Generalized random fields on Riemannian manifolds : theory and practice

Mike Pereira

PSL
UNIVERSITÉ PARIS

THÈSE DE DOCTORAT

DE L'UNIVERSITÉ PSL

Préparée à MINES ParisTech

Champs aléatoires généralisés définis sur des variétés riemanniennes: théorie et pratique

Generalized random fields on Riemannian manifolds: theory and practice

Soutenue par

Mike PEREIRA

Le 28 Novembre 2019

École doctorale n°621

Ingénierie des Systèmes, Matériaux, Mécanique, Energétique

Spécialité

Géostatistique et probabilités appliquées

Composition du jury :

Liliane BEL Professeur, AgroParisTech	*Présidente*
Denis ALLARD Directeur de recherche, INRA PACA	*Rapporteur*
Finn LINDGREN Chair of Statistics, The University of Edinburgh	*Rapporteur*
Annika LANG Associate professor, Chalmers University of Technology	*Examinatrice*
Emilio PORCU Professor, Trinity College Dublin	*Examinateur*
Cédric MAGNERON PDG, Estimages	*Invité*
Nicolas DESASSIS Chargé de recherche, MINES ParisTech	*Co-encadrant*
Hans WACKERNAGEL Directeur de recherche, MINES ParisTech	*Directeur de thèse*

MINES ParisTech | PSL

Generalized random fields on Riemannian manifolds: theory and practice.

Mike PEREIRA

PhD thesis prepared under the supervision of
Hans Wackernagel, Nicolas Desassis
and Cédric Magneron.

MINES ParisTech, PSL University
Paris, France
November 28, 2019

Acknowledgement

This PhD thesis results from a CIFRE contract that linked the French national agency for research and technology and ESTIMAGES, a French company providing geostatistical solutions for seismic noise attenuation and velocity modeling to Oil & Gas companies. As such, I would like to thank both parties for financing this project and making it possible.

I would also like to thank my PhD supervisors, Hans Wackernagel, Nicolas Desassis and Cédric Magneron, for their support throughout this period and for guiding me through this adventure. Thanks also to Jesús Angulo and Patrice Bertail for examining my work during the annual qualification meetings and giving me many fruitful pieces of advice.

Finally, I would like to thank the members of my defense committee, Liliane Bel, Denis Allard, Finn Lindgren, Annika Lang and Emilio Porcu, for taking the time to review my work and refining it with their advice, comments and thoughts. I am very honored to have my work associated with your names.

Remerciements

Ces trois années de thèse ont été riches en émotions, en casse-têtes mathématiques, en fous rires et en moments de remise en question. En regardant maintenant le travail accompli, je me dois de remercier toutes les personnes qui m'ont soutenu durant cette aventure qu'à été la thèse. Je vous dédicace à tous ce manuscrit.

Commençons par l'équipe de choc qui m'a encadré: Hans Wackernagel, mon directeur de thèse, Nicolas Desassis, mon encadrant "académique" et enfin Cédric Magneron, mon encadrant "entreprise". Hans, tu as été mon prof d'option avant d'être mon directeur de thèse et en somme j'ai eu la chance de t'avoir à mes côtés pendant toute mon aventure géostatisticienne. Je te remercie pour ta bienveillance et tes conseils avisés, que ce soit autour d'un ti'punch en Guyane ou au détour d'une visite bellifontaine pendant cette thèse. Cédric, je te remercie de la confiance sans faille que tu m'as toujours accordé. J'ai beaucoup aimé faire bouger les choses et être aux avant-gardes avec toi, que ce soit en travaillant sur des casse-têtes techniques avec Estimages ou en s'embarquant dans l'aventure "start-up" avec Showhere. Nicolas, enfin, que dire de quelqu'un qui est devenu plus qu'un encadrant, mais un véritable ami. Merci d'être toi! J'ai adoré faire de la recherche sous ton aile: que ce soit les moments passés à faire des maths au tableau, le plaisir de partager nos dernières trouvailles, l'entraide dans la bonne humeur, les cafés pour parler de tout et de rien. Tu as fais de moi le chercheur en herbe que je suis, et je t'en serai toujours reconnaissant. Dans la même veine, je me dois de mentionner le barbu qui m'a convaincu de faire une thèse: mon très cher ami Ricardo. Je me souviendrai toujours des heures passées devant un tableau noir à répondre à des questions que personne ne se pose (mais qui n'en sont pas moins importantes!), des yeux levés au ciel lorsque de mauvais mots étaient prononcés, des moments passés à refaire le monde autour d'une bière ou au Manhattan. Merci de m'avoir entrainé dans ce monde, amigo!

J'ai eu la chance de préparer ma thèse dans un environnement hors du commun, au sein d'une équipe bienveillante composée de chercheurs toujours prêts à aider les jeunes pousses. Merci l'équipe de Géostatistique des Mines! En particulier, merci à toi Lantu pour tous ces moments passés à se gratter la tête devant ton tableau noir, et bien sûr pour ta bonne humeur et ton aide précieuse. Je suis honoré d'avoir pu travailler avec toi! Merci également à Xavier, mon ex-collègue de bureau et maintenant chef d'équipe, qui malgré toutes ses responsabilités a toujours pris le temps de s'intéresser de près à mon travail. Merci infiniment pour ton travail de relecture de ma thèse et pour ton soutien à toute épreuve! Merci à Thomas et Emilie, auprès de qui j'ai toujours pu trouver une oreille attentive, que ce soit autour d'un verre ou au détour d'un cours donné ensemble. Didier, merci pour toutes les bonnes marrades, en Guyane comme à Fontainebleau (et jusqu'à Oyo?), et merci d'avoir toujours été à l'écoute de mes petits problèmes de code! Merci à toi Chantal pour ta bienveillance et ton intérêt constant pour mon travail. Merci également à Fabien, pour les bonnes barres et les pintes au pub! Je n'oublie bien sûr pas Jacques, Serge, Gaëlle et Cachou avec qui j'ai toujours pu échanger. Et un énorme merci à Nathalie (et en son temps Isabelle) qui ont su faire du bâtiment B un endroit où il fait bon travailler. Enfin, je souhaterais remercier un ancien de la maison, Denis Allard, pour avoir toujours pris le temps de m'éclairer par ses conseils et de partager avec moi son expérience.

Cette thèse, j'ai eu la chance de la préparer entouré d'un gang de doctorants incroyable qui m'ont toujours soutenu et avec qui ça a été un plaisir de passer du temps. À commencer par la

GéoFrat', les Matherons 2.0, les Millenials du vario, j'ai nommé: les doctorants (et pour certains jeunes docteurs) Géostats. Un groupe soudé composé de personnes adorables, avec le cœur sur la main, et qui va extrêmement me manquer. Marine, attentionnée et bienveillante. Laure, pleine d'énergie et toujours motivée. Léa, la battante et compagne de transilien. Jean, le chasseur au grand coeur. Alan, le catcheur de la WWE chilienne. Matthieu, le co-bureau le plus discret du monde (ahem...). Luc, avec qui on se tape de bonnes barres. Jihane, fan de Cho Cho Cho Chocolat. Anna, discrète mais toujours souriante. Merci d'avoir été des rayons de soleil à chacun de mes passages bellifontains. Je n'en oublie pas pour autant les doctorants des autres équipes et autres centres de l'école, avec qui j'ai de tendres souvenirs au Glasgow, en soirée jeux ou tout simplement à la cafet au détour d'une partie de tarot. Merci à vous, Aurélien, Angélique, Nicolas, Léonardo, Eric, Seb, Robin, Amin, Flavine, JB, Albane,... Je vous kiffe tous!

J'ai pu également bénéficier durant cette thèse du soutien de l'équipe d'Estimages! Merci à vous, chevaliers de l'ordre de KTools! J'ai été très heureux de faire partie de la Dev'team. Jean, merci de m'avoir toujours soutenu dans mon travail mais aussi pour nos conversations enflammées sur la musique ou les comics. Fred, j'ai beaucoup apprécié nos déjeuners passés à se vanner ou à parler Insta et autres trucs de "djeun's". Merci d'avoir été un chef imperturbable, taquin mais surtout loyal et droit dans ses bottes. J'attends avec impatience un repas de retrouvailles de la team au Chicco Burger! Merci également à Thibaut d'avoir toujours eu très à cœur le bien-être de tous au bureau. Merci enfin à tous les membres de la Prod, qui ont toujours été ouverts à partager leur savoir avec moi, et pour la bonne ambiance au bureau. Thomas, Natalia, Mathieu, Marion, Laura(s), Lara, Victoria, Leïla, Ivan, merci pour les bons moments. Enfin, je souhaterais remercier notre voisin de Lourcine, Edouard, pour toutes nos conversations mathématiques passionnantes et son soutien pendant les heures sombres de ma rédaction.

Je souhaite également remercier tous les amis que j'ai pu me faire en pratiquant le rugby en club pendant ces trois dernières années, que ce soit à Vincennes ou plus récemment chez les Pêchus. Merci pour les matchs gagnés à la dure, les entrainements par $-1000°$C, les 3èmes mi-temps chargées en pintes et plus généralement le plaisir de partager ensemble la pratique de ce sport. Merci en particulier aux héros de la défunte équipe 3 du RCV (maintenant 2bis si je ne me trompe pas) et ses coachs: le haut en COUleurs Dave et la star du slow motion Frei. Merci pour l'accueil, et l'esprit cloub qui régnait. Une pensée particulière à mes camarades première ligne László, Footix, Enguerrand, Bubu et Tom avec qui ça été un plaisir de souffrir en mêlée. Merci également au gang des scientifiques, Maxime, Ben, Ronan et Alexis avec qui il faisait autant plaisir de parler sciences que rugby.

Merci également à tous mes potes des Mines avec qui j'ai pu partager tant de bons souvenirs, autant avant que pendant cette thèse. Merci de votre soutien et d'avoir toujours été là pour me changer les idées. Je commence par mon coloc, et frère de coeur François. Merci pour toutes les soirées passées autour d'une pinte à refaire le monde entre gauchiasses! Et plus généralement, merci d'avoir toujours eu les bons mots et les bons conseils lorsque j'en ai eu besoin. Un grand merci à mon start-upper et co-amateur de Gladines préféré JayBee pour son soutien à toute épreuve, ainsi qu'au reste de la clique d'Ashtarak, Thibaut, Philippe et Clément L pour les souvenirs impérissables d'Arménie (#PiedDansLeFrigo). Merci également à Jean et ses auriolettes, à Oksana, véritable ange, à Farah et Olivier pour les soirées endiablées où ça partait, à mes potos Dynas pour les rires et les bons moments, à Céline pour les jeux débiles, à mon frère de couleur Clément M, au traitre des Géostats Simon et à Lotfi pour les petits cafés détente. J'ai également une pensée pour les pellots de Nantua City Beach, Anaïs et Flo, qui ont toujours été là pour moi.

Pour conclure cette tirade de remerciements, aux allures de discours aux Oscars, je souhaite remercier ma famille qui a été mon roc depuis ma naissance. Merci à toi Maman de m'avoir inculqué des valeurs que je porte haut et fort et pour ton amour et ton soutien inconditionnels, même lorsque je choisis de m'éloigner de toi. Merci à mon grand frère, Steven, mon premier défenseur et fan, avec qui j'ai toujours plaisir à passer du temps. Merci à mon petit frère Félix, petite teigne devenu homme, pour la confiance et le respect que tu me portes. Merci à vous pour les fous rires, les bons moments, le soutien et l'amour.

Contents

Introduction

Context

Geostatistics is the branch of statistics attached to model spatial phenomena through probabilistic models. Such phenomena are generally observed through measurements of their effects across a spatial domain. Within the geostatistical paradigm, we assume that the spatial phenomenon is described by a random field, that is a function that maps the points of the spatial domain to random variables. The actual reality of the phenomenon is then considered to be a particular realization of this random field, and the measurements are seen as evaluations of the realization at the same locations. The premise is then to use the statistical properties of the random field, somehow estimated from the measurements, to deduce information about the underlying phenomenon.

In many cases, one can only assume that a single realization of the phenomenon/random field is observed. Some assumptions are therefore made so that properties observed on this single realization can be generalized to describe the statistical properties of the random field. The most common one is assuming that the random field is Gaussian (Diggle et al., 1998), so that it is sufficient to only characterize its first two moments:

- its mean function, which corresponds to the expectation of the random field at each point of the domain;
- its covariance function, which corresponds to the function mapping a pair of locations on the domain to the covariance of the random field at these points.

Three recurring objectives then occur when dealing with spatial data: the inference of the parameters characterizing the mean and the covariance of a random field, the simulation of a random field, and the estimation of a random field from a set of observations. Many methods designed to perform these tasks require to build a covariance matrix between a given set of points of the domain (Chilès and Delfiner, 2012; Diggle et al., 1998; Wackernagel, 2013). We provide some examples. On one hand, the inference of the parameters characterizing a Gaussian field using a likelihood-based approach involves covariance matrices at the observed locations. On the other hand, simulations of Gaussian fields on a set of locations of a domain can be performed using the Cholesky factorization of the covariance matrix at these locations. Finally, the estimation of a Gaussian field from its partial observation, using a kriging approach, requires to invert the covariance matrix at the observed locations . Hence it is crucial to be able to properly define theses covariance matrices and to be able to work with them.

Defining the covariance matrices

The nice particular case of stationary models

To facilitate the construction of the covariance matrices, it is fairly common to consider that the random field is isotropic and second-order stationary, whenever the data lie in a nice, continuous "chunk" of space. Within this assumption, which we simply call stationarity, the possible mean

and covariance functions of the random field are simplified. On one hand, the mean function is constant over the domain. On the other hand, the covariance function is a radial function, meaning that the covariance between a pair of points will *only* depend on the (Euclidean) distance separating them.

In this context, the mean is usually estimated as the mean of the observed values and the covariance function is estimated from the data points using variogram modeling or likelihood-based approaches (Diggle et al., 1998; Wackernagel, 2013). Then, computing the covariance matrices mentioned earlier simply comes down to apply the radial covariance function to the entries of a distance matrix.

Unfortunately, as one may suspect, stationarity is a strong assumption that cannot be applied to model any spatial dataset (Fouedjio, 2017). Dealing for instance with data lying in non-Euclidean spaces, or with data for which the highly regular spatial structure implied by the stationary assumption does not apply, requires more work.

Modeling the non-stationary covariance

In the non-stationary case, the covariance function can no longer be expressed as a simple function of the distance between the points, but has an expression that depends on the location and relative position of the considered pair of points. However, we assume here that some prior structural information on the behavior of the random field across the domain is available. Namely, we assume that the random field shows *local anisotropies*. Then, around each point of the domain, there is a preferential direction along which the range of highly correlated values is maximal, whereas it is minimal in the cross-direction(s). In particular, the angles defining the preferential directions are called anisotropy angles and the size of the ranges are called anisotropy ranges.

A first challenge is to determine the expression of this covariance function from the observed data, which is tackled by imposing that the random field can be modeled in a certain way. Ideally, these models would allow to incorporate the prior structural information as it is directly linked to the definition of the covariance function.

The usual methods to model the corresponding non-stationary random fields all aim at deriving an expression of the covariance function for any pairs of points in the domain. A large review of the methods used to model non-stationary random fields was done by Fouedjio (2017). We present in the following the three more popular approaches typically encountered in practice[1].

Basis function approach The basis function approach relies on the Karhunen–Loève theorem (Lindgren, 2012), which states that any Gaussian field on a bounded domain can be decomposed as a weighted sum of orthogonal (deterministic) functions, called eigenfunctions. In particular, the weights of the linear combination are independent Gaussian variables with decreasing variances. The eigenfunctions are solutions of a set of integral equations, called Fredholm equations, which involve the expression of the covariance function. Conversely, the covariance function can be expressed as a weighted sum involving these functions (Lindgren, 2012).

Without any particular assumption about the domain, the eigenfunctions are determined by discretizing and solving the Fredholm equations. In this setting, the actual expression of the covariance function is replaced by local approximations derived from the data (Huang et al., 2001). This method assumes in particular that the data is composed of several realizations of the non-stationary process to model. Solving the discretized problem then amounts to the diagonalization of a matrix, which itself becomes a real computational bottleneck when its size (or equivalently the number of data points) increases.

Space deformation A second approach to solve the modeling problem consists in considering that a non-stationary variable observed across a spatial domain can be turned into a stationary variable after applying a (non-linear) deformation to the domain. Within this space deformation

[1] In this work, we only consider non-stationary covariances defined for spatial data. New challenges appear when dealing with space-time data given that the non-stationarity can result from both anisotropies in the spatial domain (that can change over time) and the fact that the time coordinate should generally be differentiated from the space coordinates. We refer the reader to the work of Porcu et al. (2006, 2007), who proposed a method to build models able to deal with such data.

approach, the goal is then to characterize the deformation from the observed variable so that the problem can be reformulated in a stationary framework in the deformed domain (Sampson and Guttorp, 1992). This approach relies on the idea that the covariance function of the non-stationary process can be written as the composition of a stationary (isotropic) covariance model with a deformation function. Perrin and Senoussi (2000); Porcu et al. (2010) derived characterizations of the covariance functions for which this so-called (isotropic) stationary reducibility is admissible.

The multi-dimensional scaling algorithm (Kruskal, 1964) is leveraged in this context: this algorithm associates to each data point a set of coordinates in a new, "deformed" space, so that data points with similar (resp. dissimilar) values are close to (resp. distant from) each other in the deformed space. The implementation of this method usually relies on the assumption that the data set is composed of several realizations of the same random process, although alternatives to circumvent this assumption have been proposed (Anderes and Stein, 2008; Fouedjio et al., 2015). Another approach to determine the deformation consists in working with a set of parametrized deformation functions which are fitted on the data by minimizing an objective function (Anderes and Stein, 2011; Perrin and Monestiez, 1999). Both approaches reveal to be computationnally expensive, which limit their applicability for large-scale datasets.

Besides, to the best of our knowledge, these space deformation models do not allow to easily take into account prior structural information about the non-stationarity, namely local anisotropy angles and ranges. Indeed, they all seek to directly (but approximately) characterize the overall deformation while only considering the location (and the value) of the data points. This is regrettable as these parameters are supposed to be a consequence of the (assumed) deformation process and one could think that including them in the estimation of spatial deformation would simplify the problem.

Convolution model A third approach to modeling non-stationary data is the convolution model, introduced by Higdon et al. (1999). The idea is to model the value of the non-stationary field at a given point of the domain as the result of the (spatial) convolution over the domain of a deterministic function, called kernel function, with a white noise (i.e. a random process over the domain whose values at any two distinct points are independent and identically distributed). Considering different kernel functions to compute the value of the random field at different locations of the domain then naturally yields a non-stationary field.

In order to derive a closed-form for the covariance function of the resulting field, Paciorek and Schervish (2006), Pintore and Holmes (2004), Stein (2005) and Porcu et al. (2009) proposed families of kernel functions which are parametrized at each point of the domain by the local anisotropy parameters. In particular, they represent the anisotropy parameters as positive definite matrices of the form $\boldsymbol{R}\boldsymbol{D}^2\boldsymbol{R}^T$, where $\boldsymbol{R}$ is a rotation matrix defined by the anisotropy angle(s) and $\boldsymbol{D}$ is the diagonal matrix whose entries are the inverse of the anisotropy ranges.

The covariance between two points is then expressed by averaging the representation matrix at both points, which ensures in particular that the anisotropy parameters are locally respected. The downside of this expression may be that only the information of the anisotropy at both points is taken into account in their covariance, and not the overall structure of the anisotropy, which in practice might influence the covariance.

Random fields on manifolds

Dealing with non-stationarity is not sufficient. Indeed, spatial data do not always occur on nicely contiguous domains of Euclidean spaces. The simplest example might be data measured across our planet, which arise naturally in applications such environmental science, geosciences and cosmological data analysis (Marinucci and Peccati, 2011). The use of Euclidean distance to model correlations between points of a random field defined on a sphere then becomes unrealistic.

Defining and working with random fields on a sphere is an extensively studied subject. Marinucci and Peccati (2011) provided a review of the theory surrounding random fields on a sphere. In order to retrieve a framework similar to Euclidean spaces, most of the effort was attached to characterize valid covariance functions on the sphere, that would model correlation between points using the arc length distance between them (Gneiting, 2013; Huang et al., 2011).

Stationary Gaussian random fields on a sphere are usually defined through their expansion into a basis of known (deterministic) functions called spherical harmonics (Jones, 1963). In

particular, this expansion can be seen as the counterpart of the expansion arising from the Karhunen–Loève theorem, but for fields defined on a sphere. This expansion is still being exploited to derive for instance simulation methods and to characterize the covariance structure of the resulting fields (Emery and Porcu, 2019; Lang and Schwab, 2015; Lantuéjoul et al., 2019; Marinucci and Peccati, 2011). Models have also been proposed to deal with both space-time data (Porcu et al., 2016) and anisotropy (Estrade et al., 2019) on the sphere.

However, the work done for random fields on a sphere hardly generalizes to other spatial domains, as they heavily rely on the intrinsic properties of the sphere as a surface. What then can be done if our spatial data lie on an arbitrary (smooth) surface of body? An answer to this question is provided by the theory of random fields defined on manifolds.

Basically, a manifold is a set that behaves locally like a Euclidean space. This mathematical object generalizes in particular the notions of surface and arbitrary body lying in a Euclidean space. Adler and Taylor (2009) provided a review of the theory defining such fields. They mainly focused on the geometry of their excursion sets, while dealing with brain mapping problems.

Working with covariance matrices: the big n problem

Knowing how to properly define a covariance model suited for a given spatial dataset does not guarantee that we will be able to actually use it. Indeed, a second drawback arises when trying to build and then work with covariance matrices: the so-called big n problem. By definition, the covariance matrix contains $n \times n$ covariance values that should be computed and stored, where n is the number of points of interest. In practice, n may be the number of grid points on which we desire to compute a simulation, or the number of data points. Hence, n can easily become very large, and thus, building and storing the covariance matrix quickly becomes a task requiring heavy computational and storage needs.

In fact, this problem is encountered in both the stationary and the non-stationary frameworks. Numerous solutions have been proposed in the stationary case (See Sun et al. (2012) for a review). We can for instance cite the use of compactly supported covariance functions (Gneiting, 2002) and of covariance tapering (Furrer et al., 2006; Kaufman et al., 2008), which limit the number of non-zero entries in the covariance matrix. Similarly, imposing that the considered random field is Markovian ensures that the resulting precision matrix[2] has a limited number of non-zero entries (Rue and Held, 2005). The problems are then reformulated using the precision matrix instead of the covariance matrix. If some of these solutions are transferable to the non-stationary case (see for use of compactly supported non-stationary covariance models proposed by Liang and Marcotte (2016)), they usually come at the price of a restriction on the models we can consider.

The SPDE approach, a starting place

A solution to both the modeling problem and the big n problem introduced above is proposed by Lindgren et al. (2011), with their so-called stochastic partial differential equation (SPDE) approach. The SPDE approach builds on a result from Whittle (1954) which states that Gaussian random fields Z on $\mathbb{R}^d$ with a Matérn covariance function, are the stationary solutions of the SPDE given by

$$(\kappa^2 - \Delta)^{\alpha/2} Z = \tau \mathcal{W} \quad , \tag{1}$$

where $\kappa > 0$, $\alpha > d/2$, $\tau > 0$, $(\kappa^2 - \Delta)^{\alpha/2}$ is a pseudo-differential operator (which can be seen as a generalization of the Laplacian operator and is defined using the Fourier transform) and $\mathcal{W}$ is a Gaussian white noise. In particular, for $\alpha = 2$, SPDE (1) rewrites $\kappa^2 Z - \Delta Z = \tau \mathcal{W}$ where Δ corresponds to the usual Laplacian operator.

In their approach, Lindgren et al. (2011) characterize Matérn fields as solutions of SPDE (1) rather than using their covariance function. They propose to formulate a solution for this SPDE using the finite element method: hence, the solution is expressed as a linear combination of a finite set of (user-defined) interpolation functions defined across the domain, weighted by correlated Gaussian weights. They actually provide a closed form for the precision matrix of these weights, in the case where $\alpha \in \mathbb{N}$. The precision matrix is then given as a low-degree matrix polynomial of a sparse matrix. This means in particular that solving the SPDE using this method actually yields Markovian solutions.

[2] i.e. the inverse of the covariance matrix

This approach sparked a lot of interest for several reasons. On one hand, Matérn fields are widely used in applications of geostatistical models given its ability to fit various degrees of regularity of the data with the same function by playing with a single parameter (Stein, 2012). On the other hand, the precision matrix of the weights obtained by the SPDE approach being sparse, it provides a practical solution to the big n problem when using this flexible covariance model.

Lindgren et al. (2011) and then Fuglstad et al. (2015) offer to tinker with SPDE (1) in order to provide a practical answer to the two modeling problems raised above, in the case $\alpha = 2$. In particular, their solutions conserve the desirable property that the precision matrix is sparse, and therefore the computational gains associated with it.

- *Regarding non-stationary models.* They first propose to work with spatially varying parameters κ and τ in SPDE (1), which then creates globally non-stationary fields with a locally isotropic covariance.
  A second approach they suggest is inspired by the space deformation model presented earlier, and consists in defining SPDE (1) in the deformed space. Rewriting the SPDE in the original domain using a change of variable then yields an expression of the SPDE that is locally parametrized by the Jacobian of the deformation process, or equivalently by local angles and ranges of anisotropy.

- *Regarding models on general spatial domains.* Building from the approach of Adler and Taylor (2009), they propose to define SPDE (1) directly on the general domain by seeing it as a manifold. In particular, this amounts to replace the Laplacian operator by its generalization to manifolds, called the Laplace–Beltrami operator (Lee, 2012). The resulting solution is still what is meant by a Matérn field, and is directly defined on the manifold.

Thesis statement

The starting point of our work is a simple question: can we go a little further with the solutions proposed by Lindgren et al. (2011) and Fuglstad et al. (2015)? Precisely can we design an approach

- to model both non-stationary fields from local anisotropy information and fields defined on manifolds;

- that works with a larger class of covariance functions than Matérn covariance functions;

- and that can be applied to large datasets?

As it turns out, the answer is yes, and was actually suggested by these authors in their papers. It relies on the notion of Riemannian manifold.

A *Riemannian manifold* is the association of a manifold with a locally defined metric. This metric is an application that defines around each point of the manifold a notion of length and of angles for infinitely small vectors that would be attached to that point. Hence the metric can be interpreted as an application that locally redefines the geometry of the manifold, and as such, can be seen as describing a local deformation of the manifold at each one of its points. Riemannian manifolds then seem particularly adapted to our problem, as the domain (i.e. the manifold) on which the data lie is defined together with a set of local anisotropies that in turn can be interpreted as resulting from local deformations[3] (i.e. the metric).

To see how the SPDE model extensions proposed by Lindgren et al. (2011) and Fuglstad et al. (2015) could be generalized, the focus is put not just on the solutions of the SPDE (1) now defined on the Riemannian manifold, but rather on the general mathematical object that can formally describe such solutions: *generalized random fields.* Generalized random fields are the "random" counterpart of generalized functions (also called distributions), which are widely used to formulate and derive the properties of solutions of partial derivative equations in the deterministic case (Gelfand and Shilov, 1964).

Then, the modeling problem is settled as follows. The definition of a class of generalized random fields on the Riemannian manifold that includes naturally the solutions of SPDE (1)

[3] namely a rotation and dilatation corresponding to the anisotropy angles and ranges

is introduced. Using the same principle as the finite element method, their approximation by a linear combination of predefined deterministic functions is derived, and an expression of the covariance matrix of the weights, comparable to the one obtained by Lindgren et al. (2011) in their particular case, is obtained. The fact that these fields are defined on a Riemannian manifold then ensures the applicability of the method for non-Euclidean domains (through the specification of the manifold) and for non-stationary fields (through the specification of the metric).

Remains the computational problem. As it turns out, the expression of the covariance matrix obtained in the previous step can be leveraged to derive scalable and memory-efficient algorithms for the simulation, prediction and inference of the corresponding weights. These algorithms rely on an interpretation of the Gaussian vectors defined from these covariance matrices as stochastic graph signals, that is random variables indexed by the vertices of a graph. Within this framework, called graph signal processing, generalizations of classical signal processing notions and tools, such as the Fourier transform, filtering and translation operators, are leveraged to efficiently process data indexed on graphs (Shuman et al., 2013).

As the theory (through the model specification) and the practice (through graph signal processing algorithms) of generalized random fields are laid out, we end with the concrete study of stationary and non-stationary spatial data. In particular, the simulation, the mapping, the filtering and the inference of both synthetic and real data are performed to illustrate both the flexibility and the applicability of the concepts introduced through the work.

Outline and main contributions

The dissertation is composed of two parts, reflecting the two main components of this work.

Part I aims at introducing the graph signal processing framework, as well as the algorithms that will later be used to study spatial data. In particular, we derive methods aiming at simulating stochastic graph signals, estimating their value when they are partially observed and inferring their statistical properties.

We start by setting up the mathematical framework and the main notions necessary to work with both deterministic and stochastic graph signals (Chapter 1). Following the usual graph signal processing approach, these notions are defined by drawing a parallel with classical signal processing, which we highlight throughout the chapter. Of particular interest is the concept of stationarity for stochastic graph signals, for which we propose a definition.

Then, Chapter 2 focuses on algorithms designed to perform (the equivalent of) filtering operations on graph signals. These operations play an essential role in the subsequent chapters, and as such, we lay out an extensive comparison between several approaches. It results in the introduction of the Chebyshev algorithm, which presents the best trade-off between computational cost and accuracy. This algorithm is actually the key element that ensures the scalability of the solutions proposed in this work. Applications of this algorithm to some practical problems are then presented.

Chapter 3 is devoted to the simulation of stationary graph signals. An algorithm based on Chebyshev filtering is proposed. Similar algorithms were already introduced in the literature (Hammond et al., 2011; Higham, 2008; Susnjara et al., 2015). However, we provide a study of the statistical properties of the output of this algorithm and use it to derive actionable criteria to set up its parameters. Finally, we propose a description of the algorithm in the wider framework of Krylov subspaces.

Chapter 4 then tackles the estimation of a stationary stochastic graph signal from its partial and noisy observations. We propose to solve this problem using an approach inspired by kriging theory. Two cases are treated. The first one can be interpreted as a mapping problem whereas the second one is similar to a signal extraction problem. In both cases, we lay out practical algorithms based on Chebyshev filtering. Finally, we give a formulation of these problems in a wider optimization framework, which can inspire further developments towards their efficient resolution.

Finally, Chapter 5 aims at introducing an approach to infer the statistical properties of a stochastic graph signal from its partial and noisy observations. We derive algorithms based on Chebyshev filtering to answer this problem.

Now that the study of stochastic graph signals and their properties have been introduced,

Part II aims at deriving the approximation result that allows to reduce the study of generalized random fields defined on a Riemannian manifold to the study of a stochastic graph signal.

First, a self-sufficient review of the main concepts and results of differential and Riemannian geometry used in this work is proposed in Chapter 6. In particular, we clarify the rather intuitive interpretation of Riemannian manifolds as locally deformed spaces.

Chapter 7 aims at actually presenting our solution to the modeling problem described above. The class of generalized random fields used to extend the results of Lindgren et al. (2011) is introduced, and the approximation theorem which links them to stochastic graph signals is laid out. Chapter 8 provides an application of this result when the approximation is performed using the finite element method, and a convergence result is derived.

Finally, Chapter 9 echoes the initial problem statement and illustrates the application of the framework derived in this work to both synthetic and real data. In particular, examples of simulation, mapping, filtering and inference are presented.

Disclaimer

The work presented in this dissertation is interdisciplinary: indeed, we play with notions of graph theory, classical and graph signal processing, differential and Riemannian geometry, function approximation and generalized random fields. Consequently, this dissertation was written with the intention of providing the reader with as much understanding of these subjects as needed to derive the results that are presented.

Hence, some parts of the dissertation can easily be skipped by more experienced readers. In Chapter 1, Section 1.1 consists only in basic reminders of graph theory, and Section 1.2 of reminders of classical deterministic and stochastic signal processing. Readers familiar with graph signal processing can skip Section 1.3. Readers familiar with differential and Riemannian geometry can skip Chapter 6. Finally, readers familiar with the finite element method can skip Section 8.1.

Notations

$\boldsymbol{M}^H$	Conjugate-Transpose of a matrix $\boldsymbol{M}$
$\mathrm{Cov}[\cdot,\cdot]$	Covariance between two random variables or covariance matrix between two random vectors
$\lvert\boldsymbol{M}\rvert$ or $\det \boldsymbol{M}$	Determinant of a matrix $\boldsymbol{M}$
$\mathrm{Diag}(\boldsymbol{v})$	Diagonal matrix whose entries are the entries of the vector $\boldsymbol{v}$
$\mathrm{DCT}[\cdot]$	Discrete cosine transform of a vector
$\mathrm{DFT}[\cdot]$	Discrete Fourier transform of a vector
$\lVert\cdot\rVert_2$	Euclidean norm of a vector
$\mathbb{E}[\cdot]$	Expectation of a random variable or vector
$\mathscr{F}[\cdot]$	Fourier transform of a signal
GRF	Gaussian Random Field
GRFLA	Gaussian Random Field with Local Anisotropies
GeRF	Generalized Random Field
$\mathrm{GFT}[\cdot]$	Graph Fourier transform of a graph signal
$\mathbb{1}_A$	Indicator function of a set A
$[\![\cdot,\cdot]\!]$	Interval of all integers between two integers
$\mathrm{span}\{\boldsymbol{v}_1,\ldots,\boldsymbol{v}_n\}$	Linear span of a set of $n \geq 1$ vectors $\boldsymbol{v}_1,\ldots,\boldsymbol{v}_n$
$\mathbb{P}[\cdot]$	Probability of an event
$\mathscr{M}_{p,q}(\mathbb{R})$	Set of matrices with p rows and q columns, and with real coefficients
$\mathscr{M}_n(\mathbb{R})$	Set of square matrices of size n with real coefficients
SGS	Stochastic Graph Signal
$\mathrm{supp}(\cdot)$	Support of a function
$\mathrm{Trace}(\cdot)$	Trace of a matrix
$\boldsymbol{M}^T$	Transpose of a matrix $\boldsymbol{M}$
$\mathrm{Var}[\cdot]$	Variance of a random variable or covariance matrix of a vector

List of Figures

Part I

Stochastic graph signals

1

Deterministic and stochastic graph signal processing

Contents

Résumé

Dans ce chapitre, nous introduisons un cadre mathématique minimal permettant d'étudier le traitement de signaux déterministes et stochastiques définis sur des graphes. Nous commençons par introduire les principales notions de théorie des graphes et de traitement du signal (au sens classique du terme) nécessaires à la construction de la théorie entourant le traitement du signal sur graphe. Nous présentons ensuite cette dernière en suivant la même approche que celle présente dans la littérature associée.

Introduction

Graphs are structures aiming at representing complex data as a set of objects, called vertices, and pairwise relationships between them, the edges (Bondy and Murty, 1976). These relationships usually encode a notion of similarity between the objects they connect. This type of data structure arises in applications such as social, energy, transportation and neural networks, but also biology, image processing and many more (Newman, 2010). In practice, two main scenarios arise:

- either the focus is put on the structure of the graph itself, meaning that the graph is used to model and study pairwise relationships a predefined set of objects,
- or these relationships are assumed to be known and the focus is put on modeling and studying variables that are defined on the objects.

Graph signal processing is an emerging field focusing on developing tools to process data arising from this last scenario (Shuman et al., 2013). These data are therefore modeled as variables indexed by the vertices of a known graph, and named graph signals. The goal is then to be able to perform on these graph signals common operations of continuous signal processing, such as filtering, denoising and completion.

Given that now the data domain is highly irregular, as it consists of a set of discrete vertices on an arbitrary graph, all these operations had to be redefined in a unified framework suited for graph data. This framework was built by generalizing classical signal processing notions and tools, like for instance the Fourier transform and translation operators, to graph signals (Girault, 2015a; Ortega et al., 2018; Shuman et al., 2013). This everlasting parallel between classical signal processing and graph signal processing is at the core of this new field.

This first chapter aims at introducing a minimal mathematical framework for deterministic and stochastic graph signal processing. In the first two sections, the main notions of graph theory and continuous and discrete signal processing useful to build this framework are introduced. Then, in the subsequent sections, the graph signal processing framework is introduced following the same approach as the one used by Shuman et al. (2013), Girault (2015a) and Marques et al. (2017).

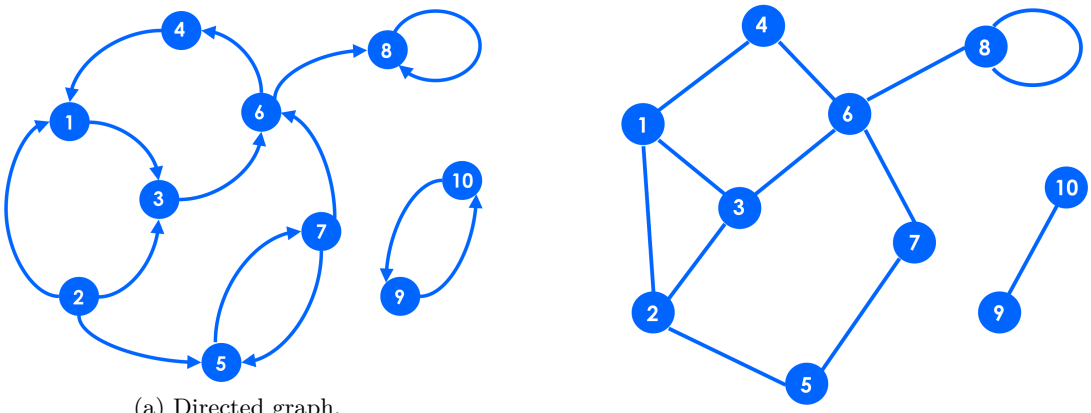
1.1 Mathematical framework for graphs

In this section we review some basic definitions and properties concerning the study of graphs. We refer the reader to (Newman, 2010, Chapter 6) for a more complete overview of the mathematical framework used in graph theory.

1.1.1 Definitions and notations

A (directed) *graph* $\mathcal{G}$ is a structure amounting to a set of objects and pairwise relationships between them. Formally it consists in a set of *vertices* $\mathcal{V}$ representing the objects and a set of *edges* $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ that represents pairwise relationships as pairs of vertices. A *subgraph* $\mathcal{H}$ of $\mathcal{G}$ is a graph whose vertex set $\mathcal{V}'$ is a subset of $\mathcal{V}$ and whose edge set is a subset of $\mathcal{E} \cap (\mathcal{V}' \times \mathcal{V}')$.

In this work, only *finite* graphs, i.e. with a finite number of vertices n, are considered. In this case, the set vertices $\mathcal{V}$ can be identified with the set of integers $[\![1, n]\!]$ and therefore vertices can be represented as integers $i \in [\![1, n]\!]$. A graph with n vertices will also be called a n-graph. They can be represented as in Figure 1.1a: each circle corresponds to a vertex and an arrow from a vertex i to a vertex j is drawn whenever $(i, j) \in \mathcal{E}$.



(a) Directed graph.

(b) Undirected graph.

Figure 1.1: Representation of a directed and an undirected graph.

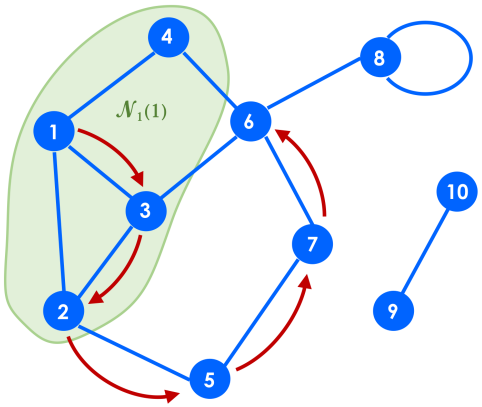


Figure 1.2: Representation of the neighborhood of order 1 of a vertex (in green) and of a path (in red). The neighborhood and the path start from vertex 1.

A *weighted* graph is a graph for which a weight (i.e. a real value) is associated to each one of its edges. The function $\mathcal{W} : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ that assigns to each pair of vertices (i, j) its weight $\mathcal{W}(i, j)$ if $(i, j) \in \mathcal{E}$ and 0 otherwise is called *weight function*. A weighted graph is therefore characterized by the triplet $(\mathcal{V}, \mathcal{E}, \mathcal{W})$. By convention, graphs with no weights are identified with weighted graphs for which all edges have a weight equal to 1.

A graph is called *undirected* (or *symmetric*) if for any pair of vertices $(i, j) \in \mathcal{V} \times \mathcal{V}$, we have $(i, j) \in \mathcal{E} \Rightarrow (j, i) \in \mathcal{E}$, and $\mathcal{W}(i, j) = \mathcal{W}(j, i)$. In this case, whenever there is an an edge between two vertices i and j, these vertices are called *adjacent* (or *connected*) and denoted $i \sim j$. Undirected graphs can be represented as in Figure 1.1b: each circle corresponds to a vertex and a straight line between a vertex i and a vertex j is drawn whenever $i \sim j$.

A *loop* is an edge between a vertex and itself. A *multi-edge* is a set of two or more edges that connect the same pair of vertices. A graph in which there are neither loops nor multi-edges is called a *simple* graph.

A *path* on an (undirected) graph $\mathcal{G}$ from a vertex i_0 to a vertex i_p is a sequence of $p + 1 \geq 1$ vertices $i_0, \ldots, i_p$ of $\mathcal{G}$ such that $\forall k \in [\![0, p - 1]\!]$, i_k and i_{k+1} are adjacent. p is called the *length* of the path. In particular, paths of length 0 correspond to the vertices of the graph and paths of length 1 correspond to its edges.

The *neighborhood* of order $k \in \mathbb{N}$ of a vertex i is the set of all vertices j such that there exists a path of length at most k from i to j. It is denoted $\mathcal{N}_k(i)$. Any $j \in \mathcal{N}_k(i)$ is called a *neighbor* (of order k) of i. Basically, any vertex in a neighborhood of order k of a vertex i can be reached from i with at most k "hops" along the edges of the graph. Figure 1.2 illustrates the neighborhood of order 1 of a given vertex (in green) of an undirected simple graph and an example of path of length 5 (in red).

An undirected graph $\mathcal{G}$ is *connected* if there exists a path between any pair of its vertices. More generally, a *connected component* of $\mathcal{G}$ is a connected subgraph $\mathcal{H}$ of $\mathcal{G}$ formed by vertices that have no neighbor other than those present in $\mathcal{H}$. It is easy to check that any graph is the disjoint union of its connected components, where the union between two graphs $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1, \mathcal{W}_1)$ and $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2, \mathcal{W}_2)$ with disjoint node sets and edge sets, is the graph $\mathcal{G}_1 \cup \mathcal{G}_2$ defined by $\mathcal{G}_1 \cup \mathcal{G}_2 = (\mathcal{V}_1 \cup \mathcal{V}_2, \mathcal{E}_1 \cup \mathcal{E}_2, \mathcal{W}_{12})$ where $\mathcal{W}_{12}$ is defined so that its restriction to edges of $\mathcal{E}_1$ (resp. $\mathcal{E}_2$) is $\mathcal{W}_1$ (resp. $\mathcal{W}_2$).

Two n-graphs $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1, \mathcal{W}_1)$ and $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2, \mathcal{W}_2)$ are *isomorphic* if there exits an edge-preserving bijection π between $\mathcal{V}_1$ and $\mathcal{V}_2$ i.e. π is a bijection from $\mathcal{V}_1$ to $\mathcal{V}_2$ such that:

$$\forall i_1, j_1 \in \mathcal{V}_1, \quad \begin{cases} i_1 \sim j_1 \Leftrightarrow \pi(i_1) \sim \pi(j_1) \\ \text{and} \\ \mathcal{W}_1(i_1, j_1) = \mathcal{W}_2(\pi(i_1), \pi(j_1)) \end{cases} .$$

Thus, two isomorphic graphs have the same "structure", meaning that they link their vertices in the same way. In particular, if $\mathcal{G}_1$ and $\mathcal{G}_2$ are two subgraphs of a graph $\mathcal{G}$ then them being isomorphic means a same layout of edges is observed at two parts of $\mathcal{G}$, thus implying that the structure they create is repeated at two different locations in $\mathcal{G}$.

Assumption 1.1. *In this work, only connected simple undirected finite graphs are considered.*

1.1.2 Matrix representations of graphs

In this section $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ denotes a graph with n vertices defined according to Assumption 1.1. Several n-matrices encompassing information on the structure of $\mathcal{G}$ are now introduced.

Adjacency matrix

Given that $\mathcal{G}$ is simple and undirected, for any pair of its vertices (i, j), there exists at most one edge between them. The *adjacency matrix* $\boldsymbol{\mathcal{W}}$ of $\mathcal{G}$ is defined as the $n \times n$ symmetric matrix whose entry $\mathcal{W}_{ij}$ is equal to the weight of the edge (i, j) if it exits, and is zero otherwise:

$$\mathcal{W}_{ij} = \begin{cases} \mathcal{W}(i, j) & \text{if } i \sim j \\ 0 & \text{otherwise} \end{cases} .$$

Hence, the adjacency matrix summarizes all the relevant information about the graph structure: the non-zero entries indicate the existence of an edge between two vertices and its weight. Besides, if $\mathcal{G}$ is composed of several connected components, then its adjacency matrix $\boldsymbol{\mathcal{W}}$ can be represented by a block matrix, where each block is the adjacency matrix of one of the connected components. Indeed, the presence of a non-zero entry outside these blocks would imply that there is an edge connecting two distinct connected components, which is impossible.

Remark 1.1.1. The fact that $\mathcal{G}$ is undirected gives that $\boldsymbol{\mathcal{W}}$ is symmetric, and the fact that it is simple ensures that the diagonal entries of $\boldsymbol{\mathcal{W}}$ are zero.

Getting back to the notion of graph isomorphism, the following result provides a link between isomorphic graphs and their adjacency matrices.

Proposition 1.1.1. *Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be two isomorphic n-graphs with adjacency matrices $\boldsymbol{\mathcal{W}}_1$ and $\boldsymbol{\mathcal{W}}_2$. Then, there exists a permutation π of $[\![1, n]\!]$ such that*

$$\boldsymbol{\mathcal{W}}_1 = \boldsymbol{P}_\pi^{-1} \boldsymbol{\mathcal{W}}_2 \boldsymbol{P}_\pi \quad ,$$

where $\boldsymbol{P}_\pi$ is the permutation matrix defined by $[\boldsymbol{P}_\pi]_{ij} = \delta_{i\pi(j)}$. In other words,

$$\forall i, j \in [\![1, n]\!], \quad [\boldsymbol{\mathcal{W}}_1]_{ij} = [\boldsymbol{\mathcal{W}}_2]_{\pi(i)\pi(j)} \quad . \tag{1.1}$$

Proof. This result is a direct consequence of the definition of isomorphic graphs. Identifying the sets of vertices of $\mathcal{G}_1$ and $\mathcal{G}_2$ with $[\![1, n]\!]$, the bijection between them defines a permutation that satisfies Equation (1.1). □

In the particular case where $\mathcal{G}_1$ and $\mathcal{G}_2$ are subgraphs of the same graph $\mathcal{G}$, the result hereafter follows.

Corollary 1.1.2. *Let $\mathcal{G}$ be a n-graph with adjacency matrix $\boldsymbol{\mathcal{W}}$ and vertex set $\mathcal{V}$.*
Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be two isomorphic subgraphs of $\mathcal{G}$, with vertex sets $\mathcal{V}_1 \subset \mathcal{V}$ and $\mathcal{V}_2 \subset \mathcal{V}$.
Then there exists a permutation π of $[\![1, n]\!]$ such that

$$\forall i, j \in \mathcal{V}_1, \quad \mathcal{W}_{ij} = [\boldsymbol{\mathcal{W}}]_{\pi(i)\pi(j)} \quad . \tag{1.2}$$

Proof. Following the definition of isomorphic graphs, consider b to be the bijection that sends $\mathcal{V}_1$ to $\mathcal{V}_2$. Then any permutation π of $[\![1, n]\!]$ such that $\forall i \in \mathcal{V}_1$, $\pi(i) = b(i)$ satisfies Equation (1.2). □

Remark 1.1.2. Both Proposition 1.1.1 and Corollary 1.1.2 are applicable to isomorphic subgraphs of a graph $\mathcal{G}$. The difference is that in the latter case, the equation is satisfied by the adjacency matrix $\boldsymbol{\mathcal{W}}$ of the graph containing $\mathcal{G}_1$ and $\mathcal{G}_2$, whereas in the former case, it involves the adjacency matrices of both subgraphs (which corresponds to sub-matrices of $\boldsymbol{\mathcal{W}}$).

Degree matrix

The *degree* d_i of a vertex $i \in \mathcal{V}$ is defined as the sum of the weights of the edges to which it is an endpoint. Hence, the degree of any vertex i can be computed from the adjacency matrix $\boldsymbol{\mathcal{W}}$ of $\mathcal{G}$ using the fact that

$$d_i = \sum_{\substack{j=1 \\ j \sim i}}^{n} \mathcal{W}(i, j) = \sum_{j=1}^{n} \mathcal{W}_{ij} = [\boldsymbol{\mathcal{W}} \mathbf{1}_n]_i \quad ,$$

where $\mathbf{1}_n$ is the n-vector of ones. Note that in the particular case were all edge weights are equal to 1, d_i is equal to the number of neighbors of order 1 of i.

The *degree matrix* $\boldsymbol{D}$ of $\mathcal{G}$ is then defined as the $n \times n$ diagonal matrix whose (diagonal) entries are the degrees of each vertex of the graph:

$$\boldsymbol{D} = \begin{pmatrix} d_1 & & \\ & \ddots & \\ & & d_n \end{pmatrix} = \mathrm{Diag}(\boldsymbol{\mathcal{W}} \mathbf{1}_{\boldsymbol{n}}) \quad .$$

Laplacian matrix

The *Laplacian matrix* (or *graph Laplacian*) $\boldsymbol{L}$ of $\mathcal{G}$ is a $n \times n$ matrix defined from its adjacency and degree matrices as

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{\mathcal{W}} \quad .$$

From its definition, the Laplacian matrix enjoys several interesting properties.

Proposition 1.1.3. *Let $\boldsymbol{L}$ be the Laplacian matrix of a simple undirected n-graph $\mathcal{G}$ with adjacency matrix $\boldsymbol{\mathcal{W}}$.*

1. *$\boldsymbol{L}$ is symmetric. Consequently, $\boldsymbol{L}$ is diagonalizable in a real orthonormal basis, and its eigenvalues are real.*

2. *The Hermitian form associated with* $\boldsymbol{L}$ *satisfies:*

$$\forall \boldsymbol{u} \in \mathbb{C}^n, \quad \boldsymbol{u}^H \boldsymbol{L} \boldsymbol{u} = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \mathcal{W}_{ij} |u_i - u_j|^2 \quad .$$

where $\boldsymbol{u}^H$ *is defined as* $\boldsymbol{u}^H := \bar{\boldsymbol{u}}^T$.

3. 0 *is an eigenvalue of* $\boldsymbol{L}$.

Proof.

1. Direct consequence of the the fact that $\boldsymbol{\mathcal{W}}$ is symmetric.
2. Let $\boldsymbol{u} \in \mathbb{C}^n$. Then, $\boldsymbol{u}^H \boldsymbol{L} \boldsymbol{u} = \boldsymbol{u}^H \boldsymbol{D} \boldsymbol{u} - \boldsymbol{u}^H \boldsymbol{\mathcal{W}} \boldsymbol{u} = \sum_i d_i \bar{u_i} u_i - \sum_i \sum_j \mathcal{W}_{ij} \bar{u_i} u_j = \sum_i \sum_j \mathcal{W}_{ij} (\bar{u_i} u_i - \bar{u_i} u_j)$. Hence, given that the indexes i and j play symmetric roles, $2\boldsymbol{u}^H \boldsymbol{L} \boldsymbol{u} = \sum_i \sum_j \mathcal{W}_{ij} (\bar{u_i} u_i - \bar{u_i} u_j) + \sum_j \sum_i W_{ji} (\bar{u_j} u_j - \bar{u_j} u_i)$.
Finally, using the fact that $\boldsymbol{\mathcal{W}}$ is symmetric and by switching the two sums of the second double sum, we get $2\boldsymbol{u}^H \boldsymbol{L} \boldsymbol{u} = \sum_i \sum_j \mathcal{W}_{ij} (\bar{u_i} u_i - \bar{u_i} u_j + \bar{u_j} u_j - \bar{u_j} u_i) = \sum_i \sum_j \mathcal{W}_{ij} (u_i - u_j)(\overline{u_i - u_j}) = \sum_i \sum j \mathcal{W}_{ij} |u_i - u_j|^2$.
3. $\boldsymbol{L} \boldsymbol{1}_n = \boldsymbol{D} \boldsymbol{1}_n - \boldsymbol{\mathcal{W}} \boldsymbol{1}_n = \boldsymbol{0}_n$.

□

Of particular interest is the case where all edge weights are non-negative as it yields stronger properties for the associated Laplacian matrix.

Proposition 1.1.4. *Let* $\boldsymbol{L}$ *be the graph Laplacian of a simple undirected graph* $\mathcal{G}$ *whose weights are non-negative. Then,*

1. $\boldsymbol{L}$ *is a positive semi-definite matrix.*
2. *The dimension of the null space of* $\boldsymbol{L}$ *(or equivalently the multiplicity of its eigenvalue* 0*) is equal to the number of connected components composing* $\mathcal{G}$.
3. *The largest eigenvalue* $\lambda_{\max}$ *of* $\boldsymbol{L}$ *satisfies*

$$\lambda_{\max} \leq \max_{i \in [\![1,n]\!]} \sqrt{2(d_i^2 + \hat{d}_i)} \quad ,$$

where $\forall i \in [\![1, n]\!]$, $d_i = \sum_{k=1}^{n} \mathcal{W}_{ik}$ *and* $\hat{d}_i = \sum_{k=1}^{n} \mathcal{W}_{ik} d_k$.

Proof.

1. According to the second point of Proposition 1.1.3, the Hermitian form associated with $\boldsymbol{L}$ is now positive, thus proving the result.
2. Denote $A = \{\boldsymbol{x} \in \mathbb{C}^n : \text{ for any pair of vertices } i, j \text{ connected in } \mathcal{G}, x_i = x_j\}$ and let r be the number of connected components. A is a vector space of dimension r as it is spanned by the set of r (linearly independent) vectors that are 1 on one of the connected components of $\mathcal{G}$ and 0 elsewhere. We now prove that A is the null space of $\boldsymbol{L}$.
Indeed, if $\boldsymbol{x}$ is in the null space of $\boldsymbol{L}$, then in particular $\boldsymbol{x}^H \boldsymbol{L} \boldsymbol{x} = 0$ and therefore using Item 2 of Proposition 1.1.3, $\boldsymbol{x} \in A$. Conversely, it is straightforward to check that any vector in A is in the null space of $\boldsymbol{L}$. Therefore, the null space of $\boldsymbol{L}$ has dimension r.
3. Let $\boldsymbol{\mathcal{W}}$ be the adjacency matrix of $\mathcal{G}$. Let $(\lambda, \boldsymbol{x})$ be an eigenpair of $\boldsymbol{L}$. Then, $\boldsymbol{L} \boldsymbol{x} = \lambda \boldsymbol{x}$ and so, $\forall i \in [\![1, n]\!]$, $\lambda x_i = \sum_k \mathcal{W}_{ik} (x_i - x_k)$. Therefore, $\forall i, j \in [\![1, n]\!]$, $\lambda (x_i - x_j) = \sum_k \mathcal{W}_{ik} (x_i - x_k) - \sum_k \mathcal{W}_{jk} (x_j - x_k)$, which gives:

$$\forall i, j \in [\![1, n]\!], |\lambda| |x_i - x_j| \leq \sum_k \left(\mathcal{W}_{ik} |x_i - x_k| + \mathcal{W}_{jk} |x_j - x_k| \right) \quad .$$

Consider the couple $(i_m, j_m) = \text{argmax}_{i,j} |x_i - x_j|$. Then,

$$\begin{aligned}|\lambda|^2|x_{i_m} - x_{j_m}| &\leq |\lambda| \sum_k (\mathcal{W}_{i_m k}|x_{i_m} - x_k| + \mathcal{W}_{j_m k}|x_{j_m} - x_k|) \\ &\leq \sum_k \mathcal{W}_{i_m k} \sum_l (\mathcal{W}_{i_m l}|x_{i_m} - x_l| + \mathcal{W}_{kl}|x_k - x_l|) \\ &+ \mathcal{W}_{j_m k} \sum_{l'} (\mathcal{W}_{j_m l'}|x_{j_m} - x_{l'}| + \mathcal{W}_{kl'}|x_k - x_{l'}|) \quad .\end{aligned}$$

By dividing by $|x_{i_m} - x_{j_m}|$ (which is non-zero otherwise $\boldsymbol{x} = \mathbf{0}$) and using the fact that $|x_{i_m} - x_{j_m}| = \max_{i,j} |x_i - x_j|$, we get:

$$\begin{aligned}|\lambda|^2 &\leq \sum_k \mathcal{W}_{i_m k} \sum_l (\mathcal{W}_{i_m l} + \mathcal{W}_{kl}) + \mathcal{W}_{j_m k} \sum_{l'} (\mathcal{W}_{j_m l'} + \mathcal{W}_{kl'}) \\ &= \sum_k \mathcal{W}_{i_m k}(d_{i_m} + d_k) + \mathcal{W}_{j_m k}(d_{j_m} + d_k) = d_{i_m}^2 + \hat{d}_{i_m} + d_{j_m}^2 + \hat{d}_{j_m} \\ &\leq 2(\max_i d_i^2 + \hat{d}_i) \quad .\end{aligned}$$

Given that this result is true for any eigenvalue of $\boldsymbol{L}$, it is true for $\lambda_{\max}$, which proves the proposition.

□

Normalized Laplacian matrix

The *normalized Laplacian matrix* (or *normalized graph Laplacian*) $\tilde{\boldsymbol{L}}$ of $\mathcal{G}$ is defined for graphs with strictly positive degrees as a scaled version of its Laplacian matrix:

$$\tilde{\boldsymbol{L}} = \boldsymbol{D}^{-1/2}\boldsymbol{L}\boldsymbol{D}^{-1/2} = \boldsymbol{I}_n - \boldsymbol{D}^{-1/2}\boldsymbol{\mathcal{W}}\boldsymbol{D}^{-1/2} \quad .$$

Its entries are therefore defined by

$$\tilde{L}_{ij} = \begin{cases} 1 & \text{if } i = j \\ -\frac{\mathcal{W}_{ij}}{\sqrt{d_i d_j}} & \text{otherwise} \end{cases} \quad .$$

Proposition 1.1.5. *Let $\tilde{\boldsymbol{L}}$ be the normalized Laplacian matrix of a simple undirected graph $\mathcal{G}$ with adjacency matrix $\boldsymbol{\mathcal{W}}$.*

1. *$\tilde{\boldsymbol{L}}$ is symmetric. Consequently, $\boldsymbol{L}$ is diagonalizable in a real orthonormal basis, and its eigenvalues are real.*
2. *The Hermitian form associated with $\tilde{\boldsymbol{L}}$ satisfies*

$$\forall \boldsymbol{u} \in \mathbb{C}^n, \quad \boldsymbol{u}^H \tilde{\boldsymbol{L}} \boldsymbol{u} = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \mathcal{W}_{ij} \left| \frac{u_i}{\sqrt{d_i}} - \frac{u_j}{\sqrt{d_j}} \right|^2 \quad .$$

3. *0 is an eigenvalue of $\tilde{\boldsymbol{L}}$.*

Proof.

1. By definition of its entries.
2. Simply notice that $\boldsymbol{u}^H \tilde{\boldsymbol{L}} \boldsymbol{u} = \left(\boldsymbol{D}^{-1/2}\boldsymbol{u}\right)^H \boldsymbol{L} \left(\boldsymbol{D}^{-1/2}\boldsymbol{u}\right)$.
3. One can easily check that $\tilde{\boldsymbol{L}} \left(\sqrt{d_1}, \ldots, \sqrt{d_n}\right)^T = \mathbf{0}_n$.

□

In the particular case where all edge weights are non-negative, the normalized Laplacian enjoys the following additional properties:

Proposition 1.1.6. *Let $\tilde{\boldsymbol{L}}$ be the normalized graph Laplacian of a simple undirected graph $\mathcal{G}$ whose weights are non-negative. Then,*

1. *$\tilde{\boldsymbol{L}}$ is a positive semi-definite matrix.*
2. *The dimension of the null space of $\tilde{\boldsymbol{L}}$ (or equivalently the multiplicity of its eigenvalue 0) is equal to the number of connected components composing $\mathcal{G}$.*
3. *The largest eigenvalues $\lambda_{\max}$ of $\tilde{\boldsymbol{L}}$ satisfies*

$$\lambda_{\max} \leq 2 \quad .$$

Proof. 1. According to Proposition 1.1.5, the Hermitian form of $\boldsymbol{L}$ is now positive, thus proving the result.

2. Notice that a vector $\boldsymbol{x} \in \mathbb{C}^n$ is in the null space of $\tilde{\boldsymbol{L}}$ iif $\boldsymbol{D}^{-1/2}\boldsymbol{x}$ is in the null space of $\boldsymbol{L}$. Consequently, the null space of $\tilde{\boldsymbol{L}}$ is the image of the null space of $\boldsymbol{L}$ under the isomorphism represented by $\boldsymbol{D}^{-1/2}$. Therefore, using the rank nullity theorem, both spaces have the same dimension.

3. First, we show that the eigenvalues of $\tilde{\boldsymbol{L}}$ and those of $\boldsymbol{D}^{-1}\boldsymbol{L}$ are the same. Indeed, the eigenvalues of $\boldsymbol{D}^{-1}\boldsymbol{L}$ are the roots of its characteristic polynomial defined by $p(\lambda) = \det(\boldsymbol{D}^{-1}\boldsymbol{L} - \lambda\boldsymbol{I}_n)$. This polynomial satisfies $p(\lambda) = \det(\boldsymbol{D}^{-1/2}(\tilde{\boldsymbol{L}} - \lambda\boldsymbol{I}_n)\boldsymbol{D}^{1/2}) = \det(\boldsymbol{D}^{-1/2})\det(\tilde{\boldsymbol{L}} - \lambda\boldsymbol{I}_n)\det(\boldsymbol{D}^{1/2}) = \det(\tilde{\boldsymbol{L}} - \lambda\boldsymbol{I}_n)$ which is the characteristic polynomial of $\tilde{\boldsymbol{L}}$, hence proving the claim.
The matrix $\boldsymbol{D}^{-1}\boldsymbol{L}$ can be seen as the Laplacian matrix of the graph whose adjacency matrix is the non-symmetric matrix $\boldsymbol{\mathcal{W}}'$ with elements are $\mathcal{W}'_{ij} = \mathcal{W}_{ij}/d_i$. Its degree matrix is then $\boldsymbol{I}_n$ (as the rows of $\boldsymbol{\mathcal{W}}'$ all sum to 1). By noticing that the proof of item 3 of Proposition 1.1.3 never uses the symmetry of $\boldsymbol{\mathcal{W}}$, the bound obtained can be extended to the non-symmetric case. In particular for $\boldsymbol{D}^{-1}\boldsymbol{L}$, this bound equals 2 (as all degrees of the corresponding graph are 1). This concludes our proof.

□

1.2 Background: Some notions of deterministic and stochastic signal processing

In this section, we turn to the second building block of the graph signal processing framework. We lay out the main notions of classical and stochastic signal processing on which we will later on build a mathematical framework for graph signal processing. In the remainder of this section $d \geq 1$ denotes an integer.

1.2.1 Harmonic analysis of continuous signals

Most of the material covered in this section is detailed in (Stein and Weiss, 1971, Chapter 1).

Signals and energy A *signal* is a function $x : \mathbb{R}^d \to \mathbb{C}$. It is called integrable if

$$\int_{\mathbb{R}^d} |x(\boldsymbol{t})| d\boldsymbol{t} < \infty \quad .$$

The *energy* $E(x)$ of a signal x is defined as the positive and possibly infinite quantity

$$E(x) = \int_{\mathbb{R}^d} |x(\boldsymbol{t})|^2 d\boldsymbol{t} \quad .$$

Signals with finite energy therefore correspond to square-integrable functions on $\mathbb{R}^d$. In the remainder of this section, only finite-energy signals are considered.

We denote $L^2(\mathbb{R}^d)$ the Hilbert space of square-integrable functions of $\mathbb{R}$ equipped with the natural inner product $\langle \cdot, \cdot \rangle_{L^2(\mathbb{R}^d)}$ defined by:

$$\forall x, y \in L^2(\mathbb{R}^d), \quad \langle x, y \rangle_{L^2(\mathbb{R}^d)} = \int_{\mathbb{R}^d} \bar{x}(\boldsymbol{t}) y(\boldsymbol{t}) d\boldsymbol{t} \quad .$$

In particular, $\forall x \in L^2(\mathbb{R}^d)$, $E(x) = \langle x, x \rangle_{L^2(\mathbb{R}^d)} = \|x\|^2_{L^2(\mathbb{R}^d)} < \infty$, where $\| \cdot \|_{L^2(\mathbb{R}^d)}$ is the norm associated with the inner product $\langle \cdot, \cdot \rangle_{L^2(\mathbb{R}^d)}$.

Fourier transform The *Fourier transform* (FT) $\mathscr{F}$ of an integrable signal x is the function $\mathscr{F}[x] : \mathbb{R}^d \to \mathbb{C}$ defined by

$$\mathscr{F}[x](\boldsymbol{\xi}) = \int_{\mathbb{R}^d} x(\boldsymbol{t}) e^{-\mathrm{i}\boldsymbol{\xi}^T \boldsymbol{t}} d\boldsymbol{t}, \quad \boldsymbol{\xi} \in \mathbb{R}^d \quad .$$

In this last equation, the variable $\boldsymbol{t}$ is referred to as belonging to the time domain whereas the variable ξ is belongs to the frequency domain.

The FT is an invertible linear operator when applied to integrable functions whose FT is itself integrable. The inverse FT $\mathscr{F}^{-1}$ is then defined for integrable functions $\tilde{x}$ on the frequency domain (which is $\mathbb{R}^d$) by

$$\mathscr{F}^{-1}[\tilde{x}](\boldsymbol{t}) = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \tilde{x}(\boldsymbol{\xi}) e^{\mathrm{i}\boldsymbol{t}^T \boldsymbol{\xi}} d\boldsymbol{\xi}, \quad \boldsymbol{t} \in \mathbb{R}^d \quad .$$

Another way of obtaining the inverse FT is by letting (Stein and Weiss, 1971, Theorem 2.4):

$$\mathscr{F}^{-1}[\tilde{x}](\boldsymbol{t}) = \mathscr{F}[\tilde{x}](-\boldsymbol{t}), \quad \boldsymbol{t} \in \mathbb{R}^d \quad . \tag{1.3}$$

Plancherel's theorem (Stein and Weiss, 1971, Theorem 2.1) states that the FT conserves the energy of an integrable signal x, i.e.

$$E(x) = \int_{\mathbb{R}^d} |x(\boldsymbol{t})|^2 d\boldsymbol{t} = \int_{\mathbb{R}^d} |\mathscr{F}[x](\boldsymbol{\xi})|^2 d\boldsymbol{\xi} = E(\mathscr{F}[x]) \quad .$$

This result is used to extend the definition of the FT to any finite energy signal x, as the limit the FT of integrable signals with finite energy converging to x. As such, the FT is a unitary operator (Stein and Weiss, 1971, Theorem 2.3) on $L^2(\mathbb{R}^d)$, meaning that

$$\forall x, y \in L^2(\mathbb{R}^d), \quad \langle x, y \rangle_{L^2(\mathbb{R}^d)} = \quad \langle \mathscr{F}[x], \mathscr{F}[y] \rangle_{L^2(\mathbb{R}^d)} \quad .$$

As for the inverse FT, it can also be extended to $L^2(\mathbb{R}^d)$ through Equation (1.3).

Convolution The *convolution product* between two signals x, y is the signal $x * y : \mathbb{R}^d \to \mathbb{C}$ defined by

$$(x * y)(\boldsymbol{t}) = \int_{\mathbb{R}^d} x(\boldsymbol{u}) y(\boldsymbol{t} - \boldsymbol{u}) d\boldsymbol{u}, \quad \boldsymbol{t} \in \mathbb{R}^d \quad .$$

The convolution theorem (Stein and Weiss, 1971, Theorem 2.6) links the notions of convolution and FT by stating that the FT of a convolution product of two signals, one of which is integrable and the other either integrable or with finite energy, is the point-wise product of their Fourier transforms:

$$\mathscr{F}\left[x * y\right] = \mathscr{F}[x] \mathscr{F}[y] \quad .$$

LTI operators See (Phillips et al., 2003, Chapter 3) for a more detailed approach. Let $d = 1$ for this particular definition. A *linear and time-invariant* (LTI) operator A is a map satisfying the following properties:

- Linearity: if x_1, x_2 are two signals, and c_1, c_2 are two scalar values, then $A[c_1 x_1 + c_2 x_2] = c_1 A[x_1] + c_2 A[x_2]$.
- Time invariance: A commutes with time shifts, i.e. $\forall \tau > 0$, $A[t \mapsto x(t - \tau)] = (t \mapsto A[x](t - \tau))$.

A LTI operator A can be entirely characterized by a single function $a : \mathbb{R} \mapsto \mathbb{C}$ called *impulse response* and such that the action of the operator on a time signal x is the convolution (in the time domain) of the impulse response and the signal:

$$A[x] = a * x \quad .$$

Equivalently, following the convolution theorem, LTI operators can also be characterized by the FT $\hat{a} = \mathscr{F}[a]$ of their impulse response, called *transfer function*. Then the action of the operator on a time signal is described as the product in the frequency domain of the transfer function and the Fourier transform of the signal:

$$\mathscr{F}\left[A[x]\right] = \hat{a} \times \mathscr{F}[x] \quad .$$

1.2.2 Harmonic analysis of discrete time signals

The material covered in this section is detailed in (Oppenheim et al., 2001, Chapter 2).

We assume that only a finite number n of samples from a signal and taken at regular time steps are observed and denote $x_1, \dots, x_n \in \mathbb{C}$ these samples. They are represented by the vector $\boldsymbol{x} = (x_1, \dots, x_n)^T$.

Harmonic analysis was extended to this setting by replacing the notion of Fourier transform by that of discrete Fourier transform. Both notions are linked as the discrete Fourier transform can be seen as the Fourier transform of a signal defined as a periodic train of n impulses corresponding to the n observed samples.

Discrete Fourier transform The *discrete Fourier transform* (DFT) of a vector of samples $\boldsymbol{x} \in \mathbb{C}^n$ is defined as the vector $\hat{\boldsymbol{x}} \in \mathbb{C}^n$ with entries

$$\hat{x}_k = \frac{1}{\sqrt{n}} \sum_{j=1}^{n} x_j e^{-\mathrm{i}\frac{2\pi}{n}(j-1)(k-1)}, \quad k \in [\![1, n]\!] \quad .$$

Each sample of $\boldsymbol{x}$ can be retrieved from the set of its DFT coefficients using the following inversion formula:

$$x_j = \frac{1}{\sqrt{n}} \sum_{k=1}^{n} \hat{x}_k e^{\mathrm{i}\frac{2\pi}{n}(j-1)(k-1)}, \quad j \in [\![1, n]\!] \quad .$$

The DFT can be seen as the projection of an input signal onto an orthonormal basis of discrete and finite signals. Indeed, let $\boldsymbol{F}$ be the matrix defined by

$$\boldsymbol{F} = \frac{1}{\sqrt{n}} \left[e^{\mathrm{i}\frac{2\pi}{n}(j-1)(k-1)} \right]_{1 \leq j,k \leq n} \quad . \tag{1.4}$$

On one hand, $\boldsymbol{F}$ entirely defines the DFT as for any $\boldsymbol{x} \in \mathbb{C}^n$:

$$\hat{\boldsymbol{x}} = \boldsymbol{F}^H \boldsymbol{x} \text{ and } \boldsymbol{x} = \boldsymbol{F}\hat{\boldsymbol{x}} \quad .$$

On the other hand, $\boldsymbol{F}$ is a unitary matrix, i.e. $\boldsymbol{F}^{-1} = \boldsymbol{F}^H$. Its columns therefore form an orthonormal basis of $\mathbb{C}^n$ for its canonical inner product $\langle ., . \rangle_{\mathbb{C}^n}$:

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle_{\mathbb{C}^n} = \boldsymbol{x}^H \boldsymbol{y} = \sum_{k=1}^{n} \overline{x_k} y_k, \quad \boldsymbol{x}, \boldsymbol{y} \in \mathbb{C}^n \quad .$$

The DFT $\hat{\boldsymbol{x}}$ of vector $\boldsymbol{x}$ therefore corresponds to the coordinates of $\boldsymbol{x}$ in this basis.

The DFT carries many of the properties of the Fourier transform. It is a linear, invertible and unitary (for $\langle ., . \rangle_{\mathbb{C}^n}$) operator of $\mathbb{C}^n$. In particular, Plancherel's theorem still holds.

Convolution The convolution between two sequences $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{C}^n$ is the vector $\boldsymbol{x} * \boldsymbol{y}$ with entries

$$[\boldsymbol{x} * \boldsymbol{y}]_k = \sum_{j=1}^{n} x_j y_{((k-j)[n])+1}, \quad k \in [\![1, n]\!] \quad ,$$

where $(k-j)[n] \in [\![0, n-1]\!]$ is the remainder of the Euclidean division of $(k-j)$ by n. Its k-th entry corresponds to the sum-product of the sequence $\boldsymbol{x}$ and a "wrapped" version of the sequence $\boldsymbol{y}$ that starts with its k-th entry.

The convolution theorem still holds with discrete sequences of samples, using now the DFT:

$$\mathrm{DFT}[\boldsymbol{x} * \boldsymbol{y}] = \mathrm{DFT}[\boldsymbol{x}] \odot \mathrm{DFT}[\boldsymbol{y}] \quad ,$$

where $\odot$ denotes the entry-wise product of two vectors.

Circular convolutive operators *Circular convolutive operators* are defined as linear operators $\boldsymbol{A} \in \mathscr{M}_n(\mathbb{C})$ for which this matrix is a circulant matrix, i.e. there exists a sequence $\boldsymbol{a} = (a_1, \ldots, a_n)^T \in \mathbb{C}^n$ such that:

$$\boldsymbol{A} = \begin{pmatrix} a_1 & a_n & \ldots & \ldots & a_2 \\ a_2 & a_1 & a_n & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & a_n \\ a_n & \ldots & \ldots & a_2 & a_1 \end{pmatrix} \quad .$$

In particular, the action of a circular convolutive operator $\boldsymbol{A}$ on a sequence of samples $\boldsymbol{x}$ can be written

$$\boldsymbol{A}\boldsymbol{x} = \left[\sum_{j=1}^{n} a_{((k-j)[n])+1} x_j \right]_{1 \leq k \leq n} = \boldsymbol{a} * \boldsymbol{x} \quad .$$

Such operators can be seen as the counterparts of LTI operators for finite sequences of samples, given that they share the same characterization using the convolution product.

Moreover, the notion of "time-invariance" can be extended to finite sequences of regular samples $\boldsymbol{x}$ by once again identifying them to periodic signals x composed of impulses corresponding to each sample. Then shifting such a signal by the sampling time is equivalent to applying a circular shift to the sequence. This last operation can be seen as applying the following permutation matrix to $\boldsymbol{x}$:

$$\boldsymbol{J} = \begin{pmatrix} 0 & 0 & \ldots & 0 & 1 \\ 1 & 0 & & & 0 \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & 1 & 0 \end{pmatrix} \in \mathscr{M}_n(\mathbb{R}) \quad . \tag{1.5}$$

This last matrix, called *circular shift matrix*, can be used to decompose any circulant matrix as

$$\boldsymbol{A} = \sum_{k=1}^{n} a_k \boldsymbol{J}^{k-1} \quad ,$$

where $\boldsymbol{J}^0 = \boldsymbol{I}$ by convention. Consequently, circular convolutive operators commute with the matrix $\boldsymbol{J}$ and therefore with time shifts.

1.2.3 Some notions regarding stochastic processes

Let $d \geq 1$ and denote $\mathcal{B}(\mathbb{R}^d)$ the set of all Borel sets of $\mathbb{R}^d$.

Weakly-stationary processes of $\mathbb{R}^d$

Let $X = \{X(\boldsymbol{t})\}_{\boldsymbol{t} \in \mathbb{R}^d}$ be a real-valued *stochastic process* indexed by $\mathbb{R}^d$, i.e. a family of real random variables $X(\boldsymbol{t})$ indexed by $\boldsymbol{t} \in \mathbb{R}^d$ and all defined on the same probability space. X is entirely characterized by the set of all *joint distribution functions* $F_{\boldsymbol{t}_1, \ldots, \boldsymbol{t}_n}$ defined by:

$$F_{\boldsymbol{t}_1, \ldots, \boldsymbol{t}_n} : (x_1, \ldots, x_n) \in \mathbb{R}^n \mapsto \mathbb{P}[X(\boldsymbol{t}_1) < x_1, \ldots, X(\boldsymbol{t}_n) < x_n] \quad ,$$

for any integer $n \geq 1$ and any $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_n \in \mathbb{R}^d$ (Parzen, 1999).

In practice, to catch a glimpse of the characteristics of X, its first two moments are preferred to the specification of all these distributions (Stein, 2012). Its first moment, called *expectation* of the process or *mean function*, is a function that assigns to any $\boldsymbol{t} \in \mathbb{R}^d$ the expectation of $X(\boldsymbol{t})$. Its second moment, called *variance function* of the process, is a function that assigns to any $\boldsymbol{t} \in \mathbb{R}^d$ the variance of $X(\boldsymbol{t})$. Of particular interest is the cross-moment of X, also called *covariance function* and defined as a function that assigns to any $\boldsymbol{t}_1, \boldsymbol{t}_2 \in \mathbb{R}^d$ the covariance $X(\boldsymbol{t}_1)$ and $X(\boldsymbol{t}_2)$.

The process X is called *weakly stationary* (or *second-order stationary*) if:

- its mean function is constant: $\exists \mu \in \mathbb{R}, \forall \boldsymbol{t} \in \mathbb{R}^d, \mathbb{E}[X(\boldsymbol{t})] = \mu$,
- there exists a function $C_X : \mathbb{R}^d \mapsto \mathbb{R}$ such that the covariance function satisfies: $\forall (\boldsymbol{t}_1, \boldsymbol{t}_2) \in \mathbb{R}^d \times \mathbb{R}^d$, $\mathrm{Cov}[X(\boldsymbol{t}_1), X(\boldsymbol{t}_2)] = C_X(\boldsymbol{t}_2 - \boldsymbol{t}_1)$.

Remark 1.2.1. Note that the variance of a weakly stationary process must consequently be finite and constant as $\forall \boldsymbol{t} \in \mathbb{R}^d$, $\mathrm{Var}[X(\boldsymbol{t})] = \mathrm{Cov}[X(\boldsymbol{t}), X(\boldsymbol{t})] = C_X(\boldsymbol{0})$.

Remark 1.2.2. The condition satisfied by the covariance function of a weakly stationary process can be expressed using the Dirac delta function:

$$\mathrm{Cov}[X(\boldsymbol{t}_1), X(\boldsymbol{t}_2)] = C_X(\boldsymbol{t}_2 - \boldsymbol{t}_1) = C_X * \delta_{\boldsymbol{t}_1}(\boldsymbol{t}_2), \quad \boldsymbol{t}_1, \boldsymbol{t}_2 \in \mathbb{R}^d \quad .$$

A weakly stationary process X is called *isotropic* if its covariance function C_X is radial, i.e. there exists a function $\tilde{C}_X : \mathbb{R}_+ \mapsto \mathbb{R}$ such that $\forall \boldsymbol{h} \in \mathbb{R}^d, C_X(\boldsymbol{h}) = \tilde{C}_X(\|\boldsymbol{h}\|)$. For sake of simplicity, the same notation C_X is from now on be used to denote both the covariance function of X and when applicable its writing as a radial function $\tilde{C}_X$.

Zero-mean weakly stationary processes admit a spectral representation (Stein, 2012, Section 2.5). Let X denote such a process. Then X can be written as the inverse Fourier transform of a complex random measure[1] M_X on $\mathbb{R}^d$:

$$X(\boldsymbol{t}) = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} e^{\mathrm{i}\langle \boldsymbol{\xi}, \boldsymbol{t} \rangle} M_X(d\boldsymbol{\xi}) = \mathscr{F}^{-1}[M_X](\boldsymbol{t}) \quad , \tag{1.6}$$

where M_X satisfies:

- $\forall B \in \mathcal{B}(\mathbb{R}^d)$, $\mathbb{E}[M_X(B)] = 0$.
- there exists a finite positive measure F_X on $\mathbb{R}^d$ such that: $\forall B \in \mathcal{B}(\mathbb{R}^d)$, $\mathrm{Var}[M_X(B)] = F_X(B)$.
- $\forall B_1, B_2 \in \mathcal{B}(\mathbb{R}^d)$ such that $B_1 \cap B_2 = \emptyset$, $\mathrm{Cov}[M_X(B_1), M_X(B_2)] = 0$.

The measure F_X is called the *spectral measure* of X. The spectral measure of a weakly stationary process X is linked to its covariance function C_X through the Fourier transform:

$$C_X(\boldsymbol{h}) = \mathscr{F}^{-1}[F_X](\boldsymbol{h}) = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} e^{\mathrm{i}\langle \boldsymbol{\xi}, \boldsymbol{h} \rangle} F(d\boldsymbol{\xi})$$

The density f_X of the spectral measure F_X, when it exists, is called the *spectral density* of X and satisfies:

$$C_X(\boldsymbol{h}) = \mathscr{F}^{-1}[f_X](\boldsymbol{h}) = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} e^{\mathrm{i}\langle \boldsymbol{\xi}, \boldsymbol{h} \rangle} f_X(\boldsymbol{\xi}) d\boldsymbol{\xi}$$

In particular, given that the Fourier transform of a radial function is also radial (Ormerod, 1979), the spectral density of an isotropic field will be a radial function. In particular, Ormerod

[1] A random measure can be considered as a stochastic process indexed by the elements of $\mathcal{B}(\mathbb{R}^d)$ and that carries out the defining properties of a measure, namely the countable sigma-additivity and the null empty-set property.

(1979) even gives the formula linking a radial covariance function C_0 (which should be both integrable and square-integrable on $\mathbb{R}^d$) and its associated spectral density f_0:

$$f_0(\|\boldsymbol{\xi}\|) = \frac{1}{(2\pi)^{d/2}}\|\boldsymbol{\xi}\|^{1-d/2}\int_0^\infty C_0(r)J_{d/2-1}(\|\boldsymbol{\xi}\|r)r^{d/2}dr, \quad \boldsymbol{\xi}\in\mathbb{R}^d \quad , \tag{1.7}$$

where $J_{d/2-1}$ denotes the J-Bessel function with parameter $d/2-1$. Conversely, the expression of the radial covariance function C_0 can be retrieved from its radial spectral density f_0 through

$$C_0(\|\boldsymbol{h}\|) = (2\pi)^{d/2}\|\boldsymbol{h}\|^{1-d/2}\int_0^\infty f_0(r)J_{d/2-1}(\|\boldsymbol{h}\|r)r^{d/2}dr, \quad \boldsymbol{h}\in\mathbb{R}^d \quad . \tag{1.8}$$

White noise

A particular generalization of stochastic processes on $\mathbb{R}^d$, which is of great interest in this dissertation, is now introduced: the white noise. A random signed measure $\mathcal{W}$ on $\mathbb{R}^d$ is called a *white noise measure* (or simply *white noise*) with variance $\sigma^2 > 0$ if it satisfies (Carrizo Vergara, 2018; Lindgren et al., 2011):

- $\forall B \in \mathcal{B}(\mathbb{R}^d)$, $\mathbb{E}[\mathcal{W}(B)] = 0$.
- $\forall B_1, B_2 \in \mathcal{B}(\mathbb{R}^d)$, $\mathrm{Cov}[\mathcal{W}(B_1), \mathcal{W}(B_2)] = \mathbb{E}[\overline{\mathcal{W}(B_1)}\mathcal{W}(B_2)] = \sigma^2\mathrm{Leb}(B_1 \cap B_2)$ where Leb denotes here the Lebesgue measure of a Borel set.

The notion of spectral density can be extended to white noises by noticing that it admits a spectral representation very similar to that of weakly stationary processes and introduced in Equation (1.6).

Proposition 1.2.1. *Let $\mathcal{W}$ denote a white noise measure with variance σ^2 on $\mathbb{R}^d$. Then there exists a complex measure $M_{\mathcal{W}}$ satisfying:*

$$\mathcal{W}(d\boldsymbol{t}) = d\boldsymbol{t}\frac{1}{(2\pi)^d}\int_{\mathbb{R}^d} e^{\mathrm{i}\langle\boldsymbol{\xi},\boldsymbol{t}\rangle}M_{\mathcal{W}}(d\boldsymbol{\xi}) \quad ,$$

and such that:

- $\forall B \in \mathcal{B}(\mathbb{R}^d)$, $\mathbb{E}[M_{\mathcal{W}}(B)] = 0$.
- $\forall B \in \mathcal{B}(\mathbb{R}^d)$, $\mathrm{Var}[M_{\mathcal{W}}(B)] = (2\pi)^d\sigma^2 Leb(B)$.
- $\forall B_1, B_2 \in \mathcal{B}(\mathbb{R}^d)$ *such that* $B_1 \cap B_2 = \emptyset$, $\mathrm{Cov}[M_{\mathcal{W}}(B_1), M_{\mathcal{W}}(B_2)] = 0$.

Proof. See Appendix C.1. □

Similarly to stationary processes, the spectral measure of the white noise is defined as the measure associated to the variance of $M_{\mathcal{W}}$. Therefore, the spectral measure of the white noise is the Lebesgue measure, scaled with a factor $(2\pi)^d\sigma^2$. This measure admits a density, which defines the spectral density of the white noise and corresponds to the constant function equal to $(2\pi)^d\sigma^2$. The white noise can therefore be seen as generalized stochastic process with a spectral measure that is not finite but rather admits a "density" that is constant across the frequency domain.

Kernel representation of stationary processes

A representation of a class of weakly stationary stochastic processes of $\mathbb{R}^d$ using a convolution product of a white noise is now presented (Higdon et al., 1999). Let $k : \mathbb{R}^d \to \mathbb{R}$ denote a square-integrable function, called *kernel function*, and introduce Z the stochastic process defined by:

$$Z(\boldsymbol{t}) = \int_{\mathbb{R}^d} k(\boldsymbol{t}-\boldsymbol{s})\mathcal{W}(d\boldsymbol{s}), \quad \boldsymbol{t}\in\mathbb{R}^d \quad . \tag{1.9}$$

Then, Z is zero-mean and its covariance function satisfies:

$$\begin{aligned}\mathrm{Cov}[Z(\boldsymbol{t}_1), Z(\boldsymbol{t}_2)] = \mathbb{E}[Z(\boldsymbol{t}_1)Z(\boldsymbol{t}_2)] &= \mathbb{E}\left[\int_{\mathbb{R}^d}\int_{\mathbb{R}^d} k(\boldsymbol{t}_1-\boldsymbol{u})k(\boldsymbol{t}_2-\boldsymbol{v})\mathcal{W}(d\boldsymbol{u})\mathcal{W}(d\boldsymbol{v})\right] \\ &= \int_{\mathbb{R}^d}\int_{\mathbb{R}^d} k(\boldsymbol{t}_1-\boldsymbol{u})k(\boldsymbol{t}_2-\boldsymbol{v})\mathbb{E}\left[\mathcal{W}(d\boldsymbol{u})\mathcal{W}(d\boldsymbol{v})\right] \\ &= \int_{\mathbb{R}^d} k(\boldsymbol{t}_1-\boldsymbol{u})k(\boldsymbol{t}_2-\boldsymbol{u})d\boldsymbol{u} = \int_{\mathbb{R}^d} k(\boldsymbol{u})k(\boldsymbol{t}_2-\boldsymbol{t}_1+\boldsymbol{u})d\boldsymbol{u} \quad ,\end{aligned}$$

which is a function of the lag $\boldsymbol{t}_2 - \boldsymbol{t}_1$. Noticing, using the Cauchy–Schwartz inequality, that its values are always finite, we can conclude that Z is a weakly stationary process with covariance function:

$$C_Z(\boldsymbol{h}) = \int_{\mathbb{R}^d} k(\boldsymbol{u})k(\boldsymbol{h}+\boldsymbol{u})d\boldsymbol{u} = \int_{\mathbb{R}^d} k(\boldsymbol{v}-\boldsymbol{h})k(\boldsymbol{v})d\boldsymbol{u} = k * \check{k}(\boldsymbol{h}) \quad ,$$

where $\check{k}$ denotes the reflection of k i.e. $\forall \boldsymbol{t} \in \mathbb{R}^d, \check{k}(\boldsymbol{u}) = k(-\boldsymbol{u})$. Moreover, it admits a spectral density f_Z satisfying:

$$f_Z(\boldsymbol{\xi}) = \mathscr{F}[C_Z](\boldsymbol{\xi}) = \mathscr{F}[k](\boldsymbol{\xi})\mathscr{F}[\check{k}](\boldsymbol{\xi}) = |\mathscr{F}[k](\boldsymbol{\xi})|^2 \quad ,$$

which clearly defines a positive finite measure given that k is square-integrable.

Conversely, given a spectral density f (i.e. a positive function defining a finite measure), a weakly process with spectral density f can be generated using Equation (1.9) by taking k as the function defined by:

$$k = \mathscr{F}^{-1}[\sqrt{f}] \quad .$$

1.3 Graph signal processing in a nutshell

Now that the two building blocks necessary to its construction have been laid out, we introduce the general framework used in graph signal processing. The notions presented in this section are part of the standard framework used in the graph signal processing community. They are also introduced in (Girault, 2015a; Ortega et al., 2018; Perraudin and Vandergheynst, 2017; Shuman et al., 2013).

1.3.1 Signals on a graph

A *graph signal* x on a n-graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ is a function $x : \mathcal{V} \to \mathbb{C}$ that assigns to each vertex i of $\mathcal{G}$ a complex number $x(i)$. Any graph signal x can be represented by a vector $\boldsymbol{x}$ such that $x_i = x(i)$. Hence, vectors of $\mathbb{C}^n$ are identified with signals on a n-graph. A signal defined on a graph $\mathcal{G}$ is called a $\mathcal{G}$-signal.

Example 1.3.1 (Digital image processing)**.** A digital image is a rectangular grid of adjacent colored points, also called pixels. A simple undirected graph $\mathcal{G}$ can be associated to a given digital image as follows: each pixel of the image is associated to a vertex of $\mathcal{G}$ and adjacent pixels define adjacent vertices on $\mathcal{G}$.

By definition, each pixel has a color. For black-and-white images, this color can be represented by a real value ranging from 0 (for black) to 1 (for white) and corresponding to a shade of grey. Hence, the function that associates to each pixel its shade of grey defines a signal on the graph $\mathcal{G}$.

The inner product of two signals $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{C}^n$ is defined as the inner product of the corresponding vectors, and is denoted:

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle := \langle \boldsymbol{x}, \boldsymbol{y} \rangle_{\mathbb{C}^n} = \sum_{i=1}^{n} \bar{x_i} y_i \quad .$$

The energy $E(\boldsymbol{x})$ of a graph signal $\boldsymbol{x} \in \mathbb{C}^n$ is defined as the square of 2-norm:

$$E(\boldsymbol{x}) = \|\boldsymbol{x}\|^2 = \langle \boldsymbol{x}, \boldsymbol{x} \rangle = \sum_{i=1}^{n} |x_i|^2$$

These definitions are natural extensions of the definition of the inner product and energy of continuous signals in classical signal processing.

1.3.2 Graph shift operators

A $n \times n$ matrix $\boldsymbol{S}$ is called a *shift operator* for the graph $\mathcal{G}$ if its entries satisfy

$$\forall i, j \in [\![1, n]\!], \quad S_{ij} \neq 0 \Rightarrow i \sim j \text{ or } i = j \quad .$$

Hence, the off-diagonal non-zero entries of a shift operator indicate the existence of an edge between two vertices of $\mathcal{G}$.

More generally, the non-zero entries of the iterates $\boldsymbol{S}^k, k \geq 2$ of $\boldsymbol{S}$ provide some knowledge about the existence of a path of length k between two given vertices of the corresponding graph. Indeed, notice that the entries of $\boldsymbol{S}^k$ can be deduced from the entries of $\boldsymbol{S}$ by:

$$\left[\boldsymbol{S}^k\right]_{ij} = \sum_{l_1=1}^{n} S_{il_1} \left[\boldsymbol{S}^{k-1}\right]_{l_1 j} = \cdots = \sum_{l_1=1}^{n} \cdots \sum_{l_{k-1}=1}^{n} S_{il_1} S_{l_1 l_2} \ldots S_{l_{k-1} j}, \quad k \geq 2 \quad .$$

Hence for $\left[\boldsymbol{S}^k\right]_{ij}$ to be non-zero, at least one of the terms $S_{il_1} S_{l_1 l_2} \ldots S_{l_{k-1} j}$ must be non-zero, meaning that there must exist a sequence of $k-1$ vertices $l_1, \ldots, l_{k-1}$ such that this term is non-zero. According to the definition of shift operators this actually means that the sequence $i, l_1, \ldots, l_{k-1}, j$ forms a path (of length k) between i and j, which consequently are linked by a path of length k.

Shift operators can be seen as linear operators on $\mathbb{C}^n$ whose action is defined by

$$\boldsymbol{S} : \boldsymbol{u} \in \mathbb{C}^n \mapsto \boldsymbol{S}\boldsymbol{u} \in \mathbb{C}^n \quad .$$

The signal $\boldsymbol{S}\boldsymbol{u}$ is then said to be shifted. Notice that, according to the non-zero pattern of $\boldsymbol{S}$, the value of the shifted signal $\boldsymbol{S}\boldsymbol{u}$ at a vertex i satisfies

$$\forall i \in [\![1, n]\!], \quad [\boldsymbol{S}\boldsymbol{u}]_i = S_{ii} u_i + \sum_{\substack{j \neq i \\ j \sim i}} S_{ij} u_j \quad .$$

Hence the value of the shifted signal $\boldsymbol{S}\boldsymbol{u}$ at a vertex i is a weighted sum of the values of $\boldsymbol{u}$ at i and its adjacent vertices and therefore can be seen as a local transformation of the original signal $\boldsymbol{u}$. Another interpretation of shifted signals, which justifies their name, consists in noticing that to compute the value of $\boldsymbol{S}\boldsymbol{u}$ at a vertex i, one needs to "shift" along the edges of the graph and towards i the values taken by $\boldsymbol{u}$ at the adjacent vertices of i, and then compute a linear combination of these values.

Example 1.3.2 (Adjacency matrix)**.** The adjacency matrix $\boldsymbol{\mathcal{W}}$ of $\mathcal{G}$ is a possible choice shift operator. Seen as an operator on $\mathbb{C}^n$, its action is defined as

$$\boldsymbol{\mathcal{W}} : \boldsymbol{u} \in \mathbb{C}^n \mapsto \boldsymbol{\mathcal{W}}\boldsymbol{u} = \left[\sum_{j=1}^{n} \mathcal{W}_{ij} u_j \right]_{1 \leq i \leq n} \in \mathbb{C}^n \quad .$$

Therefore, applying the adjacency matrix to a signal results in computing for each vertex the weighted average of the values of the signal at its adjacent vertices, the weights being defined as the edge weights.

Example 1.3.3 (Laplacian matrix)**.** The Laplacian matrix $\boldsymbol{L}$ of $\mathcal{G}$ is another possible choice of shift operator. Seen as an operator on $\mathbb{C}^n$, its action is defined as

$$\boldsymbol{L} : \boldsymbol{u} \in \mathbb{C}^n \mapsto \boldsymbol{L}\boldsymbol{u} = \left[\sum_{j=1}^{n} \mathcal{W}_{ij} (u_i - u_j) \right]_{1 \leq i \leq n} \in \mathbb{C}^n \quad .$$

Therefore, applying the Laplacian matrix to a signal results in computing for each vertex the weighted average of the differences between the value of the signal at this vertex and the values at its adjacent vertices. Hence, similarly to the discretization of the Laplacian operator of functions of $\mathbb{R}^d$ in finite differences, the graph Laplacian computes at each vertex i a weighted sum of the differences between the value of a signal at i and the value it takes in each direction. In the graph settings these directions are defined by the edges linked to i.

Note also that the inner product between a signal $\boldsymbol{x}$ and the shifted signal $\boldsymbol{L}\boldsymbol{x}$ satisfies

$$\langle \boldsymbol{x}, \boldsymbol{L}\boldsymbol{x} \rangle = \langle \boldsymbol{L}\boldsymbol{x}, \boldsymbol{x} \rangle = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \mathcal{W}_{ij} |x_i - x_j|^2$$

and can therefore be seen as a measure of the variations of the signal $\boldsymbol{x}$ along the edges of the graph.

Example 1.3.4 (normalized Laplacian matrix)**.** Just like the Laplacian matrix, the normalized Laplacian matrix $\tilde{\boldsymbol{L}}$ of $\mathcal{G}$ is also a possible shift operator. Seen as an operator on $\mathbb{C}^n$, its action is defined as

$$\tilde{\boldsymbol{L}} : \boldsymbol{u} \in \mathbb{C}^n \mapsto \tilde{\boldsymbol{L}}\boldsymbol{u} = \left[\frac{1}{\sqrt{d_i}} \sum_{j=1}^{n} \mathcal{W}_{ij} \left(\frac{u_i}{\sqrt{d_i}} - \frac{u_j}{\sqrt{d_j}} \right) \right]_{1 \leq i \leq n} \in \mathbb{C}^n \quad ,$$

and can be seen as applying a graph Laplacian to a scaled version of the signal. The scaling in question consists in scaling down the values of the signals corresponding to high degree vertices. The inner product between a signal $\boldsymbol{x}$ and the shifted signal $\tilde{\boldsymbol{L}}\boldsymbol{x}$ now writes:

$$\langle \boldsymbol{x}, \tilde{\boldsymbol{L}}\boldsymbol{x} \rangle = \langle \tilde{\boldsymbol{L}}\boldsymbol{x}, \boldsymbol{x} \rangle = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \mathcal{W}_{ij} \left| \frac{x_i}{\sqrt{d_i}} - \frac{x_j}{\sqrt{d_j}} \right|^2$$

and can therefore still be seen as a measure of the variations of the scaled signal $\boldsymbol{D}^{-1/2}\boldsymbol{x}$ along the edges of the graph.

In the remainder if this chapter, the following assumption is made on the shift operators that will be considered.

Assumption 1.2. *Only real, symmetric shift operators $\boldsymbol{S}$ are considered.*
Consequently, $\boldsymbol{S}$ is diagonalizable by a unitary matrix and has real eigenvalues. Such a decomposition is denoted as follows:

$$\boldsymbol{S} = \boldsymbol{V} \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix} \boldsymbol{V}^H \quad ,$$

where

- $\lambda_1 \leq \cdots \leq \lambda_n$ *denote the real eigenvalues of $\boldsymbol{S}$, ordered in ascending order,*
- $\boldsymbol{V} = \left[\boldsymbol{v}^{(1)} | \ldots | \boldsymbol{v}^{(n)} \right]$ *is a unitary matrix (i.e. $\boldsymbol{V}^{-1} = \boldsymbol{V}^H$)) whose columns $\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(n)}$ form an orthonormal basis of $\mathbb{C}^n$ composed of eigenvectors of $\boldsymbol{S}$ such that:*

$$\forall i \in [\![1, n]\!], \quad \boldsymbol{S}\boldsymbol{v}^{(i)} = \lambda_i \boldsymbol{v}^{(i)} \quad .$$

Remark 1.3.1. Note that $\boldsymbol{V}$ can be chosen to be a real matrix, i.e. $\boldsymbol{v}^{(1)}, \dots, \boldsymbol{v}^{(n)}$ can be chosen to be a orthonormal basis of $\mathbb{R}^n$ composed of real vectors and $\boldsymbol{V}^{-1} = \boldsymbol{V}^T$.

1.3.3 Harmonic analysis of graph signals

A starting point: the ring graph

The *ring graph* of size n is the unweighted n-graph such that each vertex $i \in [\![1, n]\!]$ is (only) linked to the vertices $i-1$ and $i+1$. By convention the label 0 corresponds to the vertex n and the label $n+1$ corresponds to the vertex 1, hence the circular property.

The ring graph is an undirected simple graph. Its adjacency matrix is the symmetric matrix $\boldsymbol{\mathcal{W}}_r$ defined as:

$$[\boldsymbol{\mathcal{W}}_r]_{ij} = \begin{cases} 1 & \text{if } (i-j) \equiv \pm 1 \pmod{n} \\ 0 & \text{otherwise} \end{cases}, \quad 1 \leq i, j \leq n \quad .$$

Equivalently, $\boldsymbol{\mathcal{W}}_r$ can be expressed using the circular permutation matrix $\boldsymbol{J}$ (cf. Equation (1.5)) as

$$\boldsymbol{\mathcal{W}}_r = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 & 1 \\ 1 & 0 & 1 & & & 0 \\ 0 & 1 & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & 1 & 0 \\ 0 & & & 1 & \ddots & 1 \\ 1 & 0 & \dots & 0 & 1 & 0 \end{pmatrix} = \boldsymbol{J} + \boldsymbol{J}^T = \boldsymbol{J} + \boldsymbol{J}^{n-1} \quad .$$

The corresponding degree matrix $\boldsymbol{D}_r$ is then given by:

$$\boldsymbol{D}_r = \boldsymbol{\mathcal{W}}_r \mathbf{1} = 2\boldsymbol{I}_n \quad .$$

Finally the Laplacian matrix $\boldsymbol{L}_r$ of the ring graph is given by:

$$\boldsymbol{L}_r = \boldsymbol{D}_r - \boldsymbol{\mathcal{W}}_r = 2\boldsymbol{I}_n - \boldsymbol{J} - \boldsymbol{J}^{n-1} \quad .$$

Let $\boldsymbol{S}_r$ denote either the adjacency matrix or the Laplacian of the ring graph. $\boldsymbol{S}$ is in particular a shift operator of this graph. In both cases, there exists a polynomial P_r such that:

$$\boldsymbol{S}_r = P_r(\boldsymbol{J})$$

Indeed, P_r is the polynomial $X \mapsto X + X^{n-1}$ if $\boldsymbol{S}_r = \boldsymbol{\mathcal{W}}_r$ and $X \mapsto 2 - X - X^{n-1}$ if $\boldsymbol{S}_r = \boldsymbol{L}_r$.

Recall that $\boldsymbol{J}$ is a diagonalizable matrix with n distinct eigenvalues which are n roots of unity and an orthonormal eigenbasis given by the DFT matrix $\boldsymbol{F}$ (cf. Equation (1.4)):

$$\boldsymbol{J} = \boldsymbol{F}\mathrm{Diag}\left(1, \omega, \dots, \omega^{n-1}\right)\boldsymbol{F}^H, \quad \boldsymbol{F} = \frac{1}{\sqrt{n}}\left[\omega^{(j-1)(k-1)}\right]_{1 \leq j,k \leq n} \quad ,$$

where $\omega = e^{\mathrm{i}\frac{2\pi}{n}}$ and $\boldsymbol{F}$ satisfies $\boldsymbol{F}^{-1} = \boldsymbol{F}^H$. In particular, the shift operator $\boldsymbol{S}_r$ verifies:

$$\boldsymbol{S}_r = P_r(\boldsymbol{J}) = \boldsymbol{F}\mathrm{Diag}\left(P_r(1), P_r(\omega), \dots, P_r(\omega^{n-1})\right)\boldsymbol{F}^H \quad .$$

The DFT can therefore be seen as projection onto an eigenbasis of a shift operator of the ring graph.

Getting back to general graph signals, let's recall that signals on a n-graph can be identified with vectors of $\mathbb{C}^n$ and hence with sequences of n samples. In particular for signals defined on the ring graph, the DFT of the corresponding sequence of samples is exactly the projection of the signal onto an eigenbasis of a shift operator of the graph on which it is defined. This observation motivates the generalization of the notion of Fourier transform of signals on more general graphs.

Graph Fourier Transform

Following Assumption 1.2, the eigenvectors of a shift operator actually form an orthonormal basis of $\mathbb{C}^n$, thus meaning that any signal can be (uniquely) decomposed as a weighted sum of these eigenvectors. This decomposition defines the notion of graph Fourier transform.

Definition 1.3.1. *Let $\boldsymbol{x} \in \mathbb{C}^n$ be a signal on a n-graph with shift operator $\boldsymbol{S}$. The graph Fourier transform (GFT) of $\boldsymbol{x}$ with respect to an orthonormal eigenbasis $\boldsymbol{V}$ of $\boldsymbol{S}$ is the vector* $\mathrm{GFT}\,[\boldsymbol{x}]$ *defined as:*

$$\mathrm{GFT}\,[\boldsymbol{x}] = \boldsymbol{V}^H \boldsymbol{x} = \begin{pmatrix} \langle \boldsymbol{v}^{(1)}, \boldsymbol{x} \rangle \\ \vdots \\ \langle \boldsymbol{v}^{(n)}, \boldsymbol{x} \rangle \end{pmatrix} \quad .$$

The GFT of a signal is therefore the vector containing its coordinates in the eigenbasis of the shift operator:

$$\boldsymbol{x} = \sum_{i=1}^{n} \langle \boldsymbol{v}^{(i)}, \boldsymbol{x} \rangle \boldsymbol{v}^{(i)} = \sum_{i=1}^{n} \left[\mathrm{GFT}\,[\boldsymbol{x}]\right]_i \boldsymbol{v}^{(i)} \quad .$$

It can be seen as a signal indexed by the eigenvalues of the shift operator, thus motivating the use of the term *graph frequency* to refer to the eigenvalues of the shift operator. The term *graph modes* then refers to the corresponding eigenvectors.

Any signal therefore has two equivalent representations:

- in the *vertex domain*: the signal is seen as the assignment of a real value to each vertex
- in the *frequency domain*: the signal is seen as a linear combination of elementary signals defined as the eigenvectors of a shift operator and is characterized by the weights involved in the combination.

The GFT with respect to $\boldsymbol{V}$ is an invertible operation. The inverse GFT of a vector $\boldsymbol{y} \in \mathbb{C}^n$ is defined as follows:

$$\mathrm{GFT}^{-1}[\boldsymbol{y}] = \left(\boldsymbol{V}^H\right)^{-1} \boldsymbol{y} = \boldsymbol{V}\boldsymbol{y} \quad .$$

1.3.4 Graph convolutions

The definition of the convolution of graph signals relies on an analogy with the classical signal processing framework. Indeed, the convolution theorem states that the FT of the convolution of two (continuous) signals equals the point-wise product of their FTs. To conserve this property in the graph signal processing framework, the *convolution of two graph signals $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{C}^n$* is defined as the graph signal $\boldsymbol{x} * \boldsymbol{y}$ satisfying

$$\boldsymbol{x} * \boldsymbol{y} = \mathrm{GFT}^{-1}\left[\mathrm{GFT}[\boldsymbol{x}] \odot \mathrm{GFT}[\boldsymbol{y}]\right] = \boldsymbol{V}\left(\left(\boldsymbol{V}^H \boldsymbol{x}\right) \odot \left(\boldsymbol{V}^H \boldsymbol{y}\right)\right) \quad .$$

From its definition, the convolution product of graph signals carries several of the important properties of the convolution of time signals. Namely, it is a commutative, associative and bilinear operation.

> **Remark 1.3.2.** The result of the convolution between two graph signals depends on the basis $\boldsymbol{V}$ chosen to define the GFT. Given that there is no uniqueness of eigendecomposition for a given shift operator, setting a shift operator is not sufficient to set the framework necessary to work with graph convolutions. The basis $\boldsymbol{V}$ should also be specified.

Graph convolutions are used to define graph translations, using an analogy with classical signal processing. Indeed, translating time-wise a signal by a delay τ is equivalent to convolution this same signal with a Dirac impulse at time τ. Both notions are now defined for the graph signal processing framework.

Definition 1.3.2. [*Dirac signal*] *Let $\mathcal{G}$ be a graph with set of vertices $\mathcal{V}$. The Dirac signal of $\mathcal{G}$ at vertex $i \in \mathcal{V}$ is the $\mathcal{G}$-signal $\boldsymbol{\delta}^{(i)}$ defined by:*

$$\forall k \in \mathcal{V}, \quad \delta_k^{(i)} = \delta_{ik}$$

Definition 1.3.3. [*Graph translation*] *Let $\mathcal{G}$ be a graph with set of vertices $\mathcal{V}$ and let $\boldsymbol{x}$ be a $\mathcal{G}$-signal. The translation of $\boldsymbol{x}$ with respect to vertex $i \in \mathcal{V}$ is the $\mathcal{G}$-signal $\boldsymbol{T}_i\boldsymbol{x}$ defined by:*

$$\boldsymbol{T}^{(i)}\boldsymbol{x} = \boldsymbol{\delta}^{(i)} * \boldsymbol{x}$$

In particular, the translation operator $\boldsymbol{T}^{(i)}$ that maps any $\mathcal{G}$-signal to its translation with respect to $i \in \mathcal{V}$ can be defined by:

$$\boldsymbol{T}^{(i)} = \boldsymbol{V}\mathrm{Diag}\left(\boldsymbol{V}^H\boldsymbol{\delta}^{(i)}\right)\boldsymbol{V}^H$$

1.3.5 Graph filters

A linear operator on graph signals is a linear mapping from $\mathbb{C}^n$ to itself. Defined as such, it can be represented by a matrix $\boldsymbol{A} \in \mathscr{M}_n(\mathbb{C})$ whose columns correspond to the image of the canonical basis of $\mathbb{C}^n$. This operator is called real if its representative matrix is real. It can then be seen as a linear mapping from $\mathbb{R}^n$ to itself.

Graph filters are a class of linear operators on graph signals that act on the frequency content of a signal. Through the GFT, any signal can be decomposed as a weighted sum of elementary signals, each associated to a given graph frequency. Graph filters aim at amplifying or attenuating the weight of some of these signals on the overall decomposition. Such an operation can be modeled using a *transfer function* A, which is a function that associates to each graph frequency λ a scaling factor $A(\lambda) \in \mathbb{C}$. Applying a graph filter with transfer function A to a signal $\boldsymbol{x}$ yields a signal $\boldsymbol{y}$ such that:

$$\mathrm{GFT}[\boldsymbol{y}]_i = A(\lambda_i)\mathrm{GFT}[\boldsymbol{x}]_i, \quad i \in [\![1, n]\!] \quad .$$

Hence the i-th spectral component of the input signal $\boldsymbol{x}$ is now scaled by a factor $A(\lambda_i)$. Note in particular that duplicated frequencies/eigenvalues are necessarily scaled by the same factor.

Applying the inverse GFT to both members of this last equation gives:

$$\boldsymbol{y} = \boldsymbol{V}\begin{pmatrix} A(\lambda_1) & & \\ & \ddots & \\ & & A(\lambda_n) \end{pmatrix}\boldsymbol{V}^H\boldsymbol{x} \quad .$$

Graph filters can therefore be represented by matrix functions $A(\boldsymbol{S})$ of the shift operator $\boldsymbol{S}$ defined by

$$A(\boldsymbol{S}) := \boldsymbol{V}\begin{pmatrix} A(\lambda_1) & & \\ & \ddots & \\ & & A(\lambda_n) \end{pmatrix}\boldsymbol{V}^H \in \mathscr{M}_n(\mathbb{C}) \quad .$$

Their action on a signal $\boldsymbol{x}$ is then $\boldsymbol{y} = A(\boldsymbol{S})\boldsymbol{x}$ and this vector is called a *filtered signal*.

Three ingredients seem necessary to define a graph filter:

- a choice of diagonalizable shift operator $\boldsymbol{S}$ with eigenvalues $\lambda_1, \ldots, \lambda_n$,
- a set of values $\{A(\lambda_1), \ldots, A(\lambda_n)\}$, called *frequency response* of the graph filter, and corresponding to the image of the set of eigenvalues of $\boldsymbol{S}$ through a (transfer) function A,
- a choice of (orthonormal) basis $\boldsymbol{V}$ for the eigendecomposition of $\boldsymbol{S}$.

The following theorem proves that the third requirement is actually not necessary, meaning that graph filters can be defined independently from the choice of the eigenbasis $\boldsymbol{V}$.

Theorem 1.3.1. *Any graph filter defined on a n-graph with shift operator $\boldsymbol{S}$ and transfer function A can be uniquely written as a matrix polynomial of $\boldsymbol{S}$ of degree at most $n-1$, i.e., there exists a unique set of coefficients $a_0, \ldots, a_{n-1} \in \mathbb{C}$ such that:*

$$A(\boldsymbol{S}) = \sum_{k=0}^{n-1} a_k \boldsymbol{S}^k$$

In particular the coefficients $a_0, \ldots, a_{n-1} \in \mathbb{C}$ are entirely defined by the frequency response of the graph filter.

Proof. Let $\boldsymbol{V}$ denote any eigenbasis of $\boldsymbol{S}$. Let $A(\boldsymbol{S})$ be a graph filter defined through $\boldsymbol{V}$ and with transfer function A and denote $(A(\lambda_1)\dots A(\lambda_n))^T \in \mathbb{C}^n$ its frequency response.

Define P_A to be the Lagrange interpolation polynomial that assigns to each $\lambda_i, i \in [\![1,n]\!]$ the value $P_A(\lambda_i) = A(\lambda_i)$. Let $n_\lambda \leq n$ be the number of distinct eigenvalues of $\boldsymbol{S}$. According to the unisolvence theorem, P_A is the only polynomial of degree $\leq n_\lambda - 1$ that interpolates A at points $\lambda_1, \dots, \lambda_n$. Denote $a_0, \dots, a_{n-1}$ the coefficients of this polynomial, some of which possibly being zero. Let $P_A(\boldsymbol{S})$ be the graph filter defined through $\boldsymbol{V}$ and with transfer function P_A.

On one hand,

$$P_A(\boldsymbol{S}) = \boldsymbol{V}\mathrm{Diag}\left(P_A(\lambda_1), \dots, P_A(\lambda_n)\right)\boldsymbol{V}^H = \boldsymbol{V}\mathrm{Diag}\left(A(\lambda_1), \dots, A(\lambda_n)\right)\boldsymbol{V}^H = A(\boldsymbol{S}) \quad .$$

On the other hand,

$$\begin{aligned} P_A(\boldsymbol{S}) &= \boldsymbol{V}\begin{pmatrix} \sum_{k=0}^{n-1} a_k\lambda_1^k & & \\ & \ddots & \\ & & \sum_{k=0}^{n-1} a_k\lambda_n^k \end{pmatrix}\boldsymbol{V}^H = \boldsymbol{V}\left(\sum_{k=0}^{n-1} a_k \begin{pmatrix} \lambda_1^k & & \\ & \ddots & \\ & & \lambda_n^k \end{pmatrix}\right)\boldsymbol{V}^H \\ &= \sum_{k=0}^{n-1} a_k \boldsymbol{V}\mathrm{Diag}\left(\lambda_1^k, \dots, \lambda_n^k\right)\boldsymbol{V}^H \quad . \end{aligned}$$

It is straightforward to show by induction that, whatever the choice of orthonormal basis $\boldsymbol{V}$, $\boldsymbol{V}\mathrm{Diag}\left(\lambda_1^k, \dots, \lambda_n^k\right)\boldsymbol{V}^H = \boldsymbol{S}^k$, which allows to conclude that $P_A(\boldsymbol{S}) = \sum_{k=0}^{n-1} a_k \boldsymbol{S}^k$.

Hence, P_A is defined independently of a choice of eigenbasis $\boldsymbol{V}$ given that it is a polynomial whose coefficients are independent of $\boldsymbol{V}$. This concludes the proof as $A(\boldsymbol{S}) = P_A(\boldsymbol{S})$. □

Graph filters are therefore uniquely specified by a choice of shift operator $\boldsymbol{S}$ and a choice of transfer function A which defines their frequency response. Note also that following Theorem 1.3.1, all graph filters can be seen as matrix polynomials, even though the associated transfer function is not derived from a polynomial function. The action of a graph filter $A(\boldsymbol{S})$ on a signal can then be expressed as:

$$y_i = [A(\boldsymbol{S})\boldsymbol{x}]_i = \sum_{k=0}^{K} a_k [\boldsymbol{S}^k\boldsymbol{x}]_i = \sum_{k=0}^{K} a_k \sum_{j=1}^{n} [\boldsymbol{S}^k]_{ij} x_j = \sum_{j=1}^{n} x_j \sum_{k=0}^{K} a_k [\boldsymbol{S}^k]_{ij} \quad ,$$

where $K \leq n-1$ is the actual order of the polynomial representing A. Hence, the value of the filtered signal at a vertex i is a linear combination of the values taken by the input signal.

More precisely, recall that the non-zero pattern of the iterates of the shift operator reflects the existence of a path between pairs of vertices. In particular, whenever there is no path of length lesser or equal to K between two vertices i and j, all elements $[\boldsymbol{S}^k]_{ij}, 0 \leq k \leq K$ are zero and therefore x_j is not used to compute the value of the filtered signal at vertex i. Formally this means that the value of the filtered signal at a vertex i is a linear combination of the values of the input signal within a K-hop neighborhood around i.

We now circle back to our ongoing analogy with classical signal processing. Graph filters are the counterparts for graph signals of the notion of linear and time-invariant (LTI) operator defined for continuous 1D signals. Both operators are linear maps and commute with translations. Indeed, for any graph signal $\boldsymbol{x}$, any graph filter $A(\boldsymbol{S})$ and any vertex $i \in \mathcal{V}$, the translation operator $\boldsymbol{T}^{(i)}$ with respect to i satisfies:

$$\boldsymbol{T}^{(i)}A(\boldsymbol{S})\boldsymbol{x} = \boldsymbol{V}\mathrm{Diag}\left(\boldsymbol{V}^H\boldsymbol{\delta}^{(i)}\right)\mathrm{Diag}\left(A(\lambda_1), \dots, A(\lambda_n)\right)\boldsymbol{V}^H\boldsymbol{x} = A(\boldsymbol{S})\boldsymbol{T}^{(i)}\boldsymbol{x} \quad .$$

Hence translating a filtered signal is the same as filtering the translated input. This is what defined time invariance for LTI operators on time signals.

Note that the same representation of LTI operators by a convolution product holds for graph filters. Indeed, it is straightforward to see that:

$$A(\boldsymbol{S})\boldsymbol{x} = \boldsymbol{a} * \boldsymbol{x} \text{ where } \boldsymbol{a} = \boldsymbol{V}\begin{pmatrix} A(\lambda_1) \\ \vdots \\ A(\lambda_n) \end{pmatrix} = \mathrm{GFT}^{-1}\left[\begin{pmatrix} A(\lambda_1) \\ \vdots \\ A(\lambda_n) \end{pmatrix}\right] \quad ,$$

where the convolution product and the vector $\boldsymbol{a}$ are defined using the same eigenbasis $\boldsymbol{V}$.

1.4 Stochastic graph signals

Now that the framework for studying (deterministic) graph signals is in place, we turn to its generalization to account for random graph signals. The aim is to provide some notions and tools that will help us work with stochastic processes defined on the vertices of a graph. The notions introduced will be compared to the existing literature on stochastic graph signal processing throughout the section.

In this section, $\mathcal{G}$ denotes a simple undirected n-graph and $\boldsymbol{S}$ denotes a shift operator of $\mathcal{G}$ following Assumption 1.2.

1.4.1 Stationary stochastic graph signals

A graph signal on a n-graph $\mathcal{G}$ is called stochastic if it assigns to each vertex of $\mathcal{G}$ a random variable. *Stochastic graphs signals* (SGS) on $\mathcal{G}$ can therefore be identified with random vectors of $\mathbb{C}^n$. As such, the first two moments of a SGS $\boldsymbol{X}$ are:

- its expectation, which is the vector of $\mathbb{C}^n$ whose elements are the expectations of the elements of $\boldsymbol{X}$: $[\mathbb{E}[\boldsymbol{X}]]_i = \mathbb{E}[X_i]$,
- its covariance matrix, which is the $n \times n$ matrix whose element (i, j) is the covariance between X_i and X_j: $\mathrm{Var}[\boldsymbol{X}] = \mathbb{E}\left[(\boldsymbol{X} - \mathbb{E}[\boldsymbol{X}])(\boldsymbol{X} - \mathbb{E}[\boldsymbol{X}])^H\right]$.

Assumption 1.3. *Unless otherwise specified, the SGS considered in this work are zero-mean, i.e.* $\mathbb{E}[\boldsymbol{X}] = \boldsymbol{0}$.

Based on this assumption, which will be discussed in Section 1.4.4, we introduce the notion of stationary graph signal that we will use in this work. This definition will be motivated in Section 1.4.2 and compared to existing definitions of stationary graph signals found in the literature in Section 1.4.3.

Definition 1.4.1. *Let $\boldsymbol{S}$ be a shift operator of $\mathcal{G}$. A zero-mean SGS $\boldsymbol{X}$ on $\mathcal{G}$ is called $\boldsymbol{S}$-stationary if its covariance matrix* $\mathrm{Var}[\boldsymbol{X}]$ *is a graph filter with a non-negative transfer function $f_X : \mathbb{R} \to \mathbb{R}_+$:*

$$\mathrm{Var}[\boldsymbol{X}] = f_X(\boldsymbol{S}) \quad .$$

The transfer function defining the covariance matrix of a $\boldsymbol{S}$-stationary SGS is called the spectral density of the SGS.

As defined, the notion of stationarity depends on the shift operator $\boldsymbol{S}$: the same SGS can therefore be stationary or not according to the choice of shift operator. Of particular interest however are white signals, that generalize to graph signals the notion of white noise and are "shift"-independent.

Example 1.4.1 (White signal)**.** A white signal on $\mathcal{G}$ is a zero-mean SGS $\boldsymbol{\mathcal{W}}$ whose components are independent zero-mean unit-variance random variables. Hence, $\boldsymbol{W}$ is a signal such that $\mathbb{E}[\boldsymbol{W}] = \boldsymbol{0}$ and $\mathrm{Var}[\boldsymbol{W}] = \boldsymbol{I}_n$.

White signals are always $\boldsymbol{S}$-stationary, for any choice of shift operator $\boldsymbol{S}$. The spectral density f_W of a white signal is the function satisfying $f_W(\lambda) = 1$ for all $\lambda \in \mathbb{R}$.

Proposition 1.4.1. *The GFT $\tilde{\boldsymbol{X}} = \mathrm{GFT}[\boldsymbol{X}]$ of a zero-mean $\boldsymbol{S}$-stationary graph signal $\boldsymbol{X}$ with spectral density f_X is a zero-mean SGS with uncorrelated components. Its covariance matrix is the diagonal matrix defined by:*

$$\mathrm{Var}[\tilde{\boldsymbol{X}}] = \boldsymbol{V}^H \mathrm{Var}[\boldsymbol{X}] \boldsymbol{V} = \begin{pmatrix} f_X(\lambda_1) & & \\ & \ddots & \\ & & f_X(\lambda_n) \end{pmatrix} \quad .$$

Proof. By linearity of the expectation, $\tilde{\boldsymbol{X}}$ is zero-mean. And $\mathrm{Var}[\tilde{\boldsymbol{X}}] = \mathrm{Cov}[\boldsymbol{V}^H \boldsymbol{X}, \boldsymbol{V}^H \boldsymbol{X}] = \boldsymbol{V}^H \mathrm{Var}[\boldsymbol{X}] \boldsymbol{V} = \boldsymbol{V}^H f_X(\boldsymbol{S}) \boldsymbol{V}$ which yields the result by definition of the graph filter $f_X(\boldsymbol{S})$. □

Following the definition of the GFT, a zero-mean $\boldsymbol{S}$-stationary graph signal $\boldsymbol{X}$ with spectral density f_X can be decomposed as:

$$\boldsymbol{X} = \sum_{i=1}^{n} \tilde{X}_i \boldsymbol{v}^{(i)} \quad ,$$

where:

- The signals $\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(n)}$ are deterministic and pairwise orthogonal (with respect to the graph scalar product).
- The weights $\tilde{X}_1, \ldots, \tilde{X}_n$ are random and pairwise uncorrelated variables with variances $f_X(\lambda_1), \ldots, f_X(\lambda_n)$.

This decomposition is therefore the GSP analogous of the Karhunen–Loève expansion of stochastic processes.

Remark 1.4.1. If f_X cancels out at a given eigenvalue λ_i then the corresponding weight X_i is a zero-mean variable with a 0 variance. It is therefore a deterministic constant set to 0, meaning that the corresponding SGS has no component along $\boldsymbol{v}^{(i)}$. More generally, the Karhunen–Loève expansion of a SGS $\boldsymbol{X}$ with spectral density f_X therefore writes:

$$\boldsymbol{X} = \sum_{i \in [\![1,n]\!] : f_X(\lambda_i) \neq 0} \tilde{X}_i \boldsymbol{v}^{(i)}$$

In particular, band-limited SGS can be defined by considering spectral densities that cancel out across a given bandwidth.

As defined, the image of a $\boldsymbol{S}$-stationary signal after application of a graph filter also defined through $\boldsymbol{S}$, is $\boldsymbol{S}$-stationary.

Theorem 1.4.2. *Let $\boldsymbol{X}$ be a $\boldsymbol{S}$-stationary SGS with spectral density f_X and let $h(\boldsymbol{S})$ be a graph filter with transfer function h. Then the filtered signal $\boldsymbol{Y} = h(\boldsymbol{S})\boldsymbol{X}$ is $\boldsymbol{S}$-stationary with spectral density $\lambda \mapsto h(\lambda)^2 f_X(\lambda)$.*

Proof. Clearly, $\boldsymbol{Y}$ is also a zero-mean SGS. Its covariance matrix is therefore given by: $\mathrm{Var}[\boldsymbol{Y}] = \mathbb{E}[\boldsymbol{Y}\boldsymbol{Y}^H] = \mathbb{E}[h(\boldsymbol{S})\boldsymbol{X}\boldsymbol{X}^H h(\boldsymbol{S})^H] = h(\boldsymbol{S})\mathbb{E}[\boldsymbol{X}\boldsymbol{X}^H]h(\boldsymbol{S})^H$. Using the fact that $h(\boldsymbol{S})$ is a Hermitian matrix gives $\mathrm{Var}[\boldsymbol{Y}] = h(\boldsymbol{S})\mathrm{Var}[\boldsymbol{X}]h(\boldsymbol{S}) = h(\boldsymbol{S})f_X(\boldsymbol{S})h(\boldsymbol{S})$ and using the fact that all these graph filters are related to the same shift operator yields: $\mathrm{Var}[\boldsymbol{Y}] = (hf_X h)(\boldsymbol{S}) = (h^2 f_X)(\boldsymbol{S})$. □

A defining property of $\boldsymbol{S}$-stationary signals is now introduced.

Theorem 1.4.3. *A (zero-mean) SGS $\boldsymbol{X}$ is $\boldsymbol{S}$-stationary with spectral density f_X iif there exists a white signal $\boldsymbol{W}$ such that:*

$$\boldsymbol{X} = \sqrt{f_X}(\boldsymbol{S})\boldsymbol{W} \quad .$$

Proof. Let $\boldsymbol{X}$ be a zero-mean $\boldsymbol{S}$-stationary SGS with spectral density f_X. Without loss of generality, let us assume that for a given $p \in [\![0, n]\!]$, $f(\lambda_1) = 0, \ldots, f(\lambda_p) = 0$ where by convention the case $p = 0$ corresponds to the case where all $f(\lambda_i)$ are non-zero. Let $\tilde{\boldsymbol{X}}$ be the GFT of $\boldsymbol{X}$. Let $\boldsymbol{W}$ denote the zero-mean SGS defined by:

$$\boldsymbol{W} = \boldsymbol{V}\left(\left(\begin{array}{c|c} \boldsymbol{0}_{p,p} & \\ \hline & \boldsymbol{D}_{n-p} \end{array}\right)\tilde{\boldsymbol{X}} + \begin{pmatrix} \boldsymbol{\epsilon}_p \\ \hline \boldsymbol{0}_{n-p} \end{pmatrix}\right) \quad ,$$

where $\boldsymbol{D}_{n-p}$ is a $(n-p)\times(n-p)$ diagonal matrix with entries $\left(\frac{1}{\sqrt{f_X(\lambda_{p+1})}}, \ldots, \frac{1}{\sqrt{f_X(\lambda_n)}}\right)$ and $\boldsymbol{\epsilon}$ is a vector with p independent zero-mean unit-variance components. Then, $\boldsymbol{W}$ is a white signal and satisfies $\sqrt{f_X}(\boldsymbol{S})\boldsymbol{W} = \boldsymbol{X}$.

The second implication of the proposition is a direct consequence of Theorem 1.4.2. □

1.4.2 Justification of the definition of stationarity

As defined, the notion of $\boldsymbol{S}$-stationarity for graph signals allows to draw direct parallels with the notion of weak stationarity that is defined for stochastic processes on $\mathbb{R}^d$.

Spectral representation

The notion of measure can be extended to the Graph Signal Processing framework as follows. A graph measure on the frequency domain is defined as a measure on the power set $\mathcal{P}(\Lambda)$ of the (finite) discrete set $\Lambda = \{\lambda_1, \ldots, \lambda_n\}$ composed of the graph Fourier frequencies of $\boldsymbol{S}$. A graph measure μ can be entirely characterized by the knowledge of the value of the measure of each singleton composing Λ. Then, the measure of any subset of $\mathcal{P}(\Lambda)$ is simply defined as:

$$\forall S \in \mathcal{P}(\Lambda), \quad \mu(S) = \sum_{i \in [\![1,n]\!] : \lambda_i \in S} \mu(\lambda_i) \quad .$$

Hence a graph measure μ can be represented by the n-vector $\boldsymbol{\mu} = (\mu(\lambda_1), \ldots, \mu(\lambda_n))^T$, which can be seen as signal on the graph frequency domain.

Remark 1.4.2. Similarly a graph measure on the vertex domain is defined as a measure on the power set $\mathcal{P}(\mathcal{V})$ of the (finite) discrete set $\mathcal{V}$ composed of the vertices of the graph.

Proposition 1.4.4. *Let $\boldsymbol{X}$ be a $\boldsymbol{S}$-stationary SGS with spectral density f_X. Then there exists a random graph measure μ_X such that:*

$$\boldsymbol{X} = \mathrm{GFT}^{-1}\left[\boldsymbol{\mu}_X\right] = \mathrm{GFT}^{-1}\left[\begin{pmatrix} \mu_X(\lambda_1) \\ \vdots \\ \mu_X(\lambda_n) \end{pmatrix}\right] \quad ,$$

where μ_X satisfies:

- $\forall S \in \mathcal{P}(\Lambda)$, $\mathbb{E}[\mu_X(S)] = 0$.
- *The positive graph measure defined from the spectral density f_X on Λ satisfies:* $\forall S \in \mathcal{P}(\Lambda)$, $\mathrm{Var}[\mu_X(S)] = f_X(S)$.
- $\forall S_1, S_2 \in \mathcal{P}(\Lambda)$ *such that* $S_1 \cap S_2 = \emptyset$, $\mathrm{Cov}[\mu_X(S_1), \mu_X(S_2)] = 0$.

Proof. Denote μ_X the measure defined by :

$$\boldsymbol{\mu}_X = \mathrm{GFT}\left[\boldsymbol{X}\right]$$

Clearly $\mathrm{GFT}^{-1}\left[\boldsymbol{\mu}_X\right] = \boldsymbol{X}$. According to Proposition 1.4.1 the vector $\boldsymbol{\mu}_X$ is a zero-mean random vector and has a diagonal covariance matrix with entries $(f_X(\lambda_1), \dots, f_X(\lambda_n))$. Hence, the corresponding graph measure is also zero-mean and satisfies $\forall S_1, S_2 \in \mathcal{P}(\Lambda)$,

$$\begin{aligned}\mathrm{Cov}[\mu_X(S_1), \mu_X(S_2)] &= \sum_{i \in [\![1,n]\!] \lambda_i \in S_1} \sum_{j \in [\![1,n]\!] : \lambda_j \in S_2} \mathrm{Cov}[\mu(\lambda_i), \mu(\lambda_j)] \\ &= \sum_{k \in [\![1,n]\!] : \lambda_k \in S_1 \cap S_2} f_X(\lambda_k) \quad .\end{aligned}$$

On one hand, if $S_1 \cap S_2 = \emptyset$, $\mathrm{Cov}[\mu_X(S_1), \mu_X(S_2)] = 0$. On the other hand, the spectral density f_X defines a positive graph measure that satisfies $\forall S \in \mathcal{P}(\Lambda)$, $\mathrm{Var}[\mu_X(S)] = f_X(S)$. □

The conventions chosen to define both notions of Fourier transform and stationarity for graph signals yield a direct correspondence with the framework of weakly stationary processes. Indeed, in both cases stationary signals can be represented as the inverse Fourier transform of a zero-mean random measure which is uncorrelated over disjoint sets and whose variance is a deterministic positive (finite) measure. Moreover, in both frameworks, the spectral density of the signal actually corresponds to the density of the spectral measure.

Convolution representation

Similarly as weakly stationary processes, a $\boldsymbol{S}$-stationary SGS can be obtained by convolving a white input with a kernel.

Proposition 1.4.5. *A (zero-mean) SGS $\boldsymbol{X}$ is $\boldsymbol{S}$-stationary with spectral density f_X iif there exists a white signal $\boldsymbol{W}$ such that:*

$$\boldsymbol{X} = \boldsymbol{k} * \boldsymbol{W}, \text{ where } \boldsymbol{k} = \mathrm{GFT}^{-1}\left[\begin{pmatrix} \sqrt{f_X}(\lambda_1) \\ \vdots \\ \sqrt{f_X}(\lambda_n) \end{pmatrix}\right] \quad .$$

Proof. Following the notations of the proposition and the definition of the convolution product of graph signals,

$$\begin{aligned}\boldsymbol{X} = \boldsymbol{V}\left(\left(\boldsymbol{V}^H\boldsymbol{k}\right) \odot \left(\boldsymbol{V}^H\boldsymbol{W}\right)\right) &= \boldsymbol{V}\left(\begin{pmatrix} \sqrt{f_X}(\lambda_1) \\ \vdots \\ \sqrt{f_X}(\lambda_n) \end{pmatrix} \odot \left(\boldsymbol{V}^H\boldsymbol{W}\right)\right) \\ = \boldsymbol{V}\mathrm{Diag}\left(\sqrt{f_X}(\lambda_1), \dots, \sqrt{f_X}(\lambda_n)\right)\boldsymbol{V}^H\boldsymbol{W} &= \sqrt{f_X}(\boldsymbol{S})\boldsymbol{W} \quad ,\end{aligned}$$

which proves the result according to Theorem 1.4.3. □

As it was the case with the framework of weakly stationary processes, stationary signals with a known spectral density are obtained by convolving white input (white noise or white graph signal) with a kernel function defined as the inverse Fourier transform of the square-root of the spectral density.

In terms of covariance, the next proposition provides a characterization of $\boldsymbol{S}$-stationary SGS, based on a convolution and similar to the one presented in Remark 1.2.2 for weakly stationary random fields.

Proposition 1.4.6. *A (zero-mean) SGS $\boldsymbol{X}$ is $\boldsymbol{S}$-stationary with spectral density f_X iif its covariance satisfies:*

$$\mathrm{Cov}[X_i, X_j] = [\boldsymbol{C} * \boldsymbol{\delta}_i]_j \quad , \tag{1.10}$$

where $\boldsymbol{C} = \mathrm{GFT}^{-1}\left[\begin{pmatrix} f_X(\lambda_1) \\ \vdots \\ f_X(\lambda_n) \end{pmatrix}\right]$ *and $\boldsymbol{\delta}_i$ is the Dirac signal at vertex i, that assigns the value 1 to vertex i and 0 to all other vertices.*

Proof. Denote $f_X(\boldsymbol{\lambda}) = (f_X(\lambda_1), \dots, f_X(\lambda_n))^T$. Then,

$$\begin{aligned}[\boldsymbol{C} * \boldsymbol{\delta}_i]_j &= \left[\boldsymbol{V}\left(f_X(\boldsymbol{\lambda}) \odot \left(\boldsymbol{V}^H \boldsymbol{\delta}_i\right)\right)\right]_j = \left[f_X(\boldsymbol{S})\boldsymbol{\delta}_i\right]_j = \left[\mathrm{Var}[\boldsymbol{X}]\boldsymbol{\delta}_i\right]_j \\ &= \left[\mathrm{Cov}[X_i, \boldsymbol{X}]\right]_j = \mathrm{Cov}[X_i, X_j] \quad .\end{aligned}$$

□

The graph spectral density plays the same role as the spectral density of weakly stationary random fields. Indeed, the covariance between a reference vertex i and any other vertex j can be expressed as the convolution between a covariance "signal" $\boldsymbol{C}$, defined as the inverse graph Fourier Transform of the spectral density and the Dirac signal at vertex i.

Remark 1.4.3. In their work, Perraudin and Vandergheynst (2017) actually use Equation (1.10) to define their notion of stationarity of graph signals, called graph wide-sense stationarity, in the particular case where the shift operator is the Graph Laplacian. It is therefore equivalent to our notion of $\boldsymbol{S}$-stationarity according to Proposition 1.4.6. They motivate their choice by explaining that they obtain a covariance that is defined by a global kernel function (our spectral density) and locally adapted to the structure of the graph to derive covariance between vertices using a convolution with a localized signal, the Dirac signal.

1.4.3 Comparison with other definitions of stationarity

In this section, we compare our definition of a stationary SGS, given in Definition 1.4.1 to existing definitions of stationarity, and discuss the underlying assumptions made by choosing ours.

Comparison with the work of T. Espinasse

In his work, Espinasse (2011) defines a notion of stationarity for a stochastic process indexed by the vertices of graph. It is based on the notion of invariant.

Definition 1.4.2. *Let S_n be the set of all permutations of $\{1, \dots, n\}$ and for $\sigma \in S_n$ denote $\boldsymbol{P}_\sigma$ the permutation matrix defined by $[\boldsymbol{P}_\sigma]_{ij} = \delta_{i\sigma(j)}$. In particular, $\boldsymbol{P}_\sigma$ is invertible and its inverse is $\boldsymbol{P}_\sigma^{-1} = \boldsymbol{M}_{\sigma^{-1}}$.*

An invariant is a function $\Phi : \mathrm{Dom}(\Phi) \subset \mathscr{M}_n(\mathcal{R}) \to \mathscr{M}_n(\mathcal{R})$ such that:

- $\forall \boldsymbol{A} \in \mathrm{Dom}(\Phi)$, $\boldsymbol{A}^T \in \mathrm{Dom}(\Phi)$ *and* $\Phi(\boldsymbol{A}^T) = \Phi(\boldsymbol{A})^T$
- $\forall \boldsymbol{A} \in \mathrm{Dom}(\Phi)$, $\forall \sigma \in S_n$, $\boldsymbol{P}_\sigma^{-1}\boldsymbol{A}\boldsymbol{P}_\sigma \in \mathrm{Dom}(\Phi)$ *and* $\Phi(\boldsymbol{P}_\sigma^{-1} A \boldsymbol{P}_\sigma) = \boldsymbol{P}_\sigma^{-1}\Phi(\boldsymbol{A})\boldsymbol{P}_\sigma$

The order of an invariant Φ is the smallest integer $r \geq 0$ such that $\forall \boldsymbol{A} \in \mathrm{Dom}(\Phi)$, $\forall i, j \in [\![1, n]\!]$, the value of $[\Phi(\boldsymbol{A})]_{ij}$ only depends on the elements $\{A_{kl} : k, l$ are within r hops from either i or $j\}$.

The notion of stationary SGS is then defined as follows.

