Second-order derivatives for shape optimization with a level-set method

Jean-Léopold Vie

UNIVERSITÉ PARIS-EST

L'École Doctorale Mathématiques et Sciences et Technologies de l'Information et de la Communication (MSTIC)

Thèse de Doctorat

Discipline : Mathématiques Appliquées

présentée par

Jean-Léopold Vié

Second-order derivatives for shape optimization with a level-set method

---

dirigée par Éric Cancès & Grégoire Allaire ,

soutenue le 16 décembre 2016 devant le jury composé de

Rapporteurs	Marc Dambrine	Université de Pau et des pays de l'adour
	Jean-Rodolphe Roche	Institut Élie Cartan de Lorraine
Directeurs de thèse	Éric Cancès	École des Ponts (CERMICS)
	Grégoire Allaire	École Polytechnique (CMAP)
Examinateurs	Marc Albertelli	Technocentre Renault
	Samuel Amstutz	Université d'Avignon et des pays du Vaucluse
	Virginie Ehrlacher	École des Ponts (CERMICS)
	Antoine Henrot	Institut Élie Cartan de Lorraine

Remerciements

Pour commencer, je profite de quelques lignes supplémentaires pour remercier, à défaut d'être exhaustif, nombre de ceux qui m'ont accompagné de près ou de loin pendant la réalisation de ce travail. Lors de cet exercice difficile mais ô combien important, la plume aura sûrement quelques maladresses qui, je l'espère ne terniront point son intention.

Comment pourrais-je ne pas commencer par remercier ceux qui m'ont encadré et dirigé pendant ces trois années ? Grégoire qui même convalescent n'a jamais rechigné à répondre à mes interrogations et à me faire profiter de ses connaissances dans le seul but que je fournisse un travail de qualité. Si son sens du détail m'a parfois poussé dans mes retranchements, je ne peux que lui en être reconnaissant au regard du résultat final. Éric, qui a su apporter un nouvel éclairage, n'a jamais hésité à tout remettre en question, et m'a faire suivre des voies que je n'aurais jamais imaginé pouvoir prendre. Sa disponibilité et sa pédagogie m'ont été extrêmement précieuses. Quant à Marc, dont j'admire l'humilité, il a non seulement accompagné mes premiers pas en optimisation mais a toujours mis son temps et ses connaissances à ma disposition. Un immense merci à tous les trois !

Je suis très honoré que Marc Dambrine et Jean-Rodolphe Roche, que je remercie chaleureusement, aient accepté de consacrer un peu de leur précieux temps à la relecture de ma thèse. Je suis également reconnaissant envers Samuel Amstutz, Virginie Ehrlacher et Antoine Henrot qui ont bien voulu prendre part au jury.

J'ai pu bénéficier au CERMICS, même si je n'y passais pas la plus grande partie de mon temps, d'une excellente ambiance de travail. J'ai pu y croiser beaucoup de personnes toujours disponibles pour répondre à quelque question que ce soit : Jean-François, Gabriel, Virginie, Laurent, Alexandre, Frédéric (sans oublier évidemment Éric), etc. Un grand merci aussi à tous les "camarades" thésards : David, William, Yannick, Laurent, Pierre, etc. Mille mercis à Axel pour les multiples fois où nous avons pu refaire le monde un certain nombre de fois, sur la terrasse ou ailleurs ! Je n'oublierai pas bien sûr Isabelle et Fatna qui jusqu'au bout m'auront aidé à régler tant de détails indispensables.

Le CMAP m'a également réservé un formidable accueil. Je ne dis pas cela uniquement pour la salle café dont j'ai fréquenté longuement le canapé généreusement légué par Matthieu, et où bon nombre de problèmes trouvent leurs solutions ! Je ne pourrai donc pas oublier de remercier Georgios, Thomas, Gwenaël, Aline et leurs successeurs en charge de la gestion du café. Merci à tous ceux que j'ai pu y croiser pour de riches discussions : François, Matthieu, Vianney, Frédéric, Xavier, Lucas, Luka, Thomas, Manon (et toute l'équipe du tournoi), Hélène, Aline, Etienne, Gustaw, Charline, Hadrien, Perle, etc. et surtout Antoine (dont je n'oublierai pas la banane !). Merci à toute l'équipe administrative, à Nasséra pour sa gentillesse, et bien sûr l'indispensable Sylvain !

Je remercie tout particulièrement Aymeric, presque mon "papa" du labo, que je n'ai eu cesse de bousculer dans le bureau ou de noyer de questions plus ou moins inutiles. Merci pour les souvenirs de Tivoli, des Karellis, de Crête ou d'ailleurs ! Mille mercis à Georgios (et son soldat Alexis !) et Charles, illustres prédécesseurs de l'équipe d'optimisation de formes ! Heureusement, pour la qualité de nos travaux respectifs, que nos thèses n'étaient pas simultanées... Merci à Gabriel et surtout Thomas toujours prêt à donner un coup de main pour débugguer !

Merci à toute l'équipe du projet RODIN : Damien, Philippe, Luca, François, Vassili, Axelle, Christian, Julien, etc. Même si mon passage est maintenant lointain j'ai passé d'excellents instants dans les équipes d'optimisation de Renault puis de l'IRT où j'ai pu initier la définition de mon sujet de thèse dans le cadre de ce même projet : les descentes de chocolat, les débats multiples sur les DS, etc. merci à CriCri, Pierre, Paul, Yves, Pascal, Fred, Timothée, Nathalie, etc.

Au-delà de la découverte californienne, je suis reconnaissant à Frédéric, Maxime, Arthur, Landry, et toute leur équipe de m'avoir fait découvrir le monde de la recherche !

Un immense merci à Bibi et M. Dubarry qui m'ont bien tendu la main, et à qui je dois tant, et bien plus que les nombreuses expressions : "C'est frizou !".

Il serait ingrat de ne point glisser un mot à l'égard du COP, duquel j'ai fait partie pendant la durée de la thèse, et de toutes les amitiés que j'ai pu y lier : Ferré (il faut rendre à César...), Zonzon, Baptiste, Marie, Guigui, Renaud, Christophe, Grégoire, Aude, Fabrice, Luc, Didier, Marc, Lauriane, etc.

Il me faut absolument remercier mes amis de longue et moins longue date. Le groupe "plâtalique" : François-Xavier qui ne me tînt (pas trop) rigueur de retards barbariens, Gautier, Augustin, Grite & Olivier (et les enfants), Bruno, Guillaume, Biket & Claire (et petit Gus), Axel, le cosaque, etc. Merci d'être là ! Philippe, Tim & Cam (et peTimophile), M & M (et Bout bien sûr), Steph et Pau, pour ces longues années de délires, de tournées et parfois de chant.

"Last but not least" dirait-on outre-manche, une pensée particulière aux membres de la famille. Merci à Papa pour l'intérêt que tu as pu porter à ce que j'entreprenais, ainsi que toutes les nombreuses discussions qui ont pu en découdre. Merci à Anne (ou Cécile) qui ne ménagea notamment pas ses visites pour me sortir de mon ermitage rédactionnel !

Merci Sklaerder de m'avoir indiqué les bons plans pour aller en confs... C'est toujours plus sympa de savoir où aller ! Quant à Maaariiie-Louise, merci de ce rire permanent qui t'anime ! Merci à Aïtita et Mamita, Caro, Thomas et Louis qui m'ont soutenu tant que possible. Merci de nous avoir prêté la maison cet été : les conditions de travail étaient plus que royales !!

Et surtout ma très chère Fred. Merci d'avoir supporté ce PC qui ne me quittait presque jamais, et les simus permanentes à surveiller. Merci d'avoir supporté les quelques nuits blanches et l'étalement des plages horaires de rédaction. Merci de m'avoir tout simplement supporté ! Merci de m'accompagner avec confiance (au Scilly ou même jusque dans l'Allier), et merci à Jacques à qui je dois bien sûr mes plus belles idées !

”réservé aux insensés”
H. Hesse

Résumé

Second-order derivatives for shape optimization with a level-set method

Résumé

Le but de cette thèse est de définir une méthode d'optimisation de formes qui conjugue l'utilisation de la dérivée seconde de forme et la méthode des lignes de niveaux pour la représentation d'une forme. On considèrera d'abord deux cas plus simples : un cas d'optimisation paramétrique et un cas d'optimisation discrète.

Ce travail est divisé en quatre parties. La première contient le matériel nécessaire à la compréhension de l'ensemble de la thèse. Le premier chapitre rappelle des résultats généraux d'optimisation, et notamment le fait que les méthodes d'ordre deux ont une convergence quadratique sous certaines hypothèses. Le deuxième chapitre répertorie différentes modélisations pour l'optimisation de formes, et le troisième se concentre sur l'optimisation paramétrique puis l'optimisation géométrique. Les quatrième et cinquième chapitres introduisent respectivement la méthode des lignes de niveaux (level-set) et la méthode des éléments-finis.

La deuxième partie commence par les chapitres 6 et 7 qui détaillent des calculs de dérivée seconde dans le cas de l'optimisation paramétrique puis géométrique. Ces chapitres précisent aussi la structure et certaines propriétés de la dérivée seconde de forme. Le huitième chapitre traite du cas de l'optimisation discrète. Dans le neuvième chapitre on introduit différentes méthodes pour un calcul approché de la dérivée seconde, puis on définit un algorithme de second ordre dans un cadre général. Cela donne la possibilité de faire quelques premières simulations numériques dans le cas de l'optimisation paramétrique (Chapitre 6) et dans le cas de l'optimisation discrète (Chapitre 7).

La troisième partie est consacrée à l'optimisation géométrique. Le dixième chapitre définit une nouvelle notion de dérivée de forme qui prend en compte le fait que l'évolution des formes par la méthode des lignes de niveaux, grâce à la résolution d'une équation eikonale, se fait toujours selon la normale. Cela permet de définir aussi une méthode d'ordre deux pour l'optimisation. Le onzième chapitre détaille l'approximation d'intégrales de surface et le douzième chapitre est consacré à des exemples numériques.

La dernière partie concerne l'analyse numérique d'algorithmes d'optimisation de formes par la méthode des lignes de niveaux. Le Chapitre 13 détaille la version discrète d'un algorithme d'optimisation de formes. Le Chapitre 14 analyse les schémas numériques relatifs à la méthodes des lignes de niveaux. Enfin le dernier chapitre fait l'analyse numérique complète d'un exemple d'optimisation de formes en dimension un, avec une étude des vitesses de convergence.

Mots-clefs

optimisation par la méthode de Newton, algorithme d'optimisation de second ordre, optimisation de formes, level-set, dérivée de forme, approximation par des schémas numériques.

Second-order derivatives for shape optimization with a level-set method

Abstract

The main purpose of this thesis is the definition of a shape optimization method which combines second-order differentiation with the representation of a shape by a level-set function. A second-order method is first designed for simple shape optimization problems : a thickness parametrization and a discrete optimization problem.

This work is divided in four parts. The first one is bibliographical and contains different necessary backgrounds for the rest of the work. Chapter 1 presents the classical results for general optimization and notably the quadratic rate of convergence of second-order methods in well-suited cases. Chapter 2 is a review of the different modelings for shape optimization while Chapter 3 details two particular modelings : the thickness parametrization and the geometric modeling. The level-set method is presented in Chapter 4 and Chapter 5 recalls the basics of the finite element method.

The second part opens with Chapter 6 and Chapter 7 which detail the calculation of second-order derivatives for the thickness parametrization and the geometric shape modeling. These chapters also focus on the particular structures of the second-order derivative. Then Chapter 8 is concerned with the computation of discrete derivatives for shape optimization. Finally Chapter 9 deals with different methods for approximating a second-order derivative and the definition of a second-order algorithm in a general modeling. It is also the occasion to make a few numerical experiments for the thickness (defined in Chapter 6) and the discrete (defined in Chapter 8) modelings.

Then, the third part is devoted to the geometric modeling for shape optimization. It starts with the definition of a new framework for shape differentiation in Chapter 10 and a resulting second-order method. This new framework for shape derivatives deals with normal evolutions of a shape given by an eikonal equation like in the level-set method. Chapter 11 is dedicated to the numerical computation of shape derivatives and Chapter 12 contains different numerical experiments.

Finally the last part of this work is about the numerical analysis of shape optimization algorithms based on the level-set method. Chapter 13 is concerned with a complete discretization of a shape optimization algorithm. Chapter 14 then analyses the numerical schemes for the level-set method, and the numerical error they may introduce. Finally Chapter 15 details completely a one-dimensional shape optimization example, with an error analysis on the rates of convergence.

Keywords

Newton optimization method, second-order optimization algorithm, shape optimization, level-set, shape derivative, numerical approximation schemes.

Contents

Introduction

This thesis is devoted to the design of a second-order algorithm for shape optimization problems when the shape is characterized by the level-set method. It takes place in the framework of the RODIN (Robust structural Optimization for Design in Industry) project [1]. This projects aims at developing a software for mechanical engineers.

For "classical" optimization problems, formulated for example as the minimization of a functional $J : \mathbb{R}^d \to \mathbb{R}$, second-order methods are known to converge in few iterations (compared to first-order methods) when the initial guess is not too far from the optimal solution, under some assumptions on the functional such as regularity and convexity. However, they require to compute and inverse the Hessian matrix. In the case of shape optimization, it is not clear that it is possible to compute the Hessian at a reasonable price, and to invert it. This raises the following question : is there any potential gain in designing a second-order method in shape optimization ? Providing (partial) answers to this question is one of the main goals of this work. The second-order derivatives are usually computationally expensive, and it is not obvious that the gain given by the second-order information can exceed the additional cost. However, before considering the ratio between the gain and additional cost of a second-order method we just want to evaluate how fast (in terms of number of iterations) such a method can be for shape optimization problems, without considering computational time issues. Then will come the question of computing only an approximation of the second-order derivative in order to save computational time.

The first step in evaluating the speed of convergence of a second-order method is to consider simple shape optimization problems. This is the reason why we shall consider the parametric case of thickness optimization, and the discrete approach. Then we move on to the general case when a shape is represented by a level-set function and the derivatives are computed in a continuous framework. This leads us to introduce a new framework for shape differentiation that takes into account the fact that when the motion of the shape is characterized by a Hamilton-Jacobi equation for the level-set function, the shape evolves along normal trajectories. This new framework for shape differentiation, which is one of the main theoretical achievements of the thesis, allows us to design a particular shape optimization algorithm and to apply it to different numerical examples.

Finally, the last topic of this work deals with the analysis of different numerical schemes. It provides some indications on the necessary ingredients to get a second-order rate of convergence.

Now we turn to a brief description of the different parts of the work, without any technical detail nor reference. We refer to the corresponding chapters for the bibliography.

Part I : Review of the state of the art

The first part of the thesis recalls the necessary background for the works detailed in the next following parts. The first chapter gives some usual results on general optimization in Banach and Hilbert spaces and explains the main interest of a second-order method. The second chapter is a review on shape optimization methods, while Chapter 3 gives some details on the computation of derivatives for the specific cases of thickness parametrization and shape variation frameworks. Then Chapter 4 is dedicated to the introduction of the level-set method and Chapter 5 focuses on the finite element method.

Chapter 1 : Optimization

The first chapter opens with general results on optimization in a Banach space $\mathcal{V}$. For a given optimization problem, it recalls the notion of global and local optima and some conditions that ensure the existence of such optima. The notion of Fréchet derivative allows one to give first and second-order optimality conditions. Then we can recall the usual gradient and Newton algorithms for solving an optimization problem in a Hilbert space. With sufficient regularity assumptions on the criterion to optimize, the Newton method is known to have a quadratic rate of convergence. If we denote by $(u_n)_{n\in\mathbb{N}}$ the iterative sequence obtained by the Newton method, and by $u^\star$ a local minimum to which the sequence converges, it means that when u^0 is close enough to $u^\star$, there exists $C > 0$ such that for $k \in \mathbb{N}$ sufficiently large,

$$\left\|u^{k+1} - u^\star\right\|_{\mathcal{V}} \leq C \left\|u^k - u^\star\right\|_{\mathcal{V}}^2 .$$

[1] This project is funded by the FUI (Fonds Unique Interministériel) and gather different partners : CMAP (École Polytechnique, LJLL (Université Pierre et Marie Curie), INRIA Bordeaux, Renault, Safran, Airbus, ESI-Group, Eurodecision, DPS, Alneos.

This implies that the Newton method converges to machine precision in few iterations. This is the main interest of second-order methods, and the main reason why considering them for shape optimization problems. The computation of the exact second-order derivative can be replaced by an approximation while keeping a super-linear rate of convergence. This is the main goal of the well-known BFGS method which is then presented with some of its properties. Finally this chapter introduces the *optimize-then-discretize* and *discretize-then-optimize* approaches for solving numerically an optimization problem. With the first approach, one approximates the exact derivatives while with the latter one computes exactly the derivatives of an approximated model.

Chapter 2 : Shape optimization methods

The second chapter starts with the introduction of different physical models that will be used for the different modelings and criteria. Then it reviews different usual modelings for shape optimization. They can be classified in three categories : parametric optimization, topology optimization and geometric optimization. In the parametric framework a shape is represented by a set of parameters such as curvatures, thickness, lengths, etc. The topology optimization methods intend to find the best repartition of holes in a shape. Then, geometric optimization considers the shape as a global variable seen as a bounded open set of $\mathbb{R}^d$. Finally, the last part of this chapter recalls that usually there is not existence of optimal shapes. It also gives some remedies to the non-existence of an optimal solution.

Chapter 3 : Examples of objective functions and optimality conditions

In the third chapter we focus on two modelings : the parametric case of thickness optimization and the geometric optimization framework. For the thickness optimization we give the first-order derivative of three criteria : the volume function, the compliance and a least square displacement criterion.

As concerns geometric optimization, we start by a brief review on previous work about second-order shape derivatives. Then we recall some useful theoretical tools of differential geometry. Computing derivatives in this framework requires to differentiate with respect to a shape. The notion of shape differentiation was first introduced by Hadamard. We recall the two usual frameworks for shape derivation with the Hadamard method : the displacement field method and the speed method. A shape is considered as a bounded open set of $\mathbb{R}^d$. With $\theta \in W^{1,\infty}(\mathbb{R}^d;\mathbb{R}^d)$ a variation of a reference shape Ω is defined by $\Omega_\theta = (\mathrm{Id}+\theta)(\Omega)$. Differentiating with respect to θ defines the shape derivative for the displacement field method. There is also the speed method : for a vector velocity field $V \in C^1\left(\mathbb{R}\times\mathbb{R}^d;\mathbb{R}^d\right)$ we can consider the solution to the following equation

$$\begin{cases} \dfrac{\partial X_V}{\partial t}(t,x) = V\big(t, X_V(t,x)\big), \\ X_V(0,x) = x, \end{cases} \tag{1}$$

for each $x \in \Omega$. This defines a time-dependent domain $\Omega_t = X_V(t,\Omega) = \{X_V(t,x),\ x\in\Omega\}$. Differentiating with respect to the time parameter leads to another notion of shape derivative. For both frameworks, the first-order shape derivative has the same structure, and in many practical cases it can be represented by a function $j \in L^1(\partial\Omega;\mathbb{R})$ such that it writes

$$\int_{\partial\Omega} (\theta\cdot\mathbf{n})\, j, \quad \text{or} \quad \int_{\partial\Omega} (V(0,\cdot)\cdot\mathbf{n})\, j.$$

The resulting structure of the first and second-order derivatives for these two frameworks are rather similar. The shape derivatives of a criterion are entirely characterized by a linear map (denoted by l_1) and a bilinear map (denoted by l_2) defined on the boundary of the shape. Finally we detail the linear map of the first-order derivative of different criteria for geometric optimization. We start by giving the derivative of a criterion that reads as the integral on the shape of a shape-independent function. That is for $f : \mathbb{R}^d \to \mathbb{R}$ sufficiently regular

$$E(\Omega) = \int_\Omega f(x)\, dx.$$

Then we consider the shape derivatives of geometric quantities such as the signed distance function, so that we can compute the shape derivatives of a maximum-thickness criterion. We also detail the shape derivatives of boundary conditions, and thus the shape derivative of a state equation. Thus we can give the shape derivative of a compliance and a least square displacement criteria.

The last part of this chapter is finally devoted to the question of regularity of an optimal shape for a simple problem of shape optimization in the two-dimensional case. When the compliance of the shape (for a given load case) is to be minimized with respect to a volume constraint in the two-dimensional case, it appears that the shape cannot have regular holes.

Chapter 4 : The level-set method

The fourth chapter is an introduction to the level-set method for shape modeling. It is very practical especially for tracking the motion of a shape. If a shape $\Omega \subset \mathbb{R}^d$ is represented by a level-set function ϕ, and evolves with a

normal velocity with speed $v \in C^1\left(\mathbb{R}^d;\mathbb{R}\right)$, the level-set function is solution to the following Hamilton-Jacobi advection equation

$$\partial_t \phi(t,x) + v(x)|\nabla_x \phi(t,x)| = 0.$$

We also briefly evoke the theoretical framework in which there is existence and uniqueness result for this kind of equation : namely the notion of *viscosity solutions*. Then we recall a numerical scheme for the resolution of this equation.

From a numerical point of view it is practical to consider level-set functions that are close to signed distance to a boundary. This is the goal of the redistanciation procedure detailed in the last part of this chapter. We recall three methods that can be used to ensure numerically that the level-set is close to a signed distance function during the resolution of the advection equation : the resolution of an additional transport equation, the fast-marching method, and a redistanciation-free method for the advection equation.

Chapter 5 : Finite element method

The first part ends with Chapter 5 which introduces the finite element method. It is well-known for approximating solutions to variational formulations of partial differential equations and boundary value problems. The underlying idea is to reduce the space where the variational formulation takes place to a finite-dimensional space, and to interpolate functions in this finite-dimensional space. We also give some error estimates on the interpolation error. Then we detail the computation of variational forms, the mass and stiffness matrices, for the $\mathbb{Q}_1$ finite-dimensional space on a square mesh. Finally, we also recall the mass-lumping method that is an approximation method for mass matrices.

Part II : Second-order derivatives and algorithms

The second part of the thesis is first devoted to the study of second-order derivatives in three different cases : the thickness parametrization, the geometric approach from a continuous point of view, and finally the geometric approach with the *discretize-then-optimize* approach. Then we define first and second-order optimization methods, with approximation or not of the second-order derivative. The underlying concepts are rather classical, but we detail here the computation of some second-order derivatives that are not easy to locate in the literature. We also give some properties of the second-order derivatives. Finally, we give a few original numerical examples in the parametric case of thickness optimization and for the *discretize-then-optimize* approach in geometric optimization.

Chapter 6 : Parametric case

This chapter focuses on the parametric case of thickness optimization. It starts with the computation of the second-order derivative of three criteria : the volume, the compliance and a least square displacement criterion. This is rather classical, even if it is hard to locate such results in the literature. It is also the opportunity to notice that the compliance criterion is convex (in this context). The rest of this chapter is devoted to the study of the structure of the second-order derivative of the compliance criterion, which is, to our knowledge, completely new. We prove that the operator corresponding to the second-order derivative is neither compact nor coercive in the scalar case as well as in the vector case. This implies that the matrix of the discretized Hessian is ill-conditioned. However we establish also in the scalar case that under regularity conditions, the Hessian of the compliance is positive definite.

Chapter 7 : Geometric case

Chapter 7 is somehow similar to Chapter 6. We first detail the computation of second-order derivatives for different criteria : the volume or any criterion that can be written as the integral on a varying domain of a shape-independent function, the compliance, a least square displacement criterion and a maximum-thickness criterion. The structure theorem for geometric shape optimization (recalled in Chapter 3) ensures that the second-order derivative is symmetric. However, when computing the derivatives, it is not clear that the obtained expressions are symmetric. Thus, this issue is is considered especially in the case of the compliance and the least square displacement criterion. Then, for the compliance criterion, we focus on the coercivity and the positiveness of the second-order derivative. Finally, we also write the Newton equation when the shape derivatives are computed in the framework of the displacement field method (also introduced in Chapter 3). In that case we show that the tangential components of the solution to the Newton equation are almost entirely determined by its normal component.

Chapter 8 : Geometric discretized case

Chapter 8 defines a discrete model for shape optimization with the level-set method. We compute derivatives of the discretized criteria (volume, compliance and least square displacement criterion). This allows us to introduce a *discretize-then-optimize* approach for shape optimization problems within the level-set method.

Chapter 9 : Numerical examples

The ninth chapter opens with the description of a second-order method for shape optimization. It also introduces a gradient-like method that can be compared to the previously defined second-order method. Then it deals with approximating the second-order derivative of the compliance criterion in a fully general framework. We define four different approximations, namely a BFGS-like method, an inexact-Newton method, an incomplete LU method and an approximate CG-Newton method. Finally we give a few numerical examples for the parametric case of thickness optimization and the *discretize-then-optimize* framework. The thickness modeling shows the potentiality of the second-order method with different approximations (see for example Figure 1). However, the *discretize-then-optimize* approach does not seem to be well suited for second-order methods.

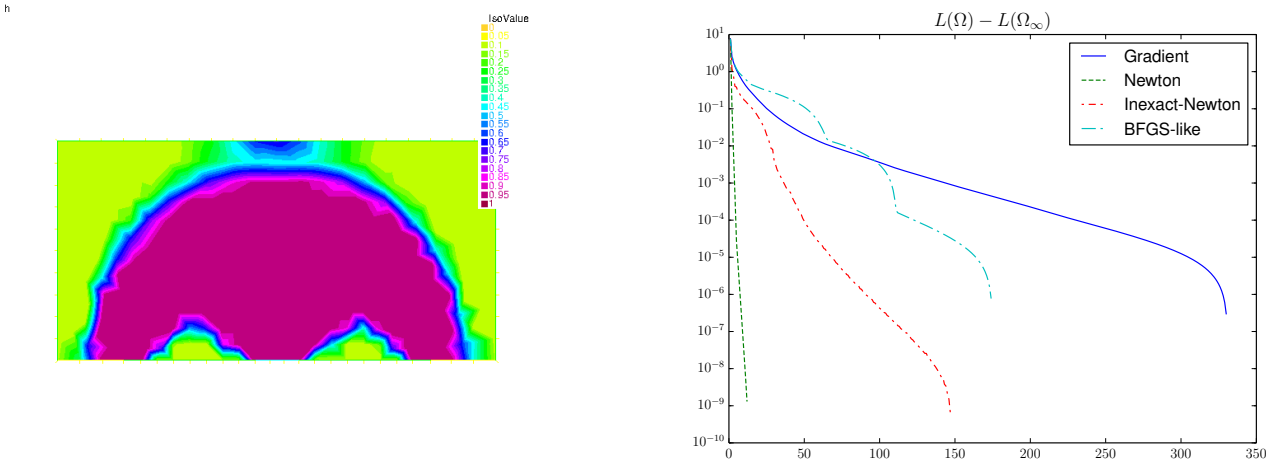


Figure 1: Minimization of the compliance criterion with a penalization on the volume for the parametric case of thickness optimization. One of the final shapes (left) and convergence of the objective function (right).

Part III : Derivation along normal trajectories

The third part opens with the definition of a new framework for computing shape derivatives : this is the main theoretical contribution of this thesis. This allows one to define different first and second-order optimization methods. Then Chapter 11 is devoted to the accurate and consistent computation of shape derivatives. Finally in Chapter 12, we implement this method for computing shape derivatives and compare, on different examples, the efficiency of the different optimization methods.

Chapter 10 : Shape derivation along normal trajectories

Chapter 10 is concerned with the geometric modeling for shape optimization. It has been seen in Chapter 3 that there are already two usual frameworks for shape differentiation : the displacement field method and the speed method. The goal of the present chapter is to define a third approach for shape differentiation by considering time-dependent shapes Ω_t that evolve in the direction of the normal vector $\mathbf{n}(t)$ to the boundary $\partial\Omega_t$. When a shape Ω is represented by a level-set function ϕ, evolving it with a normal velocity v is equivalent to solving the Hamilton-Jacobi equation

$$\begin{cases} \dfrac{\partial \varphi}{\partial t}(t,x) + v(t,x)\,|\nabla_x \varphi(t,x)| = 0, \\ \varphi(0,x) = \phi(x), \end{cases}$$

and to take the time-dependent domain Ω_t defined by $\Omega_t = \{x \in \mathbb{R}^d \mid \varphi(t,x) < 0\}$. This case is not embodied by the speed method. For a given vector field $V \in C^1\left(\mathbb{R}\times\mathbb{R}^d;\mathbb{R}^d\right)$, there is a priori no chance that it is, for each time, directed by the normal $\mathbf{n}(t)$ to the evolving domain. Solving the Hamilton-Jacobi equation is not equivalent to solve the ordinary differential equations that define X_V (see (1)) with a given velocity field V independent of the motion of the shape. However, with φ smooth solution to the Hamilton-Jacobi equation, the motion of the shape is recovered thanks to a set of bicharacteristics (x,p) solution to :

$$\begin{cases} \dfrac{dx}{dt}(t) &= \nabla_p H\big(t,x(t),p(t)\big), \\ \dfrac{dp}{dt}(t) &= -\nabla_x H\big(t,x(t),p(t)\big), \end{cases}$$

with $H(t,x,p) = v(t,x)|p(t,x)|$. For $x_0 \in \Omega$, we can define $X_v(t,x_0) = x(t)$ where (x,p) is the solution to the bicharacteristics system with $x(0) = x_0$ and $p(0) = \nabla\phi(0,x_0)$. Thus, $X_v(t,\Omega)$ defines a time evolving domain, and we can compute derivatives with respect to the time, which defines a notion of shape derivative. This new notion of differentiation leads to a different structure for the second-order shape derivatives for which there is no more influence of the tangential components of the directions of derivation.

There are two main applications for this new framework for shape differentiation. First, it allows one to define a Newton algorithm for shape optimization. Secondly, it appears to be useful for the obtention of extended descent directions, a priori defined only on the boundary of the shape, to the entire space $\mathbb{R}^d$ as required for solving the Hamilton-Jacobi equation. The particular structure of the shape derivative indicates how to choose the normal derivative of the extension on $\partial\Omega$.

Chapter 11 : Computing boundary integrals

With the geometric modeling for shape optimization, whatever the choice of framework for shape differentiation (displacement field method, speed method, derivation along normal trajectories), the shape derivatives are defined by linear and bilinear maps defined on the boundary of the shape. They are written in terms of boundary integrals. Chapter 11 is devoted to the approximation of these integrals. We start by recalling the classical approximation with a Dirac mass concentrated on the boundary. However we also observe that this is not a consistent approach for the computation of shape derivatives.

Thus we also introduce another method for computing boundary integrals. The main idea is to compute exactly the integral of a linear interpolation of a function (or the product of interpolations of functions) on the linear interpolation of the boundary. We then detail the computation of different shape derivatives with this integration method. We also give an estimate on the error due to the approximation of the boundary, and show that for regular shapes this method for computing boundary integrals is of order two with respect to the size of the mesh. Then we confront this theoretical error analysis to some numerical experiments and compare the respective accuracies and consistencies of the two approaches (see Figure 2).

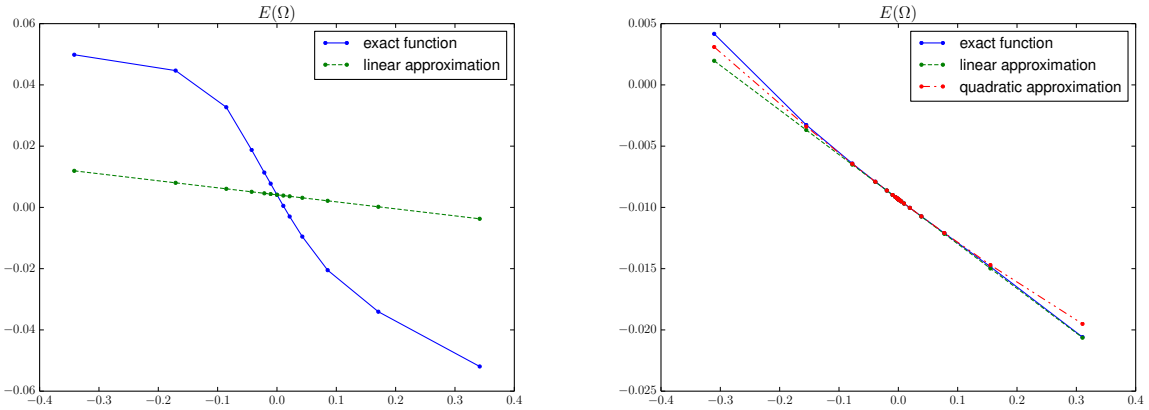


Figure 2: Comparison of consistencies with the approximation of a Dirac mass (left), and with the linear interpolation method (right).

Finally this chapter ends with a remark on the fact that a shape with piecewise linear boundary cannot always be represented by a piecewise linear interpolation of a level-set function.

Chapter 12 : Numerical examples

The twelfth chapter is a catalog of numerical examples for geometric shape optimization when the derivatives are computed along normal trajectories. We aim at comparing the efficiency of the previously described second-order methods (with the exact second-order derivative or only an approximation of it) with gradient methods. We also compare the different methods for extending a velocity field from the boundary $\partial\Omega$ to the entire space $\mathbb{R}^d$ that are introduced in Chapter 10.

The first examples deal with the minimization in the two-dimensional space of a functional of the form

$$E(\Omega) = \int_\Omega f(x)\,dx,$$

where the function f is scalar and does not depend on the domain Ω. Secondly we consider different two-dimensional mechanical examples in the framework of linearized elasticity. The functions to be minimized in these mechanical examples are either the compliance or the least square displacement criterion with a fixed penalization of the volume

of the shape. For example, in Figure 3 we display some result for the minimization of the least-square displacement criterion on the cantilever example.

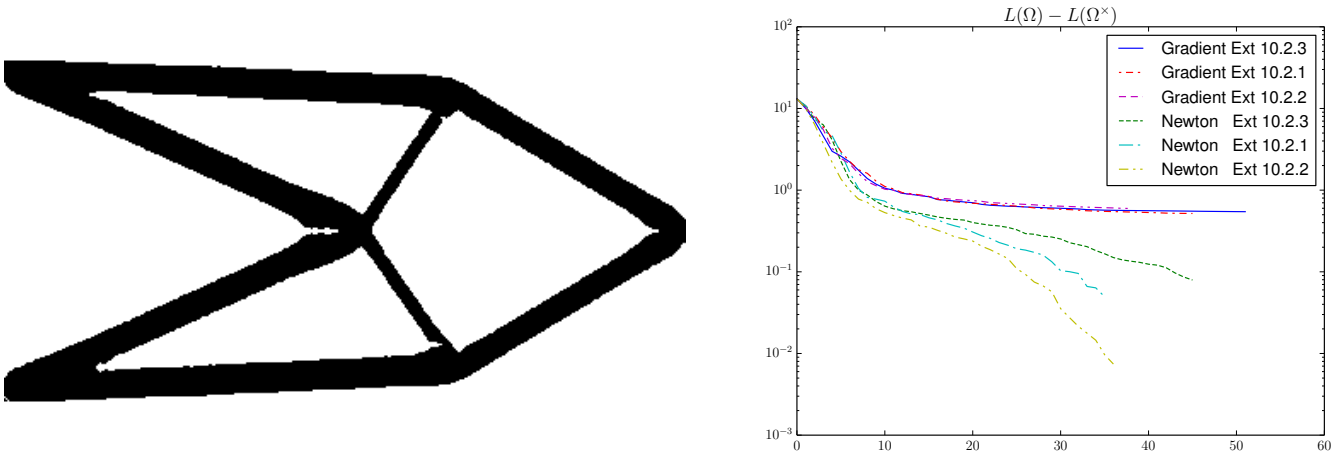


Figure 3: Minimization of the least square displacement criterion with a penalization of the volume for the cantilever. One of the final shapes (left) and evolution of the error with respect to the best shape (right).

In the following section of this chapter we focus on different approximations of the second-order derivative. It starts with describing an additional method based on the mass-lumping approximation. Then we consider the respective consistencies of different approximations for the second-order derivative : the incomplete LU approximation, the CG-Newton approximation and the mass-lumping approximation. Then we compare the efficiencies of these different approximations on some examples with a fixed extension method.

Finally, the last part of this chapter deals with three-dimensional simulations.

Part IV : Numerical analyses

The last part of the thesis is dedicated to numerical analyses of different schemes used for shape optimization when the shape is characterized by the level-set method. It also gives some hints on numerical ingredients that are necessary to get a quadratic rate of convergence. It starts with a complete discretization of a shape optimization algorithm in the two-dimensional space. Next, we focus on the numerical schemes for the level-set method : the Hamilton-Jacobi transport equation and the redistanciation process. Finally the last chapter deals with the complete analysis of a one-dimensional shape optimization example where almost all calculations can be made explicitly. This last chapter is also the opportunity to compare the theoretical rates of convergence of different optimization methods.

Chapter 13 : Full discretization of a shape optimization algorithm

In this chapter, we provide a complete discretization of a geometric shape optimization problem in the two-dimensional space for a simple model problem. The shape is modeled within the level-set method. We detail the numerical schemes for solving the Hamilton-Jacobi advection equation, and for the redistanciation process. The derivatives are computed with the *optimize-then-discretize* approach, meaning that they are discretized versions of the continuous derivatives.

Chapter 14 : Analysis of numerical schemes for the level-set method

This chapter is concerned with the analysis of numerical schemes for the level-set method. We first consider the scheme for the transport equation. In the one-dimensional case, we observe that the scheme used to redistanciate the level-set function does not keep the boundary of the shape unchanged. We also consider a particular example of shape optimization and observe that some natural choice of descent direction makes the boundary be attracted by a node of the mesh. Finally the last section of this chapter deals with the fast-marching method for the redistanciation process and shows on a simple example that this method does not keep a distance function unchanged.

Chapter 15 : A one-dimensional optimization case

In this last chapter, we introduce a one-dimensional shape optimization example for which all calculations are explicit. We take $\Omega =]0,1[$, $z \in \Omega$, $\alpha_1, \alpha_2 \in \mathbb{R}_+^\star$, $\alpha(x) = \alpha_1$ for $x \in]0,z[$, $\alpha(x) = \alpha_2$ for $x \in]z,1[$ and $u_0 \in \mathbb{R}$, with boundary conditions given by Figure 4.

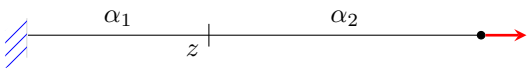


Figure 4: One-dimensional problem.

We aim at minimizing $J = (u(1) - u_0)^2$ where u is solution to

$$\left\{ \begin{array}{rcl} -\dfrac{d}{dx}\left(\alpha \dfrac{d}{dx} u\right) & = & 0 \quad \text{in } \Omega, \\ \left(\alpha \dfrac{d}{dx} u\right)(1) & = & 1, \\ u(0) & = & 0. \end{array} \right. \tag{2}$$

We start by describing the continuous problem, and its theoretical optimal solution. Then we detail a discretized version, and the corresponding optimal solution. The continuous problem can be considered from either a parametric or a shape variation point of view. Thus we detail the derivatives for the two frameworks and their discretizations. We also give the derivatives for the discretized problem.

Then, we analyze the rates of convergence of Newton's method for the different approaches : the continuous model, the *optimize-then-discretize* and *discretize-then-optimize* approaches. We also study for the continuous problem the impact on the rates of convergence of errors on the first and second-order derivatives.

Finally we compare these theoretical results and numerical experiments. In Figure 5 we plot the evolution of the error with the theoretical optimal solution with the different approaches.

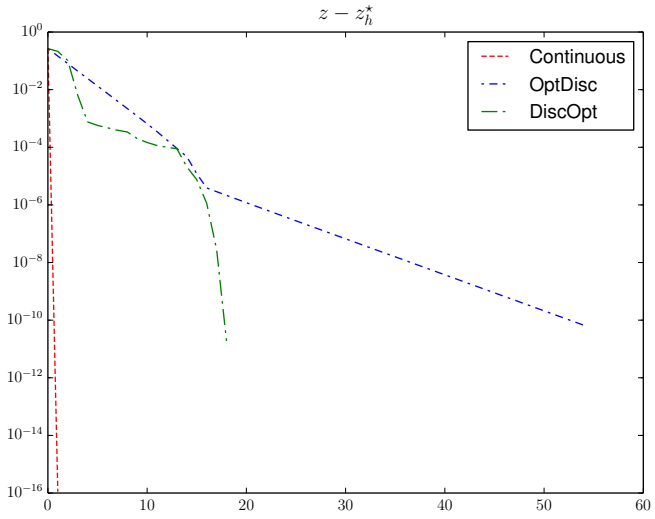


Figure 5: Evolution of the error to the optimal solution for the continuous model, the *optimize-then-discretize* and *discretize-then-optimize* approaches.

Part of the work on the new framework for shape derivative (Chapter 10 and Chapter 12) has been accepted for publication under the title :

Allaire, G.; Cancès, E. & Vié, J.-L. *Second-order shape derivatives along normal trajectories, governed by Hamilton-Jacobi equation* Structural and Multidisciplinary Optimization, 2016, 1-22.

The work of this thesis gave also rise the following poster presentation :

Vié, J.-L. *Second-order shape optimization of a plate with a parametric point of view.* CANUM 2014, Carry-le-Rouet

and the following oral presentations :

Vié, J.-L. *Méthode de second-ordre pour l'optimisation de formes. Un exemple sans contrainte.* SMAI 2015, Les Karellis.

Vié, J.-L. *Second-order shape derivatives along normal trajectories, governed by Hamilton-Jacobi equations.* PICOF 2016, Autrans.

Vié, J.-L. *Second-order shape derivatives along normal trajectories, governed by Hamilton-Jacobi equations.* ECCOMAS 2016, Crete

Vié, J.-L. *Second-order shape derivatives along normal trajectories, governed by Hamilton-Jacobi equations.* ICTAM 2016, Montreal

Part I
Review of the state of the art

"Que dites-vous ?... C'est inutile ?... Je le sais !
Mais on ne se bat pas dans l'espoir du succès !
Non ! non, c'est bien plus beau lorsque c'est inutile !"
E. Rostand

Chapter 1

Optimization

Contents

Optimization is a very important discipline that may arise everywhere when some system is wanted to be improved. The first step of an optimization process consists in defining an *objective* to be optimized. For example, one could want to minimize or maximize a cost, an energy or a time lapse. This objective may depend on different *variables* that describe the characteristics of the considered system. The variables are allowed to vary in a set usually called the *set of admissible solutions*. The objective may be optimized such that some constraints are satisfied. These constraints modify the definition of the set of admissible solutions. The choice of the *set of admissible solutions*, of *objective* and of *variables* is often named *modeling*. There are many possibilities for setting an optimization problem on a given system, since one can describe its characteristics by different ways. A simplistic choice in the modeling may lead to a straightforward optimization problem, but may not be relevant. To the contrary, when the choice is too complex, the optimization problem may have no solution or be too complicated to solve.

One can first mention different kinds of optimization problems. At first, there is the case of discrete optimization where the variables lie in a discrete set of cardinality n. Usually n can be extremely large which makes the comparison of all possible configurations completely out of reach. This can arise in problems such as minimal cuts or smallest paths in graphs. These are combinatorial problems which we will not discuss further, but one can find a lot of literature on them [24, 42, 79, 108]. Secondly, there is the case of continuous optimization where the variables can change continuously in a Banach space. The main advantage of this case relies on the fact that continuous variations in a Banach space allow one to have access to sensitivity analysis which is extremely useful from a theoretical as well as a practical point of view.

Once a model is chosen to represent a system, there are also many optimization algorithms that are available to solve the model problem. One can cite for example the famous gradient algorithm - with different possible variants - or the Newton method that requires the computation of the second-order derivative of the objective function. When the computation of the second-order derivative is too time consuming, it can be relevant to use an approximation instead. This is the main idea of the BFGS or the incomplete Newton's methods.

1.1 Continuous optimization

In this section we consider a Banach space $\mathcal{V}$ with a norm $\|\cdot\|_{\mathcal{V}}$ and an objective function $f : \mathcal{V} \to \mathbb{R}$. This space is the vectorial space where the optimization takes place. Let also $K \subset \mathcal{V}$ be a subset of $\mathcal{V}$, a set where the solution must lie. This last set is also called the set of admissible solutions. We consider the following optimization problem

$$\inf_{v \in K \subset \mathcal{V}} f(v). \tag{1.1}$$

Let us mention that if we aim at maximizing a function instead of minimizing it, it suffices to consider $-f$, and everything in the sequel applies. In the general case, there is a priori no result about the existence or uniqueness of a solution to that problem. Some additional assumptions have to be made about the objective function, the vectorial

space or the set of admissible solutions. At first there is the question of the dimension of the vectorial space $\mathcal{V}$: whether it is infinite or not. Secondly the convexity of the objective function or of the set of admissible solutions may be a matter of importance. At first we recall some classical notation and definitions. We also refer to [7, 8, 24, 25, 26, 75, 92] for more details.

1.1.1 Definitions

If the set of admissible solutions K is empty, there is of course no solution to (1.1) and by convention, the value of the infimum is $+\infty$. To the contrary when K is not empty, the value of (1.1) is the infimum of the values of f on the set K. In the case where f has no inferior bound on K this value is conventionally $-\infty$. Furthermore, when there exists a finite infimum of f on K, this value may not be attained by an element of K. When this value is attained, the infimum is said to be a *minimum*.

Definition 1.1.1. *A element $u \in K$ is said to be a local minimum of f on K if and only if*

$$\exists \delta > 0,\ \forall v \in K,\ \|u - v\|_{\mathcal{V}} \leq \delta \Rightarrow f(u) \leq f(v).$$

Furthermore, $u \in K$ is said to be a global minimum of f on K if and only if

$$\forall v \in K,\ f(u) \leq f(v).$$

Now let us define the notion of convexity for functions and spaces.

Definition 1.1.2. *A set $K \subset \mathcal{V}$ is said to be convex if for every $u, v \in K$ and $\theta \in [0,1]$, the element $\theta u + (1-\theta)v$ is also in K.*

Definition 1.1.3. *A function $f : \mathcal{V} \to \mathbb{R}$ is said to be convex if for every $u, v \in K$ and $\theta \in [0,1]$ the following inequality is satisfied*

$$f\big(\theta u + (1-\theta)v\big) \leq \theta f(u) + (1-\theta) f(v). \tag{1.2}$$

Furthermore, if the inequality is strict for every $u \neq v$ and for $\theta \in]0,1[$, the function f is said to be strictly convex.

In the case where the objective function is convex on a convex set, each local minimum is a global minimum :

Proposition 1.1.4. *Let f be a convex function on a convex set K. Every local minimum of f on K is a global minimum. If f is strictly convex, there exists at most one local minimum which is therefore a global minimum.*

In order to benefit from the sensitivity analysis, we also introduce the notion of differentiability.

Definition 1.1.5. *The function $f : \mathcal{V} \to \mathbb{R}$ is said to be Fréchet-differentiable if there exists a linear map $L \in \mathcal{V}'$ (the dual space of $\mathcal{V}$) such that*

$$f(u+w) = f(u) + L(w) + o(\|w\|_{\mathcal{V}}),$$

where $\lim_{\|w\|_{\mathcal{V}} \to 0} \dfrac{o(\|w\|_{\mathcal{V}})}{\|w\|_{\mathcal{V}}} = 0$. The linear map L is also denoted $L(w) = \langle f'(u), w \rangle$, where $\langle \cdot, \cdot \rangle$ denotes the duality product between $\mathcal{V}'$ and $\mathcal{V}$. The first-order derivative is also represented by its gradient - denoted $\nabla f(u) \in \mathcal{V}$ - that satisfies

$$\forall v \in \mathcal{V},\ \langle f'(u), v \rangle = \nabla f(u) \cdot v.$$

Similarly one can define the second-order differentiability.

Definition 1.1.6. *The function $f : \mathcal{V} \to \mathbb{R}$ is said to be twice Fréchet-differentiable at $u \in \mathcal{V}$ if it is Fréchet-differentiable in a vicinity of u and if the derivative*

$$\begin{array}{rccc} f' & : & \mathcal{V} & \to & \mathcal{V}' \\ & & v & \mapsto & f'(v) \end{array}$$

is Fréchet-differentiable at u. The derivative of f' at u is called the second-order derivative of f at u and is denoted by $f''(u)$.

With this definition of the second-order differentiability, one can recall the second-order Taylor approximation :

Lemma 1.1.7. *If f is twice Fréchet-differentiable on $\mathcal{V}$, then*

$$f(u+w) = f(u) + \langle f'(u), w \rangle + \frac{1}{2} f''(u)(w,w) + o(\|w\|_{\mathcal{V}}^2),$$

where $\lim_{\|w\|_{\mathcal{V}} \to 0} \dfrac{o(\|w\|_{\mathcal{V}}^2)}{\|w\|_{\mathcal{V}}^2} = 0$ and $f''(u)$ is identified to a bilinear continuous form on $\mathcal{V} \times \mathcal{V}$. The second-order derivative is also represented by its Hessian - denoted $D^2 f(u)$ - such that

$$\forall v, w \in \mathcal{V},\ f''(u)(v,w) = D^2 f(u) v \cdot w = v^T\, D^2 f(u)\, w = w^T\, D^2 f(u)\, v.$$

The second-order differentiability gives also a criterion for the convexity.

Lemma 1.1.8. *Let f be twice Fréchet-differentiable on $\mathcal{V}$. Then f is convex if and only if*

$$\forall u, w \in \mathcal{V},\ f''(u)(w,w) \geq 0.$$

1.1.2 Existence results

Whether the space $\mathcal{V}$ is of finite dimension or not, the existence result for (1.1) differs. Let us start with the finite-dimensional case.

Proposition 1.1.9. *Let $\mathcal{V}$ be a finite-dimensional vector space, K a closed non-empty subset of $\mathcal{V}$ and $f : \mathcal{V} \to \mathbb{R}$ a continuous function on $\mathcal{V}$. Assume also that f satisfies the following property*

$$\forall (u_n)_{n\in\mathbb{N}} \in K^{\mathbb{N}},\ \lim_{n\to\infty} \|u_n\|_{\mathcal{V}} = +\infty \Longrightarrow \lim_{n\to\infty} f(u_n) = +\infty. \tag{1.3}$$

Then there exists at least one minimum of f on K.

Remark 1.1.10. *When K is bounded, the property (1.3) is not necessary : since the space $\mathcal{V}$ is of finite dimension, K is compact, and the continuity of f ensures the existence of a minimum.*

When the space $\mathcal{V}$ is of infinite dimension, these assumptions are no longer sufficient to ensure the existence of a minimum. Existence results are recovered with additional conditions on the set K and the function f. We recall here some results of [7, 35].

Theorem 1.1.11. *Assume that K is a bounded and weakly closed subset of a reflexive Banach space, and that $f : \mathcal{V} \to \mathbb{R}$ is weakly lower semicontinuous. Then f has a global minimum on K.*

Similarly to the finite-dimensional case, the boundedness condition on the set K can be replaced by the property (1.3) stating that f is infinite at infinity (see [35, Theorem 1.2]). However, the closure condition is still necessary for the existence of a minimum. When this last condition is not satisfied but there are convexity assumptions on f and K, there may still be existence of a minimum, such as stated by the following theorem. We recall that it is a stronger assumption to assume that a space is weakly closed than to assume it is only closed.

Theorem 1.1.12. *Let K be a closed convex non-empty subset of a reflexive Banach space $\mathcal{V}$, and f be a convex continuous function on K satisfying the property (1.3). Then there exists a minimum of f on K. Furthermore, if the function f is strictly convex, then the minimum is unique.*

This existence result is hardly used in shape optimization since the convexity of the subset K and the function f as well as the closure condition on K are very strong conditions. In practice, there are lots of minimization problems for which there is not existence of a solution. For shape optimization, the natural choice of modeling makes usually the problems ill-posed (see Section 2.3).

1.1.3 Optimality conditions

Theorem 1.1.13. *Assume that K is an open set, that the function $f : K \subset \mathcal{V} \to \mathbb{R}$ has a local minimum at $u \in K$ and is Fréchet-differentiable. Then*

$$\forall w \in \mathcal{V},\ \langle f'(u), w\rangle = 0.$$

When the subset K is convex, the first-order optimality condition may be written as follows.

Theorem 1.1.14. *Assume that K is convex set and that $f : K \subset \mathcal{V} \to \mathbb{R}$ is a convex Fréchet-differentiable function. Then $u \in K$ is a minimum of f if and only f*

$$\forall v \in K,\ \langle f'(u), v - u\rangle \geq 0.$$

The convexity assumption on K and f makes the necessary optimality condition of Theorem 1.1.13 become a sufficient condition. When $K = \mathcal{V}$, there is also a second-order optimality condition :

Theorem 1.1.15. *Assume that $K = \mathcal{V}$ and that f is twice Fréchet-differentiable.*

- *If u is a local minimum of f on K then*

$$f'(u) = 0, \quad \forall w \in \mathcal{V},\ f''(u)(w, w) \geq 0.$$

- *Conversely if $f'(u) = 0$ and for every $v \in \mathcal{V}$ in a neighborhood of u*

$$\forall w \in \mathcal{V},\ f''(v)(w, w) \geq 0.$$

  then u is a local minimum of f.

Now we consider the case where the set K is not equal to $\mathcal{V}$ and can be described as a set

$$K = \{v \in \mathcal{V} \mid F(v) = 0\},$$

where $F(v) = \Big(F_1(v), \cdots, F_p(v)\Big)$ and $F_j : \mathcal{V} \to \mathbb{R}$ for all $1 \leq j \leq p$. The optimization problem now reads

$$\inf_{v\in\mathcal{V},\ F(v)=0} f(v). \tag{1.4}$$

In this framework it is useful to introduce the notion of Lagrangian.

Definition 1.1.16. *The Lagrangian of problem (1.4) is defined by*

$$\forall (v,\mu) \in \mathcal{V} \times \mathbb{R}^p, \quad \mathcal{L}(v,\mu) = f(v) + \sum_{i=1}^{p} \mu_i F_i(v) = f(v) + \mu \cdot F(v). \tag{1.5}$$

The variable $\mu \in \mathbb{R}^p$ is called the Lagrange multiplier of the constraint $F(v) = 0$.

Now the set K has no reason to still have a convex structure. When the function F is continuous, the set K is no longer open, and therefore the Theorem 1.1.13 is not available. The necessary optimality condition is given by the following theorem.

Theorem 1.1.17. *Let $u \in \mathcal{V}$ such that $F(u) = 0$. Assume that f so as F_i $(1 \leq i \leq p)$ are continuously differentiable in the vicinity of u. If u is a local minimizer of f on K and the vectors $\left(F_i'(u)\right)_{1\leq i\leq p}$ are linearly independent, then there exists a Lagrange multiplier $\lambda \in \mathbb{R}^p$ such that*

$$\frac{\partial \mathcal{L}}{\partial v}(u,\lambda) = f'(u) + \lambda \cdot F'(u) = 0 \tag{1.6}$$

and

$$\frac{\partial \mathcal{L}}{\partial \mu}(u,\lambda) = F(u) = 0. \tag{1.7}$$

The notion of Lagrangian and of Lagrange multiplier can be extended to different cases such as the case when the function F defining the set K takes its values in an infinite-dimensional space. It can also be extended to the case of inequality constraints where there are restrictions on the sign of the multiplier.

1.2 Algorithms

In the previous section, we considered the existence of solutions to a minimization problem. Now we shall be concerned with constructive aspects, that is finding sequences of approximating solutions. For the sake of simplicity, we will consider the case where $K = \mathcal{V} = \mathbb{R}^d$, so that $\mathcal{V}$ can be identified to its dual space. In particular this is a finite-dimensional case without constraints :

$$\inf_{v\in\mathcal{V}} f(v). \tag{1.8}$$

We will also assume that the function f is strictly convex and goes to $+\infty$ at infinity, which ensures by Theorem 1.1.12 the existence of a unique minimizer $u^\star$ to (1.8). In addition we also assume that f is three-times differentiable. Starting from an initial data u_0 we aim at constructing a sequence $(u^n)_{n\in\mathbb{N}}$ that converges to $u^\star$. Basically the iterates of the sequence are built on the same scheme. Given a step ρ^k and a direction w^k the iterate u^{k+1} is computed by

$$u^{k+1} = u^k + \rho^k\, w^k. \tag{1.9}$$

In view of minimizing the objective f, the sequence $f(u^n)$ is required to be decreasing. It is usually required that the direction w^k is a *descent direction*, *i.e* such that $\left\langle f'(u^k), w^k \right\rangle < 0$. It ensures that the function f can be reduced in this direction. The step ρ^k intervenes here : with a given direction w^k it can be adjusted such that the reduction of f allowed by the descent direction is effective.

There are many strategies to choose an efficient step. The question of finding a step is often called the *line search* procedure. Two main things are to be noticed about it. A particularly small step should ensure the reduction of the objective - *i.e* $f\left(u^{k+1}\right) \leq f\left(u^k\right)$ - but the change in the iterate could be extremely tiny. Therefore the convergence of the sequence to the solution would be very slow. To the contrary, with a large step the convergence can be hoped to be quicker but there is no insurance to keep the decrease property, since the first-order approximation is only local.

The most famous rules for finding a 'nice' step are given by Wolfe or Goldstein [92, §3]. These rules rely basically on the ideas that there should be a sufficient decrease between two successive values of the objective function, and that the step should not be too small. We can restrict ourselves to the Armijo backtracking procedure that is recognized to be sufficient [92, Procedure 3.1] in practice.

Algorithm 1.2.1. *Backtracking line search*

1. *Choose $\rho > 0$, $c_1, c_2 \in]0,1[$.*
2. *Compute $f(u^k + \rho w^k)$.*
3. *If $f(u^k + \rho w^k) > f(u^k) + c_2 \left\langle f'(u^k), w^k \right\rangle$ set $\rho = c_1\rho$ and go to step 2. Else set $\rho^k = \rho$ and stop.*

1.2.1 Gradient method

The main idea of gradient methods is to take directions given by the steepest-descent direction. The descent direction is therefore given by the opposite gradient direction $w^k = -\nabla f(u^k)$. In that case we straightforwardly have $\left\langle f'(u^k), w^k \right\rangle \leq 0$ with equality only at critical points. Even if in practice the choice of the backtracking line search (Algorithm 1.2.1) gives satisfactory convergence result we recall here two choices of a step for which the convergence of a gradient algorithm is established.

Optimal step

At iteration $k \in \mathbb{N}$ with a descent direction $w^k = -\nabla f(u^k)$ given by the gradient,

$$u^{k+1} = u^k + \rho^k w^k, \tag{1.10}$$

the step ρ^k may be chosen in an optimal way, that is such that

$$f\big(u^{k+1}\big) = \inf_{\rho \in \mathbb{R}_+} f\big(u^k + \rho\, w^k\big). \tag{1.11}$$

In other words, this consists in taking

$$\rho^k = \arg\inf_{\rho \in \mathbb{R}_+} f\big(u^k + \rho\, w^k\big).$$

In that case, under regularity conditions on the function f, one can recall the following convergence result ([7, Théorème 3.38]).

Theorem 1.2.2. *Assume that f is strictly convex and differentiable. Assume also that the first derivative f' is Lipschitz on $\mathcal{V}$,* i.e *there exists $C > 0$ such that*

$$\forall v, w \in \mathcal{V},\ \|f'(u) - f'(v)\|_{\mathcal{V}} \leq C \|u - v\|_{\mathcal{V}}.$$

Then the gradient algorithm given by (1.10) and (1.11) converges to $u^\star$ solution to (1.8).

Fixed step

The computation of an optimal step accelerates the convergence of the sequence $(u_n)_{n\in\mathbb{N}}$ but is quite expensive in time since at each iteration one has to solve another minimization problem. Therefore we also consider the algorithm with a fixed step :

$$u^{k+1} = u^k - \rho \nabla f(u^k), \tag{1.12}$$

where $\rho > 0$ is a priori chosen and the same for every iteration. In that case, under additional regularity conditions on the objective function f there is still a convergence result.

Theorem 1.2.3. *Assume that f is differentiable and strongly convex,* i.e *there exists $\alpha > 0$ such that*

$$\forall u, v \in \mathcal{V},\ \langle f'(u) - f'(v), u - v\rangle \geq \alpha \|u - v\|_{\mathcal{V}}.$$

Assume also that f' is Lipschitz on $\mathcal{V}$ with constant $C > 0$. With the additional condition that $0 < \rho < \dfrac{2\alpha}{C^2}$ the gradient algorithm with fixed step converges to $u^\star$.

1.2.2 Newton's method

There are many other possible choices for a direction w^k to be a descent direction. Instead of taking the steepest-descent direction, the Newton method intends to find the direction that minimizes the second-order Taylor approximation of f around u^k :

$$f(u^k + w) \simeq f(u^k) + \big\langle f'(u^k), w\big\rangle + \frac{1}{2} f''(u^k)(w, w). \tag{1.13}$$

When the quadratic form $f''(u^k)$ is positive definite, the minimization of $w \mapsto f(u^k) + \big\langle f'(u^k), w\big\rangle + \dfrac{1}{2} f''(u^k)(w, w)$ has a solution given by

$$w^k = \Big(f''(u^k)\Big)^{-1} \nabla f(u^k), \tag{1.14}$$

where in this last expression, $f''(u^k)$ is seen as an operator from $\mathcal{V}$ to $\mathcal{V}'$. With this choice of direction, the choice $\rho^k = 1$ is natural. Moreover, it appears that it ensures quadratic convergence under additional regularity conditions on f. In order to ensure the existence of w^k solution to (1.14) we also see that the the Hessian f'' has to be invertible at u^k. This is usually given by its positive-definiteness.

Theorem 1.2.4. *Assume that f is three-times differentiable in a neighborhood of a minimizer $u^\star$ where $f'(u^\star) = 0$ and $f''(u^\star)$ is positive definite. With the iteration $u^{k+1} = u^k + w^k$ where w^k is given by (1.14) then*

1. *if the initial data u_0 is close enough from $u^\star$, the sequence $(u^k)_{k\in\mathbb{N}}$ converges to $u^\star$,*
2. *furthermore the rate of convergence of $(u^k)_{k\in\mathbb{N}}$ is quadratic, meaning that there exists $C > 0$ such that*

$$\big\|u^{k+1} - u^\star\big\|_{\mathcal{V}} \leq C \big\|u^k - u^\star\big\|_{\mathcal{V}}^2,$$

3. *the sequence $f'(u^k)$ converges also quadratically to zero*

$$\Big\|f'\Big(u^{k+1}\Big)\Big\|_{\mathcal{V}} \leq C \Big\|f'\Big(u^k\Big)\Big\|_{\mathcal{V}}^2,$$

The differentiability of the second-order derivative is not necessary in this theorem, a Lipschitz continuity condition on the second-order derivative is sufficient (see [92, Theorem 3.7]).

1.2.3 The BFGS method

The computational cost of computing the full Hessian may be prohibitive. Therefore the question of approximating the second-order derivative at a monitored cost was soon considered. The BFGS method was introduced in the 70s by Broyden, Fletcher, Goldfarb and Shanno [28, 29, 62, 65, 116]. Its main idea is to approximate the second-order derivative with the difference between two successive first-order derivatives. The second-order quadratic approximation of the objective function at u^k reads for $w \in \mathbb{R}^d$

$$f(u^k) + \nabla f\big(u^k\big)\, w + \frac{1}{2} w^T\, D^2 f\big(u^k\big)\, w.$$

Now let's assume that the complete Hessian $D^2 f(u^k)$ is not known, but only an approximation : B_k. Therefore, this leads to consider the following approximation

$$m_k(w) = f\big(u^k\big) + \nabla f\big(u^k\big)\, w + \frac{1}{2} w^T\, B_k\, w.$$

The minimization of this approximation with respect to w leads to take w^k solution to

$$B_k\, w^k = -\nabla f\big(u^k\big). \tag{1.15}$$

With this direction w^k and a step ρ^k, a new iterate u^{k+1} can be computed by

$$u^{k+1} = u^k + \rho^k\, w^k.$$

Now, based on a quadratic approximation of the objective function at this new iterate, we aim at finding requirements on an approximation of the second-order derivative at this new iterate : B_{k+1}. The quadratic approximation would write

$$m_{k+1}(w) = f\big(u^{k+1}\big) + \nabla f\big(u^{k+1}\big)\, w + \frac{1}{2} w^T\, B_{k+1}\, w.$$

By construction, this approximation satisfies straightforwardly

$$m_{k+1}(0) = f\big(u^{k+1}\big), \quad \nabla m_{k+1}(0) = \nabla f\big(u^{k+1}\big).$$

The quadratic form m_{k+1} is an approximation of $f\big(u^{k+1} + w\big)$. For $w = -\rho^k\, w^k$, it could be expected that its gradient matches with the gradient of f at $u^{k+1} - \rho^k\, w^k = u^k$. Broyden, Fletcher, Goldfarb and Shanno therefore proposed the requirement

$$\nabla m_{k+1}\big(-\rho^k\, w^k\big) = \nabla f\big(u^k\big).$$

This also writes

$$B_{k+1}\, \rho^k w^k = \nabla f\big(u^{k+1}\big) - \nabla f\big(u^k\big).$$

Introducing the notation

$$s^k = u^{k+1} - u^k, \quad y^k = \nabla f\big(u^{k+1}\big) - \nabla f\big(u^k\big),$$

this leads to

$$B_{k+1}\, s^k = y^k. \tag{1.16}$$

There is absolutely no chance to have a unique matrix B_{k+1} satisfying this equation. But restricting to the case where

$$B_{k+1} = B_k + M_k,$$

it is possible to find a rank-one or rank-two matrix M_k such that B_{k+1} fulfills the requirement (1.16).

Rank-1 updates

Looking for a rank-one matrix consists in finding $c \in \mathbb{R}$ and $z \in \mathbb{R}^d$ such that

$$M_k = c z\, z^T.$$

There are many possible choices for c and z, and one can for example take

$$z = y_k - B_k\, s_k, \quad c = \frac{1}{{z_k}^T\, s_k}. \tag{1.17}$$

This choice leads to

$$\begin{aligned} B_{k+1}\, s_k &= B_k\, s_k + \frac{\big(y_k - B_k\, s_k\big)\big({y_k}^T\, s_k - {s_k}^T\, {B_k}^T\, s_k\big)}{{y_k}^T\, s_k - {s_k}^T\, B_k\, s_k} \\ &= B_k\, s_k + y_k - B_k\, s_k \\ &= y_k. \end{aligned}$$

The rank-one update of the matrix guarantees that the newly approximated matrix stays symmetric but there is no insurance to keep the positive definite character. Nevertheless, it can be proven that under regularity conditions on the objective function, the so-computed approximation of the Hessian converges in some sense to the exact Hessian.

The computation of the descent direction w^k using (1.15) requires to solve a linear system. Therefore instead of working with an approximation of the Hessian, it can be more practical to work with an approximation of its inverse. It appears that with a Sherman-Morrison formula [92, Appendix A], the rank-one update formulas for the inverse are similar.

Rank-2 updates

The non-conservation of the positive definiteness of the approximation with the rank-one update may lead to difficulties to find a descent direction. This is the main reason why one considers rank-two updates. In that case, the matrix M_k is searched under the form

$$M_k = c_1 z_1 \, {z_1}^T + c_2 z_2 \, {z_2}^T,$$

with c_1, c_2 in $\mathbb{R}$ and z_1, z_2 in $\mathbb{R}^d$. There is still non-uniqueness of a solution but one can take

$$\begin{cases} z_1 = y_k, & c_1 = \dfrac{1}{{y_k}^T \, s_k}, \\ z_2 = B_k \, s_k & c_2 = \dfrac{-1}{{s_k}^T \, B_k \, s_k}. \end{cases}$$

With this choice, the new iterate B_{k+1} writes

$$B_{k+1} = B_k + \frac{y_k \, {y_k}^T}{{y_k}^T \, s_k} - \frac{B_k \, s_k \, {s_k}^T \, {B_k}^T}{{s_k}^T \, B_k \, s_k}. \tag{1.18}$$

At first one can easily check that if the matrix B_k is symmetric, then it is the same for B_{k+1}. The question about the positive definitiveness of B_{k+1} is quite more tricky. If we assume that B_k is positive definite, one needs the additional requirement

$${s_k}^T \, y_k > 0,$$

to ensure that B_{k+1} is positive definite.

Lemma 1.2.5. *Assume that B_k is symmetric positive definite and that ${s_k}^T \, y_k > 0$. Then with*

$$B_{k+1} = B_k + \frac{y_k \, {y_k}^T}{{y_k}^T \, s_k} - \frac{B_k \, s_k \, {s_k}^T \, {B_k}^T}{{s_k}^T \, B_k \, s_k},$$

the new iterate B_{k+1} is also symmetric positive definite.

Proof. The symmetry of the new iterate B_{k+1} is straightforwardly given by the one of B_k. With H_k being the inverse of B_k, it is also possible - thanks to a Sherman-Morrison formula [92, Appendix A] - to get an update formula for H_{k+1} the inverse of B_{k+1} (with ${s_k}^T \, y_k \neq 0$)

$$H_{k+1} = \left(I - \frac{1}{{s_k}^T \, y_k} \, s_k \, {y_k}^T \right) H_k \left(I - \frac{1}{{s_k}^T \, y_k} \, y_k \, {s_k}^T \right) + \frac{1}{{s_k}^T \, y_k} \, s_k \, {s_k}^T.$$

Since ${s_k}^T \, y_k > 0$ it is easy to check that H_{k+1} is positive. Let $v \in \mathbb{R}^d$; one has

$$\begin{aligned} v^T \, H_{k+1} \, v &= \left(v - \frac{1}{{s_k}^T \, y_k} \, y_k \, {s_k}^T \, v \right)^T H_k \left(v - \frac{1}{{s_k}^T \, y_k} \, y_k \, {s_k}^T \, v \right) + \frac{1}{{s_k}^T \, y_k} \, v^T \, s_k \, {s_k}^T \, v \\ &\geq 0. \end{aligned}$$

Since H_k is positive definite, assuming that $v^T \, H_{k+1} \, v = 0$ leads to

$$\begin{aligned} v - \frac{1}{{s_k}^T \, y_k} \, y_k \, {s_k}^T \, v &= 0 \\ {s_k}^T \, v &= 0 \end{aligned}$$

and therefore $v = 0$. As a result H_{k+1} is positive definite, and so as B_{k+1}.

□

When the assumption ${s_k}^T \, y_k > 0$ is not satisfied, one can introduce

$$r_k = \theta_k \, y_k + (1 - \theta_k) \, B_k \, s_k,$$

with

$$\theta_k = \begin{cases} 1 & \text{if } {s_k}^T y_k \geq 0.2 {s_k}^T B_k s_k \\ 0.8 \times \dfrac{{s_k}^T B_k s_k}{{s_k}^T B_k s_k - {s_k}^T y_k} & \text{otherwise.} \end{cases}$$

Since B_k is assumed to be positive definite, one can easily check that $\theta_k \in [0,1]$ and that ${s_k}^T r_k > 0$. Then one replaces y_k by r_k in the update formula (1.18). With this modification the requirement (1.16) is not really satisfied since $B_{k+1} s_k = r_k$ and when $\theta_k \neq 0$ the vector r_k is different from y_k. But the condition ${s_k}^T r_k > 0$ guarantees that B_{k+1} is positive definite which is way more important for practical purposes.

1.3 Optimize-then-discretize

Different questions arise beyond the choice of a model, such as the question of discretization, the numerical aspects or the way to link discretization to optimization. With a given model, *i.e* an *objective function* and a set of *variables* the discretization mainly consists in projecting the variables in a finite-dimensional space. For example, let us consider the model

$$\inf_{v \in K \subset \mathcal{V}} f(v). \tag{1.19}$$

A discretized problem would be written

$$\inf_{v_h \in K_h \subset \mathbb{R}^{N_h}} f_h(v_h). \tag{1.20}$$

Remark 1.3.1. *Let us take an example. Let $g^0 \in C^0([0,1];\mathbb{R})$ and consider*

$$\inf_{g \in L^2([0,1];\mathbb{R})} \int_{[0,1]} |g - g^0|^2,$$

The minimum is obviously attained at $g = g^0$ but that is not the point here. In order to numerically solve this problem, we can consider $(x_i)_{1 \leq i \leq N_h}$ a subdivision of $[0,1]$ and the finite element space

$$\mathbb{P}_1 = \left\{ g \in C^0([0,1]) \mid \forall 1 \leq i \leq N_h,\ g \text{ is affine on } [x_i, x_{i+1}] \right\}.$$

Let g_h^0 be the interpolation of g^0 on $\mathbb{P}_1$. The restriction to $\mathbb{P}_1$ of the minimization problem writes

$$\inf_{g \in \mathbb{P}_1} \int_{[0,1]} |g - g_h^0|^2, \tag{1.21}$$

For $g \in \mathbb{P}_1$, the value of $\displaystyle\int_{[0,1]} |g - g_h^0|^2$ varies only with respect to the values of g at the points $(x_i)_{1 \leq i \leq N_h}$. Denoting by g_i these values, one can define

$$f_h : \mathbb{R}^{N_h} \ni (g_1, \cdots, g_{N_h}) \to \int_{[0,1]} |g - g_h^0|^2.$$

Therefore, the problem (1.21) now reads

$$\inf_{(g_i) \in \mathbb{R}^{N_h}} f_h(g_1, \cdots, g_{N_h}), \tag{1.22}$$

There are two main approaches for solving numerically an optimization problem such as (1.19). At first we mention the *Optimize-then-discretize* one. The idea is to write the optimality conditions for the real problem (1.19), to discretize the resulting problem and then to solve it. Secondly we can also consider the *Discretize-then-optimize* approach, which was first introduced for optimal control problems [23, 73, 77, 80, 101]. It consists first in discretizing the optimization problem and then in solving the resulting finite-dimensional optimization problem.

Let us consider a problem without constraint, and its discretized counterpart

$$\inf_{v \in \mathcal{V}} f(v), \qquad \inf_{v_h \in \mathbb{R}^{N_h}} f_h(v_h). \tag{1.23}$$

The first-order optimality conditions respectively read

$$\forall w \in \mathcal{V},\ \langle f'(v^\star), w \rangle_{\mathcal{V}',\mathcal{V}} = 0, \qquad \forall w_h \in \mathbb{R}^{N_h},\ \langle \nabla f_h(v_h^\star), w_h \rangle_{\mathbb{R}^{N_h},\mathbb{R}^{N_h}} = 0.$$

But the discretization of the first-order condition for the continuous problem in $\mathcal{V}$ reads

$$\forall w_h \in \mathbb{R}^{N_h},\ \langle f_h'(\tilde{v}_h^\star), w_h \rangle_{\mathbb{R}^{N_h},\mathbb{R}^{N_h}} = 0.$$

All the difference is between f_h' and ∇f_h. The first one is the discretization of the derivative of f on $\mathcal{V}$ whereas ∇f_h is the gradient in $\mathbb{R}^{N_h}$ of the discretized function f_h. There is a priori no chance for these two quantities to coincide

(in some particular case it is possible), leading to different results of the optimization process.

We recall from [127] the difference between accuracy and consistency for the computed derivatives. Accuracy is the difference with the exact derivatives of the continuous model, whereas consistency is the difference with the derivatives of the numerical model. With this point of view, the *Optimize-then-discretize* approach is accurate - the accuracy is given by the size of the discretization - but may not be consistent. As a result, there is no insurance to reach the exact optimality conditions of the continuous model. To the contrary, the *Discretize-then-optimize* approach is not accurate but is consistent. This means that it should find a solution of the numerical problem (since the derivatives are consistent), but this solution may not be a good solution at all for the continuous problem.

Chapter 2

Shape optimization methods

Contents

At the beginning of any optimization process, there is a modeling question. One has to choose a mathematical model to represent the data to be optimized. There are two main ingredients in a mathematical model for shape optimization : at first the way to represent a shape, and secondly the way to perform a sensitivity analysis.

As regards shape optimization, the most general modeling of a shape is to consider it as a variable bounded open subset of $\mathbb{R}^d$. The main trouble with this approach is that the set of bounded open sets of $\mathbb{R}^d$ does not have a vectorial space (nor a differentiable manifold) structure. Therefore, the usual mathematical results for optimization are not available. The modeling consists in finding a variable that would lie in a "nice" space - such as a Banach space - and that would represent a shape. The different methods for optimization of a shape are traditionally divided in three categories : parametric optimization, geometric optimization and topology optimization. We will present these methods thanks to some physical examples, therefore we will start by introducing different physical models.

2.1 Physical models

The physical models we will considered may be classified in different categories, whether the unknown is scalar, and whether there is a planar invariance.

2.1.1 Membrane model : $-\mathrm{div}(B\nabla u)$

We consider a membrane $\Omega_{\mathbb{R}^3}$ represented by a bounded open set $\Omega \subset \mathbb{R}^2$ and a thickness $h \in L^\infty(\Omega;\mathbb{R}_+)$:

$$\Omega_{\mathbb{R}^3} = \{X \in \mathbb{R}^3 \,|\, X = (x,y,z) \text{ with } (x,y) \in \Omega \text{ and } 0 \leq z \leq h(x,y)\}.$$

The behavior of the membrane is represented by a second-order symmetric positive definite tensor B. For a given load $f \in L^2(\Omega;\mathbb{R})$, the out-of-plane displacement of the membrane u is modeled by the following equation

$$\left\{ \begin{array}{rcll} -\mathrm{div}\,(hB\nabla u) & = & f & \text{in } \Omega, \\ u & = & 0 & \text{on } \partial\Omega. \end{array} \right. \tag{2.1}$$

The domain $\Omega_{\mathbb{R}^3}$ is formally replaced by Ω in (2.1) by integrating in the direction e_z and assuming planar invariance in this direction. We could also have considered different boundary conditions for this equation, such as Neumann boundary conditions. The displacement u is a scalar quantity from Ω to $\mathbb{R}$ and is defined a priori in $H^1(\Omega;\mathbb{R})$.

2.1.2 Planar elastic model : $-\mathrm{div}(A\varepsilon(u))$

We take the same membrane $\Omega_{\mathbb{R}^3}$ as in Section 2.1.1. Now we are concerned with in-plane displacements. Taking a load $f \in L^2(\Omega;\mathbb{R}^2)$, the in-plane displacements u are modeled by the solution to

$$\left\{\begin{array}{rcll} -\mathrm{div}\left(hA\varepsilon(u)\right) & = & f & \text{in } \Omega, \\ u & = & 0 & \text{on } \partial\Omega, \end{array}\right. \tag{2.2}$$

where the deformation tensor ε is defined by

$$\varepsilon(u) = \frac{1}{2}\left(\nabla u + \nabla u^T\right),$$

and the elastic behavior of the membrane is modeled by the Lamé's coefficients λ, μ and the Hooke's symmetric tensor A that acts on a matrix ζ as follows

$$A\zeta = 2\mu\zeta + \lambda \mathrm{Tr}(\zeta)\mathrm{Id}.$$

Like previously, the domain $\Omega_{\mathbb{R}^3}$ is replaced by Ω to get (2.2) by integrating along the direction e_z and assuming planar invariance of u in this direction. Here, the displacement u is a two-dimensional quantity from Ω to $\mathbb{R}^2$ defined a priori in $H^1(\Omega;\mathbb{R}^2)$ (given by elliptic regularity).

Instead of having f as data with a homogeneous Dirichlet boundary conditions, we could also have other boundary conditions on different parts of the boundary.

2.1.3 Planar vectorial model : $-\mathrm{div}(B\nabla u)$

We also take the same membrane $\Omega_{\mathbb{R}^3}$ as in Section 2.1.1. With $f \in L^2(\Omega;\mathbb{R}^2)$ we consider u as the solution to

$$\left\{\begin{array}{rcll} -\mathrm{div}\left(hB\nabla u\right) & = & f & \text{in } \Omega, \\ u & = & 0 & \text{on } \partial\Omega, \end{array}\right. \tag{2.3}$$

Compared to the planar elastic model of Section 2.1.2, we just replace $\varepsilon(u)$ and A by respectively ∇u and B.

2.1.4 Scalar thermal model : $-\mathrm{div}(B\nabla u)$

Here we consider a shape Ω being a bounded open set of $\mathbb{R}^d$ with $d \in \{2,3\}$. We do not assume any more any planar invariance. For an initial data $f \in L^2(\Omega;\mathbb{R})$ we consider the following boundary problem with unknown $u : \Omega \to \mathbb{R}$:

$$\left\{\begin{array}{rcll} -\mathrm{div}\left(B\nabla u\right) & = & f & \text{in } \Omega, \\ u & = & 0 & \text{on } \partial\Omega, \end{array}\right. \tag{2.4}$$

where B is symmetric second-order tensor. The unknown u may model the solution of a thermal problem. This is the reason why this model is said to be a thermal model. Similarly to the other models, we could also impose other boundary conditions on different parts of the boundary than the homogeneous Dirichlet condition on $\partial\Omega$. The tensor B is denoted as the thermal tensor.

2.1.5 Elastic model : $-\mathrm{div}(A\varepsilon(u))$

Finally we introduce the linear elastic model. We take Ω as a bounded open set of $\mathbb{R}^d$, and $f \in L^2(\Omega;\mathbb{R}^d)$ a volumic load. The displacement $u : \Omega \to \mathbb{R}^d$ of the shape under this load is modeled by the solution to

$$\left\{\begin{array}{rcll} -\mathrm{div}\left(A\varepsilon(u)\right) & = & f & \text{in } \Omega, \\ u & = & 0 & \text{on } \partial\Omega. \end{array}\right. \tag{2.5}$$

Like in Section 2.1.2, the parameters λ and μ are the Lamé's coefficients of the material. The elastic behavior of the shape is modeled by the Hooke's tensor A defined by

$$A\zeta = 2\mu\zeta + \lambda \mathrm{Tr}(\zeta)\mathrm{Id}.$$

The deformation tensor $\varepsilon(u)$ is also defined by

$$\varepsilon(u) = \frac{1}{2}\left(\nabla u + \nabla u^T\right).$$

2.2 Overview of the different methods

2.2.1 Parametric optimization

In the case of parametric optimization, the shape is characterized by a finite number of parameters. This could be a set of lengths, curvatures or thickness for example. In this section we present a parametric case of shape optimization. A reference domain Ω is taken as a bounded open set of $\mathbb{R}^2$, and does not vary. At each point $x \in \Omega$ we assign a thickness $h(x)$ (see Figure 2.1) that may evolve between two given values $h_M \geq h_m > 0$. This models the three-dimensional shape $\Omega_{\mathbb{R}^3}$ defined by

$$\Omega_{\mathbb{R}^3} = \{X \in \mathbb{R}^3 \,|\, X = (x, y, z) \text{ with } (x, y) \in \Omega \text{ and } 0 \leq z \leq h(x, y)\}.$$

At first, let us mention that this choice has a great importance on the behavior of the optimization. Considering the

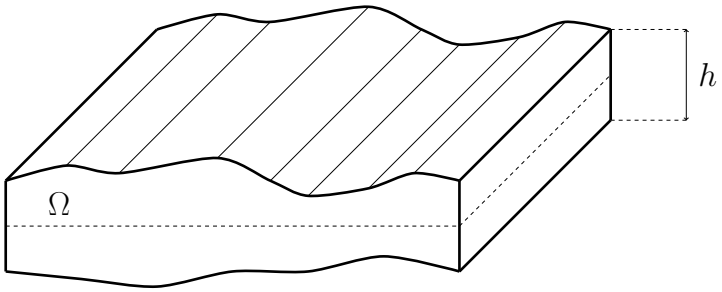


Figure 2.1: Thickness of a plate.

thickness h as a function in $L^\infty(\Omega; [h_m, h_M])$, there cannot be any holes in the shape. Moreover there should be only two boundaries in the z-direction for $\Omega_{\mathbb{R}^3}$. This is kind of a moldability constraint : the shape must have a molding direction in the z-direction (see for example [83] for more details on moldability constraints). In Figure 2.2, we give an example of a shape that cannot be represented by the thickness parametrization : at the point x_0 there is no clear definition of the thickness.

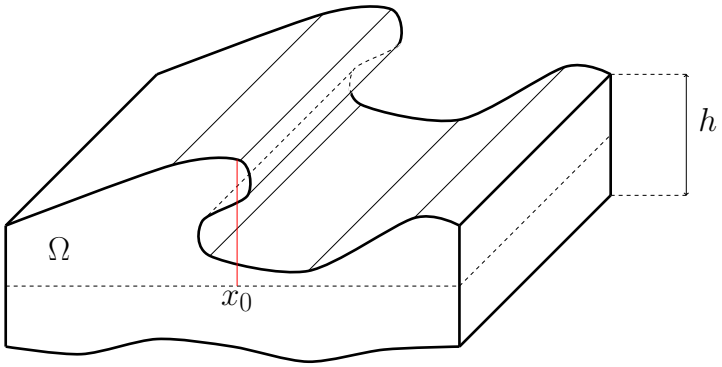


Figure 2.2: Shape that is not reachable with the thickness parametrization.

When the thickness is small the shape can be seen as a plate, and we can assume that the study takes place in the framework of small deformations. For example we can place ourselves in the framework of the *membrane model* of Section 2.1.1. The plate is assumed to be filled with a homogeneous isotropic material. In this case, the physical behavior of the plate is described by the second-order tensor B, which depends linearly on the thickness, and is defined by

$$B(h) = h\mathrm{Id}.$$

This equation characterizes the choice of the material that is made. For a given volumic load $f\, e_z$ with $f \in L^2(\Omega; \mathbb{R})$, the vertical displacements - $u(h) : \Omega \to \mathbb{R}$ - of the shape $\Omega_{\mathbb{R}^3}$ under this load are modeled by the solution to the following boundary-value problem

$$\left\{ \begin{array}{rcll} -\mathrm{div}\,(B(h)\nabla u(h)) & = & f & \text{in } \Omega, \\ u & = & 0 & \text{on } \partial\Omega. \end{array} \right. \tag{2.6}$$

This model amounts notably to assuming planar invariance in the z-direction for the physical behavior of $\Omega_{\mathbb{R}^3}$. The solution u to the problem (2.6) depends on the thickness through the tensor B. In this context, the shape optimization

reduces to the optimization of the thickness h in the convex set

$$\mathcal{U}_{ad} = \{h \in L^\infty(\Omega) \mid \forall x \in \Omega,\ 0 < h_m \leq h(x) \leq h_M\}.$$

The objective is to minimize a function J defined by

$$J(h) = \int_\Omega j\big(h, u(h)\big),$$

with $h \in \mathcal{U}_{ad}$ and $j : \Omega \times H^1(\Omega;\mathbb{R}) \to \mathbb{R}$. The optimization problem reads

$$\min_{h\in\mathcal{U}_{ad}} J(h). \tag{2.7}$$

The set $\mathcal{U}_{ad}$ is not relatively compact for the strong topology of $L^\infty(\Omega)$ (see [7, Remarque 5.4]). As a consequence there is no existence results for optimal shapes in this space. This is well explained in [7, Proposition 5.2] where a counter-example is given. It is for example possible to build a sequence of shapes with an increasing rigidity with a constant volume. This seems possible by considering shapes with a infinite number of bars whose volume tends to zero. In fact, this sequence can converge but not in the space $\mathcal{U}_{ad}$. This could be seen as the need to authorize some composite material, made by superposition of different layers of materials at the microscopic scale. It also happens that with regularity constraints on the thickness, there might be existence of optimal solutions [7, §5.2.3].

2.2.2 Topology optimization

In $\mathbb{R}^2$ optimizing the topology of a shape consists in finding the best repartition of holes in the shape. The first methods authorizing topological changes used a relaxed characteristic function. This is at the basis of the homogenization method [6, 89, 90] or the SIMP (Solid Isotropic Material with Penalization) method [20, 21] which is kind of a rough approximation of the latter. These two methods consider materials with intermediate densities, and therefore the classical notion of shape is lost. Eventually, one can interpret an optimal shape considering low densities as holes and densities close to 1 as full material : this is the role of the penalization process. The final shapes then obtained are still classical shapes, but may lose their optimal character.

One can also mention another topological method that deals with a classical notion of shape. It consists in working with a characteristic function and to compute topological sensitivities that evaluates the sensitivity with respect to nucleation of holes. We will not elaborate on this method here and refer to [17, 93, 121] for more details. The computation of topological derivatives is usually technical. Moreover, when an optimization process creates many small holes in the same area, it works as a larger hole, and at the end the topological gradient has no real sense in this area. This is one of the main difficulties of this method.

Homogenization

The most general principle for shape optimization is to optimize the characteristic function χ of a shape Ω. This function satisfies $\chi(x) = 0$ if x is in the domain Ω, and $\chi(x) = 0$ if the point is outside. The shape Ω is considered to be included in a bounded open set $\mathcal{D}$, so that the characteristic function χ lies in $L^\infty(\mathcal{D})$. There is a priori no way to benefit from a notion of derivative for this kind of variable. Moreover, from a numerical point of view, the optimization turns out to lead to combinatorial issues. With a mesh of size N, there is 2^N different characteristic functions. Therefore, even for small meshes it is unmanageable to compare all configurations and to keep the best one. In the case of thickness optimization, it is known that there is a priori no optimal solution in $L^\infty(\mathcal{D})$. There might be minimizing sequences, but they may not converge in the space $L^\infty(\mathcal{D})$ which is not relatively compact. One first way to get existence results is to restrict the admissible space by imposing more regularity on the solutions. To the contrary, the homogenization method consists in enlarging the space of admissible shapes by adding the possible limits of minimizing sequences, namely the composite solutions. A composite material is characterized by a material density, and its physical behavior given by the one of its microstructure. To that end, the problem is relaxed by replacing the characteristic function by a density function ρ varying in $[0, 1]$. When the density is 1 it corresponds to a full material, and when it vanishes it corresponds to void. Otherwise the shape is made of a composite material with density ρ.

Now we introduce some notions and results for the homogenization method, starting with the definition of an homogenized tensor. We start by the notion of periodic homogenization. Let Ω be a periodic domain of period ε. The space $\mathbb{R}^d$ is paved with ε-homotheties of the unit cell $Y = [0, 1]^d$ such that $\mathbb{R}^d = \bigcup_{i\in\mathbb{Z}^d} Y_{i,\varepsilon}$ where $Y_{i,\varepsilon} = \varepsilon(i + Y)$. We denote with an indice # a Y-periodic space : for example $L^\infty_\#(\Omega)$ is the space of $L^\infty(\Omega)$ functions that are Y-periodic. The physical behavior of the shape is modeled by the tensor A. In this case, the tensor is assumed not to be constant but to be Y-periodic :

$$\forall 1 \leq i \leq d,\ A(y + e_i) = A(y), \quad \text{where } (e_i)_{1\leq i\leq d} \text{ is the canonical basis of } \mathbb{R}^d.$$

With a source term $f \in L^2(\Omega;\mathbb{R})$ and Dirichlet boundary conditions, we consider the *scalar thermal model* of Section 2.1.4. The model problems now reads

$$\left\{ \begin{array}{rcl} -\mathrm{div}\left(A\left(\dfrac{x}{\varepsilon}\right)\nabla u_\varepsilon\right) &=& f \quad \text{in } \Omega, \\ u_\varepsilon &=& 0 \quad \text{on } \partial\Omega, \end{array} \right. \tag{2.8}$$

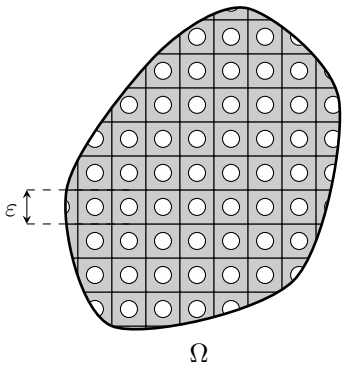


Figure 2.3: Periodic media with period ε.

Most of the time the exact knowledge of the microstructure and its physical property $A\left(\frac{x}{\varepsilon}\right)$ is not necessary. Therefore we usually prefer to consider the average behavior given by the source term f. This is the main idea of the homogenization method.

The average behavior of the solution u_ε to (2.8) can be estimated by the two-scale asymptotic expansion method. It consists in assuming that u_ε can be written under the form

$$u_\varepsilon(x) = \sum_{i=0}^{\infty} \varepsilon^i u_i\left(x, \frac{x}{\varepsilon}\right),$$

where $\Omega \times Y \ni (x, y) \mapsto u_i(x, y)$ is periodic in its second variable $y \in Y$. In that case, the gradient of u_ε reads

$$\nabla u_\varepsilon = \varepsilon^{-1} \nabla_y u_0\left(x, \frac{x}{\varepsilon}\right) + \sum_{i=0}^{\infty} \varepsilon^i \left(\nabla_y u_{i+1} + \nabla_x u_i\right)\left(x, \frac{x}{\varepsilon}\right).$$

Now (2.8) becomes

$$\begin{aligned}
&- \varepsilon^{-2}\Big(\mathrm{div}_y\big(A\nabla_y u_0\big)\Big)\left(x, \frac{x}{\varepsilon}\right) \\
&- \varepsilon^{-1}\Big(\mathrm{div}_y\big(A(\nabla_x u_0 + \nabla_y u_1)\big) + \mathrm{div}_x\big(A\nabla_y u_0\big)\Big)\left(x, \frac{x}{\varepsilon}\right) \\
&- \sum_{i=0}^{\infty} \varepsilon^i \Big(\mathrm{div}_x\big(A(\nabla_x u_i + \nabla_y u_{1+1})\big) + \mathrm{div}_y\big(A(\nabla_x u_{i+1} + \nabla_y u_{i+2})\big)\Big)\left(x, \frac{x}{\varepsilon}\right) \\
&= f(x).
\end{aligned}$$

Then we can identify each power of ε. We will not details about that and refer to [7, §7.2.2]. The equation with ε^{-2} indicates that u_0 does not depend on the periodicity variable y. The equation with ε^{-1} indicates how to introduce the cell-problem (see below (2.9)) for computing u_1. Finally, integrating the equation with ε^0 defines the homogenized tensor (see below (2.10)).

Definition 2.2.1. *Let $Y = [0,1]^d$ and $(e_i)_{1\le i\le d}$ the canonical basis of $\mathbb{R}^d$. For all $1 \le i \le d$ let w_i be the solution to the following cell problem*

$$\begin{cases} -\mathrm{div}_y\Big(A(y)(e_i + \nabla_y w_i)\Big) = 0 & \text{in } Y, \\ y \mapsto w_i(y) & Y - \text{periodic.} \end{cases} \tag{2.9}$$

Let $A^\star$ be defined by

$$A^\star_{ij} = \int_Y A(y)\big(e_i + \nabla_y w_i(y)\big) \cdot \big(e_j + \nabla_y w_j(y)\big)\, dy. \tag{2.10}$$

$A^\star$ is said to be a homogenized tensor.

Definition 2.2.2. *Let $\rho \in [0,1]$. We define by G_ρ the closure of the set of all homogenized tensors $A^\star$ such that $\int_Y \chi(y)\, dy = \rho$. This is the set of all composite materials with the two materials α, β in proportion ρ. The set G_ρ is included in the set of symmetric matrices.*

It can also be interesting to consider the case where the material is a mix of two phases α, β. Let us consider the composite material defined by the characteristic function $\chi(x)$ such that the behavior of the material is given by the law A_χ defined by

$$A_\chi(x) = \alpha\chi(x) + \beta(1 - \chi(x)).$$

Given $f \in L^2(\mathcal{D};\mathbb{R})$, let $u : \mathcal{D} \to \mathbb{R}$ be the solution to

$$\left\{ \begin{array}{rcll} -\mathrm{div}\big(A_\chi(x)\nabla u_\chi(x)\big) & = & f & \text{in } \mathcal{D}, \\ u_\chi(x) & = & 0 & \text{on } \partial\mathcal{D}. \end{array} \right.$$

We aim at minimizing an objective function J defined by

$$J(\chi) = \int_{\mathcal{D}} j\big(x, u_\chi(x)\big)\, dx,$$

where and $j : \mathcal{D} \times H^1(\mathcal{D};\mathbb{R}) \to \mathbb{R}$ and χ being in $\mathcal{U}_{ad}$ defined by

$$\mathcal{U}_{ad} = \left\{ \chi \in L^\infty\big(\mathcal{D};[0,1]\big) \right\}.$$

The optimization problem reads

$$\min_{\chi\in\mathcal{U}_{ad}} J(\chi). \tag{2.11}$$

There is the following convergence result

Theorem 2.2.3. *Let χ_ε be a sequence of characteristic functions and A_ε be the tensor defined by*

$$A_\varepsilon = \alpha\chi_\varepsilon(x) + \beta(1 - \chi_\varepsilon(x)).$$

There exists a subsequence, also denoted by ε, a density $\rho \in L^\infty(\mathcal{D};[0,1])$ and a symmetric homogenized tensor $A^\star \in L^\infty\big(\mathcal{D};\mathbb{R}^d\big)$ such that χ_ε weakly converges towards ρ and A_ε converges in the sense of homogenization towards $A^\star$. This means that for all $f \in L^2(\mathcal{D};\mathbb{R})$, the solution u_ε to

$$\left\{ \begin{array}{rcll} -\mathrm{div}\big(A_\varepsilon(x)\nabla u_\varepsilon(x)\big) & = & f & \text{in } \mathcal{D}, \\ u(x) & = & 0 & \text{on } \partial\mathcal{D}, \end{array} \right.$$

weakly converges in $H^1(\Omega;\mathbb{R})$ towards the solution u to

$$\left\{ \begin{array}{rcll} -\mathrm{div}\big(A^\star(x)\nabla u(x)\big) & = & f & \text{in } \mathcal{D}, \\ u(x) & = & 0 & \text{on } \partial\mathcal{D}. \end{array} \right.$$

Moreover, for all $x \in \Omega$, $A^\star(x) \in G_{\rho(x)}$.

Now we can define the relaxation of the initial optimization problem. Let $\mathcal{U}^\star_{ad}$ be defined by

$$\mathcal{U}^\star_{ad} = \left\{ \big(\rho, A^\star\big) \in L^\infty\Big(\mathcal{D};[0,1]\times\mathcal{S}\Big) \mid \forall x \in \Omega,\ A^\star(x) \in G_{\rho(x)} \right\},$$

where $\mathcal{S}$ is the set of symmetric tensors. Denoting by $u^\star$ the solution to

$$\left\{ \begin{array}{rcll} -\mathrm{div}\big(A^\star(x)\nabla u^\star(x)\big) & = & f & \text{in } \mathcal{D}, \\ u^\star(x) & = & 0 & \text{on } \partial\mathcal{D}, \end{array} \right.$$

the optimization problem writes

$$\inf_{(\rho,A^\star)\in\mathcal{U}^\star_{ad}} J(\rho, A^\star), \tag{2.12}$$

where

$$J(\rho, A^\star) = \int_{\mathcal{D}} j\big(x, u^\star(x)\big)\, dx.$$

Theorem 2.2.4. *The homogenized formulation (2.12) is a relaxation of (2.11) in the following sense*

1. *there exists at least one solution to (2.12);*
2. *any minimizing sequence of (2.12) converges in the sense of homogenization to a minimizer of (2.11);*
3. *any optimal solution to (2.12) is the limit of a minimizing sequence of (2.11).*

As a consequence

$$\min_{\chi\in\mathcal{U}_{ad}} J(\chi) = \min_{(\rho,A^\star)\in\mathcal{U}^\star_{ad}} J(\rho, A^\star).$$

It is at first noticeable that the homogenization is a way to avoid the non-existence of solution to the optimization problem. Furthermore, since there always exists a minimizing sequence of characteristic functions converging to an optimal solution, it is also possible to get a shape as close to an optimum as needed for the initial problem (2.11). However, the definition of the homogenized tensor is implicit, and the set G_ρ is a priori known only in a few cases, like the case of laminated materials. At the end of the optimization process, the solution to the homogenized problem is a composite material and not a classical shape. Most of the time it is preferred to obtain a classical shape described by a characteristic function and not by material densities. For this purpose, at the latest steps of the optimization process, the intermediate densities are penalized [7, §7.5.5] so that the densities get either closer to 0 or to 1.

Remark 2.2.5. *As stated by Theorem 2.2.3, for $x \in \Omega$ we have $A^\star(x) \in G_{\rho(x)}$. This does not mean that the material is periodic. But at each point $x \in \Omega$, the tensor $A^\star(x)$ can be attained by a limit of a sequence of homogenized periodic materials with density $\rho(x)$. Moreover in the case where the objective to be minimized depends only on one load-case, for any optimal configuration $(\rho, A^\star)$ one can find another equivalent configuration such as the material is periodic, and more precisely a first-order laminate (see [7, Remarque 7.22]).*

The SIMP method

The SIMP (Solid Isotropic Method with Penalization) was introduce by M.P. Bendsøe in [20]. This method has been widely developed for topology optimization [21, 132], and is very popular for it is extremely easy to implement [117]. Similarly to the homogenization method, the SIMP method intends to optimize a material density instead of a characteristic function. It is not originally linked to the homogenization method, but can be seen as a simplification of it. Given two materials with laws α, β and a density function ρ, we introduce a mixed material with law tensor A_ρ defined by

$$A_\rho = \alpha\rho^p(x) + \beta(1 - \rho^p(x)),$$

where $p \in \mathbb{N}^\star$. The parameter p plays the role of the penalization, in order to force the density to converge to a characteristic function. The SIMP method optimizes a material density which parametrizes the shape. Even if it is now referred to as a topological optimization method, it is nothing but a parametric framework for shape optimization.

The optimization problem writes

$$\min_{\rho \in L^\infty(\mathcal{D};[0,1])} J(\rho), \tag{2.13}$$

where

$$J(\rho) = \int_{\mathcal{D}} j\big(x, u(x)\big)\, dx,$$

and $u : \mathcal{D} \to \mathbb{R}$ is the solution to

$$\left\{ \begin{array}{rcll} -\mathrm{div}\big(A_\rho(x)\nabla u(x)\big) & = & f & \text{in } \mathcal{D}, \\ u(x) & = & 0 & \text{on } \partial\mathcal{D}, \end{array} \right.$$

for $f \in L^2(\mathcal{D};\mathbb{R})$ is a source term. Similarly to the previous section on homogenization, we place ourselves in the framework of the *scalar thermal model* of Section 2.1.4. In the homogenization method, the tensor A is taken as a homogenized limit of the tensor of a composite material. The local material density, as well as the microstructure of the material are taken into account in the homogenized tensor. However, for most cases, the set of homogenized tensor is not known, which is the main difficulty of the method. To the contrary, in the SIMP method, the thermal tensor is explicit, and there is only a polynomial dependency on the density. There is absolutely no influence of the microstructure in the thermal tensor. In that sense the approximation is much simpler. The polynomial dependency on the density makes also all calculations straightforward. It is quite simple to get the differentiability - with respect to the density - of the solution u to the boundary problem, and therefore to the objective function J. This is one of the main advantages of the method. However, there is no convergence results. At first, there is no existence result of an optimal solution. Assuming a numerical method converges to some local optimum, there is also no insurance to find a classical shape in its vicinity. Therefore the penalization used to force the optimization to converge to a classical solution may degrade considerably the optimality of the solution.

The approximation of the thermal behavior of the mixed material has a priori nothing to do with a homogenized tensor. When $p = 1$, it can be seen [7, §7.6.5] that it is an upper bound on the homogenized tensor, and therefore the SIMP solutions are not as good as the homogenized ones, since the material is assumed to be more rigid than it really is. In the two-dimensional case, if we consider $h_M = 1$ and $\beta = 0$, the SIMP method and the thickness optimization coincide (provided that the density satisfies $\rho(x) \geq \rho_m > 0$ in the SIMP method). A counter-example of non-existence of optimal solutions ([7, Proposition 5.2]) for the thickness optimization is also relevant for the SIMP method. Lastly let us mention that the penalization parameter p has a great influence on the behavior of numerical methods. Increasing p makes the algorithms converge to a characteristic function, but when p is too high at the beginning, the optimization may stop at a local optimum. Therefore, tuning p is a matter of great importance, but without a lot of mathematical guidelines.

Moreover, since there is no exact representation of the shape, there is no access to geometric quantities such as curvatures, thickness, or even the perimeter of the shape.

2.2.3 Geometric optimization

In geometric optimization, the variable is the shape itself, or equivalently the boundary. The shape is not governed by a set of parameters such as lengths, curvatures, etc. We first introduce the set of smooth bounded open sets : for $k \in \mathbb{N}$ let $\mathcal{O}_k$ be defined by

$$\mathcal{O}_k = \left\{ \Omega \subset \mathbb{R}^d \mid \Omega \text{ is a bounded open set of class } C^k \right\}, \tag{2.14}$$

where a set is said to be of class C^k when its boundary is of class C^k. We refer to Definition 3.2.1 for more details on smooth domains.

We consider a domain $\Omega \in \mathcal{O}_k$ filled with homogeneous isotropic material. The model is the *Elastic model* of Section 2.1.5 with Hooke's tensor A. The Hooke's tensor is constant and does not depend on the shape, and the deformation tensor ε is classically defined by $\varepsilon(u) = \frac{1}{2}\left(\nabla u + \nabla u^T\right)$. With $f \in L^2(\Omega;\mathbb{R}^d)$ let $u_\Omega : \Omega \to \mathbb{R}^d$ be the solution to

$$\left\{ \begin{array}{rcll} -\mathrm{div}\big(A\varepsilon\left(u_\Omega\right)\big) & = & f & \text{in } \Omega, \\ u_\Omega & = & 0 & \text{on } \partial\Omega. \end{array} \right. \tag{2.15}$$

The goal is to minimize a function J defined by

$$J(\Omega) = \int_\Omega j\big(\Omega, u_\Omega\big),$$

with respect to Ω being in

$$\mathcal{U}_{ad} = \mathcal{O}_k,$$

and where $j : \mathcal{U}_{ad} \times H^1\big(\mathbb{R}^d;\mathbb{R}^d\big) \to \mathbb{R}$. The optimization problem reads now

$$\min_{\Omega \in \mathcal{U}_{ad}} J(\Omega). \tag{2.16}$$

Where the sufficient smoothness of the shape Ω may be more detailed (see Section 3.2). We can notice that in this case the integration domain Ω is the optimization variable as well as the domain on which the boundary-value problem (2.15) takes place. This is the main difficulty for computing shape derivatives. The shape sensitivity analysis can be performed with respect to smooth variations of the boundary. We follow here the formalism introduced by Hadamard [66] that was widely developed [56, 72, 86, 88, 122]. With a given reference domain Ω, in the framework introduced by Murat and Simon [88, 118, 119], we consider domains of the form

$$\Omega_\theta = \big(\mathrm{Id} + \theta\big)(\Omega),$$

where $\theta \in W^{1,\infty}\big(\mathbb{R}^d;\mathbb{R}^d\big)$. Now we can define a notion of derivation with respect to the shape. We also refer to Section 3.2.2 for more details.

Definition 2.2.6. *The shape derivative of $J(\Omega)$ at Ω is defined as the Fréchet derivative in $W^{1,\infty}\big(\mathbb{R}^d;\mathbb{R}^d\big)$ at 0 of $\theta \mapsto J\Big(\big(\mathrm{Id} + \theta\big)(\Omega)\Big)$. The shape derivative is denoted $J'\left(\Omega;\theta\right)$ or J'_θ when there is no ambiguity.*

Since smooth variations of the boundary do not include topological change, a geometric method works a priori with a fixed topology. However it is possible to mix a geometric shape sensitivity analysis to a representation of the shape that allows topological changes. This is the case of the level-set method presented in Chapter 4.

2.3 Existence of optimal shapes

2.3.1 A counter-example

After the choice of representation and modeling of the shape it remains another major difficulty inherent to shape optimization. This is a theoretical aspect of shape optimization that may influence the choice of representation of a shape. This is the fact that there is a priori no existence of optimal shapes. Let us explain this problem with one example taken from [7, §6.2.1]. We start with a working domain $\mathcal{D} =]0,1[^2$, in the two-dimensional space. This domain is filled with two homogeneous, isotropic materials with properties α, β such that $\beta > \alpha > 0$ and $\beta \gg \alpha$. This consists in assuming that there is one rigid (with property β) and one soft (with property α) material. The shape to be optimized - Ω - is the area filled with the "solid" material, whereas the "soft" material is given by the shape $\mathcal{D}\backslash\Omega$. We define by $\chi(x)$ the characteristic function of the domain Ω. The global property of the domain $\mathcal{D}$ is now defined by

$$a_\chi = \alpha\chi + \beta(1-\chi).$$

With (e_1, e_2) being the canonical basis of $\mathbb{R}^2$, the domain $\mathcal{D}$ is subject to a uniform pressure load in the direction e_1, and to no volume force (see Figure 2.4). Therefore its displacement u is given by the solution to

$$\left\{ \begin{array}{rcll} -\mathrm{div}\big(a_\chi \nabla u\big) & = & 0 & \text{in } \mathcal{D}, \\ a_\chi \nabla u \mathbf{n} & = & e_1 \cdot \mathbf{n} & \text{on } \partial\mathcal{D}. \end{array} \right.$$

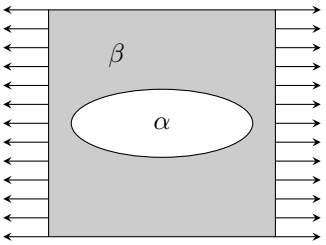


Figure 2.4: Plate under uniform load.

The function to be minimized is the following

$$J(\chi) = \int_{\partial\mathcal{D}} (e_1 \cdot \mathbf{n})\, u.$$

This consists in minimizing the deformation energy of the plate under the pressure load. Without any other condition, it is easy to see that it suffices to fill the domain $\mathcal{D}$ with the "solid" material to minimize J. This is the reason why a constraint on the proportion of this material is added : given a proportion $0 < \theta < 1$ we introduce

$$\mathcal{U}_{ad} = \left\{ \chi \in L^\infty(\mathcal{D}; \{0,1\}) \;\middle|\; \frac{1}{|\mathcal{D}|} \int_{\mathcal{D}} \chi = \theta \right\}.$$

Now the optimization problem reads

$$\min_{\chi \in \mathcal{U}_{ad}} J(\chi). \tag{2.17}$$

Theorem 2.3.1. *There is no optimal solution to (2.17).*

The proof of this theorem is rigorously detailed in [7]. The main reason why the optimization problem does not have a solution can be explained with the following picture. Starting from a shape with a given repartition of holes, it is always possible to get a better shape by adding a new hole while preserving the proportion of "solid" material. There is a theoretical lower bound on the deformation energy, and for any proportion θ of "solid" material, it is possible to get as close as we want to this lower bound, by adding new holes, such as depicted in Figure 2.5. However, it is not

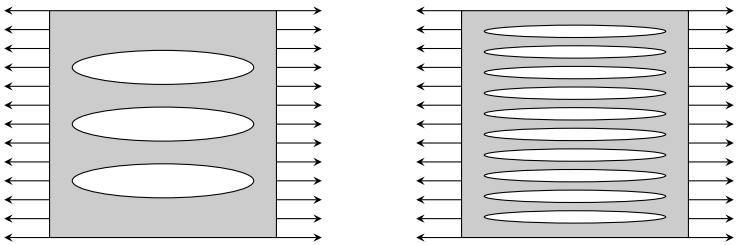


Figure 2.5: Minimizing sequence for (2.17).

possible to achieve the lower bound with a classical notion of shape.

One can find another interesting example of non-existence for a shape optimization problem in solving the heat equation with Dirichlet boundary condition in [30, Theorem 4.2.1] or [52, Theorem 2.1].

2.3.2 Remedies to the non-existence

Shape optimization problems are a priori ill-posed in the sense that there may not exist optimal shapes. In order to avoid this, there is basically two kinds of methods. On the first hand, it is possible to change the problem by restricting the set of admissible shapes. On the other hand, one can also enlarge this set by adding all possible limits of minimizing sequences.

- *Restriction of the set of admissible shapes* : the non-existence of optimal shapes is mainly due to the fact that minimizing sequences may converge to a design that is not in the set of admissible shapes. As stated in the previous counter-example, adding holes allows to improve the shape with respect to the objective function. In the end, adding too many holes leads to obtaining a shape made of composite material, which does not correspond to a "classical" shape. The restriction of the set of admissible shapes intends to ban the creation of too many holes. This has been widely detailed in the literature and we mention here a few methods for restricting this set.

- *Adding a constraint on the perimeter* : the non-existence of classical optimal shapes may comes from the fact that letting the perimeter go to infinity improves the shape on some cases. A porous or composite material is a way to allow the perimeter to go to infinity. Therefore, adding a constraint or a penalization on the perimeter may ensure the existence of classical optimal shapes (see for example [72, §4.6]).
- *Imposing more regularity* : when the number of holes goes to infinity, their sizes must go to zero. Therefore, their curvatures develop singularities. Restricting the set of admissible shapes to only regular shapes would then have the desired effect. There are many different ways to impose more regularity on a shape. For example one can require that shapes satisfy the ε-cone property (see [72, §4.3] or [7, §6.2.2]) which is kind of a regularity condition on the boundary. It is also possible to restrict the set of admissible shapes to shapes that are reachable by a smooth diffeomorphism starting from a reference shape, following the formalism introduced by Murat and Simon [88, 87].
- *Adding a topological or capacity constraint* : the topology of a connected shape - in the two-dimensional space - can be defined by the number of holes (in the three-dimensional space, it is more complicated). Imposing a topology prevents an optimization method from adding holes, and it was then proven [36, 123] in two space dimensions, under assumptions on the number of connected components, that the optimization problem has at least one solution. Other kind of constraints on the shape, such as uniform capacity constraints (see [72, §3.4.3]) may also lead to a well-posed problem.
- *Establishing a monotonous property* : It appears that when the objective function is decreasing with respect to inclusion, there is an existence theorem for minimal shapes. This result also mentioned in [72, Théorème 4.7.6] is due to [33]. This is not strictly speaking a remedy to ill-posedness of the shape optimization problem, but this case of existence is itself interesting.

- *Relaxation of the problem* : since a minimizing sequence may not converge in the set of admissible shapes, another idea is to enlarge this set with all limits of minimizing sequences. This is at the basis of the homogenization method, where generalized shapes made of composite materials are considered.

Chapter 3

Examples of objective functions and optimality conditions

Contents

In this section we detail some different objective functions in both the parametric case of thickness optimization (Section 2.2.1) and the geometric formalism (Section 2.2.3) the latter being first detailed in Section 3.2. Their first-order derivatives in the corresponding formalisms are also given whereas the higher-order derivatives are detailed in Part II. As regards mechanical criterion, we focus on the compliance and a least square criterion on the displacements. These are two usual functions widely used in the literature. The compliance is one of the simplest way to consider the rigidity of the shape since it corresponds to the deformation energy of the shape under an applied load. We also consider a least square displacement criterion which has the interest not to be self-adjoint, contrary to the compliance. For the geometric criterion, we consider the volume, which is the most natural one. We also focus on a maximum-thickness criterion, which allows one to take into account manufacturing constraints. We follow here the same line as in [83], and therefore we need to also introduce the signed distance function and its shape derivative.

3.1 Thickness optimization

We come back to the formalism presented in Section 2.2.1 in the framework of the *membrane model* of Section 2.1.1. With $\Omega \subset \mathbb{R}^2$ a smooth bounded open set as a reference domain, $0 < h_m \leq h_M$ and $h \in L^\infty\big(\Omega; [h_m, h_M]\big)$ we consider the membrane given by

$$\Omega_{\mathbb{R}^3} = \{X \in \mathbb{R}^3 \,|\, X = (x, y, z) \text{ with } (x, y) \in \Omega \text{ and } 0 \leq z \leq h(x, y)\}.$$

The boundary of Ω is divided in three parts, Γ_N, Γ_D and Γ such that

$$\partial\Omega = \Gamma_N \cup \Gamma_D \cup \Gamma,$$

where each part Γ, Γ_N and Γ_D have a non-zero measure. Given $f \in L^2(\Omega;\mathbb{R})$ and $g \in H^1(\Omega;\mathbb{R})$ such that $g = 0$ on Γ, the out-of-plane displacements of the membrane are modeled by $u : \Omega \to \mathbb{R}$ the solution to the following problem :

$$\left\{ \begin{array}{rcll} -\mathrm{div}\,(h\nabla u) & = & f & \text{in } \Omega, \\ h\partial_{\mathbf{n}} u & = & g & \text{on } \Gamma_N \cup \Gamma, \\ u & = & 0 & \text{on } \Gamma_D. \end{array} \right. \tag{3.1}$$

The domain $\Omega_{\mathbb{R}^3}$ is replaced by Ω here by integrating along the out-of-plane direction and assuming planar invariance. The set of admissible shapes is taken as

$$\mathcal{U}_{ad} = \{h \in L^\infty(\Omega;\mathbb{R}) \mid h_m \leq h \leq h_M\}.$$

We recall that for an objective function J, the optimization problem reads

$$\min_{h \in \mathcal{U}_{ad}} J(h). \tag{3.2}$$

Remark 3.1.1. *There might be any additional constraint on the optimization problem, such that the set $\mathcal{U}_{ad}$ is of the form*

$$\mathcal{U}_{ad} = \{h \in L^\infty(\Omega;\mathbb{R}) \mid h_m \leq h \leq h_M \text{ and } G(h) \leq 0\}.$$

In that case we refer to Section 1.1.3 and Theorem 1.1.17 for writing the optimality conditions that imply to differentiate both J and G.

In order to find the solution to (3.1) we introduce the variational space

$$\mathcal{V} = \left\{u \in H^1(\Omega;\mathbb{R}) \mid u = 0 \text{ on } \Gamma_D\right\}.$$

The variational formulation of (3.1) now reads

$$\left\{ \begin{array}{l} \text{find } u \in \mathcal{V}, \text{ such that }, \\ \forall v \in \mathcal{V}, \ \displaystyle\int_\Omega h\nabla u \cdot \nabla v = \int_\Omega fv + \int_{\Gamma_N} gv. \end{array} \right.$$

In this introduction to parametric optimization, we will focus on three criterion : the volume of the shape, the compliance and a least square criterion on the displacements.

3.1.1 Volume

The volume of the shape is defined by

$$V(h) = \int_\Omega h. \tag{3.3}$$

Since it is completely linear in the thickness h and the domain Ω does not vary, the differentiability is straightforward. The derivative is given by the following lemma.

Lemma 3.1.2. *The volume : $\mathcal{U}_{ad} \ni h \mapsto V(h)$ is differentiable with respect to h. Its derivative in the direction $k \in L^\infty(\Omega)$ is given by*

$$V'(h;k) = \int_\Omega k.$$

3.1.2 State equation

Provided the data f, g have the assumed regularity ($f \in L^2(\Omega;\mathbb{R})$ and $g \in H^1(\Omega;\mathbb{R})$), the displacement u solution to (3.1) is differentiable with respect to h. The same kind of result is proven in [7, Lemme 5.15] but only with volume forces and no boundary force. The sketch of the proof remains exactly the same.

Lemma 3.1.3. *The function $\mathcal{U}_{ad} \ni h \mapsto u(h) \in \mathcal{V}$ is differentiable with respect to h. Its derivative in the direction $k \in L^\infty(\Omega;\mathbb{R})$, denoted by $u'(h;k)$ - or u'_k - is solution in $\mathcal{V}$ to*

$$\left\{ \begin{array}{rcll} -\mathrm{div}\,(h\nabla u'_k) & = & \mathrm{div}\,(k\nabla u) & \text{in } \Omega, \\ h\partial_{\boldsymbol{n}} u'_k & = & -k\partial_{\boldsymbol{n}} u & \text{on } \Gamma_N \cup \Gamma, \\ u & = & 0 & \text{on } \Gamma_D. \end{array} \right. \tag{3.4}$$

3.1.3 Compliance

The compliance is defined by

$$J_1(h) = \int_\Omega h\,|\nabla u|^2 = \int_\Omega fu + \int_{\Gamma_N} gu.$$

Lemma 3.1.4. *Let $f \in L^2(\Omega;\mathbb{R})$ and $g \in H^1(\Omega;\mathbb{R})$. The compliance is differentiable with respect to h. Its derivative in the direction $k \in L^\infty(\Omega;\mathbb{R})$ is given by*

$$J_1{}'(h;k) = \int_\Omega fu'_k + \int_{\Gamma_N} gu'_k = -\int_\Omega k|\nabla u|^2.$$

Proof. Once the differentiability of u is established, since there is no variation of Ω and Γ_N with respect to the thickness h, the first expression is straightforward. To obtain the second one, it suffices to write the variational formulation for u and u'_k that are, for $v \in \mathcal{V}$

$$\int_\Omega h\nabla u\cdot\nabla v = \int_\Omega v + \int_{\Gamma_N} gv, \tag{3.5}$$

$$\int_\Omega h\nabla u'_k\cdot\nabla v = -\int_\Omega k\nabla u\cdot\nabla v. \tag{3.6}$$

Taking $v = u'_k$ in the first equality, and $v = u$ in the second one leads to the expected result. □

3.1.4 Least square displacement criterion

Given a target displacement u_0 we consider the least square criterion J_2 defined by

$$J_2(h) = \int_\Omega |u-u_0|^2.$$

Lemma 3.1.5. *Let $f \in L^2(\Omega;\mathbb{R})$ and $g \in H^1(\Omega;\mathbb{R})$. The function $h \longmapsto J_2(h)$ is differentiable on $L^\infty(\Omega;\mathbb{R})$. Let p - the adjoint state - be the solution in $\mathcal{V}$ to the following boundary-value problem*

$$\left\{\begin{array}{rcll} -\mathrm{div}\big(h\nabla p\big) &=& 2(u-u_0) & \Omega,\\ p &=& 0 & \Gamma_D,\\ h\partial_{\boldsymbol{n}}p &=& 0 & \Gamma_N\cup\Gamma. \end{array}\right. \tag{3.7}$$

Then the derivative of J_2 in the direction $k \in L^\infty(\Omega;\mathbb{R})$ is given by

$$\begin{aligned} J_2{}'(h;k) &= 2\int_\Omega (u-u_0)\,u'_k\\ &= -\int_\Omega k\,\nabla u\cdot\nabla p. \end{aligned}$$

Proof. The derivative of J_2 in the direction k immediately gives

$$J_2{}'(h;k) = 2\int_\Omega (u-u_0)\,u'_k\,dx.$$

The variational formulation associated to (3.7) is

$$\forall\varphi\in\mathcal{V},\ \int_\Omega h\,\nabla p\cdot\nabla\varphi = 2\int_\Omega (u-u_0)\varphi,$$

and the variational formulation for u'_k gives

$$\forall\varphi\in\mathcal{V},\ \int_\Omega h\,\nabla u'_k\cdot\nabla\varphi = -\int_\Omega k\,\nabla u\cdot\nabla\varphi.$$

Then

$$\begin{aligned} J_2{}'(h;k) &= 2\int_\Omega (u-u_0)\,u'_k\\ &= \int_\Omega h\,\nabla u'_k\cdot\nabla\varphi\\ &= -\int_\Omega k\,\nabla u\cdot\nabla p. \end{aligned}$$

□

Similarly to u one can check that the adjoint state p defined by (3.7) is differentiable.

Lemma 3.1.6. *With the same regularity assumption on f, g as previously, the function $h \longmapsto p(h)$ is differentiable w.r.t h. Its derivative in the direction k - namely $p'(h;k)$ or p'_k - is the solution in $\mathcal{V}$ of the following boundary-value problem*

$$\left\{\begin{array}{rcll} -\mathrm{div}\big(\nabla p'_k\big) & = & \mathrm{div}\big(k\nabla p\big) + 2u'_k & \Omega, \\ p'_k & = & 0 & \Gamma_D, \\ h\partial_{\boldsymbol{n}} p'_k & = & -k\partial_{\boldsymbol{n}} p & \Gamma_N \cup \Gamma. \end{array}\right. \tag{3.8}$$

The associated variational formulation writes

$$\forall \varphi \in \mathcal{V},\ \int_\Omega h\,\nabla p'_k \cdot \nabla\varphi = 2\int_\Omega u'_k\varphi - \int_\Omega k\,\nabla p\cdot\nabla\varphi.$$

3.2 Shape differentiation for geometric optimization

Differentiation with respect to a shape (an open subset of $\mathbb{R}^d$ with $d = 2$ or 3) is a key tool in shape optimization, that was first introduced by Hadamard [66]. It was then widely developed by many authors [7, 56, 72, 86, 88, 118], [122] (and references therein). There are two variants of the Hadamard method of shape differentiation. The first one, advocated by Murat and Simon [88, 118, 119], is based on a parametrization of shapes by displacement vector fields. Given a reference open set Ω of $\mathbb{R}^d$, a variation of this domain is of the type $\Omega_\theta = \big(\mathrm{Id} + \theta\big)(\Omega)$, where θ is a vector field from $\mathbb{R}^d$ into $\mathbb{R}^d$ and Id is the identity operator on $\mathbb{R}^d$. In other words, any point $x \in \Omega$ is moved to a new position $x + \theta(x) \in \Omega_\theta$. In this context, shape differentiation is defined as differentiation with respect to the vector field θ. The second approach is the so-called speed method, introduced by Zolésio and co-workers [55, 54, 122, 134], which is based on the flow of a vector field and on shapes evolving along this flow. For a given vector field $V(t,x)$, defined from $\mathbb{R}_+ \times \mathbb{R}^d$ into $\mathbb{R}^d$, consider the solution (or flow) of the ordinary differential equation

$$\left\{\begin{array}{l} \dfrac{\partial X_V}{\partial t}(t,x) = V\big(t, X_V(t,x)\big) \quad \text{ for } t > 0\,, \\ X_V(0,x) = x\,. \end{array}\right. \tag{3.9}$$

Then the variation of the reference domain Ω is defined, for $t \geq 0$, as $\Omega_t = X_V(t,\Omega) = \{X_V(t,x) \text{ such that } x \in \Omega\}$. In this context, shape differentiation is defined as the derivative with respect to t and it is a directional derivative in the direction of V. These two variants of the Hadamard method of shape differentiation lead to the same notion of first-order derivative by identifying the vectors fields $\theta(\cdot)$ and $V(0,\cdot)$ but to slightly different second-order derivatives. There are strong connections between them in the sense that results obtained with one method can be translated to similar results for the other one.

The second-order shape derivatives have already been studied in the literature. Simon was one of the first to study the second-order derivatives for shape optimization [119]. He notably explained the particular link between the first and second-order shape derivatives in the framework of the displacement field method. For the speed method, Delfour and Zolésio found a similar link between the first and second order derivatives [54]. Some numerical experiments can be found for electromagnetic problems in [95] where apparently there is no trouble in solving the Newton equation that requires to solve a linear system with the Hessian matrix. The particular structure of the shape Hessian was next studied by Novruzi and Pierre [94]. They proved that the second-order derivatives are composed of two maps (we denote them l_1 and l_2). More recently, there is also the work of Dambrine and co-workers [4, 48, 49, 50, 51]. They details different second-order computation, but also focus on properties of the shape Hessian such as coercivity and compactness on some cases. They also worked on the necessary assumptions on the shape Hessian to ensure the stability of a critical shape.

In this section, we first provide some background of differential geometry. Secondly we recall the two usual frameworks for computing shape derivatives, namely domain perturbation (or displacement field method) [88, 118, 119] and time moving domain (or speed method) [31, 55, 54, 122], and the respective structures of the second-order shape derivatives.

3.2.1 Differential geometry on domain boundaries

For the sake of completeness, we recall the basics of domain variations (see also [72, §5]) .

Definition 3.2.1 ([72, Theorem 2.4.5])**.** *A bounded open set Ω is said to have a Lispchitz boundary if, for any $x_0 \in \partial\Omega$, there exists a local Cartesian coordinate system with origin $x_0 = 0$, a cylinder $K = K'\times]-a,a[$ with $K' \subset \mathbb{R}^{d-1}$, an open ball with center O and radius r, and a Lipschitz function $\varphi : K' \to]-a,a[$ with $\varphi(0) = 0$ such that*

$$\begin{array}{c} \partial\Omega \cap K = \big\{\big(x', \varphi(x')\big) \mid x' \in K'\big\}, \\ \Omega \cap K = \big\{\big(x', x_N\big) \in K \mid x_N > \varphi(x')\big\}. \end{array}$$

If, in addition the function φ can be chosen of class C^k, the domain Ω is said to be of class C^k.

Let us consider a domain Ω of class C^1, and use the notation introduced in the above definition. For $x = (x', x_N) \in K$, we set $F(x) = \varphi(x') - x_N$. We have

$$\forall x \in \partial\Omega \cap K,\ F(x) = 0,$$
$$\forall x \in \Omega \cap K,\ F(x) < 0.$$

The local derivative of F gives the outer normal on $\partial\Omega \cap K$

$$\forall x \in \partial\Omega \cap K,\ \mathbf{n}(x) = \frac{1}{\sqrt{1 + |\nabla_{x'}\varphi(x')|^2}} \begin{pmatrix} \nabla_{x'}\varphi(x') \\ -1 \end{pmatrix}. \tag{3.10}$$

Let $k \in \mathbb{N}^\star$. We set

$$\mathcal{O}_k = \{\Omega \subset \mathbb{R}^d \mid \Omega \text{ is a bounded open set of class } C^k\}. \tag{3.11}$$

We now introduce some notation about tangential derivation.

Definition 3.2.2. *Let $k \geq 1$. Let V, W be two vector fields in $C^k\big(\mathbb{R}^d;\mathbb{R}^d\big)$, with respective components V_i, W_i. The j-th partial derivative of V_i is denoted by $\partial_j V_i$, and ∇V is the matrix defined by $[\nabla V]_{ij} = \partial_j V_i$. It is convenient to use Einstein's convention which means that repeated indices indicate summation from 1 to d. We define the operator $V \cdot \nabla$ that acts on scalar fields of class C^{k+1} as*

$$\begin{array}{rccc} V\cdot\nabla & : & C^{k+1}\big(\mathbb{R}^d;\mathbb{R}\big) & \to & C^k\big(\mathbb{R}^d;\mathbb{R}\big) \\ & & g & \mapsto & V_i\partial_i g, \end{array}$$

and on a vector fields of class C^{k+1} as

$$\begin{array}{rccc} V\cdot\nabla & : & C^{k+1}\big(\mathbb{R}^d;\mathbb{R}^d\big) & \to & C^k\big(\mathbb{R}^d;\mathbb{R}^d\big) \\ & & W & \mapsto & V\cdot\nabla W, \end{array}$$

where the j-th component of $V \cdot \nabla W$ is given by

$$[V \cdot \nabla W]_j = V_i \partial_i W_j.$$

Definition 3.2.3. *Let $k \geq 1$. Let $\Omega \in \mathcal{O}_{k+1}$ with outer normal $\boldsymbol{n}$. Let $V \in C^k\big(\mathbb{R}^d;\mathbb{R}^d\big)$. We define the tangential component $V_\Gamma \in C^k\big(\partial\Omega;\mathbb{R}^d\big)$ of the vector field V by*

$$V_\Gamma = V - (V \cdot \boldsymbol{n})\,\boldsymbol{n}.$$

Definition 3.2.4. *Let $k \geq 1$. Let $\Omega \in \mathcal{O}_{k+1}$ with outer normal $\boldsymbol{n}$. We define the tangential gradient ∇_Γ by*

$$\begin{array}{rccc} \nabla_\Gamma & : & C^{k+1}\big(\mathbb{R}^d;\mathbb{R}\big) & \to & C^k\big(\partial\Omega;\mathbb{R}^d\big) \\ & & v & \mapsto & \nabla_\Gamma v, \end{array}$$

where $\nabla_\Gamma v$ is the tangential component of the gradient of v :

$$\forall x \in \partial\Omega, \quad \nabla_\Gamma v(x) = \nabla v(x) - \big(\nabla v(x) \cdot \boldsymbol{n}(x)\big)\boldsymbol{n}(x).$$

For a vector field $V \in C^{k+1}\big(\mathbb{R}^d;\mathbb{R}^d\big)$, the tangential gradient $\nabla_\Gamma V$ is the matrix-valued field on $\partial\Omega$ defined by

$$\forall x \in \partial\Omega, \quad \big[\nabla_\Gamma V(x)\big]_{ij} = \big[\nabla_\Gamma V_i(x)\big]_j.$$

We now introduce an equivalence relation on $C^k\big(\mathbb{R}^d;\mathbb{R}\big)$ for functions that take the same values on $\partial\Omega$.

Definition 3.2.5. *Let $k \geq 1$. Let $\Omega \in \mathcal{O}_{k+1}$ with outer normal $\boldsymbol{n}$. Let $v_\gamma \in C^k\big(\partial\Omega;\mathbb{R}\big)$. We define the equivalence class of v_γ by*

$$cl(v_\gamma) = \{w \in C^k\big(\mathbb{R}^d;\mathbb{R}\big) \mid w_{|\partial\Omega} = v_\gamma\}.$$

We also denote by $\mathrm{ext}(v_\gamma)$ any element of $cl(v_\gamma)$. In particular $\mathrm{ext}(v_\gamma) \in C^k\big(\mathbb{R}^d;\mathbb{R}\big)$ and for $x \in \partial\Omega$, $\mathrm{ext}(v_\gamma)(x) = v_\gamma(x)$. We proceed similarly for vector-valued functions $V_\gamma \in C^k\big(\partial\Omega;\mathbb{R}^d\big)$, and use the same notation $cl(V_\gamma)$, and $\mathrm{ext}(V_\gamma)$.

Definition 3.2.6. *Let $k \geq 1$ and $\Omega \in \mathcal{O}_{k+1}$. We define the tangential gradient on $C^{k+1}\big(\partial\Omega;\mathbb{R}\big)$ by*

$$\begin{array}{rccc} \nabla_\Gamma & : & C^{k+1}\big(\partial\Omega;\mathbb{R}\big) & \to & C^k\big(\partial\Omega;\mathbb{R}^d\big) \\ & & v_\gamma & \mapsto & \nabla_\Gamma v_\gamma, \end{array}$$

where

$$\forall x \in \partial\Omega,\ \nabla_\Gamma v_\gamma(x) := \nabla \mathrm{ext}(v_\gamma)(x) - \big(\mathrm{ext}(v_\gamma)(x) \cdot \boldsymbol{n}(x)\big)\boldsymbol{n}(x). \tag{3.12}$$

Lemma 3.2.7. *The Definition 3.12 does not depend on the choice of $\mathrm{ext}(v_\gamma)$ in $cl(v_\gamma)$.*

Proof. If $v = 0$ on $\partial\Omega$, i.e if two extensions are in the same equivalence class, according to Definition 3.2.1 we have in local coordinates $v(x', \varphi(x')) = 0$. The derivation of this equality gives $\nabla_{x'} v + \nabla_{x'}\varphi \partial_{x_N} v = 0$, which leads to

$$\nabla v = \begin{pmatrix} \nabla_{x'} v \\ \partial_{x_N} v \end{pmatrix} = \partial_{x_N} v \begin{pmatrix} -\nabla_{x'}\varphi \\ 1 \end{pmatrix}.$$

With the help of (3.10), we get

$$\nabla v = -\sqrt{1 + |\nabla_{x'}\varphi(x')|^2}\, \partial_{x_N} v\, \mathbf{n}.$$

Therefore

$$\nabla_\Gamma v = 0.$$

□

Let $\xi \in C^k\big(\mathbb{R}^d; \mathbb{R}^d\big)$, and $\mathbf{n}$ be the outer normal to a shape $\Omega \in \mathcal{O}_{k+1}$. In order to simplify the notation, we set $\xi \cdot \mathbf{n} \equiv \xi_{|\partial\Omega} \cdot \mathbf{n} \in C^k\big(\partial\Omega; \mathbb{R}^d\big)$.

Definition 3.2.8. *Let $W \in C^1(\partial\Omega, \mathbb{R}^d)$. We define*

$$\mathrm{div}_\Gamma W = \mathrm{Tr}\big(D_\Gamma W\big),$$

where $\tilde{W}$ is a continuous extension of W on $\mathbb{R}^d$ and $D_\Gamma W$ is the matrix whose line i is given by $\nabla\tilde{W}_i - \big(\nabla\tilde{W}_i \boldsymbol{n}\big)\boldsymbol{n} = \nabla_\Gamma W_i$.

Definition 3.2.9. *Assume Ω is of class C^2. We define the mean curvature $\mathcal{H}$ of $\partial\Omega$ by*

$$\mathcal{H} = \mathrm{div}_\Gamma \boldsymbol{n}.$$

Finally we introduce the Laplace-Beltrami operator

Definition 3.2.10. *Let Ω be a bounded open set of class C^2. The Laplace-Beltrami operator on $\partial\Omega$, called Δ_Γ is defined by*

$$\forall u \in W^{2,1}(\partial\Omega), \quad \Delta_\Gamma u = \mathrm{div}_\Gamma\Big(\nabla_\Gamma u\Big).$$

We then have

$$\Delta_\Gamma u = \Delta u - \mathcal{H}\partial_{\boldsymbol{n}} u - \partial^2_{\boldsymbol{nn}} u.$$

3.2.2 The displacement field method

We fix $\Omega_0 \in \mathcal{O}_k$ a reference set with boundary $\partial\Omega_0$. Let $W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$ be the set of Lipschitz bounded functions of $\mathbb{R}^d$, endowed with the norm

$$\forall \theta \in W^{1,\infty}\big(\mathbb{R}^d; \mathbb{R}^d\big), \quad \|\theta\|_{W^{1,\infty}} = \sup_{y,\tilde{y}\in\mathbb{R}^d, y\neq\tilde{y}} \left\{|\theta(y)| + |\theta(y) - \theta(\tilde{y})|/|y - \tilde{y}|\right\}.$$

We can alternatively define a classical notion of shape derivative by working with $C^1(\mathbb{R}^d; \mathbb{R}^d) \cap W^{1,\infty}(\mathbb{R}^d; \mathbb{R}^d)$ instead of $W^{1,\infty}(\mathbb{R}^d; \mathbb{R}^d)$. For $k \geq 1$ we set $C^{k,\infty}(\mathbb{R}^d; \mathbb{R}^d) := C^k(\mathbb{R}^d; \mathbb{R}^d) \cap W^{k,\infty}(\mathbb{R}^d; \mathbb{R}^d)$. For any $\theta \in W^{1,\infty}(\mathbb{R}^d; \mathbb{R}^d)$, let Ω_θ be given by

$$\Omega_\theta = \left\{\big(\mathrm{Id} + \theta\big)(x),\, x \in \Omega_0\right\}.$$

When $\|\theta\|_{W^{1,\infty}(\mathbb{R}^d;\mathbb{R}^d)} < 1$, thanks to the Banach fixed-point theorem, $\mathrm{Id} + \theta$ is invertible, with $(\mathrm{Id} + \theta)^{-1} \in W^{1,\infty}\big(\mathbb{R}^d; \mathbb{R}^d\big)$, and

$$\begin{cases} \Big\|\big(\mathrm{Id} + \theta\big)^{-1} - \mathrm{Id}\Big\|_{W^{1,\infty}(\mathbb{R}^d;\mathbb{R}^d)} & \leq \quad \|\theta\|_{W^{1,\infty}} \Big(1 - \|\theta\|_{W^{1,\infty}(\mathbb{R}^d;\mathbb{R}^d)}\Big)^{-1}, \\ \Big\|\big(\mathrm{Id} + \theta\big)^{-1} - \mathrm{Id} + \theta\Big\|_{L^\infty(\mathbb{R}^d;\mathbb{R}^d)} & \leq \quad \|\theta\|_{W^{1,\infty}} \Big\|\big(\mathrm{Id} + \theta\big)^{-1} - \mathrm{Id}\Big\|_{L^\infty(\mathbb{R}^d;\mathbb{R}^d)}. \end{cases}$$

Then denoting by $\mathcal{B} = \{\theta \in W^{1,\infty}(\mathbb{R}^d; \mathbb{R}^d) \,|\, \|\theta\|_{W^{1,\infty}(\mathbb{R}^d;\mathbb{R}^d)} < 1\}$ we have that $\mathcal{B} \ni \theta \mapsto \big(\mathrm{Id} + \theta\big)^{-1} \in W^{1,\infty}(\mathbb{R}^d; \mathbb{R}^d)$ is continuous at 0 and $\mathcal{B} \ni \theta \to \big(\mathrm{Id} + \theta\big)^{-1} \in L^\infty\big(\mathbb{R}^d\big)$ is differentiable at 0 with derivative $-\mathrm{Id}$.

Definition 3.2.11. *Let $k \geq 1$ and $E(\Omega)$ be a function from $\mathcal{O}_k$ into $\mathbb{R}$. We define*

$$\begin{array}{rccc} \mathcal{E} & : & C^{k,\infty}(\mathbb{R}^d; \mathbb{R}^d) & \to & \mathbb{R} \\ & & \theta & \mapsto & E\Big(\big(\mathrm{Id} + \theta\big)(\Omega)\Big). \end{array}$$

The function E is said to be shape-differentiable at Ω if $\mathcal{E}$ is Fréchet-differentiable at 0, that is, if there exists a continuous linear map $\mathcal{E}'(0;\cdot) : C^{1,\infty}\big(\mathbb{R}^d; \mathbb{R}^d\big) \to \mathbb{R}$ such that :

$$\mathcal{E}(\theta) - \mathcal{E}(0) - \mathcal{E}'(0;\theta) = o\big(\|\theta\|_{C^{k,\infty}(\mathbb{R}^d;\mathbb{R}^d)}\big).$$

We denote $E'(\Omega;\theta) := \mathcal{E}'(0;\theta)$.

Definition 3.2.12. *The function E of Definition 3.2.11 is said to be twice shape-differentiable at Ω if $\mathcal{E}$ is Fréchet-differentiable in a neighborhood $\mathcal{U}$ of 0 in $C^{k,\infty}\big(\mathbb{R}^d;\mathbb{R}^d\big)$ and if the first derivative $\mathcal{E}'$ defined by*

$$\begin{array}{rccc} \mathcal{E}' & : & \mathcal{U} & \to & \Big(C^{k,\infty}\big(\mathbb{R}^d;\mathbb{R}^d\big)\Big)' \\ & & \theta & \mapsto & \mathcal{E}'\big(\theta;\cdot\big), \end{array}$$

is Fréchet-differentiable at 0. We denote by $\mathcal{E}''\big(0;\theta_1,\theta_2\big)$ the second Fréchet derivative at 0, θ_1 and θ_2 being respectively the first and second directions of derivation. We also denote $E''\left(\Omega;\theta_1,\theta_2\right) := \mathcal{E}''\big(0;\theta_1,\theta_2\big)$. In that case, $\mathcal{E}$ has a second-order Taylor expansion at 0 and

$$\mathcal{E}(\theta) = \mathcal{E}(0) + \mathcal{E}'(0;\theta) + \frac{1}{2}\mathcal{E}''(0;\theta,\theta) + o\big(\,\|\theta\|^2_{C^{k,\infty}(\mathbb{R}^d;\mathbb{R}^d)}\,\big).$$

Remark 3.2.13. *When $\mathcal{E}$ is Fréchet-differentiable, it is a fortiori Gâteaux-differentiable. Therefore, for $\theta \in \mathcal{U}$*

$$E'\left(\Omega;\theta\right) := \mathcal{E}'\left(0;\theta\right) = \lim_{t\to 0}\frac{1}{t}\Big(\mathcal{E}(t\theta) - \mathcal{E}(0)\Big),$$

and

$$\mathcal{E}''\left(0;\theta_1,\theta_2\right) = \lim_{t\to 0}\frac{1}{t}\Big(\mathcal{E}'\left(t\theta_2;\theta_1\right) - \mathcal{E}'\left(0;\theta_1\right)\Big),$$

where

$$\mathcal{E}'(\theta_2;\theta_1) := \lim_{s\to 0}\frac{1}{s}\Big(\mathcal{E}(\theta_2 + s\theta_1) - \mathcal{E}(\theta_2)\Big).$$

Remark 3.2.14. *As a classical result of Calculus of Variations (see [63, §II.1.2] for example), since $C^{1,\infty}\big(\mathbb{R}^d;\mathbb{R}^d\big)$ is a linear space, the second-order derivative of $\mathcal{E}$ corresponds to the derivative of its first-order derivative. However, as explained in [119], it is not the case for $E(\Omega)$ because the variation of two successive vector fields θ_1,θ_2 is not the same as the variation obtained with the sum $\theta_1+\theta_2$:*

$$\big(\mathrm{Id}+\theta_1\big)\circ\big(\mathrm{Id}+\theta_2\big) = \mathrm{Id}+\theta_2+\theta_1\circ\big(\mathrm{Id}+\theta_2\big) \neq \mathrm{Id}+\theta_2+\theta_1. \tag{3.13}$$

It follows that there is a particular relation between the second-order derivative of $\mathcal{E}$ at 0 on the one hand, and the second-order derivative of E at Ω on the other hand. Indeed

$$\begin{aligned}\mathcal{E}'(\theta_2;\theta_1) &= \lim_{s\to 0}\frac{1}{s}\Big(\mathcal{E}(\theta_2+s\theta_1) - \mathcal{E}(\theta_2)\Big)\\ &= \lim_{s\to 0}\frac{1}{s}\bigg(E\Big(\big(\mathrm{Id}+\theta_2+s\theta_1\big)(\Omega)\Big) - E\Big(\big(\mathrm{Id}+\theta_2\big)(\Omega)\Big)\bigg).\end{aligned}$$

Observing that

$$\mathrm{Id}+\theta_2+s\theta_1 = \Big(\mathrm{Id}+s\theta_1\circ\big(\mathrm{Id}+\theta_2\big)^{-1}\Big)\circ\big(\mathrm{Id}+\theta_2\big).$$

we obtain

$$\begin{aligned}\mathcal{E}'(\theta_2;\theta_1) &= \lim_{s\to 0}\frac{1}{s}\bigg(E\Big(\big(\mathrm{Id}+s\theta_1\circ\big(\mathrm{Id}+\theta_2\big)^{-1}\big)\circ\big(\mathrm{Id}+\theta_2\big)(\Omega)\Big) - E\Big(\big(\mathrm{Id}+\theta_2\big)(\Omega)\Big)\bigg)\\ &= E'\Big(\big(\mathrm{Id}+\theta_2\big)(\Omega);\theta_1\circ\big(\mathrm{Id}+\theta_2\big)^{-1}\Big).\end{aligned}$$

Since $\mathcal{E}'$ is linear in its second variable and $\theta_1\circ\big(\mathrm{Id}+t\theta_2\big)^{-1} = \theta_1 - t\,\left(\theta_2\cdot\nabla\right)\theta_1 + o(t)$, we have

$$\begin{aligned}\mathcal{E}''\big(0;\theta_1,\theta_2\big) &= \lim_{t\to 0}\frac{1}{t}\Big(\mathcal{E}'(t\theta_2;\theta_1) - \mathcal{E}'(0;\theta_1)\Big)\\ &= \lim_{t\to 0}\frac{1}{t}\bigg(E'\Big(\big(\mathrm{Id}+t\theta_2\big)(\Omega);\theta_1\circ\big(\mathrm{Id}+t\theta_2\big)^{-1}\Big) - E'\Big(\Omega;\theta_1\Big)\bigg)\\ &= \lim_{t\to 0}\frac{1}{t}\bigg(E'\Big(\big(\mathrm{Id}+t\theta_2\big)(\Omega);\theta_1\Big) - E'\big(\Omega;\theta_1\big)\bigg)\\ &\qquad + \lim_{t\to 0}E'\Big(\big(\mathrm{Id}+t\theta_2\big)(\Omega);\frac{1}{t}\big(\theta_1\circ\big(\mathrm{Id}+t\theta_2\big)^{-1}-\theta_1\big)\Big),\\ &= \Big(E'(\Omega;\theta_1)\Big)'\left(\Omega;\theta_2\right) - E'\left(\Omega;\nabla\theta_1\,\theta_2\right)\end{aligned}$$

so that

$$\mathcal{E}''\left(0;\theta_1,\theta_2\right) = \Big(E'(\Omega;\theta_1)\Big)'\left(\Omega;\theta_2\right) - E'\left(\Omega;\left(\theta_2\cdot\nabla\right)\theta_1\right). \tag{3.14}$$

We now recall the Hadamard structure theorem for the first and second-order shape derivatives.

Theorem 3.2.15 ([72, Theorem 5.9.2]). *Let $k \geq 1$, $E, \mathcal{E}$ defined in Definition 3.2.11.*

1. *Take $\Omega \in \mathcal{O}_{k+1}$ and $\mathcal{E}$ differentiable at 0 in $C^{k,\infty}\big(\mathbb{R}^d;\mathbb{R}^d\big)$. Then, there exists a continuous linear map $l_1 : C^k\big(\partial\Omega\big) \to \mathbb{R}$ such that :*
$$\forall \xi \in C^{k,\infty}\big(\mathbb{R}^d;\mathbb{R}^d\big), \quad \mathcal{E}'\left(0;\xi\right) = l_1\left(\xi \cdot \boldsymbol{n}\right).$$
2. *Take $\Omega \in \mathcal{O}_{k+2}$ and $\mathcal{E}$ twice differentiable at 0 in $C^{k,\infty}\big(\mathbb{R}^d;\mathbb{R}^d\big)$. Then there exists a continuous bilinear symmetric map $l_2 : C^k\big(\partial\Omega\big) \times C^k\big(\partial\Omega\big) \to \mathbb{R}$ such that for any $\theta, \xi \in C^{k+1,\infty}\big(\mathbb{R}^d;\mathbb{R}^d\big)$,*

$$\begin{cases} E''\left(\Omega;\theta,\xi\right) = \mathcal{E}''\left(0;\theta,\xi\right) = l_2(\theta\cdot\boldsymbol{n},\xi\cdot\boldsymbol{n}) + l_1(Z_{\theta,\xi}), \\ Z_{\theta,\xi} = \big(\left(\xi_\Gamma\cdot\nabla_\Gamma\right)\boldsymbol{n}\big)\cdot\theta_\Gamma - \nabla_\Gamma\left(\theta\cdot\boldsymbol{n}\right)\cdot\xi_\Gamma - \nabla_\Gamma\left(\xi\cdot\boldsymbol{n}\right)\cdot\theta_\Gamma. \end{cases} \tag{3.15}$$

Definition 3.2.16. *According to [72, Proposition 5.4.14, §5.9.1], denoting by d_Ω the signed distance function to $\partial\Omega$, we define $\tilde{\boldsymbol{n}}$ by : $\tilde{\boldsymbol{n}} = \nabla d_\Omega$. The vector field $\tilde{\boldsymbol{n}}$ is a unitary extension of class C^1 of the outer normal in a neighborhood of $\partial\Omega$.*

As a result $\nabla\tilde{\mathbf{n}}^T\tilde{\mathbf{n}} = 0$, and in particular $\nabla\tilde{\mathbf{n}}^T\mathbf{n} = 0$ on $\partial\Omega$ (on $\partial\Omega$, $\mathbf{n} = \tilde{\mathbf{n}}$). Therefore

$$\nabla_\Gamma\tilde{\mathbf{n}} = \nabla\tilde{\mathbf{n}} - \big(\mathbf{n}\cdot\nabla\tilde{\mathbf{n}}\big)\mathbf{n} = D^2 d_\Omega.$$

Consequently, $\nabla_\Gamma\tilde{\mathbf{n}}$ is a symmetric matrix on $\partial\Omega$. As discussed in Section 3.2.1, the tangential gradient does not depend on the extension, which means that $\nabla_\Gamma\mathbf{n}$ is symmetric on $\partial\Omega$. Therefore the vector field $Z_{\theta,\xi}$ defined by (3.15) and the second-order shape derivative $E''\left(\Omega;\theta,\xi\right)$ are also symmetric in (θ,ξ). Explicit examples of the linear maps l_1 and l_2 are given in Section 3.3 and Section 7.1.

3.2.3 The speed method

Following the approach of Sokolowski and Zolésio [55, 54, 122], shape derivatives may also be defined as the Eulerian derivatives along time trajectories, defined as flows or solutions of ordinary differential equations.

Definition 3.2.17. *Let $V \in C^{1,\infty}\big(\mathbb{R}_+\times\mathbb{R}^d;\mathbb{R}^d\big)$. For $\tau > 0$ small enough and for $x \in \mathbb{R}^d$, we define the flow of the vector field V as the unique solution $X_V : [0,\tau]\times\mathbb{R}^d \to \mathbb{R}^d$ of*

$$\begin{cases} \dfrac{\partial X_V}{\partial t}(t,x) = V\big(t, X_V(t,x)\big), \\ X_V(0,x) = x. \end{cases} \tag{3.16}$$

Define the set

$$X_V(t,\cdot)(\Omega) = \left\{X_V(t,x),\, x\in\Omega\right\}.$$

For V, W in $C^{1,\infty}\big(\mathbb{R}_+\times\mathbb{R}^d;\mathbb{R}^d\big)$ the first and second-order directional derivatives of E are defined by

$$dE\big(\Omega;V\big) := \partial_t E\big(X_V(t,\cdot)(\Omega)\big)\Big|_{t=0},$$
$$d^2E\big(\Omega;V,W\big) := \partial_t\partial_s E\big(X_V(t,\cdot)\circ X_W(s,\cdot)(\Omega)\big)\Big|_{s=t=0}.$$

Since Ω is a bounded domain, the existence time τ for (3.16) can be chosen to be uniform for any point x in a compact neighborhood of Ω.

Remark 3.2.18. *In the above definition, the velocity field V plays the same role as the vector field θ in Section 3.2.2. One can check [94, Remark 2.5] that*

$$dE\big(\Omega;V\big) = E'\left(\Omega;V\right) \quad \text{and} \quad d^2E\big(\Omega;V,W\big) = \Big(E'(\Omega;V)\Big)'\left(\Omega;W\right).$$

Contrary to Definition 3.2.11 where the shape derivative is a Fréchet derivative, the shape derivative of Definition 3.2.17 is a directional or Gâteaux derivative.

Theorem 3.2.19 ([72, Corollaire 5.9.3]). *Let $X_V(\cdot,x)$ be the solution of (3.16) where $V \in C^{1,\infty}\big(\mathbb{R}_+\times\mathbb{R}^d;\mathbb{R}^d\big)$. Define the function $e(t)$ from $[0,\tau]$ into $\mathbb{R}$ by*

$$e(t) = E\big(X_V(t,\cdot)(\Omega)\big) = \mathcal{E}\big(X_V(t,\cdot) - \mathrm{Id}\big). \tag{3.17}$$

Then e is twice differentiable on $[0,\tau]$ and

$$e'(0) = l_1\left(V\cdot\boldsymbol{n}\right),$$
$$e''(0) = l_2\big(V\cdot\boldsymbol{n}, V\cdot\boldsymbol{n}\big) + l_1(\tilde{Z}_{V,V}),$$

with the same linear map l_1 and bilinear map l_2, defined on $\partial\Omega$, as in Theorem 3.2.15, and with

$$\tilde{Z}_V = \big(\partial_t V + \left(V\cdot\nabla\right)V\big)\cdot\boldsymbol{n} + \big(\left(V_\Gamma\cdot\nabla_\Gamma\right)\boldsymbol{n}\big)\cdot V_\Gamma - 2\nabla_\Gamma\left(V\cdot\boldsymbol{n}\right)\cdot V_\Gamma.$$

Proof. Using the chain rule we get $e'(t) = \mathcal{E}'\Big(X_V(t,\cdot) - \mathrm{Id}; V\big(t, X_V(t,\cdot)\big)\Big)$ and $e''(0) = \mathcal{E}''(0;V,V) + \mathcal{E}'\left(0; \dfrac{dV}{dt}\right)$. The expected result then follows from Theorem 3.2.15. □

Although Theorem 3.2.19 is just a corollary of Theorem 3.2.15, the two notions of shape derivative differ. The first-order derivatives in Theorems 3.2.15 and 3.2.19 coincide but the second-order derivatives are different. Note however that the directional derivatives of Theorem 3.2.15 can be recovered from Theorem 3.2.19 by the special choice in (3.16) of the vector field

$$V(t,x) = \theta \circ \big(\mathrm{Id} + t\theta\big)^{-1}(x),$$

which corresponds to the solution $X(t,x) = \big(\mathrm{Id} + t\theta\big)(x)$.

Remark 3.2.20 ([72, Remarque §5.9.4])**.** *An alternative second-order shape derivative can be defined as*

$$D^2E\big(\Omega; V, W\big) = \partial_s\partial_t E\Big(X_V(s+t,\cdot)\circ X_V(s,\cdot)^{-1}\circ X_W(s,\cdot)(\Omega_0)\Big)\Big|_{t=s=0}.$$

We notice that, for s fixed, $X(t,\cdot) := X_V(s+t,\cdot)\circ X_V(s,\cdot)^{-1}$ satisfies

$$\frac{d}{dt}X(t,\cdot) = V\big(s+t, X(t,\cdot)\big), \quad X(0,\cdot) = \mathrm{Id}.$$

The derivation of $(t,s)\mapsto \mathcal{E}\big(X_V(s+t,\cdot)\circ X_V(s,\cdot)^{-1}\circ X_W(s,\cdot) - \mathrm{Id}\big)$ with respect to s and t leads to

$$\begin{aligned} D^2E\big(\Omega; V, W\big) &= \partial_s\Big(\mathcal{E}'\big(X_W(s,\cdot) - \mathrm{Id}\big); X_V\big(s, X_W(s,\cdot)\big)\Big) \\ &= \mathcal{E}''\left(\Omega; V, W\right) + \mathcal{E}'\left(\Omega; \partial_s V + W\cdot\nabla V\right). \end{aligned}$$

Now we can cite a corollary that states a connection between the two frameworks for shape differentiation. With a particular choice of a velocity field V for the speed method, we can have the exact same shape derivatives as with the displacement field method.

Corollary 3.2.21 ([94, Corollary 2.4])**.** *With the same assumptions and notation as in Corollary 3.2.19, we consider $X(t,x) = \big(\mathrm{Id} + t\xi\big)(x)$ (it would correspond to taking $\theta(t,x) = t\xi(x)$ in Theorem 3.2.15) for $\xi \in C^{k,\infty}\big(\mathbb{R}^d;\mathbb{R}^d\big)$. Then e is twice differentiable on $[0,\tau]$ and we get*

$$\begin{aligned} e'(0) &= l_1\left(\xi\cdot\boldsymbol{n}\right), \\ e''(0) &= l_2\big(\xi\cdot\boldsymbol{n}, \xi\cdot\boldsymbol{n}\big) + l_1\Big(\big(\left(\xi_\Gamma\cdot\nabla_\Gamma\right)\boldsymbol{n}\big)\cdot\xi_\Gamma - 2\nabla_\Gamma\left(\xi\cdot\boldsymbol{n}\right)\cdot\xi_\Gamma\Big). \end{aligned}$$

Proof. We just have to notice that if $X(t,x) = \big(\mathrm{Id} + t\xi\big)(x)$, then X is the flow of the vector field

$$V(t,x) = \xi\circ\big(\mathrm{Id} + t\xi\big)^{-1}(x)$$

such that $\partial_t V(0) = -\nabla\xi\,\xi$. Therefore with Corollary 3.2.19, we get

$$\big(\partial_t V + V\cdot\nabla V\big)\Big|_{t=0}\cdot\mathbf{n} = 0.$$

□

Lemma 3.2.22 ([72, Proposition 5.4.14])**.** *Let Ω be a bounded open set of class C^2, and $\tilde{\boldsymbol{n}}$ the unitary extension of the outer unit normal $\boldsymbol{n}$ introduced in Definition 3.2.16. Let also $V \in C^k\big(\mathbb{R}_+\times\mathbb{R}^d;\mathbb{R}^d\big)$ and X_V be the maximal solution to (3.16). Then, for $t\in[0,\eta[$ with $\eta > 0$ small enough,*

$$t\mapsto \boldsymbol{n}_t(x) = w(t,x)/\|w(t,x)\| \text{ with } w(t,\cdot) = \big(\nabla^T X_V(t,\cdot)^{-1}\tilde{\boldsymbol{n}}(\cdot)\big)\circ X_V(t,\cdot)^{-1},$$

is a unitary extension of the outer unit normal to $\partial\Omega_t$, and $t\mapsto \boldsymbol{n}_t \in C^0\big(\mathbb{R}^d;\mathbb{R}^d\big)$ is differentiable at 0 with

$$\partial_t\boldsymbol{n}_t(x) = -\nabla_\Gamma V(0,x)\cdot\tilde{\boldsymbol{n}}(x). \tag{3.18}$$

Remark 3.2.23. *We can now compare the structure of the shape derivatives for the two frameworks presented here. The first-order shape derivatives are the same and is given by a linear map we have until now denoted by l_1.*

The second-order shape derivatives somewhat differ from one method to the other, but the structure remains similar. It is always given by a combination of two terms. The first one is a bilinear map - denoted by l_2 - on the normal components of the velocity field on the boundary. The second-one is given by the first-order linear map l_1 on a vector - Z - that contains tangential components of the velocity field. This is only by the expression of this vector that the second-order shape derivatives differ. Moreover, this vector Z does not depend on the function to be differentiated.

Therefore, it suffices to know the linear and bilinear maps l_1 and l_2 to know the shape derivatives whatever the framework used to compute them.

3.3 Geometric objective functions

The geometric formalism which we consider here consists in considering shapes as bounded open sets of $\mathbb{R}^d$ and focusing on the displacement field method of Murat and Simon already introduced above. In this framework we detail some first-order derivative results for usual objective functions. They include a few purely geometric functions such as the volume or a maximum-thickness criterion. We also consider two usual mechanical criterion that are the compliance and a least square criterion on displacements. This section is basically a catalog of how to compute the different shape derivatives of the considered criterion. All rigorous justifications can be found in [7, 15], [72], or [52, 83]. As regards the mechanical criterion, we place ourselves in the framework of the *scalar thermal model* of Section 2.1.4. We assume that the shape Ω is smooth with its boundary divided in three disjoint parts Γ_D, Γ_N and Γ such that

$$\partial\Omega = \Gamma_D \cup \Gamma_N \cup \Gamma.$$

For $f \in L^2(\Omega;\mathbb{R})$ and $g \in H^1(\Omega;\mathbb{R})$ such that $g = 0$ on Γ let $u : \Omega \to \mathbb{R}$ be the solution to

$$\left\{ \begin{array}{rcll} -\mathrm{div}\,(\nabla u) & = & f & \text{in } \Omega, \\ \partial_{\mathbf{n}} u & = & g & \text{on } \Gamma_N \cup \Gamma, \\ u & = & 0 & \text{on } \Gamma_D. \end{array} \right. \tag{3.19}$$

In the most general setting, for an objective function J, the optimization problem reads

$$\min_{\Omega \in \mathcal{U}_{ad}} J(\Omega), \tag{3.20}$$

with

$$\mathcal{U}_{ad} = \left\{ \Omega \subset \mathbb{R}^d \mid \Omega \in \mathcal{O}_k \right\},$$

(where $k \in \mathbb{N}^\star$) or with an additional constraint $G(\Omega) \leq 0$ in the definition of $\mathcal{U}_{ad}$ such as in Remark 3.1.1. In practice it is also often assumed that the parts Γ_D and Γ_N of the boundary are not supposed to be modified. In this section we will not make this assumption on the boundary, and give the shape derivatives for different functions. We also recall that for a given direction θ sufficiently smooth, the first-order shape derivative of J at Ω in the direction θ is denoted $J'(\Omega;\theta)$ or J'_θ.

3.3.1 Integrals on a varying domain

We recall here from [72, Theorem 5.2.2, 5.4.17] how to derive integral expressions with respect to the shape.

Proposition 3.3.1. *Let Ω be a measurable bounded open set of $\mathbb{R}^d$. Let $f \in W^{1,1}(\mathbb{R}^d,\mathbb{R}^d)$. The functional $F(\Omega) = \displaystyle\int_\Omega f(x)\,dx$ is shape-differentiable with*

$$\forall \theta \in C^{1,\infty}(\mathbb{R}^d,\mathbb{R}^d), \quad F'(\Omega;\theta) = \int_\Omega \mathrm{div}(f\theta).$$

If in addition, Ω has a Lipschitz boundary, one can write

$$F'(\Omega;\theta) = \int_{\partial\Omega} (\theta \cdot \boldsymbol{n}) f.$$

Let us briefly give a sketch of the proof

Proof. We denote $\Omega_\theta := (\mathrm{Id} + \theta)(\Omega)$. Then we get

$$F(\Omega_\theta) = \int_{\Omega_\theta} f(x)\,dx.$$

Now we want to bring back this integral, by a change of variable, to the reference domain Ω. We introduce $\overline{f} = f \circ (\mathrm{Id} + \theta)$. Thus,

$$F(\Omega_\theta) = \int_\Omega \overline{f}(x) |\det(\mathrm{Id} + \nabla\theta(x))|\,dx.$$

Now we write the first-order Taylor expansions

$$\begin{aligned} \det(\mathrm{Id} + \nabla\theta) &= 1 + \mathrm{div}(\theta) + o\big(\|\theta\|_{C^{1,\infty}(\mathbb{R}^d;\mathbb{R}^d)}\big), \\ \overline{f} &= f + \nabla f\theta + o\big(\|\theta\|_{C^{1,\infty}(\mathbb{R}^d;\mathbb{R}^d)}\big). \end{aligned}$$

This leads to the expansion

$$F(\Omega_\theta) = \int_\Omega f + \int_\Omega \nabla f\theta + \mathrm{div}(\theta) + o\big(\|\theta\|_{C^{k,\infty}(\mathbb{R}^d;\mathbb{R}^d)}\big),$$

which is what was expected. □

We now consider functionals defined by a boundary integral

Proposition 3.3.2. *Assume that Ω is a bounded Lipschitz domain of class C^2 (that is $\Omega \in \mathcal{O}_2$, where $\mathcal{O}_k$ for $k \in \mathbb{N}$ is defined by (3.11)) Let also $g \in W^{2,1}(\mathbb{R}^d, \mathbb{R}^d)$. Then the functional $G(\Omega) = \displaystyle\int_{\partial\Omega} g(x)\, dx$ is shape-differentiable with*

$$\forall \theta \in C^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d), \quad G'(\Omega;\theta) = \int_{\partial\Omega} (\theta \cdot \boldsymbol{n})\Big(\partial_{\boldsymbol{n}} g + \mathcal{H} g\Big).$$

We can notice that more regularity on Ω and on the direction of derivation θ is needed in the case of boundary integral expressions. This is due to the change of variable in the integrals used to get the differentiability. A particular case is the one of the volume function which is defined by

$$V(\Omega) = \int_\Omega 1.$$

The previous results do apply to this criterion, and we can write the following lemma.

Lemma 3.3.3. *Assume that $\Omega \in \mathcal{O}_1$ with a Lipschitz boundary. The volume is shape-differentiable and*

$$\forall \theta \in W^{1,\infty}\left(\mathbb{R}^d; \mathbb{R}^d\right),\ V'(\Omega;\theta) = \int_{\partial\Omega} (\theta \cdot \boldsymbol{n}).$$

We will also consider later the case of a shape-dependent function E of the form

$$E(\Omega) = \int_\Omega f(x)\, dx, \tag{3.21}$$

where $\Omega \in \mathcal{O}_2$ and $f \in C^2(\mathbb{R}^d; \mathbb{R})$. In that case a minimization problem reads

$$\min_{\Omega \in \mathcal{O}_2} E(\Omega). \tag{3.22}$$

We have first a shape differentiability result.

Lemma 3.3.4. *Assume that $\Omega \in \mathcal{O}_1$ and that $f \in C^1(\mathbb{R}^d; \mathbb{R})$. The function E defined by (3.21) is shape-differentiable and*

$$\forall \theta \in W^{1,\infty}\left(\mathbb{R}^d; \mathbb{R}^d\right),\ E'(\Omega;\theta) = \int_{\partial\Omega} (\theta \cdot \boldsymbol{n})\, f.$$

We can also notice that the minimization problem is straightforward since the optimal shape are given by the set of negative values of f.

Proposition 3.3.5. *We assume that $f \in L^1_{loc}(\mathbb{R}^2) \cap C^0(\mathbb{R}^2)$,*

$$f(x) > 0 \text{ when } |x| \to \infty \quad \text{and} \quad |\{x \in \mathbb{R}^d \mid f(x) < 0\}| \neq 0.$$

Then $\Omega^* = \{x \in \mathbb{R}^d \mid f(x) < 0\}$ is a bounded open set solution of (3.22). The shape Ω^* is the minimal solution in the sense that for any other solution Ω we have

$$\Omega^* \subset \Omega \quad \text{and} \quad f_{\big|\Omega\setminus\Omega^*} = 0.$$

Proof. The boundedness of Ω^* is clear with the assumption made. Now let Ω_1 be a bounded open set of $\mathbb{R}^d$ such that

$$\int_{\Omega_1} f(x)\, dx \leq \int_{\Omega^*} f(x)\, dx.$$

Since

$$\Omega_1 = (\Omega_1 \cup \Omega^*) \cap (\Omega_1 \setminus \Omega^*),$$
$$\Omega^* = (\Omega_1 \cup \Omega^*) \cap (\Omega^* \setminus \Omega_1),$$

we have

$$\int_{\Omega_1\setminus\Omega^*} f(x)\, dx \leq \int_{\Omega^*\setminus\Omega_1} f(x)\, dx.$$

According to the definition of Ω^* we have $f \geq 0$ on $\Omega_1\setminus\Omega^*$ and $f < 0$ on $\Omega^*\setminus\Omega_1$. This yields

$$0 \leq \int_{\Omega_1\setminus\Omega^*} f(x)\, dx \leq \int_{\Omega^*\setminus\Omega_1} f(x)\, dx \leq 0,$$

and then

$$\int_{\Omega_1 \setminus \Omega^*} f(x)\,dx = \int_{\Omega^* \setminus \Omega_1} f(x)\,dx = 0.$$

This proves that

$$\int_{\Omega_1} f(x)\,dx = \int_{\Omega^*} f(x)\,dx.$$

Therefore Ω^* is a minimum. In addition since $f < 0$ on Ω^*, if $\Omega^* \setminus \Omega_1$ is non-empty, then

$$\int_{\Omega^* \setminus \Omega_1} f(x) < 0,$$

which is not the case. Hence $\Omega^* \setminus \Omega_1 = \emptyset$ and $\Omega^* \subset \Omega_1$. On the other hand, we have $f_{\big|\Omega \setminus \Omega^*} \geq 0$ and

$$\int_{\Omega_1 \setminus \Omega^*} f(x)\,dx = 0,$$

which means that

$$f_{\big|\Omega \setminus \Omega^*} = 0.$$

□

3.3.2 Shape derivative of the signed distance function

The maximum-thickness criterion which is to be introduced later in Section 7.1.5 requires some background about the signed distance function and its shape derivative. To that end we recall some results of [52, 83].

Definition 3.3.6.

- *Let $A \subset \mathbb{R}^d$ be a non-empty closed set. The Euclidean distance function to A, denoted by $d(., A)$ is defined as*

$$\forall x \in \mathbb{R}^d,\ d(x, A) = \min_{a \in A} |x - a|. \tag{3.23}$$

- *Let $\Omega \subset \mathbb{R}^d$ be a Lipschitz domain. The signed distance function d_Ω to Ω is defined as*

$$\forall x \in \mathbb{R}^d,\ d_\Omega(x) = \begin{cases} -d(x, \partial\Omega) & \text{if} \quad x \in \Omega, \\ 0 & \text{if} \quad x \in \partial\Omega, \\ d(x, \partial\Omega) & \text{if} \quad x \in \mathbb{R}^d \setminus (\Omega \cup \partial\Omega). \end{cases} \tag{3.24}$$

Definition 3.3.7. *Let $\Omega \subset \mathbb{R}^d$ be a Lipschitz bounded open set.*

- *For any $x \in \mathbb{R}^d$, $\prod_{\partial\Omega}(x) = \{y_0 \in \partial\Omega \mid |x - y_0| = \inf_{y \in \Omega} |x - y|\}$ is the set of orthogonal projections of x on $\partial\Omega$. It is a closed subset of $\partial\Omega$. When $\prod_{\partial\Omega}(x)$ reduces to a single point, this point is denoted as $p_{\partial\Omega}(x)$, and is called the orthogonal projection of x onto $\partial\Omega$.*
- *$\Sigma = \big\{x \in \mathbb{R}^d$ such that $(d_\Omega)^2$ is not differentiable at $x\big\}$ is called the skeleton of $\partial\Omega$ (or sometimes of Ω by a small abuse of terminology).*
- *For any $x \in \partial\Omega$, $\mathrm{ray}_{\partial\Omega}(x) = \big\{y \in \mathbb{R}^d$ such that d_Ω is differentiable at y and $p_{\partial\Omega}(y) = x\big\}$ is the ray emerging from x. Equivalently, $\mathrm{ray}_{\partial\Omega}(x) = p_{\partial\Omega}^{-1}(x)$.*

Proposition 3.3.8.

1. *Let $x \in \mathbb{R}^d \setminus \partial\Omega$ and $y \in \prod_{\partial\Omega}(x)$. If $\partial\Omega$ enjoys C^1 regularity in a neighborhood of y, then*

$$\frac{x - y}{d_\Omega(x)} = \boldsymbol{n}(y),$$

where $\boldsymbol{n}(y)$ is the unit normal vector to Ω at y, pointing outward.

2. *A point $x \notin \partial\Omega$ has a unique projection $p_{\partial\Omega}(x)$ onto $\partial\Omega$ if and only if $x \notin \Sigma$. In such a case, it satisfies $d(x, \partial\Omega) = |p_{\partial\Omega}(x) - x|$ and the gradient of d_Ω at x reads*

$$\nabla d_\Omega(x) = \boldsymbol{n}(p_{\partial\Omega}(x)) = \frac{x - p_{\partial\Omega}(x)}{d_\Omega(x)}.$$

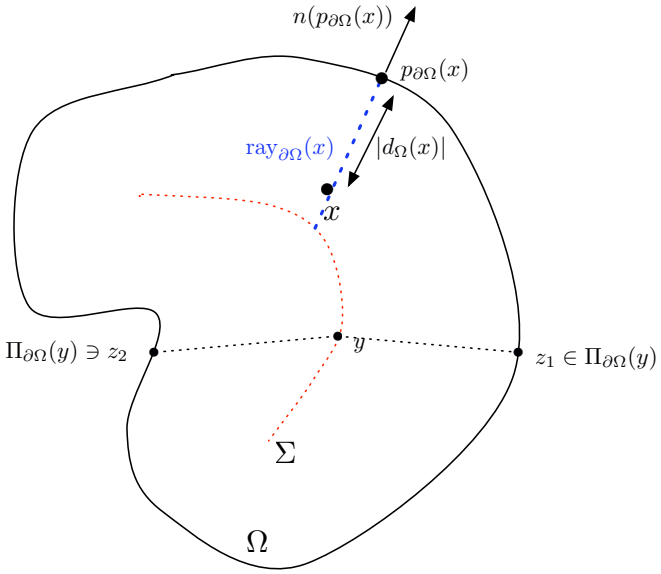


Figure 3.1: Skeleton.

3. *If Ω is of class C^k, for $k \geq 2$, then d_Ω is C^k too in a tubular neighborhood of $\partial\Omega$. In that case, d_Ω is differentiable at every point $x \in \partial\Omega$, and at such a point*

$$\nabla d_\Omega(x) = \boldsymbol{n}(x).$$

For any $x \in \partial\Omega$ we also introduce $\kappa_i(x)$ with $1 \leq i \leq d-1$ the principal curvatures of $\partial\Omega$ at x and $e_i(x)$ the associated principal directions.

Proposition 3.3.9. *For every $x \in \mathbb{R}^d$ and every $y \in \prod_{\partial\Omega}(x)$, one has*

$$\forall i \in \{1, \cdots, d-1\},\ -\kappa_i(y) d_\Omega(x) \leq 1.$$

Furthermore, denoting by Σ_{eq} the set of points $x \notin \Sigma$ such that the above equality holds for some i, we have :

$$\overline{\Sigma} = \Sigma \cup \Sigma_{eq}.$$

If $x \notin \overline{\Sigma}$, then d_Ω is twice differentiable at x and

$$-\kappa_i\big(p_{\partial\Omega}(x)\big) d_\Omega(x) < 1; \quad D^2 d_\Omega(x) = \sum_{i=1}^{d-1} \frac{\kappa_i\big(p_{\partial\Omega}(x)\big)}{1 + \kappa_i\big(p_{\partial\Omega}(x)\big) d_\Omega(x)} e_i\big(p_{\partial\Omega}(x)\big) \otimes e_i\big(p_{\partial\Omega}(x)\big).$$

Remark 3.3.10. *The orthogonal projection map $p_{\partial\Omega}$ is related to d_Ω and ∇d_Ω by*

$$\forall x \in \mathbb{R}^d \backslash \Sigma,\ p_{\partial\Omega}(x) = x - d_\Omega(x) \nabla d_\Omega(x).$$

Proposition 3.3.11. *Take $\Omega \in \mathcal{O}_1$, and fix a point $x \notin \Sigma$. Then $\theta \mapsto d_{(\mathrm{Id}+\theta)\Omega}(x)$ is Gâteaux-differentiable at $\theta = 0$, as an application from $W^{1,\infty}(\mathcal{D}, \mathbb{R}^d)$ into $\mathbb{R}$, and its derivative is*

$$d_\Omega{}'(\Omega; \theta)(x) = -\theta(p_{\partial\Omega}(x)) \cdot \boldsymbol{n}(p_{\partial\Omega}(x)).$$

Remark 3.3.12. *The signed distance function can also be seen as a solution of the following eikonal equation*

$$\begin{cases} |\nabla d_\Omega(x)| &= 1 \quad \text{in } \mathcal{D}, \\ d_\Omega(x) &= 0 \quad \text{on } \Omega. \end{cases}$$

The behavior of the variations of d_Ω with respect to the domain can be retrieved by a formal computation. Indeed, assuming that d_Ω is shape-differentiable, a formal computation yields that the directional shape derivative $d_\Omega{}'(\Omega; \theta)$ satisfies

$$\begin{cases} \nabla d_\Omega(x) \cdot \nabla d_\Omega{}'(\Omega; \theta)(x) &= 0 & \text{in } \mathcal{D}, \\ d_\Omega{}'(\Omega; \theta)(x) &= -\theta(x) \cdot \boldsymbol{n}(x) & \text{on } \partial\Omega. \end{cases}$$

In the following we will sometimes write d'_θ instead of $d_\Omega{}'(\Omega; \theta)$.

Now we are able to differentiate with respect to the shape a functional defined by an integral depending on the signed distance function.

Lemma 3.3.13. *Take $\Omega \in \mathcal{O}_1$ and $m(x,s) : \mathbb{R}^d_x \times \mathbb{R}_s \mapsto \mathbb{R}$ a function of class C^1. Define the functional $J(\Omega)$ as*

$$J(\Omega) = \int_{\mathcal{D}} m(x, d_\Omega(x))\, dx.$$

The mapping $\theta \mapsto J((\mathrm{Id}+\theta)\Omega)$ from $W^{1,\infty}(\mathcal{D},\mathbb{R}^d)$ into $\mathbb{R}$ is Gâteaux-differentiable at $\theta = 0$ and its derivative reads

$$J'(\Omega;\theta) = -\int_{\mathcal{D}} \frac{\partial m}{\partial s}(x, d_\Omega(x))\, \theta(p_{\partial\Omega}(x)) \cdot \boldsymbol{n}(p_{\partial\Omega}(x))\, dx.$$

Lemma 3.3.14. *Take $\Omega \in \mathcal{O}_2$. For $i = 1, \cdots, d-1$, denote by κ_i the principal curvatures of $\partial\Omega$ and e_i the associated principal directions. Let $x \in \mathcal{D}\backslash\overline{\Sigma}$. The orthogonal projection map $p_{\partial\Omega}$ is differentiable at x and, in the orthonormal basis $\{e_1, \cdots, e_{d-1}, \boldsymbol{n}\}(p_{\partial\Omega}(x))$ of $\mathbb{R}^d$, its gradient is the $d \times d$ diagonal matrix*

$$\nabla p_{\partial\Omega}(x) = \begin{pmatrix} 1 - \dfrac{d_\Omega(x)\kappa_1}{1 + d_\Omega(x)\kappa_1} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & 1 - \dfrac{d_\Omega(x)\kappa_{d-1}}{1 + d_\Omega(x)\kappa_{d-1}} & 0 \\ 0 & \cdots & 0 & 0 \end{pmatrix},$$

where the principal curvatures κ_i are evaluated at $p_{\partial\Omega}(x)$.

Corollary 3.3.15. *Let $\Omega \in \mathcal{O}_2$, and let φ be an integrable function over $\mathcal{D}$. Then,*

$$\int_{\mathcal{D}} \varphi(x)\, dx = \int_{\partial\Omega} \left(\int_{\mathrm{ray}_{\partial\Omega}(y)\cap\mathcal{D}} \varphi(z) \prod_{i=1}^{d-1} (1 + d_\Omega(z)\kappa_i(y))\, dz \right) dy,$$

where z denotes a point in the ray emerging from $y \in \partial\Omega$ and dz is the line integration along that ray.

Proposition 3.3.16. *Let $\Omega \in \mathcal{O}_2$, and consider a smooth variation of Ω, $\theta \in C^{2,\infty}(\mathbb{R}^d,\mathbb{R}^d) \cap W^{2,\infty}(\mathbb{R}^d,\mathbb{R}^d)$. Let $x \in \mathbb{R}^d\backslash\overline{\Sigma}$, where Σ denotes the skeleton of Ω. Then there exists $\alpha > 0$ such that the function $t \mapsto p_{\partial\Omega_{t\theta}}(x)$ is well defined and C^1-differentiable over $(0,\alpha)$. Besides,*

$$\frac{d}{dt}\big(p_{\partial\Omega_{t\theta}}(x)\big)\Big|_{t=0} = \Big(\theta(z)\cdot\boldsymbol{n}(z)\Big)\boldsymbol{n}(z) + d_\Omega(x)\Big(\mathrm{Id} + d_\Omega(x)D^2 d_\Omega(z)\Big)^{-1}\nabla_\Gamma\left(\theta\cdot\boldsymbol{n}\right)(z),$$

where $z = p_{\partial\Omega}(x)$.

Remark 3.3.17. *For the sake of simplicity, in the sequel we write $p'_{\partial\Omega,\theta}$ for the derivative of $p_{\partial\Omega}$ in the direction θ when there is no ambiguity.*

3.3.3 Shape derivative of boundary conditions

In this section, the state u is a priori not assumed to be solution to (3.19). We only assume that $u \in H^2(\Omega;\mathbb{R})$ is twice shape-differentiable. We recall some useful formal calculus from [72] to get boundary conditions on a directional derivative. The following results on boundary conditions may be enlarged to more general boundary-value problems (see [72, §5.6])

Dirichlet boundary condition

Let's consider the boundary condition

$$u = 0 \quad \text{on } \Gamma_D.$$

The shape derivative of u satisfy different boundary conditions on Γ_D.

Proposition 3.3.18. *Let u'_θ be the shape derivative of the state u in the direction θ. It satisfies the following boundary condition*

$$u'_\theta + (\theta\cdot\nabla)\, u = 0 \quad \text{on } \Gamma_D.$$

Remark 3.3.19. *Since $u = 0$ on Γ_D, $\nabla_\Gamma u = 0$ on Γ_D. Then the derivation of the boundary condition writes*

$$u'_\theta + (\theta\cdot\boldsymbol{n})\,\partial_{\boldsymbol{n}} u = 0 \quad \text{on } \Gamma_D.$$

This consistent with the well-known result stating that the shape derivative depends only on the normal component of the velocity field θ.

Proposition 3.3.20. *Let $u''_{\theta,\xi}$ be the second-order shape derivative of the state u in the direction θ, ξ. It satisfies the following boundary condition*

$$u''_{\theta,\xi} + (\theta\cdot\nabla)\,u'_\xi + (\xi\cdot\nabla)\,u'_\theta + \theta^T D^2u\,\xi = 0 \quad \text{on } \Gamma_D.$$

Proof. We have from [119, Theorem 3.3] or (3.14)

$$u''_{\theta,\xi} = (u'_\theta)'_\xi - u'_{(\xi\cdot\nabla)\theta}. \tag{3.25}$$

Proposition 3.3.18 applied to $u'_\theta + (\theta\cdot\nabla)\,u$ gives

$$\big(u'_\theta + (\theta\cdot\nabla)\,u\big)'_\xi + (\xi\cdot\nabla)\,u'_\theta + (\xi\cdot\nabla)\,(\theta\cdot\nabla)\,u = 0.$$

Then

$$u''_{\theta,\xi} + (\theta\cdot\nabla)\,u'_\xi + (\xi\cdot\nabla)\,u'_\theta + (\xi\cdot\nabla)\,(\theta\cdot\nabla)\,u = 0.$$

With the Dirichlet boundary condition satisfied by $u'_{\xi\cdot\nabla\theta}$ we get

$$u'_{(\xi\cdot\nabla)\theta} = -\big((\xi\cdot\nabla)\,\theta\cdot\nabla\big)u,$$

and

$$\begin{aligned}(\xi\cdot\nabla)\,(\theta\cdot\nabla)\,u &= \big((\xi\cdot\nabla)\,\theta\cdot\nabla\big)u + \big(\theta\cdot(\xi\cdot\nabla)\,\nabla\big)u\\ &= \big((\xi\cdot\nabla)\,\theta\cdot\nabla\big)u + \theta^T D^2u\,\xi.\end{aligned}$$

We deduce from (3.25) that

$$\begin{aligned}(u'_\theta)'_\xi &= u''_{\theta,\xi} + u'_{(\xi\cdot\nabla)\theta}\\ &= \big((\xi\cdot\nabla)\,\theta\cdot\nabla\big)u + \theta^T D^2u\,\xi + u'_{(\xi\cdot\nabla)\theta}.\end{aligned}$$

Therefore,

$$u''_{\theta,\xi} + \nabla u'_\xi\cdot\theta + \nabla u'_\theta\cdot\xi + \theta^T D^2u\,\xi = 0.$$

□

Neumann boundary condition

Let consider a boundary condition such as $\partial_{\mathbf{n}}u = g$ on Γ_N. According to [72, §5.6] we have :

Proposition 3.3.21. *If u is shape-differentiable, and satisfies $\partial_{\boldsymbol{n}}u = g$ on Γ_N, its shape derivative in the direction θ satisfies*

$$\partial_{\boldsymbol{n}}u'_\theta = (\theta\cdot\boldsymbol{n})\left(\partial_{\boldsymbol{n}}g - \partial^2_{\boldsymbol{nn}}u\right) + \nabla u\cdot\nabla_\Gamma\,(\theta\cdot\boldsymbol{n})\,.$$

This is the same result as the one stated in [7, Corollaire 6.36]. We also mention that this differs from the result in [122, Equation (3.12)], where the term $\partial_{\mathbf{n}}g$ is forgotten.

Shape derivative of the outer normal

Let Ω be in $\mathcal{O}_2$. The outer normal to the shape is shape-differentiable. In the previous section, in Lemma 3.2.22, its expression was given in the case of the speed method for shape differentiation. Since the first-order expressions for shape derivative do not change for the displacement field method we have

Lemma 3.3.22. *Let $\Omega\in\mathcal{O}_2$. The shape derivative in the direction $\theta\in C^{1,\infty}(\mathbb{R}^d;\mathbb{R}^d)$ of the outer normal is given by*

$$\boldsymbol{n}'\,(\Omega;\theta) = -\nabla_\Gamma\big(\theta\cdot\boldsymbol{n}\big).$$

Shape derivative of state equations

When the state u is solution to (3.19), with sufficient regularity conditions on the source term, it is possible to prove that it is shape-differentiable. In that case, the shape derivative is solution to a boundary-value problem.

Proposition 3.3.23. *Let $f\in H^1(\Omega;\mathbb{R})$, $g\in H^2(\Omega;\mathbb{R})$ and u be solution to (3.19). Then u is shape-differentiable on $C^{1,\infty}(\Omega)$. Its first-order shape derivative in the direction $\theta\in C^{1,\infty}(\Omega)$ solves*

$$\left\{\begin{array}{rcll} -\Delta u'_\theta &=& 0 & \text{in } \Omega,\\ u'_\theta &=& -\,(\theta\cdot\nabla)\,u & \text{on } \Gamma_D,\\ \partial_{\boldsymbol{n}}u'_\theta &=& (\theta\cdot\boldsymbol{n})\left(\partial_{\boldsymbol{n}}g - \partial^2_{\boldsymbol{nn}}u\right) + \nabla u\cdot\nabla_\Gamma\,(\theta\cdot\boldsymbol{n}) & \text{on } \Gamma_N,\\ \partial_{\boldsymbol{n}}u'_\theta &=& -\,(\theta\cdot\boldsymbol{n})\,\partial^2_{\boldsymbol{nn}}u + \nabla u\cdot\nabla_\Gamma\,(\theta\cdot\boldsymbol{n}) & \text{on } \Gamma. \end{array}\right. \tag{3.26}$$

The details of the proof are not given here, but we refer to [72, Théorème 5.5.1] for a complete proof in the case of a Neumann problem. The main argument is to work with the variational formulation, to make a change of variable in the integrals so that the integration domains are fixed (as in the proof of Proposition 3.3.1), and then to call an implicit function theorem to get the differentiability with respect to θ. We can also reformulate the Neumann boundary condition on the derivative of the state equation.

Proposition 3.3.24. *The Neumann boundary condition for u'_θ also writes*

$$\partial_{\boldsymbol{n}} u'_\theta = \mathrm{div}_\Gamma\big((\theta\cdot\boldsymbol{n})\,\nabla_\Gamma u\big) + (\theta\cdot\boldsymbol{n})\Big(\partial_{\boldsymbol{n}} g - \Delta u + \mathcal{H}\partial_{\boldsymbol{n}} u\Big).$$

Proof. It is just an expansion of $\mathrm{div}_\Gamma\big((\theta\cdot\mathbf{n})\,\nabla_\Gamma u\big)$:

$$\begin{aligned}\mathrm{div}_\Gamma\big((\theta\cdot\mathbf{n})\,\nabla_\Gamma u\big) &= \nabla u\cdot\nabla_\Gamma(\theta\cdot\mathbf{n}) + (\theta\cdot\mathbf{n})\,\Delta_\Gamma u\\ &= \nabla u\cdot\nabla_\Gamma(\theta\cdot\mathbf{n}) + (\theta\cdot\mathbf{n})\Big(\Delta u - \mathcal{H}\partial_{\mathbf{n}} u - \partial^2_{\mathbf{nn}} u\Big).\end{aligned}$$

□

Remark 3.3.25. *In the proof of the differentiability of the state equation, the implicit function theorem states first that the function $C^{1,\infty}\ni\theta\mapsto U_\theta = u_\theta\circ(\mathrm{Id}+\theta)$ is differentiable, where u_θ is the solution to (3.19) on $\Omega_\theta = (\mathrm{Id}+\theta)(\Omega)$. We have $u_\theta:\Omega_\theta\to\mathbb{R}^d$ and therefore $U_\theta:\Omega\to\mathbb{R}^d$. It appears that the regularity of the function $\theta\mapsto U_\theta$ (given by the implicit function theorem) is only limited by the regularity of f and g. But since $u_\theta = U_\theta\circ(\mathrm{Id}+\theta)$, the derivative at 0 in the direction θ reads*

$$u'_\theta = U'_\theta - \nabla u\,\theta.$$

Therefore, if f is only $L^2(\Omega;\mathbb{R})$, u is a priori only in $H^1(\Omega;\mathbb{R})$ and then ∇u is in $L^2(\Omega;\mathbb{R})$. As a result, u'_θ is a priori only in $L^2(\Omega;\mathbb{R})$. This is the main reason why stronger assumptions on f and g are made, so that u has more regularity.

3.3.4 Compliance

Let u be the solution to (3.19). The compliance is defined by

$$J_1(\Omega) = \int_\Omega fu + \int_{\Gamma_N} gu = \int_\Omega |\nabla u|^2,$$

and its first-order shape derivative is given by ([15, Theorem 7]).

Proposition 3.3.26. *Let Ω be a smooth bounded open set and $\theta\in W^{1,\infty}\left(\mathbb{R}^d;\mathbb{R}^d\right)$. Assume that the data f, g as well as u are smooth, such as $f\in H^1(\Omega;\mathbb{R})$, $g\in H^2(\Omega;\mathbb{R})$ and $u\in H^2(\Omega;\mathbb{R})$. Then J_1 is shape-differentiable in the direction θ and*

$$\begin{aligned}J_1{}'(\Omega;\theta) &= 2\int_{\Gamma_N\cup\Gamma}(\theta\cdot\boldsymbol{n})\Big(\partial_{\boldsymbol{n}}(gu) + fu + \mathcal{H}gu\Big)\\ &\quad + \int_{\Gamma_D}|\nabla u|^2(\theta\cdot\boldsymbol{n}) - \int_{\Gamma_N\cup\Gamma}|\nabla u|^2(\theta\cdot\boldsymbol{n}),\end{aligned}$$

where $\mathcal{H}$ is the mean curvature.

3.3.5 Least square displacement criterion

Let u be the solution to (3.19). Given a target displacement $u_0\in\mathbb{R}$, $\alpha\in\mathbb{N}$ and $k\in L^\infty(\Omega;\mathbb{R})$, we define a least square displacement criterion by

$$J_2(\Omega) = \left(\int_\Omega k(x)|u-u_0|^\alpha\right)^{\frac{1}{\alpha}}.$$

Under sufficient regularity conditions on f and g, the function J_2 is shape-differentiable and by [15, Theorem 7],

Proposition 3.3.27. *Let Ω be a smooth bounded open set and $\theta\in W^{1,\infty}\left(\mathbb{R}^d;\mathbb{R}^d\right)$. Assume that the data f, g as well as u are regular enough in the sense that $f\in H^1(\Omega;\mathbb{R})$, $g\in H^2(\Omega;\mathbb{R})$ and $u\in H^2(\Omega;\mathbb{R})$. Then J_1 is shape-differentiable in the direction θ and*

$$\begin{aligned}J_2{}'(\Omega;\theta) &= \frac{C_0}{\alpha}\int_{\Gamma_N\cup\Gamma}(\theta\cdot\boldsymbol{n})\Big(k|u-u_0|^\alpha + \nabla u\cdot\nabla p - fp - \partial_{\boldsymbol{n}}(gp) - \mathcal{H}gp\Big)\\ &\quad + \frac{C_0}{\alpha}\int_{\Gamma_D}(\theta\cdot\boldsymbol{n})\Big(k|u-u_0|^\alpha + \nabla u\cdot\nabla p\Big),\end{aligned}$$

where $\mathcal{H}$ is the mean curvature, p the solution to

$$\left\{\begin{array}{rcll} -\mathrm{div}\left(\nabla p\right) & = & -\alpha k|u-u_0|^{\alpha-2}(u-u_0) & \text{in } \Omega, \\ \partial_{\boldsymbol{n}} p & = & 0 & \text{on } \Gamma_N \cup \Gamma, \\ p & = & 0 & \text{on } \Gamma_D, \end{array}\right. \tag{3.27}$$

and

$$C_0 = \left(\int_\Omega k|u-u_0|^\alpha\right)^{\frac{1}{\alpha}-1}.$$

3.3.6 Maximum-thickness criterion

The main purpose of a thickness criterion in shape optimization is to model manufacturing constraints. Given $d_{max} \geq 0$, a maximum-thickness constraint may be seen as the fact that at each point $x \in \Omega$, the disk of center x and of ray $\frac{d_{max}}{2}$ intersects the void :

$$\forall x \in \Omega,\ \mathcal{B}\left(x, \frac{d_{max}}{2}\right) \cap \left(\mathbb{R}^d \backslash \overline{\Omega}\right) \neq \emptyset.$$

For example, in Figure 3.2, the maximum-thickness constraint is violated at x_1 and satisfied at x_0. Since the signed

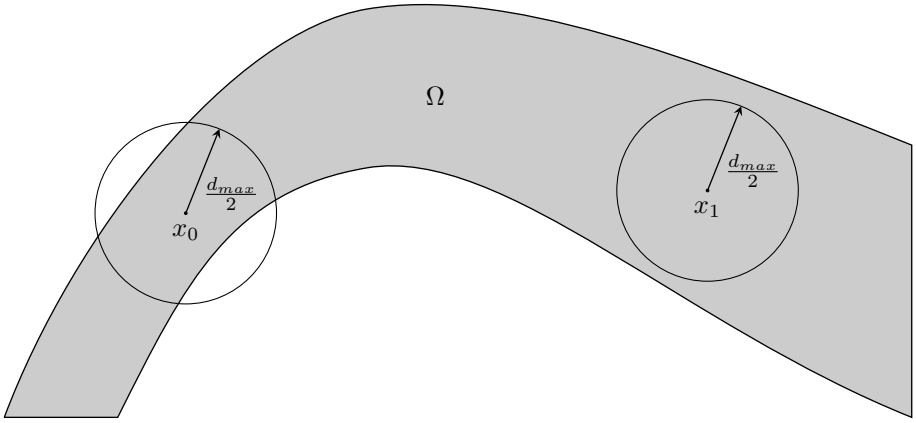


Figure 3.2: Violation of the maximum-thickness constraint on given shape.

distance function satisfies

$$d_\Omega(x) = \left\{\begin{array}{ll} -d(x,\partial\Omega) & \text{if } x \in \Omega, \\ 0 & \text{if } x \in \partial\Omega, \\ d(x,\partial\Omega) & \text{if } x \in \mathbb{R}^d \backslash \left(\Omega \cup \partial\Omega\right), \end{array}\right.$$

this constraint can be reformulated as

$$\forall x \in \Omega,\ d_\Omega(x) \geq -\frac{d_{max}}{2}.$$

The major difficulty of this kind of constraint is that it is pointwise. From a numerical point of view, this means that there are as many constraints in the optimization problem as the number of nodes in the mesh. That tends to be unmanageable. This is the main reason why Michailidis [83, §3.2.1] use a quadratic penalty function to aggregate the criterion :

$$\mathcal{P}_{Max}(\Omega) = \int_\Omega \left(\left(d_\Omega(x) + \frac{d_{max}}{2}\right)^-\right)^2 dx. \tag{3.28}$$

The penalty function is shape-differentiable and we have

Lemma 3.3.28. *The first-order shape derivative of (3.28) reads*

$$\mathcal{P}_{Max}{}'\left(\Omega;\theta\right) = \int_{\partial\Omega} -\Big(\theta(x)\cdot\boldsymbol{n}(x)\Big) \int_{\mathrm{ray}_{\partial\Omega}(x)\cap\Omega} \left(2\left(d_\Omega(z) + \frac{d_{max}}{2}\right)^- \prod_{i=1}^{d-1}\Big(1 + d_\Omega(z)\kappa_i(x)\Big)\,dz\right) dx,$$

where κ_i are the principal curvatures at the point $x \in \partial\Omega$.