Definition 1.4.3. *[(Espinasse, 2011, Definition 3.7.3)] A SGS $\boldsymbol{X}$ on a graph $\mathcal{G}$ with shift operator $\boldsymbol{S}$ is stationary of order r if its covariance matrix $\mathrm{Var}[\boldsymbol{X}]$ satisfies:*

- $\mathrm{Var}[\boldsymbol{X}]$ *is positive definite.*

- *There exists an invariant Φ of order r such that:*

$$\mathrm{Var}[\boldsymbol{X}] = \Phi(\boldsymbol{S}) \quad .$$

Similarly to the definition we introduced, this definition describes stationarity with respect to a choice of shift operator. Actually, the definition we provided falls into the scope of the definition proposed by Espinasse (2011). Simply notice that any polynomial of degree r is an invariant of order r for the set of symmetric matrices.

This definition ensures that the covariance between two pairs of vertices associated to two "large-enough" isomorphic parts of the graph stays the same, as stated by the following proposition.

Proposition 1.4.7. *Let (i_1, j_1) and (i_2, j_2) be two pairs of vertices of a graph $\mathcal{G}$ belonging to two isomorphic subsets $\mathcal{V}_1$ and $\mathcal{V}_2$ of vertices of $\mathcal{G}$ and such that i_2 (resp. j_2) is the image of i_1 (resp. j_1).*
Then for any process $\boldsymbol{X}$ stationary of order r in the sense of Definition 1.4.3, if $\mathcal{V}_1$ includes all vertices within r hops of either i_1 or j_1, then

$$\mathrm{Cov}[X_{i_1}, X_{j_1}] = \mathrm{Cov}[X_{i_2}, X_{j_2}] \quad .$$

Proof. Let us denote $\boldsymbol{W}$ the adjacency matrix of $\mathcal{G}$. Following Corollary 1.1.2, there exists a permutation σ such that $\sigma(i_1) = i_2$, $\sigma(j_1) = j_2$ and $\forall i, j \in \mathcal{V}_\infty$, $\mathcal{W}_{ij} = W_{\sigma(i)\sigma(j)} = [\boldsymbol{P}_\sigma^{-1}\boldsymbol{W}\boldsymbol{P}_\sigma]_{ij}$.

Then, given that $\boldsymbol{W}$ is shift operator, it follows from Definition 1.4.3 that

$$[\mathrm{Var}[\boldsymbol{X}]]_{i_2 j_2} = [\Phi(\boldsymbol{W})]_{i_2 j_2} = [\Phi(\boldsymbol{W})]_{\sigma(i_1)\sigma(j_1)} = [\boldsymbol{P}_\sigma^{-1}\Phi(\boldsymbol{W})\boldsymbol{P}_\sigma]_{i_1 j_1} \quad .$$

And using the fact that Φ is an invariant,

$$[\mathrm{Var}[\boldsymbol{X}]]_{i_2 j_2} = [\Phi(\boldsymbol{P}_\sigma^{-1}\boldsymbol{W}\boldsymbol{P}_\sigma)]_{i_1 j_1} \quad .$$

Hence, given that Φ is of order r, $[\mathrm{Var}[\boldsymbol{X}]]_{i_2 j_2}$ only depends on the elements of k, l of $\boldsymbol{P}_\sigma^{-1}\boldsymbol{W}\boldsymbol{P}_\sigma$ that are within r hops of $(i_1, j1)$. If $\mathcal{V}_1$ is large enough to include these vertices, then these elements are equal to those of $\boldsymbol{W}$ and therefore,

$$[\mathrm{Var}[\boldsymbol{X}]]_{i_2 j_2} = [\Phi(\boldsymbol{W})]_{i_1 j_1} = [\mathrm{Var}[\boldsymbol{X}]]_{i_1 j_1} \quad .$$

□

This result acts like a generalization for graphs of the invariance of the covariance of a stationary process by translation and symmetry, both being, similarly to graph isomorphisms, bijective transformations that preserve the structure of the objects they are applied to. This property is kept with the definition of $\boldsymbol{S}$-stationary we introduced as it is a particular case of Definition 1.4.3.

Comparison to the work of Marques et al.

In their work, Marques et al. (2017) provide three definitions of weak stationarity for a SGS.

Definition 1.4.4. [*(Marques et al., 2017, Definitions 1 and 2)*] *Let $\mathcal{G}$ be a n-graph with shift operator $\boldsymbol{S}$. A (zero-mean) SGS $\boldsymbol{X}$ is weakly stationary if it satisfies one the following requirement:*

1. *$\boldsymbol{X}$ can be written as $\boldsymbol{X} = h(\boldsymbol{S})\boldsymbol{W}$ for a graph filter $h(\boldsymbol{S})$ and a white signal $\boldsymbol{W}$.*
2. *$\mathrm{Var}[\boldsymbol{X}]$ and $\boldsymbol{S}$ are simultaneously diagonalizable.*
3. *For any integers $a, b, c, d \geq 0$ such that $a + b = c + d$:*

$$\mathbb{E}\left[(\boldsymbol{S}^a\boldsymbol{X})\left(\boldsymbol{S}^b\boldsymbol{X}\right)^T\right] = \mathbb{E}\left[(\boldsymbol{S}^c\boldsymbol{X})\left(\boldsymbol{S}^d\boldsymbol{X}\right)^T\right] \quad ,$$

or equivalently,

$$\boldsymbol{S}^a\mathrm{Var}[\boldsymbol{X}]\boldsymbol{S}^b = \boldsymbol{S}^c\mathrm{Var}[\boldsymbol{X}]\boldsymbol{S}^d \quad .$$

Note that we proved in Theorem 1.4.3 that Requirement 1 is actually equivalent to our definition of $\boldsymbol{S}$-stationary, thus linking our notion of stationarity to that of this new definition.

Marques et al. (2017) show that Requirements 2 and 3 are in fact equivalent, and that Requirement 1 implies 2 and 3. There is no equivalence between 1 and 2 as Requirement 1 implicitly imposes that the eigenspaces of $\boldsymbol{S}$ and $\mathrm{Var}[\boldsymbol{X}]$ must be the same, which is not generally the case if $\mathrm{Var}[\boldsymbol{X}]$ and $\boldsymbol{S}$ are just simultaneously diagonalizable. Indeed, let $\boldsymbol{v}^{(1)}$ and $\boldsymbol{v}^{(2)}$ be two orthogonal eigenvectors belonging to the same eigenspace of $\boldsymbol{S}$, associated to a duplicated eigenvalue λ. Then for Requirement 1 to be satisfied, the eigenvalues of $\mathrm{Var}[\boldsymbol{X}]$ associated to $\boldsymbol{v}^{(1)}$ and $\boldsymbol{v}^{(2)}$ must also be equal (to $h^2(\lambda)$) and therefore $\boldsymbol{v}^{(1)}$ and $\boldsymbol{v}^{(2)}$ are also in the same eigenspace of $\mathrm{Var}[\boldsymbol{X}]$.

Hence, defining stationarity through this Requirement 1 (or equivalently using our notion of $\boldsymbol{S}$-stationarity) yields a more restrictive notion than using the other two requirements. However, both definitions become in fact equivalent when $\boldsymbol{S}$ has no duplicated eigenvalue.

Besides, defining stationarity using Requirement 1 allows to keep the properties given by Requirements 2 and 3. In particular, Requirement 3 generalizes the invariance of the correlation operator by application of shifts by imposing that as long as the total number of times that a signal is shifted is constant, the covariance stays the same.

Comparison to the work of B. Girault

In his work, Girault (2015a) bases the definition of stationarity on an invariance of the covariance by translation, similarly as in Requirement 3 of Definition 1.4.4. But translations are now defined as the application of the following (complex) operator to a graph signal:

$$\boldsymbol{T}_S = \exp\left(-\mathrm{i}\frac{\pi}{\sqrt{\rho_S}}\sqrt{\boldsymbol{S}}\right) \quad ,$$

where only symmetric positive semi-definite shift operators $\boldsymbol{S}$ are considered and ρ_S is an upper bound on the eigenvalues of $\boldsymbol{S}$.

Contrary to the definitions based on the shift operator or Dirac signals, this definition has the particularity to conserve the energy of a graph signal which is defined as its norm. It is therefore an isometric operator. Stationarity is then defined as follows:

Definition 1.4.5. [*(Girault, 2015b, Definition 3)*] *Let $\mathcal{G}$ be a n-graph with shift operator $\boldsymbol{S}$. A (zero-mean) SGS $\boldsymbol{X}$ is wide-sense stationary if its covariance matrix satisfies*

$$\mathrm{Var}[\boldsymbol{X}] = \mathrm{Var}[\boldsymbol{T}_S\boldsymbol{X}] \quad .$$

Girault (2015b, Proposition 1) proves that wide-sense stationarity is equivalent to Requirement 2 of Definition 1.4.4. The same comparison with the notion of $\boldsymbol{S}$-stationarity therefore holds: both definitions are equivalent only if the eigenvalues of $\boldsymbol{S}$ are distinct. In the general case, $\boldsymbol{S}$-stationarity implies Definition 1.4.5 and therefore yields a more restrictive notion of stationarity.

1.4.4 A few words on the mean

Up until now, only zero-mean SGS were considered, i.e. SGS such that their mean vector is zero. However, it does not constitute a requirement for random fields to be weakly stationary. Indeed, for a random field to be stationary, its mean function should be constant. The natural counterpart of this requirement for graph signals would be to impose that the mean vector of a stationary SGS should be constant, meaning that there exists a constant m such that the expectation of a stationary SGS $\boldsymbol{Y}$ at any vertex i is $\mathbb{E}[Y_i] = m$. Hence a $\boldsymbol{S}$ stationary SGS $\boldsymbol{Y}$ would be defined as the sum of a constant vector $m\mathbf{1}$ and a stationary zero-mean SGS $\boldsymbol{X} = \boldsymbol{Y} - \mathbb{E}[\boldsymbol{Y}] = \boldsymbol{Y} - m\mathbf{1}$.

If we were to define stationary SGS like this, we would lose some of their properties, first of which being the preservation of stationary after filtering, as stated in Theorem 1.4.2. Indeed, the mean of the filtered signal $h(\boldsymbol{S})\boldsymbol{Y}$ is $mh(\boldsymbol{S})\mathbf{1}$ which is a constant signal if and only if $\mathbf{1}$ is an eigenvector of $\boldsymbol{S}$. This remark motivates the following definition of $\boldsymbol{S}$-stationary for signals that may not be zero-mean.

Definition 1.4.6. *A SGS $\boldsymbol{Y}$ is called $\boldsymbol{S}$-stationary if there exists an constant m such that:*

$$\boldsymbol{Y} - m\boldsymbol{v} \text{ is a zero-mean } \boldsymbol{S}\text{-stationary SGS,}$$

where $\boldsymbol{v}$ is an eigenvector of $\boldsymbol{S}$.

Note that whatever the choice of eigenvector $\boldsymbol{v}$ of $\boldsymbol{S}$, Theorem 1.4.2 is satisfied. Besides, in the particular case where $\boldsymbol{S}$ is the graph Laplacian, the constant signal $\mathbf{1}$ is an admissible candidate for $\boldsymbol{v}$ and therefore the mean of a stationary SGS can be considered as constant across the vertices.

Conclusion

In this chapter, we presented the mathematical framework we will use to work with variables indexed by the vertices of a (simple undirected) graph. Both the cases of deterministic and random graph signals were considered, and their respective frameworks of study were built using analogies with respectively classical signal processing theory and stochastic processes theory.

A key notion to keep in mind is that of shift operators, which are matrices aiming at representing the structure of the graph on which the signals are defined. These matrices are used to define all the key tools pertaining to both deterministic and stochastic graph signal processing. Indeed, on one hand, the graph Fourier transform but also convolutions and filtering of graph signals were all defined while relying on the eigendecomposition of a shift operator. On the other hand, the definition of stationary stochastic graph signals was also entirely based on a shift operator. The next chapter introduces practical algorithms, once again based on the shift operator, that will be used in the rest of our work.

Finally, we recall the working assumptions that will be assumed for in the remainder of this work (unless specified otherwise).

Assumption 1.1. *In this work, only connected simple undirected finite graphs are considered.*

Assumption 1.2. *Only real, symmetric shift operators $\boldsymbol{S}$ are considered.*
Consequently, $\boldsymbol{S}$ is diagonalizable by a unitary matrix and has real eigenvalues. Such a decomposition is denoted as follows:

$$\boldsymbol{S} = \boldsymbol{V} \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix} \boldsymbol{V}^H \quad ,$$

where

- *$\lambda_1 \leq \cdots \leq \lambda_n$ denote the real eigenvalues of $\boldsymbol{S}$, ordered in ascending order,*
- *$\boldsymbol{V} = \left[\boldsymbol{v}^{(1)} | \dots | \boldsymbol{v}^{(n)}\right]$ is a unitary matrix (i.e. $\boldsymbol{V}^{-1} = \boldsymbol{V}^H$)) whose columns $\boldsymbol{v}^{(1)}, \dots, \boldsymbol{v}^{(n)}$ form an orthonormal basis of $\mathbb{C}^n$ composed of eigenvectors of $\boldsymbol{S}$ such that:*

$$\forall i \in [\![1, n]\!], \quad \boldsymbol{S}\boldsymbol{v}^{(i)} = \lambda_i \boldsymbol{v}^{(i)} \quad .$$

Assumption 1.3. *Unless otherwise specified, the SGS considered in this work are zero-mean, i.e. $\mathbb{E}[\boldsymbol{X}] = \mathbf{0}$.*

2

Algorithmic toolbox for graph signal processing

Contents

Résumé

Le but de ce chapitre est d'apporter au lecteur une boite à outils d'algorithmes de traitement du signal sur graphe. Ces algorithmes sont tous basés sur des opérations de filtrage de signaux sur graphe. Ainsi, nous commençons par présenter et comparer différentes méthodes (exactes ou approchées) de filtrage de signaux sur graphe afin de motiver le choix qui est fait dans ce travail de ne recourir qu'à l'une d'entre elles: le filtrage par approximation polynômiale de Tchebychev (ou plus simplement "filtrage de Tchebychev"). Nous exposons ensuite l'utilisation de cet algorithme pour calculer trace, histogramme de valeur propres, log-déterminant et inverse de fonctions de matrices.

Introduction

In the previous chapter, the mathematical framework surrounding graph signal processing was put in place, while following a strict analogy with continuous and discrete signal processing. In particular, the notion of signal filtering on a graph $\mathcal{G}$ was introduced while relying on the definition of a matrix representation of $\mathcal{G}$ through a matrix called shift operator.

Much like in classical processing, filtering operations play a key role when processing graph signals. As we will later see in Chapters 3 and 4, algorithms aiming at simulating and estimating graph signals heavily rely on being able to compute efficient graph filtering operations. By efficient, we mean that the filtering algorithm should minimize both computational and storage costs when operated. The aim of this chapter is to introduce an approximate graph signal filtering algorithm, that we call *Chebyshev filtering*, and that will be used throughout the rest of this work.

The Chebyshev filtering algorithm has already been used for graph filtering purposes in the graph signal processing community (Hammond et al., 2011; Susnjara et al., 2015), and before that to compute approximations of matrix functions (Higham, 2008). The aim of this chapter really is to provide a rigorous justification of why it is the most appropriate algorithm in our context of application using arguments based on approximation theory and computational complexity, and also comparisons with other possible choices of algorithms[1].

We refer the reader to Appendix B for recalls on the theory of function approximation and interpolation, which are instrumental to graph filtering operations. In the first section of this chapter, we present and compare different approaches to graph filtering in order to motivate the use of Chebyshev filtering. This last algorithm is then introduced. Finally, applications of Chebyshev filtering to the computation of characteristics of graph filters are presented. They will play a key role when dealing with the inference of stochastic graph signals.

Throughout this chapter, let $h : \mathbb{R} \to \mathbb{R}$ be a transfer function and $h(\boldsymbol{S})$ be the associated graph filter with respect to a symmetric shift operator $\boldsymbol{S} \in \mathscr{M}_n(\mathbb{R})$ defined according to Assumption 1.2.

2.1 Exact algorithms for graph filtering

Let $\boldsymbol{x} \in \mathbb{R}^n$ denote a real graph signal on a graph associated with $\boldsymbol{S}$. Our goal is to filter $\boldsymbol{x}$ by the graph filter defined by h and $\boldsymbol{S}$, or equivalently evaluating the product $h(\boldsymbol{S})\boldsymbol{x}$. In this section, algorithms are derived to compute this product exactly. Two assumptions are made:

- Evaluating h on any real value is possible and achievable with a negligible computational complexity.
- The matrix $\boldsymbol{S}$ and the vector $\boldsymbol{x}$ are known and stored in memory.

The computational complexity of each proposed algorithm is derived as an order of magnitude for the count of floating-point operations performed by the algorithm. The memory requirements are also evaluated, and are defined as the amount of memory needed by the algorithm to store temporary variables used by the algorithm. They do not take into account the space used to store $\boldsymbol{S}$ and $\boldsymbol{x}$.

[1]Note however that we omit in this chapter any comparison with methods based on the Lanczos algorithm (Golub and Van Loan, 1996b, Chapter 9), as this case will be treated later in Section 3.3.

2.1.1 Filtering via eigendecomposition

A first solution to compute $h(\boldsymbol{S})\boldsymbol{x}$ consists in getting back to the definition of graph filters. Assuming that an orthonormal eigenbasis matrix $\boldsymbol{V}$ and the eigenvalues $\lambda_1, \ldots, \lambda_n$ of $\boldsymbol{S}$ have been computed and stored, the vector $h(\boldsymbol{S})\boldsymbol{x}$ can be expressed as

$$h(\boldsymbol{S})\boldsymbol{x} = \boldsymbol{V}\mathrm{Diag}(h(\lambda_1), \ldots, h(\lambda_n))\boldsymbol{V}^H\boldsymbol{x} \quad .$$

Computing the product $h(\boldsymbol{S})\boldsymbol{x}$ can be done in three steps: compute a graph Fourier transform (GFT) of $\boldsymbol{x}$, multiply the components of this vector by $h(\lambda_1), \ldots, h(\lambda_n)$ and take the inverse GFT of the result. This first approach is summed up by Algorithm 2.1.

Algorithm 2.1: Graph filtering via eigendecomposition.

Input: Shift operator $\boldsymbol{S} \in \mathscr{M}_n(\mathbb{R})$. Vector $\boldsymbol{x} \in \mathbb{R}^n$. Transfer function $h : \mathbb{R} \to \mathbb{R}$.
Output: The product $\boldsymbol{y} = h(\boldsymbol{S})\boldsymbol{x} \in \mathbb{R}^n$.

. . . . . .

Initialization: $\boldsymbol{y} = \boldsymbol{x}$;

1. Full eigendecomposition of $\boldsymbol{S}$: Use a diagonalization algorithm to compute the n eigenvalues $\lambda_1, \ldots, \lambda_n$ and an orthonormal eigenbasis $\boldsymbol{V} \in \mathscr{M}_n(\mathbb{C})$ of $\boldsymbol{S}$, and store them.
2. Graph Fourier transform: $\boldsymbol{y} \leftarrow \boldsymbol{V}^H\boldsymbol{y}$.
3. Frequency scaling: Compute a component-wise multiplication with the impulse response vector $(h(\lambda_1) \ldots h(\lambda_n))^T : \forall i \in [\![1, n]\!]$, $y_i \leftarrow h(\lambda_i)y_i$.
4. Inverse graph Fourier transform: $\boldsymbol{y} \leftarrow \boldsymbol{V}\boldsymbol{y}$

Return $\boldsymbol{y}$.

The computational bottleneck of this approach resides on its first step: the full diagonalization of the shift operator $\boldsymbol{S}$. Indeed, on one hand, the matrix $\boldsymbol{V}$ (or at least subroutines allowing to compute the products between $\boldsymbol{V}$ and a vector and between $\boldsymbol{V}^T$ and a vector) must be known to compute steps 2 and 4. On the other hand, all the eigenvalues of $\boldsymbol{S}$ must be known to compute the impulse response of the filter needed in step 3 from the expression of the transfer function h.

This full diagonalization of a $n \times n$ matrix is an expensive operation, computationally and memory-wise. $\mathcal{O}(n^3)$ operations are required to compute the full set of eigenpairs of a real symmetric matrix, using for instance the Jacobi method or the Householder tridiagonalization approach implemented in the LAPACK library (Press et al., 2007). And a storage space of order $\mathcal{O}(n^2)$ must be available to store the n vectors of size n that compose the eigenbasis $\boldsymbol{V}$ and the n eigenvalues. Such requirements become intractable as n grows as both the computational cost and the memory requirements would explode.

2.1.2 Particular case: Polynomial transfer function

A second solution for this graph filtering problem is based on the observation that in the particular case where the transfer function h is a polynomial of degree $K < n$ with coefficients $a_0, \ldots, a_K \in \mathbb{R}$, the corresponding graph filter $h(\boldsymbol{S})$ is a matrix polynomial defined by:

$$h(\boldsymbol{S}) = \sum_{k=0}^{K} a_k \boldsymbol{S}^k \quad .$$

Computing the product $h(\boldsymbol{S})\boldsymbol{x}$ can be done iteratively using Horner's scheme, as presented in Algorithm 2.2.

Algorithm 2.2 only involves products between $\boldsymbol{S}$ and various vectors: no costly factorization of the shift operator has to be applied first. In general, the computational cost of this algorithm will therefore be of order $\mathcal{O}(Kn^2)$ i.e. K times the cost of a matrix-vector product. However in the case where $\boldsymbol{S}$ is sparse, the cost of the matrix-vector product can be reduced to $\mathcal{O}(dn)$ where $d \ll n$ is the mean number of non-zero entries of $\boldsymbol{S}$ per row, thus yielding a graph filtering

Algorithm 2.2: Graph filtering with a polynomial transfer function.

Input: Shift operator $\boldsymbol{S} \in \mathscr{M}_n(\mathbb{R})$. Vector $\boldsymbol{x} \in \mathbb{R}^n$. Coefficients $a_0, \dots, a_K \in \mathbb{R}$.

Output: The product $\boldsymbol{y} = \left(\sum_{k=0}^{K} a_k \boldsymbol{S}^k \right) \boldsymbol{x} \in \mathbb{R}^n$.

Initialization: $\boldsymbol{y} = a_K \boldsymbol{x}$;
if $K > 0$ **then**
  for k **from** $K-1$ **to** 0 **do**
    $\boldsymbol{y} \leftarrow a_k \boldsymbol{x} + \boldsymbol{S}\boldsymbol{y}$;
Return $\boldsymbol{y}$.

algorithm with computational complexity $\mathcal{O}(Kdn)$. As for the storage requirements of this algorithm, they are of order n to store the temporary vector $\boldsymbol{y}$. Hence, they actually depend neither on the shift operator, nor on the transfer function (assuming both are already known and/or stored).

This last property is particularly interesting when considering the scalability of the algorithm: as long as $\boldsymbol{S}$ and small (fixed) number of vectors can be stored, Algorithm 2.2 can be used for graph filtering. This was not the case with Algorithm 2.1. Moreover, if polynomials with degree $K \ll n$ are considered, the increase of computational costs with the size n can be kept under control as they grow at most quadratically with n. The same growth rate is cubic when using Algorithm 2.1.

2.1.3 General case: Polynomial interpolation of graph filters

What about the case when h is not a polynomial function? According to Theorem 1.3.1, any graph filter $h(\boldsymbol{S})$ can be expressed as a filter whose transfer function is a polynomial P_h of degree at most $n-1$. Computing the product $h(\boldsymbol{S})$ could therefore be done using Algorithm 2.2 with P_h. This approach supposes that the analytical expression of P_h was first derived from the sole knowledge of h and $\boldsymbol{S}$. This is possible as P_h is the unique polynomial of degree at most $n-1$ interpolating h at eigenvalues $\lambda_1, \dots, \lambda_n \in \mathbb{R}$ of $\boldsymbol{S}$.

Hence, to find P_h, the full set of eigenvalues of $\lambda_1, \dots, \lambda_n$ of $\boldsymbol{S}$ must first be computed. This represents once again a rather costly step as $\mathcal{O}(n^3)$ operations are required. However, contrary to the full diagonalization approach of Algorithm 2.1, there is no need to compute and store the eigenbasis of $\boldsymbol{S}$: only the eigenvalues are needed. Less operations are in fact needed (even though the number is still of order $\mathcal{O}(n^3)$) and the storage space needs are brought down to $\mathcal{O}(n)$ using for instance a Lanczos method for the computation of eigenvalues (Press et al., 2007).

Once the eigenvalues $\lambda_1, \dots, \lambda_n$ of $\boldsymbol{S}$ are computed, the interpolating polynomial P_h can be obtained using one of the methods presented in Appendix B.1.

Keeping in mind that the expression of P_h is computed to be used in Algorithm 2.2, the Vandermonde approach seems to be the way to go as it provides directly the monomial coefficients of P_h. However a linear system that involves a (full) Vandermonde matrix of size n must be solved to compute these coefficients, which can be done in $\mathcal{O}(n^2)$ operations while requiring a storage space of $\mathcal{O}(n)$. Besides, this system is known to be numerically unstable as it becomes more and more ill-conditioned as n grows (Atkinson, 1989).

This last drawback is no longer a concern if the Newton approach is used. Indeed, computing the coefficients of P_h in the Newton polynomial basis can be done by either solving the triangular system in Equation (B.3) or using a divided-differences approach (Atkinson, 1989). Both algorithms are numerically stable and provide an exact solution in $\mathcal{O}(n^2)$ operations while requiring storage needs of order $\mathcal{O}(n)$. Then, evaluating the product $P_h(\boldsymbol{S})\boldsymbol{x}$ can be done directly with the Newton expansion of P_h by slightly modifying Horner's scheme of Algorithm 2.2, as presented in Algorithm 2.3.

Finally, one can notice that the Lagrange approach offers the desirable advantage to require no additional computations to get an expression for P_h. However, evaluating the product $P_h(\boldsymbol{S})\boldsymbol{x}$ using Equation (B.4) is less straightforward than with the other two approaches. Indeed, computing the monomial coefficients of P_h from Equation (B.4) in order to use Algorithm 2.2 requires $\mathcal{O}(n^3)$ operations as each term of the sum must be expanded first. A less expensive alternative consists in using Equation (B.4) directly to compute the product $P_h(\boldsymbol{S})\boldsymbol{x}$. Indeed, each term

Algorithm 2.3: Graph filtering with a Newton polynomial transfer function.

Input: Shift operator $\boldsymbol{S} \in \mathscr{M}_n(\mathbb{R})$. Vector $\boldsymbol{x} \in \mathbb{R}^n$. A family of interpolation points $\lambda_1, \dots, \lambda_K$ defining a Newton basis $\{\eta_k\}_{1 \leq k \leq K-1}$. Coefficients $c_0, \dots, c_{K-1} \in \mathbb{R}$.

Output: The product $\boldsymbol{y} = \left(\sum_{k=0}^{K-1} c_k \eta_k(\boldsymbol{S}) \right) \boldsymbol{x} \in \mathbb{R}^n$.

Initialization: $\boldsymbol{y} = a_{K-1}\boldsymbol{x}$;
if $K > 1$ **then**
  for k **from** $K-2$ **to** 0 **do**
    $\boldsymbol{y} \leftarrow a_k \boldsymbol{x} + \boldsymbol{S}\boldsymbol{y} - \lambda_k \boldsymbol{y}$;
Return $\boldsymbol{y}$.

of the sum can be computed using n nested multiplications, much like Horner's scheme. The resulting method is outlined in Algorithm 2.4. However, it comes at a computational cost of order $\mathcal{O}(n^2)$ as n products of n (shifted) monomials must be evaluated. On the other hand, storage needs of only $\mathcal{O}(n)$ are required.

Algorithm 2.4: Graph filtering with a Lagrange polynomial transfer function.

Input: Shift operator $\boldsymbol{S} \in \mathscr{M}_n(\mathbb{R})$. Vector $\boldsymbol{x} \in \mathbb{R}^n$. A family of interpolation points $\lambda_1, \dots, \lambda_K$ defining a Lagrange basis $\{l_k\}_{1 \leq k \leq K}$. Coefficients $h_1, \dots, h_k \in \mathbb{R}$.

Output: The product $\boldsymbol{y} = \left(\sum_{k=1}^{K} h_k l_k(\boldsymbol{S}) \right) \boldsymbol{x} \in \mathbb{R}^n$.

Initialization: $\boldsymbol{u} = \boldsymbol{0}$, $\boldsymbol{y} = \boldsymbol{0}$;
for k **from** 1 **to** K **do**
  $\boldsymbol{u} \leftarrow \boldsymbol{x}$;
  for j **from** 1 **to** K, $j \neq k$ **do**
    $\boldsymbol{u} \leftarrow \boldsymbol{S}\boldsymbol{u} - \lambda_k \boldsymbol{u}$;
  $\boldsymbol{y} \leftarrow \boldsymbol{y} + h_k \boldsymbol{u}$
Return $\boldsymbol{y}$.

2.1.4 Graph filtering via polynomial interpolation

Algorithm 2.5 sums up the general approach to graph filtering using interpolating polynomials. First, the full set of eigenvalues of $\boldsymbol{S}$ is computed. Then, an expression of the polynomial interpolating h at these eigenvalues is calculated. Finally, depending on the expression chosen at the previous step, the product $P_h(\boldsymbol{S})\boldsymbol{x}$ is computed using an iterative algorithm requiring a number of operations proportional to the size of the vectors n and the degree of the polynomial.

Algorithm 2.5: Graph filtering via polynomial interpolation.

Input: Shift operator $\boldsymbol{S} \in \mathscr{M}_n(\mathbb{R})$. Vector $\boldsymbol{x} \in \mathbb{R}^n$. Transfer function $h : \mathbb{R} \to \mathbb{R}$.
Output: The product $\boldsymbol{y} = h(\boldsymbol{S})\boldsymbol{x} \in \mathbb{R}^n$.

Initialization: $\boldsymbol{y} = \boldsymbol{x}$;

1. Eigenvalues of $\boldsymbol{S}$: Use a diagonalization algorithm to compute and store the n eigenvalues $\lambda_1, \dots, \lambda_n$ of $\boldsymbol{S}$.
2. Compute an expression of the polynomial P_h interpolating h at $\lambda_1, \dots, \lambda_n$ using either the Vandermonde, the Newton or the Lagrange approach.
3. According to the expression of P_h chosen at step 2, compute the product $\boldsymbol{y} = P_h(\boldsymbol{S})\boldsymbol{x}$ using either Algorithm 2.2, Algorithm 2.3 or Algorithm 2.4.

Return $\boldsymbol{y}$.

	Polynomial case	Full Diagonalization	Polynomial interpolation		
			Vandermonde	Newton	Legendre
Description	Algorithm 2.2	Algorithm 2.1	Algorithm 2.5		
Eigendecomposition	-	$\mathcal{O}(n^3)$	$\mathcal{O}(n^3)$		
Polynomial coefficients	0	-	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2)$	0
Product computations	$\mathcal{O}(Kdn)$	$\mathcal{O}(n^2)$	$\mathcal{O}(Kdn)$	$\mathcal{O}(Kdn)$	$\mathcal{O}(dn^2)$
Storage needs	$\mathcal{O}(n)$	$\mathcal{O}(n^2)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$

Table 2.1: Comparison of exact algorithms for graph filtering of a vector of size n. For methods involving a polynomial, its degree is denoted K (except for the Legendre approach, which has a polynomial of degree n).

2.1.5 Comparison of exact graph filtering algorithms

Table 2.1 provides a comparison of the computational and storage costs associated to the exact graph filtering algorithms presented up until now. The full diagonalization method of Algorithm 2.1 is compared to the polynomial interpolation method of Algorithm 2.5 and its three variants (namely, the choice of the Vandermonde, the Newton or the Legendre approach to express the interpolating polynomial).

The main computational bottleneck shared by these methods is the diagonalization of the matrix $\boldsymbol{S}$, which scales cubically with the size of the vectors n. Once this diagonalization step is performed, the full diagonalization approach offers the fastest way to evaluate the product $h(\boldsymbol{S})\boldsymbol{x}$. Indeed, polynomial interpolation approaches require either the computation of the coefficients of the polynomial or a tedious evaluation by nested multiplications. On the other hand, polynomial interpolation approaches require much less storage space than the full diagonalization approach given that the eigenbasis need not to be stored.

The particular case where the transfer function is polynomial yields the lowest overall computational and storage requirements. Contrary to the polynomial interpolation approach, there is no additional cost due the computation of interpolation points or more generally the diagonalization of $\boldsymbol{S}$. This motivates a new approach to solving the efficient graph filtering problem, namely finding a polynomial P_h such that:

- Computing its expression will not require any costly preliminary operations as it is the case with interpolation polynomials.
- Computing the product $P_h(\boldsymbol{S})\boldsymbol{x}$ can be done using an iterative scheme similar to those introduced in Algorithms 2.2 and 2.3.
- The products approximate well the product $h(\boldsymbol{S})\boldsymbol{x}$ in some sense to be defined.

Hence we aim at replacing the polynomial interpolation of h by its polynomial approximation, hoping that the loss of accuracy will be compensated by the gains in computational efficiency of the algorithm. This approach is presented in the next section.

2.2 Approximate algorithm for graph filtering: the Chebyshev algorithm

Following the considerations from the previous section, the idea is now to replace the costly exact computation of the product $h(\boldsymbol{S})\boldsymbol{x}$ by that of a polynomial filter $P_h(\boldsymbol{S})\boldsymbol{x}$ such that $P_h(\boldsymbol{S})\boldsymbol{x} \approx h(\boldsymbol{S})\boldsymbol{x}$ in some sense. In particular, P_h should be computed with minimal effort compared to the diagonalization step that was preliminary to all the exact methods.

2.2.1 Derivation of the algorithm

The steps leading to the Chebyshev filtering algorithm are now outlined.

Computation of the approximation error

Let us first focus on the discrepancy between $h(\boldsymbol{S})\boldsymbol{x}$ and its approximation by $P_h(\boldsymbol{S})\boldsymbol{x}$, also referred to as approximation error. Both being vectors of $\mathbb{R}^n$, it is naturally measured by the distance separating them in $\mathbb{R}^n$. This distance can be defined by any norm on $\mathbb{R}^n$. Actually, the choice of a norm is not important given that they are all equivalent in finite dimensional spaces, i.e. for any norms $\mathcal{N}_1, \mathcal{N}_2$ defined on $\mathbb{R}^n$, there exists two constants $C_1, C_2 > 0$ such that

$$\forall \boldsymbol{x} \in \mathbb{R}^n, \quad C_1\mathcal{N}_1(\boldsymbol{x}) \leq \mathcal{N}_2(\boldsymbol{x}) \leq C_2\mathcal{N}_1(\boldsymbol{x}) \quad .$$

In particular, for the Euclidean norm, the approximation error is:

$$\begin{aligned}\|h(\boldsymbol{S})\boldsymbol{x} - P_h(\boldsymbol{S})\boldsymbol{x}\|_2^2 &= \|\left(h(\boldsymbol{S}) - P_h(\boldsymbol{S})\right)\boldsymbol{x}\|_2^2 = \boldsymbol{x}^T\left(h(\boldsymbol{S}) - P_h(\boldsymbol{S})\right)^2\boldsymbol{x} \\ &= \frac{\boldsymbol{x}^T\left(h(\boldsymbol{S}) - P_h(\boldsymbol{S})\right)^2\boldsymbol{x}}{\boldsymbol{x}^T\boldsymbol{x}} \times (\boldsymbol{x}^T\boldsymbol{x}) = \mathcal{R}((h(\boldsymbol{S}) - P_h(\boldsymbol{S}))^2, \boldsymbol{x})\|\boldsymbol{x}\|_2^2 \quad ,\end{aligned}$$

where the notation $\mathcal{R}(\boldsymbol{M}, \boldsymbol{x})$ denotes the Rayleigh quotient of a Hermitian matrix $\boldsymbol{M}$ and a vector $\boldsymbol{x}$ (cf. Appendix A.2.1). Given that both $h(\boldsymbol{S})$ and $P_h(\boldsymbol{S})$ are graph filters with respect to the same shift operator $\boldsymbol{S}$, it is straightforward to check that $\left(h(\boldsymbol{S}) - P_h(\boldsymbol{S})\right)^2$ is also a graph filter with respect to $\boldsymbol{S}$ and that its eigenvalues are $(h(\lambda_1) - P_h(\lambda_1))^2, \ldots, (h(\lambda_n) - P_h(\lambda_n))^2$. Hence,

$$\min_{k\in[\![1,n]\!]}(h(\lambda_k) - P_h(\lambda_k))^2 \leq \mathcal{R}((h(\boldsymbol{S}) - P_h(\boldsymbol{S}))^2, \boldsymbol{x}) \leq \max_{k\in[\![1,n]\!]}(h(\lambda_k) - P_h(\lambda_k))^2 \quad .$$

Therefore,

$$\|h(\boldsymbol{S})\boldsymbol{x} - P_h(\boldsymbol{S})\boldsymbol{x}\|_2 \leq \left(\max_{k\in[\![1,n]\!]}|h(\lambda_k) - P_h(\lambda_k)|\right)\|\boldsymbol{x}\|_2 \quad . \tag{2.1}$$

This proves a rather intuitive result: for $P_h(\boldsymbol{S})\boldsymbol{x}$ to approximate well $h(\boldsymbol{S})\boldsymbol{x}$, it suffices that the function P_h approximates h well. More precisely, it suffices that the values of P_h are close to that of h on the set of eigenvalues $\lambda_1, \ldots, \lambda_n$ of $\boldsymbol{S}$. In particular, if these values coincide, exact graph filtering by polynomial interpolation of Section 2.1.3 is retrieved as we get $\|h(\boldsymbol{S})\boldsymbol{x} - P_h(\boldsymbol{S})\boldsymbol{x}\|_2 = 0$ and hence $h(\boldsymbol{S})\boldsymbol{x} = P_h(\boldsymbol{S})\boldsymbol{x}$.

Choice of the polynomial approximation

Assume now that some approximation error is tolerated, i.e. we want for some threshold $\epsilon_0 > 0$:

$$\|h(\boldsymbol{S})\boldsymbol{x} - P_h(\boldsymbol{S})\boldsymbol{x}\|_2 \leq \epsilon_0 \quad .$$

Then, following Equation (2.1), this condition can be enforced by imposing

$$\max_{k\in[\![1,n]\!]}|h(\lambda_k) - P_h(\lambda_k)| \leq \epsilon(\boldsymbol{x}), \quad \text{where } \epsilon(\boldsymbol{x}) = \epsilon_0/\|\boldsymbol{x}\|_2 > 0 \quad . \tag{2.2}$$

Comparing directly the values of $\{h(\lambda_k) : k \in [\![1,n]\!]\}$ and $\{P_h(\lambda_k) : k \in [\![1,n]\!]\}$ to make sure this last condition is satisfied would lead to the same problem as the one encountered in the interpolation approach: namely, the values of all the eigenvalues of $\boldsymbol{S}$ must be known and therefore $\boldsymbol{S}$ must be fully diagonalized.

However, in the context of approximation, a sufficient condition to get Equation (2.2) is if $\max_{\lambda\in[a,b]}|h(\lambda) - P_h(\lambda)| \leq \epsilon(\boldsymbol{x})$, where the interval $[a,b]$ is such that $\lambda_1, \ldots, \lambda_n \in [a,b]$. Hence, the enforcement of the condition in Equation (2.2) can be replaced by

$$\max_{\lambda\in[a,b]}|h(\lambda) - P_h(\lambda)| \leq \epsilon(\boldsymbol{x}), \quad \text{where } \epsilon(\boldsymbol{x}) = \epsilon_0/\|\boldsymbol{x}\|_2 > 0 \text{ and } \lambda_1, \ldots, \lambda_n \in [a,b] \quad . \tag{2.3}$$

Finding a polynomial approximation P_h of a function h over a segment $[a,b]$ can be done very efficiently using *Chebyshev sums* as described in details and justified in Appendix B.4. The first step consists in moving the approximation problem from the interval $[a,b]$ to $[-1,1]$. This is done by considering the (invertible) affine transform $\phi_{a,b}$ defined by

$$\phi_{a,b} : t \in [a,b] \mapsto \frac{2}{b-a}(t-a) - 1 \in [-1,1] \quad , \tag{2.4}$$

and whose inverse is the linear mapping $\phi_{a,b}^{-1}$ defined by

$$\phi_{a,b}^{-1} : t \in [-1,1] \mapsto a + \frac{b-a}{2}(t+1) \in [a,b] \quad . \tag{2.5}$$

Hence, to approximate h over $[a,b]$, we find a polynomial approximation $P_{\hat{h}}$ of the function

$$\hat{h} := h \circ \phi_{a,b}^{-1}$$

over $[-1,1]$ and return the polynomial

$$P_h := P_{\hat{h}} \circ \phi_{a,b} \quad .$$

Using Chebyshev sums, the polynomial $P_{\hat{h}}$ is given as the truncation at a given order of approximation $m \in \mathbb{N}$ of the Chebyshev series of $\hat{h}$. It is therefore written

$$P_{\hat{h}} = \frac{1}{2}c_0 T_0 + \sum_{k=1}^{m} c_k T_k \quad ,$$

where T_k denotes the k-th Chebyshev polynomial, and each coefficient c_k, $k \in [\![0,m]\!]$ is given by

$$c_k = \frac{2}{\pi}\int_0^\pi \hat{h}(\cos\theta)\cos(k\theta)d\theta, \quad k \in [\![0,m]\!] \quad . \tag{2.6}$$

These coefficients can be numerically computed either the Fast Fourier transform algorithm (Cooley and Tukey, 1965) or an algorithm designed to compute the discrete cosine transform (Chen et al., 1977; Makhoul, 1980) of a vector, as detailed in Algorithms B.1 and B.2. As for the order of the polynomial approximation m, it should be chosen to ensure Equation (2.3). Checking whether an order approximation m is large enough can be done numerically by evaluating the difference between the resulting polynomial approximation P_h and h over a fine discretization of $[a,b]$.

Remark 2.2.1. A restriction on the regularity of h over $[a,b]$ must be considered to safely apply the Chebyshev polynomial approximation: namely h should be at least of bounded variation (cf. Definition B.3.1) or Dini-Lipschitz continuous (cf. Definition B.3.2), so that any level of approximation error can be achieved by increasing the order m of the polynomial approximation (cf. Theorem B.4.4).

A method to deal with discontinuous functions is introduced in Appendix B.4.6.

Interval of approximation

The only remaining question is whether finding an interval $[a,b]$ containing all the eigenvalues of $\boldsymbol{S}$ is possible without actually computing the eigenvalues or having recourse to operations with similar computational complexities. The answer is yes and the following results provide examples of such intervals.

Proposition 2.2.1. *Let $\boldsymbol{S} \in \mathscr{M}_n(\mathbb{R})$ be a symmetric matrix and denote $\lambda_1, \ldots, \lambda_n$ its eigenvalues. Then,*

$$\forall i \in [\![1,n]\!], \quad |\lambda_i| \leq \sqrt{\mathrm{Trace}\,(\boldsymbol{S}^2)} \quad .$$

Hence, all the eigenvalues of $\boldsymbol{S}$ are contained in the interval $\left[-\sqrt{\mathrm{Trace}\,(\boldsymbol{S}^2)}, \sqrt{\mathrm{Trace}\,(\boldsymbol{S}^2)}\right]$.

Proof. This is a direct consequence of the fact that $\boldsymbol{S}^2$ has eigenvalues $\lambda_1^2, \ldots, \lambda_n^2$ and that therefore $\mathrm{Trace}\left(\boldsymbol{S}^2\right) = \sum_{j=1}^n \lambda_j^2$. □

Theorem 2.2.2 (Gerschgorin circle theorem (Gerschgorin, 1931))**.** *Any eigenvalue λ of a symmetric matrix $\boldsymbol{S} \in \mathscr{M}_n(\mathbb{R})$ satisfies:*

$$\lambda \in \bigcup_{i \in [\![1,n]\!]} [S_{ii} - r_i, S_{ii} + r_i], \qquad \textit{where } r_i = \sum_{j \neq i} |S_{ij}| \quad .$$

Hence, all the eigenvalues of $\boldsymbol{S}$ are contained in the interval $\left[\min_{i \in [\![1,n]\!]}(S_{ii} - r_i), \max_{i \in [\![1,n]\!]}(S_{ii} + r_i)\right]$.

Both Proposition 2.2.1 and Theorem 2.2.2 provide expressions of intervals containing the eigenvalues of the shift operator that can be computed with a limited complexity. Indeed, in the former case, given that $\boldsymbol{S}$ is real and symmetric, the trace of its square is equal to the sum of the square of all its elements:

$$\mathrm{Trace}(\boldsymbol{S}^2) = \sum_{j=1}^{n}\sum_{k=1}^{n} S_{ij}^2 \quad .$$

Hence, it can be computed using $\mathcal{O}(dn)$ operations, where d is at most n (when $\boldsymbol{S}$ is a full matrix). The same computational complexity can be derived in the latter case.

Remark 2.2.2. Finer intervals can be derived by using additional characteristics the shift operator may have. For instance, if $\boldsymbol{S}$ is positive (semi)-definite, then 0 is a lower bound of its eigenvalues. Consequently, the intervals proposed in Proposition 2.2.1 and Theorem 2.2.2 can be taken as

$$\left[0, \sqrt{\mathrm{Trace}\left(\boldsymbol{S}^2\right)}\right]$$

and

$$\left[0, \max_{i\in[\![1,n]\!]}(S_{ii}+r_i)\right] = \left[0, \max_{i\in[\![1,n]\!]}\sum_{j=1}^{n}|S_{ij}|\right]$$

(given that the diagonal elements of $\boldsymbol{S}$ would then be non-negative).

2.2.2 Presentation of the algorithm

At this point, the approximating polynomial P_h is expressed as a Chebyshev sum and its coefficients are computed. Computing the product $P_h(\boldsymbol{S})\boldsymbol{x}$ can be done iteratively by relying on the recurrence relation between Chebyshev polynomials described in Equation (B.11). The corresponding procedure is outlined in Algorithm 2.6.

Algorithm 2.6: Graph filtering of a Chebyshev sum.

Input: Shift operator $\boldsymbol{S} \in \mathscr{M}_n(\mathbb{R})$. Vector $\boldsymbol{x} \in \mathbb{R}^n$. A set of Chebyshev coefficients $c_0, \ldots, c_m \in \mathbb{R}$.

Output: The product $\boldsymbol{y} = \left(\frac{1}{2}c_0T_0(\boldsymbol{S}) + \sum_{k=1}^{m} c_kT_k(\boldsymbol{S})\right)\boldsymbol{x} \in \mathbb{R}^n$.

Initialization: $\boldsymbol{u}^{(-2)} = \boldsymbol{u}^{(-1)} = \boldsymbol{u} = \boldsymbol{y} = \boldsymbol{0}$;
for k **from** 0 **to** m **do**
  if $k = 0$ **then**
    $\boldsymbol{u} \leftarrow \frac{1}{2}\boldsymbol{x}$;
  else if $k = 1$ **then**
    $\boldsymbol{u} \leftarrow \boldsymbol{S}\boldsymbol{x}$
  else
    $\boldsymbol{u} \leftarrow 2\boldsymbol{S}\boldsymbol{u}^{(-1)} - \boldsymbol{u}^{(-2)}$;
  $\boldsymbol{y} \leftarrow \boldsymbol{y} + c_k\boldsymbol{u}$;
  $\boldsymbol{u}^{(-2)} \leftarrow \boldsymbol{u}^{(-1)}$;
  $\boldsymbol{u}^{(-1)} \leftarrow \boldsymbol{u}$;
Return $\boldsymbol{y}$.

To sum things up, approximate graph filtering is performed in three steps. First, an interval $[a, b]$ that contains all the eigenvalues is derived. Then a polynomial approximation of the transfer function of the filter over $[a, b]$ is derived using Chebyshev sums. Finally, the filtering operation is applied to this polynomial instead of the original transfer function, using an iterative method that only involves matrix products by the shift operator. This approach is outlined in Algorithm 2.7.

Two remarks on the outline of Algorithm 2.7 can be formulated. First, in most applications considered in this work, the transfer functions h are smooth enough so that an order of approximation of at most 10^3 are sufficient to yield almost-zero approximation errors. Second, running

Algorithm 2.7: Chebyshev filtering algorithm for graph signals.

Parameters: Order of discretization N of integrals in Algorithm B.1 or B.2
Input: Shift operator $\boldsymbol{S} \in \mathscr{M}_n(\mathbb{R})$. Vector $\boldsymbol{x} \in \mathbb{R}^n$. Transfer function $h : \mathbb{R} \to \mathbb{R}$. Approximation order $m \in \mathbb{N}$.
Output: An approximation of $\boldsymbol{h}(\boldsymbol{S})\boldsymbol{x}$.

Initialization: $\boldsymbol{y} = \boldsymbol{0}$;

1. Approximation interval:
   Find an interval $[a, b]$ that contains all the eigenvalues of $\boldsymbol{S}$. Examples are provided by Proposition 2.2.1 and Theorem 2.2.2.

2. Coefficients of the Chebyshev sum:
   Using Algorithm B.1 or B.2, compute the coefficients of the Chebyshev sum of order m of the function $t \in [-1, 1] \mapsto h(\phi_{a,b}^{-1}(t))$ where $\phi_{a,b}^{-1}(t)$ denotes the linear mapping from $[-1, 1]$ to $[a, b]$ (cf. Equation (2.5)).

3. Filtering:
   Use Algorithm 2.6 with the coefficients obtained at the previous step and using $\phi_{a,b}(\boldsymbol{S}) = \frac{2}{b-a}\boldsymbol{S} - \frac{b+a}{b-a}\boldsymbol{I}$ as shift operator on the vector $\boldsymbol{x}$.
   Store the result in $\boldsymbol{y}$.

Return $\boldsymbol{y}$.

Algorithm 2.6 with $\phi_{a,b}(\boldsymbol{S})$ as a shift operator can be done without having to actually compute (and store) this matrix. Indeed, the only requirement this algorithm has for the shift operator is the ability to compute its product with a n-vector. Yet, the product between $\phi_{a,b}(\boldsymbol{S})$ and a n-vector $\boldsymbol{u}$ can be written:

$$\phi_{a,b}(\boldsymbol{S})\boldsymbol{u} = \frac{2}{b-a}\boldsymbol{S}\boldsymbol{u} - \frac{b+a}{b-a}\boldsymbol{u} \tag{2.7}$$

Hence, any product by the shift operator in Algorithm 2.6 can effectively be replaced by the combination of a product by $\boldsymbol{S}$ and a subtraction given by Equation (2.7).

Following from this last remark, the approach to graph filtering of Algorithm 2.7 can be seen as "matrix-free" algorithm. Indeed, it does not actually require the shift operator to be stored in memory. Rather, it relies solely on being able to compute a product between the shift operator and vectors. Hence, if all that was available was a function that computed this product (without necessarily using a matrix stored in memory), the same would still apply.

This property is clearly desirable in a context where the size of the vectors and matrices may be so large that any gain in memory is appreciated. In that case, exploiting the structure of the shift operator to only keep in memory the values necessary to compute the matrix-vector product may bring great savings in storage space. This is for instance the case for circulant matrices for which just a few entries are necessary to compute a product with a vector.

2.2.3 Computational complexity of the algorithm

The computational complexity of Algorithm 2.7 is now explicitly calculated. Denote n_{nz} the number of non-zero entries of $\boldsymbol{S}$ and d the mean number of non-zero entries of a row of $\boldsymbol{S}$: $n_{nz} = d \times n$. Denote m the order of the Chebyshev approximation. The cost associated with each step (ignoring additions and multiplications by non-stored zeros) is described as follows:

- Step 1 requires $\mathcal{O}(dn)$ operations as mentioned earlier.
- Step 2 requires to apply fast Fourier transform or the discrete cosine transform algorithm to a vector of length N. The cost of this operation is $\mathcal{O}(N \log N)$ (Chen et al., 1977; Makhoul, 1980).
- Step 3 is composed of
  - $m + 1$ updates of $\boldsymbol{y}$ that consists in multiplying the entries of a n-vector by a scalar and adding them to another n-vector $\to m \times 2n$ operations

- m updates of the vector $\boldsymbol{u}$ that consists in multiplying a n-vector by $\phi_{a,b}(\boldsymbol{S})$ and subtracting another n-vector to the result ($\rightarrow n$ operations). Each product by $\phi_{a,b}(\boldsymbol{S})$ actually corresponds to a product by $\boldsymbol{S}$ ($\rightarrow dn$ operations) that is scaled by constant ($\rightarrow n$ operations) and followed by the subtraction ($\rightarrow n$ operations) of a n-vector that was also scaled by a constant ($\rightarrow n$ operations): $\rightarrow m \times (dn + 4n)$ operations.

Therefore, the overall cost of the Chebyshev filtering algorithm is $\mathcal{O}(mdn + N \log N)$ operations. Considering that in most of our applications[2] $N \ll n$, we conclude that the actual complexity of Algorithm 2.7 is of order $\mathcal{O}(mdn)$.

And regarding the storage needs, aside from $\boldsymbol{S}$, $\boldsymbol{x}$ and the coefficients which are assumed to be stored by default, the algorithm only needs enough space to work with 4 additional n-vectors. The storage needs of this algorithm are therefore of order $\mathcal{O}(n)$.

In conclusion, Algorithm 2.7 provides a solution to perform graph filtering with a computational and storage costs of the same order of the minimal case of polynomial filtering that was introduced in Algorithm 2.2. Moreover, the user can trade computational time for accuracy of the approximation using a single parameter: the order m of the Chebyshev sum. Indeed, asymptotically (when m grows to ∞), the approximation error of the Chebyshev sums goes to zero, and therefore, so does the approximation error of the vectors obtained using the Chebyshev filtering algorithm (cf. Equation (2.1)).

In the remaining of this work, the following assumption is made so that Chebyshev filtering can be applied.

Assumption 2.1. *Whenever a graph filter is considered, the associated transfer function is assumed to be regular enough for its Chebyshev series to converge over an interval* $[a, b]$ *containing all the eigenvalues of the shift operator.*

In practice, we will assume the transfer function to be Dini-Lipschitz continuous or continuous of bounded variation.

2.3 Applications of the Chebyshev filtering algorithm

In this section, a few useful algorithms designed to compute the trace, the log-determinant and the histogram of eigenvalues of a graph filter $h(\boldsymbol{S})$ defined by a shift operator following Assumption 1.2 and a transfer function h are presented. These algorithms will be particularly useful when the inference of stochastic graph signals will be considered in Chapter 5, allowing for instance to compute the likelihood of realizations of stochastic graph signals.

All the algorithms introduced in this section rely on the Chebyshev filtering algorithm, and aim at computing accurate estimates of some characteristics of a graph filter $h(\boldsymbol{S})$ in a matrix-free approach. The need to use this approach comes from the fact we want to avoid actually building and storing the graph filter $h(\boldsymbol{S})$, due to the high computational and storage costs associated. Direct methods are therefore out of the question.

Remark 2.3.1. All the algorithms presented in this section can actually be applied to draw estimates of these same characteristics for any real symmetric matrix $\boldsymbol{M}$ using the following trick: we take $\boldsymbol{S} = \boldsymbol{M}$ and set $h : x \in \mathbb{R} \rightarrow x$. Note however that if the algorithm requires h to be strictly positive, then the matrix $\boldsymbol{M}$ should be positive definite.

2.3.1 Trace of a graph filter

We present here an approach aiming at computing the trace of a graph filter. It relies on the following proposition.

[2] N actually corresponds to the order of approximation of the integrals defining the coefficients of the Chebyshev sum (cf. Equation (2.6)) as Riemann sums. N can be fixed at a few thousands in most cases. Hence it is therefore safe to assume that $N \ll n$.

Proposition 2.3.1. *Let $\boldsymbol{S} \in \mathscr{M}_n(\mathbb{R})$ be a real symmetric shift operator and let $h(\boldsymbol{S})$ be a graph filter with respect to $\boldsymbol{S}$ with transfer function $h : \mathbb{R} \mapsto \mathbb{R}$.*
Let $\boldsymbol{W}$ be a white signal, i.e. a vector composed of n independent zero-mean and unit-variance random variables.
Then $\boldsymbol{W}^T h(\boldsymbol{S})\boldsymbol{W}$ is an unbiased estimator of the trace of $h(\boldsymbol{S})$:

$$\mathbb{E}\left[\boldsymbol{W}^T h(\boldsymbol{S})\boldsymbol{W}\right] = \mathrm{Trace}\left(h(\boldsymbol{S})\right) \quad . \tag{2.8}$$

Proof. By linearity of the expectation: $\mathbb{E}\left[\boldsymbol{W}^T h(\boldsymbol{S})\boldsymbol{W}\right] = \sum_{k=1}^{n}\sum_{j=1}^{n} \left[h(\boldsymbol{S})\right]_{kj} \mathbb{E}\left[W_k W_j\right]$. By definition of $\boldsymbol{W}$, $\mathbb{E}\left[W_k W_j\right] = \mathrm{Cov}\left[W_k, W_j\right]$ is 1 if $k = j$ and 0 otherwise. Hence, $\mathbb{E}\left[\boldsymbol{W}^T h(\boldsymbol{S})\boldsymbol{W}\right] = \sum_{k=1}^{n} \left[h(\boldsymbol{S})\right]_{kk} = \mathrm{Trace}\left(h(\boldsymbol{S})\right)$. □

A stochastic approximation of the trace of a graph filter is therefore given by taking a Monte-Carlo estimate of the expectation in Equation (2.8):

$$\mathrm{Trace}\left(h(\boldsymbol{S})\right) \approx S_M \quad \text{with} \quad S_M = \frac{1}{M}\sum_{j=1}^{m} \boldsymbol{w}_j^T h(\boldsymbol{S})\boldsymbol{w}_j \quad , \tag{2.9}$$

where $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_M$ are M independent realizations of $\boldsymbol{W}$. The quadratic form in Equation (2.8) can be computed in two steps: first the product $\boldsymbol{u} = h(\boldsymbol{S})\boldsymbol{w}$ is calculated using the Chebyshev filtering algorithm, then the inner product $\boldsymbol{w}^T\boldsymbol{u}$ is returned. Hence, an approximation of the trace can be computed through Equation (2.9) for a global computational cost of filtering M signals. This method is outlined in Algorithm 2.8.

Algorithm 2.8: Trace approximation by Chebyshev filtering.

Parameters: Probability distribution $\mathcal{D}$ of a (real) random variable with mean 0 and variance 1. Any additional parameters for Chebyshev filtering.
Input: Shift operator $\boldsymbol{S} \in \mathscr{M}_n(\mathbb{R})$. Transfer function $h : \mathbb{R} \to \mathbb{R}$. Approximation order $m \in \mathbb{N}$ of the transfer function. Number of realizations M.
Output: An approximation of $\mathrm{Trace}(h(\boldsymbol{S}))$.

Initialization: $y = 0$;
for j **from** 1 **to** M **do**
  Generate $\boldsymbol{w} \in \mathbb{R}^n$ with independent entries drawn from $\mathcal{D}$;
  Compute $\boldsymbol{u} = h(\boldsymbol{S})\boldsymbol{w}$ using Chebyshev filtering at approximation order m ;
  $y \leftarrow \left((j-1)y + \boldsymbol{w}^T\boldsymbol{u}\right)/j$;
Return y.

Remark 2.3.2. In practice, the formulation of Algorithm 2.8 allows for a premature exit from the "for" loop. Indeed, at the j-th iteration, the scalar y actually contains the average over all white signals generated up until this point. Hence, one could imagine an additional criterion on the evolution of the values of y that would provoke a loop break. For instance we could stop the algorithm if, for several consecutive iterations, the difference between the current and previous values of y is below a given threshold.

Algorithm 2.8 hence provides a method to compute the trace of any graph filter using Chebyshev filtering. The computational cost of this method is dominated by the filtering steps (assuming generating the random vectors $\boldsymbol{w}$ is inexpensive and represents a cost of order $\mathcal{O}(n)$). Hence, the computational cost of Algorithm 2.8 is of order $\mathcal{O}(M \times mdn)$ where d is the mean number of non-zero entries in a row of $\boldsymbol{S}$ and M is the number of realizations used to defined

	Rademacher	Gaussian
Variance of the trace estimator	$\frac{2}{M}(\mathrm{Trace}(\boldsymbol{A}^2) - \sum_{k=1}^{n} A_{kk}^2)$	$\frac{2}{M}\mathrm{Trace}(\boldsymbol{A}^2)$
Bound on the number of samples with central limit theorem	$2\frac{F_{\mathcal{N}}^{-1}(1-\alpha/2)^2}{\epsilon^2}(\mathrm{Trace}(\boldsymbol{A}^2) - \sum_{k=1}^{n} A_{kk}^2)$	$2\frac{F_{\mathcal{N}}^{-1}(1-\alpha/2)^2}{\epsilon^2}\mathrm{Trace}(\boldsymbol{A}^2)$
Bound on the number of samples in the positive semi-definite case	$6\frac{\mathrm{Trace}(\boldsymbol{A})^2}{\epsilon^2}\log(\frac{2}{\alpha}\mathrm{rank}(\boldsymbol{A}))$	$20\frac{\mathrm{Trace}(\boldsymbol{A})^2}{\epsilon^2}\log(\frac{2}{\alpha})$

Table 2.2: Properties of the estimator S_M of the trace of a graph filter $\boldsymbol{A} = h(\boldsymbol{S})$, as defined in Equation (2.9), with respect to the distribution chosen to generate the white signals $\boldsymbol{w}$. See (Avron and Toledo, 2011) for proofs.

the stochastic estimators, m is the order of the Chebyshev approximation and n is the size of $\boldsymbol{S}$. This should be compared to the huge computational cost of the exact approach that consists in diagonalizing the graph filter, involving then $\mathcal{O}(n^3)$ operations.

Two questions remain unanswered: how to choose the distribution $\mathcal{D}$ defining the white signals and the number of realizations M that should be generated. A natural criterion for these choices consists in trying to minimize the variance of the estimator S_M in Equation (2.9). This variance directly depends on both parameters, as it is given by σ^2/M where $\sigma^2 = \mathrm{Var}[\boldsymbol{W}^T h(\boldsymbol{S})\boldsymbol{W}]$, and is linked to the approximation error $|\mathrm{Trace}(h(\boldsymbol{S})) - S_M|$ of S_M.

Indeed, the central limit theorem states that asymptotically in M, the limiting distribution of S_M is normal with mean $\mathbb{E}[S_M] = \mathrm{Trace}(h(\boldsymbol{S}))$ and variance σ^2/M. Hence the approximation error $|\mathrm{Trace}(h(\boldsymbol{S})) - S_M|$ can be estimated using the cumulative distribution function (cdf) $F_{\mathcal{N}}$ of the standard Gaussian distribution. Namely, the probability that its value is below a threshold $\epsilon > 0$ is given by

$$\mathbb{P}\left[|\mathrm{Trace}(h(\boldsymbol{S})) - S_M| \leq \epsilon\right] \approx 2F_{\mathcal{N}}\left(\frac{\epsilon}{\sqrt{\sigma^2/M}}\right) - 1 \quad (\text{as } M \to \infty).$$

Equivalently, using the inverse cdf $F_{\mathcal{N}}^{-1}$ of the standard Gaussian distribution (which is also its quantile function), we have for any risk level $0 < \alpha < 1$:

$$\mathbb{P}\left[|\mathrm{Trace}(h(\boldsymbol{S})) - S_M| \leq \sqrt{\frac{\sigma^2}{M}}F_{\mathcal{N}}^{-1}(1-\alpha/2)\right] \approx 1 - \alpha \quad (\text{as } M \to \infty).$$

Two factors directly impact the approximation error of S_M: the variance σ^2 of the quadratic forms and the number M of samples. In particular, minimizing the variance σ^2 by choosing an appropriate distribution $\mathcal{D}$ should lead to require less samples to keep the approximation error below a small threshold with high probability. Hutchinson (1989, Proposition 1) shows that σ^2 is minimal whenever the entries of $\boldsymbol{W}$ follow a Rademacher distribution i.e. they take values either $+1$ or -1 with probability $1/2$, placing this distribution as a premium candidate for $\mathcal{D}$.

Going further down this road, Avron and Toledo (2011) estimated, in the particular case where $h(\boldsymbol{S})$ is also positive semi-definite, the actual number of samples needed for the approximation error to be below a threshold ϵ with a probability $1-\alpha$ (with the asymptotic requirement of the central limit theorem). They showed that generating the entries of $\boldsymbol{w}$ using a standard Gaussian distribution demands a lower number of samples M to achieve the same accuracy (with the same probability) as when a Rademacher distribution is used. Both distributions are hence considered to run Algorithm 2.8. Table 2.2 compares them in terms of variance of the estimator, and number of samples required in the asymptotic case and in the positive definite case.

2.3.2 Histogram of eigenvalues of a shift operator

Recall that we denote $\lambda_1, \ldots, \lambda_n \in \mathbb{R}$ the eigenvalues of $\boldsymbol{S}$. Computing a histogram of these values over a interval $[a, b]$ consists in partitioning this interval into a set of $M_b \geq 1$ disjoint

subintervals of size $\tau = (b-a)/M_b$, also called bins, and counting the number of eigenvalues falling into each one of the bins.

Formally, let $n_\tau : \mathbb{R} \mapsto \mathbb{N}$ be the counting function defined for $\tau > 0$ by:

$$n_\tau(x) = \text{Card}\left\{ j \in [\![1, n]\!] : \lambda_j \in [x - \frac{\tau}{2}, x + \frac{\tau}{2}[\right\}, \quad x \in \mathbb{R} \quad . \tag{2.10}$$

Then the histogram of $\{\lambda_1, \dots, \lambda_n\}$ over an interval $[a, b] \subset \mathbb{R}$ containing them and with bin size τ is defined as the set of values:

$$\left\{ n_\tau \left(a + \left(m + \frac{1}{2} \right) \tau) \right) : m \in [\![0, \left\lceil \frac{b-a}{\tau} \right\rceil - 1]\!] \right\} \quad .$$

Hence, being able to compute the histogram of eigenvalues of a shift operator is equivalent to being able to compute values of the counting function n_τ over the interval $[a, b]$. Doing so with an efficient algorithm is the object of this section.

Let us assume that the interval $[a, b]$ is known (using for instance Proposition 2.2.1 or Theorem 2.2.2) and let $\tau > 0$ be fixed. A naive way of computing $n_\tau(x), x \in [a, b]$ consists in first computing all the eigenvalues of $\boldsymbol{S}$ and then counting how many of them fall into the bin of size τ centered at x. Doing so would be practically infeasible as the first step requires the full diagonalization of $\boldsymbol{S}$. So, for the same reasons as those presented in Section 2.1.1 to avoid graph filtering by eigendecomposition, this approach should not be considered. Instead, an approach based on Chebyshev filtering is proposed, based on the following result, already exploited by Di Napoli et al. (2016).

Proposition 2.3.2. *Let $\boldsymbol{S}$ be a real symmetric shift operator with eigenvalues $\lambda_1, \dots, \lambda_n \subset \mathbb{R}$ and let $h : [a, b] \to \mathbb{R}$ be a function defined on an interval $[a, b]$ containing all the eigenvalues of $\boldsymbol{S}$. Then,*

$$\mathbb{E}\left[\boldsymbol{W}^T h(\boldsymbol{S}) \boldsymbol{W} \right] = \sum_{k=1}^{n} h(\lambda_k) = \text{Trace}(h(\boldsymbol{S})) \quad , \tag{2.11}$$

where $\boldsymbol{W} \in \mathbb{R}^n$ is a white signal.

Proof. This a a direct consequence of Proposition 2.3.1 that relies on the fact that by definition of graph filters and using the properties of the trace function:

$$\begin{aligned} \text{Trace}(h(\boldsymbol{S})) &= \text{Trace}(\boldsymbol{V}^T \text{Diag}(h(\lambda_1), \dots, h(\lambda_n)) \boldsymbol{V}) = \text{Trace}(\text{Diag}(h(\lambda_1), \dots, h(\lambda_n)) \boldsymbol{V}\boldsymbol{V}^T) \\ &= \text{Trace}(\text{Diag}(h(\lambda_1), \dots, h(\lambda_n))) = \sum_{k=1}^{n} h(\lambda_k) \quad . \end{aligned}$$

□

In particular, note that, for any $x \in [a, b]$, the counting function can be written using indicator functions:

$$n_\tau(x) = \sum_{k=1}^{n} \mathbb{1}_{[x-\tau/2, x+\tau/2[}(\lambda_i) \quad ,$$

where $\mathbb{1}_{[x-\tau/2, x+\tau/2[}$ denotes the indicator function of the interval $[x - \tau/2, x + \tau/2[$. Hence, from Proposition 2.3.2,

$$\forall x \in [a, b], \quad n_\tau(x) = \mathbb{E}\left[\boldsymbol{w}^T \mathbb{1}_{[x-\tau/2, x+\tau/2[}(\boldsymbol{S}) \boldsymbol{w} \right] = \text{Trace}(\mathbb{1}_{[x-\tau/2, x+\tau/2[}(\boldsymbol{S})) \quad .$$

Following the results from Section 2.3.1, an idea would be to compute $n_\tau(x)$ using Algorithm 2.8. However, the function $t \mapsto \mathbb{1}_{[x-\tau/2, x+\tau/2[}(t)$ is not even continuous over $[a, b]$ as it has two discontinuities at $t = x \pm \tau/2$. Consequently, the Chebyshev series of this function will not converge uniformly and moreover, oscillations near the discontinuities will appear due to the Gibbs phenomenon (cf. Appendix B.4.6).

Nonetheless, this problem is circumvented using the approach presented in Appendix B.4.6, hence computing the coefficients of the Chebyshev sums of the discontinuous function and downscaling them using a σ-factor. This approach to compute the histogram is summed up in Algorithm 2.9, that returns a table whose first column are the midpoints of a histogram and second column contains an approximations of the counts in each bin centered at these midpoints.

Algorithm 2.9: Histogram approximation by Chebyshev filtering.

Parameters: Approximation order $m \in \mathbb{N}$ of the counting function. Number of realizations M used for the stochastic estimators. Probability distribution $\mathcal{D}$ of a (real) zero-mean random variable with variance 1. Any additional parameters for Chebyshev filtering.

Input: Shift operator $\boldsymbol{S} \in \mathscr{M}_n(\mathbb{R})$. Interval $[a, b]$ containing the eigenvalues of $\boldsymbol{S}$ and on which to compute the histogram. Bin size τ.

Output: An approximation of the histogram of eigenvalues of $\boldsymbol{S}$.

Initialization: $\boldsymbol{H} \in \mathscr{M}_{\lceil (b-a)/\tau \rceil, 2}(\mathbb{R}), x_0 = 0, y = 0$;

for k **from** 0 **to** $(\lceil (b-a)/\tau \rceil - 1)$ **do**

$x_0 \leftarrow a + (k + 1/2)\,\tau$;

Compute the coefficients $c_0, \ldots, c_m$ of the Chebyshev sum (or interpolant) of order m of the function $t \mapsto \mathbb{1}_{[x_0-\tau/2, x_0+\tau/2[}(\phi_{a,b}^{-1}(t))$ using Algorithm B.1 or B.2. Note: $\phi_{a,b}^{-1}$ is the linear map defined in Equation (2.5). ;

$y \leftarrow 0$;

for j **from** 1 **to** M **do**

Generate $\boldsymbol{w} \in \mathbb{R}^n$ with independent entries drawn from $\mathcal{D}$;

Using Algorithm 2.6, compute the product

$$\boldsymbol{u} = \sum_{k=0}^{m} \sigma\left(\frac{j}{m}\right) c_j T_j(\phi_{a,b}(\boldsymbol{S}))\boldsymbol{w} \quad ,$$

where σ is one of the σ-factors of Equations (B.22) to (B.25) and $\phi_{a,b}$ is the linear map defined in Equation (2.4).;

$y \leftarrow \left((j-1)y + \boldsymbol{w}^T\boldsymbol{u}\right)/j$;

$H_{k1} = x_0, \quad H_{k2} = y$;

Return $\boldsymbol{H}$.

The computational cost associated with Algorithm 2.9 is essentially the same as computing $\lceil (b-a)/\tau \rceil$ traces using Algorithm 2.8.

2.3.3 Log-determinant of a graph filter

We assume in this subsection that $h : \mathbb{R} \to \mathbb{R}_+^*$ is a continuous function taking strictly positive values. We are interested in estimating the log-determinant of the graph filter $h(\boldsymbol{S})$. By definition of graph filters, it is straightforward to show that this quantity equals:

$$\log\det h(\boldsymbol{S}) = \log\left(\prod_{k=1}^{n} h(\lambda_k)\right) = \sum_{k=1}^{n} \log\left(h(\lambda_k)\right) \quad . \tag{2.12}$$

Following then Proposition 2.3.2, the log-determinant of the graph filter $h(\boldsymbol{S})$ can therefore be expressed as:

$$\log\det h(\boldsymbol{S}) = \mathbb{E}\left[\boldsymbol{W}^T \log h(\boldsymbol{S})\boldsymbol{W}\right] = \mathrm{Trace}(\log h(\boldsymbol{S})) \quad , \tag{2.13}$$

where $\boldsymbol{w}$ is any white signal.

Two methods therefore arise for computing $\log\det h(\boldsymbol{S})$. The first one consists in using Equation (2.13) to notice that the log-determinant is equal to the trace of a graph filter with transfer function $t \mapsto \log(h(t))$. Hence, Algorithm 2.8 can be directly used on $\boldsymbol{S}$ and this transfer function to yield an approximation of the log-determinant.

The second method starts from Equation (2.12) and consists in directly approximating the sum over the eigenvalues of $\boldsymbol{S}$ using their histogram. Indeed, let $[a, b]$ be an interval containing the eigenvalues of $\boldsymbol{S}$ and let $\tau > 0$ be the bin size of a histogram of these eigenvalues. Let n_τ denote the counting function that yields the number of eigenvalues falling into a bin of size τ centered at any point of $[a, b]$, as defined in Equation (2.10). Then $\log\det h(\boldsymbol{S})$ can be

approximated by:

$$\log\det h(\boldsymbol{S}) = \sum_{k=1}^{n} \log\left(h(\lambda_k)\right) \approx \sum_{j=0}^{\lceil (b-a)/\tau \rceil - 1} n_\tau(a_j) \log\left(h(a_j)\right) \quad , \tag{2.14}$$

where $a_j = a + \left(j + \frac{1}{2}\right)\tau, j \in [\![0, \lceil (b-a)/\tau \rceil - 1]\!]$ are the midpoints of the histogram. Basically, the sum over all the eigenvalues is replaced by a sum over a discretization of the interval $[a, b]$ containing these eigenvalues, and weighted by the number of eigenvalues around each discretization point. One can directly see that the smoother the variations of the function $\log h$ over $[a, b]$ are, the better this approximation is. In particular, the approximation is exact whenever h is constant, i.e. has no variations.

Hence, an approximation of $\log\det h(\boldsymbol{S})$ can be obtained in two steps. First, use Algorithm 2.9 to compute a histogram of the eigenvalues of $\boldsymbol{S}$, more precisely an approximation of the weights $n_\tau(a_j)$ in Equation (2.14). Then use these counts to compute the approximation of the log-determinant as defined by Equation (2.14).

The computational cost associated with this approach is essentially that of the computation of the histogram of eigenvalues. This cost is greater than the cost of computing a single trace, as proposed in the first approach. However, once the histogram is computed, determinants for any graph filter defined through the same shift operator can be computed at virtually no cost: we only need to reevaluate Equation (2.14) for the new transfer function. In the meantime, with the first approach, changing the transfer function implies to recompute from scratch the log-determinant. Both methods therefore have their advantages and the choice between them should be made in regard with the context of use of these determinants.

2.3.4 Solving a linear system involving a graph filter

Once again let $\boldsymbol{S} \in \mathscr{M}_n(\mathbb{R})$ be a real symmetric shift operator with eigenvalues $\lambda_1, \ldots, \lambda_n$ and let $h : \mathbb{R} \to \mathbb{R}_+^*$ be a continuous function taking strictly positive values. We are now interested in finding an approximate solution of the linear system:

$$h(\boldsymbol{S})\boldsymbol{x} = \boldsymbol{b} \quad , \tag{2.15}$$

where $\boldsymbol{b} \in \mathbb{R}^n$.

$h(\boldsymbol{S})$ is positive definite given that its transfer function takes only strictly positive values. It is therefore invertible, with inverse being defined as the graph filter also defined through $\boldsymbol{S}$ but with transfer function $1/h$. Hence the solution $\boldsymbol{x} \in \mathbb{R}^n$ of Equation (2.15) is given by:

$$\boldsymbol{x} = \frac{1}{h}(\boldsymbol{S})\boldsymbol{b} \quad .$$

An approximation of this vector can then be computed using Chebyshev filtering with shift operator $\boldsymbol{S}$ and transfer function $1/h$.

Conclusion

In this chapter, we introduced the Chebyshev filtering algorithm, designed to perform filtering operations on graph signals using a polynomial approximation of the transfer function of the filter. In particular, it generates approximations of the filtered signals with a complexity that grows linearly with the order of polynomial approximation and the size of the vectors. Increasing the degree of the polynomial will improve the approximation as long as the following assumption is met.

Assumption 2.1. *Whenever a graph filter is considered, the associated transfer function is assumed to be regular enough for its Chebyshev series to converge over an interval $[a, b]$ containing all the eigenvalues of the shift operator.*

In practice, we will assume the transfer function to be Dini-Lipschitz continuous or continuous of bounded variation (cf. Theorem B.4.4).

The Chebyshev filtering algorithm was then applied to compute characteristics of a graph filter, including its trace and log-determinant, while relying solely on products between the shift operator defining the graph filter and white signals.

3

Simulation of stochastic graph signals

Contents

Résumé

Dans ce chapitre, nous présentons des algorithmes (exacts ou approchés) destinés à générer des simulations non-conditionnelles de signaux sur graphe stochastiques de propriétés de covariance connues. En particulier, nous présentons un algorithme approché de simulation basé sur le filtrage de Tchebychev, ainsi que les erreurs d'approximation numériques et statistiques qui en découlent. Nous comparons également cet algorithme aux approches par sous-espaces de Krylov.

Introduction

In the first two chapters, the focus was put on presenting a framework to study stochastic graph signals (SGS). In the next three chapters, we use this framework to perform classical tasks associated with the study of stochastic processes, namely the simulation of a SGS, its estimation from an incomplete observation and finally the inference of the parameters defining its probability distribution. In particular, we restrict ourselves to the study of stationary Gaussian SGSs, as they will play a key role in the application of the graph signal processing framework to the modeling of non-stationary Gaussian fields, which will be laid out in the second part of this dissertation.

Assumption 3.1. *Only $\boldsymbol{S}$-stationary Gaussian graph signals are considered, with $\boldsymbol{S}$ being a shift operator defined according to Assumption 1.2.*

In this chapter, algorithms to compute unconditional simulations of a SGS with known spectral density are derived. By unconditional simulation, we mean that we only aim at generating a zero-mean SGS whose covariance matrix is a graph filter with a specified positive transfer function. Hence, the SGS is drawn from its full distribution.

Two types of algorithms are presented in this chapter, much like what was done for graph filtering. On one hand, direct and exact simulation algorithms, which generate simulations with the desired statistical properties using matrix factorizations, are introduced. Then, an approximate simulation algorithm based on Chebyshev filtering is presented. Our main contributions for this part are the derivation of numerical and statistical approximation errors for the approximate simulation algorithm (cf. Section 3.2) and its comparison with Krylov subspaces approaches (cf. Section 3.3).

3.1 Simulation algorithms for Gaussian graph signals

Let $\boldsymbol{S}$ be a real symmetric shift operator, as defined in Assumption 1.2. Algorithms to compute simulations of a $\boldsymbol{S}$-stationary Gaussian SGS and the statistical properties of these algorithm are derived in this section. By Gaussian SGS we understand a SGS whose components follow a multivariate Gaussian distribution. In particular, its distribution and therefore statistical properties are entirely defined by its first two moments:

- its expectation vector, which is assumed to be $\mathbf{0}$.
- its covariance matrix, which in regard to the $\boldsymbol{S}$-stationarity assumption, is a graph filter defined by a strictly positive function called spectral density.

Let $\lambda_1, \ldots, \lambda_n$ denote the eigenvalues of $\boldsymbol{S}$ and let $\boldsymbol{V}$ be any (real or complex) orthonormal eigenbasis of $\boldsymbol{S}$. Let assume that we aim at generating realizations of a Gaussian SGS $\boldsymbol{x}$ with spectral density $f : \mathbb{R} \to \mathbb{R}_+^*$. Our goal therefore really is the simulation of a zero-mean vector with (known) covariance matrix $\boldsymbol{\Sigma} = f(\boldsymbol{S})$. We first investigate some direct simulation algorithm designed for this purpose.

3.1.1 Direct simulation of stationary graph signals

A direct method to generate samples of a Gaussian vector with known covariance matrix $\boldsymbol{\Sigma}$ consists in forming vectors $\boldsymbol{x}$ of the form (Tong, 2012)

$$\boldsymbol{x} = \boldsymbol{B}\boldsymbol{w} \quad ,$$

where $\boldsymbol{w}$ is a realization of a Gaussian white signal (i.e. a zero-mean Gaussian vector whose covariance matrix is the identity matrix) and $\boldsymbol{B}$ is a matrix such that

$$\boldsymbol{B}\boldsymbol{B}^H = \boldsymbol{\Sigma} \quad .$$

A natural candidate for such a matrix $\boldsymbol{B}$ is the Cholesky decomposition of $\boldsymbol{\Sigma}$ (Gentle, 2009). Indeed, numerous linear algebra routines allow for the computation of this matrix factorization. Algorithm 3.1 exposes this first approach to the simulation of a stationary SGS.

Algorithm 3.1: Simulation of stationary SGS by Cholesky factorization.

Input: Shift operator $\boldsymbol{S} \in \mathscr{M}_n(\mathbb{R})$. Spectral density $f : \mathbb{R} \mapsto \mathbb{R}_+$.
Output: A $\boldsymbol{S}$-stationary SGS with spectral density f.

. . . . . .

Initialization: $\boldsymbol{x} = \boldsymbol{0}$;
Build the covariance matrix $\boldsymbol{\Sigma} = f(\boldsymbol{S})$;
Compute the Cholesky factor $\boldsymbol{L}$ of $\boldsymbol{\Sigma}$;
Generate a vector $\boldsymbol{w} \in \mathbb{R}^n$ whose entries are independent standard Gaussian variables ;
$\boldsymbol{x} \leftarrow \boldsymbol{L}\boldsymbol{w}$;
Return $\boldsymbol{x}$.

Two performance issues arise when using Algorithm 3.1. First, the covariance matrix must be entirely built from the shift operator and the spectral density and stored in memory before any Cholesky factorization algorithm may be applied to it. But building $\boldsymbol{\Sigma}$ using the definition of graph filters involves to diagonalize the shift operator: this is a very expensive operation, computationally and memory-wise (cf. Section 2.1). Moreover the full covariance matrix, which is generally dense, must be stored in memory, which represents an important storage cost for large values of n.

Similarly to Chebyshev filtering (cf. Section 2.2), a cheaper alternative to diagonalization would consist in replacing $f(\boldsymbol{S})$ by a polynomial approximation, for instance by $\mathcal{S}_m[f](\boldsymbol{S})$ where $\mathcal{S}_m[f]$ denotes the Chebyshev series of order m of f. However, building the matrix $\mathcal{S}_m[f](\boldsymbol{S})$ from its polynomial expression would involve m matrix-matrix products involving $\boldsymbol{S}$. The computational cost of a single product is of order $\mathcal{O}(n^2 d)$: n^2 elements must be computed and each element requires the scalar product of a row of $\boldsymbol{S}$ which has in average d non-zero elements, with the column of another matrix. Hence the overall cost of building $\mathcal{S}_m[f](\boldsymbol{S})$ is of order $\mathcal{O}(mn^2 d)$. This cost scales quadratically with the size of the vectors n in the best case scenario (i.e. when $\boldsymbol{S}$ is sparse with $d \ll n$) and grows linearly with the approximation order. As for the memory requirements to store the result, the larger the order m is, the less sparse $\mathcal{S}_m[f](\boldsymbol{S})$ is and the more memory will be required. This can limit the order of approximation we can work with regardless of the subsequent approximation errors.

Then, once $\boldsymbol{\Sigma}$ is computed, its Cholesky factorization must be computed. The computational cost of this operation is of order $\mathcal{O}(n^3)$ whenever $\boldsymbol{\Sigma}$ is dense (Golub and Van Loan, 1996a). This cost can be greatly reduced if $\boldsymbol{\Sigma}$ is sparse: the new cost then depends on the size of the vectors n, the number of non zero entries of $\boldsymbol{\Sigma}$ and finally, its sparsity pattern which explains why a reordering of the rows and column of the matrix aiming at obtaining optimal patterns is applied beforehand. Determining the best reordering is in itself a computationally hard problem, and often the user must rely on heuristics and hope for the best (Luce and Ng, 2014).

Finally, even in the cases where the Cholesky factorization can be efficiently applied, i.e. whenever $\boldsymbol{\Sigma}$ is sparse and can easily be optimally reordered, the Cholesky factor must still be stored in memory, which represents an additional memory cost.

Faced with the important computational and storage costs associated with the direct approach presented in this subsection, we now leverage the fact that the covariance is actually a graph filter to derive a new simulation algorithm based on graph filtering.

3.1.2 Simulation of stationary graph signals by filtering

A second approach to generate simulations of a stationary SGS with a given spectral density relies on the statistical properties of stationary SGSs. Indeed, in Theorem 1.4.3, we showed that a $\boldsymbol{S}$-stationary SGS with spectral density f is the output of filtering white signals with the

graph filter $\sqrt{f}(\boldsymbol{S})$. Clearly, by definition of Gaussian vectors (cf. Appendix A.4.2), if the white signal is Gaussian, so is its filtered output given that it is a linear transformation of the white signal. Hence, the problem of generating a sample of a stationary SGS is simply reduced to that of graph filtering. Using one of the exact filtering algorithms presented in Section 2.1 to filter a vector with independent standard Gaussian entries therefore yields the desired simulation of SGS. This approach is synthesized in Algorithm 3.2.

Algorithm 3.2: Simulation of a stationary SGS by exact graph filtering.

Input: Shift operator $\boldsymbol{S} \in \mathscr{M}_n(\mathbb{R})$. Spectral density $f : \mathbb{R} \mapsto \mathbb{R}_+$.
Output: A $\boldsymbol{S}$-stationary SGS with spectral density f.

Initialization: $\boldsymbol{x} = \boldsymbol{0}$;
Generate a vector $\boldsymbol{w} \in \mathbb{R}^n$ whose entries are independent standard Gaussian variables. ;
Compute $\boldsymbol{x} = \sqrt{f}(\boldsymbol{S})\boldsymbol{w}$ using Algorithm 2.1 or 2.5. ;
Return $\boldsymbol{x}$.

The computational cost of Algorithm 3.2 is essentially due to the exact filtering step, which makes it intractable in practice. Indeed, costs similar or higher to the Cholesky approach are to be expected (cf. Section 2.1). Following then the results of Section 2.2, a workaround is provided by Chebyshev filtering, through which the exact computation of the filtered signal

$$\boldsymbol{x} = \sqrt{f}(\boldsymbol{S})\boldsymbol{w} \quad , \tag{3.1}$$

is replaced by that of the signal

$$\boldsymbol{x}^{(m)} = \mathcal{S}_m[\sqrt{f}](\boldsymbol{S})\boldsymbol{w} \quad , \tag{3.2}$$

where $\boldsymbol{w}$ is a realization of a (Gaussian) white signal and $\mathcal{S}_m[\sqrt{f}]$ is the polynomial corresponding to the Chebyshev series of order m of the function $\sqrt{f}$, over an interval containing the eigenvalues of $\boldsymbol{S}$. This approach, which we call *Chebyshev simulation*, is outlined in Algorithm 3.3.

Algorithm 3.3: Chebyshev simulation of a stationary SGS.

Input: Shift operator $\boldsymbol{S} \in \mathscr{M}_n(\mathbb{R})$. Spectral density $f : \mathbb{R} \mapsto \mathbb{R}_+$.
Output: A $\boldsymbol{S}$-stationary SGS with spectral density f.

Initialization: $\boldsymbol{x} = \boldsymbol{0}$;
Generate a vector $\boldsymbol{w} \in \mathbb{R}^n$ whose entries are independent standard Gaussian variables ;
Compute $\boldsymbol{x} = \sqrt{f}(\boldsymbol{S})\boldsymbol{w}$ using Chebyshev filtering ;
Return $\boldsymbol{x}$.

Once again, the resulting vector $\boldsymbol{x}^{(m)}$ is guaranteed to follow a zero-mean Gaussian distribution, as it is a linear transform of a zero-mean Gaussian vector. Its covariance matrix is given by:

$$\mathrm{Var}[\boldsymbol{x}^{(m)}] = \mathcal{S}_m[\sqrt{f}](\boldsymbol{S}) \left(\mathcal{S}_m[\sqrt{f}](\boldsymbol{S})\right)^H = \mathcal{S}_m[\sqrt{f}](\boldsymbol{S})^2 \quad , \tag{3.3}$$

which ensures that $\boldsymbol{x}^{(m)}$ is a $\boldsymbol{S}$-stationary SGS. However in general $\mathrm{Var}[\boldsymbol{x}^{(m)}]$ is different from the target covariance matrix, $f(\boldsymbol{S})$. Indeed the former is a $\boldsymbol{S}$-filter with transfer function $\mathcal{S}_m[\sqrt{f}]^2$ whereas the latter has transfer function f. The next section investigates the difference between the resulting vectors $\boldsymbol{x}$ and $\boldsymbol{x}^{(m)}$.

3.2 Approximation and statistical errors of Chebyshev simulations

In this section, we investigate the accuracy of the simulations generated by the Chebyshev algorithm (cf. Algorithm 3.3). Two dimensions of the problem are considered. On one hand, seeing the Chebyshev simulation algorithm as simply a graph filtering problem that was answered

using Chebyshev filtering, a numerical approximation error is derived, in the same manner as in Section 2.2. On the other hand, seeing the Chebyshev simulation algorithm as a simulation algorithm in its own right, the statistical properties of its outputs are considered and compared to the targeted ones.

3.2.1 Numerical approximation error of Chebyshev simulations

Let $\boldsymbol{X} = \sqrt{f}(\boldsymbol{S})\boldsymbol{W}$ denote a $\boldsymbol{S}$-stationary SGS with spectral density f, obtained from a Gaussian white signal $\boldsymbol{W}$. Chebyshev simulations basically replace samples of $\boldsymbol{X}$ by samples of the SGS $\boldsymbol{X}^{(m)}$ defined by (cf. Equation (3.2))

$$\boldsymbol{X}^{(m)} = \mathcal{S}_m[\sqrt{f}](\boldsymbol{S})\boldsymbol{W} \quad ,$$

for some order of approximation $m \in \mathbb{N}$. The approximation error between both SGS can be assessed using the same reasoning as in Section 2.2. Indeed, let E_m denote this approximation error, which is defined as

$$E_m := \|\boldsymbol{X} - \boldsymbol{X}^{(m)}\|_2 = \| \left(\sqrt{f}(\boldsymbol{S}) - \mathcal{S}_m[\sqrt{f}](\boldsymbol{S}) \right) \boldsymbol{W} \|_2 \quad .$$

where $\| \cdot \|_2$ denotes the Euclidean norm. In particular, E_m is a (positive) random variable. Its square can be expressed using the eigenvalues $\lambda_1, \dots, \lambda_n$ of $\boldsymbol{S}$ as

$$E_m^2 = \| \left(\sqrt{f} - \mathcal{S}_m[\sqrt{f}] \right) (\boldsymbol{S})\boldsymbol{W} \|_2^2 = \sum_{k=1}^{n} \left(\sqrt{f}(\lambda_k) - \mathcal{S}_m[\sqrt{f}](\lambda_k) \right)^2 \widetilde{W}_k^2 \quad , \tag{3.4}$$

where $\widetilde{\boldsymbol{W}} = \boldsymbol{V}^T\boldsymbol{W}$ is the graph Fourier transform of $\boldsymbol{W}$ with respect to some real orthonormal eigenbasis of $\boldsymbol{S}$. In particular, note that $\widetilde{\boldsymbol{W}}$ is also a white signal.

Following Equation (3.4), the expectation and variance of E_m^2 are given by:

$$\begin{cases} \mathbb{E}[E_m^2] = \displaystyle\sum_{k=1}^{n} \left(\sqrt{f}(\lambda_k) - \mathcal{S}_m[\sqrt{f}](\lambda_k) \right)^2 \\ \mathrm{Var}[E_m^2] = 2 \displaystyle\sum_{k=1}^{n} \left(\sqrt{f}(\lambda_k) - \mathcal{S}_m[\sqrt{f}](\lambda_k) \right)^4 \end{cases} . \tag{3.5}$$

Hence as $m \to \infty$, both the expectation and the variance of E_m^2 go to zero, meaning that asymptotically the approximation error E_m becomes zero. In particular, denote ε_m the approximation error of $\mathcal{S}_m[\sqrt{f}]$ over the interval $[a, b]$ over which it is computed, i.e.

$$\varepsilon_m := \max_{\lambda \in [a,b]} |\sqrt{f}(\lambda) - \mathcal{S}_m[\sqrt{f}](\lambda)| \quad .$$

Then, following Equation (3.5), we have $\mathbb{E}[E_m^2] = \mathcal{O}(n\varepsilon_m^2)$ and $\mathrm{Var}[E_m^2] = \mathcal{O}(n\varepsilon_m^4)$. Besides, recall that Chebyshev's inequality (Stewart, 2009, Section 8.2) ensures that, for any confidence level $\alpha > 0$:

$$\forall \alpha > 0, \quad \mathbb{P}\left[|E_m^2 - \mathbb{E}[E_m^2]| \leq \sqrt{\frac{\mathrm{Var}[E_m^2]}{\alpha}} \right] \geq 1 - \alpha, \quad \alpha > 0 \quad . \tag{3.6}$$

Hence, imposing a small enough approximation error ε_m on the Chebyshev suffices to ensure that with high probability, the approximation error E_m of the Chebyshev simulation can be made as small as we want.

A more practical concentration inequality can be derived by introducing the random variable $\widehat{E}_m$ associated to E_m by

$$E_m^2 = \sum_{k=1}^{n} \left(\sqrt{f}(\lambda_k) - \mathcal{S}_m[\sqrt{f}](\lambda_k) \right)^2 \widetilde{W}_k^2 \Longrightarrow \widehat{E}_m^2 := \varepsilon_m^2 \sum_{k=1}^{n} \widetilde{W}_k^2 \quad .$$

Then in particular, $E_m^2 \leq \widehat{E}_m^2$ and so, for any $\eta > 0$,

$$\mathbb{P}\left[E_m \leq \eta\right] = \mathbb{P}\left[E_m^2 \leq \eta^2\right] \geq \mathbb{P}\left[\widehat{E}_m^2 \leq \eta^2\right] = \mathbb{P}\left[\sum_{k=1}^{n} \widetilde{\boldsymbol{W}}_k^2 \leq \frac{\eta^2}{\varepsilon_m^2} \right] \quad .$$

Given that $\widetilde{W}_1, \dots, \widetilde{W}_n$ are standard Gaussian variables, $\sum_{k=1}^{n} \widetilde{\boldsymbol{W}}_k^2$ follows a chi-squared distribution with n degrees of freedom, denoted $\chi^2(n)$. Then,

$$\mathbb{P}\left[E_m \leq \eta\right] \geq \mathbb{P}\left[\chi^2(n) \leq \frac{\eta^2}{\varepsilon_m^2}\right] = F_{\chi^2(n)}\left(\frac{\eta^2}{\varepsilon_m^2}\right) \quad ,$$

where $F_{\chi^2(n)}$ is the cumulative distribution function of $\chi^2(n)$. Therefore, if a confidence level $\alpha > 0$ is fixed, the approximation error satisfies

$$\mathbb{P}\left[E_m \leq \varepsilon_m \sqrt{F_{\chi^2(n)}^{-1}\left(1-\alpha\right)}\right] \geq 1-\alpha, \quad \alpha > 0 \quad . \tag{3.7}$$

This last expression can be made slightly more explicit by recalling that, according to the central limit theorem, the distribution $\chi^2(n)$ actually converges to a normal distribution with mean n and variance $2n$ as n grows (Box et al., 2005). In practice, for $n > 50$, the difference between both distributions can even be neglected (Box et al., 2005). Assuming we fall in this case, the concentration inequality becomes

$$\mathbb{P}\left[E_m \leq \varepsilon_m \sqrt{n} \sqrt{1 + \sqrt{\frac{2}{n}} F_{\mathcal{N}}^{-1}\left(1-\alpha\right)}\right] \geq 1-\alpha, \quad \alpha > 0 \quad , \tag{3.8}$$

where $F_{\mathcal{N}}$ denotes the cumulative distribution function of the standard Gaussian distribution. Hence, with probability $1-\alpha$, the approximation error E_m of a Chebyshev simulation is of order $\mathcal{O}(\epsilon_m \sqrt{n})$, and is therefore entirely driven by the approximation error ε_m of the Chebyshev sum.

Equations (3.7) and (3.8) provide conditions on the approximation error of the Chebyshev sum, and therefore on the order of approximation m that should be chosen, so that with high probability the approximation error of a Chebyshev simulation is as close to 0 as one may want.

Following this approach leads to regarding the Chebyshev simulation algorithm purely as an algorithm used to approximate numerically a target SGS $\boldsymbol{x}$, which is known to have the right statistical properties (namely, Gaussian with covariance matrix $f(\boldsymbol{S})$). However, if the algorithm were to yield a simulated SGS $\boldsymbol{x}^{(m)}$ with bad approximation error, but whose statistical properties are so close to those of $\boldsymbol{x}$ that they both "seem" drawn from the same distribution, then $\boldsymbol{x}^{(m)}$ would still constitute a great output for our simulation purpose. This approach is investigated in the next section.

3.2.2 Statistical error of Chebyshev simulations

The goal of a simulation algorithm is to generate random vectors with some predefined statistical properties. In our case it comes down to generate zero-mean Gaussian vectors with covariance matrix $f(\boldsymbol{S})$. Once again, $\boldsymbol{x}^{(m)}$ denotes an output of the Chebyshev simulation algorithm, as defined by Equation (3.2). In this section, the statistical properties of $\boldsymbol{x}^{(m)}$ are exploited in order to derive a criterion on the approximation error of the Chebyshev sum that ensures that $\boldsymbol{x}^{(m)}$ can "pass" for a zero-mean Gaussian vector with covariance matrix $f(\boldsymbol{S})$.

Notice first that $\boldsymbol{x}^{(m)}$ is by definition a zero-mean Gaussian vector. The only statistical difference with $\boldsymbol{x}$ resides in the fact that the covariance matrix of $\boldsymbol{x}^{(m)}$ is $\mathrm{Var}[\boldsymbol{x}^{(m)}] = \mathcal{S}_m[\sqrt{f}]^2(\boldsymbol{S})$ (instead of $f(\boldsymbol{S})$). Hence, the question that should be answered really is: what criterion can be fixed so that Gaussian vectors with covariance matrix $\mathcal{S}_m[\sqrt{f}]^2(\boldsymbol{S})$ become statistically indiscernible from their counterparts with covariance matrix $f(\boldsymbol{S})$? An approach based on statistical tests on linear combinations obtained from both types of vectors is now outlined to answer this interrogation.

Consider a sample of N_s independent zero-mean Gaussian vectors $\left(\boldsymbol{x}_1^{(m)}, \dots, \boldsymbol{x}_{N_s}^{(m)}\right)$ with covariance matrix $\mathcal{S}_m[\sqrt{f}]^2(\boldsymbol{S})$. Each one of these vectors can be seen as an independent output of the Chebyshev simulation algorithm. Let's consider the following null hypothesis test:

$$H_0 : \left(\boldsymbol{x}_1^{(m)}, \dots, \boldsymbol{x}_{N_s}^{(m)}\right) \text{ is a sample of } N_s \text{ independent vectors with covariance matrix } f(\boldsymbol{S}) \ .$$

Recall that by definition (cf. Appendix A.4.2), a random vector $\boldsymbol{z} \in \mathbb{R}^n$ is a Gaussian vector with covariance matrix $\boldsymbol{\Sigma}$ if and only if, for any deterministic (and arbitrary) set of coefficients $\boldsymbol{c} \in \mathbb{R}^n$, $\boldsymbol{c}^T\boldsymbol{z}$ is a Gaussian variable with variance $\boldsymbol{c}^T\boldsymbol{\Sigma}\boldsymbol{c}$. Therefore, hypothesis H_0 won't be rejected if $\forall \boldsymbol{c} \in \mathbb{R}^n$, the hypothesis $H_0^{\boldsymbol{c}}$ defined by:

$$H_0^{\boldsymbol{c}} : \left(\boldsymbol{c}^T \boldsymbol{x}_1^{(m)}, \dots, \boldsymbol{c}^T \boldsymbol{x}_{N_s}^{(m)}\right) \text{ is a sample of zero-mean Gaussian variables with variance } \boldsymbol{c}^T f(\boldsymbol{S})\boldsymbol{c},$$

is not rejected.

The (two-sided) chi-square test for the variance (Snedecor and Cochran, 1989) is considered to test the null hypothesis $H_0^{\boldsymbol{c}}$ for some $\boldsymbol{c} \in \mathbb{R}^n$. Indeed, based on a sample from a population of normally distributed data, this test is used to check whether the population variance is equal to an hypothesized value. In our case, this hypothesized value is $\boldsymbol{c}^T f(\boldsymbol{S})\boldsymbol{c}$ and the sample is $\left(\boldsymbol{c}^T \boldsymbol{x}_1^{(m)}, \dots, \boldsymbol{c}^T \boldsymbol{x}_{N_s}^{(m)}\right)$.

For a given $\boldsymbol{c} \in \mathbb{R}^n$, the statistic $t(\boldsymbol{c})$ of the chi-square test for the variance is

$$t(\boldsymbol{c}) = (N_s - 1)\frac{S^2(\boldsymbol{c})}{\boldsymbol{c}^T f(\boldsymbol{S})\boldsymbol{c}} \quad ,$$

where $S^2(\boldsymbol{c})$ is the (unbiased) sample variance defined as

$$S^2(\boldsymbol{c}) = \frac{1}{N_s - 1}\sum_{k=1}^{N_s}\left(\boldsymbol{c}^T \boldsymbol{x}_k^{(m)} - m(\boldsymbol{c})\right)^2, \quad m(\boldsymbol{c}) = \frac{1}{N_s}\sum_{j=1}^{N_s} \boldsymbol{c}^T \boldsymbol{x}_j^{(m)} \quad .$$

If the null hypothesis were to be true, i.e. if the population variance were to be $\boldsymbol{c}^T f(\boldsymbol{S})\boldsymbol{c}$ then the statistic $t(\boldsymbol{c})$ would follow a chi-squared distribution with $N_s - 1$ degrees of freedom (denoted $\chi^2(N_s - 1)$). Hence, to test whether or not to reject the null hypothesis, the actual value of $t(\boldsymbol{c})$ computed from the sample is compared to "typical" values a $\chi^2(N_s - 1)$ variable should take.

Formally, we say that $H_0^{\boldsymbol{c}}$ is not rejected with significance level $\alpha > 0$ if $t(\boldsymbol{c})$ satisfies

$$\chi^2_{\frac{\alpha}{2},N_s-1} \leq t(\boldsymbol{c}) \leq \chi^2_{1-\frac{\alpha}{2},N_s-1} \quad , \tag{3.9}$$

where χ^2_{p,N_s-1} is the p-th quantile of the $\chi^2(N_s - 1)$ distribution. Recall in particular that $F_{\chi^2(N_s-1)}(\chi^2_{p,N_s-1}) = p$. If Equation (3.9) is not satisfied, we say that $H_0^{\boldsymbol{c}}$ is rejected (with significance level α).

Note that a draw from a $\chi^2(N_s - 1)$ variable would have a probability $1 - \alpha$ to fall in the interval $[\chi^2_{\frac{\alpha}{2},N_s-1}, \chi^2_{1-\frac{\alpha}{2},N_s-1}]$ that appears in Equation (3.9). This means that whenever the null hypothesis is rejected with significance α, the probability that it was true after all, and therefore that $t(\boldsymbol{c})$ is a $\chi^2(N_s - 1)$ variable, is less than α. α is also referred to as the type-I error, i.e. the probability of wrongfully rejecting the null hypothesis.

Recall now that the sample $\left(\boldsymbol{c}^T \boldsymbol{x}_1^{(m)}, \dots, \boldsymbol{c}^T \boldsymbol{x}_{N_s}^{(m)}\right)$ is generated from Chebyshev simulations. Hence, the true population variance of the sample is known and is actually equal to $\boldsymbol{c}^T \mathcal{S}_m[\sqrt{f}]^2(\boldsymbol{S})\boldsymbol{c}$. The testing procedure therefore really aims at determining whether a sample from a population of Gaussian variables with variance $\boldsymbol{c}^T \mathcal{S}_m[\sqrt{f}]^2(\boldsymbol{S})\boldsymbol{c}$ can be mistaken for a sample from a population of Gaussian variables with variance $\boldsymbol{c}^T f(\boldsymbol{S})\boldsymbol{c}$, in the sense that $H_0^{\boldsymbol{c}}$ will not be rejected.

In particular, the probability $R_\alpha(\boldsymbol{c})$ that $H_0^{\boldsymbol{c}}$ is rejected with significance α can be derived as

$$R_\alpha(\boldsymbol{c}) = 1 - \mathbb{P}\left[\chi^2_{\frac{\alpha}{2},N-1} \leq t(\boldsymbol{c}) \leq \chi^2_{1-\frac{\alpha}{2},N-1}\right] \quad .$$

Note that, as described in the previous paragraph, in the case where the true population variance is equal to the hypothesized one, this probability is equal to α. In the general case, the following result links $R_\alpha(\boldsymbol{c})$ to the accuracy of the polynomial approximation of f by $\mathcal{S}_m[\sqrt{f}]^2$ using a criterion that is actually independent of $\boldsymbol{c}$.

Proposition 3.2.1. *Let $[a, b]$ be an interval containing all the eigenvalues of $\boldsymbol{S}$. Let $\widehat{\varepsilon}_m$ denote the relative approximation error of the Chebyshev sum, defined by*

$$\widehat{\varepsilon}_m := \max_{\lambda \in [a,b]} \left| \frac{f(\lambda) - \mathcal{S}_m[\sqrt{f}](\lambda)^2}{\mathcal{S}_m[\sqrt{f}](\lambda)^2} \right| \quad . \tag{3.10}$$

Let $R_\alpha(\boldsymbol{c})$ denote the probability of rejecting, with significance $\alpha > 0$ in a chi-square test for the variance, the null hypothesis $H_0^{\boldsymbol{c}}$ (defined for N_s samples).

Then $\forall \gamma > 0$, there exists a threshold $\eta_\alpha(N_s, \gamma) > 0$ such that:

$$\widehat{\varepsilon}_m \leq \eta_\alpha(N_s, \gamma) \Rightarrow \forall \boldsymbol{c} \in \mathbb{R}^n, \quad R_\alpha(\boldsymbol{c}) \leq (1+\gamma)\alpha \quad . \tag{3.11}$$

Proof. Let $\boldsymbol{c} \in \mathbb{R}_s^N$. Denote $\sigma^2(\boldsymbol{c}) = \boldsymbol{c}^T \mathcal{S}_m[\sqrt{f}]^2(\boldsymbol{S})\boldsymbol{c}$ and $\sigma_0^2(\boldsymbol{c}) = \boldsymbol{c}^T f(\boldsymbol{S})\boldsymbol{c}$. Then, $R_\alpha(\boldsymbol{c})$ can be written

$$R_\alpha(\boldsymbol{c}) = 1 - \mathbb{P}\left[\frac{\sigma_0^2(\boldsymbol{c})}{\sigma^2(\boldsymbol{c})}\chi^2_{\frac{\alpha}{2},N_s-1} \leq t'(\boldsymbol{c}) \leq \frac{\sigma_0^2(\boldsymbol{c})}{\sigma^2(\boldsymbol{c})}\chi^2_{1-\frac{\alpha}{2},N_s-1}\right] \quad ,$$

where $t'(\boldsymbol{c})$ is the statistic defined by

$$t'(\boldsymbol{c}) = \frac{\sigma_0^2(\boldsymbol{c})}{\sigma^2(\boldsymbol{c})} t(\boldsymbol{c}) = (N_s - 1)\frac{S^2(\boldsymbol{c})}{\sigma^2(\boldsymbol{c})} \quad .$$

By definition, the sample $\left(\boldsymbol{c}^T \boldsymbol{x}_1^{(m)}, \ldots, \boldsymbol{c}^T \boldsymbol{x}_{N_s}^{(m)}\right)$ is Gaussian with variance $\sigma^2(\boldsymbol{c})$. Hence, $t'(\boldsymbol{c})$ follows a $\chi^2(N_s - 1)$ distribution. So, if $\tau(\boldsymbol{c})$ denotes the ratio

$$\tau(\boldsymbol{c}) = \frac{\sigma_0^2(\boldsymbol{c})}{\sigma^2(\boldsymbol{c})} \quad .$$

Then,

$$R_\alpha(\boldsymbol{c}) = 1 - \left(F_{\chi^2(N_s-1)}\left(\chi^2_{1-\frac{\alpha}{2},N_s-1}\tau(\boldsymbol{c})\right) - F_{\chi^2(N_s-1)}\left(\chi^2_{\frac{\alpha}{2},N_s-1}\tau(\boldsymbol{c})\right)\right) \quad .$$

The probability $R_\alpha(\boldsymbol{c})$ only depends on the ratio $\tau(\boldsymbol{c})$ (and the parameters of the test, namely N_s and α). Considering it as a function of the ratio $\tau \in [0, +\infty[$, several properties of R_α can be derived. First, given that $\forall \tau \in \mathbb{R}_+$, $R_\alpha(\tau)$ is a probability, $0 \leq R_\alpha(\tau) \leq 1$. Besides, from the fact that that $F_{\chi^2(N_s-1)}$ is a continuous cumulative distribution function, we get that R_α is also continuous (and even differentiable) and that:

$$\lim_{\tau \to 0} R_\alpha(\tau) = 1 = \lim_{\tau \to +\infty} R_\alpha(\tau) \quad . \tag{3.12}$$

Finally, from the study of the sign of its derivative (which can easily be expressed using the distribution function of $\chi^2(N_s - 1)$), we get that R_α admits a unique global minimum on $\mathbb{R}_+$ for the following value $\tau_{\min}$ of τ:

$$\tau_{\min} = \frac{N_s - 1}{\chi^2_{1-\frac{\alpha}{2},N_s-1} - \chi^2_{\frac{\alpha}{2},N_s-1}} \log\left(\frac{\chi^2_{1-\frac{\alpha}{2},N_s-1}}{\chi^2_{\frac{\alpha}{2},N_s-1}}\right) \quad .$$

In particular, R_α is strictly decreasing on $[0, \tau_{\min}[$ and strictly increasing on $]\tau_{\min}, +\infty[$.

Consequently the intermediate value theorem ensures that R_α defines a bijection between $]0, \tau_{\min}]$ and $[R_\alpha(\tau_{\min}), 1[$, but also between $[\tau_{\min}, +\infty[$ and $[R_\alpha(\tau_{\min}), 1[$.

Consider now $\gamma > 0$ such that $(1+\gamma)\alpha < 1$. Notably, we have

$$1 > (1+\gamma)\alpha > \alpha \; (= R_\alpha(1)) \geq R_\alpha(\tau_{\min}) \quad .$$

Hence, the equation $R_\alpha(\tau) = (1+\gamma)\alpha$ admits exactly two solutions $\tau_\gamma^{(1)} \in]0, \tau_{\min}[$ and $\tau_\gamma^{(2)} \in]\tau_{\min}, +\infty[$. Moreover, considering the variations of R_α, we have $\forall \tau \in [\tau_\gamma^{(1)}, \tau_\gamma^{(2)}]$, $R_\alpha(\tau) \leq (1+\gamma)\alpha$ and also $1 \in]\tau_\gamma^{(1)}, \tau_\gamma^{(2)}[$. Introduce then the threshold

$$\eta_\alpha(N_s, \gamma) = \min\{1 - \tau_\gamma^{(1)}; \tau_\gamma^{(2)} - 1\} \quad .$$

Given that by definition of $\eta_\alpha(N_s, \gamma)$, $[1 - \eta_\alpha(N_s, \gamma), 1 + \eta_\alpha(N_s, \gamma)] \subset [\tau_\gamma^{(1)}, \tau_\gamma^{(2)}]$, we have

$$\forall \tau > 0 \text{ such that } |\tau - 1| \leq \eta_\alpha(N_s, \gamma), \quad R_\alpha(\tau) \leq (1+\gamma)\alpha \quad . \tag{3.13}$$

Notice now that for a any $\boldsymbol{c} \in \mathbb{R}^n$, the quantity $|\tau(\boldsymbol{c}) - 1|$ can be expressed as

$$|\tau(\boldsymbol{c}) - 1| = \left|\frac{\sigma_0^2(\boldsymbol{c}) - \sigma^2(\boldsymbol{c})}{\sigma^2(\boldsymbol{c})}\right| = \left|\frac{\boldsymbol{c}^T(f(\boldsymbol{S}) - \mathcal{S}_m[\sqrt{f}]^2(\boldsymbol{S}))\boldsymbol{c}}{\boldsymbol{c}^T \mathcal{S}_m[\sqrt{f}]^2(\boldsymbol{S})\boldsymbol{c}}\right| = \left|\frac{\boldsymbol{c}^T(f(\boldsymbol{S}) - \mathcal{S}_m[\sqrt{f}]^2(\boldsymbol{S}))\boldsymbol{c}}{\|\mathcal{S}_m[\sqrt{f}](\boldsymbol{S})\boldsymbol{c}\|_2^2}\right| .$$

Introducing the vector $\widehat{\boldsymbol{c}} = \mathcal{S}_m[\sqrt{f}](\boldsymbol{S})\boldsymbol{c}$ and using the definition of graph filters, we get

$$|\tau(\boldsymbol{c}) - 1| = \left| \left(\frac{\widehat{\boldsymbol{c}}}{\|\widehat{\boldsymbol{c}}\|_2} \right)^T \left(\mathcal{S}_m[\sqrt{f}](\boldsymbol{S}) \right)^{-1} \left(f(\boldsymbol{S}) - \mathcal{S}_m[\sqrt{f}]^2(\boldsymbol{S}) \right) \left(\mathcal{S}_m[\sqrt{f}](\boldsymbol{S}) \right)^{-1} \left(\frac{\widehat{\boldsymbol{c}}}{\|\widehat{\boldsymbol{c}}\|_2} \right) \right|$$
$$= \left| \left(\frac{\widehat{\boldsymbol{c}}}{\|\widehat{\boldsymbol{c}}\|_2} \right)^T \left(\left(\frac{f - \mathcal{S}_m[\sqrt{f}]^2}{\mathcal{S}_m[\sqrt{f}]^2} \right)(\boldsymbol{S}) \right) \left(\frac{\widehat{\boldsymbol{c}}}{\|\widehat{\boldsymbol{c}}\|_2} \right) \right| \quad .$$

Hence $|\tau(\boldsymbol{c}) - 1|$ can be expressed as the modulus of the Rayleigh quotient of a matrix with respect to a vector depending on $\boldsymbol{c}$. It can therefore be upper-bounded, for any $\boldsymbol{c} \in \mathbb{R}^n$, by the eigenvalue of this matrix that has the largest magnitude. In our case, this gives

$$\forall \boldsymbol{c} \in \mathbb{R}^n, \quad |\tau(\boldsymbol{c}) - 1| \leq \max_{k \in [\![1,n]\!]} \left| \frac{f(\lambda_k) - \mathcal{S}_m[\sqrt{f}](\lambda_k)^2}{\mathcal{S}_m[\sqrt{f}](\lambda_k)^2} \right| \quad ,$$

where $\lambda_1, \ldots, \lambda_n$ denote the eigenvalues of $\boldsymbol{S}$. Hence by imposing for an interval $[a, b]$ containing all the eigenvalues of $\boldsymbol{S}$, the condition $\widehat{\varepsilon}_m \leq \eta_\alpha(N_s, \gamma)$ we will get in particular that $\max_{k \in [\![1,n]\!]} \left| \frac{f(\lambda_k) - \mathcal{S}_m[\sqrt{f}](\lambda_k)^2}{\mathcal{S}_m[\sqrt{f}](\lambda_k)^2} \right| \leq \eta_\alpha(N_s, \gamma)$ and therefore that for any $\boldsymbol{c} \in \mathbb{R}^n$, $|\tau(\boldsymbol{c}) - 1| \leq \eta_\alpha(N_s, \gamma)$ which concludes the proof according to Equation (3.13).

□

Therefore, if Equation (3.11) is satisfied, then, for any $\boldsymbol{c}$, hypothesis $H_0^{\boldsymbol{c}}$ is actually rejected (with significance α) with a probability less than $(1 + \gamma)\alpha$. This probability would have been equal to α if the samples were generated using the right covariance matrix. Therefore, the parameter γ represents relative increase of the rejection probability due to the fact that the samples are generated using an approximation of the target distribution.

As detailed in the proof, the bound $\eta_\alpha(N_s, \gamma)$ solely depends on the specification of the characteristics of the statistical test: the sample size N_s, the significance level α and the tolerated increase of probability of rejection γ. Namely, it is given by:

$$\eta_\alpha(N_s, \gamma) = \min\{1 - \tau_\gamma^{(1)}; \tau_\gamma^{(2)} - 1\} \quad ,$$

where $\tau_\gamma^{(1)}$ and $\tau_\gamma^{(2)}$ are the two solutions of the equation:

$$1 - \left(F_{\chi^2(N_s-1)} \left(\chi^2_{1-\frac{\alpha}{2}, N_s-1} \tau \right) - F_{\chi^2(N_s-1)} \left(\chi^2_{\frac{\alpha}{2}, N_s-1} \tau \right) \right) = (1 + \gamma)\alpha \quad . \tag{3.14}$$

Hence, once N_s, α and γ are fixed, $\eta_\alpha(N_s, \gamma)$ can be numerically computed by solving Equation (3.14) using any root finding algorithm such as the bisection method or even better Newton's method given that the derivative of the function can be analytically computed (Press et al., 2007). Besides, the fact that the disjoint intervals on which each one of the solutions lies are known can be used to ease the root finding process. Tables 3.1 and 3.2 give values of the tolerance $\eta_\alpha(N_s, \gamma)$ produced this way, for various sample sizes N_s and thresholds γ. The significance is fixed at $\alpha = 0.05$ for Table 3.1 and $\alpha = 0.01$ for Table 3.2.

Finally, note that given that $\mathcal{S}_m[\sqrt{f}]$ is defined as the truncation of a Chebyshev series at an order m, this order can be determined by specifying the parameters of the statistical test the user would want its simulations to pass, along with a tolerated error in variance. These parameters would in turn yield a value of $\eta_\alpha(N_s, \gamma)$ and therefore set a bound for the polynomial approximation error $\widehat{\varepsilon}_m$. The order of truncation m is then chosen so that $\widehat{\varepsilon}_m \leq \eta_\alpha(N_s, \gamma)$. This approach will be used in Section 9.1, when dealing with explicit examples of functions f.

This section therefore provided an actual criterion that can be used to set the order of approximation in the filtering step of the Chebyshev simulation algorithm (cf. Algorithm 3.3), so that the resulting simulations have "good enough" statistical properties. In the next section, we link the Chebyshev simulation algorithm to Krylov subspace approaches, thus providing a new insight on this algorithm.

γ	Sample size N_s					
	50	100	500	1000	5000	10000
0.1%	6.40e-04	6.20e-04	5.40e-04	4.80e-04	3.00e-04	2.40e-04
1%	5.44e-03	4.80e-03	3.04e-03	2.36e-03	1.20e-03	8.60e-04
5%	1.89e-02	1.51e-02	8.06e-03	5.94e-03	2.82e-03	2.02e-03
10%	3.00e-02	2.33e-02	1.18e-02	8.64e-03	4.02e-03	2.88e-03
20%	4.59e-02	3.48e-02	1.71e-02	1.24e-02	5.74e-03	4.08e-03
50%	7.66e-02	5.71e-02	2.75e-02	1.98e-02	9.08e-03	6.46e-03
100%	1.10e-01	8.12e-02	3.89e-02	2.80e-02	1.28e-02	9.10e-03

Table 3.1: Values of the precision threshold $\eta_\alpha(N_s, \gamma)$ for different values of sample size N_s and of degradation of the type I error γ. The significance of the test is $\alpha = 0.05$.

γ	Sample size N_s					
	50	100	500	1000	5000	10000
0.1%	4.00e-04	4.00e-04	3.60e-04	3.20e-04	2.20e-04	1.80e-04
1%	3.56e-03	3.24e-03	2.20e-03	1.74e-03	9.20e-04	6.60e-04
5%	1.33e-02	1.09e-02	6.06e-03	4.52e-03	2.18e-03	1.56e-03
10%	2.16e-02	1.71e-02	9.00e-03	6.62e-03	3.12e-03	2.24e-03
20%	3.36e-02	2.59e-02	1.31e-02	9.54e-03	4.44e-03	3.18e-03
50%	5.67e-02	4.28e-02	2.10e-02	1.52e-02	7.00e-03	5.00e-03
100%	8.11e-02	6.07e-02	2.94e-02	2.12e-02	9.76e-03	6.96e-03

Table 3.2: Values of the precision threshold $\eta_\alpha(N_s, \gamma)$ for different values of sample size N_s and of degradation of the type I error γ. The significance of the test is $\alpha = 0.01$

3.3 Relation to Krylov subspace methods

3.3.1 Background: Krylov subspace approach

Krylov subspaces provide a framework for the study of some of the most used iterative algorithms used to solve eigenvalue problems and linear systems involving a matrix $\boldsymbol{A} \in \mathscr{M}_n(\mathbb{R})$ (Del Corso et al., 2015). The idea behind such algorithms is to iteratively generate a sequence of approximate solutions of the problem while relying at each iteration on recurrence relations based on matrix-vector products involving $\boldsymbol{A}$. The approximate solution obtained at the m-th iteration step lies in the subspace $\mathcal{K}_m(\boldsymbol{A}, \boldsymbol{z})$ defined for some problem-dependent $\boldsymbol{z} \in \mathbb{R}^n$ and called Krylov subspace of dimension m generated by $\boldsymbol{A}$ and $\boldsymbol{z}$:

$$\mathcal{K}_m(\boldsymbol{A}, \boldsymbol{z}) = \text{span}\{\boldsymbol{z}, \boldsymbol{A}\boldsymbol{z}, \dots, \boldsymbol{A}^{m-1}\boldsymbol{z}\} = \{\pi(\boldsymbol{A})\boldsymbol{z} : \pi \text{ polynomial of degree } < m\} \quad .$$

In particular, $\mathcal{K}_m(\boldsymbol{A}, \boldsymbol{z})$ is a vector space of dimension at most n, the size of the matrix $\boldsymbol{A}$. An orthonormal basis of $\mathcal{K}_m(\boldsymbol{A}, \boldsymbol{z})$ can be constructed using the Lanczos algorithm (Del Corso et al., 2015; Golub and Van Loan, 1996b), which implements a Gram–Schmidt orthogonalization technique, as outlined in Algorithm 3.4.

In Algorithm 3.4 note that if $\exists j < m$ such that $\delta_{j+1} = 0$, the algorithm stops meaning that $\mathcal{K}_m(\boldsymbol{A}, \boldsymbol{z})$ has dimension j with $[\boldsymbol{v}_1 | \dots | \boldsymbol{v}_j]$ as an orthonormal basis. Besides, the orthogonality of the vectors $\{\boldsymbol{v}_j\}$ gives:

$$\boldsymbol{V}_m^T \boldsymbol{V}_m = \boldsymbol{I}_m \quad \text{and} \quad \boldsymbol{V}_m^T \boldsymbol{v}_{m+1} = \boldsymbol{0} \quad .$$

Finally, using the intermediate coefficients computed during the Lanczos algorithm, the resulting

Algorithm 3.4: Lanczos algorithm.

Input: A symmetric matrix $\boldsymbol{A} \in \mathcal{M}_n(\mathbb{R})$, a vector $\boldsymbol{z} \in \mathbb{R}^n$ with $\|\boldsymbol{z}\|_2 = 1$, $m \leq n$.
Output: An orthonormal basis of $\mathcal{K}_m(\boldsymbol{A}, \boldsymbol{z})$.

Initialization: $\boldsymbol{v}_0 = \boldsymbol{0}$, $\boldsymbol{v}_1 = \boldsymbol{z}$, $\delta_1 = 0$;
for j **from** 1 **to** m **do**
  $\boldsymbol{h} \leftarrow \boldsymbol{A}\boldsymbol{v}_j - \delta_j \boldsymbol{v}_{j-1}$;
  $\gamma_j = \boldsymbol{h}^T \boldsymbol{v}_j$;
  $\boldsymbol{k} \leftarrow \boldsymbol{h} - \gamma_j \boldsymbol{v}_j$;
  $\delta_{j+1} = \|\boldsymbol{k}\|_2$;
  $\boldsymbol{v}_{j+1} = \boldsymbol{k}/\delta_{j+1}$;
Return $\boldsymbol{V}_m = [\boldsymbol{v}_1 | \dots | \boldsymbol{v}_m] \in \mathscr{M}_{n,m}(\mathbb{R})$.

basis $\boldsymbol{V}_m$ satisfies the following relation

$$\boldsymbol{A}\boldsymbol{V}_m = \boldsymbol{V}_m \boldsymbol{T}_m + \delta_{m+1} \boldsymbol{v}_{m+1} \boldsymbol{e}_m^T \quad ,$$

where $\boldsymbol{T}_m$ is the tridiagonal matrix defined by

$$\boldsymbol{T}_m = \begin{pmatrix} \gamma_1 & \delta_2 & & \\ \delta_2 & \gamma_2 & \ddots & \\ & \ddots & \ddots & \delta_m \\ & & \delta_m & \gamma_m \end{pmatrix} \quad .$$

In particular, using the orthogonality of $\boldsymbol{V}_m$, this relation becomes

$$\boldsymbol{V}_m^T \boldsymbol{A} \boldsymbol{V}_m = \boldsymbol{T}_m \quad .$$

This last relation can be used to show that eigenvalues of $\boldsymbol{A}$ are well-approximated by those of $\boldsymbol{T}_m$ as m grows, starting from the extremal ones (Golub and Van Loan, 1996b).

Krylov subspaces arise naturally when studying iterative algorithms designed to solve linear systems of the form:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \quad , \tag{3.15}$$

where $\boldsymbol{A} \in \mathscr{M}_n(\mathbb{R})$ is assumed to be invertible and $\boldsymbol{b} \in \mathbb{R}^n$. The following proposition details this relation.

Proposition 3.3.1. *Let $\boldsymbol{A} \in \mathscr{M}_n(\mathbb{R})$ be an invertible matrix. Then there exists a polynomial π of degree at most n such that:*

$$\boldsymbol{A}^{-1} = \pi(\boldsymbol{A}) \quad .$$

Proof. Let $P_{\boldsymbol{A}}$ be the characteristic polynomial of $\boldsymbol{A}$, i.e. the polynomial defined by the relation:

$$P_{\boldsymbol{A}}(X) = |X\boldsymbol{I}_n - \boldsymbol{A}| \quad .$$

In particular, $P_{\boldsymbol{A}}$ is a polynomial of degree n, whose n-th order coefficient is 1 and 0-th order coefficient is $P_{\boldsymbol{A}}(0) = |-\boldsymbol{A}| = (-1)^n |\boldsymbol{A}| \neq 0$. Hence, there exists $c_1, \dots, c_{n-1} \in \mathbb{R}$ such that $P_{\boldsymbol{A}}(X) = X^n + c_{n-1}X^{n-1} + \cdots + c_1 X + (-1)^n |\boldsymbol{A}|$. The Cayley-Hamilton theorem states that $P_{\boldsymbol{A}}(\boldsymbol{A}) = \boldsymbol{0}$ (Friedberg et al., 2003, Theorem 5.23). Hence,

$$\frac{(-1)^{n-1}}{|\boldsymbol{A}|}(\boldsymbol{A}^{n-1} + c_{n-1}\boldsymbol{A}^{n-2} + \cdots + c_1 \boldsymbol{I}_n)\boldsymbol{A} = \boldsymbol{I}_n \quad .$$

Denoting π the polynomial of degree $n-1$ defined by $\pi(X) = ((-1)^{n-1}/|\boldsymbol{A}|)(X^{n-1} + c_{n-1}X^{n-2} + \cdots + c_1)$ then gives $\pi(\boldsymbol{A})\boldsymbol{A} = \boldsymbol{A}\pi(\boldsymbol{A}) = \boldsymbol{I}_n$ and therefore $\pi(\boldsymbol{A}) = \boldsymbol{A}^{-1}$. □

Consequently, the solution $\boldsymbol{x}^* = \boldsymbol{A}^{-1}\boldsymbol{b}$ of Equation (3.15) can also be written as $\boldsymbol{x}^* = \pi(\boldsymbol{A})\boldsymbol{b}$ for a polynomial π of degree (at most) $n-1$ and therefore $\boldsymbol{x}^*$ lies in the Krylov subspace $\mathcal{K}_n(\boldsymbol{A}, \boldsymbol{b})$. In particular, if $\boldsymbol{x}^{(0)}$ denotes an initial guess for $\boldsymbol{x}^*$:

$$\boldsymbol{x}^* - \boldsymbol{x}^{(0)} = \boldsymbol{A}^{-1}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(0)}) = \pi(\boldsymbol{A})\boldsymbol{r}^{(0)} \in \mathcal{K}_n(\boldsymbol{A}, \boldsymbol{r}^{(0)}) \quad ,$$

where $\boldsymbol{r}^{(0)} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(0)}$ denotes a vector called initial residual. A whole class of iterative algorithms, called projection methods, build on this observation to produce approximations of the solution $\boldsymbol{x}^*$ starting from an initial guess by computing orthogonal projections on Krylov subspaces of growing dimension (Saad, 2003). Among them, the generalized minimal residual (GMRES) algorithm and the conjugate gradient algorithm, designed to solve linear systems where $\boldsymbol{A}$ is respectively any invertible square matrix or symmetric positive definite matrix.

3.3.2 Link to the Chebyshev simulation algorithm

In this section, the relation between the Chebyshev simulation algorithm and Krylov subspaces is exposed, and a comparison with a more standard Krylov subspace approach to generate samples from a $\boldsymbol{S}$-stationary SGS with known spectral density is presented.

Recall that Section 3.1.2 provides a direct way to generate samples of stationary SGS with spectral density f. Denote $\boldsymbol{x}$ such a vector:

$$\boldsymbol{x} = \sqrt{f}(\boldsymbol{S})\boldsymbol{w} \quad , \tag{3.16}$$

where $\boldsymbol{w}$ is a realization of a white signal. On the other hand, the Chebyshev simulation algorithm yields, for an order of approximation $m \in \mathbb{N}$, a vector $\boldsymbol{x}^{(m)}$ given by

$$\boldsymbol{x}_C^{(m)} = \mathcal{S}_m[\sqrt{f}](\boldsymbol{S})\boldsymbol{w} \quad , \tag{3.17}$$

where $\mathcal{S}_[\sqrt{f}]$ is a polynomial of degree m defined as the Chebyshev sum (or interpolant) of order m of the function $x \mapsto \sqrt{f(x)}$ on an interval $[a, b]$ containing the eigenvalues of $\boldsymbol{S}$.

Note consequently that $\boldsymbol{x}_C^{(m)} \in \mathcal{K}_{m+1}(\boldsymbol{S}, \boldsymbol{w})$. Besides, the Chebyshev simulation algorithm can basically be seen as an iterative algorithm. Indeed, to compute $\boldsymbol{x}_C^{(m)}$ for any given m every $\boldsymbol{x}_C^{(k)}$ for $0 \leq k < m$ is successively computed and is simply updated to generate $\boldsymbol{x}_C^{(k+1)}$. This justifies the fact that Chebyshev simulations can be considered as a Krylov subspace approach.

A standard approach using Krylov subspaces to generate samples from a Gaussian vector with known covariance (or precision) matrix uses the Lanczos algorithm to come up with an approximation of $\boldsymbol{x}$ (Simpson et al., 2008). Indeed, in exact arithmetic, this algorithm can provide an orthonormal basis of $\mathcal{K}_{m+1}(\boldsymbol{S}, \boldsymbol{\varepsilon})$ (Golub and Van Loan, 1996b). $\boldsymbol{x}$ can then be approximated by (Frommer and Simoncini, 2008; Simpson et al., 2008)

$$\boldsymbol{x}_L^{(m)} = \|\boldsymbol{w}\|_2 \boldsymbol{V}_{m+1}\sqrt{f}(\boldsymbol{T}_{m+1})\boldsymbol{e}_1 \quad , \tag{3.18}$$

where $\boldsymbol{e}_1 = (1\ 0\ \ldots\ 0)^T \in \mathbb{R}^n$, $\boldsymbol{T}_{m+1}$ is a tridiagonal (symmetric) matrix of size $m+1$ and $\boldsymbol{V}_{m+1}$ is a matrix containing the $m+1$ vectors of the orthonormal basis of $\mathcal{K}_{m+1}(\boldsymbol{S}, \boldsymbol{w})$, both matrices being products of the Lanczos algorithm.

The cost associated with computing $\boldsymbol{x}_L^{(m)}$ can be decomposed as follows :

- Run the Lanczos algorithm for m iterations: this represents a cost of $\mathcal{O}(md_{nz}n)$ operations, where d_{nz} is the mean number of non-zero values in a row of $\boldsymbol{S}$ (cf. Algorithm 3.4).
- Then, compute Equation (3.18): this involves the full diagonalization of the symmetric tridiagonal matrix $\boldsymbol{T}_{m+1}$, which is an $\mathcal{O}((m+1)^3)$ operation using for instance LAPACK's eigensolvers (Demmel et al., 2008). Apply then a matrix-vector product with $\boldsymbol{V}_{m+1}$. Hence, the overall cost of this step is $\mathcal{O}((m+1)^3 + nm)$ operations.

Computing $\boldsymbol{x}_L^{(m)}$ therefore comes at an overall cost of $\mathcal{O}(md_{nz}n + m^3)$ operations. Regarding the storage needs of this process, the matrix $\boldsymbol{V}_{m+1}$ and the eigendecomposition of $\boldsymbol{T}_{m+1}$ need to be stored, which requires a storage need of $\mathcal{O}(mn + m^2)$.

From Section 2.2, it is clear that the Chebyshev simulation algorithm requires less operations and storage space to generate an approximation of $\boldsymbol{x}$ from the same Krylov subspace. But on

	Lanczos	Chebyshev
Computational cost	$\mathcal{O}(md_{nz}n + m^3)$	$\mathcal{O}(md_{nz}n)$
Storage needs	$\mathcal{O}(mn + m^2)$	$\mathcal{O}(n)$
Approximation error	$\mathcal{O}(\delta_m)$	$\mathcal{O}(\delta_m \log m)$

Table 3.3: Comparison between the Lanczos algorithm and our Chebyshev algorithm after m iterations, for the simulation of a sample from stationary SGS.

the other hand, at the same approximation order m, the quality of the approximation obtained using the Lanczos algorithm will be better than the one using the Chebyshev algorithm. Indeed, in the Lanczos case (still in exact arithmetic) this approximation error satisfies (Musco et al., 2017)

$$\|\boldsymbol{x} - \boldsymbol{x}_L^{(m)}\|_2 \leq 2\|\boldsymbol{w}\|_2 \delta_m, \quad \delta_m = \min_{\substack{\pi \text{ polynomial} \\ \text{of degree } \leq m}} \max_{x \in [\lambda_{\min}, \lambda_{\max}]} |\sqrt{f(x)} - \pi(x)| \quad ,$$

where $\lambda_{\min}$ (resp. $\lambda_{\max}$) denotes the smallest (resp. largest) eigenvalue of $\boldsymbol{S}$. Thus it yields in the Lanczos case an error of order $\mathcal{O}(\delta_m)$. In the Chebyshev case, the approximation error satisfies

$$\|\boldsymbol{x} - \boldsymbol{x}_C^{(m)}\|_2^2 = \| \left(\sqrt{f}(\boldsymbol{S}) - \mathcal{S}_m[\sqrt{f}](\boldsymbol{S}) \right) \boldsymbol{w} \|_2^2 = \frac{\boldsymbol{w}^T \left(\sqrt{f}(\boldsymbol{S}) - \mathcal{S}_m[\sqrt{f}](\boldsymbol{S}) \right) \boldsymbol{w}}{\|\boldsymbol{w}\|_2^2} \|\boldsymbol{w}\|_2^2 \quad .$$

Noting the Rayleigh equation in this last expression, we can upper-bound it by the largest eigenvalue of the matrix from which it is defined. Hence, by taking the square-root,

$$\begin{aligned} \|\boldsymbol{x} - \boldsymbol{x}_C^{(m)}\|_2 &\leq \|\boldsymbol{w}\|_2 \max_{k \in [\![1,n]\!]} |\sqrt{f(\lambda_k)} - \mathcal{S}_m[\sqrt{f}](\lambda_k)| \\ &\leq \|\boldsymbol{w}\|_2 \max_{\lambda \in [\lambda_{\min}, \lambda_{\max}]} |\sqrt{f(\lambda)} - \mathcal{S}_m[\sqrt{f}](\lambda)| \quad . \end{aligned}$$

This last estimate can be bounded using δ_m and the Lebesgue constant λ_m, thus giving for the Chebyshev approximation an error of order $\mathcal{O}(\lambda_m \delta_m) = \mathcal{O}(\delta_m \log m)$ (Mason and Handscomb, 2002). The results of the comparison between the Lanczos algorithm and our Chebyshev algorithm are summed up in Table 3.3.

For small values of m the Lanczos algorithms is more adequate as it provides an approximation with a lower error. Its main flaw resides in the fact that, contrary to our Chebyshev algorithm, the storage needs grow linearly with the order of approximation. Hence for large problems (i.e. when n is large), a restriction on the order of approximation has to be set according to the storage space available to the user.

In order to tackle this storage problem, some adjustments can be made to the original Lanczos algorithm (Aune et al., 2013). For instance, restarting procedures allow to work with a fixed number of stored basis vectors of the Krylov space. However, these methods result in a loss of approximation accuracy and push to use complex preconditioning techniques in order to improve the convergence speed of the algorithm, which in turn increases the overall computational cost (Simpson et al., 2008). The Chebyshev simulation algorithm doesn't share this storage flaw, allowing it to make up for its relative lack of precision by the possibility to work with much higher orders of approximation without the headache of finding the right variation of Lanczos algorithm[1] to use.

Another attractive feature of the Chebyshev algorithm is the statistical stopping criterion derived in Section 3.2.2. This criterion was established by using the fact that $\boldsymbol{x}_C^{(m)}$ could be written as $\boldsymbol{x}_C^{(m)} = \pi_m(\boldsymbol{S})\boldsymbol{\varepsilon}$ where the coefficients defining π_m are deterministic (which in our case means that they are not linked to $\boldsymbol{w}$) and that therefore $\boldsymbol{x}_C^{(m)}$ is a Gaussian vector with known covariance.

[1]Note also that the comparison is carried out under the assumption of exact arithmetic. In floating points computations, a loss of orthogonality of $\boldsymbol{V}_{m+1}$ is observed as m grows, leading to larger approximation errors (Musco et al., 2017) and forcing the user to adapt the algorithm using workarounds such as re-orthogonalization techniques or restart techniques (thus increasing the overall complexity of the algorithm).

This is no longer the case when considering the Lanczos algorithm given that these same coefficients would effectively depend on the entries of $\boldsymbol{w}$ as this vector is used to compute the matrices $\boldsymbol{V}_{m+1}$ and $\boldsymbol{T}_{m+1}$ used to define $\boldsymbol{x}_L^{(m)}$. The only available stopping criteria for the Lanczos algorithm are therefore linked to the actual numerical approximation error $\|\boldsymbol{x} - \boldsymbol{x}_L^{(m)}\|$ and not the statistical properties of the vector we wish to simulate. Moreover, given that in practice $\boldsymbol{x}$ is not available, the stopping criteria actually rely on the link between the Lanczos algorithm and the Conjugate Gradient algorithm, using the residuals of the latter as a bound on the approximation error (Aune et al., 2013).