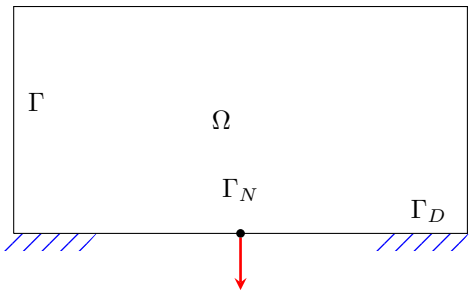


Figure 3.3: Boundary conditions.

3.3.7 Regularity of optimal shapes

In this section we consider a basic example of shape optimization in the two-dimensional space. Let us consider a bounded open shape Ω subjected to a load $g \in H^2(\Omega;\mathbb{R})$ such that $g = 0$ on Γ, and to no body forces ($f = 0$). The shape is clamped on Γ_D where Dirichlet boundary conditions are imposed such as depicted in Figure 3.3. Under these conditions, the displacement u is solution to

$$\left\{\begin{array}{rcll} -\mathrm{div}\left(\nabla u\right) & = & 0 & \text{in } \Omega, \\ \partial_{\mathbf{n}} u & = & g & \text{on } \Gamma_N \cup \Gamma, \\ u & = & 0 & \text{on } \Gamma_D. \end{array}\right. \tag{3.29}$$

We also impose that the part Γ_D and Γ_N are not subject to optimization; therefore we restrict the space of admissible to

$$\mathcal{U}_{ad} = \left\{\Omega \in \mathcal{O}_2 \mid \Gamma_D \cup \Gamma_N \subset \partial\Omega\right\},$$

and the space of derivation $\mathcal{F}$ to

$$\mathcal{F} = \left\{\theta \in C^1\left(\mathbb{R}^d;\mathbb{R}^d\right) \mid \theta = 0 \text{ on } \Gamma_D \cup \Gamma_N\right\}.$$

We now consider, for $\Lambda > 0$ the objective function

$$L(\Omega) = \int_\Omega |\nabla u|^2 + \Lambda \int_\Omega 1.$$

This consists in minimizing the compliance with a penalization on the volume of the shape. Under these assumptions, L is shape-differentiable and for $\theta \in \mathcal{F}$ we have

$$L'\left(\Omega;\theta\right) = \int_\Gamma \left(\theta \cdot \mathbf{n}\right)\left(\Lambda - |\nabla u|^2\right).$$

Proposition 3.3.29. *Let assume that there exists $\Omega \in \mathcal{O}_2$ that is a critical point of L. Therefore Ω does not have any hole.*

Proof. We have for all θ in $\mathcal{F}$, $L'\left(\Omega;\theta\right) = 0$, which implies that, on Γ

$$|\nabla u|^2 = \Lambda.$$

Moreover, on Γ we have $\partial_{\mathbf{n}} n = 0$. Therefore

$$\forall x \in \Gamma,\ |\nabla_\Gamma u|^2 = \Lambda.$$

The Sobolev embedding [27] in $\mathbb{R}^2$

$$H^2(\Omega) \subset C^{1,0}(\Omega) = C^1(\Omega)$$

implies that u is in $C^1(\Omega)$, and that ∇u is continuous over Ω. Now since $\Omega \in \mathcal{O}_2$, its normal is continuous, and therefore $\nabla_\Gamma u$ is also continuous over Γ. If there exists a hole, its boundary is necessary part of Γ let us denote it by γ. On γ the tangential gradient of u is continuous and satisfies $|\nabla_\Gamma u|^2 = \Lambda$. Since Ω lies in a subset of $\mathbb{R}^2$, the tangential gradient is constant. Let assume for example that $\nabla_\Gamma u = \sqrt{\Lambda}$. Integrating over γ leads to

$$\int_\gamma \nabla_\Gamma u = 0,$$

which means that

$$\sqrt{\Lambda}|\gamma| = 0.$$

Therefore $\Lambda = 0$ which is not possible since it is assumed positive. As a conclusion, there cannot exist any hole when the boundary is smooth. □

This property is very interesting. We have already seen that a way to get existence results for shapes was to require more regularity on admissible shapes (see Section 2.3). But here we can see that if there exists a smooth optimal shape, it must have no hole. Therefore, the restriction to smooth shapes has a really important cost, since there is only one possible topology. On the other hand it also proves that if non-smooth shapes are authorized, there will necessarily be singularities on the boundary of optimal shapes. Indeed, one can observe that optimization algorithms develop corners on shapes (see Chapter 12).

Chapter 4

The level-set method

Contents

4.1 Presentation of the level-set method

Since the early works of Osher and Sethian [97, 113], the level-set method has become a widely used method for shape modeling, especially in view of tracking the evolution of a shape, or equivalently of its boundary. It has already proven its effectiveness and its ability to solve a large variety of problems [98, 115], [129] or [10, 11, 12, 13]. It consists in working in an implicit framework where the shape Ω is represented by the set of negative values of a function $\mathbb{R}^d \ni x \mapsto \phi(x)$. In other words the level-set ϕ is a function such that

$$\begin{cases} \phi(x) < 0 & \text{if } x \in \Omega, \\ \phi(x) = 0 & \text{if } x \in \partial\Omega, \\ \phi(x) > 0 & \text{if } x \in \mathbb{R}^d \backslash (\Omega \cup \partial\Omega)\,. \end{cases}$$

One can mention two major advantages of the level-set method. At first, with this representation of a shape, a motion

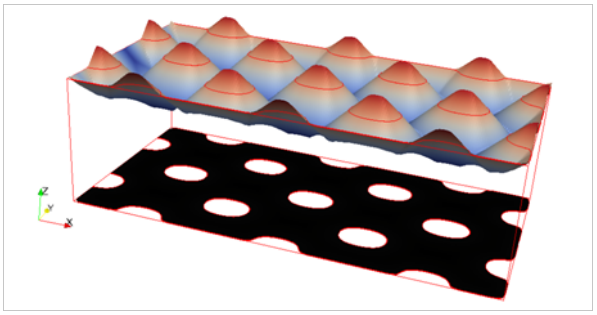


Figure 4.1: Signed distance function to a shape.

of an evolving domain $\Omega(t)$ over a time interval $[0, T]$ is tracked by a partial differential equation for a time-dependent level-set function $\phi(t, \cdot)$, namely a Hamilton-Jacobi equation. This equation admits a unique viscosity (or weak) solution global in time. The framework of viscosity solution (see Section 4.2.1) is well-suited for shape optimization since it allows topological changes. Secondly, a second advantage lies in a numerical point of view. Usually, the shape Ω is assumed to vary in a bounded working domain $\mathcal{D}$. Once this domain is meshed, it is sufficient for letting the level-set evolve. Therefore, there is no need of a costly remeshing process to track the motion of the shape.

The level-set gives also access quickly to different geometric quantities. Since the level-set satisfies $\phi(x) = 0$ on $\partial\Omega$, the normal to the shape $\mathbf{n}$ is given by the gradient of ϕ : $\mathbf{n} = \dfrac{\nabla\phi}{|\nabla\phi|}$ (at least in the vicinity of the boundary

when $\nabla\phi$ does not vanish). The mean curvature introduced in Definition 3.2.9 is computed by the divergence of the outer normal. With a Cartesian grid used to discretize the working domain $\mathcal{D}$ on which the level-set is evaluated, it becomes quite straightforward to use a finite difference scheme for computing the normal to the shape and the mean curvature. One can also introduce a sign function s, a Heaviside χ function and a Dirac function $\delta_{\partial\Omega}$. We introduce here the sign and Heaviside functions and postpone the introduction of the Dirac function to Section 11.1

Definition 4.1.1. *Let $\phi : \mathbb{R}^d \to \mathbb{R}$, and Ω a shape defined by the zero level-set of ϕ. We define the sign function s by*

$$\begin{cases} s(x) = 1 & \text{if } \phi(x) > 0, \\ s(x) = 0 & \text{if } \phi(x) = 0, \\ s(x) = -1 & \text{otherwise.} \end{cases} \tag{4.1}$$

This gives access to the characteristic function defined by

$$\chi = \frac{1}{2}(1 - s). \tag{4.2}$$

4.2 The Hamilton-Jacobi transport equation

With a level-set representation, the motion of a shape can be simply determined by a transport equation which is often called the *level-set advection equation*. In the particular case when the motion of the shape is always directed by its time-evolving outer normal, the level-set advection equation takes the form of a *Hamilton-Jacobi equation*. The ability of the Hamilton-Jacobi to track the motion of a shape has been widely studied by Osher and Sethian [97, 96, 110]. We first begin by recalling the link between shape motion and the Hamilton-Jacobi equation.

Let us consider a smooth time-evolving level-set function $\phi(t, \cdot)$ with $t \in [0, T]$. Let $\Omega_0 \in \mathcal{O}_\infty$ and ϕ_0 an associated level-set function defined by its signed distance function. Since it is a distance function, at least in the vicinity of $\partial\Omega_0$ the gradient of ϕ_0 does not vanish. For $v \in C^\infty(\mathcal{D})$ and $\phi_0 \in C^\infty(\mathcal{D}; \mathbb{R})$ assume that $\phi \in C^\infty([0, T] \times \mathcal{D})$ solves the following *Hamilton-Jacobi* equation

$$\begin{cases} \partial_t\phi(t, x) + v(x)|\nabla_x\phi(t, x)| = 0 & \text{on } (0, T) \times \mathcal{D}, \\ \phi(0, x) = \phi_0 & \text{on } \mathcal{D}. \end{cases} \tag{4.3}$$

This equation needs additional boundary conditions on $\partial\mathcal{D}$ in order to be well-posed. We will not give much detail about that, and refer to [46] for more details. For each $t \in [0, T]$ we consider a time-evolving domain $\Omega(t)$ as

$$\Omega(t) = \{x \in \mathcal{D} \mid \phi(t, x) < 0\}.$$

Since the gradient of ϕ_0 does not vanish in the neighborhood of $\partial\Omega_0$, there exists $\mathcal{O}$ a bounded open set of $\mathbb{R}^d$ containing $\partial\Omega_0$ and $\tau \in]0, T]$ such that in $[0, \tau[\times\mathcal{O}$ the level-set ϕ solution to the Hamilton-Jacobi equation (4.3) has its gradient bounded away from zero. Now, following the idea of the characteristics method, let $x_0 \in \partial\Omega_0$ and $[0, \tau] \ni t \mapsto x(t)$ given by

$$\begin{cases} \dfrac{dx}{dt}(t) &= v\big(x(t)\big)\dfrac{\nabla\phi(t, x(t))}{|\nabla\phi(t, x(t))|}, \\ x(0) &= x_0. \end{cases}$$

It appears that for each time $t \in [0, \tau]$, $x(t) \in \partial\Omega(t)$. The smoothness of ϕ ensures that

$$\begin{aligned} \frac{d\phi\big(t, x(t)\big)}{dt} &= \partial_t\phi\big(t, x(t)\big) + \frac{dx}{dt}(t) \cdot \nabla\phi\big(t, x(t)\big), \\ &= \partial_t\phi\big(t, x(t)\big) + v\big(x(t)\big)|\nabla\phi(t, x(t))|, \\ &= 0. \end{aligned}$$

Since $x(0) = x_0 \in \partial\Omega_0$ at $t = 0$, we have $\phi\big(0, x(0)\big) = 0$. Therefore, the invariance of $\phi\big(t, x(t)\big)$ with respect to t implies that

$$\forall t \in [0, \tau], \phi\big(t, x(t)\big) = 0,$$

and as a consequence

$$\forall t \in [0, \tau], x(t) \in \partial\Omega(t).$$

With all this, one can see that the motion of the boundary $\partial\Omega(t)$ is given by the speed vector $v(t, \cdot)\dfrac{\nabla\phi(t, \cdot)}{|\nabla\phi(t, \cdot)|}$. Since $\phi(t, \cdot)$ is a level-set function for $\Omega(t)$ the vector $\dfrac{\nabla\phi(t, \cdot)}{|\nabla\phi(t, \cdot)|}$ is on $\partial\Omega(t)$ the outer normal to the shape $\Omega(t)$. Therefore, the speed vector translating the motion of the boundary is always normal to the shape (see Figure 4.2).

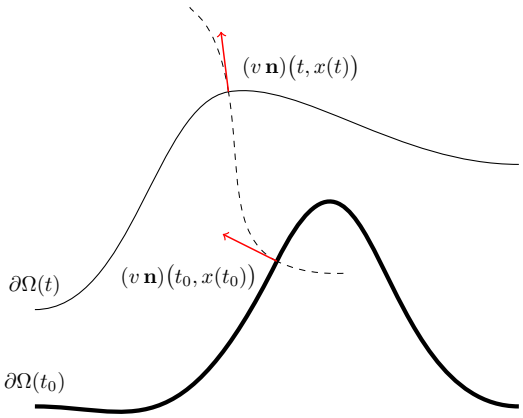


Figure 4.2: Normal evolution of a shape.

Remark 4.2.1. *It is also noticeable that the shape $\Omega(T)$ resulting from the solution to (4.3) depends only on $\Omega(0)$ which is the set $\{x \in \mathcal{D} \mid \phi_0(x) = 0\}$. This is a result from [18, 38, 61]. Indeed, let $z \to h(z)$ be an increasing function on $\mathbb{R}$ with $h'(z) > 0$ and $h(0) = 0$. When ϕ is a solution to (4.3) with initial data ϕ_0, then $h(\phi)$ is a solution for this equation with initial data $h(\phi_0)$. Since $h(0) = 0$ and h is increasing, the zero level-set of ϕ and of $h(\phi)$ are the same, meaning that $h(\phi)$ and ϕ represent the same shape.*

According to [64], the motion of a shape with respect to a vectorial velocity field V is more generally defined by the the viscosity solution to (see Section 4.2.1)

$$\begin{cases} \partial_t\phi(t,x) + V(t,x)\cdot\nabla_x\phi(t,x) = 0 & \text{on } (0,T)\times\mathcal{D}, \\ \phi(0,x) = \phi_0 & \text{on } \mathcal{D}. \end{cases}$$

4.2.1 Existence of solutions

In this section we focus on the existence of solutions to the Hamilton-Jacobi equations (4.3) on $[0,T]$. When $\Omega \in \mathcal{O}_\infty$ and T is sufficiently small, there exists a unique solution to this equation. But if T is too high, it appears that the bicharacteristics for solving the Hamiltonian system associated to the transport equation may cross. In that case, the level-set ϕ may be multi-valuated (see Remark 10.1.2). In other words, even if the initial condition ϕ_0 as well as the velocity field v are arbitrarily smooth, the solution to (4.3) may develop singularities. In oder to get an existence and uniqueness result for the solution to the Hamilton-Jacobi equation, one should enlarge the set of solutions and "choose" one of the multiple choices that may appear for the level-set. This is one achievement of the work of Crandall and Lions [46] on the *viscosity solutions* to this kind of equations. The different singularities that may appear are well described for example by Sethian [110] or [52, §1.1].

For the sake of completeness we recall here some notions about the viscosity solution theory for the Hamilton-Jacobi equation.

Definition 4.2.2. *Let $U \subset \mathbb{R}^d$ be an open set and $H : \mathbb{R}^d_x \times \mathbb{R}_u \times \mathbb{R}^d_p \times \mathcal{S}(\mathbb{R}^d) \to \mathbb{R}$ be a continuous function, where $\mathcal{S}(\mathbb{R}^d)$ is the set of $d \times d$ symmetric matrices. Consider the following general second-order Hamilton-Jacobi equation posed on $(0,T)\times U$:*

$$\frac{\partial u}{\partial t}\big(t,x\big) + H\big(x,u,\nabla u, D^2u\big)(t,x) = 0. \tag{4.4}$$

- *A function u is a viscosity subsolution of equation (4.4) if it is upper semicontinuous on U and, for any function ϕ of class C^2 on U such that $u-\phi$ reaches a local maximum at x,*

$$\frac{\partial u}{\partial t}(t,x) + H\big(x,u(x),\nabla\phi(x),D^2\phi(x)\big) \le 0.$$

- *A function u is a viscosity supersolution of equation (4.4) if it is lower semicontinuous on U, and, for any function ϕ of class C^2 on U such that $u-\phi$ reaches a local minimum at x,*

$$\frac{\partial u}{\partial t}(t,x) + H\big(x,u(x),\nabla\phi(x),D^2\phi(x)\big) \ge 0.$$

- *A function u is a viscosity solution of equation (4.4) if it is a viscosity subsolution and a viscosity supersolution.*

There are many ways to find viscosity solutions to (4.4). One can first mention Perron's method such as depicted in [64, §2.4]. It mainly consists in finding the largest viscosity subsolution. It is also more standard to consider the vicosity solution by their physical meaning : in some cases (see [18, 46]), they are seen as limits of an associated viscous equation

$$\frac{\partial u_\varepsilon}{\partial t}(t,x) + H\big(x, u_\varepsilon, \nabla u_\varepsilon, D^2 u_\varepsilon\big)(t,x) = \varepsilon \Delta u_\varepsilon. \tag{4.5}$$

while ε goes to zero.

4.2.2 Numerical scheme

Now let us focus on the numerical resolution of the Hamilton-Jacobi equation in the case of a normal evolution :

$$\begin{cases} \partial_t \phi(t,x) + v(x)|\nabla_x \phi(t,x)| = 0 & \text{on } (0,T)\times\mathcal{D}, \\ \phi(0,x) = \phi_0 & \text{on } \mathcal{D}, \end{cases} \tag{4.6}$$

where both the given scalar velocity field v and the initial data ϕ_0 are assumed to be bounded and uniformly continuous. Actually the results we refer to are also available for more general equations of the form

$$\begin{cases} \partial_t \phi(t,x) + H(x, \nabla\phi) = 0 & \text{on } (0,T)\times\mathcal{D}, \\ \phi(0,x) = \phi_0 & \text{on } \mathcal{D}. \end{cases} \tag{4.7}$$

The first case is then obtained when taking $H(x,p) = v(x)\,|p|$.

Starting from the work of Crandall and Lions [46], the numerical schemes to find viscosity solutions to Hamilton-Jacobi equations have been widely studied [105, 110, 114]. There are also many other ways to construct a "nice" numerical scheme, up to higher orders such as depicted for example in [97]. We recall here some background on this topic and detail a further used numerical scheme for (4.6). For the sake of readability we restrict ourselves - without loss of generality - to the two-dimensional case. Given $T > 0$ the space $(0,T)\times\mathbb{R}^2$ is meshed with a rectangular mesh with sizes $(\Delta t, \Delta x, \Delta y)$. Given $(n,i,j) \in \mathbb{N}\times\mathbb{Z}^2$, the point (t_n, x_i, y_j) denotes $(n\Delta t, i\Delta x, j\Delta y)$, and $x_{i,j}$ denotes $(x_i, y_j) = (i\Delta x, j\Delta y)$. For a numerical quantity U defined on the lattice $\Delta = \{(x_i,y_j)\,|\,(i,j)\in\mathbb{Z}\}$, its value at (x_i,y_j) for $t = t_n$ is written $U^n_{i,j}$. A general first-order numerical scheme may be written under the form

$$\begin{cases} \forall (n,i,j) \in \mathbb{N}\times\mathbb{Z}^2, & \phi^{n+1}_{i,j} = \phi^n_{i,j} - \Delta t \mathcal{H}\Big(x_{i,j}, D^{-x}_{i,j}\phi^n, D^{+x}_{i,j}\phi^n, D^{-y}_{i,j}\phi^n, D^{+y}_{i,j}\phi^n\Big), \\ \forall (i,j) \in \mathbb{Z}^2, & \phi^0_{i,j} = \phi_0(x_i,y_j) \end{cases} \tag{4.8}$$

where

$$D^{+x}_{i,j}\phi = \frac{\phi_{i+1,j}-\phi_{i,j}}{\Delta x}, \quad D^{-x}_{i,j}\phi = \frac{\phi_{i,j}-\phi_{i-1,j}}{\Delta x},$$
$$D^{+y}_{i,j}\phi = \frac{\phi_{i,j+1}-\phi_{i,j}}{\Delta y}, \quad D^{-y}_{i,j}\phi = \frac{\phi_{i,j}-\phi_{i,j-1}}{\Delta y}.$$

The numerical Hamiltonian $\mathcal{H}\Big(x_{i,j}, D^{-x}_{i,j}\phi^n, D^{+x}_{i,j}\phi^n, D^{-y}_{i,j}\phi^n, D^{+y}_{i,j}\phi^n\Big)$ is chosen to be an approximation of the real one $H(x,\nabla\phi)$. We also recall that the numerical Hamiltonian $\mathcal{H}$ is said to be consistent if for any $(a,b)\in\mathbb{R}^2$, and $x\in\mathbb{R}^2$

$$\mathcal{H}(x,a,a,b,b) = H\big(x,(a,b)\big).$$

Let us note that the first variable of $\mathcal{H}$ and H - namely x - is a two-dimensional one. All the other variables of $\mathcal{H}$ are scalar, whereas the second variable of H is a vector of $\mathbb{R}^2$. Moreover, the Hamiltonian $\mathcal{H}$ is said to be monotonous if it is increasing in the second and fourth arguments and decreasing in the third and fifth arguments. In the case where the numerical Hamiltonian $\mathcal{H}$ is both consistent and monotonous, Crandall and Lions [47, Theorem 1] proved that the numerical scheme converges to a viscosity solution to (4.7).

For the particular case of (4.6) when $H(x,p) = v(x)\,|p|$, we consider the following numerical scheme

$$\begin{cases} \forall (n,i,j) \in \mathbb{N}\times\mathbb{Z}^2, & \phi^{n+1}_{i,j} = \phi^n_{i,j} - \Delta t\Big(\max(v_{i,j},0)\nabla^+_{i,j}\phi^n + \min(v_{i,j},0)\nabla^-_{i,j}\phi^n\Big), \\ \forall (i,j) \in \mathbb{Z}^2, & \phi^0_{i,j} = \phi_0(x_i,y_j), \end{cases} \tag{4.9}$$

where

$$\nabla^+_{i,j}\phi = \left[\max\Big(\max(D^{-x}_{i,j}\phi,0); -\min(D^{+x}_{i,j}\phi,0)\Big)^2 + \max\Big(\max(D^{-y}_{i,j}\phi,0); -\min(D^{+y}_{i,j}\phi,0)\Big)^2\right]^{\frac{1}{2}},$$
$$\nabla^-_{i,j}\phi = \left[\max\Big(\max(D^{+x}_{i,j}\phi,0); -\min(D^{-x}_{i,j}\phi,0)\Big)^2 + \max\Big(\max(D^{+y}_{i,j}\phi,0); -\min(D^{-y}_{i,j}\phi,0)\Big)^2\right]^{\frac{1}{2}}.$$

It appears that this scheme is both consistent and monotonous, which ensures the stated convergence to a viscosity solution. A CFL condition on the time step has also to be added because this scheme is explicit in time. The scheme is said to be upwind since the computation of $\phi^{n+1}_{i,j}$ is only made with the values of $\phi^n_{i\pm1,j}$ and $\phi^n_{i,j\pm1}$ larger (respectively smaller) than $\phi^n_{i,j}$ if $v_{i,j}$ is negative (respectively positive).

4.3 Eikonal equation

The numerical scheme presented in Section 4.2.2 is a very practical way to compute a viscosity solution to a Hamilton-Jacobi equation. However it turns out that it is not numerically efficient. In practice the level-set often becomes too steep (or too flat) which causes a numerical error on the geometric properties of the shape such as for example the outer normal, and consequently the curvature, the perimeter or any boundary integral. In order to prevent these numerical inconveniences, the level-set may be periodically reinitialized such that it stays close to some signed distance function. This is what the eikonal equation aims at :

$$\left\{\begin{array}{rcll} |\nabla\phi| & = & 1 & \text{on } \mathcal{D}, \\ \phi & < & 0 & \text{in } \Omega, \\ \phi & > & 0 & \text{in } \mathcal{D}\backslash\left(\Omega\cup\partial\Omega\right), \\ \phi & = & 0 & \text{on } \partial\Omega. \end{array}\right. \tag{4.10}$$

4.3.1 Transport equation

Instead of solving exactly the eikonal equation, it is possible to solve the following transport equation :

$$\left\{\begin{array}{ll} \partial_t\phi(t,x) + s\big(\phi_0(t,x)\big)\Big(|\nabla_x\phi(t,x)| - 1\Big) = 0 & \text{on } (0,T)\times\mathcal{D}, \\ \phi(0,x) = \phi_0 & \text{on } \mathcal{D}, \end{array}\right. \tag{4.11}$$

where s is the sign function. It is not mathematically clear that this equation should have a solution. When $\nabla\phi \neq 0$ one can rewrite $s\big(\phi(t,x)\big)|\nabla_x\phi(t,x)|$ as $s\big(\phi(t,x)\big)\mathbf{n}(t,x)\cdot\nabla_x\phi(t,x)$ with $\mathbf{n}(t,x) := \dfrac{\nabla_x\phi(t,x)}{|\nabla_x\phi(t,x)|}$. When $x\in\partial\Omega(t)$, $\mathbf{n}(t,x)$ is on $\partial\Omega(t)$ the outer normal to $\Omega(t) = \{y\in\mathcal{D}\,|\,\phi(t,y)<0\}$ at x. This means that the characteristics are directed along the normal, and evolve with a speed given by the sign function.

The numerical scheme presented in Section 4.2.2 may also be used for solving this equation. Even if there is no a priori existence of a solution to (4.11), in practice this scheme always finds a numerical solution. It is in the vicinity of the boundary that it is important for the level-set to be close to a signed distance function. Therefore, the numerical scheme can also be stopped before the characteristics associated to the velocity $s(\phi)\mathbf{n}$ cross.

This way of solving the eikonal equation is very practical and is not very expensive from a computational point of view. However it turns out that even if the property $|\nabla\phi| = 1$ may be accurately preserved, the boundary of the shape is not exactly conserved. Theoretically, if $\phi(x) = 0$ on the boundary, $s(\phi(x))$ also vanishes and therefore the level-set ϕ does not change at this point. But numerically, unless the boundary contains exactly a point of a mesh grid, this should not happen. In a one-dimensional case (see Chapter 14) we will see explicitly that for a grid cell around the boundary, the conserved quantity is not the position of the interface but only the mean value of the level-set on the cell.

We could expect that the solution to (4.11) depends only on the initial shape. This would mean that it depends only on the zero level-set of ϕ_0 and not on the other values of this initial data. But there is no theoretical result about that. However, we can consider the two systems

$$\left\{\begin{array}{ll} \partial_t\phi_1(t,x) + |\nabla_x\phi_1(t,x)| - 1 = 0 & \text{on } (0,T)\times\mathcal{D}\backslash\big(\Omega\cup\partial\Omega\big), \\ \phi_1(0,x) = \phi_0 & \text{on } \mathcal{D}\backslash\big(\Omega\cup\partial\Omega\big), \\ \phi_1(x) = 0 & \text{on } \partial\Omega, \end{array}\right. \tag{4.12}$$

and

$$\left\{\begin{array}{ll} \partial_t\phi_2(t,x) + |\nabla_x\phi_2(t,x)| - 1 = 0 & \text{on } (0,T)\times\big(\Omega\cup\partial\Omega\big), \\ \phi_2(0,x) = \phi_0 & \text{on } \big(\Omega\cup\partial\Omega\big), \\ \phi_2(x) = 0 & \text{on } \partial\Omega, \end{array}\right. \tag{4.13}$$

and take

$$\left\{\begin{array}{ll} \phi(t,x) = \phi_1(t,x) & \text{on } (0,T)\times\mathcal{D}\backslash\big(\Omega\cup\partial\Omega\big), \\ \phi(t,x) = -\phi_2(t,x) & \text{on } (0,T)\times\Omega. \end{array}\right. \tag{4.14}$$

Now since the respective Hamiltonians are continuous, theses problems are well posed.

Remark 4.3.1. *The shape defined by the level-set function given by (4.14) only depends on the initial shape Ω defined by ϕ_0. We can see that in the continuous case of (4.12) the zero level-set is conserved. A fortiori, for any x such that $\phi_0(x) = 0$, since $\phi_1 = 0$ is imposed on $\partial\Omega$ as a boundary condition, we have $\phi_1(t,x) = 0$. This means that $\partial\Omega\subset\partial\Omega_t$ where $\partial\Omega_t = \{x\in\mathcal{D}\mid\phi_1(t,x) = 0\}$.*

Now reciprocally, we could check that ϕ_1 cannot vanish at any point that is not on $\partial\Omega$. We start by assuming that there exists T and $\mathcal{B}$ such that for every $x_0\in\mathcal{B}$, the solution to

$$\left\{\begin{array}{ll} \dfrac{dx}{dt} & = \dfrac{\nabla\phi_1\big(t,x(t)\big)}{|\nabla\phi_1\big(t,x(t)\big)|}, \\ x(0) & = x_0, \end{array}\right. \tag{4.15}$$

is well-defined, meaning in particular that $\nabla\phi_1\big(t,x(t)\big) \neq 0$. *Let us also assume the existence of* $(t_0,y) \in [0,T]\times\mathcal{B}$ *such that* $\phi_1(t_0,y)=0$. *Let* x *be the characteristic defined by*

$$\begin{cases} \dfrac{dx}{dt} &= \dfrac{\nabla\phi_1\big(t,x(t)\big)}{|\nabla\phi_1\big(t,x(t)\big)|}, \\ x(t_0) &= y \end{cases} \tag{4.16}$$

We can see that $\dfrac{d}{dt}\phi_1\big(t,x(t)\big) = 1$. *Then* $\phi_1\big(t_0,x(t_0)\big) = \phi_1\big(0,x(0)\big) + t_0 > 0$ *which contradicts the assumption* $\phi\big(t_0,y\big)=0$. *But the following equation*

$$\begin{cases} \dfrac{dz}{dt} &= \dfrac{\nabla\phi_1\big(t,z(t)\big)}{|\nabla\phi_1\big(t,z(t)\big)|}, \\ z(0) &= x(0) \end{cases} \tag{4.17}$$

has a unique solution which is $t\mapsto z(t)=x(0)$. *Since* $t\mapsto x(t)$ *satisfies this same equation, we deduce that* $y=x(0)$ *and that* $\phi(0,y)=0$. *This means that in the vicinity of the boundary, no point can be added to the boundary, and that* $\partial\Omega_t \subset \partial\Omega$.

Doing the same for (4.13), we can see that (4.14) conserves the boundary of Ω *(at least in the vicinity of the initial boundary).*

4.3.2 The fast-marching method

The fast-marching method is another way to solve the eikonal equation. This method, introduced by Sethian [109, 111, 112, 110], is well-suited for tracking the motion of interfaces when the velocity field depends only on the space variable and has a constant sign. This is especially efficient for the eikonal equation which is basically a static or stationary Hamilton-Jacobi equation where the speed is uniformly equals to 1 and independent of the time. Let us here consider the following eikonal equation in $\mathcal{D}\subset\mathbb{R}^2$

$$\begin{cases} |\nabla\phi| &= 1 \quad \text{on } \mathcal{D}, \\ \phi &< 0 \quad \text{in } \Omega, \\ \phi &> 0 \quad \text{in } \mathcal{D}\backslash\left(\Omega\cup\partial\Omega\right), \\ \phi &= 0 \quad \text{on } \partial\Omega. \end{cases} \tag{4.18}$$

The two-dimensional case is chosen for the sake of clarity, but everything that follows is also available in higher dimension.

The fast-marching method relies on computing the values of ϕ at grid nodes in an upwind fashion as if they were given by the propagation of a front starting from $\partial\Omega$. We describe here the propagation outside Ω, given that the propagation inside Ω is made in the exact same way. With the same notation as in Section 4.2.2 for a discretization on a grid, the upwind discretization of the gradient for a positive velocity field ($1\geq 0$) reads

$$\max\Big(\max(D^{-x}_{i,j}\phi,0);-\min(D^{+x}_{i,j}\phi,0)\Big)^2 + \max\Big(\max(D^{-y}_{i,j}\phi,0);-\min(D^{+y}_{i,j}\phi,0)\Big)^2 = 1. \tag{4.19}$$

In the upwind approximation of the gradient at (i,j), only the values smaller than $\phi_{i,j}$ are taken into account. The key ingredient is to divide the set of nodes into three categories : *accepted*, *narrow* and *far*. The *accepted* nodes are all nodes where the front has already gone through. The values of ϕ at these nodes will no longer be updated. The *narrow* nodes consist of every nodes that are not already *accepted* but are neighbors (connected by an edge) of an *accepted* node. The values of ϕ at these nodes may be updated many times. Finally, the *far* nodes are every other nodes where ϕ is initialized to ∞. Once a node is assigned as *narrow*, a trial $\tilde{\phi}$ value is computed such that it solves (4.19), where $\phi_{i,j}$ is replaced by $\tilde{\phi}$ (we recall that $\phi_{i,j}$ intervenes in the computation of every $D^{\pm x,y}_{i,j}\phi$) and only the *accepted* values are used among the set $\{\phi_{i-1,j},\phi_{i+1,j},\phi_{i,j-1},\phi_{i,j+1}\}$.

The sketch of the algorithm is the following

Algorithm 4.3.2. *The fast-marching algorithm*

1. *Initialization.*
   - *Compute the distance to the boundary of each node being in a cell crossed by the boundary* Ω. *All these nodes become* accepted.
   - *Put every neighbor of an* accepted *node in the set of* narrow *nodes.*
   - *Put every other node in the set of* far *nodes.*
2. *Loop (while the set of* narrow *nodes is non-empty)*
   - *Find the* narrow *node with the smallest value of* ϕ. *Put it in the set of* accepted *nodes. Enlarge the set of* narrow *nodes with all its neighbors.*

- *Recompute a trial value of ϕ for every* narrow *node.*

This algorithm provides a viscosity solution to (4.18). The tricky part consists in having an efficient sorting procedure so that finding the *narrow* node with the smallest value of ϕ is not too costly. Though it remains quite efficient since it converges in $O\big(N\log(N)\big)$ iterations with a Cartesian grid of size N. The main advantage of the fast-marching method lies in the fact that the boundary of the shape should be preserved. The initialization of the method starts with computing the exact distance function to the shape. Therefore the values of the new level-set preserve the position of the interface. Nevertheless this is kind of a theoretical advantage, since most of the time it is not possible to numerically compute the exact distance to the shape. Usually the exact distance to the interface is computed with linear approximation of the level-set. But once the distance is computed, the linear approximation is no longer available. Therefore one can say that the fast-marching procedure is not a projection, in the sense that distance functions are not fixed points of the procedure. In Section 14.3 we will give an example of non-conservation of the boundary by the fast-marching method.

4.3.3 Redistanciation-free method

Moving a shape requires only the normal component of a boundary vector field. When the motion of the shape is tracked by a Hamilton-Jacobi equation (4.3) the velocity field - directed along the time-dependent outer normal to the shape - has to be defined on the whole space. Indeed, the Hamilton-Jacobi equation moves all the level-sets and not only the zero level-set. The natural velocity fields given by shape derivatives are defined only on the boundary. Thus they have to be extended to the whole space. The only requirement on the extension is that its restriction to the boundary matches with the field to be extended. There are many different ways to compute extensions of velocity fields [15, 53, 32]. In this section we will briefly present an idea of Adalsteinsson and Sethian [3] for which the coupling of the advection equation with a redistanciation process does not seem necessary. The key ingredient is the construction of a time-dependent velocity field v such that at each time

$$\nabla v\cdot\nabla\phi = 0. \tag{4.20}$$

Assume that such a field exists and that ϕ solves the Hamilton-Jacobi equation (4.3). One has

$$\frac{d|\nabla\phi|^2}{dt} = 2\nabla\phi\cdot\frac{d\nabla\phi}{dt}$$

Since ϕ solves the Hamilton-Jacobi equation, the time derivative of its gradient is given by

$$\frac{d\nabla\phi}{dt} + |\nabla\phi|\nabla v + v\,\nabla|\nabla\phi| = 0.$$

Therefore

$$\frac{d|\nabla\phi|^2}{dt} = -2|\nabla\phi|\,\nabla\phi\cdot\nabla v - v\nabla\phi\cdot\nabla|\nabla\phi|,$$

and starting from a distance function with $|\nabla\phi| = 1$, the property $\nabla\phi\cdot\nabla v = 0$ leads to

$$\frac{d|\nabla\phi|^2}{dt} = 0,$$

which ensures that the level-set function remains a distance function. Therefore this choice of extension for the field v ensures that the level-set keeps being a distance-function, and that there is no need for a redistanciation.

All the difficulty remains now to build - at least numerically - a velocity field satisfying (4.20) at all times. There is a priori no chance to find a theoretical construction since it would depend on the solution to the Hamilton-Jacobi equation for which the field is required. Its construction is only implicit. The work of Adalsteinsson and Sethian consists in applying a fast-marching method for the construction of a velocity field satisfying the expected requirement. Naturally, the extension is to be recomputed at each time step of the resolution of the Hamilton-Jacobi equation. The sketch of their method is the following

Algorithm 4.3.3. *Advection*

- *Start from a level-set ϕ^n at time step $n\Delta t$.*
- *Build a temporary signed distance function $\tilde{\phi}$ that has the same zero level-set as ϕ^n.*
- *Build an extended velocity field $\tilde{v}^n$ such that it matches $\tilde{v}^{n-1}$ on the interface and that $\nabla\tilde{v}^n\cdot\nabla\tilde{\phi} = 0$.*
- *Make one step of the numerical scheme for the Hamilton-Jacobi equation with velocity $\tilde{v}^n$.*

The numerical method they implemented shows very convincing results. This method is not computationally cheaper than the coupling of the Hamilton-Jacobi equation with the eikonal equation for reinitializing the level-set. At each time step the eikonal equation is indeed used for computing a distance function and the extended velocity. It consists in solving the Hamilton-Jacobi equation with a time-dependent velocity. But the main advantage is that there is no perturbation on the current level-set. As we will see later, this should be a great improvement in the perspective of optimization.

One can also note that it is possible to start from an already extended velocity. If v is initially defined on the whole space, this means that at each time step the extended velocity should match with v on the current interface, instead of $\tilde{v}^{n-1}$.

Chapter 5

Finite element method

Contents

Partial differential equations have historically first been numerically solved by finite difference methods. The main idea was introduced by Courant, Friedrichs and Lewy [45] in the early 30s and is essentially based on Taylor approximation to compute numerical derivatives. Based on previous other formal works ([44] in the 40s), the finite element method was first introduced in mechanics in the 70s (see [133] for example). We also refer to the work of Thomée [125] for a historical review of numerical resolution of partial differential equations. It relies on the approximation of variational equalities in piecewise polynomial spaces. The finite element method is very popular for it is an easy way to solve partial differential equations on any kind of geometry. Finite difference schemes are also still widely used but preferably for time discretization in evolution problems. In this section we will focus on the finite element method, for which we recall some usual prerequisites.

5.1 Variational approximation

The general framework of variational formulations is the following. Let us consider a Hilbert space $\mathcal{V}$, a bilinear continuous and coercive form $a : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ and a linear continuous form $l : \mathcal{V} \to \mathbb{R}$. All the variational formalism is based on the Lax-Milgram theorem ([8, Théorème 3.3.1] for example) that ensures that the variational formulation

$$\text{find } u \in \mathcal{V} \text{ such that} \quad \forall v \in \mathcal{V}, \quad a(u, v) = l(v), \tag{5.1}$$

has a unique solution when the above assumptions on a and l are satisfied. This important result provides a mathematical framework for a large part of integral and partial differential equations, including in particular linear elliptic equations.

Usually the partial differential equations considered are set in Sobolev spaces such as H^1, H^1_0, H^2. The main concept of finite element method is to reduce the variational space (namely H^1, etc.) to a subspace of finite dimension. Courant [44] was the first to propose the idea to consider a triangulation adapted to the geometry of the shape and the variational space $\mathcal{V}_h$ made of continuous piecewise linear functions on the triangulation. For a given triangulation, this space is naturally of finite dimension and a subspace of H^1. Now it remains to consider what is called the *internal variational approximation*

$$\text{find } u_h \in \mathcal{V}_h \text{ such that} \quad \forall v_h \in \mathcal{V}_h, \quad a(u_h, v_h) = l(v_h). \tag{5.2}$$

Under the suitable assumptions on a (continuity and coercivity) this problem now sums up to a linear system with a finite-dimensional positive definite matrix [8, Lemme 6.1.1].

In this framework we also recall an error estimate due to Céa [8, Théorème 6.1.2] between u and u_h, the respective solutions to (5.1) and (5.2).

Lemma 5.1.1. *Let u and u_h be the solutions to (5.1) and (5.2) respectively. One has the following estimate*

$$\|u - u_h\| \leq \frac{M}{\nu} \inf_{v_h \in \mathcal{V}_h} \|u - v_h\| ,$$

where M and ν are respectively the constants of continuity and coercivity of the bilinear map a.

5.1.1 Interpolation error

In this section we consider a triangulation $\mathcal{T}_h$ of size h. For $k \geq 1$ we introduce the Lagrange finite element space $\mathcal{V}_h$ of continuous scalar functions that are piecewise polynomials of degree k :

$$\mathcal{V}_h = \left\{ u \in C(\mathcal{D}; \mathbb{R}) \mid \forall \tau \in \mathcal{T}_h,\ u_{|\tau} \text{ is a polynomial of degree } k \right\}.$$

With N_h the dimension of $\mathcal{V}_h$ and $(a_i)_{1 \leq i \leq N_h}$ being the degrees of freedom of $\mathcal{V}_h$, there exists [7, Proposition 2.35] a basis $(\varphi_i)_{1 \leq i \leq N_h}$ of $\mathcal{V}_h$ such that

$$\forall 1 \leq i, j \leq \mathbb{N}_h,\ \varphi_i(a_j) = \delta_{ij},$$

and

$$\forall v \in \mathcal{V}_h,\ v(x) = \sum_{i=1}^{N_h} v(a_i)\varphi_i(x).$$

In particular, one can notice that on each cell τ such that a_i is not part of, the basis function φ_i vanishes. Therefore, the support of φ_i is exactly the union of the cells τ that contain a_i. In the case of piecewise linear function - namely the space $\mathbb{P}_1$ - the degrees of freedom coincide with the set of vertices of the triangulation. For the space $\mathbb{P}_2$, the degrees of freedom is the union of the set of vertices of the triangulation and the set of nodes at the middle of every edges of the triangulation.

Definition 5.1.2. *For $v \in \mathcal{V}_h$ we now define the interpolation operator $\mathcal{I}^k_{\mathcal{T}_h}$ by*

$$\mathcal{I}^k_{\mathcal{T}_h} v = \sum_{i=1}^{N_h} v(a_i)\varphi_i.$$

Under suitable assumptions on the triangulation [22, Théorème IX.1.5], there is an estimate on the interpolation error :

Theorem 5.1.3. *Let $l \in \mathbb{N}$ such that $0 \leq l \leq k+1$ and $(p, q) \in \mathbb{N}^2$ with $1 \leq p \leq q \leq +\infty$ such that $W^{l,p}(\mathcal{D}) \subset C(\mathcal{D})$. There exists $c > 0$ dependent only on l, p, q such that for all $v \in W^{l,p}(\mathcal{D})$*

$$\left\| v - \mathcal{I}^k_{\mathcal{T}_h} v \right\|_{L^q(\mathcal{D})} \leq c h^{l - \frac{d}{p} + \frac{d}{q}} |v|_{W^{l,p}(\mathcal{D})},$$

where

$$|v|^p_{W^{l,p}(\mathcal{D})} := \sum_{|\alpha| = l} \|D^\alpha v\|^p_{L^p(\mathcal{D})}$$

Remark 5.1.4. *The assumption $W^{l,p}(\mathcal{D}) \subset C(\mathcal{D})$ is needed in the proof of this theorem for $\mathcal{I}^k_{\mathcal{T}_h} v$ may not be defined when v is only in $W^{l,p}(\mathcal{D})$. If l, p, k are such that the inclusion is not satisfied but v is continuous, the result remains true (see also [22]).*

Now if we take $d \leq 3$ and $k = 1$, that is we consider piecewise linear approximations, $l = 2$, $p = 2$ and $q = +\infty$. When $u \in H^2(\mathcal{D})$, Theorem 5.1.3 leads to

$$\left\| u - \mathcal{I}^k_{\mathcal{T}_h} u \right\|_{L^\infty(\mathcal{D})} \leq c h^{2 - \frac{d}{2}} |u|_{H^2(\mathcal{D})}. \tag{5.3}$$

Notice that since $d \leq 3$, the power of h in the error estimate is always positive, meaning that there is a L^∞-convergence result as soon as the function u is in $H^2(\mathcal{D})$. The regularity of the solution to (5.1) is a priori only in $H^1(\mathcal{D})$. But in view of shape differentiation, the applied loads as well as the shape have to be sufficiently regular so that u is $H^2(\mathcal{D})$ by elliptic regularity [27] (see Remark 3.3.25). In addition when u is in $H^2(\mathcal{D})$ (with a space dimension less than 3) the Sobolev embedding

$$H^2(\mathcal{D}) \subset C^{0,\alpha}(\mathcal{D}) \subset C(\mathcal{D})$$

ensures that u is continuous and therefore

$$\begin{aligned} \|u - u_h\|_{L^\infty(\mathcal{D})} &\leq \inf_{v_h \in \mathcal{V}_h} \|u - v_h\|_{L^\infty(\mathcal{D})} \\ &\leq \left\| u - \mathcal{I}^k_{\mathcal{T}_h} u \right\|_{L^\infty(\mathcal{D})} \\ &\leq c h^{2 - \frac{d}{2}} |u|_{H^2(\mathcal{D})}. \end{aligned}$$

5.1.2 $\mathbb{Q}_1$ finite element in the two-dimensional space

In numerical practice, working always with a mesh adapted to the geometry is time consuming. This is the reason why it can be interesting to work with a fixed mesh, and to track the evolution of the boundary with a level-set method for example. The choice of a square mesh may also be well-suited for it eases finite difference computations - let us mention here that the numerical scheme for solving the advection equation (see Section 4.2.2) requires finite difference approximations on the level-set. Thus, we consider a finite element approximation on a mesh with quadrangular elements denoted also $\mathcal{Q}_h$.

We could also have made the choice of working with a triangular mesh adapted to the geometry with remeshing process, and adapt the numerical scheme for advection equation to non-conforming grids. This is one of the main achievements of the work of Abgrall [1, 2], Hu and Shu [74] or Dapogny [52].

In a cell $K \in \mathcal{Q}_h$, the Lagrange finite element requires at least to work with quadratic polynomials : the problem of finding a polynomial of degree 1 with prescribed values on the vertices of a square is overdetermined. In order not to increase too much the dimension of the variational space $\mathcal{V}_h$ we deal here with quadratic approximations. In the two-dimensional space it consists in considering

$$\mathcal{V}_h = \Big\{ u \in C(\mathcal{D};\mathbb{R}) \mid \forall K \in \mathcal{Q}_h,\ u_{|K} \text{ is of the form } u_{|K}(x) = a_0 + a_1 x + a_2 y + a_3 xy \Big\}.$$

Now for $v, w \in \mathcal{V}_h$ we will detail the way to compute integrals such as

$$\int_{\mathcal{D}} v\, w, \quad \int_{\mathcal{D}} \nabla v \cdot \nabla w.$$

These two kinds of integrals are quite simple since a priori v, w are assumed to have scalar values. However the principle is exactly the same for vectorial functions. Therefore it is also easy to extend these computations to the case of homogeneous linear elasticity where the approximation matrices can be computed with the same principle.

When v, w are in $\mathcal{V}_h$ these integrals depend only on the components of v, w on a basis of $\mathcal{V}_h$, and their computations sum up to matricial evaluations. Let assume that

$$v(x) = \sum_{i=1}^{N_h} v_i \varphi_i(x), \quad w(x) = \sum_{i=1}^{N_h} w_i \varphi_i(x).$$

The integrals now read

$$\int_{\mathcal{D}} vw = \sum_{1 \le i,j \le N_h} v_i w_j \int_{\mathcal{D}} \varphi_i \varphi_j,$$
$$\int_{\mathcal{D}} \nabla v \cdot \nabla w = \sum_{1 \le i,j \le N_h} v_i w_j \int_{\mathcal{D}} \nabla \varphi_i \cdot \nabla \varphi_j.$$

Denoting by $V = (v_1, \cdots, v_{N_h})^T$, $W = (w_1, \cdots, w_{N_h})^T$ and

$$M_1 = \left(\int_{\mathcal{D}} \varphi_i \varphi_j \right)_{1 \le i,j \le N_h}, \quad M_2 = \left(\int_{\mathcal{D}} \nabla \varphi_i \cdot \nabla \varphi_j \right)_{1 \le i,j \le N_h}$$

the mass and stiffness matrices, the integrals write

$$\int_{\mathcal{D}} vw = V^T M_1 W, \quad \int_{\mathcal{D}} \nabla v \cdot \nabla w = V^T M_2 W.$$

Since the supports of the basis functions are reduced to the neighboring cells of a vertex, when i, j are such that the vertices a_i and a_j are not neighbors, the (i, j) entries of M_1 and M_2 are zero. Therefore these matrices have a sparse structure.

Following the formalism of Maday and al [22], each cell K is brought to a reference cell $\hat{K}$ such as depicted on Figure 5.1. The vertices of the cells are respectively denoted by V_i and $\hat{V}_i$. The boundary of a cell K is composed of four curves defined by $[-1, 1] \ni t \mapsto f_i(t) \in \mathbb{R}^2$ with $i \in [\![1, 4]\!]$. Each function f_i is assumed to be smooth, injective and such that

$$f_1\big([-1,1]\big) = [V_1, V_2], \quad f_2\big([-1,1]\big) = [V_2, V_3],$$
$$f_3\big([-1,1]\big) = [V_3, V_4], \quad f_4\big([-1,1]\big) = [V_4, V_1],$$

and such that

$$f_1(1) = V_2 = f_2(-1),$$
$$f_2(1) = V_3 = f_3(-1),$$
$$f_3(1) = V_4 = f_4(-1),$$
$$f_4(1) = V_1 = f_1(-1).$$

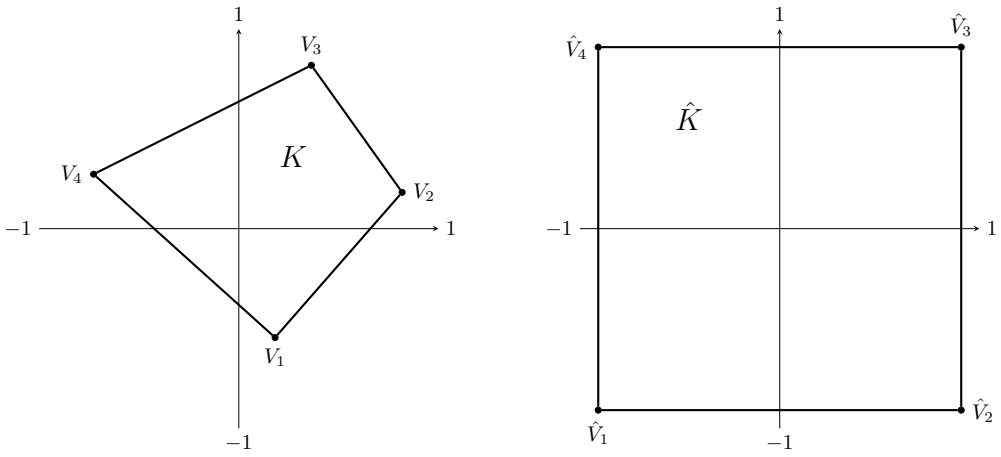


Figure 5.1: Transformation of a cell K to a reference cell $\hat{K}$.

Let $\mathbf{F} : \hat{K} \to K$ be defined by

$$\begin{aligned}\mathbf{F}(\hat{x},\hat{y}) = {} & \frac{1-\hat{x}}{2} f_1(\hat{y}) + \frac{1+\hat{x}}{2} f_3(\hat{y}) \\ & + \frac{1-\hat{y}}{2}\left(f_2(\hat{x}) - \frac{1-\hat{x}}{2} f_2(-1) - \frac{1+\hat{x}}{2} f_2(1) \right) \\ & + \frac{1+\hat{y}}{2}\left(f_4(\hat{x}) - \frac{1-\hat{x}}{2} f_4(-1) - \frac{1+\hat{x}}{2} f_4(1) \right).\end{aligned}$$

It maps the edges of $\hat{K}$ to the edges of K. When the cell K is a quadrangle and (x_i, y_i) are the coordinates of the vertices V_i one can define the f_i functions by

$$f_1(t) = \begin{pmatrix} x_1 + \frac{1}{2}(x_2 - x_1)(t+1) \\ y_1 + \frac{1}{2}(y_2 - y_1)(t+1) \end{pmatrix}, \qquad f_2(t) = \begin{pmatrix} x_2 + \frac{1}{2}(x_3 - x_2)(t+1) \\ y_2 + \frac{1}{2}(y_3 - y_2)(t+1) \end{pmatrix},$$

$$f_3(t) = \begin{pmatrix} x_3 + \frac{1}{2}(x_4 - x_3)(t+1) \\ y_3 + \frac{1}{2}(y_4 - y_3)(t+1) \end{pmatrix}, \qquad f_4(t) = \begin{pmatrix} x_4 + \frac{1}{2}(x_1 - x_4)(t+1) \\ y_4 + \frac{1}{2}(y_1 - y_4)(t+1) \end{pmatrix}.$$

Now we introduce $\hat{K} \ni (\hat{x},\hat{y}) \mapsto N_i(\hat{x},\hat{y})$ with

$$N_1(\hat{x},\hat{y}) = \frac{(1-\hat{x})(1-\hat{y})}{4}, \qquad N_2(\hat{x},\hat{y}) = \frac{(1+\hat{x})(1-\hat{y})}{4},$$

$$N_3(\hat{x},\hat{y}) = \frac{(1+\hat{x})(1+\hat{y})}{4}, \qquad N_4(\hat{x},\hat{y}) = \frac{(1-\hat{x})(1+\hat{y})}{4}.$$

In this context, the function $\mathbf{F}$ writes

$$\mathbf{F}(\hat{x},\hat{y}) = \begin{pmatrix} x_1 N_1(\hat{x},\hat{y}) + x_2 N_2(\hat{x},\hat{y}) + x_3 N_3(\hat{x},\hat{y}) + x_4 N_4(\hat{x},\hat{y}) \\ \\ y_1 N_1(\hat{x},\hat{y}) + y_2 N_2(\hat{x},\hat{y}) + y_3 N_3(\hat{x},\hat{y}) + y_4 N_4(\hat{x},\hat{y}) \end{pmatrix}.$$

Therefore, when we aim at computing the integral of a function g on K we have

$$\int_K g(x,y)\,dx\,dy = \int_{\hat{K}} g \circ \mathbf{F}(\hat{x},\hat{y}) \det\left(\partial \mathbf{F}(\hat{x},\hat{y})\right) d\hat{x}\,d\hat{y}, \tag{5.4}$$

where

$$\begin{cases} \partial_{\hat{x}} \mathbf{F}_x &= -x_1 \dfrac{1-\hat{y}}{4} + x_2 \dfrac{1-\hat{y}}{4} + x_3 \dfrac{1+\hat{y}}{4} - x_4 \dfrac{1+\hat{y}}{4}, \\ \partial_{\hat{y}} \mathbf{F}_x &= -x_1 \dfrac{1-\hat{x}}{4} - x_2 \dfrac{1+\hat{x}}{4} + x_3 \dfrac{1+\hat{x}}{4} + x_4 \dfrac{1-\hat{x}}{4}, \\ \partial_{\hat{x}} \mathbf{F}_y &= -y_1 \dfrac{1-\hat{y}}{4} + y_2 \dfrac{1-\hat{y}}{4} + y_3 \dfrac{1+\hat{y}}{4} - y_4 \dfrac{1+\hat{y}}{4}, \\ \partial_{\hat{y}} \mathbf{F}_y &= -y_1 \dfrac{1-\hat{x}}{4} - y_2 \dfrac{1+\hat{x}}{4} + y_3 \dfrac{1+\hat{x}}{4} + y_4 \dfrac{1-\hat{x}}{4}, \end{cases}$$

and when the cell K is a square with length Δx, Δy

$$\det\big(\partial\mathbf{F}(\hat{x},\hat{y})\big)) = \frac{\Delta x \Delta y}{4}.$$

Mass matrix

To assemble the global mass matrix, we need to compute the elementary mass matrix. Let $g, h \in \mathcal{V}_h$ be defined on K, and consider

$$I = \int_K g(x,y)h(x,y)\,dx\,dy.$$

With a $\mathbb{Q}_1$ finite element approximation, $g \circ \mathbf{F}$ and $f \circ \mathbf{F}$ can be decomposed on the basis $(N_i)_{1\leq i\leq 4}$.

$$\begin{aligned} g\circ\mathbf{F}(\hat{x},\hat{y}) &= g_1N_1(\hat{x},\hat{y}) + g_2N_2(\hat{x},\hat{y}) + g_3N_3(\hat{x},\hat{y}) + g_4N_4(\hat{x},\hat{y}),\\ h\circ\mathbf{F}(\hat{x},\hat{y}) &= h_1N_1(\hat{x},\hat{y}) + h_2N_2(\hat{x},\hat{y}) + h_3N_3(\hat{x},\hat{y}) + h_4N_4(\hat{x},\hat{y}). \end{aligned}$$

Introducing the vectors

$$G = \begin{pmatrix} g_1\\ g_2\\ g_3\\ g_4 \end{pmatrix}, \quad H = \begin{pmatrix} h_1\\ h_2\\ h_3\\ h_4 \end{pmatrix}, \quad N = \begin{pmatrix} N_1\\ N_2\\ N_3\\ N_4 \end{pmatrix},$$

the product $\big(g\circ\mathbf{F}\big)\big(h\circ\mathbf{F}\big)$ reads

$$G^T\,N^T\,N\,H.$$

Therefore one can write the expression of I as

$$I = \int_{\hat{K}} G^T\,N^T(\hat{x},\hat{y})\,N(\hat{x},\hat{y})\,H\det\big(\partial\mathbf{F}(\hat{x},\hat{y})\big),$$

which leads to

$$I = G^T\left(\frac{\Delta x\Delta y}{4}\int_{\hat{K}} N^T(\hat{x},\hat{y})\,N(\hat{x},\hat{y})\right)H.$$

The elementary mass matrix M_1 is

$$M_1 = \frac{\Delta x\Delta y}{4}\int_{\hat{K}} N^T(\hat{x},\hat{y})\,N(\hat{x},\hat{y}).$$

It is independent on the functions g, h and can be computed explicitly :

$$M_1 = \frac{\Delta x\Delta y}{36}\begin{pmatrix} 4 & 2 & 1 & 2\\ 2 & 4 & 2 & 1\\ 1 & 2 & 4 & 2\\ 2 & 1 & 2 & 4 \end{pmatrix}.$$

Stiffness matrix

Similarly for the stiffness matrix, with $g, h \in \mathcal{V}_h$ defined on K, the integral to be considered is

$$J = \int_K \nabla_{x,y}g(x,y)\cdot\nabla_{x,y}h(x,y)\,dx\,dy.$$

With a change of variable in the integral (with formula of (5.4)) this can be reformulated as

$$J = \int_{\hat{K}} \nabla_{x,y}\Big(g\circ\mathbf{F}\Big)\cdot\nabla_{x,y}\Big(h\circ\mathbf{F}\Big)\,\det\big(\partial\mathbf{F}(\hat{x},\hat{y})\big)\,d\hat{x}\,d\hat{y}.$$

In this equation the gradient $\nabla_{x,y}$ is computed with respect to the variables (x,y) in the original cell K. A change of variable so that the gradients are computed with respect to $\hat{x},\hat{y}$ leads to

$$J = \int_{\hat{K}} \nabla_{\hat{x},\hat{y}}\Big(g\circ\mathbf{F}\Big)\cdot\nabla_{\hat{x},\hat{y}}\Big(h\circ\mathbf{F}\Big)\,d\hat{x}\,d\hat{y}.$$

Now decomposing similarly to the previous case g, h on the basis $(N_i)_{1\leq i\leq 4}$ gives

$$\nabla_{\hat{x},\hat{y}}\big(g\circ\mathbf{F}\big)(\hat{x},\hat{y}) \;=\; \frac{1}{4}\begin{pmatrix} -g_1(1-\hat{y}) + g_2(1-\hat{y}) + g_3(1+\hat{y}) - g_4(1+\hat{y})\\ -g_1(1-\hat{x}) - g_2(1+\hat{x}) + g_3(1+\hat{y}) + g_4(1-\hat{x}) \end{pmatrix},$$

and the same for h. This can be written as

$$\nabla_{\hat{x},\hat{y}}\big(g\circ\mathbf{F}\big)(\hat{x},\hat{y}) = \nabla N^T\, G,$$

where ∇N is defined by

$$\nabla N = \begin{pmatrix} \partial_x N_1 & \partial_y N_1 \\ \partial_x N_2 & \partial_y N_2 \\ \partial_x N_3 & \partial_y N_3 \\ \partial_x N_4 & \partial_y N_4 \end{pmatrix}.$$

Therefore J can be computed by

$$J = G^T \left(\int_{\hat{K}} \nabla N(\hat{x},\hat{y})\, \nabla N^T(\hat{x},\hat{y}) \right) H.$$

The elementary stiffness matrix M_2 is

$$\int_{\hat{K}} \nabla N(\hat{x},\hat{y})\, \nabla N^T(\hat{x},\hat{y}),$$

which is also independent of g and h and whose explicit expression on a square mesh is

$$M_2 = \frac{1}{6}\begin{pmatrix} 4 & -1 & -2 & -1 \\ -1 & 4 & -1 & -2 \\ -2 & -1 & 4 & -1 \\ -1 & -2 & -1 & 4 \end{pmatrix}.$$

5.2 Mass-lumping method

The mass-lumping method, described for example in [124, Chapter 15], [37], or [130] for surface integrals in $\mathbb{R}^2$, consists in replacing the mass matrix by a diagonal matrix whose diagonal terms are the sum of each row of the mass matrix. In this section we recall some properties of the mass-lumping method. Let us consider a finite element mesh $\mathcal{T}_h$ on a domain Ω, $\mathcal{V}_h$ the space of continuous piecewise linear functions ($\mathbb{P}_1$) on this mesh. Let also N_h be the dimension of this finite element space. In particular one has $\mathcal{V}_h \subset H^1(\Omega)$. Let M be the mass matrix for the meshed domain, *i.e.* for each ϕ, ψ in $\mathcal{V}_h$ with coordinates Φ, Ψ on the canonical basis of $\mathcal{V}_h$ we have

$$\int_\Omega \phi\psi = \Phi^T\, M\, \Psi.$$

We also denote by m the bilinear map, $L^2(\Omega)\times L^2(\Omega) \ni (\phi,\psi) \mapsto m(\phi,\psi) = \displaystyle\int_\Omega \phi\psi$. For $\phi \in C(\Omega)$ we denote by $\mathcal{I}^1_{\mathcal{T}_h}\phi$ the piecewise linear interpolation on $\mathcal{V}_h$. For an element $\tau \in \mathcal{T}_h$ we introduce the linear map Q_τ defined on $C(\tau)$ by

$$Q_\tau(\phi) = \int_\tau \mathcal{I}^1_{\mathcal{T}_h}\phi.$$

Now we can define a linear map Q on $C(\Omega)$ by

$$Q(\phi) = \sum_{\tau\in\mathcal{T}_h} Q_\tau(\phi) = \int_\Omega \mathcal{I}^1_{\mathcal{T}_h}\phi. \tag{5.5}$$

We also define $\tilde{m} : C(\Omega)\times C(\Omega) \to \mathbb{R}$, an approximated mass bilinear map on $C(\Omega)$, by

$$\tilde{m}(\phi,\psi) = Q(\phi\psi).$$

The lumped-mass matrix $\tilde{M}$ is defined by the matrix of the bilinear map $\tilde{m}$, *i.e.* such that

$$\tilde{m}(\phi,\psi) = \Phi^T\, \tilde{M}\, \Psi.$$

One can therefore consider the approximation error between $\tilde{m}$ and m for continuous functions (for which the interpolation is defined).

Lemma 5.2.1. *The matrix $\tilde{M}$ is a diagonal matrix whose diagonal terms $\tilde{m}_{i,i}$ are defined by*

$$\tilde{m}_{i,i} = \sum_{j=1}^{N_h} m_{i,j},$$

where $m_{i,j}$ are the entries of the mass matrix M.

Proof. With the same notation as in Section 5.1.1, $(\varphi_i)_{1\leq i\leq N_h}$ and $(a_i)_{1\leq i\leq N_h}$ denote respectively the canonical basis of $\mathcal{V}_h$ with Lagrange finite element, and the set of edges of the mesh. Therefore, with $j \neq k$ we have for each $1 \leq i \leq N_h$, $\varphi_j(a_i)\varphi_k(a_i) = 0$. This means that the piecewise linear interpolation of the product $\varphi_k\varphi_j$ is uniformly equal to zero. Therefore $Q(\varphi_k\varphi_j) = 0$ and so is $\tilde{m}_{jk}$ for $j \neq k$. As a consequence, the matrix $\tilde{M}$ is diagonal. It remains to consider $\tilde{m}(\varphi_k, \varphi_k)$:

$$\tilde{m}(\varphi_k, \varphi_k) = Q(\varphi_k^2).$$

Since the functions φ_k^2 and φ_k take the value 1 on the vertex a_k and zero everywhere else, we have $Q(\varphi_k^2) = Q(\varphi_k)$. Therefore

$$\begin{aligned}\tilde{m}(\varphi_k, \varphi_k) &= Q(\varphi_k)\\ &= Q(\varphi_k\, u)\\ &= m(\varphi_k, u)\\ &= \sum_{j=1}^{N_h} m_{k,j}\end{aligned}$$

where $u : \Omega \ni x \mapsto u(x) = 1$. □

Now we can recall an error analysis result from [124, Lemma 15.1] or from the same author [37, Lemma 1] between the approximated lumped bilinear map $\tilde{m}$ and the mass bilinear map m.

Lemma 5.2.2. *Let $v, w \in \mathcal{V}_h$ and $\varepsilon_h = \tilde{m}(v, w) - m(v, w)$. Then we have*

$$\forall v, w \in \mathcal{V}_h, \quad |\varepsilon_h| \leq Ch^2 \left\|\nabla v\right\|_{L^2(\Omega)} \left\|\nabla w\right\|_{L^2(\Omega)}.$$

Proof. Since the quadrature formula (5.5) is exact for ϕ linear we have, by transformation to a fixed reference cell τ_0 and using the Bramble-Hilbert lemma [41, Theorem 4.1.3] that

$$\forall f \in W^{2,1}(\tau),\ \left|Q_\tau(f) - \int_\tau f\right| \leq Ch^2 \sum_{|\alpha|=2} \left\|D^\alpha f\right\|_{L^1(\tau)}.$$

With $v, w \in \mathcal{V}_h$ and $f = vw$ this leads to

$$\left|Q_\tau(vw) - \int_\tau vw\right| \leq Ch^2 \sum_{|\alpha|=2} \left\|D^\alpha (vw)\right\|_{L^1(\tau)}.$$

Since both v and w are linear in every τ this reads

$$\left|Q_\tau(vw) - \int_\tau vw\right| \leq Ch^2 \left\|\nabla v\right\|_{L^2(\tau)} \left\|\nabla w\right\|_{L^2(\tau)}.$$

And we can conclude with a Cauchy-Schwarz inequality that after summation on every τ

$$|\varepsilon_h(v, w)| \leq Ch^2 \left\|\nabla v\right\|_{L^2(\Omega)} \left\|\nabla w\right\|_{L^2(\Omega)}.$$

□

These two results state all the benefit from the mass-lumping method. Firstly, the resulting mass matrices are diagonal and not much complicated to compute, since it suffices to multiply the "true" mass matrix with a vector containing only the value 1. The diagonal structure of the resulting matrix may be a real advantage in view of linear system solving as well as optimization. Secondly, the additional error introduced by the lumped approximation is not large.

Remark 5.2.3. *According to Theorem 5.1.3 for $l \in \mathbb{N}$ such that $0 \leq l \leq k+1$ and $(p, q) \in \mathbb{N}^2$ with $1 \leq p \leq q \leq +\infty$ such that $W^{l,p}(\mathcal{D}) \subset C(\mathcal{D})$ there exists $c > 0$ dependent only on l, p, q such that for all $v \in W^{l,p}(\mathcal{D})$,*

$$\left\|v - \mathcal{I}_{\mathcal{T}_h}^k v\right\|_{L^q(\mathcal{D})} \leq ch^{l-\frac{d}{p}+\frac{d}{q}} |v|_{W^{l,p}(\mathcal{D})}.$$

For example, for $p = q = 2$, this means that

$$\left\|v - \mathcal{I}_{\mathcal{T}_h}^k v\right\|_{L^2(\mathcal{D})} \leq ch^l |v|_{H^l(\mathcal{D})}.$$

Therefore, with $v, w \in H^l(\mathcal{D})$ *one has*

$$\begin{aligned}\left|\int_{\mathcal{D}} vw - \int_{\mathcal{D}} \mathcal{I}^k_{\mathcal{T}_h} v\, \mathcal{I}^k_{\mathcal{T}_h} w\right| &\le \left|\int_{\mathcal{D}} \Big(v - \mathcal{I}^k_{\mathcal{T}_h} v\Big) w\right| + \left|\int_{\mathcal{D}} \Big(v - \mathcal{I}^k_{\mathcal{T}_h} v\Big) \mathcal{I}^k_{\mathcal{T}_h} v\right| \\ &\le \left\|v - \mathcal{I}^k_{\mathcal{T}_h} v\right\|_{L^2(\mathcal{D})} \|w\|_{L^2(\mathcal{D})} + \left\|w - \mathcal{I}^k_{\mathcal{T}_h} w\right\|_{L^2(\mathcal{D})} \left\|\mathcal{I}^k_{\mathcal{T}_h} v\right\|_{L^2(\mathcal{D})} \\ &\le c h^l \Big(|v|_{H^l(\mathcal{D})} \|w\|_{L^2(\mathcal{D})} + |w|_{H^l(\mathcal{D})} \left\|\mathcal{I}^k_{\mathcal{T}_h} v\right\|_{L^2(\mathcal{D})} \Big).\end{aligned}$$

Therefore with piecewise linear interpolation, i.e $k = 1$*, we have at most* $l = 2$ *which implies that the error on* $m(v, w)$ *is of order two. In this context the additional error of the mass-lumping method is of the same order as the interpolation error.*

Remark 5.2.4. *The mass-lumping method is widely used for evolution problems (see the above references). The main interest of the presentation of this method here is for further use in Section 9.2 where it gives some hints for approximation of second-order derivatives.*

Part II
Second-order derivatives and algorithms

"Une erreur originale
vaut peut-être mieux
qu'une idée banale."
F. Dostoïevski

Chapter 6

Parametric case

Contents

There are many different ways of modeling shape optimization problems which can be found in the literature [7, 21, 30, 40, 68, 86, 93, 106, 122]. Among these different formalisms the thickness parametrization of a shape we introduced in Section 2.2.1 is one of the simplest formalisms. Some formalisms can be very similar to this one (for example the SIMP method [21]). However, most of the time for such a framework, only the first-order derivatives are considered. In this chapter, we focus on the second-order derivatives of usual criteria for the thickness optimization. We first detail the computation of the derivatives. This is rather classical, even if it is hard to locate such results in the literature Then we study the structure of the second-order derivative which, up to my knowledge has not been considered yet.

The thickness modeling of shape a priori takes place in a two-dimensional framework. Therefore, we restrict ourselves in this whole chapter to the case where $d = 2$.

6.1 A model problem

We consider like in Section 3.1 the case of the *membrane model* (see Section 2.1.1), with $\Omega \subset \mathbb{R}^d$ a smooth bounded open reference domain and $d = 2$, $0 < h_m \leq h_M$, $h \in L^2(\Omega;[h_m, h_M])$ and

$$\Omega_{\mathbb{R}^3} = \{X \in \mathbb{R}^3 \,|\, X = (x, y, z) \text{ with } (x, y) \in \Omega \text{ and } 0 \leq z \leq h(x, y)\}.$$

The boundary of Ω is divided in three parts : $\partial\Omega = \Gamma \cup \Gamma_N \cup \Gamma_D$, each of them having a non-zero measure. For $f \in L^2(\Omega;\mathbb{R})$ and $g \in H^1(\Omega;\mathbb{R})$ with $g = 0$ on Γ the out-of-plane displacements of the membrane are modeled by

$$\left\{\begin{array}{rcll} -\mathrm{div}\,(h\nabla u) & = & f & \text{in } \Omega, \\ h\partial_{\mathbf{n}} u & = & g & \text{on } \Gamma_N \cup \Gamma, \\ u & = & 0 & \text{on } \Gamma_D. \end{array}\right. \tag{6.1}$$

With the Hilbert space $\mathcal{V}$ defined by

$$\mathcal{V} = \left\{u \in H^1(\Omega;\mathbb{R}) \mid u = 0 \text{ on } \Gamma_D\right\}. \tag{6.2}$$

the variational formulation for (6.1) reads

$$\left\{\begin{array}{l} \text{find } u \in \mathcal{V} \text{ such that} \\ \forall v \in \mathcal{V},\ \displaystyle\int_\Omega h\nabla u \cdot \nabla v = \int_\Omega fv + \int_{\Gamma_N} gv. \end{array}\right.$$

We consider shapes for which the thickness h takes its values in $[h_m, h_M]$. Therefore, we define the set of admissible shapes by

$$\mathcal{U}_{ad} = \{h \in L^\infty(\Omega;\mathbb{R}) \mid h_m \leq h \leq h_M\}.$$

In this section we are concerned with the second-order derivatives of the criteria introduced in Section 3.1, that are the volume, the compliance J_1 and a least square displacement criterion J_2 whose expressions are

$$V(h) = \int_\Omega h,$$

$$J_1(h) = \int_\Omega h|\nabla u|^2,$$

$$J_2(h) = \int_\Omega |u - u_0|^2,$$

where $u_0 \in L^2(\Omega;\mathbb{R})$ is a given target displacement.

Remark 6.1.1. *This study takes place in the two-dimensional case. However, with a physical sense to be detailed, we could also consider the same state equation as (6.1) with $d > 2$, $\Omega \in \mathbb{R}^d$. Then, the computation of the second-order derivatives in Section 6.2 remains the same. In the scalar case Section 6.3 (with $u \in H^1(\Omega;\mathbb{R})$) as well as in the vector case Section 6.4 (with $u \in H^1(\Omega;\mathbb{R}^d)$) the structure result for the second-order derivative would also be the same.*

6.2 Second-order derivatives

The computation of first and second-order derivatives is well-known for parametric optimization. For example, one can find different examples of design sensitivity analysis in the following references : [57, 67, 69, 70]. We recall here the computation of second-order derivatives for our parametric model (the first-order derivatives are recalled in Section 3.1), so that in the next section we will focus on the structure of the Hessian operator, which is our main contribution on this topic.

The case of the volume is straightforward since the function $h \mapsto V(h)$ is linear.

Lemma 6.2.1. *The function $h \mapsto V(h)$ is twice differentiable in $L^\infty(\Omega;\mathbb{R})$. For $k, l \in L^\infty(\Omega;\mathbb{R})$ the second-order derivative of the volume is given by*

$$V''(h;k,l) = 0.$$

Now we focus on the state equation and the criteria J_1, J_2.

Lemma 6.2.2. *Let $k \in L^\infty(\Omega;\mathbb{R})$. The function $h \mapsto u'_k(h)$ is differentiable on $L^\infty(\Omega;\mathbb{R})$. The second-order derivative of the state in the direction $l \in L^\infty(\Omega;\mathbb{R})$ is given by*

$$u''(h;k,l) = u''_{kl},$$

where u''_{kl} is the unique solution in $H^1(\Omega;\mathbb{R})$ of

$$\begin{cases} -\mathrm{div}\,(h\nabla u''_{kl}) = \mathrm{div}\,(l\nabla u'_k) + \mathrm{div}\,(k\nabla u'_l) & \text{in } \Omega, \\ u''_{kl} = 0 & \text{on } \Gamma_D, \\ h\partial_{\boldsymbol{n}} u''_{kl} = -k\partial_{\boldsymbol{n}} u'_l - l\partial_{\boldsymbol{n}} u'_k & \text{on } \Gamma_N \cup \Gamma. \end{cases}$$

Lemma 6.2.3. *The function $h \to J_1(h)$ is twice differentiable on $L^\infty(\Omega;\mathbb{R})$. The Hessian defines a bilinear symmetric positive form on $L^\infty(\Omega;\mathbb{R}) \times L^\infty(\Omega;\mathbb{R})$:*

$$J_1{}''(h;k,l) = -2\int_\Omega l(x)\nabla u(x)\cdot\nabla u'_k(x)\,dx = 2\int_\Omega h(x)\nabla u'_l(x)\cdot\nabla u'_k(x)\,dx,$$

where u'_k is the unique solution in $H^1(\Omega;\mathbb{R})$ of

$$\left\{\begin{array}{rcll} -\mathrm{div}\,(h\nabla u'_k) &=& \mathrm{div}\,(k\,\nabla u) & \Omega, \\ u'_k &=& 0 & \Gamma_D, \\ h\partial_{\boldsymbol{n}} u'_k &=& -k\,\partial_{\boldsymbol{n}} u & \Gamma_N \cup \Gamma. \end{array}\right. \tag{6.3}$$

We can define the Hessian operator H by

$$\begin{array}{rcccl} H & : & L^\infty(\Omega;\mathbb{R}) & \longrightarrow & L^1(\Omega;\mathbb{R}) \\ & & k & \longmapsto & -2\,\nabla u(x)\cdot\nabla u'_k(x). \end{array}$$

Remark 6.2.4. *The Hessian operator H can be seen as a bounded linear operator from $L^2(\Omega;\mathbb{R})$ to itself whenever $\nabla u \in L^\infty(\Omega;\mathbb{R}^d)$. Indeed, for $p, q \in L^2(\Omega;\mathbb{R})$ we have*

$$\begin{aligned}\langle Hp, q\rangle_{L^2(\Omega;\mathbb{R}),L^2(\Omega;\mathbb{R})} &= -2\int_\Omega q(x)\nabla u(x)\cdot\nabla u_p'(x)\,dx \\ &\leq 2\,\|q\|_{L^2(\Omega;\mathbb{R})}\,\|\nabla u\|_{L^\infty(\Omega;\mathbb{R}^d)}\,\left\|\nabla u_p'\right\|_{L^2(\Omega;\mathbb{R}^d)}.\end{aligned}$$

Since u_p' solves (6.3) we have, thanks to the variational formulation

$$\begin{aligned}h_m\left\|\nabla u_p'\right\|^2_{L^2(\Omega;\mathbb{R}^d)} &\leq \int_\Omega p(x)\nabla u(x)\cdot\nabla u_p'(x)\,dx \\ &\leq \|p\|_{L^2(\Omega;\mathbb{R})}\,\|\nabla u\|_{L^\infty(\Omega;\mathbb{R}^d)}\,\left\|\nabla u_p'\right\|_{L^2(\Omega;\mathbb{R}^d)}.\end{aligned}$$

Then

$$\left\|\nabla u_p'\right\|_{L^2(\Omega;\mathbb{R}^d)} \leq \frac{1}{h_m}\,\|p\|_{L^2(\Omega;\mathbb{R})}\,\|\nabla u\|_{L^\infty(\Omega;\mathbb{R}^d)}.$$

And

$$\langle Hp, q\rangle_{L^2(\Omega;\mathbb{R}),L^2(\Omega;\mathbb{R})} \leq \frac{2}{h_m}\,\|\nabla u\|^2_{L^\infty(\Omega;\mathbb{R}^d)}\,\|q\|_{L^2(\Omega;\mathbb{R})}\,\|p\|_{L^2(\Omega;\mathbb{R})}.$$

Remark 6.2.5. *The second expression of the Hessian in Lemma 6.2.3 proves that it is positive ($h \geq h_m > 0$) :*

$$\langle Hk, k\rangle_{L^1(\Omega;\mathbb{R}),L^\infty(\Omega;\mathbb{R})} = 2\int_\Omega h(x)|\nabla u_k'|^2\,dx \geq 0.$$

This notably proves that the compliance function is convex with respect to h.

Lemma 6.2.6. *The function $h \longmapsto J_2(h)$ is twice differentiable on $L^\infty(\Omega;\mathbb{R})$. Its second-order derivative is given by*

$$J_2{}''(h;k,l) = \int_\Omega 2u_k'u_l' - \int_\Omega \Big(l\nabla p\cdot\nabla u_k' + k\nabla p\cdot\nabla u_l'\Big),$$

where p is the unique solution in $H^1(\Omega;\mathbb{R})$ of

$$\left\{\begin{array}{rcll} -\mathrm{div}\,(h\nabla p) &=& 2(u-u_0) & \Omega, \\ p &=& 0 & \Gamma_D, \\ h\partial_{\boldsymbol{n}}p &=& 0 & \Gamma_N\cup\Gamma. \end{array}\right. \tag{6.4}$$

Proof. The derivation of

$$J_2{}'(h;k) = -\int_\Omega k\nabla u\cdot\nabla p$$

immediately gives

$$J_2{}''(h;k,l) = -\int_\Omega k\nabla u_l'\cdot\nabla p - \int_\Omega k\nabla u\cdot\nabla p_l'.$$

The variational formulation for u_k' gives

$$\forall\varphi\in\mathcal{V},\ \int_\Omega h\nabla u_k'\cdot\nabla\varphi = -\int_\Omega k\nabla u\cdot\nabla\varphi.$$

Then

$$\begin{aligned}-\int_\Omega k\nabla u\cdot\nabla p_l' &= \int_\Omega h\nabla u_k'\cdot\nabla p_l' \\ &= 2\int_\Omega u_l'u_k' - \int_\Omega l\nabla p\cdot\nabla u_k',\end{aligned}$$

and

$$J_2{}''(h;k,l) = \int_\Omega 2u_k'u_l' - \int_\Omega \Big(l\nabla p\cdot\nabla u_k' + k\nabla p\cdot\nabla u_l'\Big).$$

□

6.3 Scalar case - Structure of the Hessian of the compliance

Second-order optimization methods, such as the Newton method, have efficient convergence properties in the neighborhood of an optimal solution when the Hessian is regular enough. The main step of the Newton method consists in inverting this operator (see Section 1.2.2). In several examples of shape optimization, the bilinear map H defined by the shape Hessian is compact (see for example [4],[51], [60]), which makes it difficult to invert. For the boundary-value problem introduced here, we will study the structure of the second-order derivative of the compliance J_1. It will be seen that it is not compact, but also not coercive.

6.3.1 Non-compactness on $L^2(\Omega;\mathbb{R})$

Proposition 6.3.1. *Assume that $\nabla u \in L^\infty(\Omega;\mathbb{R})$. Then the second-order derivative operator H, seen as an operator on $L^2(\Omega;\mathbb{R})$, is not compact.*

The main idea of the proof is to build a sequence k_ε that converges weakly to 0 in $L^2(\Omega;\mathbb{R})$ such that $\langle Hk_\varepsilon, k_\varepsilon\rangle$ does not converge to 0. In order to prove this result, we use the framework of homogenization [5, 91]. With $\varepsilon > 0$, we pave $\mathbb{R}^d$ with ε-homotheties of the periodicity cell $Y = [0,1]^d$, such that $\mathbb{R}^d = \bigcup_{i\in\mathbb{Z}^d} Y_{i,\varepsilon}$ where $Y_{i,\varepsilon} = \varepsilon(i+Y)$. For $(x,y) \in \Omega \times Y$, let $k = k(x,y) \in C\left(\Omega; C^1_\#(Y)\right)$, where $C^1_\#(Y)$ is the space of $C^1(Y;\mathbb{R})$ functions that are Y-periodic. More generally, for $p \in \mathbb{N}$ we also introduce the spaces $L^p_\#(Y)$ that are the spaces of functions in $L^p(\Omega;\mathbb{R})$ that are Y-periodic. We consider the following boundary-value problem

$$\left\{\begin{array}{rcll} -\mathrm{div}\left(h\nabla v_\varepsilon\right) &=& \mathrm{div}\left(k\left(x,\frac{x}{\varepsilon}\right)\nabla u\right) & \Omega, \\ v_\varepsilon &=& 0 & \Gamma_D, \\ h\partial_{\mathbf{n}} v_\varepsilon &=& -k\left(x,\frac{x}{\varepsilon}\right)\partial_{\mathbf{n}} u & \Gamma_N \cup \Gamma, \end{array}\right. \tag{6.5}$$

where u is solution of (6.1). The variational formulation of this problem is

$$\left\{\begin{array}{l} \text{find } v_\varepsilon \in \mathcal{V} \text{ such that} \\ \forall \varphi \in \mathcal{V},\ \displaystyle\int_\Omega h(x)\nabla v_\varepsilon(x)\cdot\nabla\varphi(x)\,dx = -\int_\Omega k\left(x,\frac{x}{\varepsilon}\right)\nabla u(x)\cdot\nabla\varphi(x)\,dx. \end{array}\right. \tag{6.6}$$

Denoting $k_\varepsilon : x \mapsto k\left(x,\frac{x}{\varepsilon}\right)$, we have $v_\varepsilon = u'_{k_\varepsilon}$. The second-order derivative of J_1 in the direction k_ε is also

$$J_1{}''(h;k_\varepsilon,k_\varepsilon) = \langle Hk_\varepsilon, k_\varepsilon\rangle = 2\int_\Omega h(x)|\nabla v_\varepsilon(x)|^2\,dx. \tag{6.7}$$

We now recall some background on homogenization and two-scale convergence that will be needed. We will denote by $\mathcal{D}$ the space of $C^\infty(\Omega;\mathbb{R})$ functions with compact support.

Definition 6.3.2. *A sequence of functions v_ε in $L^2(\Omega;\mathbb{R})$ is said to two-scale converge to a limit $u_0(x,y)$ belonging to $L^2(\Omega\times Y;\mathbb{R})$ if, for any function $\varphi(x,y)$ in $\mathcal{D}\left(\Omega; C^\infty_\#(Y)\right)$, it satisfies*

$$\lim_{\varepsilon\to 0}\int_\Omega v_\varepsilon(x)\varphi\left(x,\frac{x}{\varepsilon}\right) = \int_\Omega \fint_Y u_0(x,y)\varphi(x,y)\,dxdy.$$

Theorem 6.3.3. *From each bounded sequence v_ε in $L^2(\Omega;\mathbb{R})$ one can extract a subsequence which two-scale converges to some $v_0(x,y) \in L^2(\Omega\times Y;\mathbb{R})$.*

Theorem 6.3.4. *Let v_ε be a bounded sequence in $H^1(\Omega;\mathbb{R})$. Then, up to a subsequence, v_ε two-scale converges to a limit $v(x) \in H^1(\Omega;\mathbb{R})$, and ∇v_ε two-scale converges to $\nabla_x v(x) + \nabla_y v_1(x,y)$, where the function $v_1(x,y)$ belongs to $L^2\left(\Omega; H^1_\#(Y)/\mathbb{R}\right)$.*

We also recall some useful results

Lemma 6.3.5. *Let $f \in L^2\left(\Omega; C_\#(Y)\right)$. We denote*

$$f_\varepsilon(x) = f\left(x,\frac{x}{\varepsilon}\right).$$

Then

$$\lim_{\varepsilon\to 0}\|f_\varepsilon\|_{L^2(\Omega;\mathbb{R})} = \|f\|_{L^2(\Omega\times Y;\mathbb{R})},$$
$$f_\varepsilon(x) \text{ two}-\text{scale converges to } f.$$

Remark 6.3.6. *The first statement of this lemma is explained and proved in [5, Lemma 1.3]. The second result is then easy to obtain with the first one.*

Theorem 6.3.7. *[5, Theorem 1.8] Let f_ε be a sequence of functions in $L^2(\Omega;\mathbb{R})$ that two-scale converges to a limit $f_0(x,y) \in L^2(\Omega\times Y;\mathbb{R})$. Assume that*

$$\lim_{\varepsilon\to 0}\|f_\varepsilon\|_{L^2(\Omega;\mathbb{R})} = \|f_0\|_{L^2(\Omega\times Y;\mathbb{R})}.$$

Then, for any sequence g_ε that two-scale converges to a limit $g_0(x,y) \in L^2(\Omega\times Y;\mathbb{R})$, we have

$$f_\varepsilon(x)g_\varepsilon(x) \underset{\varepsilon\to 0}{\longrightarrow} \int_Y f_0(x,y)g_0(x,y)\,dy \qquad \text{in } \mathcal{D}'(\Omega;\mathbb{R}).$$

Proposition 6.3.8. *Assume that* $\nabla u \in L^\infty\left(\Omega;\mathbb{R}^d\right)$ *and* $h \in L^\infty(\Omega;\mathbb{R})$. *Let* $k \in C\left(\Omega; C^1_\#(Y)\right)$ *such that*

1.
$$\forall x \in \Omega,\ \fint_Y k(x,y)\,dy = 0, \tag{6.8}$$

2.
$$\|\nabla_x u(x)\cdot\nabla_y k(x,y)\|_{L^2(\Omega\times Y;\mathbb{R})} \neq 0. \tag{6.9}$$

Then there exists $v_1 \in L^2\left(\Omega; C^1_\#(Y)/\mathbb{R}\right)$ *with* $\|\nabla_y v_1(x,y)\|_{L^2(\Omega\times Y;\mathbb{R}^d)} \neq 0$ *such that the solution* v_ε *of the problem (6.5) satisfies*

$$\left(v_\varepsilon(x) - \varepsilon v_1\left(x,\tfrac{x}{\varepsilon}\right)\right) \underset{\varepsilon\to 0}{\longrightarrow} 0 \quad \text{strongly in } H^1(\Omega;\mathbb{R}),$$
$$\left(\nabla v_\varepsilon(x) - \nabla_y v_1\left(x,\tfrac{x}{\varepsilon}\right)\right) \underset{\varepsilon\to 0}{\longrightarrow} 0 \quad \text{strongly in } L^2\left(\Omega;\mathbb{R}^d\right).$$

Proof. Since v_ε is solution of the variational formulation (6.6), and $v_\varepsilon \in \mathcal{V}$

$$\begin{aligned} h_m \|\nabla v_\varepsilon\|^2_{L^2(\Omega;\mathbb{R}^d)} &\leq \int_\Omega h(x)\nabla v_\varepsilon(x)\cdot\nabla v_\varepsilon(x)\,dx \\ &= -\int_\Omega k\left(x,\frac{x}{\varepsilon}\right)\nabla u(x)\cdot\nabla v_\varepsilon(x)\,dx \\ &\leq \|k\|_{L^\infty(\Omega\times Y;\mathbb{R})}\|\nabla u\|_{L^2(\Omega;\mathbb{R}^d)}\|\nabla v_\varepsilon\|_{L^2(\Omega;\mathbb{R}^d)}. \end{aligned}$$

Then v_ε is a bounded sequence in $H^1(\Omega;\mathbb{R})$, and according to Theorem 6.3.4, v_ε two-scale converges (up to a subsequence) to a limit $v \in H^1(\Omega;\mathbb{R})$, and ∇v_ε two-scale converges to $\nabla_x v(x) + \nabla_y v_1(x,y)$ with $v_1 \in L^2\left(\Omega; H^1_\#(Y)/\mathbb{R}\right)$. Given that $k(x,y)\nabla u(x) \in L^2\left(\Omega; C_\#(Y)^d\right)$, ($u$ solves (6.1), and k is continuous) Lemma 6.3.5 ensures that $k\left(x,\frac{x}{\varepsilon}\right)\nabla u$ two-scale converges to $k(x,y)\nabla u(x)$.

We now take in (6.6) a test function φ similar to the limit of v_ε, namely $\varphi(x) + \varepsilon\varphi_1\left(x,\frac{x}{\varepsilon}\right)$, where $\varphi(x) \in \mathcal{D}(\Omega;\mathbb{R})$ and $\varphi_1(x,y) \in \mathcal{D}\left(\Omega; C^\infty_\#(Y)\right)$. Its gradient is

$$\nabla\varphi(x) + \nabla_y\varphi_1\left(x,\frac{x}{\varepsilon}\right) + \varepsilon\nabla_x\varphi_1\left(x,\frac{x}{\varepsilon}\right).$$

We get

$$\begin{aligned} \int_\Omega h(x)\nabla v_\varepsilon(x)\cdot\left(\nabla\varphi(x) + \nabla_y\varphi_1\left(x,\frac{x}{\varepsilon}\right) + \varepsilon\nabla_x\varphi_1\left(x,\frac{x}{\varepsilon}\right)\right)dx = \\ -\int_\Omega k\left(x,\frac{x}{\varepsilon}\right)\nabla u(x)\cdot\left(\nabla\varphi(x) + \nabla_y\varphi_1\left(x,\frac{x}{\varepsilon}\right) + \varepsilon\nabla_x\varphi_1\left(x,\frac{x}{\varepsilon}\right)\right)dx. \end{aligned}$$

We first see that v_ε and φ_1 are bounded in $H^1(\Omega;\mathbb{R})$ and $H^1(\Omega\times Y;\mathbb{R})$ respectively, which means that

$$\lim_{\varepsilon\to 0}\varepsilon\int_\Omega\fint_Y h(x)\nabla v_\varepsilon(x)\cdot\nabla_x\varphi_1\left(x,\frac{x}{\varepsilon}\right)dxdy = 0,$$
$$\lim_{\varepsilon\to 0}\varepsilon\int_\Omega\fint_Y k\left(x,\frac{x}{\varepsilon}\right)\nabla u(x)\cdot\nabla_x\varphi_1\left(x,\frac{x}{\varepsilon}\right)dxdy = 0.$$

We look at $h(x)\left(\nabla\varphi(x) + \nabla_y\varphi_1\left(x,\frac{x}{\varepsilon}\right)\right)$ and $\nabla\varphi(x) + \nabla_y\varphi_1\left(x,\frac{x}{\varepsilon}\right)$ as test functions for the two-scale convergence of ∇v_ε and $k\left(x,\frac{x}{\varepsilon}\right)\nabla u(x)$. These test functions may not be regular enough, but they at least belong to $L^2\left(\Omega; C_\#(Y)^d\right)$ (h is in $L^\infty(\Omega;\mathbb{R})$) which is sufficient for the two-scale convergence theorem (see [5, Remark 1.11] for details). We can then pass to the two-scale limit and get

$$\begin{aligned} \int_\Omega\fint_Y h(x)\left(\nabla v(x) + \nabla_y v_1(x,y)\right)\cdot\left(\nabla\varphi(x) + \nabla_y\varphi_1(x,y)\right)dxdy = \\ -\int_\Omega\fint_Y k(x,y)\nabla u(x)\cdot\left(\nabla\varphi(x) + \nabla_y\varphi_1(x,y)\right)dxdy, \end{aligned}$$

Taking $\varphi_1 = 0$ we obtain

$$\int_\Omega\fint_Y h(x)\left(\nabla v(x) + \nabla_y v_1(x,y)\right)\cdot\nabla_x\varphi(x)\,dxdy = -\int_\Omega\fint_Y k(x,y)\nabla u(x)\cdot\nabla_x\varphi(x)\,dxdy.$$

Since φ does not depend on y, and v_1 is Y-periodic,

$$\int_\Omega h(x)\nabla v(x)\cdot\nabla\varphi(x)\,dx = -\int_\Omega \nabla u(x)\cdot\nabla\varphi(x)\left(\fint_Y k(x,y)\,dy\right)dx = 0.$$

With (6.8), since the Dirichlet boundary condition for v_ε on Γ_D is satisfied by the weak limit in $H^1(\Omega;\mathbb{R})$, we get

$$v = 0. \tag{6.10}$$

Taking $\varphi = 0$ yields to

$$\int_\Omega \fint_Y h(x)\left(\nabla v(x) + \nabla_y v_1(x,y)\right)\cdot\nabla_y\varphi_1(x,y)\,dxdy = -\int_\Omega \fint_Y k(x,y)\nabla u(x)\cdot\nabla_y\varphi_1(x,y)\,dxdy.$$

An easy integration by parts leads to

$$\begin{aligned} &-\operatorname{div}_y\left(h(x)\left(\nabla v(x)+\nabla_y v_1(x,y)\right)\right) = \operatorname{div}_y\left(k(x,y)\nabla u(x)\right),\\ &y\to v_1(x,y)\ Y-\text{periodic}. \end{aligned}$$

We have by (6.10), $v = 0$ and since we assumed (6.9) :

$$\begin{aligned} &-\operatorname{div}_y\left(h(x)\nabla_y v_1(x,y)\right) = \nabla u(x)\cdot\nabla_y k(x,y)\neq 0,\\ &y\to v_1(x,y)\ Y-\text{periodic}. \end{aligned} \tag{6.11}$$

As a result $v_1\neq 0$, $\|\nabla_y v_1\|_{L^2(\Omega\times Y;\mathbb{R}^d)}\neq 0$, and ∇v_ε two-scale converges to $\nabla_y v_1(x,y)$.

We now prove that

$$\nabla v_\varepsilon - \nabla_y v_1\left(x,\frac{x}{\varepsilon}\right)\underset{\varepsilon\to 0}{\longrightarrow} 0 \qquad \text{strongly in } L^2\left(\Omega;\mathbb{R}^d\right).$$

Since $v_\varepsilon\in H^1(\Omega;\mathbb{R})$ the variational formulation gives

$$\int_\Omega h(x)|\nabla v_\varepsilon(x)|^2\,dx = -\int_\Omega k\left(x,\frac{x}{\varepsilon}\right)\nabla u(x)\cdot\nabla v_\varepsilon(x)\,dx,$$

$$\begin{aligned}\int_\Omega h(x)\left(\nabla v_\varepsilon(x)-\nabla_y v_1\left(x,\frac{x}{\varepsilon}\right)\right)^2 dx = &-\int_\Omega k\left(x,\frac{x}{\varepsilon}\right)\nabla u(x)\cdot\nabla v_\varepsilon(x)\,dx + \int_\Omega h(x)\left|\nabla_y v_1\left(x,\frac{x}{\varepsilon}\right)\right|^2 dx\\ &-2\int_\Omega h(x)\nabla v_\varepsilon(x)\cdot\left(\nabla_y v_1\left(x,\frac{x}{\varepsilon}\right)\right)dx.\end{aligned}$$

With Lemma 6.3.5 we have

$$\lim_{\varepsilon\to 0}\left\|k\left(x,\frac{x}{\varepsilon}\right)\nabla u\right\|_{L^2(\Omega;\mathbb{R}^d)} = \|k(x,y)\nabla u(x)\|_{L^2(\Omega\times Y;\mathbb{R})}.$$

Then we can use Theorem 6.3.7 to conclude that with the two-scale convergence of $k\left(x,\frac{x}{\varepsilon}\right)\nabla u$ and ∇v_ε

$$\lim_{\varepsilon\to 0}\int_\Omega k\left(x,\frac{x}{\varepsilon}\right)\nabla u(x)\cdot\nabla v_\varepsilon(x)\,dx = \int_\Omega\fint_Y k(x,y)\nabla u(x)\cdot\nabla_y v_1(x,y)\,dxdy.$$

Furthermore, v_1 is regular; more precisely, $v_1\in L^2(\Omega;C^1_\#(Y))$. Indeed, $\nabla u(x)\cdot\nabla_y k(x,y)\in L^\infty(\Omega\times Y;\mathbb{R})$, and $h\in L^\infty(\Omega;\mathbb{R})$ so that for all $p\in\mathbb{N}$ and almost all $x\in\Omega$, $y\longmapsto v_1(x,y)\in W^{1,p}(Y;\mathbb{R})$. This implies that for almost all $x\in\Omega$, $y\longmapsto v_1(x,y)\in C^1(Y;\mathbb{R})$. Then $\nabla_y v_1(x,y)$ is regular enough to be considered as a test-function for the two scale-convergence. We can then pass to the two-scale limit and get

$$\begin{aligned} h_m\lim_{\varepsilon\to 0}&\left\|\nabla v_\varepsilon(x)-\nabla_y v_1\left(x,\frac{x}{\varepsilon}\right)\right\|^2_{L^2(\Omega;\mathbb{R}^d)}\leq\\ &-\int_\Omega\fint_Y k(x,y)\nabla u(x)\cdot\nabla_y v_1(x,y)\,dxdy - \int_\Omega\fint_Y h(x)\left|\nabla_y v_1(x,y)\right|^2 dxdy. \end{aligned} \tag{6.12}$$

In view of (6.11), considering v_1 as a test function for the variational formulation (Theorem 6.3.4 ensures that v_1 has the required regularity) the right-hand side of (6.12) is equal to zero, which gives the desired result. □

We are now able to prove Proposition 6.3.1.

Proof of Proposition 6.3.1. Since u is the solution of (6.1) with non-zero boundary conditions, u is not constant on Ω. Then, for $(e_i)_{[1,d]}$ a basis of $\mathbb{R}^d$, we assume that $\|\nabla u(x)\cdot e_1\|_{L^2(\Omega;\mathbb{R})}\neq 0$. Let k the function defined on $\Omega\times Y$ by $k(x,y)=\cos(\pi(y\cdot e_1))$. An easy computation proves that

$$\forall x\in\Omega,\ \fint_Y k(x,y)\,dy=0.$$

By construction, k is chosen such that

$$\nabla u(x)\cdot\nabla_y k(x,y)\neq 0 \text{ in } L^2(\Omega\times Y;\mathbb{R}). \tag{6.13}$$

We can now use Proposition 6.3.8. There exists $v_1\in L^2\left(\Omega;H^1_\#(Y)/\mathbb{R}\right)$, with $\nabla_y v_1(x,y)\neq 0$ in $L^2(\Omega\times Y;\mathbb{R})$, such that the solution v_ε of the homogenized problem (6.5) satisfies

$$\begin{aligned}\left(v_\varepsilon(x)-\varepsilon v_1\left(x,\tfrac{x}{\varepsilon}\right)\right) &\underset{\varepsilon\to 0}{\longrightarrow} \text{ strongly in } H^1(\Omega;\mathbb{R}),\\ \left(\nabla v_\varepsilon(x)-\nabla_y v_1\left(x,\tfrac{x}{\varepsilon}\right)\right) &\underset{\varepsilon\to 0}{\longrightarrow} \text{ strongly in } L^2\left(\Omega;\mathbb{R}^d\right).\end{aligned}$$

With the notation introduced in (6.7), we have proven that

$$\int_\Omega h(x)|\nabla v_\varepsilon(x)|^2\,dx=\langle Hk_\varepsilon,k_\varepsilon\rangle_{L^1(\Omega;\mathbb{R}),L^\infty(\Omega;\mathbb{R})}\longrightarrow\int_\Omega\fint_Y h(x)\left|\nabla_y v_1(x,y)\right|^2\,dxdy>0.$$

On the other hand, by Lemma 6.3.5 and Theorem 6.3.7, $k_\varepsilon\rightharpoonup 0$ weakly in $L^2(\Omega;\mathbb{R})$. Thus the second-order derivative H is not compact.

□

Remark 6.3.9. *Continuity assumptions on ∇u and k are only used at the beginning of the proof of Proposition 6.3.8 (we need $k(x,y)\in C\left(\Omega;C_\#(Y)\right)$ and $\nabla u\in L^2\left(\Omega;\mathbb{R}^d\right)$), and to get regularity on v_1 for (6.12) ($\nabla u\in L^\infty\left(\Omega;\mathbb{R}^d\right)$ and $\nabla_y k(x,y)\in L^2\left(\Omega;L^\infty(Y)^d\right)$ were needed).*

6.3.2 Non-coercivity on $L^2(\Omega;\mathbb{R})$

We now prove the non-coercivity of the Hessian of J_1. It implies that zero is in the spectrum of the operator H (see Section 6.3.3).

Proposition 6.3.10. *Assume that $\nabla u\in H^1\left(\Omega;\mathbb{R}^d\right)$. The second-order derivative operator H of the compliance J_1, is not coercive in $L^2(\Omega;\mathbb{R})$.*

In order to proves the non-coercivity, the main idea is to find a sequence $k_n\in L^2(\Omega;\mathbb{R})$ such that $\lim_{n\to\infty}\langle Hk_n,k_n\rangle=0$ with k_n that does not converge to zero in the following sense

$$\exists C>0,\ \forall n\in\mathbb{N},\ \|k_n\|_{L^2(\Omega;\mathbb{R})}>C.$$

For $\delta>0$ we pave $\mathbb{R}^d$ with δ-homotheties of the periodic cell $Y=[0,1]^d$, such that $\mathbb{R}^d=\bigcup_{i\in\mathbb{Z}^d}Y_{i,\delta}$ where $Y_{i,\delta}=\delta(i+Y)$. We denote $D_\delta=\left\{i\in\mathbb{N}^d\mid Y_{i,\delta}\cap\Omega\neq\emptyset\right\}$. There exists $M_0>0$ such that $|D_\delta|\leq\frac{M_0}{\delta^d}$. The proof uses some well-known lemmas that are recalled and proved later.

Proof. In each cell $Y_{i,\delta}$ of the paving, we introduce

$$\xi_i=\fint_{Y_{i,\delta}}\nabla u(x)\,dx=\frac{1}{|Y_{i,\delta}|}\int_{Y_{i,\delta}}\nabla u(x)\,dx.$$

In each cell we also choose $\xi_i^\perp\in\mathbb{R}^2$ such that

$$\left\|\xi_i^\perp\right\|=1,\qquad \xi_i\cdot\xi_i^\perp=0.$$

We define the function k on $[0,1]$ by

$$k(x)=\cos(\pi x), \tag{6.14}$$

and introduce a sequence $(k_n)_{n\in\mathbb{N}}\in L^\infty(\Omega;\mathbb{R})$ defined in each cell $Y_{i,\delta}$ by

$$k_n(x)=k\left(n\left(x\cdot\xi_i^\perp\right)\right).$$

The sequence k_n is in $L^2(\Omega;\mathbb{R})$, with $\|k_n\|_{L^2(\Omega;\mathbb{R})}=\frac{1}{2}|\Omega|(1+o(1))$ (for $n\delta\gg 1$). Since $u'_{k_n}\in H^1(\Omega;\mathbb{R})$ is solution to (6.3)

$$\langle Hk_n,k_n\rangle=\int_\Omega h|\nabla u'_{k_n}|^2\,dx=-\int_\Omega k_n\nabla u\cdot\nabla u'_{k_n}\,dx.$$

To establish the result, it is sufficient to prove that for all bounded sequence $\varphi_n \in H^1(\Omega;\mathbb{R})$

$$\lim_{n\to\infty}\int_\Omega k_n(x)\nabla u(x)\cdot\nabla\varphi_n(x)\,dx = 0. \tag{6.15}$$

Let φ_n be a bounded sequence in $H^1(\Omega;\mathbb{R})$. There exists $\varphi_\infty \in H^1(\Omega;\mathbb{R})$ such that up to an extraction

$$\begin{array}{ll}\varphi_n & \rightharpoonup \varphi_\infty \quad \text{weakly in } H^1(\Omega;\mathbb{R}),\\ \varphi_n & \to \varphi_\infty \quad \text{strongly in } L^2(\Omega;\mathbb{R}).\end{array}$$

Let

$$I_n = \int_\Omega k_n(x)\nabla u(x)\cdot\nabla\varphi_n(x)\,dx.$$

We have

$$\begin{aligned} I_n &= \sum_{i\in\mathbb{N}^d}\int_{Y_{i,\delta}} \mathbb{1}_\Omega(x)k_n(x)\nabla u(x)\cdot\nabla\varphi_n(x)\,dx\\ &= \underbrace{\sum_{i\in D_\delta}\int_{Y_{i,\delta}} \mathbb{1}_\Omega(x)k_n(x)\left(\nabla u(x)-\xi_i\right)\cdot\nabla\varphi_n(x)\,dx}_{I_{1_n}} + \underbrace{\sum_{i\in D_\delta}\int_{Y_{i,\delta}} \mathbb{1}_\Omega(x)k_n(x)\xi_i\cdot\nabla\varphi_n(x)\,dx}_{I_{2_n}}, \end{aligned}$$

and

$$I_{2_n} = \underbrace{\sum_{i\in D_\delta}\int_{Y_{i,\delta}} \mathbb{1}_\Omega(x)k_n(x)\xi_i\cdot\nabla\varphi_\infty(x)\,dx}_{I_{21_n}} + \underbrace{\sum_{i\in D_\delta}\int_{Y_{i,\delta}} \mathbb{1}_\Omega(x)k_n(x)\xi_i\cdot\left(\nabla\varphi_n(x)-\nabla\varphi_\infty\right)\,dx}_{I_{22_n}}.$$

Estimate for I_{1_n}. ξ_i was chosen such that it approximates ∇u. That is the main reason why I_{1_n} should go to zero. Since $\nabla u \in H^1\left(\Omega;\mathbb{R}^d\right)$, in each $Y_{i,\delta}$, according to Lemma 6.3.16 we have for $i \in D_\delta$ (we recall that $Y_{i,\delta}$ is convex)

$$\begin{aligned}\int_{Y_{i,\delta}} \mathbb{1}_\Omega(x)k_n(x)\left(\nabla u(x)-\xi_i\right)\cdot\nabla\varphi_n(x)\,dx &\le \|\mathbb{1}_\Omega k_n\|_{L^\infty(Y_{i,\delta};\mathbb{R})}\left\|\nabla u(x)-\xi_i\right\|_{L^2(Y_{i,\delta};\mathbb{R}^d)}\left\|\nabla\varphi_n\right\|_{L^2(Y_{i,\delta};\mathbb{R}^d)}\\ &\le \delta\left\|D^2u\right\|_{L^2\left(Y_{i,\delta};\mathbb{R}^{d^2}\right)}\left\|\nabla\varphi_n\right\|_{L^2(Y_{i,\delta};\mathbb{R}^d)}.\end{aligned}$$

Then, by Cauchy-Schwarz, summing over D_δ

$$|I_{1_n}| \le \delta\left\|D^2u\right\|_{L^2\left(\Omega;\mathbb{R}^{d^2}\right)}\left\|\nabla\varphi_n\right\|_{L^2(\Omega;\mathbb{R}^d)}.$$

Since φ_n converges weakly in $H^1(\Omega;\mathbb{R})$, $\nabla\varphi_n$ is bounded in $L^2(\Omega;\mathbb{R}^d)$. Therefore there exists $M_1 \in \mathbb{R}$ such that

$$|I_{1_n}| \le \delta M_1. \tag{6.16}$$

Estimate for I_{21_n}. We now prove that for each $i \in D_\delta$,

$$\lim_{n\to\infty}\int_{Y_{i,\delta}} k_n(x)\xi_i\cdot\nabla\varphi_\infty(x)\,dx = 0.$$

In this paragraph, the main reason why I_{21_n} is small, is that k_n is oscillating and its mean-value is zero. We denote

$$g(x,y) = k\left(y\cdot\xi_i^\perp\right)\xi_i\cdot\nabla\varphi_\infty(x).$$

We then pave $Y_{i,\delta}$ in the direction $\xi_i^\perp$, with periodicity $\frac{1}{n}$. We denote $Y_{i,\delta}^0$ the rescaled periodic cell for this paving. By construction, g is defined on $Y_{i,\delta}\times Y_{i,\delta}^0$, and

$$g \in L^2(Y_{i,\delta}; C_\#(Y_{i,\delta}^0)).$$

We can use the two-scale convergence theory in the framework of homogenization. In particular, it follows from Lemma 6.3.5 that

$$\lim_{n\to\infty}\int_{Y_i} k_n(x)\xi_i\cdot\nabla\varphi_\infty(x)\,dx = \int_{Y_{i,\delta}}\fint_{Y_{i,\delta}^0} g(x,y)\,dxdy.$$

Since the paving is chosen in the direction $\xi_i^\perp$, we have

$$\begin{aligned}\int_{Y_{i,\delta}}\fint_{Y_{i,\delta}^0} g(x,y)\,dxdy &= \int_{Y_{i,\delta}}\fint_{Y_{i,\delta}^0} k\left(y\cdot\xi_i^\perp\right)\xi_i\cdot\nabla\varphi_\infty(x)\,dxdy\\ &= \left(\int_{Y_{i,\delta}}\xi_i\cdot\nabla\varphi_\infty(x)\,dx\right)\left(\fint_{Y_{i,\delta}^0} k\left(y\cdot\xi_i^\perp\right)\,dy\right)\\ &= 0,\end{aligned}$$

because k and $Y^0_{i,\delta}$ are chosen such that the mean integral of $k\left(y\cdot\xi_i^\perp\right)$ vanishes on $Y^0_{i,\delta}$. Then for any fixed δ, I_{21_n} has a fixed number of terms and each of them tends to zero.

$$\forall\delta>0,\ \lim_{n\to 0} I_{21_n}=0. \tag{6.17}$$

Estimate for I_{22_n}. In this paragraph, there are two main arguments to prove that I_{22_n} goes to zero. On the one hand, k varies only in the direction perpendicular to ξ_i (the space is two-dimensional). On the other hand, φ_n is bounded in $H^1(\Omega;\mathbb{R})$. We have by integration by parts

$$\begin{aligned} I_{22_n} &= -\sum_{i\in D_\delta}\int_{Y_{i,\delta}} n\cancel{\left(\xi_i^\perp\cdot\xi_i\right)}k'(nx\cdot\xi^\perp)\left(\varphi_n(x)-\varphi_\infty\right)dx+\sum_{i\in D_\delta}\int_{\partial Y_{i,\delta}} k_n(x)\left(\varphi_n(x)-\varphi_\infty\right)\xi_i\cdot\mathbf{n}\,dx\\ &=\sum_{i\in D_\delta}\int_{\partial Y_{i,\delta}} k_n(x)\left(\varphi_n(x)-\varphi_\infty\right)\xi_i\cdot\mathbf{n}\,dx, \end{aligned}$$

and

$$\begin{aligned} |I_{22_n}| &\le \sum_{i\in D_\delta}\|\xi_i\cdot\mathbf{n}\|_{L^\infty(\partial Y_{i,\delta};\mathbb{R})}\|\varphi_n-\varphi_\infty\|_{L^2(\partial Y_{i,\delta};\mathbb{R})}\sqrt{|\partial Y_{i,\delta}|}\\ &\le \sum_{i\in D_\delta}\|\nabla u\|_{L^2(Y_{i,\delta};\mathbb{R}^d)}\|\varphi_n-\varphi_\infty\|_{L^2(\partial Y_{i,\delta};\mathbb{R})}\sqrt{\frac{|\partial Y_{i,\delta}|}{|Y_{i,\delta}|}}\\ &\le \left(\sum_{i\in D_\delta}\delta\|\varphi_n-\varphi_\infty\|^2_{L^2(\partial Y_{i,\delta};\mathbb{R})}\right)^{\frac12}\left(\sum_{i\in D_\delta}\|\nabla u\|^2_{L^2(Y_{i,\delta};\mathbb{R}^d)}\frac{|\partial Y_{i,\delta}|}{\delta|Y_{i,\delta}|}\right)^{\frac12}, \end{aligned}$$

where for the first inequality we used Lemma 6.3.16 below which gives

$$\|\xi_i\cdot\mathbf{n}\|_{L^\infty(\partial Y_{i,\delta})}\le|\xi|\le\frac{\|\nabla u\|_{L^2(Y_{i,\delta};\mathbb{R}^d)}}{\sqrt{|Y_{i,\delta}|}}.$$

Applying Lemma 6.3.15 below we have for all $\eta>0$

$$|I_{22_n}|\le\left(C(\eta)\|\varphi_n-\varphi_\infty\|^2_{L^2(\Omega;\mathbb{R})}+\delta^2\eta\|\nabla\varphi_n-\nabla\varphi_\infty\|^2_{L^2(\Omega;\mathbb{R}^d)}\right)^{\frac12}\frac{\|\nabla u\|_{L^2(\Omega;\mathbb{R}^d)}}{\delta}.$$

Since φ_n converges weakly in $H^1(\Omega;\mathbb{R})$, $\|\nabla\varphi_n-\nabla\varphi_\infty\|_{L^2(\Omega;\mathbb{R}^d)}$ is bounded. There exists M_2 such that, for all $n\in\mathbb{N}$, $\|\nabla\varphi_n-\nabla\varphi_\infty\|_{L^2(\Omega;\mathbb{R}^d)}\le M_2$. Then

$$|I_{22_n}|\le\|\nabla u\|_{L^2(\Omega;\mathbb{R}^d)}\left(\frac{C(\eta)}{\delta^2}\|\varphi_n-\varphi_\infty\|^2_{L^2(\Omega;\mathbb{R})}+\eta M_2\right)^{\frac12}. \tag{6.18}$$

Let $\varepsilon>0$. There exists $\delta>0$ such that, with (6.16),

$$|I_{1_n}|\le\frac{\varepsilon}{3}.$$

Using (6.17), since all terms $\displaystyle\int_{Y_i}k_n(x)\xi_i\cdot\nabla\varphi_\infty(x)\,dx$ go to zero, and δ is now fixed, there exists $n_0\in\mathbb{N}$ such that for all $n\ge n_0$

$$|I_{21_n}|\le\frac{\varepsilon}{3}.$$

We then choose $\eta>0$ such that

$$\eta\le\frac{1}{2\|\nabla u\|^2_{L^2(\Omega;\mathbb{R}^d)}M_2}\left(\frac{\varepsilon}{3}\right)^2.$$

Since φ_n converges to φ_∞ in $L^2(\Omega;\mathbb{R})$, there exists $n_1\in\mathbb{N}$ such that for all $n\ge n_1$,

$$\|\varphi_n-\varphi_\infty\|^2_{L^2(\Omega;\mathbb{R})}\le\frac{\delta^2}{2\|\nabla u\|^2_{L^2(\Omega;\mathbb{R}^d)}C(\eta)}\left(\frac{\varepsilon}{3}\right)^2,$$

which means

$$|I_{22_n}|\le\frac{\varepsilon}{3}.$$

Let $N=\max\{n_0,n_1\}$. For any $n\ge N$, we have

$$|I_n|\le\varepsilon,$$

which proves that

$$\lim_{n\to\infty}I_n=0.$$

Since k_n is of order $\frac12|\Omega|(1+o(1))$ in $L^2(\Omega;\mathbb{R})$, it follows that the second-order derivative operator H is not coercive. □

Remark 6.3.11. *In the proof, the sequence k_n oscillates in a direction roughly perpendicular to the gradient of u, so that* $\operatorname{div}(k_n\nabla u)$ *vanish in* $L^2(\Omega;\mathbb{R})$*. Therefore, according to Lemma 3.1.3 giving the first-order derivative of the state, we have* $\left\|u'_{k_n}\right\|_{H^1(\Omega;\mathbb{R})} \longrightarrow 0$*. This implies thanks to Lemma 6.2.3 that* $\langle Hk_n, k_n\rangle_{L^1(\Omega;\mathbb{R}),L^\infty(\Omega;\mathbb{R})} \longrightarrow 0$*.*

Remark 6.3.12. *In this section the regularity assumption concerns only ∇u which is needed to be in $H^1\left(\Omega;\mathbb{R}^d\right)$. This assumption is used in (6.16) in order to use Lemma 6.3.16. We do not really need the continuity of k.*

For convenience, we denote by $\|\varphi\|_{L^2(\partial\Omega;\mathbb{R})}$ the L^2-norm of the trace of a function $\varphi \in H^1(\Omega;\mathbb{R})$ on the boundary $\partial\Omega$ of the regular domain Ω.

Lemma 6.3.13. *Let Ω be a bounded open set with a regular boundary. Then*

$$\forall \eta > 0,\ \exists C(\eta),\ \forall \varphi \in H^1(\Omega;\mathbb{R}),\quad \|\varphi\|^2_{L^2(\partial\Omega;\mathbb{R})} \leq C(\eta)\,\|\varphi\|^2_{L^2(\Omega;\mathbb{R})} + \eta\,\|\nabla\varphi\|^2_{L^2(\Omega;\mathbb{R}^d)}\,. \tag{6.19}$$

Proof. By contradiction, let us assume that

$$\exists \eta > 0,\ \forall n \in \mathbb{N}^\star,\ \exists \varphi_n \in H^1(\Omega;\mathbb{R}),\quad \|\varphi_n\|^2_{L^2(\partial\Omega;\mathbb{R})} > n\,\|\varphi_n\|^2_{L^2(\Omega;\mathbb{R})} + \eta\,\|\nabla\varphi_n\|^2_{L^2(\Omega;\mathbb{R}^d)}\,.$$

We can renormalize the sequence φ_n, or assume that $\|\varphi_n\|_{L^2(\partial\Omega;\mathbb{R})} = 1$. Then

$$\begin{aligned}&\|\varphi_n\|^2_{L^2(\Omega;\mathbb{R})} \longrightarrow 0,\\ &\|\nabla\varphi_n\|^2_{L^2(\Omega;\mathbb{R}^d)} \quad \text{bounded.}\end{aligned}$$

As a result

$$\varphi_n \rightharpoonup 0 \text{ weakly in } H^1(\Omega;\mathbb{R}),$$

and consequently

$$\gamma\varphi_n \rightharpoonup 0 \text{ weakly in } H^{1/2}(\partial\Omega;\mathbb{R}),$$

where γ is the trace operator. The regularity of the boundary is nedded for the continuity of the trace operator γ, in order to get the weak convergence of $\gamma\varphi_n$. Using the compact embedding ([81, Theorem 3.27]) :

$$H^{1/2}(\partial\Omega;\mathbb{R}) \hookrightarrow L^2(\partial\Omega;\mathbb{R}),$$

we obtain

$$\gamma\varphi_n \to 0 \text{ strongly in } L^2(\partial\Omega;\mathbb{R}),$$

which contradicts the fact that $\|\varphi_n\|_{L^2(\partial\Omega;\mathbb{R})} = 1$. □

Remark 6.3.14. *Thanks to [58], Lemma 6.3.13 remains true if $\Omega = Y$, with $Y = [0,1]^d$. However, we could also simply invoke arguments of symmetry and extension by reflexion to get this result on a cube.*

Now we make precise how estimate (6.19) behaves if the cube $Y_{i,\delta}$ is rescaled.

Lemma 6.3.15. *We have*

$$\forall \eta > 0,\ \exists C(\eta) \in \mathbb{R}_+,\ \forall i \in \mathbb{N}^d,\ \forall \delta > 0,\ \forall \varphi \in H^1(Y_{i,\delta};\mathbb{R}),\quad \delta\,\|\varphi\|^2_{L^2(\partial Y_{i,\delta};\mathbb{R})} \leq C(\eta)\,\|\varphi\|^2_{L^2(Y_{i,\delta};\mathbb{R})} + \eta\delta^2\,\|\nabla\varphi\|^2_{L^2(Y_{i,\delta};\mathbb{R}^d)}\,.$$

Proof. All is about a change of variable. We use [7, Lemmas 6.21,6.23] with $T = \delta\,(\mathrm{Id} + t_i)$. It yields for $\varphi \in H^1(Y_{i,\delta};\mathbb{R})$

$$\begin{aligned}\int_{Y_{i,\delta}} \varphi(x)^2\,dx &= \delta^d \int_Y (\varphi\circ T(x))^2\,dx,\\ \int_{Y_{i,\delta}} |\nabla\varphi(x)|^2\,dx &= \delta^d \int_Y |\nabla\varphi\circ T(x)|^2\,dx\\ &= \delta^{d-2}\int_Y |\nabla\,(\varphi\circ T)\,(x)|^2\,dx,\\ \int_{\partial Y_{i,\delta}} \varphi(x)^2\,ds &= \int_{\partial Y} (\varphi\circ T(x))^2\delta^d\delta^{-1}\,ds.\end{aligned}$$

Then

$$\begin{aligned}\|\varphi\|^2_{L^2(Y_{i,\delta};\mathbb{R})} &= \delta^d\,\|\varphi\circ T\|^2_{L^2(Y;\mathbb{R})}\,,\\ \|\nabla\varphi\|^2_{L^2(Y_{i,\delta};\mathbb{R}^d)} &= \delta^{d-2}\,\|\nabla\,(\varphi\circ T)\|^2_{L^2(Y;\mathbb{R}^d)}\,,\\ \|\varphi\|^2_{L^2(\partial Y_{i,\delta};\mathbb{R})} &= \delta^{d-1}\,\|\varphi\circ T\|^2_{L^2(\partial Y;\mathbb{R})}\,.\end{aligned}$$

Applying Lemma 6.3.13 to $\varphi\circ T$ in Y, we conclude

$$\delta\,\|\varphi\|^2_{L^2(\partial Y_{i,\delta};\mathbb{R})} \leq C(\eta)\,\|\varphi\|^2_{L^2(Y_{i,\delta};\mathbb{R})} + \eta\delta^2\,\|\nabla\varphi\|^2_{L^2(Y_{i,\delta};\mathbb{R}^d)}\,.$$

□

We also recall the Poincaré-Wirtinger inequality.

Lemma 6.3.16. *(Poincaré-Wirtinger) Let $v \in H^1(\Omega;\mathbb{R})$. Then there exists $C > 0$ such that for any bounded, connected open set $\omega \subset \Omega$,*

$$\|v - \xi\|_{L^2(\omega;\mathbb{R})} \leq C\mathrm{diam}(\omega)\, \|\nabla v\|_{L^2(\omega;\mathbb{R}^d)}\,,$$

where

$$\xi = \fint_\omega v\, dx = \frac{1}{|\omega|}\int_\omega v(x)\, dx,$$

and

$$|\xi| \leq \frac{1}{\sqrt{|\omega|}}\, \|v\|_{L^2(\omega;\mathbb{R})}\,.$$

6.3.3 Ill-conditioning of the discretized Hessian

We prove here that the non-coercivity of the Hessian w.r.t the $L^2(\Omega;\mathbb{R})$ norm implies that it is ill-conditioned, meaning that there is a sequence of eigenvalues converging to zero. We use the same paving as previously : $\mathbb{R}^d = \bigcup_{i\in\mathbb{N}^d} Y_{i,\delta}$. On each cell $Y_{i,\delta}$, we introduce a triangulation $\mathcal{T}_{\rho,i}$ ($\rho \leq \delta$), with ρ the discretization parameter on each cell $Y_{i,\delta}$. Then, choosing $\mathcal{T}_\rho = \bigcup_{i\in\mathbb{N}^d} \mathcal{T}_{\rho,i}$ gives a triangulation of $\mathbb{R}^d$. We introduce X a finite element space (continuous piecewise polynomials of degree $\leq r$) on Ω, associated with the triangulation $\mathcal{T}_\rho$. We denote $\Pi_{\rho,r}$ the orthogonal projection from $L^2(\Omega;\mathbb{R})$ to X. For any $v \in L^2(\Omega;\mathbb{R})$, $\Pi_{\rho,r}$ belongs to X and satisfies

$$\forall w_\rho \in X,\ \int_\Omega (\Pi_{\rho,r}v)(x) w_\rho(x)\, dx = \int_\Omega v(x) w_\rho(x)\, dx. \tag{6.20}$$

We also use from [22, Chapter III, Theorem 1.2] an approximation result, under good conditions on the finite element space.

Theorem 6.3.17. *There exists a constant c independent on ρ such that for all $v \in H^1(\Omega;\mathbb{R})$*

$$\|v - \Pi_{\rho,r}v\|_{L^2(\Omega;\mathbb{R})} \leq c\rho r^{-1}\, \|v\|_{H^1(\Omega;\mathbb{R})}\,.$$

Proof. On a reference triangle $\mathcal{T}$ we get from [22, Chapter III, Theorem 1.2],

$$\|v\circ T - \Pi_{\rho,r}(v\circ T)\|_{L^2(\mathcal{T};\mathbb{R})} \leq cr^{-1}\, \|v\circ T\|_{H^1(\mathcal{T};\mathbb{R})}\,,$$

where T is the linear transformation from a reference triangle $\mathcal{T}$ to the triangle $\mathcal{T}_{\rho,i}$. The variational equality satisfied by $\Pi_{\rho,r}$ gives

$$\|v\circ T - \Pi_{\rho,r}(v\circ T)\|_{L^2(\mathcal{T};\mathbb{R})} = \big\|\big(v - \Pi_{\rho,r}(v)\big)\circ T\big\|_{L^2(\mathcal{T};\mathbb{R})}\,.$$

Using the inequalities written in the proof of Lemma 6.3.15, we can then deduce the expected result. □

Proposition 6.3.18. *Let H_ρ be the approximation of the Hessian H on X : $H_\rho = \Pi_{\rho,r}\, H\, \Pi_{\rho,r}$. Assume that H is not coercive in $L^2(\Omega;\mathbb{R})$ and that $\nabla u \in L^\infty(\Omega;\mathbb{R}^d)$. Then the smallest eigenvalue of H_ρ goes to zero when $\rho \to 0$.*

Proof. We have $H_\rho = \Pi_{\rho,r}\, H\, \Pi_{\rho,r}$. From the previous section we found a sequence k_n such that $\|k_n\|_{L^2(\Omega;\mathbb{R})} = 1$ and $\langle Hk_n, k_n\rangle_{L^2(\Omega;\mathbb{R}),L^2(\Omega;\mathbb{R})} \to 0$. Denoting $l_n = \Pi_{\rho,r}k_n$ we get

$$\begin{aligned}\langle H_\rho l_n, l_n\rangle_{X,X} &= \langle \Pi_{\rho,r}H\Pi_{\rho,r}k_n, \Pi_{\rho,r}k_n\rangle_{X,X}\\ &= \langle H\Pi_{\rho,r}k_n, \Pi_{\rho,r}k_n\rangle_{L^2(\Omega;\mathbb{R}),L^2(\Omega;\mathbb{R})}\,,\end{aligned}$$

where the first equality comes from $\Pi^2_{\rho,r} = \Pi_{\rho,r}$ and the last one from (6.20). Then

$$\begin{aligned}\langle H_\rho l_n, l_n\rangle_{X,X} =& \langle H(\Pi_{\rho,r}k_n - k_n), \Pi_{\rho,r}k_n\rangle_{L^2(\Omega;\mathbb{R}),L^2(\Omega;\mathbb{R})}\\ &+ \langle Hk_n, \Pi_{\rho,r}k_n - k_n\rangle_{L^2(\Omega;\mathbb{R}),L^2(\Omega;\mathbb{R})}\\ &+ \langle Hk_n, k_n\rangle_{L^2(\Omega;\mathbb{R}),L^2(\Omega;\mathbb{R})}\,.\end{aligned}$$

According to the definition of k_n (6.14) in the proof of Proposition 6.3.10, in each cell $Y_{i,\delta}$ we have $k_n \in H^1(Y_{i,\delta};\mathbb{R})$ with

$$k_n(x) = \cos\left(\pi\left(nx\cdot\xi^\perp\right)\right).$$

Since we assumed that each $\mathcal{T}_{\rho,i}$ is a triangulation on $Y_{i,\delta}$, one can apply Theorem 6.3.17 and get

$$\begin{aligned}\|\Pi_{\rho,r}k_n - k_n\|_{L^2(Y_{i,\delta};\mathbb{R})} &\leq C_3\rho\, \|k_n\|_{H^1(Y_{i,\delta};\mathbb{R})} \leq C_4\rho n|Y_{i,\delta}|^{\frac{1}{2}},\\ \|\Pi_{\rho,r}k_n - k_n\|_{L^2(\Omega;\mathbb{R})} &\leq C_4\rho n|\Omega|^{\frac{1}{2}}.\end{aligned}$$

We know from Remark 6.2.4 that, for $p, q \in L^2(\Omega)$,

$$\langle Hp, q\rangle_{L^2(\Omega;\mathbb{R}),L^2(\Omega;\mathbb{R})} \leq C_2 \|q\|_{L^2(\Omega;\mathbb{R})} \|p\|_{L^2(\Omega;\mathbb{R})}.$$

Therefore,

$$\langle H_\rho l_n, l_n\rangle_{X,X} \leq C_5\rho n + \langle Hk_n, k_n\rangle_{L^2(\Omega),L^2(\Omega)}.$$

Now we can denote λ_ρ the smallest eigenvalue of H_ρ. We then obtain

$$\lambda_\rho \|l_n\|_{L^2(\Omega;\mathbb{R})} \leq C_5\rho n + \langle Hk_n, k_n\rangle_{L^2(\Omega;\mathbb{R}),L^2(\Omega;\mathbb{R})}.$$

But,

$$\begin{aligned}\|\Pi_{\rho,r}k_n\|_{L^2(\Omega;\mathbb{R})} &\geq \|k_n\|_{L^2(\Omega;\mathbb{R})} - \|k_n - \Pi_n k_n\|_{L^2(\Omega;\mathbb{R})}\\ &\geq \|k_n\|_{L^2(\Omega;\mathbb{R})} - C_5\rho n,\\ \|l_n\|_{L^2(\Omega;\mathbb{R})} &\geq 1 - C_5\rho n.\end{aligned}$$

As a result

$$\lambda_\rho (1 - C_5\rho n) \leq C_5\rho n + \langle Hk_n, k_n\rangle_{L^2(\Omega;\mathbb{R}),L^2(\Omega;\mathbb{R})}.$$

Then, choosing $n = \dfrac{1}{\sqrt{\rho}}$ we obtain

$$\limsup_{\rho\to 0} \lambda_\rho \leq \langle Hk_n, k_n\rangle_{L^2(\Omega;\mathbb{R}),L^2(\Omega;\mathbb{R})} \underset{n\to\infty}{\longrightarrow} 0.$$

The last expression of the second-order derivative in Lemma 6.2.3 proves that H is positive. Then $\lambda_\rho \geq 0$. As a result

$$\lim_{\rho\to 0} \lambda_\rho = 0,$$

which ends the proof. □

Remark 6.3.19. *In this section we need a little more regularity on ∇u. The non-coercivity of the continuous operator H is given provided $\nabla u \in H^1\left(\Omega;\mathbb{R}^d\right)$. But we need in addition that $\nabla u \in L^\infty\left(\Omega;\mathbb{R}^d\right)$ to deduce the non-coercivity of the discretized operator.*

6.3.4 Kernel on $H^1(\Omega;\mathbb{R})$

It has just been seen that the Hessian is not coercive on the whole space $L^2(\Omega;\mathbb{R})$. We will now prove, when the source term f vanishes, that if we restrict the space to $H^1(\Omega;\mathbb{R}) \cap L^\infty(\Omega;\mathbb{R})$, the Hessian becomes definite.

Proposition 6.3.20. *Assume that $f = 0$ (in (6.1)), $h \in H^1(\Omega;\mathbb{R}) \cap L^\infty(\Omega;[h_m;h_M])$, $\nabla u \in L^\infty\left(\Omega;\mathbb{R}^d\right)$. We also assume that the set $\{x \in \Omega, \mid \nabla u(x) = 0\}$ is of zero Lebesgue measure. Then the Kernel of the Hessian H satisfies*

$$\left(H^1(\Omega;\mathbb{R}) \cap L^\infty(\Omega;\mathbb{R})\right) \cap \mathrm{Ker}(H) = \{0\}.$$

Proof. Let $k \in H^1(\Omega;\mathbb{R}) \cap L^\infty(\Omega;\mathbb{R})$ such that $Hk = 0$. We have

$$\begin{aligned}\langle Hk, k\rangle_{L^1(\Omega;\mathbb{R}),L^\infty(\Omega;\mathbb{R})} &= -2\int_\Omega k(x)\nabla u(x)\cdot\nabla u'_k(x)\,dx\\ &= 2\int_\Omega h(x)|\nabla u'_k(x)|^2\,dx.\end{aligned}$$

Then $\nabla u'_k(x) = 0$. The variational formulation of (6.3) gives

$$\forall\varphi \in \mathcal{V},\ \int_\Omega k(x)\nabla u(x)\cdot\nabla\varphi(x)\,dx = 0, \tag{6.21}$$

where we recall the definition of $\mathcal{V}$ by (6.2) $\mathcal{V} = \left\{u \in H^1(\Omega;\mathbb{R}) \mid u = 0 \text{ on } \Gamma_D\right\}$. For $\varphi \in \mathcal{V}$ we can then write

$$\begin{aligned}0 &= \int_\Omega k\nabla u\cdot\nabla\varphi\\ &= \int_\Omega \frac{k}{h}h\nabla u\cdot\nabla\varphi.\end{aligned}$$

An integration by parts then gives

$$0 = -\int_\Omega h\varphi\,\nabla u\cdot\nabla\left(\frac{k}{h}\right) - \int_\Omega \frac{k}{h}\varphi\,\mathrm{div}\,(h\nabla u) + \int_{\partial\Omega}\frac{k}{h}\varphi\,h\frac{\partial u}{\partial \mathbf{n}}.$$

Taking φ with compact support in Ω, with $\mathrm{div}\,(h\nabla u) = f = 0$ we get

$$h\nabla u\cdot\nabla\left(\frac{k}{h}\right) = 0.$$

Since h, k, u are in $H^1(\Omega;\mathbb{R})\cap L^\infty(\Omega;\mathbb{R})$ and $0 < h_m \le h \le h_M < +\infty$, $\varphi = \dfrac{ku}{h}$ is an admissible test function for (6.21) (since $u = 0$ on Γ_D, φ vanishes also on Γ_D). We have

$$\begin{aligned} 0 = \int_\Omega k(x)\nabla u(x)\cdot\nabla\varphi(x)\,dx &= \int_\Omega k(x)u(x)\nabla u(x)\cdot\nabla\left(\frac{k}{h}\right)(x)\,dx + \int_\Omega \frac{k^2(x)}{h(x)}|\nabla u(x)|^2\,dx \\ &= \int_\Omega \frac{k^2(x)}{h(x)}|\nabla u(x)|^2\,dx. \end{aligned}$$

Thus we can deduce that

$$k = 0 \text{ in } L^2(\Omega;\mathbb{R}).$$

□

Proposition 6.3.21. *Assume that $f = 0$ (in (6.1)), $h \in H^1(\Omega;\mathbb{R})\cap L^\infty(\Omega;[h_m;h_M])$, $\nabla u \in L^\infty\left(\Omega;\mathbb{R}^d\right)$. We also assume that the set $\{x\in\Omega,\ |\ \nabla u(x) = 0\}$ is of zero Lebesgue measure. Let*

$$\mathcal{F} = \left\{k\in H^1(\Omega;\mathbb{R})\ |\ k = 0 \ \text{ on } \ \Gamma_N\right\}.$$

Let $k_n \in H^1(\Omega;\mathbb{R})\cap L^\infty(\Omega;\mathbb{R})\cap\mathcal{F}$ bounded such that

$$\lim_{n\to 0}\langle Hk_n, k_n\rangle_{L^1(\Omega;\mathbb{R}),L^\infty(\Omega;\mathbb{R})} = 0.$$

Then

$$\lim_{n\to 0}\|k_n\|_{L^2(\Omega;\mathbb{R})} = 0.$$

Proof. The scheme of the proof is the same than the one of Proposition 6.3.20. From $\lim_{n\to 0}\langle Hk_n, k_n\rangle = 0$ we deduce that

$$\lim_{n\to\infty}\left\|\nabla u'_{k_n}\right\|_{L^2(\Omega;\mathbb{R})} = 0.$$

The variational formulation for u'_{k_n} gives

$$\forall\varphi\in\mathcal{V},\ \int_\Omega h(x)\nabla u'_{k_n}(x)\cdot\nabla\varphi(x)\,dx = -\int_\Omega k_n(x)\nabla u(x)\cdot\nabla\varphi(x)\,dx.$$

The strong convergence of $\nabla u'_{k_n}$ to zero in $L^2(\Omega,\mathbb{R}^d)$ implies that for any bounded sequence $\varphi_n\in\mathcal{V}$,

$$\lim_{n\to\infty}\int_\Omega h(x)\nabla u'_{k_n}(x)\cdot\nabla\varphi_n(x)\,dx = 0.$$

As a result, for any bounded sequence $\varphi_n\in H^1(\Omega;\mathbb{R})$,

$$\lim_{n\to\infty}\int_\Omega k_n(x)\nabla u(x)\cdot\nabla\varphi_n(x)\,dx = 0. \tag{6.22}$$

But we have, with an integration by parts

$$\begin{aligned} \int_\Omega k_n(x)\nabla u(x)\cdot\nabla\varphi_n(x)\,dx &= \int_\Omega \frac{k_n(x)}{h(x)}h(x)\nabla u(x)\cdot\nabla\varphi_n(x)\,dx \\ &= \int_{\partial\Omega}\varphi_n\frac{k_n}{h}h\frac{\partial u}{\partial\mathbf{n}} - \int_\Omega h(x)\varphi_n\nabla u\cdot\nabla\frac{k_n}{h} - \int_\Omega\frac{k_n}{h}\varphi_n\mathrm{div}\,(h\nabla u). \end{aligned}$$

Since $\mathrm{div}\,(h\nabla u) = f = 0$, $\nabla u\cdot\mathbf{n} = 0$ on Γ, and $k_n = 0$ on Γ_N we have

$$\int_\Omega k_n(x)\nabla u(x)\cdot\nabla\varphi_n(x)\,dx = \int_{\Gamma_D}\varphi_n k_n\frac{\partial u}{\partial\mathbf{n}} - \int_\Omega h(x)\varphi_n\nabla u\cdot\nabla\left(\frac{k_n}{h}\right).$$

Now we take

$$\varphi_n = \frac{k_n u}{h}.$$

Since $\varphi_n = 0$ on Γ_D, we have

$$\int_{\Gamma_D} \varphi_n k_n \frac{\partial u}{\partial \mathbf{n}} = 0,$$

and then

$$\lim_{n\to\infty} \int_\Omega h(x)\varphi_n \nabla u \cdot \nabla \left(\frac{k_n}{h}\right) = 0.$$

But

$$\lim_{n\to\infty} \int_\Omega k_n u \nabla u \cdot \nabla \left(\frac{k_n}{h}\right) = 0.$$

Since k_n, u are in $H^1(\Omega;\mathbb{R}) \cap L^\infty(\Omega;\mathbb{R})$, with k_n bounded in $H^1(\Omega;\mathbb{R})$, $h \in H^1(\Omega;\mathbb{R})$ and $0 < h_m \leq h \leq h_M < +\infty$, we can take $\varphi_n = \dfrac{k_n u}{h}$ in (6.22). We also have $k_n u$ bounded in $H^1(\Omega;\mathbb{R})$. Then,

$$\int_\Omega k_n(x)\nabla u(x)\nabla\varphi_n(x)\,dx = \int_\Omega k_n(x)u(x)\nabla u(x)\nabla\left(\frac{k_n}{h}\right)(x)\,dx \int_\Omega \frac{k_n^2(x)}{h(x)}|\nabla u(x)|^2\,dx,$$

with $\displaystyle\int_\Omega k_n(x)u(x)\nabla u(x)\nabla\left(\frac{k_n}{h}\right)(x)\,dx \to 0$. We deduce that

$$\lim_{n\to\infty} \int_\Omega \frac{k_n^2(x)}{h(x)}|\nabla u(x)|^2\,dx = 0,$$

and therefore that

$$\lim_{n\to\infty} \|k_n\|_{L^2(\Omega;\mathbb{R})} = 0.$$

□

Remark 6.3.22. *It might not be too restrictive to limit ourselves to $\mathcal{F}_c$. Indeed, it means that we are looking for directions k which do not allow the shape Ω to change in the region of $\partial\Omega$ where the loads are applied.*

6.4 Vector case - Structure of the Hessian of the compliance

We consider the same problem as in the previous section in the framework of the *planar vectorial case* of Section 2.1.3. Like previously said in Remark 6.1.1, in practice we take $d = 2$, however the computation of derivatives and their structure are the same if $d > 2$. Now the data f, g as well as the solution u are now vectors with two components, instead of being scalars. In particular $f, g \in H^1(\Omega;\mathbb{R}^d)$ with $g = 0$ on Γ. In this framework, all derivative expressions of Section 6.2 and Section 3.1 remain the same with the dot product " $\cdot$ " replaced by the contracted product " : " when necessary.

6.4.1 Non-compactness in $L^2\left(\Omega;\mathbb{R}^d\right)$

Proposition 6.4.1. *For the boundary-value problem (6.1), the second-order derivative of the compliance, H, is not compact in $L^2(\Omega;\mathbb{R})$.*

Proof. The proof is similar to the one of the scalar case. With $(e_i)_{1\leq i\leq d}$ a basis of $\mathbb{R}^d$, knowing that u is not constant, there exists $x_0 \in \Omega$ such that $\nabla u(x_0) \neq 0$. We can assume that $\nabla u(x_0)^T e_1 \neq 0$. The difference with the scalar case is that $\nabla u(x_0)^T e_1$ is a vector instead of being a scalar. We can take the same sequence k_ε defined by

$$k(x,y) = \cos(\pi\,(y\cdot e_1))$$
$$k_\varepsilon(x) = k\left(x, \frac{x}{\varepsilon}\right).$$

By construction, we still have $\left|\nabla u(x_0)^T \nabla_y k(x,y)\right| \neq 0$, in $L^2(\Omega\times Y;\mathbb{R}^d)$, seen as the norm of the vector $\nabla u(x_0)^T \nabla_y k(x,y)$ in $\mathbb{R}^d$. Then we will have the same conclusions : there exists $v_1 \in L^2\left(\Omega; \left(C^1_\#(Y)/\mathbb{R}\right)^d\right)$, such that

$$\int_\Omega h(x)|\nabla v_\varepsilon(x)|^2\,dx = \langle Hk_\varepsilon, k_\varepsilon\rangle_{L^1(\Omega;\mathbb{R}),L^\infty(\Omega;\mathbb{R})} \underset{\varepsilon\to 0}{\longrightarrow} \int_\Omega \fint_Y h(x)\left|\nabla_y v_1(x,y)\right|^2\,dxdy > 0,$$

whereas

$$k_n \underset{\varepsilon\to 0}{\longrightarrow} 0 \quad L^2(\Omega;\mathbb{R})-\text{weak}.$$

□

Remark 6.4.2. *For the non-compactness property in the vector case, the regularity assumptions on h, k and ∇u are exactly the same as in the scalar case.*

6.4.2 Non-coercivity in $L^2\left(\Omega;\mathbb{R}^d\right)$

Like in Section 6.3.2, for $\delta > 0$ we pave $\mathbb{R}^d$ with δ-homotheties of the periodicity cell $Y = [0,1]^d$, *i.e* there exists $(t_i)_{i\in\mathbb{N}^d} \in \left(\mathbb{R}^d\right)^{\mathbb{N}^d}$, such that $\mathbb{R}^d = \bigcup_{i\in\mathbb{N}^d} Y_{i,\delta}$ where $Y_{i,\delta} = \delta(t_i + Y)$. We introduce $\partial D_\delta = \left\{i \in \mathbb{N}^d \mid Y_{i,\delta} \cap \Gamma_D \neq \emptyset\right\}$. There exists $M_0 > 0$ such that $|\partial D_\delta| \geq \frac{M_0}{\delta^{d-1}}$. In the scalar case, the main argument was to find a sequence k_n whose variations were orthogonal to ∇u. In the vector case, the problem is now to find a sequence whose variations are orthogonal to all components of ∇u. When u has d components, for a domain $\Omega \subset \mathbb{R}^d$, there is a priori no vector in the kernel of ∇u. For the particular problem (6.1), with the homogeneous Dirichlet boundary condition $u = 0$, the tangential gradient vanishes on Γ_D

$$\nabla_\Gamma u = 0.$$

As a result, only the normal gradient of u is non-zero on Γ_D.

Proposition 6.4.3. *Let Ω be a bounded open set of $\mathbb{R}^d$ with a regular boundary. We also assume that Ω and u are sufficiently regular so that $u \in H^1(\Omega;\mathbb{R}^2)$, and that ∇u is Lipschitz with coefficient C_1. Then the second-order derivative of the compliance, H, is not coercive in $L^2(\Omega;\mathbb{R})$.*

Lemma 6.4.4. *We assume that u is such that ∇u is Lipschitz with coefficient C_1. We denote ∇u_j each row of ∇u. For $i \in \partial D_\delta$, since $\Gamma_D \cap Y_{i,\delta}$ is not empty, we choose $x_i \in \Gamma_D \cap Y_{i,\delta}$. We also define ξ_i a matrix of same size than ∇u, such that each row $\xi_{i,j}$ is*

$$\xi_{i,j} = \fint_{Y_{i,\delta}} \Big((\nabla u_j(x) \cdot \boldsymbol{n}(x_i))\, \boldsymbol{n}(x_i)\Big)\, dx.$$

Then

$$\forall \delta > 0,\ \forall i \in \partial D_\delta,\ \|\nabla u - \xi_i\|_{L^2(Y_{i,\delta})} \leq 2\delta C_1 \sqrt{|Y_{i,\delta}|},$$

and we have

$$|\xi_i| \leq \frac{1}{\sqrt{|Y_{i,\delta}|}} \|\nabla u\|_{L^2(Y_{i,\delta};\mathbb{R}^{2d})}.$$

Proof. For $x \in Y_{i,\delta}$ and $j \in [\![1,d]\!]$ one has

$$\nabla u_j(x) - \xi_{i,j} = \fint_{Y_{i,\delta}} \Big(\nabla u_j(x) - (\nabla u_j(y) \cdot \mathbf{n}(x_i))\, \mathbf{n}(x_i)\Big)\, dy.$$

Since $x_i \in \Gamma_D$ we have

$$\begin{aligned}\nabla u_j(x) - \xi_{i,j} &= \fint_{Y_{i,\delta}} \Big(\nabla u_j(x) - \nabla u_j(x_i) + \nabla u_j(x_i) - (\nabla u_j(y) \cdot \mathbf{n}(x_i))\, \mathbf{n}(x_i)\Big)\, dy \\ &= \fint_{Y_{i,\delta}} \Big(\nabla u_j(x) - \nabla u_j(x_i)\Big)\, dy + \fint_{Y_{i,\delta}} \Big(\nabla_\Gamma u_j(x_i) + (\nabla u_j(x_i) \cdot \mathbf{n}(x_i))\, \mathbf{n}(x_i) - (\nabla u_j(y) \cdot \mathbf{n}(x_i))\, \mathbf{n}(x_i)\Big)\, dy.\end{aligned}$$

Since x_i is on Γ_D we have $\nabla_\Gamma u_j(x_i) = 0$. We can now rewrite

$$\begin{aligned}\nabla u_j(x) - \xi_{i,j} =& \fint_{Y_{i,\delta}} \Big(\nabla u_j(x) - \nabla u_j(x_i)\Big)\, dy \\ &+ \left(\left(\nabla u_j(x_i) - \fint_{Y_{i,\delta}} \nabla u_j(y)\, dy\right) \cdot \mathbf{n}(x_i)\right) \mathbf{n}(x_i).\end{aligned}$$

Since ∇u is Lipschitz with coefficient C_1 we have

$$\begin{aligned}\left\|\nabla u_j(x_i) - \fint_{Y_{i,\delta}} \nabla u_j(y)\, dy\right\|_{L^2(Y_{i,\delta};\mathbb{R})} &\leq \delta C_1 \sqrt{|Y_{i,\delta}|}, \\ \left\|\fint_{Y_{i,\delta}} \Big(\nabla u_j(x) - \nabla u_j(x_0)\Big)\, dy\right\|_{L^2(Y_{i,\delta},\mathbb{R})} &\leq \delta C_1 \sqrt{|Y_{i,\delta}|}.\end{aligned}$$

Thus we obtain

$$\|\nabla u_j(x) - \xi_{i,j}\|_{L^2(\Omega;\mathbb{R}^2)} \leq 2\delta C_1 \sqrt{|Y_{i,\delta}|}.$$

In order to get the bound on $|\xi_{i,j}|$ we write

$$\begin{aligned}|\xi_{i,j}| &\leq \frac{1}{|Y_{i,\delta}|} \fint_{Y_{i,\delta}} |(\nabla u_j(x) \cdot \mathbf{n}(x_i))\, \mathbf{n}(x_i)|\, dx \\ &\leq \frac{1}{\sqrt{|Y_{i,\delta}|}} \|\nabla u_j\|_{L^2(Y_{i,\delta};\mathbb{R}^{2d})}.\end{aligned}$$

Doing the same for all $j \in [\![1,d]\!]$ we get the result.

□

Remark 6.4.5. *For $i \in \partial D_\delta$, the definition of ξ_i ensures that $\xi_{i,1}$ and $\xi_{i,2}$ have the same direction : $\boldsymbol{n}(x_i)$.*

Proof of Proposition 6.4.3. For $i \in \partial D_\delta$, there exists $x_i \in Y_{i,\delta} \cap \Gamma_D$. For each cell of the paving we define $\xi = (\xi_{i,1}, \xi_{i,2})^T$, where $\xi_{i,1}, \xi_{i,2} \in \mathbb{R}^2$, by

$$\begin{cases} \xi_{i,j} &= \fint_{Y_{i,\delta}} (\nabla u_j(x) \cdot \mathbf{n}(x_i))\, \mathbf{n}(x_i)\, dx, \quad \text{if } i \in \partial D_\delta, \\ \xi_{i,j} &= 0 \qquad\qquad\qquad\qquad\qquad\qquad\quad \text{otherwise.} \end{cases}$$

and ξ_i the matrix whose rows are $\xi_{i,j}$. In each cell we also choose $\xi_i^\perp$ such that

$$\left\|\xi_i^\perp\right\| = 1, \qquad \mathbf{n}(x_i) \cdot \xi_i^\perp = 0.$$

By construction we also have

$$\xi_{i,j} \cdot \xi_i^\perp = 0.$$

We take again the function k defined on $[0,1]$ by

$$k(x) = \cos(\pi x),$$

and introduce the sequence $(k_n)_{n\in\mathbb{N}} \in L^2(\Omega;\mathbb{R})^{\mathbb{N}}$ defined in each cell $Y_{i,\delta}$ by

$$k_n(x) = \begin{cases} \dfrac{1}{\sqrt{\delta}}\, k\left(nx \cdot \xi_i^\perp\right) & \text{if } i \in \partial D_\delta, \\ 0 & \text{else.} \end{cases}$$

Let $(\varphi_n)_{n\in\mathbb{N}}$ a bounded sequence in $\left(H^1\left(\Omega;\mathbb{R}^d\right)\right)^{\mathbb{N}}$. The goal is now to prove (6.15) :

$$\lim_{n\to\infty} \int_\Omega k_n(x) \nabla u(x) : \nabla \varphi_n(x)\, dx = 0.$$

There exists $\varphi_\infty \in H^1\left(\Omega;\mathbb{R}^d\right)$ such that

$$\begin{array}{rll} \varphi_n & \rightharpoonup \varphi_\infty & \text{weakly in } H^1\left(\Omega;\mathbb{R}^d\right), \\ \varphi_n & \to \varphi_\infty & \text{strongly in } L^2\left(\Omega;\mathbb{R}^d\right). \end{array}$$

Let

$$I_n = \int_\Omega k_n(x) \nabla u(x) : \nabla \varphi_n(x)\, dx.$$

We have

$$\begin{aligned} I_n &= \sum_{i\in\mathbb{N}^d} \int_{Y_{i,\delta}} \mathbb{1}_\Omega(x) k_n(x) \nabla u(x) : \nabla \varphi_n(x)\, dx \\ &= \underbrace{\sum_{i\in\partial D_\delta} \int_{Y_{i,\delta}} \mathbb{1}_\Omega(x) k_n(x) \left(\nabla u(x) - \xi_i\right) : \nabla \varphi_n(x)\, dx}_{I_{1_n}} + \underbrace{\sum_{i\in\partial D_\delta} \int_{Y_{i,\delta}} \mathbb{1}_\Omega(x) k_n(x) \xi_i : \nabla \varphi_n(x)\, dx}_{I_{2_n}}, \end{aligned}$$

and

$$I_{2_n} = \underbrace{\sum_{i\in\partial D_\delta} \int_{Y_{i,\delta}} \mathbb{1}_\Omega(x) k_n(x) \xi_i : \nabla \varphi_\infty(x)\, dx}_{I_{21_n}} + \underbrace{\sum_{i\in\partial D_\delta} \int_{Y_{i,\delta}} \mathbb{1}_\Omega(x) k_n(x) \xi_i : \left(\nabla \varphi_n(x) - \nabla \varphi_\infty\right)\, dx}_{I_{22_n}}.$$

Estimate for I_{1_n}. According to Lemma 6.4.4 we have for $i \in D_\delta$

$$\begin{aligned} \int_{Y_{i,\delta}} \mathbb{1}_\Omega(x) k_n(x) \left(\nabla u(x) - \xi_i\right) : \nabla \varphi_n(x)\, dx &\le \left\|\mathbb{1}_\Omega k_n\right\|_{L^\infty(Y_{i,\delta};\mathbb{R})} \left\|\nabla u(x) - \xi_i\right\|_{L^2(Y_{i,\delta};\mathbb{R}^{2d})} \left\|\varphi_n\right\|_{L^2(Y_{i,\delta};\mathbb{R}^d)} \\ &\le 2\delta C_1 \sqrt{|Y_{i,\delta}|}\, \left\|\varphi_n\right\|_{L^2(Y_{i,\delta};\mathbb{R}^{2d})}. \end{aligned}$$

Then summing over D_δ

$$|I_{1_n}| \le 2\delta C_1 \sqrt{|\Omega|}\, \left\|\varphi_n\right\|_{L^2(\Omega;\mathbb{R}^d)}.$$

Since φ_n is bounded in $L^2\left(\Omega;\mathbb{R}^d\right)$ (φ_n has a strong limit in $L^2\left(\Omega;\mathbb{R}^d\right)$), there exists $M_1 \in \mathbb{R}$ such that

$$|I_{1_n}| \le \delta M_1. \tag{6.23}$$

Estimate for I_{21_n}. This is exactly the same as in Proposition 6.3.10. We have

$$\forall \delta > 0, \ \lim_{n\to\infty} I_{21_n} = 0. \tag{6.24}$$

Estimate for I_{22_n}. This is as well exactly the same as in Proposition 6.3.10. We just have to take into account that $\|k_n\|_{L^\infty(Y_{i,\delta})} = \dfrac{1}{\sqrt{\delta}}$. With the same integration by parts (and ${\xi_i}^T \xi^\perp = 0$)

$$I_{22_n} = \sum_{i\in\partial D_\delta} \int_{\partial Y_{i,\delta}} k_n(x)\,(\varphi_n(x) - \varphi_\infty)\,\xi_i \mathbf{n}\,dx,$$

and

$$\begin{aligned}
|I_{22_n}| &\le \sum_{i\in\partial D_\delta} \|k_n\|_{L^\infty(\partial Y_{i,\delta};\mathbb{R})} \|\xi_i\mathbf{n}\|_{L^\infty(\partial Y_{i,\delta};\mathbb{R}^d)} \|\varphi_n - \varphi_\infty\|_{L^2(\partial Y_{i,\delta};\mathbb{R}^d)} \sqrt{|\partial Y_{i,\delta}|} \\
&\le \sum_{i\in\partial D_\delta} \frac{1}{\sqrt{\delta}} \|\nabla u\|_{L^2(Y_{i,\delta};\mathbb{R}^{2d})} \|\varphi_n - \varphi_\infty\|_{L^2(\partial Y_{i,\delta};\mathbb{R}^d)} \sqrt{\frac{|\partial Y_{i,\delta}|}{|Y_{i,\delta}|}} \\
&\le \left(\sum_{i\in\partial D_\delta} \delta \|\varphi_n - \varphi_\infty\|^2_{L^2(\partial Y_{i,\delta};\mathbb{R}^d)}\right)^{\frac{1}{2}} \left(\sum_{i\in\partial D_\delta} \|\nabla u\|^2_{L^2(Y_{i,\delta};\mathbb{R}^{2d})} \frac{|\partial Y_{i,\delta}|}{\delta^2 |Y_{i,\delta}|}\right)^{\frac{1}{2}},
\end{aligned}$$

$$|I_{22_n}| \le \|\nabla u\|_{L^2(\Omega;\mathbb{R}^{2d})} \sqrt{\frac{1}{\delta}} \left(\frac{C(\eta)}{\delta^2} \|\varphi_n - \varphi_\infty\|^2_{L^2(\Omega;\mathbb{R}^d)} + \eta M_3\right)^{\frac{1}{2}}. \tag{6.25}$$

Now we take $\varepsilon > 0$. We choose $\delta > 0$ such that

$$|I_{1_n}| \le \frac{\varepsilon}{3}.$$

With 6.24, there exists $n_1 \in \mathbb{N}$ such that

$$|I_{2_n}| \le \frac{\varepsilon}{3}.$$

Now we take $\eta > 0$ such that

$$\eta \le \left(\frac{\varepsilon}{3}\right)^2 \frac{\delta}{2\|\nabla u\|^2_{L^2(\Omega;\mathbb{R}^{2d})} M_3},$$

and n_2 such that for all $n \ge n_2$

$$\|\varphi_n - \varphi_\infty\|^2_{L^2(\Omega;\mathbb{R}^d)} \le \left(\frac{\varepsilon}{3}\right)^2 \frac{\delta^3}{2\|\nabla u\|^2_{L^2(\Omega;\mathbb{R}^{2d})} C(\eta)}.$$

Taking $n_3 \ge \max\{n_1, n_2\}$ we have for all $n \ge n_3$

$$I_n \le \varepsilon,$$

which proves that

$$\lim_{n\to\varepsilon} I_n = 0.$$

Now it remains to check that k_n does not go to zero in $L^2(\Omega;\mathbb{R})$. We have

$$\begin{aligned}
\|k_n\|^2_{L^2(\Omega;\mathbb{R})} &= \sum_{i\in\partial D_\delta} \int_{Y_{i,\delta}} \frac{1}{\delta} \left|k\left(nx\cdot\xi_i^\perp\right)\right|^2 dx \\
&= \sum_{i\in\partial D_\delta} \delta^{d-1} \\
&\ge \frac{M_0}{\delta^{d-1}} \delta^{d-1}.
\end{aligned}$$

Then we have

$$\begin{aligned}
&\lim_{n\to\infty} \langle H k_n, k_n\rangle = 0, \\
&\|k_n\|_{L^2(\Omega;\mathbb{R})} \nrightarrow 0,
\end{aligned}$$

which concludes the proof.

□

Remark 6.4.6. *For the non-coercivity in the vectorial case, the regularity assumptions concern ∇u and the regularity of $\partial\Omega$. But these assumptions are somehow stronger than in the scalar case.*

6.4.3 Ill-conditioning of the discretized Hessian

In the vector case, the discretization of the second-order derivative operator H is not coercive. This is exactly the same result and the same proof as in the scalar case. We keep the same notation, with a paving $\mathbb{R}^d = \bigcup_{i\in\mathbb{N}^d} Y_{i,\delta}$, and X a finite element space where ρ is the parameter of the discretization, such that the triangulation of $\mathbb{R}^d$ is the union of triangulations of all $Y_{i,\delta}$ $(\rho \leq \delta)$.

Proposition 6.4.7. *Let H_ρ be the approximation of the Hessian H on X. Assume that H is not coercive in $L^2(\Omega;\mathbb{R})$ and that $\nabla u \in L^\infty\left(\Omega;\mathbb{R}^{2d}\right)$. Then the smallest eigenvalue of H_ρ goes to zero when $\rho \to 0$.*

Chapter 7

Geometric case

Contents

The theoretical background for computing shape derivative was introduced in Chapter 3 where there is also a brief review on previous works on second-order shape derivatives. The computation of first-order shape derivatives for usual criterion was also detailed there. Now we are concerned with second-order shape derivatives for the same criteria. We also saw that there are a priori two usual frameworks for shape differentiability, namely the speed method and the displacement field method. The shape derivatives in these two frameworks are the same at first-order, but differ at the second-order. However their structures remain the same. Indeed for a given criterion they are entirely determined by a linear map l_1 and a bilinear map l_2 (see also Remark 3.2.23). In this section we detail some second-order shape derivatives for the speed method, which allows one to obtain the two maps l_1 and l_2 and then to switch to the displacement field framework if wanted.

We will keep here the notation of Section 3.2 and recall that, for $k \in \mathbb{N}$, $\mathcal{O}_k$ is the set of bounded open sets of $\mathbb{R}^d$ that are of class C^k. In the perspective of optimization, we take a priori the set of admissible shape $\mathcal{U}_{ad}$ as

$$\mathcal{U}_{ad} = \mathcal{O}_3.$$

The third-order regularity for the shape ($\Omega \in \mathcal{O}_3$) is assumed for computing second-order shape derivatives. With this notation, when we aim at minimizing a shape functional J, the problem reads

$$\min_{\Omega \in \mathcal{U}_{ad}} J(\Omega). \tag{7.1}$$

In this chapter we first detail some calculations of shape derivatives. The first two criterion (volume and pde-free example) are already known and we recall them for the sake of completeness. For the compliance and least square criteria, the main work here is to detail the calculations of the second-order derivatives. These are rather classical, and we could already find the results in the literature (see the works of Dambrine for example). In the case of the compliance we also give the derivatives in a fully general case (in Section 7.3) which doesn't seem to have been done yet. Then we also compute the second-order derivatives of the thickness criterion. This criterion has already been introduced for shape optimization [83], but the calculation of its second-order derivative is new. Finally, we are concerned with the compliance criterion in a specific load case. We focus on properties of its second-order derivatives that appear not to have been explored yet.

7.1 First and second-order shape derivatives

7.1.1 Volume

Let $\Omega \in \mathcal{O}_3$. We recall that the volume of the shape Ω is given by

$$V(\Omega) = \int_\Omega 1.$$

The volume is shape-differentiable, and the shape derivative in the direction $\theta \in W^{1,\infty}\left(\mathbb{R}^d;\mathbb{R}^d\right)$ is given by

$$V'(\Omega;\theta) = \int_{\partial\Omega} (\theta\cdot\mathbf{n}).$$

Lemma 7.1.1. *With $\Omega \in \mathcal{O}_3$, the volume is twice shape-differentiable. The second-order shape derivative in the directions $\theta,\xi \in C^{2,\infty}\left(\mathbb{R}^d;\mathbb{R}^d\right)$ is given by*

$$V''(\Omega;\theta,\xi) = \int_{\partial\Omega} \mathcal{H}\,(\theta\cdot\boldsymbol{n})\,(\xi\cdot\boldsymbol{n}) + \int_{\partial\Omega} Z_{\theta,\xi},$$

where $\mathcal{H}$ is the mean curvature of the boundary and $Z_{\theta,\xi}$ is defined by

$$Z_{\theta,\xi} = \left((\xi_\Gamma\cdot\nabla_\Gamma)\,\boldsymbol{n}\right)\cdot\theta_\Gamma - \nabla_\Gamma(\theta\cdot\boldsymbol{n})\cdot\xi_\Gamma - \nabla_\Gamma(\xi\cdot\boldsymbol{n})\cdot\theta_\Gamma.$$

Remark 7.1.2. *For the first-order shape-differentiability, the volume Ω needs only to have a Lipschitz boundary (see Proposition 3.3.1). But for the second-order shape-differentiability, Theorem 3.2.15 that details the structure of the derivative, is written for shapes that are at least in $\mathcal{O}_3$. It is not surprising to require more regularity on the shape for the second-order derivative : the derivation of the first-order shape derivative V'_θ implies indeed to derive the outer normal to the shape. However, the requirement $\Omega \in \mathcal{O}_3$ in Theorem 3.2.15 is not assumed for the existence of the second-order derivative, but for the structure result.*

In order to obtain the formula for the second-order shape derivative, it suffices to differentiate the first-order shape derivative and to use (3.14).

For the volume criterion and $v,w \in C^1(\partial\Omega;\mathbb{R}^d)$ the linear and bilinear maps l_1 and l_2 that characterize the shape derivatives are given by

$$l_1(v) = \int_{\partial\Omega} v,$$
$$l_2(v,w) = \int_{\partial\Omega} \mathcal{H}\,v\,w.$$

7.1.2 PDE-free example

The case of an objective shape-dependent function E of the form

$$E(\Omega) = \int_\Omega f(x)\,dx, \tag{7.2}$$

was also introduced in Section 3.3.1. When the shape as well as the function $f : \mathbb{R}^d \mapsto \mathbb{R}$ are sufficiently regular, the function E is twice shape-differentiable.

Lemma 7.1.3. *With $\Omega \in \mathcal{O}_3$, the function E defined by (7.2) is twice shape-differentiable. The second-order shape derivative in the directions $\theta,\xi \in C^{2,\infty}\left(\mathbb{R}^d;\mathbb{R}^d\right)$ is given by*

$$E''(\Omega;\theta,\xi) = \int_{\partial\Omega} \left(\mathcal{H}f + \partial_{\boldsymbol{n}} f\right)(\theta\cdot\boldsymbol{n})\,(\xi\cdot\boldsymbol{n}) + \int_{\partial\Omega} Z_{\theta,\xi},$$

where $\mathcal{H}$ is the mean curvature of the boundary and $Z_{\theta,\xi}$ is defined by

$$Z_{\theta,\xi} = \left((\xi_\Gamma\cdot\nabla_\Gamma)\,\boldsymbol{n}\right)\cdot\theta_\Gamma - \nabla_\Gamma(\theta\cdot\boldsymbol{n})\cdot\xi_\Gamma - \nabla_\Gamma(\xi\cdot\boldsymbol{n})\cdot\theta_\Gamma.$$

7.1.3 Compliance

In the framework of the *scalar thermal model* (see Section 2.1.4), we recall that the compliance is

$$J_1(\Omega) = \int_\Omega fu + \int_{\Gamma_N} gu = \int_\Omega |\nabla u|^2,$$

where $u : \Omega \to \mathbb{R}$ is solution to

$$\left\{ \begin{array}{rcll} -\Delta u & = & f & \text{in } \Omega, \\ \partial_{\mathbf{n}} u\, \mathbf{n} & = & g & \text{on } \Gamma_N \cup \Gamma, \\ u & = & 0 & \text{on } \Gamma_D, \end{array} \right. \tag{7.3}$$

with $f \in L^2(\Omega;\mathbb{R})$, $g \in H^2(\Omega;\mathbb{R})$. The first-order derivative of the compliance reads

$$\begin{aligned} J_1{}'(\Omega;\theta) &= 2\int_{\Gamma_N\cup\Gamma} (\theta\cdot\mathbf{n})\left(\partial_{\mathbf{n}}(gu) + fu + \mathcal{H}gu\right) \\ &\quad + \int_{\Gamma_D} |\nabla u|^2 (\theta\cdot\mathbf{n}) - \int_{\Gamma_N\cup\Gamma} |\nabla u|^2 (\theta\cdot\mathbf{n}), \end{aligned} \tag{7.4}$$

where $\mathcal{H}$ is the mean curvature. In order to get the first-order shape-differentiability, we assumed in Section 3.3, that $f \in H^1(\Omega;\mathbb{R})$, $g \in H^2(\Omega;\mathbb{R})$ so that by elliptic regularity, $u \in H^2(\Omega;\mathbb{R})$. We also had to assume that Ω was a smooth bounded open set, typically $\Omega \in \mathcal{O}_3$ in accordance with the assumptions of Theorem 3.2.15. At first we consider the case where there is no other restriction on an admissible shape than being in $\mathcal{O}_3$. Thus, for the optimization problem, the set of admissible shape is

$$\mathcal{U}_{ad} = \mathcal{O}_3.$$

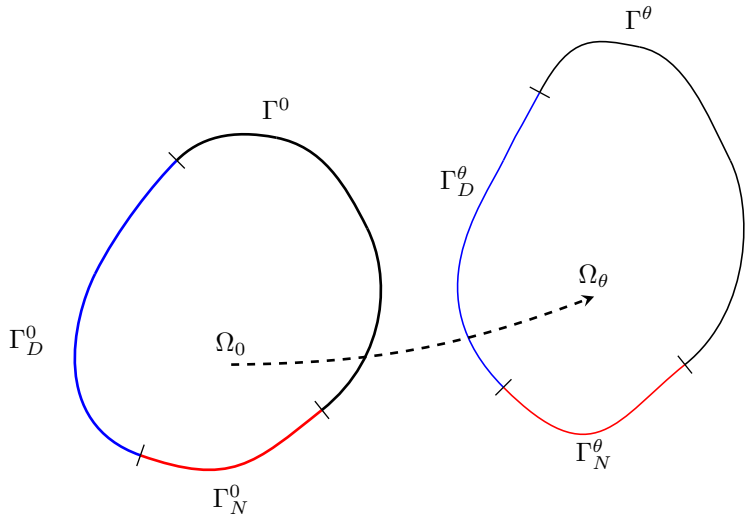


Figure 7.1: Variation of a shape Ω_0 and its boundary trough the diffeomorphism $(\mathrm{Id}+\theta)$, when the set of admissible shapes is $\mathcal{O}_3$.

In order to compute the second-order shape derivative of the compliance, we need two geometric results we introduce here.

Lemma 7.1.4. *Let $p, q, r \in C^1\left(\mathbb{R}^d;\mathbb{R}^d\right)$. One has*

$$\nabla\left(p\cdot q\right)\cdot r = \left(r\cdot\nabla\right)p\cdot q + \left(r\cdot\nabla\right)q\cdot p.$$

Proof. It is only a writing game

$$\begin{aligned} \nabla\left(p\cdot q\right)\cdot r &= \sum_{j=1}^d r_j \sum_{i=1}^d q_i\partial_j p_i + \sum_{j=1}^d r_j \sum_{i=1}^d p_i\partial_j q_i \\ &= \sum_{i=1}^d q_i\left(\sum_{j=1}^d r_j\partial_j p_i\right) + \sum_{i=1}^d p_i\left(\sum_{j=1}^d r_j\partial_j q_i\right), \\ &= \left(r\cdot\nabla\right)p\cdot q + \left(r\cdot\nabla\right)q\cdot p. \end{aligned}$$

□

Lemma 7.1.5. *Let $\theta, \xi \in C^{1,\infty}(\Omega;\mathbb{R}^d)$, and $\boldsymbol{n}$ the outer unit normal to Ω. Then*

$$\left(\xi\cdot\nabla\right)\theta\cdot\boldsymbol{n} - \left(\xi\cdot\boldsymbol{n}\right)\partial_{\boldsymbol{n}}(\theta\cdot\boldsymbol{n}) = \left(\xi_\Gamma\cdot\nabla\right)\theta\cdot\boldsymbol{n},$$

and

$$\left(\xi_\Gamma\cdot\nabla\right)\theta\cdot\boldsymbol{n} = \xi_\Gamma\cdot\nabla_\Gamma\left(\theta\cdot\boldsymbol{n}\right) - \left(\left(\xi_\Gamma\cdot\nabla_\Gamma\right)\boldsymbol{n}\right)\cdot\theta_\Gamma.$$

Proof. The decomposition of ξ in its tangential and normal components gives

$$(\xi \cdot \nabla)\,\theta \cdot \mathbf{n} = (\xi_\Gamma \cdot \nabla)\,\theta \cdot \mathbf{n} + (\xi \cdot \mathbf{n})\,(\mathbf{n} \cdot \nabla)\,\theta \cdot \mathbf{n}.$$

Given that $\|\mathbf{n}\| = 1$, we have $(\mathbf{n} \cdot \nabla)\,\mathbf{n} = 0$. Then, with Lemma 7.1.4

$$\begin{aligned}(\mathbf{n} \cdot \nabla)\,\theta \cdot \mathbf{n} &= \nabla\,(\theta \cdot \mathbf{n}) \cdot \mathbf{n} - (\mathbf{n} \cdot \nabla)\,\mathbf{n} \cdot \theta \\ &= \nabla\,(\theta \cdot \mathbf{n}) \cdot \mathbf{n},\end{aligned}$$

and

$$(\xi \cdot \nabla)\,\theta \cdot \mathbf{n} = (\xi_\Gamma \cdot \nabla)\,\theta \cdot \mathbf{n} + (\xi \cdot \mathbf{n})\,\partial_{\mathbf{n}}(\theta \cdot \mathbf{n}).$$

On the other hand,

$$\begin{aligned}(\xi_\Gamma \cdot \nabla)\,\theta \cdot \mathbf{n} &= \nabla\,(\theta \cdot \mathbf{n}) \cdot \xi_\Gamma - (\xi_\Gamma \cdot \nabla)\,\mathbf{n} \cdot \theta, \\ (\xi_\Gamma \cdot \nabla)\,\mathbf{n} \cdot \mathbf{n} &= \nabla\,(\mathbf{n} \cdot \mathbf{n}) \cdot \xi_\Gamma - (\xi_\Gamma \cdot \nabla)\,\mathbf{n} \cdot \mathbf{n} = 0.\end{aligned}$$

Therefore

$$(\xi_\Gamma \cdot \nabla)\,\theta \cdot \mathbf{n} = \nabla\,(\theta \cdot \mathbf{n}) \cdot \xi_\Gamma - (\xi_\Gamma \cdot \nabla_\Gamma)\,\mathbf{n} \cdot \theta_\Gamma.$$

□

Now we can compute the second-order shape derivative of the compliance.

Proposition 7.1.6. *Let $\Omega \in \mathcal{O}_3$, $f \in H^1(\Omega;\mathbb{R})$ and $g \in H^2(\Omega;\mathbb{R})$ such that by elliptic regularity, $u \in H^2(\Omega;\mathbb{R})$. Then the compliance J_1 is twice shape-differentiable. For $\theta, \xi \in C^{2,\infty}\left(\Omega;\mathbb{R}^d\right)$, let*

$$Z_{\theta,\xi} = \left(\left(\xi_\Gamma \cdot \nabla_\Gamma\right)\boldsymbol{n}\right) \cdot \theta_\Gamma - \nabla_\Gamma\,(\theta \cdot \boldsymbol{n}) \cdot \xi_\Gamma - \nabla_\Gamma\,(\xi \cdot \boldsymbol{n}) \cdot \theta_\Gamma.$$

The second-order shape derivative in the directions θ, ξ writes

$$\begin{aligned}J_1{}''(\Omega;\theta,\xi) =& -2\int_{\Gamma_N\cup\Gamma} (\theta\cdot\boldsymbol{n})\,\nabla u'_\xi\cdot\nabla u + 2\int_{\Gamma_D}(\theta\cdot\boldsymbol{n})\,\nabla u'_\xi\cdot\nabla u \\ &+2\int_{\Gamma_N\cup\Gamma}(\theta\cdot\boldsymbol{n})\,u'_\xi(f+\mathcal{H}g+\partial_{\boldsymbol{n}}g)+2\int_{\Gamma_N}(\theta\cdot\boldsymbol{n})\,g\nabla_\Gamma u\cdot\nabla_\Gamma(\xi\cdot\boldsymbol{n}) \\ &+2\int_{\Gamma_N\cup\Gamma}(\theta\cdot\boldsymbol{n})\,(\xi\cdot\boldsymbol{n})\Big(\mathcal{H}\big(fu+\mathcal{H}gu+\partial_{\boldsymbol{n}}(gu)\big)+\partial_{\boldsymbol{n}}\big(fu+\mathcal{H}gu+\partial_{\boldsymbol{n}}(gu)\big)\Big) \\ &+2\int_{\Gamma_N}(\theta\cdot\boldsymbol{n})\,(\xi\cdot\boldsymbol{n})\,g\Big(\partial_{\boldsymbol{n}}g-\partial^2_{\boldsymbol{nn}}u\Big)+2\int_{\Gamma_N}gu\,\nabla_\Gamma(\theta\cdot\boldsymbol{n})\cdot\nabla_\Gamma(\xi\cdot\boldsymbol{n}) \\ &-\int_{\Gamma_N\cup\Gamma}(\theta\cdot\boldsymbol{n})\,(\xi\cdot\boldsymbol{n})\Big(\partial_{\boldsymbol{n}}|\nabla u|^2+\mathcal{H}|\nabla u|^2\Big)+\int_{\Gamma_D}(\theta\cdot\boldsymbol{n})\,(\xi\cdot\boldsymbol{n})\Big(\partial_{\boldsymbol{n}}|\nabla u|^2+\mathcal{H}|\nabla u|^2\Big) \\ &+2\int_{\Gamma_N\cup\Gamma}Z_{\theta,\xi}\Big(\partial_{\boldsymbol{n}}(gu)+fu+\mathcal{H}gu\Big)+\int_{\Gamma_D}Z_{\theta,\xi}|\nabla u|^2-\int_{\Gamma_N\cup\Gamma}Z_{\theta,\xi}|\nabla u|^2,\end{aligned}$$

where u'_θ (and similarly for u'_ξ) is defined by

$$\left\{\begin{array}{rcll} -\Delta u'_\theta &=& 0 & \Omega, \\ u'_\theta &=& -(\theta\cdot\nabla)\,u & \Gamma_D, \\ \partial_{\boldsymbol{n}}u'_\theta &=& (\theta\cdot\boldsymbol{n})\Big(\partial_{\boldsymbol{n}}g-\partial^2_{\boldsymbol{nn}}u\Big)+\nabla u\cdot\nabla_\Gamma(\theta\cdot\boldsymbol{n}) & \Gamma_N, \\ \partial_{\boldsymbol{n}}u'_\theta &=& -(\theta\cdot\boldsymbol{n})\,\partial^2_{\boldsymbol{nn}}u+\nabla u\cdot\nabla_\Gamma(\theta\cdot\boldsymbol{n}) & \Gamma. \end{array}\right. \tag{7.5}$$

Theorem 3.2.15 ensures that J_1'' is symmetric. One can check this property in reformulating the expression.

Proposition 7.1.7. *Let $\theta, \xi \in C^{2,\infty}\left(\Omega;\mathbb{R}^d\right)$ and*

$$Z_{\theta,\xi} = \left(\left(\xi_\Gamma \cdot \nabla_\Gamma\right)\boldsymbol{n}\right) \cdot \theta_\Gamma - \nabla_\Gamma\,(\theta \cdot \boldsymbol{n}) \cdot \xi_\Gamma - \nabla_\Gamma\,(\xi \cdot \boldsymbol{n}) \cdot \theta_\Gamma.$$

The second-order shape derivative of the compliance is symmetric on $C^{2,\infty}\left(\mathbb{R}^d;\mathbb{R}^d\right)\times C^{2,\infty}\left(\mathbb{R}^d;\mathbb{R}^d\right)$. It can be rewritten in a symmetric expression :

$$\begin{aligned}J_1{}''(\Omega;\theta,\xi) =& 2\int_\Omega \nabla u'_\xi\cdot\nabla u'_\theta+2\int_{\Gamma_N}gu\,\nabla_\Gamma(\theta\cdot\boldsymbol{n})\cdot\nabla_\Gamma(\xi\cdot\boldsymbol{n})-2\int_{\Gamma_D}\left(u'_\xi\partial_{\boldsymbol{n}}u'_\theta+u'_\theta\partial_{\boldsymbol{n}}u'_\xi\right) \\ &+2\int_{\Gamma_N\cup\Gamma}(\theta\cdot\boldsymbol{n})\,(\xi\cdot\boldsymbol{n})\Big(\mathcal{H}\big(fu+\mathcal{H}gu+\partial_{\boldsymbol{n}}(gu)\big)+\partial_{\boldsymbol{n}}\big(fu+\mathcal{H}gu+\partial_{\boldsymbol{n}}(gu)\big)\Big) \\ &-\int_{\Gamma_N\cup\Gamma}(\theta\cdot\boldsymbol{n})\,(\xi\cdot\boldsymbol{n})\Big(\partial_{\boldsymbol{n}}|\nabla u|^2+\mathcal{H}|\nabla u|^2\Big)+\int_{\Gamma_D}(\theta\cdot\boldsymbol{n})\,(\xi\cdot\boldsymbol{n})\Big(\partial_{\boldsymbol{n}}|\nabla u|^2+\mathcal{H}|\nabla u|^2\Big) \\ &+2\int_{\Gamma_N\cup\Gamma}Z_{\theta,\xi}\Big(\partial_{\boldsymbol{n}}(gu)+fu+\mathcal{H}gu\Big)+\int_{\Gamma_D}Z_{\theta,\xi}|\nabla u|^2-\int_{\Gamma_N\cup\Gamma}Z_{\theta,\xi}|\nabla u|^2.\end{aligned}$$

In order to ease the reading, these propositions are formally proven here in the case where $f = 0$, $g = 0$ on Γ and when the parts Γ_D and Γ_N of the boundary cannot be modified. Thus, with a reference domain Ω_0, whose boundary is divided like $\partial\Omega_0 = \Gamma^0 \cup \Gamma_D^0 \cup \Gamma_N^0$ the set of admissible shapes is restricted to

$$\mathcal{U}_{ad} := \left\{\Omega \in \mathcal{O}_3 \mid \Gamma_D = \Gamma_D^0 \text{ and } \Gamma_N = \Gamma_N^0\right\}. \tag{7.6}$$

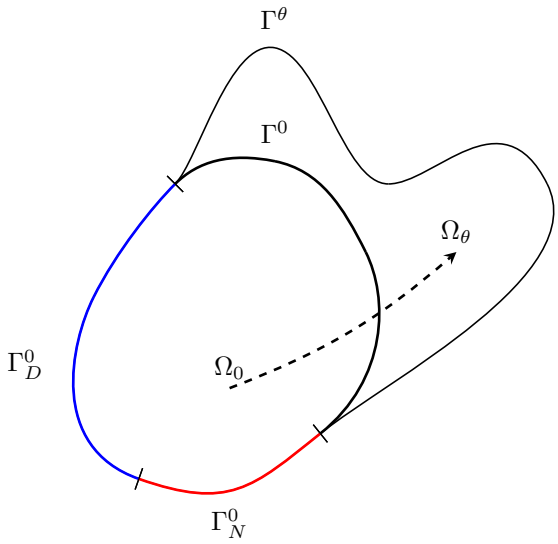


Figure 7.2: Variation of a shape Ω_0 and its boundary trough the diffeomorphism $(\mathrm{Id} + \theta)$, when the set of admissible shapes is restricted to $\mathcal{U}_{ad}$ defined by (7.6).

We also define the space $\mathcal{F}$ by

$$\mathcal{F} = \left\{\theta \in C^{2,\infty}\left(\mathbb{R}^d;\mathbb{R}^d\right) \mid \theta = 0 \text{ on } \Gamma_D \cup \Gamma_N\right\}. \tag{7.7}$$

For the more general case, the computations are given in Section 7.3.

Proposition 7.1.8. *Let $\Omega \in \mathcal{O}_3$ and f, g with the same regularity as in Proposition 7.1.6. We also assume that $g = 0$ on Γ. For $\theta, \xi \in \mathcal{F}$, the shape derivatives of the compliance in the directions θ, ξ read*

$$J_1{}'(\Omega;\theta) = -\int_\Gamma (\theta\cdot\boldsymbol{n})\,|\nabla u|^2,$$
$$J_1{}''(\Omega;\theta,\xi) = -2\int_\Gamma (\theta\cdot\boldsymbol{n})\,\nabla u'_\xi\cdot\nabla u - \int_\Gamma (\theta\cdot\boldsymbol{n})\,(\xi\cdot\boldsymbol{n})\left(\partial_{\boldsymbol{n}}|\nabla u|^2 + \mathcal{H}|\nabla u|^2\right) - \int_\Gamma Z_{\theta,\xi}|\nabla u|^2,$$

where

$$\left\{\begin{array}{rcll} -\Delta u'_\theta &=& 0 & \Omega,\\ u'_\theta &=& 0 & \Gamma_D,\\ \partial_{\boldsymbol{n}} u'_\theta &=& 0 & \Gamma_N,\\ \partial_{\boldsymbol{n}} u'_\theta &=& -\mathrm{div}_\Gamma\left((\theta\cdot\boldsymbol{n})\nabla_\Gamma u\right) & \Gamma, \end{array}\right. \tag{7.8}$$

and

$$Z_{\theta,\xi} = \left(\left(\xi_\Gamma\cdot\nabla_\Gamma\right)\boldsymbol{n}\right)\cdot\theta_\Gamma - \nabla_\Gamma(\theta\cdot\boldsymbol{n})\cdot\xi_\Gamma - \nabla_\Gamma(\xi\cdot\boldsymbol{n})\cdot\theta_\Gamma.$$

Proof. For the first-order shape derivative, we simply use the fact that $\theta = 0$ on $\Gamma_D \cup \Gamma_N$ for (7.4) :

$$J_1{}'(\Omega;\theta) = -\int_\Gamma |\nabla u|^2\,(\theta\cdot\mathbf{n}).$$

According to Proposition 3.3.23, the state u as well as the normal $\mathbf{n}$ are shape-differentiable, and therefore we can differentiate the above expression. This leads to

$$\left(J_1{}'(\Omega;\theta)\right)'(\Omega;\xi) = -2\int_\Gamma (\theta\cdot\mathbf{n})\,\nabla u'_\xi\cdot\nabla u - \int_\Gamma \left(\theta\cdot\mathbf{n}'_\xi\right)|\nabla u|^2$$
$$-\int_\Gamma (\xi\cdot\mathbf{n})\left(\partial_{\mathbf{n}}\left(|\nabla u|^2\,(\theta\cdot\mathbf{n})\right) + \mathcal{H}\,(\theta\cdot\mathbf{n})\,|\nabla u|^2\right).$$

Now we write the derivative of the outer normal $\mathbf{n}$, which exists by Lemma 3.3.22 since $\Omega \in \mathcal{O}_3$

$$\mathbf{n}'_\xi = -\nabla_\Gamma (\xi \cdot \mathbf{n}).$$

As a result

$$\begin{aligned}\Big(J_1{}'(\Omega;\theta)\Big)'(\Omega;\xi) = &-2\int_\Gamma (\theta\cdot\mathbf{n})\,\nabla u'_\xi\cdot\nabla u\\ &+\int_\Gamma |\nabla u|^2\nabla_\Gamma(\xi\cdot\mathbf{n})\cdot\theta-(\xi\cdot\mathbf{n})\,|\nabla u|^2\,\partial_{\mathbf{n}}\big(\theta\cdot\mathbf{n}\big)\\ &-\int_\Gamma (\xi\cdot\mathbf{n})\,(\theta\cdot\mathbf{n})\left(\partial_{\mathbf{n}}|\nabla u|^2+\mathcal{H}|\nabla u|^2\right).\end{aligned}$$

Now with (3.14)

$$\begin{aligned}J_1{}''(\Omega;\theta,\xi) =&\Big(J_1{}'(\Omega;\theta)\Big)'(\Omega;\xi)-J_1{}'(\Omega;(\xi\cdot\nabla)\,\theta)\\ =&-2\int_\Gamma(\theta\cdot\mathbf{n})\,\nabla u'_\xi\cdot\nabla u\\ &-\int_\Gamma(\xi\cdot\mathbf{n})\,(\theta\cdot\mathbf{n})\left(\partial_{\mathbf{n}}|\nabla u|^2+\mathcal{H}|\nabla u|^2\right)\\ &+\int_\Gamma|\nabla u|^2\Big(\nabla_\Gamma(\xi\cdot\mathbf{n})\cdot\theta-(\xi\cdot\mathbf{n})\,\partial_{\mathbf{n}}(\theta\cdot\mathbf{n})+\big(\xi\cdot\nabla\big)\theta\cdot\mathbf{n}\Big).\end{aligned}$$

But, with Lemma 7.1.5 and

$$Z_{\theta,\xi} = \big(\,(\xi_\Gamma\cdot\nabla_\Gamma)\,\mathbf{n}\big)\cdot\theta_\Gamma-\nabla_\Gamma(\theta\cdot\mathbf{n})\cdot\xi_\Gamma-\nabla_\Gamma(\xi\cdot\mathbf{n})\cdot\theta_\Gamma,$$

we get

$$\begin{aligned}J_1{}''(\Omega;\theta,\xi) &= \Big(J_1{}'(\Omega;\theta)\Big)'(\Omega;\xi)-J_1{}'(\Omega;(\xi\cdot\nabla)\,\theta)\\ &=-2\int_\Gamma(\theta\cdot\mathbf{n})\,\nabla u'_\xi\cdot\nabla u-\int_\Gamma(\xi\cdot\mathbf{n})\,(\theta\cdot\mathbf{n})\left(\partial_{\mathbf{n}}|\nabla u|^2+\mathcal{H}|\nabla u|^2\right)+\int_\Gamma Z_{\theta,\xi}|\nabla u|^2.\end{aligned}$$

□

Remark 7.1.9. *We can notice that the form of the second-order shape derivative corresponds to the structure we expect from Theorem 3.2.15. The second-order shape derivative is composed of a bilinear map l_2 and a linear map l_1 defined on $\mathcal{F}\times\mathcal{F}$ and $\mathcal{F}$ respectively, that we give here :*

$$l_2(\Omega;\theta,\xi) = -2\int_\Gamma(\theta\cdot\boldsymbol{n})\,\nabla u'_\xi\cdot\nabla u-\int_\Gamma(\xi\cdot\boldsymbol{n})\,(\theta\cdot\boldsymbol{n})\left(\partial_{\boldsymbol{n}}|\nabla u|^2+\mathcal{H}|\nabla u|^2\right), \tag{7.9}$$

$$l_1(\Omega;Z) = -\int_\Gamma Z|\nabla u|^2, \tag{7.10}$$

where l_1 is exactly the linear map of the first-order shape derivative. It just remains to check that l_2 is symmetric.

Now we consider the symmetry of the second-order shape derivative. The structure theorem ensures that structurally it is symmetric (Theorem 3.2.15) but, with the previous expression, it is not obvious.