Conclusion

In this section, algorithms to generate simulations of a stationary SGS were introduced. The focus was put on an approximate simulation algorithm, which we called Chebyshev simulation algorithm, and that was based on Chebyshev graph filtering operations.

The numerical approximation error of the simulations generated by the Chebyshev algorithm were computed, and led to concentration inequalities linking the accuracy of the polynomial approximation used in the filtering step and the error between the simulation vector and a vector that is known to have the right statistical properties.

Going then a step further, the statistical properties of the simulated vectors were directly derived and compared to the targeted ones through an approach based on statistical tests. This yielded criteria on the accuracy of the polynomial approximation used in the filtering step so that the simulated vectors could "pass" for vectors with the targeted statistical properties.

Finally, the Chebyshev simulation algorithm was presented as a Krylov subspace approach, and compared to a more standard method of simulation from this class of algorithms, based on the Lanczos algorithm. Both methods produce an estimate of the simulated output using a polynomial approximation of predefined degree. At the same degree, the Lanczos approach will yield a better estimate when considering numerical approximation error. However, the use of Chebyshev simulations is justified by their cheap computational and storage costs, the ability to evaluate statistical errors, and the guarantee that the simulations produced by the algorithm are Gaussian vectors.

4

Prediction of stationary stochastic graph signals

Contents

Résumé

Dans ce chapitre, nous nous intéressons au problème lié à l'estimation d'un signal sur graphe stochastique stationnaire à partir de l'observation partielle et/ou bruitée d'une de ses réalisation. Nous supposons par contre connue sa covariance. Nous présentons des estimateurs adaptés à cette situation, ainsi que des algorithmes et des détails d'implémentation permettant de les utiliser en pratique.

Introduction

Throughout this chapter, $\boldsymbol{S}$ denotes a shift operator of size n defined following Assumption 1.2, meaning that $\boldsymbol{S}$ is a symmetric matrix that relates to the adjacency relations of a simple undirected graph. We focus on the problem of predicting a $\boldsymbol{S}$-stationary stochastic graph signal (SGS) from its incomplete and possibly noisy observation. However, the parameters defining the covariance of the SGS, namely the shift operator $\boldsymbol{S}$ and the spectral density, are assumed to be known. The task of estimating them as well will be tackled in the next chapter.

Hence, our starting point is a vector of observed values derived from a single realization of a $\boldsymbol{S}$-stationary SGS through an affine transform: each observed value is a linear combination of entries of the SGS to which an independent noise variable with known variance is added. The goal is then to come up with a predictor of the realization that gave rise to the observation vector.

To tackle this problem, an approach based on the geostatistical paradigm is adopted, meaning that a predictor of the random signal given the observed data is built instead of trying to predict directly the realization of this random signal (Chilès and Delfiner, 2012).

In the first two sections of this chapter, predictors are derived for the cases where the noise affecting the observations is assumed to be entirely uncorrelated or arising from $\boldsymbol{S}$-stationary signals. The remaining of the chapter then focuses on algorithms used to compute these predictors, and on their implementation. In particular, the same restrictions regarding computational and storage costs as in the previous chapter still apply, meaning that a matrix-free approach is once again adopted.

4.1 Prediction of a stationary graph signal

The problem answered in this section is the following. Let $\boldsymbol{z} \in \mathbb{R}^n$ be a realization of a $\boldsymbol{S}$-stationary SGS $\boldsymbol{Z}$ with known spectral density $f : \mathbb{R} \to \mathbb{R}_+$. We aim at building a predictor of $\boldsymbol{z}$ from its incomplete observation. Formally, we assume that we do not observe $\boldsymbol{z}$ directly, but rather a vector $\boldsymbol{z}_o \in \mathbb{R}^q$, linked to $\boldsymbol{z}$ by the relation

$$\boldsymbol{z}_o = \boldsymbol{M}_o \boldsymbol{z} + \tau \boldsymbol{w}_o \quad , \tag{4.1}$$

where $\boldsymbol{M}_o \in \mathscr{M}_{q,n}(\mathbb{R})$ is a known full-rank matrix called observation matrix, $\boldsymbol{w}_o$ is a q-vector composed of realizations of independent standard Gaussian variables, and $\tau \geq 0$ is a variance parameter. Basically, it is assumed that the observed vector $\boldsymbol{z}_o$ is a linear transform of the original signal $\boldsymbol{z}$ to which a noise component of variance τ^2 is added. Note that taking $\tau = 0$ allows to consider a noise-free model.

In particular, the rather general formulation of Equation (4.1) includes the case where only a few components of a SGS are observed and must be used to reconstruct the whole signal. Then $\boldsymbol{z}_o$ is the vector composed of the components of $\boldsymbol{z}$ that are actually observed, $\boldsymbol{M}_o$ is the matrix that extracts the observed components from $\boldsymbol{z}$ and $\tau = 0$. More precisely, $\boldsymbol{M}_o$ is the matrix whose (k, j)-th element is one if z_j is the k-th observed component of $\boldsymbol{z}_o$ and 0 otherwise.

4.1.1 Kriging predictor in the zero-mean case

We aim at finding a predictor of a signal $\boldsymbol{z}$, conditionally to the observation of $\boldsymbol{z}_o$. Hence, following the geostatistical paradigm (Chilès and Delfiner, 2012), $\boldsymbol{z}$ and $\boldsymbol{z}_o$ are seen as realizations of random vectors $\boldsymbol{Z}$ and $\boldsymbol{Z}_o$ that are linked through the relation

$$\boldsymbol{Z}_o = \boldsymbol{M}_o \boldsymbol{Z} + \tau \boldsymbol{W}_o \quad , \tag{4.2}$$

where $\boldsymbol{Z}$ is a $\boldsymbol{S}$-stationary SGS with spectral density f, $\boldsymbol{W}_0$ is a zero-mean Gaussian vector with covariance matrix $\boldsymbol{I}_q$ and $\boldsymbol{M}_o$ and τ are defined as above. In particular, we call the random vector $\boldsymbol{Z}_o$ *observation process*. predictors of $\boldsymbol{z}$ are then built by considering the conditional distribution of $\boldsymbol{Z}$ given $\boldsymbol{Z}_o = \boldsymbol{z}_o$.

Proposition 4.1.1. *Let $\boldsymbol{Z}$ be a $\boldsymbol{S}$-stationary SGS with spectral density f and let $\boldsymbol{W}_o$ be a vector of independent standard Gaussian variables.*
Let $\boldsymbol{Z}_o$ be the random vector defined by Equation (4.2) for some (deterministic) matrix $\boldsymbol{M}_o \in \mathscr{M}_{q,n}(\mathbb{R})$ and variance parameter $\tau \geq 0$. Denote $\boldsymbol{z}_o$ a particular realization of $\boldsymbol{Z}_o$.

Then,

$$[\boldsymbol{Z}|\boldsymbol{Z}_o = \boldsymbol{z}_o] \sim \mathcal{N}\left(\mathbb{E}[\boldsymbol{Z}|\boldsymbol{z}_o], \mathrm{Var}[\boldsymbol{Z}|\boldsymbol{z}_o]\right) \quad , \tag{4.3}$$

where $\mathbb{E}[\boldsymbol{Z}|\boldsymbol{z}_o]$ is the conditional expectation of $\boldsymbol{Z}$ given $\boldsymbol{Z}_o = \boldsymbol{z}_o$:

$$\mathbb{E}[\boldsymbol{Z}|\boldsymbol{z}_o] = f(\boldsymbol{S})\boldsymbol{M}_o^T \left(\boldsymbol{M}_o f(\boldsymbol{S}) \boldsymbol{M}_o^T + \tau^2 \boldsymbol{I}_q\right)^{-1} \boldsymbol{z}_o \quad ; \tag{4.4}$$

and $\mathrm{Var}[\boldsymbol{Z}|\boldsymbol{z}_o] := \mathbb{E}\left[(\boldsymbol{Z} - \mathbb{E}[\boldsymbol{Z}|\boldsymbol{z}_o])(\boldsymbol{Z} - \mathbb{E}[\boldsymbol{Z}|\boldsymbol{z}_o])^T | \boldsymbol{Z}_o = \boldsymbol{z}_o\right]$ is the conditional covariance matrix of $\boldsymbol{Z}$ given $\boldsymbol{Z}_o = \boldsymbol{z}_o$:

$$\mathrm{Var}[\boldsymbol{Z}|\boldsymbol{z}_o] = f(\boldsymbol{S}) - f(\boldsymbol{S})\boldsymbol{M}_o^T \left(\boldsymbol{M}_o f(\boldsymbol{S}) \boldsymbol{M}_o^T + \tau^2 \boldsymbol{I}_q\right)^{-1} \boldsymbol{M}_o f(\boldsymbol{S}) \quad . \tag{4.5}$$

In particular, whenever f is non-zero on the set of eigenvalues of $\boldsymbol{S}$ and $\tau > 0$, the conditional expectation and covariance matrix of $\boldsymbol{Z}$ can also be expressed as

$$\mathbb{E}[\boldsymbol{Z}|\boldsymbol{z}_o] = \left((\tau^2/f)(\boldsymbol{S}) + \boldsymbol{M}_o^T \boldsymbol{M}_o\right)^{-1} \boldsymbol{M}_o^T \boldsymbol{z}_o \quad , \tag{4.6}$$

and

$$\mathrm{Var}[\boldsymbol{Z}|\boldsymbol{z}_o] = \tau^2 \left((\tau^2/f)(\boldsymbol{S}) + \boldsymbol{M}_o^T \boldsymbol{M}_o\right)^{-1} \quad . \tag{4.7}$$

Proof. See Appendix C.2. □

Circling back to the initial prediction problem, the next proposition justifies why choosing the conditional expectation $\mathbb{E}[\boldsymbol{Z}|\boldsymbol{z}_o]$ as a predictor of $\boldsymbol{Z}$ given the observations $\boldsymbol{z}_o$ is optimal in some sense. We first introduce the notion of best linear unbiased predictor. Let $\boldsymbol{Z} \in \mathbb{R}^n$ and $\boldsymbol{Z}_o \in \mathbb{R}^q$ be two random vectors defined as in Proposition 4.1.1 and $\boldsymbol{z}_o$ be a realization of $\boldsymbol{Z}_o$. A vector $\boldsymbol{z}^* \in \mathbb{R}^n$ is the *best linear unbiased predictor* (BLUP) of a random vector $\boldsymbol{Z}$ given a vector of observations $\boldsymbol{z}_o$ if it is:

- Linear: There exists a $n \times q$ weight matrix, denoted $\boldsymbol{K}$, and a vector $\boldsymbol{\mu} \in \mathbb{R}^n$ such that $\boldsymbol{z}^* = \boldsymbol{\mu} + \boldsymbol{K}\boldsymbol{z}_o$. Hence each entry of $\boldsymbol{Z}$ is predicted by a linear combination of the observations in $\boldsymbol{z}_o$.
- Unbiased: $\mathbb{E}[\boldsymbol{Z}^* - \boldsymbol{Z}] = \boldsymbol{0}$ where $\boldsymbol{Z}^* = \boldsymbol{\mu} + \boldsymbol{K}\boldsymbol{Z}_o$, , i.e. the error term $\boldsymbol{Z}^* - \boldsymbol{Z}$ is zero-mean.
- Minimal variance: $\boldsymbol{K}$ is the matrix that minimizes $\mathrm{Var}[\boldsymbol{Z}^* - \boldsymbol{Z}]$, i.e. the error term $\boldsymbol{Z}^* - \boldsymbol{Z}$ has minimal variance over all possible linear predictors of $\boldsymbol{Z}$ from $\boldsymbol{Z}_o$.

The next proposition then follows from Proposition 4.1.1.

Proposition 4.1.2. *Let $\boldsymbol{Z}$ and $\boldsymbol{Z}_o$ be two random vectors defined as in Proposition 4.1.1 and $\boldsymbol{z}_o$ be a realization of $\boldsymbol{Z}_o$, considered as an observation of $\boldsymbol{Z}$.*

The conditional expectation $\mathbb{E}[\boldsymbol{Z}|\boldsymbol{z}_o]$ of $\boldsymbol{Z}$ given $\boldsymbol{Z}_o = \boldsymbol{z}_o$ is the best unbiased linear predictor of $\boldsymbol{Z}$ given $\boldsymbol{z}_o$.

Proof. See Appendix C.2. □

Hence, $\boldsymbol{z}^* = \mathbb{E}[\boldsymbol{Z}|\boldsymbol{z}_o]$ is an optimal choice of linear predictor of $\boldsymbol{Z}$ given $\boldsymbol{z}_o$ given that it is unbiased and it ensures that the variance of the error is minimal. By analogy with the simple kriging predictor used in Geostatistics, which is defined in the same manner (Chilès and Delfiner, 2012; Wackernagel, 2013), $\boldsymbol{z}^* = \mathbb{E}[\boldsymbol{Z}|\boldsymbol{z}_o]$ is called *kriging predictor* of $\boldsymbol{Z}$ by $\boldsymbol{z}_o$.

Besides, note that $\boldsymbol{Z}^* = \mathbb{E}[\boldsymbol{Z}|\boldsymbol{Z}_o]$ is the conditional expectation of $\boldsymbol{Z}$ with respect to the random variable $\boldsymbol{Z}_o$. As such, it is also equal to the conditional expectation of $\boldsymbol{Z}$ with respect to $\sigma(\boldsymbol{Z}_o)$, the σ-algebra generated by $\boldsymbol{Z}_o$ (Feller, 1971). Whenever $\boldsymbol{Z}$ and $\boldsymbol{Z}_o$ are square-integrable random variables, $\boldsymbol{Z}^*$ defines an orthogonal projection of $\boldsymbol{Z}$ onto the space of $\sigma(\boldsymbol{Z}_o)$-measurable functions, with respect to the inner product $(X, Y) \mapsto \mathbb{E}[XY]$. As such, $\boldsymbol{Z}^*$ can be interpreted as the projection of $\boldsymbol{Z}$ on the set of random variables that encapsulate information from $\boldsymbol{Z}_o$. In this sense, $\boldsymbol{Z}^*$ is the best representation of $\boldsymbol{Z}$ achievable by a prediction based on $\boldsymbol{Z}_o$.

In the next section, kriging predictors are derived for the case where $\boldsymbol{Z}$ is not necessarily zero-mean.

4.1.2 Kriging predictor in the non-zero mean case

Recalling Definition 1.4.6, let us assume for this subsection that $\boldsymbol{Z}$ is a $\boldsymbol{S}$-stationary SGS with spectral density f and possibly non-zero mean (cf. Section 1.4.4). Hence, there exists a zero-mean $\boldsymbol{S}$-stationary SGS $\boldsymbol{Y}$ with spectral density f, an eigenvector $\boldsymbol{v} \in \mathbb{R}^n$ of $\boldsymbol{S}$ and some $m \in \mathbb{R}$ such that

$$\boldsymbol{Z} = \boldsymbol{Y} + m\boldsymbol{v} \quad . \tag{4.8}$$

Once again, we aim at predicting $\boldsymbol{Z} \in \mathbb{R}^n$ from a vector of observations $\boldsymbol{z}_o \in \mathbb{R}^q$ drawn from a observation process $\boldsymbol{Z}_o$ defined by Equation (4.2).

SGS with known mean

We first assume that both the mean eigenvector $\boldsymbol{v}$ and the mean value m in Equation (4.8) are known.

Proposition 4.1.3. *Let $\boldsymbol{Z}$ be a $\boldsymbol{S}$-stationary SGS with spectral density $f : \mathbb{R} \to \mathbb{R}_+$ and with mean $m\boldsymbol{v}$ where $m \in \mathbb{R}$ and $\boldsymbol{v} \in \mathbb{R}^n$ is an eigenvector of $\boldsymbol{S}$. Let us assume that both m and $\boldsymbol{v}$ are known.*
Then, the BLUP $\boldsymbol{Z}^*$ of $\boldsymbol{Z}$ given a vector of observations $\boldsymbol{z}_o$ given by Equation (4.1) is

$$\boldsymbol{Z}^* = \mathbb{E}[\boldsymbol{Z}|\boldsymbol{z}_o] = m\boldsymbol{v} + f(\boldsymbol{S})\boldsymbol{M}_o^T \left(\boldsymbol{M}_o f(\boldsymbol{S}) \boldsymbol{M}_o^T + \tau^2 \boldsymbol{I}_q\right)^{-1} (\boldsymbol{z}_o - m\boldsymbol{M}_o\boldsymbol{v}) \quad . \tag{4.9}$$

In the case where f is non-zero on the set of eigenvalues of $\boldsymbol{S}$ and $\tau > 0$, we have the following equivalent formulation of the kriging predictor:

$$\boldsymbol{Z}^* = \mathbb{E}[\boldsymbol{Z}|\boldsymbol{z}_o] = m\boldsymbol{v} + \left((\tau^2/f)(\boldsymbol{S}) + \boldsymbol{M}_o^T\boldsymbol{M}_o\right)^{-1} \boldsymbol{M}_o^T(\boldsymbol{z}_o - m\boldsymbol{M}_o\boldsymbol{v}) \quad . \tag{4.10}$$

Proof. See Appendix C.2. □

Remark 4.1.1. Regarding the conditional covariance matrix, $\mathrm{Var}[\boldsymbol{Z}|\boldsymbol{z}_o]$, simple calculations show that $\mathrm{Var}[\boldsymbol{Z}|\boldsymbol{z}_o] = \mathrm{Var}[\boldsymbol{Y}|\boldsymbol{y}_o]$, and therefore it keeps the same formula as in Equation (4.5) and, when applicable, Equation (4.7).

SGS with unknown mean

We now assume that the mean parameter m is unknown. However the vector $\boldsymbol{v}$ carrying the mean is assumed to be known. The BLUP of $\boldsymbol{Z}$ given $\boldsymbol{z}_o$ then has the following expression.

Proposition 4.1.4. *Let $\boldsymbol{Z}$ be a $\boldsymbol{S}$-stationary SGS with spectral density $f : \mathbb{R} \to \mathbb{R}_+$ and mean $m\boldsymbol{v}$ where $m \in \mathbb{R}$ and $\boldsymbol{v} \in \mathbb{R}^n$ is an eigenvector of $\boldsymbol{S}$. Let us assume that $\boldsymbol{v}$ is known but m is*

unknown.
Then, the BLUP $\boldsymbol{Z}^*$ *of* $\boldsymbol{Z}$ *given a vector of observations* $\boldsymbol{z}_o$ *given by Equation* (4.1) *is*

$$\boldsymbol{Z}^* = \left(\quad f(\boldsymbol{S})\boldsymbol{M}_o^T \quad \middle| \quad \boldsymbol{v} \quad \right) \left(\begin{array}{c|c} \boldsymbol{M}_o f(\boldsymbol{S}) \boldsymbol{M}_o^T + \tau^2 \boldsymbol{I}_q & \boldsymbol{M}_o \boldsymbol{v} \\ \hline (\boldsymbol{M}_o \boldsymbol{v})^T & 0 \end{array} \right)^{-1} \left(\begin{array}{c} \boldsymbol{z}o \\ \hline 0 \end{array} \right) \quad . \tag{4.11}$$

Proof. See Appendix C.2. □

This predictor introduced in the proposition above actually corresponds to the ordinary kriging predictor encountered in Geostatistics (Wackernagel, 2013) and is the BLUP of $\boldsymbol{Z}$.

4.1.3 Conditional simulations

The idea behind conditional simulations is to generate simulations of a stationary SGS that agree with some observation data when the same observation process is applied to them. Considering a $\boldsymbol{S}$-stationary SGS $\boldsymbol{Z}$ and an observation process $\boldsymbol{Z}_o$ defined by Equation (4.1), we assume that we only observe a (single) realization $\boldsymbol{z}_o$ of $\boldsymbol{Z}$. We aim at generating a simulation $\boldsymbol{z}_c$ of $\boldsymbol{Z}$ such that

$$\boldsymbol{M}_o \boldsymbol{z}_c + \tau \boldsymbol{w}_{oc} = \boldsymbol{z}_o \quad ,$$

for some realization of $\boldsymbol{w}_{oc}$ of $\boldsymbol{W}_o$. This is actually equivalent to draw $\boldsymbol{z}_c$ from the conditional distribution of $\boldsymbol{Z}$ given $\boldsymbol{Z}_o = \boldsymbol{z}_o$. Hence, following Proposition 4.1.1,

$$\boldsymbol{z}_c \sim \mathcal{N}\left(\mathbb{E}[\boldsymbol{Z}|\boldsymbol{z}_o], \mathrm{Var}[\boldsymbol{Z}|\boldsymbol{z}_o]\right) \quad , \tag{4.12}$$

where $\mathbb{E}[\boldsymbol{Z}|\boldsymbol{z}_o]$ and $\mathrm{Var}[\boldsymbol{Z}|\boldsymbol{z}_o]$ are defined in Equations (4.4) to (4.7).

Conditional simulations are widely used in Geostatistics for uncertainty assessments when studying complex (spatial) phenomena (Chilès and Delfiner, 2012; Lantuéjoul, 2013). The premise is that each conditional simulation can be interpreted as a possible picture of the phenomenon or an alternative version of the reality of the phenomenon, that is generated while honoring the limited information gathered about it. Using conjointly all these alternative scenarios allows to assess which one of them might be problematic and therefore identify possible outliers.

In the context of SGS, a possible use of conditional simulations would be to compute predictions of non linear functions of $\boldsymbol{Z}$, conditional to some observed data $\boldsymbol{z}_o$. Indeed, if $\boldsymbol{z}_c^{(1)}, \ldots, \boldsymbol{z}_c^{(N)}$ denote a set of $N > 0$ independently generated conditional simulations of $\boldsymbol{Z}$, then for any function F of $\boldsymbol{Z}$, a prediction $F(\boldsymbol{Z})^*$ of $F(\boldsymbol{Z})$ conditional to $\boldsymbol{Z} = \boldsymbol{z}_o$ is given using a Monte-Carlo approach, via the relation

$$F(\boldsymbol{Z})^* = \frac{1}{N} \sum_{k=1}^{N} F(z_c^{(k)}) \quad .$$

Direct approach to conditional simulations

Circling back to the generation of conditional simulations, a direct approach consists in noticing that any conditional simulation $\boldsymbol{z}_c$ following Equation (4.12) is a realization of a random vector $\boldsymbol{Z}_c$ that can be written

$$\boldsymbol{Z}_c = \mathbb{E}[\boldsymbol{Z}|\boldsymbol{z}_o] + \boldsymbol{Z}_{nc}^0 \quad , \tag{4.13}$$

where $\boldsymbol{Z}_{nc}^0$ is a zero-mean Gaussian vector with covariance matrix $\mathrm{Var}[\boldsymbol{Z}|\boldsymbol{z}_o]$. Hence, a conditional simulation $\boldsymbol{z}_c$ is obtained by adding the conditional expectation $\mathbb{E}[\boldsymbol{Z}|\boldsymbol{z}_o]$ to a realization of $\boldsymbol{Z}_{nc}^0$. Realizations of $\boldsymbol{Z}_{nc}^0$ may be obtained by a factorization method (cf. Section 3.1.1) given that their covariance matrix is known but does not exhibit any particular structure that could be used to bypass this method (like for instance them being graph filters). Algorithm 4.1 outlines this procedure.

Algorithm 4.1: Conditional simulation using a direct approach.

Input: Observation matrix $\boldsymbol{M}_o$, Variance parameter τ and observation vector $\boldsymbol{z}_o$.
Spectral density of the signal f of a zero-mean $\boldsymbol{S}$-stationary SGS $\boldsymbol{Z}$.
Output: A simulation of $\boldsymbol{Z}$ conditional to $\boldsymbol{z}_o$.

Compute $\mathbb{E}[\boldsymbol{Z}|\boldsymbol{z}_o]$ using Equation (4.4) or Equation (4.6) ;
Find a matrix $\boldsymbol{B} \in \mathscr{M}_n(\mathbb{R})$ such that $\boldsymbol{B}\boldsymbol{B}^T = \mathrm{Var}[\boldsymbol{Z}|\boldsymbol{z}_o]$, where $\mathrm{Var}[\boldsymbol{Z}|\boldsymbol{z}_o]$ can be equivalently expressed as Equation (4.5) or Equation (4.7) ;
Compute $\boldsymbol{z}_{nc}^0 = \boldsymbol{B}\boldsymbol{\varepsilon}$ where $\boldsymbol{\varepsilon}$ is a vector with independent standard Gaussian entries;
Return $\boldsymbol{z}_c = \mathbb{E}[\boldsymbol{Z}|\boldsymbol{z}_o] + \boldsymbol{z}_{nc}^0$;

The direct approach presented in Algorithm 4.1 has a huge bottleneck: the factorization of $\mathrm{Var}[\boldsymbol{Z}|\boldsymbol{z}_o]$. Contrary to the case where the covariance matrix is a graph filter, the factorization here supposes that first, $\mathrm{Var}[\boldsymbol{Z}|\boldsymbol{z}_o]$ is formed and stored. Forming $\mathrm{Var}[\boldsymbol{Z}|\boldsymbol{z}_o]$ involves to fully form a graph filter which must be avoided as it is a costly operation (cf. Section 4.3.1 for more details). Hence, the direct approach of Algorithm 4.1 is usually discarded when it comes to generate conditional simulation.

Kriging approach to conditional simulations

This second approach for generating conditional simulations builds on the one presented above. It allows to compute conditional simulations of SGSs as long as we know how to compute unconditional simulations of SGSs with known spectral density, and that we know how to compute conditional expectations of these SGS. The former was addressed in Chapter 3. The latter is the purpose of Sections 4.3 and 4.5. We therefore assume for this subsection that both tasks can be performed.

Starting once gain with Equation (4.13), we aim at finding a more efficient way to generate a simulation of $\boldsymbol{Z}_{nc}^0$, which is a Gaussian vector with mean $\boldsymbol{0}$ and covariance matrix $\mathrm{Var}[\boldsymbol{Z}|\boldsymbol{z}_o]$. The following proposition answers this question.

Proposition 4.1.5. *Let $\boldsymbol{Z}$ be a $\boldsymbol{S}$-stationary SGS with spectral density f. Let $\boldsymbol{Z}_o$ be the random vector defined from $\boldsymbol{Z}$ by Equation* (4.2).
Denote $\mathbb{E}[\boldsymbol{Z}|\boldsymbol{Z}_o]$ the conditional expectation of $\boldsymbol{Z}$ given $\boldsymbol{Z}_o$ (which is the random vector obtained by substituting $\boldsymbol{z}_o$ to $\boldsymbol{Z}_o$ in Equation (4.4)*).*

Then,

$$\boldsymbol{Z} - \mathbb{E}[\boldsymbol{Z}|\boldsymbol{Z}_o] \sim \mathcal{N}(\boldsymbol{0}, \mathrm{Var}[\boldsymbol{Z}|\boldsymbol{z}_o]) \quad ,$$

where $\mathrm{Var}[\boldsymbol{Z}|\boldsymbol{z}_o]$ is defined through Equation (4.5) *and only depends on $\boldsymbol{M}_o$, τ and $f(\boldsymbol{S})$.*

Proof. See Appendix C.2. □

Remark 4.1.2. Given that the corresponding expressions are equivalent, $\mathbb{E}[\boldsymbol{Z}|\boldsymbol{Z}_o]$ and $\mathrm{Var}[\boldsymbol{Z}|\boldsymbol{z}_o]$ in Proposition 4.1.5 can also be computed using respectively Equation (4.6) and Equation (4.7).

Consequently, a simulation of $\boldsymbol{Z}_{nc}^0$ can be generated by computing a realization of the random variable $\boldsymbol{Z} - \mathbb{E}[\boldsymbol{Z}|\boldsymbol{Z}_o]$, given that they both have the same distribution. This can be done in three steps:

1. Generate a realization $\boldsymbol{z}'$ of $\boldsymbol{Z}$, which is a $\boldsymbol{S}$-stationary SGS with spectral density f.
2. Compute the vector $\mathbb{E}[\boldsymbol{Z}|\boldsymbol{z}'_o]$ which is obtained by replacing $\boldsymbol{z}_o$ with $\boldsymbol{z}'_o$ in Equation (4.4), where
$$\boldsymbol{z}'_o = \boldsymbol{M}_o\boldsymbol{z}' + \tau\boldsymbol{w}'_o \quad , \tag{4.14}$$
and $\boldsymbol{w}'_o$ is a vector of independent standard Gaussian variables.
3. The actual simulation $\boldsymbol{z}_{nc}^0$ of $\boldsymbol{Z}_{nc}^0$ is given by $\boldsymbol{z}_{nc}^0 = \boldsymbol{z}' - \mathbb{E}[\boldsymbol{Z}|\boldsymbol{z}'_o]$.

Equation (4.13) then gives the following expression for a conditional simulation $\boldsymbol{z}_c$ of $\boldsymbol{z}$:

$$\boldsymbol{z}_c = \mathbb{E}[\boldsymbol{Z}|\boldsymbol{z}_o] + \boldsymbol{z}^0_{nc} = \mathbb{E}[\boldsymbol{Z}|\boldsymbol{z}_o] + \boldsymbol{z}' - \mathbb{E}[\boldsymbol{Z}|\boldsymbol{z}'_o] \quad .$$

In particular, noting that the expression of both $\mathbb{E}[\boldsymbol{Z}|\boldsymbol{z}_o]$ and $\mathbb{E}[\boldsymbol{Z}|\boldsymbol{z}'_o]$ are linear with respect to $\boldsymbol{z}_o$ and $\boldsymbol{z}'_o$, this last equation can be written as

$$\boldsymbol{z}_c = \boldsymbol{z}' + \mathbb{E}[\boldsymbol{Z}|\boldsymbol{z}_o - \boldsymbol{z}'_o] \quad ,$$

where $\mathbb{E}[\boldsymbol{Z}|\boldsymbol{z}_o - \boldsymbol{z}'_o]$ denotes the vector obtained by substituting $\boldsymbol{z}_o$ by $\boldsymbol{z}_o - \boldsymbol{z}'_o$ in Equation (4.4) or Equation (4.6):

$$\begin{aligned} \mathbb{E}[\boldsymbol{Z}|\boldsymbol{z}_o - \boldsymbol{z}'_o] &= f(\boldsymbol{S})\boldsymbol{M}_o^T \left(\boldsymbol{M}_o f(\boldsymbol{S})\boldsymbol{M}_o^T + \tau^2 \boldsymbol{I}_q\right)^{-1} (\boldsymbol{z}_o - \boldsymbol{z}'_o) \\ &= \left((\tau^2/f)(\boldsymbol{S}) + \boldsymbol{M}_o^T\boldsymbol{M}_o\right)^{-1} \boldsymbol{M}_o^T(\boldsymbol{z}_o - \boldsymbol{z}'_o) \quad . \end{aligned} \tag{4.15}$$

Equation (4.15) is used to derive the conditional simulation algorithm outlined in Algorithm 4.2. This algorithm sums up this kriging approach to conditional simulations, that yields a conditional simulation for the cost of an unconditional simulation and a linear prediction of SGS by kriging. The user should note that the second equality in Equation (4.15) is defined only if f is strictly positive over the eigenvalues of $\boldsymbol{S}$ and $\tau > 0$.

Algorithm 4.2: Conditional simulation by kriging.

Input: Observation matrix $\boldsymbol{M}_o$, Variance parameter τ and observation vector $\boldsymbol{z}_o$.
Spectral density of the signal f of a zero-mean $\boldsymbol{S}$-stationary SGS $\boldsymbol{Z}$.
Output: A simulation of $\boldsymbol{Z}$ conditional to $\boldsymbol{z}_o$.

Compute a unconditional simulation $\boldsymbol{z}'$ of $\boldsymbol{Z}$ using one of the algorithms of Section 3.1 ;
Compute $\boldsymbol{z}'_o$ using Equation (4.14) ;
Compute $\mathbb{E}[\boldsymbol{Z}|\boldsymbol{z}_o - \boldsymbol{z}'_o]$ using Equation (4.15) ;
Return $\boldsymbol{z}_c = \boldsymbol{z}' + \mathbb{E}[\boldsymbol{Z}|\boldsymbol{z}_o - \boldsymbol{z}'_o]$;

4.2 Extraction of a stationary graph signal

The prediction problem of Section 4.1 is now extended: correlated noises are indeed added in the observation process. This situation arises naturally in Geostatistics, where the noise affecting a spatial dataset can also presents spatial correlations that can be modeled. We transpose this setting to stochastic graph signals.

Let $\boldsymbol{Z} \in \mathbb{R}^n$ be once again a $\boldsymbol{S}$-stationary SGS with known spectral density $f : \mathbb{R} \to \mathbb{R}_+$. We aim at recovering a predictor of $\boldsymbol{Z}$ from its noisy observation. Formally, we assume that we do not observe $\boldsymbol{Z}$ directly, but rather a vector $\boldsymbol{z}_o \in \mathbb{R}^q$ which is a realization of an observation process $\boldsymbol{Z}_o$ defined by:

$$\boldsymbol{Z}_o = \boldsymbol{M}_o\boldsymbol{Z} + \boldsymbol{M}_1\boldsymbol{Z}_1 + \cdots + \boldsymbol{M}_p\boldsymbol{Z}_p + \tau\boldsymbol{W}_o \quad , \tag{4.16}$$

where:

- $\boldsymbol{M}_o \in \mathscr{M}_{q,n}(\mathbb{R})$ and $\boldsymbol{M}_1 \in \mathscr{M}_{q,n_1}(\mathbb{R}), \ldots, \boldsymbol{M}_p \in \mathscr{M}_{q,n_p}(\mathbb{R})$ are known observation matrices.
- $\boldsymbol{Z}_1 \in \mathbb{R}^{n_1}, \ldots, \boldsymbol{Z}_p \in \mathbb{R}^{n_p}$ are p zero-mean *independent* stationary SGS. In particular, $\forall k \in [\![1, p]\!]$, $\boldsymbol{Z}_k$ is assumed to be stationary with respect to a shift operator $\boldsymbol{S}_k$ and has spectral density f_k, both of which are known.
- $\boldsymbol{W}_o$ is a vector with q *independent* standard Gaussian entries.
- $\tau \geq 0$ is a variance parameter.

We therefore aim at extracting a particular signal $\boldsymbol{Z}$ from the observation of a superposition of independent signals $\boldsymbol{Z}_1, \dots, \boldsymbol{Z}_p, \boldsymbol{W}_o$. In particular, we call structured noises the signals $\boldsymbol{Z}_1, \dots, \boldsymbol{Z}_p$ and unstructured noise the vector $\boldsymbol{W}_0$ in order to introduce a distinction between them. The observation process involves a modification of each structured noise through an observation matrix.

This new problem is a direct generalization of the prediction problem of Section 4.1, which is retrieved when $p = 0$, i.e. where the noise of $\boldsymbol{Z}_o$, defined as the difference $\boldsymbol{Z}_o - \boldsymbol{M}_o\boldsymbol{Z}$, is purely a measurement error. This parallel allows to derive linear predictors of $\boldsymbol{Z}$ in the same way as in Section 4.1.

4.2.1 Linear predictor in the known-mean case

Let us assume for this section that $\boldsymbol{Z}$ is a S-stationary SGS with spectral density f and possibly non-zero mean $m\boldsymbol{v}$ where $m \in \mathbb{R}$ and $\boldsymbol{v} \in \mathbb{R}^n$ is an eigenvector of $\boldsymbol{S}$. In particular $\boldsymbol{Y} := \boldsymbol{Z} - m\boldsymbol{v}$ defines a zero-mean $\boldsymbol{S}$-stationary SGS $\boldsymbol{Y}$ with spectral density f.

We aim at extracting $\boldsymbol{Z} \in \mathbb{R}^n$ from a vector of observations $\boldsymbol{z}_o \in \mathbb{R}^q$ drawn from a observation process $\boldsymbol{Z}_o$ (defined by Equation (4.16)) by building a linear predictor of $\boldsymbol{Z}$ from $\boldsymbol{z}_o$. The structured noises $\boldsymbol{Z}_1, \dots, \boldsymbol{Z}_p$ are still assumed to be zero-mean, as is the unstructured noise $\boldsymbol{W}_o$.

Proposition 4.2.1. *Let $\boldsymbol{Z}$ be a $\boldsymbol{S}$-stationary SGS with spectral density f and known mean $m\boldsymbol{v}$ where $m \in \mathbb{R}$ and $\boldsymbol{v} \in \mathbb{R}^n$ is an eigenvector of $\boldsymbol{S}$. Let $\boldsymbol{z}_o \in \mathbb{R}^q$ be a realization of an observation process $\boldsymbol{Z}_o$ defined by Equation* (4.16).
Then, the BLUP $\boldsymbol{Z}^*$ of $\boldsymbol{Z}$ given $\boldsymbol{z}_o$ is the conditional expectation of $\boldsymbol{Z}$ given $\boldsymbol{Z}_o = \boldsymbol{z}_o$, that is

$$\begin{aligned}\boldsymbol{Z}^* &= \mathbb{E}[\boldsymbol{Z}|\boldsymbol{z}_o] \\ &= m\boldsymbol{v} + f(\boldsymbol{S})\boldsymbol{M}_o^T \left(\boldsymbol{M}_o f(\boldsymbol{S})\boldsymbol{M}_o^T + \sum_{k=1}^{p} \boldsymbol{M}_k f_k(\boldsymbol{S}_k)\boldsymbol{M}_k^T + \tau^2 \boldsymbol{I}_q\right)^{-1} (\boldsymbol{z}_o - m\boldsymbol{M}_o\boldsymbol{v}) \,.\end{aligned} \tag{4.17}$$

Besides, the conditional covariance matrix of $\boldsymbol{Z}$ given $\boldsymbol{Z}_o = \boldsymbol{z}_o$ is given by

$$\mathrm{Var}[\boldsymbol{Z}|\boldsymbol{z}_o] = f(\boldsymbol{S}) - f(\boldsymbol{S})\boldsymbol{M}_o^T \left(\boldsymbol{M}_o f(\boldsymbol{S})\boldsymbol{M}_o^T + \sum_{k=1}^{p} \boldsymbol{M}_k f_k(\boldsymbol{S}_k)\boldsymbol{M}_k^T + \tau^2 \boldsymbol{I}_q\right)^{-1} \boldsymbol{M}_o f(\boldsymbol{S}) \,. \tag{4.18}$$

Proof. See Appendix C.2. □

Other formulations of the solution of the extraction problem can be formulated for the particular case where the spectral density f is non-zero over the set of eigenvalues of $\boldsymbol{S}$ and $\tau > 0$.

Proposition 4.2.2. *Let $\boldsymbol{Z}$ be a zero-mean $\boldsymbol{S}$-stationary SGS with spectral density f and let $\boldsymbol{z}_o \in \mathbb{R}^q$ be a realization of an observation process $\boldsymbol{Z}_o$ defined by Equation* (4.16).
Then, the BLUP $\boldsymbol{Z}^*$ of $\boldsymbol{Z}$ given $\boldsymbol{z}_o$ and the BLUPs $\boldsymbol{Z}_1^*, \dots, \boldsymbol{Z}_p^*$ of $\boldsymbol{Z}_1, \dots, \boldsymbol{Z}_p$ given $\boldsymbol{Z}_o = \boldsymbol{z}_o$ satisfy

$$\begin{pmatrix} \boldsymbol{Z}^* - m\boldsymbol{v} \\ \boldsymbol{Z}_1^* \\ \vdots \\ \boldsymbol{Z}_p^* \end{pmatrix} = \begin{pmatrix} \boldsymbol{M}_o^T\boldsymbol{M}_o + \frac{\tau^2}{f}(\boldsymbol{S}) & \boldsymbol{M}_o^T\boldsymbol{M}_1 & \dots & \boldsymbol{M}_o^T\boldsymbol{M}_p \\ \boldsymbol{M}_1^T\boldsymbol{M}_o & \boldsymbol{M}_1^T\boldsymbol{M}_1 + \frac{\tau^2}{f_1}(\boldsymbol{S}_1) & \dots & \boldsymbol{M}_1^T\boldsymbol{M}_p \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{M}_p^T\boldsymbol{M}_o & \boldsymbol{M}_p^T\boldsymbol{M}_1 & \dots & \boldsymbol{M}_p^T\boldsymbol{M}_p + \frac{\tau^2}{f_p}(\boldsymbol{S}_p) \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{M}_o^T(\boldsymbol{z}_o - m\boldsymbol{M}_o\boldsymbol{v}) \\ \boldsymbol{M}_1^T(\boldsymbol{z}_o - m\boldsymbol{M}_o\boldsymbol{v}) \\ \vdots \\ \boldsymbol{M}_p^T(\boldsymbol{z}_o - m\boldsymbol{M}_o\boldsymbol{v}) \end{pmatrix}. \tag{4.19}$$

Proof. See Appendix C.2. □

4.2.2 Linear predictor in the unknown-mean case

Once again, we assume for this section that $\boldsymbol{Z}$ is a S-stationary SGS with spectral density f and possibly non-zero mean $m\boldsymbol{v}$ where $m \in \mathbb{R}$ and $\boldsymbol{v} \in \mathbb{R}^n$ is an eigenvector of $\boldsymbol{S}$. However we now assume that the mean value m is unknown and the mean eigenvector $\boldsymbol{v}$ is known. The BLUP of $\boldsymbol{Z}$ given $\boldsymbol{z}_o$ has the following expression.

Proposition 4.2.3. *Let $\boldsymbol{Z}$ be a $\boldsymbol{S}$-stationary SGS with spectral density f and mean $m\boldsymbol{v}$ where $m \in \mathbb{R}$ and $\boldsymbol{v} \in \mathbb{R}^n$ is an eigenvector of $\boldsymbol{S}$. Let us assume that $\boldsymbol{v}$ is known but m is unknown. Then, the BLUP $\boldsymbol{Z}^*$ of $\boldsymbol{Z}$ given a vector of observations $\boldsymbol{z}_o$ defined by Equation* (4.16) *is:*

$$\boldsymbol{z}^* = \left(\begin{array}{c|c} f(\boldsymbol{S})\boldsymbol{M}_o & \boldsymbol{v} \end{array}\right) \left(\begin{array}{c|c} \boldsymbol{M}_o f(\boldsymbol{S}) \boldsymbol{M}_o^T + \sum_{k=1}^{p} \boldsymbol{M}_k f_k(\boldsymbol{S}_k) \boldsymbol{M}_k^T + \tau^2 \boldsymbol{I}_q & \boldsymbol{M}_o \boldsymbol{v} \\ \hline (\boldsymbol{M}_o \boldsymbol{v})^T & 0 \end{array}\right)^{-1} \left(\begin{array}{c} \boldsymbol{z}_o \\ \hline 0 \end{array}\right) \tag{4.20}$$

Proof. See Appendix C.2. □

In the next sections, we present numerical methods to effectively solve the prediction and extraction problems that were introduced in the past two sections. As a matter of fact, given that the prediction problem is the particular case of an extraction problem for which there are no structured noises $\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_p$, only this last class of problems will actually be considered from now on.

4.3 Practical implementation in the known-mean case

Let us assume that we aim at extracting a signal $\boldsymbol{Z}$ with known mean $m\boldsymbol{v}$ where $m \in \mathbb{R}$ and $\boldsymbol{v} \in \mathbb{R}^n$ is an eigenvector of $\boldsymbol{S}$, from an observation vector $\boldsymbol{z}_o$ arising from an observation process $\boldsymbol{Z}_o$ defined by Equation (4.16).

4.3.1 Matrix-free formulation of the problem

Propositions 4.2.1 and 4.2.2 provide expressions for the BLUP $\boldsymbol{z}^*$ of $\boldsymbol{Z}$ given $\boldsymbol{z}_o$ that share a common formulation. Indeed, they can be written as:

$$\boldsymbol{z}^* = \boldsymbol{P}\boldsymbol{K}^{-1}\boldsymbol{b} \quad , \tag{4.21}$$

where:

- $\boldsymbol{K}$ is a symmetric positive-definite matrix defined from the covariance matrices $f(\boldsymbol{S})$, $f_1(\boldsymbol{S}_1), \ldots, f_p(\boldsymbol{S}_p)$, the observation matrices $\boldsymbol{M}_o, \boldsymbol{M}_1, \ldots, \boldsymbol{M}_p$ and the variance parameter τ. Let $n_{\boldsymbol{K}}$ be its size.
- $\boldsymbol{b}$ is a $n_{\boldsymbol{K}}$-vector defined from $\boldsymbol{z}_o$ and the observation matrices $\boldsymbol{M}_o, \boldsymbol{M}_1, \ldots, \boldsymbol{M}_p$.
- $\boldsymbol{P}$ is a $n \times n_{\boldsymbol{K}}$ matrix defined from the covariance matrix $f(\boldsymbol{S})$ and the observation matrix $\boldsymbol{M}_o$.

More precisely, the matrices $\boldsymbol{K}$, $\boldsymbol{P}$ and the vector $\boldsymbol{b}$ have the following expression (cf. Proposition 4.2.1):

$$\begin{gathered} \boldsymbol{K} = \left(\boldsymbol{M}_o f(\boldsymbol{S}) \boldsymbol{M}_o^T + \sum_{k=1}^{p} \boldsymbol{M}_k f_k(\boldsymbol{S}_k) \boldsymbol{M}_k^T + \tau^2 \boldsymbol{I}_q\right), \\ \boldsymbol{b} = \boldsymbol{z}_o - m\boldsymbol{M}_o\boldsymbol{v}, \quad \boldsymbol{P} = f(\boldsymbol{S})\boldsymbol{M}_o^T \quad . \end{gathered} \tag{4.22}$$

Whenever the spectral density f of the extracted signal is non-zero over the eigenvalues of $\boldsymbol{S}$ and $\tau > 0$, an alternative formulation is given by (cf. Proposition 4.2.2)

$$\begin{aligned} \boldsymbol{K} &= \begin{pmatrix} \boldsymbol{M}_o^T\boldsymbol{M}_o + \frac{\tau^2}{f}(\boldsymbol{S}) & \boldsymbol{M}_o^T\boldsymbol{M}_1 & \dots & \boldsymbol{M}_o^T\boldsymbol{M}_p \\ \boldsymbol{M}_1^T\boldsymbol{M}_o & \boldsymbol{M}_1^T\boldsymbol{M}_1 + \frac{\tau^2}{f_1}(\boldsymbol{S}_1) & \dots & \boldsymbol{M}_1^T\boldsymbol{M}_p \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{M}_p^T\boldsymbol{M}_o & \boldsymbol{M}_p^T\boldsymbol{M}_1 & \dots & \boldsymbol{M}_p^T\boldsymbol{M}_p + \frac{\tau^2}{f_p}(\boldsymbol{S}_p) \end{pmatrix}, \\ \boldsymbol{b} &= \begin{pmatrix} \boldsymbol{M}_o^T \\ \boldsymbol{M}_1^T \\ \vdots \\ \boldsymbol{M}_p^T \end{pmatrix} (\boldsymbol{z}_o - m\boldsymbol{M}_o\boldsymbol{v}), \quad \boldsymbol{P} = \boldsymbol{I} \quad . \end{aligned} \tag{4.23}$$

In that case, $\boldsymbol{z}^*$ actually corresponds to the best linear predictor of the vector containing the signal $\boldsymbol{Z}$ but also the p structured noise components $\boldsymbol{Z}_1, \dots, \boldsymbol{Z}_p$.

Hence, a straightforward way to get the extracted signal $\boldsymbol{z}^*$ would consist in building the matrices $\boldsymbol{P}$, $\boldsymbol{K}$ and $\boldsymbol{b}$, and actually computing $\boldsymbol{z}^*$ through Equation (4.21). This can be done in two steps:

1. First, compute the term $\boldsymbol{x}^* = \boldsymbol{K}^{-1}\boldsymbol{b}$ by either inverting $\boldsymbol{K}$ and multiplying the inverse with $\boldsymbol{b}$ or more generally by solving the linear system

$$\boldsymbol{K}\boldsymbol{x}^* = \boldsymbol{b} \quad , \tag{4.24}$$

   using any algorithm designed for this purpose.

2. Return $\boldsymbol{z}^* = \boldsymbol{P}\boldsymbol{x}^*$.

In practice, building and storing the matrices $\boldsymbol{K}$ and $\boldsymbol{P}$ in order to directly use them in Equation (4.21) quickly becomes an intractable operation. To understand this, notice that the expression of both matrices involves at least one graph filter. Hence computing and storing $\boldsymbol{K}$ and $\boldsymbol{P}$ actually requires to compute and store at least one graph filter. This can be done using the definition of graph filters, which involves the diagonalization of a shift operator. If the shift operator has size n, this approach would therefore require $\mathcal{O}(n^3)$ operations and a storage space of order $\mathcal{O}(n^2)$ given that the resulting matrix has no reason to be sparse.

Following the idea of Chebyshev filtering, we might think of computing a polynomial approximation of the graph filter. However, doing so now involves matrix-matrix products between the shift operator and a matrix of size n that becomes less and less sparse as the number of products grows. The whole point of the Chebyshev approach would therefore be lost: only low-order approximations would be considered otherwise the computation of the graph filter would be as expensive as using the diagonalization method.

Even if we assume that we are able to build any graph filters, a storage problem arises. Take for instance the case of the matrix $\boldsymbol{K}$, whose computation seems inevitable to solve the system of Equation (4.24). Storing $\boldsymbol{K}$ would require $\mathcal{O}(n^2)$ storage space, as it is in general a dense matrix.

Another approach should therefore be used to solve the system of Equation (4.24). Even though computing directly the matrix $\boldsymbol{K}$ is prohibited, computing products between $\boldsymbol{K}$ and vectors of the same size can be done in an efficient way using Chebyshev filtering. Assuming the observation matrices are sparse, the computational and storage cost of computing a product $\boldsymbol{K}\boldsymbol{x}$ can be brought down to roughly the cost of performing $p+1$ graph filtering operations.

In the case where $\boldsymbol{K}$ is defined as in Equation (4.22), a product $\boldsymbol{K}\boldsymbol{x}$ is given by

$$\boldsymbol{K}\boldsymbol{x} = \boldsymbol{M}_o f(\boldsymbol{S})\boldsymbol{M}_o^T\boldsymbol{x} + \sum_{k=1}^{p} \boldsymbol{M}_k f_k(\boldsymbol{S}_k)\boldsymbol{M}_k^T\boldsymbol{x} + \tau^2\boldsymbol{x} \quad ,$$

where each term of the form $\boldsymbol{M}_o f(\boldsymbol{S})\boldsymbol{M}_o^T\boldsymbol{x}$ can be computed in three steps. First, the vector $\boldsymbol{M}_o^T\boldsymbol{x}$ is computed (which is cheap as $\boldsymbol{M}_o$ is sparse). Then Chebyshev filtering is used on the graph filter $f(\boldsymbol{S})$ and the vector $\boldsymbol{M}_o^T\boldsymbol{x}$. And finally, the resulting vector is multiplied by $\boldsymbol{M}_o$.

Similarly, in the case where $\boldsymbol{K}$ is defined as in Equation (4.23), we have

$$\boldsymbol{K}\begin{pmatrix}\boldsymbol{x}\\ \boldsymbol{x}_1\\ \vdots\\ \boldsymbol{x}_p\end{pmatrix} = \begin{pmatrix}\boldsymbol{M}_o^T\left(\boldsymbol{M}_o\boldsymbol{x}+\sum_{k=1}^{p}\boldsymbol{M}_k\boldsymbol{x}_k\right)+\frac{\tau^2}{f}(\boldsymbol{S})\boldsymbol{x}\\ \boldsymbol{M}_1^T\left(\boldsymbol{M}_o\boldsymbol{x}+\sum_{k=1}^{p}\boldsymbol{M}_k\boldsymbol{x}_k\right)+\frac{\tau^2}{f_1}(\boldsymbol{S}_1)\boldsymbol{x}_1\\ \vdots\\ \boldsymbol{M}_p^T\left(\boldsymbol{M}_o\boldsymbol{x}+\sum_{k=1}^{p}\boldsymbol{M}_k\boldsymbol{x}_k\right)+\frac{\tau^2}{f_p}(\boldsymbol{S}_p)\boldsymbol{x}_p\end{pmatrix} \quad ,$$

where each term of the form $(\tau^2/f)(\boldsymbol{S})\boldsymbol{x}$ is computed using Chebyshev filtering. Note also that the term $(\boldsymbol{M}_o\boldsymbol{x}+\sum_{k=1}^{p}\boldsymbol{M}_k\boldsymbol{x}_k)$ can be computed once, stored, and used for every subvector of the product.

Hence efficient programs based on Chebyshev filtering can be written to compute the product $\boldsymbol{K}\boldsymbol{x}$ for any vector $\boldsymbol{x}$, and do not require to actually build the matrix $\boldsymbol{K}$. The idea is then to use "matrix-free" solvers to solve Equation (4.24). Such solvers have the desirable properties that they are able to solve linear systems using only products between vectors and the matrix defining the linear system. In particular, they do not require to explicitly have access to elements of this matrix and therefore to have them stored somewhere.

Note finally that if a method is found to efficiently solve Equation (4.24), then computing the actual extracted signal is done by simply multiplying the obtained solution by the matrix $\boldsymbol{P}$. This last operation can once again be performed using Chebyshev algorithm and therefore amounts to the cost of at most one graph filtering operation. In the following, we therefore focus solely on the numerical resolution of Equation (4.24).

4.3.2 Optimization framework

Note that the solution $\boldsymbol{x}^*$ of Equation (4.24) satisfies:

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}\in\mathbb{R}^{n_{\boldsymbol{K}}}}{\operatorname{argmin}} f_{\text{opt}}(\boldsymbol{x}), \quad \text{where} \quad f_{\text{opt}}(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{K}\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{x} \quad . \tag{4.25}$$

Indeed, given that $\boldsymbol{K}$ is a positive definite matrix, the function $f_{\text{opt}} : \mathbb{R}^{n_{\boldsymbol{K}}} \to \mathbb{R}$ is called *objective function* and is convex, and therefore its stationary point is its unique minimum. In particular,

$$\forall \boldsymbol{x} \in \mathbb{R}^{n_{\boldsymbol{K}}}, \quad \nabla f_{\text{opt}}(\boldsymbol{x}) = \boldsymbol{K}\boldsymbol{x} - \boldsymbol{b} \quad ,$$

and therefore the (unique) stationary point of f_{opt} is $\boldsymbol{x}^* = \boldsymbol{K}^{-1}\boldsymbol{b}$. Computing $\boldsymbol{x}^*$ is therefore equivalent to solving the minimization problem defined by Equation (4.25).

Remark 4.3.1. Let us denote $\|\cdot\|_{\boldsymbol{K}}$ the norm defined for any $\boldsymbol{x} \in \mathbb{R}^{n_{\boldsymbol{K}}}$ by $\|\boldsymbol{x}\|_{\boldsymbol{K}} = \sqrt{\boldsymbol{x}^T\boldsymbol{K}\boldsymbol{x}}$. Then, $\forall \boldsymbol{x} \in \mathbb{R}^{n_{\boldsymbol{K}}}$,

$$\|\boldsymbol{x} - \boldsymbol{x}^*\|_{\boldsymbol{K}}^2 = (\boldsymbol{x})^T\boldsymbol{K}\boldsymbol{x} - 2(\boldsymbol{x})^T\boldsymbol{K}\boldsymbol{x}^* + (\boldsymbol{x}^*)^T\boldsymbol{K}\boldsymbol{x}^* \quad .$$

And if we now define $\boldsymbol{x}^*$ by $\boldsymbol{x}^* = \boldsymbol{K}^{-1}\boldsymbol{b}$ we have

$$f_{\text{opt}}(\boldsymbol{x}^*) = \frac{1}{2}(\boldsymbol{x}^*)^T\boldsymbol{K}\boldsymbol{x}^* - (\boldsymbol{x}^*)^T\boldsymbol{b} = -\frac{1}{2}(\boldsymbol{x}^*)^T\boldsymbol{b} \quad .$$

Hence, by combining both equations we get

$$\|\boldsymbol{x} - \boldsymbol{x}^*\|_{\boldsymbol{K}}^2 = 2\left(f_{\text{opt}}(\boldsymbol{x}) - f_{\text{opt}}(\boldsymbol{x}^*)\right) \quad . \tag{4.26}$$

We therefore retrieve the fact that the minimum of the objective function is reached by the solution of the system $\boldsymbol{K}\boldsymbol{x} = \boldsymbol{b}$.

Besides, evaluating f_{opt} or ∇f_{opt} at any point $\boldsymbol{x} \in \mathbb{R}^{n_{\boldsymbol{K}}}$ only requires to be able to compute the product $\boldsymbol{K}\boldsymbol{x}$, and therefore can be done within a matrix-free approach. Hence a first-order

optimization method, i.e. one that is based on the gradient of the objective function, can be used to solve the problem and therefore get $\boldsymbol{x}^*$ (Nocedal and Wright, 2006). We discard in a first approach any second-order optimization method, which, even though they enjoy faster convergence rates to the solution, require to compute the Hessian matrix of the objective (Nocedal and Wright, 2006), which is here the matrix $\boldsymbol{K}$.

We rather look at algorithms that minimize both computational and storage costs in a matrix-free approach. Ideally, only a few vectors should be stored at any time during the optimization process, and each iteration should require a number as small as possible of products between vectors and the matrix $\boldsymbol{K}$, the optimal number of products being of course 1 (to compute a gradient). First order descent algorithms allow to check both boxes.

More generally, descent algorithms (Nocedal and Wright, 2006) iteratively build a sequence $\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(1)}, \dots$ that converges to $\boldsymbol{x}^*$ and whose terms follow the general recurrence relation:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{d}^{(k)}, \quad k \geq 0 \quad , \tag{4.27}$$

where $\{\boldsymbol{d}^{(k)}\}_{k\geq 0}$ is a family of vectors called descent directions and generally computed using their own recurrence relation, which involves gradient computations and $\{\alpha_k\}_{k\geq 0}$ is a family of (positive) parameters called step sizes.

4.3.3 Steepest gradient descent algorithm

The simplest example of descent algorithm is the *constant-step gradient descent algorithm* (Nocedal and Wright, 2006), which consists in choosing a constant step size for all updates in Equation (4.27), and taking $\boldsymbol{d}^{(k)} = -\nabla f_{\text{opt}}(\boldsymbol{x}^{(k)})$ which corresponds to the direction of greatest decrease of f_{opt}. Hence, we set a parameter $\alpha \in \mathbb{R}_+$ and build the sequence:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha \nabla f_{\text{opt}}(\boldsymbol{x}^{(k)}), \quad k \geq 0 \quad . \tag{4.28}$$

As one may suspect, a successful convergence of this sequence towards $\boldsymbol{x}^*$ highly depends on the choice of α: taking a value of α that is too large results in the divergence of the algorithm and taking a value of α that is too small results in a very slow convergence (Nocedal and Wright, 2006). To avoid the hassle of setting the right parameters, the next algorithm is preferred.

The *steepest gradient descent algorithm* (Nocedal and Wright, 2006) is derived by choosing at each iteration k of the gradient descent a step size $\alpha_k = \alpha_k^{\text{Steep}}$ that yields the greatest decrease of the objective function f_{opt}. Hence,

$$\alpha_k^{\text{Steep}} = \underset{\alpha\in\mathbb{R}}{\operatorname{argmax}}\, f_{\text{opt}}\left(\boldsymbol{x}^{(k)}\right) - f_{\text{opt}}\left(\boldsymbol{x}^{(k)} - \alpha\nabla f_{\text{opt}}(\boldsymbol{x}^{(k)})\right), \quad k \geq 0 \quad .$$

Given that f_{opt} is quadratic, this problem has a closed-form solution that is obtained by calculating the stationary point of the function $\alpha \mapsto f_{\text{opt}}\left(\boldsymbol{x}^{(k)}\right) - f_{\text{opt}}\left(\boldsymbol{x}^{(k)} - \alpha\nabla f_{\text{opt}}(\boldsymbol{x}^{(k)})\right)$. This gives

$$\alpha_k^{\text{Steep}} = \frac{\nabla f_{\text{opt}}(\boldsymbol{x}^{(k)})^T \nabla f_{\text{opt}}(\boldsymbol{x}^{(k)})}{\nabla f_{\text{opt}}(\boldsymbol{x}^{(k)})^T \boldsymbol{K} \nabla f_{\text{opt}}(\boldsymbol{x}^{(k)})}, \quad k \geq 0 \quad .$$

The steepest gradient algorithm is outlined in Algorithm 4.3. It assumes that only a routine allowing to compute matrix-vector products between $\boldsymbol{K}$ and any vector of size $n_{\boldsymbol{K}}$ is known. Besides, the iterations of the algorithm are carried out until "convergence is reached", which means here that a good enough approximation of the solution was reached. To assess the quality of a given iterate $\boldsymbol{x}^{(k)}$, a stopping criterion is usually set by requiring that the (Euclidean) norm of $\nabla f_{\text{opt}}(\boldsymbol{x}^{(k)})$, which is given by $\|\nabla f_{\text{opt}}(\boldsymbol{x}^{(k)})\| = \|\boldsymbol{b} - \boldsymbol{K}\boldsymbol{x}^{(k)}\|$, is below a predefined threshold (Nocedal and Wright, 2006). Other possible stopping criteria include checking the norm of the difference between successive iterates or successive values taken by f_{opt}.

The performance of the steepest gradient algorithm is determined by how fast or equivalently how many iterations are needed for the k-th approximation $\boldsymbol{x}^{(k)}$ of the solution $\boldsymbol{x}^*$ generated by the algorithm to reach a given approximation error, measured as a distance between $\boldsymbol{x}^{(k)}$ and $\boldsymbol{x}^*$. For the steepest gradient descent algorithm, this convergence rate depends (only) on the initial guess we have for $\boldsymbol{x}^*$ and on the properties of $\boldsymbol{K}$ through a quantity called the condition number of $\boldsymbol{K}$ (Nocedal and Wright, 2006; Saad, 2003).

Algorithm 4.3: Steepest gradient algorithm.

Input: For a positive definite matrix $\boldsymbol{K} \in \mathscr{M}_{n_{\boldsymbol{K}}}(\mathbb{R})$, a routine $\texttt{prod}_{\boldsymbol{K}}(\boldsymbol{v})$ that returns for any $\boldsymbol{v} \in \mathbb{R}^{n_{\boldsymbol{K}}}$ the vector $\boldsymbol{K}\boldsymbol{v}$. A vector $\boldsymbol{b} \in \mathbb{R}^{n_{\boldsymbol{K}}}$. An initial guess $\boldsymbol{x}^{(0)}$

Output: An approximation of $\boldsymbol{x}^* = \boldsymbol{K}^{-1}\boldsymbol{b}$.

$k = 0$;
$\boldsymbol{d}^{(0)} = -\nabla f_{\text{opt}}(\boldsymbol{x}^{(0)}) = \boldsymbol{b} - \boldsymbol{K}\boldsymbol{x}^{(0)}$;
while *Convergence is not reached* **do**
  $\alpha_k = \frac{(\boldsymbol{d}^{(k)})^T \boldsymbol{d}^{(k)}}{(\boldsymbol{d}^{(k)})^T \boldsymbol{K} \boldsymbol{d}^{(k)}}$;
  $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{d}^{(k)}$;
  $\boldsymbol{d}^{(k+1)} = \boldsymbol{d}^{(k)} - \alpha_k \cdot \texttt{prod}_{\boldsymbol{K}}(\boldsymbol{d}^{(k)})$;
  $k \leftarrow k + 1$;
Return $\boldsymbol{x}^{(k)}$.

Let us denote by $\|\cdot\|_2$ either the Euclidean norm of a vector or the matrix norm subordinate to the Euclidean norm as defined for matrices of $\mathscr{M}_n(\mathbb{R})$ by

$$\|\boldsymbol{A}\|_2 := \sup_{\boldsymbol{x}\neq\boldsymbol{0}} \frac{\|\boldsymbol{A}\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2} = \sup_{\boldsymbol{x}\neq\boldsymbol{0}} \sqrt{\mathcal{R}(\boldsymbol{A}^T\boldsymbol{A}, \boldsymbol{x})} = \sqrt{\lambda_{\max}(\boldsymbol{A}^T\boldsymbol{A})}, \quad \boldsymbol{A} \in \mathscr{M}_n(\mathbb{R}) \quad ,$$

where $\lambda_{\max}(.)$ denotes the largest eigenvalue of a matrix and $\mathcal{R}(\boldsymbol{M}, \boldsymbol{v})$ denotes the Rayleigh quotient of a Hermitian matrix $\boldsymbol{M}$ and a vector $\boldsymbol{v}$ (cf. Appendix A.2.1).

The condition number $\kappa(\boldsymbol{A})$ of an invertible matrix $\boldsymbol{A}$ is then defined as:

$$\kappa(\boldsymbol{A}) = \|\boldsymbol{A}\|_2 \|\boldsymbol{A}^{-1}\|_2 \quad .$$

Note that in the particular case of a (symmetric) positive definite matrix $\boldsymbol{K}$, its condition number can expressed as

$$\kappa(\boldsymbol{K}) = \frac{\lambda_{\max}(\boldsymbol{K})}{\lambda_{\min}(\boldsymbol{K})} \quad ,$$

where $\lambda_{\max}(\boldsymbol{K})$ (resp. $\lambda_{\min}(\boldsymbol{K})$) denotes the largest (resp. lowest) eigenvalue of $\boldsymbol{K}$.

Proposition 4.3.1. *The sequence $\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(1)}, \ldots$ generated by applying the steepest gradient algorithm to the minimization problem of Equation* (4.25) *satisfies*

$$\forall k \geq 0, \quad \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\|_{\boldsymbol{K}} \leq \left(\frac{\kappa(\boldsymbol{K}) - 1}{\kappa(\boldsymbol{K}) + 1}\right)^k \|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\|_{\boldsymbol{K}} \quad ,$$

where $\kappa(\boldsymbol{K})$ is the condition number of $\boldsymbol{K}$ and $\|\cdot\|_{\boldsymbol{K}}$ is the norm defined for any $\boldsymbol{x} \in \mathbb{R}^{n_{\boldsymbol{K}}}$ by $\|\boldsymbol{x}\|_{\boldsymbol{K}} = \sqrt{\boldsymbol{x}^T\boldsymbol{K}\boldsymbol{x}}$.
In particular, $\forall \epsilon > 0$,

$$k \geq \frac{1}{\log\left(\frac{\kappa(\boldsymbol{K})-1}{\kappa(\boldsymbol{K})+1}\right)} \log\left(\frac{\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\|_{\boldsymbol{K}}}{\epsilon}\right) \Rightarrow \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\|_{\boldsymbol{K}} \leq \epsilon \quad .$$

Proof. See (Sun and Yuan, 2006, Theorem 3.1.5). □

A similar result can be deduced about the convergence towards the global minimum of the objective function f_{opt} of the sequence $\{f_{\text{opt}}(\boldsymbol{x}^{(k)})\}_{k\geq 0}$.

Corollary 4.3.2. *The sequence $\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(1)}, \ldots$ generated by applying the steepest gradient algorithm to the minimization problem of Equation* (4.25) *satisfies:*

$$\forall k \geq 0, \quad \left(f_{\text{opt}}(\boldsymbol{x}^{(k)}) - f_{\text{opt}}(\boldsymbol{x}^*)\right) \leq \left(\frac{\kappa(\boldsymbol{K}) - 1}{\kappa(\boldsymbol{K}) + 1}\right)^{2k} \left(f_{\text{opt}}(\boldsymbol{x}^{(0)}) - f_{\text{opt}}(\boldsymbol{x}^*)\right) \quad ,$$

where $\kappa(\boldsymbol{K})$ is the condition number of $\boldsymbol{K}$.

Proof. This result is a direct consequence of Proposition 4.3.1 and Equation (4.26). □

Hence, the convergence rate of the steepest gradient algorithm is greatly determined by the condition number of the matrix $\boldsymbol{K}$. For ill-conditioned problems, which correspond to the case where $\kappa(\boldsymbol{K})$ is large, convergence may be very slow. In fact, an disproportionate number of iterations may be needed for the sequence $\{\boldsymbol{x}^{(k)}\}_{k\geq 0}$ to reach the minimum $\boldsymbol{x}^*$ (Nocedal and Wright, 2006). However, this flaw is not shared by the algorithm that will be introduced in the next subsection, which has the desirable property to converge in a finite number of iterations.

4.3.4 Conjugate gradient algorithm

The conjugate gradient algorithm (Nocedal and Wright, 2006) is an iterative method designed to solve linear systems of the form of Equation (4.24) where $\boldsymbol{K} \in \mathscr{M}_{n_{\boldsymbol{K}}}(\mathbb{R})$ is indeed a symmetric positive definite matrix. It builds a sequence $\{\boldsymbol{x}^{(k)}\}_{0\leq k\leq n_{\boldsymbol{K}}}$ of approximations of the solution using the following principle.

Let $\boldsymbol{x}^{(0)}$ be an initial guess for $\boldsymbol{x}^*$. Recall from Section 3.3.1 that

$$\boldsymbol{x}^* - \boldsymbol{x}^{(0)} = \boldsymbol{K}^{-1}\boldsymbol{r}^{(0)} \quad ,$$

where $\forall k \geq 0$, $\boldsymbol{r}^{(k)}$ denotes the vector defined by $\boldsymbol{r}^{(k)} = \boldsymbol{b} - \boldsymbol{K}\boldsymbol{x}^{(k)}$, called k-th residual vector. Hence, following Proposition 3.3.1, $\boldsymbol{x}^* - \boldsymbol{x}^{(0)}$ lies in the Krylov subspace of dimension $n_{\boldsymbol{K}}$ generated by $\boldsymbol{K}$ and $\boldsymbol{r}^{(0)}$, and denoted $\mathcal{K}_{n_{\boldsymbol{K}}}(\boldsymbol{K}, \boldsymbol{r}^{(0)})$ (cf. Section 3.3.1).

The conjugate gradient algorithm generates a sequence $\{\boldsymbol{x}^{(k)}\}_{k\geq 0}$ such that $\forall k \geq 0$, $\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(0)}$ is the $\boldsymbol{K}$-orthogonal projection of $\boldsymbol{x}^* - \boldsymbol{x}^{(0)}$ onto the subspace $\mathcal{K}_k(\boldsymbol{K}, \boldsymbol{r}^{(0)})$ of dimension k (Del Corso et al., 2015). Namely,

$$\boldsymbol{x}^{(k)} = \boldsymbol{x}^{(0)} + \underset{\boldsymbol{y}\in\mathcal{K}_k(\boldsymbol{K},\boldsymbol{r}^{(0)})}{\operatorname{argmin}} \|\boldsymbol{x}^* - \boldsymbol{x}^{(0)} - \boldsymbol{y}\|_{\boldsymbol{K}}, \quad k \geq 0 \quad , \tag{4.29}$$

where $\|.\|_{\boldsymbol{K}}$ is the norm defined for any $\boldsymbol{x} \in \mathbb{R}^{n_{\boldsymbol{K}}}$ by $\|\boldsymbol{x}\|_{\boldsymbol{K}} = \sqrt{\boldsymbol{x}^T\boldsymbol{K}\boldsymbol{x}}$. In particular for $k = n_{\boldsymbol{K}}$, given that $\boldsymbol{x}^* - \boldsymbol{x}^{(0)} \in \mathcal{K}_k(\boldsymbol{K}, \boldsymbol{r}^{(0)})$, the minimum in Equation (4.29) is reached for $\boldsymbol{y} = \boldsymbol{x}^* - \boldsymbol{x}^{(0)}$, and therefore

$$\boldsymbol{x}^{(n_{\boldsymbol{K}})} = \boldsymbol{x}^{(0)} + (\boldsymbol{x}^* - \boldsymbol{x}^{(0)}) = \boldsymbol{x}^* \quad .$$

Hence the conjugate gradient reaches the actual solution in (at most) $n_{\boldsymbol{K}}$ iterations.

Remark 4.3.2. Note that, using Equation (4.26), Equation (4.29) can be written as:

$$\boldsymbol{x}^{(k)} = \underset{\boldsymbol{x}\in\boldsymbol{x}^{(0)}+\mathcal{K}_k(\boldsymbol{K},\boldsymbol{r}^{(0)})}{\operatorname{argmin}} \|\boldsymbol{x}^* - \boldsymbol{x}\|_{\boldsymbol{K}} = \underset{\boldsymbol{x}\in\boldsymbol{x}^{(0)}+\mathcal{K}_k(\boldsymbol{K},\boldsymbol{r}^{(0)})}{\operatorname{argmin}} f_{\text{opt}}(\boldsymbol{x}) \quad .$$

Hence, the conjugate algorithm actually computes at each iteration k the vector in the affine space $\boldsymbol{x}^{(0)} + \mathcal{K}_k(\boldsymbol{K}, \boldsymbol{r}^{(0)})$ that minimizes the objective function f_{opt}.

In particular, the conjugate gradient is a descent algorithm. Indeed, let $(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{n_{\boldsymbol{K}}})$ be a $\boldsymbol{K}$-orthonormal basis of $\mathcal{K}_{n_{\boldsymbol{K}}}(\boldsymbol{K}, \boldsymbol{r}^{(0)})$, i.e. $\forall i \neq j \in [\![1, n_{\boldsymbol{K}}]\!]$, $\|\boldsymbol{v}_i\|_{\boldsymbol{K}} = \|\boldsymbol{v}_j\|_{\boldsymbol{K}} = 1$ and $\boldsymbol{v}_i^T\boldsymbol{K}\boldsymbol{v}_j = 0$. Such a basis can be built using a Gram-Schmidt orthogonalization technique, similarly to the Lanczos algorithm (cf. Algorithm 3.4). Doing so, it ensures that $\forall 1 \leq k \leq n_{\boldsymbol{K}}$, $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k$ is a $\boldsymbol{K}$-orthonormal basis of $\mathcal{K}_k(\boldsymbol{K}, \boldsymbol{r}^{(0)})$. Then, in particular, there exists $c_1, \ldots, c_{n_{\boldsymbol{K}}} \in \mathbb{R}$ such that $\boldsymbol{x}^* - \boldsymbol{x}^{(0)} = \sum_{j=1}^{n_{\boldsymbol{K}}} c_j\boldsymbol{v}_j$ which gives by definition of $\boldsymbol{x}^{(k)}$, $k \geq 0$, $\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(0)} = \sum_{j=1}^{k} c_j\boldsymbol{v}_j$. And therefore,

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + c_{k+1}\boldsymbol{v}_{k+1}, \quad k \geq 0 \quad .$$

Computing iteratively the vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{n_{\boldsymbol{K}}}$ using a Lanczos-like algorithm actually yields recurrence relations that are used to compute the projections defining $\boldsymbol{x}^{(k)}$ (Del Corso et al., 2015). In fact, the conjugate gradient algorithm actually computes projections using the recurrence relation:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_k\boldsymbol{d}^{(k)}, \quad k \geq 0 \quad ,$$

where the descent directions $\boldsymbol{d}^{(k)}$ follow their own recurrence relation:

$$\boldsymbol{d}^{(k+1)} = \boldsymbol{r}^{(k+1)} + \beta_k \boldsymbol{d}^{(k)}, \quad k \geq 0 \quad .$$

In particular, the descent directions are $\boldsymbol{K}$-orthogonal and the residuals are orthogonal (with respect to the Euclidean norm), which allows to derive closed-form expressions of the coefficients α_k, β_k:

$$\alpha_k = \frac{(\boldsymbol{r}^{(k)})^T \boldsymbol{r}^{(k)}}{(\boldsymbol{d}^{(k)})^T \boldsymbol{K} \boldsymbol{d}^{(k)}}, \quad \beta_k = \frac{(\boldsymbol{r}^{(k+1)})^T \boldsymbol{r}^{(k+1)}}{(\boldsymbol{r}^{(k)})^T \boldsymbol{r}^{(k)}} \quad .$$

The conjugate gradient algorithm is outlined in Algorithm 4.4.

Algorithm 4.4: Conjugate gradient algorithm.

Input: For a positive definite matrix $\boldsymbol{K} \in \mathscr{M}_{n_{\boldsymbol{K}}}(\mathbb{R})$, a routine $\texttt{prod}_{\boldsymbol{K}}(\boldsymbol{v})$ that returns for any $\boldsymbol{v} \in \mathbb{R}^{n_{\boldsymbol{K}}}$ the vector $\boldsymbol{K}\boldsymbol{v}$. A vector $\boldsymbol{b} \in \mathbb{R}^{n_{\boldsymbol{K}}}$. An initial guess $\boldsymbol{x}^{(0)}$

Output: An approximation of $\boldsymbol{x}^* = \boldsymbol{K}^{-1}\boldsymbol{b}$.

$k = 0$;
$\boldsymbol{r}^{(0)} = \boldsymbol{b} - \texttt{prod}_{\boldsymbol{K}}(\boldsymbol{x}^{(0)})$; $\boldsymbol{d}^{(0)} = \boldsymbol{r}^{(0)}$;
$\boldsymbol{p}^{(0)} = \texttt{prod}_{\boldsymbol{K}}(\boldsymbol{d}^{(0)})$;
while *Convergence is not reached* **do**
  $\alpha_k = \frac{(\boldsymbol{r}^{(k)})^T \boldsymbol{r}^{(k)}}{(\boldsymbol{d}^{(k)})^T \boldsymbol{p}^{(k)}}$;
  $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{d}^{(k)}$;
  $\boldsymbol{r}^{(k+1)} = \boldsymbol{r}^{(k)} - \alpha_k \boldsymbol{p}^{(k)}$;
  $\beta_k = \frac{(\boldsymbol{r}^{(k+1)})^T \boldsymbol{r}^{(k+1)}}{(\boldsymbol{r}^{(k)})^T \boldsymbol{r}^{(k)}}$;
  $\boldsymbol{d}^{(k+1)} = \boldsymbol{r}^{(k+1)} + \beta_k \boldsymbol{d}^{(k)}$;
  $\boldsymbol{p}^{(k+1)} = \texttt{prod}_{\boldsymbol{K}}(\boldsymbol{d}^{(k+1)})$;
  $k \leftarrow k + 1$;
Return $\boldsymbol{x}^{(k)}$.

As mentioned earlier, the conjugate gradient algorithm reaches the solution of the linear system in a finite number of iterations $n_{\boldsymbol{K}}$. In fact, the algorithm is stopped as soon as the k-th residual is null, as this means that $\boldsymbol{x}^{(k)} = \boldsymbol{x}^*$. In theory, this can happen for $k < n_{\boldsymbol{K}}$. However that, in the worst case scenario, the $k = n_{\boldsymbol{K}}$ which can be very large. Stopping the algorithm beforehand, once the k-th iterate is close enough to the solution, seems once again more adequate. Fortunately, the conjugate gradient algorithm enjoys a better convergence rate than the steepest gradient algorithm.

Proposition 4.3.3. *The sequence $\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(1)}, \ldots$ generated by applying the conjugate gradient algorithm to the minimization problem of Equation* (4.25) *satisfies*

$$\forall k \geq 0, \quad \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\|_{\boldsymbol{K}} \leq \left(\frac{\sqrt{\kappa(\boldsymbol{K})} - 1}{\sqrt{\kappa(\boldsymbol{K})} + 1} \right)^k \|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\|_{\boldsymbol{K}} \quad , \tag{4.30}$$

where $\kappa(\boldsymbol{K})$ is the condition number of $\boldsymbol{K}$.

In particular, $\forall \epsilon > 0$, $\forall k \geq 0$,

$$k \geq \frac{1}{\log\left(\frac{\sqrt{\kappa(\boldsymbol{K})}-1}{\sqrt{\kappa(\boldsymbol{K})}+1} \right)} \log\left(\frac{\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\|_{\boldsymbol{K}}}{\epsilon} \right) \Rightarrow \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\|_{\boldsymbol{K}} \leq \epsilon \quad .$$

Proof. See (Saad, 2003, Theorem 6.29 & Equation 6.128). □

Corollary 4.3.4. *The sequence $\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(1)}, \dots$ generated by applying the conjugate gradient algorithm to the minimization problem of Equation* (4.25) *satisfies:*

$$\forall k \geq 0, \quad \left(f_{\text{opt}}(\boldsymbol{x}^{(k)}) - f_{\text{opt}}(\boldsymbol{x}^*)\right) \leq \left(\frac{\sqrt{\kappa(\boldsymbol{K})} - 1}{\sqrt{\kappa(\boldsymbol{K})} + 1}\right)^{2k} \left(f_{\text{opt}}(\boldsymbol{x}^{(0)}) - f_{\text{opt}}(\boldsymbol{x}^*)\right)$$

Proof. This result is a direct consequence of Proposition 4.3.3 and Equation (4.26). □

Given that by definition, $\kappa(\boldsymbol{K}) \geq 1$, we have $\sqrt{\kappa(\boldsymbol{K})} \leq \kappa(\boldsymbol{K})$ and therefore, the conjugate gradient benefits from a faster convergence rate than the steepest gradient descent (introduced in the previous subsection). However, for some problems, $\sqrt{\kappa(\boldsymbol{K})}$ can still be quite large. In that case, preconditioning methods should be applied on top of the optimization algorithm to speed up the convergence.

4.3.5 Note on preconditioning

The idea behind preconditioning is to replace the ill-conditioned system of Equation (4.24) by another system, with a better condition number, and whose solution can easily be used to compute the solution of the original system (Saad, 2003). In our case, Equation (4.24) is replaced by

$$(\boldsymbol{P}_L \boldsymbol{K} \boldsymbol{P}_R)\, \boldsymbol{u}^* = \boldsymbol{P}_L \boldsymbol{b} \quad \text{and} \quad \boldsymbol{x}^* = \boldsymbol{P}_R \boldsymbol{u}^* \quad , \tag{4.31}$$

where $\boldsymbol{P}_L \in \mathscr{M}_n(\mathbb{R})$ (resp. $\boldsymbol{P}_R \in \mathscr{M}_n(\mathbb{R})$) is an invertible matrix called left-preconditioning (resp. right-preconditioning) matrix and is chosen so that $\kappa(\boldsymbol{P}_L \boldsymbol{K} \boldsymbol{P}_R) < \kappa(\boldsymbol{K})$ and is as small as possible.

Given the form of Equation (4.31), the algorithms presented in this section can be rewritten to solve this new system without having to actually the matrix $(\boldsymbol{P}_L \boldsymbol{K} \boldsymbol{P}_R)$. Basically, products by the preconditioning matrices are added at each iteration. Hence, $\boldsymbol{P}_L \in \mathscr{M}_n(\mathbb{R})$ and $\boldsymbol{P}_R \in \mathscr{M}_n(\mathbb{R})$ are chosen so that matrix vector products involving them come at a small computational cost, thus ensuring that the gains in terms of number of iterations to convergence are not overshadowed by the fact that each iteration comes at a greater cost.

An optimal choice for these preconditioning matrices would satisfy $\kappa(\boldsymbol{P}_L \boldsymbol{K} \boldsymbol{P}_R) = 1$, which is the lowest value a condition number can have. This corresponds to the case when $\boldsymbol{P}_L \boldsymbol{K} \boldsymbol{P}_R = c\boldsymbol{I}$ for some $c \neq 0$, which gives $\boldsymbol{K}^{-1} = \boldsymbol{P}_R \boldsymbol{P}_L$. Finding preconditioning matrices satisfying this relation is actually equivalent to computing directly the inverse of $\boldsymbol{K}$ which is here out of the question. Instead, the preconditioning matrices are chosen so that $(\boldsymbol{P}_R \boldsymbol{P}_L)^{-1}$ is somewhat close to $\boldsymbol{K}$, which ensures in general that the condition number will be reduced (Saad, 2003).

Classical choices of preconditioning matrices include (Saad, 2003):

- the Jacobi preconditioner, for which $\boldsymbol{P}_R = \boldsymbol{I}_n$ and $\boldsymbol{P}_L$ is taken to be the diagonal matrix whose entries are the inverse of the diagonal entries of $\boldsymbol{K}$.

- the Gauss-Seidel preconditioner, for which $\boldsymbol{P}_R = \boldsymbol{I}_n$ and $\boldsymbol{P}_L$ is taken to be the inverse of the lower triangular part of $\boldsymbol{K}$. Products between $\boldsymbol{P}_L$ and vectors are therefore computed by solving a triangular system.

- incomplete factorization techniques that define $\boldsymbol{P}_R^{-1}$ and $\boldsymbol{P}_L^{-1}$ as incomplete factorizations of $\boldsymbol{K}$, which are cheaply computable.

In our particular context however, $\boldsymbol{K}$ is not actually known, and we only have a routine computing its product with vectors. Moreover, as the size of the vectors $n_{\boldsymbol{K}}$ can be quite large, the calls to this routine should be limited at a strict minimum. Hence many classical preconditioners, like those mentioned above, cannot be used to accelerate the convergence of the descent algorithms used to solve Equation (4.24).

4.4 Practical implementation on the unknown-mean case

Let us now assume that we aim at extracting a signal $\boldsymbol{Z}$ with mean $m\boldsymbol{v}$ where $m \in \mathbb{R}$ is unknown (but $\boldsymbol{v} \in \mathbb{R}^n$ is a predefined eigenvector of $\boldsymbol{S}$), still from an observation vector $\boldsymbol{z}_o$ arising from an observation process $\boldsymbol{Z}_o$ defined by Equation (4.16).

4.4.1 Matrix-free formulation of the problem

The best unbiased linear solution $\boldsymbol{z}^*$ of $\boldsymbol{Z}$ given the observation $\boldsymbol{z}_o$ is now given by (cf. Proposition 4.2.3)

$$\boldsymbol{z}^* = \tilde{\boldsymbol{P}}\tilde{\boldsymbol{K}}^{-}\tilde{\boldsymbol{b}} \quad ,$$

where the matrices $\tilde{\boldsymbol{P}} \in \mathscr{M}_{n,n_{\boldsymbol{K}}+1}(\mathbb{R})$, $\tilde{\boldsymbol{K}} \in \mathscr{M}_{n_{\boldsymbol{K}}+1}(\mathbb{R})$ and $\tilde{\boldsymbol{b}} \in \mathscr{M}_{n_{\boldsymbol{K}}+1}(\mathbb{R})$ are defined by

$$\tilde{\boldsymbol{P}} = \left(\begin{array}{c|c} \boldsymbol{P} & \boldsymbol{v} \end{array}\right), \quad \tilde{\boldsymbol{K}} = \left(\begin{array}{c|c} \boldsymbol{K} & \boldsymbol{M}_o\boldsymbol{v} \\ \hline (\boldsymbol{M}_o\boldsymbol{v})^T & 0 \end{array}\right)^{-1}, \quad \tilde{\boldsymbol{b}} = \left(\begin{array}{c} \boldsymbol{b} \\ \hline 0 \end{array}\right) \quad , \tag{4.32}$$

where $\boldsymbol{K}$, $\boldsymbol{P}$ and $\boldsymbol{b}$ are defined by Equation (4.22) and therefore are the same as the ones used for the problem of extraction of a signal with mean $\boldsymbol{0}$.

Once again, $\boldsymbol{z}^*$ is computed in two steps:

1. First, compute the term $\tilde{\boldsymbol{x}}^* = \boldsymbol{K}^{-1}\boldsymbol{b}$ by solving the linear system:

$$\tilde{\boldsymbol{K}}\tilde{\boldsymbol{x}}^* = \tilde{\boldsymbol{b}} \quad . \tag{4.33}$$

2. Return $\boldsymbol{z}^* = \tilde{\boldsymbol{P}}\tilde{\boldsymbol{x}}^*$.

The same conclusions as in the known-mean case still holds here: a matrix-free approach, based on Chebyshev filtering, must be considered to perform both tasks as they involve basically the same matrices as the ones used in the known-mean case. Indeed, the matrix-vector products involving $\tilde{\boldsymbol{P}}$ and $\tilde{\boldsymbol{K}}$ can easily be expressed in function of matrix-vector products involving the matrices $\boldsymbol{K}$ and $\boldsymbol{P}$ in Equation (4.22) as

$$\tilde{\boldsymbol{P}}\begin{pmatrix} \boldsymbol{x} \\ \hline \xi \end{pmatrix} = \boldsymbol{P}\boldsymbol{x} + \xi\boldsymbol{v}, \quad \tilde{\boldsymbol{K}}\begin{pmatrix} \boldsymbol{x} \\ \hline \xi \end{pmatrix} = \begin{pmatrix} \boldsymbol{K}\boldsymbol{x} + \xi\boldsymbol{M}_o\boldsymbol{v} \\ \hline (\boldsymbol{M}_o\boldsymbol{v})^T\boldsymbol{x}, \quad \boldsymbol{x} \in \mathbb{R}^{n_{\boldsymbol{K}}}, \xi \in \mathbb{R} \end{pmatrix} \quad .$$

Even though the unknown-mean case seems quite similar to its known-mean counterpart, there is a major difference that prevents us from using the solving methods: the matrix involved in the linear system to be solved is no longer positive definite. Indeed, note for instance that if $\lambda_{\max}(\boldsymbol{K})$ denotes the largest eigenvalue of $\boldsymbol{K}$, then

$$\begin{pmatrix} \boldsymbol{M}_o\boldsymbol{v} \\ \hline -\lambda_{\max}(\boldsymbol{K}) \end{pmatrix}^T \tilde{\boldsymbol{K}} \begin{pmatrix} \boldsymbol{M}_o\boldsymbol{v} \\ \hline -\lambda_{\max}(\boldsymbol{K}) \end{pmatrix} = \|\boldsymbol{M}_o\boldsymbol{v}\|^2 \left(\frac{(\boldsymbol{M}_o\boldsymbol{v})^T\boldsymbol{K}(\boldsymbol{M}_o\boldsymbol{v})}{\|\boldsymbol{M}_o\boldsymbol{v}\|^2} - 2\lambda_{\max}(\boldsymbol{K}) \right) \quad .$$

And this last quantity is strictly negative given that the Rayleigh quotient appearing in the right side of the equation is upper bounded by $\lambda_{\max}(\boldsymbol{K}) > 0$. Hence $\tilde{\boldsymbol{K}}$ cannot be positive definite. Solving Equation (4.33) using the steepest descent algorithm or the conjugate gradient algorithm should therefore be avoided. The next section introduces an algorithm designed to tackle this new problem.

4.4.2 Conjugate residual algorithm

The conjugate residual algorithm (Saad, 2003) aims at solving a system of the form Equation (4.33) in the case that it is only required that $\tilde{\boldsymbol{K}}$ is symmetric. The idea behind this

algorithm is to get back to the positive definite problem. Indeed, by multiplying Equation (4.33) by $\tilde{\boldsymbol{K}}^T = \tilde{\boldsymbol{K}}$ we get the equivalent linear system:

$$\tilde{\boldsymbol{K}}^T \tilde{\boldsymbol{K}} \tilde{\boldsymbol{x}}^* = \tilde{\boldsymbol{K}}^T \tilde{\boldsymbol{b}} \quad , \tag{4.34}$$

where now, the matrix $\tilde{\boldsymbol{K}}^T \tilde{\boldsymbol{K}} = \tilde{\boldsymbol{K}}^2$ is positive definite. This system can therefore be solved using either one of the solvers introduced in Section 4.3. In particular, it would be sufficient to have a routine that computes the product between $\tilde{\boldsymbol{K}}$ and vectors as the product between $\tilde{\boldsymbol{K}}^T \tilde{\boldsymbol{K}} = \tilde{\boldsymbol{K}}^2$ and a vector can then be done by calling this routine twice. Note however that this approach comes at a computational price: each iteration would now cost twice as much as in the case where the system was positive definite.

Fortunately, the conjugate gradient algorithm can be cleverly rewritten to specifically solve the system of Equation (4.34) while requiring at each iteration only a single product between $\tilde{\boldsymbol{K}}$ and a vector: this approach is outlined in Algorithm 4.5. Computationally, when compared to a classical conjugate gradient algorithm, it comes at the price of storing an additional vector throughout the procedure. Algorithm 4.5 generates a set of $\tilde{\boldsymbol{K}}^T \tilde{\boldsymbol{K}}$-conjugate descent directions and ensures that the residuals are $\tilde{\boldsymbol{K}}$-conjugate (Saad, 2003).

Algorithm 4.5: Conjugate residual algorithm.

Input: For a symmetric matrix $\tilde{\boldsymbol{K}} \in \mathscr{M}_{n_{\tilde{\boldsymbol{K}}}}(\mathbb{R})$, a routine $\texttt{prod}_{\tilde{\boldsymbol{K}}}(\boldsymbol{v})$ that returns for any $\boldsymbol{v} \in \mathbb{R}^{n_{\tilde{\boldsymbol{K}}}}$ the vector $\tilde{\boldsymbol{K}}\boldsymbol{v}$. A vector $\tilde{\boldsymbol{b}} \in \mathbb{R}^{n_{\tilde{\boldsymbol{K}}}}$. An initial guess $\tilde{\boldsymbol{x}}^{(0)}$

Output: An approximation of $\tilde{\boldsymbol{x}}^* = \tilde{\boldsymbol{K}}^{-1}\tilde{\boldsymbol{b}}$.

$k = 0$;
$\boldsymbol{r}^{(0)} = \tilde{\boldsymbol{b}} - \texttt{prod}_{\tilde{\boldsymbol{K}}}(\tilde{\boldsymbol{x}}^{(0)})$; $\quad \boldsymbol{d}^{(0)} = \boldsymbol{r}^{(0)}$;
$\boldsymbol{p}^{(0)} = \texttt{prod}_{\tilde{\boldsymbol{K}}}(\boldsymbol{d}^{(0)})$; $\quad \boldsymbol{q}^{(0)} = \boldsymbol{p}^{(0)} (= \texttt{prod}_{\tilde{\boldsymbol{K}}}(\boldsymbol{r}^{(0)}))$;
while *Convergence is not reached* **do**
  $\alpha_k = \frac{(\boldsymbol{q}^{(k)})^T \boldsymbol{q}^{(k)}}{(\boldsymbol{p}^{(k)})^T \boldsymbol{p}^{(k)}} = \frac{(\boldsymbol{r}^{(k)})^T \boldsymbol{q}^{(k)}}{(\boldsymbol{p}^{(k)})^T \boldsymbol{p}^{(k)}}$;
  $\tilde{\boldsymbol{x}}^{(k+1)} = \tilde{\boldsymbol{x}}^{(k)} + \alpha_k \boldsymbol{d}^{(k)}$;
  $\boldsymbol{r}^{(k+1)} = \boldsymbol{r}^{(k)} - \alpha_k \boldsymbol{p}^{(k)}$;
  $\boldsymbol{q}^{(k+1)} = \texttt{prod}_{\tilde{\boldsymbol{K}}}(\boldsymbol{r}^{(k+1)})$;
  $\beta_k = \frac{(\boldsymbol{q}^{(k+1)})^T \boldsymbol{q}^{(k+1)}}{(\boldsymbol{q}^{(k)})^T \boldsymbol{q}^{(k)}} = \frac{(\boldsymbol{r}^{(k+1)})^T \boldsymbol{q}^{(k+1)}}{(\boldsymbol{r}^{(k)})^T \boldsymbol{q}^{(k)}}$;
  $\boldsymbol{d}^{(k+1)} = \boldsymbol{r}^{(k+1)} + \beta_k \boldsymbol{d}^{(k)}$;
  $\boldsymbol{p}^{(k+1)} = \boldsymbol{q}^{(k+1)} + \beta_k \boldsymbol{p}^{(k)}$;
  $k \leftarrow k + 1$;
Return $\boldsymbol{x}^{\tilde{(k)}}$.

Remark 4.4.1. Using the formalism of Equation (4.25), solving Equation (4.33), or equivalently Equation (4.33), is equivalent to a least-square optimization problem, defined by:

$$\tilde{\boldsymbol{x}}^* = \underset{\tilde{\boldsymbol{x}} \in \mathbb{R}^{n_{\tilde{\boldsymbol{K}}}}}{\operatorname{argmin}} \tilde{f}_{\text{opt}}(\tilde{\boldsymbol{x}}), \quad \text{where} \quad \tilde{f}_{\text{opt}}(\tilde{\boldsymbol{x}}) = \frac{1}{2}\|\tilde{\boldsymbol{K}}\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{b}}\|_2^2 \quad . \tag{4.35}$$

And at each iteration k of the algorithm:

$$\tilde{\boldsymbol{x}}^{(k)} = \underset{\tilde{\boldsymbol{x}} \in \tilde{\boldsymbol{x}}^{(0)} + \mathcal{K}_k(\boldsymbol{K}, \boldsymbol{r}^{(0)})}{\operatorname{argmin}} \tilde{f}_{\text{opt}}(\tilde{\boldsymbol{x}}) = \underset{\tilde{\boldsymbol{x}} \in \tilde{\boldsymbol{x}}^{(0)} + \mathcal{K}_k(\boldsymbol{K}, \boldsymbol{r}^{(0)})}{\operatorname{argmin}} \|\tilde{\boldsymbol{K}}\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{b}}\|_2 \quad .$$

Hence, the conjugate residual algorithm actually computes at each iteration k the vector $\tilde{\boldsymbol{x}}$ of the affine space $\tilde{\boldsymbol{x}}^{(0)} + \mathcal{K}_k(\boldsymbol{K}, \boldsymbol{r}^{(0)})$ that minimizes the norm of the residual vector $\tilde{\boldsymbol{b}} - \tilde{\boldsymbol{K}}\tilde{\boldsymbol{x}}$.

The convergence rate of the conjugate residual algorithm can be directly derived from the convergence rate of the conjugate gradient algorithm.

Proposition 4.4.1. *The sequence $\tilde{\boldsymbol{x}}^{(0)}, \tilde{\boldsymbol{x}}^{(1)}, \dots$ generated by applying the conjugate residual algorithm to the minimization problem of Equation* (4.35) *satisfies*

$$\forall k \geq 0, \quad \|\tilde{\boldsymbol{K}}\tilde{\boldsymbol{x}}^{(k)} - \tilde{\boldsymbol{b}}\|_{\boldsymbol{K}} \leq \left(\frac{\kappa(\tilde{\boldsymbol{K}}) - 1}{\kappa(\tilde{\boldsymbol{K}}) + 1}\right)^k \|\tilde{\boldsymbol{K}}\tilde{\boldsymbol{x}}^{(0)} - \tilde{\boldsymbol{b}}\|_2 \quad ,$$

where $\kappa(\tilde{\boldsymbol{K}})$ is the condition number of $\tilde{\boldsymbol{K}}$.
In particular, $\forall \epsilon > 0$, $\forall k \geq 0$,

$$k \geq \frac{1}{\log\left(\frac{\kappa(\tilde{\boldsymbol{K}})-1}{\kappa(\tilde{\boldsymbol{K}})+1}\right)} \log\left(\frac{\|\tilde{\boldsymbol{K}}\tilde{\boldsymbol{x}}^{(0)} - \tilde{\boldsymbol{b}}\|_{\boldsymbol{K}}}{\epsilon}\right) \Rightarrow \|\tilde{\boldsymbol{K}}\tilde{\boldsymbol{x}}^{(k)} - \tilde{\boldsymbol{b}}\|_2 \leq \epsilon \quad .$$

Proof. Note that $\|\tilde{\boldsymbol{K}}^T\tilde{\boldsymbol{K}}\|_2 = \sqrt{\lambda_{\max}\left((\tilde{\boldsymbol{K}}^T\tilde{\boldsymbol{K}})^2\right)} = \lambda_{\max}(\tilde{\boldsymbol{K}}^T\tilde{\boldsymbol{K}}) = \|\tilde{\boldsymbol{K}}\|_2^2$ and that similarly $\|(\tilde{\boldsymbol{K}}^T\tilde{\boldsymbol{K}})^{-1}\|_2 = \|(\tilde{\boldsymbol{K}}^{-1})(\tilde{\boldsymbol{K}}^{-1})^T\|_2 = \|\tilde{\boldsymbol{K}}^{-1}\|_2^2$ gives:

$$\kappa(\tilde{\boldsymbol{K}}^T\tilde{\boldsymbol{K}}) = \kappa(\tilde{\boldsymbol{K}})^2 \quad .$$

Substituting $\boldsymbol{K}$ in Proposition 4.3.3 by $\tilde{\boldsymbol{K}}^T\tilde{\boldsymbol{K}}$ then gives the result by noticing that $\|\tilde{\boldsymbol{v}}\|_{\tilde{\boldsymbol{K}}^T\tilde{\boldsymbol{K}}} = \|\tilde{\boldsymbol{K}}\tilde{\boldsymbol{v}}\|_2$, $\forall \tilde{\boldsymbol{v}} \in \mathbb{R}^{n_{\tilde{\boldsymbol{K}}}}$. □

4.5 Unified approach through quadratic programming

In this section the extraction problem in the known-mean case and in the unknown-mean case are unified into a single optimization framework called quadratic programming. This opens a lead to eventually use wide array of numerical solvers designed for this type of problems in order to tackle the optimization tasks arising from the computation of the BLUP of a signal.

Quadratic programs (QP) with equality constraints are stated as follows (Nocedal and Wright, 2006; Sun and Yuan, 2006). Let $\boldsymbol{Q}$ be a symmetric matrix of size N, $\boldsymbol{d} \in \mathbb{R}^N$ and $\boldsymbol{E} \in \mathscr{M}_{M,N}(\mathbb{R})$, $\boldsymbol{e} \in \mathbb{R}^M$ for some $M \geq 0$. We aim at finding $\boldsymbol{x}^* \in \mathbb{R}^N$ satisfying:

$$\begin{aligned} \boldsymbol{x}^* = &\underset{\boldsymbol{x} \in R^N}{\operatorname{argmin}} f_{\text{opt}}(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} - \boldsymbol{x}^T\boldsymbol{d} \\ &\text{subject to } \boldsymbol{E}\boldsymbol{x} = \boldsymbol{e} \end{aligned} \quad . \tag{4.36}$$

The equation $\boldsymbol{E}\boldsymbol{x} = \boldsymbol{e}$ imposes a set of M linear equations, called equality constraints, that must be satisfied by the solution $\boldsymbol{x}^*$ of the problem. In particular, if $M = 0$, no constraints are imposed while searching for a minimum of f_{opt} (i.e. $\boldsymbol{E}$ and $\boldsymbol{e}$ are not defined) and Problem 4.36 is called unconstrained QP problem. If the matrix $\boldsymbol{Q}$ is positive semidefinite (resp. definite), Problem 4.36 is called a convex QP (resp. strictly convex QP) as the function f_{opt} to minimize is convex (resp. strictly convex).

Clearly, as stated in Section 4.3.2, the solution $\boldsymbol{x}^*$ of the linear system that arises from the extraction of a known-mean signal is the solution of an unconstrained strictly convex QP defined by the matrix $\boldsymbol{Q} = \boldsymbol{K}$ and the vector $\boldsymbol{d} = \boldsymbol{b}$.

In the case where the signal to be extracted is of unknown mean, the following proposition shows that the linear system can also be seen as a strictly convex QP, but now with an equality constraint.

Proposition 4.5.1. *Let $\tilde{\boldsymbol{x}}^*$ be the solution of the linear system of Equation* (4.33)*, where the matrix $\tilde{\boldsymbol{K}}$ and the vector $\tilde{\boldsymbol{b}}$ are defined in Equation* (4.32)*.*
Then $\tilde{\boldsymbol{x}}^*$ can be decomposed as $\tilde{\boldsymbol{x}}^* = (\quad (\boldsymbol{x}^*)^T \quad |\mu)^T$ where:

$$\begin{aligned} \boldsymbol{x}^* = &\underset{\boldsymbol{x} \in R^{n_{\boldsymbol{K}}}}{\operatorname{argmin}} f_{\text{opt}}(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{K}\boldsymbol{x} - \boldsymbol{x}^T\boldsymbol{b} \\ &\text{subject to } (\boldsymbol{M}_o\boldsymbol{v})^T\boldsymbol{x} = 0 \end{aligned} \quad , \tag{4.37}$$

and

$$\mu = \frac{(\boldsymbol{M}_o\boldsymbol{v})^T(\boldsymbol{b} - \boldsymbol{K}\boldsymbol{x}^*)}{\|\boldsymbol{M}_o\boldsymbol{v}\|_2^2} \quad , \tag{4.38}$$

where $\boldsymbol{K}$ *and* $\boldsymbol{b}$ *are defined in Equation* (4.22) *(with* $m = 0$*).*

Proof. Let $N = n_{\boldsymbol{K}}$. Let $\tilde{\boldsymbol{x}}^*$ be decomposed as $\tilde{\boldsymbol{x}}^* = (\quad \widehat{\boldsymbol{x}}^T \quad , \mu)^T$ for some $\widehat{\boldsymbol{x}} \in \mathbb{R}^N$ and $\mu \in \mathbb{R}$. Let us show that $\widehat{\boldsymbol{x}} = \boldsymbol{x}^*$ and that μ satisfies Equation (4.38).

Note that the equation $\tilde{\boldsymbol{K}}\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{b}}$ implies that $\widehat{\boldsymbol{x}}$ and μ must satisfy

$$\boldsymbol{K}\widehat{\boldsymbol{x}} + \mu\boldsymbol{M}_o\boldsymbol{v} = \boldsymbol{b} \quad \text{and} \quad (\boldsymbol{M}_o\boldsymbol{v})^T\widehat{\boldsymbol{x}} = 0 \quad . \tag{4.39}$$

In particular, by denoting $\mathscr{L}$ the function defined on $\mathbb{R}^N \times \mathbb{R}$ by

$$\mathscr{L}(\boldsymbol{y}, \xi) = \frac{1}{2}\boldsymbol{y}^T\boldsymbol{K}\boldsymbol{y} - \boldsymbol{y}^T\boldsymbol{b} + \xi(\boldsymbol{M}_o\boldsymbol{v})^T\boldsymbol{y}, \quad \boldsymbol{y} \in \mathbb{R}^N, \xi \in \mathbb{R} \quad ,$$

we get from Equation (4.39) that $\nabla h(\widehat{\boldsymbol{x}}, \mu) = \boldsymbol{0}$ and therefore (x, μ) is a stationary point of $\mathscr{L}$. Noticing now that $\mathscr{L}$ is actually the Lagrangian function of the constrained minimization problem of Equation (4.37), for which μ plays the role of a Lagrange multiplier, we get that $\widehat{\boldsymbol{x}} = \boldsymbol{x}^*$.

The expression of μ with respect to $\widehat{\boldsymbol{x}} = \boldsymbol{x}^*$ follows from Equation (4.39) implying that $(\boldsymbol{M}_o\boldsymbol{v})^T(\boldsymbol{K}\boldsymbol{x}^* + \mu\boldsymbol{M}_o\boldsymbol{v}) = (\boldsymbol{M}_o\boldsymbol{v})^T\boldsymbol{b}$, which gives the result. □

Consequently, solving the linear system arising from the extraction of a signal with unknown mean can effectively be replaced by solving a strictly convex QP, defined by Equation (4.37) with a single equality constraint. This QP is actually the same QP as the one arising in the known mean case, but with an equality constraint.

Circling back to our matrix-free requirement, note that, for either one of the QPs presented in this section, a routine that evaluates the objective function f_{opt} or its gradient ∇f_{opt} can easily be derived from a routine $\texttt{prod}_{\boldsymbol{K}}$ that computes the product by $\boldsymbol{K}$ and would require a single call to $\texttt{prod}_{\boldsymbol{K}}$. Solving these QPs can actually be done by calling any optimization solver designed for quadratic (or more generally non-linear) problems that takes as an input routines to evaluate the objective function, its gradient and the constraint. This is the case for most of the implementation of these methods (Nocedal and Wright, 2006; Saad, 2003). The only constraint that we should keep in mind is to restrict the number of evaluations of the objective function and its gradient that the solver performs at each iteration.

Implementations of such solvers are available in the R packages `nloptr` (Ypma, 2018) and `mize` (Melville, 2019). Studying the characteristics and performances of the myriad of non-linear solvers that exist today exceeds the scope of this work. However, it represents an actual lead to find a solver that would perform better than the descent algorithms that we currently use.

Conclusion

In this chapter, the problem of predicting or extracting a SGS from its noisy observation was tackled. In particular, the noises considered were either composed of uncorrelated elements affecting each observation, or were a sum of linear transformations of independent stationary signals. The predictors presented were directly inspired from the kriging predictors common in Geostatistics, and are the best linear unbiased predictors.

We proposed algorithms to compute these predictors in a matrix-free approach while once again relying on the Chebyshev filtering algorithm. These algorithms all come down to solving an optimization problem, and the associated solving methods were presented. Finally, the prediction problems of this chapter were formulated as quadratic programming problems, thus expanding the possible means of solving the associated optimization problems.

5

Inference of stochastic graph signals

Contents

Résumé

Nous nous intéressons maintenant au même problème d'estimation que dans le chapitre précédent, mais sans supposer cette fois que la covariance du signal est connue. Il s'agit donc d'inférer les propriétés statistiques d'un signal partiellement observé et bruité, tout en l'estimant. Nous présentons deux approches basées sur une maximisation de vraisemblance: la première consiste à maximiser directement la vraisemblance en utilisant sa forme analytique, la seconde fait recours à l'algorithme EM ("Expectation-Maximization").

Introduction

Starting from the formalism of Section 4.1, let us assume that a zero-mean stationary SGS $\boldsymbol{Z} \in \mathbb{R}^n$ with respect to a shift operator $\boldsymbol{S}$ and with spectral density f, is observed through a realization $\boldsymbol{z}_o \in \mathbb{R}^d$ of an observation process $\boldsymbol{Z}_o \in \mathbb{R}^d$ defined by

$$\boldsymbol{Z}_o = \boldsymbol{M}_o \boldsymbol{Z} + \tau \boldsymbol{W}_o \quad , \tag{5.1}$$

where $\boldsymbol{M}_o \in \mathscr{M}_{d,n}(\mathbb{R})$ is the observation matrix of the process, $\tau \geq 0$ is the variance parameter and $\boldsymbol{W}_o \in \mathbb{R}^d$ is a vector with d independent standard Gaussian entries.

We assume that the only known quantities of the problem are the observation matrix $\boldsymbol{M}_o$ and of course the observation vector $\boldsymbol{z}_o$. This section aims at providing an algorithm designed to predict conjointly the remaining quantities, namely $\boldsymbol{S}$, f, τ and $\boldsymbol{z}$, where $\boldsymbol{z}$ is the realization of $\boldsymbol{Z}$ that gave rise to $\boldsymbol{z}_o$, i.e

$$\boldsymbol{z}_o = \boldsymbol{M}_o \boldsymbol{z} + \tau \boldsymbol{w}_o \quad ,$$

for some realization $\boldsymbol{w}_o$ of $\boldsymbol{W}_o$.

Chapter 4 provides a framework and algorithms for the case where the only quantity to estimate is $\boldsymbol{z}$. We now add to the unknowns of the problem the elements $\boldsymbol{S}$, f, τ characterizing the Gaussian distribution followed by $\boldsymbol{Z}$. Let us assume that these elements are parametrized by the entries of a vector $\boldsymbol{\theta} \in \mathbb{R}^{N_P}$, where $N_P \geq 1$. This means that the estimators of $\boldsymbol{S}$, f, τ will be chosen from families of matrices, functions and real numbers parametrized by $\boldsymbol{\theta}$. We denote by $\boldsymbol{S}_{\boldsymbol{\theta}}$, $f_{\boldsymbol{\theta}}$, $\tau_{\boldsymbol{\theta}}$ the members of these families associated with the vector of parameters $\boldsymbol{\theta}$.

In this chapter we investigate two solutions to this inference problem, both based on the maximization of the likelihood of the observed data. On one hand, the direct maximization of this likelihood, through its analytical expression, is exposed. Then, an approach based on the maximization of surrogate but more easily computable function is presented. It is based on the Expectation-Maximization algorithm (Dempster et al., 1977). Finally, the particular case where the shift operator is assumed to be known is looked into, as it yields several simplifications that lighten the overall computational and storage costs of the inference process.

5.1 Inference by direct likelihood maximization

5.1.1 Principle of the direct likelihood maximization approach

Our starting point is that following Equation (5.1), $\boldsymbol{Z}_o$ follows a Gaussian distribution with mean $\boldsymbol{0}$ and covariance matrix $\boldsymbol{\Sigma}$ given by

$$\boldsymbol{\Sigma} = \boldsymbol{M}_o f(\boldsymbol{S}) \boldsymbol{M}_o^T + \tau^2 \boldsymbol{I}_d \quad . \tag{5.2}$$

Hence, for a set of parameters $\boldsymbol{\theta} \in \mathbb{R}^{N_P}$, we denote by $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ the covariance matrix that $\boldsymbol{Z}_o$ would have had if its distribution were specified by $\boldsymbol{S}_{\boldsymbol{\theta}}$, $f_{\boldsymbol{\theta}}$, $\tau_{\boldsymbol{\theta}}$ instead of $\boldsymbol{S}$, f, τ:

$$\boldsymbol{\Sigma}_{\boldsymbol{\theta}} = \boldsymbol{M}_o f_{\boldsymbol{\theta}}(\boldsymbol{S}_{\boldsymbol{\theta}}) \boldsymbol{M}_o^T + \tau_{\boldsymbol{\theta}}^2 \boldsymbol{I}_d \quad . \tag{5.3}$$

Then, the log-likelihood $L(\boldsymbol{\theta}\,;\,\boldsymbol{z}_o)$ of $\boldsymbol{\theta}$ given $\boldsymbol{z}_o$, which is defined as the evaluation of the log of the distribution function of $\boldsymbol{Z}_o$ at $\boldsymbol{Z}_o = \boldsymbol{z}_o$, under the assumption that its is defined through $\boldsymbol{S}_{\boldsymbol{\theta}}$, $f_{\boldsymbol{\theta}}$, $\tau_{\boldsymbol{\theta}}$, can be expressed as

$$L(\boldsymbol{\theta}\,;\,\boldsymbol{z}_o) = \log \pi_{\boldsymbol{\theta}}(\boldsymbol{Z}_o = \boldsymbol{z}_o) = -\frac{1}{2}\left(\log|\boldsymbol{\Sigma}_{\boldsymbol{\theta}}| + \boldsymbol{z}_o^T \boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1} \boldsymbol{z}_o + d \log 2\pi\right) \quad . \tag{5.4}$$

A maximum likelihood approach consists in maximizing Equation (5.4) for $\boldsymbol{\theta}$. Finding an analytical expression for the maximum seems unlikely. Hence we have to rely on a generic optimization algorithm. Such algorithms require to be able to at least evaluate $L(\boldsymbol{\theta}\,;\,\boldsymbol{z}_o)$ for any $\boldsymbol{\theta}$, or even better, to compute its gradient and Hessian matrix (Nocedal and Wright, 2006). In the next subsections, we focus on the evaluation of the likelihood function $L(\boldsymbol{\theta}\,;\,\boldsymbol{z}_o)$ for any $\boldsymbol{\theta}$, as it is the base of many optimization algorithms and can then be used to approximate gradients and Hessian matrices through for instance finite difference approaches (Nocedal and Wright, 2006).

5.1.2 Evaluation of the likelihood function: the covariance approach

The sole evaluation of $L(\cdot\,;\,\boldsymbol{z}_o)$ requires to compute the log-determinant of $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ and the quadratic term $\boldsymbol{z}_o^T\boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1}\boldsymbol{z}_o$. This should once again be done in a matrix-free approach, given that building $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ is out of the question (as it involves once again to build a graph filter).

The product between $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ and any vector $\boldsymbol{v} \in \mathbb{R}^d$ is given by $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}\boldsymbol{v} = \boldsymbol{M}_o^T\left(f_{\boldsymbol{\theta}}(\boldsymbol{S}_{\boldsymbol{\theta}})\boldsymbol{M}_o\boldsymbol{v}\right) + \tau_{\boldsymbol{\theta}}^2\boldsymbol{v}$ and is computable in three steps: first the vector $\boldsymbol{v}' = \boldsymbol{M}_o\boldsymbol{v}$ is formed, then the product $\boldsymbol{y} = f_{\boldsymbol{\theta}}(\boldsymbol{S}_{\boldsymbol{\theta}})\boldsymbol{v}'$ is calculated using Chebyshev filtering and finally the vector $\boldsymbol{M}_o^T\boldsymbol{y} + \tau_{\boldsymbol{\theta}}^2\boldsymbol{v}$ is returned. This way, the matrix $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$, and in fact any other matrix except $\boldsymbol{M}_o$ and $\boldsymbol{S}_{\boldsymbol{\theta}}$, need not to actually be formed to compute $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}\boldsymbol{v}$. This is in accordance with the matrix-free framework in which we work.

In order to evaluate the log-determinant in Equation (5.4), the matrix $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$, which is symmetric and positive definite, is seen as shift operator. Notice then that, consequently to Proposition 2.3.2, its log-determinant can be written as

$$\log|\boldsymbol{\Sigma}_{\boldsymbol{\theta}}| = \mathrm{Trace}(\log(\boldsymbol{\Sigma}_{\boldsymbol{\theta}})) \quad ,$$

and corresponds therefore to the trace of the graph filter $\log(\boldsymbol{\Sigma}_{\boldsymbol{\theta}})$. Algorithm 2.8 can therefore yield an estimate of $\log|\boldsymbol{\Sigma}_{\boldsymbol{\theta}}|$ based on the Chebyshev filtering of a predefined number of white signals by the filter $\log(\boldsymbol{\Sigma}_{\boldsymbol{\theta}})$. In particular, only products between $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ and vectors of $\mathbb{R}^d$ are required to calculate this estimate.

In order to use Chebyshev filtering with $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ as shift operator, bounds on its eigenvalues must be known. However in this case, the shift operator is not explicitly formed: only its products with vectors are. The following proposition provides an estimate of these bounds in function of $\boldsymbol{\tau}_{\boldsymbol{\theta}}$, f, and the extremal eigenvalues of $\boldsymbol{M}_o^T\boldsymbol{M}_o$ and $\boldsymbol{S}_{\boldsymbol{\theta}}$, which can be computed with more classical approaches using for instance Theorem 2.2.2.

Proposition 5.1.1. *Let $n, d \geq 1$. Let $f : \mathbb{R}_+ \mapsto \mathbb{R}_+^*$, $\tau > 0$ and let $\boldsymbol{S} \in \mathscr{M}_n(\mathbb{R})$ be symmetric. For an observation matrix $\boldsymbol{M}_o \in \mathscr{M}_{d,n}(\mathbb{R})$, we denote by $\boldsymbol{\Sigma}$ the matrix defined by Equation* (5.2).

Then,

$$\lambda_{\max}(\boldsymbol{\Sigma}) \leq \tau^2 + \lambda_{\max}(\boldsymbol{M}_o^T\boldsymbol{M}_o) \max_{\lambda\in[\lambda_{\min}(\boldsymbol{S}),\lambda_{\max}(\boldsymbol{S})]} f(\lambda)$$

and

$$\lambda_{\min}(\boldsymbol{\Sigma}) \geq \tau^2 + \lambda_{\min}(\boldsymbol{M}_o^T\boldsymbol{M}_o) \min_{\lambda\in[\lambda_{\min}(\boldsymbol{S}),\lambda_{\max}(\boldsymbol{S})]} f(\lambda) \quad ,$$

where $\lambda_{\max}(\cdot)$ (resp. $\lambda_{\min}(\cdot)$) denotes the largest (resp. lowest) eigenvalue of a matrix.

Proof. See Appendix C.3. □

The computation of the quadratic term is then performed in two steps. First, the linear system

$$\boldsymbol{\Sigma}\boldsymbol{x}^* = \boldsymbol{z}_o$$

is solved for $\boldsymbol{x}^*$, and then the quadratic term is given by $\boldsymbol{z}_o^T\boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1}\boldsymbol{z}_o = \boldsymbol{z}_o^T\boldsymbol{x}^*$. Following from the approach outlined for the log-determinant, $\boldsymbol{x}^*$ can be computed using the results of Section 2.3.4 on the graph filter $\mathrm{Id}(\boldsymbol{\Sigma}_{\boldsymbol{\theta}})$, where Id denotes the identity map of $\mathbb{R}$. Hence $\boldsymbol{x}^*$ would be computed by filtering $\boldsymbol{z}_o$ with the graph filter $h(\boldsymbol{\Sigma}_{\boldsymbol{\theta}})$, where of course $h : x \mapsto 1/x$. Once again only products between $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ would be needed.

A second approach to compute $\boldsymbol{x}^*$ consists in noticing that the linear system it satisfies actually corresponds to the linear system in Equation (4.24) which is solved to compute the

kriging estimate of Proposition 4.1.1 using the approach outlined in Section 4.3. The steepest gradient or the conjugate gradient algorithm can therefore be used to solve it in a matrix-free approach, and therefore yield $\boldsymbol{x}^*$.

Algorithm 5.1: Covariance approach to the evaluation of the likehood function.

Input: Parameter vector $\boldsymbol{\theta} \in \mathbb{R}^{N_P}$. A routine $\texttt{prod}_{\boldsymbol{\Sigma}}(\boldsymbol{\theta}, \boldsymbol{v})$ that computes the product $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}\boldsymbol{v}$ for $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ defined in Equation (5.3) and $\boldsymbol{v} \in \mathbb{R}^d$.

Output: An estimate of $L(\boldsymbol{\theta}\,;\,\boldsymbol{z}_o)$ as defined in Equation (5.4).

Compute the bounds on the eigenvalues of $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ that are given by Proposition 5.1.1 ;

Compute $\log|\boldsymbol{\Sigma}_{\boldsymbol{\theta}}|$ using Algorithm 2.8 on the graph filter with shift operator $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ and transfer function $x \mapsto \log(x)$;

Compute $\boldsymbol{x}^* = \boldsymbol{\Sigma}_{\boldsymbol{\theta}}^* \boldsymbol{z}_o$ using:

- Either the steepest gradient or the conjugate gradient algorithms described in Algorithms 4.3 and 4.4.
- Or Chebyshev filtering to compute the product $h(\boldsymbol{\Sigma}_{\boldsymbol{\theta}})\boldsymbol{z}_o$ where $h : x \mapsto 1/x$.

Return $L(\boldsymbol{\theta}\,;\,\boldsymbol{z}_o) = -\frac{1}{2}\left(\log|\boldsymbol{\Sigma}_{\boldsymbol{\theta}}| + \boldsymbol{z}_o^T\boldsymbol{x}^* + d\log 2\pi\right)$.

Algorithm 5.1 sums up the method used to evaluate the likelihood of a particular parameter vector $\boldsymbol{\theta}$. Plugging this function into an optimization algorithm that only requires evaluation of the objective function will then yield the parameters $\boldsymbol{\theta}^*$ that actually maximizes $L(\cdot\,;\,\boldsymbol{z}_o)$. Examples of such algorithms include the Nelder–Meade algorithm which only relies on evaluations of the objective function, or gradient descent algorithms for which the gradients are numerically approximated from function evaluations (Press et al., 2007).

It is hard to predict in advance the number of evaluations of the likelihood function that will be necessary to find the maximum. In this regard, its cost of evaluation should be reduced at a minimum. However, in Algorithm 5.1, each evaluation requires numerous products between the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ and vectors in order to compute both the determinant and the solution of the linear system. Each one of these products may be quite costly as it involves a Chebyshev filtering step.

5.1.3 Evaluation of the likelihood function: the precision approach

In an attempt to save some computing time, an idea consists in working directly with the precision matrix $\boldsymbol{Q}_{\boldsymbol{\theta}} = \boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1}$ instead of the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$. Indeed, the likelihood $L(\boldsymbol{\theta}\,;\,\boldsymbol{z}_o)$ to maximize can be expressed in function of $\boldsymbol{Q}_{\boldsymbol{\theta}}$ as

$$L(\boldsymbol{\theta}\,;\,\boldsymbol{z}_o) = -\frac{1}{2}\left(-\log|\boldsymbol{Q}_{\boldsymbol{\theta}}| + \boldsymbol{z}_o^T\boldsymbol{Q}_{\boldsymbol{\theta}}\boldsymbol{z}_o + d\log 2\pi\right) \quad . \tag{5.5}$$

So following, the same reasoning that led to Algorithm 5.1, evaluating $L(\boldsymbol{\theta}\,;\,\boldsymbol{z}_o)$ could be done while relying only on products between $\boldsymbol{Q}_{\boldsymbol{\theta}}$ and vectors. To do so, an expression of $\boldsymbol{Q}_{\boldsymbol{\theta}}$ as a function of the parametrized objects $f_{\boldsymbol{\theta}}$, $\boldsymbol{S}_{\boldsymbol{\theta}}$ and $\tau_{\boldsymbol{\theta}}$ must be derived. Ideally, this expression should be different than simply taking $\boldsymbol{Q}_{\boldsymbol{\theta}} = \boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1} = \left(\boldsymbol{M}_o f_{\boldsymbol{\theta}}(\boldsymbol{S}_{\boldsymbol{\theta}})\boldsymbol{M}_o^T + \tau_{\boldsymbol{\theta}}^2\boldsymbol{I}_d\right)^{-1}$ as otherwise, we retrieve Algorithm 5.1.

Following from the proof of Proposition 4.1.1, we recall that the joint distribution of the vectors $\boldsymbol{Z}$ and $\boldsymbol{Z}_o$, now under a parameter $\boldsymbol{\theta}$, is actually that of a zero-mean Gaussian vector whose covariance matrix $\widetilde{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}}$ can be expressed with respect to $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ (cf. Equation (C.1)):

$$\widetilde{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}} = \begin{pmatrix} f_{\boldsymbol{\theta}}(\boldsymbol{S}_{\boldsymbol{\theta}}) & f_{\boldsymbol{\theta}}(\boldsymbol{S}_{\boldsymbol{\theta}})\boldsymbol{M}_o^T \\ \boldsymbol{M}_o f_{\boldsymbol{\theta}}(\boldsymbol{S}_{\boldsymbol{\theta}}) & \boldsymbol{\Sigma}_{\boldsymbol{\theta}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{I}_n & \\ \boldsymbol{M}_o & \boldsymbol{I}_d \end{pmatrix} \begin{pmatrix} f_{\boldsymbol{\theta}}(\boldsymbol{S}_{\boldsymbol{\theta}}) & \\ & \tau_{\boldsymbol{\theta}}^2\boldsymbol{I}_d \end{pmatrix} \begin{pmatrix} \boldsymbol{I}_n & \boldsymbol{M}_o^T \\ & \boldsymbol{I}_d \end{pmatrix} . \tag{5.6}$$

The inverse of this matrix, denoted by $\widetilde{\boldsymbol{Q}}_{\boldsymbol{\theta}}$, is then given by

$$\begin{aligned}\widetilde{\boldsymbol{Q}}_{\boldsymbol{\theta}} &= \begin{pmatrix} \boldsymbol{I}_n & -\boldsymbol{M}_o^T \\ & \boldsymbol{I}_d \end{pmatrix} \begin{pmatrix} (1/f_{\boldsymbol{\theta}})(\boldsymbol{S}_{\boldsymbol{\theta}}) & \\ & \tau_{\boldsymbol{\theta}}^{-2}\boldsymbol{I}_d \end{pmatrix} \begin{pmatrix} \boldsymbol{I}_n & \\ \boldsymbol{M}_o & \boldsymbol{I}_d \end{pmatrix} \\ &= \begin{pmatrix} (1/f_{\boldsymbol{\theta}})(\boldsymbol{S}_{\boldsymbol{\theta}}) + \tau_{\boldsymbol{\theta}}^{-2}\boldsymbol{M}_o^T\boldsymbol{M}_o & -\tau_{\boldsymbol{\theta}}^{-2}\boldsymbol{M}_o^T \\ -\tau_{\boldsymbol{\theta}}^{-2}\boldsymbol{M}_o & \tau_{\boldsymbol{\theta}}^{-2}\boldsymbol{I}_d \end{pmatrix} .\end{aligned} \tag{5.7}$$

Hence, the inverse of $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$, which is $\boldsymbol{Q}_{\boldsymbol{\theta}}$, can be expressed using a Schur complement of $\widetilde{\boldsymbol{Q}}_{\boldsymbol{\theta}}$ (cf. Equation (A.6)) as

$$\boldsymbol{Q}_{\boldsymbol{\theta}} = \boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1} = \tau_{\boldsymbol{\theta}}^{-2}\left(\boldsymbol{I}_d - \tau_{\boldsymbol{\theta}}^{-2}\boldsymbol{M}_o\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}^{-1}\boldsymbol{M}_o^T\right) \quad ,$$

where $\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}$ is the matrix defined by

$$\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}} := (1/f_{\boldsymbol{\theta}})(\boldsymbol{S}_{\boldsymbol{\theta}}) + \tau_{\boldsymbol{\theta}}^{-2}\boldsymbol{M}_o^T\boldsymbol{M}_o \quad . \tag{5.8}$$

The quadratic term $\boldsymbol{x}^T\boldsymbol{Q}_{\boldsymbol{\theta}}\boldsymbol{x}$ in Equation (5.5) therefore involves the resolution of a linear system in $\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}$ and can therefore be computed using the same approach as the one derived for the computation of the quadratic term in Algorithm 5.1.

As for the log-determinant of $\boldsymbol{Q}_{\boldsymbol{\theta}}$ that appears in Equation (5.5), given that $|\widetilde{\boldsymbol{Q}}_{\boldsymbol{\theta}}| = |(1/f_{\boldsymbol{\theta}})(\boldsymbol{S}_{\boldsymbol{\theta}})| \cdot |\tau_{\boldsymbol{\theta}}^{-2}\boldsymbol{I}_d|$, it satisfies (cf. Equation (A.7))

$$\log|\boldsymbol{Q}_{\boldsymbol{\theta}}| = -2d\log\tau_{\boldsymbol{\theta}} + \log|(1/f_{\boldsymbol{\theta}})(\boldsymbol{S}_{\boldsymbol{\theta}})| - \log|\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}| \quad .$$

In this expression, the term $\log|(1/f_{\boldsymbol{\theta}})(\boldsymbol{S}_{\boldsymbol{\theta}})|$ is the log-determinant of a graph filter defined through the shift operator $\boldsymbol{S}_{\boldsymbol{\theta}}$ and can therefore be computed using the results of Section 2.3.3 using a method requiring only products between $\boldsymbol{S}_{\boldsymbol{\theta}}$ and vectors.

The term $\log|\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}|$ can be computed using the same approach as the one outlined for the computation of $\log|\boldsymbol{\Sigma}_{\boldsymbol{\theta}}|$ in Algorithm 5.1, thus requiring products between the matrix $\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}$ and vectors. The next proposition gives an estimate of the eigenvalue bounds of $\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}$ needed to use this approach, which is summarized in Algorithm 5.2.

Proposition 5.1.2. *Let $n, d \geq 1$. Let $f : \mathbb{R}_+ \mapsto \mathbb{R}_+^*$, $\tau > 0$ and let $\boldsymbol{S} \in \mathscr{M}_n(\mathbb{R})$ be symmetric. For an observation matrix $\boldsymbol{M}_o \in \mathscr{M}_{d,n}(\mathbb{R})$, we denote by $\widehat{\boldsymbol{Q}}$ the matrix defined by*

$$\widehat{\boldsymbol{Q}} := (1/f)(\boldsymbol{S}) + \tau^{-2}\boldsymbol{M}_o^T\boldsymbol{M}_o \quad .$$

Then,

$$\lambda_{\max}(\widehat{\boldsymbol{Q}}) \leq \tau^{-2}\lambda_{\max}(\boldsymbol{M}_o^T\boldsymbol{M}_o) + \max_{\lambda\in[\lambda_{\min}(\boldsymbol{S}),\lambda_{\max}(\boldsymbol{S})]} \frac{1}{f(\lambda)}$$

and

$$\lambda_{\min}(\widehat{\boldsymbol{Q}}) \geq \tau^{-2}\lambda_{\min}(\boldsymbol{M}_o^T\boldsymbol{M}_o) + \min_{\lambda\in[\lambda_{\min}(\boldsymbol{S}),\lambda_{\max}(\boldsymbol{S})]} \frac{1}{f(\lambda)} \quad ,$$

where $\lambda_{\max}(\cdot)$ (resp. $\lambda_{\min}(\cdot)$) denotes the largest (resp. lowest) eigenvalue of a matrix.

Proof. The proof of Proposition 5.1.1 can be directly adapted to prove this result. □

Algorithms 5.1 and 5.2 both propose a similar approach to the evaluation of the likelihood function. They both rely on the computation of the log-determinant and on the resolution of a linear system involving a matrix (either $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ or $\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}$) that is not sparse a priori and whose products with a vector require to perform Chebyshev filtering operations. In one case, the approximated function is $f_{\boldsymbol{\theta}}$ (for Algorithm 5.1) and in the other case it is $(1/f_{\boldsymbol{\theta}})$ (for Algorithm 5.2). Hence, the choice between both algorithms should be made based on which one of $f_{\boldsymbol{\theta}}$ or $(1/f_{\boldsymbol{\theta}})$ requires less polynomials to be approximated by a Chebyshev series. This will ensure that we minimize the cost of evaluation of the likelihood function and therefore the cost of the overall minimization process.

Algorithm 5.2: Precision approach to the evaluation of the likehood function.

Input: Parameter vector $\boldsymbol{\theta} \in \mathbb{R}^{N_P}$. A routine $\texttt{prod}_{\widehat{\boldsymbol{Q}}}(\boldsymbol{\theta}, \boldsymbol{v})$ that computes the product $\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}\boldsymbol{v}$ for $\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}$ defined in Equation (5.8) and $\boldsymbol{v} \in \mathbb{R}^d$.

Output: An estimate of $L(\boldsymbol{\theta}\,;\,\boldsymbol{z}_o)$ as defined in Equation (5.5).

Compute $\log|(1/f_{\boldsymbol{\theta}})(\boldsymbol{S}_{\boldsymbol{\theta}})|$ using an algorithm from Section 2.3.3;

Compute the bounds on the eigenvalues of $\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}$ that are given by Proposition 5.1.2 ;

Compute $\log|\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}|$ using Algorithm 2.8 on the graph filter with shift operator $\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}$ and transfer function $x \mapsto \log(x)$;

Compute $\boldsymbol{x}^* = \widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}^{-1}\boldsymbol{M}_o^T\boldsymbol{z}_o$ using:

- Either the steepest gradient or the conjugate gradient algorithms described in Algorithms 4.3 and 4.4.
- Or Chebyshev filtering to compute the product $h(\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}})\boldsymbol{M}_o^T\boldsymbol{z}_o$ where $h : x \mapsto 1/x$.

Compute the quantity q corresponding to the quadratic term:

$$q = \tau_{\boldsymbol{\theta}}^{-2}\left(\boldsymbol{z}_o^T\boldsymbol{z}_o - \tau_{\boldsymbol{\theta}}^{-2}(\boldsymbol{M}_o^T\boldsymbol{z}_o)^T\boldsymbol{x}^*\right) \quad .$$

Return $L(\boldsymbol{\theta}\,;\,\boldsymbol{z}_o) = -\frac{1}{2}\Big(2d\log\tau_{\boldsymbol{\theta}} - \log|(1/f_{\boldsymbol{\theta}})(\boldsymbol{S}_{\boldsymbol{\theta}})| + \log|\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}| + q - d\log 2\pi\Big)$.

5.2 Inference using the Expectation-Maximization approach

We now propose an alternative to the direct maximization of the “hard-to-evaluate" likelihood function that is based on the Expectation-Maximization (EM) algorithm (Dempster et al., 1977).

5.2.1 Formulation of the EM algorithm for SGS inference

Given a parameter vector $\boldsymbol{\theta}$, recall that the associated joint distribution of $(\boldsymbol{Z}, \boldsymbol{Z}_o)$ is that of a zero-mean Gaussian vector with covariance matrix given by $\widetilde{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}}$. In particular, the log-likelihood of $\boldsymbol{\theta}$ given now a couple $(\boldsymbol{Z} = \boldsymbol{\zeta}, \boldsymbol{Z}_o = \boldsymbol{z}_o)$ with $\boldsymbol{\xi} \in \mathbb{R}^n$ would therefore be

$$\begin{aligned}\widetilde{L}(\boldsymbol{\theta}\,;\,\boldsymbol{\zeta}, \boldsymbol{z}_o) &= \log\pi_{\boldsymbol{\theta}}(\boldsymbol{Z} = \boldsymbol{\zeta}, \boldsymbol{Z}_o = \boldsymbol{z}_o) \\ &= -\frac{1}{2}\left(\log|\widetilde{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}}| + \begin{pmatrix}\boldsymbol{\zeta}\\ \boldsymbol{z}_o\end{pmatrix}^T \widetilde{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}}^{-1}\begin{pmatrix}\boldsymbol{\zeta}\\ \boldsymbol{z}_o\end{pmatrix} + (n+d)\log 2\pi\right) \quad ,\end{aligned} \tag{5.9}$$

where $\widetilde{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}}$ is given by Equation (5.6). This equation can be rewritten with respect to $\widetilde{\boldsymbol{Q}}_{\boldsymbol{\theta}} = \widetilde{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}}^{-1}$, the precision matrix of $(\boldsymbol{Z}, \boldsymbol{Z}_o)$ under the set of parameters $\boldsymbol{\theta}$:

$$\widetilde{L}(\boldsymbol{\theta}\,;\,\boldsymbol{\zeta}, \boldsymbol{z}_o) = -\frac{1}{2}\left(-\log|\widetilde{\boldsymbol{Q}}_{\boldsymbol{\theta}}| + \begin{pmatrix}\boldsymbol{\zeta}\\ \boldsymbol{z}_o\end{pmatrix}^T \widetilde{\boldsymbol{Q}}_{\boldsymbol{\theta}}\begin{pmatrix}\boldsymbol{\zeta}\\ \boldsymbol{z}_o\end{pmatrix} + (n+d)\log 2\pi\right) \quad , \tag{5.10}$$

where $\widetilde{\boldsymbol{Q}}_{\boldsymbol{\theta}}$ is given by Equation (5.6), and satisfies in particular

$$\log|\widetilde{\boldsymbol{Q}}_{\boldsymbol{\theta}}| = -2d\log\tau_{\boldsymbol{\theta}} + \log|(1/f_{\boldsymbol{\theta}})(\boldsymbol{S}_{\boldsymbol{\theta}})| \quad . \tag{5.11}$$

Hence the likelihood $\widetilde{L}(\boldsymbol{\theta}\,;\,\boldsymbol{\zeta}, \boldsymbol{z}_o)$ defined in Equation (5.10) is way cheaper to compute than its counterpart $L(\boldsymbol{\theta}\,;\,\boldsymbol{z}_o)$ of Equation (5.4). Indeed, computing the log-determinant in $\widetilde{L}(\boldsymbol{\theta}\,;\,\boldsymbol{\zeta}, \boldsymbol{z}_o)$ through Equation (5.11) requires mainly to compute the log-determinant of the graph filter defined through the shift operator $\boldsymbol{S}_{\boldsymbol{\theta}}$. Using Chebyshev filtering to estimate this quantity with the methods presented in Section 2.3.3, it requires only products between $\boldsymbol{S}_{\boldsymbol{\theta}}$, which is

generally sparse[1], and vectors. In comparison, computing the log-determinant in Equation (5.4) (resp. Equation (5.5)) required products between $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ (resp. $\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}$), and therefore $f_{\boldsymbol{\theta}}(\boldsymbol{S}_{\boldsymbol{\theta}})$ (resp. $(1/f_{\boldsymbol{\theta}})(\boldsymbol{S}_{\boldsymbol{\theta}})$), and vectors.

As for the quadratic term in $\widetilde{L}(\boldsymbol{\theta}\,;\,\boldsymbol{\zeta}, \boldsymbol{z}_o)$, it is computed with the cost of basically a Chebyshev filtering operation with $\boldsymbol{S}_{\boldsymbol{\theta}}$. Comparatively, the quadratic term in the expression of $L(\boldsymbol{\theta}\,;\,\boldsymbol{z}_o)$ requires to solve a linear system defined by $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ (or $\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}$). This property is particularly interesting when considering Markovian models. In this setting, conditional independence relations are imposed between the entries of the modeled signal, which results in its precision matrix being sparse (Rue and Held, 2005). In particular, models can easily be retrieved by imposing that $1/f_{\boldsymbol{\theta}}$ is a low-degree polynomial.