Proposition 7.1.10. *When the state u is regular enough and $\Delta u = 0$ on $\partial\Omega$, the bilinear map l_2 defined in Remark 7.1.9 is symmetric on $\mathcal{F}$, under assumption (7.6). It writes*

$$l_2(\Omega;\theta,\xi) = 2\int_\Omega\nabla u'_\theta\cdot\nabla u'_\xi-\int_\Gamma(\xi\cdot\boldsymbol{n})\,(\theta\cdot\boldsymbol{n})\,\big(\partial_{\boldsymbol{n}}|\nabla u|^2+\mathcal{H}|\nabla u|^2\big).$$

Proof. The second component of l_2 in (7.9) is obviously symmetric. Now let us consider for $\theta,\xi\in\mathcal{F}$,

$$l(\Omega;\theta,\xi) = \int_\Gamma(\theta\cdot\mathbf{n})\,\nabla u'_\xi\cdot\nabla u.$$

Since $\partial_{\mathbf{n}}u = 0$ on Γ we have

$$l(\Omega;\theta,\xi) = \int_\Gamma(\theta\cdot\mathbf{n})\,\nabla_\Gamma u'_\xi\cdot\nabla_\Gamma u.$$

Since

$$\int_{\partial\Omega} \mathrm{div}_\Gamma\Big((\theta\cdot\mathbf{n})\, u'_\xi \nabla_\Gamma u\Big) = 0,$$

and $\theta\cdot\mathbf{n} = 0$ on $\Gamma_D \cup \Gamma_N$, we obtain

$$\int_{\Gamma} \mathrm{div}_\Gamma\Big((\theta\cdot\mathbf{n})\, u'_\xi \nabla_\Gamma u\Big) = 0,$$

and

$$\int_\Gamma (\theta\cdot\mathbf{n})\, \nabla_\Gamma u'_\xi\cdot\nabla_\Gamma u = -\int_\Gamma u'_\xi\, \mathrm{div}_\Gamma\Big((\theta\cdot\mathbf{n})\,\nabla_\Gamma u\Big).$$

On the other hand, since $\partial_{\mathbf{n}} u = g = 0$ on Γ we get

$$\mathrm{div}_\Gamma\Big((\theta\cdot\mathbf{n})\,\nabla_\Gamma u\Big) = (\theta\cdot\mathbf{n})\,\Delta_\Gamma u + \nabla_\Gamma(\theta\cdot\mathbf{n})\cdot\nabla_\Gamma u.$$

Since

$$\Delta_\Gamma u = \Delta u - \mathcal{H}\partial_{\mathbf{n}} u - \partial^2_{\mathbf{nn}} u,$$

we finally get

$$\begin{aligned}
\int_\Gamma (\theta\cdot\mathbf{n})\,\nabla_\Gamma u'_\xi\cdot\nabla_\Gamma u &= -\int_\Gamma (\theta\cdot\mathbf{n})\; u'_\xi\Delta_\Gamma u - \int_\Gamma u'_\xi\,\nabla_\Gamma(\theta\cdot\mathbf{n})\cdot\nabla_\Gamma u\\
&= -\int_\Gamma (\theta\cdot\mathbf{n})\; u'_\xi\Delta u + \int_\Gamma u'_\xi\Big((\theta\cdot\mathbf{n})\;\partial^2_{\mathbf{nn}} u - \nabla_\Gamma(\theta\cdot\mathbf{n})\cdot\nabla_\Gamma u\Big)\\
&= -\int_\Gamma (\theta\cdot\mathbf{n})\; u'_\xi\Delta u - \int_\Gamma u'_\xi\,\partial_{\mathbf{n}} u'_\theta.
\end{aligned}$$

Since $\Delta u = 0$ on $\partial\Omega$ and $\Delta u'_\theta = 0$ we obtain by integration by parts

$$\begin{aligned}
l(\Omega;\theta,\xi) &= -\int_\Omega \mathrm{div}\,(\nabla u'_\theta)\, u'_\xi - \int_\Omega \nabla u'_\theta\cdot\nabla u'_\xi\\
&= -\int_\Omega \nabla u'_\theta\cdot\nabla u'_\xi.
\end{aligned}$$

Therefore the bilinear map l_2 is symmetric. □

7.1.4 Least square criterion

Let $\alpha \geq 2$ and $u_0 \in L^\alpha(\mathbb{R}^d;\mathbb{R}^d)$. Let us consider the *elastic model* (see Section 2.1.5) and the following criterion, referred to as the least square displacement criterion

$$J_2(\Omega) = \left(\int_\Omega k(x)|u(x) - u_0(x)|^\alpha\, dx\right)^{\frac{1}{\alpha}}, \tag{7.11}$$

where $g \in H^1(\Omega;\mathbb{R}^d)$ satisfies $g = 0$ on Γ and $u \in H^2(\Omega;\mathbb{R}^d)$ is solution to

$$\left\{\begin{array}{rcll} -\mathrm{div}\,(A\varepsilon(u)) &=& 0 & \text{in } \Omega,\\ A\varepsilon(u)\mathbf{n} &=& g & \text{on } \Gamma\cup\Gamma_N,\\ u &=& 0 & \text{on } \Gamma_D,\end{array}\right. \tag{7.12}$$

where the Hooke's tensor is defined on every symmetric matrix ζ by the Lamé coefficients λ, μ, and

$$A\zeta = 2\mu\zeta + \lambda\mathrm{Tr}(\zeta),$$

and where the linearized deformation tensor $\varepsilon(u)$ is defined by

$$\varepsilon(u) = \frac{1}{2}\left(\nabla u + \nabla u^T\right).$$

Remark 7.1.11. *Here we consider the mechanical equation (7.12) in the case of linear elasticity for an isotropic material. This is quite different from (7.3) since Hooke's tensor is not proportional to the identity, and since the gradient operator is replaced by its symmetrization $\varepsilon(u)$.*

Remark 7.1.12. *This criterion is usually used in the literature with $\alpha = 2$, so that it is called a least square criterion. However, we can define it for other values of $\alpha \geq 2$, and we detail the shape derivatives here in this more general case.*

The H^2 regularity is given by elliptic regularity since g is in $H^2(\Omega;\mathbb{R})$. In order to ease the calculations, we introduce

$$j_2(\Omega) = \int_\Omega k(x)|u(x) - u_0(x)|^\alpha. \tag{7.13}$$

Similarly to the previous section we also assume that only the part Γ of the boundary $\partial\Omega$ may evolve. This implies to consider the set of admissible shapes

$$\mathcal{U}_{ad} := \{\Omega \in \mathcal{O}_3 \mid \Gamma_D \cup \Gamma_N \subset \partial\Omega\}. \tag{7.14}$$

Similarly to the previous section we also introduce

$$\mathcal{F} = \left\{\theta \in C^{2,\infty}(\mathbb{R}^d;\mathbb{R}^d) \mid \theta = 0 \text{ on } \Gamma_D \cup \Gamma\right\}.$$

Lemma 7.1.13. *The state u is twice shape-differentiable.*

Proof. Since g is in $H^2(\Omega;\mathbb{R}^d)$, by elliptic regularity, u is in $H^3(\Omega;\mathbb{R}^d)$. The regularity of the shape Ω ensures then that u is twice shape-differentiable (see also Remark 3.3.25). □

Remark 7.1.14. *The shape differentiability of u in Remark 3.3.25 is a priori only considered in the framework of the* scalar thermal model. *However the same arguments work for the* elastic model.

Proposition 7.1.15. *Let $k \geq 3$, and $\Omega \in \mathcal{O}_k$. The function j_2 defined by (7.13) is twice shape-differentiable with respect to Ω. For $\theta, \xi \in \mathcal{F}$, we have*

$$j_2'(\Omega;\theta) = \alpha \int_\Omega k|u-u_0|^{\alpha-2}(u-u_0)u_\theta' + \int_\Gamma (\theta\cdot\boldsymbol{n})\, k|u-u_0|^\alpha, \tag{7.15}$$

and

$$\begin{aligned} j_2''(\Omega;\theta,\xi) = &\int_\Gamma (\xi\cdot\boldsymbol{n})\, A\varepsilon(u_\theta') : \varepsilon(p) + (\theta\cdot\boldsymbol{n})\, A\varepsilon(u_\xi') : \varepsilon(p) + \alpha(\alpha-1)\int_\Omega k|u-u_0|^{\alpha-2}u_\theta' u_\xi' \\ &+ \alpha\int_\Gamma k|u-u_0|^{\alpha-2}(u-u_0)\Big((\theta\cdot\boldsymbol{n})\,u_\xi' + (\xi\cdot\boldsymbol{n})\,u_\theta'\Big) \\ &+ \int_\Gamma (\theta\cdot\boldsymbol{n})\,(\xi\cdot\boldsymbol{n})\left(\partial_{\boldsymbol{n}}\mathscr{L} + \mathcal{H}\mathcal{L}\right) + \int_\Gamma \mathscr{L} Z_{\theta,\xi}, \end{aligned} \tag{7.16}$$

where

$$\mathscr{L} = A\varepsilon(u)\varepsilon(p) + k|u-u_0|^\alpha,$$

$$Z_{\theta,\xi} = \big((\xi_\Gamma\cdot\nabla_\Gamma)\,\boldsymbol{n}\big)\cdot\theta_\Gamma - \nabla_\Gamma(\theta\cdot\boldsymbol{n})\cdot\xi_\Gamma - \nabla_\Gamma(\xi\cdot\boldsymbol{n})\cdot\theta_\Gamma,$$

and p is the solution to

$$\left\{\begin{array}{rcll} -\mathrm{div}\,(A\varepsilon(p)) &=& -\alpha k|u-u_0|^{\alpha-2}(u-u_0) & \text{in } \Omega, \\ A\varepsilon(p)\boldsymbol{n} &=& 0 & \text{on } \Gamma_N\cup\Gamma, \\ p &=& 0 & \text{on } \Gamma_D. \end{array}\right. \tag{7.17}$$

Proof. The state u is twice shape-differentiable. Therefore j_2 is shape-differentiable. The first-order derivative is straightforward. Now, for $\theta, \xi \in \mathcal{F}$,

$$\begin{aligned} (j_2')'(\Omega;\theta,\xi) = &\alpha\int_\Omega k\,u_\theta'\Big((\alpha-2)|u-u_0|^{\alpha-4}(u-u_0)^2u_\xi' + |u-u_0|^{\alpha-2}u_\xi'\Big) \\ &+ \alpha\int_\Omega k\,|u-u_0|^{\alpha-2}(u-u_0)(u_\theta')_\xi' + \alpha\int_\Gamma (\xi\cdot\mathbf{n})\,k\,|u-u_0|^{\alpha-2}(u-u_0)u_\theta' \\ &+ \int_\Gamma \left(\theta\cdot\mathbf{n}_\xi'\right)k\,|u-u_0|^\alpha + \int_\Gamma (\theta\cdot\mathbf{n})\,k\,|u-u_0|^{\alpha-2}(u-u_0)u_\xi' \\ &+ \int_\Gamma (\xi\cdot\mathbf{n})\left(\partial_{\mathbf{n}}\Big((\theta\cdot n)\,k\,|u-u_0|^\alpha\Big) + \mathcal{H}\,(\theta\cdot\mathbf{n})\,k\,|u-u_0|^\alpha\right) \\ \\ = &\alpha(\alpha-1)\int_\Omega k\,|u-u_0|^{\alpha-2}u_\theta'u_\xi' + \alpha\int_\Omega k\,|u-u_0|^{\alpha-2}(u-u_0)(u_\theta')_\xi' \\ &+ \alpha\int_\Gamma k\,|u-u_0|^{\alpha-2}(u-u_0)\Big((\xi\cdot\mathbf{n})\,u_\theta' + (\theta\cdot\mathbf{n})\,u_\xi'\Big) \\ &- \int_\Gamma (\theta\cdot\nabla_\Gamma(\xi\cdot\mathbf{n}))\,k\,|u-u_0|^\alpha \\ &+ \int_\Gamma (\xi\cdot\mathbf{n})\left(\partial_{\mathbf{n}}\Big((\theta\cdot\mathbf{n})\,k\,|u-u_0|^\alpha\Big) + \mathcal{H}\,(\theta\cdot\mathbf{n})\,k\,|u-u_0|^\alpha\right), \end{aligned}$$

which can be rewritten as

$$\begin{aligned}(j_2')'(\Omega;\theta,\xi) = \alpha(\alpha-1)\int_\Omega k\,|u-u_0|^{\alpha-2}u_\theta' u_\xi' + \alpha\int_\Omega k\,|u-u_0|^{\alpha-2}(u-u_0)(u_\theta')_\xi'\\ + \alpha\int_\Gamma k\,|u-u_0|^{\alpha-2}(u-u_0)\Big(\,(\xi\cdot\mathbf{n})\,u_\theta' + (\theta\cdot\mathbf{n})\,u_\xi'\Big)\\ + \int_\Gamma (\theta\cdot\mathbf{n})\,(\xi\cdot\mathbf{n})\left(\partial_{\mathbf{n}}\Big(k\,|u-u_0|^\alpha\Big) + \mathcal{H}k\,|u-u_0|^\alpha\right)\\ + \int_\Gamma k\,|u-u_0|^\alpha\Big(\,(\xi\cdot\mathbf{n})\,\partial_{\mathbf{n}}(\theta\cdot n) - \Big(\theta\cdot\nabla_\Gamma\,(\xi\cdot\mathbf{n})\,\Big)\Big).\end{aligned}$$

The second-order shape derivative obtained from the derivative of the first-order shape derivative by (see also (3.14))

$$j_2''(\Omega;\theta,\xi) = (j_2')'(\Omega;\theta,\xi) - j_2'(\Omega;(\xi\cdot\nabla)\,\theta),$$

and we also have

$$j_2'(\Omega;(\xi\cdot\nabla)\,\theta) = \int_\Omega \alpha k\,|u-u_0|^{\alpha-2}(u-u_0)u'_{(\xi\cdot\nabla)\theta} + \int_\Gamma \big(\,(\xi\cdot\nabla)\,\theta\cdot\mathbf{n}\big)k\,|u-u_0|^\alpha. \tag{7.18}$$

Therefore, we get

$$\begin{aligned}j_2''(\Omega;\theta,\xi) = \alpha(\alpha-1)\int_\Omega k\,|u-u_0|^{\alpha-2}u_\theta' u_\xi' + \alpha\int_\Omega k\,|u-u_0|^{\alpha-2}(u-u_0)(u_\theta')_\xi'\\ + \alpha\int_\Gamma k\,|u-u_0|^{\alpha-2}(u-u_0)\Big(\,(\xi\cdot\mathbf{n})\,u_\theta' + (\theta\cdot\mathbf{n})\,u_\xi'\Big)\\ - \int_\Omega \alpha k\,|u-u_0|^{\alpha-2}(u-u_0)u'_{(\xi\cdot\nabla)\theta}\\ + \int_\Gamma (\theta\cdot\mathbf{n})\,(\xi\cdot\mathbf{n})\left(\partial_{\mathbf{n}}\Big(k\,|u-u_0|^\alpha\Big) + \mathcal{H}k\,|u-u_0|^\alpha\right)\\ + \int_\Gamma k\,|u-u_0|^\alpha\Big(\,(\xi\cdot\mathbf{n})\,\partial_{\mathbf{n}}(\theta\cdot n) - \Big(\theta\cdot\nabla_\Gamma\,(\xi\cdot\mathbf{n})\,\Big) - \big(\,(\xi\cdot\nabla)\,\theta\cdot\mathbf{n}\big)\Big).\end{aligned}$$

Since p is the solution to (7.17), we have

$$\begin{aligned}\alpha\int_\Omega k\,|u-u_0|^{\alpha-2}(u-u_0)(u_\theta')_\xi' &= \int_\Omega \operatorname{div}\big(A\varepsilon(p)\big)(u_\theta')_\xi'\\ &= -\int_\Omega A\varepsilon(p):\varepsilon\big((u_\theta')_\xi'\big) + \int_\Gamma A\varepsilon(p)\mathbf{n}\cdot(u_\theta')_\xi'.\end{aligned}$$

With $(u_\theta')_\xi'$ and $A\big(\varepsilon(p)\big)\mathbf{n}$ vanishing respectively on Γ_D and $\Gamma_N\cup\Gamma$ we get

$$\alpha\int_\Omega k\,|u-u_0|^{\alpha-2}(u-u_0)(u_\theta')_\xi' = \int_\Omega \operatorname{div}\Big(A\varepsilon\big((u_\theta')_\xi'\big)\Big)p - \int_\Gamma A\varepsilon\big((u_\theta')_\xi'\big)\mathbf{n}\cdot p.$$

With the help of Lemma 7.1.16 we get an expression of $\alpha\displaystyle\int_\Omega k\,|u-u_0|^{\alpha-2}(u-u_0)(u_\theta')_\xi'$ which leads to

$$\begin{aligned}j_2''(\Omega;\theta,\xi) = \int_\Gamma (\xi\cdot\mathbf{n})\,A\varepsilon(u_\theta'):\varepsilon(p) + (\theta\cdot\mathbf{n})\,A\varepsilon(u_\xi'):\varepsilon(p)\\ + \int_\Gamma (\xi\cdot\mathbf{n})\,(\theta\cdot\mathbf{n})\left(\partial_{\mathbf{n}}\Big(A\varepsilon(u):\varepsilon(p)\Big) + \mathcal{H}A\varepsilon(u):\varepsilon(p)\right)\\ + \int_\Gamma A\varepsilon(u):\varepsilon(p)\Big(\,(\xi\cdot\mathbf{n})\,\partial_{\mathbf{n}}(\theta\cdot\mathbf{n}) - \big(\theta\cdot\nabla_\Gamma\,(\xi\cdot\mathbf{n})\,\big)\Big)\\ + \alpha(\alpha-1)\int_\Omega k\,|u-u_0|^{\alpha-2}u_\theta' u_\xi'\\ + \alpha\int_\Gamma k\,|u-u_0|^{\alpha-2}(u-u_0)\Big(\,(\xi\cdot\mathbf{n})\,u_\theta' + (\theta\cdot\mathbf{n})\,u_\xi'\Big)\\ - \int_\Omega \alpha k\,|u-u_0|^{\alpha-2}(u-u_0)u'_{(\xi\cdot\nabla)\theta}\\ + \int_\Gamma (\theta\cdot\mathbf{n})\,(\xi\cdot\mathbf{n})\left(\partial_{\mathbf{n}}\Big(k\,|u-u_0|^\alpha\Big) + \mathcal{H}k\,|u-u_0|^\alpha\right)\\ + \int_\Gamma k\,|u-u_0|^\alpha\Big(\,(\xi\cdot\mathbf{n})\,\partial_{\mathbf{n}}(\theta\cdot n) - \Big(\theta\cdot\nabla_\Gamma\,(\xi\cdot\mathbf{n})\,\Big) - \big(\,(\xi\cdot\nabla)\,\theta\cdot\mathbf{n}\big)\Big).\end{aligned}$$

Now we consider specifically the term with $u'_{(\xi\cdot\nabla)\theta}$. Thanks to the variational formulation for the adjoint p, we have:

$$\begin{aligned}
-\alpha\int_\Omega k\,|u-u_0|^{\alpha-2}(u-u_0)u'_{(\xi\cdot\nabla)\theta} &= -\int_\Omega \operatorname{div}\big(A\varepsilon(p)\big)u'_{(\xi\cdot\nabla)\theta}\\
&= \int_\Omega A\varepsilon(p):\varepsilon\left(u'_{(\xi\cdot\nabla)\theta}\right)-\int_\Gamma A\varepsilon(p)\mathbf{n}\cdot u'_{(\xi\cdot\nabla)\theta}\\
&= -\int_\Omega \operatorname{div}\Big(A\varepsilon\left(u'_{(\xi\cdot\nabla)\theta}\right)\Big)p+\int_\Gamma A\varepsilon\left(u'_{(\xi\cdot\nabla)\theta}\right)\mathbf{n}\cdot p\\
&= \int_\Gamma \operatorname{div}_\Gamma\Big(\big(\,(\xi\cdot\nabla)\,\theta\cdot\mathbf{n}\big)A\varepsilon(u)\Big)p\\
&= -\int_\Gamma\big(\,(\xi\cdot\nabla)\,\theta\cdot\mathbf{n}\big)A\varepsilon(u):\varepsilon(p).
\end{aligned}$$

Therefore

$$\begin{aligned}
j_2''(\Omega;\theta,\xi) = &\int_\Gamma(\xi\cdot\mathbf{n})\,A\varepsilon(u'_\theta):\varepsilon(p)+(\theta\cdot\mathbf{n})\,A\varepsilon(u'_\xi):\varepsilon(p)\\
&+\int_\Gamma(\xi\cdot\mathbf{n})\,(\theta\cdot\mathbf{n})\left(\partial_{\mathbf{n}}\Big(A\varepsilon(u):\varepsilon(p)\Big)+\mathcal{H}A\varepsilon(u):\varepsilon(p)\right)\\
&+\int_\Gamma A\varepsilon(u):\varepsilon(p)\Big(\,(\xi\cdot\mathbf{n})\,\partial_{\mathbf{n}}(\theta\cdot\mathbf{n})-\big(\theta\cdot\nabla_\Gamma\,(\xi\cdot\mathbf{n})\,\big)-\big((\xi\cdot\nabla)\,\theta\cdot\mathbf{n}\big)\,\Big)\\
&+\alpha(\alpha-1)\int_\Omega k\,|u-u_0|^{\alpha-2}u'_\theta u'_\xi\\
&+\alpha\int_\Gamma k\,|u-u_0|^{\alpha-2}(u-u_0)\Big(\,(\xi\cdot\mathbf{n})\,u'_\theta+(\theta\cdot\mathbf{n})\,u'_\xi\Big)\\
&+\int_\Gamma(\theta\cdot\mathbf{n})\,(\xi\cdot\mathbf{n})\left(\partial_{\mathbf{n}}\Big(k\,|u-u_0|^\alpha\Big)+\mathcal{H}k\,|u-u_0|^\alpha\right)\\
&+\int_\Gamma k\,|u-u_0|^\alpha\Big(\,(\xi\cdot\mathbf{n})\,\partial_{\mathbf{n}}(\theta\cdot n)-\Big(\theta\cdot\nabla\,(\xi\cdot\mathbf{n})\,\Big)-\big(\,(\xi\cdot\nabla)\,\theta\cdot\mathbf{n}\big)\Big),
\end{aligned}$$

and then

$$\begin{aligned}
j_2''(\Omega;\theta,\xi) = &\int_\Gamma(\xi\cdot\mathbf{n})\,A\varepsilon(u'_\theta):\varepsilon(p)+(\theta\cdot\mathbf{n})\,A\varepsilon(u'_\xi):\varepsilon(p)+\alpha(\alpha-1)\int_\Omega k\,|u-u_0|^{\alpha-2}u'_\theta u'_\xi\\
&+\alpha\int_\Gamma k\,|u-u_0|^{\alpha-2}(u-u_0)\Big(\,(\xi\cdot\mathbf{n})\,u'_\theta+(\theta\cdot\mathbf{n})\,u'_\xi\Big)\\
&+\int_\Gamma(\theta\cdot\mathbf{n})\,(\xi\cdot\mathbf{n})\left(\partial_{\mathbf{n}}\mathscr{L}+\mathcal{H}\mathscr{L}\right)+\int_\Gamma\mathscr{L}Z_{\theta,\xi},
\end{aligned}$$

where

$$\begin{aligned}
\mathscr{L} &= A\varepsilon(u):\varepsilon(p)+k|u-u_0|^\alpha,\\
Z_{\theta,\xi} &= \big(\,(\xi_\Gamma\cdot\nabla_\Gamma)\,\mathbf{n}\big)\cdot\theta_\Gamma-\nabla_\Gamma\,(\theta\cdot\mathbf{n})\cdot\xi_\Gamma-\nabla_\Gamma\,(\xi\cdot\mathbf{n})\cdot\theta_\Gamma.
\end{aligned}$$

□

Lemma 7.1.16. *For $\theta,\xi\in\mathcal{F}$, the second-order shape derivative $(u'_\theta)'_\xi$ of the state u satisfies*

$$\begin{aligned}
\int_\Gamma A\varepsilon\big((u'_\theta)'_\xi\big)\boldsymbol{n}\cdot v = &-\int_\Gamma(\xi\cdot\boldsymbol{n})\,A\varepsilon(u'_\theta):\varepsilon(v)+(\theta\cdot\boldsymbol{n})\,A\varepsilon(u'_\xi):\varepsilon(v)\\
&-\int_\Gamma(\xi\cdot\boldsymbol{n})\,(\theta\cdot\boldsymbol{n})\left(\partial_{\boldsymbol{n}}\Big(A\varepsilon(u):\varepsilon(v)\Big)+\mathcal{H}A\varepsilon(u):\varepsilon(v)\right)\\
&+\int_\Gamma A\varepsilon(u):\varepsilon(v)\left(\big(\theta\cdot\nabla_\Gamma\,(\xi\cdot\boldsymbol{n})\,\big)-(\xi\cdot\boldsymbol{n})\,\partial_{\boldsymbol{n}}(\theta\cdot\boldsymbol{n})\right).
\end{aligned}$$

Proof. The variational formulation for u reads

$$\forall v\in\mathcal{V},\ \int_\Omega A\varepsilon(u):\varepsilon(v)=\int_\Gamma gv,$$

where

$$\mathcal{V}=\left\{v\in H^1(\Omega)\mid v=0\text{ on }\Gamma_D\right\}.$$

The shape derivation of this equality leads to

$$\forall v \in \mathcal{V},\ \int_\Omega A\varepsilon(u'_\theta) : \varepsilon(v) + \int_\Gamma (\theta\cdot\mathbf{n})\, A\varepsilon(u) : \varepsilon(v) = 0,$$

and the second-order derivation gives $\forall v \in \mathcal{V}$,

$$\begin{aligned} 0 = &\int_\Omega A\varepsilon\big((u'_\theta)'_\xi\big) : \varepsilon(v) + \int_\Gamma (\xi\cdot\mathbf{n})\, A\varepsilon(u'_\theta) : \varepsilon(v) \\ &+ \int_\Gamma \big(\theta\cdot\mathbf{n}'_\xi\big) A\varepsilon(u) : \varepsilon(v) + \int_\Gamma (\theta\cdot\mathbf{n})\, A\varepsilon(u'_\xi) : \varepsilon(v) \\ &+ \int_\Gamma (\xi\cdot\mathbf{n}) \left(\partial_\mathbf{n}\Big((\theta\cdot\mathbf{n})\, A\varepsilon(u) : \varepsilon(v)\Big) + \mathcal{H}\,(\theta\cdot\mathbf{n})\, A\varepsilon(u) : \varepsilon(v)\right). \end{aligned}$$

An integration by part of the volumic term gives

$$\begin{aligned} 0 = &-\int_\Omega \operatorname{div}\Big(A\varepsilon\big((u'_\theta)'_\xi\big)\Big) v + \int_\Gamma A\varepsilon\big((u'_\theta)'_\xi\big)\mathbf{n}\cdot v \\ &+ \int_\Gamma (\xi\cdot\mathbf{n})\, A\varepsilon(u'_\theta) : \varepsilon(v) + \int_\Gamma \big(\theta\cdot\mathbf{n}'_\xi\big) A\varepsilon(u) : \varepsilon(v) + \int_\Gamma (\theta\cdot\mathbf{n})\, A\varepsilon(u'_\xi) : \varepsilon(v) \\ &+ \int_\Gamma (\xi\cdot\mathbf{n}) \left(\partial_\mathbf{n}\Big((\theta\cdot\mathbf{n})\, A\varepsilon(u) : \varepsilon(v)\Big) + \mathcal{H}\,(\theta\cdot\mathbf{n})\, A\varepsilon(u) : \varepsilon(v)\right). \end{aligned}$$

With v varying in $\mathcal{V}$ we obtain that $\operatorname{div}\Big(A\varepsilon\big((u'_\theta)'_\xi\big)\Big) = 0$ and

$$\begin{aligned} \int_\Gamma A\varepsilon\big((u'_\theta)'_\xi\big)\mathbf{n}\cdot v = &-\int_\Gamma (\xi\cdot\mathbf{n})\, A\varepsilon(u'_\theta) : \varepsilon(v) + (\theta\cdot\mathbf{n})\, A\varepsilon(u'_\xi) : \varepsilon(v) \\ &- \int_\Gamma (\xi\cdot\mathbf{n})\,(\theta\cdot\mathbf{n}) \left(\partial_\mathbf{n}\Big(A\varepsilon(u) : \varepsilon(v)\Big) + \mathcal{H}A\varepsilon(u) : \varepsilon(v)\right) \\ &+ \int_\Gamma A\varepsilon(u) : \varepsilon(v)\left(\big(\theta\cdot\nabla_\Gamma\,(\xi\cdot\mathbf{n})\,\big) - (\xi\cdot\mathbf{n})\,\partial_\mathbf{n}(\theta\cdot\mathbf{n})\right). \end{aligned}$$

□

We can observe that the second-order shape derivative of the displacement is symmetric in θ, ξ. We can also write it in another way that could be more useful from a numerical point of view.

Proposition 7.1.17. *With the same notation as previously, when the adjoint p is shape-differentiable, the second-order shape derivative of j_2 also reads*

$$\begin{aligned} j_2''(\Omega;\theta,\xi) = &\int_\Gamma (\theta\cdot\boldsymbol{n})\, A\varepsilon(p'_\xi) : \varepsilon(u) + (\theta\cdot\boldsymbol{n})\, A\varepsilon(u'_\xi) : \varepsilon(p) \\ &+ \alpha\int_\Gamma k\,|u-u_0|^{\alpha-2}(u-u_0)\,(\theta\cdot\boldsymbol{n})\, u'_\xi \\ &+ \int_\Gamma (\theta\cdot\boldsymbol{n})\,(\xi\cdot\boldsymbol{n})\left(\partial_{\boldsymbol{n}}\mathscr{L} + \mathcal{H}\mathscr{L}\right) + \int_\Gamma \mathscr{L}\, Z_{\theta,\xi}. \end{aligned}$$

Proof. The variational formulation for the adjoint p is

$$\forall v \in \mathcal{V},\ -\int_\Omega \operatorname{div}\Big(A\varepsilon(p)\Big) v = -\int_\Omega \alpha k\,|u-u_0|(u-u_0)v.$$

After integration by parts we obtain

$$\forall v \in \mathcal{V},\ \int_\Omega A\varepsilon(p) : \varepsilon(v) + \int_\Omega \alpha k\,|u-u_0|(u-u_0)v = 0.$$

For $v \in \mathcal{V}$ and $\xi \in \mathcal{F}$, the derivation with respect to the shape leads to

$$\begin{aligned} 0 = &\int_\Omega A\varepsilon(p'_\xi) : \varepsilon(v) + \int_\Gamma (\xi\cdot\mathbf{n})\, A\varepsilon(p) : \varepsilon(v) \\ &+ \int_\Omega \alpha(\alpha-1)k\,|u-u_0|^{\alpha-2}u'_\xi v + \int_\Gamma (\xi\cdot\mathbf{n})\,\alpha k\,|u-u_0|^{\alpha-2}(u-u_0)v. \end{aligned}$$

Now, taking $v = u'_\theta$ and integrating by parts gives

$$\begin{aligned}0 = &- \int_\Omega \operatorname{div}\Big(A\varepsilon(u'_\theta)\Big)p'_\xi + \int_\Gamma A\varepsilon(u'_\theta)\mathbf{n}\cdot p'_\xi \\ &+ \int_\Gamma (\xi\cdot\mathbf{n})\, A\varepsilon(p) : \varepsilon(u'_\theta) + \int_\Omega \alpha(\alpha-1)k\,|u-u_0|^{\alpha-2}u'_\xi u'_\theta \\ &+ \int_\Gamma (\xi\cdot\mathbf{n})\,\alpha k\,|u-u_0|^{\alpha-2}(u-u_0)u'_\theta.\end{aligned}$$

But $\operatorname{div}\Big(A\varepsilon(u'_\theta)\Big) = 0$ and

$$\int_\Gamma A\varepsilon(u'_\theta)\mathbf{n}\cdot p'_\xi = \int_\Gamma \operatorname{div}_\Gamma\Big((\theta\cdot\mathbf{n})\, A\varepsilon(u)\Big)p'_\xi = -\int_\Gamma (\theta\cdot\mathbf{n})\, A\varepsilon(u):\varepsilon(p'_\xi).$$

Therefore

$$\begin{aligned}\int_\Gamma (\theta\cdot\mathbf{n})\, A\varepsilon(u):\varepsilon(p'_\xi) = &\int_\Gamma (\xi\cdot\mathbf{n})\, A\varepsilon(p):\varepsilon(u'_\theta) + \int_\Omega \alpha(\alpha-1)k\,|u-u_0|^{\alpha-2}u'_\xi u'_\theta \\ &+ \int_\Gamma (\xi\cdot\mathbf{n})\,\alpha k\,|u-u_0|^{\alpha-2}(u-u_0)u'_\theta,\end{aligned}$$

which gives the expected result. □

Proposition 7.1.18. *Let $k \geq 2$, $\Omega \in \mathcal{O}_k$ and $\theta, \xi \in C^{k,\infty}(\mathbb{R}^d;\mathbb{R}^d)$. The derivatives of the displacement criterion J_2 reads*

$$\begin{aligned}J_2'(\Omega;\theta) &= \frac{1}{\alpha} j_2'(\Omega;\theta)\, j_2(\Omega)^{\frac{1}{\alpha}-1}, \\ J_2''(\Omega;\theta,\xi) &= \frac{1}{\alpha} j_2''(\Omega;\theta,\xi)\, j_2(\Omega)^{\frac{1}{\alpha}-1} + \frac{1}{\alpha}(\frac{1}{\alpha}-1) j_2'(\Omega;\theta)\, j_2'(\Omega,\xi)\, j_2(\Omega)^{\frac{1}{\alpha}-2}.\end{aligned}$$

Proof. This is straightforwardly given by the chain rule and the relations

$$\begin{aligned}j_2''(\Omega;\theta,\xi) &= (j_2')'(\Omega;\theta,\xi) - j_2'(\Omega;\nabla\theta\xi), \\ J_2''(\Omega;\theta,\xi) &= (J_2')'(\Omega;\theta,\xi) - J_2'(\Omega;\nabla\theta\xi).\end{aligned}$$

□

7.1.5 Maximum-thickness criterion

In order to consider a maximum-thickness criterion we use the quadratic penalty function

$$\mathcal{P}_{Max}(\Omega) = \int_\Omega \left(\left(d_\Omega(x) + \frac{d_{max}}{2}\right)^-\right)^2 dx. \tag{7.19}$$

This criterion was introduced by G. Michailidis [83]. He explained the computation of the first-order derivative, whose result we recall here. We also refer to Section 3.3.2 for the definition of the projection, the skeleton, and other geometric quantities we need here.

Lemma 7.1.19. *The shape derivative of (7.19) reads*

$$\mathcal{P}_{Max}{}'(\Omega;\theta) = \int_{\partial\Omega} -\Big(\theta(x)\cdot\boldsymbol{n}(x)\Big)\int_{\mathrm{ray}_{\partial\Omega}(x)\cap\Omega}\left(2\left(d_\Omega(z)+\frac{d_{max}}{2}\right)^-\prod_{i=1}^{d-1}\Big(1+d_\Omega(z)\kappa_i(x)\Big)\,dz\right)dx.$$

Now we are concerned with the computation of second-order derivative of such a criterion. We need first the following lemma.

Lemma 7.1.20. *Let $\Omega \subset \mathbb{R}^d$ be a bounded domain of class C^2 and $m(s) : \mathbb{R} \mapsto \mathbb{R}$ a function of class C^2. We assume that m vanishes on $\overline{\Sigma}$, where Σ is the skeleton. Define the functional*

$$J_3(\Omega) = \int_\Omega m(y)\,dx.$$

The function J_3 is twice shape-differentiable and its derivatives read

$$\begin{aligned}
J_3{}'(\Omega;\theta) =& -\int_\Omega (\theta\cdot\boldsymbol{n})(z)\, m'(y)\,dx + m(0)\int_{\partial\Omega}(\theta\cdot\boldsymbol{n})(x)\,dx,\\
J_3{}''(\Omega;\theta,\xi) =& -\int_\Omega y\Big(\big(1+yD^2d_\Omega(z)\big)^{-1}\nabla_\Gamma(\xi\cdot\boldsymbol{n})(z)\cdot\nabla_\Gamma(\theta\cdot\boldsymbol{n})(z)\Big)m'(y)\,dx\\
&+\int_\Omega (\xi\cdot\boldsymbol{n})(z)\ (\theta\cdot\boldsymbol{n})(z)\,m''(y)\,dx\\
&+m(0)\int_{\partial\Omega}(\xi\cdot\boldsymbol{n})(\theta\cdot\boldsymbol{n})\mathcal{H}\,dx - m'(0)\int_{\partial\Omega}(\xi\cdot\boldsymbol{n})(\theta\cdot\boldsymbol{n})\ dx\\
&+\int_\Omega Z_{\theta,\xi}\circ p_{\partial\Omega}(x)\,m'(y)\,dx - m(0)\int_{\partial\Omega}Z_{\theta,\xi}\,dx,
\end{aligned}$$

where $z = p_{\partial\Omega}(x)$, $y = d_\Omega(x)$ and

$$Z_{\theta,\xi} = \big((\xi_\Gamma\cdot\nabla_\Gamma)\,\boldsymbol{n}\big)\cdot\theta_\Gamma - \nabla_\Gamma(\theta\cdot\boldsymbol{n})\cdot\xi_\Gamma - \nabla_\Gamma(\xi\cdot\boldsymbol{n})\cdot\theta_\Gamma.$$

Proof. The first-order derivation formula yields to

$$\begin{aligned}
J_3{}'(\Omega;\theta) &= \int_\Omega m'(y)d'_\theta(x)\,dx + m(0)\int_{\partial\Omega}(\theta\cdot\mathbf{n})(x)\,dx\\
&= -\int_\Omega(\theta\cdot\mathbf{n})(z)\,m'(y)\,dx + m(0)\int_{\partial\Omega}(\theta\cdot\mathbf{n})(x)\,dx.
\end{aligned}$$

Since we assumed that $\Omega\in\mathcal{O}_3$, $p_{\partial\Omega}$ is shape-differentiable (see Proposition 3.3.16) on $\mathcal{D}\backslash\overline{\Sigma}$. Now, since m vanishes on $\overline{\Sigma}$ the regularity assumption on m allows one to derive $J_3{}'$.

$$\begin{aligned}
\Big(J_3{}'(\Omega;\theta)\Big)'(\Omega;\xi) =& -\int_\Omega\Big(\big(p'_{\partial\Omega,\xi}(x)\cdot\nabla\big)\theta(z)\cdot\mathbf{n}(z) + \big(\theta\cdot\mathbf{n}'_\xi\big)(z) + \big(p'_{\partial\Omega,\xi}(x)\cdot\nabla\big)\mathbf{n}(z)\cdot\theta(z)\Big)m'(y)\,dx\\
&+\int_\Omega(\xi\cdot\mathbf{n})(z)\ (\theta\cdot\mathbf{n})(z)\,m''(y)\,dx\\
&-\int_{\partial\Omega}(\theta\cdot\mathbf{n})(\xi\cdot\mathbf{n})\,m'(y)\,dx + m(0)\int_{\partial\Omega}\big(\theta\cdot\mathbf{n}'_\xi\big)(x)\,dx\\
&+m(0)\int_{\partial\Omega}(\xi\cdot\mathbf{n})\Big(\partial_{\mathbf{n}}(\theta\cdot\mathbf{n}) + \mathcal{H}(\theta\cdot\mathbf{n})\Big)\,dx.
\end{aligned}$$

We can compute $J_3{}''(;,)$:

$$\begin{aligned}
J_3{}''(\Omega;\theta,\xi) =& \Big(J_3{}'(\Omega;\theta)\Big)'(\Omega;\xi) - J_3{}'(\Omega;(\xi\cdot\nabla)\theta)\\
&-\int_\Omega\Big(\big(p'_{\partial\Omega,\xi}(x)\cdot\nabla\big)\theta(z)\cdot\mathbf{n}(z) + \big(\theta\cdot\mathbf{n}'_\xi\big)(z) + \big(p'_{\partial\Omega,\xi}(x)\cdot\nabla\big)\mathbf{n}(z)\cdot\theta(z)\Big)m'(y)\,dx\\
&+\int_\Omega(\xi\cdot\mathbf{n})(z)\ (\theta\cdot\mathbf{n})(z)\,m''(y)\,dx\\
&-m'(0)\int_{\partial\Omega}(\theta\cdot\mathbf{n})(\xi\cdot\mathbf{n})\ dx + m(0)\int_{\partial\Omega}\big(\theta\cdot\mathbf{n}'_\xi\big)\ dx\\
&+m(0)\int_{\partial\Omega}(\xi\cdot\mathbf{n})\Big(\partial_{\mathbf{n}}(\theta\cdot\mathbf{n}) + \mathcal{H}(\theta\cdot\mathbf{n})\Big)\,dx\\
&+\int_\Omega\Big((\xi\cdot\nabla)\theta(z)\cdot\mathbf{n}(z)\Big)m'(y)\,dx - m(0)\int_{\partial\Omega}(\xi\cdot\nabla)\theta\cdot\mathbf{n}\,dx.
\end{aligned}$$

Let us introduce

$$\begin{aligned}
A =& -\int_\Omega\Big(\big(p'_{\partial\Omega,\xi}(x)\cdot\nabla\big)\theta(z)\cdot\mathbf{n}(z) + \big(\theta\cdot\mathbf{n}'_\xi\big)(z) + \big(p'_{\partial\Omega,\xi}(x)\cdot\nabla\big)\mathbf{n}(z)\cdot\theta(z)\Big)m'(y)\,dx\\
&+\int_\Omega\Big((\xi(z)\cdot\nabla)\theta(z)\cdot\mathbf{n}(z)\Big)m'(y)\,dx,\\
B =& m(0)\int_{\partial\Omega}\big(\theta\cdot\mathbf{n}'_\xi\big)\ dx + m(0)\int_{\partial\Omega}(\xi\cdot\mathbf{n})\,\partial_{\mathbf{n}}(\theta\cdot\mathbf{n})\,dx - m(0)\int_{\partial\Omega}(\xi\cdot\nabla)\theta\cdot\mathbf{n}\,dx.
\end{aligned}$$

With

$$p'_{\partial\Omega,\xi} = \Big(\xi(z)\cdot\mathbf{n}(z)\Big)\mathbf{n}(z) + y\Big(1+y\mathcal{H}d_\Omega(z)\Big)^{-1}\nabla_\Gamma(\xi\cdot\mathbf{n})(z),$$

we get

$$\begin{aligned}
A =& -\int_\Omega (\xi\cdot\mathbf{n})(z)\Big(\left(\mathbf{n}(z)\cdot\nabla\right)\theta(z)\cdot\mathbf{n}(z)\Big)m'(y)\,dx\\
&-\int_\Omega m'(y)y\left(\left(1+yD^2d_\Omega(z)\right)^{-1}\nabla_\Gamma(\xi\cdot\mathbf{n})(z)\cdot\nabla\right)\theta(z)\cdot\mathbf{n}(z)\,dx\\
&+\int_\Omega\Big(\theta(z)\cdot\nabla_\Gamma(\xi\cdot\mathbf{n})(z)\Big)m'(y)\\
&-\int_\Omega(\xi\cdot\mathbf{n})(z)\Big(\left(\mathbf{n}(z)\cdot\nabla\right)\mathbf{n}(z)\cdot\theta(z)\Big)m'(y)\,dx\\
&-\int_\Omega m'(y)y\left(\left(1+yD^2d_\Omega(z)\right)^{-1}\nabla_\Gamma(\xi\cdot\mathbf{n})(z)\cdot\nabla\right)\mathbf{n}(z)\cdot\theta(z)\,dx\\
&+\int_\Omega(\xi\cdot\mathbf{n})(z)\Big(\left(\mathbf{n}(z)\cdot\nabla\right)\theta(z)\cdot\mathbf{n}(z)\Big)m'(y)\,dx\\
&+\int_\Omega\Big(\left(\xi_\Gamma(z)\cdot\nabla\right)\theta(z)\cdot\mathbf{n}(z)\Big)m'(y)\,dx.
\end{aligned}$$

With Lemma 7.1.5, we have

$$\begin{aligned}
A =& -\int_\Omega m'(y)y\left(\left(1+yD^2d_\Omega(z)\right)^{-1}\nabla_\Gamma(\xi\cdot\mathbf{n})(z)\cdot\nabla\right)\theta(z)\cdot\mathbf{n}(z)\,dx\\
&-\int_\Omega m'(y)y\left(\left(1+yD^2d_\Omega(z)\right)^{-1}\nabla_\Gamma(\xi\cdot\mathbf{n})(z)\cdot\nabla\right)\mathbf{n}(z)\cdot\theta(z)\,dx\\
&+\int_\Omega\Big(\theta_\Gamma\cdot\nabla_\Gamma(\xi\cdot\mathbf{n})+\xi_\Gamma\cdot\nabla_\Gamma(\theta\cdot\mathbf{n})-(\xi_\Gamma\cdot\nabla)\,\mathbf{n}\cdot\theta_\Gamma\Big)m'(y)\,dx,
\end{aligned}$$

and

$$\begin{aligned}
A =& -\int_\Omega m'(y)y\Big(\nabla(\theta\cdot\mathbf{n})\cdot\left(1+yD^2d_\Omega(z)\right)^{-1}\nabla_\Gamma(\xi\cdot\mathbf{n})(z)\Big)\,dx\\
&-\int_\Omega Z(\theta,\xi)\circ p_{\partial\Omega}(x)\,m'(y)\,dx.
\end{aligned}$$

Applying the same lemma to B gives

$$(\xi\cdot\nabla)\,\theta\cdot\mathbf{n}=(\xi\cdot\mathbf{n})\,\partial_{\mathbf{n}}(\theta\cdot\mathbf{n})+\xi_\Gamma\cdot\nabla_\Gamma(\theta\cdot\mathbf{n})-(\xi_\Gamma\cdot\nabla)\,\mathbf{n}\cdot\theta_\Gamma,$$

and

$$\begin{aligned}
B =& m(0)\int_{\partial\Omega}-\theta\cdot\nabla_\Gamma(\xi\cdot\mathbf{n})+(\xi\cdot\mathbf{n})\,\partial_{\mathbf{n}}(\theta\cdot\mathbf{n})\\
&-m(0)\int_{\partial\Omega}(\xi\cdot\mathbf{n})\,\partial_{\mathbf{n}}(\theta\cdot\mathbf{n})+\xi_\Gamma\cdot\nabla_\Gamma(\theta\cdot\mathbf{n})-(\xi_\Gamma\cdot\nabla)\,\mathbf{n}\cdot\theta_\Gamma\\
=& m(0)\int_{\partial\Omega}Z_{\theta,\xi}\,dx.
\end{aligned}$$

Then

$$\begin{aligned}
J_3{}''(\Omega;\theta,\xi) =& -\int_\Omega m'(y)y\Big(\nabla(\theta\cdot\mathbf{n})\cdot\left(1+yD^2d_\Omega(z)\right)^{-1}\nabla_\Gamma(\xi\cdot\mathbf{n})(z)\Big)\,dx\\
&+\int_\Omega(\xi\cdot\mathbf{n})(z)\;(\theta\cdot\mathbf{n})(z)\,m''(y)\,dx\\
&-m'(0)\int_{\partial\Omega}(\theta\cdot\mathbf{n})(\xi\cdot\mathbf{n})\;dx+m(0)\int_{\partial\Omega}\mathcal{H}(\xi\cdot\mathbf{n})(\theta\cdot\mathbf{n})\;dx\\
&-\int_\Omega Z_{\theta,\xi}\circ p_{\partial\Omega}(x)\,m'(y)\,dx+m(0)\int_{\partial\Omega}Z_{\theta,\xi}\,dx.
\end{aligned}$$

□

Lemma 7.1.21. *Let $\Omega\subset\mathbb{R}^d$ be a bounded domain of class C^2 and $m(s):\mathbb{R}\to\mathbb{R}$ a function of class C^2. We assume that m vanishes on $\overline{\Sigma}$, where Σ is the skeleton. Define the functional*

$$J_4(\Omega)=\int_\Omega m\big(d_\Omega(x)\big)d_\Omega(x)^p\,dx.$$

This functional is twice shape-differentiable at $\theta = 0$ and its derivatives read

$$
\begin{aligned}
J_4{}'(\Omega;\theta) &= -\int_\Omega (\theta\cdot\boldsymbol{n})(z)\left(m'(y)y^p + p\,m(y)y^{p-1}\right)dx,\\
J_4{}''(\Omega;\theta,\xi) &= -\int_\Omega y\left(\left(1+y\mathcal{H}d_\Omega(z)\right)^{-1}\nabla_\Gamma(\xi\cdot\boldsymbol{n})(z)\cdot\nabla_\Gamma(\theta\cdot\boldsymbol{n})(z)\right)\left(m'(y)y^p + p\,m(y)y^{p-1}\right)dx\\
&\quad+\int_\Omega (\xi\cdot\boldsymbol{n})(z)\ (\theta\cdot\boldsymbol{n})(z)\left(m''(y)y^p + 2p\,m'(y)y^{p-1} + p(p-1)\,m(y)y^{p-2}\right)dx\\
&\quad-\int_\Omega Z_{\theta,\xi}\circ p_{\partial\Omega}(x)\left(m'(y)y^p + p\,m(y)y^{p-1}\right)dx,
\end{aligned}
$$

where $z = p_{\partial\Omega}(x)$, $y = d_\Omega(x)$ and

$$
Z_{\theta,\xi} = \left(\left(\xi_\Gamma\cdot\nabla_\Gamma\right)\boldsymbol{n}\right)\cdot\theta_\Gamma - \nabla_\Gamma(\theta\cdot\boldsymbol{n})\cdot\xi_\Gamma - \nabla_\Gamma(\xi\cdot\boldsymbol{n})\cdot\theta_\Gamma.
$$

Proof. Since $\forall x\in\partial\Omega$, $d_\Omega(x) = 0$ the first-order derivation yields

$$
\begin{aligned}
J_4{}'(\Omega;\theta) &= \int_\Omega p d'_\theta(x) m(y)y^{p-1} + d'_\theta(x)m'(y)y^p + \int_{\partial\Omega}(\theta\cdot\mathbf{n})\,m(y)y^p\\
&= -\int_\Omega (\theta\cdot\mathbf{n})(z)\Big(m'(y)y^p + p\,m(y)y^{p-1}\Big)dx.
\end{aligned}
$$

We recall that since $y = d_\Omega(x)$, we have $y = 0$ on $\partial\Omega$. Then

$$
\begin{aligned}
\Big(J_4{}'(\Omega;\theta)\Big)'(\Omega;\xi) &= -\int_\Omega\Big(\left(p'_{\partial\Omega,\xi}(x)\cdot\nabla\right)\theta(z)\cdot\mathbf{n}(z) + \left(p'_{\partial\Omega,\xi}(x)\cdot\nabla\right)\mathbf{n}(z)\cdot\theta(z)\Big)\Big(m'(y)y^p + p\,m(y)y^{p-1}\Big)dx\\
&\quad-\int_\Omega\left(\theta\cdot\mathbf{n}'_\xi\right)(z)\Big(m'(y)y^p + p\,m(y)y^{p-1}\Big)dx\\
&\quad-\int_\Omega(\theta\cdot\mathbf{n})(z)\Big(pd'_\xi m'(y)y^{p-1} + p(p-1)d'_\xi m(y)y^{p-2} + d'_\xi m''(y)y^p + pd'_\xi m'(y)y^{p-1}\Big)dx\\
&\quad-\int_{\partial\Omega}(\xi\cdot\mathbf{n})(\theta\cdot\mathbf{n})\Big(pm(y)y^{p-1} + m'(y)y^p\Big)dx.
\end{aligned}
$$

With $d'_\xi = -(\xi\cdot\mathbf{n})(z)$ we have

$$
\begin{aligned}
\Big(J_4{}'(\Omega;\theta)\Big)'(\Omega;\xi) &= -\int_\Omega\Big(\left(p'_{\partial\Omega,\xi}(x)\cdot\nabla\right)\theta(z)\cdot\mathbf{n}(z) + \left(p'_{\partial\Omega,\xi}(x)\cdot\nabla\right)\mathbf{n}(z)\cdot\theta(z)\Big)\Big(m'(y)y^p + p\,m(y)y^{p-1}\Big)dx\\
&\quad-\int_\Omega\left(\theta\cdot\mathbf{n}'_\xi\right)(z)\Big(m'(y)y^p + p\,m(y)y^{p-1}\Big)dx\\
&\quad+\int_\Omega(\theta\cdot\mathbf{n})(z)(\xi\cdot\mathbf{n})(z)\Big(2pm'(y)y^{p-1} + p(p-1)m(y)y^{p-2} + m''(y)y^p + pd'_\xi m'(y)y^{p-1}\Big)dx.
\end{aligned}
$$

As a result,

$$
\begin{aligned}
J_4{}''(\Omega;\theta,\xi) &= \Big(J_4{}'(\Omega;\theta)\Big)'(\Omega;\xi) - J_4{}'(\Omega;(\xi\cdot\nabla)\theta)\\
&= -\int_\Omega\Big(\left(p'_{\partial\Omega,\xi}(x)\cdot\nabla\right)\theta(z)\cdot\mathbf{n}(z) + \left(p'_{\partial\Omega,\xi}(x)\cdot\nabla\right)\mathbf{n}(z)\cdot\theta(z)\Big)\Big(m'(y)y^p + p\,m(y)y^{p-1}\Big)dx\\
&\quad-\int_\Omega\left(\theta\cdot\mathbf{n}'_\xi\right)(z)\Big(m'(y)y^p + p\,m(y)y^{p-1}\Big)dx\\
&\quad+\int_\Omega(\theta\cdot\mathbf{n})(z)(\xi\cdot\mathbf{n})(z)\Big(2pm'(y)y^{p-1} + p(p-1)m(y)y^{p-2} + m''(y)y^p + pd'_\xi m'(y)y^{p-1}\Big)dx\\
&\quad+\int_\Omega\left((\xi\cdot\nabla)\theta\cdot\mathbf{n}\right)(z)\Big(pm(y)y^{p-1} + m'(y)y^p\Big).
\end{aligned}
$$

The same computation as in the previous lemma gives

$$
\begin{aligned}
J_4{}''(\Omega;\theta,\xi) &= -\int_\Omega y\big(pm(y)y^{p-1} + m'(y)y^p\big)\Big(\nabla(\theta\cdot\mathbf{n})\cdot\left(1+y\mathcal{H}d_\Omega(z)\right)^{-1}\nabla_\Gamma(\xi\cdot\mathbf{n})(z)\Big)dx\\
&\quad+\int_\Omega(\theta\cdot\mathbf{n})(z)(\xi\cdot\mathbf{n})(z)\Big(2pm'(y)y^{p-1} + p(p-1)m(y)y^{p-2} + m''(y)y^p + pd'_\xi m'(y)y^{p-1}\Big)dx\\
&\quad-\int_\Omega Z_{\theta,\xi}\circ p_{\partial\Omega}(x)\Big(m'(y)y^p + p\,m(y)y^{p-1}\Big)dx.
\end{aligned}
$$

□

7.2 Structure of the second-order shape derivative

Now we are concerned with the structure of the second-order shape derivative in the framework of geometric shape optimization. To that end, we come back to the *scalar thermal model* (see Section 2.1.4), and the compliance criterion

$$J_1(\Omega) = \int_\Omega |\nabla u|^2,$$

where the state u is solution to

$$\left\{ \begin{array}{rcll} -\Delta u &=& 0 & \text{in } \Omega, \\ \partial_{\mathbf{n}} u &=& g & \text{on } \Gamma_N, \\ \partial_{\mathbf{n}} u &=& 0 & \text{on } \Gamma, \\ u &=& 0 & \text{on } \Gamma_D. \end{array} \right. \tag{7.20}$$

We consider shape variations such that the parts $\Gamma_D \cup \Gamma_N$ are not modified. We therefore introduce $\mathcal{F}$ by

$$\mathcal{F} = \left\{ \theta \in C^{2,\infty}(\mathbb{R}^d;\mathbb{R}^d) \mid \theta = 0 \text{ on } \Gamma_D \cup \Gamma_N \right\}.$$

For $\theta, \xi \in \mathcal{F}$, we recall that the shape derivatives are given by

$$\begin{aligned} J_1{}'(\Omega;\theta) &= -\int_\Gamma (\theta\cdot\mathbf{n})\,|\nabla u|^2, \\ J_1{}''(\Omega;\theta,\xi) &= -2\int_\Gamma (\theta\cdot\mathbf{n})\,\nabla u'_\xi\cdot\nabla u - \int_\Gamma (\theta\cdot\mathbf{n})\,(\xi\cdot\mathbf{n})\left(\partial_{\mathbf{n}}|\nabla u|^2 + \mathcal{H}|\nabla u|^2\right) - \int_\Gamma Z_{\theta,\xi}|\nabla u|^2, \end{aligned}$$

where

$$\left\{ \begin{array}{rcll} -\Delta u'_\theta &=& 0 & \Omega, \\ u'_\theta &=& 0 & \Gamma_D, \\ \partial_{\mathbf{n}} u'_\theta &=& 0 & \Gamma_N, \\ \partial_{\mathbf{n}} u'_\theta &=& -\mathrm{div}_\Gamma\left((\theta\cdot\mathbf{n})\nabla_\Gamma u\right) & \Gamma, \end{array} \right. \tag{7.21}$$

and

$$Z_{\theta,\xi} = \left((\xi_\Gamma\cdot\nabla_\Gamma)\,\mathbf{n}\right)\cdot\theta_\Gamma - \nabla_\Gamma(\theta\cdot\mathbf{n})\cdot\xi_\Gamma - \nabla_\Gamma(\xi\cdot\mathbf{n})\cdot\theta_\Gamma.$$

7.2.1 Coercivity, positiveness

In this section we are looking at the coercivity and the positiveness of the shape Hessian of the compliance. In [49], it is proven that there is an equivalence between coercivity and positiveness. We recall the corresponding result.

Theorem 7.2.1. *[49, Theorem 3.2] Let $\Omega^\star \in \mathcal{O}_5$ and J a shape functional, invariant with respect to a translation of the shape, twice shape-differentiable on a neighborhood of $\Omega^\star$ such that*

1. ***Structural hypotheses** : there exists $0 \le s_1 < s_2 \le 1$ such that*
   (a) *the bilinear map l_2 of the second-order derivative of J can be written under the form $l_2 = l_m + l_r$, where l_m is lower semi-continuous in $H^{s_2}(\partial\Omega^\star;\mathbb{R})$ and $\forall\varphi \in C^\infty(\partial\Omega^\star;\mathbb{R})$, $l_m(\varphi,\varphi) \ge c_1|\varphi|^2_{H^{s_2}}$ and l_r is continuous in H^{s_1}.*
   (b) *J satisfies $C_{H^{s_2}}$. This is a continuity assumption (see [49])*
2. ***Necessary optimality conditions** :*
   (a) *$\Omega^\star$ is a critical domain for J.*
   (b) *$\Omega^\star$ is a stable shape for J, that is such that*

$$\forall\varphi \in H^{s_2}(\partial\Omega;\mathbb{R})\backslash\{0\},\ l_2(\varphi,\varphi) > 0.$$

Then $\Omega^\star$ is a stable local minimum of J in a neighborhood of $\Omega^\star$ (in a precise sense defined by [49, Theorem 3.1]).

For the case of the compliance, the part of the second-order derivative that could correspond to l_m is the following.

$$\begin{array}{rcccl} l & : & \mathcal{F}\times\mathcal{F} & \longmapsto & \mathbb{R} \\ & & (\theta,\xi) & \mapsto & \displaystyle\int_\Omega \nabla u'_\theta\cdot\nabla u'_\xi. \end{array} \tag{7.22}$$

We recall that u'_θ is solution to (7.21). We are now looking to the coercivity of this bilinear map. We first establish that $\nabla_\Gamma u$ does not vanish on any bounded open set of Γ.

Lemma 7.2.2. *Assume that $\Omega \in \mathcal{O}_3$. We also assume that the solution u of (7.3) is in $H^2(\Omega;\mathbb{R})$. Let $x \in \Gamma$ such that $\nabla_\Gamma u(x) = 0$. Assume that there exists a neighborhood $\mathscr{V}$ of x in Γ such that*

$$\forall y \in \mathscr{V},\ \nabla_\Gamma u(y) = 0.$$

Then

$$u = 0 \quad \text{on } \Omega.$$

In order to prove this lemma, we recall a result from [39, Corollary 1.38].

Lemma 7.2.3. *Let Ω be a connected open set of class C^2. Let γ an open subset of $\partial\Omega$. Let $u \in H^2(\Omega;\mathbb{R})$ such that $\Delta u = 0$ in Ω and $u = \partial_{\boldsymbol{n}} u = 0$ on γ. Then u vanishes identically on Ω.*

Proof of Lemma 7.2.2. Let $\gamma = \mathcal{V} \cap \Gamma$. On γ we have by assumption

$$\begin{aligned} \partial_{\mathbf{n}} u &= 0, \\ \nabla_\Gamma u &= 0. \end{aligned}$$

The condition $\nabla_\Gamma u = 0$ ensures that u is constant on γ. Let C be the value of u on γ. Now we introduce $w = u - C$. We have $w \in H^2(\Omega;\mathbb{R})$ and

$$\begin{aligned} \Delta w &= 0 \quad \text{on} \quad \Omega, \\ w &= 0 \quad \text{on} \quad \gamma, \\ \partial_{\mathbf{n}} w &= 0 \quad \text{on} \quad \gamma. \end{aligned}$$

Then Lemma 7.2.3 ensures that $w = 0$ in Ω. In particular, on Γ_D we have $u = w = 0$. Therefore $C = 0$, and u vanishes identically on Ω. □

We also recall a result proven in [103].

Proposition 7.2.4. *Let $\mathcal{O}$ be a bounded open set of $\mathbb{R}^d$. Suppose that $u \in C^2(\overline{\Omega};\mathbb{R})$, $c \in C^1(\Omega;\mathbb{R})$ and $|\nabla u| \geq k_1$ in $\mathcal{O}$, where k_1 is positive. For any $f \in L^\infty(\Omega;\mathbb{R})$, consider the following problem with unknown α*

$$\nabla u \cdot \nabla \alpha + c(x)\alpha = f. \tag{7.23}$$

For any $P \in \mathcal{O}$, we denote $\mathscr{C}$ the characteristics of (7.23) containing P, and Q its root on Γ. The union of such point Q as P ranges over $\mathcal{O}$ is referred to as the inflow boundary Γ_1. Then (7.23) has a unique solution $\alpha \in C^1(\Omega;\mathbb{R})$ assuming prescribed values along Γ_1.

We denote by s the curvilinear abscissa along $\mathscr{C}$, and take $s = 0$ at P. We also denote by L the arc length of $\mathscr{C}$ between P and Q. The solution α can be expressed as

$$\alpha(P) = \alpha(Q) \exp\left(- \int_{-L}^{0} \frac{c}{|\nabla u|} ds \right) + \int_{-L}^{0} \frac{f}{|\nabla u|} \exp\left(- \int_{-s}^{0} \frac{c}{|\nabla u|} ds' \right) ds. \tag{7.24}$$

Proposition 7.2.5. *Assume that $\Gamma \subset \mathbb{R}^{d-1}$, and that there exists $k_1 > 0$ such that $|\nabla_\Gamma u| \geq k_1$ on Γ. Then the bilinear map l defined by (7.22) is positive on $\mathcal{F}$.*

Proof. Let $\theta \in \mathcal{F}$. Assume that $l(\theta,\theta) = 0$. Then we have

$$\int_\Omega |\nabla u'_\theta|^2 = 0.$$

As a result, we have $u'_\theta = 0$ in $H^1(\Omega;\mathbb{R})$. According to (7.21), we have

$$\operatorname{div}_\Gamma\Big((\theta \cdot \mathbf{n}) \nabla_\Gamma u \Big) = 0,$$

or equivalently

$$\nabla_\Gamma (\theta \cdot \mathbf{n}) \cdot \nabla_\Gamma u - (\theta \cdot \mathbf{n}) \partial^2_{\mathbf{nn}} u = 0.$$

Since $\Gamma \subset \mathbb{R}^{d-1}$, one can apply Proposition 7.2.4, with $\mathcal{O} = \Gamma$, $f = 0$ and $c = \Delta u$. The boundary condition on θ on the boundary of Γ is $\theta = 0$. Then the uniqueness of the solution of (7.23) ensures that

$$\theta = 0.$$

□

In the same framework, we prove the following result which is similar to Proposition 6.3.21 for thickness optimization.

Proposition 7.2.6. *Assume that $\Gamma \subset \mathbb{R}^{d-1}$, and that there exists $k_1 > 0$ such that $|\nabla_\Gamma u| \geq k_1$ on Γ, and that $\partial^2_{\boldsymbol{nn}} u$ is in $L^\infty(\Gamma;\mathbb{R})$. Then for θ_k a sequence of elements of $\mathcal{F}$ with*

$$\lim_{k\to\infty} l(\theta_k,\theta_k) = 0,$$

we have

$$\theta_{k|\Gamma} \longrightarrow 0 \text{ in } C^0\left(\Gamma;\mathbb{R}^d\right).$$

Proof. Let θ_k be sequence of elements of $\mathcal{F}$ such that

$$\lim_{k\to\infty} l(\theta_k,\theta_k) = 0.$$

Let

$$f_k = \nabla_\Gamma\left(\theta_k\cdot\mathbf{n}\right)\cdot\nabla_\Gamma u - \left(\theta_k\cdot\mathbf{n}\right)\partial^2_{\mathbf{nn}} u. \tag{7.25}$$

Considering $(\theta_k\cdot\mathbf{n})$ as an unknown, this equation has the same form as (7.23). Like it is explained in [103], the assumption $|\nabla_\Gamma u| \geq k_1 > 0$ ensures that for each point P of Γ there exists a characteristic for equation (7.25) containing P. Moreover, this assumption also implies that there is an upper bound L_M on the length of the characteristics.

Let $P \in \Gamma$, and $\mathscr{C}$ be the characteristic of θ_k through P. The origin of $\mathscr{C}$ is on the boundary of Γ where θ_k vanishes. Like in Proposition 7.2.4, we denote by s the curvilinear abscissa such that $s = 0$ at P, and L the arc length of $\mathscr{C}$ between P and its root. Then Proposition 7.2.4 ensures that the solution of (7.25) at P is given by

$$\theta_k(P) = \int_{-L}^{0} \frac{f_k}{|\nabla u|}\exp\left(-\int_{-s}^{0}\frac{\partial^2_{\mathbf{nn}}u}{|\nabla u|}ds'\right)ds,$$

Let $M = \left\|\partial^2_{\mathbf{nn}}u\right\|_{L^\infty(\Gamma;\mathbb{R})}$. We then have

$$|\theta_k(P)| \leq \|f_k\|_{L^\infty(\Gamma;\mathbb{R})}\,\frac{L_M}{k_1}\exp\left(\frac{ML_M}{k_1}\right).$$

We can observe that the bound for $|\theta_k(P)|$ is independent of P. Thus, for all $P \in \Gamma$ $\theta_k(P)$ converges uniformly to zero, which gives the desired result. □

When Γ is not included in $\mathbb{R}^{d-1}$, it means that this part of the boundary is not plane. According to [72, §5.4.1], when Ω is Lipschitz, its boundary can be represented by a set of graphs. For each one we can compute the corresponding change of variables, and (7.25) will conserve exactly the same structure and properties. Then the results on positiveness and coercivity of l are the same.

Remark 7.2.7. *The results established in this section are most theoretical than practical. Indeed, it is not clear that the requirement $|\nabla_\Gamma u| > k_1 > 0$ could be satisfied.*

7.2.2 Counter-example for the coercivity

In the case where ∇u vanishes at one point on Γ, we can prove that the bilinear map is not coercive on $H^{\frac{1}{2}}\left(\Gamma;\mathbb{R}^d\right)$. This is the subject of the following example.

Example 7.2.8. *Consider the case of $\mathbb{R}^d$ with $d = 2$. Let $\Omega = \mathbb{R}\times\mathbb{R}_+$ and $\Gamma = \mathbb{R}\times 0$. Let*

$$u = -\frac{1}{2}e^{-(x^2+y^2)}.$$

The function u is solution of

$$\begin{cases} \Delta u &= (2-2(x^2+y^2))e^{-(x^2+y^2)} & \text{on } \Omega \\ \partial_{\boldsymbol{n}} u &= 0 & \text{on } \Gamma \\ u &= 0 & \text{at } \infty \end{cases}$$

We have on Γ,

$$\nabla_\Gamma u = \begin{pmatrix} xe^{-x^2} \\ 0 \end{pmatrix}, \quad \boldsymbol{n} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Let v_n and θ_n defined on Γ by

$$v_n(x) = e^{-nx^2}, \quad \theta_n(x) = v_n(x)\boldsymbol{n}(x),$$

$$\operatorname{div}_\Gamma\Big(v_n\nabla_\Gamma u\Big) = \frac{d}{dx}\Big(v_n(x)xe^{-x^2}\Big).$$

Now we will prove that

$$\lim_{n\to\infty} \|v_n\|_{H^{1/2}(\Gamma;\mathbb{R})} \neq 0$$
$$\lim_{n\to\infty} \left\|\mathrm{div}_\Gamma\left(v_n \nabla_\Gamma u\right)\right\|_{H^{-1/2}(\Gamma;\mathbb{R})} = 0.$$

In order to compute the Sobolev norms, we have to compute different Fourier transforms.

$$\mathscr{F}\big(v_n\big)(\xi) = \sqrt{\frac{\pi}{n}} e^{-\frac{\xi^2}{4n}},$$

$$\begin{aligned}\mathrm{div}_\Gamma\big(v_n \nabla_\Gamma u\big) &= \frac{d}{dx}\left(v_n(x) x e^{-x^2}\right) \\ &= -\frac{1}{2(n+1)} \frac{d^2}{dx^2}\left(e^{-(n+1)x^2}\right),\end{aligned}$$

$$\mathscr{F}\Big(\mathrm{div}_\Gamma\big(v_n \nabla_\Gamma u\big)\Big)(\xi) = \frac{\xi^2}{2(n+1)} \sqrt{\frac{\pi}{n+1}} e^{-\frac{\xi^2}{4(n+1)}}.$$

Then we have

$$\begin{aligned}\|v_n\|^2_{H^{1/2}(\Gamma;\mathbb{R})} &= \int_{-\infty}^{+\infty} (1+\xi^2)^{\frac{1}{2}} \left|\mathscr{F}((\theta_n \cdot \boldsymbol{n}))(\xi)\right|^2 d\xi \\ &= \frac{\pi}{n} \int_{-\infty}^{+\infty} (1+\xi^2)^{\frac{1}{2}} e^{-\frac{\xi^2}{2n}} d\xi \\ &= \pi \int_{-\infty}^{+\infty} \left(\frac{1}{n} + z^2\right)^{\frac{1}{2}} e^{-\frac{z^2}{2}} dz,\end{aligned}$$

where we set $\xi = \sqrt{n} z$. *We also have*

$$\begin{aligned}\left\|\mathrm{div}_\Gamma\left(v_n \nabla_\Gamma u\right)\right\|^2_{H^{-1/2}(\Gamma;\mathbb{R})} &= \frac{\pi}{4(n+1)^3} \int_{-\infty}^{+\infty} (1+\xi^2)^{-\frac{1}{2}} \xi^4 e^{-\frac{\xi^2}{2(n+1)}} d\xi \\ &= \frac{\pi}{4\sqrt{n+1}} \int_{-\infty}^{+\infty} (1+(n+1)z^2)^{-\frac{1}{2}} z^4 e^{-\frac{z^2}{2}} dz,\end{aligned}$$

with the change of variable $\xi = \sqrt{n+1} z$. *Now with dominated convergence we easily see that*

$$\lim_{n\to\infty} \|v_n\|_{H^{1/2}(\Gamma;\mathbb{R})} = 2\pi \int_0^{+\infty} z e^{-\frac{z^2}{2}} dz = 2\pi$$
$$\lim_{n\to\infty} \left\|\mathrm{div}_\Gamma\big(v_n \nabla_\Gamma u\big)\right\|_{H^{-1/2}(\Gamma;\mathbb{R})} = 0.$$

We deduce from the definition of u'_θ *by (7.21) that*

$$\lim_{n\to\infty} \left\|u'_{\theta_n}\right\|_{H^1(\Gamma;\mathbb{R})} = 0,$$

and from the definition of l *that*

$$\lim_{n\to\infty} l(\theta_n, \theta_n) = \lim_{n\to\infty} \int_\Omega |\nabla u'_{\theta_n}|^2 = 0.$$

Since $\lim_{n\to\infty} \|(\theta_n \cdot \boldsymbol{n})\|_{H^{1/2}(\Gamma;\mathbb{R})} \neq 0$ *we conclude that* l *is not coercive on* $H^{\frac{1}{2}}(\Gamma;\mathbb{R}^d)$.

7.2.3 Newton's equation and tangential components

Solving Newton's equation is equivalent to minimizing the quadratic approximation of an objective function. In this section we consider a twice shape-differentiable function J. According to Theorem 3.2.15 for $\Omega \in \mathcal{O}_3$, the shape derivatives of J in the directions $\theta, \xi \in C^{2,\infty}\left(\mathbb{R}^d;\mathbb{R}^d\right)$ are given by

$$\begin{cases} J'(\Omega;\theta) &= l_1(\theta \cdot \mathbf{n}), \\ J''(\Omega;\theta,\xi) &= l_2(\theta \cdot \mathbf{n}, \xi \cdot \mathbf{n}) + l_1(Z_{\theta,\xi}), \end{cases}$$

where

$$Z_{\theta,\xi} = \big(\left(\xi_\Gamma \cdot \nabla_\Gamma\right)\mathbf{n}\big) \cdot \theta_\Gamma - \nabla_\Gamma\left(\theta \cdot \mathbf{n}\right) \cdot \xi_\Gamma - \nabla_\Gamma\left(\xi \cdot \mathbf{n}\right) \cdot \theta_\Gamma.$$

Now, in view of minimizing J with the Newton method, the Newton equation for $\theta \in C^{2,\infty}\left(\mathbb{R}^d;\mathbb{R}^d\right)$ reads

$$\forall \xi \in C^{2,\infty}\left(\mathbb{R}^d;\mathbb{R}^d\right),\ J''(\Omega;\theta,\xi) = -J'(\Omega;\xi).$$

Lemma 7.2.9. *Let $\Omega \in \mathcal{O}_3$ and J be twice shape-differentiable. The Newton equation is equivalent to*

$$\forall \xi \in C^{2,\infty}\left(\mathbb{R}^d;\mathbb{R}^d\right), \begin{cases} l_2(\theta\cdot\boldsymbol{n},\xi\cdot\boldsymbol{n}) + l_1\big(\nabla_\Gamma(\xi\cdot\boldsymbol{n})\cdot\theta_\Gamma\big) = -l_1(\xi\cdot\boldsymbol{n}), \\ l_1\Big(\xi_\Gamma\cdot\big(\left(\theta_\Gamma\cdot\nabla_\Gamma\right)\boldsymbol{n} - \nabla_\Gamma\left(\theta\cdot\boldsymbol{n}\right)\big)\Big) = 0. \end{cases} \tag{7.26}$$

Proof. The Newton equation reads

$$\forall \xi \in C^{2,\infty}\left(\mathbb{R}^d;\mathbb{R}^d\right), l_2(\theta\cdot\mathbf{n},\xi\cdot\mathbf{n}) + l_1(Z_{\theta,\xi}) = -l_1(\xi\cdot\mathbf{n}).$$

We start by writing it with $\xi \in C^{2,\infty}\left(\mathbb{R}^d;\mathbb{R}^d\right)$ such that $\xi\cdot\mathbf{n} = 0$. This leads to

$$l_1(Z_{\theta,\xi}) = 0.$$

More particularly, since $\xi\cdot\mathbf{n} = 0$ and $\nabla_\Gamma\mathbf{n}$ is symmetric we have for all $\xi \in C^{2,\infty}\left(\mathbb{R}^d;\mathbb{R}^d\right)$ with $\xi\cdot\mathbf{n} = 0$

$$l_1\Big(\xi_\Gamma\cdot\big(\left(\theta_\Gamma\cdot\nabla_\Gamma\right)\mathbf{n} - \nabla_\Gamma\left(\theta\cdot\mathbf{n}\right)\big)\Big) = 0,$$

which is the second equation we are looking for. The first equality is then straightforward with this result. □

Remark 7.2.10. *In the previous lemma, it is interesting to focus on the second equation of (7.26). In particular we have for every $\xi_\Gamma \in C^{2,\infty}\left(\partial\Omega;\mathbb{R}^d\right)$,*

$$l_1\Big(\xi_\Gamma\cdot\big(\left(\theta_\Gamma\cdot\nabla_\Gamma\right)\boldsymbol{n} - \nabla_\Gamma\left(\theta\cdot\boldsymbol{n}\right)\big)\Big) = 0.$$

This means that the vector $\left(\theta_\Gamma\cdot\nabla_\Gamma\right)\boldsymbol{n} - \nabla_\Gamma\left(\theta\cdot\boldsymbol{n}\right)$ should be in the kernel of l_1. When this kernel reduces to $\{0\}$, this means that Moreover, when the linear map l_1 is of the form

$$\forall v \in C^1(\partial\Omega;\mathbb{R}),\ l_1(v) = \int_{\partial\Omega} jv,$$

it means that this vector vanishes whenever j is non-zero.

$$\left(\theta_\Gamma\cdot\nabla_\Gamma\right)\boldsymbol{n} = \nabla_\Gamma\left(\theta\cdot\boldsymbol{n}\right). \tag{7.27}$$

This is a geometric condition between the tangent and the normal components - namely θ_Γ and $\theta\cdot\boldsymbol{n}$ - which does not depend on the criterion but only on the shape trough the outer normal $\boldsymbol{n}$.

In the case of the compliance with u solution to (7.3), Lemma 7.2.2 ensures that u cannot vanish on open subsets of the boundary. This means that $|\nabla u|$ - which defines the first-order derivative - may vanish only on isolated points. Therefore, equation (7.27) is satisfied for a solution to the Newton equation.

This property is interesting more from a theoretical than from a practical point of view. It is a priori not obvious to have such a relation between the tangential and normal components of the directions. Moreover, when the kernel of the linear map l_1 reduces to $\{0\}$, this relation depends only on the shape and not on the criterion. It is not really used in practice since it is quite difficult to compute numerically the tangential components of directions. We will also see later in Chapter 10 that we can consider a framework where only normal evolutions of the shape are considered.

7.3 Appendix

7.3.1 Computation of the shape Hessian

Proposition 7.3.1. *For every θ,ξ in $C^{2,\infty}(\Omega;\mathbb{R}^d)$ the shape Hessian of the compliance in the directions θ,ξ reads*

$$\begin{aligned} J_1{}''\left(\Omega;\theta,\xi\right) =& -2\int_{\Gamma_N\cup\Gamma}\left(\theta\cdot\boldsymbol{n}\right)\nabla u'_\xi\cdot\nabla u + 2\int_{\Gamma_D}\left(\theta\cdot\boldsymbol{n}\right)\nabla u'_\xi\cdot\nabla u \\ &+ 2\int_{\Gamma_N\cup\Gamma}\left(\theta\cdot\boldsymbol{n}\right)u'_\xi(f+\mathcal{H}g+\partial_{\boldsymbol{n}}g) + 2\int_{\Gamma_N}\left(\theta\cdot\boldsymbol{n}\right)g\nabla_\Gamma u\cdot\nabla_\Gamma\left(\xi\cdot\boldsymbol{n}\right) \\ &+ 2\int_{\Gamma_N\cup\Gamma}\left(\theta\cdot\boldsymbol{n}\right)\left(\xi\cdot\boldsymbol{n}\right)\Big(\mathcal{H}\big(fu+\mathcal{H}gu+\partial_{\boldsymbol{n}}(gu)\big)+\partial_{\boldsymbol{n}}\big(fu+\mathcal{H}gu+\partial_{\boldsymbol{n}}(gu)\big)\Big) \\ &+ 2\int_{\Gamma_N}\left(\theta\cdot\boldsymbol{n}\right)\left(\xi\cdot\boldsymbol{n}\right)g\Big(\partial_{\boldsymbol{n}}g-\partial^2_{\boldsymbol{nn}}u\Big) + 2\int_{\Gamma_N}gu\,\nabla_\Gamma\left(\theta\cdot\boldsymbol{n}\right)\cdot\nabla\left(\xi\cdot\boldsymbol{n}\right) \\ &- \int_{\Gamma_N\cup\Gamma}\left(\theta\cdot\boldsymbol{n}\right)\left(\xi\cdot\boldsymbol{n}\right)\Big(\partial_{\boldsymbol{n}}|\nabla u|^2+\mathcal{H}|\nabla u|^2\Big) + \int_{\Gamma_D}\left(\theta\cdot\boldsymbol{n}\right)\left(\xi\cdot\boldsymbol{n}\right)\Big(\partial_{\boldsymbol{n}}|\nabla u|^2+\mathcal{H}|\nabla u|^2\Big) \\ &+ 2\int_{\Gamma_N\cup\Gamma}Z_{\theta,\xi}\Big(\partial_{\boldsymbol{n}}(gu)+fu+\mathcal{H}gu\Big) + \int_{\Gamma_D}Z_{\theta,\xi}|\nabla u|^2 - \int_{\Gamma_N\cup\Gamma}Z_{\theta,\xi}|\nabla u|^2. \end{aligned}$$

Proof. We start from the first-order shape derivative of the compliance

$$J_1{}'(\Omega;\theta) = 2\int_{\Gamma_N\cup\Gamma}(\theta\cdot\mathbf{n})\Big(\partial_{\mathbf{n}}(gu)+fu+\mathcal{H}gu\Big) + \int_{\Gamma_D}|\nabla u|^2(\theta\cdot\mathbf{n}) - \int_{\Gamma_N\cup\Gamma}|\nabla u|^2(\theta\cdot\mathbf{n}),$$

and we introduce

$$\begin{aligned} J_a{}'(\Omega;\theta) &= \int_{\Gamma_N\cup\Gamma}(\theta\cdot\mathbf{n})\Big(\partial_{\mathbf{n}}(gu)+fu+\mathcal{H}gu\Big),\\ J_b{}'(\Omega;\theta) &= \int_{\Gamma_D}|\nabla u|^2(\theta\cdot\mathbf{n}),\\ J_c{}'(\Omega;\theta) &= -\int_{\Gamma_N}|\nabla u|^2(\theta\cdot\mathbf{n}), \end{aligned}$$

and

$$\begin{aligned} J_a{}''(\Omega;\theta,\xi) &= \Big(J_a{}'(\Omega;\theta)\Big)'(\Omega;\xi) - J_a{}'(\Omega;(\xi\cdot\nabla)\theta),\\ J_b{}''(\Omega;\theta,\xi) &= \Big(J_b{}'(\Omega;\theta)\Big)'(\Omega;\xi) - J_b{}'(\Omega;(\xi\cdot\nabla)\theta),\\ J_c{}''(\Omega;\theta,\xi) &= \Big(J_c{}'(\Omega;\theta)\Big)'(\Omega;\xi) - J_c{}'(\Omega;(\xi\cdot\nabla)\theta). \end{aligned}$$

Then we use equation (3.14), and the linearity of J_a', J_b', J_c' to compute the second-order shape derivative.

$$\Big(J_1{}'(\Omega;\theta)\Big)'(\Omega;\xi) = 2\Big(J_a{}'(\Omega;\theta)\Big)'(\Omega;\xi) + \Big(J_b{}'(\Omega;\theta)\Big)'(\Omega;\xi) + \Big(J_c{}'(\Omega;\theta)\Big)'(\Omega;\xi).$$

Case of J_a'. Deriving J_a' leads to

$$\begin{aligned} \Big(J_a{}'(\Omega;\theta)\Big)'(\Omega;\xi) = &\int_{\Gamma_N\cup\Gamma}\big(\theta\cdot\mathbf{n}_\xi'\big)\big(fu+\mathcal{H}gu+\partial_{\mathbf{n}}(gu)\big)\\ &+\int_{\Gamma_N\cup\Gamma}(\theta\cdot\mathbf{n})\Big(fu_\xi'+\mathcal{H}gu_\xi'+\partial_{\mathbf{n}}(gu_\xi')+\nabla(gu)\cdot\mathbf{n}_\xi'+\mathcal{H}_\xi'gu\Big)\\ &+\int_{\Gamma_N\cup\Gamma}(\xi\cdot\mathbf{n})\Big(\mathcal{H}(\theta\cdot\mathbf{n})\Big(f+\mathcal{H}gu+\partial_{\mathbf{n}}(gu)\Big)+\partial_{\mathbf{n}}\Big((\theta\cdot\mathbf{n})\big(f+\mathcal{H}gu+\partial_{\mathbf{n}}(gu)\big)\Big)\Big)\\ &-\int_{\Gamma_N\cup\Gamma}\Big(\big((\xi\cdot\nabla)\theta\big)\cdot\mathbf{n}\Big)\Big(f+\mathcal{H}gu+\partial_{\mathbf{n}}(gu)\Big). \end{aligned}$$

Since $\mathcal{H}=\operatorname{div}(\mathbf{n})$, we have $\mathcal{H}_\xi'=\operatorname{div}(\mathbf{n}_\xi')$. From Proposition 3.3.22, we then have

$$\begin{aligned} \mathbf{n}_\xi' &= -\nabla_\Gamma(\xi\cdot\mathbf{n}),\\ \mathcal{H}_\xi' &= -\operatorname{div}(\nabla_\Gamma(\xi\cdot\mathbf{n})) = -\Delta_\Gamma(\xi\cdot\mathbf{n}),\\ \big(\xi\cdot\nabla\theta\big)\cdot\mathbf{n} &= (\xi\cdot\mathbf{n})\,\partial_{\mathbf{n}}(\theta\cdot\mathbf{n}) + \xi_\Gamma\cdot\nabla_\Gamma(\theta\cdot\mathbf{n}) - \nabla_\Gamma\mathbf{n}\xi_\Gamma\cdot\theta_\Gamma. \end{aligned}$$

Using Lemma 7.1.5 and introducing

$$Z_{\theta,\xi} = \big((\xi_\Gamma\cdot\nabla_\Gamma)\,\mathbf{n}\big)\cdot\theta_\Gamma - \nabla_\Gamma(\theta\cdot\mathbf{n})\cdot\xi_\Gamma - \nabla_\Gamma(\xi\cdot\mathbf{n})\cdot\theta_\Gamma,$$

we have

$$\begin{aligned} J_a{}''(\Omega;\theta,\xi) = &\int_{\Gamma_N\cup\Gamma}Z_{\theta,\xi}\Big(fu+\mathcal{H}gu+\partial_{\mathbf{n}}(gu)\Big)\\ &+\int_{\Gamma_N\cup\Gamma}(\xi\cdot\mathbf{n})(\theta\cdot\mathbf{n})\Big(\mathcal{H}\big(f+\mathcal{H}gu+\partial_{\mathbf{n}}(gu)\big)+\partial_{\mathbf{n}}\big(f+\mathcal{H}gu+\partial_{\mathbf{n}}(gu)\big)\Big)\\ &+\underbrace{\int_{\Gamma_N\cup\Gamma}(\theta\cdot\mathbf{n})\Big(fu_\xi'+\mathcal{H}gu_\xi'+\partial_{\mathbf{n}}(gu_\xi')-\nabla(gu)\cdot\nabla_\Gamma(\xi\cdot\mathbf{n})-\Delta_\Gamma(\xi\cdot\mathbf{n})gu\Big)}_{J_d''}. \end{aligned}$$

Now we consider the last term of J_a'', and use expression

$$\partial_{\mathbf{n}}u_\xi' = (\xi\cdot\mathbf{n})\big(\partial_{\mathbf{n}}g-\partial_{\mathbf{nn}}^2u\big)+\nabla_\Gamma u\cdot\nabla_\Gamma((\xi\cdot\mathbf{n})),$$

to get

$$J_d'' = \int_{\Gamma_N\cup\Gamma} (\theta\cdot\mathbf{n})\left(fu_\xi' + \mathcal{H}gu_\xi' + u_\xi'\partial_{\mathbf{n}}g + g\,(\xi\cdot\mathbf{n})\left(\partial_{\mathbf{n}}g - \partial^2_{\mathbf{nn}}u\right)\right) - \int_{\Gamma_N\cup\Gamma} (\theta\cdot\mathbf{n})\left(u\nabla g\cdot\nabla_\Gamma\left((\xi\cdot\mathbf{n})\right) + \Delta_\Gamma\,(\xi\cdot\mathbf{n})\,gu\right).$$

Integrating by parts the term with the Laplace-Beltrami operator, we obtain

$$-\int_{\partial\Omega} (\theta\cdot\mathbf{n})\,\Delta_\Gamma\,(\xi\cdot\mathbf{n})\,gu = \int_{\partial\Omega} \nabla_\Gamma\left((\theta\cdot\mathbf{n})\,gu\right)\cdot\nabla_\Gamma\left((\xi\cdot\mathbf{n})\right).$$

Since $u=0$ on Γ_D, $\nabla_\Gamma u$ vanishes as well on Γ_D. Then the previous equality is available on $\Gamma_N\cup\Gamma$ instead of $\partial\Omega$:

$$-\int_{\Gamma_N\cup\Gamma} (\theta\cdot\mathbf{n})\,\Delta_\Gamma\,(\xi\cdot\mathbf{n})\,gu = \int_{\Gamma_N\cup\Gamma} gu\nabla_\Gamma\,(\theta\cdot\mathbf{n})\cdot\nabla_\Gamma\,(\xi\cdot\mathbf{n}) + \int_{\Gamma_N\cup\Gamma} g\,(\theta\cdot\mathbf{n})\,\nabla_\Gamma u\cdot\nabla_\Gamma\,(\xi\cdot\mathbf{n}) + u\,(\theta\cdot\mathbf{n})\,\nabla_\Gamma g\cdot\nabla_\Gamma\xi\cdot\mathbf{n}.$$

As a result, we can rewrite J_d'' as

$$J_d'' = \int_{\Gamma_N\cup\Gamma} (\theta\cdot\mathbf{n})\,u_\xi'\Big(f + \mathcal{H}g + \partial_{\mathbf{n}}g\Big) + \int_{\Gamma_N\cup\Gamma} (\theta\cdot\mathbf{n})\,(\xi\cdot\mathbf{n})\,g\Big(\partial_{\mathbf{n}}g - \partial^2_{\mathbf{nn}}u\Big) + \int_{\Gamma_N\cup\Gamma} gu\nabla_\Gamma\,(\theta\cdot\mathbf{n})\cdot\nabla_\Gamma\,(\xi\cdot\mathbf{n}) + \int_{\Gamma_N\cup\Gamma} g\,(\theta\cdot\mathbf{n})\,\nabla_\Gamma u\cdot\nabla_\Gamma\,(\xi\cdot\mathbf{n}),$$

and J_a'' as

$$\begin{aligned} J_a{}''\,(\Omega;\theta,\xi) = &\int_{\Gamma_N\cup\Gamma} Z_{\theta,\xi}\Big(fu + \mathcal{H}gu + \partial_{\mathbf{n}}(gu)\Big) \\ &+ \int_{\Gamma_N\cup\Gamma} (\xi\cdot\mathbf{n})\,(\theta\cdot\mathbf{n})\left(\mathcal{H}\big(f + \mathcal{H}gu + \partial_{\mathbf{n}}(gu)\big) + \partial_{\mathbf{n}}\big(f + \mathcal{H}gu + \partial_{\mathbf{n}}(gu)\big)\right) \\ &+ \int_{\Gamma_N\cup\Gamma} (\theta\cdot\mathbf{n})\,u_\xi'\Big(f + \mathcal{H}g + \partial_{\mathbf{n}}g\Big) + \int_{\Gamma_N\cup\Gamma} (\theta\cdot\mathbf{n})\,(\xi\cdot\mathbf{n})\,g\Big(\partial_{\mathbf{n}}g - \partial^2_{\mathbf{nn}}u\Big) \\ &+ \int_{\Gamma_N\cup\Gamma} gu\nabla_\Gamma\,(\theta\cdot\mathbf{n})\cdot\nabla_\Gamma\,(\xi\cdot\mathbf{n}) + \int_{\Gamma_N\cup\Gamma} g\,(\theta\cdot\mathbf{n})\,\nabla_\Gamma u\cdot\nabla_\Gamma\,(\xi\cdot\mathbf{n}). \end{aligned}$$

Case of J_b' and J_3'. The derivation of these terms is done in the proof of Proposition 7.1.8. We find

$$J_b{}''\,(\Omega;\theta,\xi) = -\int_{\Gamma_N\cup\Gamma} Z_{\theta,\xi}|\nabla u|^2 - 2\int_{\Gamma_N\cup\Gamma} (\theta\cdot\mathbf{n})\,\nabla u_\xi'\cdot\nabla u - \int_{\Gamma_N\cup\Gamma} (\theta\cdot\mathbf{n})\,(\xi\cdot\mathbf{n})\left(\partial_{\mathbf{n}}|\nabla u|^2 + \mathcal{H}|\nabla u|^2\right),$$

and

$$J_c{}''\,(\Omega;\theta,\xi) = \int_{\Gamma_D} Z_{\theta,\xi}|\nabla u|^2 + 2\int_{\Gamma_D} (\theta\cdot\mathbf{n})\,\nabla u_\xi'\cdot\nabla u + \int_{\Gamma_D} (\theta\cdot\mathbf{n})\,(\xi\cdot\mathbf{n})\left(\partial_{\mathbf{n}}|\nabla u|^2 + \mathcal{H}|\nabla u|^2\right).$$

Finally

$$\begin{aligned} J_1{}''\,(\Omega;\theta,\xi) = &\,2\int_{\Gamma_N\cup\Gamma} Z_{\theta,\xi}\Big(fu + \mathcal{H}gu + \partial_{\mathbf{n}}(gu)\Big) + \int_{\Gamma_D} Z_{\theta,\xi}|\nabla u|^2 - \int_{\Gamma_N\cup\Gamma} Z_{\theta,\xi}|\nabla u|^2 \\ &+ 2\int_{\Gamma_N\cup\Gamma} (\xi\cdot\mathbf{n})\,(\theta\cdot\mathbf{n})\left(\mathcal{H}\big(f + \mathcal{H}gu + \partial_{\mathbf{n}}(gu)\big) + \partial_{\mathbf{n}}\big(f + \mathcal{H}gu + \partial_{\mathbf{n}}(gu)\big)\right) \\ &+ 2\int_{\Gamma_N\cup\Gamma} (\theta\cdot\mathbf{n})\,(\xi\cdot\mathbf{n})\,g\Big(\partial_{\mathbf{n}}g - \partial^2_{\mathbf{nn}}u\Big) + 2\int_{\Gamma_N\cup\Gamma} gu\nabla_\Gamma\,(\theta\cdot\mathbf{n})\cdot\nabla_\Gamma\,(\xi\cdot\mathbf{n}) \\ &- \int_{\Gamma_N\cup\Gamma} (\theta\cdot\mathbf{n})\,(\xi\cdot\mathbf{n})\left(\partial_{\mathbf{n}}|\nabla u|^2 + \mathcal{H}|\nabla u|^2\right) + \int_{\Gamma_D} (\theta\cdot\mathbf{n})\,(\xi\cdot\mathbf{n})\left(\partial_{\mathbf{n}}|\nabla u|^2 + \mathcal{H}|\nabla u|^2\right) \\ &- 2\int_{\Gamma_N\cup\Gamma} (\theta\cdot\mathbf{n})\,\nabla u_\xi'\cdot\nabla u + 2\int_{\Gamma_D} (\theta\cdot\mathbf{n})\,\nabla u_\xi'\cdot\nabla u \\ &+ 2\int_{\Gamma_N\cup\Gamma} (\theta\cdot\mathbf{n})\,u_\xi'\Big(f + \mathcal{H}g + \partial_{\mathbf{n}}g\Big) + 2\int_{\Gamma_N\cup\Gamma} g\,(\theta\cdot\mathbf{n})\,\nabla_\Gamma u\cdot\nabla_\Gamma\,(\xi\cdot\mathbf{n}). \end{aligned} \tag{7.28}$$

□

7.3.2 Symmetry of the shape Hessian

We consider here the general case, with no particular assumption on f or on the vanishing of the derivation directions.

Proof of Proposition 7.1.7. The shape Hessian of the compliance reads :

$$\begin{aligned}
J_1{}''(\Omega;\theta,\xi) = {} & 2\int_\Omega \nabla u'_\xi\cdot\nabla u'_\theta + 2\int_{\Gamma_N} gu\,\nabla_\Gamma(\theta\cdot\mathbf{n})\cdot\nabla_\Gamma(\xi\cdot\mathbf{n}) - 2\int_{\Gamma_D}\Big(u'_\xi\partial_{\mathbf{n}}u'_\theta + u'_\theta\partial_{\mathbf{n}}u'_\xi\Big)\\
& + 2\int_{\Gamma_N\cup\Gamma}(\theta\cdot\mathbf{n})(\xi\cdot\mathbf{n})\Big(\mathcal{H}\big(fu+\mathcal{H}gu+\partial_{\mathbf{n}}(gu)\big)+\partial_{\mathbf{n}}\big(fu+\mathcal{H}gu+\partial_{\mathbf{n}}(gu)\big)\Big)\\
& - \int_{\Gamma_N\cup\Gamma}(\theta\cdot\mathbf{n})(\xi\cdot\mathbf{n})\Big(\partial_{\mathbf{n}}|\nabla u|^2+\mathcal{H}|\nabla u|^2\Big)+\int_{\Gamma_D}(\theta\cdot\mathbf{n})(\xi\cdot\mathbf{n})\Big(\partial_{\mathbf{n}}|\nabla u|^2+\mathcal{H}|\nabla u|^2\Big)\\
& + 2\int_{\Gamma_N\cup\Gamma} Z(\theta,\xi)\Big(\partial_{\mathbf{n}}(gu)+fu+\mathcal{H}gu\Big)+\int_{\Gamma_D}|\nabla u|^2 Z_{\theta,\xi}-\int_{\Gamma_N\cup\Gamma}|\nabla u|^2 Z_{\theta,\xi}.
\end{aligned}$$

Since

$$Z_{\theta,\xi} = \big((\xi_\Gamma\cdot\nabla_\Gamma)\,\mathbf{n}\big)\cdot\theta_\Gamma - \nabla_\Gamma(\theta\cdot\mathbf{n})\cdot\xi_\Gamma - \nabla_\Gamma(\xi\cdot\mathbf{n})\cdot\theta_\Gamma,$$

we see that $Z_{\theta,\xi}$ is symmetric. Only the terms in the last two lines of (7.28), may be non-symmetric. Let

$$\begin{aligned}
J_{ns} = {} & -\int_{\Gamma_N\cup\Gamma}(\theta\cdot\mathbf{n})\,\nabla u'_\xi\cdot\nabla u+\int_{\Gamma_D}(\theta\cdot\mathbf{n})\,\nabla u'_\xi\cdot\nabla u\\
& +\int_{\Gamma_N\cup\Gamma}(\theta\cdot\mathbf{n})\,u'_\xi\Big(f+\mathcal{H}g+\partial_{\mathbf{n}}g\Big)+\int_{\Gamma_N\cup\Gamma} g\,(\theta\cdot\mathbf{n})\,\nabla_\Gamma u\cdot\nabla_\Gamma(\xi\cdot\mathbf{n}).
\end{aligned}$$

The aim is now to prove that J_{ns} is symmetric. Let

$$J_{ns1} = -\int_{\Gamma_N\cup\Gamma}(\theta\cdot\mathbf{n})\,\nabla u'_\xi\cdot\nabla u.$$

We recall that

$$\begin{aligned}
\partial_{\mathbf{n}}u'_\xi &= (\xi\cdot\mathbf{n})\Big(\partial_{\mathbf{n}}g-\partial^2_{\mathbf{nn}}u\Big)+\nabla_\Gamma u\cdot\nabla_\Gamma(\xi\cdot\mathbf{n})\\
&= \mathrm{div}_\Gamma\Big((\xi\cdot\mathbf{n})\,\nabla_\Gamma u\Big)+(\xi\cdot\mathbf{n})\Big(\partial_{\mathbf{n}}g-\Delta u+\mathcal{H}\partial_{\mathbf{n}}u\Big).
\end{aligned}$$

Then

$$J_{ns1} = -\int_{\Gamma_N\cup\Gamma}(\theta\cdot\mathbf{n})\,g\partial_{\mathbf{n}}u'_\xi-\int_{\Gamma_N\cup\Gamma}(\theta\cdot\mathbf{n})\,\nabla_\Gamma u'_\xi\cdot\nabla_\Gamma u.$$

But $u'_\xi = 0$ and $u=0$ on Γ_D. Therefore $\nabla_\Gamma u = 0$ on Γ_D and

$$\begin{aligned}
\int_{\Gamma_N\cup\Gamma}(\theta\cdot\mathbf{n})\,\nabla_\Gamma u'_\xi\cdot\nabla_\Gamma u &= \int_{\partial\Omega}(\theta\cdot\mathbf{n})\,\nabla_\Gamma u'_\xi\cdot\nabla_\Gamma u\\
&= -\int_{\partial\Omega} u'_\xi\mathrm{div}_\Gamma\Big((\theta\cdot\mathbf{n})\,\nabla_\Gamma u\Big)\\
&= -\int_{\Gamma_N\cup\Gamma} u'_\xi\mathrm{div}_\Gamma\Big((\theta\cdot\mathbf{n})\,\nabla_\Gamma u\Big)\\
&= -\int_{\Gamma_N\cup\Gamma} u'_\xi\partial_{\mathbf{n}}u'_\theta+\int_{\Gamma_N\cup\Gamma} u'_\xi\,(\theta\cdot\mathbf{n})\Big(\partial_{\mathbf{n}}g+f+\mathcal{H}g\Big).
\end{aligned}$$

As a result

$$\begin{aligned}
J_{ns1} = {} & -\int_{\Gamma_N\cup\Gamma} g\,(\theta\cdot\mathbf{n})\,(\xi\cdot\mathbf{n})\Big(\partial_{\mathbf{n}}g-\partial^2_{\mathbf{nn}}u\Big)-\int_{\Gamma_N\cup\Gamma} g\,(\theta\cdot\mathbf{n})\,\nabla_\Gamma u\cdot\nabla_\Gamma(\xi\cdot\mathbf{n})\\
& +\int_{\Gamma_N\cup\Gamma} u'_\xi\partial_{\mathbf{n}}u'_\theta-\int_{\Gamma_N\cup\Gamma} u'_\xi\,(\theta\cdot\mathbf{n})\Big(\partial_{\mathbf{n}}g+f+\mathcal{H}g\Big),
\end{aligned}$$

and

$$\begin{aligned}
J_{ns} = {} & \int_{\Gamma_D}(\theta\cdot\mathbf{n})\,\nabla u'_\xi\cdot\nabla u\\
& +\int_{\Gamma_N\cup\Gamma}(\theta\cdot\mathbf{n})\,u'_\xi\Big(f+\mathcal{H}g+\partial_{\mathbf{n}}g\Big)+\int_{\Gamma_N\cup\Gamma} g\,(\theta\cdot\mathbf{n})\,\nabla_\Gamma u\cdot\nabla_\Gamma(\xi\cdot\mathbf{n})\\
& -\int_{\Gamma_N\cup\Gamma} g\,(\theta\cdot\mathbf{n})\,(\xi\cdot\mathbf{n})\Big(\partial_{\mathbf{n}}g-\partial^2_{\mathbf{nn}}u\Big)-\int_{\Gamma_N\cup\Gamma} g\,(\theta\cdot\mathbf{n})\,\nabla_\Gamma u\cdot\nabla_\Gamma(\xi\cdot\mathbf{n})\\
& +\int_{\Gamma_N\cup\Gamma} u'_\xi\partial_{\mathbf{n}}u'_\theta-\int_{\Gamma_N\cup\Gamma} u'_\xi\,(\theta\cdot\mathbf{n})\Big(\partial_{\mathbf{n}}g+f+\mathcal{H}g\Big),
\end{aligned}$$

which leads to

$$J_{ns} = \int_{\Gamma_D} (\theta \cdot \mathbf{n}) \nabla u'_\xi \cdot \nabla u + \int_{\Gamma_N \cup \Gamma} u'_\xi \partial_{\mathbf{n}} u'_\theta - \int_{\Gamma_N \cup \Gamma} g (\theta \cdot \mathbf{n}) (\xi \cdot \mathbf{n}) \left(\partial_{\mathbf{n}} g - \partial^2_{\mathbf{nn}} u \right).$$

Now, we see that

$$\begin{aligned} \int_{\Gamma_N \cup \Gamma} u'_\xi \partial_{\mathbf{n}} u'_\theta &= \int_{\partial\Omega} u'_\xi \partial_{\mathbf{n}} u'_\theta - \int_{\Gamma_D} u'_\xi \partial_{\mathbf{n}} u'_\theta \\ &= \int_\Omega \nabla u'_\theta \cdot \nabla u'_\xi + \int_\Omega u'_\xi \Delta u'_\theta - \int_{\Gamma_D} u'_\xi \partial_{\mathbf{n}} u'_\theta. \end{aligned}$$

We also know the Dirichlet boundary condition on u'_θ : $u'_\theta = -(\theta \cdot \mathbf{n}) \partial_{\mathbf{n}} u$. Then, with $\nabla_\Gamma u = 0$ on Γ_D,

$$\begin{aligned} \int_{\Gamma_D} (\theta \cdot \mathbf{n}) \nabla u'_\theta \cdot \nabla u &= \int_{\Gamma_D} (\theta \cdot \mathbf{n}) \partial_{\mathbf{n}} u'_\xi \partial_{\mathbf{n}} u \\ &= - \int_{\Gamma_D} u'_\theta \partial_{\mathbf{n}} u'_\xi. \end{aligned}$$

As a result

$$J_{ns} = - \int_{\Gamma_N \cup \Gamma} g (\theta \cdot \mathbf{n}) (\xi \cdot \mathbf{n}) \left(\partial_{\mathbf{n}} g - \partial^2_{\mathbf{nn}} u \right) + \int_\Omega \nabla u'_\theta \cdot \nabla u'_\xi - \int_{\Gamma_D} \left(u'_\xi \partial_{\mathbf{n}} u'_\theta + u'_\theta \partial_{\mathbf{n}} u'_\xi \right),$$

which is entirely symmetric. □

Chapter 8

Geometric discretized case

Contents

The computation of shape derivatives from a continuous point of view performed in the previous chapters is suitable for the *optimize-then-discretize* methodology (see Section 1.3) In this chapter we are concerned with the *discretize-then-optimize* approach for shape optimization with the level-set method (see Chapter 4) when the update of the level-set is governed by a Hamilton-Jacobi transport equation. The material domain is described implicitly by a level-set function which is positive outside the material domain, negative inside and zero on its boundary. Considering a finite element discretization, the level-set function allows one to compute a material fraction - or a material density - in each element. Like in density-based method such as the Solid Isotropic Material with Penalization (SIMP) method [20], every criterion can be computed thanks to the density. In the *discretize-then-optimize* methodology we introduce here, we proceed as follows for computing derivatives of a criterion : we first differentiate the criterion with respect to the material density, and compute the derivative of the material density with respect to the level-set function. Then we compose the two derivatives. It remains finally to compose again with the variation of the level-set trough the Hamilton-Jacobi equation.

This approach has already been studied for shape optimization [126, 127] but not when the update of the level-set function is made through a Hamilton-Jacobi transport equation. We would like to test if the coupling of a discrete approach with a Hamilton-Jacobi equation for tracking the motion of the shape is possible for first and second-order optimization methods. The discrete approach is also known for being consistent even if not accurate. We recall from Section 1.3 the difference between consistency and accuracy : accuracy is the difference between the computed derivative and the exact derivative of the continuous model whereas consistency is the difference with the derivatives of the discretized model. Therefore, in the present framework, we would also like to evaluate the consistency of the *discretize-then-optimize* approach.

This chapter is dedicated to first and second-order calculations for usual criteria in this discrete approach. The use of these derivatives are considered later in Section 9.4 as well as the question consistency.

8.1 Model Problem

Let Ω_0 be a reference domain. Its boundary is divided in three parts with $\partial\Omega_0 = \Gamma^0 \cup \Gamma_D^0 \cup \Gamma_N^0$. We want to optimize the part Γ^0 of this shape. Thus, we define the set of admissible shapes by

$$\mathcal{U}_{ad} := \left\{\Omega \in \mathcal{O}_3 \mid \Gamma_D = \Gamma_D^0 \text{ and } \Gamma_N = \Gamma_N^0\right\} = \left\{\Omega \in \mathcal{O}_3 \mid \Gamma_D^0 \cup \Gamma_N^0 \subset \partial\Omega\right\}. \tag{8.1}$$

We also introduce a bounded open set $\mathcal{D} \subset \mathbb{R}^d$ called the working domain or the computational domain, and impose that all admissible shapes are included in this domain. This means that we consider the set of admissible shapes as

$$\mathcal{U}_{ad} = \{\Omega \in \mathcal{O}_1 \mid \Gamma_D^0 \cup \Gamma_N^0 \subset \partial\Omega \text{ and } \Omega \subset \mathcal{D}\}.$$

In this section, the C^1-regularity for the shape is sufficient since we only need for the normal to be well-defined. For

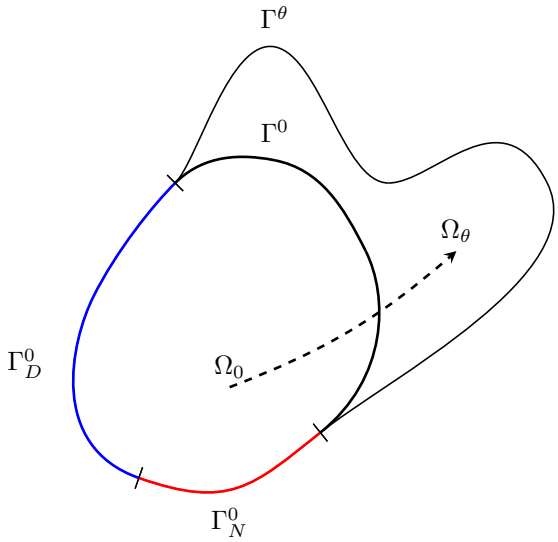


Figure 8.1: Variation of a shape Ω_0 and its boundary trough the diffeomorphism $(\mathrm{Id}+\theta)$, when the set of admissible shapes is restricted to $\mathcal{U}_{ad}$ defined by (8.1).

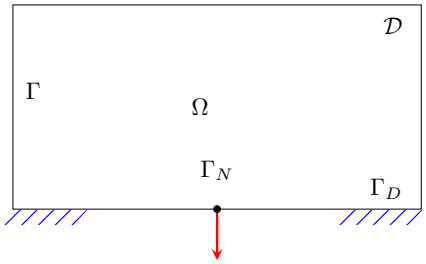


Figure 8.2: Boundary condition for the arch.

a given $f \in L^2\left(\mathcal{D};\mathbb{R}^d\right)$ and $g \in H^1\left(\mathcal{D};\mathbb{R}^d\right)$ such that $g = 0$ on Γ, we consider the following boundary-value problem, with unknown $u \in H^1(\Omega;\mathbb{R}^d)$:

$$\left\{\begin{array}{rcll} -\mathrm{div}\big(A\varepsilon(u)\big) & = & 0 & \text{in } \Omega, \\ u & = & 0 & \text{on } \Gamma_D, \\ A\varepsilon(u)\mathbf{n} & = & g & \text{on } \Gamma_N, \\ A\varepsilon(u)\mathbf{n} & = & 0 & \text{on } \Gamma, \end{array}\right. \tag{8.2}$$

where

$$A\xi = 2\mu\xi + \lambda \mathrm{Tr}(\xi)\mathrm{Id},$$
$$\varepsilon(u) = \frac{1}{2}\Big(\nabla u + \nabla u^T\Big).$$

This is the framework of the *elastic model* presented in Section 2.1.5. With the Hilbert space

$$\mathcal{V} = \left\{u \in H^1(\Omega;\mathbb{R}^d) \mid u = 0 \text{ on } \Gamma_D\right\},$$

the variational formulation of (8.2) is

$$\left\{\begin{array}{l} \text{find } u \in \mathcal{V}, \\ \forall \varphi \in \mathcal{V},\ \displaystyle\int_\Omega A\varepsilon(u) : \varepsilon(\varphi) = \int_{\Gamma_N} g \cdot \varphi. \end{array}\right. \tag{8.3}$$

We consider three criteria : the compliance, a least square displacement criterion and the volume of the shape. First, the compliance reads

$$J_1(\Omega) = \int_\Omega A\varepsilon(u) : \varepsilon(u) = \int_{\Gamma_N} g \cdot u,$$

Secondly, we consider a displacement criterion. For $\alpha \geq 2$, $u_0 \in L^\alpha\left(\Omega;\mathbb{R}^2\right)$ and $k \in L^\infty(\mathbb{R}^2;\mathbb{R}^2)$ let J_2 be defined by

$$J_2(\Omega) = \left(\int_\Omega k|u-u_0|^\alpha\right)^{\frac{1}{\alpha}}.$$

The volume is simply

$$V(\Omega) = \int_\Omega 1.$$

8.2 Discretization

We consider a square mesh with elements $K \in \mathcal{Q}_h$ of size h and introduce the finite element space $\mathbb{Q}_1\big(\mathcal{Q}_h;\mathbb{R}^2\big)$ for the state equation (8.2). Let $(\varphi_i)_{i\in\mathcal{I}}$ a basis of $\mathbb{Q}_1\big(\mathcal{Q}_h;\mathbb{R}^2\big)$. For the $\mathbb{Q}_1$ finite element space, the set $\mathcal{I}$ is the set of vertices of the mesh $\mathcal{Q}_h$. We introduce a density function ρ which plays the role of the characteristic function of Ω. It is defined on each cell by

$$\rho(K) = \begin{cases} 1 & \text{if } K \subset \Omega, \\ 0 & \text{if } K \subset \mathcal{D}\backslash\Omega, \\ \dfrac{|K\cap\Omega|}{|K|} & \text{otherwise.} \end{cases}$$

Notation. *For $v \in \mathbb{Q}_1\big(\mathcal{Q}_h;\mathbb{R}^2\big)$, we denote V_i its components in the basis $(\varphi_i)_{i\in\mathcal{I}}$, i.e $v = \sum_{i\in\mathcal{I}} V_i\varphi_i$.*

On each element K of the mesh, one can extract from $(\varphi_i)_{i\in\mathcal{I}}$ a basis $(\varphi_p)_{p\in\mathcal{I}_K}$ of $\mathbb{Q}_1\big(K;\mathbb{R}^2\big)$. The set $\mathcal{I}_K$ corresponds to the vertices of the element K. Any v, w in $\mathbb{Q}_1\big(\mathcal{Q}_h;\mathbb{R}^2\big)$ can be represented by their components V, W in the basis $(\varphi_i)_{i\in\mathcal{I}}$. Then their restriction to $\mathbb{Q}_1\big(K;\mathbb{R}^2\big)$ can be represented by the components $V_{|K}, W_{|K}$ in the basis $(\varphi_p)_{p\in\mathcal{I}_K}$. The elementary stiffness matrix $A_{\mathcal{Q}_h}$ is the matrix such that for any v, w with components V, W in the basis $(\varphi_i)_{i\in\mathcal{I}}$,

$$\int_K A\varepsilon(v) : \varepsilon(w) = {V_{|K}}^T A_{\mathcal{Q}_h} W_{|K}.$$

Taking into account the fact that the element K may not be embedded in Ω, we take advantage of the density function to approximate

$$\int_{K\cap\Omega} A\varepsilon(v) : \varepsilon(w) \simeq \rho(K){V_{|K}}^T A_{\mathcal{Q}_h} W_{|K}.$$

Therefore we have for any v, w in $\mathbb{Q}_1\big(\mathcal{Q}_h;\mathbb{R}^2\big)$

$$\int_\Omega A\varepsilon(v) : \varepsilon(w) \simeq \sum_{K\in\mathcal{Q}_h} \rho(K){V_{|K}}^T A_{\mathcal{Q}_h} W_{|K}. \tag{8.4}$$

Definition 8.2.1. *We define a third-order stiffness tensor $\mathscr{A}$ by*

$$\forall v, w \in \mathbb{Q}_1\big(\mathcal{Q}_h;\mathbb{R}^2\big),\ \mathscr{A}\big(\rho, v, w\big) \equiv \sum_{K\in\mathcal{Q}_h} \rho(K){V_{|K}}^T A_{\mathcal{Q}_h} W_{|K}. \tag{8.5}$$

Due to the symmetry of the elementary stiffness matrix $A_{\mathcal{Q}_h}$, $\mathscr{A}$ is symmetric with respect to v, w, i.e $\mathscr{A}\big(\rho, v, w\big) = \mathscr{A}\big(\rho, w, v\big)$. We also build another third-order mass tensor $\mathscr{M}$ by

$$\forall v, w \in \mathbb{Q}_1\big(\mathcal{Q}_h;\mathbb{R}^2\big),\ \mathscr{M}\big(\rho, v, w\big) \equiv \sum_{K\in\mathcal{Q}_h} \rho(K){V_{|K}}^T M_{\mathcal{Q}_h} W_{|K}, \tag{8.6}$$

where $M_{\mathcal{Q}_h}$ is the usual elementary mass matrix for $\mathbb{Q}_1$, that is such that

$$\forall v, w \in \mathbb{Q}_1(\mathbb{R}^2;\mathbb{R}^2),\ \int_K vw = {V_{|K}}^T M_{\mathcal{Q}_h} W_{|K}.$$

The mass tensor is straightforwardly symmetric.

Definition 8.2.2. *We define the stiffness matrix A by*

$$A = \mathscr{A}\big(\rho, \cdot, \cdot\big).$$

For the discretized version of (8.2), we will consider the load g as being a point load applied on a vertex of the mesh $\mathcal{Q}_h$. Denoting G the components of the $\mathbb{Q}_1$ interpolation of g, this vector would have only one non-zero entry. Then, for solving (8.2), we consider the following discretized version

$$\forall v \in \mathbb{Q}_1\big(\mathcal{Q}_h;\mathbb{R}^2\big),\ \mathscr{A}\big(\rho, u, v\big) = V^T G, \tag{8.7}$$

which can also be written as

$$AU = G. \tag{8.8}$$

With all these ingredients, it is possible to compute the criteria (J_1, J_2 and V) and their gradients which we detail in the following section. For the resolution of the Hamilton-Jacobi transport equation, we will use the classical numerical scheme we introduced in Chapter 4.

8.3 Discrete derivation

In the previous section, we constructed a discrete model for the optimization problem. Now we are concerned with the derivation of this discrete model.

8.3.1 Density function

The density is computed from a level-set function ϕ. The level-set function is defined on each node $i \in \mathcal{I}$, whereas the density function is defined on each cell $K \in \mathcal{Q}_h$. A cell fully include in the domain Ω has a full density ($\rho = 1$), and a cell that does not intersect the domain has a void density ($\rho = \varepsilon \ll 1$). The use of an ersatz density instead of void is made to avoid the ill-conditioning of the stiffness matrix A. In the sequel, we only focus on cells that are intersected by the boundary. They are characterized by the fact that the sign of ϕ changes on their four nodes. At their center we introduce an additional node (in Figure 8.4 this node is denoted by C). This additional node is necessary for there might be an uncertainty about the position of the shape (see Figure 8.3). A priori we do not know the value of ϕ

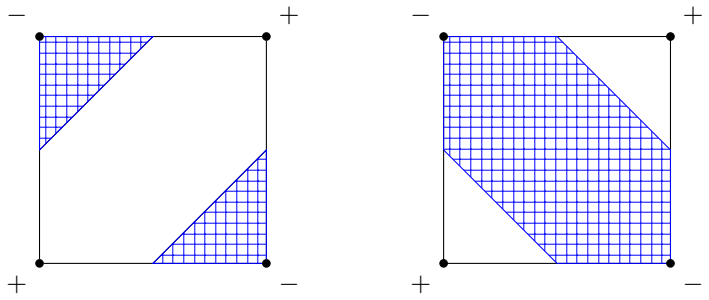


Figure 8.3: Non-uniqueness of a shape (grid domain) for a given level-set function defined by its values at the four corners of a square.

at this node. We introduce at this node a value of ϕ such that it is the mean value of ϕ on the square cell. (This is exactly the $\mathbb{Q}_1$ approximation). Now the cell is divided in four similar triangles $\mathcal{T}_i$ ($i \in [\![1, 4]\!]$).

Figure 8.4 shows a square cell, and the subdividing triangles. It also shows all the possible ways for the domain Ω to intersect a triangle. In each triangle $\mathcal{T} = ABC$ we compute a linear interpolation of ϕ and $\mathscr{D}$, the linear approximation of the curve $\phi = 0$. We denote P_1, P_2, P_3 the intersection points between $\mathscr{D}$ and the boundary of the triangle, according to whether they are respectively on $[A, C]$, $[B, C]$ or $[A, B]$. The intersection points being on $[A, C]$, $[B, C]$ or $[A, B]$ are entirely determined by the sign of ϕ at the nodes A, B and C. Now we choose a coordinate system for the triangle, so that $\mathscr{D}$ can be parametrized by the x-coordinate. Depending on the choice of the coordinate system, we are looking for the plane $\mathscr{P} \subset \mathbb{R}^3$ containing the points

$$(a)\begin{cases}\big(0,0,\phi_C\big),\\ \big(\frac{\sqrt{2}}{2},0,\phi_A\big),\\ \big(0,\frac{\sqrt{2}}{2},\phi_B\big).\end{cases}\quad (b)\begin{cases}\big(0,0,\phi_B\big),\\ \big(\frac{\sqrt{2}}{2},0,\phi_C\big),\\ \big(\frac{\sqrt{2}}{2},\frac{\sqrt{2}}{2},\phi_A\big).\end{cases}\quad (c)\begin{cases}\big(0,0,\phi_A\big),\\ \big(1,0,\phi_B\big),\\ \big(\frac{1}{2},\frac{1}{2},\phi_C\big).\end{cases}$$

This amounts to doing a linear approximation of ϕ. This is quite accurate, since the function ϕ is often redistanciated during the resolution of the Hamilton-Jacobi transport equation to be an approximation of the signed distance function. Let $ax + by + cz + d = 0$ be an equation of $\mathscr{P}$. Taking $c = 1$ leads to

$$(a)\begin{cases}a &=& \sqrt{2}\big(\phi_C - \phi_A\big),\\ b &=& \sqrt{2}\big(\phi_C - \phi_B\big),\\ c &=& 1,\\ d &=& -\phi_C.\end{cases}\quad (b)\begin{cases}a &=& \sqrt{2}\big(\phi_B - \phi_C\big),\\ b &=& \sqrt{2}\big(\phi_C - \phi_A\big),\\ c &=& 1,\\ d &=& -\phi_B.\end{cases}\quad (c)\begin{cases}a &=& \phi_A - \phi_B,\\ b &=& \phi_A + \phi_B - 2\phi_C,\\ c &=& 1,\\ d &=& -\phi_A.\end{cases}$$

Now we have the equation of the straight-line $\mathscr{D}$, intersection of $\mathscr{P}$ with the plane whose equation is $z = 0$. Since ϕ is very close to a signed distance function, it can at most vanish on two points on the triangle (assuming Ω is a "nice" domain, well discretized on $\mathcal{Q}_h$). The equation of $\mathscr{D}$ is then $ax + by + d = 0$. The choice of coordinates for the triangle is made so that $b \neq 0$. The area of $\Omega \cap \mathcal{T}$ can be expressed only in terms of the values of ϕ at the nodes A, B, C. When $\mathscr{D}$ coincides with an edge of the triangle $\mathcal{T}$, the area is either 0 or $\frac{1}{4}h^2$. Otherwise, we can compute it

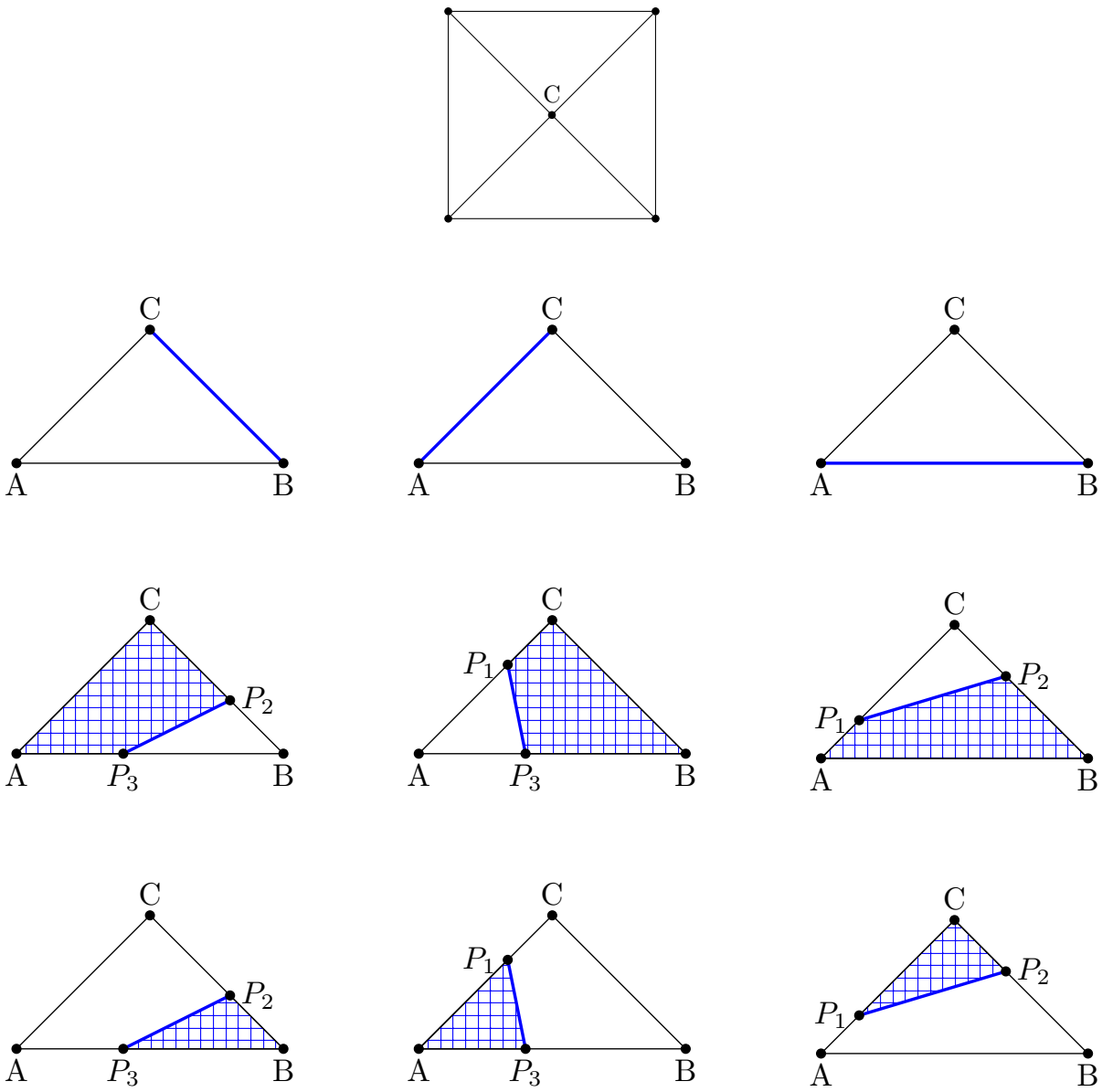


Figure 8.4: All possible configurations of a triangle cut by the level-set function.

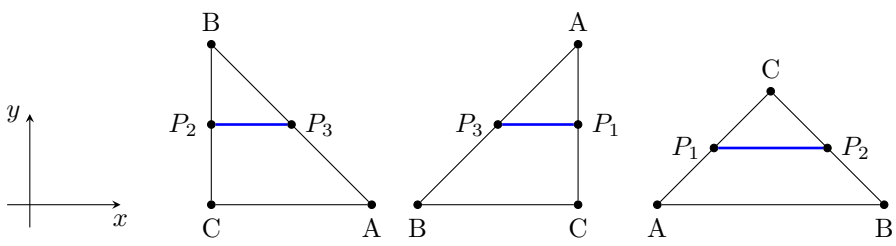


Figure 8.5: Choice of coordinate system for the triangle ABC in the cases (a) (left), (b) (middle) and (c) (right).

analytically.

$$(a)\ |\Omega \cap \mathcal{T}| = \begin{cases} \dfrac{\phi_A\phi_B - \phi_A\phi_C + \phi_B\phi_C}{4(\phi_A - \phi_B)(\phi_B - \phi_C)} & \text{if } \phi_A < 0, \\ \dfrac{-\phi_B^2}{4(\phi_A - \phi_B)(\phi_B - \phi_C)} & \text{otherwise.} \end{cases} \tag{8.9}$$

$$(b)\ |\Omega \cap \mathcal{T}| = \begin{cases} \dfrac{\phi_A^2}{4(\phi_A - \phi_B)(\phi_A - \phi_C)} & \text{if } \phi_A < 0, \\ \dfrac{-(\phi_A\phi_B + \phi_A\phi_C - \phi_B\phi_C)}{4(\phi_A - \phi_B)(\phi_A - \phi_C)} & \text{otherwise.} \end{cases} \tag{8.10}$$

$$(c)\ |\Omega \cap \mathcal{T}| = \begin{cases} -\dfrac{\phi_A\phi_C - \phi_A\phi_B + \phi_B\phi_C}{4(\phi_A - \phi_C)(\phi_B - \phi_C)} & \text{if } \phi_A < 0, \\ \dfrac{\phi_C^2}{4(\phi_A - \phi_C)(\phi_B - \phi_C)} & \text{otherwise.} \end{cases} \tag{8.11}$$

Adding all contributions on the four triangles of a cell, we get $\mathscr{A}_\tau$, the value of the area of $\Omega \cap Y_\tau$. The density ρ is obtained on Y_τ by

$$\rho(\tau) = \frac{\mathscr{A}_\tau}{|Y_\tau|}.$$

It depends only on the values of ϕ at the nodes of the square cell Y_τ.

Lemma 8.3.1. *Assume that ϕ does not vanish on the grid points, and that in any cell the mean value of ϕ on the vertices does not vanish. Then the discrete density function is differentiable with respect to ϕ, and its derivative is*

$$\delta\rho(\phi; d\phi) = \nabla\rho\ d\phi, \tag{8.12}$$

as long as ϕ does not vanish on one of the vertices of the triangles.

Proof. On each triangle in each cell τ, the dependency of the density with respect to the values of ϕ is analytic. This expression is then differentiable with respect to ϕ. The lack of regularity occurs when ϕ vanishes on one of the vertices of a triangle. In this case, with a variation of ϕ at the vertex where it vanishes, the density may switch between two analytic expressions (8.9),(8.10), (8.11). Once we have computed the gradient of the density in each cell τ with respect to ϕ, one can assemble it to get the global gradient of ρ. □

8.3.2 Compliance

The discrete expression of the compliance reads

$$J_{1,d} = \mathscr{A}\big(\rho, U, U\big), \tag{8.13}$$

where U solves

$$AU = G.$$

Lemma 8.3.2. *Let $d\rho_1, d\rho_2$ be two variations of ρ such that $d\rho_1 = d\rho_2 = 0$ on $\Gamma_D \cup \Gamma_N$. The discrete compliance $J_{1,d}$ is twice differentiable with respect to ρ, with*

$$\delta J_{1,d}(\rho; d\rho_1) = -\mathscr{A}\big(d\rho_1, u, u\big).$$

$$\begin{aligned} \delta^2 J_{1,d}(\rho; d\rho_1, d\rho_2) &= -2\mathscr{A}\big(d\rho_1, du(d\rho_2), u\big), \\ &= 2\mathscr{A}\big(\rho, du(d\rho_1), u(d\rho_2)\big). \end{aligned}$$

Proof. The variations $d\rho_1, d\rho_2$ are assumed not to modify $\Gamma_D \cup \Gamma_N$ so that there is no variation of the right-hand side of (8.8). Differentiating (8.13) with respect to ρ gives

$$\delta J_{1,d}(\rho; d\rho_1) = \mathscr{A}\big(d\rho_1, u, u\big) + \mathscr{A}\big(\rho, du(d\rho_1), u\big) + \mathscr{A}\big(\rho, u, du(d\rho_1)\big).$$

The symmetry of $\mathscr{A}$ allows us to write

$$\delta J_{1,d}(\rho; d\rho_1) = 2\mathscr{A}\big(\rho, du(d\rho_1), u\big) + \mathscr{A}\big(d\rho_1, u, u\big).$$

We can differentiate the state equation (8.8). Therefore

$$\forall v \in \mathbb{Q}_1\big(\mathcal{Q}_h;\mathbb{R}^2\big),\ \mathscr{A}\big(d\rho_1,u,v\big)+\mathscr{A}\big(\rho,du(d\rho_1),v\big)=0, \tag{8.14}$$

which leads to the announced result for the first derivative. Since $\mathscr{A}$ is linear in ρ, the state equation is smooth with respect to ρ, and so is then the compliance $J_{1,d}$. The second-order differentiability and the first expression of the second-order derivative are then straightforwardly obtained. The variation of the state equation (8.14) with $v=u(d\rho_2)$ gives the result.

□

Lemma 8.3.3. *Assuming that ρ is twice differentiable with respect to ϕ (under the assumptions of Lemma 8.3.1). Let $d\phi_1, d\phi_2 \in \mathbb{Q}_1\big(\mathbb{R}^2;\mathbb{R}^2\big)$ such that $\nabla\rho\, d\phi_i = 0$ on $\Gamma_D\cup\Gamma_N$ (with $i=1,2$). The compliance is twice differentiable with respect to ϕ with*

$$\delta[J_{1,d}\circ\rho](\phi;d\phi_1)=-\mathscr{A}\big(\nabla\rho\ d\phi_1,u,u\big),$$

and

$$\delta^2[J_{1,d}\circ\rho](\phi;d\phi_1,d\phi_2)=-\mathscr{A}\Big(d^2\rho(\phi;d\phi_1;d\phi_2),u,u\Big)+2\mathscr{A}\Big(\rho,du\big(\nabla\rho\,d\phi_1\big),du\big(\nabla\rho\,d\phi_2\big)\Big),$$

where $d^2\rho(\phi;d\phi_1;d\phi_2)$ is the vector defined by

$$\begin{pmatrix} d\rho(1)(\phi;d\phi_1;d\phi_2)\\ \vdots\\ d\rho(n)(\phi;d\phi_1;d\phi_2)\end{pmatrix}$$

with n being the number of cells τ, and $d\rho(\tau)(\phi;d\phi_1;d\phi_2)$ is the second-order variation of $\rho(\tau)$ in the directions $d\phi_1, d\phi_2$.

Proof. It is just a chain rule differentiation, thanks to (8.12). The requirement $\nabla\rho\,d\phi_i=0$ on $\Gamma_D\cup\Gamma_N$ implies that the space $\mathcal{V}$ as well as the right-hand side of (8.8) do not vary with respect to the shape. □

8.3.3 Displacement

Let $\alpha\geq 2$, $u_0\in L^\alpha(\Omega;\mathbb{R}^2)$ be a target displacement and $k\in L^\infty(\mathbb{R}^2;\mathbb{R}^2)$. The displacement criterion reads

$$J_2(\Omega)=\bigg(\int_\Omega k|u-u_0|^\alpha\bigg)^{\frac{1}{\alpha}},$$

where u is the solution to (8.2). The discrete displacement criterion reads

$$J_{2,d}(\rho)=\bigg(\sum_{\tau\in\mathcal{T}_h}\rho(\tau)k(\tau)|u(\tau)-u_0(\tau)|^\alpha\bigg)^{\frac{1}{\alpha}}=\mathscr{M}(\rho,k|u-u_0|^\alpha,1)^{\frac{1}{\alpha}}.$$

We introduce

$$j_2(\Omega)=\int_\Omega k|u-u_0|^\alpha,$$

whose discrete expression is

$$j_{2,d}(\rho)=\sum_{\tau\in\mathcal{T}_h}\rho(\tau)k(\tau)|u(\tau)-u_0(\tau)|^\alpha=\mathscr{M}(\rho,k|u-u_0|^\alpha,1).$$

Lemma 8.3.4. *Let $d\rho_1$ be a variation of ρ such that $d\rho_1=0$ on $\Gamma_D\cup\Gamma_N$. The discrete displacement $j_{2,d}$ is differentiable with respect to ρ, with*

$$\delta j_{2,d}(\rho;d\rho_1)=\mathscr{M}(d\rho_1,k|u-u_0|^\alpha,1)+\mathscr{A}\big(d\rho_1,u,p\big),$$

where p is the solution to

$$\forall v\in\mathbb{Q}_1(\mathbb{R}^2;\mathbb{R}^2),\ \mathscr{A}\big(\rho,p,v\big)=-\mathscr{M}(\rho,\alpha k\,|u-u_0|^{\alpha-2}(u-u_0),v).$$

Proof. The derivation with respect to ρ leads to

$$\delta j_{2,d}(\rho;d\rho_1)=\mathscr{M}(d\rho_1,k|u-u_0|^\alpha,1)+\mathscr{M}(\rho,\alpha k\,|u-u_0|^{\alpha-2}(u-u_0),du(d\rho_1)).$$

Introducing p solution to

$$\forall v\in\mathbb{Q}_1(\mathbb{R}^2;\mathbb{R}^2),\ \mathscr{A}\big(\rho,p,v\big)=-\mathscr{M}(\rho,\alpha k\,|u-u_0|^{\alpha-2}(u-u_0),v),$$

with $v = du(d\rho_1)$, we have

$$\mathscr{M}(\rho, \alpha k\,|u-u_0|^{\alpha-2}(u-u_0), du(d\rho_1)) = -\mathscr{A}\big(\rho, p, du(d\rho_1)\big),$$

and then with the help of (8.14)

$$\mathscr{M}(\rho, \alpha k\,|u-u_0|^{\alpha-2}(u-u_0), du(d\rho_1)) = \mathscr{A}\big(d\rho_1, p, u\big),$$

which is the expected result. □

Lemma 8.3.5. *Let $d\rho$ be a variation of ρ such that Γ_N is not modified. Let also p be the solution to*

$$\forall v \in \mathbb{Q}_1(\mathbb{R}^2;\mathbb{R}^2),\ \mathscr{A}\big(\rho, p, v\big) = -\mathscr{M}(\rho, \alpha k\,|u-u_0|^{\alpha-2}(u-u_0), v).$$

Then for all $v \in \mathbb{Q}_1(\mathbb{R}^2;\mathbb{R}^2)$ we have

$$\begin{aligned} 0 = {} & \mathscr{A}(d\rho, p, v) + \mathscr{A}(\rho, dp(d\rho), v) \\ & + \mathscr{M}(d\rho, \alpha k\,|u-u_0|^{\alpha-2}(u-u_0), v) \\ & + \mathscr{M}(d\rho, \alpha(\alpha-1)k\,|u-u_0|^{\alpha-2}du(d\rho), v). \end{aligned} \tag{8.15}$$

Proof. This is nothing but a chain rule. □

Lemma 8.3.6. *Let $d\rho_1, d\rho_2$ be two variations of ρ such that $d\rho_1 = d\rho_2 = 0$ on $\Gamma_D \cup \Gamma_N$. Then $j_{2,d}$ is twice differentiable with respect to ρ, with*

$$\begin{aligned} \delta^2 j_{2,d}(\rho; d\rho_1, d\rho_2) = {} & \mathscr{A}\big(d\rho_1, du(d\rho_2), p\big) + \mathscr{A}\big(d\rho_2, du(d\rho_1), p\big), \\ & + \mathscr{M}\big(\rho, k\alpha(\alpha-1)|u-u_0|^{\alpha-2}, du(d\rho_1)du(d\rho_2)\big) \\ & + \mathscr{M}\big(d\rho_1, k\alpha|u-u_0|^{\alpha-2}(u-u_0), du(d\rho_2)\big) \\ & + \mathscr{M}\big(d\rho_2, k\alpha|u-u_0|^{\alpha-2}(u-u_0), du(d\rho_1)\big). \end{aligned}$$

Proof. It suffices to combine the chain rule differentiation with (8.15). □

8.3.4 Volume

The discrete expression of the volume reads

$$V = \sum_{\tau \in \mathcal{T}_h} \rho(\tau)$$

Lemma 8.3.7. *The volume function is twice differentiable with respect to ρ, its derivatives are given by*

$$\delta V(\rho; d\rho) = \sum_{\tau \in \mathcal{T}_h} d\rho,$$

$$\delta^2 V(\rho; d\rho_1; d\rho_2) = 0.$$

Proof. The result comes from the linearity of the discrete expression of the volume. □

Lemma 8.3.8. *Assuming that the density ρ is differentiable with respect to ϕ, the volume is also differentiable with respect to ϕ. For $d\phi_1, d\phi_2 \in \mathbb{Q}_1\big(\mathcal{Q}_h;\mathbb{R}^2\big)$ we have*

$$\begin{aligned} \delta V(\phi; d\phi) &= \delta V\big(\phi; \nabla\rho\, d\phi\big), \\ \delta^2 V(\phi; d\phi_1, d\phi_2) &= \delta V\big(\phi; d^2\rho(\phi; d\phi_1; d\phi_2)\big). \end{aligned}$$

8.3.5 Transport equation

Finally, after computing derivatives of the various criteria $J_{1,d}$, $J_{2,d}$ and V with respect to a variation of the level-set, we consider the variations of a level-set function through a Hamilton-Jacobi transport equation. We first recall briefly the numerical scheme (see Section 4.2.2). For a given normal velocity v, the level-set ϕ_0 is updated with :

$$\begin{cases} \forall (n,i,j) \in \mathbb{N}\times\mathbb{Z}^2, & \phi_{i,j}^{n+1} = \phi_{i,j}^n - \Delta t\Big(\max(v_{i,j},0)\nabla_{i,j}^+\phi^n + \min(v_{i,j},0)\nabla_{i,j}^-\phi^n\Big), \\ \forall (i,j) \in \mathbb{Z}^2, & \phi_{i,j}^0 = \phi_0(x_i,y_j), \end{cases} \tag{8.16}$$

where $\nabla_{i,j}^{\pm}\phi$ is a numerical approximation by finite difference of the gradient of ϕ. The level-set function is often redistanciated, then when ϕ_0 is smooth, at least at the first step we have $|\nabla\phi_0| = 1$. Therefore we have

$$\phi^1 - \phi^0 = -v\Delta t.$$

We then consider that the variation $d\phi$ given by the velocity v reads

$$d\phi = -v.$$

As a result, the previous discrete derivatives with respect to normal velocities can be easily obtained. For example, let us consider the volume criterion. Let $v, w \in \mathbb{Q}_1(\mathbb{R}^2; \mathbb{R})$ be two normal velocities. The derivatives of the volume in these direction read

$$\begin{aligned} \delta V(\Omega; v) &= -\delta V\big(\phi; \nabla\rho\, v\big), \\ \delta^2 V(\Omega; v, w) &= \delta V\big(\phi; d^2\rho(\phi; v, w)\big). \end{aligned}$$

Chapter 9

Numerical examples

Contents

After performing second-order analysis in the previous chapters, we would like now to benefit from it in view of optimization. For finite-dimensional optimization, it is well-known that under regularity conditions on objective functions, the Newton method converges quadratically. In the case of shape optimization we would like to see if the second-order analysis allows one to have this quadratic convergence. At first we detail the optimization methods we use and compare them. We focus on a quasi-Newton method that we compare to a gradient-like method. Secondly we will introduce different approximation of the Newton method. Finally we will compare these methods on numerical examples for the parametric thickness optimization and on the *discretize-then-optimize* approach for shape optimization with the level-set method. The case of geometric shape optimization (see Chapter 7) is not considered here for the numerical examples but only in Chapter 12.

9.1 Optimization method

In this section we consider the question of minimizing a real function E on a set $\mathcal{U}_{ad}$:

$$\min_{\Omega \in \mathcal{U}_{ad}} E(\Omega). \tag{9.1}$$

The shape Ω can be represented by a thickness such as in Chapter 6, or by a material density like in Chapter 8, a level-set function like in Chapter 4 or any other model. We will only assume here that we can compute first and second-order derivatives of the function $E : \mathcal{U}_{ad} \to \mathbb{R}$ with respect to the model.

For simplicity we consider the unconstrained minimization of an objective function $E(\Omega)$ over all possible shapes $\Omega \subset \mathcal{U}_{ad}$. Of course, in practice there are always additional constraints, whether directly on the domain Ω (like geometrical or manufacturing constraints) or on the mechanical performances of Ω. Nevertheless, for simplicity we focus on the definition of the Newton algorithm in the unconstrained case and we recall that the Newton method can be extended to the constrained setting [92] without theoretical difficulties, although it may be quite intricate in numerical practice. Starting from an initial shape Ω_0, we build an iterative sequence $(\Omega_p)_{p\in\mathbb{N}}$ in order to minimize the objective function $E(\Omega)$.

For a direction v and a time-step t we will denote by $\Omega(p, t, v)$ the update of Ω_p in the direction v with the time-step t. We also assume that there is a scaling invariance between t and v, *i.e.* such that for $\alpha > 0$, $\Omega(p, \frac{t}{\alpha}, \alpha v) = \Omega(p, t, v)$. The new shape is defined as $\Omega_{p+1} = \Omega(p, t_p, v^p)$ where $t_p > 0$ is chosen in order to decrease the reduced objective function $t \mapsto E\left(\Omega(p, t, v^p)\right)$. With the notation of Chapter 1 it would correspond to take

$$u^{p+1} = u^p + t^p\, v^p.$$

At an iterate p, we will denote respectively $l_{p,1}$ and $l_{p,2}$ the first and second-order derivatives of the objective function E with respect to the model. We denote $\mathcal{F}_p$ the space where the derivatives take place. The use of the index p on $\mathcal{F}_p$ means that this space may depend on the shape Ω_p.

9.1.1 Newton's algorithm

In view of the scaling invariance $\Omega(p, \frac{t}{\alpha}, \alpha v^p) = \Omega(p, t, v^p)$, there is no loss of generality in choosing the parameter t_p equal to 1. Then, for small enough $v \in \mathcal{F}_p$, the following quadratic approximation holds true

$$E\big(p, \Omega(1, v)\big) \simeq E(\Omega_p) + l_{p,1}(v) + \frac{1}{2} l_{p,2}(v, v),$$

The principle of the Newton method is to minimize this quadratic approximation in order to obtain the descent direction v^p. First, we solve the following problem, referred to as the Newton problem

$$\inf_{v \in \mathcal{F}_p} \quad l_{p,1}(v) + \frac{1}{2} l_{p,2}(v, v). \tag{9.2}$$

There is a priori no reason for the Newton problem (9.2) to have a finite infimum. Indeed, the Hessian operator $l_{p,2}$ may have zero or negative eigenvalues. To be convinced, consider the examples of Chapter 7 where the Hessian depends on the normal vector $\mathbf{n}$ but also on the shape curvature $\mathcal{H}$ and has thus no clear sign. In such a case, the infimum in (9.2) could be $-\infty$. For this reason, instead of solving the Newton problem (9.2) we consider a trust-region variant (see for example [92, Section 6.4], [131])

$$\begin{array}{l} \min \\ v \in \mathcal{F}_p, \\ \|v\|_{L^\infty} \leq v_M, \end{array} \qquad l_{p,1}(v) + \frac{1}{2} l_{p,2}(v, v)\,, \tag{9.3}$$

for some finite positive bound v_M. Such a bound implies that, at least after finite-dimensional discretization, (9.3) admits a minimizer. If the solution v^p to (9.3) satisfies a strict bound $\|v^p\|_{L^\infty} < v_M$, then it is also a solution of (9.2) and it satisfies the optimality condition or Newton equation

$$l_{p,2}(v^p, w) = -l_{p,1}(w) \quad \forall w \in \mathcal{F}_p\,.$$

After having found a good descent direction v^p, we also look for a good descent step t_p to ensure an efficient decrease of the reduced objective function $t \mapsto E\big(\Omega(p, t, v^p)\big)$, or equivalently $t \mapsto E\big(\Omega(p, 1, tv^p)\big)$, due to the scaling invariance. Ideally we should choose t_p as the global minimizer of the reduced objective function, but repeated evaluations of the objective function are way too costly. Therefore, our strategy is a trade-off which ensures a substantial decrease of the objective function, at an affordable computing price. We use a well-known backtracking procedure [92, Procedure 3.1, Chapter 3] :

Algorithm 9.1.1. *Backtracking line search.*

1. *Choose $t_p > 0$, c_1, c_2 in $]0, 1[$.*
2. *Set $t_p := c_1 t_p$.*
3. *If $E\big(\Omega(p, 1, t_p v^p)\big) \leq E\big(\Omega_p\big) + c_2\, t_p\, l_{p,1}(v^p)$ stop. Else go to step 2.*

The following algorithm will be referred to as the Newton algorithm in the sequel.

Algorithm 9.1.2. *Newton's algorithm*

1. ***Initialization*** *Set $p = 0$. Choose an initial domain Ω_0, two convergence thresholds $\varepsilon > 0$, $\eta > 0$, and two coefficients c_0, c_1 in $]0, 1[$.*
2. ***Newton's direction*** *Compute the solution v^p of (9.3).*
3. ***Time Step*** *Compute a time step t_p with Algorithm 9.1.1.*
4. ***Update*** *Set $\Omega_{p+1} := \Omega(p, t_p, v^p)$ and $p = p + 1$. If $\|v^p\|_{\mathcal{F}_p} \leq \varepsilon$ or $t_p \leq \eta$ stop. Else return to step 2.*

Remark 9.1.3. *Since the Hessian operator $l_{p,2}$ may have negative eigenvalues, the solution v^p to (9.3) may fail to be a descent direction,* i.e. *we may have $l_{p,1}(v^p) > 0$. When it occurs, we take its opposite in order to ensure that $l_{p,1}(v^p) \leq 0$.*

9.1.2 Gradient algorithm

Gradient algorithms are based on a first-order Taylor expansion for sufficiently small v

$$E\big(\Omega(p,1,v)\big) \simeq E(\Omega_p) + l_{p,1}(v).$$

The steepest-descent direction is defined by the solution to

$$\min_{\|v\|_{\mathcal{F}}=1,} \quad l_{p,1}(v).$$

Then, one looks for a descent step $t_p > 0$ such that the objective function decreases. This is the classical gradient-type method. However, in order to make a fair comparison with the previous Newton algorithm which is coupled to a trust-region method, we change this usual algorithm and couple it again with a trust-region method. More precisely, we compute a descent direction as the solution of

$$\min_{\substack{v \in \mathcal{F}_p,\\ \|v\|_{L^\infty} \leq v_M}} \quad l_{p,1}(v) + \frac{1}{2}\|v\|^2_{\mathcal{F}_p}. \tag{9.4}$$

Comparing with (9.3), we keep the same L^∞ bound and we replace the Hessian operator by the simpler $\mathcal{F}_p$ scalar product. Note that this quadratic term is not really necessary to make (9.4) well-posed but it helps for making a fair comparison with (9.3). In particular, when $\mathcal{F}$ is a Banach space, if v^p is a solution to (9.4) which does not saturate the bound, $\|v^p\|_{L^\infty} < v_M$, then the optimality condition of (9.4) yields that v^p is the steepest-descent direction.

The main reason for choosing this approach for the gradient algorithm is to ease the comparison with the Newton method. Indeed, the only difference between (9.3) and (9.4) is the replacement of the Hessian operator by the identity operator on $\mathcal{F}_p$.

The following algorithm will be referred to as the gradient algorithm in the sequel.

Algorithm 9.1.4. *Gradient algorithm*

1. ***Initialization*** *Set $p = 0$. Choose an initial domain Ω_0, two convergence thresholds $\varepsilon > 0$, $\eta > 0$, and two coefficients c_0, c_1 in $]0,1[$.*
2. ***Gradient direction*** *Compute the solution v^p of (9.4).*
3. ***Time Step*** *Compute a time step t_p with Algorithm 9.1.1.*
4. ***Update*** *Set $\Omega_{p+1} := \Omega(p, t_p, v^p)$ and $p = p + 1$. If $\|v^p\|_{\mathcal{F}_p} \leq \varepsilon$ or $t_p \leq \eta$ stop. Else return to step 2.*

9.2 Approximation of the Hessian

In the case of smooth objective functions to be optimized, the Newton method may improve a classical gradient method by accelerating the convergence rate. Indeed it could become quadratic instead of being only linear. Therefore the use of the second-order analysis usually decreases significantly the number of iteration of an optimization process. However each iteration may become way more expensive from a computational point of view. In case of shape optimization, when the criterion depends on the solution of an elliptic partial differential equation, the computation of the second-order derivative implies to invert the corresponding elliptic operator, or at least to solve a lot of linear systems with this operator. It can become prohibitive especially when the number of variables in the discretization increases. This is the main reason why considering approximations of the second-order derivative.

The underlying idea of the methods we propose here is to approximate the inverse of the elliptic operator and then to keep the exact structure of the second-order derivative. This idea does not really depend on the chosen framework for shape optimization : it can be relevant for the thickness optimization (see Chapter 6) as well as for the geometric shape optimization with the *optimize-then-discretize* approach (see Chapter 7) or for the *discretize-then-optimize* approach (see Chapter 8). In order to ease the reading we will focus on the compliance criterion.

Let $\mathcal{U}_{ad}$ be a set of designs. For instance $\Omega \in \mathcal{U}_{ad}$ could be a thickness distribution as well as a bounded open set in $\mathbb{R}^d$. We only assume that $\mathcal{U}_{ad}$ is a Banach space. We also denote $\mathcal{F}$ the Banach space to which the directions of derivation belong. Let $\mathcal{A}(\Omega)$ be an invertible symmetric elliptic bilinear map (for instance the linearized elasticity operator) defined on a Banach space $\mathcal{V}$, and f be a source term. The operator $\mathcal{A}$ is a priori dependent on the design Ω, contrary to the source term f which is assumed not to vary with respect to the shape. For $v \in \mathcal{V}$, we denote $\mathcal{A}(\Omega)v$ the linear map : $\mathcal{V} \ni w \mapsto \mathcal{A}(\Omega)(v,w)$. We consider the state u_Ω solution to a "physical system"

$$\mathcal{A}(\Omega)u_\Omega = f. \tag{9.5}$$

In this section we are concerned with the approximation of the second-order derivative of a criterion that depends on the state u_Ω. We will focus on the case of the compliance that reads

$$J_1(\Omega) = \mathcal{A}(\Omega)(u_\Omega, u_\Omega). \tag{9.6}$$

Even if this example is one of the simplest criterion, the approximation method applies to any other criterion implying the state. The derivation of the function J_1 with respect to the design in the direction $\theta \in \mathcal{F}$ formally reads

$$J_1{}'(\Omega;\theta) = 2\,\mathcal{A}(\Omega)(u_\Omega, u'_{\Omega,\theta}) + \mathcal{A}'(\Omega;\theta)\,(u_\Omega, u_\Omega),$$

where $u'_{\Omega,\theta}$ and $\mathcal{A}'(\Omega;\theta)$ correspond respectively to the derivation with respect to the design of the state and the elliptic operator. We can also write formally the derivation of the state equation (9.5) :

$$\mathcal{A}(\Omega)u'_{\Omega,\theta} = -\mathcal{A}'(\Omega;\theta)\,u_\Omega, \tag{9.7}$$

which leads to

$$J_1{}'(\Omega;\theta) = -\mathcal{A}'(\Omega;\theta)\,(u_\Omega, u_\Omega)$$

Therefore, for $\theta, \xi \in \mathcal{F}$, the second-order derivative of J_1 reads

$$J_1{}''(\Omega;\theta,\xi) = -2\,\mathcal{A}'(\Omega;\theta)\,(u_\Omega, u'_{\Omega,\xi}) - \mathcal{A}''(\Omega;\theta,\xi)\,(u_\Omega, u_\Omega).$$

In order to ease the reading here, we denote by $g(u,\theta)$ the right-hand side of (9.7) :

$$g(u,\theta) = -\mathcal{A}'(\Omega;\theta)\,u_\Omega.$$

With the first-order derivation of the sate equation, this can be written as

$$J_1{}''(\Omega;\theta,\xi) = 2\,\mathcal{A}(\Omega)(u'_{\Omega,\theta}, u'_{\Omega,\xi}) - \mathcal{A}''(\Omega;\theta,\xi)\,(u_\Omega, u_\Omega),$$

or

$$J_1{}''(\Omega;\theta,\xi) = 2\,\mathcal{A}(\Omega)^{-1}\Big(g(u,\xi), g(u,\theta)\Big) - \mathcal{A}''(\Omega;\theta,\xi)\,(u_\Omega, u_\Omega). \tag{9.8}$$

Remark 9.2.1. *The computation here is only formal. However, it has the benefit to show the form of the second-order derivatives. For example, in the parametric case of thickness optimization in Chapter 6, the second-order derivative of the compliance reads (see Lemma 6.2.3)*

$$J_1{}''(h;k,l) = -2\int_\Omega l(x)\nabla u(x)\cdot\nabla u'_k(x)\,dx = 2\int_\Omega h(x)\nabla u'_l(x)\cdot\nabla u'_k(x)\,dx.$$

The second term of (9.8) does not appear since the elasticity operator depends linearly on the design. In this case, the linear elasticity operator reads

$$\mathcal{A}(\Omega)(v,w) = \int_\Omega h\nabla v\cdot\nabla w.$$

Similarly for the geometric framework for shape optimization of Chapter 7, the second-order derivative reads

$$J_1{}''(\Omega;\theta,\xi) = 2\int_\Omega \nabla u'_\theta\cdot\nabla u'_\xi - \int_\Gamma (\theta\cdot\boldsymbol{n})\,(\xi\cdot\boldsymbol{n})\left(\partial_{\boldsymbol{n}}|\nabla u|^2 + \mathcal{H}|\nabla u|^2\right) - \int_\Gamma Z_{\theta,\xi}|\nabla u|^2,$$

where

$$Z_{\theta,\xi} = \big((\xi_\Gamma\cdot\nabla_\Gamma)\,\boldsymbol{n}\big)\cdot\theta_\Gamma - \nabla_\Gamma(\theta\cdot\boldsymbol{n})\cdot\xi_\Gamma - \nabla_\Gamma(\xi\cdot\boldsymbol{n})\cdot\theta_\Gamma.$$

The first term $2\displaystyle\int_\Omega \nabla u'_\theta\cdot\nabla u'_\xi$ corresponds to $2\,\mathcal{A}(\Omega)(u'_{\Omega,\theta}, u'_{\Omega,\xi})$ whereas the remaining terms can be seen as a variation of the elliptic operator with respect to the design.

Finally for optimization with the discretize-then-optimize *framework described in Chapter 8, the second-order derivative reads (see Lemma 8.3.3)*

$$\delta^2[J_{1,d}\circ\rho](\phi; d\phi_1, d\phi_2) = -\mathscr{A}\Big(d^2\rho(\phi; d\phi_1; d\phi_2), u, u\Big) + 2\mathscr{A}\Big(\rho, du\big(\nabla\rho\,d\phi_1\big), du\big(\nabla\rho\,d\phi_2\big)\Big),$$

which is again of the same form.

The computation of the second-order derivative implies to invert the elliptic operator $\mathcal{A}$. Even if the last term implying the derivative of the elliptic operator with respect to the design may lead to numerical difficulties, the computationally expensive part of the second-order derivative is the inversion of $\mathcal{A}$. This is the reason why in this section we focus on approximating the first component :

$$\mathcal{A}(\Omega)^{-1}\Big(g(u,\xi), g(u,\theta)\Big).$$

To that end, we present different methods based on the idea of approximating the inverse operator $\mathcal{A}(\Omega)^{-1}$. The main advantage is to reduce the cost of the computation while keeping the structure of the derivative, since the right-hand sides $g(u,\theta)$ and $g(u,\xi)$ will be exactly computed.

With these approximations, we can then use the exact same method as the Newton method presented in Section 9.1.1. For the approximated CG-Newton method, we replace the resolution of (9.3) by Algorithm 9.2.4. For all other approximation methods, for solving (9.3) we use the approximated second-order derivative instead of $l_{p,2}$.

9.2.1 BFGS-like method

The BFGS method intends to build iteratively a sequence $(B_k)_{k\in\mathbb{N}}$ as an approximation of the second-order derivative of a function, thanks to the variation of the design and the gradient of the function between two successive iterations (see Section 1.2.3). At iteration k, a new approximation of the Hessian B_{k+1} is searched such that

$$B_{k+1}s_k = y_k, \tag{9.9}$$

where s_k (and respectively y_k) is the difference between the last two designs (respectively gradients). The iterate B_{k+1} is usually built as a rank-one or rank-two update of the previous approximation B_k. It is also possible to build an approximation of the inverse of B_{k+1} instead, with the same update-rules and requirement.

In our case, we will also build an iterative sequence $(H_k)_{k\in\mathbb{N}}$ as an approximation of the inverse of the operator $\mathcal{A}$. We will consider the same kinds of rank-one or rank-two updates of the inverse, but with a requirement different from (9.9). We will impose that

$$H_{k+1}f = u_{\Omega_k}, \tag{9.10}$$

where u_{Ω_k} is the state, solution to $\mathcal{A}(\Omega_k)u_{\Omega_k} = f$. For instance, the rank-two update will look like

$$H_{k+1} = \left(I - \frac{1}{{u_{\Omega_k}}^T f}\, u_{\Omega_k}\, f^T\right) H_k \left(I - \frac{1}{{u_{\Omega_k}}^T f}\, f\, {u_{\Omega_k}}^T\right) + \frac{1}{{u_{\Omega_k}}^T f}\, u_{\Omega_k}\, {u_{\Omega_k}}^T. \tag{9.11}$$

Remark 9.2.2. *In case of geometric optimization, the BFGS method is quite tricky to apply. For instance, the shape derivative is only defined on the boundary of a shape. Therefore for comparing two successive shape derivatives, one should first map one shape to the other one, and then transport the shape derivative accordingly. From a numerical point of view, the discretization of the shape derivative may be defined for example on every node in the vicinity of the boundary. For two successive shapes, the set of nodes in the vicinity of the boundary may change, which implies that two successive shape derivatives may not have the same dimensions.*

The void domain is often represented by an ersatz material [6, 15]. During the optimization process, some parts of the shape may be disconnected. The ersatz material allows one to avoid ill-conditioning of the stiffness matrix in this case. The use of an ersatz material makes also the stiffness matrix as well as the state u_Ω be defined on the whole computational domain that does not change. This means that the stiffness matrix and the state u_Ω have the same dimensions from one iteration to the next one. Therefore it is possible to compare these quantities between two successive iterates and use an update rule like (9.11).

9.2.2 Inexact-Newton method

The costly part of the computation of the second-order derivative is to invert the operator $\mathcal{A}$. If the operator were diagonal, it would not be expensive to invert. This is the reason why we would like to approximate the stiffness matrix by its diagonal. This approximation is extremely rough but it keeps the structure of the second-order derivative since the right-hand side $g(u,\theta)$ are exactly computed. As we will see later (see Section 15.5.4), the second-order derivative can be less accurate than the first-order derivative while keeping the quadratic rate of convergence. Therefore the roughness of the approximation may not be that problematic.

9.2.3 Incomplete LU-factorization method

Looking at the structure of the second-order derivative, we can observe that we don't exactly need to invert the elliptic operator $\mathcal{A}$. It is only required to compute

$$\mathcal{A}(\Omega)^{-1} g(u_\Omega, \theta), \tag{9.12}$$

for every $\theta \in \mathcal{F}$. After discretization, the number of linear systems to solve is the dimension of $\mathcal{F}$. However, the matrix of the operator is still the same for each right-hand side. Then if it is possible to factorize the operator $\mathcal{A}$ with a Cholesky or a LU method for example, the computational cost may not be prohibitive. In this section we will consider the LU factorization method. The idea is to compute two matrices L and U that are respectively lower and upper triangular, and such that their product is equal to the matrix to factorize : $LU = \mathcal{A}$. With such a factorization, solving a linear system for $\mathcal{A}$ becomes straightforward : it suffices to solve two triangular systems.

When the matrix is sparse and of high dimension, the LU factorization method can be approximated. It briefly consists in computing the pivots only for non-zero entries of the matrix to factorize. We will not enter into the details of this approximation method and refer to [16, Exercise 6.10] for example. Once an approximated factorization is performed, it is possible to solve all linear systems (9.12) very rapidly, and then to compute quite easily an approximation of the second-order derivative.

9.2.4 Approximate CG-Newton method

The Newton method intends to find at each iteration the minimum of the quadratic approximation of the objective :

$$\mathcal{F} \ni \theta \mapsto Q(\theta) = {J_1}'(\Omega;\theta) + \frac{1}{2}{J_1}''(\Omega;\theta,\theta)\,. \tag{9.13}$$

When the bilinear map $(\theta, \xi) \mapsto J_1{}''(\Omega; \theta, \theta)$ is positive definite, one can use the well-known conjugate-gradient algorithm to find a solution to the Newton equation. Since for shape optimization problems the second-order derivative may not be positive definite, we use a modified version of this algorithm. This is an iterative method for minimizing (9.13). In order to be consistent with the trust-region imposed for the Newton method Section 9.1.1 we also project at each iteration of this modified conjugate-gradient algorithm the state θ_k on the trust-region.

In order to ease the reading, we start by recalling the classical conjugate-gradient algorithm for minimizing $F(x) = \frac{1}{2} x^T A\, x - b^T\, x$ with $x \in \mathbb{R}^n$ and A is a $n \times n$ symmetric, positive definite matrix.

Algorithm 9.2.3. *[92, Algorithm 5.2]*

1. *The initial state is* $x_0 = 0$.
2. *The initial residual* r_0 *is defined by* $r_0 = A\, x_0 - b$.
3. *The initial descent direction is given by* $w_0 = r_0$.
4. *Loop*
   while $r_k \neq 0$ **do**
   (a) *Set*
   $$\alpha_k = \frac{{r_k}^T\, r_k}{{w_k}^T A\, w_k}.$$
   (b) *The new state is updated with* $x_{k+1} = x_k - \alpha_k w_k$.
   (c) *Calculate the new residual* $r_{k+1} = r_k - \alpha_k A\, w_k$.
   (d) *Set*
   $$\gamma_k = \frac{{r_{k+1}}^T\, r_{k+1}}{{r_k}^T\, r_k}.$$
   (e) *Update the descent direction with* $w_{k+1} = r_{k+1} + \gamma_k w_k$.
   end
5. **return** x_k.

In the case of the functional Q defined by (9.13), we replace the mapping $(x, y) \mapsto x^T A\, y$ by the mapping $(\theta, \xi) \mapsto J_1{}''(\Omega; \theta, \xi)$. We also adapt the classical conjugate-gradient algorithm to the case when the bilinear map J_1 is not positive definite. Finally we truly thank C. Dapogny for all his ideas and advices, and that was a great help on this topic.

Algorithm 9.2.4. *Approximated conjugate-gradient algorithm.*

1. *The initial state is* $\theta_0 = 0$.
2. *The initial residual* r_0 *is defined by the unique solution to*
   $$\forall \xi \in \mathcal{F}, \quad r_0 \cdot \xi = J_1{}'(\Omega; \xi) + J_1{}''(\Omega; \theta_0, \xi)$$
3. *The initial descent direction is given by* $w_0 = r_0$.
4. *Loop*
   for k=0 ... until convergence **do**
   (a) *Calculate the mapping* $\xi \mapsto J_1{}''(\Omega; w_k, \xi)$
   (b) *If* $J_1{}''(\Omega; w_k, w_k)$ *is negative or "very small"* **break**. *Else calculate the coefficient*
   $$\alpha_k = \frac{r_k \cdot r_k}{J_1{}''(\Omega; w_k, w_k)}.$$
   (c) *The new state is updated with* $\theta_{k+1} = \theta_k - \alpha_k w_k$. *The new state is projected on the space* $\{v \in L^\infty \mid \|L^\infty\|_v < v_M\}$
   (d) *Calculate the new residual which is the solution to*
   $$\forall \xi \in \mathcal{F}, \quad r_{k+1} \cdot \xi = r_k \cdot \xi - \alpha_k J_1{}''(\Omega; w_k, \xi)\,.$$
   (e) *If the residual is "small", that is if*
   $$\frac{r_{k+1} \cdot r_{k+1}}{r_0 \cdot r_0} \leq \varepsilon,$$
   for a user-defined threshold ε, **break**. *Else, evaluate*
   $$\gamma_k = \frac{r_{k+1} \cdot r_{k+1}}{r_k \cdot r_k}.$$

(f) Update the descent direction with $w_{k+1} = r_{k+1} + \gamma_k w_k$.

end

5. **return** θ_k.

This algorithm is not exactly a conjugate-gradient algorithm, since it stops if the direction w_k comes in a region where the bilinear map is not positive definite. However, it appears that the direction is always a descent direction.

In the numerical examples, the use of the approximated conjugate-gradient algorithm above will replace the resolution of (9.3).

One of the main advantages of this method is that there is no need to assemble the whole mapping $(\theta, \xi) \mapsto J_1''(\Omega; \theta, \xi)$. We only need to compute mappings $\xi \mapsto J_1''(\Omega; \theta, \xi)$ for a given θ.

9.3 Thickness optimization

For the thickness optimization of a plate, we consider only the following load case with a thickness varying between $h_m = 0.1$ and $h_M = 1$. With a Lagrange multiplier $\Lambda = 5$ we aim at minimizing the following Lagrangian

$$L(h) = J(h) + \Lambda V(h),$$

where J is the compliance for the considered load case and V is the volume of the shape. The force applied is unitary, the Young's modulus and the Poisson's coefficient are defined by $E = 1$ and $\nu = 0.3$.

The shape is characterized by the thickness map h_p, and the descent direction v takes place in $L^\infty(\Omega; \mathbb{R})$. Then at iteration $p \in \mathbb{N}$, for a time t, the update of the shape reads

$$h_{p+1} = \max\Big(\min(h_p + tv, h_M), h_m\Big),$$

so that the update stays in $L^\infty(\Omega; [h_m; h_M])$.

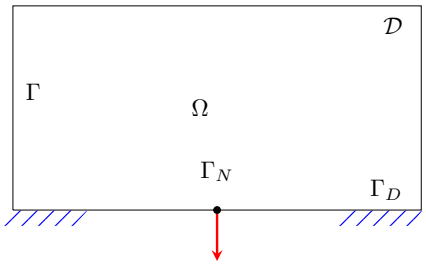


Figure 9.1: Boundary condition for the arch.

As regards the optimization method, we will compare the gradient method (see Section 9.1.2) to different second-order methods (see Section 9.1.1) where the second-order derivative - namely $l_{p,2}$ - is computed exactly or only approximated. At first we consider the Newton method with the exact Hessian. Secondly, we try the Newton method with two approximations of the Hessian. These approximations are presented in Section 9.2.3 (respectively Section 9.2.1) and referred to as the *Inexact Newton's method* (respectively the *BFGS-like method*). The different optimal thicknesses are represented in Figure 9.2, and the final numerical values are given in Table 9.2.

	Iterations	Lagrangian
Gradient	330	18.301767
Newton	12	18.243249
Inexact Newton	147	18.301749
BFGS-like	174	18.301782

Table 9.2: Convergence table.

For each method, we denote by Ω_∞ the final shape. We also define $\Omega^\times$ as being the final shape for which the objective function is minimal. The numerical experiments are done with FreeFem [71]. The exact second-order method

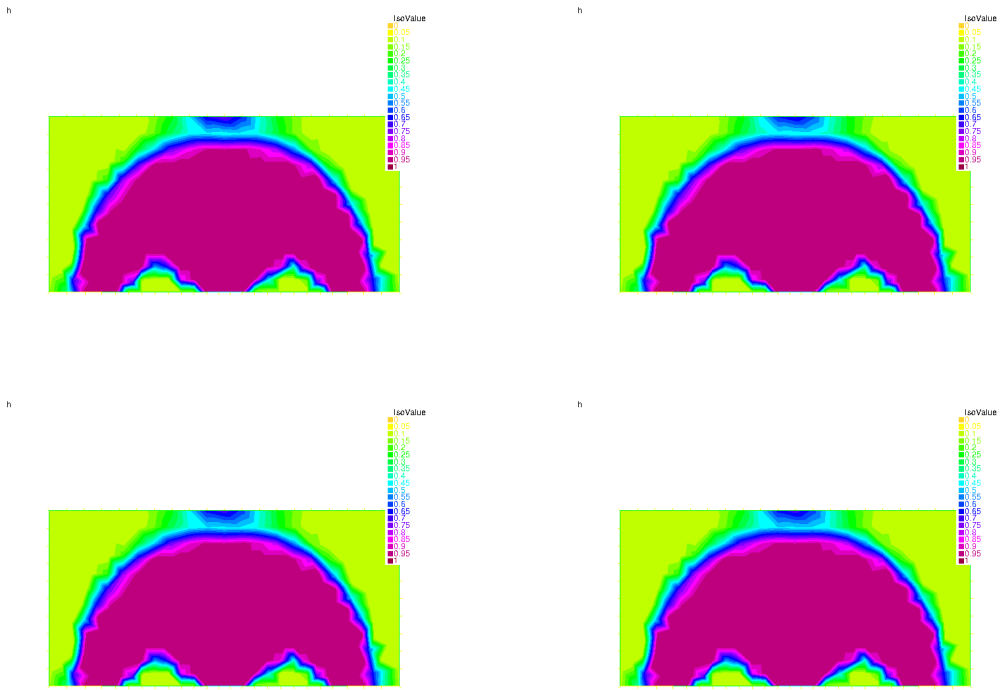


Figure 9.2: Final domains. Gradient method (top left), Newton's method (top right), Inexact Newton's method (bottom left) and BFGS-like method (bottom right).

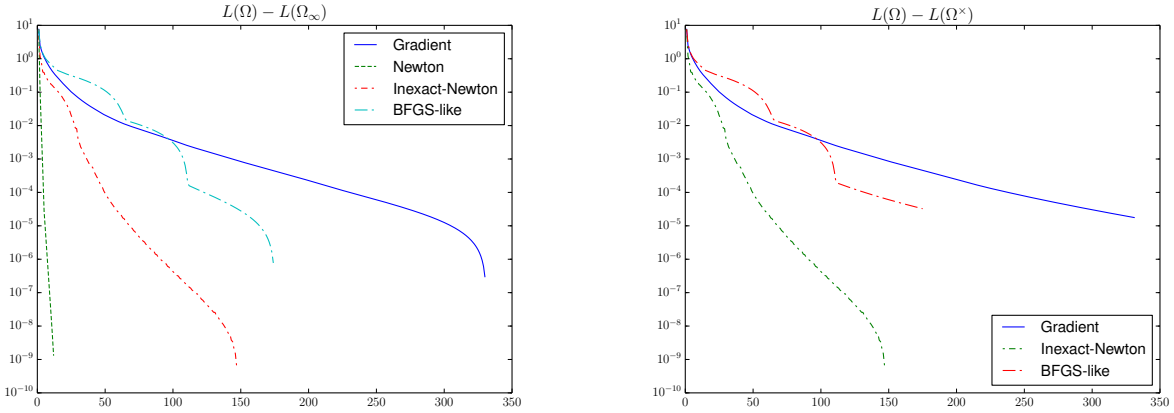


Figure 9.3: Convergence of $L(\Omega) - L(\Omega_\infty)$ (left) and $L(\Omega) - L(\Omega^\times)$ (right) with respect to the number of iterations.

converges much more rapidly to a shape whose objective function is much better than with the other method. Thus, we will no plot the convergence of $L(\Omega) - L(\Omega^\times)$ for this method.

We can observe that the Newton method converges way faster than the other methods, and to the best of all final shapes. The other two second-order methods also seem to converge more rapidly than the gradient method but the improvement is less impressive.

9.4 Discrete optimization

Now we consider the case of discrete optimization of Chapter 8. Like previously, we consider the minimization of a Lagrange function

$$L(\Omega) = J_1(\Omega) + \Lambda V(\Omega),$$

where J_1 is the compliance for the load case of Figure 9.1, and the Lagrange multiplier is fixed to $\Lambda = 0.1$. The applied point load is also $g = 1$, with a Young modulus $E = 1$ and a Poisson's coefficient $\nu = 0.3$.

The shape is modeled by the level-set method. The descent direction is found by a gradient-like equation (9.4) or the Newton's equation (9.3). Similarly to the *optimize-then-discretize* framework for shape optimization, the shape derivatives depend only on the values on the boundary of the velocity field. Therefore, the direction found by the gradient-like or Newton's method has to be extended to the whole computational domain. This is done by a classical regularization technique (Method 10.2.3). The shape is updated by the Hamilton-Jacobi transport equation, with a time parameter given by a backtracking line search. The numerical scheme used for the transport equation is the one of Section 4.2.2. We do not give many details here about the optimization method and refer to Section 10.2 and Section 10.3. The principle is exactly the same, and the only difference is the expression of the derivatives.

In this framework for optimization, we will compare the gradient algorithm with two second-order methods : the Newton method with the exact second-order derivative, and the approximate CG-Newton method.

9.4.1 Arch - initialization without holes

We consider the *discretize-then-optimize* approach for the minimization of the compliance under the same boundary conditions as in Figure 9.1.

At first we consider the consistency of the derivative, in particular for the compliance criterion J_1. For a given shape Ω_p, with a given velocity field v, we plot $J_1(\Omega(p, t, v))$ for different time steps t. We compare it to the linear approximation $J_1(\Omega) + tJ_1(\Omega; v)$. We can observe on Figure 9.4 that the derivative is consistent, in the sense that the linear approximation is tangent to the real curve. This result was expected since the consistency is a priori the main advantage of the *discretize-then-optimize* approach.

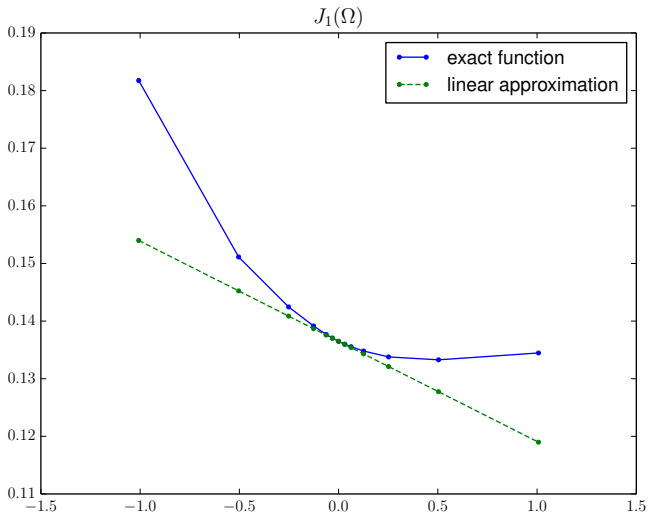


Figure 9.4: Consistency of the derivative.

Now we can have a look at the final results for shape optimization in Table 9.4. The final values of the Lagrangian are also similar, even if it is quite better with the gradient-like method. Looking at the iteration number, we can also see that there is no gain in using the Newton method or an approximated second-order method. As concerns the consistency of the approach, we also include in Table 9.4 an indicator of the decrease of the gradient. With v^0 and v^∞ being the first and last directions taken by the optimization process, we consider the ratio

$$\frac{\delta L(\Omega; v^\infty)}{\delta L(\Omega; v^0)}. \tag{9.14}$$

This quantity is reported in the third column of Table 9.4. We can observe that it is rather small, meaning that at the end there is no real hope to decrease the objective function. This indicates also that the discrete optimality conditions are somehow reached.

The Figure 9.6 represents the initial shape and all final shapes. At first we can notice that the final shapes look all the same. We denote Ω_∞ each final shape, and $\Omega^\times$ the best of the three final shapes and plot the convergence curves of $L(\Omega) - L(\Omega_\infty)$ and $L(\Omega) - L(\Omega^\times)$ (see Figure 9.5).

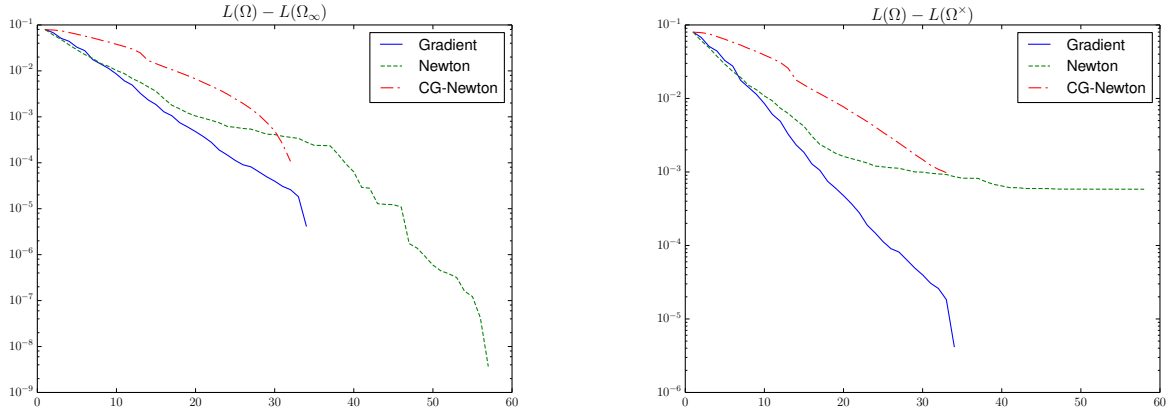


Figure 9.5: Convergence of $L(\Omega) - L(\Omega_\infty)$ (left) and $L(\Omega) - L(\Omega^\times)$ (right) for Section 9.4.1 with respect to the number of iterations.

The gradient method converges more rapidly to a better solution than the two second-order methods. We can also see that the second-order methods do not perform well, and there is no interest in computing the second-order derivatives. These last method converges indeed with more iterations to a shape that decreases less the objective function.

9.4.2 Arch - initialization with holes

Here we consider the exact same minimization problem but with a different initialization. The initial design is plotted at the top-left corner of Figure 9.7. Similarly to the previous example, we also report in Table 9.6 the final values of the objective, the number of iterations, and the ratio

$$\frac{\delta L(\Omega; v^\infty)}{\delta L(\Omega; v^0)}. \tag{9.15}$$

The conclusions are similar to the previous example. Indeed, there is no gain in computing second-order derivatives.

We keep the same notation as in the previous numerical example as concerns Ω_∞ and $\Omega^\times$, and plot the convergence curve of $L(\Omega) - L(\Omega_\infty)$ and $L(\Omega) - L(\Omega^\times)$.

	Iterations	Lagrangian	Ratio (9.14)
Gradient	35	0.235502	5.620×10^{-3}
Newton	58	0.236085	3.000×10^{-3}
CG-Newton	33	0.236480	2.488×10^{-2}

Table 9.4: Convergence table for Section 9.4.1.

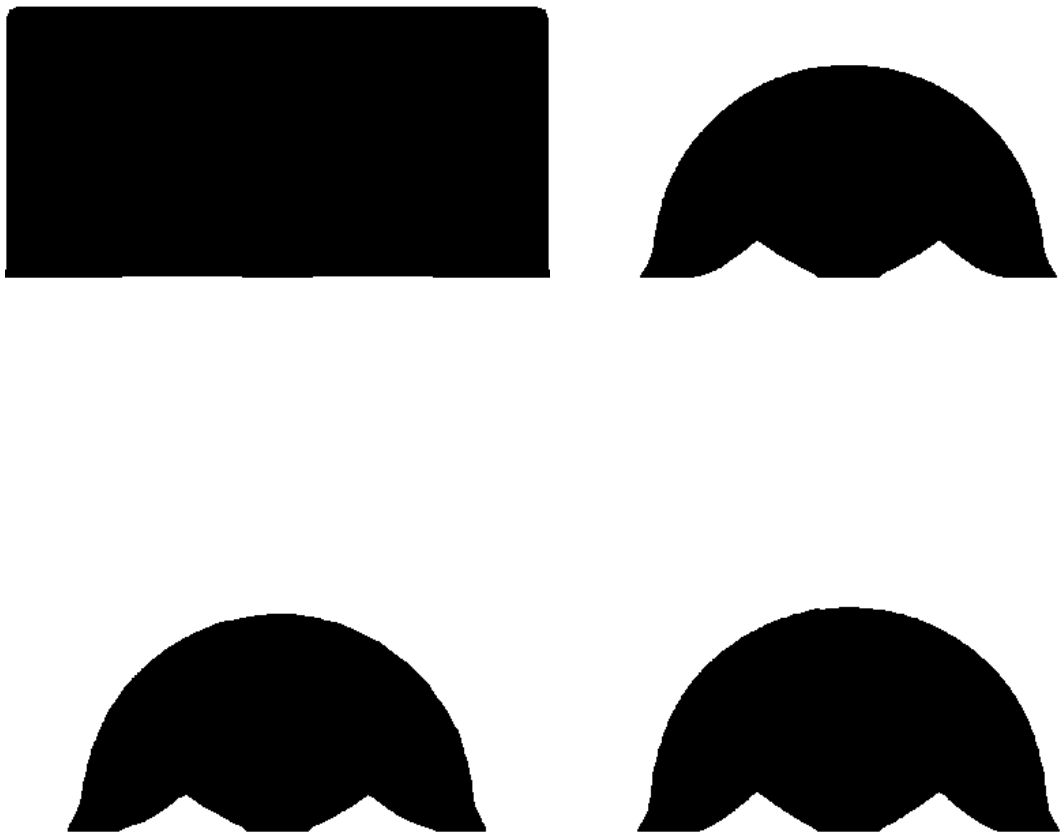


Figure 9.6: Final domains for Section 9.4.1. Gradient method (top left), Newton's method (top right), Inexact Newton's method (bottom left) and BFGS-like method (bottom right).

	Iterations	Lagrangian	Ratio (9.14)
Gradient	46	0.235509	1.433×10^{-3}
Newton	43	0.235991	4.685×10^{-3}
CG-Newton	43	0.236462	6.118×10^{-3}

Table 9.6: Convergence table for Section 9.4.2.

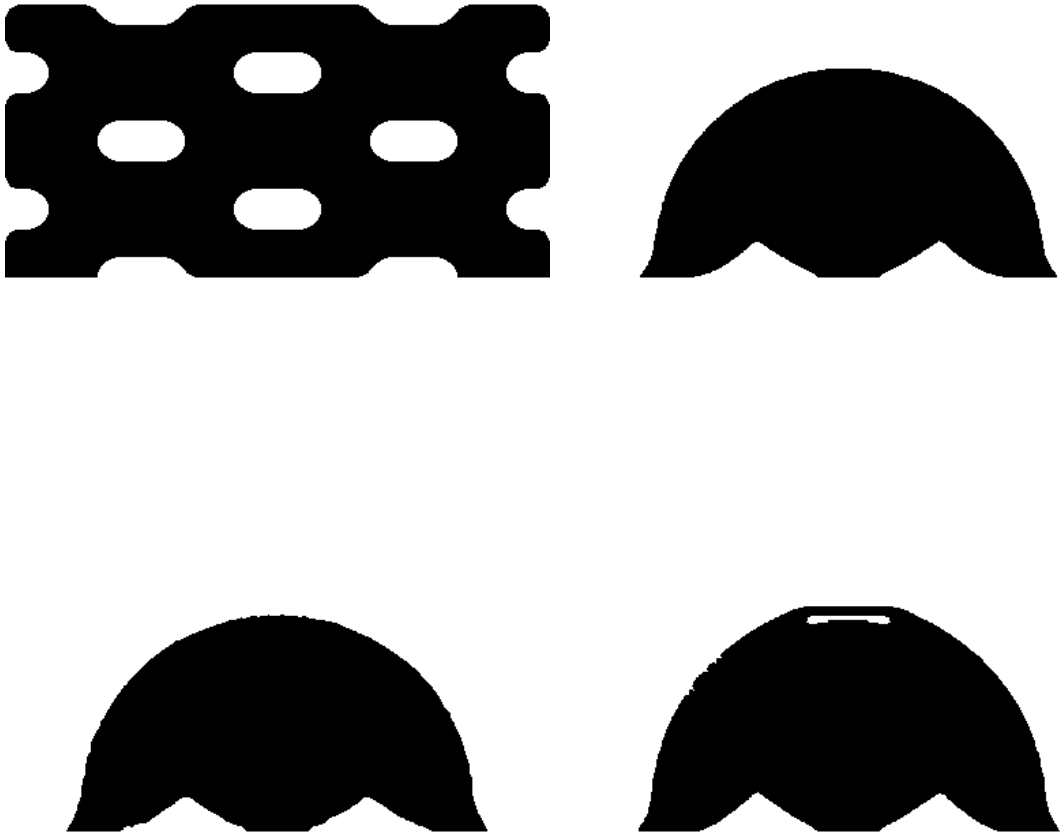


Figure 9.7: Final domains for Section 9.4.2. Gradient method (top left), Newton's method (top right), Inexact Newton's method (bottom left) and BFGS-like method (bottom right).

The non-efficiency of second-order derivatives is very questioning. Indeed, after discretization, the optimization problem takes place in a finite-dimensional space where the Newton method as well as approximate second-order methods have already proven to give superlinear convergence rate. There might be different explanations.

At first, we can wonder if the sufficient regularity required (see Section 1.2) is reached. There are many reasons why this could not be the case. For example, if we consider the dependency of the density ρ with respect to the level-set function ϕ, it is assumed that the level-set never vanishes at a node of the mesh (otherwise there are discontinuities of the derivatives). This assumption is quite strong since there is no reason for the level-set not to vanish on a node, or very close to a node of the mesh. Therefore, there should be discontinuities in the derivative of the density with respect to the level-set, especially for second-order derivatives. Moreover we can have a look at the dependency of the level-set on a velocity field through the Hamilton-Jacobi equation. The numerical scheme used might not be sufficiently smooth since there are min and max functions.

As a conclusion, we could say that the *discretize-then-optimize* approach might be efficient for first-order algorithms, since it appears to be consistent. However, for second-order optimization, the whole numerical method seems not to be sufficiently smooth to lead to superlinear convergence.

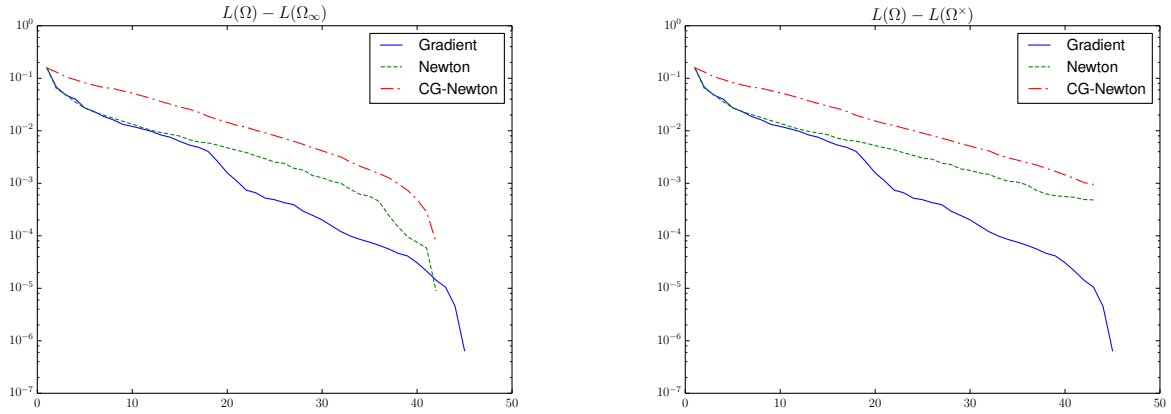


Figure 9.8: Convergence of $L(\Omega) - L(\Omega_\infty)$ (left) and $L(\Omega) - L(\Omega^\times)$ (right) for Section 9.4.2 with respect to the number of iterations.

Part III
Derivation along normal trajectories

"We are such stuff
as dreams are made on."
W. Shakespeare

Chapter 10

Shape derivation along normal trajectories

Contents

We recall from Section 3.2 that there are two variants for shape differentiation with the Hadamard method. The first one - the displacement field method - consists in considering parametrization of shapes by vector fields. Given a reference domain $\Omega \in \mathbb{R}^d$, a variation of this domain is of the type $\Omega_\theta = (\mathrm{Id} + \theta)(\Omega)$ where θ is a smooth vector field from $\mathbb{R}^d$ to $\mathbb{R}^d$. There is also a second approach, called the speed method, based on shapes evolving along the flow of a vector field. For a given vector field V defined from $\mathbb{R}_+ \times \mathbb{R}^d$ to $\mathbb{R}^d$, the evolving shape is defined for $t \geq 0$ by

$$\Omega_t := \{X_V(t,x) \mid x \in \Omega\},$$

where X_V is the solution (or flow) of the partial differential equation

$$\begin{cases} \dfrac{\partial X_V}{\partial t}(t,x) = V\big(t, X_V(t,x)\big) & \text{for } t > 0\,, \\ X_V(0,x) = x\,. \end{cases} \tag{10.1}$$

We refer to Section 3.2 for more details on this two usual frameworks for computing shape derivatives, and the respective structures of the second-order shape derivatives.

The present chapter is the subject of a joint work with G. Allaire and E. Cancès [9]. Its goal is to define a third approach of Hadamard shape differentiation by considering again a family of shapes Ω_t, evolving with time $t \geq 0$ in the direction of the normal vector $\mathbf{n}(t)$ to the boundary $\partial\Omega_t$. This new approach is especially adapted to the level-set method for shape and topology optimization [14], [15], [129] where the shape is indeed advected in its normal direction by solving an eikonal or Hamilton-Jacobi equation. Following the lead of Osher and Sethian [97], a shape Ω is represented by a level-set function ϕ which, by definition, satisfies

$$\begin{cases} x \in \Omega & \text{iff } \phi(x) < 0, \\ x \in \partial\Omega & \text{iff } \phi(x) = 0, \\ x \in \mathbb{R}^d \setminus (\Omega \cup \partial\Omega) & \text{iff } \phi(x) > 0. \end{cases} \tag{10.2}$$

In other words, the boundary of Ω is given by the zero level-set $\{x \in \mathbb{R}^d \mid \phi(x) = 0\}$. Let $v(t,x)$ be a smooth function from $\mathbb{R}_+ \times \mathbb{R}^d$ into $\mathbb{R}$. Evolving the shape Ω_t with a normal velocity v is equivalent to solving the following eikonal or Hamilton-Jacobi equation

$$\begin{cases} \dfrac{\partial \varphi}{\partial t}(t,x) + v(t,x)\,|\nabla_x \varphi(t,x)| = 0, \\ \varphi(0,x) = \phi(x). \end{cases} \tag{10.3}$$

For $t \geq 0$, the shape Ω_t is recovered as the set of negative values of $\varphi(t,\cdot)$, namely $\Omega_t = \{x \in \mathbb{R}^d \mid \varphi(t,x) < 0\}$. Here we give a rigorous definition of shape differentiation for such evolutions. The key ingredient is the method of

bicharacteristics for solving (10.3) when its solution φ is smooth. It is well known [102] that smooth solutions of (10.3) can be computed by solving the following system of two ordinary differential equations

$$\left\{ \begin{array}{lll} \dfrac{dx}{dt}(t) & = & \nabla_p H\big(t, x(t), p(t)\big), \\ \dfrac{dp}{dt}(t) & = & -\nabla_x H\big(t, x(t), p(t)\big), \end{array} \right. \tag{10.4}$$

where $H(t,x,p) = v(t,x)|p|$ is the Hamiltonian defined on $\mathbb{R}_+ \times \mathbb{R}^d \times \mathbb{R}^d$. For smooth evolutions[1], an equivalent definition of Ω_t is then $\partial\Omega_t = \{x(t)$ solution of (10.4) with $x(0) \in \partial\Omega\}$. By using (10.4) we can introduce a new definition of shape derivative which is different from the two previous ones (note that (10.4) cannot be put under the simpler form (10.1)). It turns out that the first-order shape derivatives for all three variants are the same and it is only the second-order shape derivative which is different and simpler with our approach since it does not involve any tangential displacement of the boundary as in the two previous methods. Our main theoretical result is Theorem 10.1.4.

There are two main applications of the knowledge of second-order derivatives. The most obvious one is the definition of the Newton algorithm. We explore this issue in Section 10.3. We are not the first ones to study second-order shape derivatives and the Newton algorithm for shape optimization. Let us mention the stability results in [48], [50], and the numerical algorithms in [95], [104]. Much more is even known in the context of control theory [78]. Another application of second-order derivative is presented in Section 10.2 for the first time, to the best of our knowledge. It turns out to be useful for the extension of a descent direction, merely known on the boundary $\partial\Omega$, to the entire space $\mathbb{R}^d$, as required for solving the Hamilton-Jacobi equation (10.3). The main idea is that the structure of the second-order shape derivative gives a hint on the choice of the normal derivative of the extension on $\partial\Omega$. Our numerical experiments indicate that this idea leads to better extensions in the sense that the convergence of the optimization process is improved.

This chapter is organized as follows. Section 10.1 describes the proposed new setting of shape differentiation when the shape evolves in its normal direction which is the typical case in the level-set method for shape optimization. In Section 10.2 we take advantage of the structure of the second-order shape derivative to present new extension methods for a descent direction, from $\partial\Omega$ to $\mathbb{R}^d$. Section 10.3 is concerned with optimization algorithms presented in Section 9.1 and how to use them in the current framework. Technical details as well as numerical examples will be discussed later in Chapter 11 and Chapter 12.

10.1 Shape derivation with respect to normal evolution

In this section we introduce a new variant of shape differentiation which makes sense for domains that evolve in the direction of their normal vector, as it is the case in the level-set method [97], [46] (see also Figure 10.1). Note that the case of a normal evolution is a priori not covered by the flow of equation (10.1) since it is not always obvious that there exists a velocity field V which stays parallel to the normal vector for times $t > 0$, even if it is so at the initial time $t = 0$ (in the sense of : orthogonal to the "vertical" direction $\mathbf{n}_0$). There are simple examples of initial geometries and normal flows : indeed, take an initial disk with a radial velocity. But it is easy to construct instances of non-normal flows too. Take for example an initially flat part of the boundary (with constant unit normal n_0) with a velocity $V(x)$ which is everywhere parallel to n_0 and affine (non-constant) with respect to the tangential variable $(x - (x \cdot n_0)n_0)$. Clearly, for any later time $t > 0$, the boundary is not anymore flat since points on the initially flat part move with different velocities. Therefore, the unit normal to this part of $\partial\Omega(t)$ is no longer parallel to n_0 and thus to V. By a blow-up argument, this simple example can be extended to more general situations (at the price of some technicalities). Nevertheless, the arguments below prove the existence of at least one normal velocity V (which is not explicit in terms of the initial boundary), as stated in Remark 10.1.6. The main new idea here is to introduce a set of two coupled ordinary differential equations (the so-called bicharacteristics) which are equivalent to the level-set equation, in the case of smooth domains. We first recall the method of bicharacteristics for solving Hamilton-Jacobi equations, of which the level-set equation is a particular case.