The idea of the *EM algorithm* is to replace the maximization of $L(\cdot\,;\,\boldsymbol{z}_o)$ with the maximization of an objective function defined through $L(\cdot\,;\,\boldsymbol{\zeta}, \boldsymbol{z}_o)$ and which is hoped to be easier to compute. To do so, note that the log-likelihood $L(\boldsymbol{\theta}\,;\,\boldsymbol{z}_o) = \log \pi_{\boldsymbol{\theta}}(\boldsymbol{Z}_o = \boldsymbol{z}_o)$ can be expressed as the log of a marginal distribution of the joint distribution of $(\boldsymbol{Z}_o, \boldsymbol{Z})$, and so,

$$L(\boldsymbol{\theta}\,;\,\boldsymbol{z}_o) = \log \int \pi_{\boldsymbol{\theta}}(\boldsymbol{Z} = \boldsymbol{\zeta}, \boldsymbol{Z}_o = \boldsymbol{z}_o) d\boldsymbol{\zeta} = \log \int \exp \widetilde{L}(\boldsymbol{\theta}\,;\,\boldsymbol{\zeta}, \boldsymbol{z}_o) d\boldsymbol{\zeta} \quad .$$

The EM algorithm leverages this expression to maximize $L(\cdot\,;\,\boldsymbol{z}_o)$ through an iterative approach. A sequence $\{\boldsymbol{\theta}^{(k)}\}_{k\geq 0}$ converging to a local maximum of $L(\cdot\,;\,\boldsymbol{z}_o)$ is generated through a recurrence that comprises two steps

- *Expectation step*: Find an expression for the expectation function $E_{\boldsymbol{\theta}^{(k)}}$ defined by

$$E_{\boldsymbol{\theta}^{(k)}} : \boldsymbol{\theta} \mapsto \mathbb{E}\left[\widetilde{L}(\boldsymbol{\theta}\,;\,\boldsymbol{Z}_{\boldsymbol{\theta}^{(k)}}, \boldsymbol{z}_o)\right] \quad \text{where} \quad \boldsymbol{Z}_{\boldsymbol{\theta}^{(k)}} = [\boldsymbol{Z} | \boldsymbol{Z}_o = \boldsymbol{z}_o\,;\,\boldsymbol{\theta}^{(k)}] \quad . \tag{5.12}$$

- *Maximization step*: Maximize the expectation function $E_{\boldsymbol{\theta}^{(k)}}$:

$$\boldsymbol{\theta}^{(k+1)} = \underset{\boldsymbol{\theta}\in\mathbb{R}^{N_P}}{\operatorname{argmax}}\, E_{\boldsymbol{\theta}^{(k)}}(\boldsymbol{\theta}) \quad . \tag{5.13}$$

Basically, to compute the value of the expectation function $E_{\boldsymbol{\theta}^{(k)}}$ at some $\boldsymbol{\theta}$, the observed data $\boldsymbol{Z}_o = \boldsymbol{z}_o$ are completed with a vector $\boldsymbol{Z} = \boldsymbol{Z}_{\boldsymbol{\theta}^{(k)}}$ that is drawn from the conditional distribution of $\boldsymbol{Z}$ given $\boldsymbol{Z}_o = \boldsymbol{z}_o$ and under the current estimate $\boldsymbol{\theta}^{(k)}$ of the maximum. Then, $E_{\boldsymbol{\theta}^{(k)}}(\boldsymbol{\theta})$ is defined as the "average" over all completion vectors $\boldsymbol{Z}_{\boldsymbol{\theta}^{(k)}}$ drawn this way, of the log-likelihood of $\boldsymbol{\theta}$ with respect to the completed pair $(\boldsymbol{Z} = \boldsymbol{Z}_{\boldsymbol{\theta}^{(k)}}, \boldsymbol{Z}_o = \boldsymbol{z}_o)$.

In the next two subsections, we show two ways of performing the Expectation step of the EM algorithm in our particular inference problem.

5.2.2 EM by trace approximation

First; note that Proposition 4.1.1 actually gives the distribution of $\boldsymbol{Z}_{\boldsymbol{\theta}^{(k)}}$:

$$\boldsymbol{Z}_{\boldsymbol{\theta}^{(k)}} = [\boldsymbol{Z} | \boldsymbol{Z}_o = \boldsymbol{z}_o\,;\,\boldsymbol{\theta}^{(k)}] \sim \mathcal{N}\left(\tau^{-2}_{\boldsymbol{\theta}^{(k)}} \widehat{\boldsymbol{Q}}^{-1}_{\boldsymbol{\theta}^{(k)}} \boldsymbol{M}_o^T \boldsymbol{z}_o;\quad \widehat{\boldsymbol{Q}}^{-1}_{\boldsymbol{\theta}^{(k)}}\right) \quad , \tag{5.14}$$

where $\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}^{(k)}}$ is once again the matrix defined in Equation (5.8), but with $\boldsymbol{\theta} = \boldsymbol{\theta}^{(k)}$. We now derive the expression of $E_{\boldsymbol{\theta}^{(k)}}(\boldsymbol{\theta})$ from this observation. First, note that using the linearity of the expectation, we have

$$E_{\boldsymbol{\theta}^{(k)}}(\boldsymbol{\theta}) = -\frac{1}{2}\left(-\log|\widetilde{\boldsymbol{Q}}_{\boldsymbol{\theta}}| + \left(\mathbb{E}\left[\boldsymbol{Z}^T_{\boldsymbol{\theta}^{(k)}} \widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}} \boldsymbol{Z}_{\boldsymbol{\theta}^{(k)}}\right] - \frac{2}{\tau^2_{\boldsymbol{\theta}}} \boldsymbol{z}_o^T \boldsymbol{M}_o \mathbb{E}[\boldsymbol{Z}_{\boldsymbol{\theta}^{(k)}}] + \frac{1}{\tau^2_{\boldsymbol{\theta}}} \boldsymbol{z}_o^T \boldsymbol{z}_o\right)\right) + C \,,$$

where C is a constant. Note then that, following Proposition A.3.5, we have

$$\mathbb{E}[(\boldsymbol{Z}_{\boldsymbol{\theta}^{(k)}})^T \widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}} \boldsymbol{Z}_{\boldsymbol{\theta}^{(k)}}] = \operatorname{Trace}(\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}} \operatorname{Var}[\boldsymbol{Z}_{\boldsymbol{\theta}^{(k)}}]) + \mathbb{E}[\boldsymbol{Z}_{\boldsymbol{\theta}^{(k)}}]^T \widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}} \mathbb{E}[\boldsymbol{Z}_{\boldsymbol{\theta}^{(k)}}] \quad .$$

[1]Recall indeed that $\boldsymbol{S}_{\boldsymbol{\theta}}$ is supposed to be a shift operator, and as such its sparsity pattern is directly linked to the amount of "connections" in the graph it represents. In many real-world applications, and in particular in the ones that will be presented in this work, these graphs are sparsely connected, and therefore yield sparse shift operators.

Injecting this relation in the previous equation then gives,

$$E_{\boldsymbol{\theta}^{(k)}}(\boldsymbol{\theta}) = -\frac{1}{2}\Bigg(-\log|\widetilde{\boldsymbol{Q}}_{\boldsymbol{\theta}}| + \mathrm{Trace}(\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}^{(k)}}^{-1}) + \boldsymbol{\mu}_{\boldsymbol{\theta}^{(k)}}^T\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}\boldsymbol{\mu}_{\boldsymbol{\theta}^{(k)}} - \frac{2}{\tau_{\boldsymbol{\theta}}^2}\boldsymbol{z}_o^T\boldsymbol{M}_o\boldsymbol{\mu}_{\boldsymbol{\theta}^{(k)}} + \frac{1}{\tau_{\boldsymbol{\theta}}^2}\boldsymbol{z}_o^T\boldsymbol{z}_o\Bigg) + C \quad , \tag{5.15}$$

where, following Equation (5.14),

$$\boldsymbol{\mu}_{\boldsymbol{\theta}^{(k)}} := \mathbb{E}[\boldsymbol{Z}_{\boldsymbol{\theta}^{(k)}}] = \tau_{\boldsymbol{\theta}^{(k)}}^{-2}\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}^{(k)}}^{-1}\boldsymbol{M}_o^T\boldsymbol{z}_o \quad .$$

Thus, the steps of the EM algorithm come down to the computation of a sequence of parameters vectors $\boldsymbol{\theta}^{(0)}, \boldsymbol{\theta}^{(1)}, \ldots$ through the recurrence relation of the maximization step, i.e. Equation (5.13), where $E_{\boldsymbol{\theta}^{(k)}}(\boldsymbol{\theta})$ is given by Equation (5.15). In particular, by removing all constant terms and all additive terms that do not depend on $\boldsymbol{\theta}$ in Equation (5.15), and using Equation (5.11), we get the following equivalent formulation of the recurrence relation:

$$\boldsymbol{\theta}^{(k+1)} = \underset{\boldsymbol{\theta}\in\mathbb{R}^{N_P}}{\mathrm{argmin}}\Bigg(\mathrm{Trace}(\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}^{(k)}}^{-1}) + \boldsymbol{\mu}_{\boldsymbol{\theta}^{(k)}}^T\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}\boldsymbol{\mu}_{\boldsymbol{\theta}^{(k)}} + \frac{1}{\tau_{\boldsymbol{\theta}}^2}\boldsymbol{z}_o^T\left(\boldsymbol{z}_o - 2\boldsymbol{M}_o\boldsymbol{\mu}_{\boldsymbol{\theta}^{(k)}}\right) + 2d\log\tau_{\boldsymbol{\theta}} - \log|(1/f_{\boldsymbol{\theta}})(\boldsymbol{S}_{\boldsymbol{\theta}})|\Bigg) \quad , \tag{5.16}$$

where $\boldsymbol{\mu}_{\boldsymbol{\theta}^{(k)}}$ does not depend on $\boldsymbol{\theta}$ and can therefore be computed once and for all prior to the minimization process of Equation (5.16), and so be used at each evaluation of the objective function.

Evaluating the objective function in Equation (5.16) for a particular $\boldsymbol{\theta}$ requires mainly to:

- Compute the log-determinant $\log|(1/f_{\boldsymbol{\theta}})(\boldsymbol{S}_{\boldsymbol{\theta}})| = -\log|f_{\boldsymbol{\theta}}(\boldsymbol{S}_{\boldsymbol{\theta}})|$ which is done through Equation (5.11) and involves a limited number of Chebyshev filtering operations with $\boldsymbol{S}_{\boldsymbol{\theta}}$.
- Compute the quadratic term $\boldsymbol{\mu}_{\boldsymbol{\theta}^{(k)}}^T\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}\boldsymbol{\mu}_{\boldsymbol{\theta}^{(k)}}$, which requires a single product between $\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}$ and $\boldsymbol{\mu}_{\boldsymbol{\theta}^{(k)}}$, as $\boldsymbol{\mu}_{\boldsymbol{\theta}^{(k)}}$ is computed and stored once and for all. Hence, the cost of this operation is basically that of a single Chebyshev filtering operation with $\boldsymbol{S}_{\boldsymbol{\theta}}$.
- Compute the trace term $\mathrm{Trace}(\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}^{(k)}}^{-1})$.

The trace term in Equation (5.16) poses a problem. Indeed, as building the matrices $\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}$ and $\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}^{(k)}}$ is out of the question, this term should be approximated using an approach similar to the one outlined for the trace of graph filters (cf. Section 2.3.1). Indeed, we can for instance write

$$\mathrm{Trace}(\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}^{(k)}}^{-1}) = \mathbb{E}[\boldsymbol{W}^T\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}^{(k)}}^{-1}\boldsymbol{W}] = \mathbb{E}[(\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}\boldsymbol{W})^T\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}^{(k)}}^{-1}\boldsymbol{W}] \quad , \tag{5.17}$$

where $\boldsymbol{W}$ is a zero-mean random vector with covariance matrix $\boldsymbol{I}_n$ (cf. Proposition A.3.5). This term can therefore be approximated using a Monte-Carlo estimate, similarly to what was done for the trace of graph filters in Section 2.3.1. Precisely, if $\boldsymbol{w}^{(1)}, \ldots, \boldsymbol{w}^{(N)}$ denote N realizations of $\boldsymbol{W}$, then we write

$$\mathrm{Trace}(\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}^{(k)}}^{-1}) \approx \frac{1}{N}\sum_{i=1}^{N}\left(\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}\boldsymbol{w}^{(i)}\right)^T\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}^{(k)}}^{-1}\boldsymbol{w}^{(i)} \quad . \tag{5.18}$$

In practice, as in Section 2.3.1, the entries of $\boldsymbol{W}$ are independent and identically distributed variables following either a Gaussian or a Rademacher distribution. Hence the same conclusions regarding the link between the sample size N and the approximation accuracy can be extended to this case.

Computing the approximation in Equation (5.18) then requires to:

- Compute N products between $\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}$ and a vector: as seen earlier, such products amount to the cost of a Chebyshev filtering operation with shift operator $\boldsymbol{S}_{\boldsymbol{\theta}}$ (and transfer function $(1/f_{\boldsymbol{\theta}})$).

- Solving N linear system defined by the matrix $\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}^{(k)}}$: this can be done using a descent algorithm (or eventually a Chebyshev filtering operation with shift operator $\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}^{(k)}}$ and transfer function $\lambda \mapsto 1/\lambda$).

When solving the minimization problem of Equation (5.16), given that the objective function is evaluated several times, several evaluations of the trace term are performed for a fixed value of $\boldsymbol{\theta}^{(k)}$ but varying values of $\boldsymbol{\theta}$. In this case, we can actually reuse the solutions of the linear system from one evaluation to the other as they depend only on $\boldsymbol{\theta}^{(k)}$. Hence, they can be computed once and for all at the beginning of the minimization process, thus reducing the cost of evaluating the trace term to that of performing the products between $\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}$ and vectors. Algorithm 5.3 summarizes this approach of likelihood maximization by EM.

Algorithm 5.3: EM algorithm for likelihood maximization by trace approximations.

Input: An observation vector $\boldsymbol{z}_o$ from a process defined by Equation (5.1).
Families of spectral densities $\{f_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}}$, variance parameters $\{\tau_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}}$ and shift operators $\{\boldsymbol{S}_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}}$ parametrized by the same parameter vector $\boldsymbol{\theta} \in \mathbb{R}^{N_P}$.
An initial guess of parameter vector $\boldsymbol{\theta}^{(0)}$.
Output: An estimate of the parameter vector maximizing the likelihood given $\boldsymbol{z}_o$.

$k = 0$;
while *Convergence is not achieved* **do**

- Expectation step

Compute $\boldsymbol{\mu}_{\boldsymbol{\theta}^{(k)}} = \tau^{-2}_{\boldsymbol{\theta}^{(k)}} \widehat{\boldsymbol{Q}}^{-1}_{\boldsymbol{\theta}^{(k)}} \boldsymbol{M}_o^T \boldsymbol{z}_o$ (where $\widehat{\boldsymbol{Q}}^{-1}_{\boldsymbol{\theta}^{(k)}}$ is defined in Equation (5.8)) using a descent algorithm (cf. Algorithm 4.3 or 4.4) ;
for $i = 1, \ldots, N$ **do**
Generate and store a vector $\boldsymbol{w}^{(i)} \in \mathbb{R}^n$ with independent zero-mean and unit-variance entries ;
Compute and store $\boldsymbol{x}^{(i)} = \widehat{\boldsymbol{Q}}^{-1}_{\boldsymbol{\theta}^{(k)}} \boldsymbol{w}^{(i)}$ using a descent algorithm (cf. Algorithm 4.3 or 4.4) ;

- Maximization step

Solve the following minimization problem (using a general-purpose optimization algorithm):

$$\boldsymbol{\theta}^{(k+1)} = \underset{\boldsymbol{\theta} \in \mathbb{R}^{N_P}}{\operatorname{argmin}} \left(\frac{1}{N} \sum_{i=1}^{N} \left(\boldsymbol{x}^{(i)} \right)^T \widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}} \boldsymbol{w}^{(i)} + \boldsymbol{\mu}^T_{\boldsymbol{\theta}^{(k)}} \widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}} \boldsymbol{\mu}_{\boldsymbol{\theta}^{(k)}} + \frac{1}{\tau^2_{\boldsymbol{\theta}}} \boldsymbol{z}_o^T \left(\boldsymbol{z}_o - 2\boldsymbol{M}_o \boldsymbol{\mu}_{\boldsymbol{\theta}^{(k)}} \right) \right.$$
$$\left. + 2d \log \tau_{\boldsymbol{\theta}} - \log \left| (1/f_{\boldsymbol{\theta}})(\boldsymbol{S}_{\boldsymbol{\theta}}) \right| \right) .$$

$k \leftarrow k + 1$;

Return $\boldsymbol{\theta}^{(k)}$.

Each iteration of Algorithm 5.3 can be decomposed into two steps:

- A preprocessing step that amounts to generate and store N random n-vectors, solving $N + 1$ linear systems involving $\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}^{(k)}}$ and storing the results (which are n-vectors). Note that each product between $\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}^{(k)}}$ and a vector involves a Chebyshev filtering operation and that a total of $2N + 1$ n-vectors need to be stored.

- An optimization step that consists in minimizing a function whose evaluation amounts to $N + 1$ products between $\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}$ and vectors and a Chebyshev filtering operation.

The memory requirements of Algorithm 5.3 can be reduced by using a different approach to the approximation of the trace term $\mathrm{Trace}(\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}} \widehat{\boldsymbol{Q}}^{-1}_{\boldsymbol{\theta}^{(k)}})$ than the one presented in Equation (5.18). Indeed, following Proposition A.4.11, we have

$$\mathrm{Cov}[\boldsymbol{W}^T \widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}} \boldsymbol{W}, \boldsymbol{W}^T \widehat{\boldsymbol{Q}}^{-1}_{\boldsymbol{\theta}^{(k)}} \boldsymbol{W}] = 2\mathrm{Trace}(\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}} \widehat{\boldsymbol{Q}}^{-1}_{\boldsymbol{\theta}^{(k)}}) \quad ,$$

where $\boldsymbol{W}$ is a zero-mean *Gaussian* vector with covariance matrix $\boldsymbol{I}_n$. Using once again a Monte-Carlo estimate, the trace term can therefore be approximated using a sequence $\boldsymbol{w}^{(1)}, \dots, \boldsymbol{w}^{(N)}$ of N independent realizations of $\boldsymbol{W}$ as the sample covariance of the set of pairs

$$\left\{ \left((\boldsymbol{w}^{(i)})^T \widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}} \boldsymbol{w}^{(i)}, (\boldsymbol{w}^{(i)})^T \widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}^{(k)}}^{-1} \boldsymbol{w}^{(i)} \right) \right\}_{i \in [\![1,N]\!]} .$$

Hence,

$$\mathrm{Trace}(\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}} \widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}^{(k)}}^{-1}) \approx \frac{1}{2(N-1)} \sum_{i=1}^{N} \left(t_{\boldsymbol{\theta}}^{(i)} - \bar{t}_{\boldsymbol{\theta}} \right) \left(s_{\boldsymbol{\theta}^{(k)}}^{(i)} - \bar{s}_{\boldsymbol{\theta}^{(k)}} \right) = \frac{1}{2(N-1)} \sum_{i=1}^{N} t_{\boldsymbol{\theta}}^{(i)} \left(s_{\boldsymbol{\theta}^{(k)}}^{(i)} - \bar{s}_{\boldsymbol{\theta}^{(k)}} \right) \tag{5.19}$$

where on one hand,

$$t_{\boldsymbol{\theta}}^{(i)} = (\boldsymbol{w}^{(i)})^T \widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}} \boldsymbol{w}^{(i)} \quad \text{and} \quad \bar{t}_{\boldsymbol{\theta}} = \frac{1}{N} \sum_{i=1}^{N} t_{\boldsymbol{\theta}}^{(i)} \quad ,$$

and on the other hand,

$$s_{\boldsymbol{\theta}^{(k)}}^{(i)} = (\boldsymbol{w}^{(i)})^T \widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}^{(k)}}^{-1} \boldsymbol{w}^{(i)} \quad \text{and} \quad \bar{s}_{\boldsymbol{\theta}^{(k)}} = \frac{1}{N} \sum_{i=1}^{N} s_{\boldsymbol{\theta}^{(k)}}^{(i)} \quad .$$

Computing the approximation in Equation (5.19) now requires to

- Compute $t_{\boldsymbol{\theta}}^{(i)}$ for $i \in [\![1, N]\!]$ by computing a product between between $\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}$ and a vector (this be done for the cost of a Chebyshev filtering operation).

- Compute $s_{\boldsymbol{\theta}^{(k)}}^{(i)}$ for $i \in [\![1, N]\!]$ by solving a linear system defined by $\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}^{(k)}}$ and can be done using for instance a descent algorithm.

The computational cost associated with this trace approximation is therefore basically the same as the cost associated with the previous one (in Equation (5.18)). The difference between them is in the quantities which are stored when several evaluations of the trace term are performed for a fixed value of $\boldsymbol{\theta}^{(k)}$ but varying values of $\boldsymbol{\theta}$. In this case, note that we can now reuse the coefficients $s_{\boldsymbol{\theta}^{(k)}}^{(i)}$, which only depend on $\boldsymbol{\theta}^{(k)}$. Hence, they can be computed once and for all at the beginning of the minimization process, thus reducing the cost of evaluating the trace term to that of computing the coefficients $t_{\boldsymbol{\theta}}^{(i)}$. Algorithm 5.4 summarizes this approach of likelihood maximization by EM.

Each iteration of Algorithm 5.4 can be decomposed into two steps:

- A preprocessing step that amounts to generate and store N random n-vectors, solving $N+1$ linear systems involving $\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}^{(k)}}$ and storing one of these solutions and $N+1$ scalar values. Hence, we need to store N less n-vectors compared to Algorithm 5.3.

- An optimization step that consists in minimizing a function whose evaluation amounts to $N+1$ products between $\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}$ and vectors and a Chebyshev filtering operation.

Hence, for basically the same computational cost as Algorithm 5.3, Algorithm 5.4 allows to save on the memory requirements by storing less vectors.

Let us quickly compare the direct maximization of $L(\cdot\,; \boldsymbol{z}_o)$ through its evaluations with Algorithm 5.1 with the minimization problem of Equation (5.16) induced by the EM approach.

On one hand, in Algorithms 5.3 and 5.4, heavy calculations requiring to solve a linear system involving $\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}^{(k)}}$, are precomputed once and for all so that the subsequent evaluations of the objective function only require a limited number of Chebyshev filtering operations with $\boldsymbol{S}_{\boldsymbol{\theta}}$. In comparison, when $L(\cdot\,; \boldsymbol{z}_o)$ is directly maximized, such systems have to be solved at each evaluation of the objective function.

On the other hand it should be noted that within the EM approach, an optimization problem must be solved at each iteration whereas a single optimization problem is solved in the likelihood approach.

Algorithm 5.4: Memory-saving EM algorithm for likelihood maximization by trace approximations.

Input: An observation vector $\boldsymbol{z}_o$ from a process defined by Equation (5.1).
Families of spectral densities $\{f_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}}$, variance parameters $\{\tau_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}}$ and shift operators $\{\boldsymbol{S}_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}}$ parametrized by the same parameter vector $\boldsymbol{\theta} \in \mathbb{R}^{N_P}$.
An initial guess of parameter vector $\boldsymbol{\theta}^{(0)}$.
Output: An estimate of the parameter vector maximizing the likelihood given $\boldsymbol{z}_o$.

$k = 0$;
while *Convergence is not achieved* **do**

- Expectation step

Compute $\boldsymbol{\mu}_{\boldsymbol{\theta}^{(k)}} = \tau^{-2}_{\boldsymbol{\theta}^{(k)}} \widehat{\boldsymbol{Q}}^{-1}_{\boldsymbol{\theta}^{(k)}} \boldsymbol{M}_o^T \boldsymbol{z}_o$ (where $\widehat{\boldsymbol{Q}}^{-1}_{\boldsymbol{\theta}^{(k)}}$ is defined in Equation (5.8)) using a descent algorithm (cf. Algorithm 4.3 or 4.4) ;
for $i = 1, \ldots, N$ **do**
Generate and store a vector $\boldsymbol{w}^{(i)} \in \mathbb{R}^n$ with independent zero-mean and unit-variance entries ;
Compute $\boldsymbol{x}^{(i)} = \widehat{\boldsymbol{Q}}^{-1}_{\boldsymbol{\theta}^{(k)}} \boldsymbol{w}^{(i)}$ using a descent algorithm (cf. Algorithm 4.3 or 4.4) ;
Store $s^{(i)}_{\boldsymbol{\theta}^{(k)}} = (\boldsymbol{w}^{(i)})^T \boldsymbol{x}^{(i)}$;
Store $\bar{s}_{\boldsymbol{\theta}^{(k)}} = \frac{1}{N} \sum_{i=1}^N s^{(i)}_{\boldsymbol{\theta}^{(k)}}$;

- Maximization step

Solve the following minimization problem (using a general purpose optimization algorithm):

$$\boldsymbol{\theta}^{(k+1)} = \underset{\boldsymbol{\theta} \in \mathbb{R}^{N_P}}{\operatorname{argmin}} \left(\frac{1}{2(N-1)} \sum_{i=1}^N \left(s^{(i)}_{\boldsymbol{\theta}^{(k)}} - \bar{s}_{\boldsymbol{\theta}^{(k)}} \right) t^{(i)}_{\boldsymbol{\theta}} + \boldsymbol{\mu}^T_{\boldsymbol{\theta}^{(k)}} \widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}} \boldsymbol{\mu}_{\boldsymbol{\theta}^{(k)}} + \frac{1}{\tau^2_{\boldsymbol{\theta}}} \boldsymbol{z}_o^T \left(\boldsymbol{z}_o - 2 \boldsymbol{M}_o \boldsymbol{\mu}_{\boldsymbol{\theta}^{(k)}} \right) + 2d \log \tau_{\boldsymbol{\theta}} - \log \left| (1/f_{\boldsymbol{\theta}})(\boldsymbol{S}_{\boldsymbol{\theta}}) \right| \right) \quad ,$$

where $t^{(i)}_{\boldsymbol{\theta}} = (\boldsymbol{w}^{(i)})^T \widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}} \boldsymbol{w}^{(i)}$. ;
$k \leftarrow k + 1$;
Return $\boldsymbol{\theta}^{(k)}$.

5.2.3 EM by conditional simulations

Starting from the formulation of the EM algorithm through its two steps, another approach can be taken to maximize the expectation function. Indeed, a Monte-Carlo estimate can be used to directly approximate $E_{\boldsymbol{\theta}^{(k)}}(\boldsymbol{\theta})$ using a set of conditional simulations of $\boldsymbol{Z}$. The expectation over $[\boldsymbol{Z} | \boldsymbol{z}_o \,;\, \boldsymbol{\theta}^{(k)}]$ in $E_{\boldsymbol{\theta}^{(k)}}(\boldsymbol{\theta})$ is then replaced by an average over a set of of N realizations $\boldsymbol{z}^{(1)}_{\boldsymbol{\theta}^{(k)}}, \ldots, \boldsymbol{z}^{(N)}_{\boldsymbol{\theta}^{(k)}}$ of this random vector, namely:

$$E_{\boldsymbol{\theta}^{(k)}}(\boldsymbol{\theta}) = \mathbb{E}\left[\widetilde{L}(\boldsymbol{\theta} \,;\, \boldsymbol{Z}_{\boldsymbol{\theta}^{(k)}}, \boldsymbol{z}_o) \right] \approx \frac{1}{N} \sum_{i=1}^n \widetilde{L}(\boldsymbol{\theta} \,;\, \boldsymbol{z}^{(i)}_{\boldsymbol{\theta}^{(k)}}, \boldsymbol{z}_o) \quad .$$

This approach was introduced by Wei and Tanner (1990) and is called *Monte-Carlo EM algorithm*.

Each conditional simulation $\boldsymbol{z}^{(i)}_{\boldsymbol{\theta}^{(k)}}$ is generated through Algorithm 4.2. They come at the price of a Chebyshev filtering operation with $\boldsymbol{S}_{\boldsymbol{\theta}^{(k)}}$ (for the non-conditional simulation) and the solving of a linear system involving $\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}^{(k)}}$ (for the conditioning through kriging). Note that the conditional simulations can be precomputed during the expectation step as they only depend on the parameter $\boldsymbol{\theta}^{(k)}$, which is fixed during the maximization step. Then, the maximization step

is reduced to the following optimization problem:

$$\boldsymbol{\theta}^{(k+1)} = \underset{\boldsymbol{\theta}\in\mathbb{R}^{N_P}}{\operatorname{argmin}} \left(\frac{1}{\tau_{\boldsymbol{\theta}}^2} \boldsymbol{z}_o^T (\boldsymbol{z}_o - 2\boldsymbol{M}_o \bar{\boldsymbol{z}}_{\boldsymbol{\theta}^{(k)}}) + \frac{1}{N} \sum_{i=1}^{n} \left(\boldsymbol{z}_{\boldsymbol{\theta}^{(k)}}^{(i)} \right)^T \widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}} \boldsymbol{z}_{\boldsymbol{\theta}^{(k)}}^{(i)} + 2d \log \tau_{\boldsymbol{\theta}} - \log \left| (1/f_{\boldsymbol{\theta}})(\boldsymbol{S}_{\boldsymbol{\theta}}) \right| \right) \quad , \tag{5.20}$$

where $\bar{\boldsymbol{z}}_{\boldsymbol{\theta}^{(k)}}$ denotes the mean of the conditional simulations:

$$\bar{\boldsymbol{z}}_{\boldsymbol{\theta}^{(k)}} = \frac{1}{N} \sum_{i=1}^{n} \boldsymbol{z}_{\boldsymbol{\theta}^{(k)}}^{(i)} \quad .$$

If the conditional simulations $\boldsymbol{z}_{\boldsymbol{\theta}^{(k)}}^{(1)}, \ldots, \boldsymbol{z}_{\boldsymbol{\theta}^{(k)}}^{(N)}$ are precomputed and stored, the evaluation of the objective function in Equation (5.20) requires that of N quadratic forms defined by $\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}$ and that of the log-determinant of the graph filter $(1/f_{\boldsymbol{\theta}})(\boldsymbol{S}_{\boldsymbol{\theta}})$. Algorithm 5.5 summarizes this new formulation of the EM algorithm.

Algorithm 5.5: EM algorithm for likelihood maximization by conditional simulations.

Input: An observation vector $\boldsymbol{z}_o$ from a process defined by Equation (5.1).
Families of spectral densities $\{f_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}}$, variance parameters $\{\tau_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}}$ and shift operators $\{\boldsymbol{S}_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}}$ parametrized by the same parameter vector $\boldsymbol{\theta} \in \mathbb{R}^{N_P}$.
An initial guess of parameter vector $\boldsymbol{\theta}^{(0)}$.
Output: An estimate of the parameter vector maximizing the likelihood given $\boldsymbol{z}_o$.

$k = 0$;
while *Convergence is not achieved* **do**
- Expectation step

for $i = 1, \ldots, N$ **do**
Generate a vector $\boldsymbol{w} \in \mathbb{R}^n$ with independent standard Gaussian entries ;
Compute a non-conditional simulation of $\boldsymbol{Z}$ under $\boldsymbol{\theta}^{(k)}$ by computing the vector $\boldsymbol{z}' = \sqrt{f_{\boldsymbol{\theta}^{(k)}}}(\boldsymbol{S}_{\boldsymbol{\theta}^{(k)}})\boldsymbol{w}$;
Generate a vector $\boldsymbol{w}_o' \in \mathbb{R}^d$ with independent standard Gaussian entries ;
Compute the residual kriging estimate, which is the solution $\boldsymbol{x}'$ of the linear system $\boldsymbol{x}' = \tau_{\boldsymbol{\theta}^{(k)}}^{-2} \widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}^{(k)}}^{-1} \boldsymbol{M}_o^T (\boldsymbol{z}_o - (\boldsymbol{M}_o \boldsymbol{z}' + \tau_{\boldsymbol{\theta}^{(k)}} \boldsymbol{w}_o'))$, (where $\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}^{(k)}}^{-1}$ is defined in Equation (5.8)) using a descent algorithm (cf. Algorithm 4.3 or 4.4) ;
Store $\boldsymbol{z}_{\boldsymbol{\theta}^{(k)}}^{(i)} = \boldsymbol{z}' + \boldsymbol{x}'$;

Store $\bar{\boldsymbol{z}}_{\boldsymbol{\theta}^{(k)}} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{z}_{\boldsymbol{\theta}^{(k)}}^{(i)}$;

- Maximization step

Solve the following minimization problem (using a general purpose optimization algorithm):

$$\boldsymbol{\theta}^{(k+1)} = \underset{\boldsymbol{\theta}\in\mathbb{R}^{N_P}}{\operatorname{argmin}} \left(\frac{1}{\tau_{\boldsymbol{\theta}}^2} \boldsymbol{z}_o^T (\boldsymbol{z}_o - 2\boldsymbol{M}_o \bar{\boldsymbol{z}}_{\boldsymbol{\theta}^{(k)}}) + \frac{1}{N} \sum_{i=1}^{n} \left(\boldsymbol{z}_{\boldsymbol{\theta}^{(k)}}^{(i)} \right)^T \widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}} \boldsymbol{z}_{\boldsymbol{\theta}^{(k)}}^{(i)} + 2d \log \tau_{\boldsymbol{\theta}} - \log \left| (1/f_{\boldsymbol{\theta}})(\boldsymbol{S}_{\boldsymbol{\theta}}) \right| \right)$$

$k \leftarrow k + 1$;
Return $\boldsymbol{\theta}^{(k)}$.

Each iteration of Algorithm 5.5 can be decomposed into two steps:

- A preprocessing step that amounts to generate and store N conditional simulations, and therefore amounts to N Chebyshev filtering operations with shift operator $\boldsymbol{S}_{\boldsymbol{\theta}^{(k)}}$ and transfer function $\sqrt{f_{\boldsymbol{\theta}^{(k)}}}$; and solving N linear systems defined by $\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}^{(k)}}$. In total, we need to store $N + 1$ n-vectors at this step.

- An optimization step that consists in minimizing a function whose evaluation amounts to $N+1$ products between $\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}$ and vectors and a Chebyshev filtering operation.

Hence, Algorithms 5.4 and 5.5 basically operate with the same computational complexity and storage needs. One advantage of Algorithm 5.5 over Algorithm 5.4 would be that at each iteration of the algorithm, we actually compute estimators of the underlying field $\boldsymbol{Z}$ given the data $\boldsymbol{z}_o$, which are given by the conditional simulations and their average. Hence, in a context where the ultimate goal is SGS estimation, the estimators are readily available each step of the way using Algorithm 5.5.

5.3 Particular case: Inference with a known shift operator

In this section, we look into the particular case when the shift operator is fixed to a single value and known value $\boldsymbol{S}$, i.e. $\forall \boldsymbol{\theta}, \boldsymbol{S}_{\boldsymbol{\theta}} = \boldsymbol{S}$. As we may see, several simplifications of the algorithms introduced in the previous sections can be made to alleviate their computational and storage costs.

5.3.1 General remark

Whenever the shift operator is fixed, the following trick can be used to lighten the computational cost of the direct likelihood maximization relying on Algorithm 5.2 and Algorithms 5.3 to 5.5 based on the EM approach. Indeed, computational savings can be made for the evaluation of the log-determinant term $\log|(1/f_{\boldsymbol{\theta}})(\boldsymbol{S}_{\boldsymbol{\theta}})| = -\log|f_{\boldsymbol{\theta}}(\boldsymbol{S}_{\boldsymbol{\theta}})| = \log|(1/f_{\boldsymbol{\theta}})(\boldsymbol{S})| = -\log|f_{\boldsymbol{\theta}}(\boldsymbol{S})|$ that appears systematically in the objective function of the associated optimization problems.

Following the method introduced in Section 2.3.3, we can compute the histogram of eigenvalues of $\boldsymbol{S}$ once and for all using Algorithm 2.9, and use it as an additional input of Algorithms 5.2 to 5.5. Then, the log-determinant $\log|(1/f_{\boldsymbol{\theta}})(\boldsymbol{S})|$ can be estimated for any $f_{\boldsymbol{\theta}}$ using Equation (2.14), and therefore requiring only direct evaluation of a function on points of $\mathbb{R}$. Hence, the evaluation of the log-determinant would now require no graph filtering operation at all, and would in fact be totally inexpensive to compute compared to the other terms involved in the objective function.

In particular, for the implementation of the EM approaches of Algorithms 5.3 to 5.5, using this trick ensures that the cost of evaluation of the objective function in the optimization step is reduced to that of a predefined number of quadratic forms defined by the matrix $\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}$ (given in Equation (5.8)). This number is fixed by the user and corresponds to the degree of the approximation of the Monte-Carlo estimates used in these implementations.

5.3.2 Particular case: Polynomial spectral densities

We still assume in this subsection that the shift operator $\boldsymbol{S}$ of $\boldsymbol{Z}$ is fixed and known, and we aim at determining its spectral density f and the variance parameter of its observation process τ using parametrized families of both of them. We assume in particular in this section that the spectral density $f_{\boldsymbol{\theta}}$, or rather its inverse, is chosen from a family of polynomial functions of fixed degree and deduce desirable simplification for the implementation of the EM approaches of Algorithms 5.3 to 5.5.

For a vector parameter $\boldsymbol{\theta} = (\theta_1, \dots, \theta_{N_P})^T \in \mathbb{R}^{N_p}$ we therefore fix:

$$\frac{1}{f_{\boldsymbol{\theta}}} = \left(\sum_{k=1}^{N_P - 1} \theta_k \tilde{T}_{k-1} \right)^2 \quad \text{and} \quad \frac{1}{\tau_{\boldsymbol{\theta}}} = e^{\theta_{N_P}} \quad , \tag{5.21}$$

where $\tilde{T}_{k-1}$ denotes the $(k-1)$-th Chebyshev polynomial, shifted on an interval containing the eigenvalues of $\boldsymbol{S}$. Hence, we ensure that $\tau_{\boldsymbol{\theta}} > 0$ and that $f_{\boldsymbol{\theta}}(\boldsymbol{S})$ defines a covariance matrix.

Remark 5.3.1. As mentioned earlier, taking $1/f_{\boldsymbol{\theta}}$ to be a (low-degree) polynomial is actually equivalent to assuming an underlying Markovian model between the entries of the resulting SGS. This hypothesis is not unusual when working with Gaussian vectors. Indeed, the sparsity of the resulting precision matrices of their discretization allows for instance fast sample computations and likelihood computations (Rue and Held, 2005).

Denote by $p_{\boldsymbol{\theta}}$ the polynomial given by

$$p_{\boldsymbol{\theta}} = \sum_{k=1}^{N_P-1} \theta_k \tilde{T}_{k-1} \quad .$$

Hence $(1/f_{\boldsymbol{\theta}}) = p_{\boldsymbol{\theta}}^2$. Also, the matrix $\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}}$ defined in Equation (5.8) and appearing in the expression of the objective functions of Algorithms 5.3 to 5.5 now writes

$$\widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}} = p_{\boldsymbol{\theta}}^2(\boldsymbol{S}_{\boldsymbol{\theta}}) + \tau_{\boldsymbol{\theta}}^{-2} \boldsymbol{M}_o^T \boldsymbol{M}_o = \left(\sum_{k=1}^{N_P-1} \theta_k \tilde{T}_{k-1}(\boldsymbol{S}) \right)^2 + e^{2\theta_{N_P}} \boldsymbol{M}_o^T \boldsymbol{M}_o \quad .$$

Injecting Equation (5.21) in the expression of the objective functions of Algorithms 5.3 to 5.5 allows to actually derive an analytical expression for their gradients, and therefore to use for instance descent algorithms without having to estimate the gradients from evaluations of the function.

Indeed, note that these objective functions are the sum of four main types of terms: for $\boldsymbol{u}, \boldsymbol{v}$ vectors independent of $\boldsymbol{\theta}$, we have either quadratic terms of the form $\boldsymbol{v}^T \widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}} \boldsymbol{v}$, or the log-determinant $(1/f_{\boldsymbol{\theta}})(\boldsymbol{S})$, or terms of the form $\tau_{\boldsymbol{\theta}}^{-2} \boldsymbol{u}^T \boldsymbol{v}$ or the log of $\tau_{\boldsymbol{\theta}}$. Using the derivative formulas by Petersen and Pedersen (2008), the gradient of these terms (with respect to $\boldsymbol{\theta} \in \mathbb{R}^{N_p}$) is then given by

$$\nabla(\boldsymbol{v}^T \widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}} \boldsymbol{v}) = 2 \begin{pmatrix} \boldsymbol{v}^T \tilde{T}_0(\boldsymbol{S}) p_{\boldsymbol{\theta}}(\boldsymbol{S}) \boldsymbol{v} \\ \vdots \\ \boldsymbol{v}^T \tilde{T}_{N_P-2}(\boldsymbol{S}) p_{\boldsymbol{\theta}}(\boldsymbol{S}) \boldsymbol{v} \\ \hline \tau_{\boldsymbol{\theta}}^{-2} \boldsymbol{v}^T \boldsymbol{M}_o^T \boldsymbol{M}_o \boldsymbol{v} \end{pmatrix},$$

$$\nabla(\tau_{\boldsymbol{\theta}}^{-2} \boldsymbol{u}^T \boldsymbol{v}) = 2 \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \hline \tau_{\boldsymbol{\theta}}^{-2} \boldsymbol{u}^T \boldsymbol{v} \end{pmatrix}, \quad \nabla(\log \tau_{\boldsymbol{\theta}}) = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \hline -\theta_{N_P} \end{pmatrix},$$

$$\nabla(\log |(1/f_{\boldsymbol{\theta}})(\boldsymbol{S})|) = \nabla(2 \log |p_{\boldsymbol{\theta}}(\boldsymbol{S})|) = 2 \begin{pmatrix} \mathrm{Trace}\left(\tilde{T}_0(\boldsymbol{S}) p_{\boldsymbol{\theta}}(\boldsymbol{S})^{-1}\right) \\ \vdots \\ \mathrm{Trace}\left(\tilde{T}_{N_P-2}(\boldsymbol{S}) p_{\boldsymbol{\theta}}(\boldsymbol{S})^{-1}\right) \\ \hline 0 \end{pmatrix} \quad .$$

Assuming that the trick of Section 5.3.1 is used, computing the gradient $\nabla(\log |(1/f_{\boldsymbol{\theta}})(\boldsymbol{S})|)$ is as cheap as computing $\log |(1/f_{\boldsymbol{\theta}})(\boldsymbol{S})|$. Indeed, simply note that the entries of this gradient vector satisfy

$$\forall j \in [\![0, N_P - 2]\!], \quad \mathrm{Trace}\left(\tilde{T}_j(\boldsymbol{S}) p_{\boldsymbol{\theta}}(\boldsymbol{S})^{-1}\right) = \sum_{i=1}^{n} \frac{T_j}{p_{\boldsymbol{\theta}}}(\lambda_i) \quad ,$$

where $\lambda_1, \ldots, \lambda_n$ denote the actual eigenvalues of $\boldsymbol{S}$. This last sum can be approximated using the precomputed histogram of eigenvalues of $\boldsymbol{S}$ in the same way as in Section 2.3.3. Hence, the cost of computing the gradient of the objective function comes to that of evaluating the gradients of the form $\nabla(\boldsymbol{v}^T \widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}} \boldsymbol{v})$. Such gradients can easily by computed using two runs of the Chebyshev filtering with graph filter $p_{\boldsymbol{\theta}}(\boldsymbol{S})$ and the vector $\boldsymbol{v}$:

- The first run is actually used to compute the product $p_{\boldsymbol{\theta}}(\boldsymbol{S})\boldsymbol{v}$ and involves exactly $N_P - 2$ products between (a matrix as sparse as) $\boldsymbol{S}$ and vectors.
- For the second run, instead of using them to form the vector $p_{\boldsymbol{\theta}}(\boldsymbol{S})\boldsymbol{v}$, each product $\tilde{T}_j(\boldsymbol{S})\boldsymbol{v}$, $0 \leq j \leq N_P - 2$, generated during the run is extracted and used to compute the $(j+1)$-th entry of $\nabla(\boldsymbol{v}^T \widehat{\boldsymbol{Q}}_{\boldsymbol{\theta}} \boldsymbol{v})$.

Thus, computing the gradient of the objective function comes roughly at the cost of evaluating the objective function twice. Hence, optimization algorithms for non-linear problems using the gradient (or more generally first-order derivatives) of the objective function can easily be used to tackle the optimization task of Algorithms 5.3 to 5.5. We can for instance cite the gradient descent and the conjugate gradient algorithms who both find adaptation in the context of non-linear problems (Bertsekas, 1997).

Conclusion

In this chapter, we introduced two classes of algorithms designed to perform inference based on a noisy and partial observation of a stationary SGS. On one hand, the likelihood of the vector of observations was directly maximized using an optimization algorithm. The main drawback of this approach is the high cost associated with the evaluation of the objective function of the optimization problem. That is why an approach based on the EM algorithm was introduced as a possible alternative.

Three implementations of the EM algorithm were proposed. They all iterate two steps: a preprocessing step involving a predefined number of linear systems to solve, followed by an optimization step where the cost of evaluating the objective function was drastically reduced when compared to the direct approach. Finally simplifications and computational tricks were presented for the cases where the shift operator is assumed to be known, and when a Markov model is assumed on the graph signals.

This chapter actually concludes the first part of our work: practical solutions for the simulation, the estimation and the inference of SGSs have been introduced. Now that our algorithmic toolbox is complete, we turn to the motivation of this work: working with non-stationary Gaussian fields and complex domains. The aim for the second part of this dissertation is to present the framework and the results allowing to take on this challenge, and how they relate to the graph signal processing framework.

Part II

Generalized random fields

6

Differential and Riemannian geometry

Contents

Résumé

Le but de ce chapitre est d'introduire des notions de géométrie différentielle et riemannienne qui seront utilisées dans la suite du manuscrit. Il s'agit d'un mini-cours, basés sur plusieurs ouvrages de référence, et cherchant à apporter au lecteur une bonne intuition sur ces sujets.

Introduction

Gaussian random fields (GRF) are widely used to model spatially correlated data in environmental and earth sciences Chilès and Delfiner (2012); Lantuéjoul (2013); Wackernagel (2013). These data usually correspond to samples of a regionalized variable z, i.e. a variable defined on a spatial domain. Following the geostatistical paradigm, this regionalized variable is modeled in a probabilistic framework by a GRF: z is then seen as a particular realization of a GRF Z. Rather than characterizing directly the features of the regionalized variable z from its samples, the focus is set on deducing from these samples some features of the GRF Z. Conditioning methods are then used to revert back to the data and honor them in some sense.

Working with GRFs capable of modeling truthfully the particularities of the spatial data at hand is instrumental to the use of geostatistical methods. In some applications, these data can be defined on complex spatial domains such as arbitrary surfaces of a three-dimensional space, or showcase preferential directions of high correlation (also called anisotropy directions) that change over the domain. In both cases, the GRFs used in the geostatistical models should reflect these particular features.

The objective of the second part of this dissertation is to provide a general framework that can be used to define GRFs that account for the complex geometric features listed above. This framework is actually summarized by the title of this dissertation: "Generalized random fields on Riemannian manifolds". The basic idea is to define GRFs (or rather generalized random fields) on a mathematical object that allows to model both surfaces and local deformations on a spatial domain (the so-called Riemannian manifold).

The outline of the second part of this dissertation is as follows.

- We first introduce the reader to basic notions of differential and Riemannian geometry and to the central object they model: Riemannian manifolds. We show in particular why Riemannian manifolds are suited to the modeling problem we are trying to tackle (Chapter 6).

- Then, the framework allowing to work with (generalized) random fields on Riemannian manifolds is studied. We prove a theorem which links these fields to stochastic graph signals, thus opening the way to work with them using the framework and the tools introduced in the first part of this dissertation (Chapter 7).

- Next, this theorem is applied to derive finite element approximations of the modeled non-stationary fields, similarly as what is proposed by (Lindgren et al., 2011), and the convergence of this approximation is studied (Chapter 8).

- Finally, the power of this new framework is illustrated by applying it to practical problems involving real and synthetic data (Chapter 9).

As mentioned above, this particular chapter aims at providing the reader with some basic understanding of the notions of differential and Riemannian geometry used in this work. Several concepts, such as the notions of orientability and connections were deliberately omitted in order to focus the text on the key concepts that will actually be used in the next chapters. This summary is intended to be self-sufficient and is a condensed version of textbooks on differential and Riemannian geometry (listed hereafter).

For a more comprehensive understanding of the subject, the reader is referred to the books used to write this chapter. For an introduction on differential geometry, see (Abraham et al., 2012), (Lang, 2012), (Lee, 2012). For an introduction on Riemannian and spectral geometry, see (Bérard, 2006), (Canzani, 2013), (Craioveanu et al., 2013),(Jost, 2008), (Lablée, 2015).

6.1 Manifolds and differential geometry

6.1.1 Manifolds, charts, atlases and functions

A *manifold* $\mathcal{M}$ *of dimension* $d \geq 1$, also called *d-manifold*, is a topological space such that:

- $\mathcal{M}$ is a Hausdorff space: $\forall \boldsymbol{p}, \boldsymbol{q} \in \mathcal{M}$, there exists open subsets $U_{\boldsymbol{p}}, U_{\boldsymbol{q}}$ of $\mathcal{M}$ such that $\boldsymbol{p} \in U_{\boldsymbol{p}}$, $\boldsymbol{q} \in U_{\boldsymbol{q}}$ and $U_{\boldsymbol{p}} \cap U_{\boldsymbol{q}} = \emptyset$.
- $\mathcal{M}$ is second-countable, i.e. there exists a countable family $\mathcal{U} = \{U_i\}_{i \in \mathbb{N}}$ of open subsets of $\mathcal{M}$ such that any open subset $U \subset \mathcal{M}$ can be written as the union of a subfamily of $\mathcal{U}$.
- $\mathcal{M}$ is locally Euclidean of dimension d: every $\boldsymbol{p} \in \mathcal{M}$ has a neighborhood homeomorphic to an open set of $\mathbb{R}^d$.

Assumption 6.1. *All manifolds encountered in this work are assumed to be (topologically) connected, i.e. they cannot be expressed as the disjoint union of two open sets.*

Formally, for any point $\boldsymbol{p} \in \mathcal{M}$ there exists an open set $U_{\boldsymbol{p}}$ containing $\boldsymbol{p}$ and there exists $\phi : U_{\boldsymbol{p}} \to \widehat{U}_{\boldsymbol{p}} \subset \mathbb{R}^n$ that maps $U_{\boldsymbol{p}}$ towards a open subset $\widehat{U}_{\boldsymbol{p}}$ of $\mathbb{R}^d$, and such that ϕ is continuous, bijective and its inverse is also continuous (hence, ϕ is a homeomorphism). Manifolds can be seen as generalizations of the notions of curves and surfaces to higher dimensions. Each point of a manifold can be seen as described by a set of d "coordinates" given by its image through the homeomorphism ϕ.

Example 6.1.1. The simplest example of a d-manifold may be open domains of $\mathbb{R}^d$. Indeed, if $B \subset \mathbb{R}^d$ denotes an open domain of $\mathbb{R}^d$, equipped with the same topology as $\mathbb{R}^d$, then the three requirements that define a manifold are clearly verified by B. In particular, the identity map defines a homeomorphism between any open neighborhood of $\boldsymbol{p} \in B$ and an open set of $\mathbb{R}^d$.

In particular, $\mathbb{R}^d$ itself but also open balls of $\mathbb{R}^d$ of any (strictly positive) radius are d-manifolds.

Example 6.1.2. Let $\mathbb{S}^2$ denote the unit sphere of $\mathbb{R}^3$ (equipped with its natural Euclidean topology):

$$\mathbb{S}^2 = \{\boldsymbol{p} \in \mathbb{R}^3 : \|\boldsymbol{p}\|_2 = \sqrt{p_1^2 + p_2^2 + p_3^2} = 1\} \quad .$$

$\mathbb{S}^2$ inherits a topology from $\mathbb{R}^3$: indeed, open sets of $\mathbb{S}^2$ can be defined as intersections of $\mathbb{S}^2$ with open sets of $\mathbb{R}^3$. Hence $\mathbb{S}^2$ is second-countable as $\mathbb{R}^3$ is. Besides, with this topology, $\mathbb{S}^2$ is Hausdorff. Indeed, for any distinct points $\boldsymbol{p}, \boldsymbol{q} \in \mathbb{S}^2$ we can find a small enough open ball of $\mathbb{R}^3$ around each one of them such that the balls do not intersect. The open sets of $\mathbb{S}^2$ defined as the intersection of these balls with $\mathcal{S}$ then satisfy the Hausdorff property.

Now let $\boldsymbol{p} \in \mathbb{S}^2$ and consider the applications ψ, ψ' defined over the open set

$$\widehat{U} =]-\pi, \pi[\times]-\frac{\pi}{3}, \frac{\pi}{3}[$$

by:

$$\psi : (\theta, \xi) \in \widehat{U} \mapsto \begin{pmatrix} \cos(\theta)\cos(\xi) \\ \sin(\theta)\cos(\xi) \\ \sin(\xi) \end{pmatrix}, \quad \tilde{\psi} : (\theta, \xi) \in \widehat{U} \mapsto \begin{pmatrix} \sin(\xi) \\ \sin(\theta)\cos(\xi) \\ \cos(\theta)\cos(\xi) \end{pmatrix} \quad .$$

ψ and $\tilde{\psi}$ actually represent parametrizations of parts of a unit sphere using spherical coordinates (cf. Figure 6.1). As such they define two diffeomorphisms from open sets of $\mathbb{S}^2$ that cover $\mathbb{S}^2$, to open sets of $\mathbb{R}^2$. This proves that $\mathbb{S}^2$ is locally Euclidean of dimension 2.

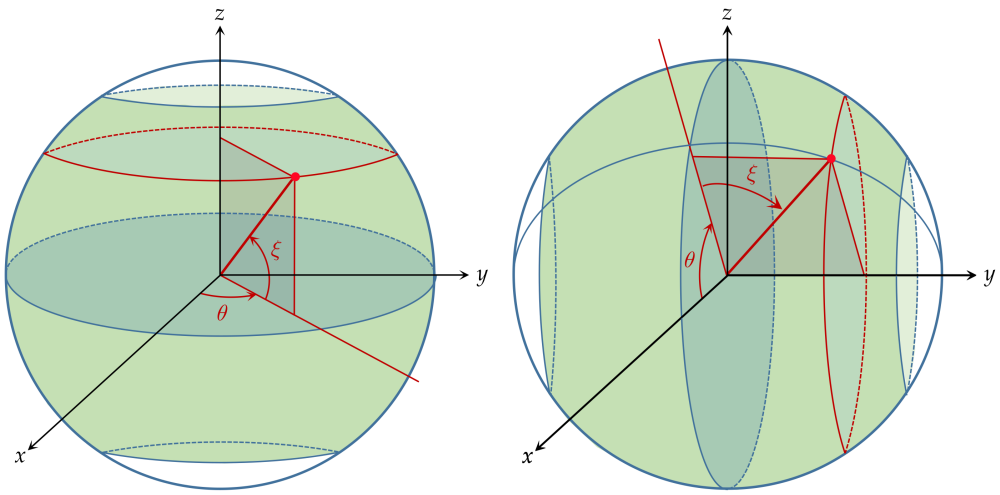


Figure 6.1: Illustration of the two parametrizations of $\mathbb{S}^2$ defined on Example 6.1.2. The figure on the left corresponds to ψ and the figure on the right corresponds to $\tilde{\psi}$. Any point of $\mathbb{S}^2$ can be retrieved by at least one of these diffeomorphisms.

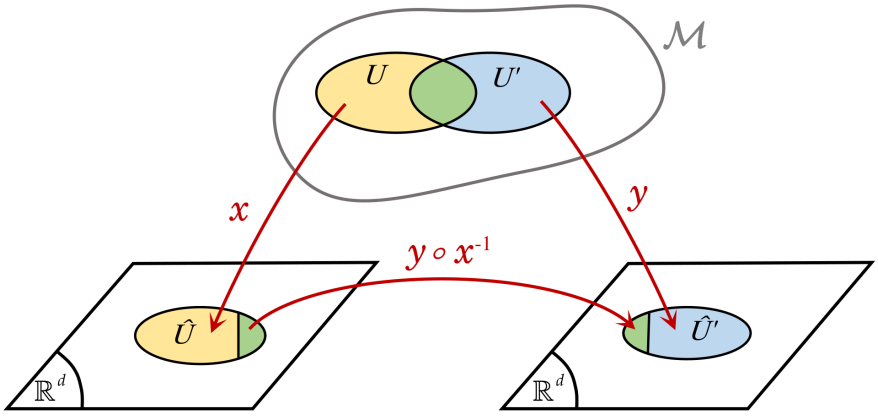


Figure 6.2: Illustration of a transition map. Two subsets U_α (in yellow) and U_β (in blue) of a manifold $\mathcal{M}$ and their intersection (in green) are represented.

More generally, if U is an open subset of $\mathcal{M}$ and $x : U \to \mathbb{R}^d$ a homeomorphism that maps U to an open subset $\widehat{U} = x(U)$ of $\mathbb{R}^d$, then the pair (U, x) is called a *coordinate chart* (or simply *chart*). Following the definition of manifolds, any point $\boldsymbol{p} \in \mathcal{M}$ is contained in the domain U of some coordinate chart (U, x): we then say that the chart (U, x) contains the point $\boldsymbol{p}$. In particular, the *coordinate functions* of x, denoted $x = (x_1, \ldots, x_d)$ and such that $\forall \boldsymbol{p} \in U$, $x(\boldsymbol{p}) = (x_1(\boldsymbol{p}), \ldots, x_d(\boldsymbol{p}))$ are called the *local coordinates* on U.

Let (U, x) and (U', y) denote two charts such that $U \cap U' \neq \emptyset$. The application $y \circ x^{-1} : U \cap U' \to \mathbb{R}^d$ is called *transition map (between U and U')*: it can actually be interpreted as an application turning local coordinates on U into local coordinates on U', as illustrated in Figure 6.2. Note that given that x and y are homeomorphisms, their associated transition map $y \circ x^{-1}$ is also a homeomorphism, with inverse $x \circ y^{-1}$. If besides $y \circ x^{-1}$ and its inverse are $\mathscr{C}^k$-differentiable, then by definition, $y \circ x^{-1}$ is a $\mathscr{C}^k$-diffeomorphism and the charts (U, x) and (U', y) are said to be $\mathscr{C}^k$*-compatible*. In particular, $\mathscr{C}^\infty$-compatible charts are also called *smoothly compatible* charts.

An *atlas* $\mathcal{A}$ is a collection of coordinate charts $\mathcal{A} = \{(U^{(\alpha)}, x^{(\alpha)}) : \alpha \in I\}$ of $\mathcal{M}$ indexed by a set I and such that $\cup_{\alpha \in I} U^{(\alpha)} = \mathcal{M}$. An atlas is said to be $\mathscr{C}^k$-differentiable if $\forall \alpha, \beta \in I$ such that $U^{(\alpha)} \cap U^{(\beta)} \neq \emptyset$, the transition map $x^{(\beta)} \circ (x^{(\alpha)})^{-1}$ is $\mathscr{C}^k$-differentiable. In particular, a $\mathscr{C}^\infty$-differentiable atlas is also called *smooth atlas*. Hence, a $\mathscr{C}^k$-differentiable (resp. smooth) atlas is simply a collection of charts that are pairwise $\mathscr{C}^k$-compatible (resp. smoothly compatible).

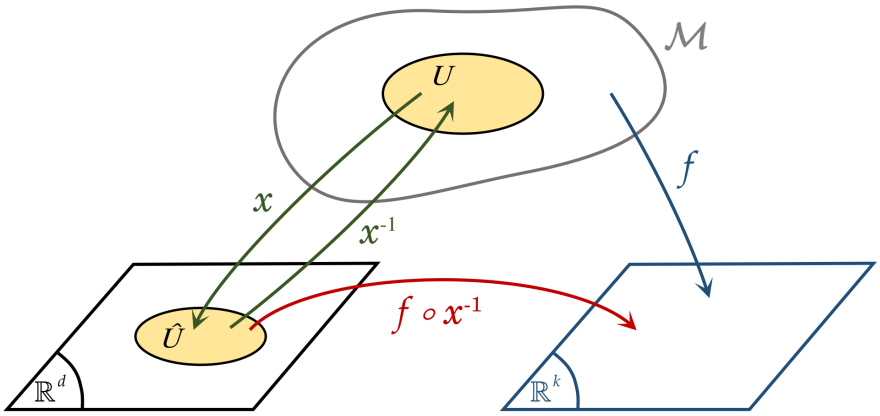


Figure 6.3: Illustration of a coordinate representation of a function.

Example 6.1.3. Following the notations of Example 6.1.2, denote $x = \psi^{-1}$ and $y = \tilde{\psi}^{-1}$. Let then U (resp. $\tilde{U}$) be the open subset of $\mathbb{S}^2$ defined by $U = \psi(\widehat{U})$ (resp. $U' = \tilde{\psi}(\widehat{U})$). Then, both (U, x) and $(U'y)$ are charts of $\mathbb{S}^2$.
Besides, $\mathcal{A} = \{(U, x); (U', y)\}$ defines an atlas of $\mathbb{S}^2$.

Two smooth atlases $\mathcal{A}_1$ and $\mathcal{A}_2$ are *compatible* if their union $\mathcal{A}_1 \cup \mathcal{A}_2$ is also a smooth atlas: in particular, this means that any chart in $\mathcal{A}_1$ is smoothly compatible with all charts in $\mathcal{A}_2$ (and vice-versa). One can check that atlas compatibility defines an equivalence relation.

Given some atlas of reference $\mathcal{A}$, let $\mathcal{C}_\mathcal{A}$ be an equivalence class for this relation that contains $\mathcal{A}$, i.e. $\mathcal{C}_\mathcal{A}$ is the set of all atlases that are compatible with $\mathcal{A}$. Then all atlases in $\mathcal{C}_\mathcal{A}$ are included inside a single smooth atlas, called maximal smooth atlas, and such that it contains any chart that is smoothly compatible with all charts in $\mathcal{A}$. The notion of *smooth manifold* is then defined as the association $(\mathcal{M}, \mathcal{A})$ of a manifold $\mathcal{M}$ with a maximal smooth atlas $\mathcal{A}$ (or equivalently its equivalence class of compatible atlases $\mathcal{C}_\mathcal{A}$). The notion of $\mathscr{C}^k$-differentiable manifold is defined similarly, by considering collections of $\mathscr{C}^k$-differentiable charts.

Let $(\mathcal{M}, \mathcal{A})$ be a smooth d-manifold and let $k \geq 1$. A function $f : \mathcal{M} \to \mathbb{R}^k$ is a *smooth function* if for any chart $(U, x) \in \mathcal{A}$, the function $f \circ x^{-1}$, called *coordinate representation of* f, is a smooth function of $x(U) \subset \mathbb{R}^d$ (cf. Figure 6.3). Of particular interest is the case where $k = 1$, i.e. f is real-valued. The set of real-valued smooth functions of $\mathcal{M}$ is denoted $\mathscr{C}^\infty(\mathcal{M})$.

6.1.2 Submanifolds of $\mathbb{R}^n$

Let $n \geq 1$. Of particular interest in this thesis are (embedded) submanifolds of $\mathbb{R}^n$, which are subsets of $\mathbb{R}^n$ having the defining properties of a manifold. They are embedded in $\mathbb{R}^n$ through the inclusion map, meaning that the topology on submanifolds of $\mathbb{R}^n$ is actually the *trace topology* of $\mathbb{R}^n$. Hence, open sets of a submanifold of $\mathbb{R}^n$ are defined as the intersection of open sets of $\mathbb{R}^n$ with the subset of $\mathbb{R}^n$ defining the submanifold.

Formally, for $d \leq n$, a *d-submanifold of* $\mathbb{R}^n$ is a subset $\mathcal{M} \subset \mathbb{R}^n$ such that $\forall \boldsymbol{p} \in \mathcal{M}$, there exists an open neighborhood of $\boldsymbol{p}$, denoted $\mathscr{V}(\boldsymbol{p}) \subset \mathbb{R}^n$ and a diffeomorphism $\phi : \mathbb{R}^n \to \mathbb{R}^n$ such that

$$\phi(\mathcal{M} \cap \mathscr{V}(\boldsymbol{p})) = \phi(\mathscr{V}(\boldsymbol{p})) \cap \left(\mathbb{R}^d \times \{\mathbf{0}_{n-d}\}\right)$$

Therefore, ϕ associates to any point $\boldsymbol{q} \in \mathcal{M} \cap \mathscr{V}(\boldsymbol{p})$, a unique set of d real values, which corresponds to the first d entries of $\phi(\boldsymbol{q}) \in \mathbb{R}^n$, the $n-d$ remaining entries of this n-vector being always zero. The pair $(\mathcal{M} \cap \mathscr{V}(\boldsymbol{p}), \phi)$ hence corresponds to a chart as defined for abstract manifolds.

6.1.3 Tangent space

The notion of tangent space of a manifold generalizes that of tangent line of a parametrized curve: a tangent space at a point $\boldsymbol{p} \in \mathcal{M}$ can therefore be thought of as a "linear" approximation of $\mathcal{M}$ in a small neighborhood of $\boldsymbol{p}$. These notions are generalized to the rather abstract case of

manifolds by defining tangent vectors (i.e. the elements of a tangent space) through their action on smooth functions defined on the manifold, much like tangent vectors of $\mathbb{R}^2$ can be seen as directional derivatives of smooth curves defined on this same domain.

A *tangent vector of $\mathcal{M}$ at a point $\boldsymbol{p} \in \mathcal{M}$* is a map $t_{\boldsymbol{p}} : \mathscr{C}^\infty(\mathcal{M}) \to \mathbb{R}$ that satisfies the following properties:

- Linearity: $\forall f, g \in \mathscr{C}^\infty(\mathcal{M})$, $\forall \alpha \in \mathbb{R}$: $t_{\boldsymbol{p}}(\alpha f + g) = \alpha t_{\boldsymbol{p}}(f) + t_{\boldsymbol{p}}(g)$.
- Leibniz rule: $\forall f, g \in \mathscr{C}^\infty(\mathcal{M})$: $t_{\boldsymbol{p}}(fg) = g(p)t_{\boldsymbol{p}}(f) + f(p)t_{\boldsymbol{p}}(g)$.

One can show (Lee, 2012, Corollary 3.3) that the set $T_{\boldsymbol{p}}\mathcal{M}$ of all tangent vectors at a point $\boldsymbol{p} \in \mathcal{M}$, which is called *tangent space at* $\boldsymbol{p}$, is a vector space of dimension d defined by

$$T_{\boldsymbol{p}}\mathcal{M} = \operatorname{span}\left\{ \left.\frac{\partial}{\partial x_i}\right|_{\boldsymbol{p}} : i \in [\![1, d]\!] \right\} \quad , \tag{6.1}$$

where the tangent vectors $\partial/\partial x_i|_{\boldsymbol{p}}, i \in [\![1, d]\!]$ are called *directional derivatives* and are defined, for a choice of chart $(U, x) \in \mathcal{A}$ containing $\boldsymbol{p}$, by:

$$\forall f \in \mathscr{C}^\infty(\mathcal{M}), \quad \left.\frac{\partial}{\partial x_i}\right|_{\boldsymbol{p}} (f) = \frac{\partial f}{\partial x_i}(\boldsymbol{p}) := \partial_i(f \circ x^{-1})(x(\boldsymbol{p}))$$

Here, $\partial_i(f \circ x^{-1})(x(\boldsymbol{p}))$ denotes the usual i-th partial derivative at the point $x(\boldsymbol{p}) \in \widehat{U}$ of the function $f \circ x^{-1} : \widehat{U} \subset \mathbb{R}^d \to \mathbb{R}$:

$$\partial_i(f \circ x^{-1})(x(\boldsymbol{p})) = \lim_{t \to 0} \frac{f \circ x^{-1}(x(\boldsymbol{p}) + t\boldsymbol{e}_i) - f \circ x^{-1}(x(\boldsymbol{p}))}{t} = \lim_{t \to 0} \frac{f \circ x^{-1}(x(\boldsymbol{p}) + t\boldsymbol{e}_i) - f(\boldsymbol{p})}{t}$$

Note that, given that $\mathcal{M}$ is a smooth manifold, $f \circ x^{-1}$ is a smooth function of $\mathbb{R}^d$ and therefore this quantity is well defined.

Following Equation (6.1), any tangent vector $t_{\boldsymbol{p}} \in T_{\boldsymbol{p}}\mathcal{M}$ can be represented by a vector $\boldsymbol{t}_{\boldsymbol{p}}^x \in \mathbb{R}^d$ such that

$$t_{\boldsymbol{p}} = \sum_{i=1}^{d} [\boldsymbol{t}_{\boldsymbol{p}}^x]_i \left.\frac{\partial}{\partial x_i}\right|_{\boldsymbol{p}} \tag{6.2}$$

The vector $\boldsymbol{t}_{\boldsymbol{p}}^x \in \mathbb{R}^d$, called *representative vector* of $t_{\boldsymbol{p}}$ with respect to the chart (U, x), simply contains the coordinates of $t_{\boldsymbol{p}}$ in the particular basis given in Equation (6.1). Conversely any $\boldsymbol{t}_{\boldsymbol{p}}^x \in \mathbb{R}^d$, defines an element $t_{\boldsymbol{p}}$ of $T_{\boldsymbol{p}}\mathcal{M}$ by Equation (6.2). Hence tangent vectors can be seen as both directional derivatives and vectors of $\mathbb{R}^d$ attached to a particular point of the manifold.

Example 6.1.4. Let $B \subset \mathbb{R}^d$ be an open domain of $\mathbb{R}^d$, seen as a d-manifold. The chart (B, x^{Euc}) where x^{Euc} maps points of B to their Cartesian coordinates, covers the whole manifold. Note that x^{Euc} is actually the restriction to B of the identity map of $\mathbb{R}^d$.

Let $\boldsymbol{p} \in B$. Then for every $k \in [\![1, d]\!]$, the directional derivative $\partial/\partial x_k^{\text{Euc}}|_{\boldsymbol{p}}$, corresponds exactly to the application that maps a smooth function on $B \subset \mathbb{R}^d$ to its usual k-th partial derivative at $\boldsymbol{p}$: $\partial/\partial x_k^{\text{Euc}}|_{\boldsymbol{p}} = \partial_k|_{\boldsymbol{p}}$.

Moreover for a tangent vector $t_{\boldsymbol{p}} \in T_{\boldsymbol{p}}B$ with representative vector $\boldsymbol{t}_{\boldsymbol{p}}^{\text{Euc}} \in \mathbb{R}^d$ with respect to the chart (B, x^{Euc}), we have

$$\forall f \in \mathscr{C}^\infty(B), \quad t_{\boldsymbol{p}}(f) = \sum_{i=1}^{d} [\boldsymbol{t}_{\boldsymbol{p}}^{\text{Euc}}]_i \partial_i f(\boldsymbol{p}) = \nabla f(\boldsymbol{p})^T \boldsymbol{t}_{\boldsymbol{p}}^{\text{Euc}} = \lim_{h \to 0} \frac{f(\boldsymbol{p} + h\boldsymbol{t}_{\boldsymbol{p}}^{\text{Euc}}) - f(\boldsymbol{p})}{h} \quad ,$$

where $\nabla f(\boldsymbol{p})$ denotes the gradient of $f : B \subset \mathbb{R}^d \to \mathbb{R}$ at $\boldsymbol{p}$. Hence $t_{\boldsymbol{p}}(f)$ is the (usual) directional derivative of f at $\boldsymbol{p}$ along the direction $\boldsymbol{t}_{\boldsymbol{p}}^{\text{Euc}}$.

Note that if another chart (U', y) is chosen to define the basis of Equation (6.1), then the chain rule (cf. Theorem A.1.1) allows to conclude that the relation between both basis is given by

$$\left.\frac{\partial}{\partial y_i}\right|_{\boldsymbol{p}} = \sum_{j=1}^{d} \frac{\partial x_j}{\partial y_i}(\boldsymbol{p}) \left.\frac{\partial}{\partial x_j}\right|_{\boldsymbol{p}} \quad , \tag{6.3}$$

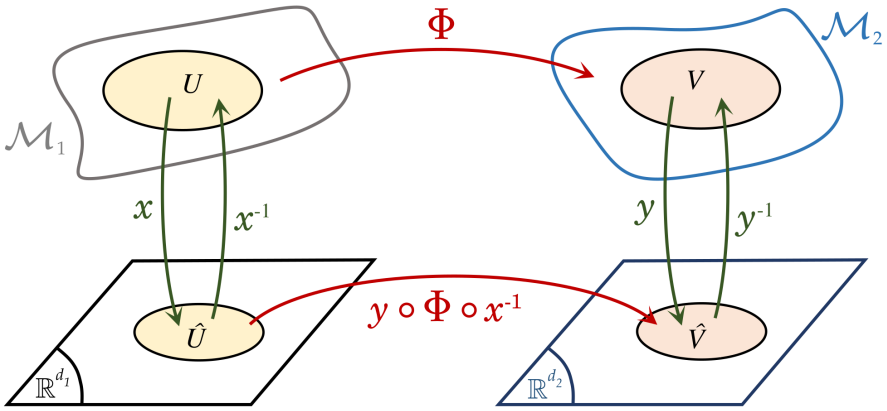


Figure 6.4: Illustration of a map between manifolds.

where $\partial x_j/\partial y_i(\boldsymbol{p})$ is the image of the function $\boldsymbol{p} \mapsto x_j(\boldsymbol{p})$ through the tangent vector $\partial/\partial y_i|_{\boldsymbol{p}}$, or equivalently the i-th partial derivative (with respect to the coordinate system (U', y) also containing $\boldsymbol{p}$) of the j-th component of the coordinate function x. In particular, applying this relation to Equation (6.2) gives a link between the coordinates of a tangent vector $t_{\boldsymbol{p}}$ in both bases.

Proposition 6.1.1. *Let $\mathcal{M}$ be a d-manifold and for $\boldsymbol{p} \in \mathcal{M}$ let $t_{\boldsymbol{p}} \in T_{\boldsymbol{p}}\mathcal{M}$.*

Then for any coordinate charts (U, x) and (U', y) containing $\boldsymbol{p}$, the representative vectors of $t_{\boldsymbol{p}}$ denoted $\boldsymbol{t}^x_{\boldsymbol{p}} \in \mathbb{R}^d$ in the basis of directional derivatives $\{\partial/\partial x_k|_{\boldsymbol{p}}\}_{1\leq k\leq d}$ and $\boldsymbol{t}^y_{\boldsymbol{p}} \in \mathbb{R}^d$ in the basis of directional derivatives $\{\partial/\partial y_k|_{\boldsymbol{p}}\}_{1\leq k\leq d}$ satisfy

$$\boldsymbol{t}^x_{\boldsymbol{p}} = J_{x\circ y^{-1}}(y(\boldsymbol{p}))\boldsymbol{t}^y_{\boldsymbol{p}} \quad ,$$

where $J_{x\circ y^{-1}}(y(\boldsymbol{p}))$ denotes the Jacobian matrix of the application $x \circ y^{-1} : y(U') \subset \mathbb{R}^d \to x(U) \subset \mathbb{R}^d$ at the point $y(\boldsymbol{p}) \in \mathbb{R}^d$.

Finally, the *tangent bundle* of $\mathcal{M}$, denoted $T\mathcal{M}$, is the disjoint union of all the tangent spaces of $\mathcal{M}$:

$$T\mathcal{M} = \bigsqcup_{\boldsymbol{p}\in\mathcal{M}} T_{\boldsymbol{p}}\mathcal{M} \quad \text{(disjoint union)} \quad .$$

6.1.4 Maps and differentials

Let $(\mathcal{M}_1, \mathcal{A}_1)$ be a smooth d_1-manifold and $(\mathcal{M}_2, \mathcal{A}_2)$ be a smooth d_2-manifold. A map $\Phi : \mathcal{M}_1 \to \mathcal{M}_2$ is a *smooth map* if:

- For all $\boldsymbol{p} \in \mathcal{M}_1$ there exists a chart $(U, x) \in \mathcal{A}_1$ containing $\boldsymbol{p}$ and a chart $(V, y) \in \mathcal{A}_2$ containing $\Phi(\boldsymbol{p})$ such that $\Phi(U) \subset V$.
- The composite map $y \circ \Phi \circ x^{-1}$ from $\widehat{U} = x(U)$ to $\widehat{V} = y(V)$ is smooth.

An illustration of the different building blocks of a smooth map is provided in Figure 6.4. In particular, smooth maps are continuous, and composition of smooth maps are also smooth. Examples of smooth maps include constant maps (i.e. applications that map all $\boldsymbol{p} \in \mathcal{M}_1$ to the same point $\boldsymbol{q} \in \mathcal{M}_2$) and the identity map (from $\mathcal{M}_1$ to $\mathcal{M}_1$).

A map $\Phi : \mathcal{M}_1 \to \mathcal{M}_2$ is a *diffeomorphism between manifolds* if it is a bijective smooth map whose inverse is also a smooth map. If such a map exists, then $\mathcal{M}_1$ and $\mathcal{M}_2$ are said to be diffeomorphic. In particular, only manifolds having the same dimension can be diffeomorphic.

Let $\Phi : \mathcal{M}_1 \to \mathcal{M}_2$ be a smooth map. The *differential of Φ at a point $\boldsymbol{p} \in \mathcal{M}_1$* is the map $d\Phi_{\boldsymbol{p}}$ from the tangent space of $\mathcal{M}_1$ at $\boldsymbol{p}$ to the tangent space of $\mathcal{M}_2$ at $\Phi(\boldsymbol{p}) \in \mathcal{M}_2$:

$$d\Phi_{\boldsymbol{p}} : T_{\boldsymbol{p}}\mathcal{M}_1 \to T_{\Phi(\boldsymbol{p})}\mathcal{M}_2 \quad ,$$

such that $\forall t_{\boldsymbol{p}} \in T_{\boldsymbol{p}}\mathcal{M}_1$, $d\Phi_{\boldsymbol{p}}(t_{\boldsymbol{p}})$ is the tangent vector of $\mathcal{M}_2$ at $\Phi(\boldsymbol{p})$ defined by:

$$\forall f \in \mathscr{C}^\infty(\mathcal{M}_2), \quad d\Phi_{\boldsymbol{p}}(t_{\boldsymbol{p}})(f) = t_{\boldsymbol{p}}(f \circ \Phi) \quad .$$

In particular, this last equation is well-defined given that Φ is a smooth map and so, $f \circ \Phi$ is a smooth function from $\mathcal{M}_1$ to $\mathbb{R}$.

Two important properties of differentials of smooth maps should be kept in mind. First, they define linear maps between tangent spaces. Second, whenever the smooth map Φ is a diffeomorphism, then the differential at any point $\boldsymbol{p} \in \mathcal{M}_1$ is a bijective map that satisfies

$$(d\Phi_{\boldsymbol{p}})^{-1} = d(\Phi^{-1})_{\Phi(\boldsymbol{p})} : T_{\Phi(\boldsymbol{p})}\mathcal{M}_2 \to T_{\boldsymbol{p}}\mathcal{M}_1 \quad .$$

Hence, the inverse of the differential of Φ at $\boldsymbol{p} \in \mathcal{M}_1$ is the differential of the inverse of Φ at $\Phi(\boldsymbol{p}) \in \mathcal{M}_2$ (and therefore is also linear).

The action of the differential $d\Phi_{\boldsymbol{p}}$ on a tangent vector $t_{\boldsymbol{p}} \in T_{\boldsymbol{p}}\mathcal{M}_1$ can be made explicit using directional derivatives and the notion of Jacobian matrix, which we now define. Consider a chart $(U, x) \in \mathcal{A}_1$ containing $\boldsymbol{p}$ and a chart $(V, y) \in \mathcal{A}_2$ containing $\Phi(\boldsymbol{p})$. The *Jacobian matrix* $J_\Phi(\boldsymbol{p}) \in \mathscr{M}_{d_2,d_1}(\mathbb{R})$ *of* Φ *at* $\boldsymbol{p}$ with respect to the charts (U, x) and (V, y), is defined as the (usual) Jacobian matrix of the function $\widehat{\Phi} = y \circ \Phi \circ x^{-1} : x(U) \subset \mathbb{R}^{d_1} \to y(V) \subset \mathbb{R}^{d_2}$ at the point $x(\boldsymbol{p}) \in \mathbb{R}^{d_1}$:

$$J_\Phi(\boldsymbol{p}) := J_{y\circ\Phi\circ x^{-1}}(x(\boldsymbol{p})) = \left[\partial_j (y \circ \Phi \circ x^{-1})_i (x(\boldsymbol{p}))\right]_{\substack{1\le i\le d_2 \\ 1\le j\le d_1}}, \quad \boldsymbol{p} \in \mathcal{M}_1 \quad ,$$

where for $1 \le i \le d_2$, $(y \circ \Phi \circ x^{-1})_i$ denotes the i-th coordinate function of function $y \circ \Phi \circ x^{-1}$.

Proposition 6.1.2. *Let $(\mathcal{M}_1, \mathcal{A}_1)$ be a smooth d-manifold, $(\mathcal{M}_2, \mathcal{A}_2)$ a smooth $\tilde{d}$-manifold and $\Phi : \mathcal{M}_1 \to \mathcal{M}_2$ a smooth map.*
Let $\boldsymbol{p} \in \mathcal{M}_1$. Consider then a chart $(U, x) \in \mathcal{A}_1$ containing $\boldsymbol{p}$ and a chart $(V, y) \in \mathcal{A}_2$ containing $\Phi(\boldsymbol{p})$.

Then, $\forall t_{\boldsymbol{p}} \in T_{\boldsymbol{p}}\mathcal{M}_1$, with representative vector $\boldsymbol{t}^x_{\boldsymbol{p}} \in \mathbb{R}^d$ with respect to the chart (U, x), the image of $t_{\boldsymbol{p}}$ by the differential $d\Phi_{\boldsymbol{p}}$ of Φ at $\boldsymbol{p}$ satisfies

$$d\Phi_{\boldsymbol{p}}(t_{\boldsymbol{p}}) = \sum_{i=1}^{d_2} [J_\Phi(\boldsymbol{p})\boldsymbol{t}^x_{\boldsymbol{p}}]_i \left.\frac{\partial}{\partial y_i}\right|_{\Phi(\boldsymbol{p})} \in T_{\Phi(\boldsymbol{p})}\mathcal{M}_2 \quad ,$$

where $J_\Phi(\boldsymbol{p}) \in \mathscr{M}_{d_2,d_1}(\mathbb{R})$ is the Jacobian matrix of Φ at $\boldsymbol{p}$ with respect to the charts (U, x) and (V, y).

Hence the differential $d\Phi_{\boldsymbol{p}}$ maps the representative vector of a tangent vector of $T_{\boldsymbol{p}}\mathcal{M}_1$ to its product with the Jacobian matrix of Φ at $\boldsymbol{p}$.

Proof. This property is a direct consequence of the chain rule. □

6.2 Riemannian manifolds

The notion of geometry is now introduced on smooth manifolds, while relying on the same concepts as those used in Euclidean spaces. This seems a natural choice given that by definition, manifolds are locally Euclidean. In particular, the notions of length and angles between vectors "attached" to a point of a manifold are defined by introducing an inner product that is defined on each tangent space of the manifold. These inner products are chosen so that they define a "smooth" structure on the manifold called Riemannian metric, and the association of a manifold with a Riemannian metric is called a Riemannian manifold.

Defining a Riemannian metric on a manifold allows to define familiar geometric concepts on the manifold, such as lengths, angles and distances. The aim of this section is to introduce both the concept of Riemannian metric and its use to define the aforementioned geometric concepts. The next section will then focus on the development of an integration theory on (smooth) manifold, while once again relying on Riemannian metrics.

6.2.1 Riemannian metric

Let $\mathcal{M}$ be a smooth d-manifold. A *Riemannian metric* g on $\mathcal{M}$ is an application that "smoothly" associates to each point $\boldsymbol{p} \in \mathcal{M}$ a symmetric positive definite bilinear form $g(\boldsymbol{p})$ (also denoted

$g_{\boldsymbol{p}}$) defined on its tangent space $T_{\boldsymbol{p}}\mathcal{M}$. Namely, g associates to each $\boldsymbol{p} \in \mathcal{M}$ an application $g_{\boldsymbol{p}}$ defined by

$$g_{\boldsymbol{p}} : \begin{array}{ccc} T_{\boldsymbol{p}}\mathcal{M} \times T_{\boldsymbol{p}}\mathcal{M} & \to & \mathbb{R} \\ (u_{\boldsymbol{p}}, v_{\boldsymbol{p}}) & \mapsto & g_{\boldsymbol{p}}(u_{\boldsymbol{p}}, v_{\boldsymbol{p}}) \end{array} ,$$

such that

- $g_{\boldsymbol{p}}$ is symmetric bilinear: $\forall u_{\boldsymbol{p}}, v_{\boldsymbol{p}}, w_{\boldsymbol{p}} \in T_{\boldsymbol{p}}\mathcal{M}$, $\forall \lambda \in \mathbb{R}$: $g_{\boldsymbol{p}}(u_{\boldsymbol{p}}, v_{\boldsymbol{p}}) = g_{\boldsymbol{p}}(v_{\boldsymbol{p}}, u_{\boldsymbol{p}})$, $\quad g_{\boldsymbol{p}}(u_{\boldsymbol{p}} + w_{\boldsymbol{p}}, v_{\boldsymbol{p}}) = g_{\boldsymbol{p}}(u_{\boldsymbol{p}}, v_{\boldsymbol{p}}) + g_{\boldsymbol{p}}(w_{\boldsymbol{p}}, v_{\boldsymbol{p}})$, $\quad g_{\boldsymbol{p}}(\lambda u_{\boldsymbol{p}}, v_{\boldsymbol{p}}) = \lambda g_{\boldsymbol{p}}(u_{\boldsymbol{p}}, v_{\boldsymbol{p}})$
- $g_{\boldsymbol{p}}$ is positive definite: $\forall u_{\boldsymbol{p}} \in T_{\boldsymbol{p}}\mathcal{M}$, $u_{\boldsymbol{p}} \neq 0 \Rightarrow g_{\boldsymbol{p}}(u_{\boldsymbol{p}}, u_{\boldsymbol{p}}) > 0$.

The association $(\mathcal{M}, g)$ of a smooth manifold $\mathcal{M}$ and a Riemannian metric g defined on this manifold is then called a *Riemannian manifold*.

In particular, note that $g_{\boldsymbol{p}}$ actually defines an inner product on the vector space $T_{\boldsymbol{p}}\mathcal{M}$ and can be expressed using the local coordinates from a chart (U, x) containing $\boldsymbol{p}$ as

$$g_{\boldsymbol{p}}(u_{\boldsymbol{p}}, v_{\boldsymbol{p}}) = (\boldsymbol{u}_{\boldsymbol{p}}^x)^T \boldsymbol{G}^x(\boldsymbol{p}) \boldsymbol{v}_{\boldsymbol{p}}^x = \sum_{i=1}^{d} \sum_{j=1}^{d} G_{ij}^x(\boldsymbol{p}) [\boldsymbol{u}_{\boldsymbol{p}}^x]_i [\boldsymbol{v}_{\boldsymbol{p}}^x]_j \quad ,$$

where $\boldsymbol{u}_{\boldsymbol{p}}^x, \boldsymbol{v}_{\boldsymbol{p}}^x$ are the representative vectors of $u_{\boldsymbol{p}}, v_{\boldsymbol{p}} \in T_{\boldsymbol{p}}\mathcal{M}$ with respect to the chart (U, x) (as defined in Equation (6.2)), and $\boldsymbol{G}^x(\boldsymbol{p})$ is a symmetric positive definite matrix of size d, called *representative matrix of the metric g at $\boldsymbol{p} \in \mathcal{M}$ with respect to the chart* (U, x), and whose entries are defined by

$$[\boldsymbol{G}^x(\boldsymbol{p})]_{ij} = G_{ij}^x(\boldsymbol{p}) = g_{\boldsymbol{p}} \left(\left. \frac{\partial}{\partial x_i} \right|_{\boldsymbol{p}}, \left. \frac{\partial}{\partial x_j} \right|_{\boldsymbol{p}} \right), \quad 1 \leq i, j \leq d \quad . \tag{6.4}$$

The requirement that the Riemannian metric g "smoothly" maps points of the manifold to inner products on their tangent spaces then corresponds to requiring that $\forall k, j \in [\![1, d]\!]$, the maps $\boldsymbol{p} \mapsto G_{kj}^x(\boldsymbol{p})$ define smooth functions from U to $\mathbb{R}$.

Note that the representative matrix of a metric actually depends on the considered chart containing $\boldsymbol{p} \in \mathcal{M}$, as underlined by the superscript x in Equation (6.4). The following result provides a link between representative matrices of the same metric for different charts.

Proposition 6.2.1. *Let $(\mathcal{M}, g)$ be a Riemannian manifold and let $\boldsymbol{p} \in \mathcal{M}$. Consider (U, x) and (U', y) two charts of $\mathcal{M}$ containing $\boldsymbol{p}$. Then, the representative matrices of g with respect to both charts, as defined in Equation* (6.4)*, satisfy*

$$\boldsymbol{G}^y(\boldsymbol{p}) = J_{x \circ y^{-1}}(y(\boldsymbol{p}))^T \boldsymbol{G}^x(\boldsymbol{p}) J_{x \circ y^{-1}}(y(\boldsymbol{p})) \quad , \tag{6.5}$$

where $J_{x \circ y^{-1}}(y(\boldsymbol{p}))$ denotes the (usual) Jacobian matrix of the function $x \circ y^{-1} : y(U') \subset \mathbb{R}^d \to x(U) \subset \mathbb{R}^d$ at the point $y(\boldsymbol{p})$.

Proof. This result is a consequence of Proposition 6.1.1. □

Example 6.2.1 (Euclidean Metric)**.** Let B be an open domain of $\mathbb{R}^d$. The chart (B, x^{Euc}), where x^{Euc} is the inclusion map into $\mathbb{R}^d$, covers the whole manifold. The *Euclidean metric*, denoted g^{Euc}, is the Riemannian metric on B defined as the bilinear form that associates to any pair of tangent vectors of $T_{\boldsymbol{p}}B$ (where $\boldsymbol{p} \in B$) the dot product of their representative vectors with respect to the canonical chart (B, x^{Euc}). Hence, for any $\boldsymbol{p} \in B$,

$$\forall \boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^d, \quad g_{\boldsymbol{p}}^{\text{Euc}} \left(\sum_{i=1}^{d} [\boldsymbol{u}]_i \left. \frac{\partial}{\partial x_i^{\text{Euc}}} \right|_{\boldsymbol{p}}, \sum_{j=1}^{d} [\boldsymbol{v}]_j \left. \frac{\partial}{\partial x_j^{\text{Euc}}} \right|_{\boldsymbol{p}} \right) := \sum_{i=1}^{d} [\boldsymbol{u}]_i [\boldsymbol{v}]_i = \boldsymbol{u}^T \boldsymbol{v} \quad .$$

In particular, the representative matrix of the Euclidean metric g^{Euc} at $\boldsymbol{p}$ and with respect to (B, x^{Euc}) is the identity matrix.

One way to define a Riemannian metric on a manifold is to inherit it from another manifold equipped with its own metric, as detailed in the following result.

Proposition 6.2.2. *Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be two smooth manifolds, and let us assume that $\mathcal{M}_2$ is equipped with a Riemannian metric g'.*
Let us also assume that there exists a smooth map $\Phi : \mathcal{M}_1 \to \mathcal{M}_2$ such that its differential $d\Phi_{\boldsymbol{p}}$ at any point $\boldsymbol{p} \in \mathcal{M}_1$ is injective.
Then g' and Φ induce a Riemannian metric $\Phi^* g'$ on $\mathcal{M}$ which is defined as:

$$\forall \boldsymbol{p} \in \mathcal{M}, \forall u_{\boldsymbol{p}}, v_{\boldsymbol{p}} \in T_{\boldsymbol{p}}\mathcal{M}_1, \quad (\Phi^* g')_{\boldsymbol{p}}(u_{\boldsymbol{p}}, v_{\boldsymbol{p}}) = g'(d\Phi_{\boldsymbol{p}}(u_{\boldsymbol{p}}), d\Phi_{\boldsymbol{p}}(v_{\boldsymbol{p}}))$$

Proof. The injectivity of $d\Phi_{\boldsymbol{p}}$ ensures that $(\Phi^* g')_{\boldsymbol{p}}$ defines an inner product on $T_{\boldsymbol{p}}\mathcal{M}_1$ and the smoothness of g' and Φ ensures the smoothness of the metric $(\Phi^* g')$. □

In particular, following the notations of Proposition 6.2.2, $\Phi^* g'$ is called the *pullback metric* of g' by Φ and $(\mathcal{M}_1, \Phi^* g')$ defines a Riemannian manifold. A consequence of the proposition is that any smooth manifold $\mathcal{M}$ admits a Riemannian metric, that can be built by "gluing" together pullback metrics of Euclidean metrics defined on domains of charts of $\mathcal{M}$.

Theorem 6.2.3. *Every smooth manifold admits a Riemannian metric.*

Proof. See (Lee, 2012, Proposition 13.3) □

Hence, any smooth manifold can be seen as a Riemannian manifold, which is why we will focus on Riemannian manifolds for the rest of this chapter.

6.2.2 A few geometric notions on Riemannian manifolds

The metric of a Riemannian manifold allows to locally define classical geometric notions on the tangent space of each point of the manifold. Namely, if $(\mathcal{M}, g)$ denotes a Riemannian manifold, and $\boldsymbol{p} \in \mathcal{M}$:

- The *length of a tangent vector* $t_{\boldsymbol{p}} \in T_{\boldsymbol{p}}\mathcal{M}$ is defined as $\|t_{\boldsymbol{p}}\|_{g_{\boldsymbol{p}}} = \sqrt{g_{\boldsymbol{p}}(t_{\boldsymbol{p}}, t_{\boldsymbol{p}})}$. In particular, $\forall v_{\boldsymbol{p}} \in T_{\boldsymbol{p}}\mathcal{M}$ such that $v_{\boldsymbol{p}} \neq 0$, $v_{\boldsymbol{p}}/\sqrt{g_{\boldsymbol{p}}(v_{\boldsymbol{p}}, v_{\boldsymbol{p}})}$ has length 1.
- The *angle θ between two tangent vectors* $u_{\boldsymbol{p}}, v_{\boldsymbol{p}} \in T_{\boldsymbol{p}}\mathcal{M}$ is defined as

$$\cos\theta = \frac{g_{\boldsymbol{p}}(u_{\boldsymbol{p}}, v_{\boldsymbol{p}})}{\|u_{\boldsymbol{p}}\|_{g_{\boldsymbol{p}}} \|v_{\boldsymbol{p}}\|_{g_{\boldsymbol{p}}}} \quad .$$

- Two tangent vectors $u_{\boldsymbol{p}}, v_{\boldsymbol{p}} \in T_{\boldsymbol{p}}\mathcal{M}$ are called *orthogonal* if $g_{\boldsymbol{p}}(u_{\boldsymbol{p}}, v_{\boldsymbol{p}}) = 0$ i.e. if either one of them is zero or the angle between them is $\pi/2$.
- Two tangent vectors $u_{\boldsymbol{p}}, v_{\boldsymbol{p}} \in T_{\boldsymbol{p}}\mathcal{M}$ are called *orthonormal* if they are orthogonal and have length 1.

The notion of distance between points of a manifold is also introduced thanks to the Riemannian metric and the notion of curve along the manifold. A *parametrized curve* (resp. *smooth curve*) γ of $\mathcal{M}$ is a map from an open interval $I \subset \mathbb{R}$ to $\mathcal{M}$ that is continuous (resp. smooth). This means that for any $t_0 \in I$, the function $t \in]t_0 - \epsilon, t_0 + \epsilon[\mapsto x \circ \gamma(t)$, defined for a chart (U, x) containing $\gamma(t)$ and a small enough $\epsilon > 0$, is continuous (resp. smooth) at $t = t_0$.

Let $[a, b] \subset \mathbb{R}$ be a segment of $\mathbb{R}$. A map $\gamma : [a, b] \to \mathcal{M}$ is called a *curve segment* from $\gamma(a) = \boldsymbol{p}_1 \in \mathcal{M}$ to $\gamma(b) = \boldsymbol{p}_2 \in \mathcal{M}$ if, for some $\epsilon > 0$, there exists a parametrized curve $\tilde{\gamma} :]a - \epsilon, b + \epsilon[\to \mathcal{M}$ that agrees with γ on $[a, b]$. In particular, γ is called *smooth curve segment* if $\tilde{\gamma}$ is smooth, and *piecewise smooth curve segment* if there exits a subdivision of $[a, b]$, denoted $t_0 = a \leq t_1 \leq \cdots \leq t_N \leq t_{N+1} = b$, for which the restriction of γ to any segment $[t_k, t_{k+1}]$ is a smooth curve segment (from $\gamma(t_k)$ to $\gamma(t_{k+1})$).

Then the *length of a (piecewise) smooth curve segment* of $\mathcal{M}$, parametrized by $\gamma : [a, b] \to \mathcal{M}$ is defined from the Riemannian metric g of $\mathcal{M}$ as

$$L_g(\gamma) = \int_a^b \|\gamma'(t)\|_{g_{\gamma(t)}} dt \quad ,$$

where $\gamma'(t) \in T_{\gamma(t)}\mathcal{M}$ is the tangent vector defined as:

$$\forall f \in \mathscr{C}^\infty(\mathcal{M}), \quad \gamma'(t)(f) = \frac{d(f \circ \gamma)}{dt}(t) \quad .$$

This quantity is actually independent from the parametrization γ of the curve, i.e. $\forall \psi : [c, d] \to [a, b]$ diffeomorphism, $L_g(\gamma) = L_g(\gamma \circ \psi)$.

The *distance between two points* $\boldsymbol{p}_1, \boldsymbol{p}_2 \in \mathcal{M}$ is finally defined as the infimum of the length of piecewise smooth curve segments γ between $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$:

$$d_g(\boldsymbol{p}_1, \boldsymbol{p}_2) = \inf_{\substack{\gamma:[a,b]\to\mathcal{M} \text{ piecewise smooth} \\ \gamma(a)=\boldsymbol{p}_1, \gamma(b)=\boldsymbol{p}_2}} L_g(\gamma), \quad \boldsymbol{p}_1, \boldsymbol{p}_2 \in \mathcal{M} \quad . \tag{6.6}$$

In particular, a Riemannian manifold $(\mathcal{M}, g)$ is a metric space with respect to the Riemannian distance function d_g, and the topology induced by this distance function is the same as the original topology of $\mathcal{M}$. This means that open sets $U \subset \mathcal{M}$ defined in the original topology of $\mathcal{M}$ are also open sets in the topology induced by d_g, i.e. sets such that

$$\forall \boldsymbol{p} \in U, \quad \exists \epsilon > 0 \text{ such that } \forall \boldsymbol{q} \in \mathcal{M} : \quad d_g(\boldsymbol{p}, \boldsymbol{q}) < \epsilon \Rightarrow \boldsymbol{q} \in U \quad .$$

In other words, for any point of U there exists a (small enough) ball around that point that is fully contained in U, where the notion of ball is defined through d_g.

As a metric space, the notions of boundedness and completeness can be extended to a Riemannian manifold $(\mathcal{M}, g)$. Any $B \subset \mathcal{M}$ is *bounded* if $\exists C \geq 0 \ \forall \boldsymbol{p}_1, \boldsymbol{p}_2 \in B$, $d_g(\boldsymbol{p}_1, \boldsymbol{p}_2) \leq C$. $(\mathcal{M}, g)$ is called *complete* if the metric space $(\mathcal{M}, d_g)$ is complete, i.e. any Cauchy sequence of $\mathcal{M}$ converges in $\mathcal{M}$. Hence if $(\mathcal{M}, g)$ is complete and $(\boldsymbol{p}_k)_{k\in\mathbb{N}}$ is a sequence of points of $\mathcal{M}$ such that:

$$\forall \epsilon > 0, \exists N \in \mathbb{N} \text{ such that } \forall m, n \in \mathbb{N} : \quad m \geq n \geq N \Rightarrow d_g(\boldsymbol{p}_m, \boldsymbol{p}_n) < \epsilon$$

Then $(\boldsymbol{p}_k)_{k\in\mathbb{N}}$ converges and its limit is a point of $\mathcal{M}$.

6.2.3 Geodesics

A *geodesic* on $\mathcal{M}$ is a smooth curve $\gamma : [a, b] \to \mathcal{M}$ that minimizes the energy functional E_g defined as

$$E_g(\gamma) = \frac{1}{2} \int_a^b \|\gamma'(t)\|^2_{g_{\gamma(t)}} dt \quad .$$

The expression of E_g has the following physical interpretation. Consider a particle of unit mass moving freely on $\mathcal{M}$ and whose position at a time t is given by $\gamma(t)$. To obtain the equation of motion of this particle, the principle of least action can be applied. It consists in finding the trajectory γ that minimizes the integral of the Lagrangian of the system, which in the case of a free particle is reduced to its instantaneous kinetic energy $1/2\|\gamma'(t)\|^2_{\gamma(t)}$. Hence, as defined, the geodesic γ represents the trajectory of a particle moving freely on the manifold from $\gamma(a)$ to $\gamma(b)$.

The existence of geodesics between points sharing a chart is a consequence of the fact that this minimization problem can be turned into a second order differential equation through the Euler–Lagrange equations of functionals, as one would do in physics. This underlines the locality of geodesics, that are not necessarily defined for any pair of points on the manifold.

Defined as such, geodesics have two noticeable properties. First, they have a constant velocity, meaning that if $\gamma : [a, b] \to \mathcal{M}$ is a geodesic, there exists a constant c such that $\forall t \in [a, b]$, $\|\gamma'(t)\|_{\gamma(t)} = c$. Consequently, geodesics are parametrized by their length:

$$\gamma : [a, b] \to \mathcal{M} \text{ geodesic } \Rightarrow \forall t \in [a, b] : L_g(\gamma|_{[a,t]}) = c(t - a) \quad .$$

This property explains why geodesics on manifolds are sometimes referred as the generalization of Euclidean “straight lines".

Second, geodesics locally minimize the distance between points along them:

$$\gamma : [a,b] \to \mathcal{M} \text{ geodesic } \Rightarrow \forall t_1, t_2 \in [a,b] : L_g(\gamma|_{[t_1,t_2]}) = d_g(\gamma(t_1), \gamma(t_2)) \quad ,$$

where d_g denotes the distance defined in Equation (6.6). Hence geodesics locally defined paths of minimal length on the manifold.

A theorem (Jost, 2008, Theorem 1.4.3) states that for any point $\boldsymbol{p}$ of a Riemannian manifold, there exist a diffeomorphism, called *exponential map* of $\mathcal{M}$ at $\boldsymbol{p}$ that maps tangent vectors of $T_{\boldsymbol{p}}\mathcal{M}$ of length less than some $\epsilon > 0$ to an open neighborhood of $\boldsymbol{p}$ of size less than ϵ. Formally, the exponential map $\exp_{\boldsymbol{p}}$ yields a one-to-one correspondence between tangent vectors $u_{\boldsymbol{p}} \in T_{\boldsymbol{p}}\mathcal{M}$ such that $\|u_{\boldsymbol{p}}\|_{g_{\boldsymbol{p}}} < \epsilon$ and points $\boldsymbol{q} \in \mathcal{M}$ such that $d_g(\boldsymbol{p}, \boldsymbol{q}) < \epsilon$. In particular, $\exp_{\boldsymbol{p}}(u_{\boldsymbol{p}})$ is given as the endpoint of the geodesic of length $\|u_{\boldsymbol{p}}\|_{g_{\boldsymbol{p}}}$ that starts at $\boldsymbol{p}$ in the direction $\boldsymbol{u}_{\boldsymbol{p}}$. Hence small vectors in the tangent space of a point $\boldsymbol{p}$ can be seen, through the exponential map, as small displacements from a $\boldsymbol{p}$ along the geodesics of the manifold.

6.3 Integration on Riemannian manifolds

As we saw in the previous section, endowing a smooth manifold with a Riemannian metric allows to introduce geometric concepts on it, namely lengths, angles and distances. In this section, integration theory on a manifold is defined using once again a Riemannian metric. In particular, as we may see, a volume element can be introduced on manifolds, that corresponds locally to the deformation of the Euclidean volume element induced by the metric. Integrals of real functions defined on the manifold are then defined by “gluing” together integrals defined using this volume measure on subsets covering the manifold.

6.3.1 Integrals on a Riemannian manifold

Let $(\mathcal{M}, g)$ be a Riemannian manifold. Let $A \subset \mathcal{M}$ be an open subset of $\mathcal{M}$. A function $f : A \to \mathbb{R}$ is called *measurable* on A if for any chart (U, x) of $\mathcal{M}$ containing A the map $f \circ x^{-1} : x(A) \subset \mathbb{R}^d \to \mathbb{R}$ is measurable, i.e. if the preimage of any Borel set of $\mathbb{R}$ is a Borel set of $x(A)$. In this case, the *integral of f over the open subset $A \subset \mathcal{M}$* is denoted[1] $\int_A f dV_g$ and is defined as the following Lebesgue integral over $x(A)$:

$$\begin{aligned} \int_A f dV_g :&= \int_{x(A)} \left(f \circ x^{-1}(\boldsymbol{x})\right) \cdot \left(|\boldsymbol{G}^x|^{1/2}(x^{-1}(\boldsymbol{x}))\right) d\boldsymbol{x} \\ &= \int_{x(A)} \left(f \cdot |\boldsymbol{G}^x|^{1/2}\right) \circ x^{-1}(\boldsymbol{x}) d\boldsymbol{x} \quad , \end{aligned} \tag{6.7}$$

where $|\boldsymbol{G}^x|^{1/2}$ is the smooth function that maps any point of $\boldsymbol{p} \in U$ to the square-root of the determinant of $\boldsymbol{G}^x(\boldsymbol{p})$, the representative matrix of g at $\boldsymbol{p}$ with respect to the chart (U, x) as defined in Equation (6.4).

This quantity is independent from the choice of chart containing A. Indeed, if (U', y) denotes another chart containing A, the change of coordinates formula of integrals on $\mathbb{R}^d$ yields

$$\begin{aligned} \int_A f dV_g &= \int_{(x \circ y^{-1}) \circ y(A)} \left(f \cdot |\boldsymbol{G}^x|^{1/2}\right) \circ y^{-1} \circ (y \circ x^{-1})(\boldsymbol{x}) d\boldsymbol{x} \\ &= \int_{y(A)} \left(f \cdot |\boldsymbol{G}^x|^{1/2}\right) \circ y^{-1}(\boldsymbol{y}) |J_{x \circ y^{-1}}(\boldsymbol{y})| d\boldsymbol{y} \\ &= \int_{x(A)} \left(f \circ y^{-1}(\boldsymbol{y})\right) \cdot \left(|J_{x \circ y^{-1}}(\boldsymbol{y})^T| \cdot |\boldsymbol{G}^x(y^{-1}(\boldsymbol{y}))| \cdot |J_{x \circ y^{-1}}(\boldsymbol{y})|\right)^{1/2} d\boldsymbol{y} \quad . \end{aligned}$$

Using the change of map formula of Equation (6.5), this last equation becomes

$$\int_A f dV_g = \int_{x(A)} \left(f \cdot |\boldsymbol{G}^x|^{1/2}\right) \circ x^{-1}(\boldsymbol{x}) d\boldsymbol{x} = \int_{y(A)} \left(f \cdot |\boldsymbol{G}^y|^{1/2}\right) \circ y^{-1}(\boldsymbol{y}) d\boldsymbol{y} \quad .$$

[1] For the moment, writing $\int_A f dV_g$ the integral of f over A should be purely taken as a notation. In the next subsection, this notation will be justified by interpreting the term V_g as a measure on the manifold, and dV_g as the corresponding volume element.

Hence, as defined in Equation (6.7), the integral over A is independent of the choice of a coordinate map over A.

The integral of a function f over a subset of a manifold can be seen as the integral of its coordinate representation $f \circ x^{-1}$ through a chart x mapping A, scaled by a smooth function $|\boldsymbol{G}^x|^{1/2}$ (independent of f) that corrects the volume element so that it takes into account the actual geometry of the manifold (as defined by its metric). Equivalently, it can also be seen as the integral of $f \circ x^{-1}$ over $x(A) \subset \mathbb{R}^d$ with respect to a positive measure with density $\boldsymbol{x} \mapsto |\boldsymbol{G}^x(x^{-1}(\boldsymbol{x}))|^{1/2}$ with respect to the Lebesgue measure.

To go from this local definition of integrals to integrals defined on the whole manifold $\mathcal{M}$, local integrals defined over subsets covering $\mathcal{M}$ are "glued" together using the notion of partition of unity, which is now defined. Let $\mathcal{A} = \left\{(U^{(\alpha)}, x^{(\alpha)}) : \alpha \in I\right\}$ denote an atlas of $\mathcal{M}$ (indexed by a set I). A *partition of unity subordinate to the atlas* $\mathcal{A}$ is a set of functions $\{\phi_\alpha : \mathcal{M} \to [0,1] : \alpha \in I\}$ (also indexed by I) such that:

- $\forall \alpha \in I$, $\operatorname{supp} \phi_\alpha \subset U^{(\alpha)}$, where $\operatorname{supp} \phi_\alpha$ is the support of ϕ_α, i.e. the closure of the set of all points $\boldsymbol{p} \in \mathcal{M}$ such that $\phi_\alpha(\boldsymbol{p}) \neq 0$. Note that consequently, ϕ_α is zero outside $U^{(\alpha)}$.
- $\forall \boldsymbol{p} \in \mathcal{M}$, $\phi_\alpha(\boldsymbol{p})$ is non-zero only for a finite number of indexes $\alpha \in I$.
- $\forall \boldsymbol{p} \in \mathcal{M}$, $\sum_{\alpha \in I} \phi_\alpha(\boldsymbol{p}) = 1$.

Then, the *integral of a function* $f : \mathcal{M} \to \mathbb{R}$ *over the manifold* $\mathcal{M}$ is denoted $\int_{\mathcal{M}} f dV_g$ and is defined as the sum over the covering open sets composing an atlas $\mathcal{A}$ of $\mathcal{M}$ of local integrals of f, weighted by a partition of unity:

$$\int_{\mathcal{M}} f dV_g = \sum_{\alpha \in I} \int_{U^{(\alpha)}} \phi_\alpha f dV_g \quad . \tag{6.8}$$

In particular, measurable functions on $\mathcal{M}$ are defined as functions that are measurable on any chart of $\mathcal{M}$, and therefore for which each integral in Equation (6.8) is defined. Note also that the definition of the integral of a function over $\mathcal{M}$ is actually independent from the choices of the atlas $\mathcal{A}$ and its subordinate partition of unity. This is due to the fact that the local integrals are chart-invariant and that each function composing the partition of unity is zero outside an open set of $\mathcal{M}$.

Hence, the integration of a function of $\mathcal{M}$ requires to choose an atlas and a subordinate partition of unity, which may become a tedious task. However, in some cases, integrals over a manifold can be expressed as usual Lebesgue integral over open sets of $\mathbb{R}^d$ and therefore be calculated explicitly by classical methods.

Example 6.3.1 (Integration over an open set)**.** Let us assume that the Riemannian manifold $(\mathcal{M}, g)$ is such that $\mathcal{M}$ is diffeomorphic to an open set $A \subset \mathbb{R}^d$ and denote by x this diffeomorphism. Then the set $\mathcal{A} = \{(A, x)\}$ is an atlas for $\mathcal{M}$ composed of a single chart. This situation is particularly desirable as the function mapping all points of A to 1 can be chosen as a partition of unity. Hence the integral of a function $f : \mathcal{M} \to \mathbb{R}$ over $\mathcal{M}$ reduces to an integral over $A \subset \mathbb{R}^d$:

$$\int_{\mathcal{M}} f dV_g = \int_A f dV_g \quad ,$$

which in turn is computed using Equation (6.7).

This case arises when $\mathcal{M}$ is itself an open subset of $\mathbb{R}^d$. Then, x can be chosen to be the identity map and the integral over $(\mathcal{M}, g)$ is given by

$$\int_{\mathcal{M}} f dV_g = \int_{\mathcal{M}} f(\boldsymbol{p}) \cdot |\boldsymbol{G}(\boldsymbol{p})|^{1/2} d\boldsymbol{p} \quad (\mathcal{M} \subset \mathbb{R}^d \text{ open }) \quad , \tag{6.9}$$

where $\boldsymbol{G}$ is the representative matrix of the metric g with respect to the chart obtained by considering the identity map (cf. Equation (6.4)). Hence, the integral of f over the Riemannian manifold $(\mathcal{M}, g)$ is reduced to a "common" integral over a subset of $\mathbb{R}^d$ (which here is $\mathcal{M}$) of the function $\boldsymbol{p} \mapsto f(\boldsymbol{p}) \cdot |\boldsymbol{G}(\boldsymbol{p})|^{1/2}$.

6.3.2 Measure on a Riemannian manifold

The integration of a measurable function over a subset of a manifold is defined using the definition of the integral over the whole manifold. Indeed, the integral of $f : \mathcal{M} \to \mathbb{R}$ over any subset $M \subset \mathcal{M}$ is denoted $\int_M f dV_g$ and is given by

$$\int_M f dV_g = \int_{\mathcal{M}} (\mathbb{1}_M \cdot f) dV_g \quad ,$$

where $\mathbb{1}_M : \mathcal{M} \to \mathbb{R}$ is the indicator function of the subset M. Similarly, a measure V_g can be defined over subsets $M \subset \mathcal{M}$, as

$$V_g(M) = \int_{\mathcal{M}} \mathbb{1}_M dV_g = \int_M dV_g \quad .$$

It is straightforward to check that V_g is well-defined as a positive measure over $\mathcal{M}$. It is called the *canonical measure associated to the Riemannian manifold* $(\mathcal{M}, g)$. In particular,

$$V_g(M) = \sum_{\alpha \in I} \int_{x^{(\alpha)}(U^{(\alpha)} \cap M)} \left(\phi_\alpha \cdot |\boldsymbol{G}^{x^{(\alpha)}}|^{1/2}\right) \circ \left(x^{(\alpha)}\right)^{-1} (\boldsymbol{x}) d\boldsymbol{x} \quad .$$

$M \subset \mathcal{M}$ is called a *null set of* $\mathcal{M}$ whenever $V_g(M) = 0$. This is equivalent to imposing that for any chart (U, x) of $\mathcal{M}$, the set $x(U \cap M)$ is a null set for the measure of $\mathbb{R}^d$ with density $\boldsymbol{x} \mapsto |\boldsymbol{G}^x(x^{-1}(\boldsymbol{x}))|^{1/2}$ with respect to the Lebesgue measure. In particular, for any null set M and for any measurable function $f : \mathcal{M} \to \mathbb{R}$:

$$M \text{ null set} \quad \Rightarrow \quad \int_M f dV_g = 0 \quad \text{and} \quad \int_{\mathcal{M}} f dV_g = \int_{\mathcal{M} \setminus M} f dV_g \quad .$$

In practice, this last property can be used to compute integrals over manifolds, as they can be reduced to a more easy to compute integral over a subset of the manifold by removing null sets from the manifold. This is illustrated in the next example.

Example 6.3.2 (Integration on a sphere)**.** Let us assume that the Riemannian manifold $(\mathcal{M}, g)$ is such that $\mathcal{M}$ is the sphere $\mathbb{S}^2 \subset \mathbb{R}^3$. Contrary to the previous example, $\mathcal{M}$ cannot be covered entirely with a single chart. However, a chart covering $\mathbb{S}^2$ except for "negligible" parts can easily be built, so that carrying out the integration over $\mathbb{S}^2$ without these parts is the same as carrying out the integration over $\mathbb{S}^2$ entirely. Indeed, the map

$$\begin{array}{rccc} \phi : &]-\pi, \pi[\times]-\frac{\pi}{2}, \frac{\pi}{2}[& \to & \mathbb{S}^2 \setminus \{(0,0,1); (0,0,-1)\} \\ & (\theta, \xi) & \mapsto & (\cos\theta \cos\xi, \sin\theta \cos\xi, \sin\xi) \end{array} ,$$

defines a diffeomorphism from an open set of $\mathbb{R}^2$ to the unit-sphere minus two poles. These poles form a null set as their images by any coordinate chart will be isolated points in $\mathbb{R}^3$ which are null sets for the Lebesgue measure. Hence, integration over $(\mathbb{S}^2, g)$ is given by:

$$\int_{\mathbb{S}^2} f dV_g = \int_{]-\pi,\pi[} \int_{]-\frac{\pi}{2},\frac{\pi}{2}[} f \circ \phi(\theta, \xi) \sqrt{|\boldsymbol{G}^\phi(\phi(\theta, \xi))|} d\xi d\theta \tag{6.10}$$

where G^ϕ is the representative matrix of the metric g with respect to the chart obtained from ϕ (cf. Equation (6.4)).

6.3.3 Integrability on a Riemannian manifold

We assume in this section that $\mathcal{M}$ is a compact manifold.

A function $f : \mathcal{M} \to \mathbb{R}$ is called *integrable* if $\int_{\mathcal{M}} |f| dV_g < \infty$ and *square-integrable* if $|f|^2$ is integrable. Let $\overset{L^2(\mathcal{M})}{\sim}$ be the binary relation defined over the set of square-integrable functions by

$$f_1 \overset{L^2(\mathcal{M})}{\sim} f_2 \Leftrightarrow \int_{\mathcal{M}} (f_1 - f_2)^2 dV_g = 0, \quad f_1, f_2 \text{ square-integrable} . \tag{6.11}$$

In particular, $\overset{L^2(\mathcal{M})}{\sim}$ is an equivalence relation over the set of square-integrable functions of $\mathcal{M}$, and the set of equivalence classes under $\overset{L^2(\mathcal{M})}{\sim}$ is denoted by $L^2(\mathcal{M})$.

Hence, any element of $L^2(\mathcal{M})$ actually corresponds to a set of square integrable functions such that any pair of them satisfies Equation (6.11). However, using a common abuse of notation, elements of $L^2(\mathcal{M})$ will also be called square-integrable functions of $L^2(\mathcal{M})$ and we will write

$$L^2(\mathcal{M}) = \left\{ f : \mathcal{M} \to \mathbb{R} \text{ measurable } : \int_{\mathcal{M}} f^2 dV_g < \infty \right\} \quad .$$

Hence the equivalence classes defining $L^2(\mathcal{M})$ are identified with the functions composing these classes.

$L^2(\mathcal{M})$ can be equipped with the inner-product $\langle \cdot, \cdot \rangle_{L^2(\mathcal{M})}$ defined by

$$\langle f_1, f_2 \rangle_{L^2(\mathcal{M})} = \int_{\mathcal{M}} f_1 f_2 dV_g, \quad f_1, f_2 \in L^2(\mathcal{M}) \quad , \tag{6.12}$$

with associated norm $\| \cdot \|_{L^2(\mathcal{M})}$ given by

$$\|f\|_{L^2(\mathcal{M})} = \sqrt{\langle f, f \rangle_{L^2(\mathcal{M})}}, \quad f_2 \in L^2(\mathcal{M}) \quad . \tag{6.13}$$

$L^2(\mathcal{M})$ then defines a Hilbert space (Craioveanu et al., 2013).

Remark 6.3.1. The set $L^2(\mathcal{M})$ can equivalently be defined as the completion via Cauchy sequences with respect to the norm $\|.\|_{L^2(\mathcal{M})}$ of the set of smooth functions with compact support over $\mathcal{M}$.

6.4 Manifolds with boundary

Manifolds with boundary are a generalization of manifolds as defined in the previous sections, and called here ordinary manifolds. They allow to extend the notion of edge (or border) to manifolds.

6.4.1 Definitions and first properties

Formally, the definition of a *manifold with boundary* is the same as the definition of an ordinary manifold, except that it is now required that a neighborhood of any point of the manifold be homeomorphic to either an open subset of $\mathbb{R}^d$ or an open subset of $\mathbb{H}^d = \mathbb{R}^{d-1} \times \mathbb{R}_+$. In particular, open subsets of $\mathbb{H}^d$ are defined as the intersection of open sets of $\mathbb{R}^d$ with $\mathbb{H}^d$.

Hence, a coordinate chart (U, x) of a d-manifold with boundary $\mathcal{M}$ is either

- a *regular chart*, i.e. x is a homeomorphism from $U \subset \mathcal{M}$ to an open subset of $\mathbb{R}^d$. Then $x(U)$ is open set of $\mathbb{R}^d$ that is homeomorphic to an open subset U of $\mathcal{M}$,
- or a *boundary chart*, i.e. x is a homeomorphism from $U \subset \mathcal{M}$ to an open subset of $\mathbb{H}^d$ which means that

  $$\forall \boldsymbol{p} \in U, \quad x(\boldsymbol{p}) = (x_1(\boldsymbol{p}), \ldots, x_d(\boldsymbol{p})) \in \mathbb{R}^d \text{ and } x_d(\boldsymbol{p}) \geq 0 \quad .$$

  Then $x(U)$ is the intersection of an open set of $\mathbb{R}^d$ with $\mathbb{H}^d$.

Then, a point $\boldsymbol{p} \in \mathcal{M}$ is called an *interior point* if there exists a regular chart that contains $\boldsymbol{p}$. Otherwise, $\boldsymbol{p}$ is called a *boundary point*: in this case, if (U, x) is a boundary chart containing $\boldsymbol{p}$, then $x_d(\boldsymbol{p}) = 0$.

The set $\text{Int}(\mathcal{M})$ of all interior points of $\mathcal{M}$ is called the *interior of* $\mathcal{M}$ and the set $\partial\mathcal{M}$ of all boundary points of $\mathcal{M}$ is called the *boundary of* $\mathcal{M}$. Basically, for a boundary point $\boldsymbol{p} \in \partial\mathcal{M}$, we see that even an infinitesimal perturbations of its coordinates $x(\boldsymbol{p})$ can push us off the "edge" of the manifold: indeed, as soon as the d-th component of the perturbed coordinates is strictly negative, its preimage by x will not fall into $\mathcal{M}$.

For a d-manifold with boundary $\mathcal{M}$, we have:

$$\mathcal{M} = \text{Int}(\mathcal{M}) \cup \partial\mathcal{M} \quad .$$

Ordinary manifolds are the particular case of manifolds with an empty boundary: that is why they are also called manifolds without boundary. More generally, $\text{Int}(\mathcal{M})$ is an ordinary d-manifold and $\partial\mathcal{M}$ is an ordinary $(d-1)$-manifold.

The other definitions introduced for ordinary manifolds still hold for manifolds with boundary, as long as requirements on charts account for both regular and boundary charts, i.e. $\mathbb{R}^d$ can be replaced by $\mathbb{H}^d$ as the mapping destination of coordinates charts. This is how notions like smoothness of manifolds and maps or tangent spaces are naturally extended to manifolds with boundaries.

Some particular points concerning tangent spaces should be noted. Let $\mathcal{M}$ denote a manifold with boundary. On one hand, if $\boldsymbol{p} \in \text{Int}(\mathcal{M})$ then $\boldsymbol{p}$ can basically be seen as point of the ordinary manifold $\text{Int}(\mathcal{M})$ and $T_{\boldsymbol{p}}\mathcal{M} = T_{\boldsymbol{p}}\text{Int}(\mathcal{M})$. On the other hand, if $\boldsymbol{p} \in \partial\mathcal{M}$ then two cases arise:

- either $\boldsymbol{p}$ is seen as a point of the d-manifold with boundary $\mathcal{M}$ and then its tangent space $T_{\boldsymbol{p}}\mathcal{M}$ is also a d-dimensional vector space spanned by directional derivatives along coordinate charts.
- or $\boldsymbol{p}$ is seen as a point of the $(d-1)$-manifold without boundary $\partial\mathcal{M}$ and then its tangent space $T_{\boldsymbol{p}}\partial\mathcal{M}$ can be seen as a restriction of $T_{\boldsymbol{p}}\mathcal{M}$. Indeed, let (U, x) be a boundary chart containing $\boldsymbol{p}$, such that $x_d(\boldsymbol{p}) = 0$. Then, $T_{\boldsymbol{p}}\partial\mathcal{M}$ is spanned by $\{\partial/\partial x_1, \ldots, \partial/\partial x_{d-1}\}$.

Note in particular that $T_{\boldsymbol{p}}\partial\mathcal{M}$ is a vector subspace of dimension $d-1$ of $T_{\boldsymbol{p}}\mathcal{M}$, which is a vector space of dimension d.

Manifolds defined by an implicit function are a particular case of manifold with boundary. This is formalized in the next proposition.

Proposition 6.4.1. *Let $F : \mathbb{R}^d \mapsto \mathbb{R}$ be a smooth function of $\mathbb{R}^d$ such that*

$$\{\boldsymbol{p} \in \mathbb{R}^d : F(\boldsymbol{p}) = 0\} \neq \emptyset$$

and such that

$$\forall \boldsymbol{p} \in \mathbb{R}^d, \quad F(\boldsymbol{p}) = 0 \Rightarrow \nabla F(\boldsymbol{p}) \neq \mathbf{0} \quad ,$$

where $\nabla F(\boldsymbol{p}) = (\partial_1 F(\boldsymbol{p}), \ldots, \partial_k F(\boldsymbol{p}))^T$ is the usual gradient of a function of $\mathbb{R}^d$ (with respect to the Cartesian coordinates).

Then, the set

$$\mathcal{M} = \{\boldsymbol{p} \in \mathbb{R}^d : F(\boldsymbol{p}) \leq 0\}$$

is a d-manifold with boundary such that

- *its interior is $\text{Int}\,\mathcal{M} = \{\boldsymbol{p} \in \mathbb{R}^d : F(\boldsymbol{p}) < 0\}$, which is a d-manifold without boundary;*
- *its boundary is $\partial\mathcal{M} = \{\boldsymbol{p} \in \mathbb{R}^d : F(\boldsymbol{p}) = 0\}$, which is a $(d-1)$-manifold without boundary;*
- *both $\text{Int}\,\mathcal{M}$ and $\partial\mathcal{M}$ are submanifolds of $\mathbb{R}^d$.*

Proof. Let $\mathcal{M}_1 = \{\boldsymbol{p} \in \mathbb{R}^d : F(\boldsymbol{p}) < 0\}$ and $\mathcal{M}_2 = \{\boldsymbol{p} \in \mathbb{R}^d : F(\boldsymbol{p}) = 0\}$. Clearly, $\mathcal{M} = \mathcal{M}_1 \cup \mathcal{M}_2$.

First, the smoothness of F is used to prove that $\mathcal{M}_1$ is an ordinary d-manifold, with the usual topology of the Euclidean space $\mathbb{R}^d$. Indeed, note that given that $\mathcal{M}_1$ is the preimage by F of the open set $]-\infty, 0[$ of $\mathbb{R}$, by continuity of F, F is an open set of $\mathbb{R}^d$. And so as such, it defines an (ordinary) d-submanifold of $\mathbb{R}^d$.

Then, let $\boldsymbol{p} \in \mathcal{M}_2$ and let us assume, without loss of generality, that $\partial_d F(\boldsymbol{p}) \neq 0$. The implicit function theorem (Wilfred, 2002, Section 2.10) states that, as long as $\partial_d F(\boldsymbol{p}) \neq 0$, there exists an open set U of $\mathbb{R}^{d-1}$ containing $(p_1, \ldots, p_{d-1})$ and a unique (smooth) map $\phi : U \to \mathbb{R}$ such that

$$\phi(p_1, \ldots, p_{d-1}) = p_d \quad \text{and} \quad \forall \boldsymbol{x} \in U, \quad F(\boldsymbol{x}, \phi(\boldsymbol{x})) = 0 \quad .$$

Note that in particular $\forall \boldsymbol{x} \in U$, $(\boldsymbol{x}, \phi(\boldsymbol{x})) \in \mathcal{M}_2$. Hence, ϕ defines a coordinate chart between an open set of $\mathcal{M}_2$ around $\boldsymbol{p}$ and an open set $\mathbb{R}^{d-1}$. $\mathcal{M}_2$ is therefore a $(d-1)$-submanifold of $\mathbb{R}^d$, and in particular an ordinary $(d-1)$-manifold.

The set $\mathcal{M} = \mathcal{M}_1 \cup \mathcal{M}_2$ then defines a manifold with boundary, with interior $\mathcal{M}_1$ and boundary $\mathcal{M}_2$.

□

Example 6.4.1. The unit ball $\mathbb{B}^3$ of $\mathbb{R}^3$ is defined as the set of points

$$\mathbb{B}^3 = \{(x, y, z) \in \mathbb{R}^3 : x^2 + y^2 + z^2 \leq 1\} \quad .$$

By denoting $F : (x, y, z) \in \mathbb{R}^3 \mapsto x^2+y^2+z^2-1$, we have $\mathbb{B}^3 = \{(x, y, z) \in \mathbb{R}^3 : F(x, y, z) \leq 0\}$. In particular, F is a smooth function of $\mathbb{R}^3$ and satisfies $\nabla F(x, y, z) = (2x, 2y, 2z)^T$. Hence, $\mathbb{B}^3$ is a 3-manifold with boundary and its boundary, given by $\{(x, y, z) \in \mathbb{R}^3 : F(x, y, z) = 0\}$, is the 2-sphere $\mathbb{S}^2$.

This result actually still holds for other dimensions: the unit-ball $\mathbb{B}^d$ of $\mathbb{R}^d$ is a d-manifold with boundary and satisfies $\partial\mathbb{B}^d = \mathbb{S}^{d-1}$.

Remark 6.4.1. It should be noted that the boundary ∂M of a manifold with boundary $\mathcal{M}$ generally differs from the boundary of $\mathcal{M}$ seen as a (subset of a) topological space. To distinguish both notions we call $\partial\mathcal{M}$ the manifold boundary of $\mathcal{M}$ and we call topological boundary the second kind of boundary. Both types of boundary are fundamentally different, and thus, $\mathcal{M}$ can have (or not) a manifold boundary regardless of the fact that it has (or not) a topological boundary.

To illustrate this point, consider the unit sphere $\mathbb{S}^2$ of $\mathbb{R}^3$. As we saw, it defines a manifold without boundary. However, seen as a subset of (the topological space) $\mathbb{R}^3$, its topological boundary is also $\mathbb{S}^2$ itself. Consider now the unit ball $\mathbb{B}^3$ of $\mathbb{R}^3$. As we saw, it defines a manifold with boundary, whose manifold boundary $\mathbb{S}^2 \subset \mathbb{R}^3$. Seen as a subset of (the topological space) $\mathbb{R}^3$, its topological boundary is also $\mathbb{S}^2 \subset \mathbb{R}^3$. But if now we see $\mathbb{B}^3$ as a subset of $\mathbb{R}^4$, its topological boundary becomes $\mathbb{B}^3$ itself.

6.4.2 Riemannian manifolds with boundary

A manifold with boundary can also be equipped with a Riemannian metric and then defines a *Riemannian manifold with boundary*. Indeed, the tangent spaces at any point of a manifold with boundary have the same dimension, and the notion of Riemannian metric on a manifold with boundary can then be naturally extended using the same definition as in the ordinary case.

Let then $(\mathcal{M}, g)$ denote a Riemannian manifold with boundary, and g its metric. Then the boundary $\partial\mathcal{M}$ of $\mathcal{M}$ can be endowed with its own metric, inherited from the metric of $\mathcal{M}$. Indeed, given that $\forall \boldsymbol{p} \in \partial\mathcal{M}$, $T_{\boldsymbol{p}}\partial\mathcal{M} \subset T_{\boldsymbol{p}}\mathcal{M}$ then the tensor field ∂g defined at any point $\boldsymbol{p} \in \partial\mathcal{M}$ by

$$\partial g_{\boldsymbol{p}} : (u_{\boldsymbol{p}}, v_{\boldsymbol{p}}) \in T_{\boldsymbol{p}}\partial\mathcal{M} \times T_{\boldsymbol{p}}\partial\mathcal{M} \mapsto \partial g_{\boldsymbol{p}}(u_{\boldsymbol{p}}, v_{\boldsymbol{p}}) = g_{\boldsymbol{p}}(u_{\boldsymbol{p}}, v_{\boldsymbol{p}})$$

defines a Riemannian metric on $\partial\mathcal{M}$. Hence, $(\partial\mathcal{M}, \partial g)$ is a Riemannian ordinary $(d-1)$-manifold.

Integrating a function over a smooth d-manifold with boundary $\mathcal{M}$ that is equipped with a Riemannian metric g is actually equivalent to integrating the same function over the interior $\mathrm{Int}(\mathcal{M})$ seen as a Riemannian manifold also equipped with the metric g. Indeed, by definition of the boundary of a manifold, the image of a point $\boldsymbol{p} \in \partial\mathcal{M}$ will always lie in the boundary of the domain of integration in the right side of Equation (6.7), and can therefore be discarded.

On the other hand, integration can be defined over just the boundary $\partial\mathcal{M}$ of a smooth d-manifold with boundary $\mathcal{M}$. In this case, $\partial\mathcal{M}$ is seen a smooth $(d-1)$-manifold equipped with the Riemannian metric ∂g and we denote dS_g the volume element of $\partial\mathcal{M}$ associated with ∂g: $dS_g = dV_{\partial g}$.

Both types of integrals intervene in Green's theorem, which will be stated in Section 6.5.1, and which plays a key role in the theory of analysis of functions on Riemannian manifolds.

6.4.3 Normal vector at the boundary

Let $(\mathcal{M}, g)$ be a Riemannian manifold with boundary. The orthogonal subspace of $T_{\boldsymbol{p}}\partial\mathcal{M}$ in $T_{\boldsymbol{p}}\mathcal{M}$ is defined by

$$T_{\boldsymbol{p}}\partial\mathcal{M}^{\perp} = \{u_{\boldsymbol{p}} \in T_{\boldsymbol{p}}\mathcal{M} : \quad \forall v_{\boldsymbol{p}} \in T_{\boldsymbol{p}}\partial\mathcal{M}, \quad g_{\boldsymbol{p}}(u_{\boldsymbol{p}}, v_{\boldsymbol{p}}) = 0\} \quad .$$

In particular, $T_{\boldsymbol{p}}\partial\mathcal{M}$ and $T_{\boldsymbol{p}}\partial\mathcal{M}^{\perp}$ are in direct sum, meaning tangent vectors in $T_{\boldsymbol{p}}\partial\mathcal{M}$ can be uniquely decomposed as the sum of an element of $T_{\boldsymbol{p}}\partial\mathcal{M}$ and an element of $T_{\boldsymbol{p}}\partial\mathcal{M}^{\perp}$ and vice-versa:

$$T_{\boldsymbol{p}}\mathcal{M} = T_{\boldsymbol{p}}\partial\mathcal{M} \oplus T_{\boldsymbol{p}}\partial\mathcal{M}^{\perp} \quad .$$

Any vector of $T_{\boldsymbol{p}}\partial\mathcal{M}^{\perp}$ is called a normal vector of $\partial\mathcal{M}$ at $\boldsymbol{p}$.

Given that $T_{\boldsymbol{p}}\partial\mathcal{M}$ is a vector space of dimension $d-1$ and that $T_{\boldsymbol{p}}\mathcal{M}$ is a vector space of dimension d, $T_{\boldsymbol{p}}\partial\mathcal{M}^{\perp}$ is a vector space of dimension 1. It is therefore spanned by any non-zero element it contains. Let then $n_{\boldsymbol{p}} \in T_{\boldsymbol{p}}\mathcal{M}^{\perp}$ be the tangent vector such that $g_{\boldsymbol{p}}(n_{\boldsymbol{p}}, n_{\boldsymbol{p}}) = 1$ and $g_{\boldsymbol{p}}(n_{\boldsymbol{p}}, \partial/\partial x_d) < 0$ where (U, x) is a boundary chart containing $\boldsymbol{p}$ and such that $x_d(\boldsymbol{p}) = 0$. $\boldsymbol{n_p}$ is called *outward unit normal vector* of $\partial\mathcal{M}$ at $\boldsymbol{p}$ and satisfies

$$T_{\boldsymbol{p}}\partial\mathcal{M}^{\perp} = \operatorname{span}\{\boldsymbol{n_p}\} \quad .$$

It can be shown that $\boldsymbol{p} \mapsto n_{\boldsymbol{p}}$ is a well-defined continuous vector field over $\partial\mathcal{M}$, i.e. an application that maps each point of a manifold to one of its tangent vector.

Normal vectors of a manifold with boundary defined as in Proposition 6.4.1 can be easily deduced from the expression of their defining equation.

Proposition 6.4.2. *Let $(\mathcal{M}, g)$ be a Riemannian manifold with boundary defined through a smooth function $F : \mathbb{R}^d \to \mathbb{R}$ by:*

$$\mathcal{M} = \{\boldsymbol{p} \in \mathbb{R}^d : F(\boldsymbol{p}) \leq 0\} \quad ,$$

where $\boldsymbol{p} \mapsto \nabla F(\boldsymbol{p})$ is non-zero on $\partial\mathcal{M}$. Let us assume that $\mathcal{M}$ is equipped with the Euclidean metric $\bar{g}$.

Then $\forall \boldsymbol{p} \in \mathcal{M}$, the unit outward normal vector $n_{\boldsymbol{p}} \in T_{\boldsymbol{p}}\mathcal{M}$ at $\boldsymbol{p} \in \partial\mathcal{M}$ is represented in the basis of Cartesian directional derivatives $\{\partial_1|_{\boldsymbol{p}}, \ldots, \partial_d|_{\boldsymbol{p}}\}$ by the vector

$$\boldsymbol{n_p} = \frac{1}{\|\nabla F(\boldsymbol{p})\|_2} \nabla F(\boldsymbol{p}) \quad .$$

Proof. See Appendix C.4. □

Example 6.4.2. Following Proposition 6.4.2, the unit outward normal vector of the unit-ball $\mathbb{B}^d$ at one of its point $\boldsymbol{p} \in \mathbb{B}^d$ is given by

$$\boldsymbol{n_p} = \frac{\boldsymbol{v_p}}{\|\boldsymbol{v_p}\|_2}, \quad \text{where } \boldsymbol{v_p} = \nabla F(\boldsymbol{p}) = 2\boldsymbol{p} \quad .$$

6.4.4 Manifolds with corners

Geometric objects like rectangles, triangles, cubes or more generally polyhedrons of $\mathbb{R}^d$ often arise as spatial domains on which a phenomenon is studied. Clearly, such subsets of $\mathbb{R}^d$ are manifolds with boundary. However, they will not have a smooth structure due to the fact that they have "corners". That is why the notion of manifold with corners is introduced.

A *d-manifold with corner* is a d-manifold with boundary such that any of its coordinate charts (U, x) is either

- a regular chart,
- a boundary chart,

- or a *chart with corners*, i.e. x is a homeomorphism from U to an open subset[2] of $(\mathbb{R}_+)^d$, which means

$$\forall \boldsymbol{p} \in U, \quad x(\boldsymbol{p}) \in \{\boldsymbol{x} \in \mathbb{R}^d : x_1 \geq 0, \ldots, x_d \geq 0\} \quad .$$

As it is the case for manifolds with or without boundary, a manifold with corners is called *smooth* if it can be covered by smoothly compatible charts with corners (cf. Section 6.1.1).

Let us assume from now on that $\mathcal{M}$ is a smooth manifold with corners. If the image of $\boldsymbol{p} \in \mathcal{M}$ through a chart with corner (U, x) falls on one of the "edges" of $(\mathbb{R}_+)^d$, i.e. if $x(\boldsymbol{p})$ has more than one coordinate equal to zero, then $\boldsymbol{p}$ is called *corner point* of $\mathcal{M}$. In smooth manifolds with corners, this property is actually independent from the choice of chart. As a quick reminder, boundary points of $\mathcal{M}$ correspond to points of $\boldsymbol{p}$ for which exactly one coordinate vanishes. Hence the image through a coordinate chart of a corner point of $\mathcal{M}$ lies on one the edges of $(\mathbb{R}_+)^d$.

Once again, the notions introduced for smooth manifolds with or without boundary, such as smooth maps, partitions of unity, tangent vectors and Riemannian metrics, can be extended to smooth manifolds with corners by considering now smoothly compatible charts with corners.

Regarding the integration of a function over a (Riemannian) manifold with corners, the same definition as the one stated for manifold with boundary holds (cf. Section 6.4.2). Hence, if $\mathcal{M}$ is a smooth manifold with corners equipped with a metric g, then the integral of a function over $\mathcal{M}$ can be reduced to the integral of the same function over $\mathrm{Int}(\mathcal{M})$ (also equipped with g).