10.1.1 Bicharacteristics method for solving Hamilton-Jacobi equation

Let $H(t,x,p) : \mathbb{R}_+ \times \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a smooth function, called a Hamiltonian in this context. For $(t,x) \in \mathbb{R}_+ \times \mathbb{R}^d$, we consider the scalar first-order partial differential equation

$$\partial_t \phi(t,x) + H\big(t, x, \nabla_x \phi(t,x)\big) = 0, \tag{10.5}$$

with unknown $\phi : \mathbb{R}_+ \times \mathbb{R}^d \to \mathbb{R}$. Usually, (10.5) is complemented by initial and boundary conditions. Here, ∂_t, ∇_x and ∇_p respectively denote the derivation with respect to the first variable, the following d and the last d variables of $\mathbb{R}_+ \times \mathbb{R}^d \times \mathbb{R}^d$. We recall a classical result for solving (10.5).

[1] The Hamiltonian $H(t,x,p) = v(t,x)|p|$ is not smooth at $p = 0$ but it is not an issue as explained in Remark 10.1.3.

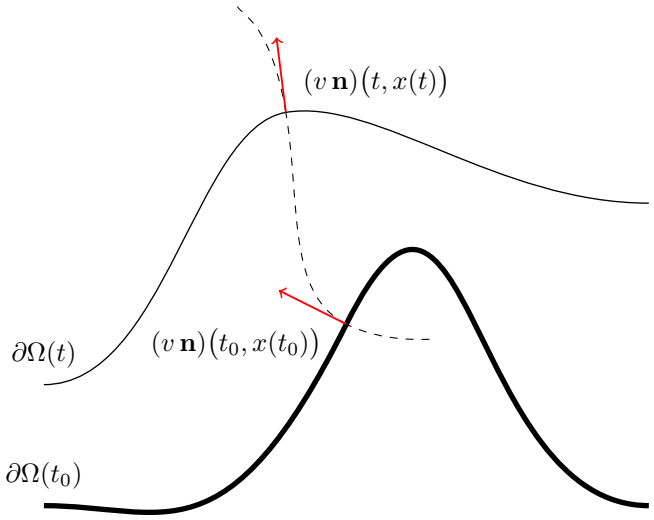


Figure 10.1: Normal motion of a shape

Lemma 10.1.1 ([102, Lemma 5.I.2])**.** *Let $\tau > 0$ and $\mathcal{O}$ be an open set of $\mathbb{R}^d$. Assume that ϕ is a smooth solution of (10.5) for $(t,x) \in [0,\tau] \times \mathcal{O}$, and that $(x(t),p(t))$, defined from $\mathbb{R}_+$ into $\mathbb{R}^d \times \mathbb{R}^d$, solves*

$$\left\{ \begin{array}{rcl} \dfrac{dx}{dt}(t) & = & \nabla_p H\big(t,x(t),p(t)\big), \\ \dfrac{dp}{dt}(t) & = & -\nabla_x H\big(t,x(t),p(t)\big). \end{array} \right. \tag{10.6}$$

If $p(0) = \nabla_x \phi(0,x(0))$, then $p(t) = \nabla_x \phi\big(t,x(t)\big)$ as long as $(t,x(t)) \in [0,\tau] \times \mathcal{O}$. Furthermore, let $t \ni \mathbb{R}_+ \mapsto (y(t),r(t)) \in \mathbb{R} \times \mathbb{R}$ be the solution of

$$\left\{ \begin{array}{rl} \dfrac{dy}{dt}(t) & = -\partial_t H\big(t,x(t),p(t)\big), \\ \dfrac{dr}{dt}(t) & = y(t) + p(t) \cdot \nabla_p H\big(t,x(t),p(t)\big), \end{array} \right. \tag{10.7}$$

with $y(0) = \partial_t \phi(0,x(0))$, $r(0) = \phi\big(0,x(0)\big)$. Then $y(t) = \partial_t \phi(t,x(t))$ and $r(t) = \phi\big(t,x(t)\big)$ as long as $(t,x(t)) \in [0,\tau] \times \mathcal{O}$.

Remark 10.1.2. *Equation* (10.6) *is a Hamiltonian system, called the bicharacteristics system of* (10.5)*, which admits global solution in time since the value of the Hamiltonian function $H\big(t,x(t),p(t)\big)$ is conserved along the flow. Lemma 10.1.1 roughly states that, in order to find a smooth solution of the Hamilton-Jacobi equation* (10.5)*, it is enough to solve the Hamiltonian system* (10.6)*. This construction breaks down as soon as the projection on the configuration space $\mathbb{R}^d$ of the bicharacteristics $(x(t),p(t))$ in the phase space $\mathbb{R}^d \times \mathbb{R}^d$ cross, implying that $\phi\big(t,x(t)\big)$ is multivalued. In such a case,* (10.5) *has no smooth solution and one should resort to the notion of viscosity solutions [46]. However, for short time τ there exists a unique smooth solution of* (10.5) *[102]. Note that* (10.7) *is decoupled from* (10.6) *and is just a kind of post-processing phase of the bicharacteristics method.*

For the sake of completeness, we provide a proof of Lemma 10.1.1.

Proof. To simplify the notation we write $\dot{z}(t)$ instead of $\dfrac{dz}{dt}(t)$. Let ϕ be a smooth solution of (10.5) on $[0,\tau] \times \mathcal{O}$ and define $X(t) : [0,\tau] \to \mathbb{R}^d$ the solution of the following ordinary differential equation (o.d.e.)

$$\left\{ \begin{array}{l} \dot{X}(t) = \nabla_p H\Big(t,X(t),\nabla_x \phi\big(t,X(t)\big)\Big), \\ X(0) = x(0), \end{array} \right. \tag{10.8}$$

which exists, at least for short times, by virtue of the Cauchy-Lipschitz theorem. Let Y,P,R be defined by

$$Y(t) = \partial_t \phi\big(t,X(t)\big), \quad P(t) = \nabla_x \phi\big(t,X(t)\big), \quad R(t) = \phi\big(t,X(t)\big).$$

We now prove that X, Y, P, R coincide with x, y, p, r. By rewriting (10.8) and differentiating the definitions of Y, P, R, we obtain

$$\dot{X}(t) = \nabla_p H\Big(t, X(t), P(t)\Big), \qquad \dot{Y}(t) = \partial_{tt}\phi\big(t, X(t)\big) + \partial_t \nabla_x \phi\big(t, X(t)\big) \cdot \dot{X}(t),$$
$$\dot{P}(t) = \partial_t \nabla_x \phi\big(t, X(t)\big) + \nabla_x^2 \phi\big(t, X(t)\big)\dot{X}(t), \quad \dot{R}(t) = \partial_t \phi\big(t, X(t)\big) + \nabla_x \phi\big(t, X(t)\big) \cdot \dot{X}(t).$$

With the definition of Y, P and the o.d.e. for X, we already get an o.d.e. for R, similar to that for r in (10.7),

$$\dot{R}(t) = Y(t) + P(t) \cdot \nabla_p H\big(t, X(t), P(t)\big).$$

Differentiating the Hamilton-Jacobi equation (10.5) with respect to t and x respectively leads to

$$\partial_{tt}\phi + \partial_t H + \partial_t \nabla_x \phi \cdot \nabla_p H = 0, \quad \text{and} \quad \partial_t \nabla_x \phi + \nabla_x H + \nabla_x^2 \phi \nabla_p H = 0. \tag{10.9}$$

Evaluating (10.9) at $\big(t, X(t)\big)$ (for small time, we are sure that $X(t) \in \mathcal{O}$), and using $\dot{X} = \nabla_p H$, we get

$$\partial_{tt}\phi + \partial_t \nabla \phi \cdot \dot{X} = -\partial_t H, \quad \text{and} \quad \partial_t \nabla \phi + \nabla_x^2 \phi \dot{X} = -\nabla_x H,$$

which are exactly the o.d.e.'s, similar to those for y and p,

$$\dot{Y}(t) = -\partial_t H\Big(t, X(t), P(t)\Big), \quad \text{and} \quad \dot{P}(t) = -\nabla_x H\Big(t, X(t), P(t)\Big).$$

As a result, (x, y, p, r) and (X, Y, P, R) satisfy the same differential system with the same initial condition since $X(0) = x(0)$ and $p(0) = \nabla_x \phi(0, x(0))$ by assumption. Therefore $(x, y, p, r) = (X, Y, P, R)$ and, furthermore, from the equivalent definitions of Y, P, R we deduce

$$y(t) = \partial_t \phi\big(t, x(t)\big), \quad p(t) = \nabla_x \phi\big(t, x(t)\big), \quad r(t) = \phi\big(t, x(t)\big).$$

□

A particular case of (10.5), which is crucial for the level-set method, is the eikonal equation corresponding to the Hamiltonian $H(t, x, p) = v(t, x)|p|$ where v is a smooth function from $\mathbb{R}_+ \times \mathbb{R}^d$ into $\mathbb{R}$. In other words, we specialize (10.5) to the case

$$\begin{cases} \partial_t \phi(t, x) + v(t, x)\, |\nabla_x \phi(t, x)| = 0, \\ \phi(0, x) = \phi_0(x), \end{cases} \tag{10.10}$$

where ϕ_0 is a smooth function, satisfying for some smooth bounded open set Ω_0,

$$\begin{cases} \phi_0(x) < 0 & \text{if } x \in \Omega_0, \\ \phi_0(x) = 0 & \text{if } x \in \partial\Omega_0, \\ \phi_0(x) > 0 & \text{if } x \in \mathbb{R}^d \setminus (\Omega_0 \cup \partial\Omega_0). \end{cases}$$

The evolution of the eikonal equation defines a family of domains Ω_t :

$$\Omega_t := \big\{ x \in \mathbb{R}^d \mid \phi(t, x) < 0 \big\}.$$

Applying Lemma 10.1.1 to (10.10), in the case of smooth solutions, yields $\dot{p} = \nabla_x \phi$ and

$$\dot{x} = \nabla_p H = v \frac{\nabla_x \phi}{|\nabla_x \phi|} = v\, \mathbf{n} \tag{10.11}$$

because the derivative of the function $|p|$ is $p/|p|$ and $\nabla_x \phi / |\nabla_x \phi|$ defines a smooth extension of the unit exterior normal vector $\mathbf{n}$ to the boundary $\partial\Omega_t$. We thus recover from (10.11) that the level-set equation (10.10) corresponds to an evolution with a normal velocity v.

Furthermore, if we restrict the solutions of the Hamiltonian system (10.6) to those initial positions $x(0) \in \partial\Omega_0$, with $p(0) = \nabla_x \phi_0(x(0))$, then the boundary of Ω_t can be obtained simply by

$$\partial\Omega_t = \{ x(t) \text{ solution of (10.6) with } x(0) \in \partial\Omega_0 \},$$

without the necessity to solve the other o.d.e.'s (10.7).

Remark 10.1.3. *Although the function $\mathbb{R}^d \ni p \mapsto |p| \in \mathbb{R}$ is not smooth at 0 (contrary to our assumption), it is not a problem for the level-set method which focuses only on the evolution of the zero level-set (the shape boundary) where $p = \nabla_x \phi$ is proportional to the normal vector to the boundary and thus does not vanish, at least for small times when the initial value $p(0)$ is uniformly bounded away from 0. In other words, the non-smooth function $|p|$ can be regularized near 0 without changing the evolution of the zero level-set.*

10.1.2 Derivation along normal trajectories

We now introduce our new variant of the Hadamard method of shape differentiation, based on the Hamiltonian system (10.6).

Theorem 10.1.4. *Let $k \geq 1$, $E : \mathcal{O}_{k+2} \to \mathbb{R}$ be twice differentiable at $\Omega_0 \in \mathcal{O}_{k+2}$ and $v(t,x)$ be a C^k scalar function from $\mathbb{R}_+ \times \mathbb{R}^d$ into $\mathbb{R}$. Let ϕ_0 be the signed distance function associated to Ω_0. For some time $\tau > 0$, let $\phi \in C^{k+2}\big([0,\tau] \times \mathbb{R}^d\big)$ be a smooth solution (see Remark 10.1.5) of*

$$\begin{cases} \partial_t \phi(t,x) + v(t,x)\,|\nabla_x \phi(t,x)| = 0, \\ \phi(0,x) = \phi_0(x). \end{cases} \tag{10.12}$$

Define $\Omega_t := \big\{x \in \mathbb{R}^d \mid \phi(t,x) < 0\big\}$ and $\epsilon(t) := E\big(\Omega_t\big)$. Then ϵ is twice differentiable at 0 and it holds

$$\epsilon'(0) = l_1\big(v(0,\cdot)\big) \quad \text{and} \quad \epsilon''(0) = l_2\big(v(0,\cdot), v(0,\cdot)\big) + l_1(\hat{Z}_{v,v}(0,\cdot)),$$

with the same linear form l_1 and bilinear form l_2, defined on $\partial\Omega$, as in Theorem 3.2.15, and where $\hat{Z}_{v,v} = \partial_t v + v\partial_{\boldsymbol{n}} v$ with $\partial_{\boldsymbol{n}} v = \boldsymbol{n} \cdot \nabla_x v$.

Remark 10.1.5. *As already said in Remark 10.1.2, (10.12) may not have smooth solutions for all times, even if the Hamiltonian H and the initial data are smooth. Nevertheless, there always exists a small enough time $\tau > 0$ such that there exists a unique classical (smooth) solution of (10.12). To obtain solutions for all times, Crandall and Lions [46] introduced the notion of viscosity solutions for Hamilton-Jacobi equations, which yields the uniqueness and global-in-time existence of such solutions.*

Remark 10.1.6. *Once again, the first-order shape derivative in Theorem 10.1.4 is the same as in the two variants of the Hadamard method, discussed in Section 3.2. However, the second-order derivative is different from the previous ones in Theorems 3.2.15 and 3.2.19. The family of shape evolutions along the normal direction is somehow smaller than the previous two classes of shape evolutions since it completely eliminates the possibility of tangential displacements. Therefore, the second-order derivative is simpler in the present new setting. The formula of the second-order derivative in Theorem 10.1.4 coincides with that of Theorem 3.2.19 if one can find a vector field $V(t,x)$ such that $V(t,x) = v(t,x)\boldsymbol{n}(t,x)$, where $\boldsymbol{n}(t,x)$ is the unit normal vector to the evolving shape Ω_t. The existence of such a vector field was a priori not obvious but it is actually provided by the method of bicharacteristics in Lemma 10.1.1. Indeed, after solving (10.6), it is enough to take $V(t,x) = \nabla_p H(t,x,p(t))$, where $H(t,x,p) = v(t,x)|p|$. Note that such a velocity field V depends strongly on the initial shape Ω_0 and that it is far from being explicit.*

Proof. We apply the result of Section 10.1.1 with $H(t,x,p) = v(t,x)|p|$. By definition, we have $\epsilon(t) = E\big(\Omega_t\big) = \mathcal{E}\big(x(t,\cdot) - \mathrm{Id}\big)$ where $x(t,x_0)$ is the solution of (10.6) with $x(0) = x_0$. Arguing as in the proof of Theorem 3.2.19 and using Theorem 3.2.15, the shape derivatives are given by the chain rule

$$\begin{aligned} \epsilon'(t) &= \mathcal{E}'\Big(x(t,\cdot) - \mathrm{Id}; \dot{x}(t,\cdot)\Big), \\ \epsilon''(t) &= \mathcal{E}''\Big(x(t,\cdot) - \mathrm{Id}; \dot{x}(t,\cdot), \dot{x}(t,\cdot)\Big) + \mathcal{E}'\Big(x(t,\cdot) - \mathrm{Id}; \ddot{x}(t,\cdot)\Big). \end{aligned}$$

In (10.11) we already computed $\dot{x}(t,x_0) = v(t,x_0)\,\mathbf{n}(t,x_0)$ and thus $\epsilon'(0) = l_1\Big(\dot{x}(0,\cdot)\cdot\mathbf{n}\Big) = l_1\Big(v(0,\cdot)\Big)$. To obtain the second-order derivative, we remark that, by definition the tangential component of $\dot{x}(0,\cdot)$ vanishes, i.e., $\dot{x}(0,\cdot)_\Gamma = 0$ (or $\dot{x}(0,\cdot)$ is a normal vector), which implies $Z_{\dot{x}(0,\cdot),\dot{x}(0,\cdot)} = 0$ (where Z is defined in (3.15)) and thus

$$\mathcal{E}''\Big(x(0,\cdot) - \mathrm{Id}; \dot{x}(0,\cdot), \dot{x}(0,\cdot)\Big) = l_2\Big(\dot{x}(0,\cdot)\cdot\mathbf{n}, \dot{x}(0,\cdot)\cdot\mathbf{n}\Big),$$

so that

$$\epsilon''(0) = l_2\Big(\dot{x}(0,\cdot)\cdot\mathbf{n}, \dot{x}(0,\cdot)\cdot\mathbf{n}\Big) + l_1\Big(\ddot{x}(0,\cdot)\cdot\mathbf{n}\Big). \tag{10.13}$$

It remains to compute $\ddot{x}$, starting from $\dot{x}(t,\cdot) = v(t,\cdot)\,\mathbf{n}(t,\cdot)$. We first compute the time derivative of $\mathbf{n} = p/|p|$:

$$\dot{\mathbf{n}} = \frac{\dot{p}}{|p|} - p\frac{(p\cdot\dot{p})}{|p|^3}.$$

Since $\dot{p} = -\nabla_x H = -|p|\nabla_x v$, $p = \nabla\phi$ and $\mathbf{n} = \dfrac{\nabla_x\phi}{|\nabla_x\phi|}$, we get

$$\dot{\mathbf{n}} = -\nabla_x v + \frac{p}{|p|^3}|p|\,(\nabla_x v\cdot p) = -\nabla_x v + (\nabla_x v\cdot\mathbf{n})\,\mathbf{n}.$$

Here $\mathbf{n}$ is an extension in $\mathbb{R}^d$ of the unit exterior normal to $\partial\Omega_t$. If we restrict ourselves to the boundary $\partial\Omega_t$, this implies that $\dot{\mathbf{n}} = -\nabla_\Gamma v$. By using a suitable extension of $\nabla_\Gamma v$, this result is valid everywhere in $\mathbb{R}^d$. Then, differentiating $\dot{x}(t,\cdot) = v(t,\cdot)\,\mathbf{n}(t,\cdot)$ we deduce

$$\ddot{x} = \Big(\partial_t v + \nabla_x v\cdot\dot{x}\Big)\,\mathbf{n} + v\dot{\mathbf{n}} = \partial_t v\,\mathbf{n} + v\big(\nabla_x v\cdot\mathbf{n}\big)\,\mathbf{n} - v\,\nabla_\Gamma v, \tag{10.14}$$

which implies $\ddot{x}(0,\cdot)\cdot\mathbf{n} = \partial_t v + v\big(\nabla_x v\cdot\mathbf{n}\big)$ since $\nabla_\Gamma v\cdot\mathbf{n} = 0$. This proves the desired result. □

10.1.3 Comparison with the existing derivation frameworks

Now there are three frameworks for shape derivation with the Hadamard method. We briefly sum up the differences between them. In each case, time variations of the shape can be put under the form

$$\Omega_t = X(t, \Omega),$$

where $X : \mathbb{R}_+ \times \mathbb{R}^d \to \mathbb{R}^d$. Thus, the time derivatives in the directions $d_1, d_2 : \mathbb{R}_+ \times \mathbb{R}^d \to \mathbb{R}$ of a shape-dependent function E can be written as

$$\begin{aligned} \frac{d}{dt} E \circ X &= l_1(d_1), \\ \frac{d^2}{dt^2} E \circ X &= l_2(d_1, d_2) + l_1(Z), \end{aligned} \tag{10.15}$$

where $Z : \mathbb{R}^d \to \mathbb{R}$.

Displacement field method

For the displacement field method, a transformation of a shape is given by a diffeomorphism $\theta : \mathbb{R}^d \to \mathbb{R}^d$. The time-dependent transformation map X is given by (see Figure 10.2)

$$X(t, \Omega) = (\mathrm{Id} + t\theta)(\Omega).$$

In that case, for $\theta, \xi : \mathbb{R}^d \to \mathbb{R}^d$ the shape derivatives read like in (10.15) with $d_1 = \theta \cdot \mathbf{n}$, $d_2 = \xi \cdot \mathbf{n}$ and

$$Z = Z_{\theta,\xi} = \big((\xi_\Gamma \cdot \nabla_\Gamma)\, \mathbf{n} \big) \cdot \theta_\Gamma - \nabla_\Gamma (\theta \cdot \mathbf{n}) \cdot \xi_\Gamma - \nabla_\Gamma (\xi \cdot \mathbf{n}) \cdot \theta_\Gamma.$$

The shape derivatives can be recovered by a time-dependent variation (see also Section 3.2.3), but the derivatives are really computed with respect to the velocity fields θ, ξ.

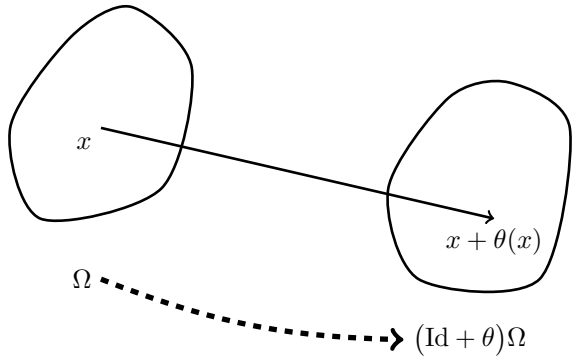


Figure 10.2: Transformation of a shape by $\mathrm{Id} + \theta$

Speed method

For the speed method, with velocity $V : \mathbb{R}_+ \times \mathbb{R}^d \to \mathbb{R}^d$ the transformation of a reference domain is given by X_V which is defined by a set of characteristics in $[0, \tau]$.

$$\begin{cases} \dfrac{\partial X_V}{\partial t}(t, x) = V\big(t, X_V(t, x)\big) \quad \text{for } t > 0\,, \\ X_V(0, x) = x\,. \end{cases} \tag{10.16}$$

The shape derivatives in the directions V, V have the same structure as in (10.15) with $d_1 = V(0) \cdot \mathbf{n}$, $d_2 = V(0) \cdot \mathbf{n}$ and

$$Z = \tilde{Z}_V = \big(\partial_t V + (V \cdot \nabla)\, V\big) \cdot \mathbf{n} + Z_{V,V}.$$

Contrary to the displacement field method, the trajectories are taken into account here (see Figure 10.3).

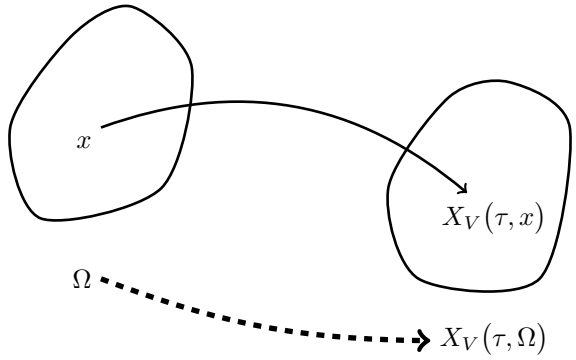


Figure 10.3: Transformation of a shape by X_V

Variation along normal trajectories

In the case of evolution along normal trajectories, the transformation map is implicit and given by a bicharacteristics system. With a velocity $v : \mathbb{R}^d \to \mathbb{R}$, ϕ solution to

$$\partial_t \phi(t,x) + H\big(t, x, \nabla_x \phi(t,x)\big) = 0, \tag{10.17}$$

and x, p solution to

$$\begin{cases} \dfrac{dx}{dt}(t) &= \nabla_p H\big(t, x(t), p(t)\big), \\ \dfrac{dp}{dt}(t) &= -\nabla_x H\big(t, x(t), p(t)\big), \end{cases} \tag{10.18}$$

where $p(0) = \nabla_x \phi(0, x(0))$ and $x(0) \in \mathbb{R}^d$ (see Lemma 10.1.1), the transformation map is given by

$$X_v(t, x(0)) = \nabla_p H\big(t, x(t), p(t)\big).$$

The shape derivatives in the directions v, v still have the same structures with $d_1 = d_2 = v$ and

$$Z = \hat{Z}_{v,v} = \partial_t v + v \partial_{\mathbf{n}} v.$$

In this framework, similarly to the speed method, the trajectories are taken into account for computing derivatives.

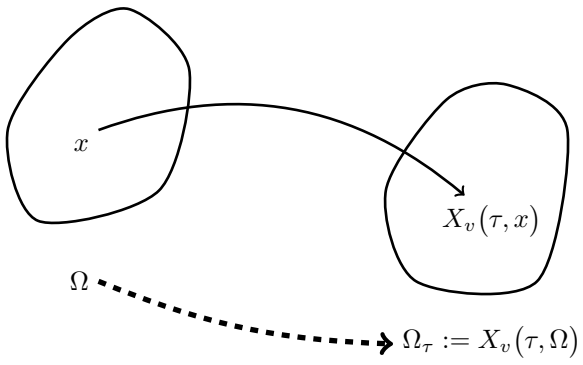


Figure 10.4: Transformation of a shape by X_v

10.2 Extension of the shape derivative

In this section we discuss a first application of the formula for the second-order shape derivative, which is concerned with the problem of extending the shape derivative or the normal velocity in the eikonal equation (10.12) from the shape boundary to the entire space $\mathbb{R}^d$. Indeed, as stated in Theorems 3.2.15, 3.2.19 and 10.1.4, the first-order shape derivative is of the type $\mathcal{E}'(0;\theta) = l_1(\theta \cdot \mathbf{n})$ where l_1 is a linear form on $C^k\big(\partial\Omega\big)$ $(k \geq 1)$. In full generality, l_1 can be

represented by a distribution. Nevertheless, in many practical cases, it is smoother, meaning that it can be represented by a function $j \in L^1(\partial\Omega;\mathbb{R})$. For simplicity, we assume it is the case. Thus, there exists an integrand $j(x)$, depending on the function $\mathcal{E}$ and defined only on the boundary $\partial\Omega$, such that

$$\mathcal{E}'(0;\theta) = l_1(\theta\cdot\mathbf{n}) \equiv \int_{\partial\Omega} j\ (\theta\cdot\mathbf{n})\ .$$

A possible descent direction is to choose $(\theta\cdot\mathbf{n}) = -j$ (other choices are possible, see [15], [32], [53]). In the sequel we call v_γ a descent direction, a function from $\partial\Omega$ into $\mathbb{R}$ such that

$$l_1(v_\gamma) = \int_{\partial\Omega} j\, v_\gamma \le 0\,. \tag{10.19}$$

A priori, v_γ is defined only on $\partial\Omega$, while the level-set method requires to solve (10.12) with a velocity v_γ defined everywhere in $\mathbb{R}^d$. Therefore, one needs to extend v_γ from $\partial\Omega$ to $\mathbb{R}^d$. This issue is discussed in [15], [32], [53] and here we propose new approaches based on the following Ansatz. For small $t \ge 0$, according to Theorem 10.1.4, we write the second-order Taylor expansion

$$E\big(\Omega_t\big) = \epsilon(t) = \epsilon(0) + t\, l_1(v_\gamma) + \frac{1}{2}t^2\Big(l_2\big(v_\gamma, v_\gamma\big) + l_1(\partial_t v_\gamma + v_\gamma\partial_{\mathbf{n}} v_\gamma)\Big) + \mathcal{O}(t^3). \tag{10.20}$$

For simplicity we restrict ourselves to time-independent descent directions, namely $\partial_t v_\gamma = 0$. Taking into account the bilinear form l_2 in (10.20) is at the basis of the Newton algorithm which will be discussed in the next section. For the moment we focus on the other second-order term and our main idea is to build an extension of v_γ such that

$$l_1(v_\gamma\partial_{\mathbf{n}} v_\gamma) \le 0. \tag{10.21}$$

To enforce (10.21), and since v_γ is already bound to satisfy (10.19), we can choose a suitable normal derivative $\partial_{\mathbf{n}} v_\gamma$. We propose here two new extension methods based on this idea and we recall a third classical method which, under some variants, can be found in [15], [32], [53].

Method 10.2.1. *Given a descent direction v_γ on $\partial\Omega$, we compute its extension $v(t,x)$ to $\mathbb{R}^d$ as the solution of the following linear transport equation*

$$\begin{cases} \partial_t v + \mathrm{sign}(\phi)\big(\boldsymbol{n}\cdot\nabla_x v - 1\big) = 0 & \text{in } [0,\tau]\times\mathbb{R}^d,\\ v(t,x) = v_\gamma & \text{on } \partial\Omega,\\ v(0,x) = 0 & \text{on } \mathbb{R}^d\backslash\partial\Omega, \end{cases} \tag{10.22}$$

where $\boldsymbol{n}$ is an extension of the unit outer normal (cf. Remark 3.2.16) and ϕ is a level-set function for Ω. The stationary solution v of (10.22) satisfies

$$\boldsymbol{n}\cdot\nabla_x v = \partial_{\boldsymbol{n}} v_\gamma = 1 \text{ in } \mathbb{R}^d, \quad \text{and} \quad v = v_\gamma \text{ on } \partial\Omega.$$

Thus, we obtain $l_1(v) = l_1(v_\gamma)$ and $l_1(v\partial_{\boldsymbol{n}} v) = l_1(v) = l_1(v_\gamma) \le 0$.

As already said in Remark 3.2.16, there exists a smooth extension of the unit outer normal $\mathbf{n}$ in the neighborhood of $\partial\Omega$ [72]. A global smooth extension is easily deduced if the unit norm constraint is not enforced away from $\partial\Omega$. In numerical practice, the extension of $\mathbf{n}$ is given by $\frac{\nabla_x\phi}{|\nabla_x\phi|}$ where ϕ is the level-set function (such an extension is not smooth on the skeleton of the shape Ω). Equation (10.22) is very much inspired from the classical re-initialization (or redistanciation) technique in the level-set algorithm. In particular, the velocity being opposed on each side of $\partial\Omega$, it implies that the information from the boundary condition v_γ is carried away from $\partial\Omega$. It is not mathematically clear that there exists a stationary solution of (10.22). In numerical practice (with our choice $\mathbf{n} = \frac{\nabla_x\phi}{|\nabla_x\phi|}$), we always find one and, in any case, since the extension is only needed locally close to $\partial\Omega$, we could stop the time evolution of (10.22) as soon as the characteristics associated to the velocity $\mathrm{sign}(\phi)\mathbf{n}$ have reached a certain distance away from $\partial\Omega$. The same comments apply to the following alternative extension method.

Method 10.2.2. *Let v_γ be a descent direction defined on $\partial\Omega$. Assume that an extension of the product jv_γ is already known in $\mathbb{R}^d$ (we keep the same notation jv_γ for the extension). We compute another extension $v(t,x)$ of v_γ as a solution of the linear transport equation*

$$\begin{cases} \partial_t v + \mathrm{sign}(\phi)\big(\boldsymbol{n}\cdot\nabla_x v + jv_\gamma\big) = 0 & \text{in } [0,\tau]\times\mathbb{R}^d,\\ v(t,x) = v_\gamma & \text{on } \partial\Omega,\\ v(0,x) = 0 & \text{on } \mathbb{R}^d\backslash\partial\Omega, \end{cases} \tag{10.23}$$

where ϕ is a level-set function for Ω. The stationary solution v of (10.23) satisfies

$$\boldsymbol{n}\cdot\nabla_x v = -jv_\gamma \text{ in } \mathbb{R}^d, \quad \text{and} \quad v = v_\gamma \text{ on } \partial\Omega.$$

Thus, we obtain $l_1(v) = l_1(v_\gamma)$ and

$$l_1(v\partial_{\boldsymbol{n}} v) = l_1(-jv_\gamma^2) = -\int_{\partial\Omega} j^2 v_\gamma^2 \le 0.$$

Eventually, for the sake of comparison, we recall a standard regularization and extension approach which can be found in [15], [32], [53].

Method 10.2.3. *Let v_γ be a descent direction defined on $\partial\Omega$. We compute an extension $v(x)$ of v_γ as a solution of the following variational problem on $\mathbb{R}^d$*

$$\eta^2 \int_{\mathbb{R}^d} \nabla v \cdot \nabla w + \int_{\partial\Omega} vw = \int_{\partial\Omega} v_\gamma w, \quad \forall w \in H^1\big(\mathbb{R}^d\big), \tag{10.24}$$

where $\eta > 0$ is a small parameter (typically of the order of a few mesh cell diameters in numerical practice). In practice too, the full space $\mathbb{R}^d$ is replaced by a bounded computational domain. On the one hand, the solution v is not strictly speaking an extension of v_γ since $v_\gamma \neq v$ on $\partial\Omega$. On the other hand, the direction v is more regular than v_γ on $\partial\Omega$. In the case of the obvious descent direction $v_\gamma = -j$, one can check that the solution v of (10.24) remains a descent direction since

$$l_1(v) = \int_{\partial\Omega} jv = -\int_{\partial\Omega} v_\gamma v = -\Big(\eta^2 \int_{\mathbb{R}^d} |\nabla v|^2 + \int_{\partial\Omega} v^2\Big) \leq 0.$$

Remark 10.2.4. *Method 10.2.3 can be used to compute the extension of jv_γ which is assumed to be already known at the beginning of Method 10.2.2. Even though such an extension is not equal to jv_γ on $\partial\Omega$, one can show by the same argument that Method 10.2.2 produces again an extension satisfying $l_1(v\partial_{\boldsymbol{n}} v) \leq 0$.*

10.3 Optimization method

The Newton method as well as the gradient method have already been introduced in Section 9.1 for general shape optimization problems. Here we consider the case of shape optimization with the level-set method, and give some more details on these two methods.

We consider the unconstrained minimization of an objective function $E(\Omega)$ over all possible shapes $\Omega \subset \mathbb{R}^d$. Starting from an initial shape Ω_0, we build an iterative sequence $(\Omega_p)_{p\in\mathbb{N}}$ in order to minimize the objective function $E(\Omega)$. For a given iteration number p, we denote by ϕ_p the level-set function associated to $\partial\Omega_p$. To build the new shape Ω_{p+1} from Ω_p, we solve the Hamilton-Jacobi equation

$$\begin{cases} \partial_t \phi_{p+1}(t,x) + v^p(x)\,|\nabla_x \phi_{p+1}(t,x)| = 0, \\ \phi_{p+1}(0,x) = \phi_p(x). \end{cases} \tag{10.25}$$

For any pseudo-time or descent parameter $t \geq 0$ we define an associated shape

$$\Omega(t, v^p) = \big\{ x \in \mathbb{R}^d \mid \phi_{p+1}(t,x) < 0 \big\}.$$

The new shape is defined as $\Omega_{p+1} = \Omega(t_p, v^p)$ where $t_p > 0$ is chosen in order to decrease the reduced objective function $t \mapsto E\left(\Omega(t, v^p)\right)$.

In (10.25) the normal velocity v^p is a descent direction (properly extended to $\mathbb{R}^d$ by one of the three methods presented in Section 10.2), evaluated at Ω_p with a gradient or the Newton method that we now describe. The linear and bilinear forms l_1 and l_2, defined in Theorem 3.2.15, depend on the shape Ω_p (cf. the explicit examples in Chapter 7 or Chapter 12). In this section, we denote them by $l_{p,1}$ and $l_{p,2}$ for the corresponding shape Ω_p.

10.3.1 Newton's algorithm

Thanks to the scaling invariance $\Omega(\frac{t}{\alpha}, \alpha v^p) = \Omega(t, v^p)$ the parameter t_p is chosen equal to 1. Then, for small enough v, the following quadratic approximation holds true

$$E\big(\Omega(1,v)\big) \simeq E(\Omega_p) + l_{p,1}(v) + \frac{1}{2}\Big(l_{p,2}(v,v) + l_{p,1}(v\partial_{\mathbf{n}} v)\Big),$$

where only the traces of v and $\partial_{\mathbf{n}} v$ on $\partial\Omega_p$ play a role. The Newton method intends to minimize this quadratic approximation in order to obtain the descent direction v^p. We now observe that, on $\partial\Omega_p$, v and $\partial_{\mathbf{n}} v$ may be considered as independent variables and that $\partial_{\mathbf{n}} v$ does not enter the advection equation (10.25). Therefore, $\partial_{\mathbf{n}} v$ can be used only in an extension process of the shape derivative (as explained in the previous section) and it makes sense to minimize separately in v and $\partial_{\mathbf{n}} v$ (although other choices are possible). First, we solve the following problem, referred to as the Newton problem

$$\min_{v \in C^{1,\infty}(\partial\Omega_p;\mathbb{R})} \quad l_{p,1}(v) + \frac{1}{2} l_{p,2}(v,v). \tag{10.26}$$

Second, knowing the optimal v^p in (10.26), we should minimize $l_{p,1}(v^p \partial_{\mathbf{n}} v)$ with respect to $\partial_{\mathbf{n}} v$. This linear problem does not have a solution, and this is precisely the content of the previous section. Note that, if we impose a trust-region constraint (say $\|\partial_{\mathbf{n}} v\|_{L^2(\partial\Omega_p;\mathbb{R})} \leq m$), then this linear problem has a solution $\partial_{\mathbf{n}} v$ which is proportional to jv and we are back to Method 10.2.2 for extending the descent direction away from $\partial\Omega$. Another possibility could be to first

extend v by Method 10.2.3 and then consider $\partial_{\mathbf{n}} v$ to be the image of the extended v by the Dirichlet-to-Neumann map. It results in a quadratic approximation featuring only v. It is not clear how it can be implemented in practice but, at least, it shows that there are different ways of extracting useful information from the second-order approximation of the objective function.

Like it is said in the general case (see Section 9.1), the Newton problem (10.26) has no reason to be bounded, and to have a finite minimum. Indeed, the Hessian operator $l_{p,2}$ usually depends on the normal vector and the curvature and therefore may not be positive definite. In such a case, the minimal value of (10.26) could be $-\infty$. As a result, the Newton problem (10.26) is replaced by a trust-region variant (see for example [92, Section 6.4], [131]) that writes :

$$\min_{\substack{v \in C^{1,\infty}\left(\partial\Omega_p;\mathbb{R}\right),\\ \|v\|_{L^\infty(\partial\Omega_p;\mathbb{R})} \leq v_M,}} \quad l_{p,1}(v) + \frac{1}{2} l_{p,2}(v,v)\,, \tag{10.27}$$

for some finite positive bound v_M. When the solution v^p to (10.27) satisfies a strict bound $\|v^p\|_{L^\infty(\partial\Omega;\mathbb{R})} < v_M$, then it is also a solution of (10.26) and it satisfies Newton's equation

$$l_{p,2}(v^p, w) = -l_{p,1}(w) \quad \forall w \in C^{1,\infty}\left(\partial\Omega_p;\mathbb{R}\right).$$

With a good descent direction v^p, we also have to look for a good descent step t_p. We use for that the usual backtracking procedure of Algorithm 9.1.1.

Now we recall the Newton algorithm in this context.

Algorithm 10.3.1. *Newton's algorithm*

1. ***Initialization*** *Set $p = 0$. Choose an initial domain Ω_0, two convergence thresholds $\varepsilon > 0$, $\eta > 0$, and two coefficients c_0, c_1 in $]0,1[$.*
2. ***Newton's direction*** *Compute the solution v^p of (10.27).*
3. ***Extension*** *Compute an extension of v^p with one of the three methods detailed in Section 10.2.*
4. ***Time Step*** *Compute a time step t_p with Algorithm 9.1.1.*
5. ***Update*** *Set $\Omega_{p+1} = \Omega(t_p, v^p)$ and $p = p+1$. If $\displaystyle\int_{\partial\Omega_p} |v^p_\gamma|^2 \leq \varepsilon$ or $t_p \leq \eta$ stop. Else return to step 2.*

Remark 10.3.2. *Like explained in Remark 9.1.3, when the bilinear map $l_{p,2}$ is not positive definite, the direction v^p solution to (10.27) may not be a descent direction. In such a case, we replace v^p by its opposite.*

10.3.2 Gradient algorithm

In the context of shape optimization with the level-set method, the gradient-like descent direction is computed as the solution to

$$\min_{\substack{v \in C^{1,\infty}\left(\partial\Omega_p;\mathbb{R}\right),\\ \|v\|_{L^\infty(\partial\Omega_p;\mathbb{R})} \leq v_M}} \quad l_{p,1}(v) + \frac{1}{2}\int_{\partial\Omega_p} v^2. \tag{10.28}$$

The $L^\infty\left(\partial\Omega_p;\mathbb{R}\right)$ bound is added for the same reason explained in Section 9.1, that is in order to ease the comparison with the Newton method.

In this context we write the gradient algorithm.

Algorithm 10.3.3. *Gradient algorithm*

1. ***Initialization*** *Set $p = 0$. Choose an initial domain Ω_0, two convergence thresholds $\varepsilon > 0$, $\eta > 0$, and two coefficients c_0, c_1 in $]0,1[$.*
2. ***Gradient direction*** *Compute the solution v^p of (10.28).*
3. ***Extension*** *Compute an extension of v^p with one of the three methods detailed in Section 10.2.*
4. ***Time Step*** *Compute a time step t_p with Algorithm 9.1.1.*
5. ***Update*** *Set $\Omega_{p+1} = \Omega(t_p, v^p)$ and $p = p+1$. If $\displaystyle\int_{\partial\Omega_p} |v^p|^2 \leq \varepsilon$ or $t_p \leq \eta$ stop. Else return to step 2.*

Chapter 11

Computing boundary integrals

Contents

Both gradient and Newton's algorithms are based on shape derivatives which are written in terms of boundary integrals. It is therefore crucial to compute these integrals accurately to obtain efficient optimization methods. We first recall the standard procedure in the level-set method for structural optimization which is based on the so-called approximate Dirac mass for the boundary. Then we introduce a better approximation using a piecewise linear reconstruction of the boundary, in the spirit of earlier results of Min and Gibou [84, 85].

11.1 Dirac mass function

11.1.1 Principle

Given a level-set function ϕ, we can define a sign and a characteristic function.

Definition 11.1.1. *Let $\phi : \mathbb{R}^d \to \mathbb{R}$, and Ω a shape defined by the zero level-set of ϕ. We define the sign function s by*

$$\begin{cases} s(x) = 1 & \text{if } \phi(x) > 0, \\ s(x) = 0 & \text{if } \phi(x) = 0, \\ s(x) = -1 & \text{otherwise.} \end{cases} \tag{11.1}$$

This gives access to the Heaviside function defined by

$$\chi = \frac{1}{2}(1 - s). \tag{11.2}$$

Now we can also introduce a Dirac mass concentrated on the boundary, seen as the weak derivative (in the sense of distributions) of the sign function.

Definition 11.1.2. *Let $\Omega \in \mathcal{O}_2$ be a subset of a bounded open set $\mathcal{D} \subset \mathbb{R}^d$. Let also C_c be the set of functions in $C^\infty(\mathcal{D}; \mathbb{R})$ with compact support in $\mathcal{D}$. We denote by $\delta_{\partial\Omega}$ the single-layer distribution defined by*

$$\forall \psi \in C_c, \quad \langle \delta_{\partial\Omega}, \psi \rangle = \int_{\partial\Omega} \psi \, .$$

The distribution $\delta_{\partial\Omega}$ is called the Dirac mass concentrated on the boundary of Ω.

Lemma 11.1.3. *Let $\Omega \in \mathcal{O}_2$. The gradient of the sign function, seen as a distribution of ${C_c}^d$ satisfies*

$$\forall \psi \in {C_c}^d, \quad \langle \nabla s, \psi \rangle = 2 \langle \delta_{\partial\Omega}, (\psi \cdot \boldsymbol{n}) \rangle \, .$$

Proof. With $\psi \in {C_c}^d$ one has

$$\begin{aligned}\int_{\mathcal{D}} s\,\mathrm{div}(\psi) &= -\int_{\Omega} \mathrm{div}(\psi) + \int_{\mathcal{D}\setminus\Omega} \mathrm{div}(\psi),\\ &= -2\int_{\partial\Omega} \psi\cdot\mathbf{n} + \int_{\partial\mathcal{D}} \psi\cdot\mathbf{n}.\end{aligned}$$

The support of the function ψ being compact in $\mathcal{D}$, this leads to

$$\int_{\mathcal{D}} s\,\mathrm{div}(\psi) = -2\int_{\partial\Omega} \psi\cdot\mathbf{n},$$

and therefore

$$\langle \nabla s, \psi\rangle = 2\left\langle \delta_{\partial\Omega}, (\psi\cdot\mathbf{n})\right\rangle.$$

□

This has a numerical interest since, with a regularization of the sign function, an approximation of its gradient allows one to compute a regular approximation of the Dirac mass $\delta_{\partial\Omega}$.

Remark 11.1.4. *Let $s_n \in C^\infty(\mathcal{D})$ be a sequence of smooth functions converging in the sense of the distributions to s. In other words*

$$\forall \psi \in C_c, \quad \lim_{n\to\infty}\int_{\mathcal{D}} s_n\psi = \int_{\mathcal{D}} s\psi.$$

Now let $\tilde{\boldsymbol{n}}$ be an extension of the outer normal to Ω and define δ_n by

$$\delta_n = \frac{1}{2}\nabla s_n\cdot\tilde{\boldsymbol{n}}.$$

For all $\psi \in C_c$, we have

$$\int_{\mathcal{D}} s_n \mathrm{div}(\psi\tilde{\boldsymbol{n}}) = -\int_{\mathcal{D}} \nabla s_n\cdot(\psi\tilde{\boldsymbol{n}}) = -2\int_{\mathcal{D}} \delta_n\psi.$$

Since

$$\lim_{n\to\infty}\int_{\mathcal{D}} s_n\mathrm{div}(\psi\tilde{\boldsymbol{n}}) = \int_{\mathcal{D}} s\,\mathrm{div}(\psi\tilde{\boldsymbol{n}}) = -\left\langle\nabla s, \psi\tilde{\boldsymbol{n}}\right\rangle,$$

and

$$\left\langle\nabla s, \psi\tilde{\boldsymbol{n}}\right\rangle = 2\left\langle\delta_{\partial\Omega}, \psi\right\rangle,$$

we get

$$\lim_{n\to\infty}\int_{\mathcal{D}} \delta_n\psi = \left\langle\delta_{\partial\Omega}, \psi\right\rangle.$$

Therefore, the sequence δ_n converges to $\delta_{\partial\Omega}$ in the sense of distributions.

We also recall that for any Lipchitz domain Ω, the signed distance function d_Ω to Ω is defined by

$$\forall x\in\mathbb{R}^d,\ d_\Omega(x) = \begin{cases} -d(x,\partial\Omega) & \text{if} \quad x\in\Omega,\\ 0 & \text{if} \quad x\in\partial\Omega,\\ d(x,\partial\Omega) & \text{if} \quad x\in\mathbb{R}^d\setminus(\Omega\cup\partial\Omega).\end{cases} \tag{11.3}$$

Introducing a small regularization parameter $\varepsilon > 0$, we can compute an approximation of the sign function by

$$s_{1,\varepsilon} = \frac{d_\Omega}{\sqrt{d_\Omega^2+\varepsilon^2}}.$$

Since the gradient of $s_{1,\varepsilon}$ is directed along ∇d_Ω (the direction is independent of ε) which is an extension of the outer normal, we can consider

$$\delta_{1,\varepsilon} = \frac{1}{2}\nabla s_{1,\varepsilon}\cdot\tilde{\mathbf{n}} = \frac{1}{2}|\nabla s_{1,\varepsilon}|, \tag{11.4}$$

which has the same convergence property as the approximation δ_n introduced in Remark 11.1.4. Boundary integrals can then be approximated as

$$\int_{\partial\Omega} f \simeq \int_{\mathcal{D}} \delta_{1,\varepsilon} f,$$

and the previous remark ensures the convergence of the approximation. Finally, using a finite difference scheme and standard quadrature formulas makes this last approximation fully discrete (see [15] for details).

In the literature [100, 107] we could also find another approximation for the signed distance function

$$s_{2,\varepsilon} = \frac{\phi}{\sqrt{\phi^2 + h^2\left|\nabla\phi\right|^2 + \varepsilon^2}}. \tag{11.5}$$

Instead of (11.4), we could also have written (see [15, 43]) $\nabla s = 2\delta\mathbf{n}$. Therefore we can take

$$\delta_{2,\varepsilon} = \frac{1}{2}|\nabla s_{2,\varepsilon}|, \tag{11.6}$$

that leads to another approximation of the Dirac mass.

Now, with these two approximations of the Dirac mass $\delta_{\partial\Omega}$, we consider the consistency of the methods for computing boundary integrals.

11.1.2 Lack of consistency

Computing boundary integrals with the Dirac mass function is very practical from a numerical point of view, since with finite difference it is possible to compute an approximation of the weak derivative of the sign function. However, it seems that this approach is not numerically consistent. We recall that consistency is the difference between the continuous model and the numerical model. A priori only the *discretize-then-optimize* approach is known to be consistent. Therefore, it would not be very surprising to derive from other approaches models that are not perfectly consistent. But it appears that the numerical error introduced by the use of a Dirac function concentrated on the boundary to compute shape derivatives can be very large.

Let us consider the following example.

Example 11.1.5. *Let $(a_0, b_0) \in \mathbb{R}^2$ and f be given by*

$$\begin{aligned} f(x,y) &= (x^2+y^2)^5 - 2a_0^5\mathrm{Re}\big((x+iy)^5\big) + a_0^{10} - b_0^{10} \\ &= (x^2+y^2)^5 - 2a_0^5(x^5 - 10x^3y^2 + 5xy^4) + a_0^{10} - b_0^{10}. \end{aligned} \tag{11.7}$$

We consider the minimization problem

$$\inf_{\Omega\in\mathcal{O}_2} E(\Omega), \tag{11.8}$$

where

$$E(\Omega) = \int_\Omega f(x)\,dx,$$

is the objective function to be minimized. We start from an initial shape Ω_0 given by the zero value of

$$\phi(x,y) = ((x-a_1)^2+y^2)((x+a_1)^2+y^2) - b_1^4,$$

and take

$$a_0 = 0.95,\ b_0 = 0.953,\ a_1 = 0.5,\ b_1 = 0.51.$$

This simple model will also be further explored for consistency considerations in Section 11.3.2, and for numerical optimization in Section 12.1. Here we focus only on the consistency (like in Section 9.4.1) of the shape derivative computed with a Dirac approximation at the initial shape. The initial shape is the one displayed in Figure 11.1. For $\theta \in C^{1,\infty}(\mathbb{R}^2;\mathbb{R}^2)$, the first-order shape derivative of the function E is given by

$$E'(\Omega;\theta) = \int_{\partial\Omega} f(x)\,(\theta\cdot\mathbf{n})\,(x)\,dx.$$

Now for a velocity field v defined on $\partial\Omega$ by v_γ and extended with Method 10.2.3, we consider for $t \in [0,\tau]$

$$\Omega_t = \Omega(t,v) = \left\{x \in \mathbb{R}^2 \mid \phi(t,x) < 0\right\},$$

where ϕ is solution to the following Hamilton-Jacobi equation

$$\begin{cases} \partial_t\phi(t,x) + v(x)\,|\nabla_x\phi(t,x)| = 0, \\ \phi(0,x) = \phi_0(x), \end{cases} \tag{11.9}$$

and ϕ_0 is the signed distance function to Ω_0. For different values of the time parameter τ, and v_γ given by the solution to the gradient equation (10.28), we plot in Figure 11.2 the two curves

$$\begin{aligned} &t \mapsto E(\Omega_t), \\ &t \mapsto E(\Omega_0) + tE'(\Omega_0;v), \end{aligned}$$

where the first-order derivative is approximated by

$$E'(\Omega_0;v) \simeq \int_{\mathcal{D}} \delta_{i,\varepsilon} f v.$$

We take $\mathcal{D} =]-1.15, 1.15[$ that is endowed with a square mesh of size 120×120. The size of a cell is then $h = \frac{2.3}{120}$. For the Dirac approximation, we take $\varepsilon = \frac{h}{20}$. We can observe that the linear approximation computed with to the Dirac mass approximation is far from being tangent to the curve $t \mapsto E(\Omega_t)$.

Figure 11.1: Initial shape.

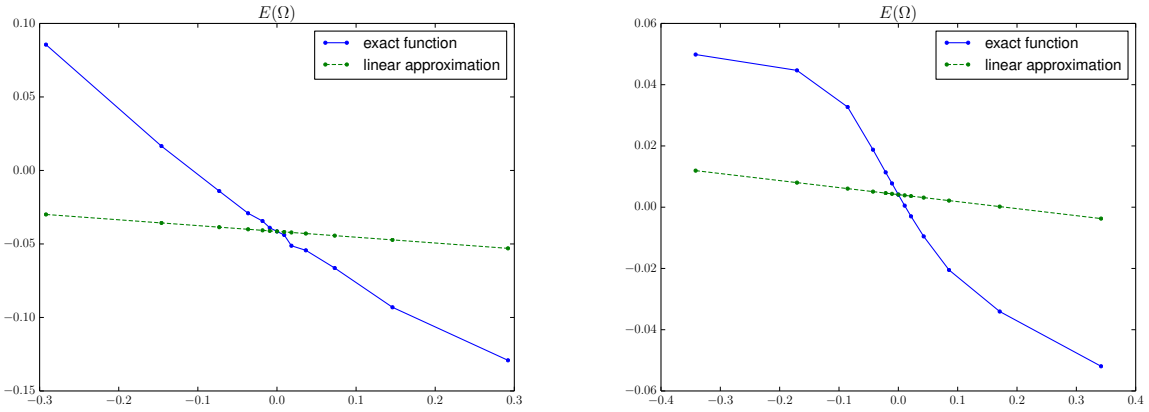


Figure 11.2: Consistency of the derivative with $\delta_{1,\varepsilon}$ (left) and $\delta_{2,\varepsilon}$ (right) with $h = \frac{2.3}{120}$ and $\varepsilon = \frac{h}{20}$.

11.2 Approximation with a linear interpolation method

The previous Dirac mass approach is very simple to implement in practice but may not be consistent enough in some cases. We therefore introduce another approach to evaluate boundary integrals. The main idea is to reconstruct a piecewise linear approximation of the boundary and compute exactly the boundary integral for a linear interpolation of the integrand (or for products of linear interpolations). There have been many works on improving the computation of surface integrals in the context of the level-set method [19], [59], [84], [120]. Here we follow an idea of Min and Gibou [84, 85] and we compute boundary integrals like for building mass and stiffness matrices in a finite element method.

11.2.1 Principle

In numerical practice we work in 2-d with a regular square mesh $\mathcal{Q}_h$ of the computational domain in which all admissible shapes are included (extension to the 3-d case is conceptually simple although more cumbersome than in 2-d). Like in Chapter 8, each rectangular cell (whose nodes are denoted i_i for $i \in [\![1, 4]\!]$) is divided in four triangles separated by its diagonals (see Figure 11.4).

Let $\mathcal{T}_h$ be the resulting triangular mesh. The level-set function ϕ is discretized on the square mesh. To obtain its $\mathbb{P}_1$ finite element interpolation on the triangular mesh $\mathcal{T}_h$ we simply assign to the additional node at the center of each square (denoted further by C in the figures) the average value of the level-set at the four corner nodes. This consists in considering ϕ as being in $\mathbb{Q}_1(\mathcal{Q}_h; \mathbb{R})$, and in taking $\mathcal{I}^1_{\mathcal{T}_h}(\phi)$ its linear interpolation on $\mathcal{T}_h$, where $\mathcal{I}^1_{\mathcal{T}_h}$ is the linear interpolation operator for the Lagrange finite element (see Definition 5.1.2). By definition, $\mathcal{I}^1_{\mathcal{T}_h}(\phi)$ coincides with ϕ on each node of $\mathcal{T}_h$. Once we have a $\mathbb{P}_1$ interpolation $\phi_h = \mathcal{I}^1_{\mathcal{T}_h}(\phi)$ of ϕ on the triangular mesh, it is standard to obtain a discretization Ω_h of Ω which is defined by $\Omega_h = \{x \in \mathbb{R}^d \,|\, \phi_h(x) < 0\}$. By construction, the boundary of Ω_h is piecewise linear (more precisely it is the continuous union of line segments in each triangle). To compute an

approximation of an integral on $\partial\Omega$, we compute the exact integral on $\partial\Omega_h$ of an interpolation of the integrand.

Remark 11.2.1. *The use of a virtual node C is suggested by the non-uniqueness of domains whose level-set functions take given values on the nodes of a square (see Figure 11.3). This does not occur on a triangle.*

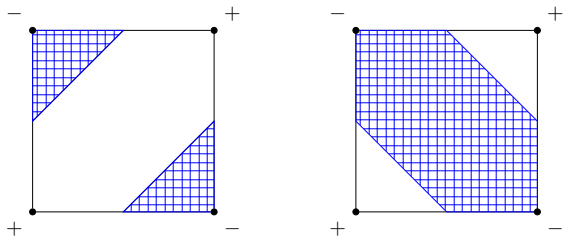


Figure 11.3: Non-uniqueness of a shape (grid domain) for a given level-set function defined by its values at the four corners of a square.

Let us start by describing the linear approximation of the boundary. An element that is not crossed by the boundary does not contribute to the boundary integral. In the sequel, we only focus on elements that are intersected by the boundary. They are characterized by the fact that the sign of ϕ is not the same on the four nodes of the element. With the additional node, a cell is divided in four similar triangles $\mathcal{T}_i$ ($i \in [\![1,4]\!]$). Figure 11.4 shows a square cell, and the subdividing triangles. It also shows all the possible ways for the domain Ω to intersect a triangle.

In each triangle $\tau = ABC$ we can compute a linear interpolation of ϕ and then the straight line $\mathscr{D}_\tau := \tau \cap \partial\Omega_h$ as the curve $\phi_h = 0$. We denote by P_1,P_2, or P_3 the intersection points of $\mathscr{D}_\tau$ and the boundary of the triangle, depending on whether they are respectively on $[A,C]$, $[B,C]$ or $[A,B]$. The edges intersected by $\mathscr{D}_\tau$ - $[A,C]$, $[B,C]$ or $[A,B]$ - are entirely determined by the sign of ϕ_h at the nodes A, B and C. Now we choose a Cartesian coordinate system of the triangle, so that $\mathscr{D}_\tau$ can be parametrized by the x-coordinate (see Figure 11.5). Depending on the choice of coordinate system, we are looking for the plane $\mathscr{P}_\tau \subset \mathbb{R}^3$ containing the points

$$(a)\begin{cases}\big(0,0,\phi(C)\big)\\ \big(\frac{\sqrt{2}}{2},0,\phi(A)\big)\\ \big(0,\frac{\sqrt{2}}{2},\phi(B)\big)\end{cases}\quad(b)\begin{cases}\big(0,0,\phi(B)\big)\\ \big(\frac{\sqrt{2}}{2},0,\phi(C)\big)\\ \big(\frac{\sqrt{2}}{2},\frac{\sqrt{2}}{2},\phi(A)\big)\end{cases}\quad(c)\begin{cases}\big(0,0,\phi(A)\big)\\ \big(1,0,\phi(B)\big)\\ \big(\frac{1}{2},\frac{1}{2},\phi(C)\big)\end{cases}\tag{11.10}$$

This is exactly the linear interpolation of ϕ given by $\mathcal{I}^1_{\mathcal{T}_h}$. We recall that with the operator $\mathcal{I}^1_{\mathcal{T}_h}$, the values of ϕ and ϕ_h at the nodes A, B or C are the same. Let $ax+by+cz+d=0$ be an equation of $\mathscr{P}_\tau$. Taking $c=1$ leads to

$$(a)\begin{cases}a &=& \sqrt{2}\big(\phi(C)-\phi(A)\big)\\ b &=& \sqrt{2}\big(\phi(C)-\phi(B)\big)\\ c &=& 1\\ d &=& -\phi(C),\end{cases}\quad(b)\begin{cases}a &=& \sqrt{2}\big(\phi(B)-\phi(C)\big)\\ b &=& \sqrt{2}\big(\phi(C)-\phi(A)\big)\\ c &=& 1\\ d &=& -\phi(B),\end{cases}\quad(c)\begin{cases}a &=& \phi(A)-\phi(B)\\ b &=& \phi(A)+\phi(B)-2\phi(C)\\ c &=& 1\\ d &=& -\phi(A).\end{cases}\tag{11.11}$$

Now we have the equation of the straight-line $\mathscr{D}_\tau$, intersection of $\mathscr{P}_\tau$ with the plane of equation $z=0$. Since ϕ is almost a signed distance function, it is reasonable to assume that ϕ_h does not vanish identically in the triangle. The equation of $\mathscr{D}_\tau$ is then $ax+by+d=0$. The orientation of the triangle is made so that $b \neq 0$ (see Figure 11.5). Now we have explicitly, for each possible configuration, the equation of the linear approximation of the boundary depending only on the values of ϕ_h on the nodes of the triangle. Since the value of ϕ at node C depends linearly on its values at nodes i_i, the equation of $\mathscr{D}_\tau$ can be written only in terms of $\phi(i_i)$.

We also want to integrate a field $v \in \mathbb{Q}_1(\mathcal{Q}_h;\mathbb{R})$ on the boundary. We will explicitly integrate its linear interpolation on $\mathcal{T}_h$: $\mathcal{I}^1_{\mathcal{T}_h}(v)$. This interpolation coincides with v on every node of $\mathcal{T}_h$. Similarly to the approximation of ϕ, we find the equation of $\mathcal{I}^1_{\mathcal{T}_h}(v)$:

$$\mathcal{I}^1_{\mathcal{T}_h}(v)(x,y) = -d_v - b_v y - a_v x,$$

where, depending on the choice of the orientation

$$(a)\begin{cases}a_v &=& \sqrt{2}\big(v(C)-v(A)\big)\\ b_v &=& \sqrt{2}\big(v(C)-v(B)\big)\\ c_v &=& 1\\ d_v &=& -v(C)\end{cases}\quad(b)\begin{cases}a_v &=& \sqrt{2}\big(v(B)-v(C)\big)\\ b_v &=& \sqrt{2}\big(v(C)-v(A)\big)\\ c_v &=& 1\\ d_v &=& -v(B)\end{cases}\quad(c)\begin{cases}a_v &=& v(A)-v(B)\\ b_v &=& v(A)+v(B)-2v(C)\\ c_v &=& 1\\ d_v &=& -v(A)\end{cases}\tag{11.12}$$

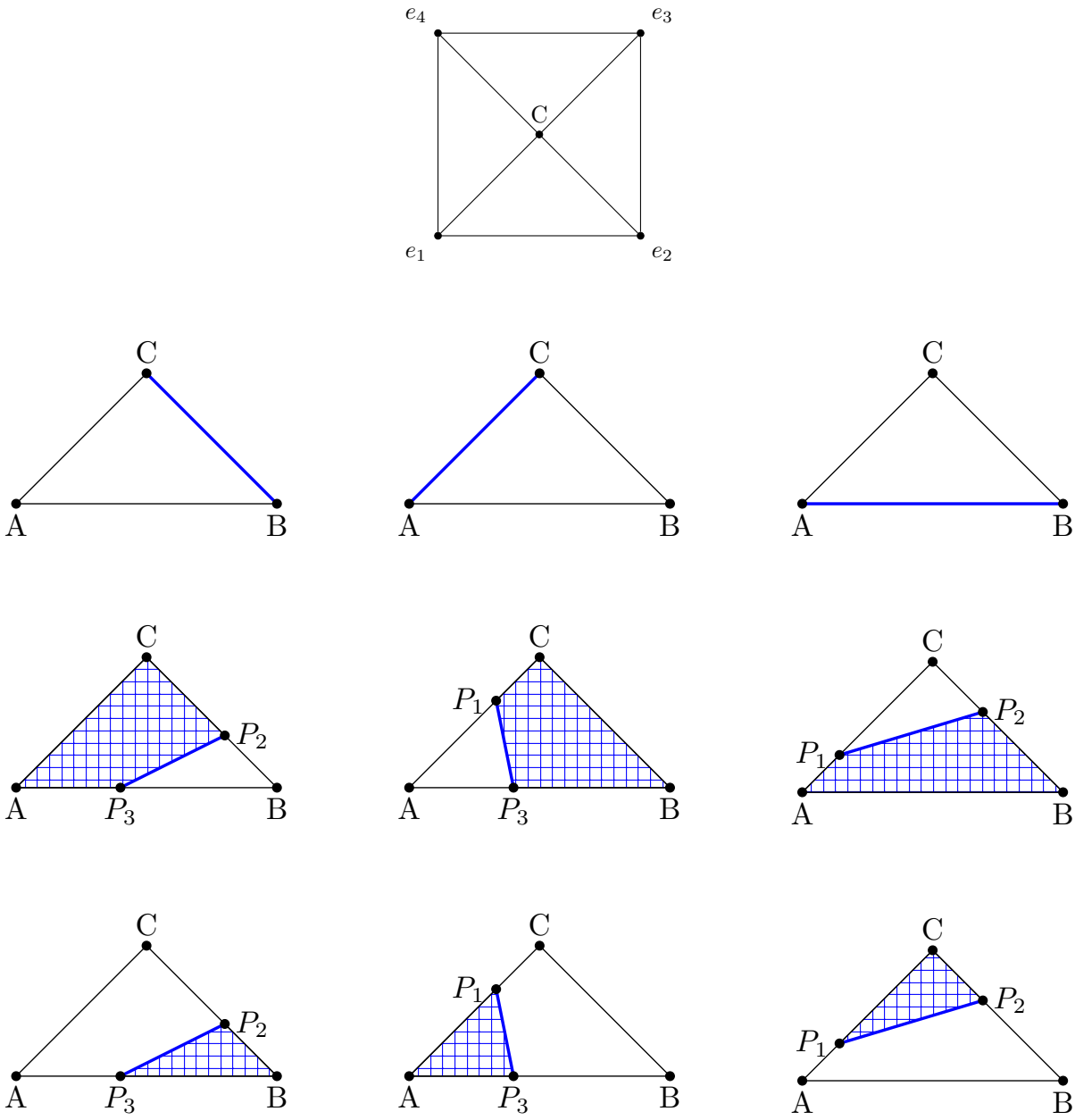


Figure 11.4: All possible configurations of a triangle cut by the level-set function.

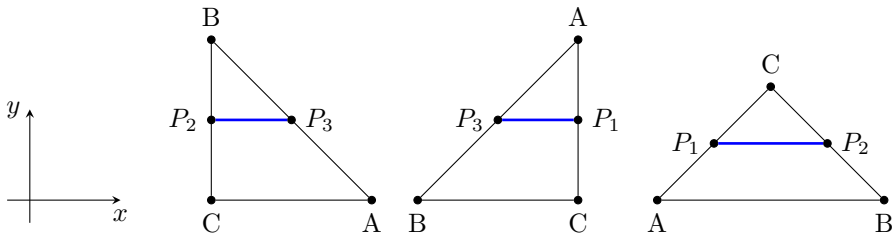


Figure 11.5: Choice of coordinate system for the triangle ABC in the cases (a), (b) and (c) in (11.10) .

Knowing the equation of $\mathscr{D}_\tau$ we can parametrize it with a function r. We denote (x_i, y_i),(x_j, y_j) the coordinates of P_i,P_j, $x_{\min} = \min(x_i, x_j)$ and $x_{\max} = \max(x_i, x_j)$. Since the coordinate system of the triangle is chosen such that $b \neq 0$, we can write $\mathscr{D}_\tau = r\big([x_{\min}, x_{\max}]\big)$, where $r(t) = \big(t, -\frac{d}{b} - \frac{a}{b}t\big)$ with a, b and d given by (11.11). Then

$$\int_{\mathscr{D}_\tau} \mathcal{I}^1_{\mathcal{T}_h}(v)\, ds = \int_{x_{\min}}^{x_{\max}} \big(\mathcal{I}^1_{\mathcal{T}_h}(v) \circ r(t)\big)|r'(t)|_2\, dt. \tag{11.13}$$

Now the contribution of the triangle to the boundary integral is expressed only in terms of the values of v and of ϕ at the nodes A, B, C. Like $\phi(C)$, the value of $v(C)$ can be written only in terms of $v(i_i)$ for $i \in [\![1, 4]\!]$, so that the elementary contribution (11.13) can be written only in terms $\phi(e_i)$ and $v(e_i)$ for $i \in [\![1, 4]\!]$. Summing all the contributions on each cell, we get the total boundary integral after rescaling by the size of the mesh h (in our the parametrization of the triangle, the lengths of the edges are 1, $\frac{\sqrt{2}}{2}$, $\frac{\sqrt{2}}{2}$, instead of h, $\frac{h\sqrt{2}}{2}$, $\frac{h\sqrt{2}}{2}$).

In order to get more accuracy, for $v_1, v_2, v_3 \in \mathbb{Q}_1(\mathcal{Q}_h; \mathbb{R})$ and $v_{i,h} = \mathcal{I}^1_{\mathcal{T}_h}(v_i)$, we also compute explicitly the boundary integrals

$$\int_{\mathscr{D}_\tau} v_{1,h}\, v_{2,h}\, v_{3,h}\, ds = \int_{x_{\min}}^{x_{\max}} \big(v_{1,h} \circ r(t)\big)\big(v_{2,h} \circ r(t)\big)\big(v_{3,h} \circ r(t)\big)|r'(t)|_2\, dt, \tag{11.14}$$

in terms of $\phi(i_i)$, and $v_j(i_i)$ for $i \in [\![1, 4]\!]$ and $j \in [\![1, 3]\!]$. With the same principle, we can also compute bulk integrals. We denote by Ω_h the linear approximation of the shape; note that $\forall \tau \in \mathcal{T}_h$, $\partial\Omega_h \cap \tau = \mathscr{D}_\tau$. Then we can also compute explicitly

$$\int_{\tau \cap \Omega_h} v_{1,h}\, v_{2,h}\, v_{3,h}\, dx. \tag{11.15}$$

11.2.2 Criteria evaluations

We can then compute for each triangle τ of the mesh $\mathcal{T}_h$, the third-order tensors defined for $v_1, v_2, v_3 \in \mathbb{Q}_1(\mathcal{Q}_h; \mathbb{R})$ by

$$\begin{aligned} (v_j(i_i))_{i \in [\![1,4]\!],\, j \in [\![1,3]\!]} &\longmapsto \int_{\mathscr{D}_\tau} \mathcal{I}^1_{\mathcal{T}_h}(v_1)\, \mathcal{I}^1_{\mathcal{T}_h}(v_2)\, \mathcal{I}^1_{\mathcal{T}_h}(v_3)\, ds, \\ (v_j(i_i))_{i \in [\![1,4]\!],\, j \in [\![1,3]\!]} &\longmapsto \int_{\tau \cap \Omega_h} \mathcal{I}^1_{\mathcal{T}_h}(v_1)\, \mathcal{I}^1_{\mathcal{T}_h}(v_2)\, \mathcal{I}^1_{\mathcal{T}_h}(v_3)\, dx, \end{aligned} \tag{11.16}$$

where $\mathcal{I}^1_{\mathcal{T}_h}(v_i) \in \mathbb{P}_1(\mathcal{T}_h; \mathbb{R})$ are the linear functions that coincide with v_i on each node of $\mathcal{T}_h$. The operator $\mathcal{I}^1_{\mathcal{T}_h}$ is the linear interpolation operator for the Lagrange finite element (see Definition 5.1.2). We are now able to compute usual criteria and their gradients.

Volume

For the volume criterion, we only need to compute

$$V(\Omega) = \int_\Omega 1,$$

and the respectively linear and bilinear maps l_1, l_2 defined for $w_1, w_2 \in C^1\left(\mathbb{R}^d; \mathbb{R}\right)$ by

$$\begin{aligned} l_1(w_1) &= \int_{\partial\Omega} w_1, \\ l_2(w_1, w_2) &= \int_{\partial\Omega} \mathcal{H}\, w_1\, w_2, \end{aligned}$$

that characterize the first and second-order shape derivatives (we recall that $\mathcal{H}$ is the mean curvature of the boundary).

To that end, we assemble a volume mass matrix M, and a boundary mass matrix M_Γ. For $w_1, w_2 \in \mathbb{Q}_1(\mathcal{Q}_h;\mathbb{R})$, we define W_1, W_2 by $W_1 = (w_1(e_1), \cdots, w_1(e_{N_h}))^T$ and $W_2 = (w_2(e_1), \cdots, w_2(e_{N_h}))^T$ where $(e_i)_{1\leq i\leq N_h}$ is the set of nodes of $\mathbb{Q}_1(\mathcal{Q}_h;\mathbb{R})$. The mass matrices M, M_Γ are defined by

$$
\begin{aligned}
{W_1}^T M W_2 &= \sum_{\tau\in\mathcal{T}_h} \int_{\tau\cap\Omega_h} \mathcal{I}^1_{\mathcal{T}_h}(w_1)\, \mathcal{I}^1_{\mathcal{T}_h}(w_2)\, dx,\\
{W_1}^T M_\Gamma W_2 &= \sum_{\tau\in\mathcal{T}_h} \int_{\mathscr{D}_\tau} \mathcal{I}^1_{\mathcal{T}_h}(w_1)\, \mathcal{I}^1_{\mathcal{T}_h}(w_2)\, ds.
\end{aligned}
$$

In order to get the value of the volume, it suffices to take $w_1 = w_2 = 1$ and to compute ${W_1}^T M W_2$. The first-order shape derivative is also obtained by computing $M_\Gamma W_1$ with $W_1 = 1$.

For the bilinear map of the second-order derivative, it suffices similarly to assemble the matrix H defined by

$$
{W_1}^T H W_2 = \sum_{\tau\in\mathcal{T}_h} \int_{\mathscr{D}_\tau} \mathcal{I}^1_{\mathcal{T}_h}(\mathcal{H}_h)\, \mathcal{I}^1_{\mathcal{T}_h}(w_1)\, \mathcal{I}^1_{\mathcal{T}_h}(w_2)\, ds. \tag{11.17}
$$

We recall that the curvature $\mathcal{H}$ is defined by $\mathrm{div}_\Gamma \mathbf{n}$. Thus, with finite difference scheme, one can get $\mathcal{H}_h$ an approximation of the curvature.

Thus, we have

$$
\begin{aligned}
V(\Omega) &\simeq E^T M E,\\
l_1(w_1) &\simeq {W_1}^T M_\Gamma E,\\
l_2(w_1, w_2) &\simeq {W_1}^T H W_2,
\end{aligned}
$$

where $E \in \mathbb{R}^{N_h}$ with $E = (1, \cdots, 1)$.

Finally, in order to ease the numerical aspects, we use the mass-lumping approximation (see Section 5.2) for M, M_Γ and H. The error analysis stays the same than in the classical case (see Lemma 5.2.2).

Let us also mention that the boundary mass matrix M_Γ (or more precisely its mass-lumping approximation) is used in the optimization method : in the gradient algorithm for computing a descent direction (10.28), or for the extension of the descent direction in Method 10.2.3.

Compliance

For the compliance criterion, we will consider here the framework of the *scalar thermal model* (see Section 2.1.4). The criterion reads

$$
J_1(\Omega) = \int_\Omega |\nabla u|^2,
$$

where $u \in H^1(\Omega;\mathbb{R})$ solves

$$
\left\{
\begin{array}{rcll}
-\mathrm{div}\,(\nabla u) &=& 0 & \text{in } \Omega,\\
\partial_{\mathbf{n}} u &=& g & \text{on } \Gamma_N \cup \Gamma,\\
u &=& 0 & \text{on } \Gamma_D,
\end{array}
\right. \tag{11.18}
$$

for $g \in H^2(\Omega)$ with $g = 0$ on Γ. With the Hilbert space $\mathcal{V} = \left\{p \in H^1(\Omega;\mathbb{R}) \mid p = 0 \text{ on } \Gamma_D\right\}$, we recall that the variational formulation reads

$$
\left\{
\begin{array}{l}
\text{find } u \in \mathcal{V} \text{ such that },\\
\forall p \in \mathcal{V},\ \displaystyle\int_\Omega \nabla u \cdot \nabla p = \int_{\Gamma_N} gp.
\end{array}
\right. \tag{11.19}
$$

We also recall that, for $w_1, w_2 \in C^1(\mathbb{R}^d,\mathbb{R})$, when only the part Γ can be optimized, the linear and bilinear maps l_1 and l_2 that characterize the first and second-order shape derivatives are defined by

$$
\begin{aligned}
l_1(w_1) &= -\int_\Gamma |\nabla u|^2 w_1\\
l_2(w_1, w_2) &= 2\int_\Omega \nabla u'_{w_1} \cdot \nabla u'_{w_2} - \int_\Gamma \Big(\mathcal{H}|\nabla u|^2 + \partial_{\mathbf{n}}|\nabla u|^2\Big) w_1\, w_2,
\end{aligned}
$$

where for $w \in H^1(\Omega;\mathbb{R})$, u'_w is defined by

$$
\left\{
\begin{array}{rcll}
-\mathrm{div}\,(\nabla u'_w) &=& 0 & \text{in } \Omega,\\
\partial_{\mathbf{n}} u'_w &=& 0 & \text{on } \Gamma_N,\\
\partial_{\mathbf{n}} u'_w &=& \mathrm{div}\,(w\nabla_\Gamma u) & \text{on } \Gamma,\\
u'_w &=& 0 & \text{on } \Gamma_D.
\end{array}
\right. \tag{11.20}
$$

From a numerical point of view, we look for an approximate solution u_h of (11.18) in $\mathbb{Q}_1(\mathcal{Q}_h;\mathbb{R})$. Thus, the partial derivatives of u_h - $\partial_x u_h, \partial_y u_h$ - are linear in each triangular element τ of $\mathcal{T}_h$. For $u_h, p_h \in \mathbb{Q}_1(\mathcal{Q}_h;\mathbb{R})$, with U, P defined by $U = (u_h(e_1), \cdots, u_h(e_{N_h}))^T$ and $P = (p_h(e_1), \cdots, p_h(e_{N_h}))^T$ we can introduce the stiffness matrix A_h defined by

$$P^T A_h U = \sum_{\tau \in \mathcal{T}_h} \int_{\tau \cap \Omega_h} \partial_x u_h \, \partial_x p_h + \partial_y u_h \, \partial_y p_h.$$

It is common for the stiffness matrix to use an ersatz material (see [15] for details), so that it prevents this matrix from being ill-conditioned. In the same spirit, with an ersatz material with thermal conductivity $\varepsilon \ll 1$, we replace the matrix A_h by the matrix A_ε such that

$$P^T A_\varepsilon U = P^T A_h U + \varepsilon P^T A_{\mathcal{D}\setminus\Omega,h} U,$$

where the matrix $A_{\mathcal{D}\setminus\Omega,h}$ defined by

$$P^T A_{\mathcal{D}\setminus\Omega,h} U = \sum_{\tau \in \mathcal{T}_h} \int_{\tau \cap \Big(\mathcal{D}\setminus\big(\Omega_h \cap \partial\Omega_h\big)\Big)} \partial_x u_h \, \partial_x p_h + \partial_y u_h \, \partial_y p_h,$$

is nothing but the stiffness matrix associated to the complementary domain of Ω in $\mathcal{D}$.

If we denote $G = (g(e_1), \cdots, g(e_{N_h}))^T$, where the source term g is assumed to be in $\mathbb{Q}_1(\mathcal{Q}_h;\mathbb{R})$, the variational formulation (11.19) now reads

$$\left\{ \begin{array}{l} \text{find } U \in \mathbb{R}^{N_h}, \text{ such that }, \\ \forall P \in \mathbb{R}^{N_h}, \; P^T A_\varepsilon U = P^T M_\Gamma G, \end{array} \right. \tag{11.21}$$

where N_h is the dimension of $\mathbb{Q}_1(\mathcal{Q}_h;\mathbb{R})$.

In order to compute the gradient, we may proceed similarly. Once the state u_h is computed by (11.21), we just have to compute l_1 for $w_h \in \mathbb{Q}_1(\mathcal{Q}_h;\mathbb{R})$. Thanks to (11.16), given that $\partial_x u_h, \partial_y u_h$ are linear in each triangle τ of $\mathcal{T}_h$, we numerically compute the first-order shape derivative with

$$l_1(w_h) \simeq -\sum_{\tau \in \mathcal{T}_h} \int_{\mathscr{D}_\tau \cap \Gamma_h} \Big((\partial_x u_h)^2 + (\partial_y u_h)^2\Big) \mathcal{I}^1_{\mathcal{T}_h}(w_h),$$

where Γ_h is the linear approximation of Γ.

As concerns the second-order derivative of the compliance, it is somehow more tricky. We start by writing the variational formulation for (11.20) :

$$\left\{ \begin{array}{l} \text{find } u'_w \in \mathcal{V}, \text{ such that }, \\ \forall p \in \mathcal{V}, \; \displaystyle\int_\Omega \nabla u'_w \cdot \nabla p = -\int_\Gamma w \nabla u \cdot \nabla p. \end{array} \right. \tag{11.22}$$

It suffices to use an integration by part on the right-hand side, and the fact that, since $\partial_{\mathbf{n}} u = 0$ on Γ, we have $\nabla_\Gamma u = \nabla u$. As usual, thanks to (11.16) we can assemble a matrix $F(u_h)$ such that for $w_h, p_h \in \mathbb{Q}_1(\mathcal{Q}_h;\mathbb{R})$

$$P^T F(u_h) W = -\sum_{\tau \in \mathcal{T}_h} \int_{\mathscr{D}_\tau \cap \Gamma_h} \Big(\partial_x u_h \, \partial_x p_h + \partial_y u_h \, \partial_y p_h\Big) \mathcal{I}^1_{\mathcal{T}_h}(w_h). \tag{11.23}$$

Thus, the discretization of the variational formulation (11.22) reads

$$\left\{ \begin{array}{l} \text{find } U_w \in \mathbb{R}^{N_h}, \text{ such that }, \\ \forall P \in \mathbb{R}^{N_h}, \; P^T A_\varepsilon U_w = P^T F(u) W. \end{array} \right. \tag{11.24}$$

As a result, for $w_{1,h}, w_{2,h} \in \mathbb{Q}_1(\mathcal{Q}_h;\mathbb{R})$ we can approximate the first term of l_2 by

$$\begin{aligned} l_2(w_{1,h}, w_{2,h}) &= {U_{w_1}}^T A_\varepsilon U_{w_2} \\ &= \Big(A_\varepsilon^- 1 F(u_h) W_1\Big)^T \, A_\varepsilon \, \Big(A_\varepsilon^- 1 F(u_h) W_2\Big) \\ &= {W_1}^T F(u_h)^T A_\varepsilon^{-1} F(u_h) W_2. \end{aligned}$$

For the last part of the second-order derivative, $|\nabla u_h|$ is a priori not continuous from one triangle of $\mathcal{T}_h$ to another one. Therefore, with this linear interpolation framework it is not possible to compute its normal gradient. We then have to proceed differently. Thus, we compute the term $\mathcal{H}_h |\nabla u_h|^2 + \partial_{\mathbf{n}}\big(|\nabla u_h|^2\big)$ with finite difference and proceed like (11.17) to get the bilinear map.

Remark 11.2.2. *This numerical method for computing boundary integrals is more precise than the approximation with a Dirac function. It is more time consuming but the additional burden is completely negligible in front of the CPU time required for the finite element analysis.*

Remark 11.2.3. *For the displacement criterion in the framework of the* elastic model*, we can proceed in exactly the same way.*

11.2.3 Error analysis

After having introduced a numerical method for computing boundary integrals, we are concerned with its error analysis. There are two sources of error. First, errors can originate from the approximation of the integrand in (11.16). These errors are due to the difference between $\mathcal{I}^1_{\mathcal{T}_h}(v)$ and v for $v \in C^0\left(\mathbb{R}^d;\mathbb{R}\right)$. The analysis of these errors is classical for finite element methods. Therefore, we will rather focus on the other source of numerical error, originated by the linear approximation of the domain. A shape Ω is represented implicitly by a level-set function ϕ. The linear approximation Ω_h of the shape Ω is done implicitly by a linear interpolation of the level-set, that is

$$\Omega_h = \left\{x \in \mathcal{D} \mid \mathcal{I}^1_{\mathcal{T}_h}(\phi)(x) < 0\right\}.$$

We can find similar error estimates on boundary integrals than the ones we consider here, for example in the framework of the *boundary elements method* [76, 82]. The main difference is that most of the time, the approximated shape is a polygonal domain that coincides with the triangulation. This means that the boundary is made of edges of the mesh. In our case, the approximated shape is defined implicitly by the interpolation of a level-set function, and thus the mesh is not adapted to the triangulation. As a result, the boundary has a priori no chance to be made of edges of the mesh, which is another source of error.

We start by recalling that when there exists a point where the gradient of the level-set function does not vanish, the boundary of a shape can be defined (at least locally) explicitly by the graph of a function.

Proposition 11.2.4. *Let $\phi \in C^k\big(\mathbb{R}^2;\mathbb{R}\big)$. Let $(x_0,y_0) \in \mathbb{R}^2$ such that $\phi(x_0,y_0)=0$, and assume that $\partial_y\phi(x_0,y_0) \neq 0$. For $r>0$ and $(x,y) \in B\big((x_0,y_0),r\big)$ we introduce*

$$T_x(y) = y - \big(\partial_y\phi(x_0,y_0)\big)^{-1}\phi(x,y).$$

Then if r is sufficiently small, there exists φ in $C^k\big(B(x_0,r);\mathbb{R}\big)$ such that $\forall (x,y) \in B\big((x_0,y_0),r\big)$,

$$\phi(x,y(x)) = 0 \Leftrightarrow y = \varphi(x), \tag{11.25}$$

and $\varphi(x)$ is the unique fixed point of $B(y_0,r) \ni y \mapsto T_x(y)$.

Proof. For $(x,y) \in B\big((x_0,y_0),r\big)$ we have

$$dT_x(y) = \mathrm{Id} - \big(\partial_y\phi(x_0,y_0)\big)^{-1}\partial_y\phi(x,y),$$

and $dT_{x_0}(y_0)=0$. The continuity of dT ensures that if r is sufficiently small

$$\forall (x,y) \in B\big((x_0,y_0),r\big), \quad |dT_x(y)| \leq \frac{1}{2}.$$

Therefore T_x is a contraction with constant $\frac{1}{2}$, and

$$\begin{aligned}|T_x(y)-y_0| &\leq |T_x(y)-T_x(y_0)| + |T_x(y_0)-y_0| \\ &\leq \frac{1}{2}|y-y_0| + \left|\big(\partial_y\phi(x_0,y_0)\big)^{-1}\phi(x,y_0)\right|.\end{aligned}$$

Since ϕ is continuous and $\phi(x_0,y_0)=0$, when r is sufficiently small we have

$$\left|\big(\partial_y\phi(x_0,y_0)\big)^{-1}\phi(x,y_0)\right| \leq \frac{r}{2}.$$

We then get $|T_x(y)-y_0| \leq r$ and

$$T_x\big(B(y_0,r)\big) \subset B(y_0,r).$$

Picard's fixed point theorem then ensures the existence of $\varphi(x)$ the unique fixed point of T_x, and that φ is of class C^k. □

Thanks to this implicit function theorem, the shape as well as its linear approximation can be represented (at least locally) by the graph of a function. Assuming that the level-set ϕ and its linear approximation coincide at some point, we can get an error estimate on the two different graphs.

Corollary 11.2.5. *Let $\phi, \phi_h \in C^k\big(\mathbb{R}^2;\mathbb{R}\big)$. Let $(x_0,y_0) \in \mathbb{R}^2$, such that $\phi(x_0,y_0) = \phi_h(x_0,y_0) = 0$. Assume that $\partial_y\phi(x_0,y_0) \neq 0$ and $\partial_y\phi_h(x_0,y_0) \neq 0$. Then there exists $r>0$ and two functions φ and φ_h in $C^k\big(B(x_0,r);\mathbb{R}\big)$, such that $\forall (x,y) \in B\big((x_0,y_0),r\big)$,*

$$\phi(x,y) = 0 \Leftrightarrow y = \varphi(x), \quad \text{and} \quad \phi_h(x,y) = 0 \Leftrightarrow y = \varphi_h(x) \tag{11.26}$$

and

$$\begin{aligned}\frac{1}{2}|\varphi(x)-\varphi_h(x)| \leq \; & \left|\Big(\big(\partial_y\phi_h(x_0,y_0)\big)^{-1} - \big(\partial_y\phi(x_0,y_0)\big)^{-1}\Big)\phi_h(x,y)\right. \\ & \left. + \big(\partial_y\phi(x_0,y_0)\big)^{-1}\Big(\phi_h(x,y)-\phi(x,y)\Big)\right|\end{aligned} \tag{11.27}$$

Proof. For $(x,y)\in\mathbb{R}^2$ we introduce

$$T_x(y) = y - \big(\partial_y\phi(x_0,y_0)\big)^{-1}\phi(x,y)$$
$$S_x(y) = y - \big(\partial_y\phi_h(x_0,y_0)\big)^{-1}\phi_h(x,y)$$

Since $\partial_y\phi(x_0,y_0)\neq 0$ and $\partial_y\phi_h(x_0,y_0)\neq 0$, there exists $r>0$, $\varphi,\varphi_h\in C^k\big(B(x_0,r);\mathbb{R}\big)$ such that (11.25) is satisfied (we have two radii r_1,r_2, for ϕ and ϕ_h and we take $\min\{r_1,r_2\}$), and also such that for $x\in B(x_0,r)$, the functions $B(y_0,r)\ni y\mapsto T_x(y)$ and $B(y_0,r)\ni y\mapsto S_x(y)$ are contractions with constant $\frac{1}{2}$. Now we have for $x\in B(x_0,r)$

$$\begin{aligned}|\varphi(x)-\varphi_h(x)| &\le |T_x\big(\varphi(x)\big)-S_x\big(\varphi_h(x)\big)|\\ &\le |T_x\big(\varphi(x)\big)-S_x\big(\varphi(x)\big)|+|S_x\big(\varphi(x)\big)-S_x\big(\varphi_h(x)\big)|\\ &\le |T_x\big(\varphi(x)\big)-S_x\big(\varphi(x)\big)|+\frac{1}{2}|\varphi(x)-\varphi_h(x)|\end{aligned}$$

Therefore

$$|\varphi(x)-\varphi_h(x)|\le 2|T_x\big(\varphi(x)\big)-S_x\big(\varphi(x)\big)|.$$

Since

$$\begin{aligned}T_x\big(\varphi(x)\big)-S_x\big(\varphi(x)\big) =&\big(\partial_y\phi_h(x_0,y_0)\big)^{-1}\phi_h(x,y)-\big(\partial_y\phi(x_0,y_0)\big)^{-1}\phi(x,y)\\ =&\Big(\big(\partial_y\phi_h(x_0,y_0)\big)^{-1}-\big(\partial_y\phi(x_0,y_0)\big)^{-1}\Big)\phi_h(x,y)\\ &+\big(\partial_y\phi(x_0,y_0)\big)^{-1}\Big(\phi_h(x,y)-\phi(x,y)\Big),\end{aligned}$$

we have the expected result. □

Remark 11.2.6. *Coming back to Remark 5.1.4 and taking $p=+\infty$, we have for all $\phi\in C^1\left(\mathbb{R}^d;\mathbb{R}\right)\cap L^\infty\left(\mathbb{R}^d;\mathbb{R}\right)$*

$$\left\|\phi-\mathcal{I}^1_{\mathcal{T}_h}(\phi)\right\|_{L^\infty(\mathbb{R}^d;\mathbb{R})}\le ch^2\left\|\phi\right\|_{L^\infty(\mathbb{R}^d;\mathbb{R})}. \tag{11.28}$$

We also have

$$\left\|\nabla\phi-\nabla\mathcal{I}^1_{\mathcal{T}_h}(\phi)\right\|_{L^\infty(\mathbb{R}^d;\mathbb{R})}=O(h).$$

The level-set function is often taken close to a signed distance function d_Ω that satisfies $|\nabla d_\Omega|=1$. Therefore, it is possible to choose a coordinate system (x,y) such that both the partial derivatives of ϕ and $\mathcal{I}^1_{\mathcal{T}_h}(\phi)$ in the direction y are away from 0. As a consequence we have by (11.27) in the proof of Corollary 11.2.5

$$|\varphi(x)-\varphi_h(x)|\le O(h)\phi_h(x,y)+O(h^2). \tag{11.29}$$

Finally if we take $r=O(h)$ in this proof, since ϕ is close to a distance function and $\phi(x_0,y_0)=0$ we also have $\phi(x,y)=O(h)$ and therefore

$$|\varphi(x)-\varphi_h(x)|=O(h^2) \tag{11.30}$$

for $x,y\in B\big((x_0,y_0),r\big)$.

Remark 11.2.7. *We can also mention that the linear interpolation $\mathcal{I}^1_{\mathcal{T}_h}$ is piecewise C^1, and C^1 in each cell τ of the mesh $\mathcal{T}_h$. Since the estimate (11.30) is only local, it can be obtained on each domain of regularity of $\mathcal{I}^1_{\mathcal{T}_h}$. Therefore the regularity assumption in Corollary 11.2.5 can be replaced by the same piecewise regularity assumption.*

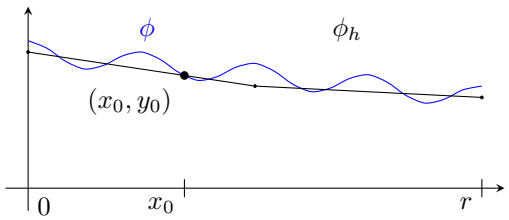


Figure 11.6: Interpolation error between ϕ and ϕ_h.

Now assuming that the second-order accuracy of (11.30) is satisfied, we can get an approximation error on boundary integrals.

Proposition 11.2.8. *Let $\phi \in C^k\left(\mathbb{R}^d;\mathbb{R}\right)$ and $\phi^1_{\mathcal{T}_h}(\phi)$ its $\mathbb{P}_1$-interpolation on $\mathcal{T}_h$. Let $(x_0,y_0) \in \mathbb{R}^d$ such that $\phi(x_0,y_0) = \phi^1_{\mathcal{T}_h}(\phi)(x_0,y_0)$. According to Corollary 11.2.5, there exists $r>0$, φ and φ_h defined on $[0,r]$ such that*

$$\forall (x,y) \in [0,r]\times B(y_0,r),\ \phi(x,y)=0 \Leftrightarrow y=\varphi(x) \quad \text{and} \quad \mathcal{I}^1_{\mathcal{T}_h}(\phi)(x,y)=0 \Leftrightarrow y=\varphi_h(x).$$

Let now assume that on $[x_0,r]$,

$$|\nabla\varphi| = O(h) \quad \text{and} \quad |\nabla\varphi_h| = O(h). \tag{11.31}$$

We denote γ and γ_h the respective boundaries parametrized by $T:[0,r]\ni t\mapsto (t,\varphi(t))$ and $T_h:[0,r]\ni t\mapsto (t,\varphi_h(t))$. Then, for $v\in C^1(\mathbb{R}^d;\mathbb{R})$, we have

$$\left|\int_\gamma v(x)\,dx - \int_{\gamma_h} v(x)\,dx\right| = O(rh^2). \tag{11.32}$$

Proof. Let ε be defined by

$$\varepsilon = \int_\gamma v(x)\,dx - \int_{\gamma_h} v(x)\,dx.$$

We have by a change of variable

$$\begin{aligned}
\varepsilon &= \int_{[0,r]} \Big(v\circ T(x)|\nabla T(x)| - v\circ T_h(x)|\nabla T(x)|\Big)\,dx,\\
&\le \left|\int_{[0,r]} \Big(v\circ T(x) - v\circ T_h(x)\Big)|\nabla T_h(x)|\,dx + \int_{[0,r]} v\circ T_h(x)\Big(|\nabla T(x)| - |\nabla T_h(x)|\Big)\,dx\right|.
\end{aligned}$$

Since v is of class C^1, it is Lipschitz. Therefore, there exists $L>0$ such that

$$\forall X,Y \in \left([0,r]\times B(y_0,r)\right)^2,\ |v(X)-v(Y)| \le L|X-Y|.$$

Thus, for $x\in[0,r]$, we have

$$\begin{aligned}
\Big|v\circ T(x)|\nabla T(x)| - v\circ T_h(x)\Big| &\le L|T(x)-T_h(x)|\\
&\le L|\varphi(x)-\varphi_h(x)|\\
&= O(h^2)
\end{aligned}$$

As a consequence, the first term of ε is of third-order with respect to h :

$$\int_{[0,r]} \Big(v\circ T(x) - v\circ T_h(x)\Big)|\nabla T_h(x)|\,dx = O(rh^2).$$

Now we can also write

$$\begin{aligned}
|\nabla T(x)| &= \sqrt{1+|\nabla\varphi(x)|^2},\\
|\nabla T_h(x)| &= \sqrt{1+|\nabla\varphi_h(x)|^2}.
\end{aligned}$$

Thus,

$$|\nabla T(x)| - |\nabla T_h(x)| = \frac{|\nabla\varphi(x)|^2 - |\nabla\varphi_h(x)|^2}{\sqrt{1+|\nabla\varphi(x)|^2}+\sqrt{1+|\nabla\varphi_h(x)|^2}},$$

which is of order two with respect to h, whenever $|\nabla\varphi| = O(h)$ and $|\nabla\varphi_h| = O(h)$:

$$|\nabla T(x)| - |\nabla T_h(x)| = O(h^2).$$

Since v is continuous and bounded, we finally obtain that

$$|\varepsilon| = O(rh^2).$$

□

Remark 11.2.9. *In Proposition 11.2.8, the assumption that $|\nabla\varphi|$ as well as $\nabla\varphi_h$ are of order $O(h)$ is not that strong. We can choose the coordinate system such that on x_0 we have $\nabla\varphi(x_0)=0$. Thus, $\nabla\varphi$ stays of order h in $[0,r]$, and since the error between $\nabla\varphi$ and $\nabla\varphi_h$ is of the same order, we also have $\nabla\varphi_h = O(h)$.*

Remark 11.2.10. *Finally it remains to sum up all these estimates. We assume that there exists finite families* $(x_i, y_i)_{1\leq i\leq N}$, $(r_i)_{1\leq i\leq N}$, $\varphi_i : [0, r_i] \to \mathbb{R}$ *and* $\varphi_{h,i} : [0, r_i] \to \mathbb{R}$ *such that*

$$\forall 1 \leq i \leq N, \quad \phi(x_i, y_i) = \mathcal{I}^1_{\mathcal{T}_h}(\phi)(x_i, y_i) = 0, \tag{11.33}$$

$$\forall 1 \leq i \leq N, \ \forall (x, y) \in [0, r_i] \times B(y_i, r_i), \ \phi(x, y) = 0 \Leftrightarrow y = \varphi_i(x) \tag{11.34}$$

$$\forall 1 \leq i \leq N, \ \forall (x, y) \in [0, r_i] \times B(y_i, r_i), \ \phi_h(x, y) = 0 \Leftrightarrow y = \varphi_{h,i}(x). \tag{11.35}$$

We also assume that

$$\{(x, y) \in \mathbb{R}^d \mid \exists i \in [\![0, N]\!], \ x \in [0, r_i] \text{ and } y = \varphi_i(x)\} \supset \partial\Omega, \tag{11.36}$$

and

$$\forall 1 \leq i \leq N, \quad |\nabla\varphi_i| = O(h) \text{ and } |\nabla\varphi_{h,i}| = O(h).$$

Adding all the error estimates (11.32) leads to

$$\left| \int_{\partial\Omega} v - \int_{\partial\Omega_h} v \right| = \sum_{i=1}^{N} r_i O(h^2). \tag{11.37}$$

Since for regular domains $\sum_{i=1}^{N} r_i$ *can be chosen of the same order as the measure of* $\partial\Omega$*, we finally get*

$$\left| \int_{\partial\Omega} v - \int_{\partial\Omega_h} v \right| = O(h^2).$$

Remark 11.2.11. *When the function* ϕ *is in* $\mathbb{Q}_1(\mathcal{Q}_h; \mathbb{R}^2)$*, it is linear on each edge of a cell* K*. Thus it coincides with its* P_1 *interpolation* $\mathcal{I}^1_{\mathcal{T}_h}(\phi)$ *on the edges. Thus the boundaries* $\partial\Omega$ *and* $\partial\Omega_h$ *intersect on the edges of the cell (points* A *or* C *in Figure 11.7). Since* $\Omega \in \mathbb{Q}_1(\mathcal{Q}_h; \mathbb{R})$ *the zero level-set of* ϕ *is a hyperbola in the cell* K*. It might be at most*

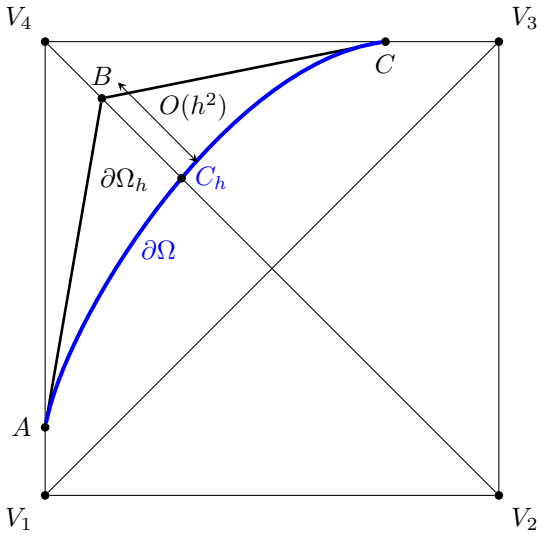


Figure 11.7: Interpolation error between ϕ and ϕ_h in cell K.

divided in two parts like in the case described in Figure 11.3. Let us denote γ *one of this part. Since* ϕ *is in* $\mathbb{Q}_1$ *there exists a coordinate system and a graph such that*

$$(x, y) \in \gamma \Leftrightarrow y = \varphi(x).$$

This means that the graph given by the implicit function theorem works for the whole part γ*. We can also choose the coordinate system such that there exists* x_0 *where* $\nabla\varphi(x_0) = 0$*. Then, since* φ *is smooth and the cell is of size* h *we have easily* $|\nabla\varphi| = O(h)$ *in the cell. Knowing that the boundaries of* Ω *and* Ω_h *coincide on the edges of the cell we can use Proposition 11.2.8 to conclude that in a cell, the integration error is of order* $O(h^3)$*. Therefore we can sum over all the cells crossed by the boundary, and conclude that the global integration error is of order* $O(h^2)$*.*

11.3 Accuracy and consistency comparison

We have seen in Section 11.1 that the computation of a boundary integral with the help of a Dirac function concentrated on the boundary may not be sufficiently consistent. We introduced therefore in Section 11.2 another numerical method for computing boundary integrals, based on linear interpolation. Now we aim at comparing the respective consistencies and accuracies of the two methods.