And to integrate a function over the boundary $\partial\mathcal{M}$ of $\mathcal{M}$, one "chops up" the integral over $\partial\mathcal{M}$ into integrals over subsets of $\partial\mathcal{M}$ that can be considered as ordinary $(d-1)$-manifolds or d-manifolds with boundary, equipped with the metric ∂g. In particular, the boundary points of $\mathcal{M}$ will lie in the boundaries of these chopped up pieces, and will effectively be discarded in the integration process.

The results that will be presented in the remainder of this chapter and in the subsequent ones rely on the so-called spectral theory of Riemannian manifolds. This branch of differential geometry aims at deriving tools to work with functions defined over a Riemannian manifold using their decomposition as a sum of *fixed* smooth functions satisfying a differential equation (called eigenvalue problem). The next section aims at introducing these concepts.

Remark 6.4.2. In the remainder of this work, (smooth) manifolds with corners will be identified with (smooth) manifolds with boundary. Indeed, the results of spectral theory that will be used rely on boundary conditions being assumed on the considered functions, so that their integral over the boundary is always discarded. Consequently, the presence of corners on the boundary will have no effect on the derived results.

6.5 Differential operators

The gradient and the Laplacian of functions defined over a Riemannian manifold are now introduced. The central piece of this section is the spectral theorem, which provides a decomposition of any square-integrable function defined on a compact Riemannian manifold. This decomposition will later be used to define (generalized) random fields on a Riemannian manifold, which can be considered for now as a randomized version of the notion of distribution that will also be introduced in this section.

In the remainder of this section, $(\mathcal{M}, g)$ denotes a Riemannian manifold with or without boundary and $\mathcal{C}^\infty(\mathcal{M})$ is the set of smooth functions of $\mathcal{M}$.

6.5.1 Gradient, Laplacian and Green's theorem

Let $f \in \mathscr{C}^\infty(\mathcal{M})$. The *gradient of f on $\mathcal{M}$* is the application $\nabla_{\mathcal{M}} f : \mathcal{M} \mapsto T\mathcal{M}$ such that $\forall \boldsymbol{p} \in \mathcal{M}$, $\nabla_{\mathcal{M}} f(\boldsymbol{p}) \in T_{\boldsymbol{p}}\mathcal{M}$ and

$$\forall u_{\boldsymbol{p}} \in T_{\boldsymbol{p}}\mathcal{M}, \quad g_{\boldsymbol{p}}(\nabla_{\mathcal{M}} f(\boldsymbol{p}), u_{\boldsymbol{p}}) = u_{\boldsymbol{p}}(f) \quad .$$

[2] for the trace topology.

In particular, $\nabla_{\mathcal{M}} f$ is a vector field. In local coordinates of a chart (U, x) of $\mathcal{M}$ the gradient is given by

$$\nabla_{\mathcal{M}} f(\boldsymbol{p}) = \sum_{i=1}^{d} \sum_{j=1}^{d} [\boldsymbol{G}^x(\boldsymbol{p})^{-1}]_{ij} \frac{\partial f}{\partial x_j}(\boldsymbol{p}) \left. \frac{\partial}{\partial x_i} \right|_{\boldsymbol{p}}, \quad \boldsymbol{p} \in U \quad .$$

For $\boldsymbol{p} \in U$, the representative vector of $\nabla_{\mathcal{M}} f(\boldsymbol{p})$ with respect to the chart (U, x) is denoted by $\nabla_x f$ and is given by

$$\nabla_x f(\boldsymbol{p}) = \boldsymbol{G}^x(\boldsymbol{p})^{-1} \begin{pmatrix} \frac{\partial f}{\partial x_1}(\boldsymbol{p}) \\ \vdots \\ \frac{\partial f}{\partial x_d}(\boldsymbol{p}) \end{pmatrix} \in \mathbb{R}^d \quad .$$

The *Laplace–Beltrami operator*, simply called *Laplacian* here, is a generalization on Riemannian manifolds of the Laplace operator (or Laplacian) of smooth functions of $\mathbb{R}^d$. In local coordinates of a chart (U, x) of $\mathcal{M}$ the Laplacian of $f \in \mathscr{C}^\infty(\mathcal{M})$ is the smooth function $\Delta_{\mathcal{M}} f \in \mathscr{C}^\infty(\mathcal{M})$ defined by

$$\Delta_{\mathcal{M}} f(\boldsymbol{p}) = \frac{1}{\sqrt{|\boldsymbol{G}^x(\boldsymbol{p})|}} \sum_{i=1}^{d} \sum_{j=1}^{d} \frac{\partial}{\partial x_i} \left(\sqrt{|\boldsymbol{G}^x|} \left[(\boldsymbol{G}^x)^{-1} \right]_{ij} \frac{\partial f}{\partial x_j} \right) \Bigg|_{\boldsymbol{p}}, \quad \boldsymbol{p} \in U \quad .$$

Green's theorem holds for integration on Riemannian manifolds. However it is required of them that they are compact. A *compact manifold* is a manifold with possibly empty boundary which is compact as a topological space. In particular submanifolds (with or without boundary) of $\mathbb{R}^d$ that are topologically compact in $\mathbb{R}^d$ are compact manifolds.

We first introduce the following notations. Let $f_1, f_2 \in \mathscr{C}^\infty(\mathcal{M})$. We write

$$\int_{\mathcal{M}} g\left(\nabla_{\mathcal{M}} f_1, \nabla_{\mathcal{M}} f_2\right) dV_g := \int_{\mathcal{M}} \left(\boldsymbol{p} \mapsto g_{\boldsymbol{p}}\left(\nabla_{\mathcal{M}} f_1(\boldsymbol{p}), \nabla_{\mathcal{M}} f_2(\boldsymbol{p})\right)\right) dV_g$$

and

$$\int_{\partial \mathcal{M}} f_1 g(n, \nabla_{\mathcal{M}} f_2) dS_g := \int_{\partial \mathcal{M}} f_1 \cdot \left(\boldsymbol{p} \mapsto g_{\boldsymbol{p}}(n_{\boldsymbol{p}}, \nabla_{\mathcal{M}} f_2(\boldsymbol{p}))\right) dS_g \quad ,$$

where dS_g denotes the restriction of the measure dV_g of $\mathcal{M}$ on the boundary $\partial\mathcal{M}$ (cf. Section 6.4.2) and $n_{\boldsymbol{s}}$ denotes the unit outward normal vector at a point $\boldsymbol{p} \in \partial\mathcal{M}$.

Theorem 6.5.1 (Green's theorem)**.** *Let $(\mathcal{M}, g)$ be a compact connected Riemannian manifold with (or without) boundary and $f_1, f_2 \in \mathscr{C}^\infty(\mathcal{M})$.*
Then,

$$\int_{\mathcal{M}} f_1 \cdot \Delta_{\mathcal{M}} f_2 dV_g = -\int_{\mathcal{M}} g\left(\nabla_{\mathcal{M}} f_1, \nabla_{\mathcal{M}} f_2\right) dV_g + \int_{\partial \mathcal{M}} f_1 g(n, \nabla_{\mathcal{M}} f_2) dS_g \quad ,$$

where n denotes the vector field associating to each point $\boldsymbol{s} \in \partial\mathcal{M}$ its unit outward normal vector.

Proof. See (Lang, 2012, Theorem 3.4). □

This result still holds when $\mathcal{M}$ is not compact but either f_1 or f_2 is a compactly supported function of $\mathscr{C}^\infty(\mathcal{M})$ (Lang, 2012). Besides, there exist three cases for which Green's theorem simplifies and yields interesting results for functions of $L^2(\mathcal{M})$. These three cases are:

- *Closed condition*: $\mathcal{M}$ is a compact connected manifold without boundary.
- *Dirichlet boundary conditions*: $\mathcal{M}$ is a compact connected manifold with boundary $\partial\mathcal{M}$. $f \in \mathscr{C}^\infty(\mathcal{M})$ follows Dirichlet boundary conditions if

$$\forall \boldsymbol{p} \in \partial\mathcal{M}, \quad f(\boldsymbol{p}) = 0 \quad .$$

- *Neumann boundary conditions*: $\mathcal{M}$ is a compact connected manifold with boundary $\partial\mathcal{M}$. $f \in \mathscr{C}^\infty(\mathcal{M})$ follows Neumann boundary conditions if

$$\forall \boldsymbol{p} \in \partial\mathcal{M}, \quad g_{\boldsymbol{p}}(n_{\boldsymbol{p}}, \nabla_{\mathcal{M}} f(\boldsymbol{p})) = 0 \quad ,$$

  where $n_{\boldsymbol{p}}$ denotes the unit normal vector at a point $\boldsymbol{p} \in \partial\mathcal{M}$.

In either one of these cases, the following corollary of Green's theorem is valid.

Corollary 6.5.2. *Let $(\mathcal{M}, g)$ be a compact connected Riemannian manifold and let $f_1, f_2 \in \mathscr{C}^\infty(\mathcal{M})$. If either $\partial\mathcal{M} = \emptyset$ or $\partial\mathcal{M} \neq \emptyset$ and f_1, f_2 follow Dirichlet or Neumann boundary conditions then,*

$$\langle f_1, -\Delta_{\mathcal{M}} f_2 \rangle_{L^2(\mathcal{M})} = \langle \nabla_{\mathcal{M}} f_1, \nabla_{\mathcal{M}} f_2 \rangle_{L^2(\mathcal{M})} = \langle -\Delta_{\mathcal{M}} f_1, f_2 \rangle_{L^2(\mathcal{M})} \quad ,$$

where the notation $\langle \nabla_{\mathcal{M}} f_1, \nabla_{\mathcal{M}} f_2 \rangle_{L^2(\mathcal{M})}$ symbolizes the integral over $\mathcal{M}$ given by

$$\langle \nabla_{\mathcal{M}} f_1, \nabla_{\mathcal{M}} f_2 \rangle_{L^2(\mathcal{M})} = \int_{\mathcal{M}} g\left(\nabla_{\mathcal{M}} f_1, \nabla_{\mathcal{M}} f_2\right) dV_g \quad .$$

Proof. This is a direct consequence of the fact that, within the requirement of this corollary, the integral over $\partial\mathcal{M}$ that appears in Theorem 6.5.1 is zero. □

Remark 6.5.1. Note that if $\nabla_{\mathcal{M}} f_2(\boldsymbol{p})$ and $\nabla_{\mathcal{M}} f_2(\boldsymbol{p})$ have support in a coordinate chart (U, x) then

$$\langle \nabla_{\mathcal{M}} f_1, \nabla_{\mathcal{M}} f_2 \rangle_{L^2(\mathcal{M})} = \int_U \nabla_x f_1^T \boldsymbol{G}^x \nabla_x f_2 dV_g = \int_U \begin{pmatrix} \frac{\partial f_1}{\partial x_1} \\ \vdots \\ \frac{\partial f_1}{\partial x_d} \end{pmatrix}^T (\boldsymbol{G}^x)^{-1} \begin{pmatrix} \frac{\partial f_2}{\partial x_1} \\ \vdots \\ \frac{\partial f_2}{\partial x_d} \end{pmatrix} dV_g \quad .$$

Consequently, whenever $(\mathcal{M}, g)$ be a compact connected Riemannian manifold, $-\Delta_{\mathcal{M}}$ defines a formally self-adjoint operator on functions of $\mathscr{C}^\infty(\mathcal{M})$ that satisfy appropriate boundary conditions. Moreover, it is a positive semi-definite operator as $\forall f \in \mathscr{C}^\infty(\mathcal{M})$ with boundary conditions when needed,

$$\langle f, -\Delta_{\mathcal{M}} f \rangle_{L^2(\mathcal{M})} = \langle \nabla_{\mathcal{M}} f, \nabla_{\mathcal{M}} f \rangle_{L^2(\mathcal{M})} \geq 0 \quad .$$

This result can be leveraged to prove the so-called spectral theorem that is introduced in the next subsection.

6.5.2 Spectral theorem

The spectral theorem is a fundamental result of differential geometry. It relies on the notion of eigenvalue problem that is now introduced.

Let $(\mathcal{M}, g)$ be a compact connected Riemannian manifold with (possibly empty) boundary $\partial\mathcal{M}$. An *eigenvalue problem* answers the following question: find all pairs (λ, ϕ) where $\lambda \in \mathbb{R}$ and $\phi \in \mathscr{C}^\infty(\mathcal{M})$, $\phi \neq 0$, such that

$$-\Delta_{\mathcal{M}} \phi = \lambda \phi \quad , \tag{6.14}$$

For such a pair (λ, ϕ) , λ is a called *eigenvalue* and ϕ is called *eigenfunction associated to the eigenvalue* λ. In particular, for a given eigenvalue λ, the set of all eigenfunctions associated to λ forms a vector space E_λ, called *eigenspace* of λ, and whose dimension is called *multiplicity* of λ. The set of all eigenvalues corresponding to an eigenvalue problem is called *spectrum* of $-\Delta_{\mathcal{M}}$ (for this problem).

Different eigenvalue problems corresponds to different requirements on the value of the eigenfunctions on the boundary $\partial\mathcal{M}$:

- The *closed eigenvalue problem* consists in finding pairs (λ, ϕ) that are solutions of Equation (6.14) in the case where $\partial\mathcal{M} = \emptyset$.
- The *Dirichlet eigenvalue problem* consists in finding pairs (λ, ϕ) that are solutions of Equation (6.14) and such that ϕ follows Dirichlet boundary conditions (in the case where $\partial\mathcal{M} \neq \emptyset$).
- The *Neumann eigenvalue problem* consists in finding pairs (λ, ϕ) that are solutions of Equation (6.14) and such that ϕ follows Neumann boundary conditions (in the case where $\partial\mathcal{M} \neq \emptyset$).

The next theorem provides a result on solutions of these eigenvalue problems.

Theorem 6.5.3 (Spectral theorem)**.** *Let $(\mathcal{M}, g)$ be a compact connected Riemannian manifold with (possibly empty) boundary $\partial\mathcal{M}$. The following assertions are true for the closed, the Dirichlet and the Neumann eigenvalue problems.*

- *The spectrum of $-\Delta_{\mathcal{M}}$ is an infinite (countable) sequence of real values*

$$0 \leq \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_k \leq \dots \quad ,$$

  where each eigenvalue is repeated in the sequence $\{\lambda_k\}_{k\in\mathbb{N}}$ as many times as its multiplicity. Besides, $\lim_{k\to\infty} \lambda_k = +\infty$.
- *Each eigenvalue has finite multiplicity and the eigenspaces corresponding to distinct eigenspaces are $L^2(\mathcal{M})$-orthogonal. Hence, for any eigenvalues λ_k, λ_j:*

$$\lambda_k \neq \lambda_j \Rightarrow \forall f_i \in E_{\lambda_i}, f_j \in E_{\lambda_j}, \quad \langle f_i, f_j \rangle_{L^2(\mathcal{M})} = 0 \quad .$$

- *Each eigenfunction is $\mathscr{C}^\infty$-smooth and analytic, and the direct sum of all eigenspaces is dense in $L^2(\mathcal{M})$ for the norm $\|.\|_{L^2(\mathcal{M})}$. Hence, there exists a $L^2(\mathcal{M})$-orthonormal basis $\{e_k\}_{k\in\mathbb{N}}$ of $L^2(\mathcal{M})$ such that $\forall k \in \mathbb{N}$, $e_k \in \mathscr{C}^\infty(\mathcal{M})$ is an eigenfunction associated to the eigenvalue λ_k:*

$$-\Delta_{\mathcal{M}} e_k = \lambda_k e_k \quad , \quad \|e_k\|_{L^2(\mathcal{M})} = 1 \quad \text{and} \quad k \neq j \Rightarrow \langle e_k, e_j \rangle_{L^2(\mathcal{M})} = 0 \quad .$$

  In particular,

$$\forall f \in L^2(\mathcal{M}), \quad \left\| f - \sum_{k\in\mathbb{N}} \langle e_j, f \rangle_{L^2(\mathcal{M})} e_j \right\|_{L^2(\mathcal{M})} = 0 \quad .$$

Proof. See (Lablée, 2015, Proposition 4.3.1 & Section 4.4) or (Jost, 2008, Theorem 3.2.1). □

This theorem provides a decomposition of any function $f \in L^2(\mathcal{M})$ onto an orthonormal basis $\{e_k\}_{k\in\mathbb{N}}$ of eigenfunctions of the negative Laplacian, as

$$f = \sum_{k\in\mathbb{N}} \langle e_j, f \rangle_{L^2(\mathcal{M})} e_j \quad ,$$

where the equality is understand in the L^2-sense.

The next result gives an estimate of the growth rate of the eigenvalues of the Laplacian of a compact Riemannian manifold.

Theorem 6.5.4 (Weyl asymptotic formula)**.** *Let $(\mathcal{M}, g)$ be a compact connected Riemannian d-manifold with (possibly empty) boundary $\partial\mathcal{M}$ and let $\{\lambda_k\}_{k\in\mathbb{N}}$ denote the eigenvalues of $-\Delta_{\mathcal{M}}$ as described in Theorem 6.5.3.*

Then,

$$\lambda_k \underset{k\to\infty}{\sim} \left(\frac{(2\pi)^d}{\beta_d V_g(\mathcal{M})} \right)^{2/d} k^{2/d} \quad , \tag{6.15}$$

where $\beta_d = \pi^{d/2}/\Gamma(d/2+1)$ is the volume of the (usual) unit ball of $\mathbb{R}^d$ and $V_g(\mathcal{M}) = \int_{\mathcal{M}} dV_g$.

Proof. See Section 7.6 of (Lablée, 2015). □

In the next subsection, the domain of definition of the Laplace–Beltrami operator is extended to a wider class of functions than just $\mathscr{C}^\infty(\mathcal{M})$. This extension relies on the notion of distribution on $\mathcal{M}$ that will be introduced.

6.5.3 Sobolev spaces and distributions on a Riemannian manifold

In this section, the notion of distribution on a Riemannian manifold is introduced in order to safely define the Laplacian of a non-smooth function of the manifold. This step is important as many of the functions whose Laplacian will be considered in the remainder of this work will not be smooth but merely piecewise differentiable (cf. Chapter 8).

Throughout this section, $(\mathcal{M}, g)$ denotes a compact connected Riemannian manifold with (possibly empty) boundary $\partial\mathcal{M}$. Let $\mathcal{C}_0^\infty(\mathcal{M}) \subset \mathcal{C}^\infty(\mathcal{M})$ be the set of smooth functions of $\mathcal{M}$ with compact support in $\mathrm{Int}(\mathcal{M}) = \mathcal{M}\backslash\partial\mathcal{M}$.

Distributions on a Riemannian space

The notion of distribution on $\mathcal{M}$ is now introduced. Let $\mathscr{D}(\mathcal{M})$ denote either $\mathcal{C}^\infty(\mathcal{M})$ or $\mathcal{C}_0^\infty(\mathcal{M})$. A *distribution T with test function space $\mathscr{D}(\mathcal{M})$* is a linear map from $\mathscr{D}(\mathcal{M})$ to $\mathbb{R}$ which is also continuous i.e. for any sequence $\{u_k\}_{k\in\mathbb{N}}$ of functions of $\mathscr{D}(\mathcal{M})$ converging to a function $u \in \mathscr{D}(\mathcal{M})$, the sequence $\{T(u_k)\}_{k\in\mathbb{N}}$ converges to $T(u)$.

In particular, given that $\mathcal{M}$ is compact and that therefore $\mathscr{D}(\mathcal{M}) \subset L^2(\mathcal{M})$, we can associate to any $f \in L^2(\mathcal{M})$ the distribution T_f with test function space $\mathscr{D}(\mathcal{M})$ defined by

$$T_f : u \in \mathscr{D}(\mathcal{M}) \mapsto T_f(u) = \langle f, u\rangle_{L^2(\mathcal{M})} \tag{6.16}$$

Note in particular that Equation (6.16) is actually defined for $u \in L^2(\mathcal{M})$ and that therefore T_f can also be considered as a linear continuous map $T_f : L^2(\mathcal{M}) \to \mathbb{R}$.

More generally, the fact that $\mathscr{D}(\mathcal{M})$ is dense in $L^2(\mathcal{M})$ (Bérard, 2006, Chapter III, Point (13)) allows to extend the domain of definition of some distributions.

Lemma 6.5.5. *Let $f \in L^2(\mathcal{M})$ and denote T_f the distribution with test function space $\mathscr{D}(\mathcal{M})$ defined by Equation* (6.16)*. Let T be any other distribution with test function space $\mathscr{D}(\mathcal{M})$.*

If T and T_f agree on $\mathscr{D}(\mathcal{M})$ then T admits a continuous linear extension on $L^2(\mathcal{M})$ defined by

$$\forall \phi \in L^2(\mathcal{M}), \quad T(\phi) := T_f(\phi) = \langle f, \phi\rangle_{L^2(\mathcal{M})} \quad .$$

Proof. Let $\phi \in L^2(\mathcal{M})$ and $\{\phi_k\}_{k\in\mathbb{N}}$ be a sequence of functions of $\mathscr{D}(\mathcal{M})$ converging to ϕ. Define $T(\phi) := \lim\limits_{k\to\infty} T(\phi_k)$. Then,

$$\lim_{k\to\infty} T(\phi_k) = \lim_{k\to\infty} T_f(\phi_k) = \lim_{k\to\infty} \langle f, \phi_k\rangle_{L^2(\mathcal{M})} = \langle f, \phi\rangle_{L^2(\mathcal{M})} := T_f(\phi) \in \mathbb{R}$$

□

Note in particular that Lemma 6.5.5 allows to actually identify arbitrary distributions with (the distributions associated with) functions of $L^2(\mathcal{M})$, as long as they coincide on the test function space.

Corollary 6.5.6. *Let $f_1, f_2 \in L^2(\mathcal{M})$ and denote T_{f_1}, T_{f_2} the associated distributions defined by Equation* (6.16)*.*

If T_{f_1} and T_{f_2} agree on $\mathscr{D}(\mathcal{M})$ then $f_1 = f_2$ in the L^2-sense.

Corollary 6.5.6 allows to identify distributions and functions of $L^2(\mathcal{M})$ and will be leveraged to extend the domain of definition of the Laplace operator.

Sobolev spaces on a Riemannian space

We now introduced three subsets $L^2(\mathcal{M})$ onto which the definition of the Laplacian operator can be extended. These sets of functions are referred to as *Sobolev spaces* of $\mathcal{M}$.

First, denote $\|.\|_{H^1(\mathcal{M})}$ the norm associated with the inner product $\langle .,. \rangle_{H^1(\mathcal{M})}$ on $\mathcal{C}^\infty(\mathcal{M})$ defined by:

$$\forall \varphi_1, \varphi_2 \in \mathcal{C}^\infty(\mathcal{M}), \quad \langle \varphi_1, \varphi_2 \rangle_{H^1(\mathcal{M})} = \langle \varphi_1, \varphi_2 \rangle_{L^2(\mathcal{M})} + \langle \nabla_{\mathcal{M}} \varphi_1, \nabla_{\mathcal{M}} \varphi_2 \rangle_{L^2(\mathcal{M})} \quad .$$

The first Sobolev space we will be working with is $H^1(\mathcal{M})$.

Definition 6.5.1. *$H^1(\mathcal{M})$ is defined as the closure of $\mathcal{C}^\infty(\mathcal{M})$ in $L^2(\mathcal{M})$ for the norm $\|.\|_{H^1(\mathcal{M})}$.*

$H^1(\mathcal{M})$ is therefore the smallest closed subset of $L^2(\mathcal{M})$ containing $\mathcal{C}^\infty(\mathcal{M})$, and can be seen as the set containing $\mathcal{C}^\infty(\mathcal{M})$ and the functions of $L^2(\mathcal{M})$ that are limit (with respect to the norm $\|.\|_{H^1(\mathcal{M})}$) of a sequence of elements of $\mathcal{C}^\infty(\mathcal{M})$. The elements of $H^1(\mathcal{M})$ are functions of $L^2(\mathcal{M})$ whose first derivatives (in the sense of distributions) can be identified to elements $L^2(\mathcal{M})$ (as in Lemma 6.5.5). In particular,

$$\forall \varphi_1, \varphi_2 \in H^1(\mathcal{M}), \quad \langle \nabla_{\mathcal{M}} \varphi_1, \nabla_{\mathcal{M}} \varphi_2 \rangle_{L^2(\mathcal{M})} < \infty \quad .$$

The second Sobolev space we will be working with is $H^1_0(\mathcal{M})$.

Definition 6.5.2. *$H^1_0(\mathcal{M})$ is defined as the closure of $\mathcal{C}^\infty_0(\mathcal{M})$ in $L^2(\mathcal{M})$ for the norm $\|.\|_{H^1(\mathcal{M})}$.*

The elements of $H^1_0(\mathcal{M})$ correspond to the elements of $H^1(\mathcal{M})$ that follow Dirichlet boundary conditions in the weak sense, i.e.

$$\forall \varphi \in H^1_0(\mathcal{M}), \forall u \in \mathscr{C}^0(\mathcal{M}), \quad \int_{\partial \mathcal{M}} u(\boldsymbol{s}) \cdot \varphi(\boldsymbol{s}) d\boldsymbol{s} = 0 \quad .$$

Remark 6.5.2. Note that by definition, $\mathcal{C}^\infty(\mathcal{M})$ (resp. $\mathcal{C}^\infty_0(\mathcal{M})$) is dense in $H^1(\mathcal{M})$ (resp. $H^1_0(\mathcal{M})$) for the norm $\|.\|_{H^1(\mathcal{M})}$.

Extensions of the Laplacian operator

The definition of the Laplacian operator is extended to functions in Sobolev spaces of $\mathcal{M}$, at least in the distribution sense, in a way that it coincides with the actual definition of the Laplacian when the functions are regular enough. Three extensions of the Laplacian operators corresponding to the three boundary conditions described earlier are now presented.

Closed Laplacian Let us assume that $\mathcal{M}$ is a manifold without boundary, i.e. $\partial \mathcal{M} = \emptyset$. For $\varphi \in H^1(\mathcal{M})$ denote T^C_φ the linear application defined by

$$\begin{array}{rccl} T^C_\varphi : & H^1(\mathcal{M}) & \to & \mathbb{R} \\ & u & \mapsto & T^C_\varphi(u) = \langle \nabla_{\mathcal{M}} \varphi, \nabla_{\mathcal{M}} u \rangle_{L^2(\mathcal{M})} \end{array} \tag{6.17}$$

Noting that $\mathscr{C}^\infty(\mathcal{M}) \subset H^1(\mathcal{M})$, T^C_φ actually defines a distribution on $\mathcal{M}$ with test function space $\mathscr{C}^\infty(\mathcal{M})$. In particular, if we assume that $\varphi \in H^1(\mathcal{M})$ is such that $-\Delta_{\mathcal{M}} \varphi$ can be computed from its current definition (cf. Section 6.5.1) and satisfies $-\Delta_{\mathcal{M}} \varphi \in L^2(\mathcal{M})$, we have from Green's theorem that

$$\forall u \in \mathscr{C}^\infty(\mathcal{M}), \quad T^C_\varphi(u) = \langle -\Delta_{\mathcal{M}} \varphi, u \rangle_{L^2(\mathcal{M})} \quad . \tag{6.18}$$

thus giving that T^C_φ coincides with $-\Delta_{\mathcal{M}} \varphi$ in the sense of distributions. Note also that, using the density of $\mathscr{C}^\infty(\mathcal{M})$ in $L^2(\mathcal{M})$, Equation (6.18) actually holds $\forall u \in H^1(\mathcal{M}) \subset L^2(\mathcal{M})$. Hence T^C_φ can be identified with the linear map $u \mapsto \langle -\Delta_{\mathcal{M}} \varphi, u \rangle_{L^2(\mathcal{M})}$ defined from the Laplacian of φ.

In the more general case where we only assume that $\varphi \in H^1(\mathcal{M})$, the Laplacian of φ is directly defined as the linear map T^C_φ given in Equation (6.17), and is then denoted $-\Delta_\mathcal{M}\varphi$ so that we can write

$$\forall \varphi_1, \varphi_2 \in H^1(\mathcal{M}), \quad \langle -\Delta_\mathcal{M}\varphi_1, \varphi_2 \rangle_{L^2(\mathcal{M})} := T^C_{\varphi_1}(\varphi_2) \quad .$$

Consequently we have $\forall \varphi_1, \varphi_2 \in H^1(\mathcal{M})$:

$$\langle -\Delta_\mathcal{M}\varphi_1, \varphi_2 \rangle_{L^2(\mathcal{M})} = \langle \nabla_\mathcal{M}\varphi_1, \nabla_\mathcal{M}\varphi_2 \rangle_{L^2(\mathcal{M})} = \langle -\Delta_\mathcal{M}\varphi_2, \varphi_1 \rangle_{L^2(\mathcal{M})} \tag{6.19}$$

Dirichlet Laplacian Let us assume that $\mathcal{M}$ is a manifold with non-empty boundary $\partial\mathcal{M}$. For $\varphi \in H^1_0(\mathcal{M})$ denote T^D_φ the linear application defined by

$$\begin{array}{rccl} T^D_\varphi : & H^1_0(\mathcal{M}) & \to & \mathbb{R} \\ & u & \mapsto & T^D_\varphi(u) = \langle \nabla_\mathcal{M}\varphi, \nabla_\mathcal{M} u \rangle_{L^2(\mathcal{M})} \end{array} \tag{6.20}$$

Note that the only difference between Equation (6.17) and Equation (6.20) is the domain of definition of the map. The same reasoning as the one used in the closed case can therefore be applied. It shows that T^D_φ once again defines a distribution on $\mathcal{M}$, but with test function space $\mathscr{C}^\infty_0(\mathcal{M})$.

Hence, when $\varphi \in H^1_0(\mathcal{M})$, the Laplacian of φ is directly defined as the linear map T^D_φ given in Equation (6.20), and is then denoted $-\Delta_\mathcal{M}\varphi$. In particular Equation (6.19) holds now $\forall \varphi_1, \varphi_2 \in H^1_0(\mathcal{M})$.

Neumann Laplacian Let us assume that $\mathcal{M}$ is a manifold with non-empty boundary $\partial\mathcal{M}$. Let $\varphi \in H^1(\mathcal{M})$ such that φ follows Neumann boundary conditions in the L^2-sense, meaning that

$$\forall u \in \mathscr{C}^0(\partial\mathcal{M}), \quad \int_{\partial\mathcal{M}} u(\boldsymbol{s}) \cdot g_{\boldsymbol{s}}(n_{\boldsymbol{s}}, \nabla_\mathcal{M}\varphi(\boldsymbol{s}))d\boldsymbol{s} = 0 \tag{6.21}$$

Denote then T^N_φ the linear application defined by

$$\begin{array}{rccl} T^N_\varphi : & H^1(\mathcal{M}) & \to & \mathbb{R} \\ & u & \mapsto & T^N_\varphi(u) = \langle \nabla_\mathcal{M}\varphi, \nabla_\mathcal{M} u \rangle_{L^2(\mathcal{M})} \end{array} \tag{6.22}$$

Note that this is actually the same definition as Equation (6.17): only the domain from which the function φ was chosen changed. The same reasoning as in the "closed" case can then be used to define the Laplacian of φ from the map T^N_φ.

Namely, when $\varphi \in H^1(\mathcal{M})$ follows Neumann boundary conditions, the Laplacian of φ is directly defined as the linear map T^N_φ given in Equation (6.22), and is then denoted $-\Delta_\mathcal{M}\varphi$.

6.6 Riemannian geometry and local deformations

To conclude this chapter on Riemannian geometry, we reintroduce the main defining properties of Riemannian manifolds using a "practical" and rather intuitive perspective. Indeed, as we may now see, Riemannian manifolds are a mathematical object particularly suited to model spatial domains undergoing local deformations. This parallel will be leveraged later in this work to interpret (generalized) random fields defined on Riemannian manifolds as locally deformed (generalized) random fields (cf. Chapter 7).

6.6.1 Link to Continuum mechanics

In this subsection, a parallel is drawn between the study of finite deformations in continuum mechanics and Riemannian manifolds, which provides an interpretation of the notion of Riemannian metric as being linked to local deformations (Fiala, 2008; Simo and Marsden, 1984).

Let B_R denote a body that occupies a portion of a spatial domain. Formally, B_R can be seen as a continuous and connected subset of $\mathbb{R}^d$. Let us assume that the body B_R is deformed from its initial (reference) configuration B_R into a deformed one $B_D \subset \mathbb{R}^d$. This process, which is

assumed to be reversible, is called a finite deformation and can be modeled as a diffeomorphism $\Phi : B_R \to \Phi(B_R) = B_D$ that maps any point $\boldsymbol{p} \in B_R$ in the reference configuration to its position $\boldsymbol{q} = \Phi(\boldsymbol{p}) \in B_D$ in the deformed configuration.

Let $\boldsymbol{p} \in B_R$ and let $\boldsymbol{q} \in B_D$ be its position in the deformed configuration. Let $d\boldsymbol{p}$ be an infinitesimal displacement from $\boldsymbol{p}$ to a point $(\boldsymbol{p} + d\boldsymbol{p}) \in B_R$ (infinitely close to $\boldsymbol{p}$ in B_R). Then the displacement $d\boldsymbol{q}$ between both points in the deformed body B_D can be written as

$$d\boldsymbol{q} = \Phi(\boldsymbol{p} + d\boldsymbol{p}) - \Phi(\boldsymbol{p}) \quad .$$

Using a Taylor development of first order around $\boldsymbol{p}$, this last equation gives

$$d\boldsymbol{q} = \Phi(\boldsymbol{p}) + J_\Phi(\boldsymbol{p})d\boldsymbol{p} + o(\|d\boldsymbol{p}\|) - \Phi(\boldsymbol{p}) = J_\Phi(\boldsymbol{p})d\boldsymbol{p} + o(\|d\boldsymbol{p}\|) \quad ,$$

where $J_\phi(\boldsymbol{p})$ denotes the Jacobian matrix of Φ at point $\boldsymbol{p} \in B_R$. Hence, if the terms of higher order are neglected (due to the infinitesimal nature of $d\boldsymbol{p}$), then the displacements in the reference and in the deformed configurations are linked by

$$d\boldsymbol{q} = \boldsymbol{F}(\boldsymbol{p})d\boldsymbol{p} \quad ,$$

where $\boldsymbol{F}$ is a tensor field, called deformation gradient tensor field, that associates to any point $\boldsymbol{p}$ in the reference configuration a tensor $F(\boldsymbol{p})$ defined by

$$F(\boldsymbol{p}) : d\boldsymbol{p} \mapsto \boldsymbol{F}(\boldsymbol{p})d\boldsymbol{p} = J_\Phi(\boldsymbol{p})d\boldsymbol{p} \quad .$$

In particular, to characterize length changes and angle changes around a point $\boldsymbol{p} \in B_R$ after the deformation process, it is useful to see how inner products between displacement vectors vary. Let then $d\boldsymbol{p}_1, d\boldsymbol{p}_2$ be two displacement vectors from $\boldsymbol{p}$ and let $d\boldsymbol{q}_1$ and $d\boldsymbol{q}_2$ be their images in the deformed configuration. We have

$$\langle d\boldsymbol{q}_1, d\boldsymbol{q}_2 \rangle = \langle \boldsymbol{F}(\boldsymbol{p})d\boldsymbol{p}_1, \boldsymbol{F}(\boldsymbol{p})d\boldsymbol{p}_2 \rangle = d\boldsymbol{p}_1^T \boldsymbol{C}(\boldsymbol{p})d\boldsymbol{p}_2 \quad ,$$

where $\boldsymbol{C}$ is a tensor field, called (right) Cauchy-Green deformation tensor, that associates to any point $\boldsymbol{p}$ in the reference configuration a tensor $C(\boldsymbol{p})$ defined by

$$C(\boldsymbol{p}) : (d\boldsymbol{p}_1, d\boldsymbol{p}_2) \longmapsto d\boldsymbol{p}_1^T \boldsymbol{C}(\boldsymbol{p})d\boldsymbol{p}_2 = d\boldsymbol{p}_1^T \boldsymbol{F}(\boldsymbol{p})^T \boldsymbol{F}(\boldsymbol{p})d\boldsymbol{p}_2 \quad .$$

The Cauchy-Green deformation tensor informs on how lengths and angles of small vectors around a point $\boldsymbol{p}$ in the reference configuration are modified after the deformation process, and therefore how the geometry around that point is modified. Indeed, for any vectors $d\boldsymbol{p}, d\boldsymbol{p}_1, d\boldsymbol{p}_2$ around any point $\boldsymbol{p}$ in the reference configuration, the length $d\boldsymbol{p}$ and the angle θ between $d\boldsymbol{p}_1$ and $d\boldsymbol{p}_2$ are modified according to:

$$\begin{array}{rcl} \|d\boldsymbol{p}\| & \text{becomes} & \|d\boldsymbol{q}\| = \sqrt{d\boldsymbol{p}^T \boldsymbol{C}(\boldsymbol{p})d\boldsymbol{p}} \quad , \\ \cos\theta = \frac{d\boldsymbol{p}_1^T d\boldsymbol{p}_2}{\|d\boldsymbol{p}_1\|\|d\boldsymbol{p}_2\|} & \text{becomes} & \cos\theta' = \frac{(d\boldsymbol{p}_1')^T d\boldsymbol{p}_2'}{\|d\boldsymbol{p}_1'\|\|d\boldsymbol{p}_2'\|} = \frac{d\boldsymbol{p}_1^T \boldsymbol{C}(\boldsymbol{p})d\boldsymbol{p}_2}{\sqrt{d\boldsymbol{p}_1^T \boldsymbol{C}(\boldsymbol{p})d\boldsymbol{p}_1}\sqrt{d\boldsymbol{p}_2^T \boldsymbol{C}(\boldsymbol{p})d\boldsymbol{p}_2}} \quad . \end{array}$$

Circling back to the subject of this section, Riemannian manifolds actually provide a natural mathematical framework for the study of deformations. Indeed, consider now that B_R and B_D are submanifolds of $\mathbb{R}^d$, and that B_D is equipped with the Euclidean metric, denoted g^{Euc}. The deformation diffeomorphism Φ therefore defines a smooth map between two smooth manifolds, B_R and B_D. Hence, the pullback metric of g^{Euc} by Φ defines a Riemannian metric on B_R by:

$$\forall \boldsymbol{p} \in \mathcal{M}, \forall u_{\boldsymbol{p}}, v_{\boldsymbol{p}} \in T_{\boldsymbol{p}}B_R, \quad \Phi^* g^{\text{Euc}}(u_{\boldsymbol{p}}, v_{\boldsymbol{p}}) = g^{\text{Euc}}(d\Phi_{\boldsymbol{p}}(u_{\boldsymbol{p}}), d\Phi_{\boldsymbol{p}}(v_{\boldsymbol{p}})) \quad ,$$

where $d\Phi_{\boldsymbol{p}}$ denotes the differential of the map $\Phi : B_R \to B_D$ at the point $\boldsymbol{p} \in \mathbb{B}$. Using the definition of the Euclidean metric, this last equation becomes

$$\Phi^* g^{\text{Euc}}(u_{\boldsymbol{p}}, v_{\boldsymbol{p}}) = \sum_{k=1}^{d} [J_\Phi(\boldsymbol{p})\boldsymbol{u}_{\boldsymbol{p}}]_k [J_\Phi(\boldsymbol{p})\boldsymbol{v}_{\boldsymbol{p}}]_k = \langle J_\Phi(\boldsymbol{p})\boldsymbol{u}_{\boldsymbol{p}}, J_\Phi(\boldsymbol{p})\boldsymbol{v}_{\boldsymbol{p}} \rangle = \boldsymbol{u}_{\boldsymbol{p}}^T J_\Phi(\boldsymbol{p})^T J_\Phi(\boldsymbol{p})\boldsymbol{v}_{\boldsymbol{p}} \quad .$$

Identifying (through the exponential map) the roles of the (representative) vectors $\boldsymbol{u}_{\boldsymbol{p}}, \boldsymbol{v}_{\boldsymbol{p}}$ defined on the tangent space of B_R seen as a manifold with the displacement vectors along the body $d\boldsymbol{p}_1, d\boldsymbol{p}_2$ of the continuum mechanics approach, we retrieve the expression of the Cauchy-Green deformation tensor. Hence the deformation tensor Φ simply corresponds to the pullback metric of the Euclidean metric by the deformation diffeomorphism Φ, and therefore defines its own Riemannian metric on the undeformed body B_R. The geometry induced by a Riemannian metric $g = \Phi^* g^{\text{Euc}}$ on a manifold B_R can be interpreted as the geometry that would exist on the body B_R after it has been deformed through Φ.

6.6.2 Laplacian as a change of coordinates

In this subsection, the Laplace–Beltrami operator, which plays a central role in the spectral analysis of Riemannian manifolds is reintroduced using the same formalism as the one used in the previous subsection. We show that the Laplace–Beltrami operator defined in B_R can be identified to a classical Laplacian operator defined on B_D through the change of coordinates induced by the deformation transformation Φ. We assume that the functions considered in this subsection follow Dirichlet boundary conditions, i.e. they are zero on the boundary of B_R (or B_D)

Once again B_D is purely seen as a domain of $\mathbb{R}^d$ (or equivalently as a d-submanifold of $\mathbb{R}^d$ endowed with the Euclidean metric). Hence the definition of the gradient and the Laplacian of functions of B_D corresponds to the classical definition of such objects for functions of $\mathbb{R}^d$, i.e. using partial derivatives with respect to Cartesian coordinates. Denote then $\nabla_{\mathbb{R}^d}$ the gradient operator and $-\Delta_{\mathbb{R}^d}$ the negative Laplacian operator as defined on $\mathbb{R}^d$.

The Laplacian of a (sufficiently smooth) function $f : B_D \to \mathbb{R}$ can also be defined as a distribution with test function space $\mathscr{C}_0^\infty(B_D)$ (i.e. the set of smooth functions of B_D that are zero on the boundary ∂B_D). It then maps any $u \in \mathscr{C}_0^\infty(B_D)$ to the scalar value $\langle -\Delta_{\mathbb{R}^d} f, u\rangle_{L^2(\mathbb{R}^d)} \in \mathbb{R}$ defined by

$$\langle -\Delta_{\mathbb{R}^d} f, u\rangle_{L^2(\mathbb{R}^d)} := \langle \nabla_{\mathbb{R}^d} f, \nabla_{\mathbb{R}^d} u\rangle_{L^2(\mathbb{R}^d)} := \sum_{k=1}^{d} \langle \frac{\partial f}{\partial q_k}, \frac{\partial u}{\partial q_k}\rangle_{L^2(\mathbb{R}^d)}, \quad u \in \mathscr{C}_0^\infty(B_D) \quad , \tag{6.23}$$

where $\langle ., . \rangle_{L^2(\mathbb{R}^d)}$ denotes the inner-product associated with square-integrable functions of $\mathbb{R}^d$ (i.e. the Lebesgue integral of their product). In particular, if $f \in \mathscr{C}^2(B_D)$, $-\Delta_{\mathbb{R}^d} f \in \mathscr{C}^0(B_D)$ and we have

$$\langle -\Delta_{\mathbb{R}^d} f, u\rangle_{L^2(\mathbb{R}^d)} = \int_{B_D} (-\Delta_{\mathbb{R}^d} f)(\boldsymbol{q}) u(\boldsymbol{q}) d\boldsymbol{q}, \quad u \in \mathscr{C}_0^\infty(B_D) \quad . \tag{6.24}$$

Let $u \in \mathscr{C}_0^\infty(B_D)$ and let $f \in \mathscr{C}^2(B_D)$. We denote $\tilde{f} = f \circ \Phi : B_R \to \mathbb{R}$ the function of B_R canonically associated with f through Φ. We therefore have $f = \tilde{f} \circ \Phi^{-1}$ and the chain rule (cf. Theorem A.1.1) gives an expression of the partial derivative of f with respect to those of $\tilde{f}$:

$$\forall k \in [\![1, d]\!], \quad \frac{\partial f}{\partial q_k}(\boldsymbol{q}) = \sum_{l=1}^{d} \frac{\partial \tilde{f}}{\partial p_l}(\Phi^{-1}(\boldsymbol{q})) \frac{\partial [\Phi^{-1}]_l}{\partial q_k}(\boldsymbol{q}) \quad .$$

Injecting this last equation in Equation (6.23) then gives

$$\begin{aligned}
\langle -\Delta_{\mathbb{R}^d} f, u\rangle_{L^2(\mathbb{R}^d)} &= \sum_{k=1}^{d} \int_{B_D} \sum_{l=1}^{d} \sum_{l'=1}^{d} \frac{\partial \tilde{f}}{\partial p_l}(\Phi^{-1}(\boldsymbol{q})) \frac{\partial [\Phi^{-1}]_l}{\partial q_k}(\boldsymbol{q}) \frac{\partial \tilde{u}}{\partial p_{l'}}(\Phi^{-1}(\boldsymbol{q})) \frac{\partial [\Phi^{-1}]_{l'}}{\partial q_k}(\boldsymbol{q}) d\boldsymbol{q} \\
&= \int_{B_D} \sum_{l=1}^{d} \sum_{l'=1}^{d} \frac{\partial \tilde{f}}{\partial p_l}(\Phi^{-1}(\boldsymbol{q})) \frac{\partial \tilde{u}}{\partial p_{l'}}(\Phi^{-1}(\boldsymbol{q})) \sum_{k=1}^{d} \frac{\partial [\Phi^{-1}]_l}{\partial q_k}(\boldsymbol{q}) \frac{\partial [\Phi^{-1}]_{l'}}{\partial q_k}(\boldsymbol{q}) d\boldsymbol{q} \\
&= \sum_{l'=1}^{d} \int_{B_D} \frac{\partial \tilde{u}}{\partial p_{l'}}(\Phi^{-1}(\boldsymbol{q})) \sum_{l=1}^{d} \frac{\partial \tilde{f}}{\partial p_l}(\Phi^{-1}(\boldsymbol{q})) [J_{\Phi^{-1}}(\boldsymbol{q}) J_{\Phi^{-1}}(\boldsymbol{q})^T]_{ll'} d\boldsymbol{q} \ ,
\end{aligned}$$

where of course, $\tilde{u} = u \circ \Phi$ and $J_{\Phi^{-1}}(\boldsymbol{q})$ denotes the Jacobian matrix of $\Phi^{-1} : B_D \to B_R$ at the point $\boldsymbol{q} \in B_D$. Operating a change of variables $\boldsymbol{q} = \Phi(\boldsymbol{p})$ in the last equation then gives (cf. Theorem A.1.2)

$$\langle -\Delta_{\mathbb{R}^d} f, u\rangle_{L^2(\mathbb{R}^d)} = \sum_{l'=1}^{d} \int_{B_R} \frac{\partial \tilde{u}}{\partial p_{l'}}(\boldsymbol{p}) \sum_{l=1}^{d} \frac{\partial \tilde{f}}{\partial p_l}(\boldsymbol{p}) [J_{\Phi^{-1}}(\Phi(\boldsymbol{p})) J_{\Phi^{-1}}(\Phi(\boldsymbol{p}))^T]_{ll'} |\det J_\Phi(\boldsymbol{p})| d\boldsymbol{p} \ ,$$

where $J_\Phi(\boldsymbol{p})$ denotes the (usual) Jacobian matrix of $\Phi : B_R \to B_D$.

Note in particular that the chain rule also yields that $\forall \boldsymbol{p} \in B_R$, $J_\Phi(\boldsymbol{p})^{-1} = J_{\Phi^{-1}}(\Phi(\boldsymbol{p}))$. Hence, for any $\boldsymbol{p} \in B_R$, by denoting $\boldsymbol{G}(\boldsymbol{p})$ the matrix defined by

$$\boldsymbol{G}(\boldsymbol{p}) = J_\Phi(\boldsymbol{p})^T J_\Phi(\boldsymbol{p}), \quad \boldsymbol{p} \in B_R \quad , \tag{6.25}$$

we get

$$\langle -\Delta_{\mathbb{R}^d} f, u \rangle_{L^2(\mathbb{R}^d)} = \sum_{l'=1}^{d} \int_{B_R} \frac{\partial \tilde{u}}{\partial p_{l'}}(\boldsymbol{p}) \sum_{l=1}^{d} \frac{\partial \tilde{f}}{\partial p_l}(\boldsymbol{p}) [\boldsymbol{G}(\boldsymbol{p})^{-1}]_{ll'} \sqrt{\det \boldsymbol{G}(\boldsymbol{p})} d\boldsymbol{p} \quad .$$

Finally, given that $u \in \mathscr{C}_0^\infty(B_D)$, we have $\tilde{u} \in \mathscr{C}_0^\infty(B_R)$ and so, the integration by parts formula gives

$$\begin{aligned} \langle -\Delta_{\mathbb{R}^d} f, u \rangle_{L^2(\mathbb{R}^d)} &= -\sum_{l'=1}^{d} \int_{B_R} \tilde{u}(\boldsymbol{p}) \frac{\partial}{\partial p_{l'}} \left(\sqrt{\det \boldsymbol{G}} \sum_{l=1}^{d} [\boldsymbol{G}^{-1}]_{ll'} \frac{\partial \tilde{f}}{\partial p_l} \right)(\boldsymbol{p}) d\boldsymbol{p} \\ &= -\int_{B_R} \tilde{u}(\boldsymbol{p}) \mathrm{div}_{\mathbb{R}^d} \left(\sqrt{\det \boldsymbol{G}} \sum_{l=1}^{d} [\boldsymbol{G}^{-1}]_{ll'} \frac{\partial \tilde{f}}{\partial p_l} \right)(\boldsymbol{p}) d\boldsymbol{p} \quad , \end{aligned}$$

by definition of the divergence operator $\mathrm{div}_{\mathbb{R}^d}$ acting on functions $\mathbb{R}^d$.

On the other hand, a direct change of coordinates $\boldsymbol{q} = \Phi(\boldsymbol{p})$ in Equation (6.24) gives

$$\begin{aligned} \langle -\Delta_{\mathbb{R}^d} f, u \rangle_{L^2(\mathbb{R}^d)} &= \int_{B_D} -\Delta_{\mathbb{R}^d} f(\boldsymbol{q}) u(\boldsymbol{q}) d\boldsymbol{q} = \int_{B_R} -\Delta_{\mathbb{R}^d} f(\Phi(\boldsymbol{p})) u(\Phi(\boldsymbol{p})) |\det J_\Phi(\boldsymbol{p})| d\boldsymbol{p} \\ &= -\int_{B_R} \widetilde{\Delta_{\mathbb{R}^d} f}(\boldsymbol{p}) \sqrt{\det G(\boldsymbol{p})} \tilde{u}(\boldsymbol{p}) d\boldsymbol{p} \quad , \end{aligned}$$

where $\widetilde{\Delta_{\mathbb{R}^d} f} = (\Delta_{\mathbb{R}^d} f) \circ \Phi$ denotes the function of B_R canonically associated with $\Delta_{\mathbb{R}^d} f$ through Φ.

Identifying these two expressions of $\langle -\Delta_{\mathbb{R}^d} f, u \rangle_{L^2(\mathbb{R}^d)}$, which are true $\forall u \in \mathscr{C}_0^\infty(B_D)$, then gives

$$\widetilde{\Delta_{\mathbb{R}^d} f} = (\Delta_{\mathbb{R}^d} f) \circ \Phi = \frac{1}{\sqrt{\det \boldsymbol{G}}} \mathrm{div}_{\mathbb{R}^d} \left(\sqrt{\det \boldsymbol{G}} \sum_{l=1}^{d} [\boldsymbol{G}^{-1}]_{ll'} \frac{\partial f \circ \Phi}{\partial p_l} \right), \quad f \in \mathscr{C}^2(B_D) \, . \tag{6.26}$$

We recognize in the right member of the equation the expression (in local coordinates) of the Laplace–Beltrami operator applied to the function $\tilde{f} = f \circ \Phi : B_R \to \mathbb{R}$, where B_R is now seen as a Riemannian d-manifold endowed with a metric g defined from the field of positive-definite matrices $\{\boldsymbol{G}(\boldsymbol{p})\}_{\boldsymbol{p} \in B_R}$ given by Equation (6.25). We therefore retrieve the same construction of a Riemannian manifold from a body B through deformation transformation Φ, as the one presented in Section 6.6.1.

Hence, applying the Laplace–Beltrami operator to a (sufficiently smooth) function $\tilde{f}$ of the Riemannian manifold (B_R, g) is equivalent to applying the classical Laplacian operator of $\mathbb{R}^d$ on the function $\tilde{f} \circ \Phi^{-1}$ defined on the deformed body $B_D = \Phi(B_R)$. The Laplace–Beltrami operator on (B_R, g) can therefore be seen as a classical Laplacian operator on the deformed configuration B_D, seen through the change of coordinates induced by Φ.

Conclusion

In this chapter, we introduced basic notions of differential and Riemannian geometry. The focus was set on (compact) Riemannian manifolds, which can be seen as locally Euclidean spaces for which the geometry around each point is defined by a spatially varying inner product called Riemannian metric. In particular, integration and differential calculus were reintroduced in these spaces.

We provided a more "physical" interpretation of Riemannian manifolds, which actually relates them the spatial deformation models used in Geostatistics to model non-stationary data (Sampson and Guttorp, 1992). The Riemannian metric was then simply interpreted as an application allowing to compute lengths and angles as if the spatial domain on which it is defined was deformed.

The Laplace–Beltrami operator, which corresponds to the generalization of the Laplace operator to Riemannian manifolds, was introduced. As we may see in the subsequent chapters, this operator plays a key role when working with "functions" defined on the manifold. We indeed stated the spectral theorem, which ensures that its eigenfunctions act like a decomposition basis for any square-integrable function defined on the Riemannian manifold.

The next chapter will build on this result to build a class of (generalized) random fields that can be seen as the counterparts, on a Riemannian manifold, of isotropic stationary Gaussian random fields of $\mathbb{R}^d$. As we may see, working with these fields will answer the modeling problem posed in this thesis.

7

Generalized random fields on Riemannian manifolds

Contents

Résumé

Dans ce chapitre, nous présentons un cadre mathématique permettant de définir et de travailler avec des champs gaussiens définis sur des domaines complexes ou caractérisés par des anisotropies locales. L'idée est d'étendre aux variétés riemanniennes la notion de champ gaussien isotrope et stationnaire telle que définie sur des domaines euclidiens. Travailler sur des variétés riemanniennes permet à la fois de modéliser des champs définis sur des domaines seulement localement euclidiens, mais aussi de modéliser des anisotropies locales en définissant une métrique appropriée.

Nous commençons par introduire une classe de champs aléatoires généralisés définie à partir des fonctions propres et des valeurs propres de l'opérateur de Laplace–Beltrami de la variété riemannienne. Nous en étudions ensuite les propriétés statistiques, et plus particulièrement leur covariance afin de montrer en quoi cette classe de champ répond à notre problématique initiale. Enfin, nous proposons une discrétisation de Ritz–Galerkin de ces champs, qui sera destinée aux application numériques.

Introduction

In this chapter we circle back to our initial modeling problem, that is defining a framework that allows to easily work with Gaussian random fields defined on complex spatial domains or characterized by local anisotropies. As it turns out, this problem is answered by transposing the notion of isotropic stationary Gaussian random fields (as defined in $\mathbb{R}^d$) to Riemannian manifolds. Indeed, as we saw in the previous chapter, these objects can naturally represent complex domains and local deformations of space.

We will propose a passage from the definition of random fields on $\mathbb{R}^d$ to Riemannian manifolds using their characterization by a pseudo-differential operator (Lang and Potthoff, 2011). This allows to redefine the notion of stationarity without involving a covariance function, and therefore in a way that is independent of the actual geometry of the manifold. Doing otherwise would indeed have forced us to find a counterpart to the notions of "invariance by translation and rotation" that characterize the covariance of isotropic random fields in $\mathbb{R}^d$, and which are obviously geometry-dependent. We therefore end up with a framework that can easily be transposed to a wide range of domains.

However, the fact that we are working with pseudo-differential operators forces us to generalize the notion of random field to more than just a stochastic process indexed by the spatial domain. This is why the notion of generalized random field is introduced. It allows us to justify the fact that we work with both pseudo-differential operators, and processes/fields that may not be smooth.

The approach we present is similar to the approach used by Lindgren et al. (2011) to generalize the definition of a class of stochastic partial differential equations to manifolds in order to define Matérn field on them. Bolin et al. (2018) also used this approach to derive results on the numerical approximation of solutions of SPDEs defined by a fractional power of an elliptic differential operator on a bounded domain of $\mathbb{R}^d$.

We extend both approaches to the case where the domain of study is a compact Riemannian manifold. In particular, the generalized random fields that will be considered are defined by leveraging the spectral theorem on compact Riemannian manifolds (cf. Theorem 6.5.3). As we may see, this approach has several advantages:

- the proposed construction of generalized random fields holds for any compact connected Riemannian manifold,
- the covariance properties of the resulting (generalized) random fields can easily be linked to the covariance properties of usual random fields defined on $\mathbb{R}^d$, and in particular those that display local anisotropies,
- the resulting generalized random fields can be discretized using a very general approach and doing so, can be numerically computed.

In a first section, we introduce the class of generalized random fields which will be used in this work, and the surrounding framework. Our main contributions are presented in the two subsequent sections.

On one hand, we leverage the notion of metric to show how they can relate to the definition of local anisotropies on the resulting random fields. This is done by looking into the covariance properties of these generalized random fields.

On the other hand, a method of discretization of these generalized random fields, based on the Ritz–Galerkin approximation approach, is presented: the generalized random fields are approximated by a weighted sum of linearly independent (deterministic) functions defined on the manifold and a theorem describing the statistical properties of the weights is stated (and proven). As we may see, this discretization is linked to the notion of stochastic graph signal, and will be leveraged in the subsequent chapters to numerically work with the generalized random fields defined here.

Note however that the work presented in this chapter only concerns zero-mean Gaussian fields.

Assumption 7.1. *All (generalized) Gaussian fields in this work are assumed to be zero-mean.*

7.1 Generalized random fields: mathematical framework

In this section, the mathematical framework leading to the definition of a particular class of generalized random fields on a compact Riemannian manifold is presented.

7.1.1 Functions of the Laplacian

The aim of this subsection is to introduce a class of operators acting on $L^2(\mathcal{M})$, called functions of the Laplacian, and derived from the spectral theorem (cf. Theorem 6.5.3). These operators are classically used to express solutions of some differential equations and to prove the Weyl asymptotic formula that was introduced in Theorem 6.5.4 (Bouclet, 2012). We will be using these operators to define the class of generalized random fields with which we will be working.

Consider then $\gamma : \mathbb{R}_+ \mapsto \mathbb{R}$ such that γ is bounded. We introduce $\gamma(-\Delta_{\mathcal{M}})$ the (linear) operator on $L^2(\mathcal{M})$ whose action is defined by:

$$\forall f \in L^2(\mathcal{M}), \quad \gamma(-\Delta_{\mathcal{M}})f = \sum_{k\in\mathbb{N}} \gamma(\lambda_k)\langle e_k, f\rangle_{L^2(\mathcal{M})} e_k \quad . \tag{7.1}$$

$\gamma(-\Delta_{\mathcal{M}})$ is called *function of the Laplacian*. The next proposition details the action of this operator.

Proposition 7.1.1. *The operator $\gamma(-\Delta_{\mathcal{M}})$ defined in Equation (7.1) satisfies*

$$\gamma(-\Delta_{\mathcal{M}}) : L^2(\mathcal{M}) \to L^2(\mathcal{M}) \quad .$$

Besides, its definition does not depend on the orthonormal basis of eigenfunctions of $-\Delta_{\mathcal{M}}$ used in Equation (7.1).

Proof. γ is bounded, and therefore, so is γ^2. Hence, there exists $M \in \mathbb{R}$ such that $\forall \lambda \in \mathbb{R}_+$, $\gamma(\lambda)^2 < M$. Take then $f \in L^2(\mathcal{M})$, and let $\{\tilde{f}_p\}_{p\in\mathbb{N}}$ be the sequence defined by

$$\tilde{f}_p = \sum_{k=0}^{p} \gamma(\lambda_k)\langle e_k, f\rangle_{L^2(\mathcal{M})} e_k, \quad p \in \mathbb{N} \quad .$$

Note in particular that $\forall p, q \in \mathbb{N}$ such that $q > p$ we have

$$\|\tilde{f}_q - \tilde{f}_p\|^2_{L^2(\mathcal{M})} = \sum_{k=p+1}^{q} \gamma(\lambda_k)^2 \langle e_k, f\rangle^2_{L^2(\mathcal{M})} \leq M \sum_{k=p+1}^{q} \langle e_k, f\rangle^2_{L^2(\mathcal{M})} \underset{p,q\to+\infty}{\longrightarrow} 0 \quad ,$$

given that $\sum_{k\in\mathbb{N}} \langle e_k, f\rangle^2_{L^2(\mathcal{M})} = \|f\|^2_{L^2(\mathcal{M})} < \infty$. Hence $\{\tilde{f}_p\}_{p\in\mathbb{N}}$ is a Cauchy sequence of $L^2(\mathcal{M})$. It is therefore convergent in $L^2(\mathcal{M})$ given that $L^2(\mathcal{M})$ is complete.

Finally, simply notice that by definition,

$$\gamma(-\Delta_{\mathcal{M}})f := \lim_{p\to+\infty} \tilde{f}_p \quad .$$

to conclude the proof. □

$\gamma(-\Delta_{\mathcal{M}})$ defines a linear (and continuous) operator from $L^2(\mathcal{M})$ to $L^2(\mathcal{M})$, which basically scales the coordinates of an input function f by the evaluation of γ on each corresponding eigenvalue of $-\Delta_{\mathcal{M}}$.

$\gamma(-\Delta_{\mathcal{M}})$ can be seen as a generalization of pseudo-differential operators of $\mathbb{R}^d$ on the Riemannian manifold $(\mathcal{M}, g)$. Indeed, a *pseudo-differential operator P of* $\mathbb{R}^d$ is an operator on real-valued functions of $\mathbb{R}^d$ whose action on a particular function φ is defined by

$$P\varphi = \mathscr{F}^{-1}\left[\boldsymbol{\xi} \in \mathbb{R}^d \mapsto p(\boldsymbol{\xi}) \cdot \mathscr{F}[\varphi](\boldsymbol{\xi})\right] \quad , \tag{7.2}$$

where p is a smooth function called symbol of P, whose derivatives are required to be polynomially bounded.

Dealing now with a Riemannian manifold (instead of $\mathbb{R}^d$), the notion of Fourier transform can be naturally extended by noticing that the Fourier transform of $\mathbb{R}^d$ actually corresponds to the decomposition of a function into the continuously indexed set of functions $\{\boldsymbol{x} \in \mathbb{R}^d \mapsto e^{\mathrm{i}\boldsymbol{x}^T\boldsymbol{\xi}}\}_{\boldsymbol{\xi}\in\mathbb{R}^d}$. It is straightforward to check that these functions actually are eigenfunctions of the negative Laplacian of $\mathbb{R}^d$, associated with eigenvalues $\{\|\boldsymbol{\xi}\|^2\}_{\boldsymbol{\xi}\in\mathbb{R}^d}$. Hence the Fourier transform in $\mathbb{R}^d$ can be interpreted as a decomposition of a function into a weighted "sum" of eigenfunctions of the Laplacian.

Extending now this observation to Riemannian manifolds, the notion of Fourier transform can hence be identified with the decomposition of a function into the countable basis of eigenfunctions of the Laplace-Beltrami operator. Denote then $\mathscr{F}_{\mathcal{M}} : L^2(\mathcal{M}) \to \ell^2(\mathbb{N})$ the map that associates to any $f \in L^2(\mathcal{M})$ its coordinates in the basis $\{e_k\}_{k\in\mathbb{N}}$:

$$\forall f \in L^2(\mathcal{M}), \quad \mathscr{F}_{\mathcal{M}}[f] = \{\langle e_k, f\rangle_{L^2(\mathcal{M})}\}_{k\in\mathbb{N}} \quad .$$

This operator is invertible and its inverse $\mathscr{F}_{\mathcal{M}}^{-1} : \ell^2(\mathbb{N}) \to L^2(\mathcal{M})$ is given by:

$$\forall \{c_k\}_{k\in\mathbb{N}} \in \ell^2(\mathbb{N}), \quad \mathscr{F}_{\mathcal{M}}^{-1}[\{c_k\}_{k\in\mathbb{N}}] = \sum_{k\in\mathbb{N}} c_k e_k \in L^2(\mathcal{M}) \quad .$$

Then the definition of the operator $\gamma(-\Delta_{\mathcal{M}})$ in Equation (7.1) can be written

$$\gamma(-\Delta_{\mathcal{M}}) = \mathscr{F}_{\mathcal{M}}^{-1}\left[\{\gamma(\lambda_k) \cdot \langle e_k, f\rangle_{L^2(\mathcal{M})}\}_{k\in\mathbb{N}}\right] \quad . \tag{7.3}$$

Equation (7.3) presents a form similar in all aspects to Equation (7.2). The function γ in Equation (7.3) plays the role of the symbol function in Equation (7.2), and functions defined on the continuous space $\mathbb{R}^d$ are replaced by countable sequences.

This observation justifies the parallel that is drawn between pseudo differential operators and the functional operators studied in this section. A more in-depth comparison between them is carried out in Appendix D.1.1, in the case where the Riemannian manifold considered is a bounded box of $\mathbb{R}^d$.

7.1.2 Generalized random fields of $L^2(\mathcal{M})$

General definitions and notions

A *generalized random field* (GeRF) $\mathcal{Z}$ on $\mathcal{M}$ is a linear and continuous functional that associates to any $\varphi \in \mathscr{C}^\infty(\mathcal{M})$ a random variable $\mathcal{Z}(\varphi) \in \mathbb{R}$ (Gelfand and Shilov, 1964). A GeRF $\mathcal{Z}$ is characterized by its *probability distribution*, which is the set of all joint distributions $F_{\varphi_1,\dots,\varphi_m}$ defined by

$$F_{\varphi_1,\dots,\varphi_m} : (a_1, \dots, a_m) \in \mathbb{R}^m \mapsto \mathbb{P}\left[\mathcal{Z}(\varphi_1) \leq a_1, \dots, \mathcal{Z}(\varphi_m) \leq a_m\right]$$

for any $m \geq 1$ and $\varphi_1, \dots, \varphi_m \in \mathscr{C}^\infty(\mathcal{M})$.

The *mean* of $\mathcal{Z}$ is the linear and continuous functional $\mu_{\mathcal{Z}}$ which associates to any $\varphi \in \mathscr{C}^\infty(\mathcal{M})$, the expectation of $\mathcal{Z}(\varphi)$:

$$\mu_{\mathcal{Z}}(\varphi) := \mathbb{E}[\mathcal{Z}(\varphi)], \quad \varphi \in \mathscr{C}^\infty(\mathcal{M}) \quad .$$

In particular $\mathcal{Z}$ is called zero-mean if $\forall \varphi \in \mathscr{C}^\infty(\mathcal{M})$, $\mu_{\mathcal{Z}}(\varphi) = 0$.

If the expectation of the product $\mathcal{Z}(\varphi_1)\mathcal{Z}(\varphi_2)$ exists for any $\varphi_1, \varphi_2 \in \mathscr{C}^\infty(\mathcal{M})$ and is continuous in (φ_1, φ_2), then the *covariance functional* $C_{\mathcal{Z}} : \mathscr{C}^\infty(\mathcal{M}) \times \mathscr{C}^\infty(\mathcal{M}) \to \mathbb{R}$ of $\mathcal{Z}$ is the positive definite functional defined as

$$C_{\mathcal{Z}}(\varphi_1, \varphi_2) := \mathrm{Cov}[\mathcal{Z}(\varphi_1), \mathcal{Z}(\varphi_2)] = \mathbb{E}[\mathcal{Z}(\varphi_1)\mathcal{Z}(\varphi_2)] - \mathbb{E}[\mathcal{Z}(\varphi_1)]\mathbb{E}[\mathcal{Z}(\varphi_2)], \quad \varphi_1, \varphi_2 \in \mathscr{C}^\infty(\mathcal{M}) \, .$$

Finally, the *characteristic functional* $\Psi_{\mathcal{Z}}$ of $\mathcal{Z}$ is the functional that associates to any $\varphi \in \mathscr{C}^\infty(\mathcal{M})$ the value of the characteristic function of $\mathcal{Z}(\varphi)$ at 1, namely

$$\Psi_{\mathcal{Z}} : \varphi \in \mathscr{C}^\infty(\mathcal{M}) \mapsto \mathbb{E}[e^{\mathrm{i}\mathcal{Z}(\varphi)}] \quad .$$

The characteristic function of a GeRF is continuous on $\mathscr{C}^\infty(\mathcal{M})$ and satisfies $\Psi_{\mathcal{Z}}(0) = 1$. Besides, Minlos' theorem ensures that the characteristic functional of a GeRF entirely characterizes its probability distribution (Gelfand and Shilov, 1964; Lang, 2007).

Gaussian GeRF and white noise

A GeRF $\mathcal{Z}$ is called *Gaussian GeRF*, or *generalized Gaussian field* (GeGF), if for any $m \geq 1$ and any linearly independent $\varphi_1, \ldots, \varphi_m \in \mathscr{C}^\infty(\mathcal{M})$, the random vector $(\mathcal{Z}(\varphi_1), \ldots, \mathcal{Z}(\varphi_m))^T$ is a non-singular Gaussian vector. The characteristic functional of a zero-mean GeGF $\mathcal{Z}$ is then given by (cf. Theorem A.4.4):

$$\forall \varphi \in \mathscr{C}^\infty(\mathcal{M}), \quad \Psi_{\mathcal{Z}}(\varphi) = e^{-\frac{1}{2}C_{\mathcal{Z}}(\varphi,\varphi)} \quad ,$$

where $C_{\mathcal{Z}}$ is once again the covariance functional of $\mathcal{Z}$. Conversely, given a continuous, symmetric and positive-definite bilinear form Q on $\mathscr{C}^\infty(\mathcal{M}) \times \mathscr{C}^\infty(\mathcal{M})$, the functional defined by

$$\varphi \in \mathscr{C}^\infty(\mathcal{M}) \mapsto e^{-\frac{1}{2}Q(\varphi,\varphi)} \quad ,$$

is the characteristic function of a GeGF with covariance functional Q (Gelfand and Shilov, 1964).

In particular, considering as bilinear form the inner product of $L^2(\mathcal{M})$, yields the functional

$$\varphi \in \mathscr{C}^\infty(\mathcal{M}) \mapsto e^{-\frac{1}{2}\langle \varphi, \varphi \rangle_{L^2(\mathcal{M})}} \quad . \tag{7.4}$$

Any GeRF with characteristic function given by Equation (7.4) is a GeGF called *Gaussian white noise on* $\mathcal{M}$. A characterization of Gaussian white noises based on the Hilbert space $L^2(\mathcal{M})$ is given by the following proposition.

Proposition 7.1.2. *Let $\{W_j\}_{j\in\mathbb{N}}$ be a sequence of independent, standard Gaussian variables.*

Then, the linear functional $\mathcal{W}$ defined over $L^2(\mathcal{M})$ by

$$\mathcal{W} : \varphi \in L^2(\mathcal{M}) \mapsto \sum_{j\in\mathbb{N}} W_j \langle \varphi, e_j \rangle_{L^2(\mathcal{M})} \tag{7.5}$$

is a Gaussian white noise on $\mathcal{M}$. In particular, it satisfies

$$\forall \varphi \in L^2(\mathcal{M}), \quad \mathbb{E}\left[\mathcal{W}(\varphi)\right] = 0 \tag{7.6}$$

and

$$\forall \varphi_1, \varphi_2 \in L^2(\mathcal{M}), \quad \mathrm{Cov}\left[\mathcal{W}(\varphi_1), \mathcal{W}(\varphi_2)\right] = \langle \varphi_1, \varphi_2 \rangle_{L^2(\mathcal{M})} \quad . \tag{7.7}$$

Proof. Note that given that $\mathscr{C}^\infty(\mathcal{M}) \subset L^2(\mathcal{M})$, $\mathcal{W}$ can be seen as a GeRF on $\mathcal{M}$.
Let $\varphi \in \mathscr{C}^\infty(\mathcal{M})$. Following from the mutual independence of the W_j, the characteristic function of $\mathcal{W}$ satisfies:

$$\Psi_{\mathcal{W}}(\varphi) = \mathbb{E}\left[e^{\mathrm{i}\sum_{j\in\mathbb{N}} W_j \langle \varphi, e_j\rangle_{L^2(\mathcal{M})}}\right] = \prod_{j\in\mathbb{N}} \mathbb{E}\left[e^{\mathrm{i}W_j\langle\varphi,e_j\rangle_{L^2(\mathcal{M})}}\right] = \prod_{j\in\mathbb{N}} \Psi_{\mathcal{N}(0,1)}(\langle\varphi, e_j\rangle_{L^2(\mathcal{M})}) \ ,$$

where $\Psi_{\mathcal{N}(0,1)}$ denotes the characteristic function of the standard Gaussian distribution, which is given by $\Psi_{\mathcal{N}(0,1)}(t) = e^{-t^2/2}$, $\forall t \in \mathbb{R}$. Hence,

$$\Psi_{\mathcal{W}}(\varphi) = \prod_{j\in\mathbb{N}} e^{-\frac{1}{2}\langle\varphi,e_j\rangle^2_{L^2(\mathcal{M})}} = e^{-\frac{1}{2}\sum_{j\in\mathbb{N}}\langle\varphi,e_j\rangle^2_{L^2(\mathcal{M})}} = e^{-\frac{1}{2}\langle\varphi,\varphi\rangle^2_{L^2(\mathcal{M})}} \ ,$$

which is the characteristic function of a Gaussian white noise. Hence $\mathcal{W}$ is Gaussian white noise.
Equations (7.6) and (7.7) then follow from the fact that the Gaussian white noise is a zero-mean generalized random process with covariance functional $\langle .,.\rangle_{L^2(\mathcal{M})}$; and by density of $\mathscr{C}^\infty(\mathcal{M})$ in $L^2(\mathcal{M})$. □

Seen as the functional defined in Proposition 7.1.2, the Gaussian white noise has several properties related to $L^2(\mathcal{M})$. For one, it is defined on $L^2(\mathcal{M})$ and not only on $\mathscr{C}^\infty(\mathcal{M})$. Moreover, for any $m \geq 1$, and for any $\varphi_1, \ldots, \varphi_m \in L^2(\mathcal{M})$, we have:

$$\begin{pmatrix} \mathcal{W}(\varphi_1) \\ \vdots \\ \mathcal{W}(\varphi_m) \end{pmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{pmatrix} \langle\varphi_1,\varphi_1\rangle_{L^2(\mathcal{M})} & \cdots & \langle\varphi_1,\varphi_m\rangle_{L^2(\mathcal{M})} \\ \vdots & \ddots & \vdots \\ \langle\varphi_n,\varphi_1\rangle_{L^2(\mathcal{M})} & \cdots & \langle\varphi_m,\varphi_m\rangle_{L^2(\mathcal{M})} \end{pmatrix}\right) \ ,$$

which means that $(\mathcal{W}(\varphi_1)\ldots\mathcal{W}(\varphi_m))^T$ defines a zero-mean Gaussian vector. Finally, note that

$$\forall\varphi \in L^2(\mathcal{M}), \quad \mathrm{Var}[\mathcal{W}(\varphi)] = \mathbb{E}\left[|\mathcal{W}(\varphi)|^2\right] = \|\varphi\|^2_{L^2(\mathcal{M})} < \infty \ .$$

Hence all random variables $\mathcal{W}(\varphi)$ have a finite variance.

$L^2(\mathcal{M})$-valued GeGF

We now introduce (and denote by) $L^2(\Omega, \mathcal{M})$ the set of $L^2(\mathcal{M})$-valued random variables defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and satisfying

$$\forall\mathcal{Z} \in L^2(\Omega,\mathcal{M}), \quad \mathbb{E}[\mathcal{Z}] = 0_{L^2(\mathcal{M})} \text{ and } \mathbb{E}[\|\mathcal{Z}\|^2_{L^2(\mathcal{M})}] < \infty \ . \tag{7.8}$$

In particular, this means that any $\mathcal{Z} \in L^2(\Omega,\mathcal{M})$ is almost surely in $L^2(\mathcal{M})$. This condition is actually enforced by Equation (7.8). Indeed, according to Markov's inequality (Stewart, 2009, Section 8.1), $\forall N \geq 1$, $\mathbb{P}\left[\|\mathcal{Z}\|^2_{L^2(\mathcal{M})} \geq N\right] \leq \mathbb{E}[\|\mathcal{Z}\|^2_{L^2(\mathcal{M})}]/N$. And taking the limit as $N \to \infty$ then gives $\mathbb{P}\left[\|\mathcal{Z}\|^2_{L^2(\mathcal{M})} = +\infty\right] = 1 - \mathbb{P}\left[\mathcal{Z} \in L^2(\mathcal{M})\right] = 0$. Consequently, any $\mathcal{Z} \in L^2(\Omega,\mathcal{M})$ can be represented in the basis $\{e_j\}_{j\in\mathbb{N}}$ as

$$\mathcal{Z} = \sum_{j\in\mathbb{N}} Z_j e_j \ , \tag{7.9}$$

where $Z_1, Z_2, \ldots$ are real-valued random variables satisfying $\mathbb{E}[Z_j] = 0$ and $\mathbb{E}[Z_j^2] < \infty$ (Tone, 2011).

$L^2(\Omega,\mathcal{M})$ is a Hilbert space when equipped with the scalar product $\langle .,.\rangle_{L^2(\Omega,\mathcal{M})}$ (and associated norm $\|.\|_{L^2(\Omega,\mathcal{M})}$) defined by:

$$\forall\mathcal{Z},\mathcal{Z}' \in L^2(\Omega,\mathcal{M}), \quad \langle\mathcal{Z},\mathcal{Z}'\rangle_{L^2(\Omega,\mathcal{M})} = \mathbb{E}\left[\langle\mathcal{Z},\mathcal{Z}'\rangle_{L^2(\mathcal{M})}\right] \ .$$

Note in particular that if $\mathcal{Z}$ and $\mathcal{Z}'$ are represented as in Equation (7.9), we have

$$\langle\mathcal{Z},\mathcal{Z}'\rangle_{L^2(\Omega,\mathcal{M})} = \sum_{j\in\mathbb{N}} \mathbb{E}[Z_j Z_j'] \quad \text{and} \quad \|\mathcal{Z}\|^2_{L^2(\Omega,\mathcal{M})} = \sum_{j\in\mathbb{N}} \mathbb{E}[Z_j^2] \ .$$

The next result introduces a class of GeGFs defined through the white noise that can be identified with elements of $L^2(\Omega, \mathcal{M})$.

Theorem 7.1.3. *Let $\{W_j\}_{j\in\mathbb{N}}$ be a sequence of independent standard Gaussian variables defining a Gaussian white noise $\mathcal{W}$ as in Proposition 7.1.2.*
For $\gamma : \mathbb{R}_+ \mapsto \mathbb{R}$ such that $\sum_{j\in\mathbb{N}} \gamma(\lambda_j)^2 < \infty$, denote $\gamma(-\Delta_{\mathcal{M}})\mathcal{W}$ the GeGF of $\mathcal{M}$ defined on $L^2(\mathcal{M})$ by

$$(\gamma(-\Delta_{\mathcal{M}})\mathcal{W})(\varphi) := \mathcal{W}\left(\gamma(-\Delta_{\mathcal{M}})\varphi\right), \quad \varphi \in L^2(\mathcal{M}) \quad , \tag{7.10}$$

where $\gamma(-\Delta_{\mathcal{M}})$ is the function of the Laplacian defined in Equation (7.1).

Then, $\gamma(-\Delta_{\mathcal{M}})\mathcal{W}$ can be identified with the element $\mathcal{Z} \in L^2(\Omega, \mathcal{M})$ defined by

$$\mathcal{Z} = \sum_{j\in\mathbb{N}} W_j \gamma(\lambda_j) e_j \quad , \tag{7.11}$$

through the linear functional of $L^2(\mathcal{M})$ defined by: $\varphi \in L^2(\mathcal{M}) \mapsto \langle \mathcal{Z}, \varphi \rangle_{L^2(\mathcal{M})}$.

Proof. Clearly, $\mathcal{Z}$ is an element of $L^2(\Omega, \mathcal{M})$ given that $\mathbb{E}[\mathcal{Z}] = 0_{L^2(\mathcal{M})}$ and

$$\|\mathcal{Z}\|^2_{L^2(\Omega,\mathcal{M})} = \mathbb{E}\left[\|\mathcal{Z}\|^2_{L^2(\mathcal{M})}\right] = \sum_{j\in\mathbb{N}} \gamma(\lambda_j)^2 < \infty \, .$$

We now show that the linear functional $\varphi \in L^2(\mathcal{M}) \mapsto \langle \mathcal{Z}, \varphi \rangle_{L^2(\mathcal{M})}$ is equal to $\gamma(-\Delta_{\mathcal{M}})\mathcal{W}$. Indeed, $\forall \varphi \in L^2(\mathcal{M})$, we have from Proposition 7.1.2:

$$\begin{aligned}(\gamma(-\Delta_{\mathcal{M}})\mathcal{W})(\varphi) &= \mathcal{W}\left(\sum_{j\in\mathbb{N}} \gamma(\lambda_j)\langle e_j, \varphi\rangle_{L^2(\mathcal{M})} e_j\right) \\ &= \sum_{j\in\mathbb{N}} W_j \gamma(\lambda_j)\langle e_j, \varphi\rangle_{L^2(\mathcal{M})} = \langle \mathcal{Z}, \varphi\rangle_{L^2(\mathcal{M})} \, ,\end{aligned}$$

which concludes the proof. □

From now on, GeGFs of the form $\gamma(-\Delta_{\mathcal{M}})\mathcal{W}$ will be directly identified with their representation $\mathcal{Z}$ in $L^2(\Omega, \mathcal{M})$, and we will write them as:

$$\mathcal{Z} = \gamma(-\Delta_{\mathcal{M}})\mathcal{W} = \sum_{j\in\mathbb{N}} W_j \gamma(\lambda_j) e_j \quad , \tag{7.12}$$

where $\{W_j\}_{j\in\mathbb{N}}$ is a sequence of independent, standard Gaussian variables. As such, they are considered as linear applications that map $L^2(\mathcal{M})$ to zero-mean Gaussian variables such that

$$\forall \varphi \in L^2(\mathcal{M}), \quad \mathcal{Z}(\varphi) = \sum_{j\in\mathbb{N}} W_j \gamma(\lambda_j)\langle e_j, \varphi\rangle_{L^2(\mathcal{M})} \quad ,$$

and

$$\begin{aligned}\forall u, v \in L^2(\mathcal{M}), \quad \mathrm{Cov}[\mathcal{Z}(u), \mathcal{Z}(v)] &= \langle \gamma(-\Delta_{\mathcal{M}})u, \gamma(-\Delta_{\mathcal{M}})v\rangle_{L^2(\mathcal{M})} \\ &= \sum_{j\in\mathbb{N}} \gamma(\lambda_j)^2 \langle e_j, u\rangle_{L^2(\mathcal{M})}\langle e_j, v\rangle_{L^2(\mathcal{M})} \quad .\end{aligned} \tag{7.13}$$

In particular, γ will be taken to be a non-negative square-integrable function on $\mathbb{R}_+$, to ensure that Equation (7.12) is well-defined. In the next section, the statistical properties of such fields, and in particular their covariance, are investigated and related to those of usual random fields.

7.2 Covariance properties of generalized Gaussian fields

The aim of this section is to show how the covariance properties of the GeGFs defined in the previous section by Equation (7.12) relate to the usual description of the covariance properties of

random fields of $\mathbb{R}^d$. In particular, we show they can basically be seen as random fields defined on the manifold, whose spectral density is given by γ and with local anisotropies defined by the Riemannian metric.

To come up with these conclusions, we first consider the case where the Riemannian manifold $(\mathcal{M}, g)$ is a compact domain of $\mathbb{R}^d$ endowed with the Euclidean metric (cf. Example 6.2.1). This allows to draw a direct parallel between GeGFs defined on this trivial manifold and the so-called Karhunen–Loève expansion of random fields. In particular, we deduce a practical interpretation of γ. Then, this same manifold is endowed with a Riemannian metric to derive the conclusion on local anisotropies.

7.2.1 Generalized random fields and Karhunen–Loève expansion

In this subsection, we draw a parallel between the definition of GeGFs we proposed in Section 7.1.2 and the Karhunen–Loève expansion of Gaussian random fields in the particular case where the domain we consider is a hypercube of $\mathbb{R}^d$. We first recall the definition of this expansion.

Let B denote the unit hypercube of $\mathbb{R}^d$ and let Z be a zero-mean Gaussian random field defined on B. Denote $c_Z : B \times B \to \mathbb{R}$ the covariance function of Z, i.e.

$$\forall \boldsymbol{x}, \boldsymbol{y} \in B, \quad c_Z(\boldsymbol{x}, \boldsymbol{y}) = \mathrm{Cov}[Z(\boldsymbol{x}), Z(\boldsymbol{y})] = \mathbb{E}[Z(\boldsymbol{x})Z(\boldsymbol{y})] \quad .$$

Denote by $L^2(B)$ the set of square-integrable functions of B. We can associate to the covariance function c_Z an operator $\mathcal{C}_Z : L^2(B) \to L^2(B)$, called *covariance operator*, which maps any $\varphi \in L^2(B)$ to a function $\mathcal{C}_Z[\varphi] \in L^2(B)$ given by

$$\mathcal{C}_Z[\varphi](\boldsymbol{x}) = \int_B c_Z(\boldsymbol{x}, \boldsymbol{y})\varphi(\boldsymbol{y})d\boldsymbol{y} \quad . \tag{7.14}$$

Similarly as what was done for the Laplacian (cf. Section 6.5.2), a function $\phi \in L^2(B)$ is called eigenfunction of $\mathcal{C}_Z$ with associated eigenvalue $\eta \in \mathbb{R}$ if it satisfies

$$\mathcal{C}_Z[\phi] = \eta\phi \quad . \tag{7.15}$$

The Karhunen–Loève theorem then states the following results (Lindgren, 2012).

Theorem 7.2.1 (Karhunen–Loève theorem)**.** *Let Z be a (continuous in quadratic mean) Gaussian random field with covariance operator $\mathcal{C}_Z$, defined on the hypercube B.*

On one hand, there exists a complete (countable) orthogonal[1] basis of $L^2(B)$ consisting of eigenfunctions $\{\phi_k\}_{k\in\mathbb{N}}$ of $\mathcal{C}_Z$.

On the other hand, if $\{\eta_k\}_{k\in\mathbb{N}}$ denotes the eigenvalues associated with $\{\phi_k\}_{k\in\mathbb{N}}$, then $\forall k \in \mathbb{N}$, $\eta_k \geq 0$ and Z can be decomposed as

$$Z = \sum_{k\in\mathbb{N}} W_k\sqrt{\eta_k}\phi_k \quad , \tag{7.16}$$

where $\{W_k\}_{k\in\mathbb{N}}$ is a set of zero-mean uncorrelated (Gaussian) random variables with unit variance. Equation (7.16) *is called the Karhunen–Loève expansion of Z.*

Remark 7.2.1. Note that Z can be identified with a zero-mean GeGF[2] $\mathcal{Z}$ with covariance functional $C_{\mathcal{Z}}$ (cf. Section 7.1.2) given by:

$$\forall u, v \in \mathscr{C}^\infty(B), \quad C_{\mathcal{Z}}(u, v) = \int_B c_Z(\boldsymbol{x}, \boldsymbol{y})u(\boldsymbol{x})v(\boldsymbol{y})d\boldsymbol{y} \quad .$$

Then the eigenfunctions ϕ_k and eigenvalues η_k of the covariance operator $\mathcal{C}_Z$ also correspond to eigenfunctions and eigenvalues of the covariance functional $C_{\mathcal{Z}}$ in the sense that

$$\forall u \in \mathscr{C}^\infty(B), \quad C_{\mathcal{Z}}(\phi_k, u) = \eta_k\langle\phi_k, u\rangle_{L^2(B)} \quad .$$

Hence, Theorem 7.2.1 can also be stated using the covariance functional instead of the covariance operator.

[1] For the usual inner product on $L^2(B)$.

We now circle back to the class of GeGFs considered in this work and characterized in Theorem 7.1.3. We show in particular that the expansion in Equation (7.11) can be identified with the Karhunen–Loève expansion of the GeGF.

Indeed, let $\mathcal{Z}$ be now a GeGF defined as in Equation (7.11). Then, $\forall u \in \mathscr{C}^\infty(B)$, the eigenfunctions $\{e_k\}_{k\in\mathbb{N}}$ of the negative Laplace-Beltrami operator on B satisfy

$$\begin{aligned} C_{\mathcal{Z}}(e_k, u) :&= \mathrm{Cov}[\mathcal{Z}(e_k), \mathcal{Z}(u)] = \langle \gamma(-\Delta_B)e_k, \gamma(-\Delta_B)u\rangle_{L^2(B)} \\ &= \left\langle \sum_{l\in\mathbb{N}} \gamma(\lambda_l)\langle e_l, e_k\rangle_{L^2(B)} e_l, \sum_{l'\in\mathbb{N}} \gamma(\lambda_{l'})\langle e_{l'},\ u\rangle_{L^2(B)} e_{l'} \right\rangle_{L^2(B)} = \gamma(\lambda_k)^2 \langle e_k, u\rangle_{L^2(B)} \quad . \end{aligned}$$

Hence, the eigenfunctions $\{e_k\}_{k\in\mathbb{N}}$ of the negative Laplace-Beltrami operator are eigenfunctions of the covariance functional of $\mathcal{Z}$ with associated eigenvalues $\{\gamma(\lambda_k)^2\}_{k\in\mathbb{N}}$. Then, following Theorem 7.2.1, the expansion of $\mathcal{Z}$ in Equation (7.11) corresponds to a Karhunen–Loève expansion.

Hence, using the formalism of Karhunen–Loève expansions, we identify the GeGF $\mathcal{Z}$ with the zero-mean random function series given by

$$\mathcal{Z}(\boldsymbol{x}) = \sum_{j\in\mathbb{N}} W_j \gamma(\lambda_j) e_j(\boldsymbol{x}), \quad \boldsymbol{x} \in B_D \quad ,$$

where $\{W_j\}_{j\in\mathbb{N}}$ denotes a sequence of independent standard Gaussian variables, and $\{\lambda_j\}_{j\in\mathbb{N}}$ (resp. $\{e_j\}_{j\in\mathbb{N}}$) are the eigenvalues (resp. eigenfunctions) of the Laplacian on B_D. Its covariance function is then obtained as

$$\mathrm{Cov}[\mathcal{Z}(\boldsymbol{x}), \mathcal{Z}(\boldsymbol{y})] = \sum_{j\in\mathbb{N}}\sum_{k\in\mathbb{N}} \mathbb{E}[W_k W_j]\gamma(\lambda_j)\gamma(\lambda_k) e_j(\boldsymbol{x}) e_k(\boldsymbol{y})), \quad \boldsymbol{x}, \boldsymbol{y} \in B_D \quad .$$

which gives

$$\mathrm{Cov}[\mathcal{Z}(\boldsymbol{x}), \mathcal{Z}(\boldsymbol{y})] = \sum_{j\in\mathbb{N}} \gamma(\lambda_j)^2 e_j(\boldsymbol{x}) e_j(\boldsymbol{y}), \quad \boldsymbol{x}, \boldsymbol{y} \in B_D \quad . \tag{7.17}$$

In their work on Gaussian process regression, Solin and Särkkä (2014), show that away from the boundary of B_D, the covariance function defined by Equation (7.17) yields a good approximation of the isotropic covariance function defined as the inverse Fourier transform of the function γ^2, which we denote C_0:

$$C_0 = \mathcal{F}^{-1}[\gamma^2] \quad . \tag{7.18}$$

Another proof is provided by (Huang et al., 2001), and relies on the identification of the Karhunen–Loève expansion of $\mathcal{Z}$ with the discretized spectral representation of a Gaussian random field with covariance function C_0.

Hence we have that for points $\boldsymbol{x}, \boldsymbol{y} \in B_D$ away from the boundaries,

$$\mathrm{Cov}[\mathcal{Z}(\boldsymbol{x}), \mathcal{Z}(\boldsymbol{y})] = C_0(\|\boldsymbol{x} - \boldsymbol{y}\|_2) \quad . \tag{7.19}$$

where C_0 is given by Equation (7.18).

Remark 7.2.2. The link between our definition of GeGFs and the Karhunen–Loève expansion exhibited in this subsection actually provides an additional justification to the fact the Laplacian functions can be considered as the transposition of pseudo-differential operators to compact domains of $\mathbb{R}^d$ (cf. Section 7.1.1).

Indeed, following a characterization from (Lang and Potthoff, 2011), a stationary field with covariance C_0 on $\mathbb{R}^d$ can be identified with a generalized random field of $\mathbb{R}^d$ defined by

$$Z = \mathcal{L}_\gamma \mathcal{W} \quad , \tag{7.20}$$

where once again C_0 and γ^2 are linked through Equation (7.18), $\mathcal{L}_\gamma$ denotes the pseudo-differential with symbol function γ and $\mathcal{W}$ is a Gaussian white noise on $\mathbb{R}^d$. In particular,

[2]This identification can actually be seen as (formally) defining the GeGF $\mathcal{Z}$ as the map $\mathcal{Z} : u \in \mathscr{C}^\infty(B) \mapsto \int_B u(\boldsymbol{x}) Z(\boldsymbol{x}) d\boldsymbol{x}$.

Z is therefore seen as a linear application mapping $\mathscr{C}_c^\infty(\mathbb{R}^d)$ (the set of compactly-supported smooth functions of $\mathbb{R}^d$) to zero-mean Gaussian variables such that

$$\forall u, v \in \mathscr{C}^\infty(\mathbb{R}^d), \quad \mathrm{Cov}[Z(u), Z(v)] = \langle \mathcal{L}_\gamma u, \mathcal{L}_\gamma v \rangle_{L^2(\mathbb{R}^d)} \quad . \tag{7.21}$$

Comparing Equation (7.12) with Equation (7.20) and Equation (7.13) with Equation (7.21) then allows to conclude that the definition Laplacian functions play the exact same role as the pseudo-differential operators do when defining stationary fields.

7.2.2 Generalized random fields on a compact domain of $\mathbb{R}^d$ equipped with a metric

In this subsection, we use the formalism presented in Section 6.6. $B_R \subset \mathbb{R}^d$ denotes a compact and connected set of $\mathbb{R}^d$, with (piecewise) smooth boundary, called reference configuration. $\Phi : B_R \to B_D = \Phi(B)$ denotes a diffeomorphism that maps any point $\boldsymbol{p} \in B_R$ to a point $\boldsymbol{q} = \Phi(\boldsymbol{p}) \in B_D$ in the set $B_D \subset \mathbb{R}^d$, which is called deformed configuration.

In particular, the body B_R is seen as a compact d-submanifold B_R of $\mathbb{R}^d$, equipped with a Riemannian metric g, represented by a field of positive definite matrices $\{\boldsymbol{G}(\boldsymbol{p})\}_{\boldsymbol{p}\in B_R}$. In particular, $B_D = \Phi(B_R)$, and we assume that Φ is linked to g through Equation (6.25). Let then Δ_{B_R} be the Laplace-Beltrami operator on (B_R, g).

Following Equation (7.12), we now define a GeGF $\mathcal{Z}_R$ on B_R through

$$\mathcal{Z}_R = \sum_{k\in\mathbb{N}} W_{\boldsymbol{k}} \gamma(\lambda_k^R) e_k^R \quad ,$$

where $\{W_k\}_{k\in\mathbb{N}}$ is a set of independent standard Gaussian variables, and $\{\lambda_k^R\}_{k\in\mathbb{N}}$ (resp. $\{e_k^R\}_{k\in\mathbb{N}}$) are the eigenvalues (resp. eigenfunctions) of $-\Delta_{B_R}$.

Let $-\Delta_{\mathbb{R}^d}$ denote the classical Laplacian of $\mathbb{R}^d$, defined on functions of B_D. Note that, following Equation (6.26), the eigenfunctions of $-\Delta_{B_R}$ satisfy

$$\forall k \in \mathbb{N}, \quad -\Delta_{B_R} e_k^R = \left(-\Delta_{\mathbb{R}^d}(e_k^R \circ \Phi^{-1})\right) \circ \Phi = \lambda_k^R e_k^R \quad .$$

And therefore the function $e_k^D := e_k^R \circ \Phi^{-1}$ is an eigenfunction of $-\Delta_{\mathbb{R}^d}$ on B_D, associated with the eigenvalue $\lambda_k^D := \lambda_k^R$. Hence,

$$\mathcal{Z}_D := \mathcal{Z}_R \circ \Phi^{-1} = \sum_{k\in\mathbb{N}} W_k \gamma(\lambda_k^R) e_k^R \circ \Phi^{-1} = \sum_{k\in\mathbb{N}} W_k \gamma(\lambda_k^D) e_k^D$$

defines a GeGF on B_D. In particular, following from Section 7.2.1, $\mathcal{Z}_D$ can be seen as an isotropic stationary random field with spectral density γ^2 and covariance function satisfying Equation (7.19).

Consider now two points $\boldsymbol{p} \in B_R$ and $\boldsymbol{p}+d\boldsymbol{p} \in B_R$ separated by an infinitesimal displacement vector $d\boldsymbol{p} \in \mathbb{R}^d$. Following the results of Section 6.6.1 we have,

$$\begin{aligned} \mathrm{Cov}[\mathcal{Z}_R(\boldsymbol{p}), \mathcal{Z}_R(\boldsymbol{p}+d\boldsymbol{p})] &= \mathrm{Cov}[\mathcal{Z}_D(\Phi(\boldsymbol{p})), \mathcal{Z}_D(\Phi(\boldsymbol{p}+d\boldsymbol{p}))] \\ &= C(\|\Phi(\boldsymbol{p}+d\boldsymbol{p}) - \Phi(\boldsymbol{p})\|_2) = C\left(\sqrt{d\boldsymbol{p}\boldsymbol{G}(\boldsymbol{p})d\boldsymbol{p}}\right) \quad . \end{aligned} \tag{7.22}$$

Besides, $\boldsymbol{G}(\boldsymbol{p})$ being a positive-definite and symmetric matrix, it can be diagonalized as

$$\boldsymbol{G}(\boldsymbol{p}) = \boldsymbol{R}(\boldsymbol{p})^T \mathrm{Diag}(\rho_1(\boldsymbol{p}), \ldots, \rho_d(\boldsymbol{p})) \boldsymbol{R}(\boldsymbol{p}) \quad , \tag{7.23}$$

where $\boldsymbol{R}(\boldsymbol{p}) \in \mathscr{M}_d(\mathbb{R})$ is an orthogonal matrix (i.e. $\boldsymbol{R}(\boldsymbol{p})^T\boldsymbol{R}(\boldsymbol{p}) = \boldsymbol{R}(\boldsymbol{p})\boldsymbol{R}(\boldsymbol{p})^T = \boldsymbol{I}_d$) and $\rho_1(\boldsymbol{p}), \ldots, \rho_d(\boldsymbol{p}) > 0$. For $d \in \{2, 3\}$, whenever $\det \boldsymbol{R}(\boldsymbol{p}) = 1$, $\boldsymbol{R}(\boldsymbol{p})$ represents a rotation transformation[3]. In this case, Equation (7.22) becomes

$$\mathrm{Cov}[\mathcal{Z}_R(\boldsymbol{p}), \mathcal{Z}_R(\boldsymbol{p}+d\boldsymbol{p})] = C(\|\mathrm{Diag}(1/\sqrt{\rho_1(\boldsymbol{p})}, \ldots, 1/\sqrt{\rho_d(\boldsymbol{p})})\boldsymbol{R}(\boldsymbol{p})d\boldsymbol{p}\|_2) \quad . \tag{7.24}$$

[3]If $\det \boldsymbol{R}(\boldsymbol{p}) = -1$, $\boldsymbol{R}(\boldsymbol{p})$ represents a reflection transformation (Friedberg et al., 2003).

Hence, the covariance of $\mathcal{Z}_R$ around $\boldsymbol{p} \in B_R$ acts basically like an isotropic field with covariance C defined on a neighborhood of $\boldsymbol{p}$ deformed by the rotation induced by $\boldsymbol{R}(\boldsymbol{p})$ and dilatation with factors $1/\sqrt{\rho_1(\boldsymbol{p})}, \dots, 1/\sqrt{\rho_d(\boldsymbol{p})}$ [4].

Conversely, given fields of axis lengths $\{1/\sqrt{\rho_1(\boldsymbol{p})}\}_{\boldsymbol{p}\in B_R}$, ..., $\{1/\sqrt{\rho_d(\boldsymbol{p})}\}_{\boldsymbol{p}\in B_R}$ and of rotation matrices $\{\boldsymbol{R}(\boldsymbol{p})\}_{\boldsymbol{p}\in B_R}$ and defined across a domain B_R, a GeGF that behaves locally as in Equation (7.22) can be generated and characterized as a GeGF defined on the Riemannian manifold obtained by equipping B_R with a Riemannian metric defined by Equation (7.23). This idea will be discussed in Section 7.4.2 and leveraged to model a class of non-stationary fields defined on B_R, called *Gaussian random fields with local anisotropies*.

Note in particular, there is no need to actually specify the transformation deformation Φ that was associated to the metric in our formalism, as it does not intervene in the characterization of the metric or the pseudo-differential operators defining the fields once $\{\boldsymbol{G}(\boldsymbol{p})\}_{\boldsymbol{p}\in B_R}$ is fixed.

7.3 Discretization of generalized Gaussian fields

In the last section, we showed how the covariance of the GeGFs defined through Equation (7.12) could be linked to the covariance function of a field whose spectral density is γ^2. Besides, endowing a manifold with a Riemannian metric proved to be a natural way to define local anisotropies on the manifold.

We now aim at computing numerical approximations of such fields using a discretization of the functions of the Laplacian and of the resulting GeGFs we have been working with. It leads to their approximation by a weighted sum of user-defined deterministic functions called basis functions and defined on the manifold. The discretization we propose, based on the Ritz–Galerkin approximation theory, can be seen as an extension to general functions of the Laplacians and to Riemannian manifolds of the approach proposed by Bolin et al. (2018) to derive numerical approximation results for fractional elliptic stochastic partial differential equations.

Our main contribution is the derivation of Theorem 7.3.5 which provides a complete characterization of the weights of such an approximation. The study of the convergence properties of these approximations is delayed to the next chapter, for a particular set of basis functions.

The following notations are adopted in this section. Let $H(\mathcal{M})$ denote either $H^1(\mathcal{M})$ if a closed or a Neumann Laplacian is considered, or $H^1_0(\mathcal{M})$ if a Dirichlet Laplacian is considered (cf. Section 6.5.3). Take $n \geq 1$ and $\{\psi_k\}_{1\leq k\leq n}$ a family of linearly independent functions of $H(\mathcal{M})$. $V_n \subset H(\mathcal{M})$ denotes its linear span:

$$V_n = \operatorname{span}\{\psi_k : k \in [\![1, n]\!]\} \quad .$$

In particular, V_n is a n-dimensional vector space included in $H(\mathcal{M})$.

7.3.1 Ritz–Galerkin discretization of functions of the Laplacian

Let $\varphi \in H(\mathcal{M})$. Following the Ritz–Galerkin approximation approach (Brenner and Scott, 2007; Strang and Fix, 1973), the discretization of $-\Delta_{\mathcal{M}}\varphi$ over a n-dimensional space $V_n \subset H(\mathcal{M})$ is defined as the element of V_n, which we denote $-\Delta_n\varphi \in V_n$, that agrees with $-\Delta_{\mathcal{M}}\varphi$ over V_n. Formally, and following the definition of $-\Delta_{\mathcal{M}}\varphi$ provided in Section 6.5.3, $-\Delta_n\phi$ is defined as the element of V_n satisfying:

$$\forall v \in V_n, \quad \langle -\Delta_n\phi, v\rangle_{L^2(\mathcal{M})} = \langle \nabla_{\mathcal{M}}\phi, \nabla_{\mathcal{M}} v\rangle_{L^2(\mathcal{M})} \quad . \tag{7.25}$$

Consider now the operator $-\Delta_n$ that associates to any $\varphi \in H(\mathcal{M})$ its discretization $-\Delta_n\varphi \in V_n$ as defined by Equation (7.25). $-\Delta_n$ is called the *Ritz–Galerkin approximation* of the operator $-\Delta_{\mathcal{M}}$. In particular, if $\{f_k\}_{1\leq k\leq n}$ denotes *any* orthonormal basis of V_n (with respect to the scalar product $\langle\cdot,\cdot\rangle_{L^2(\mathcal{M})}$), $-\Delta_n$ satisfies

$$\begin{aligned} -\Delta_n : V_n &\to V_n \\ \varphi &\mapsto -\Delta_n\varphi = \sum_{k=1}^{n} \langle \nabla_{\mathcal{M}} f_k, \nabla_{\mathcal{M}}\varphi\rangle_{L^2(\mathcal{M})}\, f_k \end{aligned} \quad . \tag{7.26}$$

[4] This is actually equivalent to saying that around $\boldsymbol{p} \in B_R$, $\mathcal{Z}_R$ acts like a stationary field with geometric anisotropy (Chilès and Delfiner, 2012)

Let $\boldsymbol{C}$ and $\boldsymbol{R}$ be the n-matrices respectively called *mass matrix* and *stiffness matrix*, and defined by

$$\begin{aligned}\boldsymbol{C} &= \left[\langle \psi_k, \psi_l \rangle_{L^2(\mathcal{M})}\right]_{1\leq k,l\leq n} \quad , \\ \boldsymbol{R} &= \left[\langle \nabla_{\mathcal{M}}\psi_k, \nabla_{\mathcal{M}}\psi_l \rangle_{L^2(\mathcal{M})}\right]_{1\leq k,l\leq n} \quad .\end{aligned} \tag{7.27}$$

Lemma 7.3.1. *Let $\boldsymbol{C}$ and $\boldsymbol{R}$ be the matrices defined in Equation* (7.27).
Then, $\boldsymbol{C}$ is a symmetric positive definite matrix and $\boldsymbol{R}$ is a symmetric positive semi-definite matrix.

Proof. On one hand, note that $\boldsymbol{C}$ is symmetric since the functions $\{\psi_k\}_k$ are real-valued. Also, $\forall \boldsymbol{x} \in \mathbb{R}^n$,

$$\boldsymbol{x}^T\boldsymbol{C}\boldsymbol{x} = \sum_{k=1}^{n}\sum_{l=1}^{n} x_k \langle \psi_k, \psi_l \rangle_{L^2(\mathcal{M})} x_l = \| \sum_{k=1}^{n} x_k \psi_k \|^2_{L^2(\mathcal{M})} \geq 0 \ .$$

Given that the functions $\{\psi_k\}_k$ are linearly independent, this quantity is zero only if $\boldsymbol{x} = \boldsymbol{0}$. Hence, $\boldsymbol{C}$ is positive definite.
On the other hand, $\boldsymbol{R}$ is symmetric by definition of its entries (cf. Corollary 6.5.2). And, $\forall \boldsymbol{x} \in \mathbb{R}^n$,

$$\boldsymbol{x}^T\boldsymbol{R}\boldsymbol{x} = \langle \sum_{k=1}^{n} x_k \nabla_{\mathcal{M}}\psi_k, \sum_{l=1}^{n} x_l \nabla_{\mathcal{M}}\psi_l \rangle_{L^2(\mathcal{M})} = \| \sum_{k=1}^{n} x_k \nabla_{\mathcal{M}}\psi_k \|^2_{L^2(\mathcal{M})} \geq 0$$

Hence $\boldsymbol{G}$ is positive semi-definite. □

Remark 7.3.1. Following the proof of Lemma 7.3.1, note that $\boldsymbol{R}$ is positive definite (and therefore invertible) whenever $\forall \varphi \in V_n$, $\nabla_{\mathcal{M}}\varphi = 0_{L^2(\mathcal{M})} \Rightarrow \varphi = 0_{L^2(\mathcal{M})}$.

We denote by $\boldsymbol{C}^{1/2}$ the principal square-root[5] of the mass matrix $\boldsymbol{C}$. In particular, $\boldsymbol{C}^{1/2}$ is invertible and we denote by $\boldsymbol{C}^{-1/2}$ its inverse. The following result provides a link between the matrices $\boldsymbol{C}$ and $\boldsymbol{R}$ and the endomorphism $-\Delta_n$ of V_n.

Theorem 7.3.2. *Let $\{\psi_k\}_{1\leq k\leq n}$ be a family of linearly independent functions of $H(\mathcal{M})$, satisfying Dirichlet or Neumann boundary conditions whenever $\partial\mathcal{M} \neq \emptyset$. Let V_n denote its linear span.*

Then the endomorphism $-\Delta_n$ defined by Equation (7.26) *is diagonalizable and its eigenvalues are those of the matrix $\boldsymbol{S}$ defined by*

$$\boldsymbol{S} = \boldsymbol{C}^{-1/2}\boldsymbol{R}\boldsymbol{C}^{-1/2} \quad , \tag{7.28}$$

where the matrices $\boldsymbol{C}$ and $\boldsymbol{R}$ are defined in Equation (7.27) *and $\boldsymbol{C}^{-1/2}$ is the inverse of the principal square-root of $\boldsymbol{C}$.*
In particular, the application $E : \mathbb{R}^n \to V_n$, defined by

$$E : \boldsymbol{v} \in \mathbb{R}^n \mapsto \sum_{k=1}^{n} [\boldsymbol{C}^{-1/2}\boldsymbol{v}]_k \psi_k \quad , \tag{7.29}$$

is an isometric isomorphism that maps the eigenvectors of $\boldsymbol{S}$ to the eigenfunctions of $-\Delta_n$.

Proof. Note first that $\boldsymbol{S}$ is real symmetric and is therefore diagonalizable. Take then λ an eigenvalue of $\boldsymbol{S}$ and denote $\boldsymbol{v} \neq \boldsymbol{0}$ an associated eigenvector. Then,

$$\boldsymbol{S}\boldsymbol{v} = \boldsymbol{C}^{-1/2}\boldsymbol{R}\boldsymbol{C}^{-1/2}\boldsymbol{v} = \lambda\boldsymbol{v} = \lambda\boldsymbol{C}^{1/2}\boldsymbol{C}^{-1/2}\boldsymbol{v} \quad ,$$

and so, $\boldsymbol{R}\boldsymbol{u} = \lambda\boldsymbol{C}\boldsymbol{u}$ where $\boldsymbol{u} = \boldsymbol{C}^{-1/2}\boldsymbol{v}$. Hence, using Equation (7.27),

$$\forall k \in [\![1, n]\!], \quad \sum_{l=1}^{n} \langle \nabla_{\mathcal{M}}\psi_k, \nabla_{\mathcal{M}}\psi_l \rangle_{L^2(\mathcal{M})} u_l = \lambda \sum_{l=1}^{n} \langle \psi_k, \psi_l \rangle_{L^2(\mathcal{M})} u_l \quad ,$$

[5] Hence, $\boldsymbol{C}^{1/2}$ is obtained by applying the square-root function to the eigenvalues of $\boldsymbol{C}$, in the same way as graph filters were defined (cf. Section 1.3.5).

which gives using Equation (7.29),

$$\forall k \in [\![1,n]\!], \quad \langle \nabla_{\mathcal{M}} \psi_k, \nabla_{\mathcal{M}} E(\boldsymbol{v}) \rangle_{L^2(\mathcal{M})} = \lambda \langle \psi_k, E(\boldsymbol{v}) \rangle_{L^2(\mathcal{M})} \quad . \tag{7.30}$$

Note then that $\{\psi_k\}_k$ is also a basis of V_n as it is a family of linearly independent functions spanning V_n. Denote then $\boldsymbol{A} \in \mathscr{M}_n(\mathbb{R})$ the invertible change-of-basis matrix between $\{\psi_k\}_k$ and the orthonormal basis $\{f_k\}_k$ of V_n in Equation (7.26). In particular, $\boldsymbol{A}$ satisfies

$$\forall k \in [\![1,n]\!], \quad \psi_k = \sum_{l=1}^{n} A_{kl} f_l \quad .$$

Then Equation (7.30) can be written

$$\boldsymbol{A} \begin{pmatrix} \langle \nabla_{\mathcal{M}} f_1, \nabla_{\mathcal{M}} E(\boldsymbol{v}) \rangle_{L^2(\mathcal{M})} \\ \vdots \\ \langle \nabla_{\mathcal{M}} f_n, \nabla_{\mathcal{M}} E(\boldsymbol{v}) \rangle_{L^2(\mathcal{M})} \end{pmatrix} = \lambda \boldsymbol{A} \begin{pmatrix} \langle f_1, E(\boldsymbol{v}) \rangle_{L^2(\mathcal{M})} \\ \vdots \\ \langle f_n, E(\boldsymbol{v}) \rangle_{L^2(\mathcal{M})} \end{pmatrix} \quad .$$

Multiplying both members of this equality by $\boldsymbol{A}^{-1}$ then yields that

$$\forall k \in [\![1,n]\!], \quad \langle \nabla_{\mathcal{M}} f_k, \nabla_{\mathcal{M}} E(\boldsymbol{v}) \rangle_{L^2(\mathcal{M})} = \lambda \langle f_k, E(\boldsymbol{v}) \rangle_{L^2(\mathcal{M})} \quad .$$

And so, given that $E(\boldsymbol{v}) \in V_n$,

$$-\Delta_n E(\boldsymbol{v}) = \sum_{k=1}^{n} \langle \nabla_{\mathcal{M}} f_k, \nabla_{\mathcal{M}} E(\boldsymbol{v}) \rangle_{L^2(\mathcal{M})} f_k = \lambda \sum_{k=1}^{n} \langle f_k, E(\boldsymbol{v}) \rangle_{L^2(\mathcal{M})} f_k = \lambda E(\boldsymbol{v}) \quad .$$

Therefore λ is an eigenvalue of $-\Delta_n$ and E maps the eigenvectors of $\boldsymbol{S}$ to the eigenfunctions of $-\Delta_n$.
Note then that, $\forall \boldsymbol{x} \in \mathbb{R}^n$,

$$\|E(\boldsymbol{x})\|^2_{L^2(\mathcal{M})} = \sum_{k=1}^{n} \sum_{l=1}^{n} [\boldsymbol{C}^{-1/2}\boldsymbol{x}]_k \langle \psi_k, \psi_l \rangle_{L^2(\mathcal{M})} [\boldsymbol{C}^{-1/2}\boldsymbol{x}]_l = \left(\boldsymbol{C}^{-1/2}\boldsymbol{x}\right)^T \boldsymbol{C}\boldsymbol{C}^{-1/2}\boldsymbol{x} = \|\boldsymbol{x}\|_2^2 \, .$$

Hence, given that it is also linear, E is an isometry between $\mathbb{R}^n$ (with the metric $\|.\|_2$) and V_n (with the metric $\|.\|_{L^2(\mathcal{M})}$). Consequently E is injective: indeed, $\forall \boldsymbol{x} \in \mathbb{R}^n$, $E(\boldsymbol{x}) = 0 \Rightarrow \|\boldsymbol{x}\|_2^2 = \|E(\boldsymbol{x})\|^2_{L^2(\mathcal{M})} = 0$ and so, $\boldsymbol{x} = 0$. And finally, using the rank–nullity theorem Friedberg et al. (2003), E is bijective (as an injective application between two vector spaces with same dimension). □

Denote by $\{\lambda_{k,n}\}_{1 \leq k \leq n} \subset \mathbb{R}_+$ the eigenvalues of the matrix $\boldsymbol{S}$ in Theorem 7.3.2, and let $\{\boldsymbol{v}_k\}_{1 \leq k \leq n} \subset \mathbb{R}^n$ be an orthonormal basis of $\mathbb{R}^n$ composed of real eigenvectors of $\boldsymbol{S}$ satisfying $\forall k \in [\![1,n]\!]$, $\boldsymbol{S}\boldsymbol{v}_k = \lambda_{k,n}\boldsymbol{v}_k$. Denoting by $\boldsymbol{V} \in \mathscr{M}_n(\mathbb{R})$ the matrix

$$\boldsymbol{V} = (\boldsymbol{v}_1 | \ldots | \boldsymbol{v}_n) \quad ,$$

we then have

$$\boldsymbol{S} = \boldsymbol{V} \begin{pmatrix} \lambda_{1,n} & & \\ & \ddots & \\ & & \lambda_{n,n} \end{pmatrix} \boldsymbol{V}^T, \quad \boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{V}\boldsymbol{V}^T = \boldsymbol{I}_n \quad . \tag{7.31}$$

Given that the application E defined in Equation (7.29) is a linear isometry, it maps orthonormal sequences in $\mathbb{R}^n$ (with respect to $\langle ., . \rangle_2$) to orthonormal sequences in V_n (with respect to $\langle ., . \rangle_{L^2(\mathcal{M})}$). Hence, the sequence $\{e_{k,n}\}_{1 \leq k \leq n} \subset V_n$, where

$$\forall k \in [\![1,n]\!], \quad e_{k,n} = E(\boldsymbol{v}_k) \quad ,$$

is an orthonormal family of functions of V_n.

Moreover, given that E is linear and bijective, $\{E(\boldsymbol{v}_k)\}_{1\leq k\leq n}$ is actually a basis of V_n since $\{\boldsymbol{v}_k\}_{1\leq k\leq n}$ is a basis of $\mathbb{R}^n$. Consequently, $\{e_{k,n}\}_{1\leq k\leq n}$ defines an orthonormal basis of V_n composed of eigenfunctions of $-\Delta_n$.

Take $\gamma : \mathbb{R}_+ \to \mathbb{R}$. Following the definition of the discretized operator $-\Delta_n$ and analogously to the definition of the operator $\gamma(-\Delta_{\mathcal{M}})$ from the operator $-\Delta_{\mathcal{M}}$, the discretization of the operator $\gamma(-\Delta_{\mathcal{M}})$ on V_n is then defined as the endomorphism $\gamma(-\Delta_n)$ of V_n given by

$$\begin{aligned}\gamma(-\Delta_n) : V_n &\to V_n \\ \varphi &\mapsto \gamma(-\Delta_n)\varphi := \sum_{k=1}^{n} \gamma(\lambda_{k,n})\langle \varphi, e_{k,n}\rangle_{L^2(\mathcal{M})} e_{k,n} \quad .\end{aligned} \tag{7.32}$$

Lemma 7.3.3. *The definition of $\gamma(-\Delta_n)$ in Equation (7.32) does not depend on the choice of orthonormal basis $\{e_{k,n}\}_{1\leq k\leq n}$ of eigenfunctions of $-\Delta_n$ satisfying $\forall k \in [\![1,n]\!]$, $-\Delta_n e_{k,n} = \lambda_{k,n} e_{k,n}$.*

Proof. Let $\{e_{k,n}\}_{1\leq k\leq n}$ and $\{e'_{k,n}\}_{1\leq k\leq n}$ denote two orthonormal basis of V_n such that $\forall k \in [\![1,n]\!]$, $-\Delta_n e_{k,n} = \lambda_{k,n} e_{k,n}$ and $-\Delta_n e'_{k,n} = \lambda_{k,n} e'_{k,n}$. Assume that $\gamma(-\Delta_n)$ is defined by Equation (7.32).
Let $\boldsymbol{A} \in \mathscr{M}_n(\mathbb{R})$ be the change-of-basis matrix between $\{e_{k,n}\}_k$ and $\{e'_{k,n}\}_k$, i.e.

$$\forall k \in [\![1,n]\!], \quad e_{k,n} = \sum_{l=1}^{n} A_{kl} e'_{l,n} \quad .$$

The orthonormality of $\{e_{k,n}\}_k$ and $\{e'_{k,n}\}_k$ gives that

$$\begin{aligned}\forall k,k' \in [\![1,n]\!], \quad \langle e_{k,n}, e_{k',n}\rangle_{L^2(\mathcal{M})} &= \sum_{l=1}^{n}\sum_{l'=1}^{n} A_{kl}\langle e'_{l,n}, e'_{l',n}\rangle_{L^2(\mathcal{M})} A_{k'l'} = \sum_{l=1}^{n}\sum_{l'=1}^{n} A_{kl}\delta_{ll'}A_{k'l'} \\ &= \sum_{l=1}^{n} A_{kl}A_{k'l} = [\boldsymbol{A}\boldsymbol{A}^T]_{kk'} = \delta_{kk'} \quad .\end{aligned}$$

Hence, $\boldsymbol{A}\boldsymbol{A}^T = \boldsymbol{I}_n = \boldsymbol{A}^T\boldsymbol{A}$.
On the other hand, the fact that $\{e_{k,n}\}_k$ and $\{e'_{k,n}\}_k$ are eigenfunctions of $-\Delta_n$ gives $\forall k \in [\![1,n]\!]$, $-\Delta_n e_{k,n} = \sum_{l=1}^n A_{kl}(-\Delta_n e'_{l,n}) = \sum_{l=1}^n \lambda_{l,n} A_{kl} e'_{l,n} = \lambda_{k,n} e_{k,n} = \lambda_{k,n}\sum_{l=1}^n A_{kl} e'_{l,n}$. Hence,

$$\forall k,l \in [\![1,n]\!], \quad \lambda_{k,n}A_{kl} = \lambda_{l,n}A_{kl}$$

Consequently, note that $\forall k,l \in [\![1,n]\!]$, $\gamma(\lambda_{k,n})A_{kl} = \gamma(\lambda_{l,n})A_{kl}$ still holds (this can be verified with a simple proof by contradiction). Therefore, we have

$$\gamma(\boldsymbol{\Lambda})\boldsymbol{A} = \boldsymbol{A}\gamma(\boldsymbol{\Lambda}), \quad \text{where} \quad \gamma(\boldsymbol{\Lambda}) := \begin{pmatrix} \gamma(\lambda_{1,n}) & & \\ & \ddots & \\ & & \gamma(\lambda_{n,n}) \end{pmatrix} \quad .$$

Finally, note that $\forall \varphi \in V_n$,

$$\begin{aligned}\gamma(-\Delta_n)\varphi &= \sum_k \gamma(\lambda_{k,n}) \left\langle \varphi, \sum_l A_{kl} e'_{l,n} \right\rangle_{L^2(\mathcal{M})} \sum_{l'} A_{kl'} e'_{l',n} \\ &= \sum_{k,l,l'} \gamma(\lambda_{k,n}) A_{kl} A_{kl'} \left\langle \varphi, e'_{l,n} \right\rangle_{L^2(\mathcal{M})} e'_{l',n} \\ &= \sum_{l,l'} [\boldsymbol{A}^T\gamma(\boldsymbol{\Lambda})\boldsymbol{A}]_{ll'} \left\langle \varphi, e'_{l,n} \right\rangle_{L^2(\mathcal{M})} e'_{l',n} = \sum_{l,l'} [\boldsymbol{A}^T\boldsymbol{A}\gamma(\boldsymbol{\Lambda})]_{ll'} \left\langle \varphi, e'_{l,n} \right\rangle_{L^2(\mathcal{M})} e'_{l',n} \\ &= \sum_{l,l'} [\boldsymbol{I}_n\gamma(\boldsymbol{\Lambda})]_{ll'} \left\langle \varphi, e'_{l,n} \right\rangle_{L^2(\mathcal{M})} e'_{l',n} = \sum_{l} \gamma(\lambda_{l,n}) \left\langle \varphi, e'_{l,n} \right\rangle_{L^2(\mathcal{M})} e'_{l,n} \ ,\end{aligned}$$

which proves the result. □

7.3.2 Ritz–Galerkin discretization of GeGFs

Let $\mathcal{W}_n$ be the V_n-valued random variable defined by

$$\mathcal{W}_n = \sum_{k=1}^{n} W_k e_{k,n} \quad , \tag{7.33}$$

where $W_1, \dots, W_n$ are independent standard Gaussian variables. Then, $\mathcal{W}_n$ is called *white noise on* V_n. This definition of white noise is coherent with the characterization of Gaussian white noises introduced in Proposition 7.1.2. Indeed, the linear functional $\varphi \in V_n \mapsto \langle \mathcal{W}_n, \varphi \rangle_{L^2(\mathcal{M})} = \sum_{k=1}^{n} W_k \langle e_{k,n}, \varphi \rangle_{L^2(\mathcal{M})}$ maps elements of V_n to Gaussian variables satisfying $\forall \varphi \in V_n$, $\mathbb{E}[\langle \mathcal{W}_n, \varphi \rangle_{L^2(\mathcal{M})}] = 0$ and $\forall \varphi_1, \varphi_2 \in V_n$,

$$\mathrm{Cov}[\langle \mathcal{W}_n, \varphi \rangle_{L^2(\mathcal{M})}, \langle \mathcal{W}_n, \varphi \rangle_{L^2(\mathcal{M})}] = \sum_{k=1}^{n} \langle e_{k,n}, \varphi_1 \rangle_{L^2(\mathcal{M})} \langle e_{k,n}, \varphi_2 \rangle_{L^2(\mathcal{M})} = \langle \varphi_1, \varphi_2 \rangle_{L^2(\mathcal{M})} \, .$$

In particular, by independence of the W_k, the characteristic function of this functional is

$$\varphi \in V_n \mapsto \mathbb{E}[e^{\mathrm{i}\langle \mathcal{W}_n, \varphi \rangle_{L^2(\mathcal{M})}}] = \prod_{k=1}^{n} \Psi_{\mathcal{N}(0,1)}(\langle e_{k,n}, \varphi \rangle_{L^2(\mathcal{M})}) = e^{-\frac{1}{2}\langle \varphi, \varphi \rangle_{L^2(\mathcal{M})}} \, ,$$

which is the expected form defined in Equation (7.4).

Proposition 7.3.4. *Let $\mathcal{W}_n$ be a white noise on V_n.*

Then $\mathcal{W}_h$ can be written

$$\mathcal{W}_n = \sum_{k=1}^{n} \tilde{W}_k \psi_k \quad , \tag{7.34}$$

where the weights $\tilde{W}_1 \dots \tilde{W}_n$ for a Gaussian vector defined by $(\tilde{W}_1 \dots \tilde{W}_n)^T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{C}^{-1})$.

Proof. Using the linearity of the map E defined in Theorem 7.3.2, the definition of $\mathcal{W}_n \in V_n$ in Equation (7.33) can be written $\mathcal{W}_n = \sum_{k=1}^{n} W_k E(\boldsymbol{v}_k) = E\left(\sum_{k=1}^{n} W_k \boldsymbol{v}_k\right) = E\left(\boldsymbol{V}(W_1, \dots, W_n)^T\right)$ where $(W_1, \dots, W_n)^T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.
But also, denoting $\tilde{W}_1, \dots, \tilde{W}_n$ the coordinates of $\mathcal{W}_n$ in the basis $\{\psi_k\}_k$, we get from Equation (7.29), $\mathcal{W}_n = \sum_{k=1}^{n} \tilde{W}_k \psi_i = E\left(\boldsymbol{C}^{1/2}(\tilde{W}_1 \dots \tilde{W}_n)^T\right)$. Hence, using the fact that E is bijective, we get $(\tilde{W}_1 \dots \tilde{W}_n)^T = \boldsymbol{C}^{-1/2}\boldsymbol{V}(W_1 \dots W_n)^T$ which proves the result. □

Theorem 7.3.5. *Let $\mathcal{Z}_n$ be the V_n-valued random variable defined by*

$$\mathcal{Z}_n = \gamma(-\Delta_n)\mathcal{W}_n \quad , \tag{7.35}$$

where $\gamma(-\Delta_n)$ is the mapping of Equation (7.32) and $\mathcal{W}_n$ is a Gaussian white noise on V_n.

Then, $\mathcal{Z}_n$ can be decomposed in the basis $\{\psi_k\}_{1 \le k \le n}$ as

$$\mathcal{Z}_n = \sum_{k=1}^{n} Z_k \psi_k \quad , \tag{7.36}$$

The weights $Z_1, \dots, Z_n$ form a Gaussian vector $\boldsymbol{Z} = (Z_1, \dots, Z_n)^T$ with mean $\mathbf{0}$ and covariance matrix

$$\mathrm{Var}[\boldsymbol{Z}] = \boldsymbol{C}^{-1/2}\gamma^2(\boldsymbol{S})\boldsymbol{C}^{-1/2} \quad , \tag{7.37}$$

where $\boldsymbol{C}$ and $\boldsymbol{S}$ are defined in Equations (7.27) and (7.28) , $\boldsymbol{C}^{-1/2}$ is the inverse of the principal square-root of $\boldsymbol{C}$ and $\gamma^2(\boldsymbol{S})$ denotes the graph filter with shift operator $\boldsymbol{S}$ and transfer function $\lambda \mapsto \gamma(\lambda)^2$.

Proof. Notice that $\mathcal{Z}_n \in V_n$, hence there exists some random vector $\boldsymbol{Z} \in \mathbb{R}^n$ such that $\mathcal{Z}_n = \sum_{k=1}^{n} Z_k \psi_k$. And following Equation (7.29), $\mathcal{Z}_n = E\left(\boldsymbol{C}^{1/2}\boldsymbol{Z}\right)$.

But also, following the definition of $\mathcal{W}_n$ in Equation (7.33) and the linearity of E, $\mathcal{Z}_n = \gamma(-\Delta_n)\mathcal{W}_n = \sum_{k=1}^n \gamma(\lambda_{k,n}) W_k E(\boldsymbol{v}_k) = E\left(\sum_{k=1}^n \gamma(\lambda_{k,n}) W_k \boldsymbol{v}_k\right)$ which gives,

$$\mathcal{Z}_n = \gamma(-\Delta_n)\mathcal{W}_n = E\left(\boldsymbol{V}\begin{pmatrix}\gamma(\lambda_{1,n}) & & \\ & \ddots & \\ & & \gamma(\lambda_{n,n})\end{pmatrix}\begin{pmatrix}W_1 \\ \vdots \\ W_n\end{pmatrix}\right) \quad .$$

Therefore, given that E is bijective,

$$\boldsymbol{Z} = \boldsymbol{C}^{-1/2}\boldsymbol{V}\begin{pmatrix}\gamma(\lambda_{1,n}) & & \\ & \ddots & \\ & & \gamma(\lambda_{n,n})\end{pmatrix}\begin{pmatrix}W_1 \\ \vdots \\ W_n\end{pmatrix} \quad ,$$

where $(W_1 \dots W_n)^T \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, which proves the result. □

Theorem 7.3.5 provides an explicit expression for the the covariance matrix of the weights of V_n-valued random variables. Consequently, generating realizations of such random functions can easily be done by simulating a zero-mean Gaussian vector of weights with covariance matrix given in Equation (7.37) and then building the weighted sum Equation (7.36). More generally, the statistical properties of such fields are entirely specified by those of its random weights.

Following Equation (7.37), note that the vector

$$\boldsymbol{X} = \boldsymbol{C}^{1/2}\boldsymbol{Z}$$

is a zero-mean Gaussian vector with covariance matrix $\gamma^2(\boldsymbol{S})$. It can therefore be seen as a $\boldsymbol{S}$-stationary stochastic graph signal with spectral density γ^2, on a n-graph $\mathcal{G}$ for which $\boldsymbol{S}$ can be a shift operator. This means in particular that the two vertices i, j of $\mathcal{G}$ such that $i \neq j$ should be adjacent whenever $S_{ij} = [\boldsymbol{C}^{-1/2}\boldsymbol{R}\boldsymbol{C}^{-1/2}]_{ij} \neq 0$. The results and algorithms presented in the first two parts of this work can therefore be applied to $\boldsymbol{X}$ seen as a graph signal. For instance the simulation algorithm of Section 3.1 can be used to generate realizations of $\boldsymbol{X}$ and therefore of $\boldsymbol{Z} = \boldsymbol{C}^{-1/2}\boldsymbol{X}$.

Of particular interest is the case where the matrix $\boldsymbol{S}$ is sparse, as then the graph filtering algorithms become computationally efficient. In the next chapter, a particular family of subspaces V_n of $L^2(\mathcal{M})$ yielding sparse shift operators $\boldsymbol{S}$ is presented: the subspaces arising from the finite element method. Convergence results of the discretization of a GeGF onto such subspaces are derived.

7.4 Discussion

7.4.1 Comparison with the Karhunen–Loève expansion

In Section 7.2.1 we provided a link between our construction of GeGFs and Karhunen–Loève expansions. The latter are classically used to derive numerical approximations of a Gaussian field by truncating at a given order the expansion in Equation (7.16). The main drawback of this approach is the determination of the eigenfunctions (and eigenvalues) of the covariance operator defining the expansion.

For some simple domains, the analytical expression of the eigenfunctions is known, and the Karhunen–Loève expansion becomes an efficient modeling tool for isotropic fields defined on them (Solin and Särkkä, 2014). In the general case however, the determination of approximation eigenfunctions can easily require heavy computations: the integral (eigenvalue) problem of Equations (7.14) and (7.15) is indeed discretized, and the resulting matrix eigenvalue problem is solved by diagonalization (Huang et al., 2001).

On the other hand, with our description of GeGFs, no problem-specific diagonalization is actually needed. Indeed, the weights of the (Ritz–Galerkin) discretization of any (isotropic) GeGF are given by Theorem 7.3.5 and can be leveraged using the graph signal processing techniques presented in the early chapters of this work. Moreover, the extension to more complex domains (i.e. arbitrary smooth submanifolds of $\mathbb{R}^d$) and to fields with local anisotropies is straightforward using our approach: it only affects the definition of the entries of the mass and stiffness matrices in Theorem 7.3.5. Doing the same with Karhunen–Loève expansions would suppose first to define the covariance operator, which is far from trivial for these problems.

7.4.2 Accounting for local anisotropies

We assume for this subsection that B denotes a compact connected domain of $\mathbb{R}^d$ and that $d \in \{2,3\}$. The goal of this section is to highlight how our characterization of GeGFs on Riemannian manifolds relates to a particular class of non-stationary Gaussian random fields on B.

Namely, we call *Gaussian random field with local anisotropies*[6] (GRFLA) on B any non-stationary Gaussian random field Z defined on B such that its covariance function satisfies

$$\forall \boldsymbol{p} \in B, \quad \mathrm{Cov}[Z(\boldsymbol{p}), Z(\boldsymbol{p}+\boldsymbol{h})] \underset{\boldsymbol{h}\to\boldsymbol{0}}{\sim} C_0(\|\boldsymbol{A}(\boldsymbol{p})\boldsymbol{h}\|_2) = C_0\left(\sqrt{\boldsymbol{h}^T\boldsymbol{A}(\boldsymbol{p})^T\boldsymbol{A}(\boldsymbol{p})\boldsymbol{h}}\right) \quad , \tag{7.38}$$

where $\boldsymbol{A}(\boldsymbol{p}) \in \mathscr{M}_d(\mathbb{R})$ is an invertible matrix and C_0 is an isotropic covariance function. Hence, Z corresponds to a random field that can be made locally isotropic around each point $\boldsymbol{p} \in B$ by the linear change of variable $\boldsymbol{h} \to \boldsymbol{h}' = \boldsymbol{A}(\boldsymbol{p})\boldsymbol{h}$. In particular, the matrix $\boldsymbol{A}(\boldsymbol{p})^T\boldsymbol{A}(\boldsymbol{p})$ is called *anisotropy matrix* and is symmetric positive definite.

The anisotropy matrices defining a GRFLA actually have a geometric interpretation. Indeed, for $d \in \{2,3\}$, the anisotropy matrices can be written as the composition of a rotation matrix, a diagonal matrix and the inverse of rotation matrix (cf. Appendix A.2.3). Around each point of the domain, the covariance of Z then acts like the covariance of a stationary field with geometric anisotropy defined by these matrices. Hence, working with GRFLA allows to handle a large spectrum of non-stationary random fields, as the local behavior of the resulting fields is parametrized by interpretable geometric parameters, namely a rotation and scalings along principal coordinate axes.

Following the results of Section 7.2.2, building a GeGF on B that acts like a GRFLA can be done by endowing B with a Riemannian metric. Then if a field of local anisotropy parameters (namely rotation angles $\theta, \theta_1, \theta_2, \theta_3$ and ranges ρ_1, ρ_2, ρ_3 for the scalings) are defined across a domain B, we can endow B with the metric defined by

$$\boldsymbol{G}(\boldsymbol{p}) = \begin{cases} \boldsymbol{V}_{\theta(\boldsymbol{p})}\mathrm{Diag}\left(1/\rho_1(\boldsymbol{p})^2, 1/\rho_2(\boldsymbol{p})^2\right)\boldsymbol{V}_{\theta(\boldsymbol{p})}^T & \text{if } d=2 \\ \boldsymbol{V}_{\theta_1(\boldsymbol{p}),\theta_2(\boldsymbol{p}),\theta_3(\boldsymbol{p})}\mathrm{Diag}\left(1/\rho_1(\boldsymbol{p})^2, 1/\rho_2(\boldsymbol{p})^2, 1/\rho_3(\boldsymbol{p})^2\right)\boldsymbol{V}_{\theta_1(\boldsymbol{p}),\theta_2(\boldsymbol{p}),\theta_3(\boldsymbol{p})}^T & \text{if } d=3 \end{cases}, \tag{7.39}$$

where $\boldsymbol{V}_*$ denotes a two(or-three)-dimensional rotation matrix (cf. Appendix A.2.3). Then our construction of GeGFs using the Laplace-Beltrami operator associated with this metric will yield a random field on B which respects the prescribed local anisotropies.

The advantage of this method is that it allows to easily incorporate into the model of a non-stationary random field information about its local behavior, as described geometrically by the anisotropy parameters. In the remainder of this subsection, we draw parallels between this approach and other approaches aiming at modeling non-stationary random fields. A complete review of such models can be found in (Fouedjio, 2017).

Space deformation

Within the space deformation approach, a non-stationary field Z defined on B is modeled as

$$\forall \boldsymbol{p} \in B, \quad Z(\boldsymbol{p}) = Y(\Phi(\boldsymbol{p})) \quad , \tag{7.40}$$

where Φ is a deterministic non-linear smooth bijective function defined over B and Y is an isotropic random field on $\Phi(B)$ which covariance function is denoted by C. The covariance function of Z then satisfies

$$\forall \boldsymbol{p}_1, \boldsymbol{p}_2 \in B, \quad \mathrm{Cov}[Z(\boldsymbol{p}_1), Z(\boldsymbol{p}_2)] = C(\|\Phi(\boldsymbol{p}_1) - \Phi(\boldsymbol{p}_2)\|_2) \tag{7.41}$$

A first-order Taylor approximation of Equation (7.41) allows to retrieve Equation (7.38) where $\boldsymbol{A}(\boldsymbol{p})$ is set to be the Jacobian matrix of Φ at $\boldsymbol{p}$. Hence, Z is a GRFLA.

In practice, problems involving Z are transposed to the isotropic field Y by determining the transformation Φ from observations of Z, which can be done using a multi-dimensional scaling algorithm (Kruskal, 1964). This approach is detailed in (Sampson and Guttorp, 1992).

[6] This notion corresponds, in the zero-mean case, to the notion of *locally stationary field* introduced by (Matheron, 1971).

Following the results of Sections 6.6 and 7.2.2, the random fields defined by Equation (7.40) can be directly interpreted as instances of one of our GeGFs, defined on B equipped with the Riemannian metric defined from the Jacobian matrix of Φ (cf. Equation (6.25)). In a context where the anisotropy parameters are known, using the formalism of GeRFs on Riemannian manifolds rather than the space deformation representation of Equation (7.40) allows to actually work with Z without having to specify the deformation transformation Φ, that actually may not exist[7]. Indeed, we simply set the metric on B using Equation (7.39) and then use Theorem 7.3.5 to characterize (numerical approximations) of the resulting GeGF.

Convolution model

Within the convolution approach (Higdon et al., 1999), a non-stationary field Z defined on B is modeled at each point $\boldsymbol{p} \in B$ as the result of a (stochastic) convolution on $\mathbb{R}^d$ of a so-called kernel function $q_{\boldsymbol{p}}$ with a Gaussian white noise $\mathcal{W}$:

$$\forall \boldsymbol{p} \in B, \quad Z(\boldsymbol{p}) = \int_B q_{\boldsymbol{p}}(\boldsymbol{x}) \mathcal{W}(d\boldsymbol{x}) \quad . \tag{7.42}$$

Note that the non-stationarity of Z is a consequence of the fact that we allow the kernel functions $\{q_{\boldsymbol{p}}\}_{\boldsymbol{p} \in B}$ to vary with $\boldsymbol{p} \in B$.

In the case where a field of anisotropy parameters is defined on B, Paciorek and Schervish (2006) proposed to set $q_{\boldsymbol{p}}$ as the density of a multivariate Gaussian distribution centered at $\boldsymbol{p}$ and with covariance matrix $\boldsymbol{G}(\boldsymbol{p})$, as given by Equation (7.39). This yields a closed-form for the covariance function of the resulting field Z:

$$\mathrm{Cov}[Z(\boldsymbol{p}_1), Z(\boldsymbol{p}_2)] = \frac{1}{\pi^{d/2} \sqrt{\det \boldsymbol{A}(\boldsymbol{p}_1, \boldsymbol{p}_2)}} e^{-(\boldsymbol{p}_2 - \boldsymbol{p}_1)^T \boldsymbol{A}(\boldsymbol{p}_1, \boldsymbol{p}_2)^{-1} (\boldsymbol{p}_2 - \boldsymbol{p}_1)}, \quad \boldsymbol{p}_1, \boldsymbol{p}_2 \in B \tag{7.43}$$

where

$$\boldsymbol{A}(\boldsymbol{p}_1, \boldsymbol{p}_2) := \frac{\boldsymbol{G}(\boldsymbol{p}_1) + \boldsymbol{G}(\boldsymbol{p}_2)}{2} \quad .$$

Hence, Z can be seen as GRFLA if we consider that, for any $\boldsymbol{p} \in B$, if we take $\boldsymbol{h} \to \boldsymbol{0}$ then $\det \boldsymbol{A}(\boldsymbol{p}, \boldsymbol{p} + \boldsymbol{h})$ can be considered as constant. In particular, Z then corresponds to a non-stationary Gaussian covariance function. Generalizations of Equation (7.43) have been proposed for Matérn and Cauchy covariance functions (Stein, 2005). They yield the same forms of covariance function as in Equation (7.43), except that the Gaussian covariance function is replaced by the appropriate one.

Contrary to the space deformation approach, taking field of anisotropies into account is done readily when setting the kernel functions through $\{\boldsymbol{G}(\boldsymbol{p})\}_{\boldsymbol{p} \in B}$ in Equation (7.39). However, when considering the expression of the resulting covariance function, we see that the covariance between two points depends only "what is happening" at these two points specifically. Indeed, the covariance between $Z(\boldsymbol{p}_1)$ and $Z(\boldsymbol{p}_2)$ in Equation (7.43) can be seen as the covariance, between $\boldsymbol{p}_1 \in B$ and $\boldsymbol{p}_2 \in B$, of a random function on B with (global) geometric anisotropy defined by the averaged anisotropy matrix $\boldsymbol{A}(\boldsymbol{p}_1, \boldsymbol{b}_2)$. Hence, the structure of the anisotropy field between $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ is not taken into account in Equation (7.43).

This property is not shared by the space deformation approach, which in this sense is more flexible. Indeed, the deformation process Φ in Equation (7.41), makes it so that the covariance between any two points of the domain B depends on the overall structure of the anisotropy field. This is due to the fact that this structure is actually defined by the function Φ.

Hence, our GeGFs on Riemannian manifolds allow to take the best of the two approaches presented in this discussion. They ally the ease of taking into account fields of local anisotropies (of the convolution model) to the definition of covariance functions that assimilate them as a whole (as space deformation models do). In summary, the GeGF approach allows to easily take into account local anisotropies in a global model of covariance. However, we lose the closed-form expression of the covariance model, which can only be computed numerically using Theorem 7.3.5.

[7] At least if we consider transformations Φ from $\mathbb{R}^d$ to $\mathbb{R}^d$.... However, Perrin and Meiring (2003) showed that a non-stationary field (with moments at least of order 2) defined on $\mathbb{R}^d$ can always be seen as a stationary field defined $\mathbb{R}^{2d}$, which points towards considering deformations into space with higher dimensions.

7.4.3 Link to stochastic partial differential equation approach

In this subsection, we show how the class of GeGFs we introduced relates to the stochastic partial differential equation (SPDE) approach introduced by Lindgren et al. (2011).

Within the SPDE approach, stationary Gaussian random fields Z on $\mathbb{R}^d$ with a Matérn covariance function, are characterized as the stationary solutions of the SPDE defined in $\mathbb{R}^d$ by

$$(\kappa^2 - \Delta)^{\alpha/2} Z = \tau \mathcal{W} \tag{7.44}$$

where $\kappa > 0$, $\alpha > d/2$, $\tau > 0$, $(\kappa^2 - \Delta)^{\alpha/2}$ is the pseudo-differential operator with symbol function $p(\boldsymbol{\xi}) = \kappa^2 + \|\boldsymbol{\xi}\|^2$ (cf. Equation (7.2)) and $\mathcal{W}$ is a Gaussian white noise (Whittle, 1954). Hence, Equation (7.44) can actually be seen as a particular case of the more general set of SPDEs defined as

$$\mathcal{L}_p Z = \mathcal{W} \quad , \tag{7.45}$$

where p is a strictly positive radial function of $\mathbb{R}^d$, i.e. for some $p_0 : \mathbb{R}_+ \to \mathbb{R}_+^*$,

$$\forall \boldsymbol{\xi} \in \mathbb{R}^d, \quad p(\boldsymbol{\xi}) = p_0(\|\boldsymbol{\xi}\|^2) \quad ,$$

and $\mathcal{L}_p$ is the pseudo-differential operator of $\mathbb{R}^d$ with symbol function p. In particular, the equality is here understood in the second-order sense, meaning that Z is seen as a generalized random field and both sides have the same covariance functional.

The class of SPDEs defined by Equation (7.45) was extensively studied by (Carrizo Vergara et al., 2018), who derived conditions on the symbol function p for the existence (and uniqueness) of stationary solutions. Precisely, they show that existence and uniqueness of a stationary solutions are guaranteed if p_0 is a continuous non-negative function satisfying the following conditions:

- p_0 is polynomially upper-bounded,
- p_0 is lower-bounded by the inverse of a strictly positive polynomial,
- $\exists N > 0, \quad \int_{\mathbb{R}^d} |p_0(\|\boldsymbol{\omega}\|^2)|^{-2}(1 + \|\boldsymbol{\omega}\|^2)^{-N} d\boldsymbol{\omega} < \infty$.

They show that the solution is the obtained as the generalized random field Z of $\mathbb{R}^d$ defined by

$$Z = \mathcal{L}_{1/p} \mathcal{W} \quad ,$$

where $\mathcal{L}_{1/p}$ is the pseudo-differential operator with symbol function $1/p$ (Carrizo Vergara et al., 2018, Theorem 1 & Remark 2). In particular, Z is defined as in Remark 7.2.2.

Following Remark 7.2.2, we conclude that the class of GeGFs we have been working with includes the solutions of Equation (7.45) (when transposed to the manifold) and therefore the solutions of the SPDE in Lindgren et al. (2011), which are retrieved by taking

$$p_0(\|\boldsymbol{\xi}\|^2) = \frac{1}{\tau}(\kappa^2 + \|\boldsymbol{\xi}\|^2)^{\alpha/2}, \quad .$$

In particular one may notice that the expressions of the covariance matrix of the weights of the finite element approximation of Matérn fields proposed by Lindgren et al. (2011) are retrieved by setting $\gamma = 1/p_0$ in Theorem 7.3.5.

Conclusion

Generalized random fields on Riemannian manifolds were introduced as a tool allowing to model Gaussian fields on complex spatial domains and with local anisotropies. They can be seen as the transposition of isotropic stationary random fields of $\mathbb{R}^d$ to compact Riemannian manifolds. They were defined using a general approach based on the properties of the Laplace-Beltrami operator associated with the Riemannian manifold: this operator actually takes on the "transposition" process mentioned above given that it accounts for both the geometry of the manifold and the eventual presence of anisotropies (through the Riemannian metric used to define it). The approach presented in this section can therefore be applied on any compact Riemannian manifold to define non-stationary field from the expression of a (radial) spectral density.

The discretization of these generalized random fields was then tackled. Given a set of deterministic "basis" functions defined on the domain, we looked for approximations that would be written as a weighted sum of the basis functions. We derived a theorem that entirely characterizes the random weights of this linear combination, thus providing a numerical model for the generalized random fields.

In the next section, we apply this decomposition theorem to the basis functions obtained from the finite element method and derive a convergence result for the approximation.

8

Finite element approximation of generalized Gaussian fields

Contents

Résumé

Dans ce chapitre, nous proposons de discrétiser les champs gaussiens généralisés introduits au chapitre précédent à l'aide de la méthode des éléments finis, en nous basant sur les résultats d'approximation de Ritz–Galrkin déjà obtenus.

Nous commençons par introduire la méthode des éléments finis et à en présenter des exemples de mise en œuvre. Puis nous appliquons cette méthode à la discrétisation de champs gaussiens généralisés, en prenant soin de détailler le problème des conditions limites. Nous présentons également une analyse d'erreur de l'approximation obtenue, débouchant sur un résultat de convergence de l'approximation vers le champ lorsque la taille de maillage se réduit. Enfin, nous donnons un exemple complet de construction de cette approximation afin de mettre en évidence l'intérêt de travailler avec des éléments finis.

Introduction

The aim of this chapter is to build from the results of Theorem 7.3.5, and provide an example of construction of a discretization of a generalized random field defined on a Riemannian manifold. The set of basis functions used to define the approximation are derived from the finite element method, and we start by recalling its principle. Convergence results of the finite element approximation are then exposed. Finally, the full construction of this approximation is carried out on a simple example.

8.1 Introduction to the finite element method

In this section, we recall the principle of the finite element method and the mathematical objects it involves. We refer the reader to (Brenner and Scott, 2007; Raviart et al., 1998; Strang and Fix, 1973) for a complete review of the method and its main convergence properties.

8.1.1 Mathematical construction of finite elements

Definition of a finite element

Let $K \subset \mathbb{R}^d$ be a compact and connected set, with a non-empty interior. Let $X = \{\boldsymbol{x}^{(i)}\}_{i=1}^N$ be a set of N points of K: $\boldsymbol{x}^{(i)} \in K$. Finally, let P be a finite-dimensional vector space of functions mapping K to $\mathbb{R}$. The triplet (K, X, P) is called a *Lagrange finite element* if $\forall \boldsymbol{\alpha} \in \mathbb{R}^d$, there exists a unique element $p \in P$ such that $\forall j \in [\![1, N]\!]$, $p(\boldsymbol{x}^{(j)}) = \alpha_j$. In this case, we say that the set X is *P-unisolvent*. Hence, all elements of P are uniquely defined by the values they take over the points of K constituting X.

From now on (K, X, P) denotes a Lagrange finite element. Consider then the family $\{p^{(i)}\}_{i=1}^N$ of functions of P defined by:

$$\forall i \in [\![1, N]\!], \quad \forall j \in [\![1, N]\!], \quad p^{(i)}(\boldsymbol{x}^{(j)}) = \delta_{ij} \quad .$$

The functions $\{p^{(i)}\}_{i=1}^N$ are called *shape functions* of the finite element. They define a basis of the vector space P as any element $p \in P$ can be uniquely written as

$$p = \sum_{i=1}^N p(\boldsymbol{x}^{(i)}) p^{(i)}, \quad p \in P \quad .$$

More generally, consider the operator Π_K that associates to any $v : K \to \mathbb{R}$ the element of P defined by

$$\Pi_K v = \sum_{i=1}^N v(\boldsymbol{x}^{(i)}) p^{(i)} \in P \tag{8.1}$$

and called *P-interpolator* associated with the finite element (K, X, P). In particular, $\Pi_K v$ is called *P-interpolate* of v: $\Pi_K v$ is indeed the unique element of P that interpolates v over the set of points X.

Starting from the definition of a single Lagrange finite element, the next proposition is used to build a whole family of finite elements.

Proposition 8.1.1. *Let (K, X, P) be a Lagrange finite element.*

Then for any bijective function $F : K \to \widehat{K} = F(K) \subset \mathbb{R}^d$, the triplet $(\widehat{K}, \widehat{X}, \widehat{P})$ defined by

$$\widehat{K} = F(K), \quad \widehat{X} = F(X) = \{F(\boldsymbol{x}^{(i)})\}_{i=1}^{N}, \quad \widehat{P} = P \circ F^{-1} = \{p \circ F^{-1} : p \in P\} \tag{8.2}$$

is also a Lagrange finite element.

In particular, (K, X, P) and $(\widehat{K}, \widehat{X}, \widehat{P})$ are said to be equivalent. If besides, F is an affine function, then (K, X, P) and $(\widehat{K}, \widehat{X}, \widehat{P})$ are called affine-equivalent.

Proof. By definition, $\widehat{X}$ is a set of $N = \text{Card } X$ points of $\widehat{K}$ and $\widehat{P}$ is a vector space of functions mapping $\widehat{K}$ to $\mathbb{R}$ which has the same dimension as P. We now show that $\widehat{X}$ is $\widehat{P}$-unisolvent.

Let $\boldsymbol{\alpha} \in \mathbb{R}^d$ and assume that there exists $\widehat{p}_1, \widehat{p}_2 \in \widehat{P}$ such that $\forall j \in [\![1, N]\!]$, $\widehat{p}_1(F(\boldsymbol{x}^{(j)})) = \widehat{p}_2(F(\boldsymbol{x}^{(j)})) = \alpha_j$. By definition of $\widehat{P}$, there exists $p_1, p_2 \in P$ such $\widehat{p}_1 = p_1 \circ F^{-1}$ and $\widehat{p}_2 = p_2 \circ F^{-1}$. Hence, $\forall j \in [\![1, N]\!]$, $p_1(\boldsymbol{x}^{(j)}) = p_2(\boldsymbol{x}^{(j)}) = \alpha_j$, and so, by given that X is P-unisolvent, $p_1 = p_2$. This then gives, $\widehat{p}_1 = \widehat{p}_2$, ans so, $\widehat{X}$ is $\widehat{P}$-unisolvent.

Therefore, $(\widehat{K}, \widehat{X}, \widehat{P})$ defines a Lagrange finite element. □

Finite elements defined from simplices

We now restrict ourselves to the case where the compact set K is a *d-simplex*, which we denote T. Namely, T is the convex hull of $(d+1)$ points $\{\boldsymbol{a}^{(i)}\}_{i=1}^{d+1}$ of $\mathbb{R}^d$ such that there is no hyperplane of $\mathbb{R}^d$ containing all of them. T is a polyhedron and particular, for $d = 2$, T is a triangle and for $d = 3$, T is a tetrahedron.

It can be shown that, given that the points $\{\boldsymbol{a}^{(i)}\}_{i=1}^{d+1}$ of $\mathbb{R}^d$ do not lie in a single hyperplane of $\mathbb{R}^d$, the matrix $\boldsymbol{A} \in \mathscr{M}_{d+1}(\mathbb{R})$ defined by

$$\boldsymbol{A} = \left(\begin{array}{c|c|c} & & \\ \boldsymbol{a}^{(1)} & \dots & \boldsymbol{a}^{(d+1)} \\ & & \\ \hline 1 & \dots & 1 \end{array}\right) = \begin{pmatrix} a_1^{(1)} & & a_1^{(d+1)} \\ \vdots & \dots & \vdots \\ a_d^{(1)} & & a_d^{(d+1)} \\ \hline 1 & \dots & 1 \end{pmatrix} \tag{8.3}$$

is invertible. Indeed, $\forall \boldsymbol{y} \in \mathbb{R}^{d+1}$, $\boldsymbol{A}\boldsymbol{y} = \boldsymbol{0} \Rightarrow \sum_{k=1}^{d+1} y_k \boldsymbol{a}^{(k)} = \boldsymbol{0}$ and $\sum_{k=1}^{d+1} y_k = 0$, which gives $\sum_{k=1}^{d} y_k(\boldsymbol{a}^{(k)} - \boldsymbol{a}^{(d+1)}) = \boldsymbol{0}$ and so otherwise To any point $\boldsymbol{x} \in \mathbb{R}^d$ we can therefore associate a set of $(d+1)$ coefficients gathered in a vector $\boldsymbol{b}(\boldsymbol{x}) = (b_1(\boldsymbol{x}), \dots, b_{d+1}(\boldsymbol{x}))^T \in \mathbb{R}^{d+1}$ defined as the solution of the system

$$\boldsymbol{A}\boldsymbol{b}(\boldsymbol{x}) = \left(\begin{array}{c} \boldsymbol{x} \\ \hline 1 \end{array}\right) \quad . \tag{8.4}$$

These coefficients are called *barycentric coordinates of $\boldsymbol{x}$ with respect to T* and can be seen as the unique set of coefficients $b_1(\boldsymbol{x}), \dots, b_{d+1}(\boldsymbol{x}) \in \mathbb{R}$ such that

$$\boldsymbol{x} = \sum_{i=1}^{d+1} b_i(\boldsymbol{x})\boldsymbol{a}^{(i)}, \quad \text{with} \quad \sum_{i=1}^{d+1} b_i(\boldsymbol{x}) = 1, \quad \boldsymbol{x} \in \mathbb{R}^d \quad . \tag{8.5}$$

In particular, $\forall i \in [\![1, d+1]\!]$, the barycentric coordinates of $\boldsymbol{a}^{(i)}$ are given by the i-th canonical basis vector of $\mathbb{R}^{d+1}$: $b_j(\boldsymbol{a}^{(i)}) = \delta_{ij}$, $1 \le i, j \le d+1$.

The barycentric coordinates $\boldsymbol{b}$ of a simplex T provide a characterization of the points it contains:

$$\begin{aligned} T &= \{\boldsymbol{x} \in \mathbb{R}^d : \forall i \in [\![1, d+1]\!],\ b_i(\boldsymbol{x}) \in [0, 1]\} \\ &= \left\{ \sum_{i=1}^{d+1} c_i \boldsymbol{a}^{(i)} : \sum_{i=1}^{d+1} c_i = 1 \text{ and } \forall i \in [\![1, d+1]\!],\ c_i \in [0, 1] \right\} \quad . \end{aligned} \tag{8.6}$$

In particular, Figure 8.1 provides a graphical interpretation of the barycentric coordinates of a triangle (i.e. a 2-simplex).

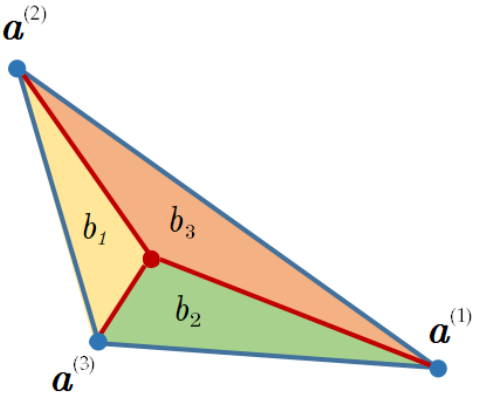


Figure 8.1: Illustration of the barycentric coordinates of a triangle. The i-th barycentric coordinate b_i of a point lying inside the triangle is equal to the ratio between the corresponding colored area and the total area of the triangle.

For $1 \leq i \leq d+1$ and a d-simplex T with barycentric coordinates $\boldsymbol{b}$, the subset $U_i \subset T$ defined by

$$U_i = \{\boldsymbol{x} \in T : \ b_i(\boldsymbol{x}) = 0\} \tag{8.7}$$

is called a *face* of T. In particular, U_i is one of the faces of the polyhedron of $\mathbb{R}^d$ defined by T. Note also that, following Equation (8.6), U_i is actually a $(d-1)$-simplex defined by the points $\{\boldsymbol{a}^{(k)}\}_{k \in [\![1,d+1]\!] \setminus \{i\}}$ using the characterization of simplices given by Equations (8.5) and (8.6). The barycentric coordinates with respect to a face $U_i \subset T$ of a point $\boldsymbol{x} \in U_i$ are therefore equal to its barycentric coordinates with respect to T, where the i-th barycentric coordinate (which is zero) is omitted.

For $m \geq 0$, let P_m be the set of all polynomial functions from $\mathbb{R}^d$ to $\mathbb{R}$ with degree at most m. Hence, any element $p \in P_m$ can be written as

$$p(\boldsymbol{x}) = \sum_{\boldsymbol{k} \in \mathbb{N}^d, |\boldsymbol{k}| \leq m} c_{\boldsymbol{k}} \boldsymbol{x}^{\boldsymbol{k}}, \quad \boldsymbol{x} \in \mathbb{R}^d \quad ,$$

where $\boldsymbol{x}^{\boldsymbol{k}} := \prod_{j=1}^d x_j^{k_j}$, $|\boldsymbol{k}| := \sum_{j=1}^d |k_j|$ and the coefficients $c_{\boldsymbol{k}} \in \mathbb{R}$ are indexed by the index vectors $\boldsymbol{k} \in \mathbb{N}^d, |\boldsymbol{k}| \leq m$. In particular, P_k is a vector space of dimension

$$N = \dim(P_m) = \binom{d+m}{m} = \frac{(d+m)!}{m!d!} \quad .$$

Note that we will also denote the restrictions of P_m to subsets of $\mathbb{R}^d$ with non-empty interior by P_m.

Let then X_m be the set of points of a d-simplex T defined from their barycentric coordinates $\boldsymbol{b}$ by

$$X_m = \left\{ \boldsymbol{x} \in \mathbb{R}^d : \quad \forall j \in [\![1, d+1]\!], \quad b_j(\boldsymbol{x}) \in \left\{ 0, \frac{1}{m}, \ldots, \frac{m-1}{m}, 1 \right\} \right\} \quad , \tag{8.8}$$

and for the particular case where $m = 0$, take,

$$X_0 = \left\{ \boldsymbol{x} \in \mathbb{R}^d : \quad \forall j \in [\![1, d+1]\!], \quad b_j(\boldsymbol{x}) = \frac{1}{d+1} \right\} \quad . \tag{8.9}$$

Note that X_m is therefore composed of $N = \dim(P_m)$ points. Indeed, for any $\boldsymbol{x} \in X_m$ can be uniquely identified by the vector $\boldsymbol{b}'(\boldsymbol{x}) = m(b_1(\boldsymbol{x}) \ \cdots \ b_d(\boldsymbol{x}))^T$ which satisfies $\boldsymbol{b}'(\boldsymbol{x}) \in \mathbb{N}^d$ and $|\boldsymbol{b}'(\boldsymbol{x})| = m \sum_{j=1}^d b_j(\boldsymbol{x}) = m(1 - b_{d+1}(\boldsymbol{x})) \leq m$. Hence $b'(\boldsymbol{x})$ is the multi-index of a monomial in P_k, which proves the statement.

Consequently, using an extension of Lagrange interpolation to the multivariate case (Saniee, 2008), we can deduce that the polynomial of P_m interpolating a function $v : T \mapsto \mathbb{R}$ over the points of X_m is uniquely defined. Hence, X_m is P_m-unisolvent and the triplet (T, X_m, P_m) defines a Lagrange finite element called *d-simplex of type (m)*.

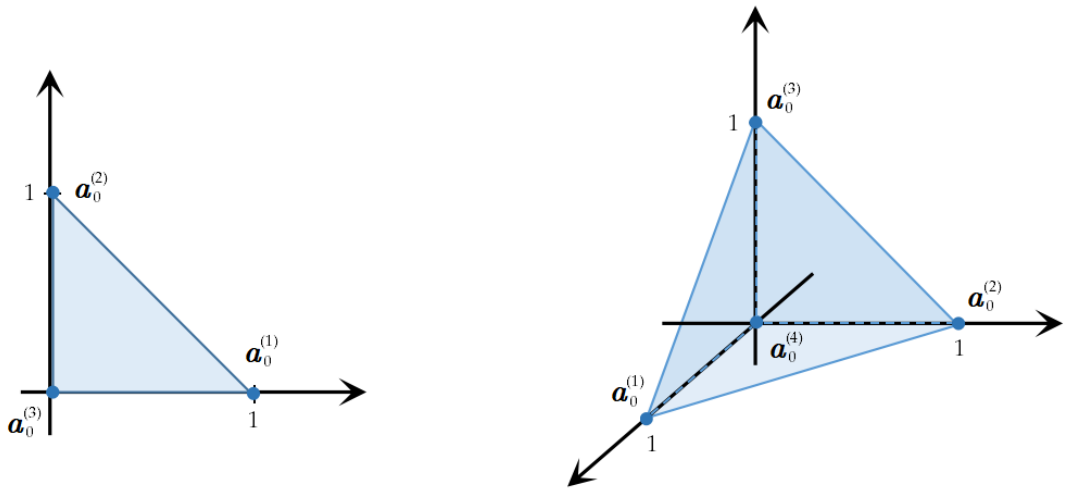


Figure 8.2: Illustration of the standard d-simplices for $d = 2$ (left) and $d = 3$ (right).

Remark 8.1.1. If we restrict the functions of P_m on one of the faces U_i of T, then $X_m \cap U_i$ is $P_{m|U_i}$-unisolvent, meaning that the values a function $p \in P_m$ takes at the points of X_m that lie on a face U_i uniquely define the values p takes on the whole face. This is a direct consequence of the fact that $X_m \cap U_i$ actually defines the interpolating set (as defined in Equation (8.8)) of the $(d-1)$-simplex of type (m) associated with the face U_i.

A very useful property of simplices is that if T and $\widehat{T}$ denote two d-simplices, then their associated d-simplices of type (m) are equivalent finite elements. Indeed, the application F that maps any point $\boldsymbol{x} \in T$ with barycentric coordinates (with respect to T) $\boldsymbol{b}(\boldsymbol{x}) \in \mathbb{R}^{d+1}$ to the point of $\widehat{\boldsymbol{x}} = F(\boldsymbol{x}) \in \widehat{T}$ with barycentric coordinates (with respect to $\widehat{T}$) $\widehat{\boldsymbol{b}}(\widehat{\boldsymbol{x}}) = \boldsymbol{b}(\boldsymbol{x}) \in \mathbb{R}^{d+1}$ is a bijective transform sending T to $\widehat{T}$. Following Proposition 8.1.1, it is then straightforward to check that the Lagrange finite element defined by Equation (8.2) actually corresponds to the d-simplex of type (m) built from $\widehat{T}$.

In particular, following the definition of barycentric coordinates as solution of the linear system in Equation (8.4), the points $\boldsymbol{x} \in T$ and $\widehat{\boldsymbol{x}} = F(\boldsymbol{x}) \in \widehat{T}$ satisfy

$$\begin{pmatrix} \widehat{\boldsymbol{x}} \\ \hline 1 \end{pmatrix} = \widehat{\boldsymbol{A}}\widehat{\boldsymbol{b}}(\widehat{\boldsymbol{x}}) = \widehat{\boldsymbol{A}}\boldsymbol{b}(\boldsymbol{x}) = \widehat{\boldsymbol{A}}\boldsymbol{A}^{-1} \begin{pmatrix} \boldsymbol{x} \\ \hline 1 \end{pmatrix} \quad , \tag{8.10}$$

where the matrices $\boldsymbol{A}$ and $\widehat{\boldsymbol{A}}$ are given by Equation (8.3) using the vertices defining the simplices T and $\widehat{T}$. This gives the relation

$$\widehat{\boldsymbol{x}} = F(\boldsymbol{x}) = \boldsymbol{M}\boldsymbol{x} + \boldsymbol{c} \quad ,$$

where $\boldsymbol{M} \in \mathscr{M}_d(\mathbb{R})$ is the matrix containing the d first rows and columns of $\widehat{\boldsymbol{A}}\boldsymbol{A}^{-1}$ and $\boldsymbol{c} \in \mathbb{R}^d$ is the vector containing the d first entries of the last column of $\widehat{\boldsymbol{A}}\boldsymbol{A}^{-1}$. Hence, the d-simplices of type (m) associated with T and $\widehat{T}$ are in fact affine-equivalent.

In conclusion, all d-simplices of type (m) are in bijection with one another, through an affine transform. In practice, they are all defined from a single reference d-simplex of type (m) which is now defined.

Construction of the standard finite element

Let T_0 be the d-simplex defined from the following points of $\mathbb{R}^d$: $\boldsymbol{a}_0^{(1)} = (1, 0, \ldots, 0)$, $\boldsymbol{a}_0^{(2)} = (0, 1, 0, \ldots, 0)$, ..., $\boldsymbol{a}_0^{(d)} = (0, \ldots, 0, 1)$ and $\boldsymbol{a}_0^{(d+1)} = (0, 0, \ldots, 0)$. T_0 is called the standard d-simplex (cf. Figure 8.2) .

In particular, the barycentric coordinates $\boldsymbol{b}^0$ of a standard d-simplex satisfy for any $\boldsymbol{x} \in \mathbb{R}^d$ the relation $\boldsymbol{A}^0\boldsymbol{b}^0(\boldsymbol{x}) = (\boldsymbol{x}^T \ |1)^T$ where

$$\boldsymbol{A}^0 = \left(\begin{array}{c|c} \boldsymbol{I}_d & \boldsymbol{0}_d \\ \hline \boldsymbol{1}_d^T & 1 \end{array} \right) \quad .$$

Equivalently, the barycentric coordinates with respect to T_0 are given by

$$\forall \boldsymbol{x} \in \mathbb{R}^d, \qquad b_i^0(\boldsymbol{x}) = x_i \quad \forall i \in [\![1,d]\!] \quad \text{and} \quad b_{d+1}^0(\boldsymbol{x}) = 1 - \sum_{i=1}^{d} x_i \quad . \tag{8.11}$$

Hence, the first d barycentric coordinates of a point $\boldsymbol{x} \in \mathbb{R}^d$ with respect to T_0 correspond to its actual Cartesian coordinates.

The d-simplex of type (m) defined from T_0 is called the *standard d-simplex of type* (m) and is denoted (T_0, X_m^0, P_m). In particular, following Equations (8.8) and (8.11), we have (for $m \geq 1$)

$$X_m^0 = \left\{ \boldsymbol{x} \in \mathbb{R}^d : \quad \left\{ \begin{array}{l} \forall j \in [\![1,d]\!], \quad x_j \in \left\{0, \frac{1}{m}, \ldots, \frac{m-1}{m}, 1\right\} \\ 1 - \sum_{j=1}^d x_j \in \left\{0, \frac{1}{m}, \ldots, \frac{m-1}{m}, 1\right\} \end{array} \right. \right\} \quad ,$$

and we denote $p_0^{(i)}, i \in [\![1, \text{Card } X_m^0]\!]$ the shape functions of (T_0, X_m^0, P_m).

Any d-simplex of type (m) can be deduced from (T_0, X_m^0, P_m) using their affine equivalence. Using Equation (8.10) and the particular form of $\boldsymbol{A}^0$, the (bijective) affine map F_T that sends T_0 to a given d-simplex T (while conserving its barycentric coordinates) is given by

$$F_T : \boldsymbol{x}_0 \in T_0 \mapsto \boldsymbol{x} = F(\boldsymbol{x}_0) = \boldsymbol{a}^{(d+1)} + \boldsymbol{M}\boldsymbol{x}_0 \in T \quad , \tag{8.12}$$

where $\boldsymbol{M} \in \mathscr{M}_d(\mathbb{R})$ is the (invertible) matrix defined by

$$\boldsymbol{M} = \left(\begin{array}{c|c|c} & & \\ \boldsymbol{a}^{(1)} - \boldsymbol{a}^{(d+1)} & \ldots & \boldsymbol{a}^{(d)} - \boldsymbol{a}^{(d+1)} \\ & & \end{array} \right) \quad . \tag{8.13}$$

The inverse of F, which maps T to T_0 is therefore given by

$$F_T^{-1} : \boldsymbol{x} \in T \mapsto \boldsymbol{x}_0 = F_T(\boldsymbol{x}) = \boldsymbol{M}^{-1}(\boldsymbol{x} - \boldsymbol{a}^{(d+1)}) \in T_0 \quad . \tag{8.14}$$

In particular, note that given that F_T maintains the barycentric coordinates and following Equation (8.11), F_T^{-1} simply corresponds to the function that maps $\boldsymbol{x} \in T$ to its first d barycentric coordinates (with respect to T). These transformations are illustrated in Figure 8.3.

Any d-simplex of type (m) (T, X_m, P_m) can then be retrieved from (T_0, X_m^0, P_m) through

$$T = F_T(T_0), \quad X_m = F_T(X_m^0), \quad P_m = \text{span}\left\{ p^{(i)} = p_0^{(i)} \circ F_T^{-1} : i \in [\![1, \text{Card } X_m]\!] \right\} \quad , \tag{8.15}$$

where $p_0^{(i)}, i \in [\![1, \text{Card } X_m]\!]$ denote the shape functions of (T_0, X_m^0, P_m), i.e. the polynomial of P_m satisfying which is 1 at the i-th point of X_m^0 and 0 at any other point of X_m^0. We can therefore restrict ourselves to the study of the standard d-simplex of type (m).

In the particular case where $m \leq 2$, we now derive the expression of the shape functions of (T_0, X_m^0, P_m). First, we introduce the following notations:

$$\begin{aligned} \boldsymbol{a}_0^{(0)} &= \frac{1}{d+1} \sum_{i=1}^{d+1} \boldsymbol{a}_0^{(i)} = \frac{1}{d+1} \boldsymbol{1}_d \quad , \\ \boldsymbol{a}_0^{(ij)} &= \frac{1}{2}(\boldsymbol{a}_0^{(i)} + \boldsymbol{a}_0^{(j)}), \quad 1 \leq i < j \leq d+1 \quad , \end{aligned}$$

where once again $\{\boldsymbol{a}_0^{(i)}\}_{i=1}^{d+1}$ denotes the $(d+1)$ points whose convex hull defines the d-simplex T_0, and given by

$$\forall i \in [\![1, d+1]\!], \quad \forall j \in [\![1,d]\!], \quad [\boldsymbol{a}_0^{(i)}]_j = \delta_{ij} \quad .$$

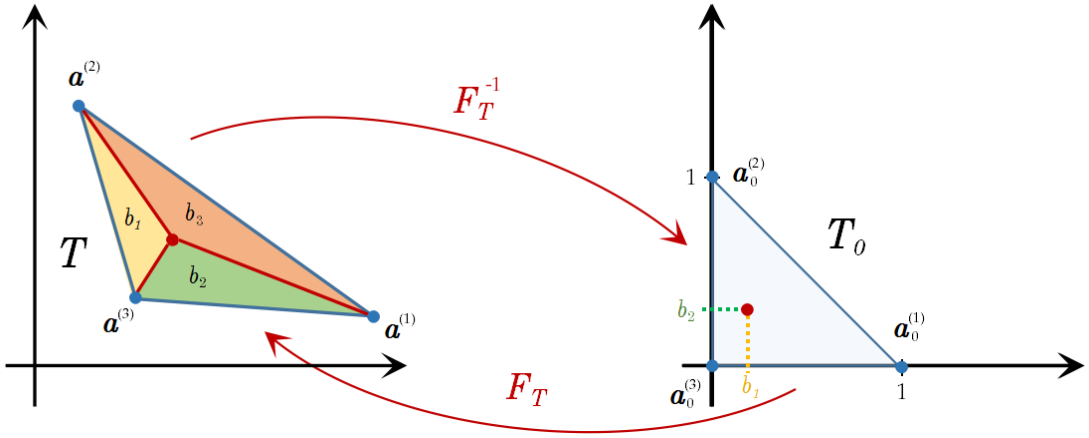


Figure 8.3: Illustration of the affine transformation from a general 2-simplex T to the standard 2-simplex T_0. b_1, b_2, b_3 denote the barycentric coordinate functions of T.

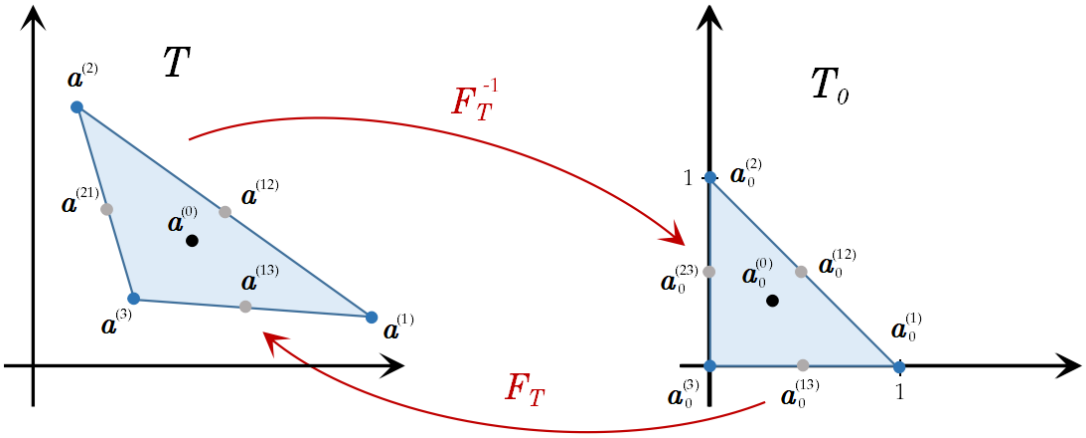


Figure 8.4: Illustration of the possible interpolation points from a general 2-simplex T (left) and for the standard 2-simplex T_0 (right).

	Interpolating points X_m^0	Shape functions	Number
$m=0$	$\{\boldsymbol{a}_0^{(0)}\}$	$p_0^{(0)}(\boldsymbol{x})=1$	1
$m=1$	$\{\boldsymbol{a}_0^{(i)}\}_{1\leq i\leq d+1}$	$p_0^{(i)}(\boldsymbol{x})=x_i$ where $x_{d+1}:=1-\sum_{i=1}^d x_i$	$d+1$
$m=2$	$\{\boldsymbol{a}_0^{(i)}\}_{1\leq i\leq d+1}\cup\{\boldsymbol{a}_0^{(ij)}\}_{1\leq i<j\leq d+1}$	$p_0^{(i)}(\boldsymbol{x})=x_i(2x_i-1)$ $p_0^{(ij)}(\boldsymbol{x})=4x_ix_j$ where $x_{d+1}:=1-\sum_{i=1}^d x_i$	$\frac{(d+1)(d+2)}{2}$

Table 8.1: Interpolating points X_m^0 and associated shape functions for the standard d-simplex of type m with $d\geq 1$ and $0\leq m\leq 2$.

Then, for $m\leq 2$, the points in X_m^0 are taken from $\{\boldsymbol{a}_0^{(0)}\}\cup\{\boldsymbol{a}_0^{(i)}\}_{1\leq i\leq d+1}\cup\{\boldsymbol{a}_0^{(ij)}\}_{1\leq i<j\leq d+1}$. These interpolation points are illustrated in Figure 8.4 for $d=2$.

Table 8.1 then gives the expression of the points composing X_m^0 and the corresponding shape functions of T_0 for $m\leq 3$. We denote by $\boldsymbol{p}_0^{(*)}$ the shape function associated with the point $\boldsymbol{a}_0^{(*)}\in X_m^0$.

Remark 8.1.2. As we may see in the subsequent sections of this chapter, the finite element method relies on the computation of integrals defined on simplices. The relations in Equation (8.15) are then used to express integrals over arbitrary d-simplices as integrals on the standard d-simplex, through a change of variable (cf. Theorem A.1.2).

Indeed, if $\varphi: T\to\mathbb{R}$ is a measurable function on a d-simplex T, then its integral over T can be written as an integral over the standard d-simplex T_0 as

$$\int_T \varphi(\boldsymbol{x})d\boldsymbol{x}=|\boldsymbol{M}|\int_{T_0}\varphi\circ F(\boldsymbol{x}_0)d\boldsymbol{x}_0 \quad ,$$

where $|\boldsymbol{M}|$ is the determinant of the matrix $\boldsymbol{M}$ defined in Equation (8.13). In particular, this determinant actually corresponds to twice the surface (resp. 6 times the volume) of the triangle (resp. tetrahedron) T when $d=2$ (resp. $d=3$).

8.1.2 Finite element method

We first assume that $\mathcal{M}\subset\mathbb{R}^d$ is a compact polyhedral set, i.e. $\mathcal{M}$ is a compact set formed by a finite union of polyhedrons of $\mathbb{R}^d$. A triangulation $\mathcal{T}_h$ of $\mathcal{M}$ is a finite decomposition of $\mathcal{M}$

$$\mathcal{M}=\bigcup_{T\in\mathcal{T}_h} T \quad ,$$

such that:

- Each element $T\in\mathcal{T}_h$ is a d-simplex.
- Two distinct simplices of $\mathcal{T}_h$ have disjoint interiors.
- Any face of a simplex $T_1\in\mathcal{T}_h$ is either the face of a distinct simplex $T_2\in\mathcal{T}_h$ or is part of the boundary of $\mathcal{M}$.

In particular, note that the intersection of two distinct simplices of $\mathcal{T}_h$ is either empty or it is a common face or a common vertex. The index h is called the size of the triangulation and denotes the largest diameter h_T of an element $T\in\mathcal{T}_h$:

$$h=\max_{T\in\mathcal{T}_h} h_T, \quad \text{where} \quad h_T:=\sup_{\boldsymbol{p}_1,\boldsymbol{p}_2\in T} d(\boldsymbol{p}_1,\boldsymbol{p}_2) \quad .$$

For a d-simplex T, let ρ_T be the radius of the largest ball of $\mathbb{R}^d$ that can be contained in T, and let h_T be the diameter of T. A family of triangulations $\{\mathcal{T}_h\}_{h\in H}$, where $H\subset]0,+\infty[$, is

called *shape regular* if there exists a constant $C > 0$ such that $\forall h \in H$, $\forall T \in \mathcal{T}_h$, $\rho_T \geq Ch_T$. Besides, $\{\mathcal{T}_h\}_{h\in H}$ is called *quasi-uniform* if there exists a constant $C' > 0$ such that $\forall h \in H$, $\forall T \in \mathcal{T}_h$, $h_T \geq Ch$.

Let $m \geq 0$. We associate to each $T \in \mathcal{T}_h$ a Lagrange finite element (T, X_m, P_m) which is a d-simplex of type (m). Consider then the set $\mathcal{X}_h \subset \mathcal{M}$ defined as the union of all the interpolating sets X_m corresponding to the finite elements of the triangulation $\mathcal{T}_h$:

$$\mathcal{X}_h = \bigcup_{\substack{(T,X_m,P_m)\\ T\in\mathcal{T}_h}} X_m \quad .$$

The elements of $\mathcal{X}_h$ are points of $\mathcal{M}$ called *nodes of the triangulation* $\mathcal{T}_h$.

Note that if we consider two of such finite elements (T, X_m, P_m) and $(\widehat{T}, \widehat{X}_m, P_m)$ such that T and $\widehat{T}$ have a common face U, then the interpolating points of X_m and $\widehat{X}_m$ that lie in U coincide, i.e. $X_m \cap U = \widehat{X}_m \cap U$. Indeed, this is a direct consequence of the definition of X_m and $\widehat{X}_m$ (cf. Equation (8.8)) and of the fact that the barycentric coordinates of a point of U are the same whether they are considered with respect to T or $\widehat{T}$. Using then the fact that $X_m \cap U$ and $\widehat{X}_m \cap U$ are $P_{m|U}$-unisolvent, we deduce that the functions of P_m defined on T or $\widehat{T}$ coincide along any common face $U = T \cap \widehat{T}$ as long as they coincides on the points $X_m \cap U = \widehat{X}_m \cap U$.

Let then $\Pi_h\varphi$ be the function defined for any square-integrable function $\varphi : \mathcal{M} \to \mathbb{R}$ by

$$\forall T \in \mathcal{T}_h, \quad \forall \boldsymbol{x} \in T, \quad \Pi_h\varphi(\boldsymbol{x}) = \Pi_T\varphi(\boldsymbol{x}) \quad ,$$

where Π_T is the P_m-interpolator associated with the finite element (T, X_m, P_m), as defined in Equation (8.1). Π_h is therefore well-defined (including along the faces of the simplices of $\mathcal{T}_h$) and is a continuous function of $\mathcal{M}$. Besides, on each simplex $T \in \mathcal{T}_h$ with associated finite element (T, X_m, P_m), it coincides with the P_m-interpolate of φ.

We now introduce the set V_h of (continuous) functions of $\mathcal{M}$ defined by

$$V_h = \{\Pi_h\varphi : \varphi \in L^2(\mathcal{M})\} \quad .$$

Then V_h is a vector subspace of $L^2(\mathcal{M})$ of dimension $N_h = |\mathcal{X}_h|$, and is called *finite element space*. Indeed, if we denote

$$\mathcal{X}_h = \{\boldsymbol{x}^{(j)}\}_{1\leq j\leq N_h} \quad ,$$

then a basis for V_h is provided by the set of functions $\{\psi_j\}_{1\leq j\leq N_h} \subset V_h$ where for each $j \in [\![1, N_h]\!]$, the function ψ_j is defined by the relation:

$$\forall k \in [\![1, N_h]\!], \quad \psi_j(\boldsymbol{x}^{(k)}) = \delta_{jk} \quad . \tag{8.16}$$

Hence we have

$$V_h = \operatorname{span}\{\{\psi_j\}_{1\leq j\leq N_h}\} \quad , \tag{8.17}$$

where ψ_j is the unique function of V_h that is 1 at the node $\boldsymbol{x}^{(j)} \in \mathcal{X}_h$ and 0 at any other node of the triangulation. In particular,

$$\forall v \in V_h, \quad v = \sum_{j=1}^{N_h} v(\boldsymbol{x}^{(j)})\psi_j \quad .$$

Note that the basis functions $\{\psi_j\}_{1\leq j\leq N_h}$ have a limited support: indeed if $\boldsymbol{x}^{(j)} \in T$ then ψ_j coincides with the shape function associated with the interpolating point $\boldsymbol{x}^{(j)}$ of T (cf. for instance Table 8.1), and otherwise ψ_j is zero over T. Hence the support of ψ_j is limited to the simplices that contain $\boldsymbol{x}^{(j)}$.

8.1.3 Triangulation of non-polyhedral sets

In this section, we no longer assume that $\mathcal{M}$ is a compact polyhedral set of $\mathbb{R}^d$. Instead, we now take $\mathcal{M}$ to be a compact subset of $\mathbb{R}^d$ with a (piecewise) smooth boundary $\partial\mathcal{M}$. The idea is to approximate $\mathcal{M}$ by a polyhedral set $\mathcal{M}_h$ such that any vertex on the boundary of $\mathcal{M}_h$ is a point of $\partial\mathcal{M}$. Then, $\mathcal{M}_h$ is triangulated as described above by a triangulation $\mathcal{T}_h$, according to the boundary conditions prescribed by the problem.

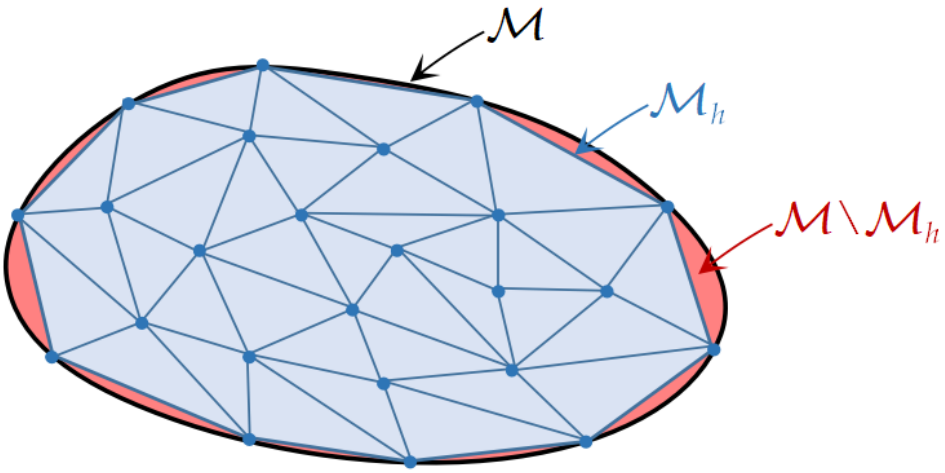


Figure 8.5: Triangulation of a non-polyhedral set $\mathcal{M}$ (delimited by the black boundary). The approximating polyhedral set $\mathcal{M}_h$ is represented in blue and the skin $\mathcal{M}\backslash\mathcal{M}_h$ in red.

In the general case where $\mathcal{M}$ has a curved boundary, the set $\mathcal{M}\backslash\mathcal{M}_h$, also called "skin", will be non-empty (cf. Figure 8.5). To account for the fact that the skin is not part of the triangulation, small adjustments can be made to extend the definition of the basis functions to the skin.

In the Dirichlet case, given that the value of the basis functions is zero on the boundary of $\mathcal{M}_h$ and that we also want them to be zero on $\partial\mathcal{M}$, we may simply set the value of all basis functions over the skin to be zero (Strang and Fix, 1973).

Using the same approach for the Neumann case however would result in a discontinuity of the basis functions across $\partial\mathcal{M}_h$. Instead, when linear shape functions are considered ($m = 1$), we may for instance extend into each piece of the skin the shape functions of the adjacent simplex (Strang and Fix, 1973). Given that their derivatives are piecewise constant, their values on the faces of $\partial\mathcal{M}_h$ will be "propagated" on the skin.

Another possible method to account for curved boundaries consists in deforming the simplices approximating $\mathcal{M}$ on its boundary so that their faces that lie on $\partial\mathcal{M}_h$ are themselves curved (Strang and Fix, 1973). Such elements are called isoparametric and are defined through a bijective transformation that maps the standard d-simplex T_0 to a deformed d-simplex $\tilde{T}$. In particular, the faces of the deformed simplex $\tilde{T}$ are polynomial surfaces defined using the same shape functions as the one used to build the finite elements.

8.1.4 Triangulation of surfaces of $\mathbb{R}^3$

We now consider the case where $\mathcal{M}$ is a smooth surface embedded in $\mathbb{R}^3$, and defined either parametrically or implicitly. $\mathcal{M}$ can therefore be seen as 2-submanifold of $\mathbb{R}^3$. Triangulating $\mathcal{M}$ consists in defining a locally planar surface $\mathcal{M}_h$ composed of triangles $T \subset \mathcal{M}_h$ that approximate locally $\mathcal{M}$, in the sense that $\forall \boldsymbol{p} \in T$, $\text{dist}(\boldsymbol{p}, \mathcal{M}) \leq \epsilon$, for some threshold $\epsilon > 0$ fixed in advance. Hence, we can write

$$\mathcal{M}_h = \bigcup_{T \in \mathcal{T}_h} T \quad ,$$

where $\mathcal{T}_h$ denotes the triangulation of $\mathcal{M}$, i.e. the set of triangles defining $\mathcal{M}_h$. In particular, the triangles of $\mathcal{T}_h$ must satisfy the following requirements:

- Given that each triangle $T \in \mathcal{T}_h$ can be seen as a 2-simplex defined by 3 points $\{\boldsymbol{a}^{(i)}\}_{1 \leq i \leq 3}$ of $\mathbb{R}^3$, we impose that these points lie in the original surface $\mathcal{M}$: $\boldsymbol{a}^{(i)} \in \mathcal{M}$.
- $\forall T, T' \in \mathcal{T}_h$, either $T = T'$, or $T \cap T' = \emptyset$, or $T \cap T'$ is a common edge or vertex of T and T'.

The notions of shape regular and quasi-uniform are directly extended from the case of the triangulation of compact sets of $\mathbb{R}^d$.

It can be showed that, for a small enough mesh size, each triangle $T \in \mathcal{T}_h$ can be mapped to a curved triangle $\tilde{T} \subset \mathcal{M}$, where curved triangles are defined as the image of the standard 2-simplex through a bijective application that maps it to $\mathcal{M}$. This is a consequence of the local coordinate mappings defining the surface $\mathcal{M}$. Conversely, in order to avoid double coverings, the triangulation $\mathcal{T}_h$ is built so that each point of $\mathcal{M}$ can be associated to (at most) and one point of $\mathcal{T}_h$, meaning that the $\mathcal{M}$ and $\mathcal{M}_h$ are in bijection.

Hence functions defined on $\mathcal{M}$ can be seen as defined on $\mathcal{M}_h$ and vice versa, using the bijection between both surfaces. Indeed, if $a : T \to \tilde{T}$ denotes the mapping that sends $T \in \mathcal{T}_h$ to its curved counterpart $\tilde{T} \in \mathcal{M}$, then we can associate to any $\varphi : T \to \mathbb{R}$ the function $\tilde{v} = v \circ a^{-1} : \tilde{T} \to \mathbb{R}$. Consequently the function space V_h on $\mathcal{T}_h$, which is defined in the same manner as it would be defined for the triangulation of a compact set of $\mathbb{R}^2$, can be seen as a set of functions defined on $\mathcal{M}$.

Finally, note that each triangle $T \in \mathcal{T}_h \subset \mathbb{R}^3$ in the triangulation of $\mathcal{M}_h$ is actually in bijection with the standard 2-simplex $T_0 \subset \mathbb{R}^2$. If $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \boldsymbol{x}^{(3)} \in \mathbb{R}^3$ denote the three vertices of T and $\boldsymbol{M}$ denotes the matrix defined by

$$\boldsymbol{M} = \left(\begin{array}{c|c} \boldsymbol{x}^{(1)} - \boldsymbol{x}^{(3)} & \boldsymbol{x}^{(2)} - \boldsymbol{x}^{(3)} \end{array} \right) \in \mathscr{M}_{3,2}(\mathbb{R}) \quad ,$$

then the application F defined by

$$F_T : \boldsymbol{y} \in T_0 \mapsto F(\boldsymbol{y}) = \boldsymbol{x}^{(3)} + \boldsymbol{M}\boldsymbol{y} \in T \tag{8.18}$$

is a bijective map sending $T_0 \subset \mathbb{R}^2$ to $T \subset \mathbb{R}^3$. Its inverse is given by

$$F_T^{-1} : \boldsymbol{x} \in T \mapsto F^{-1}(\boldsymbol{x}) = (\boldsymbol{M}^T\boldsymbol{M})^{-1}\boldsymbol{M}^T(\boldsymbol{x} - \boldsymbol{x}^{(3)}) \in T_0 \quad . \tag{8.19}$$

In particular, F_T^{-1} sends a point of T to its first two barycentric coordinates as defined by Equation (8.5). Hence finite elements can be built on $\mathcal{M}_h$ using the fact that all triangles are affine-equivalent to the standard 2-simplex.

8.2 Generalized random field approximation

Circling back to the discretization problem introduced in Section 7.3, finite element spaces are used to define the set of approximating functions V_n used to discretize generalized Gaussian fields (GeGFs) on a compact manifold $\mathcal{M}$. In particular, in the remainder of this chapter, these sets of function will rather be denoted by V_h where h will correspond to the mesh size of the triangulation, as the latter is directly linked to the dimension of the set.

8.2.1 Accounting for boundary conditions

The sets of basis functions V_h arising from finite element spaces are used to approximate GeGFs defined on the domain $\mathcal{M}$. The boundary conditions defining the eigenvalue problems on $\mathcal{M}$ should be accounted for as they are a key building block of the construction of GeGFs. In particular the set of approximating functions should be chosen as a subset of the domain of definition of the Laplacian. For the Dirichlet Laplacian/boundary conditions, this set is $H_0^1(\mathcal{M})$ and for the closed and Neumann case, the set is $H^1(\mathcal{M})$.

The closed eigenvalue problem arises when $\mathcal{M}$ is a manifold without boundary. In particular, this is the case when $\mathcal{M}$ is a closed surface, i.e. a surface that is topologically compact but has no boundary as a manifold (ex: sphere, torus). In that case, the set V_h arising from the triangulation of $\mathcal{M}$ can directly be used as a set of approximating functions of the problem given that no restriction is required and that it is a subset of $H^1(\mathcal{M})$.

The Dirichlet eigenvalue problem can be considered when $\mathcal{M}$ is a compact manifold with non-empty boundary. This is the case when $\mathcal{M}$ is a topological compact of $\mathbb{R}^d$ (with smooth boundary). Approximation functions reflecting this boundary condition should be used: hence, the approximation function should also be zero on the boundary of $\mathcal{M}$ so that they can lie in $H_0^1(\mathcal{M})$. Consequently, the set of approximating function that should be chosen is:

$$V_h^0 = \{\varphi \in V_h : \varphi_{|\partial\mathcal{M}} = 0\} \quad ,$$

where V_h is the set of basis functions defined by the triangulation of $\mathcal{M}$. In particular, V_h^0 is a vector subspace of V_h, whose dimension is equal to the number of unconstrained nodes of the triangulation, i.e. the number of nodes that are not on the boundary of $\mathcal{M}$ (or rather the boundary of the the polyhedron formed by the simplices of the triangulation). Indeed, V_h^0 is spanned by the basis functions of V_h (defined by Equation (8.16)) associated with these nodes (and only these nodes). Besides, $V_h^0 \subset H_0^1(\mathcal{M})$.

As for the Neumann eigenvalue problem, it can be considered for the same types of domains $\mathcal{M}$ as the Dirichlet problem. The set of approximation functions can be taken to be the whole set of basis functions V_h, which is a subset of $H^1(\mathcal{M})$. The boundary conditions will be implicitly enforced by definition of the Neumann Laplacian.

8.2.2 Error analysis of the finite element approximation

In this section, a convergence result of the finite element approximation of a GeGF is exposed. This result is simply an extension of Theorem 2.10 in (Bolin et al., 2018), and is proved in the exact same way.

First, we recall some notations. Let $(V_h)_{h\in]0,1]}$ be a family of finite element spaces indexed by a mesh size h over a domain $\mathcal{M} \subset \mathbb{R}^d$. In particular, following the previous subsection, the finite element spaces are defined so that they account for boundary conditions. We denote $n_h = \dim(V_h)$ the number of basis functions associated with the triangulation[1] of $\mathcal{M}$ with mesh size h.

Let $-\Delta_{\mathcal{M}}$ denote the Laplace-Beltrami operator, defined over $L^2(\mathcal{M})$, and let $-\Delta_h$ denote its discretization over V_h, as defined in Equation (7.26). Let $\{\lambda_j\}_{j\in\mathbb{N}}$ and $\{\lambda_{k,h}\}_{1\leq k\leq n_h}$ be the eigenvalues of $-\Delta_{\mathcal{M}}$ and $-\Delta_h$, listed in non-decreasing order.

Let $\gamma : \mathbb{R}_+ \to \mathbb{R}$ such that $\sum_{j\in\mathbb{N}} \gamma(\lambda_j)^2 < \infty$.

The following assumptions are considered to derive an error bound between a GeGF Z defined by Equation (7.12) and its finite element approximation defined by Equation (7.36).

Assumption 8.1 (Growth of the eigenvalues of $-\Delta_{\mathcal{M}}$)**.** *There exist three constants $\alpha > 0$, $c_\lambda > 0$ and $C_\lambda > 0$ such that the eigenvalues $\{\lambda_j\}_{j\in\mathbb{N}}$ satisfy*

$$\forall j \in \mathbb{N}, \quad \lambda_j > 0 \Rightarrow c_\lambda j^\alpha \leq \lambda_j \leq C_\lambda j^\alpha \quad .$$

Assumption 8.2 (Derivative of γ)**.** *$\gamma : \mathbb{R}_+ \to \mathbb{R}$ is derivable on $\mathbb{R}_+$, and there exist $C_{\text{Deriv}} > 0$ and $a \geq 0$ such that*

$$\forall x > 0, \quad |\gamma'(x)| \leq \frac{C_{\text{Deriv}}}{x^a} \quad .$$

Assumption 8.3 (Asymptotic behavior of γ)**.** *There exists a constant $\beta > 0$ such that $\gamma : \mathbb{R}_+ \to \mathbb{R}$ satisfies $|\gamma(\lambda)| = \underset{\lambda\to+\infty}{\mathcal{O}}\left(\lambda^{-\beta}\right)$, i.e.*

$$\exists C_\gamma > 0, \exists R_\gamma > 0, \quad \lambda \geq R_\gamma \Rightarrow |\gamma(\lambda)| \leq C_\gamma \lambda^{-\beta} \quad .$$

Assumption 8.4 (Dimension of the finite element space)**.** *There exist two constants $\tilde{d} > 0$, $C_{\text{FES}} > 0$ such that*

$$n_h = \dim(V_h) = C_{\text{FES}} h^{-\tilde{d}} \quad .$$

Assumption 8.5 (Mesh size)**.** *The mesh size h shall satisfy:*

$$h \leq \left(\frac{1}{C_{\text{FES}}} \left\lceil \left(\frac{R_\gamma}{c_\lambda} \right)^{1/\alpha} \right\rceil \right)^{-1/\tilde{d}} \quad ,$$

[1] In particular n_h is equal to either the total number of interpolation points (for closed and Neumann boundary conditions) or the number of interpolation points that do not lie on the boundary of the domain (for Dirichlet boundary conditions).

where C_{FES}, R_γ, α and c_λ are the constants defined in Assumptions 8.1, 8.3 and 8.4. In particular, following Assumptions 8.1 and 8.4, for all $j \geq n_h$, $\lambda_j \geq R_\gamma$.

Assumption 8.6 (Eigenvalues and eigenvectors of $-\Delta_h$)**.** *There exist constants $H_0 \in]0,1[$, $C_1, C_2 > 0$, and exponents $r, s, q > 0$ such that*

$$\forall h \in]0, H_0[, \quad \forall k \in [\![1, n_h]\!], \quad \begin{cases} 0 \leq \lambda_{k,h} - \lambda_k \leq C_1 h^r \lambda_j^q \\ \|e_{k,h} - e_k\|^2_{L^2(\mathcal{M})} \leq C_2 h^{2s} \lambda_k^q \end{cases} ,$$

where $\{\lambda_{k,h}\}_{1 \leq k \leq n_h}$ and $\{e_{k,h}\}_{1 \leq k \leq n_h}$ are the eigenvalues and eigenvectors of the discretized operator $-\Delta_h$ associated with a mesh size h.

Following the notations of the previous sections, let $\mathcal{Z}$ and $\mathcal{Z}_h$ be the random fields defined by:

$$\mathcal{Z} = \gamma(L)\mathcal{W} = \sum_{j \in \mathbb{N}} W_j \gamma(\lambda_j) e_j \tag{8.20}$$

and

$$\mathcal{Z}_h = \gamma(L_h)\mathcal{W}_h = \sum_{k=1}^{n_h} W_k \gamma(\lambda_{k,h}) e_{k,h} \quad , \tag{8.21}$$

where $\{W_j\}_{j \in \mathbb{N}}$ is a sequence of independent standard Gaussian variables. The expected approximation error of $\mathcal{Z}$ by $\mathcal{Z}_h$ is then defined by :

$$\|\mathcal{Z} - \mathcal{Z}_h\|_{L^2(\Omega;\mathcal{M})} = \sqrt{\mathbb{E}\left[\|\mathcal{Z} - \mathcal{Z}_h\|^2_{L^2(\mathcal{M})}\right]} \tag{8.22}$$

and can be bounded using the following result.

Theorem 8.2.1. *Let V_h, γ, $-\Delta_{\mathcal{M}}$ and $-\Delta_h$ satisfying Assumptions 8.1 to 8.6. Assume that the function γ is such that $a < q + \beta$ in Assumption 8.2 and that the growth of eigenvalues α, defined in Assumption 8.1, satisfies*

$$\frac{1}{2\beta} < \alpha \leq \min\left\{\frac{2s}{q\tilde{d}}; \frac{r}{(q+\beta-a)\tilde{d}}\right\} \quad . \tag{8.23}$$

Then, for $h > 0$ sufficiently small, the approximation error of the GeGF $\mathcal{Z}$ (defined by Equation (8.20)) by its finite element discretization $\mathcal{Z}_n$ (defined by Equation (8.21)) is bounded by

$$\|\mathcal{Z} - \mathcal{Z}_h\|_{L^2(\Omega;\mathcal{M})} \leq M h^{\min\left\{s;\, \tilde{d}(\alpha\beta - 1/2);\, r\right\}} \quad , \tag{8.24}$$

where $M > 0$ is a constant independent of h.

Proof. See Appendix D.2 for a proof of this theorem. □

In the applications that will be presented in the next chapter, Assumptions 8.1 to 8.6 are satisfied and therefore, Theorem 8.2.1 applies. Indeed,

- Assumption 8.1 is a direct consequence of the Weyl asymptotic formula (cf. Theorem 6.5.4), and will be satisfied as long as a compact connected Riemannian manifold is considered;
- Assumptions 8.2 and 8.3 depend only on a suitable choice of spectral density (or equivalently covariance function) of the random fields with which we work;
- Assumptions 8.4 and 8.5 depend only on the size of the triangulation, which is also set by the user;
- Assumption 8.6 is a consequence of (Strang and Fix, 1973, Theorems 6.1 & 6.2) for fine enough triangulations.

In particular, we refer the reader to the work of Bolin et al. (2018) for an example of possible values taken by the parameters defined in these assumptions.

8.3 Example of construction of a finite element approximation

We assume in this section that a GeGF $\mathcal{Z}$ is built from a Neumann Laplacian on a compact 2-manifold $\mathcal{M}$. We assume that a finite element space $V_h = \text{span}\{\psi_j : j \in [\![1, N_h]\!]\}$ has been built on $\mathcal{M}$ from shape functions taken in P_1. Hence the functions in V_h are piecewise-linear and continuous functions of $\mathbb{R}^2$.

We seek to built the discretization $\mathcal{Z}_{N_h}$ of $\mathcal{Z}$ described in Theorem 7.3.5. This comes down to building the matrices $\boldsymbol{S}$ and $\boldsymbol{C}$ defining the covariance matrix of the weights in Equation (7.36), through the relation in Equation (7.37).

Recall first that each basis function ψ_j is related to the node $\boldsymbol{x}^{(j)}$ of the triangulation through Equation (8.17). In particular, on each triangle containing $\boldsymbol{x}^{(j)}$, ψ_j coincides with the basis function associated with $\boldsymbol{x}^{(j)}$; and on the triangles that do not contain $\boldsymbol{x}^{(j)}$, ψ_j is zero.

To each triangle $T \in \mathcal{T}_h$ we associate a set $(j_1, \dots, j_{d+1}) \in [\![1, N_h]\!]^{d+1}$ such that the points $\boldsymbol{x}^{(j_1)}, \dots, \boldsymbol{x}^{(j_{d+1})}$ are the vertices of T. In particular, if T_0 denotes the standard d-simplex, the map

$$F_T : \boldsymbol{y} \in T_0 \mapsto F_T(\boldsymbol{y}) = \boldsymbol{x}^{(j_{d+1})} + \boldsymbol{M}_T \boldsymbol{y} \in T \quad , \tag{8.25}$$

where

$$\boldsymbol{M}_T = \left(\boldsymbol{x}^{(j_1)} - \boldsymbol{x}^{(j_{d+1})} | \dots | \boldsymbol{x}^{(j_d)} - \boldsymbol{x}^{(j_{d+1})} \right)$$

is a bijective map that sends T_0 to T. Note that F_T^{-1} is given by

$$F_T^{-1} : \boldsymbol{x} \in T \mapsto F_T^{-1}(\boldsymbol{x}) = \boldsymbol{P}_T \left(\boldsymbol{x} - \boldsymbol{x}^{(j_3)} \right) \in T_0 \quad , \tag{8.26}$$

where $\boldsymbol{P}_T = \boldsymbol{M}_T^{-1}$ if $\mathcal{M}$ is a d-submanifold of $\mathbb{R}^d$ (for instance a polyhedral set of $\mathbb{R}^d$) and $\boldsymbol{P}_T = (\boldsymbol{M}_T^T \boldsymbol{M}_T)^{-1} \boldsymbol{M}_T^T$ if $\mathcal{M}$ is a d-submanifold of $\mathbb{R}^{d+1}$ (for instance a surface in $\mathbb{R}^3$). Then, F_T^{-1} maps any point of T to its first two barycentric coordinates.

Given a point $\boldsymbol{x}^{(j)} \in \mathcal{X}_h$, we denote

$$\mathcal{T}_h^{(j)} = \{T \in \mathcal{T}_h : \boldsymbol{x}^{(j)} \text{ is one the vertices of } T\} \quad . \tag{8.27}$$

Consider then some $T = (j_1, \dots, j_{d+1}) \in \mathcal{T}_h^{(j)}$ and denote $k_j \in [\![1, d+1]\!]$ the index such that $j = j_{k_j}$. Then the restriction of ψ_j to T is given by

$$\psi_{j|T} = p_0^{(k_j)} \circ F_T^{-1} \quad ,$$

where the expression of the function $p_0^{(k_j)}$ is given in Table 8.1. As for the gradient of ψ_j on T it is therefore given by

$$\forall \boldsymbol{x} \in T, \quad \nabla \psi_j(\boldsymbol{x}) = \boldsymbol{P}_T^T \boldsymbol{c}_{k_j} \quad ,$$

where $\forall k \in [\![1, d]\!]$, $\boldsymbol{c}_k$ denotes the k-th canonical basis vector of $\mathbb{R}^d$ and we set

$$\boldsymbol{c}_{d+1} = -\sum_{k=1}^{d} \boldsymbol{c}_k = -\boldsymbol{1}_d$$

As expected, note that the gradient of ψ_j is constant over each triangle of the triangulation.

8.3.1 Construction of the mass matrix

Recall the expression of the elements of the mass matrix $\boldsymbol{C} \in \mathscr{M}_{N_h}(\mathbb{R})$ in Equation (7.27). It is a common practice to actually replace the matrix $\boldsymbol{C}$ by a diagonal matrix (also denoted $\boldsymbol{C}$) with entries given by

$$\forall j \in [\![1, N_h]\!], \quad C_{jj} = \langle \psi_j, 1 \rangle_{L^2(\mathcal{M})} \quad .$$

This approach, called mass lumping, bears negligible effects on the outcome of the approximation while bringing major simplifications (Chen and Thomée, 1985; Lindgren et al., 2011). Indeed, the matrix $\boldsymbol{C}$ being now diagonal, its (inverse) principal square-root is given with no extra

computational effort by taking the (inverse) square-root of its diagonal entries. This property will be particularly useful when computing the scaled stiffness matrix $\boldsymbol{S}$.

Mass lumping is applied in the following. The elements C_{jj} are defined by integrals over $(\mathcal{M}, g)$. Given that the triangulation of $\mathcal{M}$ is a partition of this set, the integral over $\mathcal{M}$ can be split into a sum of integrals over each simplex $T \in \mathcal{T}_h$. On each simplex T, assuming that there exists a coordinate chart (U_T, x_T) containing T, the integral can be expressed using local coordinates, thus giving

$$C_{jj} = \int_{\mathcal{M}} \psi_j dV_g = \sum_{T \in \mathcal{T}_h} \int_T \psi_j dV_g = \sum_{T \in \mathcal{T}_h} \int_{x_T(T)} \psi_j \circ x_T^{-1}(\boldsymbol{t}) \sqrt{|g^{x_T}|(x_T^{-1}(\boldsymbol{t}))} d\boldsymbol{t} \quad ,$$

where $x_T(T) \subset \mathbb{R}^d$ is the image of T through x_T and $|g^{x_T}|$ denotes the determinant of the representative matrix of the metric g with respect to the coordinate chart (U_T, x_T) at any point of $T \subset U_T$. In particular, given that the simplices T are portions of $\mathbb{R}^d$, we can choose x_T to be the identity mapping Id and U_T to be a small enough open set of $\mathcal{M}$ containing T. This gives

$$C_{jj} = \sum_{T \in \mathcal{T}_h} \int_T \psi_j(\boldsymbol{p}) \sqrt{|g|(\boldsymbol{p})} d\boldsymbol{p} = \sum_{T \in \mathcal{T}_h^{(j)}} \int_T \psi_j(\boldsymbol{p}) \sqrt{|g|(\boldsymbol{p})} d\boldsymbol{p} \quad ,$$

where $\mathcal{T}_h^{(j)}$ is defined in Equation (8.27) for any $\boldsymbol{p} \in T$, $|g|(\boldsymbol{p})$ now denotes the determinant of the representative matrix $\boldsymbol{G}(\boldsymbol{p})$ of the metric g with respect to the chart (U_T, Id). In particular, $\boldsymbol{G}(\boldsymbol{p})$ can be seen as a matrix defining local anisotropies over T through Equation (7.23).

Finally, a change of variable $\boldsymbol{y} = F_T(\boldsymbol{p})$ in these integrals allows to express them as integrals over the same domain T_0

$$C_{jj} = \sum_{T \in \mathcal{T}_h^{(j)}} \int_{T_0} \psi_j \circ F_T(\boldsymbol{y}) \sqrt{|g| \circ F_T(\boldsymbol{y})} \sqrt{\det J_{F_T}(\boldsymbol{y})^T J_{F_T}(\boldsymbol{y})} d\boldsymbol{y} \quad ,$$

where J_{F_T} denotes the Jacobian matrix of F_T. This gives the following expression of F_T and ψ_j:

$$C_{jj} = \sum_{T \in \mathcal{T}_h^{(j)}} \sqrt{\det \boldsymbol{M}_T^T \boldsymbol{M}_T} \int_{T_0} p_0^{(k_j)}(\boldsymbol{y}) \sqrt{|g| \circ F_T(\boldsymbol{y})} d\boldsymbol{y} \quad , \tag{8.28}$$

where $k_j \in [\![1, d+1]\!]$ is the vertex index of the triangulation node $\boldsymbol{x}^{(j)}$ in $T \in \mathcal{T}_h^{(j)}$ and the expression of $p_0^{(*)}$ is given in Table 8.1.

In practice, the computation of these elements is eased by assuming that the field of matrices $\boldsymbol{G}(\boldsymbol{p})$ is constant across each triangle:

$$\forall T \in \mathcal{T}_h, \quad \forall \boldsymbol{p} \in T, \quad \boldsymbol{G}(\boldsymbol{p}) = \boldsymbol{G}_T \quad ,$$

for some (symmetric) positive definite matrix $\boldsymbol{G}_T$. For fine triangulation and smoothly varying matrices $\boldsymbol{G}(\boldsymbol{p})$ this approximation is valid and we usually take $\boldsymbol{G}_T$ to be the value of $\boldsymbol{G}(\boldsymbol{p})$ at the center of gravity of T or the mean of the values $\boldsymbol{G}(\boldsymbol{p})$ at the vertices of T. Hence,

$$C_{jj} = \sum_{T \in \mathcal{T}_h^{(j)}} \sqrt{\det \boldsymbol{M}_T^T \boldsymbol{M}_T} \sqrt{\det \boldsymbol{G}_T} \int_{T_0} p_0^{(k_j)}(\boldsymbol{y}) d\boldsymbol{y} \quad ,$$

where the remaining integral is actually the volume of the $d+1$ standard simplex (cf. (Stein, 1966)). Hence,

$$C_{jj} = \frac{1}{(d+1)!} \sum_{T \in \mathcal{T}_h^{(j)}} \sqrt{\det \boldsymbol{M}_T^T \boldsymbol{M}_T} \sqrt{\det \boldsymbol{G}_T} \quad . \tag{8.29}$$

8.3.2 Construction of the stiffness matrix

Following Equation (7.27), the elements of the stiffness matrix $\boldsymbol{R}$ are given by

$$\forall i, j \in [\![1, N_h]\!], \quad R_{ij} = \langle \nabla_{\mathcal{M}} \psi_i, \nabla_{\mathcal{M}} \psi_j \rangle_{L^2(\mathcal{M})} \quad .$$

Once again the integral over $\mathcal{M}$ is decomposed as a sum of integrals over each simplex of the triangulation, thus giving

$$R_{ij} = \sum_{T \in \mathcal{T}_h} \int_T \nabla\psi_i(\boldsymbol{p})^T \boldsymbol{G}(\boldsymbol{p})^{-1} \nabla\psi_j(\boldsymbol{p}) \sqrt{|g|(\boldsymbol{p})} d\boldsymbol{p} \quad ,$$

where $\boldsymbol{G}$ and $|g|$ are defined as in the previous section. Using the fact that the gradients have limited support, we get

$$R_{ij} = \sum_{T \in \mathcal{T}_h^{(i)} \cap \mathcal{T}_h^{(j)}} \int_T \nabla\psi_i(\boldsymbol{p})^T \boldsymbol{G}(\boldsymbol{p})^{-1} \nabla\psi_j(\boldsymbol{p}) \sqrt{|g|(\boldsymbol{p})} d\boldsymbol{p} \quad .$$

Applying once again the change of variable $\boldsymbol{y} = F_T(\boldsymbol{p})$ finally gives,

$$R_{ij} = \sum_{T \in \mathcal{T}_h^{(i)} \cap \mathcal{T}_h^{(j)}} \sqrt{\det \boldsymbol{M}_T^T \boldsymbol{M}_T} \int_{T_0} \boldsymbol{c}_{k_i}^T \boldsymbol{P}_T \boldsymbol{G}(F_T(\boldsymbol{y}))^{-1} \boldsymbol{P}_T^T \boldsymbol{c}_{k_j} \sqrt{|g|(F_T(\boldsymbol{y}))} d\boldsymbol{y}$$

or equivalently

$$R_{ij} = \sum_{T \in \mathcal{T}_h^{(i)} \cap \mathcal{T}_h^{(j)}} \sqrt{\det \boldsymbol{M}_T^T \boldsymbol{M}_T} \cdot \boldsymbol{c}_{k_i}^T \boldsymbol{P}_T \boldsymbol{H}_T \boldsymbol{P}_T^T \boldsymbol{c}_{k_j} \quad , \tag{8.30}$$

where $\boldsymbol{H}_T$ is the matrix defined by

$$\boldsymbol{H}_T = \int_{T_0} \sqrt{|g|(F_T(\boldsymbol{y}))} \boldsymbol{G}(F_T(\boldsymbol{y}))^{-1} d\boldsymbol{y} \quad ,$$

and the integral of a matrix is understood as the integral of its entries. Note that if we once gain assume that the matrices $\boldsymbol{G}(\boldsymbol{p})$ are constant on each triangle the coefficients R_{ij} would be given by

$$R_{ij} = \frac{1}{d!} \sum_{T \in \mathcal{T}_h^{(i)} \cap \mathcal{T}_h^{(j)}} \sqrt{\det \boldsymbol{M}_T^T \boldsymbol{M}_T} \sqrt{\det \boldsymbol{G}_T} \cdot \boldsymbol{c}_{k_i}^T \boldsymbol{P}_T \boldsymbol{G}_T^{-1} \boldsymbol{P}_T^T \boldsymbol{c}_{k_j} \quad . \tag{8.31}$$

For the element R_{ij} to be non-zero, the nodes $\boldsymbol{x}^{(i)} \in \mathcal{X}_h$ and $\boldsymbol{x}^{(j)} \in \mathcal{X}_h$ must be the vertices of at least one common simplex T. This means that $\boldsymbol{x}^{(i)}$ and $\boldsymbol{x}^{(j)}$ must form the edge of one of the triangles (or tetrahedron) of the triangulation. Thus, the number of non-zero entries of $\boldsymbol{R}$ is equal to the number of simplex edges in the triangulation, thus yielding the fact that the matrix $\boldsymbol{R}$ will be sparse.

Recall now the concluding remarks of Section 7.3, which pointed out the link between approximation weights and graph signals. The particular form of Equation (8.31) actually allows to specify the graph on which the signal lies. Indeed, denote by $\mathcal{G}_h$ the graph whose vertices are the nodes of the triangulation $\mathcal{X}_h$ and such that $i \sim j$ whenever $\boldsymbol{x}^{(i)}$ and $\boldsymbol{x}^{(j)}$ form the edge of one of the simplices of the triangulation. In particular, $\mathcal{G}_h$ is an undirected simple graph. Then $\boldsymbol{R}$ is a shift operator of $\mathcal{G}_h$. The following proposition even goes a step further.

Proposition 8.3.1. *Let $\mathcal{G}_h$ be the graph defined from the vertices of the triangulation of a domain $\mathcal{M}$ (as described above) with linear basis functions $\{\psi_j\}_{j \in [\![1, N_h]\!]}$. Let assume that each edge (i, j) of $\mathcal{G}_h$ has weight w_{ij} given by*

$$w_{ij} = -\langle \nabla_{\mathcal{M}} \psi_i, \nabla_{\mathcal{M}} \psi_j \rangle_{L^2(\mathcal{M})}, \quad i \sim j \quad .$$

Then the stiffness matrix $\boldsymbol{R}$ defined from the basis functions is the graph Laplacian of $\mathcal{G}_h$.

Proof. Note that $\forall i \neq j$ vertices of the graph/triangulation, $w_{ii} = 0$ and $w_{ij} = -R_{ij}$. Note then that the degree d_i of the vertex i of $\mathcal{G}_h$ is given by

$$\begin{aligned} d_i = \sum_{j=1}^n w_{ij} &= -\sum_{\substack{j=1 \\ j \neq i}}^n \sum_{T \in \mathcal{T}_h^{(i)} \cap \mathcal{T}_h^{(j)}} \sqrt{\det \boldsymbol{M}_T^T \boldsymbol{M}_T} \cdot \boldsymbol{c}_{k_i}^T \boldsymbol{P}_T \boldsymbol{H}_T \boldsymbol{P}_T^T \boldsymbol{c}_{k_j} \\ &= -\sum_{T \in \mathcal{T}_h^{(i)}} \sum_{\substack{j \in \mathcal{X}_h \cap T \\ j \neq i}} \sqrt{\det \boldsymbol{M}_T^T \boldsymbol{M}_T} \cdot \boldsymbol{c}_{k_i}^T \boldsymbol{P}_T \boldsymbol{H}_T \boldsymbol{P}_T^T \boldsymbol{c}_{k_j} \quad . \end{aligned}$$

Noting then that $\sum_{j\in\mathcal{X}_h\cap T} \boldsymbol{c}_{k_j} = \boldsymbol{0}$ by definition of the vectors $\boldsymbol{c}_k$ and of the indices k_j, we get

$$\sum_{j=1}^{n} w_{ij} = -\sum_{T\in\mathcal{T}_h^{(i)}} \sqrt{\det \boldsymbol{M}_T^T\boldsymbol{M}_T}\cdot c_{k_i}^T \boldsymbol{P}_T\boldsymbol{H}_T\boldsymbol{P}_T^T(-\boldsymbol{c}_{k_i}) = R_{ii} \quad .$$

Hence the off-diagonal entries of $\boldsymbol{R}$ are minus the weights of $\mathcal{G}_h$ and the diagonal entries of $\boldsymbol{R}$ are the degrees of the vertices of $\mathcal{G}_h$. $\boldsymbol{R}$ is therefore the graph Laplacian of $\mathcal{G}_h$.

□

Note that, given that $\boldsymbol{C}$ is taken diagonal, the scaled stiffness matrix $\boldsymbol{S} = \boldsymbol{C}^{-1/2}\boldsymbol{S}\boldsymbol{C}^{-1/2}$ is also a shift operator $\mathcal{G}_h$ and is a sparse matrix. Using graph filtering algorithms for the simulations, prediction and inference of the approximation weights is therefore expected to yield good computational and storage performances.

8.3.3 Particular case: constant anisotropy on a 2D grid

In this section, we look into the particular case where:

- The domain of study $\mathcal{M}$ is a rectangular domain of $\mathbb{R}^2$.
- The coefficients of the metric are constant, meaning that the field of matrices $\{\boldsymbol{G}(\boldsymbol{p})\}_{\boldsymbol{p}\in\mathcal{M}}$ is constant over $\mathcal{M}$. We then denote by $\boldsymbol{G}$ its value.

As we may see, we can leverage the redundancy of the entries of the stiffness matrix for storage and computational gains.

The triangulation $\mathcal{M}$ is performed in two steps. First, a regular grid, with steps (l_1, l_2) is defined over $\mathcal{M}$. Then each rectangle $dx \times dy$ is divided into two triangles by cutting them along the same diagonal. We assume here that all rectangles were cut along their top-left to bottom-right. We call $\mathcal{T}_h^r$ this "grid" triangulation of $\mathcal{M}$.

To each triangle $T \in \mathcal{T}_h^r$, we associate the vertex indices (j_1, j_2, j_3) such that $\boldsymbol{x}^{(j_3)}$ is the corner of T and $\boldsymbol{x}^{(j_1)}$ (resp $\boldsymbol{x}^{(j_2)}$) is the vertex of T horizontally (resp. vertically) aligned with $\boldsymbol{x}^{(j_3)}$. Then, by definition of the matrices $\boldsymbol{M}_T$ we have

$$\forall T \in \mathcal{T}_h^r, \quad \boldsymbol{M}_T = \boldsymbol{M} = \epsilon \begin{pmatrix} l_1 & \\ & l_2 \end{pmatrix}, \quad \epsilon \in \{-1, 1\}$$

The matrices $\boldsymbol{M}_T = \boldsymbol{M}$, and $\boldsymbol{P}_T = \boldsymbol{M}_T^{-1} = \boldsymbol{M}^{-1} = \boldsymbol{P}$, are therefore independent of T.

Leveraging the fact that the metric coefficients are constant, the expressions of the coefficients C_{jj} and R_{ij} are therefore simplified to

$$C_{jj} = \frac{1}{6}\sum_{T\in\mathcal{T}_h^{(j)}} l_1 l_2\sqrt{\det \boldsymbol{G}} = \frac{l_1 l_2}{6}\sqrt{\det \boldsymbol{G}}\,\mathrm{Card}\left\{T \in \mathcal{T}_h^r : \boldsymbol{x}^{(j)} \in T\right\}, \quad j \in [\![1, N_h]\!] \tag{8.32}$$

and

$$R_{ij} = \frac{1}{2}\sum_{T\in\mathcal{T}_h^{(i)}\cap\mathcal{T}_h^{(j)}} l_1 l_2\sqrt{\det \boldsymbol{G}}\cdot c_{k_i}^T \boldsymbol{P}\boldsymbol{G}^{-1}\boldsymbol{P}^T\boldsymbol{c}_{k_j}, \quad i, j \in [\![1, N_h]\!] \quad . \tag{8.33}$$

Note that we now have $\forall i, j \in [\![1, N_h]\!]$,

$$R_{ij} = \sum_{T\in\mathcal{T}_h^{(i)}\cap\mathcal{T}_h^{(j)}} \boldsymbol{c}_{k_i}^T\tilde{\boldsymbol{H}}\boldsymbol{c}_{k_j}$$

where

$$\tilde{\boldsymbol{H}} = \frac{1}{2} l_1 l_2\sqrt{\det \boldsymbol{G}}\cdot \boldsymbol{P}\boldsymbol{G}^{-1}\boldsymbol{P}^T \quad .$$

These coefficients are non-zero only if $i = j$ or i and j form the edge of one of the triangles of the grid triangulation, i.e. i and j must be adjacent vertices in the triangulation graph. For a triangulation point $\boldsymbol{x}^{(i)}$, denote $(i_1, i_2) \in [\![1, n_1]\!] \times [\![1, n_2]\!]$ its grid coordinates. The only possible

neighbors of $\boldsymbol{x}^{(i)}$ are the points $\boldsymbol{x}^{(j)}$ with grid coordinates (j_1, j_2) such that $j_1 - i_1 \in \{-1, 0, 1\}$ and $j_2 - i_2 \in \{-1, 0, 1\} \backslash \{j_1 - i_1\}$.

Let us look at the different cases that arise. If $j_1 = i_1 + 1$ and $j_2 = i_2$, then the number of triangles containing i and j is equal to:

- 0 whenever $\boldsymbol{x}^{(i)}$ is on the right border of the grid,
- 1 whenever $\boldsymbol{x}^{(i)}$ is either on the top or the bottom border,
- 2 in any other case.

When such triangles exist, (i, j) is their edge along the direction x_1. Each triangle yields either $k_i = 3$ and $k_j = 1$ or $k_i = 3$ and $k_j = 1$, meaning finally that $R_{ij} = \delta_{i_1 n_1}(2 - \delta_{i_2 1} - \delta_{i_2 n_2}) \times \boldsymbol{c}_1^T \tilde{\boldsymbol{H}} \boldsymbol{c}_3 = -\delta_{i_1 n_1}(2 - \delta_{i_2 1} - \delta_{i_2 n_2}) \times (\tilde{H}_{11} + \tilde{H}_{12})$.

The same reasoning gives:

- If $j_1 = i_1 - 1$ and $j_2 = i_2$, $R_{ij} = -(1 - \delta_{i_1 1})(2 - \delta_{i_2 1} - \delta_{i_2 n_2}) \times (\tilde{H}_{11} + \tilde{H}_{12})$
- If $j_1 = i_1$ and $j_2 = i_2 + 1$, $R_{ij} = -(1 - \delta_{i_2 n_2})(2 - \delta_{i_1 1} - \delta_{i_1 n_1}) \times (\tilde{H}_{22} + \tilde{H}_{12})$
- If $j_1 = i_1$ and $j_2 = i_2 - 1$, $R_{ij} = -(1 - \delta_{i_2 1})(2 - \delta_{i_1 1} - \delta_{i_1 n_1}) \times (\tilde{H}_{22} + \tilde{H}_{12})$
- If $j_1 = i_1 + 1$ and $j_2 = i_2 - 1$, $R_{ij} = 2(1 - \delta_{i_1 n_1})(1 - \delta_{i_2 1})\tilde{H}_{12}$
- If $j_1 = i_1 - 1$ and $j_2 = i_2 + 1$, $R_{ij} = 2(1 - \delta_{i_2 n_2})(1 - \delta_{i_1 1})\tilde{H}_{12}$

The graph filtering algorithms we plan to use to compute for instance simulations of the approximation weights heavily rely on products between the scaled stiffness matrix $\boldsymbol{S} = \boldsymbol{C}^{-1/2}\boldsymbol{R}\boldsymbol{C}^{-1/2}$ and vectors $\boldsymbol{u}$. Such products are computed in three steps. First, the product $\boldsymbol{v} = \boldsymbol{C}^{-1/2}\boldsymbol{u}$ is computed, and amounts to the entry-wise multiplication of the entries of $\boldsymbol{u}$ by the diagonal elements of $\boldsymbol{C}$. Then the product $\boldsymbol{R}\boldsymbol{v}$ is computed. And finally the result is once again multiplied by $\boldsymbol{C}^{-1/2}$.

Following the particular expression of the elements of $\boldsymbol{R}$, the product $\boldsymbol{R}\boldsymbol{v}$ can actually be identified with the application of a convolution on an image with $n_1 \times n_2$ pixels and where the value at a pixel (i_1, i_2) is v_i (at least if we disregard the borders of the image). In particular, the convolution matrix of this operation is

$2\tilde{H}_{12}$	$-2(\tilde{H}_{22} + \tilde{H}_{12})$	0
$-2(\tilde{H}_{11} + \tilde{H}_{12})$	$4(\tilde{H}_{11} + \tilde{H}_{12} + \tilde{H}_{22})$	$-2(\tilde{H}_{11} + \tilde{H}_{12})$
0	$-2(\tilde{H}_{22} + \tilde{H}_{12})$	$2\tilde{H}_{12}$

and is constant across the image. Hence, the actual computation of the product $\boldsymbol{R}\boldsymbol{v}$ can be replaced in this case by an optimized image convolution algorithm, and the matrix $\boldsymbol{R}$ becomes useless as the convolution matrix is entirely defined by the entries of $\tilde{\boldsymbol{H}}$. Some minor adjustment will be needed for the entries corresponding to border nodes but can be done a posteriori.

Conclusion

We introduced in this chapter an implementation of the discretization of GeGFs presented in Section 7.3. The finite element method was presented and used to derive a finite-dimensional vector space of basis functions on which GeGFs defined on a compact Riemannian manifold can be approximated. In particular, these so-called finite element spaces consist of (deterministic) interpolation functions attached to the nodes of a triangulation of the manifold. Computing the weights using Theorem 7.3.5 then yields a numerical approximation of the GeGF, which can then be used for simulation, estimation and inference purposes (cf. Chapters 3 to 5). This will be illustrated in the next chapter.

A result of convergence of the finite element discretization of a GeGF was also introduced and proven. As one way expect it, the convergence rate of the discretization towards the GeGF depends mainly on the mesh size of the triangulation and on the regularity and speed of decrease of the spectral density (and its derivative) defining the GeGF.

9 Applications

Contents

Résumé

Dans ce chapitre, nous illustrons comment les champs généralisés définis sur des variétés riemanniennes que nous avons introduit dans ce travail nous permettent de répondre aux problématiques de modélisation que nous nous posons, à savoir travailler sur des domaines spatiaux complexes et avec des anisotropies locales.

Nous nous intéressons à trois tâches classiquement rencontrées en Géostatistique: la simulation de champs gaussiens, leur estimation à partir de l'observation incomplète et bruitée d'une réalisation et enfin l'inférence de leur propriétés de covariance dans le même cas. Nous montrons comment notre cadre permet d'aborder ces trois tâches avec une approche "sans matrice" qui ouvre le champ à des implémentations capables de travailler avec des grands jeux de données tout en minimisant coûts computationels et coûts de mémoire (grâce à l'algorithme de filtrage de Tchebychev). Nous présentons des exemples synthétiques ainsi que des exemples issus de données réelles et répondant à des problématiques posées par la société ESTIMAGES.

Introduction

In this chapter we illustrate how the framework of generalized random fields on Riemannian manifolds introduced in this dissertation allows to take on the two challenges that motivated our work, that is the extension of classical isotropic Gaussian random fields of $\mathbb{R}^d$ to complex (and bounded) domains and the definition of random fields with predefined local anisotropies.

On one hand, as long as the complex domain can be described as a d-manifold, our approach is applicable. Hence, using a triangulation of the domain, a finite element approximation of a generalized random field with spectral density γ^2 can be defined from the functions of a finite element space using Equation (7.36). On the other hand, when dealing with local anisotropies, we saw how they could be used to define a Riemannian metric, which in turn would ensure that the resulting random field respects them.

In this context three classical tasks in Geostatistics are performed: the simulation of Gaussian random fields, their estimation from an incomplete (and possibly noisy) observation and the inference of the covariance parameters from once again an incomplete observation of the field. Concrete case studies are presented, using both synthetic and real data. In particular, the case studies on real data come from the ongoing application of the modeling approach (and of the algorithms) presented in this work within the activities of the ESTIMAGES company.

9.1 Simulation

In this section, we leverage our construction of generalized random fields on manifolds to yield simulations of Gaussian random fields on various domains. More precisely, simulations of the finite element approximation of our generalized random fields, defined by Equation (7.36), are computed. Such simulations simply amount to the simulation of the Gaussian weights $\boldsymbol{Z} = (Z_1, \dots, Z_{n_h})$ through which they are defined. Given that the vector $\boldsymbol{U} = \boldsymbol{C}^{1/2}\boldsymbol{Z}$ can be seen as $\boldsymbol{S}$-stationary graph signal (on the triangulation graph), we use the simulation algorithms of Section 3.1 to generate a realization $\boldsymbol{u}$ of $\boldsymbol{U}$ and then obtain a realization of $\boldsymbol{Z}$ with the vector $\boldsymbol{z} = \boldsymbol{C}^{-1/2}\boldsymbol{u}$.

It should be noted that using the Chebyshev filtering algorithm to generate the weights turns the simulation process into a convolution process, with a possibly spatially varying kernel. Indeed, note first that, given that linear finite elements are used, each weight z_i actually corresponds to the value taken by the realization of the finite element approximation $\mathcal{Z}_{n_h}$ at the i-th triangulation node. Besides, through Chebyshev filtering, z_i can be expressed as

$$z_i = \frac{1}{\sqrt{C_{ii}}} u_i = \frac{1}{\sqrt{C_{ii}}} \left[\mathcal{S}_m[\gamma^2](\boldsymbol{S})\boldsymbol{w}\right]_i \quad ,$$

where $\boldsymbol{w}$ is a realization of a Gaussian white signal and $\mathcal{S}_m[\gamma^2]$ is the Chebyshev approximation polynomial of γ^2 at some order of approximation m. $\mathcal{S}_m[\gamma^2](\boldsymbol{S})$ being a polynomial graph filter, its non-zero entries $[\mathcal{S}_m[\gamma^2](\boldsymbol{S})]_{ij}$ on its i-th row correspond to the vertices j of the grid that are within a m-hop neighborhood around i. Hence z_i is given as a linear combination of the values of $\boldsymbol{w}$ at the vertices within a neighborhood of size m of i, and the weights are given by the entries of the i-th row of $\mathcal{S}_m[\gamma^2]$.

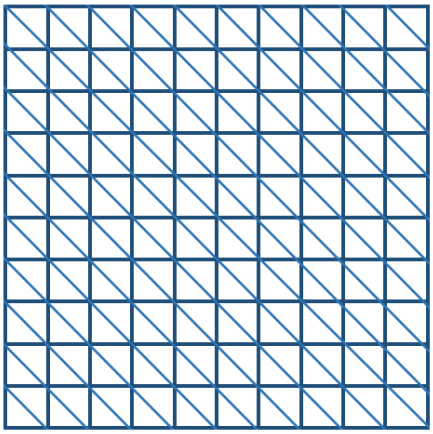


Figure 9.1: Illustration of a grid triangulation. Each cell of a regular grid is split along its diagonal to yield a triangulation of the domain initially covered by the grid.

In particular if we work with a triangulated grid (cf. Figure 9.1), the i-th entry of any n_h-vector can be seen as the value of the i-th pixel of an image having the same dimension as the grid. In that case, z_i can be seen as the result of applying at the i-th pixel of the image whose pixel values are determined by $\boldsymbol{w}$, the convolution kernel of size m with coefficients given by the entries the i-th row of $\mathcal{S}_m[\gamma^2]$. In the following we will represent some of these convolution kernels in the stationary case as then, the kernel matrix is identical for all nodes far enough from the borders of the image.

9.1.1 Simulation of stationary Matérn models

Presentation of the model

The Matérn model is a widely used covariance model in Geostatistics due to its great flexibility (Stein, 2012). For a lag distance $h \in \mathbb{R}_+$, its isotropic formulation is (Chilès and Delfiner, 2012):

$$C(h) = \frac{\sigma^2}{2^{\nu-1}\Gamma(\nu)} \left(\frac{h}{\phi}\right)^{\nu} K_\nu \left(\frac{h}{\phi}\right) \quad ,$$

where $\sigma^2 > 0$ is the marginal variance, $\phi > 0$ is a scaling parameter, $\nu > 0$ is a shape parameter and K_ν is the modified Bessel function of the second kind of order ν. The parameter ν can be seen as a "smoothness" parameter as the underlying process is $\lfloor \nu \rfloor$-time mean-square differentiable.

Following the results from Whittle (1954), Gaussian random fields defined over $\mathbb{R}^d$ and with a Matérn covariance can be seen as solutions of the following stochastic partial differential equation (SPDE):

$$(\kappa^2 - \Delta)^{\alpha/2} Z = \tau \mathcal{W} \quad , \tag{9.1}$$

with $\mathcal{W}$ a spatial Gaussian white noise and $\kappa > 0$, $\tau > 0$, $\alpha > d/2$ and the pseudo-differential operator $(\kappa^2 - \Delta)^{\alpha/2}$ is defined by

$$(\kappa^2 - \Delta)^{\alpha/2}[.] = \mathscr{F}^{-1}\left[w \mapsto (\kappa^2 + \|\omega\|^2)^{\alpha/2} \mathscr{F}[.](\omega)\right] \quad .$$

The parameters of the SPDE are linked to the parameters of the covariance function through

$$\kappa = 1/\phi, \quad \alpha = \nu + d/2, \quad \tau = \sigma \kappa^{\nu} \sqrt{(4\pi)^{d/2} \Gamma(\nu + d/2)/\Gamma(\nu)} \quad .$$

Consequently, their spectral density is given by (Lang and Potthoff, 2011):

$$f(\boldsymbol{w}) = \frac{\tau^2}{(\kappa^2 + \|\boldsymbol{w}\|^2)^{\alpha}} \quad . \tag{9.2}$$

We can therefore generate simulations of such fields using the finite element approximation given in Theorem 7.3.5. All that is required is to generate a set of coefficients from a multivariate

normal distribution with covariance matrix in Equation (7.37). Note that Lindgren et al. (2011) defined a Markovian approximation of the same field using also the finite element method to solve the SPDE (9.1) in the case where α is an integer. Their formula for the covariance matrix of the weights actually coincides with ours.

We generate simulations of Gaussian random fields with a Matérn covariance function (which we simply call *Matérn fields* in the following) using two methods for comparison purposes. On one hand the Cholesky factorisation algorithm applied to the expression of the covariance matrix in Equation (7.37) is used to generate the approximation weights, and on the other hand the Chebyshev filtering algorithm is used. Only Matérn fields on a two-dimensional grid are generated in this section, and with integer smoothing parameters ν, in order to facilitate the comparison with the Cholesky method. Indeed, in this case, the precision matrix of the weights will be sparse, rendering the Cholesky factorization tractable.

Order of the polynomial approximation

First, the effect of the order of the polynomial approximation on the resulting simulation is investigated. To do so, simulations of a Matérn field are generated on a 200x200 grid, with range 25, sill 1 and smoothness parameter 1, and with a growing order. In Figure 9.3, simulations obtained for degree values of 1, 5, 20 and 100 and the associated variogram (averaged over 50 simulations) are displayed. As a comparison, the same model simulated using the classical Cholesky factorisation algorithm is displayed in Figure 9.2.

As noticed in Figure 9.3, increasing the order of the polynomial tends to add smoothness and structure to the simulation. This is expected from a convolution algorithm as the size of the kernel, which is directly linked to the order of the polynomial, grows (center images in Figure 9.3). Moreover, there seems to be a point from which adding more polynomials does not change the simulation, meaning that the Chebyshev polynomial approximation basically converged.

Note also that the variogram of the simulations in Figure 9.3 tends to be respected as the order of the polynomial grows. This fact was predictable and is due to the fact that the proposed algorithm ensures that any linear combinations of the vectors generated by the algorithm have the right variance within a given tolerance. Consequently, this will ensure that the variogram is respected given that its value at particular lag h is just the variance of the difference between two particular entries of the simulated vector that correspond to nodes of the triangulation separated by an Euclidean distance of h.

Influence of the model

The influence of the covariance model parameters on the resulting approximations is now investigated. To do so, simulations of Matérn fields with different values of range and smoothness parameters are generated (cf. Figure 9.4). For each set of parameters, the order of approximation is set so that the probability of rejection on the statistical tests with significance $a = 0.05$ is

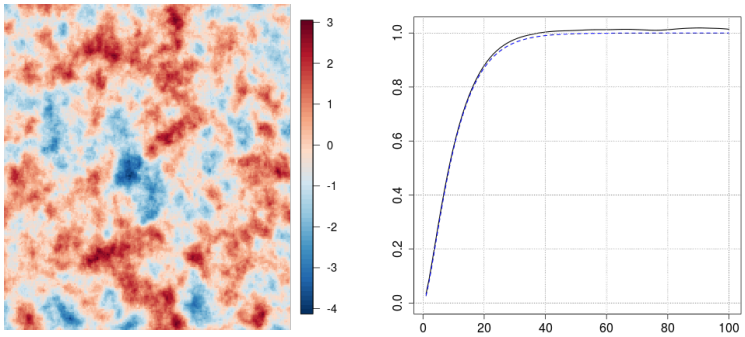
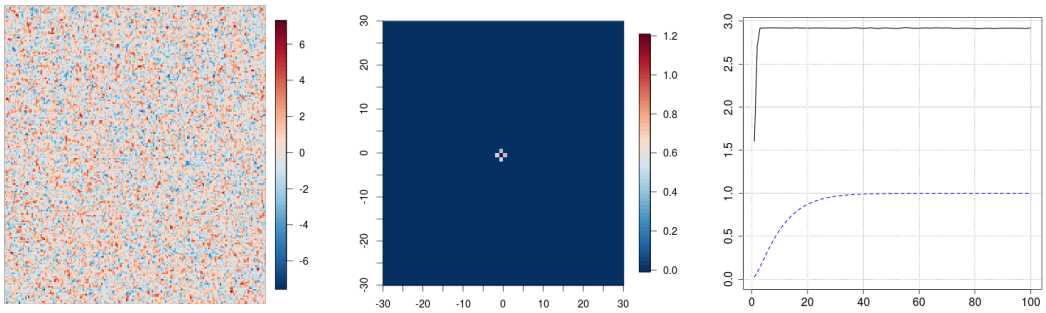
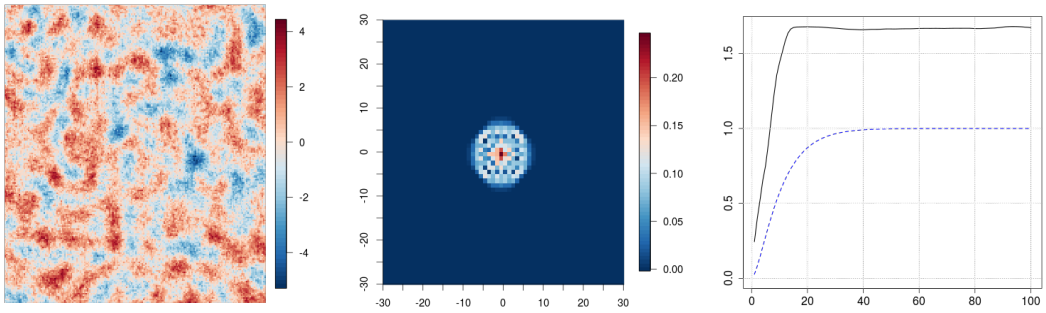


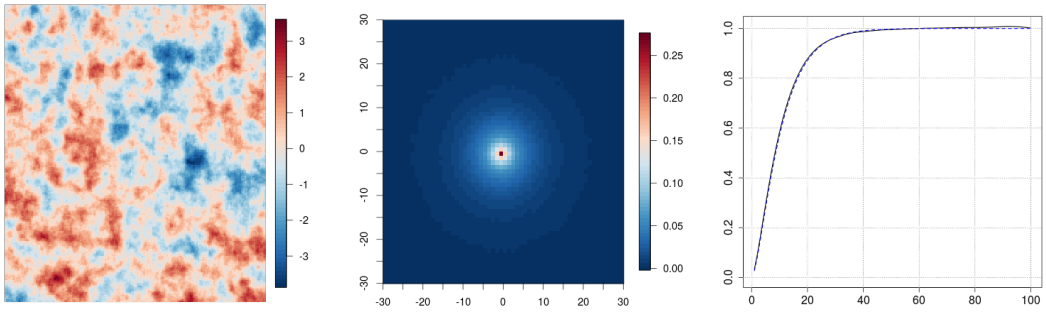
Figure 9.2: *(Left)* Simulations of a Matérn model with range 25, sill 1 and smoothness parameter 1 on a 200x200 grid using Cholesky factorisation. *(Right)* Mean variograms over 50 simulations (solid line) and model (dotted line).



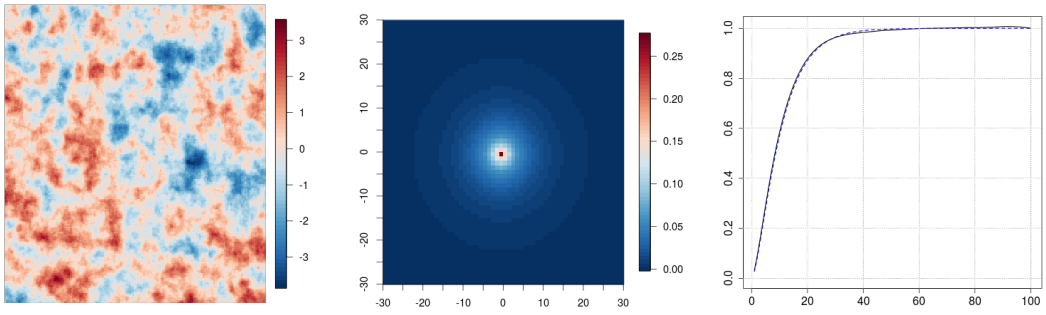
(a) Order of approximation =1.



(b) Order of approximation = 10.

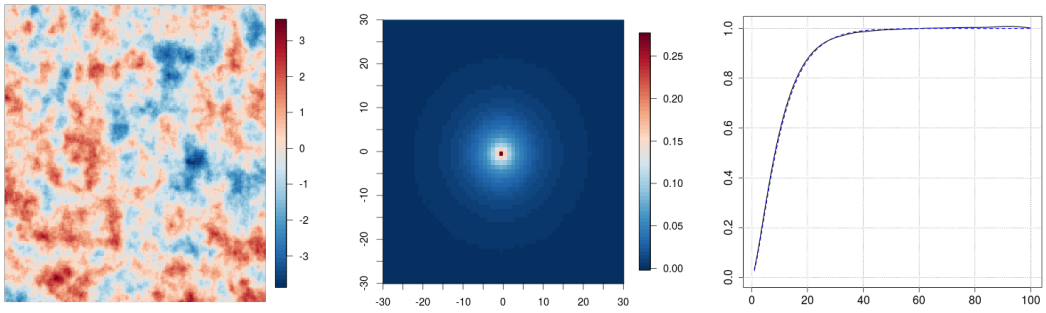


(c) Order of approximation = 50.

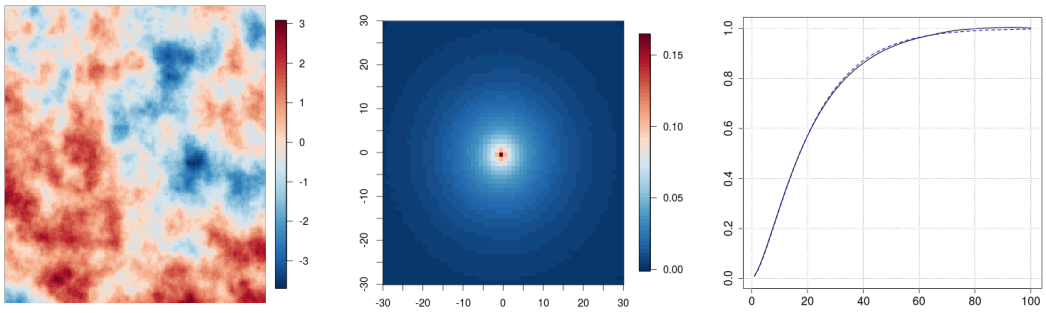


(d) Order of approximation = 100.

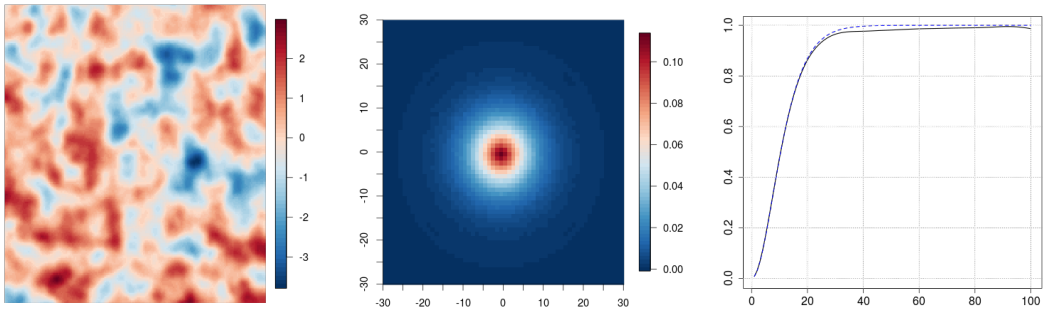
Figure 9.3: *(Left)* Simulations of a Matérn model with range 25, sill 1 and smoothness parameter 1 on a 200x200 grid using Chebyshev approximation with growing order. *(Center)* Convolution kernels associated with the simulation. *(Right)* Mean variograms over 50 simulations (solid line) and model (dotted line).



(a) Range = 25, Smoothness = 1. Order of approximation = 76.



(b) Range = 50, Smoothness = 1. Order of approximation = 166.



(c) Range = 25, Smoothness = 3. Order of approximation = 84.

Figure 9.4: *(Left)* Simulations using Chebyshev approximation of a Matérn field on a 200x200 grid with various model parameters. *(Center)* Convolution kernels associated with the simulation. *(Right)* Mean variograms over 50 simulations (solid line) and model (dotted line).

equal to $(1+10\%)a$. Following the results of Section 3.2.2, this choice corresponds to a threshold on the approximation error of 3.0e-02 (cf. Table 3.1).

The order of approximation used for each simulation is reported in Figure 9.4. It can be noticed that increasing the range results in significantly higher orders of approximation to achieve the same accuracy, whereas the effect of the smoothness parameters seems more limited. This is just another consequence of the "convolution" nature of the algorithm, as explained at the beginning of the section. The larger the range is, the larger the size of the kernel used to generate the simulation from a white noise image should be as larger "spots" must be created, and therefore the larger the order of approximation is. On the other hand, the smoothness parameter mainly affects the smoothness of the kernel, not its size.

9.1.2 Simulation of general covariance models

In the previous subsection, only Matérn fields with integer smoothing parameters were considered. However, our simulation approach does not change at all if non-integer smoothing parameters are considered. All we need is the expression of the (radial) spectral density of the

field we wish to simulate. In particular, this expression is given by Equation (9.2) for the Matérn covariance. Simulations of Matérn fields on a 2D grid with non-integer smoothing parameters are displayed in Figure 9.5. Note in particular that, contrary to the integer case, such fields are not Markovian.

As an illustration of the flexibility of the method, Figure 9.6 displays simulations of Matérn fields for various spectral densities.

Note in particular the simulation of the spherical model, which is a covariance model with compact support. Its covariance function (in the case where scale parameter and variance parameter are both set to 1) is defined by

$$C(\boldsymbol{h}) = \mathbb{1}_{\mathscr{B}} * \mathbb{1}_{\mathscr{B}}(\boldsymbol{h}) \quad ,$$

where $\mathbb{1}_{\mathscr{B}}$ denotes the indicator function of the ball of radius $1/2$ of $\mathbb{R}^2$, centered at $\mathbf{0}$. Its spectral density is given by

$$f(\boldsymbol{\xi}) = \frac{1}{\pi\|\boldsymbol{\xi}\|^2} J_1\left(\frac{\|\boldsymbol{\xi}\|}{2}\right)^2 \quad , \tag{9.3}$$

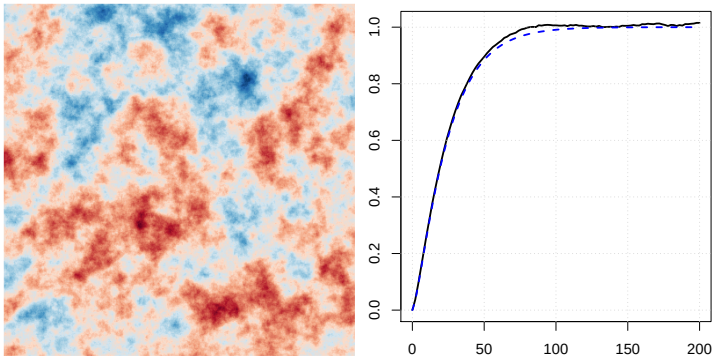
where J_1 denotes here the J-Bessel function with parameter 1 (Lantuéjoul, 2013).

Besides, we also simulated oscillating Matérn models, which were introduced in (Lindgren et al., 2011, Section 3.3) as solution of an oscillatory SPDE. Their spectral density is given by

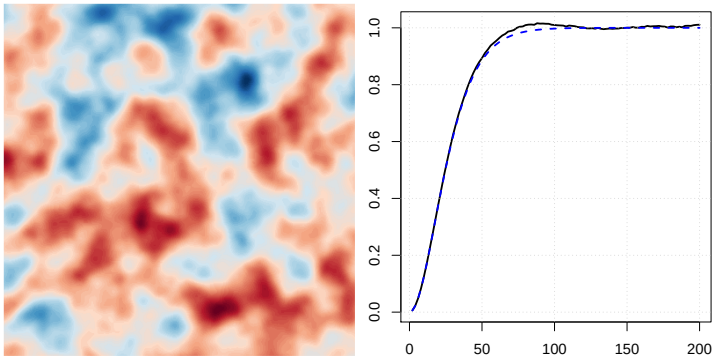
$$f(\boldsymbol{\xi}) = \frac{1}{(2\pi)^2}\left(\kappa^4 + 2\cos(\pi\theta)\kappa^2\|\boldsymbol{\xi}\|^2 + \|\boldsymbol{\xi}\|^4\right)^{\alpha/2} \quad , \tag{9.4}$$

where $\kappa > 0$ is a scale parameter, $\theta \in [0, 1[$ is a parameter linked to the oscillation frequency and $\alpha > 1$ is parameter playing a role similar to the shape parameters of ordinary Matérn fields. In particular for $\alpha = 2$, the covariance function (in $\mathbb{R}^2$) of this model is given by

$$C(\boldsymbol{h}) = \frac{1}{4\mathrm{i}\pi\sin(\pi\theta)\kappa^2}\left(K_0\left(\kappa\|\boldsymbol{h}\|e^{-\mathrm{i}\frac{\pi\theta}{2}}\right) - K_0\left(\kappa\|\boldsymbol{h}\|e^{\mathrm{i}\frac{\pi\theta}{2}}\right)\right) \quad .$$

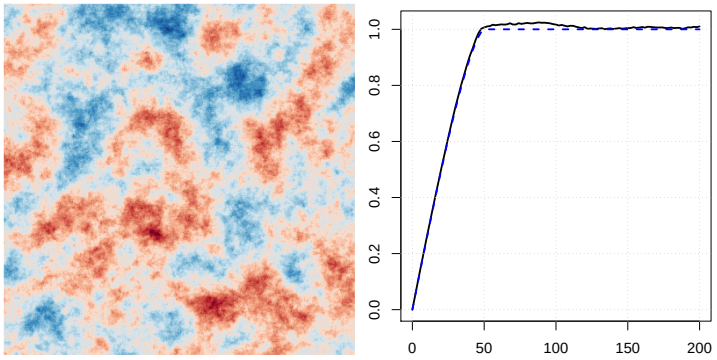


(a) Matérn with smoothness parameter $\nu = \pi/4$.

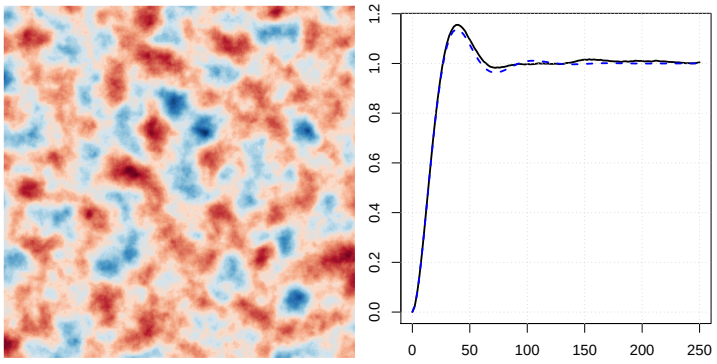


(b) Matérn with smoothness parameter $\nu = e^1$.

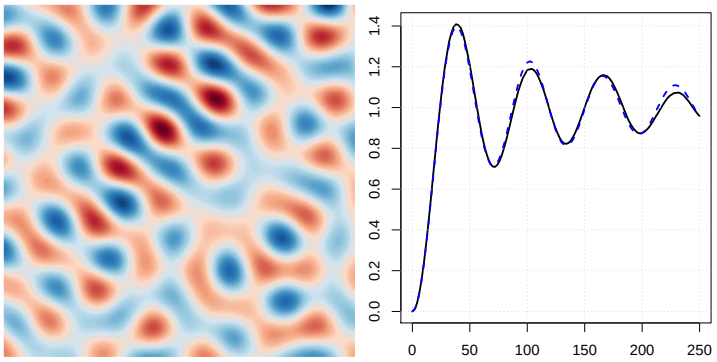
Figure 9.5: Simulations on a 400x400 grid of Matérn fields with real smoothness parameters using the spectral density expression in Equation (9.2) and associated mean variograms over 50 simulations (solid line) and model (dotted line).



(a) Spectral density : Equation (9.3).



(b) Spectral density : Equation (9.4) with $\kappa = 0.1, \theta = 0.8, \alpha = 1$.



(c) Spectral density : Equation (9.4) with $\kappa = 0.1, \theta = 0.95, \alpha = \pi$.

Figure 9.6: Simulations on a 400x400 grid of random fields with integrable spectral densities expression and associated mean variograms over 50 simulations (solid line) and model when available (dotted line).

We generated simulations for $\alpha = 1.5$ and $\alpha = \pi$, for which we could not find an expression of the associated covariance function. However, in order to check the validity of the simulation obtained by our algorithm, we computed it numerically using the Fourier transform of the spectral density.

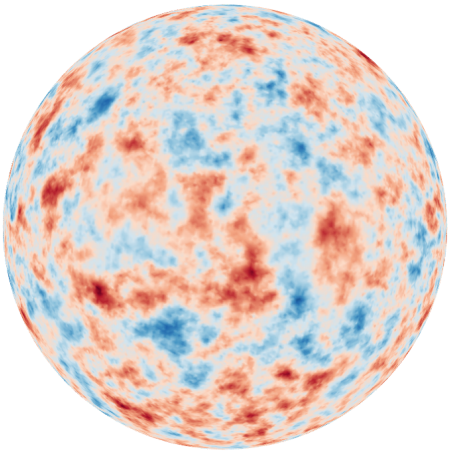
9.1.3 Simulation on manifold

Figure 9.7 displays examples of simulations of Matérn fields on (smooth) surfaces of $\mathbb{R}^2$. The simulation process doesn't change from the previous examples. Only the triangulation step, which is now carried out on the manifold, changes. Starting from a triangulation of each one of these surfaces, the mass and stiffness matrices were built (using the Euclidean metric while integrating on each triangle) and the weights were simulated using Chebyshev filtering. The covariance model is the same for all these simulations, and consists of a Matérn model with shape parameter $\nu = 3$ and scale parameter $\phi = 0.12R$ where R is the size of the smallest ball containing the surface.

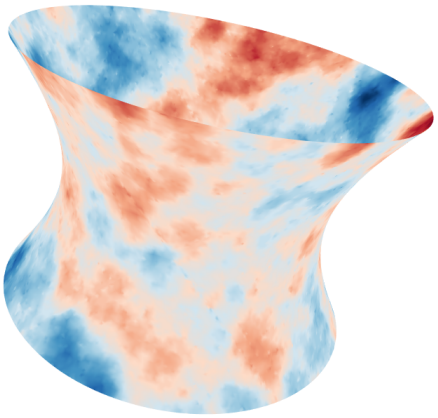
The meshes of the sphere and the hyperboloid were generated using the mesh processing software MeshLab (Cignoni et al., 2008). As for the mesh of the duck and the cow surfaces, they

are part of the Keenan Crane's 3D model repository (Crane, 2019).

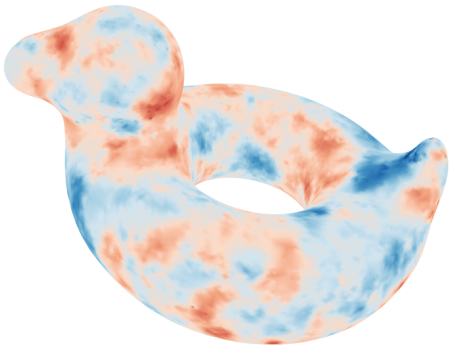
Our simulation approach generalizes easily to three-dimensional domains: only the formulas used to compute the mass and the stiffness matrices change. This is illustrated in Figure 9.8, where a Matérn field with shape parameter $\nu = 1$ was simulated on a solid torus.



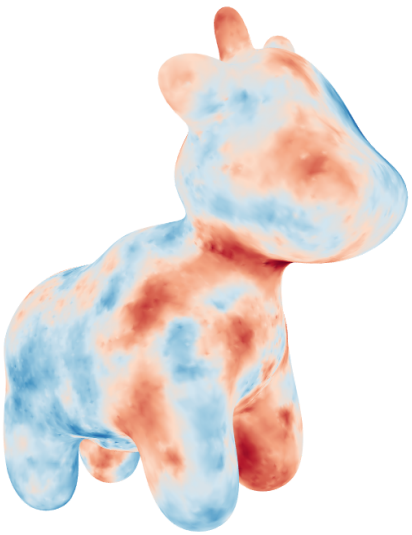
(a) Sphere.



(b) Hyperboloid.



(c) Toroidal duck.



(d) Cow.

Figure 9.7: Simulations of Matérn fields on smooth two-dimensional surfaces.

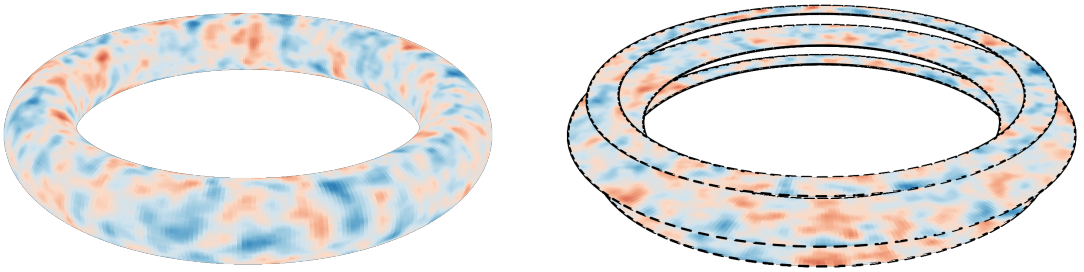
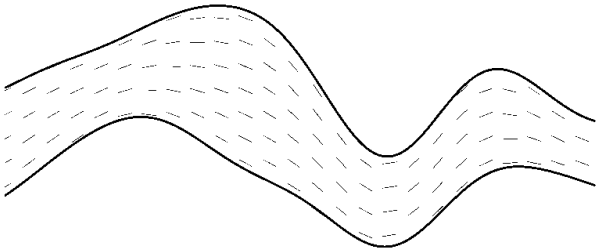


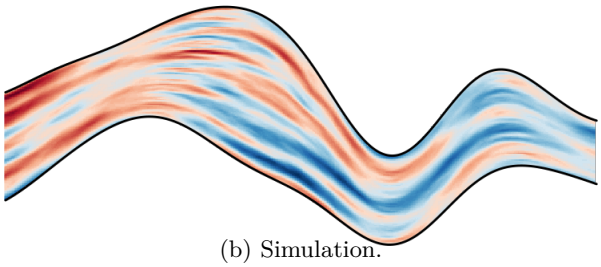
Figure 9.8: Simulation of a Matérn field on a solid torus *(Left)* and slices of the same torus *(Right)*.

9.1.4 Simulation of fields with local anisotropy

Following the approach described in the introduction of this section, a metric was defined from fields of anisotropy defined on each domain. Then the simulation process was carried out in the same manner as in the other cases. Hence, accounting for local anisotropies only impacts the construction of the mass matrix and the stiffness matrix.

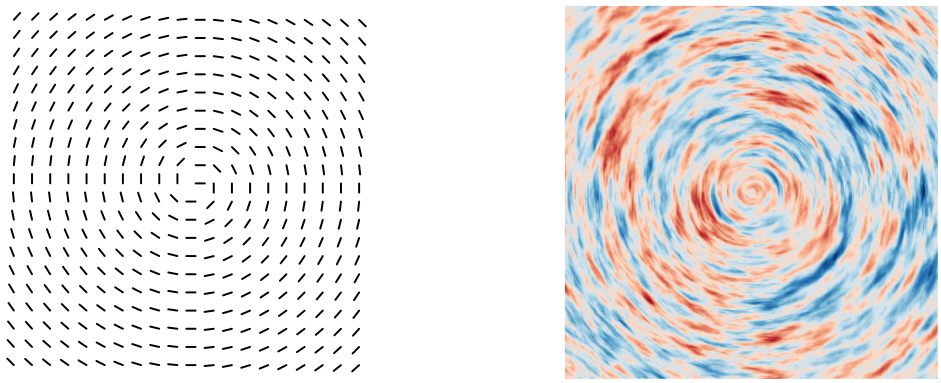


(a) Anisotropy field.

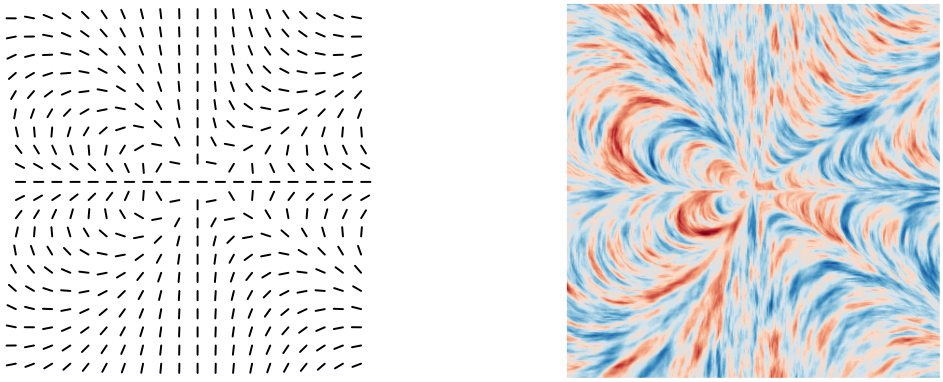


(b) Simulation.

Figure 9.9: Simulation of a non-stationary Matérn field with global range 150 and sill 1, and local anisotropies, carried out on a "geological layer" with overall extension 500x200.



(a) Circular anisotropies.



(b) Arbitrary field of anisotropy.

Figure 9.10: Simulations on a 400x400 grid of Matérn fields with local anisotropies. *(Left)* Map giving the principal direction of the anisotropy *(Right)* Resulting simulation.

Figure 9.9 displays a simulation carried out on a domain resembling a geological layer. The direction of the anisotropies follows the curvature of the layer and the anisotropy ratio is locally proportional to the thickness of the layer (with a maximum value of 1.5).

Figure 9.10 displays Matérn fields with shape parameter $\nu = 1$ and range 50, generated on a grid in which local anisotropies were specified. The directions of the major axes of anisotropy are represented as straight lines in each case, and the anisotropy ratio, i.e. the ratio between the lengths of the two main directions of anisotropy, are kept constant in these examples (and equal to 1/5).

9.2 Prediction of non-stationary fields

In this section, we apply our framework for modeling random fields with local anisotropies to mapping and filtering tasks using kriging estimates. The main limitations in usual applications come from the construction of covariance matrices relating to covariance models that would take into account the local anisotropies. Besides, such matrices are doomed to be full in general, rendering this approach hardly scalable.

Our approach answers both problems. On one hand, we showed that building the model from the Laplace-Beltrami operator defined from the local anisotropies ensures that the covariance is locally anisotropic. As the model is defined globally across the domain, through the eigenfunctions of the Laplace-Beltrami operator, the global coherence of the model is ensured. On the other hand, the Chebyshev trick allows to render calculations scalable.

9.2.1 Kriging estimate of non-stationary fields

First, we look into the mapping problem. Namely, we assume that we observe the value of a random field Z on a domain $\mathcal{M}$ at n_o locations and aim at building an estimator of Z over $\mathcal{M}$ from these observations. We also assume that Z displays local anisotropies that are known at any point of $\mathcal{M}$, and that the parameters defining the covariance model of Z are also known. Finally we assume that the observations are tainted by a measurement noise, modeled by independent zero-mean Gaussian variables with standard deviation τ.

The estimator we choose to answer this problem is the kriging estimate, as it is the best unbiased linear estimator we can build using the observation data. Denoting $\boldsymbol{p}_1, \dots, \boldsymbol{p}_n$ the observation points, and assuming a Gaussian model, the kriging estimator at a point $\boldsymbol{p}$ is given by (Wackernagel, 2013):

$$Z^*(\boldsymbol{p}) = \mathbb{E}[Z(\boldsymbol{p}) | Z(\boldsymbol{p}_1), \dots, Z(\boldsymbol{p}_{n_o})], \quad \boldsymbol{p} \in \mathcal{M} \quad .$$

If we now assume that Z is the finite element approximation of a generalized random field, we can write

$$Z(\boldsymbol{p}) = \sum_{k=1}^{n_h} Z_k \psi(\boldsymbol{p}), \quad \boldsymbol{p} \in \mathcal{M} \tag{9.5}$$

where $\{\psi_k\}_{1 \leq k \leq n_h}$ is the set of n_h interpolating functions defined from the triangulation of $\mathcal{M}$ (cf. Section 8.1), $Z_1, \dots, Z_{n_h}$ is a collection of correlated Gaussian weights whose covariance matrix is given by Theorem 7.3.5.

We can then rewrite the kriging estimate as

$$Z^*(\boldsymbol{p}) = \mathbb{E}[\boldsymbol{\psi}_{\boldsymbol{p}}^T \boldsymbol{Z} | \boldsymbol{M}_o \boldsymbol{Z} + \tau \boldsymbol{W}_o] = \boldsymbol{\psi}_{\boldsymbol{p}}^T \mathbb{E}[\boldsymbol{Z} | \boldsymbol{M}_o \boldsymbol{Z} + \tau \boldsymbol{W}_o], \quad \boldsymbol{p} \in \mathcal{M} \quad , \tag{9.6}$$

where $\boldsymbol{W}_o$ is a vector of n_o independent standard Gaussian variables, $\boldsymbol{Z} = (Z_1, \dots, Z_{n_h})^T$, $\forall \boldsymbol{p} \in \mathcal{M}$, $\boldsymbol{\psi}_{\boldsymbol{p}} := (\psi_1(\boldsymbol{p}), \dots, \psi_{n_h}(\boldsymbol{p}))^T$ and in particular

$$\boldsymbol{M}_o = \begin{pmatrix} \boldsymbol{\psi}_{\boldsymbol{p}_1}^T \\ \hline \vdots \\ \hline \boldsymbol{\psi}_{\boldsymbol{p}_{n_o}}^T \end{pmatrix} \quad . \tag{9.7}$$

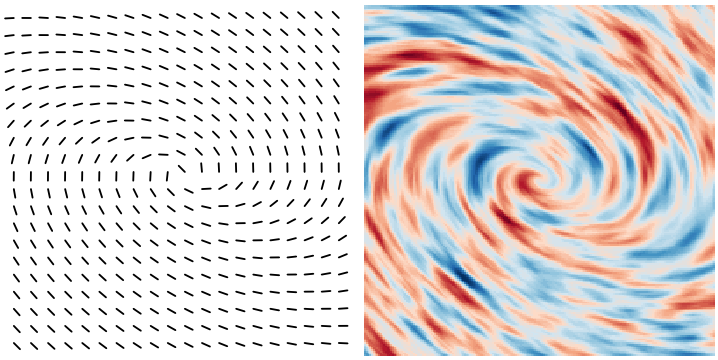
Hence, the kriging estimate of Z at any point $\boldsymbol{p}$ is obtained as a weighted sum of the entries of the conditional expectation of the vector $\boldsymbol{Z}$.

The kriging estimate in Equation (9.6) is computed as follows. The weights $\boldsymbol{\psi_p}$ are entirely determined by the location of $\boldsymbol{p} \in \mathcal{M}$ and the basis functions of the finite element space. Regarding then the computation of the conditional expectation, recall that, following Theorem 7.3.5, $\boldsymbol{Z}$ can be seen as a (linear transform of a) stochastic graph signal with covariance matrix given by Equation (7.37). In particular, $\boldsymbol{Z}$ is a stationary stochastic graph signal having the same spectral density as the Gaussian field Z. Hence, the conditional expectation in Equation (9.6) can be computed using the algorithms presented in Section 4.1, which concludes the computation of the kriging estimate.

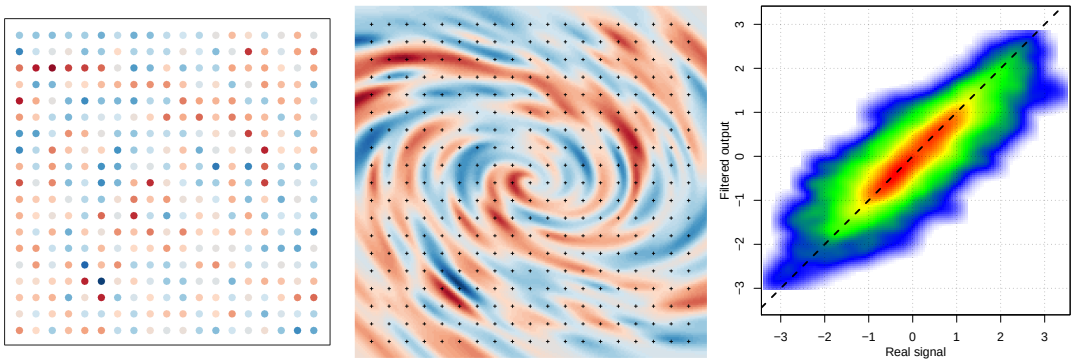
This approach to kriging is now tested in two situations, using synthetic and real data.

Synthetic data

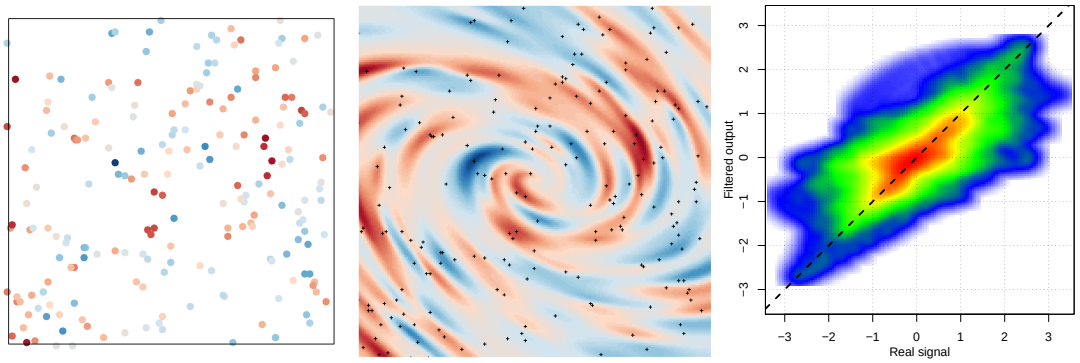
We first apply this method to synthetic data. We generated a simulation of a Matérn field on a grid, following a specific anisotropy pattern (cf. Figure 9.11a). Then we sampled the field using two strategies: on one hand the samples were chosen equidistant on the grid at a rather



(a) Simulation a non-stationary field, and associated local anisotropies.



(b) Kriging estimate with regular sampling.



(c) Kriging estimate with uniform sampling.

Figure 9.11: Kriging estimate from simulated data. For (b) and (c): the sampled points are represented in the left figure, the kriging estimate in the middle figure and the right figure represents a density plot of the correlation between the estimate and the original (simulated) field. High densities are in red.

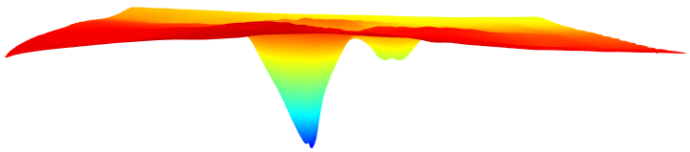
dense frequency, and on the other hand, 200 uniform points were drawn from the grid. The two kriging estimates are presented in Figures 9.11b and 9.11c.

In both cases we can observe the smoothing effect of the kriging estimate, which is due to the fact that it is basically a linear estimator (Wackernagel, 2013). Also, even with a sparse sampling, the estimate has no trouble recreating the local anisotropies of the field. Hence, the kriging procedure presented here seems particularly appropriate to efficiently take into account anisotropy-related information in mapping problems.

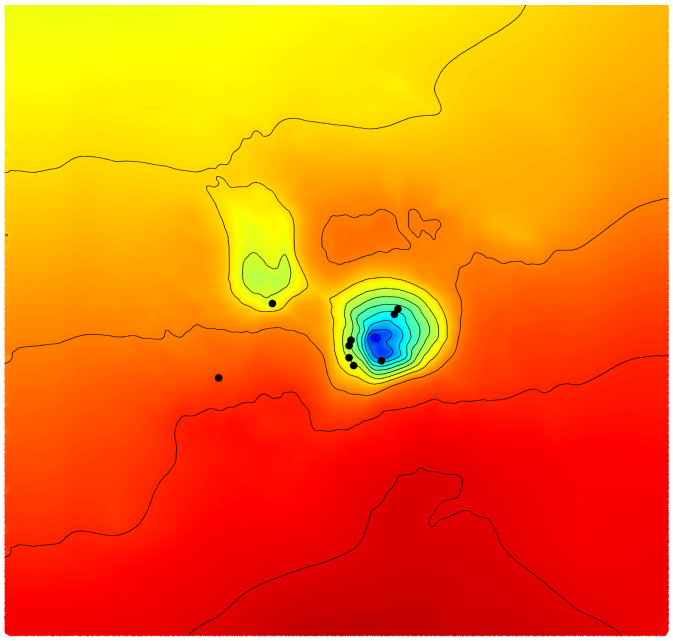
Real data

The anisotropic mapping method presented in this work was put to test on real geological data. When trying to locate an oil reservoir in the subsurface of a field, a go-to method consists in using seismic imaging. Through this process, an echography of the subsurface is obtained by "shooting" acoustic waves into the subsurface, and then studying their reflection on the different geological interfaces composing the subsurface. In particular, complex processing methods are performed to turn the detected times of arrival of these waves after reflection on a given interface into the actual depth of this surface.

In this case study, our starting point is a map $\{Z_s(\boldsymbol{p}) : \boldsymbol{p} \in \mathcal{D}\}$ of the depth of an interface across a field $\mathcal{D}$, which was obtained by processing seismic data (cf. Figure 9.12 for an example). In general, small disparities are observed between the depth obtained after processing the seismic data and the actual depth of an interface, which is observed while digging wells. Hence, a step of calibration of the "seismic" depths is performed so that both depths agree at the points where well data are available.



(a) Representation of the geological interface obtained from the seismic data, as a 3D surface.



(b) Depth map obtained from seismic data. The continuous lines represent level sets, and the black dots represent well locations.

Figure 9.12: Representations of the seismic data from the ODA field.

Denote by $\{Z_w(\boldsymbol{x}_i) : i \in [\![1, N_w]\!]\}$ the depths observed on the wells dug at the locations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N_w} \in \mathcal{D}$. The calibration step is carried out by kriging over the field $\mathcal{D}$ the residuals $R(\boldsymbol{x}_i) = Z_w(\boldsymbol{x}_i) - Z_s(\boldsymbol{x}_i), i \in [\![1, N_w]\!]$ computed at the well locations. Indeed, the interpolating nature of the resulting kriging estimate $\{R^*(\boldsymbol{p}) : \boldsymbol{p} \in \mathcal{D}\}$ will ensure that the corrected depth Z_c defined by

$$Z_c(\boldsymbol{p}) = Z_s(\boldsymbol{p}) + R^*(\boldsymbol{p}), \quad \boldsymbol{p} \in \mathcal{D},$$

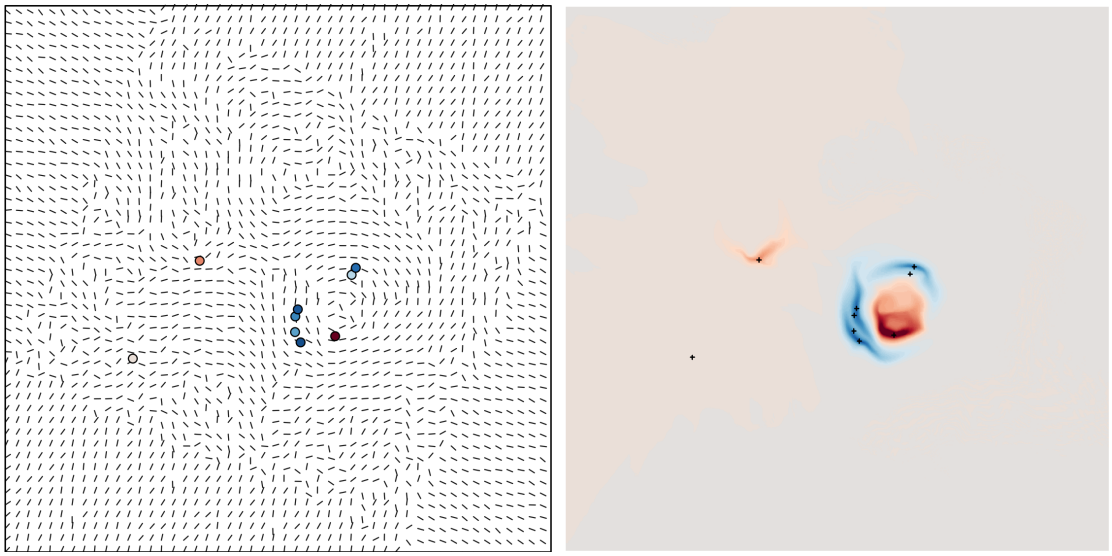
coincides with the well depths at the well locations.

Outside the grid locations, the kriging estimate R^* corrects the seismic depths by spatializing the residuals. It was observed in practice that these residuals tend to vary smoothly along the level sets of the seismic depths: the same is therefore expected for their spatialization. Hence, performing the kriging step using an isotropic covariance model would not yield satisfying results.

Instead, a locally anisotropic mapping is performed using the method presented in this section. First, a map of local anisotropies over $\mathcal{D}$ is defined from the level sets of the seismic data Z_s: basically, these local anisotropies correspond at each $\boldsymbol{p} \in \mathcal{D}$ to the angles defining the tangent to the level set of Z_s passing through $\boldsymbol{p}$. These angles can be determined using image processing algorithms based on gradient computations (Rahmat and Harris-Birtill, 2018).

Then the kriging step is performed by considering that the residuals are samples from a Gaussian random field displaying these local anisotropies. As seen earlier, the spatialization of the residual resulting from the kriging estimate will then recreate the shape of the level sets of the seismic data, thus answering our concern.

As part of a study conducted for SPIRIT ENERGY, this workflow was applied on seismic and well data from the the ODA field (cf. Figure 9.13), which is located in the Norwegian North Sea (licence PL405). The results are presented in Figure 9.13. The proposed correction was later validated by a new campaign of well digging.



(a) Local anisotropies on the field and location of the observation points.

(b) Kriging estimate.

Figure 9.13: Kriging estimate of residual points between well and seismic data from the ODA field.

9.2.2 Filtering of non-stationary fields

Geostatistical filtering relies on the assumption that a complex regionalized phenomenon can be seen as a superposition of independent simpler phenomena, each characterized by its own range of influence and its own spatial structure (Hoeber et al., 2003; Piazza et al., 2015). Formally, this means that an observed complex signal Z can be decomposed as a sum of independent (Gaussian) random fields, each characterized by its own variogram and anisotropy models. Filtering then consists in extracting one of these components, denoted Z_s, from an observation of Z that can

be expressed as

$$Z = Z_s + Z_n^{(1)} + \cdots + Z_n^{(m)} \quad , \tag{9.8}$$

where $Z_n^{(1)}, \ldots, Z_n^{(m)}$ denote the m independent noise fields that compose Z together with the "true" (unspoiled) signal Z_s.

The Factorial Kriging (FKr) method, proposed by Matheron (1982), solves this problem by estimating the value of the true signal using the observed noisy signal in a kriging approach (Wackernagel, 2013). Formally, if $Z(\boldsymbol{p}_1), \ldots, Z(\boldsymbol{p}_{n_o})$ denote the observations of the noisy signal at n_o locations $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_{n_o}$, then estimates $Z_s^*(\boldsymbol{p}_1), \ldots, Z_s^*(\boldsymbol{p}_{n_o})$ of the true signal at a location $\boldsymbol{p}$ are given by

$$Z_s^*(\boldsymbol{p}) = \boldsymbol{\sigma}_s(\boldsymbol{p})^T \left(\boldsymbol{\Sigma}_s + \boldsymbol{\Sigma}_n^{(1)} + \cdots + \boldsymbol{\Sigma}_n^{(p)} \right)^{-1} \begin{pmatrix} Z(\boldsymbol{p}_1) \\ \vdots \\ Z(\boldsymbol{p}_{n_o}) \end{pmatrix} \quad , \tag{9.9}$$

where $\boldsymbol{\sigma}_s(\boldsymbol{p}) \in \mathbb{R}^{n_o}$ denotes the vector containing covariances of the true signal Z_s between $\boldsymbol{p}$ and each data location and $\boldsymbol{\Sigma}_n^{(1)}, \ldots, \boldsymbol{\Sigma}_n^{(m)}$ denote the covariance matrices of the noise fields $Z_n^{(1)}, \ldots, Z_n^{(m)}$.

We propose to use the characterization of random fields on manifolds presented in our work to perform geostatistical filtering, while relying on graph filtering algorithms. Indeed, the same approach as the one outlined in Section 9.2.1 can be followed. The true signal and the noises are both written as finite element approximations (cf. Equation (9.5)). We assume that the observed values of the noisy signal are affected by a small measurement error modeled by independent (zero-mean) Gaussian variables with variance τ^2.

Then, using the same notions as for Equation (9.5), Equation (9.9) can be rewritten as

$$Z_s^*(\boldsymbol{p}) = \boldsymbol{\psi}_{\boldsymbol{p}}^T \mathbb{E}[\boldsymbol{Z} | \boldsymbol{M}_s \boldsymbol{Z}_s + \boldsymbol{M}_1 \boldsymbol{Z}_n^{(1)} + \cdots + \boldsymbol{M}_m \boldsymbol{Z}_n^{(m)} + \tau \boldsymbol{W}_o], \quad \boldsymbol{p} \in \mathcal{M} \quad , \tag{9.10}$$

where the vector $\boldsymbol{Z}_s$ (resp. $\boldsymbol{Z}_n^{(1)}, \ldots, \boldsymbol{Z}_n^{(m)}$) contains the weight of the finite element representation of Z_s (resp. $Z_n^{(1)}, \ldots, Z_n^{(m)}$) and the matrix $\boldsymbol{M}_s$ (resp. $\boldsymbol{M}_1, \ldots, \boldsymbol{M}_m$) is defined as in Equation (9.7) using the basis functions associated with Z_s (resp. $Z_n^{(1)}, \ldots, Z_n^{(m)}$). Finally, $\boldsymbol{W}_o$ is once again a vector with n_o independent standard Gaussian entries.

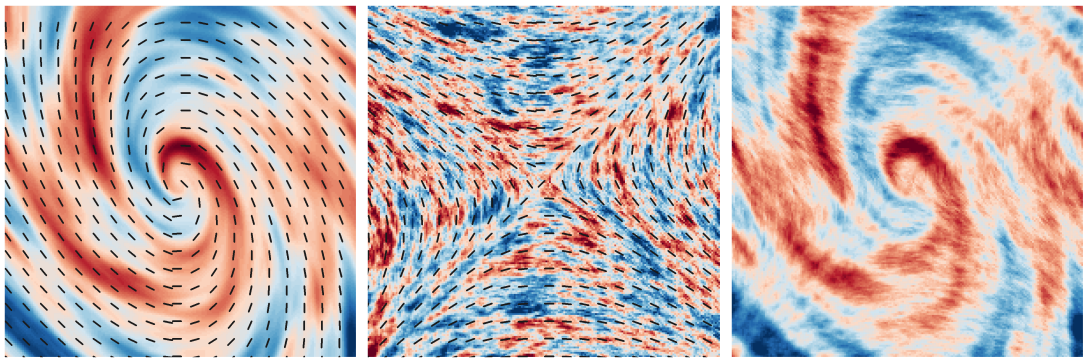
Hence, as it was the case for the kriging estimate in Section 9.2.1, the factorial kriging estimate can be estimated using the fact that $\boldsymbol{Z}_s, \boldsymbol{Z}_n^{(1)}, \ldots, \boldsymbol{Z}_n^{(m)}$ is interpreted as independent stochastic graph signals. This time, the results of Section 4.2 are used to compute the conditional expectation in Equation (9.10). Then the factorial kriging estimate at any point $\boldsymbol{p}$ of the domain is given as a linear combination of the entries of this conditional expectation vector, weighted by the vector $\boldsymbol{\psi}_{\boldsymbol{p}}$. This approach is now applied to two case studies using synthetic and real data.

Synthetic data

In this first case study, we simulated two Matérn fields on a 400x400 grid, some of which presenting local anisotropies, and added them to define our input (noisy) signal (cf. Figure 9.14). Precisely, the noisy signal to be filtered is composed of:

- A non-stationary field defined by a Matérn covariance function with smoothness parameter 3, ranges 100 and 20 along its principal directions, and sill 1. It has local anisotropies that describe a vortex-like shape. This field is the true signal we want to extract.
- A non-stationary field defined by an exponential covariance function with ranges 25 and 8 along its principal directions, and sill 0.4. It has local anisotropies that describe a "X" shape.

The filtering process was launched on the noisy image. The covariance parameters and the angles defining the anisotropies of the true signal and of the noise were used to compute the factorial kriging estimate of Equation (9.10) at each point of the grid. The output obtained from the filtering process is presented in Figure 9.15. As we see, the filtering process successfully extracted the true signal from the noisy observation. However we seem to obtain a smoothed version of the input: this is a consequence of the fact that the true signal is estimated through a kriging approach, which tends to yield smoothed outputs (Wackernagel, 2013).

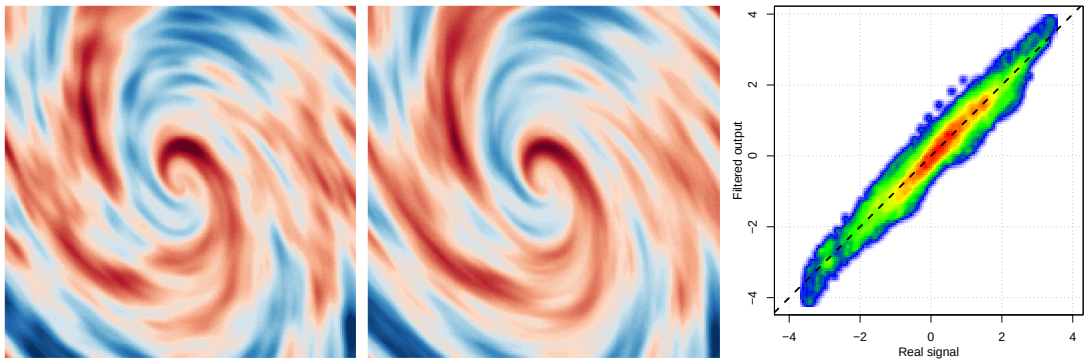


(a) Simulated "signal" and associated local anisotropies.

(b) Simulated "noise" and associated local anisotropies.

(c) Noisy signal.

Figure 9.14: Simulated data for the filtering test. The noisy signal (c) is the sum of the simulated "signal" (a) and the simulated "noise" (b).



(a) Filtered output.

(b) Original signal.

(c) Density scatter plot.

Figure 9.15: Results of the filtering algorithm applied to the simulated data.

Real dataset

Once again, the methods developed in this work were tested on real geological data. This time, the idea is to filter out noises from the seismic data. Such noises are a consequence of the methods applied to acquire the seismic data. If not removed, they make it difficult to identify zones of interest in the subsurface from the seismic data. These noises tend to be spatially correlated inside the "seismic" cube, which is the three-dimensional volume formed by the domain of acquisition $\mathcal{D}$ and the seismic depth. That is why geostatistical filtering is a good choice of algorithm, as it allows to specify spatially correlated models for the data.

In particular, using the approach presented in this section to perform geostatistical filtering allows to easily specify the local directions of the noise as local anisotropies. They will then naturally be taken into account in the filtering process. In particular, these directions can be identified on the data using image processing techniques.

The case study corresponds to the application of the geostatistical filtering on a vintage 2D seismic line located in the Amadeus basin (onshore Australia), and provided by the Australian company CENTRAL PETROLEUM. The data was originally acquired in 1966, reprocessed in 1984 and vectorized from a hardcopy in 2010. It is displayed in Figure 9.16. As one may notice, the image is very noisy due to its long history. Moreover, in some parts of the image, the signal is almost completely attenuated by the noise: this is the case in high dip areas, where the high slope of the geological interface made it hard to retrieve a satisfying level of signal from the seismic measurements.

The first step was to derive a generic variogram model composed of the signal and noise structures. Random and linear noises were identified and characterized through stationary covariance functions. This was done following the same approach as the one described by Magneron et al. (2009). In particular, a smooth component corresponding to the true signal was identified, and 5 additional noisy components characterized by global geometric anisotropies were identified.

This work was done by expert geophysicists, who used their prior knowledge of the dataset to separate what is supposed to be the noise from the signal (during the variogram modeling step).

Then, local dips were assigned to the signal to be consistent with the geological structure. This was done using the *Paleoscan*$^{\text{TM}}$ software from the French company ELIIS, which allows to identify the global shape of some geological interfaces from noisy seismic images (cf. Figure 9.17). The angles describing locally these interfaces were extracted and interpolated on the whole domain. They serve as local anisotropy angles defining the signal to extract with the filtering process.

Thus, with local anisotropies defined, the geostatistical filtering approach allowed the filtering

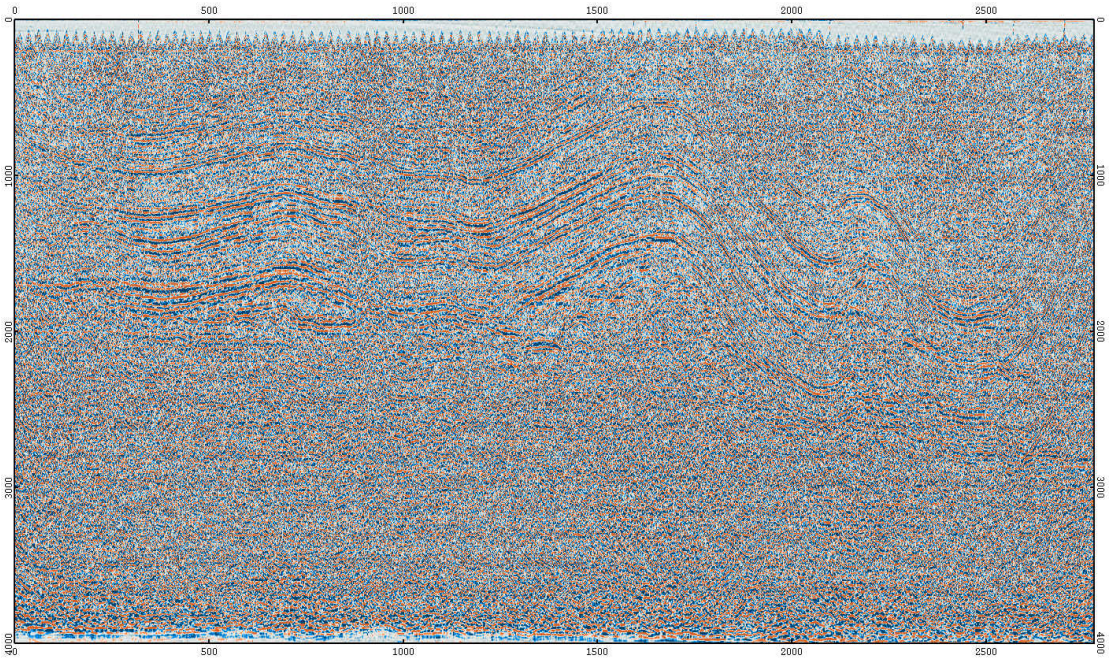


Figure 9.16: Input seicmic data from the Amedeus basin (Courtesy of CENTRAL PETROLEUM). The data form a 2778x1001 grid.

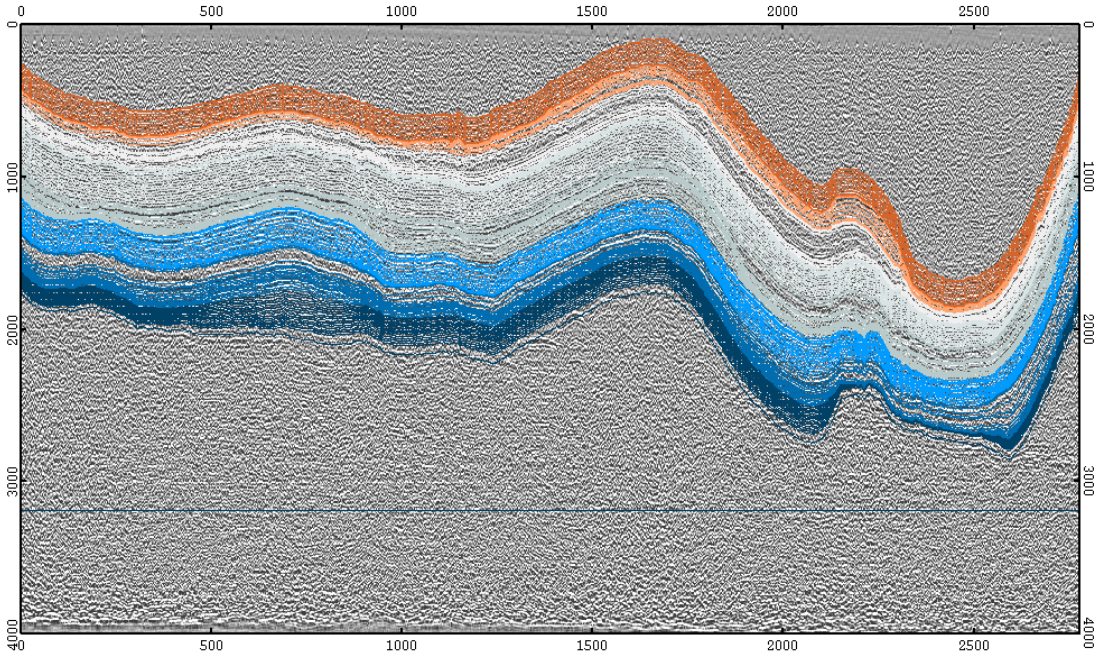
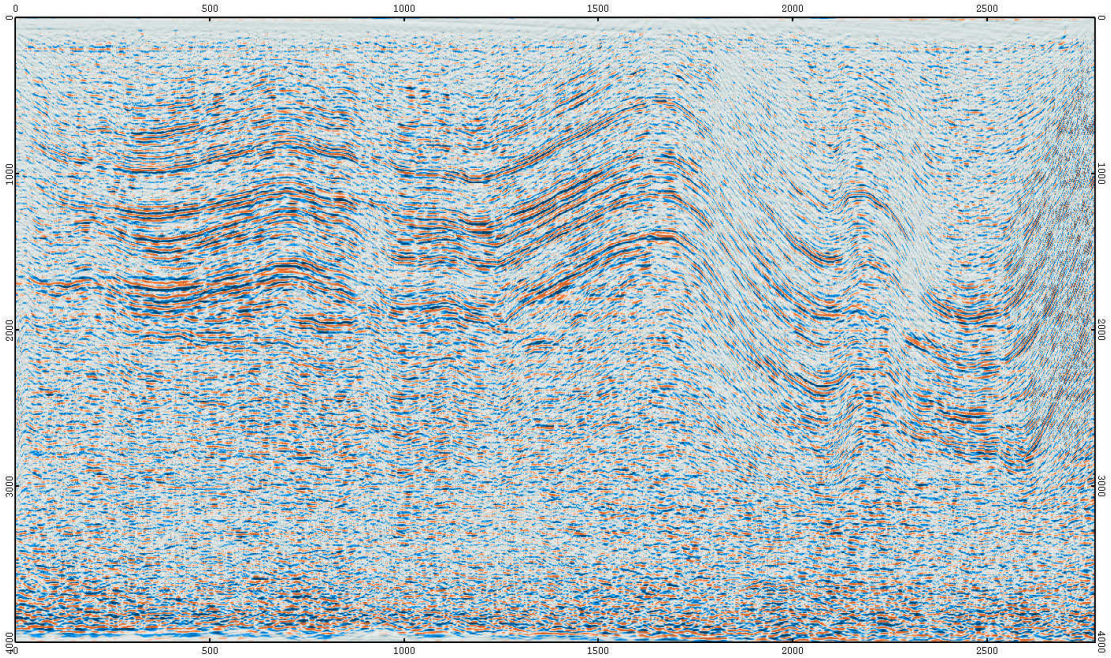


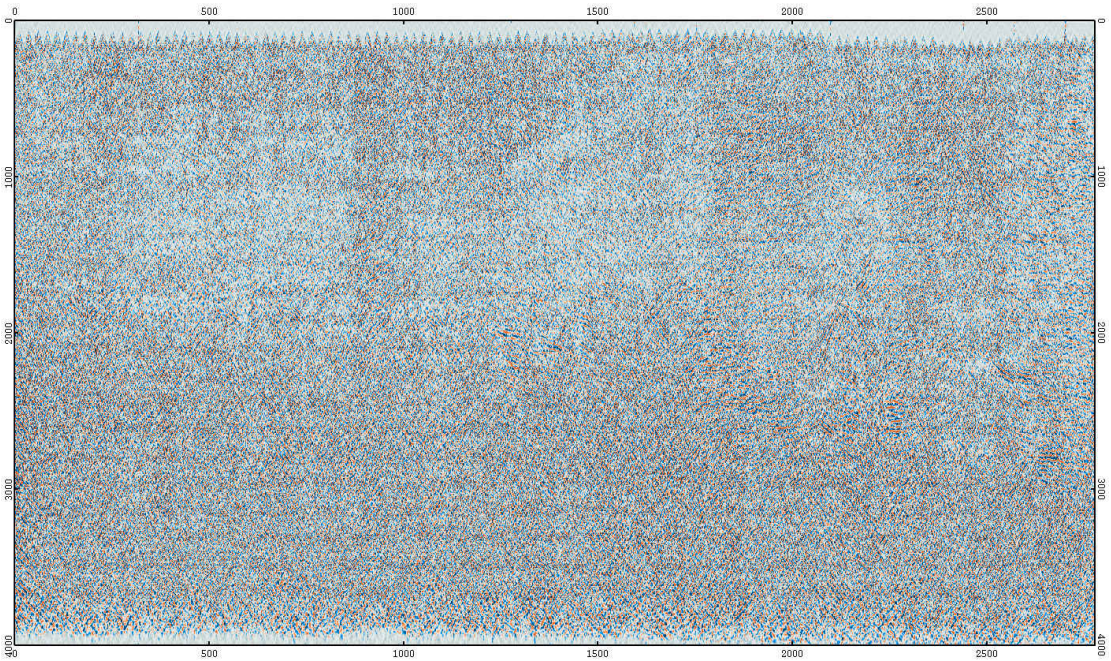
Figure 9.17: Identification of some geological interfaces from the noisy data of the Amadeus basin using the *Paleoscan*$^{\text{TM}}$ software.

out of noise while preserving the true signal. This is clearly demonstrated in Figure 9.18 where there is no obvious signal remaining in the noise image. More impressive was the restoration of the signal in the high dip areas, which was only possible using our filtering approach, since the linear noise interferes strongly with the signal in these parts of the image.

The results were validated by CENTRAL PETROLEUM, and it was proposed to use this approach of geostatistical filtering solution as a valid alternative to expensive full seismic reprocessing of seismic lines.



(a) Filtered output.



(b) Noise extracted.

Figure 9.18: Results obtained from the filtering process to the noisy data of the Amadeus basin.

9.3 Inference of non-stationary fields

Following the work presented in the last two sections, we describe here how the problem of inferring the covariance parameters of a non-stationary Gaussian field or of a field defined on a smooth surface, can be linked to the inference of the covariance parameters of a stochastic graph signal.

Let us assume that a non-stationary Gaussian random field Z is observed at n_o locations $\boldsymbol{p}_1, \dots, \boldsymbol{p}_{n_o}$ of a domain $\mathcal{M}$ (which can be a smooth surface). The idea is to once again model Z through the finite element representation given by Equation (9.5). Then, following the same approach as in Section 9.2.1, we get

$$\begin{pmatrix} Z(\boldsymbol{p}_1) \\ \vdots \\ Z(\boldsymbol{p}_{n_o}) \end{pmatrix} = \boldsymbol{M}_o \boldsymbol{Z}$$

where $\boldsymbol{M}_o$ is given by Equation (9.7) and $\boldsymbol{Z}$ is a stochastic graph signal having the same spectral density as the Gaussian field Z. In particular, $\boldsymbol{Z}$ is $\boldsymbol{S}$-stationary where $\boldsymbol{S}$ is the scaled stiffness matrix in Theorem 7.3.5 and is therefore defined by the triangulation of the domain and the field of local anisotropies.

By assuming that the observed values of Z are affected by a measurement error modeled by independent zero-mean Gaussian variables with variance τ^2, the inference problem of Z falls under the scope of the inference of stochastic graph signals studied in Chapter 5. Hence, using parametrized families of spectral densities, measurement error variances and fields of anisotropies (which in turn can define parametrized families of shift operators), the algorithms presented in Chapter 5 could tackle the inference linked to non-stationary fields and to fields defined on complex domains.

We now present an implementation of this inference paradigm on a simple case. We generate and then sample a simulation of a non-stationary field on a 200x200 grid (cf. Figure 9.19). We assume that the local anisotropies are known, as well as the measurement error variance (which was set to a negligible value). Only the parameters of the covariance function/spectral density of the field are therefore estimated. We therefore fall into the scope of Section 5.3 and an EM approach is used (and in particular the one described in Algorithm 5.5).

Following the advice of Section 5.3, the inference is performed while using the Markovian assumption, meaning that the inverse of the spectral density is parametrized as a polynomial. Besides, the shift operator was computed and stored once and for all, and its eigenvalues were computed once and for all to ease all determinant computations. The quality of this approximation is actually evaluated in Figures 9.20 and 9.21.

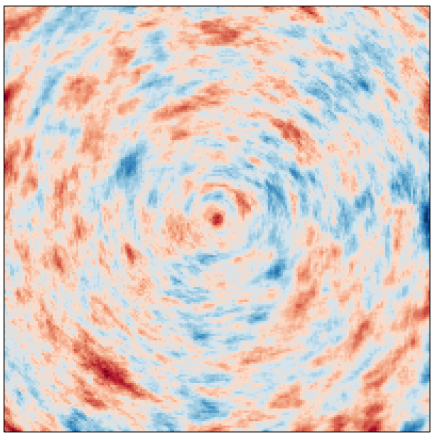


Figure 9.19: Simulated field used for the inference study. Matérn field with smoothness parameter $\pi/4$, ranges along the two principal axes of 50 and 10, sill 1, and local anisotropies distributed along concentric circles.

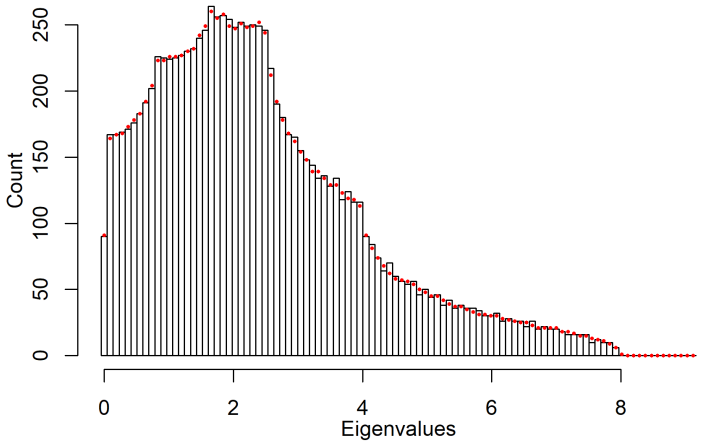


Figure 9.20: Comparison of the true histogram (black) and the one estimated using the approach of Section 2.3.2 (red). For computational reasons (due to the fact that the real eigenvalues are computed), this study was performed on a 100x100 grid with the same anisotropies as in Figure 9.19.

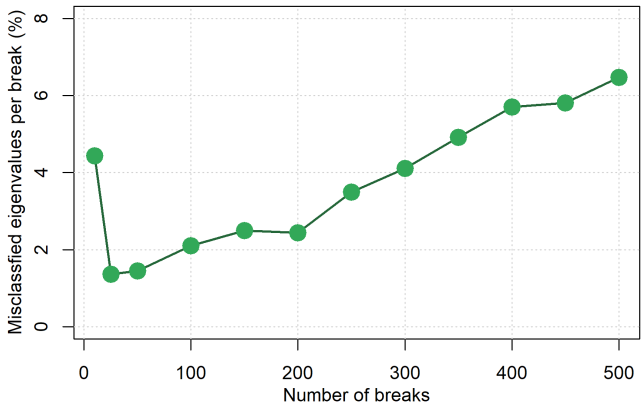
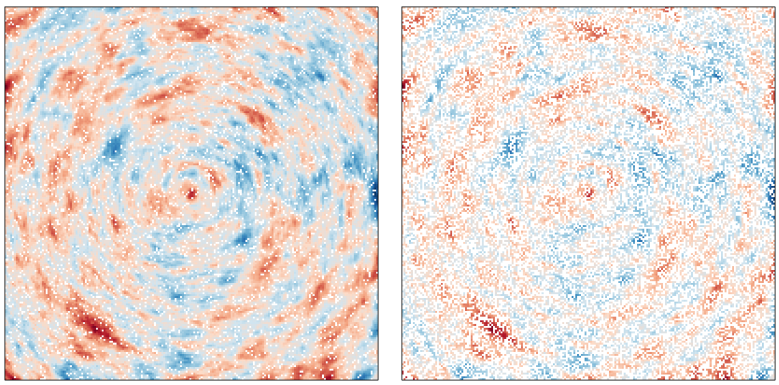


Figure 9.21: Evolution, with the number of breaks, of the mean proportion approximated eigenvalues that are missclassified in histogram bins. For computational reasons (due to the fact that the real eigenvalues are computed), this study was performed on a 100x100 grid with the same anisotropies as in Figure 9.19.

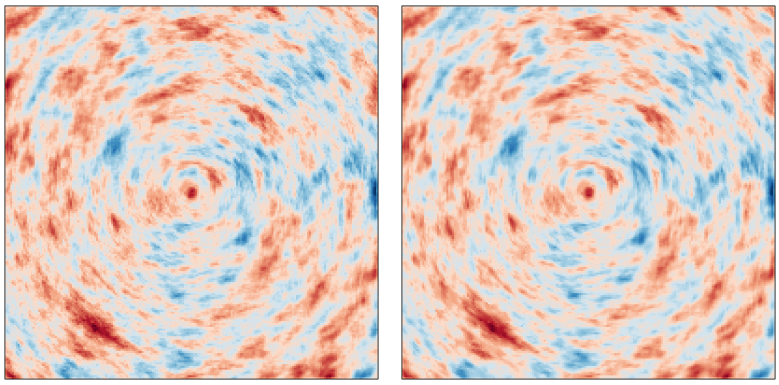
The results obtained from the inference are displayed in Figure 9.22. In particular, given that Algorithm 5.5 is used, the kriging estimate and conditional simulations are byproducts of the inference task. They are also displayed in Figure 9.22 (for the last iteration of the algorithm). Hence, the spectral density of the field is perfectly reconstituted when a small number of points are removed from the simulation. However the quality of the approximation deteriorates quickly when more points are removed. In particular, we observe that when only a third of the grid points are left, the algorithm does not yield an accurate estimate. We think that this is due to the fact that the sampling does not provide enough information to recreate the covariance structure.

Conclusion

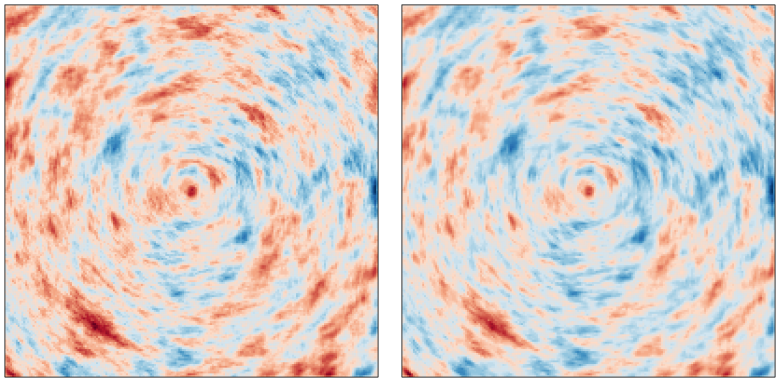
We showed in this chapter how the link between, on one hand, non-stationary random fields and fields defined on complex domains, and on the other hand stochastic graph signal processing could be leveraged to yield new approaches to the simulation, the prediction and the inference of such fields. This link comes from the identification of the Gaussian fields to generalized random fields defined on an appropriate Riemannian manifold, and then discretizing these generalized random fields using for instance the finite element method.



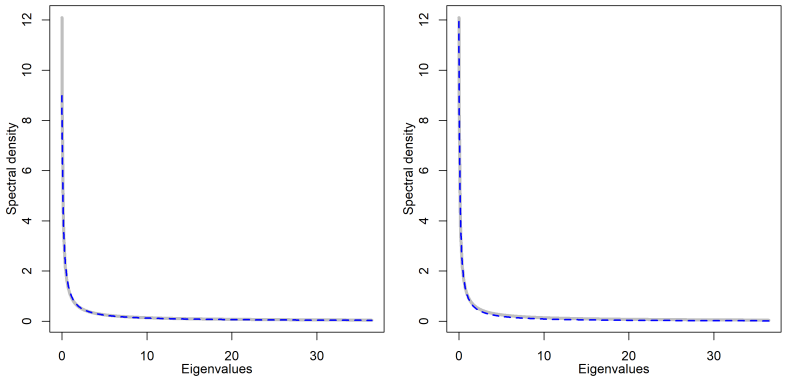
(a) Points removed.



(b) Kriging estimate.



(c) Example of conditional simulation.



(d) Spectral density obtained.

Figure 9.22: Results from the inference process. The left images correspond to the case where 10% of the grid points were removed and the right image to the case where 50% of the grid points were removed.

Hence the tasks of simulating, predicting and inferring these complex Gaussian random fields were reduced to performing the same tasks on stochastic graph signals. The methods introduced in Chapters 3 to 5 were then applied in this context. Examples of applications to both synthetic and real data were showcased.

The approach for modeling Gaussian random fields presented in this work is promising. Within the same framework both complex domains and local anisotropies are treated, which frees the user from wondering which type of approach to use. Moreover the ability to accurately and easily incorporate prior information on the structure of a Gaussian field into its model offers a great flexibility to the practitioner, as illustrated in the case studies.

Conclusion

Summary and contributions

This work aimed at designing an approach to circumvent two problems often encountered when dealing with spatial data:

- Finding a model suited to complex geostatistical data: by complex we mean data that either lie on a space that cannot be considered as a "chunk" of Euclidean space (eg. a smooth surface in $\mathbb{R}^3$) or has to be modeled with a non-stationary model.
- The big n problem: building the covariance matrices necessary to perform simulations, prediction and inference based on the spatial data becomes untractable in large-scale settings.

Let Z denote a zero-mean Gaussian random field that we which to use to model our spatial data, as described above. We assume that if Z is non-stationary, its covariance is characterized by a *known* field of local anisotropy parameters. Hence, the proposed modeling approach should allow to take in this information as smoothly as possible.

To solve the problems stated above, we propose to redefine the Gaussian fields used to model the spatial data within the framework of Riemannian manifolds, using the notion of generalized random field. Let us first recall quickly these notions.

A Riemannian manifold is the association of a manifold $\mathcal{M}$ with a Riemannian metric g. On one hand, the manifold $\mathcal{M}$ is a set that can locally be considered as Euclidean. On the other hand, the Riemannian metric g is an application that smoothly associates to each point $\boldsymbol{p} \in \mathcal{M}$ an inner product on the tangent space of $\mathcal{M}$ at $\boldsymbol{p}$. In particular, g can be represented by a field of positive definite matrices $\{\boldsymbol{G}(\boldsymbol{p})\}_{\boldsymbol{p}\in\mathcal{M}}$ defining these inner products.

Hence, the two components that define a Riemannian manifold $(\mathcal{M}, g)$ are particularly suited to answer the modeling question asked in this work. Indeed, the manifold $\mathcal{M}$ is used to represent the domain on which the data lie. If Z is non-stationary, the Riemannian metric g is used to model the local anisotropies defining the non-stationary model. In particular, for $\boldsymbol{p} \in \mathcal{M}$, the matrix $\boldsymbol{G}(\boldsymbol{p})$ defining the metric is built as

$$\boldsymbol{G}(\boldsymbol{p}) = \boldsymbol{R}(\boldsymbol{p})\boldsymbol{D}(\boldsymbol{p})^2\boldsymbol{R}(\boldsymbol{p})^T \quad ,$$

where $\boldsymbol{R}(\boldsymbol{p})$ is the rotation matrix defined from the anisotropy angle(s) at $\boldsymbol{p}$ and $\boldsymbol{D}(\boldsymbol{p})$ is the diagonal matrix whose entries are the inverse of the anisotropy ranges (cf. Section 7.4.2).

On this problem-dependent Riemannian manifold, we propose to work with a class of generalized random fields defined from the Laplace-Beltrami operator of $(\mathcal{M}, g)$ and which satisfies

$$\mathcal{Z} = \sum_{j\in\mathbb{N}} \gamma(\lambda_j) W_j e_j \quad , \tag{9.11}$$

where

- $\{\lambda_j\}_{j\in\mathbb{N}}$ (resp. $\{e_j\}_{j\in\mathbb{N}}$) is the set of eigenvalues (resp. eigenvectors) of the negative Laplace-Beltrami operator of $(\mathcal{M}, g)$,

- γ is the square-root of the spectral density of an isotropic covariance function C.

As defined, $\mathcal{Z}$ is the counterpart on the Riemannian manifold of an isotropic random field with covariance $C = \mathscr{F}^{-1}[\gamma^2]$. Hence, $\mathcal{Z}$ answers the modeling problem initially posed. On one hand, the fact that it is defined from a manifold-dependent operator ensures that it is adapted to geometry of the manifold (and therefore of the domain on which the data lie). On the other hand, the local geometry induced by the Riemannian metric ensures that the field of local anisotropies is honored (cf. Section 7.2.2).

Working directly with Equation (9.11) would require to compute the eigenfunctions and eigenvalues of the Laplace-Beltrami operator. We rather propose to approximate $\mathcal{Z}$ by a weighted sum of user-defined deterministic and n linearly independent functions $\{\psi_k\}_{1\leq k\leq n}$. Finally, we use this sum to characterize the field Z we wished to model from the start, thus giving:

$$Z = \sum_{k=1}^{n} Z_k \psi_k \quad , \tag{9.12}$$

where $Z_1, \ldots, Z_n$ is a set of zero-mean (correlated) Gaussian weights.

In order to characterize Z, we derived in Theorem 7.3.5 the expression of the covariance matrix of the weights in Equation (9.12). This result gives that $\boldsymbol{Z} = (Z_1, \ldots, Z_n)$ forms a Gaussian vector with covariance matrix

$$\mathrm{Var}[\boldsymbol{Z}] = \boldsymbol{C}^{-1/2}\gamma^2(\boldsymbol{S})\boldsymbol{C}^{-1/2} \quad , \tag{9.13}$$

where $\boldsymbol{C}$ is a diagonal matrix with entries $C_{ii} = \langle \psi_i, 1\rangle_{L^2(\mathcal{M})} > 0$ and $\boldsymbol{S}$ is a symmetric positive semi-definite matrix with entries $S_{ij} = \langle \nabla_{\mathcal{M}}\psi_i, \nabla_{\mathcal{M}}\psi_j\rangle_{L^2(\mathcal{M})}/\sqrt{C_{ii}C_{jj}}$. Note in particular that the entries of the matrices $\boldsymbol{C}$ and $\boldsymbol{S}$ are defined using the inner product (and the gradient operator) defined on the Riemannian manifold, and therefore account for the local anisotropies through the metric.

At this point the modeling problem stated at the beginning of this conclusion is solved. Tackling the big n problem is then done in two steps.

First, we propose to set the functions $\{\psi_k\}_{1\leq k\leq n}$ in Equation (9.12) to be the basis functions of the finite element method. Hence the domain $\mathcal{M}$ is triangulated and each ψ_k is attached to one of the nodes of the triangulation and has a support limited to the triangles to which the node belongs. Consequently, the resulting matrix $\boldsymbol{S}$ in Equation (9.13) is sparse. The idea is then to look for so-called matrix-free approaches, that would allow to work with the covariance matrix of Equation (9.13) without actually having to build it. In particular, the approach we end up proposing only relies on product by $\boldsymbol{S}$, which sparse. Hence both computational and storage costs are saved.

Indeed, we note that Z is in fact entirely determined by the weight vector $\boldsymbol{Z}$, which in turn can be deduced from the vector $\boldsymbol{X}$ defined by

$$\boldsymbol{X} = \boldsymbol{C}^{1/2}\boldsymbol{Z} \quad .$$

Hence, the inference, simulation and prediction tasks we wish to perform on Z can be transferred to $\boldsymbol{X}$, which is a Gaussian vector with covariance matrix given by

$$\mathrm{Var}[\boldsymbol{X}] = \gamma^2(\boldsymbol{S}) \quad .$$

As defined, the vector $\boldsymbol{X}$ can be interpreted as a (stationary) stochastic graph signal on the graph whose vertices and edges are the vertices and edges of the triangulation of $\mathcal{M}$. Using the framework of graph signal processing, we introduced the Chebyshev filtering algorithm, which aims at computing (an approximation of) products by a matrix function $h(\boldsymbol{S})$ (for a real function h). This algorithm basically replaces a product $h(\boldsymbol{S})$ by a product by a polynomial matrix, defined from the Chebyshev series approximation of h over the set of eigenvalues of $\boldsymbol{S}$. Then the product by the polynomial matrix is performed with a linear complexity through an iterative approach that only involves products by $\boldsymbol{S}$.

We proposed methods based on the Chebyshev filtering algorithm to perform simulations, estimations and inference of stochastic graph signals (cf. Chapters 3 to 5). The derived algorithms can all be considered as matrix-free algorithms and as such, are highly scalable. These

algorithms were derived while keeping in mind that they would be used as proxies to perform the same tasks on a Gaussian field Z defined by Equation (9.12).

For the simulation algorithm, we evaluated both the numerical and the statistical errors induced by using the approximations of the Chebyshev filtering algorithm. In particular we derived criteria on the approximation error of the polynomial approximation so that the resulting simulation would be statistically indiscernible from simulations performed with an exact algorithm. The estimation problems were formulated to echo the kriging (and the factorial kriging) estimates classically encountered in Geostatistics. As for the inference problem, approaches based on likelihood maximization and on the EM algorithm were proposed.

Circling back to our initial problem, we applied the modeling approach introduced in this work to several case studies involving both real and synthetic data (cf. Chapter 9). It showed how the method was able to accurately account for non-stationary models and for the geometry of the domain. Hence this new approach provides new tools to better model spatial data, and easily take into account prior structural information about the field we wish to model.

Future work

Estimation of local anisotropies

A first extension of this work concerns the inference of non-stationary fields. Indeed, in this work, we assumed that the field of local anisotropies characterizing the random field was known. If we drop this assumption, then the field of anisotropies must also be inferred from the data. The inference algorithms introduced in Chapter 5 actually account for this case. Indeed, they assumed that the shift operator, which is defined from the local anisotropies, could also parametrized. Hence, we just need to define a parametrization of the local anisotropies across the domain. For a two-dimensional domain, Fuglstad et al. (2015) proposed to represent the local anisotropies as a vector field, defined as the gradient of a real-valued sinusoidal function of $\mathbb{R}^2$. This idea could be generalized to more general functions.

Another idea to model local anisotropies consists in estimating the field of values of each parameter directly across the domain. In an inference context, given an anisotropy parameter θ, we could set up a number of "anchor" points across the domain, and define the value of θ at each point of the domain from the values of θ at the anchor points. This can be done using for instance splines or Gaussian processes.

Finally, we could once again identify the field of local anisotropies with a Riemannian metric defined across the domain and try to directly estimate the metric from the data. Indeed, some approaches aiming to estimate the metric of a Riemannian manifold, given the observation of some data points in this manifold, have been introduced in Machine Learning applications (Lebanon, 2002; Peltonen et al., 2004), and might represent an alternative to the inference process proposed above.

Spatio-temporal models

A second extension of the work presented in this dissertation would be spatio-temporal modeling. A starting point could be to consider non-separable space-time models defined from transport and diffusion stochastic partial differential equations (SPDEs) (Lindgren et al., 2011). These models were extensively studied by Carrizo Vergara (2018). They naturally generalize SPDEs used to define spatial random fields, to a spatio-temporal framework. In particular, we propose to consider (generalized) random fields defined on a Riemannian manifold $(\mathcal{M}, g)$ as solutions of a SPDE of the form

$$\frac{\partial Z}{\partial t}(t, \boldsymbol{p}) + f(-\Delta_{\mathcal{M}})Z(t, \boldsymbol{p}) = \mathcal{E}(t, \boldsymbol{p}) \tag{9.14}$$

where f is a function of the real line (with appropriate regularity) and $\mathcal{E}$ is a stochastic spatio-temporal noise field. Such SPDEs are the counterparts on Riemannian manifolds of the evolutions models introduced by Carrizo Vergara (2018) (using the analogy between functions of Laplacian and pseudo-differential operators).

Such models can be numerically solved through a discretization of the equation using an Euler or a Crank-Nicholson scheme associated with a spatial finite element approach (Lindgren et al., 2011; Thomas, 2013). In particular, the solutions obtained at each time step then depend

on the stiffness matrix defined from the finite element approach, and would therefore open the path to using matrix-free algorithms to ensure the scalability of the procedure.

Another idea to deal with Equation (9.14) would be to try to solve it directly. Indeed, notice that by setting $\mathcal{E} = 0$ and f to be the identity map, Equation (9.14) reduces to the heat equation defined on $(\mathcal{M}, g)$. Extensive literature exists on the characterization of the solutions of the heat equation on a Riemannian manifold, as they are directly linked to the eigendecomposition of the Laplacian $-\Delta_{\mathcal{M}}$ (cf. (Craioveanu et al., 2013)). Extending the resolution method of the heat equation to the "generalized" version proposed in Equation (9.14) could provide a starting point to derive solutions of Equation (9.14).

Finally, note that working with a Riemannian manifold $(\mathcal{M}, g)$ would once again offer a way to define fields on $\mathcal{M}$ with local anisotropies by setting the metric g accordingly. Moreover, transport terms can also be introduced by adjusting the metric appropriately. Indeed, if $(\mathcal{M}, g)$ is a Riemannian d-manifold, with Laplace-Beltrami operator $\Delta_{\mathcal{M}}$ and gradient $\nabla_{\mathcal{M}}$, and if $a \in \mathscr{C}^{\infty}(\mathcal{M})$ such that a is strictly positive, then $g' = a \cdot g$ also defines a Riemannian metric on $\mathcal{M}$ and the Laplace-Beltrami operator $\Delta'_{\mathcal{M}}$ associated with $(\mathcal{M}, g')$ satisfies (Craioveanu et al., 2013, Equation 2.21):

$$\Delta'_{\mathcal{M}} = \frac{1}{a}\Delta_{\mathcal{M}} + \left(1 - \frac{d}{2}\right)\frac{1}{a^2}\nabla_{\mathcal{M}} a \quad ,$$

where $\nabla_{\mathcal{M}} a$ is defined in Section 6.5.1 and can therefore be interpreted as a transport vector.

A Mathematical toolbox

A.1 Differential calculus

Let $f : A \subset \mathbb{R}^d \to \mathbb{R}$ be a differentiable function defined on a set $A \subset \mathbb{R}^d$. The gradient of f at a point $\boldsymbol{x} \in A$, is the vector $\nabla f(\boldsymbol{x}) \in \mathbb{R}^d$ defined by

$$[\nabla f(\boldsymbol{x})]_i = \partial_i f(\boldsymbol{x}), \quad i \in [\![1, d]\!] \quad ,$$

where, $\forall i \in [\![1, d]\!]$, $\partial_i f(\boldsymbol{x})$ denotes the i-th partial derivative of f at the point $\boldsymbol{x}$, i.e.

$$\partial_i(f)(\boldsymbol{x}) = \lim_{t \to 0} \frac{f(\boldsymbol{x} + t\boldsymbol{c}_i) - f(\boldsymbol{x})}{t} \quad ,$$

where $\boldsymbol{c}_i \in \mathbb{R}^d$ is the i-th vector of the canonical base of $\mathbb{R}^d$: $\forall k \in [\![1, d]\!]$, $[\boldsymbol{c}_i]_k = \delta_{ik}$.

Let $\phi : \mathscr{D}(\phi) \subset \mathbb{R}^d \to \mathbb{R}^n$ be a differentiable function defined on a set $\mathscr{D}(\phi) \subset \mathbb{R}^d$. In particular, ϕ can be written as $\phi = (\phi_1, \ldots, \phi_n)$ where $\phi_1, \ldots, \phi_n$ are real-valued differentiable functions defined on $\mathscr{D}(\phi)$ called coordinate functions of ϕ and satisfying:

$$\forall \boldsymbol{x} \in \mathscr{D}(\phi), \quad \phi(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}), \ldots, \phi_n(\boldsymbol{x}))^T \in \mathbb{R}^n \quad .$$

The Jacobian matrix of ϕ at a point $\boldsymbol{x} \in \mathbb{R}^d$ is the matrix $J_\phi(\boldsymbol{x}) \in \mathscr{M}_{n,d}(\mathbb{R})$ defined by

$$\forall i \in [\![1, n]\!], j \in [\![1, d]\!], \quad [J_\phi(\boldsymbol{x})]_{ij} = \partial_j \phi_i(\boldsymbol{x}) \quad .$$

In particular, $J_\phi(\boldsymbol{x})$ is the matrix whose rows are the gradients of each coordinate function of ϕ.

Let then introduce a second differentiable function $\psi : \mathscr{D}(\psi) \subset \mathbb{R}^n \to \mathbb{R}^m$ defined on a domain $\mathscr{D}(\psi) \subset \mathbb{R}^n$ that contains the image of $\mathscr{D}(\phi)$ by ϕ, i.e. $\phi(\mathscr{D}(\phi)) \subset \mathscr{D}(\psi)$. Then, the composition $\psi \circ \phi : \mathscr{D}(\phi) \to \mathbb{R}^m$ is well-defined. The following theorem, called *chain rule*, expresses the derivatives of $\psi \circ \phi$ in function of the derivatives of ϕ and ψ.

Theorem A.1.1 (Chain rule)**.** *Let $\phi : \mathscr{D}(\phi) \subset \mathbb{R}^d \to \mathbb{R}^n$ and $\psi : \mathscr{D}(\psi) \subset \mathbb{R}^n \to \mathbb{R}^m$ be two differentiable functions defined as above.*

Then, the Jacobian matrix of $\psi \circ \phi$ satisfies

$$\forall \boldsymbol{x} \in \mathscr{D}(\phi), \quad J_{\psi \circ \phi}(\boldsymbol{x}) = J_\psi(\phi(\boldsymbol{x})) J_\phi(\boldsymbol{x}) \quad .$$

Hence, we have $\forall \boldsymbol{x} \in \mathscr{D}(\phi)$, and $\forall i \in [\![1, m]\!], j \in [\![1, d]\!]$,

$$\partial_j(\psi \circ \phi)_i(\boldsymbol{x}) = \sum_{k=1}^{n} \partial_k \psi_i(\phi(\boldsymbol{x})) \cdot \partial_j \phi_k(\boldsymbol{x}) \quad .$$

Proof. See (Wilfred, 2002, Sections 2.8 & 2.9). □

Theorem A.1.2 (Change of variable)**.** *Let U, V be open sets in $\mathbb{R}^n$, $\Phi : U \to V$ be a diffeomorphism and $f : V \to \mathbb{R}$ a continuous function. Then for any compact $A \subset U$,*

$$\int_A f \circ \Phi(y) |\det J_\phi(\boldsymbol{y})| d\boldsymbol{y} = \int_{\Phi(A)} f(\boldsymbol{z}) d\boldsymbol{z}$$

where $\det J_\phi(\boldsymbol{y})$ is the Jacobian matrix of Φ at $\boldsymbol{y}$.

A.2 Linear algebra

A.2.1 Rayleigh quotient

Let $\boldsymbol{n} \in \mathbb{N}^*$. Let $\boldsymbol{M} = (M_{ij})_{i,j \in [\![1,n]\!]}$ be a real symmetric $n \times n$ matrix. $\boldsymbol{M}$ is diagonalizable in an orthonormal basis. Denote $\lambda_{\min} = \lambda_1 \leq \cdots \leq \lambda_n = \lambda_{\max}$ its eigenvalues, and $(\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(n)})$ the corresponding eigenvectors (forming the orthonormal basis).

Definition A.2.1. *The **Rayleigh quotient** $R(\boldsymbol{M}, \boldsymbol{x})$ associated with $\boldsymbol{M}$ and $\boldsymbol{x} \in (\mathbb{R}^n)^*$ is the ratio:*

$$R(\boldsymbol{M}, \boldsymbol{x}) = \frac{\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}} = \left(\frac{\boldsymbol{x}}{\|\boldsymbol{x}\|} \right)^T \boldsymbol{M} \left(\frac{\boldsymbol{x}}{\|\boldsymbol{x}\|} \right)$$

Proposition A.2.1. $\forall \boldsymbol{x} \in (\mathbb{R}^n)^*, \quad \lambda_{\min} \leq R(\boldsymbol{M}, \boldsymbol{x}) \leq \lambda_{\max}$

Proof. Simply notice that $\boldsymbol{x}$ can be decomposed in the orthonormal basis $(\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(n)})$ as: $\boldsymbol{x} = \sum_{i=1}^n \alpha_i \boldsymbol{v}^{(i)}$ for some $(\alpha_1, \ldots, \alpha_n)^T \in (\mathbb{R}^n)^*$. Then,

$$R(\boldsymbol{M}, \boldsymbol{x}) = \frac{\sum_{i=1}^n \lambda_i \alpha_i^2}{\sum_{j=1}^n \alpha_j^2} = \sum_{i=1}^n \lambda_i \frac{\alpha_i^2}{\sum_{j=1}^n \alpha_j^2}$$

which is just a weighted sum of the eigenvalues with positive weights. □

A.2.2 Block matrices

Let $n \geq 1$ and $k \in [\![1, n-1]\!]$. For a vector $\boldsymbol{b} \in \mathbb{R}^n$ we denote

$$\boldsymbol{b} = \begin{pmatrix} \boldsymbol{b}_k \\ \boldsymbol{b}_{-k} \end{pmatrix}$$

the partition of $\boldsymbol{A}$ such that $\boldsymbol{b}_k \in \mathbb{R}^k$ corresponds to its k first entries and $\boldsymbol{b}_{-k} \in \mathbb{R}^{n-k}$ corresponds to the remaining ones. For a matrix $\boldsymbol{A} \in \mathscr{M}_n(\mathbb{R})$. We denote

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A}_{k,k} & \boldsymbol{A}_{k,-k} \\ \boldsymbol{A}_{-k,k} & \boldsymbol{A}_{-k,-k} \end{pmatrix} \tag{A.1}$$

the partition of $\boldsymbol{A}$ such that $\boldsymbol{A}_{k,k} \in \mathscr{M}_k(\mathbb{R})$ corresponds to its first k rows and first k columns; $\boldsymbol{A}_{-k,k} \in \mathscr{M}_{n-k,k}(\mathbb{R})$ corresponds to its last $n-k$ rows and first k columns; $\boldsymbol{A}_{k,-k} \in \mathscr{M}_{k,n-k}(\mathbb{R})$ corresponds to its first k rows and last k columns; $\boldsymbol{A}_{-k,-k} \in \mathscr{M}_{n-k}(\mathbb{R})$ corresponds to its last $n-k$ rows and last $n-k$ columns of $\boldsymbol{A}$.

Let $\boldsymbol{A} \in \mathscr{M}_n(\mathbb{R})$ be an invertible matrix. Denote $\boldsymbol{B} = \boldsymbol{A}^{-1}$ its inverse, which we partition as:

$$\boldsymbol{B} = \begin{pmatrix} \boldsymbol{B}_{k,k} & \boldsymbol{B}_{k,-k} \\ \boldsymbol{B}_{-k,k} & \boldsymbol{B}_{-k,-k} \end{pmatrix} \tag{A.2}$$

The Schur complement of the block $\boldsymbol{A}_{k,k}$ of $\boldsymbol{A}$ is the matrix $[\boldsymbol{A}|\boldsymbol{A}_{k,k}] \in \mathscr{M}_{n-k}(\mathbb{R})$ defined by:

$$[\boldsymbol{A}|\boldsymbol{A}_{k,k}] = \boldsymbol{A}_{-k,-k} - \boldsymbol{A}_{-k,k} \boldsymbol{A}_{k,k}^{-1} \boldsymbol{A}_{k,-k}$$

Similarly, the Schur complement of the block $\boldsymbol{A}_{-k,-k}$ of $\boldsymbol{A}$ is the matrix $[\boldsymbol{A}|\boldsymbol{A}_{-k,-k}] \in \mathscr{M}_k(\mathbb{R})$ defined by:

$$[\boldsymbol{A}|\boldsymbol{A}_{-k,-k}] = \boldsymbol{A}_{k,k} - \boldsymbol{A}_{k,-k} \boldsymbol{A}_{-k,-k}^{-1} \boldsymbol{A}_{-k,k}$$

The Schur complements can be used to provide a factorization of a square matrix as

$$\begin{aligned}\boldsymbol{A} &= \begin{pmatrix} \boldsymbol{I} & \\ \boldsymbol{A}_{-k,k}\boldsymbol{A}_{k,k}^{-1} & \boldsymbol{I} \end{pmatrix} \begin{pmatrix} \boldsymbol{A}_{k,k} & \\ & [\boldsymbol{A}|\boldsymbol{A}_{k,k}] \end{pmatrix} \begin{pmatrix} \boldsymbol{I} & \boldsymbol{A}_{k,k}^{-1}\boldsymbol{A}_{k,-k} \\ & \boldsymbol{I} \end{pmatrix} \\ &= \begin{pmatrix} \boldsymbol{I} & \boldsymbol{A}_{k,-k}\boldsymbol{A}_{-k,-k}^{-1} \\ & \boldsymbol{I} \end{pmatrix} \begin{pmatrix} [\boldsymbol{A}|\boldsymbol{A}_{-k,-k}] & \\ & \boldsymbol{A}_{-k,-k} \end{pmatrix} \begin{pmatrix} \boldsymbol{I} & \\ \boldsymbol{A}_{-k,-k}^{-1}\boldsymbol{A}_{-k,k} & \boldsymbol{I} \end{pmatrix}\end{aligned} \tag{A.3}$$

Hence, the determinant of $\boldsymbol{A}$ can be expressed in function of those of its Schur complements:

$$|\boldsymbol{A}| = |[\boldsymbol{A}|\boldsymbol{A}_{k,k}]| \cdot |\boldsymbol{A}_{k,k}| = |[\boldsymbol{A}|\boldsymbol{A}_{-k,-k}]| \cdot |\boldsymbol{A}_{-k,-k}| \tag{A.4}$$

As for the inverse of $\boldsymbol{A}$, it can be also be factorized using Shur complements, yielding the relation

$$\begin{aligned}\boldsymbol{A}^{-1} &= \begin{pmatrix} \boldsymbol{I} & -\boldsymbol{A}_{k,k}^{-1}\boldsymbol{A}_{k,-k} \\ & \boldsymbol{I} \end{pmatrix} \begin{pmatrix} \boldsymbol{A}_{k,k}^{-1} & \\ & [\boldsymbol{A}|\boldsymbol{A}_{k,k}]^{-1} \end{pmatrix} \begin{pmatrix} \boldsymbol{I} & \\ -\boldsymbol{A}_{-k,k}\boldsymbol{A}_{k,k}^{-1} & \boldsymbol{I} \end{pmatrix} \\ &= \begin{pmatrix} \boldsymbol{I} & \\ -\boldsymbol{A}_{-k,-k}^{-1}\boldsymbol{A}_{-k,k} & \boldsymbol{I} \end{pmatrix} \begin{pmatrix} [\boldsymbol{A}|\boldsymbol{A}_{-k,-k}]^{-1} & \\ & \boldsymbol{A}_{-k,-k}^{-1} \end{pmatrix} \begin{pmatrix} \boldsymbol{I} & -\boldsymbol{A}_{k,-k}\boldsymbol{A}_{-k,-k}^{-1} \\ & \boldsymbol{I} \end{pmatrix}\end{aligned} \tag{A.5}$$

This gives in particular the following correspondence between the blocks of $\boldsymbol{B} = \boldsymbol{A}^{-1}$ and Schur complements:

$$\begin{aligned}\boldsymbol{A}^{-1} = \boldsymbol{B} &= \begin{pmatrix} \boldsymbol{B}_{k,k} & \boldsymbol{B}_{k,-k} \\ \boldsymbol{B}_{-k,k} & \boldsymbol{B}_{-k,-k} \end{pmatrix} = \begin{pmatrix} [\boldsymbol{A}|\boldsymbol{A}_{-k,-k}]^{-1} & -\boldsymbol{A}_{k,k}^{-1}\boldsymbol{A}_{k,-k}[\boldsymbol{A}|\boldsymbol{A}_{k,k}]^{-1} \\ -[\boldsymbol{A}|\boldsymbol{A}_{k,k}]^{-1}\boldsymbol{A}_{-k,k}\boldsymbol{A}_{k,k}^{-1} & [\boldsymbol{A}|\boldsymbol{A}_{k,k}]^{-1} \end{pmatrix} \\ &= \begin{pmatrix} [\boldsymbol{A}|\boldsymbol{A}_{-k,-k}]^{-1} & -[\boldsymbol{A}|\boldsymbol{A}_{-k,-k}]^{-1}\boldsymbol{A}_{k,-k}\boldsymbol{A}_{-k,-k}^{-1} \\ -\boldsymbol{A}_{-k,-k}^{-1}\boldsymbol{A}_{-k,k}[\boldsymbol{A}|\boldsymbol{A}_{-k,-k}]^{-1} & [\boldsymbol{A}|\boldsymbol{A}_{k,k}]^{-1} \end{pmatrix}\end{aligned} \tag{A.6}$$

In particular, this relation provides an alternative expression for Equation (A.4):

$$|\boldsymbol{A}| = \frac{|\boldsymbol{A}_{k,k}|}{|\boldsymbol{B}_{-k,-k}|} = \frac{|\boldsymbol{A}_{-k,-k}|}{|\boldsymbol{B}_{k,k}|} \tag{A.7}$$

A.2.3 Geometric interpretation of positive-definite matrices

Let B be a domain of $\mathbb{R}^d$ and let $\{\boldsymbol{G}(\boldsymbol{p})\}_{\boldsymbol{p}\in B_R}$ be a set of positive definite symmetric matrices indexed by the points of B. Let Z be a random field defined on B such that

$$\forall \boldsymbol{p} \in B, \quad \mathrm{Cov}[Z(\boldsymbol{p}), Z(\boldsymbol{p}+\boldsymbol{h})] \underset{\boldsymbol{h}\to\boldsymbol{0}}{\sim} C_0(\sqrt{\boldsymbol{h}^T\boldsymbol{G}(\boldsymbol{p})\boldsymbol{h}}) \tag{A.8}$$

where C_0 denotes an isotropic covariance function.

For $d = 2$, given that the matrix $\boldsymbol{G}(\boldsymbol{p})$, $\boldsymbol{p} \in B$, is positive definite and symmetric, its diagonalization in an orthonormal basis can be written as

$$\boldsymbol{G}(\boldsymbol{p}) = \boldsymbol{V}_{\theta(\boldsymbol{p})} \begin{pmatrix} 1/\rho_1(\boldsymbol{p})^2 & \\ & 1/\rho_2(\boldsymbol{p})^2 \end{pmatrix} \boldsymbol{V}_{\theta(\boldsymbol{p})}^T, \quad \boldsymbol{V}_{\theta(\boldsymbol{p})} = \begin{pmatrix} \cos\theta(\boldsymbol{p}) & -\sin\theta(\boldsymbol{p}) \\ \sin\theta(\boldsymbol{p}) & \cos\theta(\boldsymbol{p}) \end{pmatrix}$$

where $\boldsymbol{\theta}(\boldsymbol{p}) \in]\pi/2, \pi/2]$ and $\rho_1(\boldsymbol{p}), \rho_2(\boldsymbol{p}) > 0$. Note that $\boldsymbol{V}_{\theta(\boldsymbol{p})}$ is a rotation matrix, and that $\boldsymbol{V}_{\theta(\boldsymbol{p})}^T = \boldsymbol{V}_{\theta(\boldsymbol{p})}^{-1} = \boldsymbol{V}_{-\theta(\boldsymbol{p})}$. Hence the change of coordinates

$$\boldsymbol{h}' = \begin{pmatrix} 1/\rho_1(\boldsymbol{p}) & \\ & 1/\rho_2(\boldsymbol{p}) \end{pmatrix} \boldsymbol{V}_{\theta(\boldsymbol{p})}^T \boldsymbol{h} \tag{A.9}$$

yields $\mathrm{Cov}[Z(\boldsymbol{p}), Z(\boldsymbol{p}+\boldsymbol{h})] = C_0(\|\boldsymbol{h}'\|)$. Thus the non-stationary covariance model of Z, which satisfies Equation (A.8) in the reference coordinate system, is locally turned into an isotropic model through the transformation of Equation (A.9). In particular, the parameters $\theta(\boldsymbol{p})$ and $\rho_1(\boldsymbol{p}), \rho_2(\boldsymbol{p})$ can be interpreted as follows

- $\theta(\boldsymbol{p})$ and $\theta(\boldsymbol{p}) + \pi/2$ define the main directions of anisotropy, i.e. the two directions along which the covariance function C behaves like an isotropic model. Hence, along one of these directions $\boldsymbol{v}_{\boldsymbol{\theta}(\boldsymbol{p})}$, we have $\forall h \in \mathbb{R}$, $C(h\boldsymbol{v}_{\boldsymbol{\theta}(\boldsymbol{p})}) = C_0(|h|/\rho)$ where $\rho = \rho_1(\boldsymbol{p})$ (resp. $\rho = \rho_2(\boldsymbol{p})$) if $\boldsymbol{v}_{\boldsymbol{\theta}(\boldsymbol{p})}$ is the first (resp. second) column of $\boldsymbol{V}_{\boldsymbol{\theta}(\boldsymbol{p})}$.
- $\rho_1(\boldsymbol{p})$ and $\rho_2(\boldsymbol{p})$ define the ranges of the covariance model C along these two directions, once multiplied by the range of the covariance model C_0 of the .

In practice these three parameters are used to describe the anisotropy at a point $\boldsymbol{p}$, and are often graphically represented by an ellipse with semi-major axis along the direction $\theta(\boldsymbol{p})$ and axis lengths $\rho_1(\boldsymbol{p})$ and $\rho_2(\boldsymbol{p})$.

The extension to the case $d = 3$ is done by considering a supplementary range parameter $\rho_3(\boldsymbol{p})$ and using a three-dimensional rotation matrix. Such matrices are characterized by the three Euler angles $\theta_1(\boldsymbol{p}), \theta_2(\boldsymbol{p}), \theta_3(\boldsymbol{p})$, which describe rotation around each one of the Cartesian coordinates axes of $\mathbb{R}^3$. In particular, the expression of the corresponding rotation matrix $\boldsymbol{V}_{\theta_1(\boldsymbol{p}),\theta_2(\boldsymbol{p}),\theta_3(\boldsymbol{p})}$ is:

$$\boldsymbol{V}_{\theta_1(\boldsymbol{p}),\theta_2(\boldsymbol{p}),\theta_3(\boldsymbol{p})} = \begin{pmatrix} \cos\theta_3(\boldsymbol{p}) & -\sin\theta_3(\boldsymbol{p}) & 0 \\ \sin\theta_3(\boldsymbol{p}) & \cos\theta_3(\boldsymbol{p}) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_2(\boldsymbol{p}) & 0 & -\sin\theta_2(\boldsymbol{p}) \\ 0 & 1 & 0 \\ \sin\theta_2(\boldsymbol{p}) & 0 & \cos\theta_2(\boldsymbol{p}) \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_1(\boldsymbol{p}) & -\sin\theta_1(\boldsymbol{p}) \\ 0 & \sin\theta_1(\boldsymbol{p}) & \cos\theta_1(\boldsymbol{p}) \end{pmatrix}$$

The local anisotropies are now characterized by ellipsoids with axes of lengths $\rho_1(\boldsymbol{p}), \rho_2(\boldsymbol{p}), \rho_3(\boldsymbol{p})$ along the directions defined by the columns of $\boldsymbol{V}_{\theta_1(\boldsymbol{p}),\theta_2(\boldsymbol{p}),\theta_3(\boldsymbol{p})}$.

A.3 Random vector

A random vector of size $n \geq 1$ is a collection of (real) n random variables defined on the same probability space. They are denoted as vectors $\boldsymbol{X} \in \mathbb{R}^n$ whose entries $X_1, \ldots, X_n$ are (univariate) random variables. A random vector is entirely defined by its probability distribution, which is the joint distribution of its entries.

The mean and the covariance matrix of a random vector $\boldsymbol{X} \in \mathbb{R}^n$ are respectively the vector $\mathbb{E}[\boldsymbol{X}] \in \mathbb{R}^n$ and the $n \times n$ matrix $\mathrm{Var}[\boldsymbol{X}]$ defined by:

$$\mathbb{E}[\boldsymbol{X}] = \begin{pmatrix} \mathbb{E}[X_1] \\ \mathbb{E}[X_2] \\ \vdots \\ \mathbb{E}[X_n] \end{pmatrix} \quad \text{and} \quad \mathrm{Var}[\boldsymbol{X}] = \begin{pmatrix} \mathrm{Cov}[X_1, X_1] & \mathrm{Cov}[X_1, X_2] & \ldots & \mathrm{Cov}[X_1, X_n] \\ \mathrm{Cov}[X_2, X_1] & \ddots & & \vdots \\ \vdots & & \ddots & \\ \mathrm{Cov}[X_n, X_1] & \mathrm{Cov}[X_n, X_2] & \ldots & \mathrm{Cov}[X_n, X_n] \end{pmatrix}$$

In particular note that the covariance matrix is by definition symmetric and satisfies

$$\mathrm{Var}[\boldsymbol{X}] = \mathbb{E}[\boldsymbol{X}\boldsymbol{X}^T] - \mathbb{E}[\boldsymbol{X}]\mathbb{E}[\boldsymbol{X}]^T$$

Also, by bilinearity of the covariance,

$$\forall \boldsymbol{c} \in \mathbb{R}^n, \quad \boldsymbol{c}^T \mathbf{Var}[\boldsymbol{X}]\boldsymbol{c} = \mathrm{Cov}[\boldsymbol{c}^T\boldsymbol{X}, \boldsymbol{c}^T\boldsymbol{X}] = \mathrm{Var}[\boldsymbol{c}^T\boldsymbol{X}] \geq 0$$

Hence, the covariance matrix of a random vector is also positive semi-definite.

Proposition A.3.1. *Let $\boldsymbol{X} \in \mathbb{R}^n$ be a random vector with mean vector $\boldsymbol{\mu}_{\boldsymbol{X}}$ and covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{X}}$ and for $k \geq 1$, let $\boldsymbol{b} \in \mathbb{R}^k$ and $\boldsymbol{C} \in \mathscr{M}_{k,n}(\mathbb{R})$.*
Then the random vector $\boldsymbol{Y} = \boldsymbol{b} + \boldsymbol{C}\boldsymbol{X}$ of size k has $\boldsymbol{\mu}_{\boldsymbol{Y}}$ and covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{Y}}$ given by

$$\boldsymbol{\mu}_{\boldsymbol{Y}} = \boldsymbol{b} + \boldsymbol{C}\boldsymbol{\mu}_{\boldsymbol{X}} \quad \text{and} \quad \boldsymbol{\Sigma}_{\boldsymbol{Y}} = \boldsymbol{C}\boldsymbol{\Sigma}_{\boldsymbol{X}}\boldsymbol{C}^T$$

Proof. This is a direct consequence of the linearity of the expectation and the bilinearity of the covariance. □

The characteristic function of a random vector $\boldsymbol{X}$ is the function $\psi_{\boldsymbol{X}} : \mathbb{R}^n \to \mathbb{C}$ defined by:

$$\psi_{\boldsymbol{X}}(\boldsymbol{t}) = \mathbb{E}\left[e^{\mathrm{i}\boldsymbol{t}^T\boldsymbol{X}}\right]$$

Characteristic functions play an important role in the study of random vector: indeed, a characteristic function determines uniquely the distribution of a random vector (and vice-versa). The following propositions follow directly from the definition of characteristic functions.

Proposition A.3.2. *Let $\boldsymbol{X} \in \mathbb{R}^n$ be a random vector with characteristic function $\psi_{\boldsymbol{X}}$. For $k \geq 1$, let $\boldsymbol{b} \in \mathbb{R}^k$ and $\boldsymbol{C} \in \mathscr{M}_{k,n}(\mathbb{R})$.*

Then the characteristic function $\psi_{\boldsymbol{Y}}$ of the random vector $\boldsymbol{Y} = \boldsymbol{b} + \boldsymbol{C}\boldsymbol{X}$ of size k is given by

$$\forall \boldsymbol{t} \in \mathbb{R}^k, \quad \psi_{\boldsymbol{Y}}(\boldsymbol{t}) = e^{\mathrm{i}\boldsymbol{t}^T\boldsymbol{b}}\psi_{\boldsymbol{X}}(\boldsymbol{C}^T\boldsymbol{t})$$

Proposition A.3.3. *Let $\boldsymbol{X}_1 \in \mathbb{R}^n$ and $\boldsymbol{X}_2 \in \mathbb{R}^k$ be two independent random vectors.*
Then the characteristic function $\psi_{\boldsymbol{Y}}$ of $\boldsymbol{Y} = (\boldsymbol{X}_1^T \quad \boldsymbol{X}_2^T)^T \in \mathbb{R}^{n+k}$ satisfies:

$$\forall \boldsymbol{t} \in \mathbb{R}^{n+k}, \quad \psi_{\boldsymbol{Y}}(\boldsymbol{t}) = \psi_{\boldsymbol{X}_1}(\boldsymbol{t}_1)\psi_{\boldsymbol{X}_2}(\boldsymbol{t}_2)$$

where $\boldsymbol{t}$ was decomposed as $\boldsymbol{t} = (\boldsymbol{t}_1^T \quad \boldsymbol{t}_2^T)^T$ with $\boldsymbol{t}_1 \in \mathbb{R}^n, \boldsymbol{t}_2 \in \mathbb{R}^k$; and for $j \in \{1, 2\}$, $\psi_{\boldsymbol{X}_j}$ is the characteristic function of $\boldsymbol{X}_j$.

Proof. Simply note that $\psi_{\boldsymbol{Y}}(\boldsymbol{t}) = \mathbb{E}[e^{\mathrm{i}(\boldsymbol{t}_1^T X_1 + \boldsymbol{t}_2^T \boldsymbol{X}_2)}] = \mathbb{E}[e^{\mathrm{i}\boldsymbol{t}_1^T\boldsymbol{X}_1}] \cdot \mathbb{E}[e^{\mathrm{i}\boldsymbol{t}_2^T\boldsymbol{X}_2}]$ since $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ are independent. □

The result below follows immediately (by induction).

Corollary A.3.4. *Let $\boldsymbol{X} \in \mathbb{R}^n$ be a random vector whose entries $X_1, \ldots, X_n$ are (pairwise) independent random variables.*
Then the characteristic function $\psi_{\boldsymbol{X}}$ of $\boldsymbol{X}$ satisfies:

$$\forall \boldsymbol{t} \in \mathbb{R}^n, \quad \psi_{\boldsymbol{X}}(\boldsymbol{t}) = \prod_{j=1}^{n} \psi_{X_j}(t_j)$$

where ψ_{X_j} is the (univariate) characteristic function of the random variable X_j.

Example A.3.1. Let $\boldsymbol{W} \in \mathbb{R}^n$ be a random vector whose entries are n independent standard Gaussian random variables. Then $\boldsymbol{W}$ has mean $\boldsymbol{0}$, covariance matrix $\boldsymbol{I}_n$ and characteristic function $\psi_{\boldsymbol{W}}$ defined by:

$$\forall \boldsymbol{t} \in \mathbb{R}^n, \quad \psi_{\boldsymbol{W}}(\boldsymbol{t}) = \prod_{j=1}^{n} \psi_{W_j}(t_j) = \prod_{j=1}^{n} e^{-\frac{1}{2}t_j^2} = e^{-\frac{1}{2}\boldsymbol{t}^T\boldsymbol{t}}$$

Proposition A.3.5. *Let $\boldsymbol{X} \in \mathbb{R}^n$ be a random vector and let $\boldsymbol{A} \in \mathscr{M}_n(\mathbb{R})$ be a deterministic matrix.*
Then, $\mathbb{E}[\boldsymbol{A}\boldsymbol{X}] = \boldsymbol{A}\mathbb{E}[\boldsymbol{X}]$ and

$$\mathbb{E}\left[\boldsymbol{X}^T\boldsymbol{A}\boldsymbol{X}\right] = \mathrm{Trace}\left(\boldsymbol{A}\mathrm{Var}[\boldsymbol{X}]\right) + \mathbb{E}[\boldsymbol{X}]^T\boldsymbol{A}\mathbb{E}[\boldsymbol{X}]$$

Proof. The linearity of the expectation yields $\mathbb{E}\left[\boldsymbol{A}\boldsymbol{X}\right] = \boldsymbol{A}\mathbb{E}\left[\boldsymbol{X}\right]$. Then, using the properties of the trace of matrix and once again the linearity of the expectation

$$\begin{aligned}\mathbb{E}\left[\boldsymbol{X}^T\boldsymbol{A}\boldsymbol{X}\right] &= \mathbb{E}\left[\text{Trace}\left(\boldsymbol{X}^T\boldsymbol{A}\boldsymbol{X}\right)\right] = \mathbb{E}\left[\text{Trace}\left(\boldsymbol{A}\boldsymbol{X}\boldsymbol{X}^T\right)\right] = \text{Trace}\left(\boldsymbol{A}\mathbb{E}\left[\boldsymbol{X}\boldsymbol{X}^T\right]\right) \\ &= \text{Trace}(\boldsymbol{A}(\text{Var}[\boldsymbol{X}] + \mathbb{E}[\boldsymbol{X}]\mathbb{E}[\boldsymbol{X}]^T)) = \text{Trace}(\boldsymbol{A}(\text{Var}[\boldsymbol{X}]) + \text{Trace}(\boldsymbol{A}\mathbb{E}[\boldsymbol{X}]\mathbb{E}[\boldsymbol{X}]^T) \\ &= \text{Trace}(\boldsymbol{A}(\text{Var}[\boldsymbol{X}]) + (\mathbb{E}[\boldsymbol{X}]^T\boldsymbol{A}\mathbb{E}[\boldsymbol{X}])\end{aligned}$$

□

A.4 Gaussian vectors

A.4.1 Gaussian distribution

The Gaussian distribution is a univariate probability distribution defined by the following density:

$$f_{\mu;\sigma^2} = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

It is entirely defined by two parameters: its mean $\mu \in \mathbb{R}$ and its variance $\sigma^2 > 0$. If X is a random variable following a Gaussian distribution with mean μ and variance σ^2 is called a Gaussian variable and is denoted $X \sim \mathcal{N}(\mu; \sigma^2)$. In that case, $\mathbb{E}[X] = \mu$ and $\text{Var}[X] = \sigma^2$. In particular, if $X\mathcal{N}(0,1)$ we say that X is a standard Gaussian variable.

The characteristic function f_X of $X \sim \mathcal{N}(\mu; \sigma^2)$ is given by:

$$\forall t \in \mathbb{R}, \quad \psi_X(t) = \mathbb{E}\left[e^{\mathrm{i}tX}\right] = e^{\mathrm{i}\mu t - \frac{1}{2}\sigma^2 t^2}$$

and provides an alternative definition of Gaussian variables. In particular, if $\sigma^2 = 0$, this expression is still defined and actually corresponds to the characteristic function of a constant equal to μ. That is why deterministic constants can be seen as Gaussian variables with variance 0.

Note that all Gaussian variables can be derived from standard Gaussian ones. Indeed if $X \sim \mathcal{N}(0,1)$ then $Y = \mu + \sigma X \sim \mathcal{N}(\mu, \sigma^2)$. This fact can easily be proved with an argument based on characteristic functions.

Proposition A.4.1. *Let $X \sim \mathcal{N}(0,1)$. Then, $\mathbb{E}[X] = 0$, $\mathbb{E}[X^2] = 1$, $\mathbb{E}[X^3] = 0$ and $\mathbb{E}[X^4] = 3$.*

Proposition A.4.2. *Let X_1, X_2, X_3, X_4 be four independent standard Gaussian variables and let $\boldsymbol{X} = (X_1 \quad X_2 \quad X_3 \quad X_4)^T \in \mathbb{R}^4$. Then, $\forall i,j,k,l \in [\![1,4]\!]$,*

$$\mathbb{E}[X_iX_jX_kX_l] = \delta_{ij}\delta_{kl} + \delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}$$

where δ_{ij} denotes the Kronecker symbol: $\delta_{ij} = 1$ if $i = j$ and 0 otherwise.

Proof. Given that $\forall m \in [\![1,4]\!]$, $\mathbb{E}[X_m]$ and that X_1, X_2, X_3, X_4 are independent, $\mathbb{E}[X_iX_jX_kX_l]$ is zero as soon as one of the indexes $i,j,k,l \in [\![1,4]\!]$ is not repeated. Hence, for $\mathbb{E}[X_iX_jX_kX_l]$ to be non-zero, the set of indexes $\{i,j,k,l\}$ must be separable into two (disjoint) sets of two indexes having the same value in $[\![1,4]\!]$.
Note that there exists exactly three possible separations of $\{i,j,k,l\}$ into two sets of cardinal 2: $\{i,j,k,l\} = \{i,j\} \cup \{k,l\} = \{i,k\} \cup \{j,l\} = \{i,l\} \cup \{j,k\}$. We therefore require that, given one of these partitions, the indexes in each subset be equal. Let m and m' be the common value of the indexes of each subset. Then two cases arise:

- if $m = m'$, i.e. if $i = j = k = l = m$, then $\mathbb{E}[X_iX_jX_kX_l] = \mathbb{E}[X_m^4] = 3$, as the fourth moment of a standard Gaussian variable.
- if $m \neq m'$, i.e. if either $(i = j = m)\&(k = l = m')$ or $(i = k = m)\&(j = l = m')$ or $(i = l = m)\&(j = k = m')$, then $\mathbb{E}[X_iX_jX_kX_l] = \mathbb{E}[X_m^2X_{m'}^2] = \mathbb{E}[X_m^2]\mathbb{E}[X_{m'}^2] = 1$.

Using Kronecker symbols to enforce these cases then gives the result. □

The Gaussian distribution plays an important role in the study of natural phenomena thanks to the central limit theorem.

Theorem A.4.3 (Central Limit Theorem)**.** *Let $Z_1, \dots, Z_n$ be a sequence of independent and identically distributed random variables with mean μ and variance σ^2. Let S_n denote the sample mean of this sequence:*

$$S_n = \frac{1}{n}\sum_{k=1}^{n} Z_k$$

Then, the random variable $\sqrt{n}(S_n - \mu)$ converges in probability towards $\mathcal{N}(0, \sigma^2)$ as $n \to \infty$.

Finally, of particular interest for statistical tests is the chi-square distribution which is defined as the distribution of the sum of the square of independent standard Gaussian variables. Namely, the chi-square distribution with $n \geq 1$ degrees of freedom, denoted $\chi^2(n)$, is the probability distribution the the variable Q_n defined by

$$Q_n = \sum_{k=1}^{n} X_k^2$$

where $X_1, \dots, X_n$ is a sequence of independent standard Gaussian variables.

A.4.2 Gaussian vectors

We call Gaussian vector any random vector $\boldsymbol{X} \in \mathbb{R}^n$ such that for some $r \in [\![1, n]\!]$ there exists a $n \times r$ real matrix $\boldsymbol{C}_r$ of rank r, and a (deterministic) vector $\boldsymbol{\mu} \in \mathbb{R}^n$ such that:

$$\boldsymbol{X} = \boldsymbol{\mu} + \boldsymbol{C}_r \boldsymbol{W}_r \tag{A.10}$$

where $\boldsymbol{W}_r \in \mathbb{R}^r$ is a random vector with r independent standard Gaussian variables. The mean of a Gaussian vector defined by Equation (A.10) is $\boldsymbol{\mu}$ and its covariance matrix is $\boldsymbol{\Sigma} = \boldsymbol{C}_r \boldsymbol{C}_r^T$, which is a symmetric and positive semi-definite matrix of rank r.

Gaussian vectors can be alternatively defined by their characteristic function.

Theorem A.4.4. *$\boldsymbol{X} \in \mathbb{R}^n$ is a Gaussian vector if and only if there exists a vector $\boldsymbol{\mu} \in \mathbb{R}^n$ and a positive semi-definite matrix $\boldsymbol{\Sigma} \in \mathscr{M}_n(\mathbb{R})$ such that the characteristic function of $\boldsymbol{X}$ is given by*

$$\psi_{\boldsymbol{X}}(\boldsymbol{t}) = \mathbb{E}\left[e^{\mathrm{i}\boldsymbol{t}^T \boldsymbol{X}}\right] = e^{\mathrm{i}\boldsymbol{t}^T \boldsymbol{\mu} - \boldsymbol{t}^T \boldsymbol{\Sigma} \boldsymbol{t}} \tag{A.11}$$

In particular, $\boldsymbol{\mu}$ is the mean vector of $\boldsymbol{X}$ and $\boldsymbol{\Sigma}$ its covariance matrix.

Proof. Assume that $\boldsymbol{X}$ is a Gaussian vector. Then Equation (A.11) follows directly from Corollary A.3.4 and proposition A.3.2 and Example A.3.1.
Conversely, assume $\boldsymbol{X}$ is a random vector with characteristic function defined by Equation (A.11). $\boldsymbol{\Sigma} \geq 0$ is symmetric, and therefore diagonalizable in an orthonormal basis $\boldsymbol{v}_1, \dots, \boldsymbol{v}_n$ associated with eigenvalues $\lambda_1, \dots, \lambda_n$. Let $r \leq n$ be the rank of $\boldsymbol{\Sigma}$, we can assume that without loss of generality, $\lambda_1 > 0, \dots, \lambda_r > 0$ and the remaining eigenvalues (if their are any) are zero.
Denoting

$$\boldsymbol{V} = [\boldsymbol{v}_1 | \dots | \boldsymbol{v}_n] = \begin{pmatrix} \boldsymbol{V}_{r,r} & \boldsymbol{V}_{r,\bar{r}} \\ \boldsymbol{V}_{\bar{r},r} & \boldsymbol{V}_{\bar{r},\bar{r}} \end{pmatrix}$$

and $\boldsymbol{\Lambda}_r = \mathrm{Diag}(\lambda_1, \dots, \lambda_r)$ we have

$$\boldsymbol{\Sigma} = \boldsymbol{V} \begin{pmatrix} \boldsymbol{\Lambda}_r & \\ & \boldsymbol{0}_{n-r} \end{pmatrix} \boldsymbol{V}^T = \begin{pmatrix} \boldsymbol{V}_{r,r} \\ \boldsymbol{V}_{\bar{r},r} \end{pmatrix} \boldsymbol{\Lambda}_r \begin{pmatrix} \boldsymbol{V}_{r,r} \\ \boldsymbol{V}_{\bar{r},r} \end{pmatrix}^T = \boldsymbol{V}_{n,r} \boldsymbol{\Lambda}_r \boldsymbol{V}_{n,r}^T$$

where $\boldsymbol{V}_{n,r}$ is the matrix containing the r-first columns of $\boldsymbol{V}$ and $\boldsymbol{V}_{\bar{r},r}$ is the matrix containing its remaining columns. In particular, $\boldsymbol{V}_{n,r}^T \boldsymbol{V}_{n,r} = \boldsymbol{I}_r$ and that $\boldsymbol{V}_{n,r}^T \boldsymbol{V}_{n,r} = \boldsymbol{0}$.

Let $\tilde{\boldsymbol{X}}_r \in \mathbb{R}^r$ and $\tilde{\boldsymbol{X}}_{\bar{r}} \in \mathbb{R}^{n-r}$ be the random vectors defined by $\tilde{\boldsymbol{X}}_r = \boldsymbol{V}_{n,r}^T(\boldsymbol{X} - \boldsymbol{\mu})$ and $\tilde{\boldsymbol{X}}_{\bar{r}} = \boldsymbol{V}_{n,\bar{r}}^T(\boldsymbol{X} - \boldsymbol{\mu})$. Note that

$$\boldsymbol{X} - \boldsymbol{\mu} = \boldsymbol{V}\boldsymbol{V}^T(\boldsymbol{X} - \boldsymbol{\mu}) = \boldsymbol{V}\begin{pmatrix}\tilde{\boldsymbol{X}}_r \\ \tilde{\boldsymbol{X}}_{\bar{r}}\end{pmatrix} = \boldsymbol{V}_{n,r}\tilde{\boldsymbol{X}}_r + \boldsymbol{V}_{n,\bar{r}}\tilde{\boldsymbol{X}}_{\bar{r}}$$

On one hand, following Corollary A.3.4 and proposition A.3.2, the characteristic function of $\tilde{\boldsymbol{X}}_r$ is given by

$$\psi_{\tilde{\boldsymbol{X}}_r}(\boldsymbol{t}) = e^{-\frac{1}{2}\boldsymbol{t}^T\boldsymbol{V}_{n,r}^T\boldsymbol{\Sigma}\boldsymbol{V}_{n,r}\boldsymbol{t}} = e^{-\frac{1}{2}\boldsymbol{t}^T\boldsymbol{\Lambda}_r\boldsymbol{t}}$$

which is the characteristic function of a random vector with r independent zero-mean Gaussian entries with variances $\lambda_1, \dots, \lambda_r$. Hence, we can write $\tilde{\boldsymbol{X}}_r = \mathrm{Diag}(\sqrt{\lambda_1}, \dots, \sqrt{\lambda_r})\boldsymbol{W}_r$ where $\boldsymbol{W}_r$ is a random vector with r independent standard Gaussian entries.

On the other hand, the characteristic function of $\tilde{\boldsymbol{X}}_{\bar{r}}$ is given by

$$\psi_{\tilde{\boldsymbol{X}}_{\bar{r}}}(\boldsymbol{t}) = e^{-\frac{1}{2}\boldsymbol{t}^T\boldsymbol{V}_{\bar{r},r}^T\boldsymbol{\Sigma}\boldsymbol{V}_{\bar{r},r}\boldsymbol{t}} = e^{-\frac{1}{2}\cdot 0} = 1$$

which is the characteristic function of a random vector with r constant entries equal to zero. Hence, we can write $\tilde{\boldsymbol{X}}_r = \boldsymbol{0}$.
Consequently, we have:

$$\boldsymbol{X} = \boldsymbol{\mu} + \boldsymbol{V}_{n,r}\mathrm{Diag}(\sqrt{\lambda_1}, \dots, \sqrt{\lambda_r})\boldsymbol{W}_r$$

which proves that $\boldsymbol{X}$ is a Gaussian vector as $\boldsymbol{V}_{n,r}$ is of rank r. Note in particular that the mean of $\boldsymbol{X}$ therefore is $\mathbb{E}[\boldsymbol{X}] = \boldsymbol{\mu}$ and its covariance matrix is

$$\mathrm{Var}[\boldsymbol{X}] = \boldsymbol{V}_{n,r}\mathrm{Diag}(\sqrt{\lambda_1}, \dots, \sqrt{\lambda_r})(\boldsymbol{V}_{n,r}\mathrm{Diag}(\sqrt{\lambda_1}, \dots, \sqrt{\lambda_r}))^T = \boldsymbol{\Sigma}$$

□

If $\boldsymbol{X}$ is a Gaussian vector of size n with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma} \geq 0$, we denote $\boldsymbol{X} \sim \mathcal{N}_n(\boldsymbol{\mu}; \boldsymbol{\Sigma})$. Note indeed that a mean vector $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{\Sigma}$ are sufficient to characterize a Gaussian vector as they determine its characteristic function.

A Gaussian vector is called non-singular if $r = n$, i.e. if $\boldsymbol{C}_r = \boldsymbol{C}_n$ (or equivalently $\boldsymbol{\Sigma}$) is an invertible $n \times n$ matrix. Otherwise, a Gaussian vector is called singular. Non-singular Gaussian vectors follow a non-degenerate multivariate distribution of mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma} > 0$, which is defined by the following density function:

$$f_{\boldsymbol{\mu};\boldsymbol{\Sigma}}(\boldsymbol{x}) = \frac{1}{(2\pi)^{n/2}|\boldsymbol{\Sigma}|^{1/2}}\exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})\boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu})\right)$$

Another defining property of Gaussian vectors is given in the following proposition.

Proposition A.4.5. *Let $\mu \in \mathbb{R}$ and $\boldsymbol{\Sigma} \in \mathscr{M}_n(\mathbb{R})$ be a symmetric and positive semi-definite matrix.*
$\boldsymbol{X} \in \mathbb{R}^n$ is a Gaussian vector with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ if and only if $\forall \boldsymbol{c} \in \mathbb{R}^n$, $\boldsymbol{c}^T\boldsymbol{X}$ is a Gaussian variable with mean $\boldsymbol{c}^T\boldsymbol{\mu}$ and variance $\boldsymbol{c}^T\boldsymbol{\Sigma}\boldsymbol{c}$.

Proof. Simply notice that $\forall t \in \mathbb{R}$, $\forall c \in \mathbb{R}^n$, $\psi_{\boldsymbol{c}^T\boldsymbol{X}}(t) = \mathbb{E}[e^{\mathrm{i}t\boldsymbol{c}^T\boldsymbol{X}}] = \psi_{\boldsymbol{X}}(t\boldsymbol{c})$ and identify the characteristic functions to conclude. □

Gaussian vectors stay Gaussian after a linear transform.

Proposition A.4.6. *Let $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\mu}; \boldsymbol{\Sigma})$. Let $\boldsymbol{A} \in \mathscr{M}_{k,n}(\mathbb{R})$ and $\boldsymbol{b} \in \mathbb{R}^k$. Then,*

$$[\boldsymbol{b} + \boldsymbol{A}\boldsymbol{X}] \sim \mathcal{N}(\boldsymbol{b} + \boldsymbol{A}\boldsymbol{\mu}; \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^T)$$

Proof. Direct consequence of Proposition A.3.2. □

The concatenation of two independent Gaussian vectors is also a Gaussian vector.

Proposition A.4.7. *Let $\boldsymbol{X}_1 \in \mathbb{R}^n$ and $\boldsymbol{X}_2 \in \mathbb{R}^k$ be two independent Gaussian vectors such that $\boldsymbol{X}_1 \sim \mathcal{N}(\boldsymbol{\mu}_1; \boldsymbol{\Sigma}_1)$ and $\boldsymbol{X}_2 \sim \mathcal{N}(\boldsymbol{\mu}_2; \boldsymbol{\Sigma}_2)$. Then,*

$$\begin{pmatrix} \boldsymbol{X}_1 \\ \boldsymbol{X}_2 \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix} ; \begin{pmatrix} \boldsymbol{\Sigma}_1 & \\ & \boldsymbol{\Sigma}_{22} \end{pmatrix} \right)$$

Proof. Direct consequence of Proposition A.3.3. □

The finite sum of independent Gaussian vectors is also Gaussian.

Proposition A.4.8. *Let $n_1, \dots, n_p \geq 1$. Let $\boldsymbol{X}_1 \in \mathbb{R}^{n_1}, \dots, \boldsymbol{X}_p \in \mathbb{R}^{n_p}$ be p independent Gaussian vectors with respective means $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_p$ and covariance matrices $\boldsymbol{\Sigma}_1, \dots, \boldsymbol{\Sigma}_p$. Then for any $k \geq 1$, any $\boldsymbol{b} \in \mathbb{R}^k$ and any matrices $\boldsymbol{A}_1 \in \mathscr{M}_{k,n_1}(\mathbb{R}), \dots, \boldsymbol{A}_p \in \mathscr{M}_{k,n_p}(\mathbb{R})$,*

$$[\boldsymbol{b} + \boldsymbol{A}_1\boldsymbol{X}_1 + \cdots + \boldsymbol{A}_p\boldsymbol{X}_p] \sim \mathcal{N}(\boldsymbol{b} + \boldsymbol{A}_1\boldsymbol{\mu}_1 + \cdots + \boldsymbol{A}_p\boldsymbol{\mu}_p;\ \boldsymbol{A}_1\boldsymbol{\Sigma}_1\boldsymbol{A}_1^T + \cdots + \boldsymbol{A}_p\boldsymbol{\Sigma}_p\boldsymbol{A}_p^T)$$

Proof. Direct consequence of Propositions A.4.6 and A.4.7 by noticing that $\boldsymbol{Y} = \boldsymbol{b} + \boldsymbol{A}_1\boldsymbol{X}_1 + \cdots + \boldsymbol{A}_p\boldsymbol{X}_p$ can be written:

$$\boldsymbol{Y} = \boldsymbol{b} + \boldsymbol{A}\boldsymbol{X} \quad \text{where} \quad \boldsymbol{A} = \begin{pmatrix} \boldsymbol{A}_1 & & \\ & \ddots & \\ & & \boldsymbol{A}_p \end{pmatrix}, \quad \boldsymbol{X} = \begin{pmatrix} \boldsymbol{X}_1 \\ \vdots \\ \boldsymbol{X}_p \end{pmatrix}$$

□

The marginal distribution of a subvector of a Gaussian vector is a multivariate Gaussian distribution.

Proposition A.4.9. *Let $\boldsymbol{X} \in \mathbb{R}^n$ such that $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\mu}; \boldsymbol{\Sigma})$ with $\boldsymbol{\Sigma} > 0$ and let $k \in [\![1, n-1]\!]$. Consider the following partition of $\boldsymbol{X}$, $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$ and $\boldsymbol{Q} = \boldsymbol{\Sigma}^{-1}$:*

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{X}_k \\ \boldsymbol{X}_{\bar{k}} \end{pmatrix}, \quad \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_k \\ \boldsymbol{\mu}_{\bar{k}} \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{k,k} & \boldsymbol{\Sigma}_{k,\bar{k}} \\ \boldsymbol{\Sigma}_{\bar{k},k} & \boldsymbol{\Sigma}_{\bar{k},\bar{k}} \end{pmatrix}, \quad \boldsymbol{Q} = \begin{pmatrix} \boldsymbol{Q}_{k,k} & \boldsymbol{Q}_{k,\bar{k}} \\ \boldsymbol{Q}_{\bar{k},k} & \boldsymbol{Q}_{\bar{k},\bar{k}} \end{pmatrix} \tag{A.12}$$

Then the marginal distribution of $\boldsymbol{X}_k$ is:

$$\boldsymbol{X}_k \sim \mathcal{N}(\boldsymbol{\mu}_k;\ \boldsymbol{\Sigma}_{k,k}) \tag{A.13}$$

where in particular, $\boldsymbol{\Sigma}_{k,k}$ is also given by $\boldsymbol{\Sigma}_{k,k} = (\boldsymbol{Q}_{k,k} - \boldsymbol{Q}_{k,\bar{k}}\boldsymbol{Q}_{\bar{k},\bar{k}}^{-1}\boldsymbol{Q}_{\bar{k},k})^{-1}$.

Proof. Consequence of Proposition A.4.6 given that $\boldsymbol{X}_k = \begin{pmatrix} \boldsymbol{I}_k & \boldsymbol{0}_{k,n-k} \end{pmatrix} \boldsymbol{X}$.

□

The conditional distribution of a subvector of a Gaussian vector given the remaining entries is a multivariate Gaussian distribution.

Proposition A.4.10. *Let $\boldsymbol{X} \in \mathbb{R}^n$ such that $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\mu}; \boldsymbol{\Sigma})$ with $\boldsymbol{\Sigma} > 0$ and let $k \in [\![1, n-1]\!]$. Consider the following partition of $\boldsymbol{X}$, $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$ and $\boldsymbol{Q} = \boldsymbol{\Sigma}^{-1}$ given by Equation* (A.12)*. Then the conditional distribution of $\boldsymbol{X}_k$ given $\boldsymbol{X}_{\bar{k}} = \boldsymbol{x}_{\bar{k}}$ for some $\boldsymbol{x}_{\bar{k}} \in \mathbb{R}^{n-k}$ is:*

$$[\boldsymbol{X}_k | \boldsymbol{X}_{\bar{k}} = \boldsymbol{x}_{\bar{k}}] \sim \mathcal{N}(\boldsymbol{\mu}_{k|\bar{k}};\ \boldsymbol{\Sigma}_{k|\bar{k}}) \tag{A.14}$$

where

$$\boldsymbol{\mu}_{k|\bar{k}} = \boldsymbol{\mu}_k + \boldsymbol{\Sigma}_{k,\bar{k}}\boldsymbol{\Sigma}_{\bar{k},\bar{k}}^{-1}(\boldsymbol{x}_{\bar{k}} - \boldsymbol{\mu}_{\bar{k}}) = \boldsymbol{\mu}_k - \boldsymbol{Q}_{k,k}^{-1}\boldsymbol{Q}_{k,\bar{k}}(\boldsymbol{x}_{\bar{k}} - \boldsymbol{\mu}_{\bar{k}}) \tag{A.15}$$

and

$$\boldsymbol{\Sigma}_{k|\bar{k}} = \boldsymbol{\Sigma}_{kk} - \boldsymbol{\Sigma}_{k,\bar{k}}\boldsymbol{\Sigma}_{\bar{k}\bar{k}}^{-1}\boldsymbol{\Sigma}_{\bar{k},k} = \boldsymbol{Q}_{k,k}^{-1} \tag{A.16}$$

Proof. See (Tong, 2012, Theorem 3.3.4). □

Remark A.4.1. Note in particular that

$$\boldsymbol{\mu}_{k|\bar{k}} = \mathbb{E}[\boldsymbol{X}_k|\boldsymbol{X}_{\bar{k}} = \boldsymbol{x}_{\bar{k}}]$$

is the conditional expectation of $\boldsymbol{X}_k$ given $\boldsymbol{X}_{\bar{k}} = \boldsymbol{x}_{\bar{k}}$ and

$$\boldsymbol{\Sigma}_{k|\bar{k}} = \mathrm{Var}[\boldsymbol{X}_k|\boldsymbol{X}_{\bar{k}} = \boldsymbol{x}_{\bar{k}}] = \mathbb{E}\left[\left(\boldsymbol{X}_k - \boldsymbol{\mu}_{k|\bar{k}}\right)\left(\boldsymbol{X}_k - \boldsymbol{\mu}_{k|\bar{k}}\right)^T \middle| \boldsymbol{X}_{\bar{k}} = \boldsymbol{x}_{\bar{k}}\right]$$

is the conditional covariance matrix of $\boldsymbol{X}_k$ given $\boldsymbol{X}_{\bar{k}} = \boldsymbol{x}_{\bar{k}}$. Actually, the value of the latter does not depend on the conditioning data $\boldsymbol{x}_{\bar{k}}$.

Quadratic form evaluated on Gaussian vectors are directly linked to the trace function.

Proposition A.4.11. *Let $\boldsymbol{A}$ and $\boldsymbol{B}$ be two symmetric matrices of size $d \geq 1$ and let $\boldsymbol{W} \in \mathbb{R}^d$ be a random vector composed of d independent and identically distributed standard Gaussian variables.*
Then, $\mathbb{E}[\boldsymbol{W}^T\boldsymbol{A}\boldsymbol{W}] = \mathrm{Trace}(\boldsymbol{A})$, $\mathbb{E}[\boldsymbol{W}^T\boldsymbol{B}\boldsymbol{W}] = \mathrm{Trace}(\boldsymbol{B})$ and

$$\mathrm{Cov}[\boldsymbol{W}^T\boldsymbol{A}\boldsymbol{W}, \boldsymbol{W}^T\boldsymbol{B}\boldsymbol{W}] = 2\mathrm{Trace}(\boldsymbol{A}\boldsymbol{B})$$

Proof. Note that by definition of $\boldsymbol{W}$, $\mathbb{E}[\boldsymbol{W}] = \boldsymbol{0}$ and $\mathrm{Var}[\boldsymbol{W}] = \mathbb{E}[\boldsymbol{W}\boldsymbol{W}^T] = \boldsymbol{I}$. Proposition A.3.5 therefore yields $\mathbb{E}[\boldsymbol{W}^T\boldsymbol{A}\boldsymbol{W}] = \mathrm{Trace}(\boldsymbol{A})$ and $\mathbb{E}[\boldsymbol{W}^T\boldsymbol{B}\boldsymbol{W}] = \mathrm{Trace}(\boldsymbol{B})$.
Consequently, note that

$$\begin{aligned}\mathrm{Cov}[\boldsymbol{W}^T\boldsymbol{A}\boldsymbol{W}, \boldsymbol{W}^T\boldsymbol{B}\boldsymbol{W}] &= \mathbb{E}[\boldsymbol{W}^T\boldsymbol{A}\boldsymbol{W} \cdot \boldsymbol{W}^T\boldsymbol{B}\boldsymbol{W}] - \mathbb{E}[\boldsymbol{W}^T\boldsymbol{A}\boldsymbol{W}] \cdot \mathbb{E}[\boldsymbol{W}^T\boldsymbol{B}\boldsymbol{W}] \\ &= \mathbb{E}[\boldsymbol{W}^T\boldsymbol{A}\boldsymbol{W} \cdot \boldsymbol{W}^T\boldsymbol{B}\boldsymbol{W}] - \mathrm{Trace}(\boldsymbol{A}) \cdot \mathrm{Trace}(\boldsymbol{B})\end{aligned}$$

And, $\mathbb{E}[\boldsymbol{W}^T\boldsymbol{A}\boldsymbol{W} \cdot \boldsymbol{W}^T\boldsymbol{B}\boldsymbol{W}] = \mathbb{E}[\mathrm{Trace}(\boldsymbol{W}^T\boldsymbol{A}\boldsymbol{W} \cdot \boldsymbol{W}^T\boldsymbol{B}\boldsymbol{W})] = \mathbb{E}[\mathrm{Trace}(\boldsymbol{A}\boldsymbol{V}_{\boldsymbol{W}}\boldsymbol{B}\boldsymbol{V}_{\boldsymbol{W}})]$ where $\boldsymbol{V}_{\boldsymbol{W}} = \boldsymbol{W}\boldsymbol{W}^T$. Developing this last expression gives,

$$\mathbb{E}[\boldsymbol{W}^T\boldsymbol{A}\boldsymbol{W} \cdot \boldsymbol{W}^T\boldsymbol{B}\boldsymbol{W}] = \sum_{i=1}^{d}\sum_{j=1}^{d}\sum_{k=1}^{d}\sum_{l=1}^{d} A_{ik}B_{jl}\mathbb{E}[W_iW_jW_kW_l]$$

where, following Proposition A.4.2, $\mathbb{E}[W_iW_jW_kW_l] = \delta_{ij}\delta_{kl} + \delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}$. Hence, by switching some of sums,

$$\begin{aligned}\mathbb{E}[\boldsymbol{W}^T\boldsymbol{A}\boldsymbol{W} \cdot \boldsymbol{W}^T\boldsymbol{B}\boldsymbol{W}] = &\sum_{i=1}^{d}\sum_{j=1}^{d}\delta_{ij}\sum_{k=1}^{d}A_{ik}\sum_{l=1}^{d}B_{jl}\delta_{kl} + \sum_{i=1}^{d}\sum_{k=1}^{d}A_{ik}\delta_{ik}\sum_{j=1}^{d}\sum_{l=1}^{d}B_{jl}\delta_{jl} \\ &+ \sum_{l=1}^{d}\sum_{i=1}^{d}\delta_{il}\sum_{k=1}^{d}A_{ik}\sum_{j=1}^{d}B_{jl}\delta_{jk}\end{aligned}$$

And finally,

$$\begin{aligned}\mathbb{E}[\boldsymbol{W}^T\boldsymbol{A}\boldsymbol{W} \cdot \boldsymbol{W}^T\boldsymbol{B}\boldsymbol{W}] &= \sum_{i=1}^{d}\sum_{k=1}^{d}A_{ik}B_{ik} + \sum_{i=1}^{d}A_{ii}\sum_{j=1}^{d}B_{jj} + \sum_{l=1}^{d}\sum_{k=1}^{d}A_{lk}B_{kl} \\ &= \mathrm{Trace}(\boldsymbol{A}\boldsymbol{B}^T) + \mathrm{Trace}(\boldsymbol{A})\mathrm{Trace}(\boldsymbol{B}) + \mathrm{Trace}(\boldsymbol{A}\boldsymbol{B}) \\ &= 2\mathrm{Trace}(\boldsymbol{A}\boldsymbol{B}) + \mathrm{Trace}(\boldsymbol{A})\mathrm{Trace}(\boldsymbol{B})\end{aligned}$$

Therefore, $\mathrm{Cov}[\boldsymbol{W}^T\boldsymbol{A}\boldsymbol{W}, \boldsymbol{W}^T\boldsymbol{B}\boldsymbol{W}] = 2\mathrm{Trace}(\boldsymbol{A}\boldsymbol{B})$. □

A.5 Multivariate Fourier series and transform

Let $g(\boldsymbol{x})$ be a 2π-periodic function of $\mathbb{R}^d$, i.e. g is 2π-periodic with respect to each variable $x_1, \ldots, x_d$ and suppose that $g \in L^2([-\pi, \pi]^d)$. Then g can be represented as the limit on L^2 of its Fourier series $\mathcal{S}_f[g]$ defined by (Osborne, 2010):

$$g \overset{L^2([-\pi,\pi]^d)}{=} \mathcal{S}_F[g](\boldsymbol{x}) = \sum_{\boldsymbol{j} \in \mathbb{Z}^d} c_{\boldsymbol{j}}(g) e^{i \boldsymbol{j}^T \boldsymbol{x}}$$

where $\boldsymbol{j} = (j_1 \ldots j_d) \in \mathbb{Z}^d$ is a multi-index. The coefficients $c_{\boldsymbol{j}}(g)$ in this expression are given by:

$$c_{\boldsymbol{j}}(g) = \frac{1}{(2\pi)^d} \int_{[-\pi,\pi]^d} e^{-i \boldsymbol{j}^T \boldsymbol{x}} g(\boldsymbol{x}) d\boldsymbol{x}, \quad \boldsymbol{j} \in \mathbb{Z}^d$$

The Fourier transform of g is then defined from its Fourier series as a train of impulses (corresponding to the Fourier transform of each term of Fourier series):

$$\mathscr{F}[g] := \mathscr{F}\left[\mathcal{S}_F[g]\right] = (2\pi)^d \sum_{\boldsymbol{j} \in \mathbb{Z}^d} c_{\boldsymbol{j}}(g) \delta_{\boldsymbol{j}} \tag{A.17}$$

where $\mathscr{F}$ denotes the Fourier transform operator (as defined on $\mathbb{R}^d$) and $\delta_{\boldsymbol{j}}$ denotes the Dirac function at $\boldsymbol{j}$ and is a distribution with test function space $\mathscr{C}_c^\infty(\mathbb{R}^d)$, the set of compactly supported smooth functions of $\mathbb{R}^d$. In particular, $\forall \boldsymbol{j} \in \mathbb{Z}^d$,

$$\forall u \in \mathscr{C}_c^\infty(\mathbb{R}^d), \quad \delta_{\boldsymbol{j}}(u) = u(\boldsymbol{j})$$

Hence, the Fourier transform of g is a distribution whose action on functions of $\mathscr{C}_c^\infty(\mathbb{R}^d)$ is given by

$$\forall u \in \mathscr{C}_c^\infty(\mathbb{R}^d), \quad \mathscr{F}[g](u) = (2\pi)^d \sum_{\boldsymbol{j} \in \mathbb{Z}^d} c_{\boldsymbol{j}}(g) u(\boldsymbol{j})$$

Note that the inverse Fourier transform of a distribution $\widehat{T}$ is well-defined as the distribution $\mathscr{F}^{-1}\left[\widehat{T}\right]$ acting on functions of $\mathscr{C}_c^\infty(\mathbb{R}^d)$ by

$$\forall u \in \mathscr{C}_c^\infty(\mathbb{R}^d), \quad \mathscr{F}^{-1}\left[\widehat{T}\right](u) = \widehat{T}(\mathscr{F}^{-1}[u])$$

Note in particular that for g defined as above we have

$$\mathscr{F}^{-1}\left[\mathscr{F}[g]\right] = \mathcal{S}_F[g]$$

where the equality stands in the sense of distributions, i.e.

$$\forall u \in \mathscr{C}_c^\infty(\mathbb{R}^d), \quad \mathscr{F}^{-1}\left[\mathscr{F}[g]\right](u) = \int_{\mathbb{R}^d} \mathcal{S}_F[g](\boldsymbol{x}) u(\boldsymbol{x}) d\boldsymbol{x}$$

B

Interpolation and approximation of functions

We recall in this appendix some basic definitions and theorems regarding the interpolation and the approximations of real-valued functions over a closed interval of $\mathbb{R}$. We refer the reader to (Atkinson, 1989; Mason and Handscomb, 2002; Trefethen, 2013) for a complete overview of the subject.

Throughout this appendix, h will denote a real function defined on interval $[a, b] \subset \mathbb{R}$ and $m \in \mathbb{N}$. $\mathscr{P}_m$ will denote the set of polynomials with real coefficients and with degree at most m.

B.1 Interpolation of functions

The polynomial interpolation problem consists in building a polynomial finding a polynomial of $\mathscr{P}_m$ that interpolates h over a set of $m+1$ distinct points $t_0, \ldots, t_m$ sampled from $[a, b]$. Such a polynomial is called an *interpolant of degree* m of h and is denoted P_h. This problem can be formulated as:

$$\text{Find } P_h \in \mathscr{P}_m \text{ such that } \forall j \in [\![0, m]\!], \quad P_h(t_j) = h(t_j) \quad . \tag{B.1}$$

For a given set of distinct interpolation points $t_0, \ldots, t_m$ this problem has a unique solution, according to the unisolvence theorem (Atkinson, 1989, Theorem 3.1). This solution can be retrieved using either one of the following approaches (Atkinson, 1989):

- *Vandermonde approach.* P_h is written as

$$P_h(t) = \sum_{k=0}^{m} a_k t^k, \quad t \in \mathbb{R} \quad ,$$

  where the coefficients $a_0, \ldots, a_m$ are the solution of the linear system obtained by enforcing the $(m+1)$ equalities of (B.1). This last problem can be formulated using the Vandermonde matrix:

$$\begin{pmatrix} 1 & t_0 & \ldots & t_0^m \\ 1 & t_1 & \ldots & t_1^m \\ \vdots & \vdots & & \vdots \\ 1 & t_m & \ldots & t_m^m \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{pmatrix} = \begin{pmatrix} h(t_0) \\ h(t_1) \\ \vdots \\ h(t_m) \end{pmatrix} \quad . \tag{B.2}$$

- *Newton approach.* P_h is written as:

$$P_h(t) = \sum_{k=0}^{m} b_k \eta_k(t), \quad t \in \mathbb{R} \quad ,$$

where η_k denotes the k-th Newton polynomial, defined for $k \in [\![0, m]\!]$ by

$$\eta_k(t) = \begin{cases} 1 & \text{if } k = 0 \\ \prod_{j=0}^{k-1}(t - t_j) & \text{if } 1 \leq k \leq m \end{cases} \quad .$$

By once again enforcing the $(m+1)$ equalities of (B.1), the coefficients $b_0, \ldots, b_m$ are the solution of the following *(lower-)triangular* system:

$$\begin{pmatrix} \eta_0(t_0) & & & & \\ \eta_0(t_1) & \eta_1(t_1) & & & \\ \eta_0(t_2) & \eta_1(t_2) & \eta_2(t_2) & & \\ \vdots & \vdots & \vdots & \ddots & \\ \eta_0(t_m) & \eta_1(t_m) & \eta_2(t_m) & \ldots & \eta_m(t_m) \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_m \end{pmatrix} = \begin{pmatrix} h(t_0) \\ h(t_1) \\ h(t_2) \\ \vdots \\ h(t_n) \end{pmatrix} \quad . \tag{B.3}$$

- *Lagrange approach.* P_h is directly written as:

$$P_h(t) = \sum_{k=0}^{m} h(t_k) l_k(t), \quad t \in \mathbb{R} \quad , \tag{B.4}$$

where l_k denotes the k-th Lagrange polynomial, defined for $k \in [\![0, m]\!]$ by

$$l_k(t) = \prod_{\substack{j=0 \\ j \neq k}}^{m} \frac{t - t_j}{t_k - t_j}, \quad 0 \leq k \leq m \quad .$$

In the subsequent, we turn to the problem of approximating a function over a closed interval, building partly from the interpolation results introduced above.

B.2 Approximation theory

Let $h : [a, b] \to \mathbb{R}$ be defined over a subset $[a, b] \subset \mathbb{R}$. Unless otherwise specified, h is assumed to be continuous over $[a, b]$. The goal is to find a polynomial P_h of fixed degree (at most) m that approximates h over $[a, b]$. Formally, we seek a polynomial of degree at most m such that it has a "small" approximation error, defined as the infinite norm over $[a, b]$ of the difference $h - P_h$, i.e.:

$$\|h - P_h\|_\infty = \max_{t \in [a,b]} |h(x) - P_h(x)| \quad .$$

In particular the polynomial that minimizes this last norm, denoted P_h^*, exists and is unique (Atkinson, 1989, Theorem 4.10):

$$P_h^* = \operatorname*{argmin}_{P \in \mathscr{P}_m} \|h - P\|_\infty \quad . \tag{B.5}$$

Computing P_h^* is in general a difficult task. In practice, a near-optimal polynomial is preferred to P_h, i.e. a polynomial whose approximation error is close to that of P_h^*.

Let $\mathcal{A}_m$ denote a linear operator from $\mathscr{C}([a, b])$ to $\mathscr{P}_m$ that associates to any $h \in \mathscr{C}([a, b])$ a polynomial $P_h = \mathcal{A}_m[h] \in \mathscr{P}_m$, and such that $h \in \mathscr{P}_m \Rightarrow \mathcal{A}_m[h] = h$ (hence, $\mathcal{A}_m$ is a projector on $\mathscr{P}_m$). In particular, we think of $\mathcal{A}_m[h]$ as a polynomial approximation of degree at most m of h.

The accuracy of the polynomial approximations delivered by $\mathcal{A}_m$ are assessed by its Lebesgue constant $\Lambda(\mathcal{A}_m)$, which is its operator norm of $\mathcal{A}_m$ with respect to the uniform norm:

$$\Lambda(\mathcal{A}_m) = \sup_{h \in \mathscr{C}([a,b])} \frac{\|\mathcal{A}_m[h]\|_\infty}{\|h\|_\infty} \quad .$$

The Lebesgue constant therefore quantifies the discrepancy between the magnitude of the variations of an arbitrary continuous function and its approximation by $\mathcal{A}_m$.

In particular, the Lebesgue constant links the approximation error of the polynomial derived from $\mathcal{A}_m$ to the approximation error of the best approximation of h in $\mathscr{P}_m$, as defined in Equation (B.5) (Trefethen, 2013, Theorem 15.1):

$$\forall h \in \mathscr{C}([a,b]), \quad \|h - P_h^*\|_\infty \leq \|h - \mathcal{A}_m[h]\|_\infty \leq (1 + \Lambda(\mathcal{A}_m)) \|h - P_h^*\|_\infty \quad .$$

The Lebesgue constant quantifies how much the error of the approximation $\mathcal{A}_m[h]$ of an arbitrary continuous function h can be far from the minimal error achievable by a polynomial of same degree. In the remainder of this section several approaches to compute a polynomial approximation of a continuous function over a segment are considered, and arguments based on their Lebesgue constants are provided to compare them.

For sake of simplicity, in the remainder of this section, we consider that the interval $[a,b]$ on which the function h is approximated is $[-1,1]$. More general intervals $[a,b]$ can be retrieved by using the (bijective) linear mapping $\phi_{a,b}$ from $[a,b]$ to $[-1,1]$:

$$\phi_{a,b} : t \in [a,b] \mapsto \frac{2}{b-a}(t-a) - 1 \in [-1,1] \quad ,$$

whose inverse is the linear mapping $\phi_{a,b}^{-1}$ from $[-1,1]$ to $[a,b]$ given by:

$$\phi_{a,b}^{-1} : t \in [-1,1] \mapsto a + \frac{b-a}{2}(t+1) \in [a,b] \quad .$$

Hence, to approximate a function h over $[a,b]$, one can find an approximation $\hat{h}$ of the function $h \circ \phi_{a,b}^{-1}$ over $[-1,1]$ and return the function $\hat{h} \circ \phi_{a,b}$.

B.3 Approximation by interpolation

A first approach to obtain a polynomial approximation of $h \in \mathscr{C}([-1,1])$ consists in building an interpolant of h of degree m over a predefined set of $m+1$ (distinct) points $t_0, \ldots, t_m$ sampled from $[-1,1]$.

Choosing the right sampling is instrumental to the quality of approximation of the resulting interpolant. For instance, regularly spaced points over $[a,b]$ are known to be a poor choice as the resulting interpolant may lead to high approximation errors, even for smooth functions. This is formalized in the following theorem.

Theorem B.3.1 (Lebesgue constant for interpolation at equispaced points)**.** *Let $\mathcal{Q}_m$ denote the linear operator that associates to any $h \in \mathscr{C}([-1,1])$ the polynomial $\mathcal{Q}_m[h] \in \mathscr{P}_m$ that interpolates h at $m+1$ equispaced points in $[-1,1]$.*
The Lebesgue constant $\Lambda(\mathcal{Q}_m)$ satisfies:

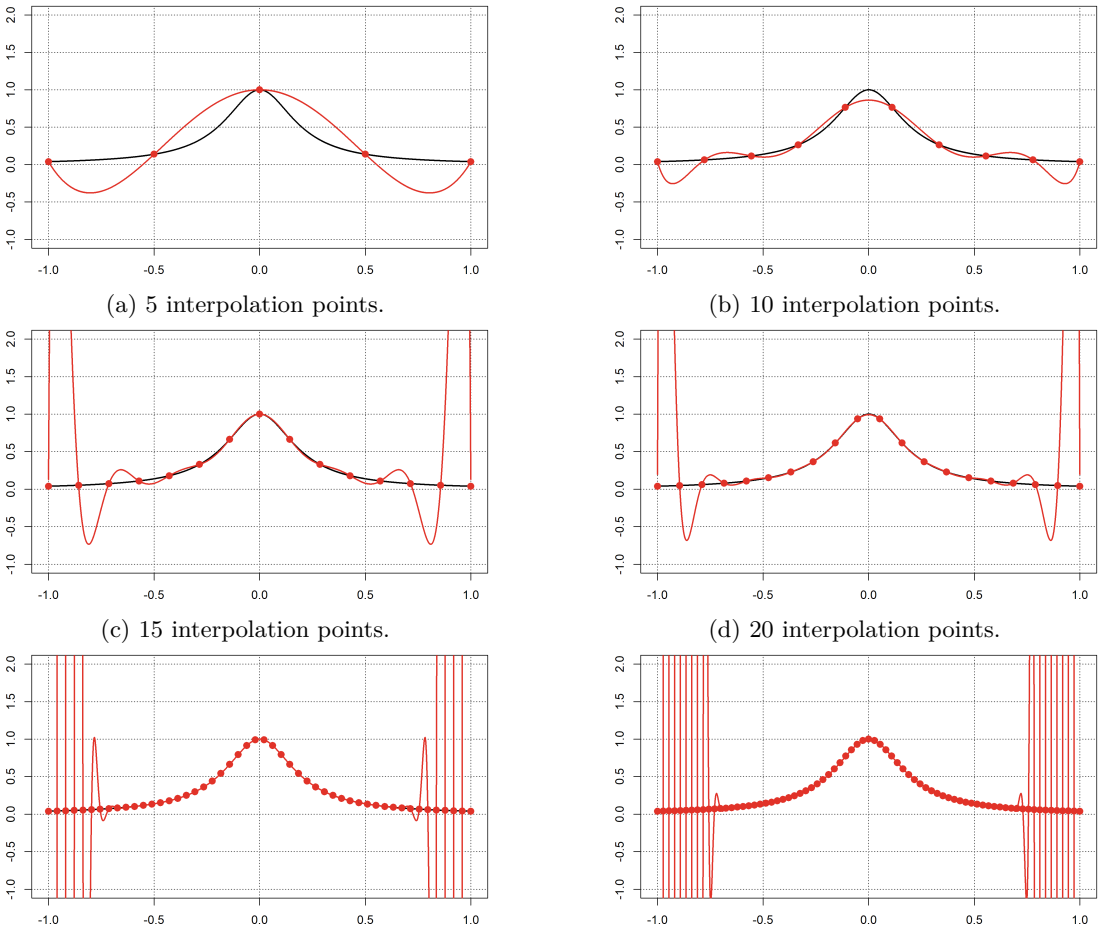
$$\Lambda(\mathcal{Q}_m) > \frac{2^{m-2}}{m^2} \quad \text{and} \quad \Lambda(\mathcal{Q}_m) \underset{m\to\infty}{\sim} \frac{2^{m+1}}{em\log m} \quad , \tag{B.6}$$

where $e = \exp(1)$ denotes Euler's number.

> *Proof.* The inequality was derived by Trefethen and Weideman (1991) and the asymptotic equivalence is a result independently discovered by Turetskii (1940) and Schönhage (1961). □

Hence, the Lebesgue constant of polynomial interpolation at equispaced points grows exponentially with the number of the interpolation points, showing the limits of the derived interpolants. Indeed, $\mathcal{Q}_m[h]$ is not a good substitute for P_h^* given that for a fixed order of polynomials m, its approximation error can be very large compared to that of P_h^*.

On top of that, this discrepancy intensifies exponentially as m grows, thus showing that adding more interpolation points may worsen things instead of helping correct the problem. An illustration of these flaws is provided by the Runge phenomenon, as illustrated in Figure B.1: the equispaced interpolants of a smooth function can exhibit large oscillations near the endpoints of the approximation interval. Moreover the magnitude of these oscillations tends to grow with the number of interpolating points.



(a) 5 interpolation points.

(b) 10 interpolation points.

(c) 15 interpolation points.

(d) 20 interpolation points.

(e) 50 interpolation points.

(f) 75 interpolation points.

Figure B.1: Runge phenomenon. The Runge function (plotted in black), defined over $[-1, 1]$ by $t \mapsto 1/(1 + 25t^2)$, is interpolated using equispaced points (plotted in red). Figures (a) to (f) represent different numbers of interpolation points.

Value of m	Value of Λ^*
10^2	3.453
10^3	4.919
10^4	6.385
10^5	7.851
10^6	9.316

Table B.1: Values of the lower-bound Λ^* of the Lebesgue constant of an interpolator over m distinct points.

Can the approximation accuracy of polynomial interpolation be improved by a better choice of interpolation points? Compared to equispaced points, the answer is yes. However, whatever the sampling of the approximation interval, one should not expect polynomial interpolators to perform well on any continuous function, as formalized by the next theorem.

Theorem B.3.2 (Lebesgue constant for interpolation at distinct points)**.** *Let $t_0, \dots, t_m$ denote $m+1$ (arbitrary) distinct points from $[-1,1]$ and let $\mathcal{A}_m^t$ denote the linear operator that associates to any $h \in \mathscr{C}([-1,1])$ the polynomial $\mathcal{A}_m^t[h] \in \mathscr{P}_m$ that interpolates h at $t_0, \dots, t_m$.*
The Lebesgue constant $\Lambda(\mathcal{A}_m^t)$ satisfies

$$\Lambda(\mathcal{A}_m^t) \geq \Lambda^* \quad \textit{with} \quad \Lambda^* := \frac{2}{\pi}\log(m+1) + C \quad , \tag{B.7}$$

where $C = (2/\pi)(\gamma + \log(4/\pi))$ and $\gamma \approx 0.57722$ is the Euler–Mascheroni constant.

Proof. Erdős (1961) proved the inequality and Brutman (1978) provided the expression of the constant. □

Hence, for any choice of interpolation points, the Lebesgue constant of the associated interpolator grows at least logarithmically with the number of points. Assuming we have found an set of interpolating points $t_0, \dots, t_m$ such that $\Lambda(\mathcal{A}_m^t) = \Lambda^*$, the growth rate of $\Lambda(\mathcal{A}_m^t$ with m would be quite slow. Even for relatively large values of m, its value would be relatively small, as illustrated in Table B.1.

Unfortunately, there is no general approach to find the interpolation points that would lead to this minimal Lebesgue constant. An excellent and readily available substitute are Chebyshev nodes. Given a number of points $m+1$, the Chebyshev nodes are the $m+1$ points of $[-1,1]$ defined as

$$t_j = \cos\left(\left(j + \frac{1}{2}\right)\frac{\pi}{m+1}\right), \quad j \in [\![0, m]\!] \quad . \tag{B.8}$$

They can be interpreted as the projection on the real axis of a set of $m+1$ points regularly distributed points over a unit half-circle, as illustrated in Figure B.2.

Interpolants built from Chebyshev nodes generally have much better approximation qualities than equispaced interpolants, as stated by the following theorem.

Theorem B.3.3 (Lebesgue constant for interpolation at Chebyshev nodes)**.** *Let $\mathcal{I}_m$ denote the linear operator that associates to any $h \in \mathscr{C}([-1,1])$ the polynomial $\mathcal{I}_m[h] \in \mathscr{P}_m$ that interpolates h at $m+1$ Chebyshev nodes of $[-1,1]$ (defined in Equation (B.8)).*
The Lebesgue constant $\Lambda(\mathcal{I}_m)$ satisfies:

$$\Lambda(\mathcal{I}_m) \leq \frac{2}{\pi}\log(m+1) + 1 \quad \textit{and} \quad \Lambda(\mathcal{I}_m) \underset{m\to\infty}{\sim} \frac{2}{\pi}\log m \quad . \tag{B.9}$$

Proof. The inequality is derived from (Ehlich and Zeller, 1966, Theorem 4) and the asymptotic equivalence then follows from Theorem B.3.2. □

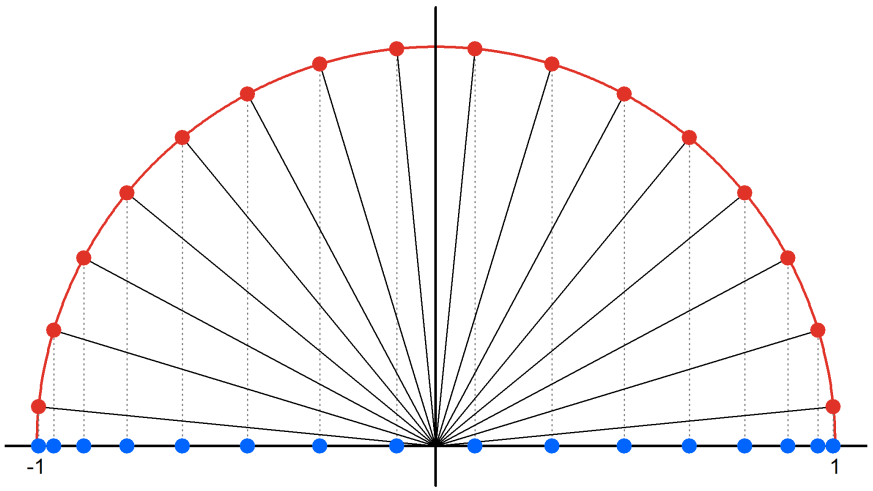


Figure B.2: Regularly spaced points on a unit half-circle and link to Chebyshev nodes. Two consecutive points on the circle are separated by an arc of length $\pi/(m+1)$ rad, where $m+1$ is the total number of points (here, $m = 15$). The projection of these points onto the horizontal axis defines the Chebyshev nodes over $[-1, 1]$.

In particular, given that $\Lambda(\mathcal{I}_m) \geq \Lambda^*$, we have

$$0 \leq \Lambda(\mathcal{I}_m) - \Lambda^* \leq 1 - C \approx 0.4787 \quad ,$$

where C is the constant defined in Theorem B.3.2. Hence, Chebyshev nodes can be considered as a nearly optimal choice of interpolation points given that the Lebesgue constants of the corresponding interpolator is consistently close to the best interpolator over the set of continuous functions.

Moreover, using Chebyshev nodes to approximate a function h over an interval $[a, b]$ actually ensures that the approximation error of $\mathcal{I}_m[h]$ goes to zero as m grows, provided that the function h is "slightly" more than continuous over $[a, b]$. Consequently, adding more interpolation points improves the approximation, until the interpolant coincides with the function. This result is formalized by the notions of bounded variation and Dini-Lipschitz continuity that are now introduced.

Definition B.3.1. [*Variations of a continuous function*] *Let $h \in \mathscr{C}([-1, 1])$. The total variation V_h of h over $[-1, 1]$ is the quantity defined by:*

$$V_h = \sup_{\substack{N \in \mathbb{N}^* \\ -1 = x_0 < x_1 < \cdots < x_{N-1} < x_N = 1}} \sum_{k=0}^{N-1} |h(x_k) - h(x_{k+1})| \quad .$$

h is said to be of bounded variation if $V_h < \infty$.

To get a grip on what the total variation of a function represents, imagine that a point travels along the curve of a continuous function h, starting at the point $(-1, h(-1))$ and ending up at the point $(1, h(1))$. The total variation h corresponds to the length of the path traveled by the projection of that point on the y-axis. In other words, it is the cumulative sum of the heights of the ups and downs of the curve. Imposing that h is of bounded total variation on $[-1, 1]$ actually prevents h from displaying an infinite amount of oscillations.

Definition B.3.2. [*Modulus of continuity and Dini-Lipschitz continuity*] *Let $h : [-1, 1] \to \mathbb{R}$. The modulus of continuity w_h of h is defined for any $\delta > 0$ by:*

$$w_h(\delta) = \sup_{\substack{t_1, t_2 \in [-1,1] \\ |t_1 - t_2| < \delta}} |h(t_1) - h(t_2)|, \quad \delta > 0 \quad .$$

If $w_h(\delta) \log(\delta) \underset{\delta \to 0}{\longrightarrow} 0$, then h is said to be Dini-Lipschitz continuous over $[-1, 1]$.

The condition on the Dini-Lipschitz continuity is a slightly stronger assumption on the approximated function h than continuity, as it imposes some restriction on the difference between the values of h at two infinitely close points of $[-1,1]$: namely if $\delta \to 0$ is the distance separating these points, the difference should go to 0 faster than $\log \delta$ goes to $-\infty$. In particular, Lipschitz-continuous functions, but also derivable functions over $[-1,1]$ are Dini-Lipschitz continuous as they both have moduli of continuity of order $\mathcal{O}(\delta)$.

Theorem B.3.4 (Convergence of Chebyshev interpolants)**.** *Let $h \in \mathscr{C}([-1,1])$ be either of bounded variation or Dini-Lipschitz continuous.*
Let $\mathcal{I}_m[h] \in \mathscr{P}_m$ be the polynomial that interpolates h at $m+1$ Chebyshev nodes of $[-1,1]$. Then,

$$\|h - \mathcal{I}_m[h]\|_\infty \underset{m\to\infty}{\to} 0 \quad .$$

Proof. See Theorem 5.7 in (Mason and Handscomb, 2002) for a proof. Also, the result on Dini-Lipschitz continuity is a direct consequence of Theorems 1.4 and 4.1 in (Rivlin, 1969). □

Theorem B.3.4 hence ensures that if a sufficiently large amount of interpolation points are chosen, the resulting Chebyshev interpolation can yield an approximation of a function that is of bounded variation or Dini-Lipschitz continuous at any level of accuracy. This results explains the renown of Chebyshev interpolants in approximation theory as they successfully approximate a large class of functions.

Finally the rate of convergence of Chebyshev interpolants towards the approximated function h are now derived. These rates of convergence essentially depend on the regularity of h: the more regular it is, the faster the convergence.

Theorem B.3.5. *Let $\nu \geq 1$ be an integer. If $h : \mathbb{R} \to \mathbb{R}$ is such that its derivatives $h, h', \ldots, h^{(\nu-1)}$ are continuous and that $h^{(\nu)}$ is of bounded variation.*
Let $\mathcal{I}_m[h] \in \mathscr{P}_m$ be the polynomial that interpolates h at $m+1$ Chebyshev nodes of $[-1,1]$.
Then,

$$\forall m > \nu, \quad \|h - \mathcal{I}_m[h]\|_\infty \leq \frac{4V}{\pi\nu(m-\nu)^\nu} \quad ,$$

where V denotes the total variation of $h^{(\nu)}$.

Besides if there exists $\rho > 1$ such that the complex function $z \in \mathbb{C} \mapsto h(z)$ is holomorphic inside the ellipse E_ρ centered at 0, with foci $z = \pm 1$ and semi-major (resp. semi-minor) axis of length $(\rho + \rho^{-1})/2$ (resp. $(\rho - \rho^{-1})/2$), then:

$$\forall m \geq 0, \quad \|h - \mathcal{I}_m[h]\|_\infty \leq \frac{4M}{\rho^m(\rho-1)} \quad ,$$

where $M = \sup_{z\in E_\rho} |h(z)|$.

Proof. Mason and Handscomb (2002, Section 5.7) provides a proof of this theorem. □

Hence, whenever h is continuously differentiable, the Chebyshev interpolants $(\mathcal{I}_m[h])_{m\geq 0}$ converge polynomially to h and the order of convergence is equal to its degree of differentiability. Besides, if h is even smoother, quicker rates of convergence can be achieved. Indeed, if h is holomorphic on a ellipse with foci $z = \pm 1$, then in particular it is infinitely differentiable over $[-1,1]$. A geometric convergence with rate depending on the size of this ellipse is achieved.

B.4 Approximation by projection

A second approach to polynomial approximation of a continuous function h consists in looking for a polynomial P_h that is expressed as a weighted sum of polynomials taken from a family that has “suitable" properties. These properties should allow to easily find an expression for

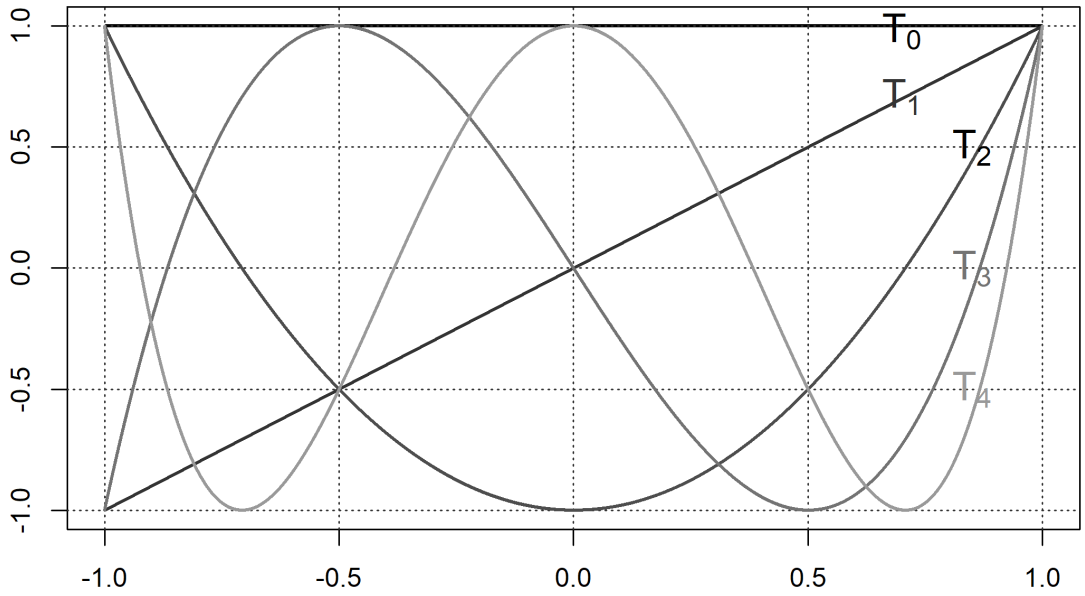


Figure B.3: First 5 Chebyshev polynomials over $[-1, 1]$.

the weights that fit best h. This notion of fit is usually understood in the least-square sense, meaning that the weights are chosen so that they minimize a weighted integral of the square of the difference $h - P_h$ over the approximation interval.

Chebyshev polynomials are an example of family of polynomials widely used for polynomial approximation in this context (Trefethen, 2013). Other families, such as Legendre and Hermite polynomials could also be considered. However Chebyshev polynomials present a double advantage: they yield minimal approximation errors and the weights of the decomposition of the approximators can be computed very efficiently for any function of $[-1, 1]$ using highly optimized algorithms (Mason and Handscomb, 2002; Trefethen, 2013). Hence, the focus of this section will remain on them.

B.4.1 Chebyshev polynomials

The Chebyshev polynomials (of the first kind) are the family $(T_k)_{k\in\mathbb{N}}$ of polynomials defined over $[-1, 1]$ by:

$$\forall k \in \mathbb{N}, \quad \forall \theta \in [-\pi, \pi], \quad T_k(\cos\theta) = \cos(k\theta) \quad , \tag{B.10}$$

or equivalently via the recurrence relation:

$$T_0(t) = 1, \quad T_1(t) = t, \quad T_{k+1}(t) = 2tT_k(t) - T_{k-1}(t) \quad k \geq 1 \quad , \tag{B.11}$$

For instance, the explicit expressions of the first five Chebyshev polynomials are:

$$T_0(t) = 1, \quad T_1(t) = t, \quad T_2(t) = 2t^2 - 1,$$
$$T_3(t) = 4t^3 - 3t, \quad T_4(t) = 8t^4 - 8t^2 + 1.$$

A graphical representation of these polynomials is provided in Figure B.3.

Several properties of Chebyshev polynomials can be derived directly from Equations (B.10) and (B.11) and are now listed. For any $k \in \mathbb{N}$, T_k is a polynomial of degree k. Besides, T_k is an even (resp. odd) polynomial if k is even (resp. odd). Also, $\forall t \in [-1, 1]$, $|T_k(t)| \leq 1$. T_k actually has exactly $k + 1$ extrema, that are either -1 or $+1$ and are located at the following points:

$$\left\{\cos\left(j\frac{\pi}{k}\right) : j \in [\![0, k]\!]\right\} \quad ,$$

Finally, T_k has exactly k distinct roots, whose expressions are:

$$\left\{\cos\left(\left(j + \frac{1}{2}\right)\frac{\pi}{k}\right) : j \in [\![0, k-1]\!]\right\} \quad ,$$

In particular, note that for $m \geq 0$, the Chebyshev nodes $t_0, \ldots, t_m$ defined in Equation (B.8) are exactly the roots of T_{m+1}.

B.4.2 Orthogonality of Chebyshev polynomials and Chebyshev sums

An important property of Chebyshev polynomials, directly linked to the context of least-square approximation, is now exposed. Let $L^2_c([-1,1])$ be the set of functions of $[-1,1]$ that are square-integrable with respect to the weight function $t \mapsto (1-t^2)^{-1/2}$:

$$L^2_c([-1,1]) = \{f : [-1,1] \to \mathbb{R} \text{ such that } \int_{-1}^{1} f(t)^2 \frac{dt}{\sqrt{1-t^2}} < \infty\} \quad .$$

$L^2([-1,1]$ is a Hilbert space when equipped with the inner product $\langle ., . \rangle_c$ (and associated norm $\|.\|_c$) defined by:

$$\begin{aligned} \langle f, g \rangle_c &= \int_{-1}^{1} f(t)g(t)\frac{dt}{\sqrt{1-t^2}}, \qquad f, g \in L^2_c([-1,1]) \quad , \\ \|f\|_c &= \sqrt{\langle f, f \rangle_c}, \qquad\qquad\qquad f \in L^2_c([-1,1]) \quad . \end{aligned}$$

In particular, the Chebyshev polynomials $(T_k)_{k\in\mathbb{N}}$ are in $L^2_c([-1,1])$ and satisfy

$$\forall i, j \in \mathbb{N}, \quad \langle T_i, T_j \rangle_c = \begin{cases} \pi & \text{if } i = j = 0 \\ \pi/2 & \text{if } i = j \neq 0 \\ 0 & \text{if } i \neq j \end{cases} \quad .$$

Therefore, the Chebyshev polynomials form a family of orthogonal polynomials of $L^2_c([-1,1])$, with respect to its inner product. As such, the Chebyshev sum (of order m) of any function $f \in L^2_c([-1,1])$ can be defined as the polynomial of degree (at most) m given by:

$$\mathcal{S}_m[f](t) = \frac{1}{2}c_0 T_0(t) + \sum_{k=0}^{m} c_k T_k(t), \quad t \in [-1,1] \quad ,$$

where the coefficients c_k are defined by:

$$c_k = \frac{2}{\pi}\int_{-1}^{1} f(t)T_k(t)\frac{1}{\sqrt{1-t^2}}dt \quad . \tag{B.12}$$

Hence the Chebyshev sum of order m of f is the sum of the polynomials T_k, $k \in [\![0, m]\!]$, weighted by the coefficients $\langle f, T_k \rangle_c / \langle T_k, T_k \rangle_c$, and can thus be interpreted as an orthogonal projection of f onto the first $m+1$ Chebyshev polynomials.

B.4.3 Chebyshev sums as approximators

The Chebyshev sums of a function $f \in L^2_c([-1,1])$ converge to f in the L^2_c-sense as m goes to infinity, as stated by the following theorem.

Theorem B.4.1 (L^2_c-convergence of Chebyshev sum)**.** *The Chebyshev sums of any square-integrable function of* $[-1,1]$ *(with respect to the weight function* $t \mapsto (1-t^2)^{-1/2}$*) are convergent in the* L^2_c*-sense to the function itself:*

$$\forall f \in L^2_c([-1,1]), \quad \|f - \mathcal{S}_m[f]\|^2_c = \int_{-1}^{1} \left(f(t) - \mathcal{S}_m[f](t)\right)^2 \frac{dt}{\sqrt{1-t^2}} \underset{m\to\infty}{\longrightarrow} 0 \quad .$$

Proof. Mason and Handscomb (2002, Section 5.3.1) provides a proof of this theorem based on an analogy between Fourier series and Chebyshev sums. □

Chebyshev sums hence provide a least-square polynomial approximation of any square-integrable function (with respect to the weight function $t \mapsto (1-t^2)^{-1/2}$). However it is not guaranteed that a Chebyshev sum of order m of a function of $L^2_c([-1,1]$ provides a good approximation of the function itself. Indeed, the value $\mathcal{S}_m[f](t), t \in [-1,1]$ may not even converge as $m \to \infty$. Results on the point-wise and uniform convergence of Chebyshev sums can be derived, but come at a price: restrictions on the regularity of the considered functions.

Theorem B.4.2 (Point-wise convergence of Chebyshev sums)**.** *The Chebyshev sums of any continuous function of* $[-1,1]$ *are point-wise convergent to the function itself:*

$$\forall f \in \mathscr{C}([-1,1]), \quad \forall t \in [-1,1], \quad \mathcal{S}_m[f](t) \underset{m\to\infty}{\longrightarrow} f(t) \quad .$$

Proof. See Mason and Handscomb (2002, Section 5.3.2) for a proof based on an analogy between Fourier series and Chebyshev sums. □

Following this result, the notion of Chebyshev series can be defined. The Chebyshev series $\mathcal{S}[f]$ of a function $f \in \mathscr{C}([a,b])$ is the function of $[-1,1]$ such that

$$\forall x \in [-1,1], \quad \mathcal{S}[f](x) = \lim_{m\to\infty} \mathcal{S}_m[f](x) = \frac{1}{2}c_0 T_0(x) + \sum_{k=0}^{\infty} c_k T_k(x) \quad .$$

Chebyshev series can be seen as orthogonal projections of functions of $[-1,1]$ over the entire family of Chebyshev polynomials $(T_k)_{k\geq 0}$. In particular, following Theorem B.4.2, $\mathcal{S}[f] = f$ for any $f \in \mathscr{C}([-1,1])$.

Moreover, Theorem B.4.2 ensures that for any point x in the approximation interval $[a,b]$, the value of the Chebyshev sum of a continuous function f at x will converge to $f(x)$. Hence, the Chebyshev sums approximate the function at each point. Considered as approximations of continuous functions, Chebyshev sums can be compared to interpolants introduced in the previous section in terms of Lebesgue constants.

Theorem B.4.3 (Lebesgue constant for Chebyshev sums)**.** *Let* $\mathcal{S}_m$ *denote the linear operator that associates to any* $f \in \mathscr{C}([-1,1])$ *its Chebyshev sum of order* m*,* $\mathcal{S}_m[f] \in \mathscr{P}_m$*.*
The Lebesgue constants $\Lambda(\mathcal{S}_m)$*,* $m \geq 0$ *satisfy*

$$\Lambda(\mathcal{S}_m) < \frac{4}{\pi^2}\log(m+1) + 3 \quad \textit{and} \quad \Lambda(\mathcal{S}_m) \underset{m\to\infty}{\sim} \frac{4}{\pi^2}\log m \quad . \tag{B.13}$$

Proof. The inequality is derived from Lemma 2.2 in (Rivlin, 1969) and the asymptotic equivalence was established by Fejér (1910). □

Comparing this last result to Theorems B.3.2 and B.3.3, Chebyshev sums have asymptotically lower Lebesgue constants than interpolants, by a factor $(2/\pi)$.

However, as it was the case for Chebyshev interpolants, there is no guarantee that low approximation errors can be achieved using Chebyshev sums to approximate arbitrary continuous functions. Indeed, the convergence rate of $\mathcal{S}_m[f](x)$ towards $f(x)$ can be arbitrary fast or slow depending on the point x. In other words, $\mathcal{S}_m[f](x)$ can approximate f at each individual point $x \in [-1,1]$ (for different values of m depending on x), but there is no guarantee that for given a m, $\mathcal{S}_m[f]$ can approximate f over $[-1,1]$. Uniform convergence is needed to ensure it.

Theorem B.4.4 (Uniform convergence of Chebyshev sums)**.** *The Chebyshev sums* $\{\mathcal{S}_m[f]\}_{m\geq 0}$ *of any function* f *of* $[-1,1]$ *that is of bounded variation or Dini-Lipschitz continuous converge uniformly to the function itself, i.e.*

$$\|f - \mathcal{S}_m[f]\|_\infty \underset{m\to\infty}{\longrightarrow} 0 \quad .$$

Proof. See once again Mason and Handscomb (2002, Section 5.3.2) for a proof based on an analogy between Fourier series and Chebyshev sums. □

Hence, provided some mild additional assumptions on a continuous function (namely that it is of bounded variation or Dini-Lipschitz continuous), Chebyshev sums provide a great approximation tool, as Theorem B.4.4 ensures that the approximation error of sum of order m goes to 0 as m grows. Approximation at any desired accuracy can therefore be obtained by taking an order m large enough. Note that it is the same result as the one obtained for Chebyshev interpolants in Theorem B.3.4.

Finally the rate of convergence of Chebyshev sums towards the approximated function h is described. Similarly to Chebyshev interpolants, these rates of convergence depend on the regularity of h.

Theorem B.4.5. *Let $\nu \geq 1$ be an integer. If $h : \mathbb{R} \to \mathbb{R}$ is such that its derivatives $h, h', \ldots, h^{(\nu-1)}$ are continuous and that $h^{(\nu)}$ is of bounded variation, then:*

$$\forall m \geq \nu, \quad \|h - \mathcal{S}_m[h]\|_\infty \leq \frac{2V}{\pi\nu(m-\nu)^\nu} \quad ,$$

where V denotes the total variation of $h^{(\nu)}$.

Besides if there exists $\rho > 1$ such that the complex function $z \in \mathbb{C} \mapsto h(z)$ is holomorphic inside the ellipse E_ρ centered at 0 and with foci $z = \pm 1$ and semi-major (resp. semi-minor) axis of length $(\rho + \rho^{-1})/2$ (resp. $(\rho - \rho^{-1})/2$), then:

$$\forall m \geq 0, \quad \|h - \mathcal{S}_m[h]\|_\infty \leq \frac{2M}{\rho^m(\rho-1)} \quad ,$$

where $M = \sup_{z \in E_\rho} |h(z)|$.

Proof. Mason and Handscomb (2002, Section 5.7) provides a proof of this theorem. □

Hence, the same rates of convergence as Chebyshev interpolants hold for Chebyshev sums, up to a multiplication factor (equal to 2). Note that these rates of convergence are directly linked to the rate of decrease of the coefficients of the Chebyshev sums as m grows, as stated in the following theorem.

Theorem B.4.6. *Let $\nu \geq 1$ be an integer. If $h : \mathbb{R} \to \mathbb{R}$ is such that its derivatives $h, h', \ldots, h^{(\nu-1)}$ are continuous and that $h^{(\nu)}$ is of bounded variation, then the coefficients $(c_m)_{m \geq 0}$ of the Chebyshev sums of h satisfy*

$$\forall m \geq \nu + 1, \quad |c_m| \leq \frac{2V}{\pi(m-\nu)^{\nu+1}} \quad ,$$

where V denotes the total variation of $h^{(\nu)}$.

Besides if there exists $\rho > 1$ such that the complex function $z \in \mathbb{C} \mapsto h(z)$ is holomorphic inside the ellipse E_ρ centered at 0 and with foci $z = \pm 1$ and semi-major (resp. semi-minor) axis of length $(\rho + \rho^{-1})/2$ (resp. $(\rho - \rho^{-1})/2$), then

$$\forall m \geq 1, \quad |c_m| \leq \frac{2M}{\rho^m} \quad ,$$

where $M = \sup_{z \in E_\rho} |h(z)|$.

Proof. Trefethen (2013, Theorems 7.1 & 8.1) provides a proof of this theorem. □

B.4.4 Computation of Chebyshev sums and link to Chebyshev interpolants

Computing the coefficients of the Chebyshev sum of a function $h \in \mathscr{C}([-1,1])$ consists in evaluating integrals defined by Equation (B.12). Applying the change of variable $t \to \cos\theta$ to these integrals gives

$$\forall k \in [\![0, m]\!], \quad c_k = \frac{2}{\pi} \int_0^\pi h(\cos\theta)\cos(k\theta) d\theta \quad . \tag{B.14}$$

An analytical expression of Equation (B.14) is seldom available. Instead, the coefficients c_k are numerically evaluated by discretizing the integral in Equation (B.14) using Riemann sums. Hence, first the interval of integration $[0, \pi]$ is discretized into an number N of equispaced points, denoted $\theta_0, \ldots, \theta_N$ and given by

$$\theta_j = j\frac{\pi}{N}, \quad j \in [\![0, N]\!] \quad .$$

If the Left rule is used, the integral of a function f defined over $[0, \pi]$ is approximated by the integral of a piecewise constant function, taking the value $f(\theta_j)$ in any interval $[\theta_j, \theta_{j+1}[$, for $0 \leq j \leq N-1$. Getting back to Equation (B.14), this means that the coefficients c_k are approximated by coefficients $c_k^{(L)}$ defined by

$$c_k^{(L)} = \frac{2}{\pi}\sum_{j=0}^{N-1}(\theta_{j+1} - \theta_j)h(\cos\theta_j)\cos(k\theta_j) = \frac{2}{N}\sum_{j=0}^{N-1} h\left(\cos\left(j\frac{\pi}{N}\right)\right)\cos\left(kj\frac{\pi}{N}\right), \quad k \in [\![0, m]\!] \, .$$

Distinguishing the case where k is even ($k = 2p$) or odd ($k = 2p+1$) gives the following set of equations:

$$p \in [\![0, \lfloor m/2 \rfloor]\!], \quad \left| \begin{array}{l} c_{2p}^{(L)} = \dfrac{2}{N}\mathfrak{Re}\left(\displaystyle\sum_{j=0}^{N-1} h\left(\cos\left(j\frac{\pi}{N}\right)\right)e^{-\mathrm{i}\frac{2\pi}{N}pj}\right) \\ c_{2p+1}^{(L)} = \dfrac{2}{N}\mathfrak{Re}\left(\displaystyle\sum_{j=0}^{N-1} e^{-\mathrm{i}\frac{\pi}{N}j}h\left(\cos\left(j\frac{\pi}{N}\right)\right)e^{-\mathrm{i}\frac{2\pi}{N}pj}\right) \end{array} \right. , \tag{B.15}$$

where $\mathfrak{Re}$ denotes the real part of a complex number. Assuming that $p < N$, the sums in Equation (B.15) are the expressions of $(p+1)$-th component of the discrete Fourier transform (DFT) of two vectors of $\mathbb{C}^N$ (cf. Section 1.2.2):

$$c_{2p}^{(L)} = \frac{2}{N}\mathfrak{Re}\left(\mathrm{DFT}\left[\boldsymbol{h}_e\right]_{p+1}\right) \quad \text{and} \quad c_{2p+1}^{(L)} = \frac{2}{N}\mathfrak{Re}\left(\mathrm{DFT}\left[\boldsymbol{h}_o\right]_{p+1}\right), \quad p \in [\![0, \lfloor m/2 \rfloor]\!] \quad , \tag{B.16}$$

where $\boldsymbol{h}_e, \boldsymbol{h}_o \in \mathbb{C}^N$ are the vectors define by:

$$\forall j \in [\![0, N-1]\!], \quad \left| \begin{array}{l} [\boldsymbol{h}_e]_{j+1} = h\left(\cos\left(j\dfrac{\pi}{N}\right)\right) \\ [\boldsymbol{h}_o]_{j+1} = e^{-\mathrm{i}\frac{\pi}{N}j}h\left(\cos\left(j\dfrac{\pi}{N}\right)\right) \end{array} \right. . \tag{B.17}$$

The DFT appearing in Equation (B.16) can be computed using the fast Fourier transform (FFT) algorithm of Cooley and Tukey (1965). This algorithm, widely used in signal processing applications, computes the DFT of any (complex) vector of size N at a computational complexity of $\mathcal{O}(N \log N)$. Algorithm B.1 sums up this first approach to compute Chebyshev coefficients.

Algorithm B.1: Computation of Chebyshev coefficients by FFT

Input: A function $h \in L_c^2([-1, 1])$. A number $(m+1)$ of coefficients to be computed. An order of approximation of the integrals $N > m$.

Output: Approximations of the coefficients of the Chebyshev sum of order m of h.

Initialization: $c_0^{(L)}, \ldots, c_m^{(L)} = 0$;

1. Compute the vectors $\boldsymbol{h}_e, \boldsymbol{h}_o \in C^N$ defined by Equation (B.17).
2. Compute the discrete Fourier transforms $\boldsymbol{y}_e, \boldsymbol{y}_o \in \mathbb{C}^N$ of both vectors using the FFT algorithm: $\boldsymbol{y}_e = \mathrm{DFT}[\boldsymbol{h}_e]$ and $\boldsymbol{y}_o = \mathrm{DFT}[\boldsymbol{h}_o]$.
3. **for** p **from** 0 **to** $\lfloor m/2 \rfloor$ **do**
   - $c_{2p}^{(L)} \leftarrow \frac{2}{N}\mathfrak{Re}([\boldsymbol{y}_e]_{p+1})$;
   - $c_{2p+1}^{(L)} \leftarrow \frac{2}{N}\mathfrak{Re}([\boldsymbol{y}_o]_{p+1})$;

Return $c_0^{(L)}, \ldots, c_m^{(L)}$.

Another possible way to numerically compute the coefficients of a Chebyshev sums consists in approximating the integral in Equation (B.14) using the Midpoint rule. In that case the integral of a function f defined over $[0, \pi]$ is once again approximated by the integral of a piecewise constant function, taking now the value $f((\theta_j + \theta_{j+1}))/2)$ in any interval $[\theta_j, \theta_{j+1}[$, for $0 \leq j \leq N-1$. Getting back to Equation (B.14), this means that the coefficients c_k are now approximated by coefficients $c_k^{(M)}$ defined by

$$c_k^{(M)} = \frac{2}{N} \sum_{j=0}^{N-1} h\left(\cos\left(\left(j+\frac{1}{2}\right)\frac{\pi}{N}\right)\right) \cos\left(k\left(j+\frac{1}{2}\right)\frac{\pi}{N}\right), \quad k \in [\![0, m]\!] \quad . \tag{B.18}$$

If we denote $\boldsymbol{h} \in \mathbb{R}^N$ the vector defined by

$$[\boldsymbol{h}]_{j+1} = h\left(\cos\left(\left(j+\frac{1}{2}\right)\frac{\pi}{N}\right)\right), \quad j \in [\![0, N-1]\!] \quad , \tag{B.19}$$

then Equation (B.18) is the expression of k-th component of the discrete cosine transform (of type II, which we denote DCT) (Oppenheim et al., 2001) of $\boldsymbol{h}$.

Note in particular that the expression of the vector $\boldsymbol{h}$ in Equation (B.19) corresponds to the evaluation of the function h at the Chebyshev nodes (of order N) $t_0, \ldots, t_{N-1}$ as defined in Equation (B.8), or equivalently at the zeros of T_N. In short, the approximation of the Chebyshev coefficients using a Midpoint rule are given by

$$\begin{pmatrix} c_0^{(M)} \\ \vdots \\ c_m^{(M)} \end{pmatrix} = \frac{2}{N} \left[\mathrm{DCT}\left[\boldsymbol{h}\right]\right]_{1:(m+1)} = \frac{2}{N} \left[\mathrm{DCT}\left[\begin{pmatrix} h(t_0) \\ \vdots \\ h(t_{N-1}) \end{pmatrix}\right]\right]_{1:(m+1)} \quad , \tag{B.20}$$

where $t_j = \cos\left(\left(j+\frac{1}{2}\right)\frac{\pi}{N}\right), j \in [\![0, N-1]\!]$ and the notation $[\boldsymbol{x}]_{1:(m+1)}$ stands for the restriction of a vector $\boldsymbol{x}$ to its first $(m+1)$ components.

DCTs can be computed using FFT algorithms at the cost some pre-processing and post-processing steps applied to the vectors which are similar to those presented in Algorithm B.1 (Chen et al., 1977; Makhoul, 1980). Other algorithms specially designed for DCTs also allow to compute these sums with an improved complexity with respect to adaptations of the FFT (Chen et al., 1977). Algorithm B.2 sums up this second approach to the computation of Chebyshev coefficients.

Algorithm B.2: Computation of Chebyshev coefficients by discrete cosine transform.

Input: A function $h \in L_c^2([-1, 1])$. A number $(m+1)$ of coefficients to be computed . An order of approximation of the integrals $N > m$.

Output: Approximations of the coefficients of the Chebyshev sum of order m of h.

Initialization: $c_0^{(M)}, \ldots, c_m^{(M)} = 0$;

1. Compute the vectors $\boldsymbol{h} \in \mathbb{R}^N$ defined by Equation (B.19).
2. Compute the discrete cosine transform $\boldsymbol{y} \in \mathbb{R}^N$ of this vector using a suitable algorithm: $\boldsymbol{y} = \mathrm{DCT}[\boldsymbol{h}]$.
3. **for** k **from** 0 **to** m **do**
   $c_k^{(M)} \leftarrow \frac{2}{N}[\boldsymbol{y}]_{k+1}$;

Return $c_0^{(M)}, \ldots, c_m^{(M)}$.

B.4.5 Link between Chebyshev sums an Chebyshev interpolant

In this subsection, we highlight the fundamental that exists between Chebyshev sums and interpolants.

Let $h \in \mathscr{C}([-1, 1])$ and let $\mathcal{I}_m[h]$ be its Chebyshev interpolant of order m. $\mathcal{I}_m[h]$ can be written in the basis of Chebyshev polynomials as:

$$\mathcal{I}_m[h](t) = \frac{1}{2} c_0^{(I)} T_0(t) + \sum_{k=1}^{m} c_k^{(I)} T_k(t) \quad .$$

The expression of the coefficients $c_k^{(I)}$ is obtained by enforcing for any $j \in [\![0, m]\!]$ that $\mathcal{I}_m[h](t_j) = h(t_j)$ where t_j denote the Chebyshev nodes, or equivalently, the zeros of T_{m+1}. This gives:

$$h(t_j) = \frac{1}{2}c_0^{(I)}T_0(t_j) + \sum_{k=0}^{m} c_k^{(I)}T_k(t_j) = \frac{1}{2}c_0^{(I)} + \sum_{k=0}^{m} c_k^{(I)} \cos\left(k\left(j+\frac{1}{2}\right)\frac{\pi}{m+1}\right) \quad .$$

This last sum is the expression of the j-th component of the DCT of type III, which corresponds to the inverse of the DCT of type II, of the vector $(c_0^{(I)}, \dots, c_m^{(I)})^T \in \mathbb{R}^{m+1}$. Hence, this last vector is given by taking the DCT of the vector $(h(t_0), \dots, h(t_m))^T$. In short, the coefficients of the decomposition of the Chebyshev interpolant into the Chebyshev polynomial basis are

$$\begin{pmatrix} c_0^{(I)} \\ \vdots \\ c_m^{(I)} \end{pmatrix} = \frac{2}{m+1}\mathrm{DCT}\left[\begin{pmatrix} h(t_0) \\ \vdots \\ h(t_m) \end{pmatrix}\right] : \quad t_j = \cos\left(\left(j+\frac{1}{2}\right)\frac{\pi}{m+1}\right), \quad j \in [\![0, m]\!] \quad . \tag{B.21}$$

This last equation is extremely similar to Equation (B.20), which is used to approximate the coefficients of the Chebyshev series of h using the Midpoint rule. In fact, if the order of discretization of integrals N is taken so that $N = m + 1$, both equations coincide. Hence, the $(m + 1)$ coefficients of the Chebsyhev interpolant $\mathcal{I}_m[h]$ in the Chebyshev basis can be viewed as approximations of the $(m+1)$ coefficients of the Chebyshev sum $\mathcal{S}_m[h]$ using a Midpoint rule with discretization order $N = m + 1$.

Hence, the Chebyshev interpolant of order m of a function is an approximation of its Chebyshev sum of order m obtained by replacing the coefficients of the Chebyshev sum (cf. Equation (B.14)) by their approximation using a Midpoint rule with exactly $m + 1$ points. This explains the great similarity between the asymptotic properties of both approximators, as stated in Theorems B.3.3 and B.4.3 but also Theorems B.3.4 and B.4.4. Indeed, the larger the order of approximation m, the closer the discretization of the integral is to the actual integral. In particular, the discrepancy between these two sets of coefficients is of order $\mathcal{O}(1/m^2)$.

In practice thought, the order N of discretization of the integrals , which is one of the parameters of Algorithm B.2 (and Algorithm B.1) is chosen almost independently from the number of desired coefficients m. Indeed, N can be chosen quite large given that the algorithms computing the Chebyshev coefficients are very cheap: they have computational complexities of order $\mathcal{O}(N \log N)$. Rather than sticking to the Chebyshev interpolant by choosing $N = m + 1$, one can aim at directly computing highly accurate approximations of the coefficients of the Chebyshev sums using larger values of N.

B.4.6 Chebyshev sums of discontinuous functions

Until this subsection, only continuous functions $h : [-1, 1] \mapsto \mathbb{R}$ were considered. This restriction allowed for Chebyshev sums (and interpolants) to benefit from point-wise convergence properties, but also uniform convergence by requiring slightly more than continuity. In this section, we investigate what tools are at hand when h is discontinuous over $[-1, 1]$, more precisely when h has a finite number of step discontinuities over $[-1, 1]$ and is continuous otherwise.

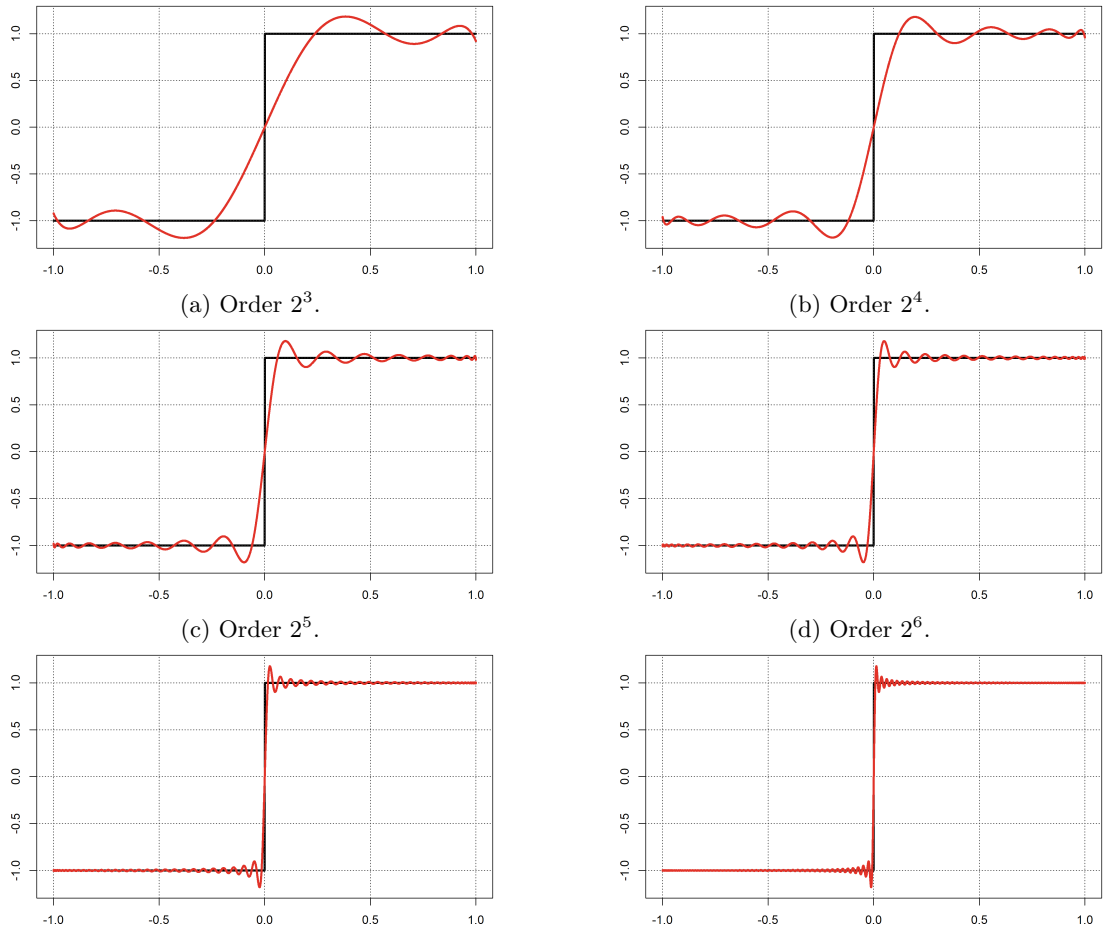
Theorem B.4.7. *Let $h : [-1, 1] \to \mathbb{R}$ such that there exists $N_d \geq 1$ and $t_1, \dots, t_{N_d} \in [-1, 1]$ such that*

$$\forall j \in [\![1, N_d]\!], \quad h(t_j^-) := \lim_{\substack{t \to t_j \\ t < t_j}} h(t) \neq h(t_j^+) := \lim_{\substack{t \to t_j \\ t > t_j}} h(t) \quad ,$$

and such that h is continuous in any subinterval of $[-1, 1]$ that does not contain any of the points $t_1, \dots, t_{N_d}$. Then, the Chebyshev sums of h satisfy

$$\forall t \in [-1, 1], \quad \lim_{m \to \infty} \mathcal{S}_m[h](t) = \begin{cases} \frac{h(t_j^-)+h(t_j^+)}{2} & \text{if } t = t_j \text{ for some } j \in [\![1, N_d]\!] \\ h(t) & \text{otherwise} \end{cases} \quad .$$

Proof. Mason and Handscomb (2002, Section 5.3.2) provides a proof of this statement. □



(a) Order 2^3. (b) Order 2^4.

(c) Order 2^5. (d) Order 2^6.

(e) Order 2^7. (f) Order 2^8.

Figure B.4: Gibbs phenomenon. The sign function (plotted in black), defined over $[-1, 1]$ is approached by Chebyshev sums at various orders (plotted in red in Figures (a) to (f)).

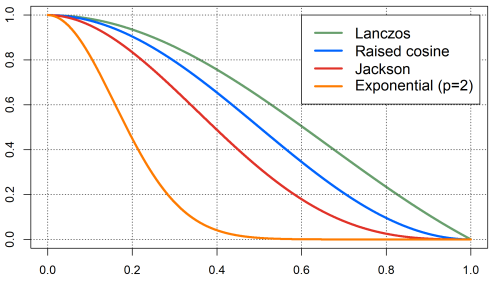
This theorem therefore states that the Chebyshev sums actually converge point-wise even though the function has a finite number of discontinuities. At these points of discontinuity it converges to the mean of the right and left limits of the function. The notion of Chebyshev series is therefore still well defined as a function of $[-1, 1]$.

However, using the Chebyshev sums (or interpolants) as approximators of such functions will not work out due to what is referred to the so-called *Gibbs phenomenon*. Indeed, near a discontinuity, the Chebyshev sums exhibit oscillations that fade out starting from the discontinuity. As the order of the sum increases, the oscillations fade out quicker but their magnitude near the discontinuity does not die out, and rather tends to a fixed constant (Trefethen, 2013). In particular, due to these oscillations, the value of the maximum of the Chebyshev sum near the discontinuity function will be higher than that of the function itself, even as the order grows to infinity. The Gibbs phenomenon is illustrated in Figure B.4 for the case of the sign function sg being approached by Chebyshev sums. Recall that the sign function is defined over $[-1, 1]$ by

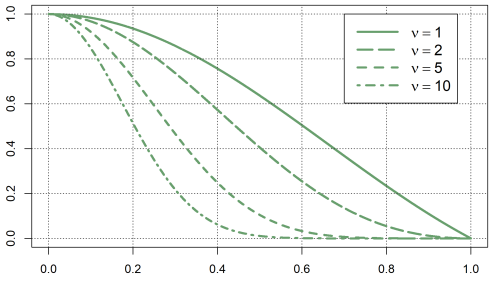
$$\mathrm{sg}(t) = \begin{cases} 1 & \text{if } t > 0 \\ 0 & \text{if } t = 0 \\ -1 & \text{otherwise} \end{cases} .$$

In that case, as the order of the sums goes to infinity, the magnitude of the oscillations tends to a value of around 1.179 for Chebyshev sums, 1.282 for Chebyshev interpolants with odd orders, and 1.066 for Chebyshev interpolants with even orders (Trefethen, 2013, Theorems 9.1 & 9.2).

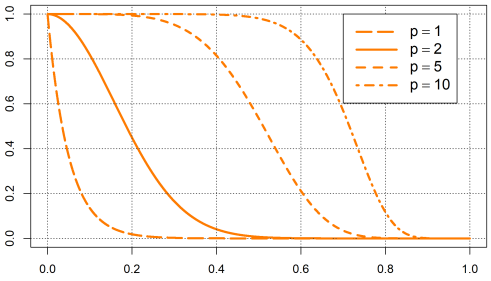
In order to reduce the oscillation of the Gibbs phenomenon when dealing with Chebyshev approximations of discontinuous functions, Lanczos (1988) introduced the use of σ-factors. A



(a) Comparative of plots of σ-factors over $[0, 1]$.



(b) Lanczos σ-factor elevated at various powers ν.



(c) Exponential σ-factors for various values of p.

Figure B.5: σ-factors over $[0, 1]$.

σ-factor is a continuous functions σ defined over $[0, 1]$ that is used to downscale the coefficients of a Chebyshev sum. More precisely, the j-th coefficient of a sum of order m is multiplied by a factor $\sigma(j/m)$. σ-factors are chosen so that high order coefficients are more affected than low order ones, namely $\sigma(t) \approx 1$ if $t \approx 0$ and $\sigma(t) \approx 0$ if $t \approx 1$. A σ-approximation $\mathcal{S}_m^\sigma[h]$ of a Chebyshev sum $\mathcal{S}_m[f]$ is the sum of Chebyshev polynomials defined as follows:

$$\text{if} \quad \mathcal{S}_m[f](t) = \sum_{k=0}^{m} c_k T_k(t), \quad \text{then} \quad \mathcal{S}_m^\sigma[f](t) = \sum_{k=0}^{m} \sigma\left(\frac{k}{m}\right) c_k T_k(t) \quad .$$

To understand why σ-approximations represent a smooth version of their associated Chebyshev sums, recall that a Chebyshev polynomial T_j oscillates $j + 1$ times between the values $+1$ and -1. Hence the higher j is, the more oscillations T_j exhibits. In a Chebyshev sum, having polynomials T_j associated to relatively high coefficients c_j may therefore lead to a Chebyshev sum that also displays the same high-frequency oscillations as T_j: this is the origin of the Gibbs phenomenon.

In the σ-approximation, the scaling of the coefficients c_j corresponding to high values of j by a factor near zero ensures that the highly oscillatory Chebyshev polynomial will not impact the overall behavior of the sum. However, the reduction of the oscillatory behavior comes at a price: an increase of the approximation error near the discontinuity.

Usual choices of σ-factors include (Di Napoli et al., 2016; Gelb and Gottlieb, 2007):

- the Lanczos σ-factor defined by

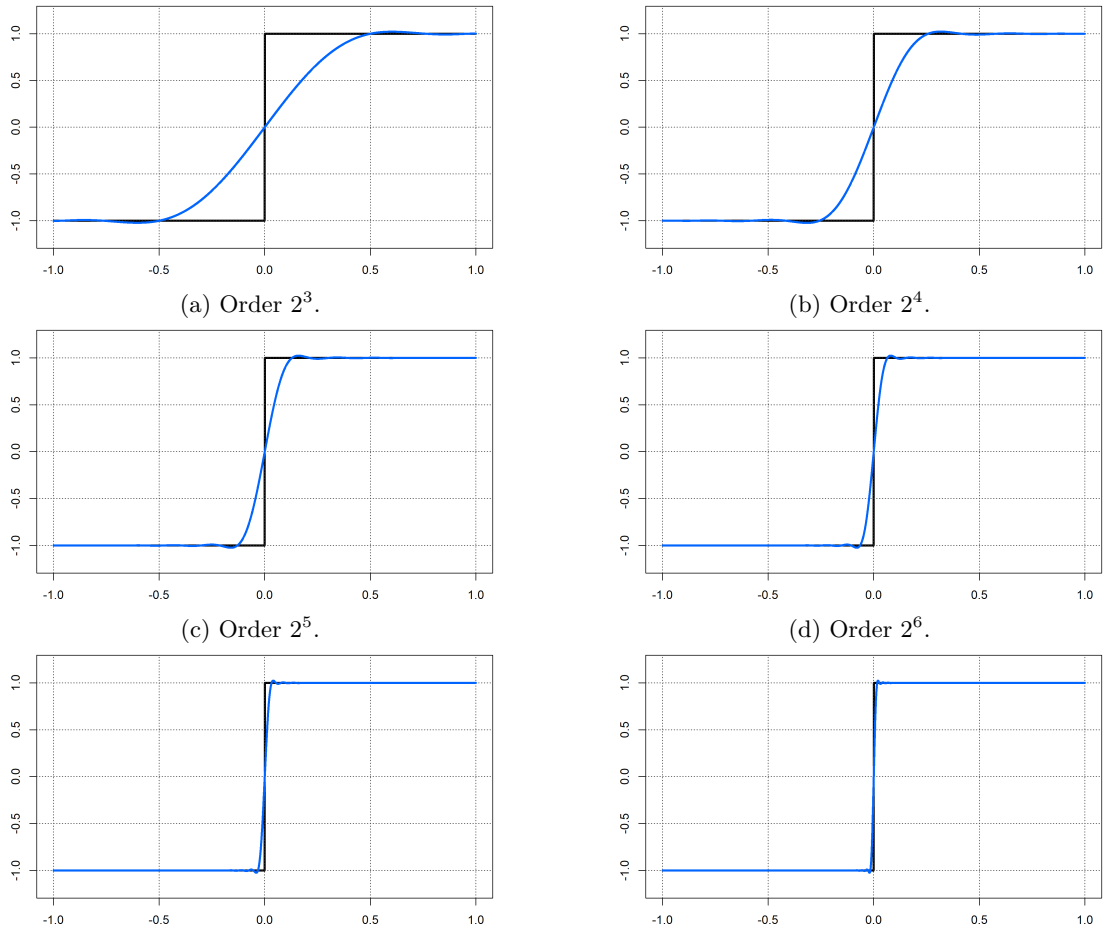
$$\sigma(t) = \frac{\sin(\pi t)}{\pi t}, \quad t \in [0, 1] \quad , \tag{B.22}$$

- the Raised cosine σ-factor defined by

$$\sigma(t) = \frac{1}{2}(1 + \cos(\pi t)), \quad t \in [0, 1] \quad , \tag{B.23}$$

- the Jackson σ-factor defined by

$$\sigma(t) = (1 - t)\cos(\pi t) + \frac{\sin(\pi t)}{\pi}, \quad t \in [0, 1] \quad , \tag{B.24}$$



(a) Order 2^3. (b) Order 2^4.

(c) Order 2^5. (d) Order 2^6.

(e) Order 2^7. (f) Order 2^8.

Figure B.6: Lanczos correction for the sign function. The sign function (plotted in black), defined over $[-1, 1]$ is approached by σ-approximations of Chebyshev sums at various orders (plotted in blue in Figures (a) to (f)). The σ-factor that was used is the Lanczos σ-factor.

- The Exponential σ-factor defined by

$$\sigma(t) = e^{-\alpha t^{\nu}}, \quad t \in [0, 1] \quad , \tag{B.25}$$

  where α is fixed so that $\sigma(1) = e^{-\alpha} \approx 0$ (for instance to the machine accuracy) and p is an additional parameter controlling the speed of decay to zero of the parameter σ.

Note that the first three σ-factors can also be elevated at a given power $\nu \geq 1$ so as to increase the speed at which the factor goes to zero when $t \to 1$. Indeed, it can easily be showed that elevating one of these factors at higher and higher powers ν yields functions that decrease from 1 to a near-zero values faster and faster. This is actually the same effect as one observes when taking smaller an smaller values of p for Exponential σ-factors. All these factors and the effect of the parameters ν and p are illustrated in Figure B.5. An illustration of the effect of Lanczos σ-factors on the approximation of the sign function is presented in Figure B.6.

In conclusion, this subsection exposed the adjustments that should be made when dealing with piecewise continuous functions (with a finite number of discontinuities). Chebyshev sums (and interpolants) can still be used as approximators of such functions. However, no uniform convergence of the sums towards the function should be expected and oscillations near the discontinuities will appear. To overcome this last issue, the coefficients of the sum should be downscaled using a σ-factor so as to reduce the impact of high order Chebyshev polynomials. The resulting σ-approximation is basically a smoothed version of its oscillatory counterpart. However the gain in smoothness is paid by a higher approximation error as the σ-approximation does not approximate the discontinuous function anymore, but rather a smoothed version of this function.

C Proofs

C.1 Chapter 1

Proposition 1.2.1

Proof. Let $M_{\mathcal{W}}$ be the complex random measure defined from $\mathcal{W}$ by:

$$M_{\mathcal{W}}(d\boldsymbol{\xi}) = d\boldsymbol{\xi} \int_{\mathbb{R}^d} e^{-\mathrm{i}\langle \boldsymbol{\xi}, \boldsymbol{t}\rangle} \mathcal{W}(d\boldsymbol{t}) \quad .$$

On one hand, for any $B \in \mathcal{B}(\mathbb{R}^d)$,

$$\begin{aligned}\int_B d\boldsymbol{t} \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} e^{\mathrm{i}\langle \boldsymbol{\xi}, \boldsymbol{t}\rangle} M_{\mathcal{W}}(d\boldsymbol{\xi}) &= \frac{1}{(2\pi)^d} \int_B \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} e^{\mathrm{i}\langle \boldsymbol{\xi}, \boldsymbol{t}-\boldsymbol{u}\rangle} \mathcal{W}(d\boldsymbol{u}) d\boldsymbol{\xi} d\boldsymbol{t} \\ &= \frac{1}{(2\pi)^d} \int_B \int_{\mathbb{R}^d} (2\pi)^d \delta_{\boldsymbol{t}-\boldsymbol{u}} \mathcal{W}(d\boldsymbol{u}) d\boldsymbol{t} = \int_B \mathcal{W}(d\boldsymbol{t}) \quad .\end{aligned}$$

On the other hand, for any $B, B_1, B_2 \in \mathcal{B}(\mathbb{R}^d)$, $M_{\mathcal{W}}$ satisfies $\mathbb{E}[M_{\mathcal{W}}(B)] = 0$ and

$$\begin{aligned}\mathrm{Cov}[M_{\mathcal{W}}(B_1), M_{\mathcal{W}}(B_2)] &= \mathbb{E}[M_{\mathcal{W}}(B_1) M_{\mathcal{W}}(B_2)^*] \\ &= \int_{B_1} \int_{B_2} \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} e^{-\mathrm{i}\langle \boldsymbol{\xi}_1, \boldsymbol{t}_1\rangle} e^{\mathrm{i}\langle \boldsymbol{\xi}_2, \boldsymbol{t}_2\rangle} \mathbb{E}[\mathcal{W}(d\boldsymbol{t}_2)\mathcal{W}(d\boldsymbol{t}_1)] d\boldsymbol{\xi}_2 d\boldsymbol{\xi}_1 \\ &= \int_{B_1} \int_{B_2} \int_{\mathbb{R}^d} e^{-\mathrm{i}\langle \boldsymbol{\xi}_1 - \boldsymbol{\xi}_2, \boldsymbol{t}_1\rangle} \sigma^2 d\boldsymbol{t}_1 d\boldsymbol{\xi}_2 d\boldsymbol{\xi}_1 \quad .\end{aligned}$$

Denoting then $\delta_{\boldsymbol{\xi}}$ the Dirac delta function concentrated at $\boldsymbol{\xi} \in \mathbb{R}^d$, we get:

$$\begin{aligned}\mathrm{Cov}[M_{\mathcal{W}}(B_1), M_{\mathcal{W}}(B_2)] &= \sigma^2 \int_{B_1} \int_{B_2} (2\pi)^d \delta_{\boldsymbol{\xi}_1 - \boldsymbol{\xi}_2} d\boldsymbol{\xi}_2 d\boldsymbol{\xi}_1 \\ &= (2\pi)^d \sigma^2 \int_{B_1 \cap B_2} d\boldsymbol{\xi}_1 = (2\pi)^d \sigma^2 \mathrm{Leb}(B_1 \cap B_2) \quad .\end{aligned}$$

On one hand, if $B_1 \cap B_2 = \emptyset$, $\mathrm{Cov}[M_{\mathcal{W}}(B_1), M_{\mathcal{W}}(B_2)] = 0$. On the other hand, $\mathrm{Var}[M_{\mathcal{W}}(B)] = (2\pi)^d \sigma^2 \mathrm{Leb}(B)$. □

C.2 Chapter 4

Proposition 4.1.1

Proof. First, note that the vector $(\boldsymbol{Z}, \boldsymbol{W}_o)$ defines a Gaussian vector as it is the concatenation of two (independent) Gaussian vectors. In particular,

$$\begin{pmatrix} \boldsymbol{Z} \\ \boldsymbol{W}_o \end{pmatrix} \sim \mathcal{N}\left(\boldsymbol{0}, \begin{pmatrix} \mathrm{Var}[\boldsymbol{Z}] & \\ & \boldsymbol{I}_d \end{pmatrix}\right) \quad .$$

Denote then $\tilde{\boldsymbol{Z}}$ the random vector obtained by concatenating $\boldsymbol{Z}$ and $\boldsymbol{Z}_o$ and let $\boldsymbol{A}$ be the matrix defined by

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{I}_n & \boldsymbol{0}_{n,d} \\ \boldsymbol{M}_o & \tau \boldsymbol{I}_d \end{pmatrix} \quad .$$

In particular, $\tilde{\boldsymbol{Z}}$ satisfies

$$\tilde{\boldsymbol{Z}} = \begin{pmatrix} \boldsymbol{Z} \\ \boldsymbol{Z}_o \end{pmatrix} = \boldsymbol{A} \begin{pmatrix} \boldsymbol{Z} \\ \boldsymbol{W}_o \end{pmatrix}$$

and therefore also defines a Gaussian vector (as a linear transform of a Gaussian vector). Its covariance matrix is given by

$$\mathrm{Var}[\tilde{\boldsymbol{Z}}] = \boldsymbol{A}\mathrm{Var}\left[\begin{pmatrix} \boldsymbol{Z} \\ \boldsymbol{W}_o \end{pmatrix}\right] \boldsymbol{A}^T = \begin{pmatrix} f(\boldsymbol{S}) & f(\boldsymbol{S})\boldsymbol{M}_o^T \\ \boldsymbol{M}_o f(\boldsymbol{S}) & \boldsymbol{M}_o f(\boldsymbol{S}) \boldsymbol{M}_o^T + \tau^2 \boldsymbol{I}_d \end{pmatrix} \quad . \tag{C.1}$$

Hence, the conditional distribution of $[\boldsymbol{Z}|\boldsymbol{Z}_o = \boldsymbol{z}_o]$ can be seen as the conditional distribution of a subset of components of $\tilde{\boldsymbol{Z}}$ given its remaining components. Hence (cf. Proposition A.4.10),

$$[\boldsymbol{Z}|\boldsymbol{Z}_o = \boldsymbol{z}_o] \sim \mathcal{N}\left(\mathbb{E}[\boldsymbol{Z}|\boldsymbol{z}_o], \mathrm{Var}[\boldsymbol{Z}|\boldsymbol{z}_o]\right) \quad ,$$

where $\mathbb{E}[\boldsymbol{Z}|\boldsymbol{z}_o]$, $\mathrm{Var}[\boldsymbol{Z}|\boldsymbol{z}_o]$ are given by Equations (4.4) and (4.5).

Finally, whenever f is non-zero on the set of eigenvalues of $\boldsymbol{S}$ and $\tau > 0$, the precision matrix of $\tilde{\boldsymbol{Z}}$, is well-defined and given by

$$\mathrm{Var}[\tilde{\boldsymbol{Z}}]^{-1} = \left(\boldsymbol{A}^{-1}\right)^T \begin{pmatrix} (1/f)(\boldsymbol{S}) & \\ & \boldsymbol{I}_d \end{pmatrix} \boldsymbol{A}^{-1} \quad \text{where} \quad \boldsymbol{A}^{-1} = \begin{pmatrix} \boldsymbol{I}_n & \boldsymbol{0}_{n,d} \\ -\tau^{-1}\boldsymbol{M}_o & \tau^{-1}\boldsymbol{I}_d \end{pmatrix} \quad .$$

This gives

$$\begin{aligned} \mathrm{Var}[\tilde{\boldsymbol{Z}}]^{-1} &= \begin{pmatrix} (1/f)(\boldsymbol{S}) + \tau^{-2}\boldsymbol{M}_o^T\boldsymbol{M}_o & -\tau^{-2}\boldsymbol{M}_o^T \\ -\tau^{-2}\boldsymbol{M}_o & \tau^{-2}\boldsymbol{I}_d \end{pmatrix} \\ &= \tau^{-2} \begin{pmatrix} (\tau^2/f)(\boldsymbol{S}) + \boldsymbol{M}_o^T\boldsymbol{M}_o & -\boldsymbol{M}_o^T \\ -\boldsymbol{M}_o & \boldsymbol{I}_d \end{pmatrix} \quad , \end{aligned} \tag{C.2}$$

from which we deduce Equations (4.6) and (4.7) (cf. Proposition A.4.10).

□

Proposition 4.1.2

Proof. Let $i \in [\![1, n]\!]$ and let $\boldsymbol{k} \in \mathbb{R}^d$. Let z_i^* be a estimator of Z_i (the i-th entry of $\boldsymbol{Z}$) defined by $z_i^* = \boldsymbol{k}^T \boldsymbol{z}_o$. Let Z_i^* be the randomization of z_i^*, defined by $Z_i^* = \boldsymbol{k}^T \boldsymbol{Z}_o$.

The estimation error $Z_i^* - Z_i$ has mean zero: indeed, $\mathbb{E}[Z_i^* - Z_i] = \boldsymbol{k}^T\mathbb{E}[\boldsymbol{Z}_o] - \mathbb{E}[Z_i] = 0$. Using the fact that $\boldsymbol{Z}$ and $\boldsymbol{W}_o$ are independent, its variance can be seen as a function v :

$\mathbb{R}^n \to \mathbb{R}$ of $\boldsymbol{k}$ given by

$$\begin{aligned} v(\boldsymbol{k}) &= \mathrm{Var}[Z_i^* - Z_i] = \mathrm{Var}[\boldsymbol{k}^T(\boldsymbol{M}_o\boldsymbol{Z} + \tau\boldsymbol{W}_o) - Z_i] \\ &= \mathrm{Var}[\boldsymbol{k}^T\boldsymbol{M}_o\boldsymbol{Z}] + \mathrm{Var}[\tau\boldsymbol{k}^T\boldsymbol{W}_o] + \mathrm{Var}[Z_i] - 2\mathrm{Cov}[\boldsymbol{k}^T\boldsymbol{M}_o\boldsymbol{Z}, Z_i] \\ &= \boldsymbol{k}^T\boldsymbol{M}_o\boldsymbol{\Sigma}\boldsymbol{M}_o^T\boldsymbol{k} + \tau^2\boldsymbol{k}^T\boldsymbol{k} - 2\boldsymbol{k}^T\boldsymbol{M}_o\boldsymbol{\sigma}_i + \mathrm{Var}[Z_i] \quad , \end{aligned}$$

where $\boldsymbol{\Sigma} = \mathrm{Var}[\boldsymbol{Z}] = f(\boldsymbol{S})$ and $\boldsymbol{\sigma}_i = \mathrm{Cov}(\boldsymbol{Z}, Z_i)$ is also the i-th column of $\boldsymbol{\Sigma}$. Finding the set of vectors $\boldsymbol{k}$ that minimize $v(\boldsymbol{k})$ is then done by finding a stationary point $\boldsymbol{k}_i$ of v:

$$\nabla v(\boldsymbol{k}_i) = 2\left(\boldsymbol{M}_o\boldsymbol{\Sigma}\boldsymbol{M}_o^T\boldsymbol{k}_i + \tau^2\boldsymbol{k}_i - \boldsymbol{M}_o\boldsymbol{\sigma}_i\right) = \boldsymbol{0} \quad ,$$

which gives $\boldsymbol{k}_i = (\boldsymbol{M}_o\boldsymbol{\Sigma}\boldsymbol{M}_o^T + \tau^2\boldsymbol{I}_d)^{-1}\boldsymbol{M}_o\boldsymbol{\sigma}_i$. As defined, $\boldsymbol{k}_i$ therefore minimizes the variance of the estimation error of Z_i by $\boldsymbol{k}_i^T\boldsymbol{Z}_o$.

Let $\boldsymbol{K}$ be the $n \times d$ matrix whose rows are the vectors $\boldsymbol{k}_i^T$. Then clearly,

$$\boldsymbol{K} = \boldsymbol{\Sigma}\boldsymbol{M}_o^T(\boldsymbol{M}_o\boldsymbol{\Sigma}\boldsymbol{M}_o^T + \tau^2\boldsymbol{I}_d)^{-1} \quad ,$$

and the estimator $\boldsymbol{Z}^* = \boldsymbol{K}\boldsymbol{Z}_o$ is the best linear estimator of $\boldsymbol{Z}$ given $\boldsymbol{Z}_o$. Finally, notice that $\boldsymbol{Z}^* = \mathbb{E}[\boldsymbol{Z}|\boldsymbol{Z}_o]$ according to Equation (4.4). □

Proposition 4.1.3

Proof. Using the linearity of the (conditional) expectation, Equation (4.8) gives

$$\mathbb{E}[\boldsymbol{Z}|\boldsymbol{z}_o] = \mathbb{E}[\boldsymbol{Y} + m\boldsymbol{v}|\boldsymbol{M}_o(\boldsymbol{Y} + m\boldsymbol{v}) + \tau\boldsymbol{W}_o = \boldsymbol{z}_o] = m\boldsymbol{v} + \mathbb{E}[\boldsymbol{Y}|\boldsymbol{Y}_o = \boldsymbol{y}_o] \quad ,$$

where $\boldsymbol{Y} = \boldsymbol{Z} - m\boldsymbol{v}$ is a zero-mean $\boldsymbol{S}$-stationary SGS and the vectors $\boldsymbol{Y}_o$ and $\boldsymbol{y}_o$ are defined by

$$\boldsymbol{y}_o = \boldsymbol{z}_o - m\boldsymbol{M}_o\boldsymbol{v}, \quad \text{and} \quad \boldsymbol{Y}_o = \boldsymbol{Z}_o - m\boldsymbol{M}_o\boldsymbol{v} = \boldsymbol{M}_o\boldsymbol{Y} + \tau\boldsymbol{W}_o \quad . \tag{C.3}$$

Note in particular that $\boldsymbol{Y}$, $\boldsymbol{Y}_o$ and $\boldsymbol{y}_o$ follow all the requirements of Propositions 4.1.1 and 4.1.2. Hence, the best unbiased linear estimator of $\boldsymbol{Y}$ given $\boldsymbol{y}_o$ is

$$\boldsymbol{y}^* = \mathbb{E}[\boldsymbol{Y}|\boldsymbol{y}_o] = f(\boldsymbol{S})\boldsymbol{M}_o^T\left(\boldsymbol{M}_o f(\boldsymbol{S})\boldsymbol{M}_o^T + \tau^2\boldsymbol{I}_d\right)^{-1}\boldsymbol{y}_o \quad ,$$

and therefore, $\mathbb{E}[\boldsymbol{Z}|\boldsymbol{z}_o]$ satisfies Equation (4.9).

In the case where f is non-zero on the set of eigenvalues of $\boldsymbol{S}$ and $\tau > 0$, the same arguments gives Equation (4.10) as a consequence of Equation (4.6) in Proposition 4.1.1.

Let $\boldsymbol{Z}^* = \mathbb{E}[\boldsymbol{Z}|\boldsymbol{Z}_o]$. We now show that $\boldsymbol{Z}^*$ is indeed the BLUE of $\boldsymbol{Z}$ given $\boldsymbol{Z}_o$. Clearly, $\boldsymbol{Z}^*$ is a linear estimator. Besides, $\mathbb{E}[\boldsymbol{Z}^* - \boldsymbol{Z}] = \mathbb{E}[m\boldsymbol{v} + \boldsymbol{Y}^* - (\boldsymbol{Y} + m\boldsymbol{v})] = \mathbb{E}[\boldsymbol{Y}^* - \boldsymbol{Y}] = \boldsymbol{0}$ which proves the unbiasedness. Similarly, note that $\mathrm{Var}[\boldsymbol{Z}^* - \boldsymbol{Z}] = \mathrm{Var}[\boldsymbol{Y}^* - \boldsymbol{Y}]$ and therefore is minimal by definition of $\boldsymbol{Y}^*$. □

Proposition 4.1.4

Proof. Let $z_i^* = \boldsymbol{k}^T\boldsymbol{z}_o$ be a linear estimator of Z_i by $\boldsymbol{z}$ and let $Z_i^* = \boldsymbol{k}^T\boldsymbol{z}_o$ be the associated randomized estimator. Following Equations (4.2) and (4.8), the unbiasedness requirement now writes:

$$\mathbb{E}[Z_i^* - Z_i] = \mathbb{E}[Z_i^*] - \mathbb{E}[Z_i] = \boldsymbol{k}^T\mathbb{E}[\boldsymbol{Z}_o] - \mathbb{E}[Z_i] = \boldsymbol{k}^T\mathbb{E}[\boldsymbol{M}_o(\boldsymbol{Y} + m\boldsymbol{v}) + \tau\boldsymbol{W}_o] - mv_i = 0 \quad ,$$

which gives the relation:

$$m(\boldsymbol{k}^T\boldsymbol{M}_o\boldsymbol{v} - v_i) = 0 \quad .$$

Given that m is unknown, this relation should hold whatever the actual value of m to guarantee that Z_i^* is an unbiased estimator. This is achieved by imposing the following constraint on the vector of coefficients $\boldsymbol{k} \in \mathbb{R}^d$:

$$\boldsymbol{k}^T\boldsymbol{M}_o\boldsymbol{v} = v_i \quad . \tag{C.4}$$

Hence, the variance minimization requirement now becomes a constrained minimization problem: finding the vector $\boldsymbol{k}$ that minimizes the estimation error $\mathrm{Var}[Z_i^* - Z_i]$ under the constraint of Equation (C.4). Note in paricular that,

$$\begin{aligned}\mathrm{Var}[Z_i^* - Z_i] &= \mathrm{Var}[\boldsymbol{k}^T(\boldsymbol{M}_o\boldsymbol{Z} + \tau\boldsymbol{W}_o) - Z_i] \\ &= \mathrm{Var}[\boldsymbol{k}^T\boldsymbol{Y}_o - Y_i + m\boldsymbol{k}^T\boldsymbol{M}_o\boldsymbol{v} - mv_i] = \mathrm{Var}[\boldsymbol{k}^T\boldsymbol{Y}_o - Y_i] \quad ,\end{aligned}$$

where $\boldsymbol{Y}$ and $\boldsymbol{Y}_o$ are defined as in Equations (4.8) and (C.3). Noting that $\boldsymbol{Y}$ and $\boldsymbol{Y}_o$ follow the requirements of Proposition 4.1.2, we can conclude that $\mathrm{Var}[Z_i^* - Z] = \mathrm{Var}[Y_i^* - Y_i]$ can be computed using the same formula as the one derived in the proof of Proposition 4.1.2.

Hence, the vector $\boldsymbol{k}_i$ defining the BLUE of Z_i by $\boldsymbol{z}_o$ is

$$\boldsymbol{k}_i = \underset{\substack{\boldsymbol{k}\in\mathbb{R}^d \\ (\boldsymbol{M}_o\boldsymbol{v})^T\boldsymbol{k}=v_i}}{\mathrm{argmin}} \quad \boldsymbol{k}^T(\boldsymbol{M}_o\boldsymbol{\Sigma}\boldsymbol{M}_o^T + \tau^2\boldsymbol{I}_d)\boldsymbol{k} - 2\boldsymbol{k}^T\boldsymbol{M}_o\boldsymbol{\sigma}_i \quad , \tag{C.5}$$

where $\boldsymbol{\Sigma} = \mathrm{Var}[\boldsymbol{Z}] = f(\boldsymbol{S})$ and $\boldsymbol{\sigma}_i = \mathrm{Cov}(\boldsymbol{Z}, Z_i)$ is the i-th column of $\boldsymbol{\Sigma}$. This minimization problem can be solved using a Lagrange multiplier to enforce the constraint. Namely, if $\boldsymbol{k}_i$ is given by Equation (C.5), then there exists some $\mu_i \in \mathbb{R}$ such that $(\boldsymbol{k}_i, \mu_i)$ is a stationary point of the Lagrange function $\mathcal{L}$ defined by

$$\mathcal{L}(\boldsymbol{k}, \mu) = \boldsymbol{k}^T(\boldsymbol{M}_o\boldsymbol{\Sigma}\boldsymbol{M}_o^T + \tau^2\boldsymbol{I}_d)\boldsymbol{k} - 2\boldsymbol{k}^T\boldsymbol{M}_o\boldsymbol{\sigma}_i + 2\mu\left((\boldsymbol{M}_o\boldsymbol{v})^T\boldsymbol{k} - v_i\right), \quad (\boldsymbol{k}, \mu) \in \mathbb{R}^d \times \mathbb{R} \, .$$

Requiring that $(\boldsymbol{k}_i, \mu_i)$ be a stationary point of this new objective function then gives the following equations (when taking the derivatives with respect to $\boldsymbol{k}$ and μ):

$$\begin{cases} 2((\boldsymbol{M}_o\boldsymbol{\Sigma}\boldsymbol{M}_o^T + \tau^2\boldsymbol{I}_d)\boldsymbol{k}_i - \boldsymbol{M}_o\boldsymbol{\sigma}_i + \mu_i\boldsymbol{M}_o\boldsymbol{v}) = \boldsymbol{0} \\ 2((\boldsymbol{M}_o\boldsymbol{v})^T\boldsymbol{k}_i - v_i) = 0 \end{cases} .$$

Hence $(\boldsymbol{k}_i, \mu_i)$ is the solution of the following linear system:

$$\left(\begin{array}{c|c} (\boldsymbol{M}_o\boldsymbol{\Sigma}\boldsymbol{M}_o^T + \tau^2\boldsymbol{I}_d) & \boldsymbol{M}_o\boldsymbol{v} \\ \hline (\boldsymbol{M}_o\boldsymbol{v})^T & 0 \end{array}\right)\left(\begin{array}{c} \boldsymbol{k}_i \\ \hline \mu_i \end{array}\right) = \left(\begin{array}{c} \boldsymbol{M}_o\boldsymbol{\sigma}_i \\ \hline v_i \end{array}\right) \quad . \tag{C.6}$$

All vectors $\boldsymbol{k}_i$, $i \in [\![1, n]\!]$ are now consolidated into one matrix $\boldsymbol{K}$ whose lines are the vectors $\boldsymbol{k}_i^T$, as done in the proof of Proposition 4.1.2, so that multiplying $\boldsymbol{K}$ by $\boldsymbol{z}_o$ yields a vector whose entries are BLUEs of the entries of $\boldsymbol{Z}$. Hence $\boldsymbol{Z}^* = \boldsymbol{K}\boldsymbol{z}_o$ will be the BLUE of $\boldsymbol{Z}$ given $\boldsymbol{z}_o$. Equation (C.6) can then be used to derive a linear system satisfied by $\boldsymbol{K}$:

$$\left(\begin{array}{c|c} (\boldsymbol{M}_o\boldsymbol{\Sigma}\boldsymbol{M}_o^T + \tau^2\boldsymbol{I}_d) & \boldsymbol{M}_o\boldsymbol{v} \\ \hline (\boldsymbol{M}_o\boldsymbol{v})^T & 0 \end{array}\right)\left(\begin{array}{c} \boldsymbol{K}^T \\ \hline \boldsymbol{\mu}^T \end{array}\right) = \left(\begin{array}{c} \boldsymbol{M}_o\boldsymbol{\Sigma} \\ \hline \boldsymbol{v}^T \end{array}\right) \quad , \tag{C.7}$$

where $\boldsymbol{\mu} = (\mu_1 \dots \mu_n)^T \in \mathbb{R}^n$ is a vector whose entries are the Lagrange multipliers of each vector $\boldsymbol{k}_i$. Hence,

$$\left(\begin{array}{c|c} \boldsymbol{K} & \boldsymbol{\mu} \end{array}\right) = \left(\begin{array}{c|c} \boldsymbol{\Sigma}\boldsymbol{M}_o^T & \boldsymbol{v} \end{array}\right)\left(\begin{array}{c|c} (\boldsymbol{M}_o\boldsymbol{\Sigma}\boldsymbol{M}_o^T + \tau^2\boldsymbol{I}_d) & \boldsymbol{M}_o\boldsymbol{v} \\ \hline (\boldsymbol{M}_o\boldsymbol{v})^T & 0 \end{array}\right)^{-1} .$$

Finally, Equation (4.11) is obtained by noticing that $\boldsymbol{Z}^* = \boldsymbol{K}\boldsymbol{z}_o$ can also be written $\boldsymbol{Z}^* = (\boldsymbol{K}|\boldsymbol{\mu})\left(\begin{smallmatrix}\boldsymbol{z}_o \\ 0\end{smallmatrix}\right)$.

□

Proposition 4.1.5

Proof. Following Equation (4.5), let $\boldsymbol{K}$ be the matrix defined by

$$\boldsymbol{K} = f(\boldsymbol{S})\boldsymbol{M}_o^T \left(\boldsymbol{M}_o f(\boldsymbol{S})\boldsymbol{M}_o^T + \tau^2 \boldsymbol{I}_d\right)^{-1} \quad .$$

Then, $\boldsymbol{Z} - \mathbb{E}[\boldsymbol{Z}|\boldsymbol{Z}_o] = \boldsymbol{Z} - \boldsymbol{K}\boldsymbol{Z}_o = (\boldsymbol{I} - \boldsymbol{K}\boldsymbol{M}_o)\boldsymbol{Z} + \tau\boldsymbol{K}\boldsymbol{W}_o$. Hence, $\mathbb{E}[\boldsymbol{Z}|\boldsymbol{Z}_o]$ is the sum of two independent Gaussian vectors. So it is a Gaussian vector whose mean is the sum of their means and whose covariance matrix is the sum of their covariance matrices (cf. Proposition A.4.8). This gives that $\boldsymbol{Z} - \mathbb{E}[\boldsymbol{Z}|\boldsymbol{Z}_o]$ is a zero-mean Gaussian vector with covariance matrix $\boldsymbol{\Sigma}$ given by

$$\boldsymbol{\Sigma} = (\boldsymbol{I} - \boldsymbol{K}\boldsymbol{M}_o)f(\boldsymbol{S})(\boldsymbol{I} - \boldsymbol{K}\boldsymbol{M}_o)^T + \tau^2 \boldsymbol{K}\boldsymbol{K}^T \quad .$$

Developing this last expression then gives $\boldsymbol{\Sigma} = \mathrm{Var}[\boldsymbol{Z}|\boldsymbol{z}_o]$. □

Proposition 4.2.1

Proof. We first consider that $\boldsymbol{Z}$ is zero-mean, i.e. $m = 0$. Equations (4.17) and (4.18) are proved in the exact same way as Equations (4.4) and (4.5) in Proposition 4.1.1, except that the vector $\tilde{\boldsymbol{Z}} = (\boldsymbol{Z} \quad \boldsymbol{Z}_o)^T$ now has covariance matrix:

$$\mathrm{Var}[\tilde{\boldsymbol{Z}}] = \begin{pmatrix} f(\boldsymbol{S}) & f(\boldsymbol{S})\boldsymbol{M}_o^T \\ \boldsymbol{M}_o f(\boldsymbol{S}) & \boldsymbol{M}_o f(\boldsymbol{S})\boldsymbol{M}_o^T + \sum_{k=1}^{p} \boldsymbol{M}_k f_k(\boldsymbol{S}_k)\boldsymbol{M}_k^T + \tau^2 \boldsymbol{I}_d \end{pmatrix} \quad .$$

This is a direct consequence of the fact that $\boldsymbol{Z}_o$ is a sum of independent Gaussian vectors (cf. Proposition A.4.8). Similarly, the fact that $\boldsymbol{Z}^*$ is the BLUE of $\boldsymbol{Z}$ by $\boldsymbol{z}_o$ is proved using the same reasoning as the one used for Proposition 4.1.2.

The case $m \neq 0$ is then proved in the exact same way Equation (4.9) is proved in Proposition 4.1.3. □

Proposition 4.2.2

Proof. Equation (4.19) is proved in the exact same way as Equation (4.6) in Proposition 4.1.1, except that the vector $\tilde{\boldsymbol{Z}}$ is now defined as:

$$\tilde{\boldsymbol{Z}} = \begin{pmatrix} \boldsymbol{Z} \\ \boldsymbol{Z}_1 \\ \vdots \\ \boldsymbol{Z}_p \\ \hline \boldsymbol{Z}_o \end{pmatrix} = \boldsymbol{A} \begin{pmatrix} \boldsymbol{Z} \\ \boldsymbol{Z}_1 \\ \vdots \\ \boldsymbol{Z}_p \\ \hline \boldsymbol{W}_o \end{pmatrix} \text{ where } \boldsymbol{A} = \left(\begin{array}{cccc|c} \boldsymbol{M}_o & & & & \\ & \boldsymbol{M}_1 & & & \\ & & \ddots & & \\ & & & \boldsymbol{M}_p & \\ \hline \boldsymbol{M}_o & \boldsymbol{M}_1 & \dots & \boldsymbol{M}_p & \tau\boldsymbol{I}_d \end{array}\right) .$$

□

Proposition 4.2.3

Proof. The same proof as the one used for Proposition 4.1.4 can be used to prove this result. Simply notice that the random vector $\tilde{\boldsymbol{W}}_o = \boldsymbol{M}_1\boldsymbol{Z}_1 + \cdots + \boldsymbol{M}_p\boldsymbol{Z}_p + \tau\boldsymbol{W}_o$ is a zero-mean Gaussian vector with covariance matrix $\tilde{\boldsymbol{\Sigma}} = \sum_{k=1}^{p} \boldsymbol{M}_k f_k(\boldsymbol{S}_k)\boldsymbol{M}_k^T + \tau^2\boldsymbol{I}_d$. Replacing $\boldsymbol{W}_o$ by $\tilde{\boldsymbol{W}}_o$ in the proof of Proposition 4.1.4 then gives the result. □

C.3 Chapter 5

Proposition 5.1.1

Proof. Following from the bounds of its Rayleigh quotients, any symmetric matrix $\boldsymbol{A}$ satisfies for any vector $\boldsymbol{v}$ of appropriate size $\boldsymbol{v}^T\boldsymbol{A}\boldsymbol{v} \in [\lambda_{\min}(\boldsymbol{A})\boldsymbol{v}^T\boldsymbol{v}, \lambda_{\max}(\boldsymbol{A})\boldsymbol{v}^T\boldsymbol{v}]$.
Note then that $\forall \boldsymbol{v} \in \mathbb{R}^d$, we have $\boldsymbol{v}^T\boldsymbol{\Sigma}\boldsymbol{v} = (\boldsymbol{M}_o\boldsymbol{v})^T f(\boldsymbol{S})(\boldsymbol{M}_o\boldsymbol{v}) + \tau^2\boldsymbol{v}^T\boldsymbol{v}$. Hence, on one hand,

$$\boldsymbol{v}^T\boldsymbol{\Sigma}\boldsymbol{v} \leq \lambda_{\max}(f(\boldsymbol{S}))(\boldsymbol{M}_o\boldsymbol{v})^T(\boldsymbol{M}_o\boldsymbol{v}) + \tau^2\boldsymbol{v}^T\boldsymbol{v} \leq (\lambda_{\max}(f(\boldsymbol{S}))\lambda_{\max}(\boldsymbol{M}_o^T\boldsymbol{M}_o) + \tau^2)\boldsymbol{v}^T\boldsymbol{v} \ ,$$

and on the other hand,

$$\boldsymbol{v}^T\boldsymbol{\Sigma}\boldsymbol{v} \geq \lambda_{\min}(f(\boldsymbol{S}))(\boldsymbol{M}_o\boldsymbol{v})^T(\boldsymbol{M}_o\boldsymbol{v}) + \tau^2\boldsymbol{v}^T\boldsymbol{v} \geq (\lambda_{\min}(f(\boldsymbol{S}))\lambda_{\min}(\boldsymbol{M}_o^T\boldsymbol{M}_o) + \tau^2)\boldsymbol{v}^T\boldsymbol{v} \ .$$

Let $l_{\min} = \lambda_{\min}(f(\boldsymbol{S}))\lambda_{\min}(\boldsymbol{M}_o^T\boldsymbol{M}_o) + \tau^2$ and $l_{\max} = \lambda_{\max}(f(\boldsymbol{S}))\lambda_{\max}(\boldsymbol{M}_o^T\boldsymbol{M}_o) + \tau^2$. Then the Rayleigh quotient $\mathcal{R}(\boldsymbol{\Sigma}, \boldsymbol{v})$ satisfies $\mathcal{R}(\boldsymbol{\Sigma}, \boldsymbol{v}) \in [l_{\min}, l_{\max}]$. Given that $\lambda_{\max}(\boldsymbol{\Sigma})$ (resp. $\lambda_{\min}(\boldsymbol{\Sigma})$) is the supremum (resp. infimum) of $\mathcal{R}(\boldsymbol{\Sigma}, \boldsymbol{v})$ over $\boldsymbol{v} \in \mathbb{R}^d$ with $\boldsymbol{v} \neq \boldsymbol{0}$, we therefore get $\lambda_{\max}(\boldsymbol{\Sigma}) \leq l_{\max}$ and $\lambda_{\min}(\boldsymbol{\Sigma}) \geq l_{\min}$.
Finally noting that following the definition of graph filters,

$$\lambda_{\min}(f(\boldsymbol{S})) \geq \min_{\lambda \in [\lambda_{\min}(\boldsymbol{S}), \lambda_{\max}(\boldsymbol{S})]} f(\lambda) \quad \text{and} \quad \lambda_{\max}(f(\boldsymbol{S})) \leq \max_{\lambda \in [\lambda_{\min}(\boldsymbol{S}), \lambda_{\max}(\boldsymbol{S})]} f(\lambda) \ ,$$

gives the result stated in the proposition. □

C.4 Chapter 6

Proposition 6.4.2

Proof. Let $\boldsymbol{p} = (p_1, \ldots, p_d)^T \in \partial\mathcal{M} \subset \mathbb{R}^d$. In particular $F(\boldsymbol{p}) = 0$ and the implicit function theorem ensures that there exists a neighborhood $\mathscr{V} \subset \mathbb{R}^{d-1}$ of $(p_1, \ldots, p_{d-1})^T \in \mathbb{R}^{d-1}$ and a map $\phi : \mathscr{V} \to \mathbb{R}$ satisfying:

$$p_d = \phi(p_1, \ldots, p_{d-1}) = \quad \text{and} \quad \forall \boldsymbol{y} \in \mathscr{V} \mapsto (\boldsymbol{y}, \phi(\boldsymbol{y}))^T \in \partial\mathcal{M} \quad .$$

Moreover, the (Cartesian) partial derivatives of ϕ (as a function of $\mathbb{R}^{d-1}$) satisfy

$$\forall \boldsymbol{x} \in V, \quad \partial_j\phi(\boldsymbol{x}) = -\frac{1}{\partial_d F(\boldsymbol{x}, \phi(\boldsymbol{x}))}\partial_j F(\boldsymbol{x}, \phi(\boldsymbol{x})), \quad 1 \leq j \leq d-1$$

A coordinate chart (U, x) between a neighborhood U of $\boldsymbol{p}$ in $\mathcal{M}$ and $V \times R_+$ can then be built by:

$$\forall \boldsymbol{q} \in U, \quad x(\boldsymbol{q}) = (x_1(\boldsymbol{q}) = q_1, \ldots, x_{d-1}(\boldsymbol{q}) = q_{d-1}, x_d(\boldsymbol{q}) = \epsilon(\phi(q_1, \ldots, q_{d-1}) - q_d)) \ , \quad ,$$

where $\epsilon \in \{\pm 1\}$ has its sign determined by whether perturbing positively the d-th coordinate of a point of $\partial\mathcal{M}$ near $\boldsymbol{p}$ pushes us outside of $\mathcal{M}$ or not. In particular, $\epsilon = \text{sign}(\partial_d F(\boldsymbol{p}))$. The inverse ψ of x on $x(U)$ is then given by

$$\forall \boldsymbol{q}' \in x(U), \quad \psi(\boldsymbol{q}') = \big(\psi_1(\boldsymbol{q}') = q'_1, \ldots, \psi_{d-1}(\boldsymbol{q}') = q'_{d-1}, \psi_d(\boldsymbol{q}') = \phi(q'_1, \ldots, q'_{d-1}) - \epsilon q'_d\big) \quad .$$

(U, x) is a coordinate chart containing $\boldsymbol{p}$ and such that $x_d(\boldsymbol{p}) = 0$. Hence,

$$T_{\boldsymbol{p}}\partial\mathcal{M} = \text{span}\left\{ \left.\frac{\partial}{\partial x_1}\right|_{\boldsymbol{p}}, \ldots, \left.\frac{\partial}{\partial x_{d-1}}\right|_{\boldsymbol{p}} \right\} \ .$$

Let $f \in \mathscr{C}^\infty(\mathcal{M})$. The action of one of these tangent vectors on f can be rewritten using the chain rule for $1 \leq j \leq d-1$ as

$$\left.\frac{\partial f}{\partial x_j}\right|_{\boldsymbol{p}} = \partial_j(f \circ \psi)(x(\boldsymbol{p})) = \sum_{k=1}^{d} \partial_k f|_{\psi(x(\boldsymbol{p}))}\partial_j\psi_k|_{x(\boldsymbol{p})} = \partial_j f|_{\boldsymbol{p}} + \partial_j\phi|_{x(\boldsymbol{p})}\partial_d f|_{\boldsymbol{p}}, \quad 1 \leq j \leq d-1 \ .$$

Hence these tangent vectors can be decomposed in the basis $\{\partial_1, \dots, \partial_d\}$ of the directional derivatives of the Cartesian coordinates of $\mathbb{R}^d$ as

$$\left.\frac{\partial}{\partial x_j}\right|_{\boldsymbol{p}} = \partial_j|_{\boldsymbol{p}} + \partial_j\phi|_{x(\boldsymbol{p})}\partial_d|_{\boldsymbol{p}} = \partial_j|_{\boldsymbol{p}} - \frac{1}{\partial_d F(\boldsymbol{p})}\partial_j F(\boldsymbol{p})\partial_d|_{\boldsymbol{p}}, \quad 1 \leq j \leq d-1 \quad .$$

Consider now the vector $v_{\boldsymbol{p}} \in T_{\boldsymbol{p}}\mathcal{M}$ defined as

$$v_{\boldsymbol{p}} = \sum_{k=1}^{d} \partial_k F(\boldsymbol{p})\partial_k|_{\boldsymbol{p}} \quad .$$

Then, for all $1 \leq j \leq d-1$, we have

$$\bar{g}_{\boldsymbol{p}}(v_{\boldsymbol{p}}, \left.\frac{\partial}{\partial x_j}\right|_{\boldsymbol{p}}) = \partial_j F(\boldsymbol{p}) \times 1 + \partial_d F(\boldsymbol{p}) \times \left(-\frac{1}{\partial_d F(\boldsymbol{p})}\partial_j F(\boldsymbol{p})\right) = 0 \quad .$$

Hence $v_{\boldsymbol{p}} \in T_{\boldsymbol{p}}\partial\mathcal{M}^{\perp}$.

Notice finally that the tangent vector $\partial/\partial x_d$ can also be decomposed using the Chain rule:

$$\left.\frac{\partial}{\partial x_d}\right|_{\boldsymbol{p}} = -\epsilon\partial_d|_{\boldsymbol{p}} \quad .$$

Hence,

$$\bar{g}_{\boldsymbol{p}}\left(v_{\boldsymbol{p}}, \left.\frac{\partial}{\partial x_d}\right|_{\boldsymbol{p}}\right) = -\epsilon\partial_d F(\boldsymbol{p}) < 0 \quad .$$

Hence the vector $v_{\boldsymbol{p}}/\sqrt{\bar{g}_{\boldsymbol{p}}(v_{\boldsymbol{p}}, v_{\boldsymbol{p}})} = n_{\boldsymbol{p}}$ satisfies all the requirements of a unit normal vector at $\boldsymbol{p}$.

□

D

Pseudo-differential operators and Laplacian

D.1 Laplacian and Fourier transform

D.1.1 Generalized random fields and pseudo-differential operators

We first consider the case $\mathcal{M} = [0, \pi]^d$. We assume that $\mathcal{M}$ is equipped with the Euclidean metric, making it a Riemannian manifold. The eigenvalues $\{\lambda_{\boldsymbol{k}}\}_{\boldsymbol{k}\in\mathbb{N}^d}$ and eigenfunctions $\{e_{\boldsymbol{k}}\}_{\boldsymbol{k}\in\mathbb{N}^d}$ of the Laplacian $-\Delta_{\mathcal{M}}$ on $\mathcal{M}$, when Dirichlet boundary conditions are considered, are given by (Grebenkov and Nguyen, 2013):

$$\forall \boldsymbol{k} \in \mathbb{N}^d, \quad \forall \boldsymbol{x} \in \mathcal{M}, \quad e_{\boldsymbol{k}}(\boldsymbol{x}) = \left(\frac{2}{\pi}\right)^d \prod_{l=1}^{d} \sin(k_l x_l), \quad \lambda_{\boldsymbol{k}} = \sum_{l=1}^{d} k_l^2 \tag{D.1}$$

Proposition D.1.1. *Let $f \in L^2(\mathcal{M})$ and define $\tilde{f} : \mathbb{R}^d \to \mathbb{R}$ by:*

$$\begin{cases} \tilde{f}(\boldsymbol{x}) = f(\boldsymbol{x}), & \forall \boldsymbol{x} \in [0, \pi]^d \\ \tilde{f}(x_1, \dots, -x_k, \dots, x_d) = -\tilde{f}(x_1, \dots, x_k, \dots, x_d) & \forall \boldsymbol{x} \in \mathbb{R}^d, 1 \leq k \leq d \\ \tilde{f}(\boldsymbol{x} + 2\pi\boldsymbol{n}) = \tilde{f}(\boldsymbol{x}), & \forall \boldsymbol{x} \in \mathbb{R}^d, \boldsymbol{n} \in \mathbb{Z}^d \end{cases} \tag{D.2}$$

Then the coefficients $c_{\boldsymbol{j}}(\tilde{f})$ of the Fourier series of $\tilde{f}$ satisfy $\forall \boldsymbol{j} \in \mathbb{Z}^d$:

$$c_{\boldsymbol{j}}(\tilde{f}) = \frac{1}{(2i)^d} \varepsilon(\boldsymbol{j}) \langle f, e_{|\boldsymbol{j}|} \rangle_{L^2(\mathcal{M})} \tag{D.3}$$

where $\varepsilon(\boldsymbol{j}) = \prod_{l=1}^{d} \operatorname{sign}(j_l)$, $|\boldsymbol{j}| := (|j_1| \dots |j_d|)^T \in \mathbb{N}^d$ and $e_{|\boldsymbol{j}|}$ is an eigenfunction of $-\Delta_{\mathcal{M}}$ on $\mathcal{M} = [0, \pi]^d$ with Dirichlet boundary conditions, as defined in (D.1)*. (Note :* sign *denotes the sign function* $\operatorname{sign} : x \in \mathbb{R} \mapsto 1$ *if* $x \geq 0$, -1 *otherwise)*
In particular, the Fourier series of $\tilde{f}$ (restricted to $[0, \pi]^d$) is equal (up to a normalization constant) to the development of f in the eigenbasis of the Laplacian.

Proof.

$$c_{\boldsymbol{j}}(\tilde{f}) = \frac{1}{(2\pi)^d} \int_{[-\pi,\pi]^d} e^{-i\boldsymbol{j}^T \boldsymbol{x}} \tilde{f}(\boldsymbol{x}) d\boldsymbol{x} = \frac{1}{(2\pi)^d} \int_{[-\pi,\pi]^{d-1}} e^{-i \sum_{k=1}^{d-1} j_l x_l} \int_{[-\pi,\pi]} e^{-i j_d x_d} \tilde{f}(\boldsymbol{x}) d\boldsymbol{x}$$

And,

$$\int_{[-\pi,\pi]} e^{-ij_d x_d}\tilde{f}(\boldsymbol{x})dx_d = \int_{[-\pi,0]} e^{-ij_d x_d}\tilde{f}(\boldsymbol{x})dx_d + \int_{[0,\pi]} e^{-ij_d x_d}\tilde{f}(\boldsymbol{x})dx_d$$
$$= \int_{[0,\pi]} (-e^{ij_d x_d} + e^{-ij_d x_d})\tilde{f}(\boldsymbol{x})dx_d = -2i\int_{[0,\pi]} \sin(j_d x_d)\tilde{f}(\boldsymbol{x})dx_d$$

So,

$$c_{\boldsymbol{j}}(\tilde{f}) = \frac{-2i}{(2\pi)^d}\int_{[-\pi,\pi]^{d-1}} e^{-i\sum_{l=1}^{d-1} j_l x_l}\int_{[0,\pi]} \sin(j_d x_d)\tilde{f}(\boldsymbol{x})d\boldsymbol{x}$$

By induction, the same process yields,

$$c_{\boldsymbol{j}}(\tilde{f}) = \frac{(-2i)^d}{(2\pi)^d}\int_{[0,\pi]^d}\prod_{l=1}^{d}\sin(j_l x_l)\tilde{f}(\boldsymbol{x})d\boldsymbol{x} = \frac{1}{(i\pi)^d}\int_{[0,\pi]^d}\prod_{l=1}^{d}\sin(j_l x_l)f(\boldsymbol{x})d\boldsymbol{x}$$
$$= \frac{1}{(i\pi)^d}\int_{[0,\pi]^d}\prod_{l=1}^{d}\sin(\mathrm{sign}(j_l)|j_l|x_l)f(\boldsymbol{x})d\boldsymbol{x} = \frac{1}{(i\pi)^d}\int_{[0,\pi]^d}\boldsymbol{\varepsilon}(\boldsymbol{j})\prod_{l=1}^{d}\sin(|j_l|x_l)f(\boldsymbol{x})d\boldsymbol{x}$$
$$= \frac{1}{(i\pi)^d}\boldsymbol{\varepsilon}(\boldsymbol{j})\int_{[0,\pi]^d}\left(\frac{\pi}{2}\right)^d e_{|\boldsymbol{j}|}(\boldsymbol{x})f(\boldsymbol{x})d\boldsymbol{x} = \frac{1}{(2i)^d}\boldsymbol{\varepsilon}(\boldsymbol{j})\langle f, e_{|\boldsymbol{j}|}\rangle_{L^2(\mathcal{M})}$$

Besides, since $f \in L^2(\mathcal{M})$, it can be decomposed in the orthonormal basis of eigenfunctions $\{e_{\boldsymbol{k}}\}_{\boldsymbol{k}\in\mathbb{N}^d}$ of the Laplacian $-\Delta_{\mathcal{M}}$. In particular,

$$f = \sum_{\boldsymbol{k}\in\mathbb{N}^d}\langle f, e_{\boldsymbol{k}}\rangle_{L^2(\mathcal{M})}e_{\boldsymbol{k}}$$

where the equality is understood in the L^2 sense. Note that using Euler's formula, it is quite straightforward to derive an alternative expression of the eigenfunctions $\{e_{\boldsymbol{k}}\}_{\boldsymbol{k}\in\mathbb{N}^d}$:

$$\forall \boldsymbol{k}\in\mathbb{N}^d, \forall \boldsymbol{x}\in\mathcal{M}, \quad e_{\boldsymbol{k}}(\boldsymbol{x}) = \left(\frac{2}{\pi}\right)^d\prod_{l=1}^{d}\sin(k_l x_l) = \left(\frac{2}{\pi}\right)^d\cdot\frac{1}{(2i)^d}\sum_{\boldsymbol{j}\in\mathbb{Z}^d:|\boldsymbol{j}|=\boldsymbol{k}}\boldsymbol{\varepsilon}(\boldsymbol{j})e^{i\boldsymbol{j}^T\boldsymbol{x}}$$

Therefore, $\forall \boldsymbol{x}\in\mathcal{M}$,

$$\sum_{\boldsymbol{k}\in\mathbb{N}^d}\langle f, e_{\boldsymbol{k}}\rangle_{L^2(\mathcal{M})}e_{\boldsymbol{k}}(\boldsymbol{x}) = \frac{1}{(i\pi)^d}\sum_{\boldsymbol{k}\in\mathbb{N}^d}\langle f, e_{\boldsymbol{k}}\rangle_{L^2(\mathcal{M})}\sum_{\boldsymbol{j}\in\mathbb{Z}^d:|\boldsymbol{j}|=\boldsymbol{k}}\boldsymbol{\varepsilon}(\boldsymbol{j})e^{i\boldsymbol{j}^T\boldsymbol{x}}$$
$$= \frac{1}{(i\pi)^d}\sum_{\boldsymbol{j}\in\mathbb{Z}^d}\boldsymbol{\varepsilon}(\boldsymbol{j})\langle f, e_{|\boldsymbol{j}|}\rangle_{L^2(\mathcal{M})}e^{i\boldsymbol{j}^T\boldsymbol{x}} = \frac{(2i)^d}{(i\pi)^d}\sum_{\boldsymbol{j}\in\mathbb{Z}^d}c_{\boldsymbol{j}}e^{i\boldsymbol{j}^T\boldsymbol{x}}$$
$$= \left(\frac{2}{\pi}\right)^d\mathcal{S}_F[\tilde{f}](\boldsymbol{x})$$

□

It is therefore possible to define the Fourier transform of a function of $f \in L^2(\mathcal{M})$ as the Fourier transform of its associated 2π-periodic function (of $\mathbb{R}^d$) $\tilde{f}$ defined as in (D.2). Using this convention, the Fourier transform $\mathscr{F}[f]$ of $f \in L^2(\mathcal{M})$ is the distribution given by:

$$\mathscr{F}[f] = (2\pi)^d\sum_{\boldsymbol{j}\in\mathbb{Z}^d}c_{\boldsymbol{j}}(\tilde{f})\delta_{\boldsymbol{j}} = (-i\pi)^d\sum_{\boldsymbol{j}\in\mathbb{Z}^d}\varepsilon(\boldsymbol{j})\langle f, e_{|\boldsymbol{j}|}\rangle_{L^2(\mathcal{M})}\delta_{\boldsymbol{j}} \tag{D.4}$$

Proposition D.1.2. *Let $\gamma : \mathbb{R}_+ \mapsto \mathbb{R}$.*
Let $\phi \in L^2(\mathcal{M})$ such that $\gamma(-\Delta_{\mathcal{M}})\phi \in L^2(\mathcal{M})$, where $\gamma(-\Delta_{\mathcal{M}})$ is the operator defined in Equation (7.1).

Then, the Fourier series representation $\gamma(-\Delta_{\mathcal{M}})\phi$ is equal $\mathscr{F}^{-1}\left[(\boldsymbol{w} \mapsto \gamma(\|\boldsymbol{w}\|^2)) \cdot \mathscr{F}[\phi]\right]$ in the sense of distributions, where $\mathscr{F}$ is the Fourier transform operator.

Proof. On one hand, by definition of $\gamma(-\Delta_{\mathcal{M}})$,

$$\gamma(-\Delta_{\mathcal{M}})\phi = \sum_{\boldsymbol{k}\in\mathbb{N}^d} \gamma(\lambda_{\boldsymbol{k}})\langle \phi, e_{\boldsymbol{k}}\rangle_{L^2(\mathcal{M})} e_{\boldsymbol{k}}$$

where $\lambda_{\boldsymbol{k}}$ and $e_{\boldsymbol{k}}$ are defined in Equation (D.1). Hence, following Equation (D.4),

$$\mathscr{F}\left[\gamma(-\Delta_{\mathcal{M}})\phi\right] = (-i\pi)^d \sum_{\boldsymbol{j}\in\mathbb{Z}^d} \varepsilon(\boldsymbol{j})\gamma(\lambda_{|\boldsymbol{j}|})\langle \phi, e_{|\boldsymbol{j}|}\rangle_{L^2(\mathcal{M})}\delta_{\boldsymbol{j}}$$

On the other hand, consider the distribution $T_{\gamma,\phi}$ defined as the product of the function $\boldsymbol{w} \mapsto \gamma(\|\boldsymbol{w}\|^2)$ with $\mathscr{F}[\phi]$, i.e.

$$\forall u \in \mathscr{C}_c^\infty(\mathbb{R}^d), \quad T_{\gamma,\phi}(u) := \mathscr{F}[\phi](\gamma \cdot u) = (-i\pi)^d \sum_{\boldsymbol{j}\in\mathbb{Z}^d} \varepsilon(\boldsymbol{j})\langle \phi, e_{|\boldsymbol{j}|}\rangle_{L^2(\mathcal{M})}\gamma(\|\boldsymbol{j}\|^2)u(\boldsymbol{j})$$

Then, by definition of the $\lambda_{\boldsymbol{j}}$,

$$\begin{aligned}\forall u \in \mathscr{C}_c^\infty(\mathbb{R}^d), \qquad T_{\gamma,\phi}(u) &= (-i\pi)^d \sum_{\boldsymbol{j}\in\mathbb{Z}^d} \varepsilon(\boldsymbol{j})\langle \phi, e_{|\boldsymbol{j}|}\rangle_{L^2(\mathcal{M})}\gamma(\lambda_{|\boldsymbol{j}|})u(\boldsymbol{j}) \\ &= \mathscr{F}\left[\gamma(-\Delta_{\mathcal{M}})\phi\right](u)\end{aligned}$$

Therefore, $T_{\gamma,\phi} = \mathscr{F}\left[\gamma(-\Delta_{\mathcal{M}})\phi\right]$ in the sense of distributions, which proves the result. □

Hence, in the particular case of the manifold $\mathcal{M} = [0,\pi]^d$, the operator $\gamma(-\Delta_{\mathcal{M}})$ acts on $L^2(\mathcal{M})$ exactly as a pseudo-differential operator.

D.2 Convergence of finite element approximations of generalized random fields

In this section, the proof of the convergence result of Theorem 8.2.1 is exposed. First, two lemmas used in the proof are introduced.

Lemma D.2.1. *Let $m \in \mathbb{R}$ such that $m \neq -1$ and let $n \in \mathbb{N}$, $n \geq 1$.*

$$\frac{1}{1+m}\left(1 - \frac{1}{n^{1+m}}\right) + \frac{1}{n^{\max\{1,m+1\}}} \leq \frac{1}{n}\sum_{k=1}^{n}\left(\frac{k}{n}\right)^m \leq \frac{1}{1+m}\left(1 - \frac{1}{n^{1+m}}\right) + \frac{1}{n^{\min\{1,m+1\}}}$$

Proof. Let $n \geq 1$ and let S_n denote the sum $S_n = \sum_{k=1}^{n}\left(\frac{k}{n}\right)^m$.
First, assume that $m \leq 0$. Then,

$$\forall k \in [\![1,n]\!], \forall t \in \left[\frac{k}{n}, \frac{k+1}{n}\right], \quad \left(\frac{k+1}{n}\right)^m \leq t^m \leq \left(\frac{k}{n}\right)^m$$

Integrating both inequalities between over the their segment of definition and then summing them for $1 \leq k \leq n-1$ gives:

$$\frac{1}{n}\left(S_n - \frac{1}{n^m}\right) \leq I_n \leq \frac{1}{n}(S_n - 1)$$

where

$$I_n = \int_{1/n}^{1} t^m dt = \frac{1}{1+m}\left(1 - \frac{1}{n^{1+m}}\right)$$

Hence, we have

$$I_n + \frac{1}{n} \leq \frac{1}{n} S_n \leq I_n + \frac{1}{n^{m+1}}$$

Similarly, if $m \geq 0$ we get,

$$I_n + \frac{1}{n^{m+1}} \leq \frac{1}{n} S_n \leq I_n + \frac{1}{n}$$

Hence, $\forall m \neq -1$, we have,

$$I_n + \frac{1}{n^{\max\{1,m+1\}}} \leq \frac{1}{n} S_n \leq I_n + \frac{1}{n^{\min\{1,m+1\}}}$$

□

Lemma D.2.2. *Let $m \in \mathbb{R}$ such that $m > 1$ and let $J \in \mathbb{N}$, $J \geq 1$.*

$$\frac{1}{(m-1)(J+1)^{m-1}} \leq \sum_{j>J} j^{-m} \leq \frac{1}{(m-1)J^{m-1}} \tag{D.5}$$

Proof. This result can easily be derived by upper-bounding and lower-bounding the integrals $\int_1^J t^{-m} dt$, $\int_1^{J+1} t^{-m} dt$ and $\int_{J+1}^{+\infty} t^{-m} dt$. □

The proof of Theorem 8.2.1 is now derived. The proof follows directly the approach outlined in the proof of Theorem 2.10 in (Bolin et al., 2018).

Proof. Let $\mathcal{Z}^{(n_h)}$ be the random field defined as the truncation of $\mathcal{Z}$ after n_h terms:

$$\mathcal{Z}^{(n_h)} = \sum_{k=1}^{n_h} W_j \gamma(\lambda_j) e_j \, .$$

Then, from the triangle inequality,

$$\|\mathcal{Z} - \mathcal{Z}_h\|_{L^2(\Omega;\mathcal{M})} \leq \|\mathcal{Z} - \mathcal{Z}^{(n_h)}\|_{L^2(\Omega;\mathcal{M})} + \|\mathcal{Z}^{(n_h)} - \mathcal{Z}_h\|_{L^2(\Omega;\mathcal{M})} \, . \tag{D.6}$$

Truncation error term $\|\mathcal{Z} - \mathcal{Z}^{(n_h)}\|_{L^2(\Omega;\mathcal{M})}$

$$\|\mathcal{Z} - \mathcal{Z}^{(n_h)}\|^2_{L^2(\Omega;\mathcal{M})} = \mathbb{E}\left[\| \sum_{j>n_h} W_j \gamma(\lambda_j) e_j \|^2_{L^2(\mathcal{M})}\right] = \mathbb{E}\left[\sum_{j>n_h} W_j^2 \gamma(\lambda_j)^2\right] = \sum_{j>n_h} \gamma(\lambda_j)^2 \, .$$

Then from Assumptions 8.1, 8.3 and 8.5 and Equation (D.5), we have

$$\|\mathcal{Z} - \mathcal{Z}^{(n_h)}\|^2_{L^2(\Omega;\mathcal{M})} \leq C_\gamma^2 \sum_{j>n_h} \lambda_j^{-2\beta} \leq C_\gamma^2 c_\lambda^{-2\beta} \sum_{j>n_h} j^{-2\alpha\beta} \leq \frac{C_\gamma^2 c_\lambda^{-2\beta}}{(2\alpha\beta - 1)} \times \frac{1}{n_h^{2\alpha\beta-1}} \, .$$

Finally, Assumption 8.4 yields

$$\|\mathcal{Z} - \mathcal{Z}^{(n_h)}\|^2_{L^2(\Omega;\mathcal{M})} \leq \frac{C_\gamma^2 c_\lambda^{-2\beta}}{(2\alpha\beta - 1) C_{\text{FES}}^{2\alpha\beta-1}} \times h^{\tilde{d}(2\alpha\beta-1)} \, . \tag{D.7}$$

Finite element discretization error $\|\mathcal{Z}^{(n_h)} - \mathcal{Z}_h\|_{L^2(\Omega;\mathcal{M})}$

From the triangular identity,

$$\begin{aligned}\|\mathcal{Z}^{(n_h)} - \mathcal{Z}_h\|_{L^2(\Omega;\mathcal{M})} &= \left\|\sum_{k=1}^{n_h} W_k\gamma(\lambda_k)e_k - \sum_{k=1}^{n_h} W_k\gamma(\lambda_{k,h})e_{j,h}\right\|_{L^2(\Omega;\mathcal{M})} \\ &\leq \underbrace{\left\|\sum_{k=1}^{n_h} W_k\gamma(\lambda_k)e_k - \sum_{k=1}^{n_h} W_k\gamma(\lambda_k)e_{k,h}\right\|_{L^2(\Omega;\mathcal{M})}}_{:=(\mathrm{I})} \\ &\quad + \underbrace{\left\|\sum_{k=1}^{n_h} W_k\gamma(\lambda_k)e_{k,h} - \sum_{k=1}^{n_h} W_k\gamma(\lambda_{k,h})e_{k,h}\right\|_{L^2(\Omega;\mathcal{M})}}_{:=(\mathrm{II})} .\end{aligned}$$

On one hand, using the independence of the weight $\{W_k\}_k$,

$$\begin{aligned}(\mathrm{I})^2 &= \left\|\sum_{k=1}^{n_h} W_k\gamma(\lambda_k)(e_k - e_{k,h})\right\|^2_{L^2(\Omega;\mathcal{M})} \\ &= \sum_{k=1}^{n_h}\sum_{l=1}^{n_h}\gamma(\lambda_k)\gamma(\lambda_l)\mathbb{E}\left[W_kW_l\right]\left\langle e_k - e_{k,h}, e_l - e_{l,h}\right\rangle_{L^2(\mathcal{M})} \\ &= \sum_{k=1}^{n_h}\gamma(\lambda_k)^2\|e_k - e_{k,h}\|^2_{L^2(\mathcal{M})} .\end{aligned}$$

So, following Assumption 8.6, $(\mathrm{I})^2 \leq C_2h^{2s}\sum_{k=1}^{n_h}\gamma(\lambda_k)^2\lambda_k^q$.

Let then K_0 be the integer defined by $K_0 = \left\lceil\left(\frac{R_\gamma}{c_\lambda}\right)^{1/\alpha}\right\rceil$. According to Assumptions 8.4 and 8.5, $n_h \geq K_0$. Hence, we can write $(\mathrm{I})^2 \leq C_2h^{2s}\left(S_0 + \sum_{k=K_0}^{n_h}\gamma(\lambda_k)^2\lambda_k^q\right)$ where S_0 is the constant defined by $S_0 = \sum_{k=1}^{K_0-1}\gamma(\lambda_k)^2\lambda_k^q$.

Finally, by definition of K_0 and according to Assumption 8.1, we have $k \geq K_0 \Rightarrow \lambda_k \geq R_\gamma$ and therefore, following Assumptions 8.1 and 8.3,

$$(\mathrm{I})^2 \leq C_2h^{2s}\left(S_0 + C_\gamma^2\sum_{k=K_0}^{n_h}\lambda_k^{-2\beta}\lambda_k^q\right) \leq C_2h^{2s}\left(S_0 + C_\gamma^2\widehat{C}_\lambda^{q-2\beta}\sum_{k=K_0}^{n_h}k^{\alpha(q-2\beta)}\right) , \qquad \text{(D.8)}$$

where $\widehat{C}_\lambda = C_\lambda$ if $q - 2\beta > 0$ and $\widehat{C}_\lambda = c_\lambda$ otherwise. Note in particular that:

$$\sum_{k=K_0}^{n_h}k^{\alpha(q-2\beta)} \leq \sum_{k=1}^{n_h}k^{\alpha(q-2\beta)} = n_h^{\alpha(q-2\beta)+1}\frac{1}{n_h}\sum_{k=1}^{n_h}\left(\frac{k}{n}\right)^{\alpha(q-2\beta)} .$$

If $\alpha(q-2\beta) > 0$, we have directly

$$\sum_{k=K_0}^{n_h}k^{\alpha(q-2\beta)} \leq n_h^{\alpha(q-2\beta)+1}\frac{1}{n_h}\sum_{k=1}^{n_h}1 = n_h^{1+\alpha(q-2\beta)} .$$

And if $\alpha(q-2\beta) \leq 0$, we use Lemma D.2.1 to derive

$$\sum_{k=K_0}^{n_h}k^{\alpha(q-2\beta)} \leq \frac{1}{1+\alpha(q-2\beta)}\left(n_h^{\alpha(q-2\beta)+1} - 1\right) + 1 .$$

Hence in both cases we can write

$$\sum_{k=K_0}^{n_h}k^{\alpha(q-2\beta)} \leq B_1n_h^{\alpha(q-2\beta)+1} + B_2 , \qquad \text{(D.9)}$$

where B_1 and B_2 are two constants depending solely on the parameters α, β, q. Injecting this last expression in Equation (D.8) anthen gives,

$$(\mathrm{I})^2 \leq C_2 h^{2s} \left(S_0 + C_\gamma^2 \widehat{C}_\lambda^{q-2\beta} B_2 + C_\gamma^2 \widehat{C}_\lambda^{q-2\beta} B_1 n_h^{\alpha(q-2\beta)+1} \right) .$$

And from Assumption 8.4 we get,

$$\begin{aligned}(\mathrm{I})^2 \leq & C_2 \left(S_0 + C_\gamma^2 \widehat{C}_\lambda^{q-2\beta} B_2 \right) \cdot h^{2s} \\ & + \left(C_2 C_\gamma^2 \widehat{C}_\lambda^{q-2\beta} B_1 C_{\mathrm{FES}}^{\alpha(q-2\beta)+1} \right) \cdot h^{(2s-\tilde{d}\alpha q)+\tilde{d}(2\alpha\beta-1)} .\end{aligned} \tag{D.10}$$

On the other hand,

$$\begin{aligned}(\mathrm{II})^2 &= \left\| \sum_{k=1}^{n_h} (\gamma(\lambda_k) - \gamma(\lambda_{k,h})) W_k e_{k,h} \right\|_{L^2(\Omega;\mathcal{M})} \\ &= \mathbb{E}\left[\sum_{k=1}^{n_h} (\gamma(\lambda_k) - \gamma(\lambda_{k,h}))^2 W_k^2 \right] = \sum_{k=1}^{n_h} (\gamma(\lambda_k) - \gamma(\lambda_{k,h}))^2 .\end{aligned} \tag{D.11}$$

In particular, using the mean value theorem, for all $1 \leq k \leq n_h$ there exists $l_k \in [\lambda_k, \lambda_{k,h}]$ such that:

$$\gamma(\lambda_k) - \gamma(\lambda_{k,h}) = \gamma'(l_k)(\lambda_{k,h} - \lambda_k) .$$

So, using Assumption 8.2,

$$|\gamma(\lambda_k) - \gamma(\lambda_{k,h})| = |\gamma'(l_k)||\lambda_{k,h} - \lambda_k| \leq \frac{C_{\mathrm{Deriv}}}{l_k^a} |\lambda_{k,h} - \lambda_k| \leq \frac{C_{\mathrm{Deriv}}}{\lambda_k^a} |\lambda_{k,h} - \lambda_k| .$$

And using Assumptions 8.1 and 8.6,

$$|\gamma(\lambda_k) - \gamma(\lambda_{k,h})| \leq \frac{C_{\mathrm{Deriv}}}{(c_\lambda k^\alpha)^a} C_1 h^r (C_\lambda k^\alpha)^q .$$

Therefore, injecting this last expression in Equation (D.11) gives

$$(\mathrm{II})^2 \leq \left(C_{\mathrm{Deriv}} c_\lambda^{-a} C_1 C_\lambda^q \right)^2 \cdot h^{2r} \sum_{k=1}^{n_h} k^{2\alpha(q-a)} .$$

Note that using the fact that $\alpha(q - a + \beta) \leq r/\tilde{d}$ (cf. Equation (8.23)), we have

$$\sum_{k=1}^{n_h} k^{2\alpha(q-a)} = \sum_{k=1}^{n_h} k^{2\alpha(q-a+\beta)} \cdot k^{-2\alpha\beta} \leq \sum_{k=1}^{n_h} k^{2r/\tilde{d}} \cdot k^{-2\alpha\beta} .$$

Using the same reasoning as the one used to derive Equation (D.9), we then get

$$\sum_{k=1}^{n_h} k^{2\alpha(q-a)} \leq B_1' n_h^{2r/\tilde{d}-2\alpha\beta+1} + B_2' ,$$

where B_1' and B_2' are two constants depending only on α, q, a. Injecting this observation in Equation (D.11) and using Assumption 8.4 finally gives

$$(\mathrm{II})^2 \leq \left(C_{\mathrm{Deriv}} c_\lambda^{-a} C_1 C_\lambda^q \right)^2 \left(B_2' \cdot h^{2r} + B_1' C_{\mathrm{FES}}^{2(r/\tilde{d}-\alpha\beta)+1} \cdot h^{2\tilde{d}\alpha\beta-\tilde{d}} \right) . \tag{D.12}$$

Combining the terms (I) and (II) finally gives:

$$\begin{aligned}\| \mathcal{Z}^{(n_h)} - \mathcal{Z}_h \|_{L^2(\Omega;\mathcal{M})} \leq & \sqrt{M_1 \cdot h^{2s} + M_2 \cdot h^{(2s-\tilde{d}\alpha q)+\tilde{d}(2\alpha\beta-1)}} \\ & + \sqrt{M_3 \cdot h^{2r} + M_4 \cdot h^{2\tilde{d}(\alpha\beta-1/2)}} ,\end{aligned} \tag{D.13}$$

where M_1, M_2, M_3, M_4 are constants independent of h.

Total error Using the fact that $h < 1$, the bounds in Equations (D.7) and (D.13) can actually be simplified by noticing that all the terms h^u $(u \geq 0)$ can be bounded by the one with the smallest exponent. This gives

$$\|\mathcal{Z} - \mathcal{Z}_h\|_{L^2(\Omega;\mathcal{M})} \leq M h^{\min\left\{s, s-\tilde{d}q\alpha/2+\tilde{d}(\alpha\beta-1/2), r, \tilde{d}(\alpha\beta-1/2)\right\}} ,$$

where M is a constant independent of h. And finally, using the fact that $s - \tilde{d}q\alpha/2 \geq 0$ (cf. Equation (8.23)) we have

$$\|\mathcal{Z} - \mathcal{Z}_h\|_{L^2(\Omega;\mathcal{M})} \leq M h^{\min\left\{s, r, \tilde{d}(\alpha\beta-1/2)\right\}} .$$

□

Bibliography

Abraham, R., Marsden, J. E., and Ratiu, T. (2012). *Manifolds, tensor analysis, and applications*, volume 75. Springer Science & Business Media.

Adler, R. J. and Taylor, J. E. (2009). *Random fields and geometry*. Springer Science & Business Media.

Anderes, E. B. and Stein, M. L. (2008). Estimating deformations of isotropic gaussian random fields on the plane. *The Annals of Statistics*, 36(2):719–741.

Anderes, E. B. and Stein, M. L. (2011). Local likelihood estimation for nonstationary random fields. *Journal of Multivariate Analysis*, 102(3):506–520.

Atkinson, K. E. (1989). *An introduction to numerical analysis (2nd Edition)*. John Wiley & Sons.

Aune, E., Eidsvik, J., and Pokern, Y. (2013). Iterative numerical methods for sampling from high dimensional gaussian distributions. *Statistics and Computing*, 23(4):501–521.

Avron, H. and Toledo, S. (2011). Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. *Journal of the ACM (JACM)*, 58(2):8.

Bérard, P. H. (2006). *Spectral geometry: direct and inverse problems*, volume 1207. Springer.

Bertsekas, D. P. (1997). Nonlinear programming. *Journal of the Operational Research Society*, 48(3):334–334.

Bolin, D., Kirchner, K., and Kovács, M. (2018). Numerical solution of fractional elliptic stochastic pdes with spatial white noise. *IMA Journal of Numerical Analysis*.

Bondy, J. A. and Murty, U. S. R. (1976). *Graph theory with applications*. Macmillan.

Bouclet, J.-M. (2012). An introduction to pseudo-differential operators. *Lecture Notes, Available at http://www. math. univ-toulouse. fr/˜ bouclet*.

Box, G. E., Hunter, J. S., and Hunter, W. G. (2005). Statistics for experimenters. In *Wiley Series in Probability and Statistics*. Wiley Hoboken, NJ, USA.

Brenner, S. and Scott, R. (2007). *The mathematical theory of finite element methods*, volume 15. Springer Science & Business Media.

Brutman, L. (1978). On the lebesgue function for polynomial interpolation. *SIAM Journal on Numerical Analysis*, 15(4):694–704.

Canzani, Y. (2013). Analysis on manifolds via the laplacian. *Lecture Notes available at: http://www. math. harvard. edu/canzani/docs/Laplacian. pdf*.

Carrizo Vergara, R. (2018). *Development of geostatistical models using Stochastic Partial Differential Equations*. PhD thesis.

Carrizo Vergara, R., Allard, D., and Desassis, N. (2018). A general framework for spde-based stationary random fields. *arXiv preprint arXiv:1806.04999*.

Chen, C. M. and Thomée, V. (1985). The lumped mass finite element method for a parabolic problem. *The ANZIAM Journal*, 26(3):329–354.

Chen, W.-H., Smith, C., and Fralick, S. (1977). A fast computational algorithm for the discrete cosine transform. *IEEE Transactions on communications*, 25(9):1004–1009.

Chilès, J.-P. and Delfiner, P. (2012). *Geostatistics : Modeling Spatial uncertainty. 2nd Edition.* Wiley Series In Probability and Statistics.

Cignoni, P., Callieri, M., Corsini, M., Dellepiane, M., Ganovelli, F., and Ranzuglia, G. (2008). Meshlab: an open-source mesh processing tool. The Eurographics Association.

Cooley, J. W. and Tukey, J. W. (1965). An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301.

Craioveanu, M.-E., Puta, M., and RASSIAS, T. (2013). *Old and new aspects in spectral geometry*, volume 534. Springer Science & Business Media.

Crane, K. (2019). Keenan's 3d model repository.

Del Corso, G. M., Menchi, O., and Romani, F. (2015). *Krylov subspace methods for solving linear systems.* Technical Report del Dipartimento di Informatica. Università di Pisa, Pisa, IT.

Demmel, J. W., Marques, O. A., Parlett, B. N., and Vömel, C. (2008). Performance and accuracy of lapack's symmetric tridiagonal eigensolvers. *SIAM Journal on Scientific Computing*, 30(3):1508–1526.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22.

Di Napoli, E., Polizzi, E., and Saad, Y. (2016). Efficient estimation of eigenvalue counts in an interval. *Numerical Linear Algebra with Applications*, 23(4):674–692.

Diggle, P. J., Tawn, J. A., and Moyeed, R. (1998). Model-based geostatistics. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 47(3):299–350.

Ehlich, H. and Zeller, K. (1966). Auswertung der normen von interpolationsoperatoren. *Mathematische Annalen*, 164(2):105–112.

Emery, X. and Porcu, E. (2019). Simulating isotropic vector-valued gaussian random fields on the sphere through finite harmonics approximations. *Stochastic Environmental Research and Risk Assessment*.

Erdős, P. (1961). Problems and results on the theory of interpolation. ii. *Acta Mathematica Hungarica*, 12(1-2):235–244.

Espinasse, T. (2011). *Champs et processus gaussiens indexés par des graphes, estimation et prédiction.* PhD thesis, Université de Toulouse, Université Toulouse III-Paul Sabatier.

Estrade, A., Fariñas, A., and Porcu, E. (2019). Covariance functions on spheres cross time: Beyond spatial isotropy and temporal stationarity. *Statistics & Probability Letters*, 151:1–7.

Fejér, L. (1910). Lebesguessche konstanten und divergente fourierreihen. *Journal für die reine und angewandte Mathematik*, 138:22–53.

Feller, W. (1971). *An introduction to probability theory and its applications*, volume 2. John Wiley & Sons, 2nd edition.

Fiala, Z. (2008). Geometry of finite deformations, linearization, and incremental deformations under initial stress/strain. In *Proceedings of International Conference Engineering Mechanics*, volume 20.

Fouedjio, F. (2017). Second-order non-stationary modeling approaches for univariate geostatistical data. *Stochastic environmental research and risk assessment*, 31(8):1887–1906.

Fouedjio, F., Desassis, N., and Romary, T. (2015). Estimation of space deformation model for non-stationary random functions. *Spatial Statistics*, 13:45–61.

Friedberg, S., Insel, A., and Spence, L. (2003). *Linear Algebra*. Featured Titles for Linear Algebra (Advanced) Series. Pearson Education.

Frommer, A. and Simoncini, V. (2008). Matrix functions. In *Model order reduction: theory, research aspects and applications*, pages 275–303. Springer.

Fuglstad, G.-A., Lindgren, F., Simpson, D., and Rue, H. (2015). Exploring a new class of non-stationary spatial gaussian random fields with varying local anisotropy. *Statistica Sinica*, pages 115–133.

Furrer, R., Genton, M. G., and Nychka, D. (2006). Covariance tapering for interpolation of large spatial datasets. *Journal of Computational and Graphical Statistics*, 15(3):502–523.

Gelb, A. and Gottlieb, S. (2007). The resolution of the gibbs phenomenon for fourier spectral methods. *Advances in The Gibbs Phenomenon. Sampling Publishing, Potsdam, New York*.

Gelfand, I. M. and Shilov, G. E. (1964). *Generalized functions, Vol. 4: applications of harmonic analysis*. Academic Press.

Gentle, J. E. (2009). *Computational statistics*, volume 308. Springer.

Gerschgorin, S. (1931). Uber die abgrenzung der eigenwerte einer matrix. *Izv. Akad. Nauk. USSR. Otd. Fiz-Mat. Nauk*, 7:749–754.

Girault, B. (2015a). *Signal processing on graphs-contributions to an emerging field*. PhD thesis, Ecole normale supérieure de lyon-ENS LYON.

Girault, B. (2015b). Stationary graph signals using an isometric graph translation. In *2015 23rd European Signal Processing Conference (EUSIPCO)*, pages 1516–1520. IEEE.

Gneiting, T. (2002). Compactly supported correlation functions. *Journal of Multivariate Analysis*, 83(2):493–508.

Gneiting, T. (2013). Strictly and non-strictly positive definite functions on spheres. *Bernoulli*, 19(4):1327–1349.

Golub, G. H. and Van Loan, C. F. (1996a). Matrix computations. *Johns Hopkins University*.

Golub, G. H. and Van Loan, C. F. (1996b). Matrix computations. 1996. *Johns Hopkins University, Press, Baltimore, MD, USA*, pages 374–426.

Grebenkov, D. S. and Nguyen, B.-T. (2013). Geometrical structure of laplacian eigenfunctions. *SIAM Review*, 55(4):601–667.

Hammond, D. K., Vandergheynst, P., and Gribonval, R. (2011). Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150.

Higdon, D., Swall, J., and Kern, J. (1999). Non-stationary spatial modeling. *Bayesian statistics*, 6(1):761–768.

Higham, N. J. (2008). *Functions of matrices: theory and computation*, volume 104. Siam.

Hoeber, H., Coléou, T., Le Meur, D., Angerer, E., Lanfranchi, P., and Lecerf, D. (2003). On the use of geostatistical filtering techniques in seismic processing. In *SEG Technical Program Expanded Abstracts 2003*, pages 2024–2027. Society of Exploration Geophysicists.

Huang, C., Zhang, H., and Robeson, S. M. (2011). On the validity of commonly used covariance and variogram functions on the sphere. *Mathematical Geosciences*, 43(6):721–733.

Huang, S., Quek, S., and Phoon, K. (2001). Convergence study of the truncated karhunen–loeve expansion for simulation of stochastic processes. *International journal for numerical methods in engineering*, 52(9):1029–1043.

Hutchinson, M. (1989). A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 18(3):1059–1076.

Jones, R. H. (1963). Stochastic processes on a sphere. *The Annals of mathematical statistics*, 34(1):213–218.

Jost, J. (2008). *Riemannian geometry and geometric analysis*, volume 42005. Springer.

Kaufman, C. G., Schervish, M. J., and Nychka, D. W. (2008). Covariance tapering for likelihood-based estimation in large spatial data sets. *Journal of the American Statistical Association*, 103(484):1545–1555.

Kruskal, J. B. (1964). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27.

Lablée, O. (2015). *Spectral Theory in Riemannian Geometry*. EMS textbooks in mathematics. European Mathematical Society.

Lanczos, C. (1988). *Applied analysis*. Courier Corporation.

Lang, A. (2007). *Simulation of stochastic partial differential equations and stochastic active contours*. PhD thesis, Universität Mannheim.

Lang, A. and Potthoff, J. (2011). Fast simulation of gaussian random fields. *Monte Carlo Methods and Applications*, 17(3):195–214.

Lang, A. and Schwab, C. (2015). Isotropic gaussian random fields on the sphere: regularity, fast simulation and stochastic partial differential equations. *The Annals of Applied Probability*, 25(6):3047–3094.

Lang, S. (2012). *Fundamentals of differential geometry*, volume 191. Springer Science & Business Media.

Lantuéjoul, C. (2013). *Geostatistical simulation: models and algorithms*. Springer Science & Business Media.

Lantuéjoul, C., Freulon, X., and Renard, D. (2019). Spectral simulation of isotropic gaussian random fields on a sphere. *Mathematical Geosciences*.

Lebanon, G. (2002). Learning riemannian metrics. In *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*, pages 362–369. Morgan Kaufmann Publishers Inc.

Lee, J. (2012). *Introduction to Smooth Manifolds*, volume 218. Springer Science & Business Media, 2nd edition.

Liang, M. and Marcotte, D. (2016). A class of non-stationary covariance functions with compact support. *Stochastic environmental research and risk assessment*, 30(3):973–987.

Lindgren, F., Rue, H., and Lindström, J. (2011). An explicit link between gaussian fields and gaussian markov random fields: the spde approach (with discussion). *JR 671 Stat Soc, Series B*, 73:423–498.

Lindgren, G. (2012). *Stationary stochastic processes: theory and applications*. Chapman and Hall/CRC.

Luce, R. and Ng, E. G. (2014). On the minimum flops problem in the sparse cholesky factorization. *SIAM Journal on Matrix Analysis and Applications*, 35(1):1–21.

Magneron, C., Bourges, M., and Jeannee, N. (2009). M-factorial kriging for seismic data noise attenuation. In *11th International Congress of the Brazilian Geophysical Society & EXPOGEF 2009, Salvador, Bahia, Brazil, 24-28 August 2009*, pages 1651–1654. Society of Exploration Geophysicists and Brazilian Geophysical Society.

Makhoul, J. (1980). A fast cosine transform in one and two dimensions. *IEEE Transactions on Acoustics Speech and Signal Processing*, 28(1):27–34.

Marinucci, D. and Peccati, G. (2011). *Random fields on the sphere: representation, limit theorems and cosmological applications*, volume 389. Cambridge University Press.

Marques, A. G., Segarra, S., Leus, G., and Ribeiro, A. (2017). Stationary graph processes and spectral estimation. *IEEE Transactions on Signal Processing*, 65(22):5911–5926.

Mason, J. C. and Handscomb, D. C. (2002). *Chebyshev polynomials*. CRC Press.

Matheron, G. (1971). The theory of regionalized variables and its applications, vol. 5. *Paris: École National Supérieure des Mines*, 211.

Matheron, G. (1982). Pour une analyse krigeante des données régionalisées. *Centre de Géostatistique, Report N-732, Fontainebleau*.

Melville, J. (2019). *mize: Unconstrained Numerical Optimization Algorithms*. R package version 0.2.2.

Musco, C., Musco, C., and Sidford, A. (2017). Stability of the Lanczos Method for Matrix Function Approximation. *arXiv*.

Newman, M. (2010). *Networks: An Introduction*. Oxford University Press.

Nocedal, J. and Wright, S. (2006). *Numerical optimization*. Springer Science & Business Media.

Oppenheim, A. V., Buck, J. R., and Schafer, R. W. (2001). *Discrete-time signal processing*. Upper Saddle River, NJ: Prentice Hall.

Ormerod, N. (1979). A theorem on fourier transforms of radial functions. *Journal of Mathematical Analysis and Applications*, 69(2):559 – 562.

Ortega, A., Frossard, P., Kovačević, J., Moura, J. M., and Vandergheynst, P. (2018). Graph signal processing: Overview, challenges, and applications. *Proceedings of the IEEE*, 106(5):808–828.

Osborne, A. R. (2010). Multidimensional fourier series. In Osborne, A. R., editor, *Nonlinear Ocean Waves and the Inverse Scattering Transform*, volume 97 of *International Geophysics*, pages 115 – 145. Academic Press.

Paciorek, C. J. and Schervish, M. J. (2006). Spatial modelling using a new class of nonstationary covariance functions. *Environmetrics: The official journal of the International Environmetrics Society*, 17(5):483–506.

Parzen, E. (1999). *Stochastic processes*, volume 24. SIAM.

Peltonen, J., Klami, A., and Kaski, S. (2004). Improved learning of riemannian metrics for exploratory analysis. *Neural Networks*, 17(8-9):1087–1100.

Perraudin, N. and Vandergheynst, P. (2017). Stationary signal processing on graphs. *IEEE Transactions on Signal Processing*, 65(13):3462–3477.

Perrin, O. and Meiring, W. (2003). Nonstationarity in $\mathbb{R}^n$ is second-order stationarity in $\mathbb{R}^{2n}$. *Journal of applied probability*, 40(3):815–820.

Perrin, O. and Monestiez, P. (1999). Modelling of non-stationary spatial structure using parametric radial basis deformations. In *geoENV II—Geostatistics for Environmental Applications*, pages 175–186. Springer.

Perrin, O. and Senoussi, R. (2000). Reducing non-stationary random fields to stationarity and isotropy using a space deformation. *Statistics & probability letters*, 48(1):23–32.

Petersen, K. B. and Pedersen, M. S. (2008). The matrix cookbook. *Technical University of Denmark*, 7(15):510.

Phillips, C. L., Parr, J. M., and Riskin, E. A. (2003). *Signals, systems, and transforms*. Prentice Hall Upper Saddle River.

Piazza, J., Magneron, C., Demongin, T., and Müller, N. (2015). M-factorial kriging-an efficient aid to noisy seismic data interpretation. In *Petroleum Geostatistics 2015*.

Pintore, A. and Holmes, C. C. (2004). Spatially adaptive non-stationary covariance functions via spa-tially adaptive spectra.

Porcu, E., Bevilacqua, M., and Genton, M. G. (2016). Spatio-temporal covariance and cross-covariance functions of the great circle distance on a sphere. *Journal of the American Statistical Association*, 111(514):888–898.

Porcu, E., Gregori, P., and Mateu, J. (2006). Nonseparable stationary anisotropic space-time covariance functions. *Stochastic Environmental Research and Risk Assessment*, 21(2):113–122.

Porcu, E., Mateu, J., and Bevilacqua, M. (2007). Covariance functions that are stationary or nonstationary in space and stationary in time. *Statistica Neerlandica*, 61(3):358–382.

Porcu, E., Mateu, J., and Christakos, G. (2009). Quasi-arithmetic means of covariance functions with potential applications to space–time data. *Journal of Multivariate Analysis*, 100(8):1830 – 1844.

Porcu, E., Matkowski, J., and Mateu, J. (2010). On the non-reducibility of non-stationary correlation functions to stationary ones under a class of mean-operator transformations. *Stochastic environmental research and risk assessment*, 24(5):599–610.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (2007). *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press.

Rahmat, R. and Harris-Birtill, D. (2018). Comparison of level set models in image segmentation. *IET Image Processing*, 12(12):2212–2221.

Raviart, P.-A., Thomas, J.-M., Ciarlet, P. G., and Lions, J. L. (1998). *Introduction à l'analyse numérique des équations aux dérivées partielles*, volume 2. Dunod Paris.

Rivlin, T. J. (1969). An introduction to the approximation of functions.

Rue, H. and Held, L. (2005). *Gaussian Markov random fields: theory and applications*. Chapman and Hall/CRC.

Saad, Y. (2003). *Iterative methods for sparse linear systems*, volume 82. siam.

Sampson, P. D. and Guttorp, P. (1992). Nonparametric estimation of nonstationary spatial covariance structure. *Journal of the American Statistical Association*, 87(417):108–119.

Saniee, K. (2008). A simple expression for multivariate lagrange interpolation.

Schönhage, A. (1961). Fehlerfortpflanzung bei interpolation. *Numerische Mathematik*, 3(1):62–71.

Shuman, D., Narang, S., Frossard, P., Ortega, A., and Vandergheynst, P. (2013). The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 3(30):83–98.

Simo, J. and Marsden, J. (1984). Stress tensors, riemannian metrics and the alternative descriptions in elasticity. In *Trends and Applications of Pure Mathematics to Mechanics*, pages 369–383. Springer.

Simpson, D. P., Turner, I. W., and Pettitt, A. N. (2008). Fast sampling from a gaussian markov random field using krylov subspace approaches. Technical report.

Snedecor, G. W. and Cochran, W. G. (1989). *Statistical methods. Eight Edition.* Iowa University Press.

Solin, A. and Särkkä, S. (2014). Hilbert space methods for reduced-rank gaussian process regression. *arXiv preprint arXiv:1401.5508*.

Stein, E. M. and Weiss, G. (1971). Introduction to fourier analysis on euclidean spaces (pms-32).

Stein, M. L. (2005). Nonstationary spatial covariance functions. *Technical report*.

Stein, M. L. (2012). *Interpolation of spatial data: some theory for kriging*. Springer Science & Business Media.

Stein, P. (1966). A note on the volume of a simplex. *The American Mathematical Monthly*, 73(3):299–301.

Stewart, W. J. (2009). *Probability, Markov chains, queues, and simulation: the mathematical basis of performance modeling*. Princeton university press.

Strang, G. and Fix, G. J. (1973). *An analysis of the finite element method*, volume 212. Prentice-hall Englewood Cliffs, NJ.

Sun, W. and Yuan, Y.-X. (2006). *Optimization theory and methods: nonlinear programming*, volume 1. Springer Science & Business Media.

Sun, Y., Li, B., and Genton, M. G. (2012). Geostatistics for large datasets. In *Advances and challenges in space-time modelling of natural events*, pages 55–77. Springer.

Susnjara, A., Perraudin, N., Kressner, D., and Vandergheynst, P. (2015). Accelerated filtering on graphs using lanczos method. *arXiv preprint arXiv:1509.04537*.

Thomas, J. W. (2013). *Numerical partial differential equations: finite difference methods*, volume 22. Springer Science & Business Media.

Tone, C. (2011). Central limit theorems for hilbert-space valued random fields satisfying a strong mixing condition. *ALEA*, 8:77–94.

Tong, Y. L. (2012). *The multivariate normal distribution*. Springer Science & Business Media.

Trefethen, L. N. (2013). *Approximation theory and approximation practice*, volume 128. Siam.

Trefethen, L. N. and Weideman, J. (1991). Two results on polynomial interpolation in equally spaced points. *Journal of Approximation Theory*, 65(3):247–260.

Turetskii, A. (1940). The bounding of polynomials prescribed at equally distributed points. In *Proc. Pedag. Inst. Vitebsk*, volume 3, pages 117–127.

Wackernagel, H. (2013). *Multivariate geostatistics: an introduction with applications*. Springer Science & Business Media.

Wei, G. C. and Tanner, M. A. (1990). A monte carlo implementation of the em algorithm and the poor man's data augmentation algorithms. *Journal of the American statistical Association*, 85(411):699–704.

Whittle, P. (1954). On stationary processes in the plane. *Biometrika*, pages 434–449.

Wilfred, K. (2002). Advanced calculus.

Ypma, J. (2018). *nloptr: R Interface to NLopt*. R package version 1.2.1.

RÉSUMÉ

La géostatistique est la branche des statistiques s'intéressant à la modélisation des phénomènes ancrés dans l'espace au travers de modèles probabilistes. En particulier, le phénomène en question est décrit par un champ aléatoire (généralement gaussien) et les données observées sont considérées comme résultant d'une réalisation particulière de ce champ aléatoire. Afin de faciliter la modélisation et les traitements géostatistiques qui en découlent, il est d'usage de supposer ce champ comme stationnaire et donc de supposer que la structuration spatiale des données se répète dans le domaine d'étude.
Cependant, lorsqu'on travaille avec des jeux de données spatialisées complexes, cette hypothèse devient inadaptée. En effet, comment définir cette notion de stationnarité lorsque les données sont indexées sur des domaines non euclidiens (comme des sphères ou autres surfaces lisses)? Quid également du cas où les données présentent structuration spatiale qui change manifestement d'un endroit à l'autre du domaine d'étude? En outre, opter pour des modèles plus complexes, lorsque cela est possible, s'accompagne en général d'une augmentation drastique des coûts opérationnels (calcul et mémoire), fermant alors la porte à leur application à de grands jeux de données.
Dans ce travail, nous proposons une solution à ces problèmes s'appuyant sur la définition de champs aléatoires généralisés sur des variétés riemanniennes. D'une part, travailler avec des champs aléatoires généralisés permet d'étendre naturellement des travaux récents s'attachant à tirer parti d'une caractérisation des champs aléatoires utilisés en géostatistique comme des solutions d'équations aux dérivées partielles stochastiques. D'autre part, travailler sur des variétés riemanniennes permet à la fois de définir des champs sur des domaines qui ne sont que localement euclidiens, et sur des domaines vus comme déformés localement (ouvrant donc la porte à la prise en compte du cas non stationnaire). Ces champs généralisés sont ensuite discrétisés en utilisant une approche par éléments finis, et nous en donnons une formule analytique pour une large classe de champs généralisés englobant les champs généralement utilisés dans les applications. Enfin, afin de résoudre le problème du passage à l'échelle pour les grands jeux de données, nous proposons des algorithmes inspirés du traitement du signal sur graphe permettant la simulation, la prédiction et l'inférence de ces champs par des approches "matrix-free".

MOTS CLÉS

Champ aléatoire généralisé, Variété riemannienne, Traitement du signal sur graphe, Equation aux dérivées partielles stochastique, Méthode des éléments finis

ABSTRACT

Geostatistics is the branch of statistics attached to model spatial phenomena through probabilistic models. In particular, the spatial phenomenon is described by a (generally Gaussian) random field, and the observed data are considered as resulting from a particular realization of this random field. To facilitate the modeling and the subsequent geostatistical operations applied to the data, the random field is usually assumed to be stationary, thus meaning that the spatial structure of the data replicates across the domain of study.
However, when dealing with complex spatial datasets, this assumption becomes ill-adapted. Indeed, how can the notion of stationarity be defined (and applied) when the data lie on non-Euclidean domains (such as spheres or other smooth surfaces)? Also, what about the case where the data clearly display a spatial structure that varies across the domain? Besides, using more complex models (when it is possible) generally comes at the price of a drastic increase in operational costs (computational and storage-wise), rendering them impossible to apply to large datasets.
In this work, we propose a solution to both problems, which relies on the definition of generalized random fields on Riemannian manifolds. On one hand, working with generalized random fields allows to naturally extend ongoing work that is done to leverage a characterization of random fields used in Geostatistics as solutions of stochastic partial differential equations. On the other hand, working on Riemannian manifolds allows to define such fields on both (only) locally Euclidean domains and on locally deformed spaces (thus yielding a framework to account for non-stationary cases). The discretization of these generalized random fields is undertaken using a finite element approach, and we provide an explicit formula for a large class of fields comprising those generally used in applications. Finally, to solve the scalability problem, we propose algorithms inspired from graph signal processing to tackle the simulation, the estimation and the inference of these fields using matrix-free approaches.

KEYWORDS

Generalized random field, Riemannian manifold, Graph signal processing, Stochastic partial differential equation, Finite element method