11.3.1 Accuracy

In order to check that this computation is relevant, we first compute the perimeters of a circle of radius 1 and of two squares. The first square - referred to as `square1` - is such that its size is exactly a multiple of the mesh size, so that its boundary is always on the edges of the mesh. The number of cells on each edge of the square is given by $2\lfloor\frac{0.85}{h}\rfloor$. The second square - referred to as `square2` - has a side of size 1.96. The boundary is not made of edges of elements of the mesh. The size of the mesh is (100×100), and the working domain is of size 2.3×2.3 thus h is given by $h = \frac{2.3}{100}$. For the approximations of the Dirac mass, the parameter ε is defined by $\varepsilon = \frac{h}{20}$. We compare the linear interpolation

Figure 11.8: Shapes for consistency study.

method (introduced in Section 11.2) respectively with the quadrature methods using the approximations $\delta_{1,\varepsilon}$ and $\delta_{2,\varepsilon}$ of Dirac mass $\delta_{\partial\Omega}$ given by (11.4) and (11.6). The results are shown in Table 11.1.

	Theoretical perimeter	Linear interpolation method	Approximation $\delta_{1,\varepsilon}$	Approximation $\delta_{2,\varepsilon}$
circle	6.283185	6.282757	6.602255	6.262644
square1	6.808000	6.808000	6.781012	6.762995
square2	7.840000	7.811479	7.793377	7.761437

Table 11.1: Numerical computation of perimeters for $h = \frac{2.3}{100}$ and $\varepsilon = \frac{h}{20}$.

We can observe that using the linear interpolation framework for computing boundary integrals seems to be much more accurate than using an approximation if the Dirac mass $\delta_{\partial\Omega}$. It is also noticeable that in the case of `square1` when the boundary is part of the edges of the mesh, the linear interpolation method leads to an exact computation of the boundary. This was expected since when the level-set function vanishes on two vertices of an element, the length of the boundary is exactly the length of the edge.

In order to complete the validation of the numerical method we study on the same cases the evolution of the error on the perimeter when decreasing the mesh size h. We can compare the convergence of the absolute error for different methods, in logarithmic scale. In figures 11.9, 11.10, 11.11 we respectively plot the convergence of the error compared to the decrease of h the size of the mesh for the circle, the first square and the second square.

In the case of the circle, the linear interpolation method of Section 11.2 seems to be second-order in accordance with the theoretical results obtained in Section 11.2.3. We can also see that the accuracy of the computation with the Dirac approximation $\delta_{1,\varepsilon}$ is not very satisfactory. For the first square, in Figure 11.10, the curve for the linear interpolation method is not plotted since the numerical error is always reduced to the machine precision. However, in the case of the square whose boundary does not consist of edges of the mesh, all approximation methods seem to have the same rate of convergence. We can also observe that the linear interpolation method is not second-order any more. This could be explained by the fact that the shape has corners that are not accurately captured by the set of edges of the mesh. Therefore in a cell containing a corner, the linear interpolation of the level-set is not accurate, leading to a larger global error.

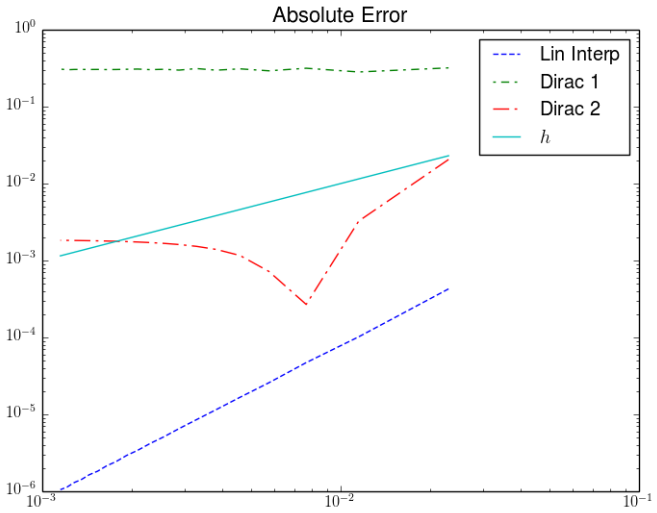


Figure 11.9: Absolute error on the circle.

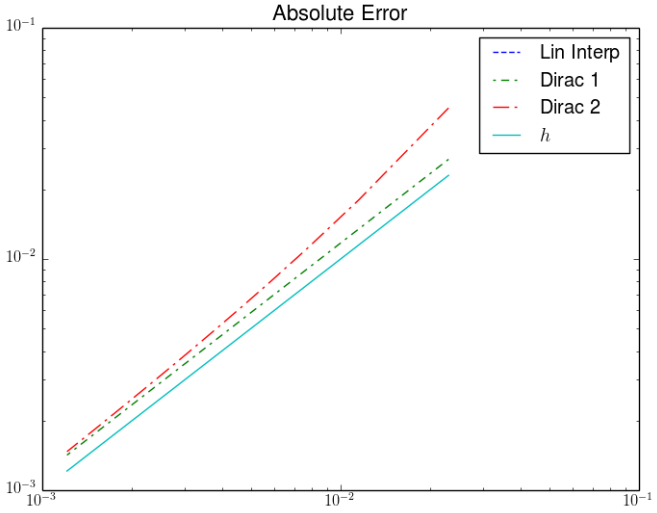


Figure 11.10: Absolute error on the first square.

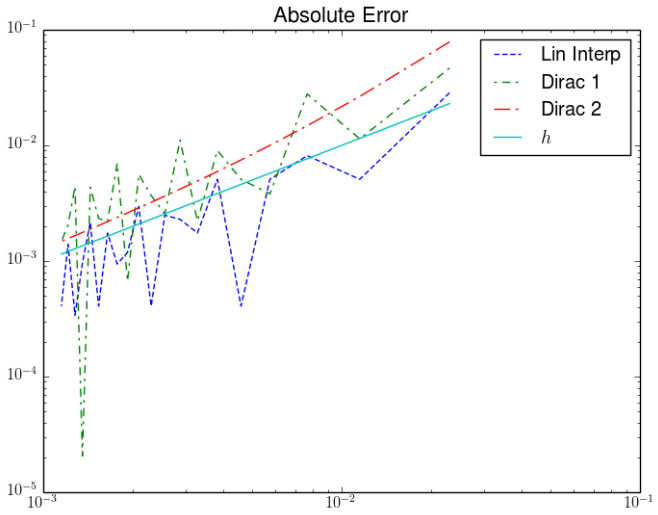


Figure 11.11: Absolute error on the second square.

11.3.2 Consistency

Secondly, we want to check that for a function vanishing on the boundary, the boundary integral vanishes. For this purpose, we go back to Example 11.1.5. The objective function to be minimized reads

$$E(\Omega) = \int_\Omega f,$$

where f is defined by (11.7). We recall from Section 3.3.1 and Section 7.1.2 that when Ω is sufficiently regular, $\Omega \mapsto E(\Omega)$ is shape-differentiable. In that case, for $v, w \in C^1(\mathbb{R}^d;\mathbb{R})$, the linear and bilinear maps l_1 and l_2 that characterize the shape derivatives of E read

$$l_1(v) = \int_{\partial\Omega} fv,$$
$$l_2(v,w) = \int_{\partial\Omega} \big(\mathcal{H}f + \partial_{\mathbf{n}} f\big)vw.$$

We also recall that $\Omega^\star = \{x \in \mathbb{R}^d \mid f(x) < 0\}$ is the optimal shape for the minimization of E. By construction, we have $f = 0$ on $\partial\Omega^\star$. We compare the approximate values of $\|f\|_{L^2(\partial\Omega_0;\mathbb{R})}$ and $\|f\|_{L^2(\partial\Omega^\star;\mathbb{R})} = 0$ obtained by the linear interpolation method (Section 11.2) and the two approximations of the Dirac mass $\delta_{\partial\Omega_0}$ and $\delta_{\partial\Omega^\star}$ (Section 11.1). For a mesh of size $h = \frac{2.3}{120}$ and a small parameter $\varepsilon = \frac{h}{20}$, the results are given in Table 11.2.

	Linear approximation	Approximation $\delta_{1,\varepsilon}$	Approximation $\delta_{2,\varepsilon}$
$\|f\|_{L^2(\Omega_0;\mathbb{R})}$	$1.907891\,10^{-1}$	$1.972436\,10^{-1}$	$1.560282\,10^{-1}$
$\|f\|_{L^2(\Omega^\star;\mathbb{R})}$	$5.049777\,10^{-10}$	$2.065797\,10^{-1}$	$2.163689\,10^{-1}$

Table 11.2: Approximation of $\|f\|_{L^2(\Omega_0;\mathbb{R})}$ and $\|f\|_{L^2(\Omega^\star;\mathbb{R})}$.

With the linear interpolation method, the value of the L^2-norm of f on $\partial\Omega^\star$ is almost zero, whereas with the approximations $\delta_{1,\varepsilon}$, $\delta_{2,\varepsilon}$ of the Dirac mass, the integral of f^2 on $\partial\Omega^\star$ is even larger than on $\partial\Omega_0$. The L^2-norm of f on the boundary is a norm of the first-order shape derivative. Therefore with the approximations of the Dirac mass, the gradient does not decrease between the initial and optimal shape. Usually an optimization process is stopped when the gradient is close enough to zero. In this case with the approximations $\delta_{1,\varepsilon}$, $\delta_{2,\varepsilon}$ of the Dirac mass, such a stopping criterion may never be satisfied.

Let us focus on the first approximation of the Dirac mass $\delta_{1,\varepsilon}$, with the same parameters $h = \frac{2.3}{120}$, $\varepsilon = \frac{h}{20}$. We can compare the values of $f(x)\delta_{1,\varepsilon}(x)$ on the domain $\mathcal{D}$ for the initial and optimal shapes. At the optimal shape (bottom of Figure 11.12) we can see that the quantity $\delta_{1,\varepsilon} f$ goes rapidly from positive values to negative values of roughly the same magnitudes when crossing the interface $\partial\Omega^\star$. Its zero level-set coincides with the interface $\partial\Omega^\star$. But summing over $\mathcal{D}$ leads to a non-zero value.

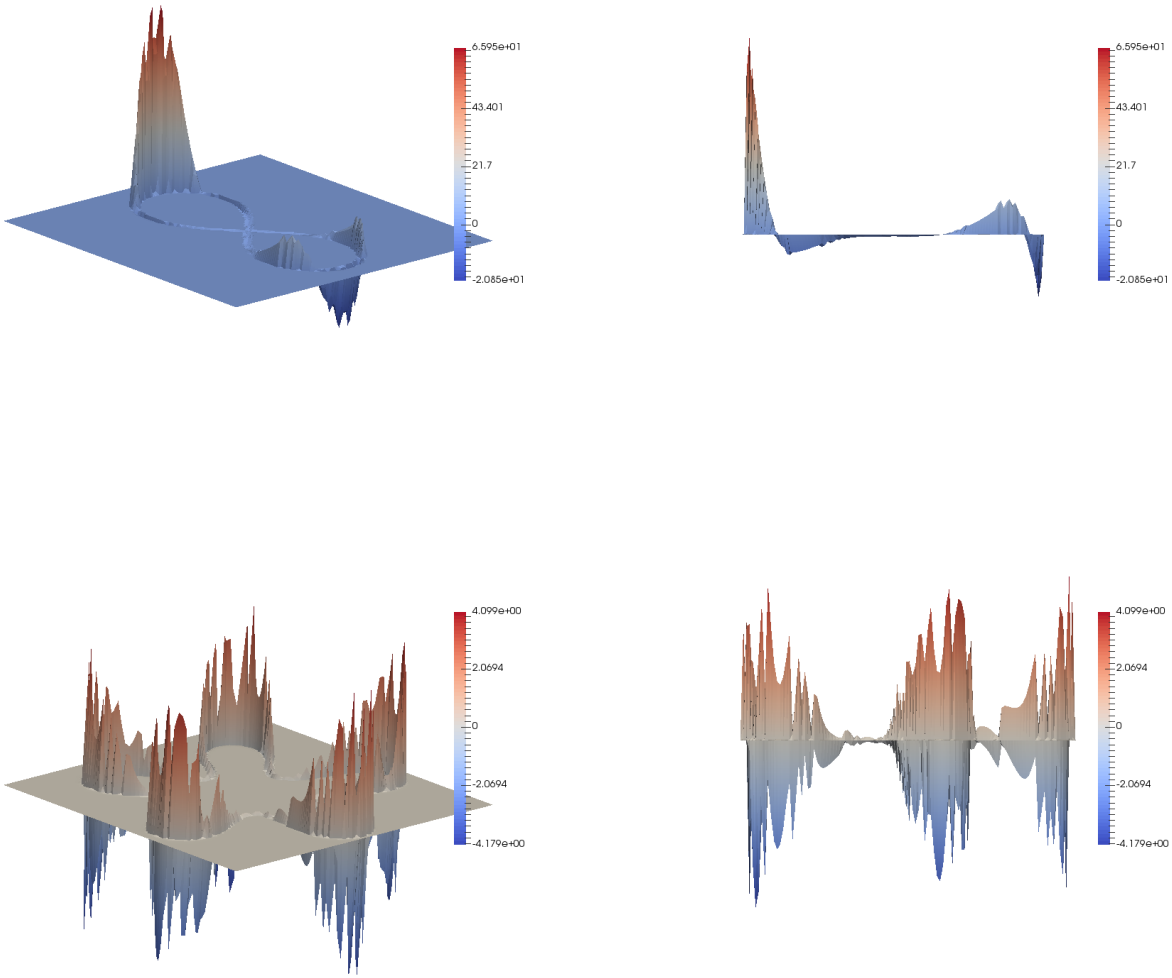


Figure 11.12: Values of $f(x)\delta_{1,\varepsilon}(x)$ at the initial shape (top) and at the optimal shape (bottom) for $h = \frac{2.3}{120}$ and $\varepsilon = \frac{h}{20}$.

Now we proceed as in Section 11.1.2 to compare the consistency of the linear interpolation method with the consistency of the Dirac approximation. Like in Section 11.1.2, we take v_γ given by the solution to the gradient equation (10.28), v being the extension of v_γ by Method 10.2.3. For different values of the final time τ of resolution of the Hamilton-Jacobi equation we plot in Figure 11.14 the values of

$$\tau \mapsto E(\Omega_\tau),$$
$$\tau \mapsto E(\Omega_0) + \tau E'(\Omega_0; v),$$
$$\tau \mapsto E(\Omega_0) + \tau E'(\Omega_0; v) + \frac{1}{2}\tau^2 E''(\Omega_0; v, v).$$

In order to ease the reading for the comparison we plot again in Figure 11.13 what gave the approximations $\delta_{1,\varepsilon}$, $\delta_{2,\varepsilon}$ of the Dirac mass for the same example (see Section 11.1.2).

We can now observe in Figure 11.14 that the linear approximation of E where the boundary integrals are computed with the linear interpolation method of Section 11.2 is tangent to the real curve $t \mapsto E(\Omega_t)$. This means that the approach is consistent, since the numerical computation of the continuous derivative coincides with the exact derivative. We can also see that the quadratic approximation is roughly relevant. In order to look at the second-order approximation we also plot in Figure 11.15 the following curves for the linear interpolation method

$$\tau \mapsto E(\Omega_\tau) - E(\Omega_0) - \tau E'(\Omega_0; v), \tag{11.38}$$
$$\tau \mapsto E(\Omega_\tau) - E(\Omega_0) - \tau E'(\Omega_0; v) - \frac{1}{2}\tau^2 E''(\Omega_0; v, v). \tag{11.39}$$

We can observe on the right picture of Figure 11.15 that both the first-order and second-order approximation look to converge superlinearly to zero with respect to τ. The superlinear convergence of the first-order approximation could

be explained by the fact that the curvature in the given direction is rather small, as we can see in Figure 11.14. Finally we can say that the linear interpolation method for computing boundary integrals is both accurate and consistent.

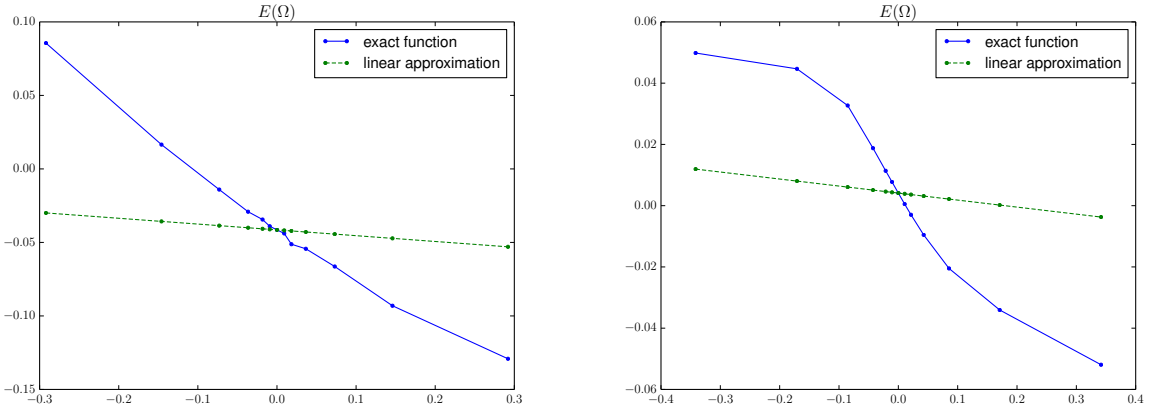


Figure 11.13: Consistency of the derivative with $\delta_{1,\varepsilon}$ (left) and $\delta_{2,\varepsilon}$ (right) for $h = \frac{2.3}{120}$ and $\varepsilon = \frac{h}{20}$.

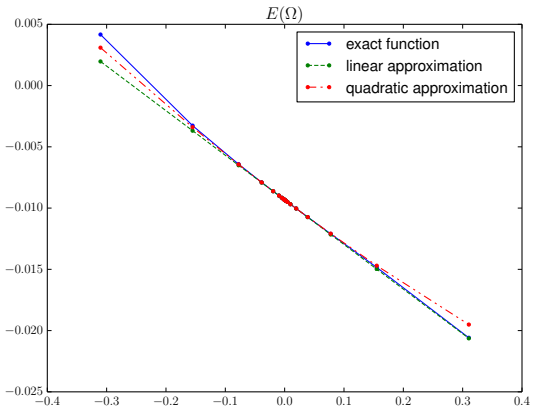


Figure 11.14: Consistency of the derivative with the linear interpolation method.

11.4 Shapes defined by level-sets

In this section we are concerned with the fact that every shape with piecewise linear boundary cannot be represented with the piecewise linear interpolation of a level-set function.

Definition 11.4.1. *For a mesh $\mathcal{T}_h$, we denote by $\mathcal{O}_h$ the set*

$$\mathcal{O}_h = \{\Omega_h \subset \mathbb{R}^d \,|\, \exists \phi \in \mathbb{P}_1(\mathcal{T}_h, \mathbb{R}),\ \Omega_h = \phi^{-1}\big(]-\infty, 0[\big)\}.$$

The set $\mathcal{O}_h$ is not the set of domains whose boundaries are piecewise linear. For example, let us consider a triangular mesh. Figure 11.16 shows different configurations of a domain with piecewise linear boundary intersecting two triangles. Only the first two (a and b) are domains of $\mathcal{O}_h$. Indeed, let us assume there exists $\phi \in \mathbb{P}_1$ such that in the configuration c, $\Omega = \phi^{-1}\big(]-\infty, 0[\big)$. We observe that the boundary in the triangle ABD is parallel to the line AB. Therefore, $\phi(A) = \phi(B)$. As a result, the boundary of Ω in the triangle ABC is necessary parallel to AB (as an application of Thales theorem).

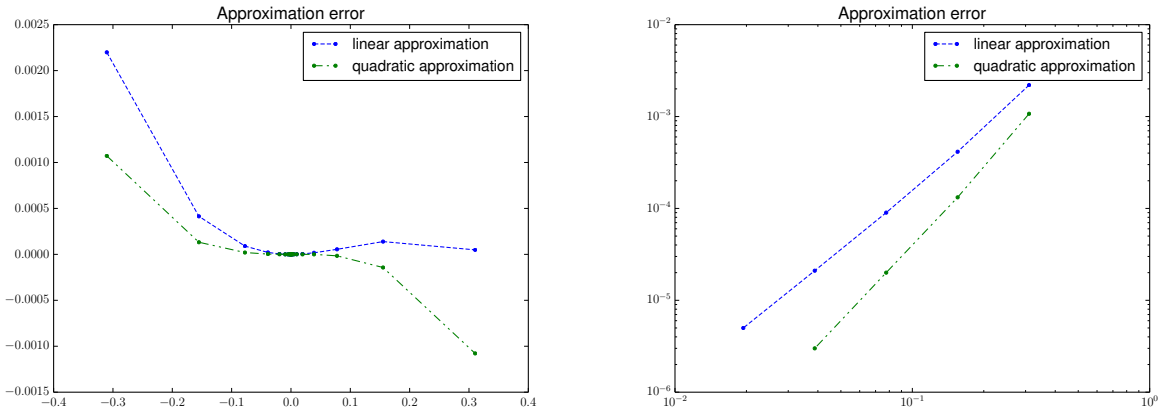
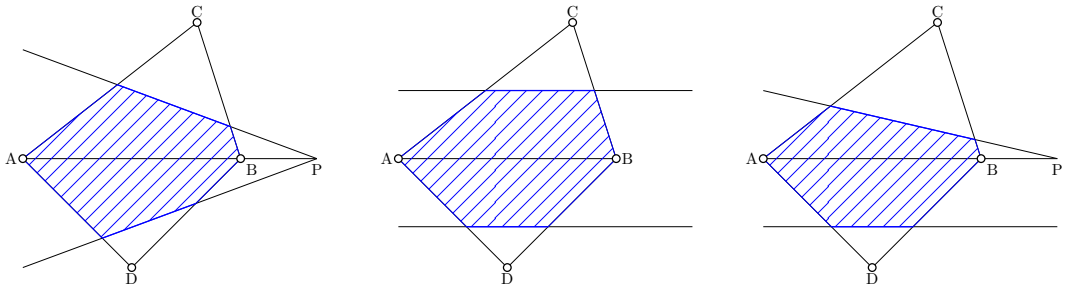


Figure 11.15: Plot of the curves (11.38) and (11.39) with respect to the time τ in linear (left) and log (right) scales.



(a) (b) (c)

Figure 11.16: Different configurations.

Chapter 12

Numerical examples

Contents

We have introduced a new framework for shape differentiation, and shown how it can be used for the definition of a Newton-like methods. It was also seen that the structure of the second-order shape derivative along normal trajectories gives indication on how we can compute extensions of velocity fields defined on a boundary in the perspective of shape optimization. In this chapter we are concerned with the comparison of the newly defined Newton method (Algorithm 10.3.1) with an adapted gradient method (Algorithm 10.3.3). We also want to compare the extension methods (namely Method 10.2.1, and Method 10.2.2) induced by the structure of the second-order shape derivative with a usual method (Method 10.2.3) for defining velocity fields on the whole space. This is the subject of the first two sections where we consider first the case of an example without partial differential equation and then some mechanical examples.

Finally we are also interested with the question of approximating the second-order shape derivative instead of computing it exactly. Therefore we compare the different approximation methods of Section 9.2 to the gradient-like method (Algorithm 10.3.3).

12.1 Comparison of algorithms and extensions on PDE-free examples

12.1.1 Setting of the problem

Let $f \in C^2(\mathbb{R}^d;\mathbb{R})$. We consider the minimization problem introduced in (11.1.5)

$$\inf_{\Omega\in\mathcal{O}_2} E(\Omega), \tag{12.1}$$

where

$$E(\Omega) = \int_\Omega f(x)\,dx$$

is the function to be minimized. We recall from Proposition 3.3.5 that the optimal shape is given by

$$\Omega^\star = \{x \in \mathbb{R}^d \mid f(x) < 0\}. \tag{12.2}$$

Since f is smooth, the objective function E is twice shape-differentiable at any $\Omega \in \mathcal{O}_2$. The shape derivatives in directions $\theta, \xi \in C^{2,\infty}(\partial\Omega; \mathbb{R}^2)$ in the sense of Theorem 3.2.15 are given by the linear and bilinear maps

$$l_1(\theta) = \int_{\partial\Omega} (\theta \cdot \mathbf{n})\, f,$$
$$l_2(\theta, \xi) = \int_{\partial\Omega} (\theta \cdot \mathbf{n})\, (\xi \cdot \mathbf{n}) \left(\mathcal{H} f + \partial_{\mathbf{n}} f \right),$$

where $\mathcal{H}(x)$ is the curvature of $\partial\Omega$ at $x \in \partial\Omega$.

Remark 12.1.1. *In this example it is also important to note that the computation of the second shape derivative is not very expensive. We only need to compute $\mathcal{H}f + \partial_{\mathbf{n}} f$ on the boundary. In this very specific case, the computation time of the descent directions are similar for the Newton and gradient algorithms.*

In the following examples, we will always take the initial shape given by the set of negative values of

$$\phi(x, y) = ((x - a_1)^2 + y^2)((x + a_1)^2 + y^2) - b_1^4,$$

with

$$a_1 = 0.5,\ b_1 = 0.51.$$

This shape is plot in Figure 12.1

Figure 12.1: Initial shape for the PDE-free example.

We now discuss some implementation issues. We use a single regular square mesh $\mathcal{Q}_h$ for the Hamilton-Jacobi equation and the evaluation of the objective function. The Hamilton-Jacobi equation is coupled with the transport equation for the redistanciation of the level-set. Both equations are solved with a second-order finite difference scheme. In these PDE-free examples, choosing the transport equation or the fast-marching method for the redistanciation of the level-set does not have much impact on the consistency (see Section 12.2.3). The objective function is computed with the linear interpolation approach for computing bulk and boundary integrals (see Section 11.2 for details). These algorithms are implemented in Scilab [34] with some routines (concerning the evaluation of the objective function and their gradients) in C.

We use the gradient algorithm 10.3.3 or the Newton algorithm 10.3.1, and the corresponding quadratic problems (10.27), (10.28) are solved by using Ipopt [128]. We take the domain $\mathcal{D}$ as $\mathcal{D} = [-1.15, 1.15]$, and mesh it with a mesh of size 120×120. This consists in taking the mesh parameter h as $h = \frac{2.3}{120}$. The numerical parameters ε, η for both algorithms 10.3.3 and 10.3.1 are set to $\varepsilon = \eta = 10^{-6}$. In this PDE-free example the two stopping criteria (with ε or η) can stop the algorithm. The descent direction is a priori defined only on $\partial\Omega$, and we need to extend it to the entire domain $\mathcal{D}$ in order to solve the Hamilton-Jacobi equation (10.25). We compare the three extension methods proposed in Section 10.2.

The computation of the matrix discretizing the Hessian operator, and more precisely the bilinear form l_2, is somehow tricky. When computing derivatives, the shape Ω is assumed to be smooth. In numerical practice, the optimal structures may exhibit corners, i.e., are not smooth. In such a case the mean curvature $\mathcal{H} = \mathrm{div}_\Gamma\, \mathbf{n}$ is not well defined. To mitigate this effect, we truncate the value of the discretized version of $\mathcal{H}$ by a maximum (absolute) value of the order of $1/h$ where h is the mesh size.

12.1.2 Cassini curve

First we consider the case when the objective function E is defined by the following function f

$$\begin{aligned} f(x,y) &= (x^2+y^2)^5 - 2a_0^5 \mathrm{Re}\big((x+iy)^5\big) + a_0^{10} - b_0^{10} \\ &= (x^2+y^2)^5 - 2a_0^5(x^5 - 10x^3y^2 + 5xy^4) + a_0^{10} - b_0^{10}, \end{aligned}$$

where

$$a_0 = 0.95,\ b_0 = 0.953.$$

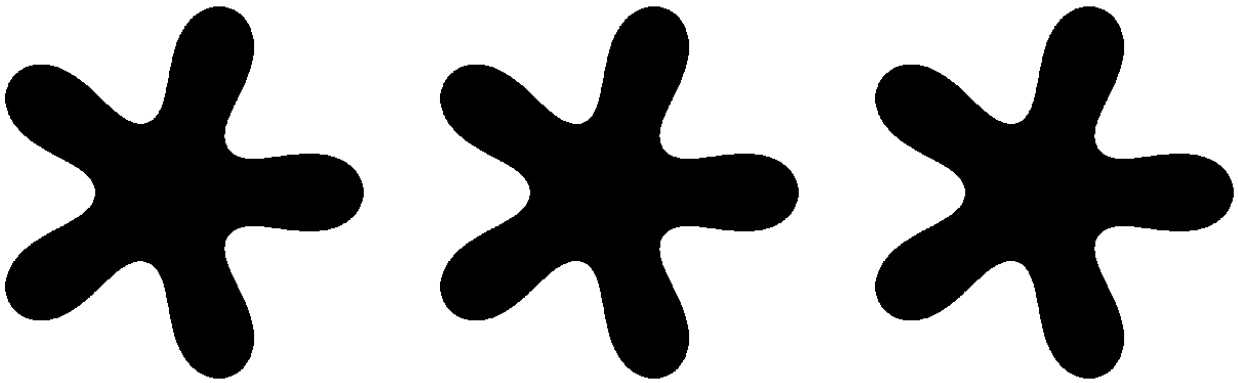


Figure 12.2: Final domains for the gradient algorithm for the problem of Section 12.1.2 with, from left to right, extension Method 10.2.3, Method 10.2.1 and Method 10.2.2.

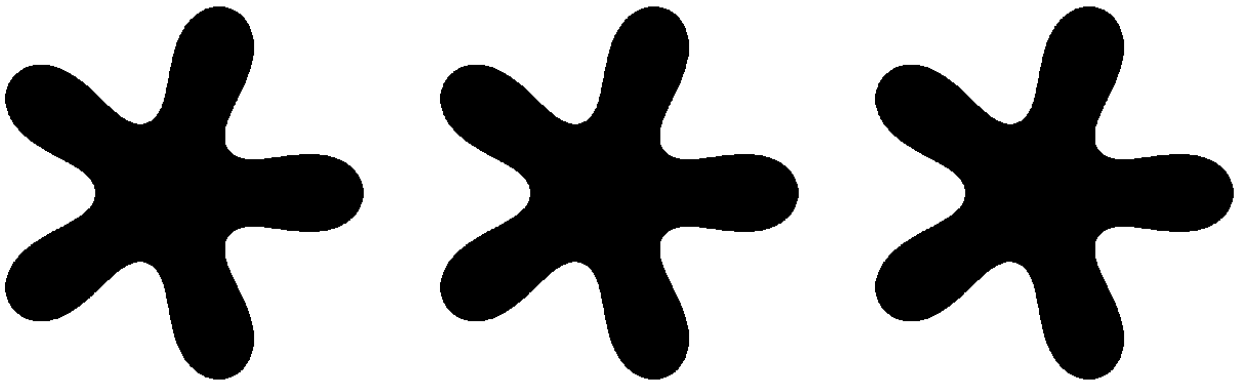


Figure 12.3: Final domains for the Newton algorithm for the problem of Section 12.1.2 with, from left to right, extension Method 10.2.3, Method 10.2.1, and Method 10.2.2.

The final shapes are plotted in Figure 12.2 and Figure 12.3, and the final value of the Lagrangian and the number of iterations needed to get convergence are transcripted in Table 12.1. We denote Ω_∞ the final shape for each method, and $\Omega^\times$ the best of all six final shapes (which are very similar) obtained by the various numerical methods. The shape $\Omega^\times$ is very close to $\Omega^\star$ (defined by (12.2)). In this example, according to Table 12.1, the shape $\Omega^\times$ is obtained by the gradient method with extension Method 10.2.3. We keep this notation in the sequel. Due to the line search, each iteration may need several evaluations of the objective function. Therefore, we also give the number of evaluations needed to get convergence. In order to have an idea on how the first-order optimality condition is satisfied, we additionally compute

$$\frac{E'\left(\Omega_\infty; v^\infty\right)}{E'\left(\Omega_0; v^0\right)} = \frac{\displaystyle\int_{\partial\Omega_\infty} f v^\infty}{\displaystyle\int_{\partial\Omega_0} f v^0}, \tag{12.3}$$

where v is the descent direction taken by the algorithm. This quantity is referred to as `Gradient ratio` in Table 12.1. In Figure 12.4 and Figure 12.5 we plot respectively the convergence of $E(\Omega) - E(\Omega^\times)$ and $E(\Omega) - E(\Omega_\infty)$.

We can see that the Newton method, whatever the choice of extension, converges always in fewer iterations (and evaluations) than the gradient method. The gain is also in terms of computation time, since here the computation time of a descent direction is similar for the gradient and Newton algorithms (see Remark 12.1.1). Comparing the convergence curves, we can also conclude that both the second-order derivative and the choice of extension improve the gradient method with extension Method 10.2.3. We can also look at the gradient ratio defined by (12.3) in Table 12.1, and see that for all methods, this ratio tends to zero. This means that the first-order optimality condition is not far from being reached.

		Gradient	Newton
Extension 10.2.3	Lagrangian	-0.28092703	-0.28092666
	Iterations	99	31
	Evaluations	458	69
	Gradient ratio	4.298×10^{-6}	4.432×10^{-6}
Extension 10.2.1	Lagrangian	-0.28092641	-0.28092682
	Iterations	70	10
	Evaluations	222	33
	Gradient ratio	2.275×10^{-5}	4.125×10^{-6}
Extension 10.2.2	Lagrangian	-0.28092667	-0.28092701
	Iterations	76	11
	Evaluations	390	31
	Gradient ratio	1.580×10^{-5}	1.977×10^{-6}

Table 12.1: Performance of each algorithm for the problem of Section 12.1.2.

Finally we are also concerned with the rate of convergence. Theoretically, the Newton method should be second-order (see Theorem 1.2.4). To that end, we compare the evolution of the error to the optimal shape from one iteration to the other. More precisely, for the Newton method we expect to observe that for $p \in \mathbb{N}$

$$|E(\Omega_{p+1}) - E(\Omega_\infty)| \leq C|E(\Omega_p) - E(\Omega_\infty)|^\alpha, \tag{12.4}$$

with $\alpha = 2$, which would lead to

$$\log\Big(|E(\Omega_{p+1}) - E(\Omega_\infty)|\Big) \leq \log(C) + \alpha \log\Big(|E(\Omega_p) - E(\Omega_\infty)|\Big).$$

To that end, we plot on Figure 12.6 the evolution of $\log\Big(|E(\Omega_{p+1}) - E(\Omega_\infty)|\Big)$ with respect to $\log\Big(|E(\Omega_p) - E(\Omega_\infty)|\Big)$. We also plot the linear regressions (thanks to a least square resolution) of this curve, and register the slope α of the linear regression in Table 12.2. We do it also for the gradient method, and all extension methods.

Remark 12.1.2. *According to Theorem 1.2.4 we should observe the second-order convergence rate on the error between Ω_p and $\Omega^\star$. However, since there is no natural norm for evaluating the difference between to shapes, we focus on the error on the objective function. When the second-order derivative of E is positive definite at the optimal shape, we can prove that the convergence rate on the objective (12.4) is the same as the convergence rate on the data Ω (such as in Theorem 1.2.4).*

	Gradient	Newton
Extension 10.2.3	0.97	1.60
Extension 10.2.1	0.86	1.97
Extension 10.2.2	0.91	1.43

Table 12.2: Slopes of the linear regressions of Figure 12.6.

We can observe on Figure 12.6 that the convergence rate for the Newton method is higher than the rate for the gradient method. Looking at Table 12.2, we can see that the slope of the linear regression is about 1 for the gradient method, and about 1.5 or 2 for the Newton method. This confirm that the Newton method for this example is at least super linear and that the gradient method has only a first-order convergence rate.

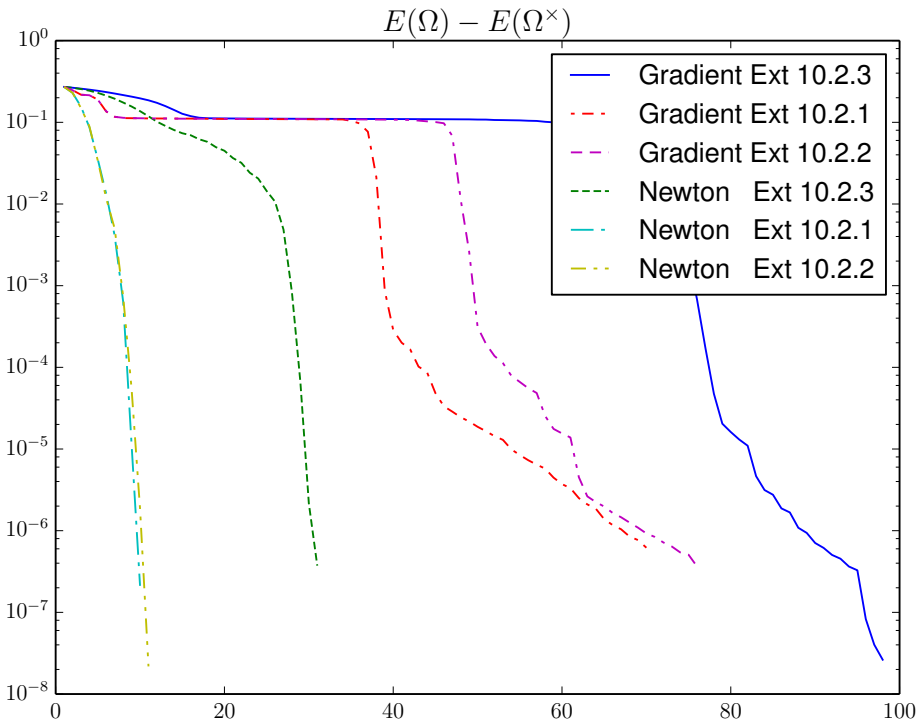


Figure 12.4: Convergence of $E(\Omega) - E(\Omega^\times)$ for the problem of Section 12.1.2.

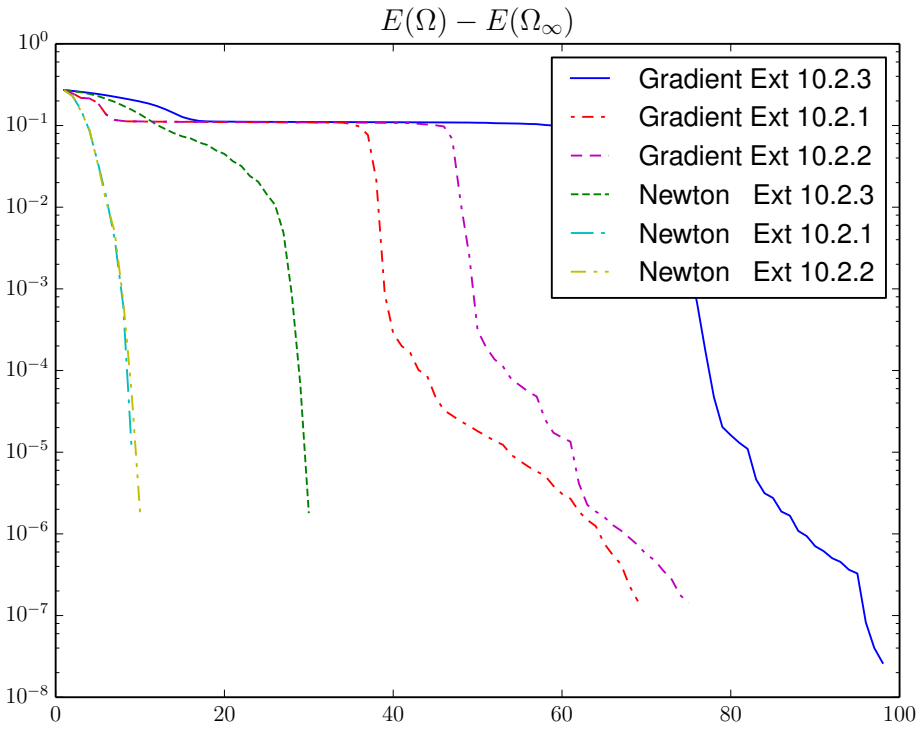


Figure 12.5: Convergence of $E(\Omega) - E(\Omega_\infty)$ for the problem of Section 12.1.2.

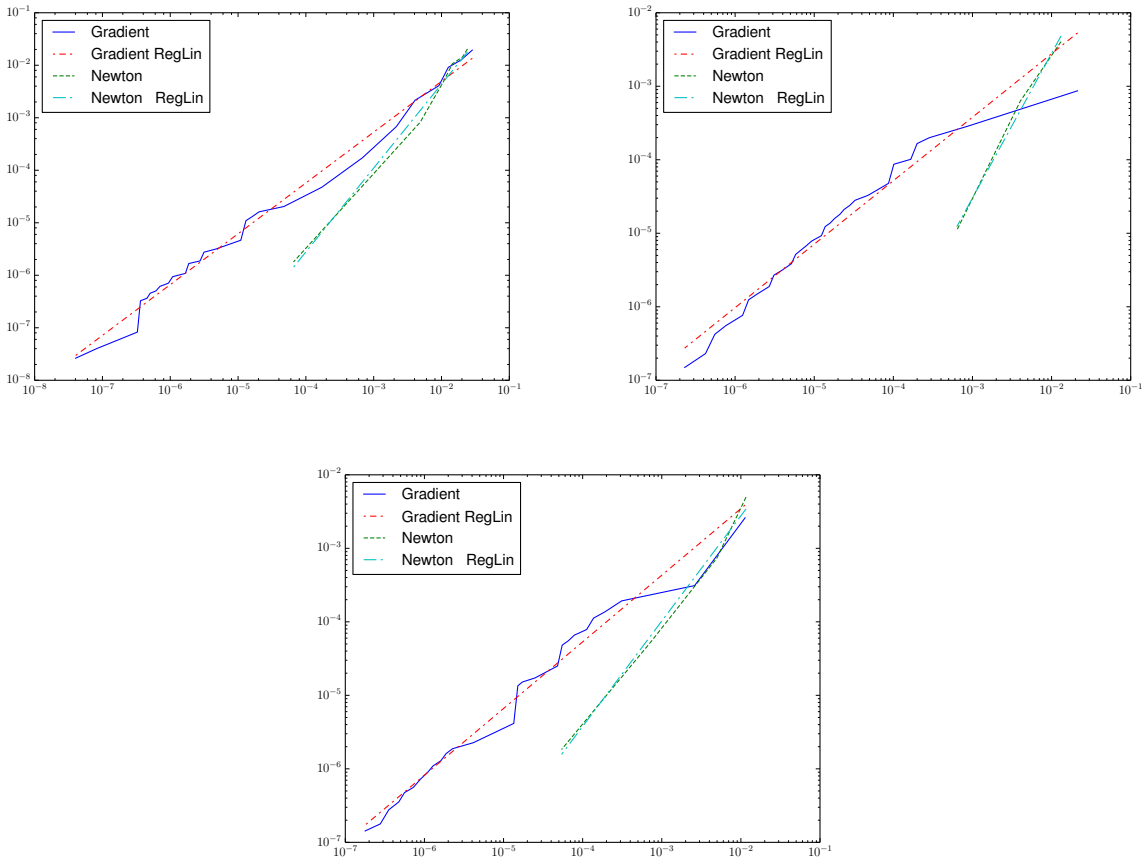


Figure 12.6: Linear regressions of the error with the best shape for the problem of Section 12.1.2. The extension is made with Method 10.2.3 (top-left), Method 10.2.1 (top-right), and Method 10.2.2 (bottom). Evolution of $E(\Omega_{n+1}) - E(\Omega_\infty)$ with respect to $E(\Omega_n) - E(\Omega_\infty)$.

12.1.3 Cardioid

Now we consider the case where the objective function E is defined by the following function f that describes a cardioid :

$$f(x, y) = (x^2 + y^2 - ax)^2 - a^2(x^2 + y^2),$$

with $a = 0.6$.

We plot all final shapes in Figure 12.7 and Figure 12.8. We can observe that they all look very similar. With the Newton method though, the cusp seems to be better captured. Thus, we can observe in Table 12.3 that the final value for the objective function E is always better with the Newton method. In this example, the shape $\Omega^\times$ is obtained by the Newton method with extension Method 10.2.2. The plots of the convergence curves $E(\Omega) - E(\Omega^\times)$ and $E(\Omega) - E(\Omega_\infty)$ respectively in Figure 12.9 and Figure 12.10 also indicate that the Newton method allows significant gains in terms of iterations compared to the gradient method. We also see that the use of Method 10.2.1 or Method 10.2.2 instead of Method 10.2.3 improves the convergence for both the Newton and gradient methods.

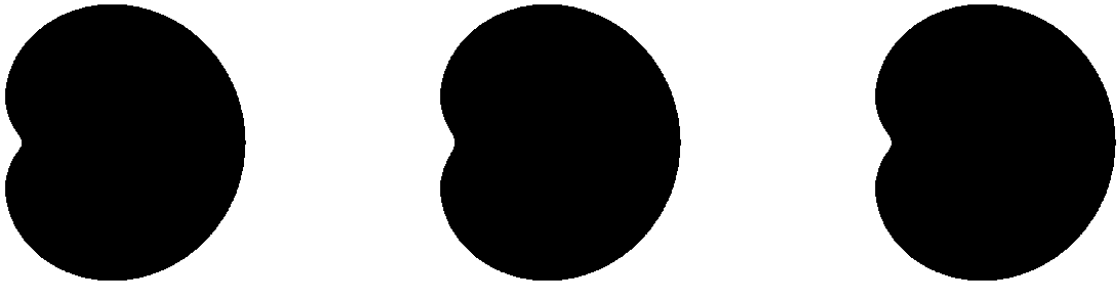


Figure 12.7: Final domains for the gradient algorithm for the problem of Section 12.1.3 with, from left to right, extension Method 10.2.3, Method 10.2.1 and Method 10.2.2.

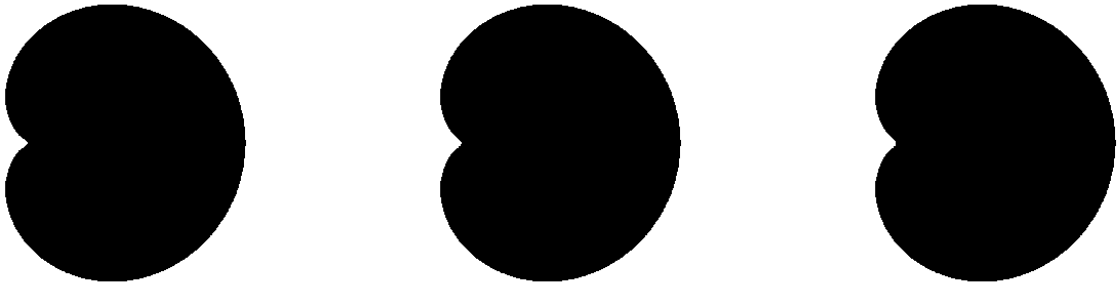


Figure 12.8: Final domains for the Newton algorithm for the problem of Section 12.1.3 with, from left to right, extension Method 10.2.3, Method 10.2.1, and Method 10.2.2.

For the rate of convergence, we can see on Figure 12.11 an in Table 12.4 that the gradient method converges linearly. However, the rate of convergence for the Newton method is not as close to 2 as in the test case presented in Section 12.1.2. The convergence of the Newton method remains superlinear, but is not quadratic. This could be explained by the singularity of the optimal shape due to the cusp. Therefore, the second-order shape derivative - that depends on the curvature - is not well-conditioned and may not be sufficiently regular for preserving the quadratic convergence rate.

		Gradient	Newton
Extension 10.2.3	Lagrangian	-0.16016584	-0.16016598
	Iterations	53	25
	Evaluations	298	83
	Gradient ratio	1.311×10^{-7}	8.690×10^{-9}
Extension 10.2.1	Lagrangian	-0.16016559	-0.16016598
	Iterations	49	9
	Evaluations	62	32
	Gradient ratio	6.124×10^{-7}	4.554×10^{-8}
Extension 10.2.2	Lagrangian	-0.16016581	-0.16016598
	Iterations	39	12
	Evaluations	257	54
	Gradient ratio	2.886×10^{-7}	2.152×10^{-8}

Table 12.3: Performance of each algorithm for the problem of Section 12.1.3.

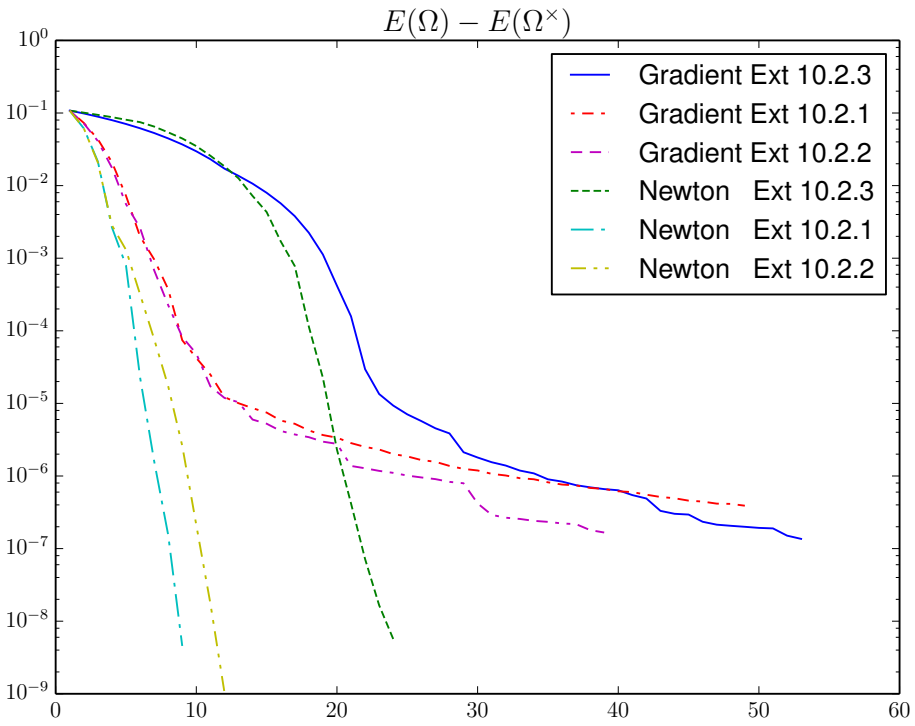


Figure 12.9: Convergence of $E(\Omega) - E(\Omega^\times)$ for the problem of Section 12.1.3.

	Gradient	Newton
Extension 10.2.3	0.98	1.08
Extension 10.2.1	0.92	1.06
Extension 10.2.2	0.93	1.11

Table 12.4: Slopes of the linear regressions of Figure 12.11.

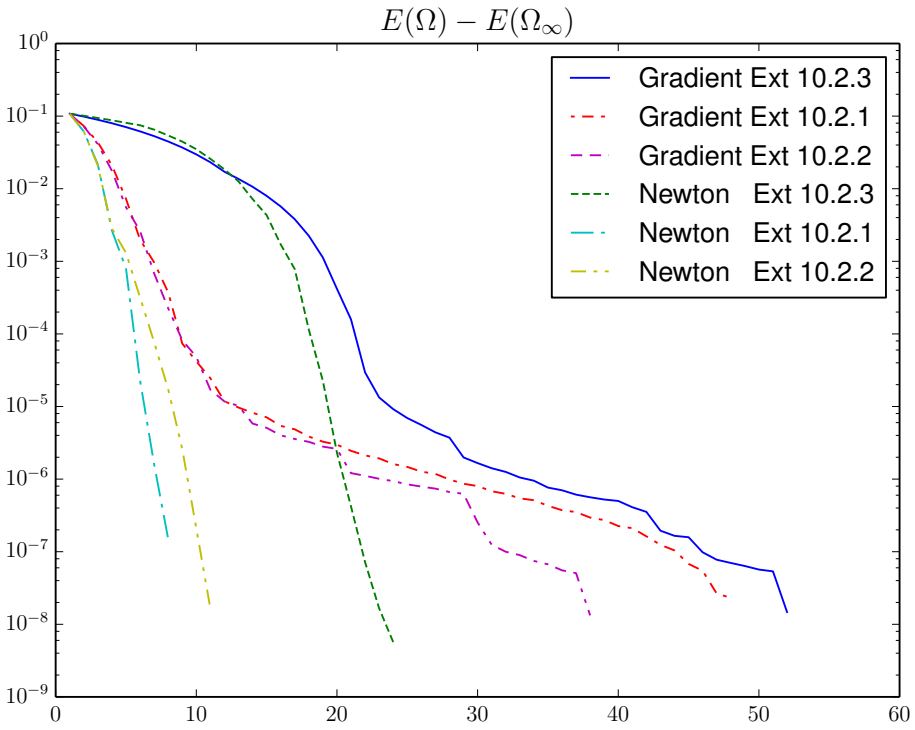


Figure 12.10: Convergence of $E(\Omega) - E(\Omega_\infty)$ for the problem of Section 12.1.3.

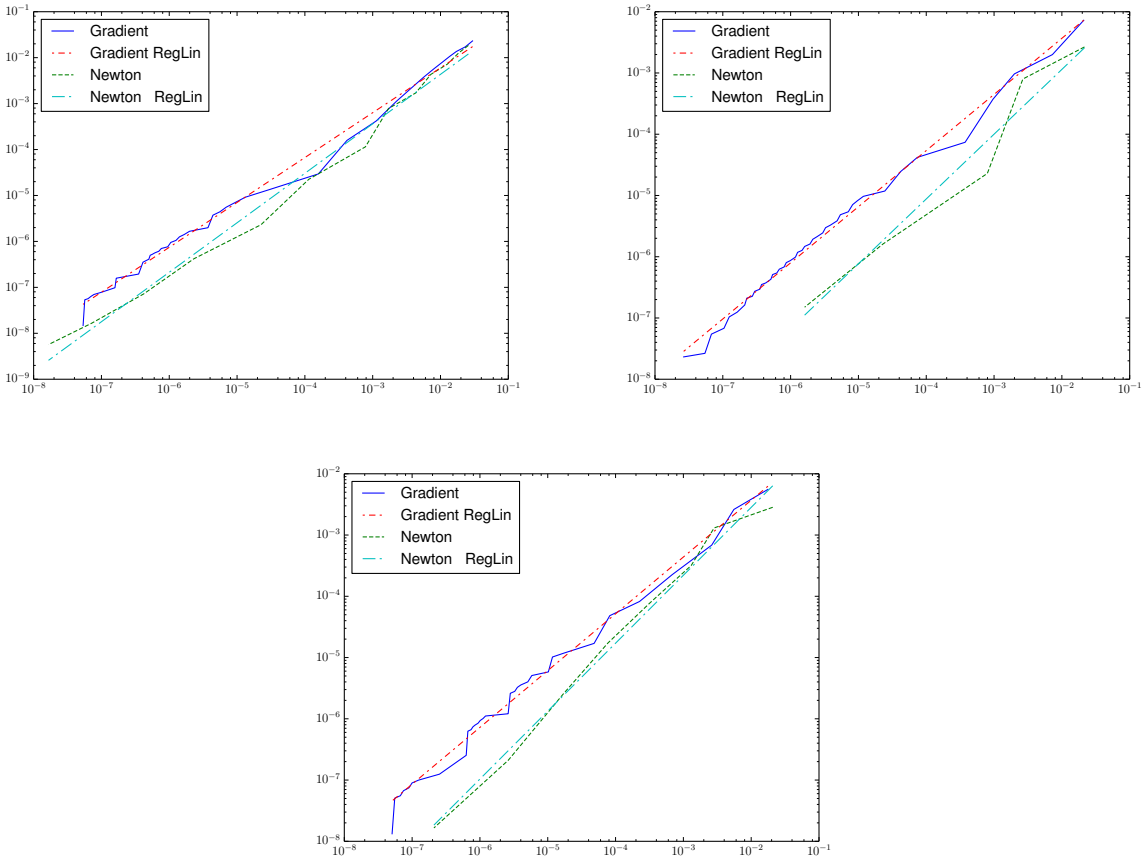


Figure 12.11: Linear regressions of the error with the best shape for the problem of Section 12.1.3. The extension is made with Method 10.2.3 (top-left), Method 10.2.1 (top-right), and Method 10.2.2 (bottom). Evolution of $E(\Omega_{n+1}) - E(\Omega_\infty)$ with respect to $E(\Omega_n) - E(\Omega_\infty)$.

12.1.4 Rotated Square

Here, the objective function E is defined by the following function f that describes a square :

$$f(x,y) = \max\Big(x^2 - a^2; y^2 - a^2\Big),$$

with $a = 0.6$. We also change the coordinate system by rotating it by an angle of $\frac{\pi}{4}$. Therefore, the orientation of the mesh changes with respect to the shape. In particular, the singularity of the boundary is not on a vertex of the mesh, and the directions of the corner are no longer aligned with an edge of the mesh. Therefore, the square will not be exactly captured by the level-set representation.

We plot all final shapes in Figure 12.12 and Figure 12.13. The performance of each algorithm is given in Table 12.5, and the convergence curves $E(\Omega) - E(\Omega^\times)$ and $E(\Omega) - E(\Omega_\infty)$ are respectively plotted in Figure 12.14 and Figure 12.15 In this example, we can first observe that the gradient method and the Newton method converge similarly. Whatever the choice of extension, they require almost the same number of iterations to converge. We can also see that the choice of extension by Method 10.2.1 or Method 10.2.2 improves the convergence - compared when the extension is given by Method 10.2.3 - for both the gradient or the Newton method.

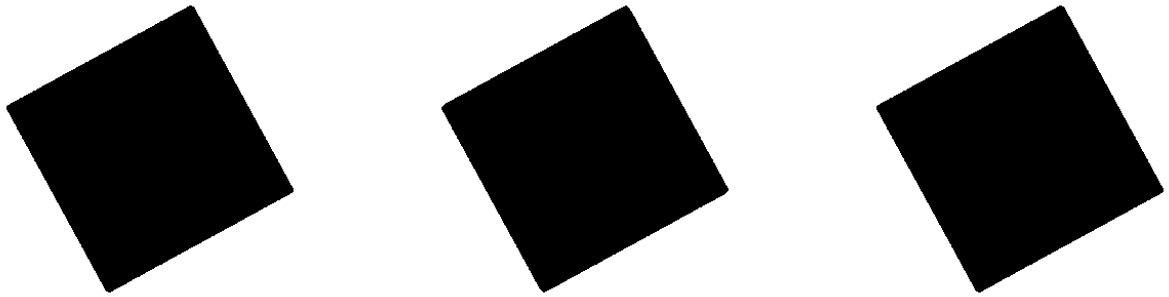


Figure 12.12: Final domains for the gradient algorithm for the problem of Section 12.1.4 with, from left to right, extension Method 10.2.3, Method 10.2.1 and Method 10.2.2.

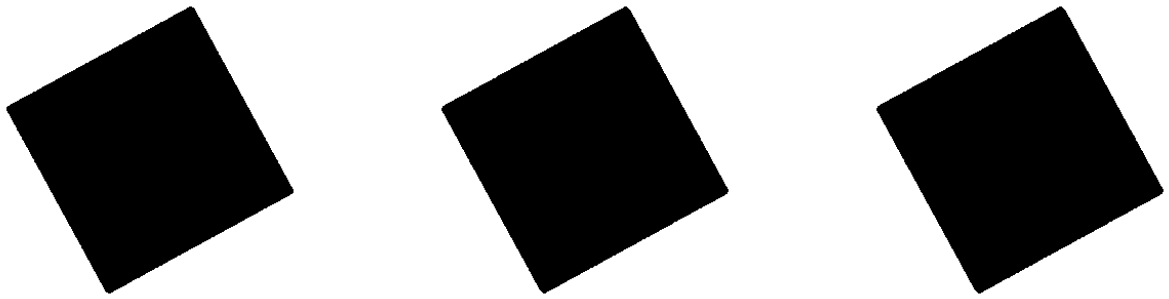


Figure 12.13: Final domains for the Newton algorithm for the problem of Section 12.1.4 with, from left to right, extension Method 10.2.3, Method 10.2.1, and Method 10.2.2.

These first observations are not in favor of the Newton method for this example. It indeed appears that there is no gain - for this particular case - in computing the second-order shape derivatives. Looking at the convergence rates in Figure 12.16 and Table 12.6 also confirms this impression. Even if for the Newton method the slope (see Table 12.6) is - for two extension methods - close to 1.2, it can be seen as a numerical artifact. The slope is indeed computed by a least square minimization, and looking at the evolution of the real error (in the Newton case in Figure 12.16) we can see that it is very erratic, and that it is close to the gradient error. We conjecture here that this is the singularity of the corner that cause the loss of the second-order rate of convergence.

		Gradient	Newton
Extension 10.2.3	Lagrangian	-0.25902578	-0.25902570
	Iterations	26	18
	Evaluations	89	60
	Gradient ratio	7.899×10^{-7}	4.934×10^{-7}
Extension 10.2.1	Lagrangian	-0.25902529	-0.25902538
	Iterations	10	9
	Evaluations	39	36
	Gradient ratio	6.353×10^{-6}	2.028×10^{-6}
Extension 10.2.2	Lagrangian	-0.25902577	-0.25902561
	Iterations	9	10
	Evaluations	50	53
	Gradient ratio	1.042×10^{-6}	1.026×10^{-6}

Table 12.5: Performance of each algorithm for the problem of Section 12.1.4.

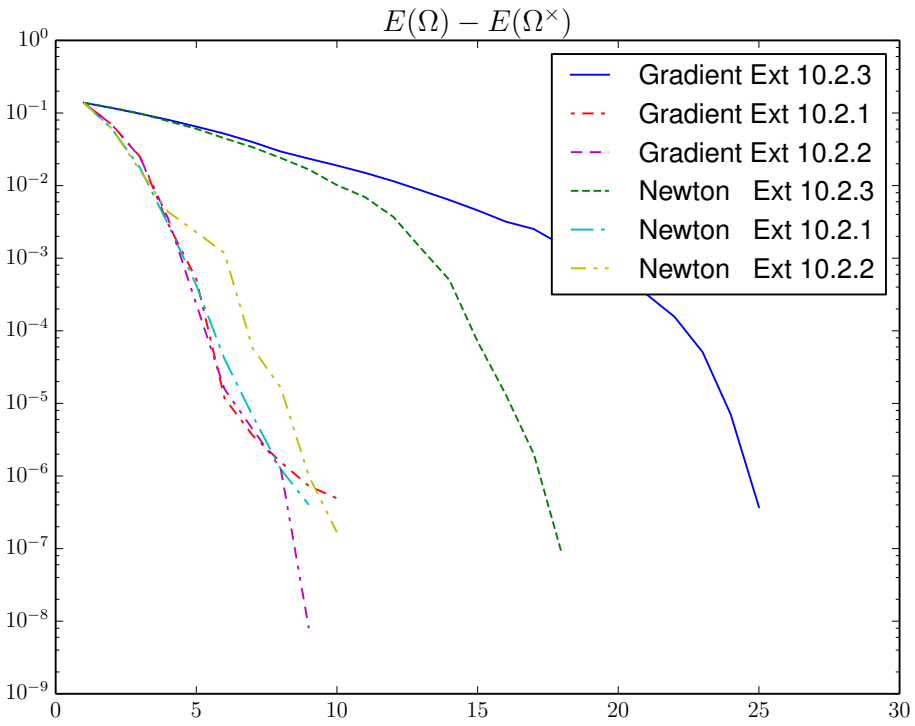


Figure 12.14: Convergence of $E(\Omega) - E(\Omega^\times)$ for the problem of Section 12.1.4.

	Gradient	Newton
Extension 10.2.3	1.28	1.23
Extension 10.2.1	0.88	1.02
Extension 10.2.2	0.87	1.21

Table 12.6: Slopes of the linear regressions of Figure 12.16.

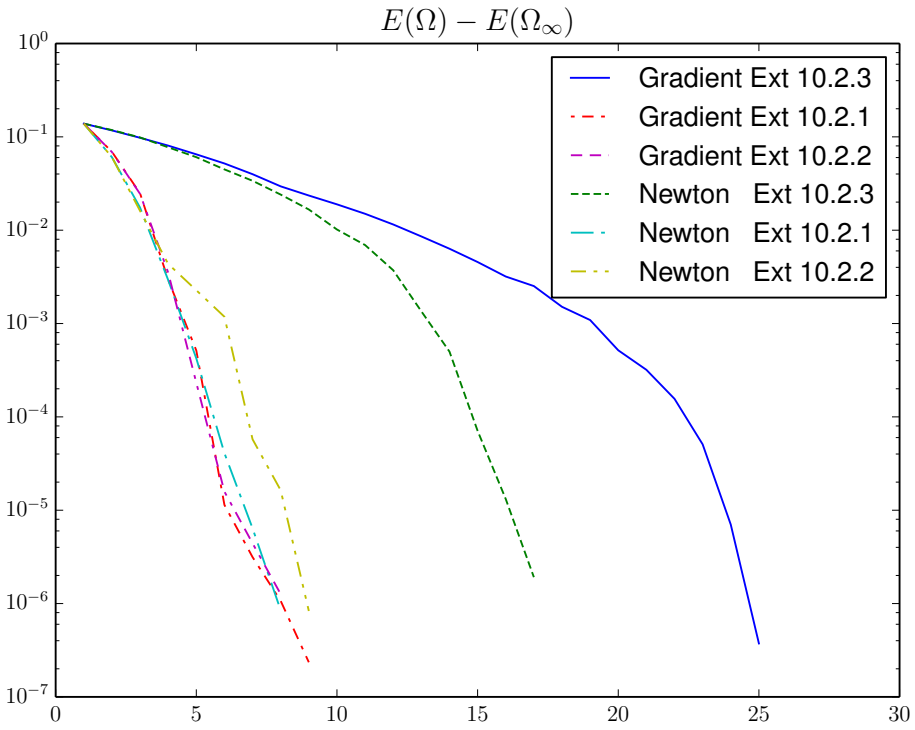


Figure 12.15: Convergence of $E(\Omega) - E(\Omega_\infty)$ for the problem of Section 12.1.4.

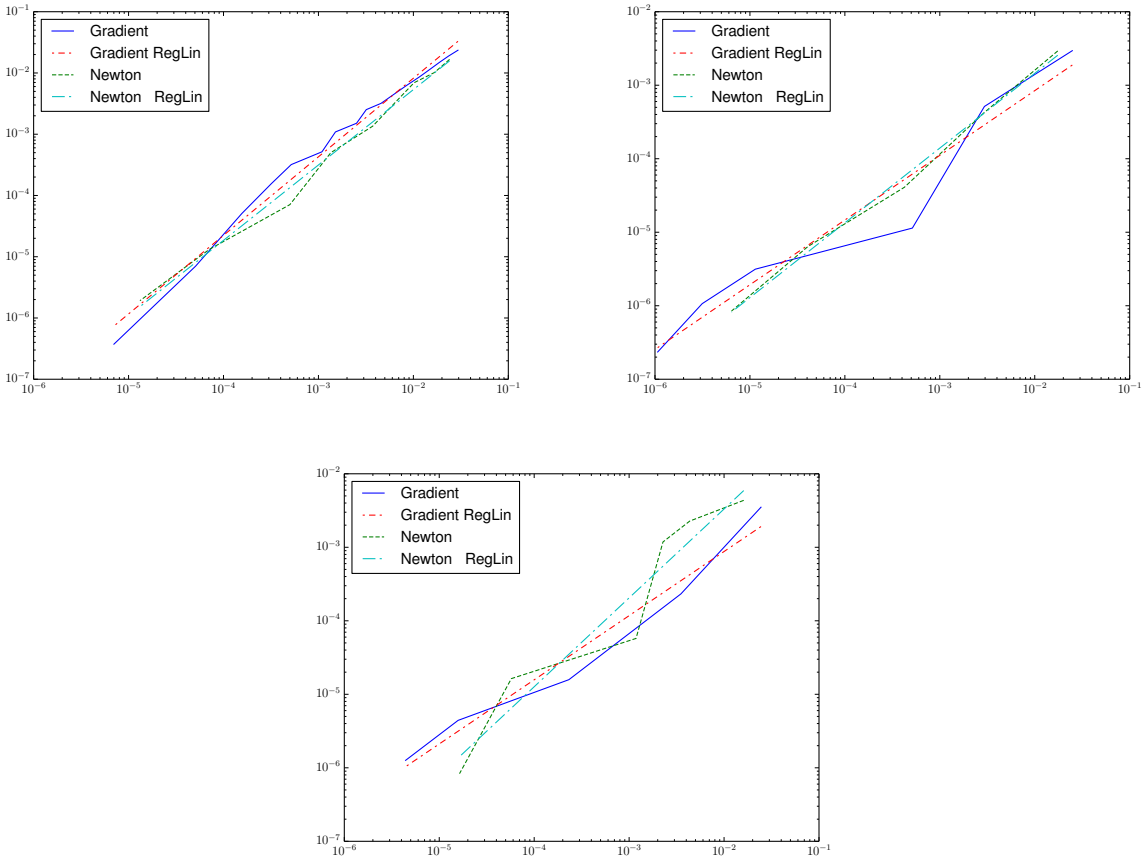


Figure 12.16: Linear regressions of the error with the best shape for the problem of Section 12.1.4. The extension is made with Method 10.2.3 (top-left), Method 10.2.1 (top-right), and Method 10.2.2 (bottom). Evolution of $E(\Omega_{n+1}) - E(\Omega_\infty)$ with respect to $E(\Omega_n) - E(\Omega_\infty)$.

12.2 Comparison of algorithms and extensions on mechanical examples

12.2.1 Setting of the problem

We consider the case of the *elastic model* described in Section 2.1.5. Let $\Omega \in \mathbb{R}^d$ be a smooth bounded open set, filled with a homogeneous isotropic material, with Hooke's tensor A, and Lamé's coefficients λ, μ. For a symmetric tensor ζ, we have $A\zeta = 2\mu\zeta + \lambda \mathrm{Tr}(\zeta)\mathrm{Id}$. We consider a partition of the boundary of Ω as

$$\partial\Omega = \Gamma \cup \Gamma_N \cup \Gamma_D\,,$$

where Γ_N and Γ_D remain unchanged, whereas Γ is the part to be optimized. We introduce a working domain $\mathcal{D}$ containing all admissible shapes. For $g \in H^1(\mathcal{D};\mathbb{R}^d)$ such that $g = 0$ on Γ, we consider the following boundary problem, with unknown u :

$$\left\{\begin{array}{rcll} -\mathrm{div}\big(A\varepsilon(u)\big) & = & 0 & \text{in } \Omega, \\ u & = & 0 & \text{on } \Gamma_D, \\ A\varepsilon(u)\mathbf{n} & = & g & \text{on } \Gamma_N, \\ A\varepsilon(u)\mathbf{n} & = & 0 & \text{on } \Gamma, \end{array}\right. \tag{12.5}$$

with the strain tensor $\varepsilon(u) = \dfrac{1}{2}\Big(\nabla u + (\nabla u)^T\Big)$. We introduce the Hilbert space

$$\mathcal{V} = \left\{u \in H^1(\Omega;\mathbb{R}^d) \mid u = 0 \text{ on } \Gamma_D\right\}.$$

The variational formulation of (12.5) is

$$\left\{\begin{array}{l} \text{find } u \in \mathcal{V} \text{ such that} \\ \forall \varphi \in \mathcal{V},\ \displaystyle\int_\Omega A\varepsilon(u) : \varepsilon(\varphi) = \int_{\Gamma_N} g \cdot \varphi. \end{array}\right. \tag{12.6}$$

Our aim is to minimize an objective function subject to a volume constraint. We focus on two criteria, the compliance J_1 defined by

$$J_1(\Omega) = \int_\Omega A\varepsilon(u) : \varepsilon(u) = \int_{\Gamma_N} g \cdot u,$$

and the least square displacement criterion error defined by

$$J_2(\Omega) = \sqrt{\int_\Omega k|u - u_0|^2},$$

where $k \in L^\infty(\mathcal{D})$ and $u_0 \in L^2\left(\Omega;\mathbb{R}^d\right)$. In the following examples for J_2, we shall take $u_0 = 0$ and k will be the characteristic function of a thin neighborhood of Γ_N. The volume is defined as

$$V(\Omega) = \int_\Omega 1\, dx.$$

For simplicity, in the sequel, we shall fix the Lagrange multiplier $\Lambda > 0$ for the volume constraint and minimize, without any constraint, the Lagrangian L defined by

$$L(\Omega) = J(\Omega) + \Lambda V(\Omega),$$

where J is either J_1 or J_2. Optimizing without constraints the Lagrangian L allows us to focus exclusively on the comparison between the gradient and the Newton algorithms.

12.2.2 Shape derivatives and implementation issues

When the shape Ω and the boundary load g are sufficiently smooth, the objective function and the volume are shape-differentiable [72], [15]. We recall without proofs these shape derivatives.

Proposition 12.2.1. *Assume that Ω is of class C^3. Then, the volume $V(\Omega)$ is twice shape-differentiable. For $\theta, \xi \in C^{2,\infty}\big(\mathbb{R}^d;\mathbb{R}^d\big)$, the maps l_1, l_2 (defined in Theorem 3.2.15) corresponding to the derivatives of $V(\Omega)$ are defined by*

$$l_1(\theta) = \int_{\partial\Omega} (\theta \cdot \boldsymbol{n}) \quad \text{and} \quad l_2(\theta,\xi) = \int_{\partial\Omega} \mathcal{H}\,(\theta \cdot \boldsymbol{n})\,(\xi \cdot \boldsymbol{n})\,,$$

where $\mathcal{H} = \mathrm{div}_\Gamma\, \boldsymbol{n}$ is the mean curvature of the boundary.

The formulas for first-order shape derivatives of the compliance and the least square criteria are classical results in shape optimization, and can for example be found in [15]. For the second-order shape derivatives, all calculations are made in Chapter 7.

Proposition 12.2.2. *Assume that Ω is of class C^3 and that Γ_D and Γ_N are fixed. Then, the compliance J_1 is twice shape-differentiable. For $\theta, \xi \in C^{2,\infty}\big(\mathbb{R}^d;\mathbb{R}^d\big)$, the maps l_1, l_2 corresponding to the derivatives of J_1 are defined by*

$$l_1(\theta) = -\int_\Gamma (\theta\cdot\boldsymbol{n})\, A\varepsilon(u) : \varepsilon(u),$$
$$l_2(\theta,\xi) = 2\int_\Omega A\varepsilon(u'_\theta) : \varepsilon(u'_\xi) - \int_\Gamma (\theta\cdot\boldsymbol{n})\,(\xi\cdot\boldsymbol{n})\left(\mathcal{H}A\varepsilon(u):\varepsilon(u) + \partial_{\boldsymbol{n}}\Big(A\varepsilon(u):\varepsilon(u)\Big)\right),$$

where $\mathcal{H}$ is the mean curvature of the boundary Γ and u'_θ is the solution of

$$\left\{\begin{array}{rcll} -\mathrm{div}\big(A\varepsilon(u'_\theta)\big) &=& 0 & \text{in } \Omega,\\ u'_\theta &=& 0 & \text{on } \Gamma_D,\\ A\varepsilon(u'_\theta)\boldsymbol{n} &=& 0 & \text{on } \Gamma_N,\\ A\varepsilon(u'_\theta)\boldsymbol{n} &=& \mathrm{div}_\Gamma\big((\theta\cdot\boldsymbol{n})\,A\varepsilon(u)\big) & \text{on } \Gamma. \end{array}\right. \tag{12.7}$$

For the least square criterion, we denote $\tilde{J}_2(\Omega) = \big(J_2(\Omega)\big)^2 = \displaystyle\int_\Omega k|u-u_0|^2$. In order to ease the reading, we give the derivatives of $\tilde{J}_2$ instead of J_2.

Proposition 12.2.3. *Assume that the domain Ω is of class C^3. The least square criterion $\tilde{J}_2$ is twice shape-differentiable. For $\theta, \xi \in C^{2,\infty}\big(\mathbb{R}^d;\mathbb{R}^d\big)$, the map l_1 corresponding to the first shape derivative of $\tilde{J}_2$ is defined by*

$$l_1(\theta) = \int_\Gamma (\theta\cdot\boldsymbol{n})\left(k|u-u_0|^2 + A\varepsilon(u):\varepsilon(p)\right),$$

where the adjoint state p is the solution to

$$\left\{\begin{array}{rcll} -\mathrm{div}\big(A\varepsilon(p)\big) &=& -2k(u-u_0) & \text{in } \Omega,\\ p &=& 0 & \text{on } \Gamma_D,\\ A\varepsilon(p)\boldsymbol{n} &=& 0 & \text{on } \Gamma_N,\\ A\varepsilon(p)\boldsymbol{n} &=& 0 & \text{on } \Gamma. \end{array}\right. \tag{12.8}$$

The load g, the target displacement u_0 as well as the shape Ω are assumed to be sufficiently regular so that $p \in H^2\left(\Omega;\mathbb{R}^d\right)$. The map l_2 corresponding to the second-order shape derivative of $\tilde{J}_2$ is defined by

$$\begin{aligned} l_2(\theta,\xi) = &\int_\Gamma (\xi\cdot\boldsymbol{n})\,A\varepsilon(u'_\theta):\varepsilon(p) + (\theta\cdot\boldsymbol{n})\,A\varepsilon(u'_\xi):\varepsilon(p) + 2\int_\Omega k\,u'_\theta\cdot u'_\xi\\ &+ 2\int_\Gamma k(u-u_0)\Big((\theta\cdot\boldsymbol{n})\,u'_\xi + (\xi\cdot\boldsymbol{n})\,u'_\theta\Big)\\ &+ \int_\Gamma (\theta\cdot\boldsymbol{n})\,(\xi\cdot\boldsymbol{n})\Big(\partial_{\boldsymbol{n}}\mathcal{J} + \mathcal{H}\mathcal{J}\Big), \end{aligned}$$

where $\mathcal{H}$ is the mean curvature o the boundary Γ, u'_θ is the smooth solution to (12.7) and

$$\mathcal{J} = k|u-u_0|^2 + A\varepsilon(u):\varepsilon(p)\,.$$

Let us now focus on some implementation issues. We use a single regular square mesh $\mathcal{Q}_h$ (the value of h is detailed in each example) for both the Hamilton-Jacobi equation and the linearized elasticity systems, like (12.5), (12.7), (12.8). The Hamilton-Jacobi equation is solved by a second-order finite difference scheme, and the level-set is redistanciated with the fast-marching procedure Section 4.3.2. The elasticity equations are solved by $\mathbb{Q}_1$ finite elements with an ersatz material approach for the void region (see Section 11.2 for details). These algorithms are implemented in Scilab [34] with some routines (concerning the evaluation of the objective function and their gradients) in C.

Like for the PDE-free examples, we use the gradient algorithm 10.3.3 or the Newton algorithm 10.3.1, and the corresponding quadratic problems (10.27), (10.28) are solved by using Ipopt [128]. Here we set the numerical parameters ε, η for both algorithms 10.3.3 and 10.3.1 to $\varepsilon = \eta = 10^{-5}$. Even if we can observe a decrease on the L^2-norm of the descent direction, it appears that - in the following examples - it is always the criterion on the descent step that stops the algorithms. The descent direction is still only defined on $\partial\Omega$ and needs to be extended for solving the Hamilton-Jacobi equation. We compare the three extension methods proposed in Section 10.2.

The most tricky part is the computation of the matrix discretizing the Hessian operator, and more precisely the bilinear form l_2. Note that the discretization of the linear form l_1 (or the first-order derivative) is standard [15]. A first remark concerns the assumed smoothness of the shape Ω for deriving the above formulas. Like explained in Section 12.1 or in Section 3.3.7, the optimal structures may not be smooth. Thus, the curvature $\mathcal{H} = \mathrm{div}_\Gamma\,\mathbf{n}$ is not well defined, and therefore its numerical value is truncated by a maximum (absolute) value of order $1/h$. A second remark deals with the computation of normal derivatives of $A\varepsilon(u):\varepsilon(u)$ or $\mathcal{J}$ in the formulas for l_2. When u and p are $\mathbb{Q}_1$, we need to interpolate their gradients as $\mathbb{Q}_1$ finite elements before computing these normal derivatives. The

third and most important comment is about the matrix arising from the discretization of l_2. It is a full matrix of size proportional to the number of nodes belonging to the mesh cells cut by the boundary $\partial\Omega$. Therefore its storage is a real issue, at least for 3-d problems, (see Section 12.4). A possible cure could be to store only elements which are above some threshold, in absolute value. Furthermore, the evaluation of the Hessian matrix requires the knowledge of the shape derivatives of the displacement, namely u'_θ and u'_ξ. For a given $\theta \in \mathbb{P}_1\big(\mathcal{T}_h;\mathbb{R}^2\big)$, computing u'_θ amounts to solving the linear system with the stiffness matrix and the right-hand side parametrized by θ in (12.7). In 2-d, the stiffness matrix is already factorized for the computation of u. There are as many different right-hand sides as the number of nodes in the vicinity of the boundary (this number is usually much smaller than the number of cells in the mesh). This computation of the Hessian is not too expensive if the stiffness matrix is factorized and stored, what can be easily done in 2-d. However, in 3-d, when iterative solvers are used, the computational cost may be problematic. For example in 2-d, and without any performance optimization (using Scilab), for a mesh of size 120×60 (as used in the next section) the computation of the Hessian requires about $5s$ of CPU time, whereas the computation of the gradient requires about $100ms$ of CPU time.

12.2.3 Consistency issue with the redistanciation

In the level-set method, the motion of the shape is tracked by the resolution of a Hamilton-Jacobi equation. For numerical reasons, it is convenient to take the signed distance function to $\partial\Omega$ as level-set function. This resolution of the Hamilton-Jacobi equation does not ensures that the level-set function stays a distance function. Therefore, the level-set is often redistanciated (see Section 4.3) to prevent it from being too steep or too flat.

However, it appears that the redistanciation process may have negative effects on the consistency of the derivatives. It is well known that the resolution of the transport equation (see Section 4.3.1) for the redistanciation of the level-set does not conserve exactly the boundary of the shape. This was not problematic for the previous PDE-free examples in Section 12.1. In the mechanical examples, we rather choose the fast-marching method (see Section 4.3.2) instead, and we will see that there are still numerical issues.

For example, we take the case of the compliance criterion on the following example of the arch with an initial shape with holes. As usual, with v given by the extension (by Method 10.2.3) of the solution to the gradient equation (10.28), we plot the following functions

$$\begin{aligned}
&\tau \mapsto J_1(\Omega_\tau),\\
&\tau \mapsto J_1(\Omega_0) + \tau {J_1}'(\Omega_0; v)\,,\\
&\tau \mapsto J_1(\Omega_0) + \tau {J_1}'(\Omega_0; v) + \frac{1}{2}\tau^2 {J_1}''(\Omega_0; v, v)\,.
\end{aligned}$$

In Figure 12.17, we compare the consistency of the derivatives depending on whether the transport equation is coupled to a redistanciation process or not. On the right picture of Figure 12.17 (when the level-set is redistanciated during the advection) we can see that there is a bump in the curve $\tau \mapsto J_1(\Omega_\tau)$ at $\tau = 0$. To the contrary, when there is no redistanciation, the derivatives are consistent. The redistanciation acts as if a constant error were added to this curve. In the right picture, the error added by the redistanciation is negative : the value $J_1(\Omega_0)$ is above the curve $\tau \mapsto J_1(\Omega_\tau)$. It is not problematic since the objective is to be minimized. Thus, any small positive time step τ lead to a decrease of the criterion J_1. However, the error may be positive like in Figure 12.18. Even if this last figure is just a cartoon illustrating the idea, such a case sometimes occurs. In that case, the shape Ω_0 is seen as a local minimum of the criterion J_1. Thus, the line-search may not find any time step to decrease the criterion. This may lead to the termination of the optimization process for a shape not optimal at all.

Thus, the fast-marching method also seems to modify the zero level-set. It appears that the initial computation of the distance function for nodes in the vicinity of the boundary is based on linear approximations. Thus, this could lead to slight modifications of the boundary. This question is further considered in Section 14.3.

The modification of the shape induced by the redistanciation process is also the reason why it could be interesting to consider instead the approach of Adalsteinsson and Sethian [3] that we briefly introduced in Section 4.3.3.

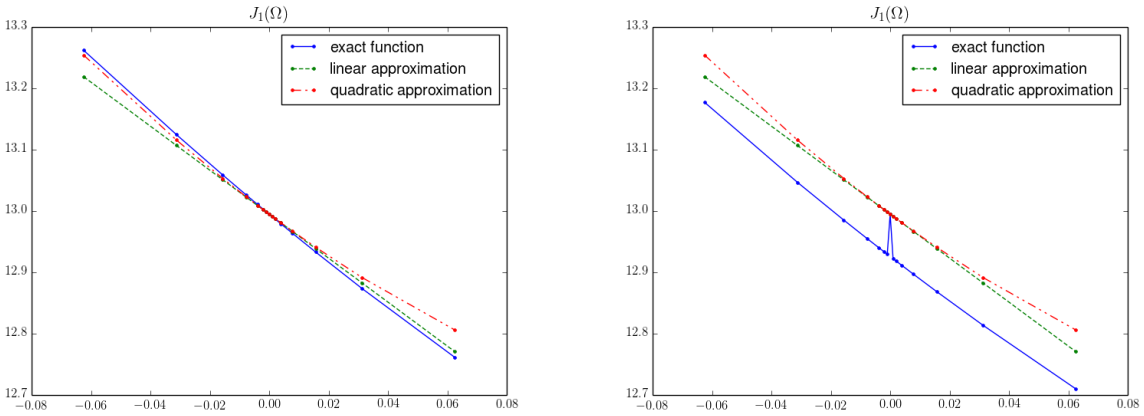


Figure 12.17: Consistency without (left) or with (right) redistanciation.

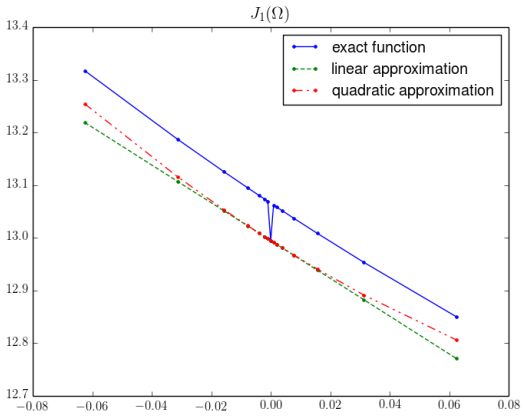


Figure 12.18: Positive additional error due to redistanciation.

12.2.4 Arch

As a first test case, we consider an arch under boundary conditions described by Figure 12.19. We consider a working domain of size 2×1 with a square mesh of size 120×60. Thus the value of h is $h = \frac{1}{60}$. For both the compliance and the least square criteria, the applied force is a point load $(0, -1)$ applied at the middle of the bottom boundary. It is classical for numerical experiments to apply a point load. However, for second-order optimization, this may not be a very good idea since the state u is probably not regular enough to apply the differentiability results of the previous section. The Young modulus and the Poisson coefficient are respectively $E = 1.0$ and $\nu = 0.3$. In the definition of the Lagrangian, the Lagrange multiplier is $\Lambda = 20$ for the initialization with holes and $\Lambda = 8$ for the initialization without holes.

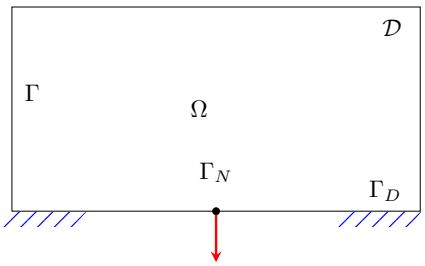


Figure 12.19: Boundary condition for the arch.

Least square criterion - Initialization with holes

In this example we consider the least square error criterion with target displacement $(0, 0)$ on the boundary Γ_N. Since both components of the displacement are minimized on Γ_N this least square objective function is different from the compliance. The initial shape is the one of Figure 12.20.

Figure 12.20: Initial shape for the arch.

Like previously, we denote Ω_∞ the final shape for each method, and $\Omega^\times$ the best of all six final shapes (which are plotted in Figure 12.21 and Figure 12.22). The final value of the Lagrangian and the number of iterations (a maximum number iteration is set to 200) needed to get convergence are transcripted in Table 12.7. Due to the line search, each iteration may need several finite element computations for the evaluation of the criterion. Therefore, we also give the number of finite element computations needed to get convergence. We also register in this table the `Gradient ratio` defined by

$$\frac{L'\left(\Omega_\infty; v^\infty\right)}{L'\left(\Omega_0; v^0\right)} \tag{12.9}$$

In this first example one can observe that the best shape is obtained with the Newton method and the extension Method 10.2.3. The convergence of $L(\Omega) - L(\Omega^\times)$ and $L(\Omega) - L(\Omega_\infty)$ are plotted respectively in Figure 12.23 and Figure 12.24. Looking at Figure 12.24 we see that all the Newton method converge with fewer iterations than the gradient methods. In this case, the choice of the extension method does not lead to major changes in the behavior of the algorithms and the final results.

Finally we consider the convergence rate for the error with respect to the final value for each algorithm (Figure 12.25 and Table 12.8). This indicator is as well deceptive here. We can see that the rates of convergence are similar for all methods, and around 1. There can be many reasons for that. First, all final shapes have corners. Like in the case of the square in Section 12.1.4 this singularity in the boundary is probably responsible for the degradation of the

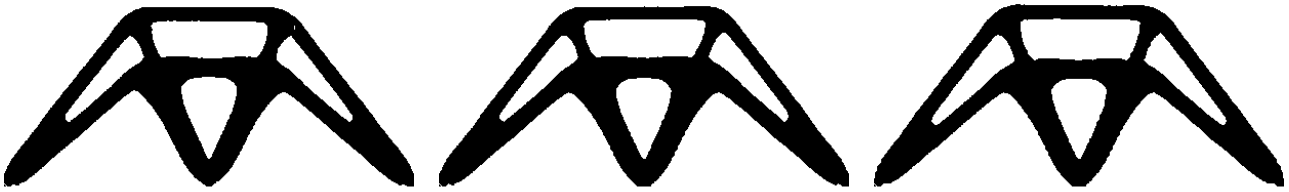


Figure 12.21: Final domains for the gradient algorithm for the least square criterion on the arch problem of Section 12.2.4 and initialization with the shape of Figure 12.20 with, from left to right, extension Method 10.2.3, Method 10.2.1 and Method 10.2.2.

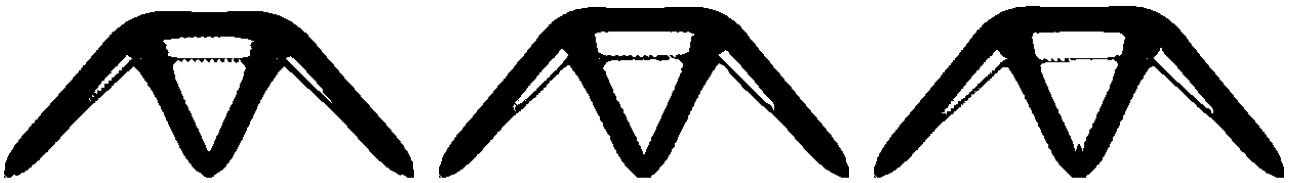


Figure 12.22: Final domains for the Newton algorithm for the least square criterion on the arch problem of Section 12.2.4 and initialization with the shape of Figure 12.20 with, from left to right, extension Method 10.2.3, Method 10.2.1, and Method 10.2.2.

		Gradient	Newton
Extension 10.2.3	Lagrangian	26.65396800	26.15282400
	Iterations	199	27
	FE	2533	74
	Gradient ratio	1.038×10^{-1}	1.209×10^{-2}
Extension 10.2.1	Lagrangian	26.69840400	26.22945200
	Iterations	199	23
	FE	2233	68
	Gradient ratio	1.383×10^{-1}	1.458×10^{-2}
Extension 10.2.2	Lagrangian	26.74557000	26.24563200
	Iterations	199	24
	FE	2477	84
	Gradient ratio	8.366×10^{-2}	1.694×10^{-2}

Table 12.7: Performance of each algorithm for the least square criterion on the arch problem of Section 12.2.4 and initialization with the shape of Figure 12.20.

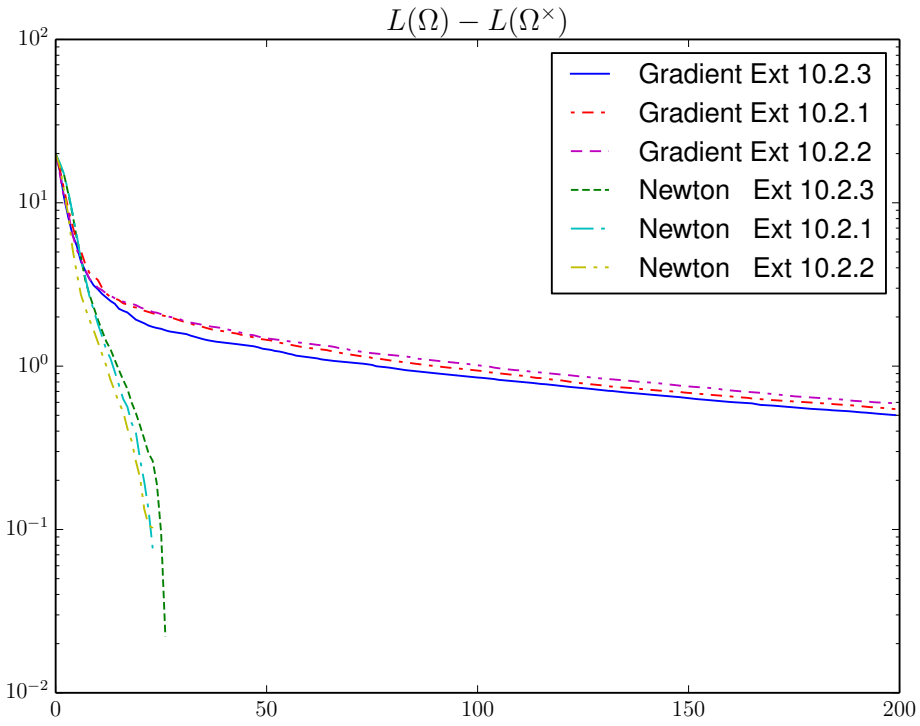


Figure 12.23: Convergence of $L(\Omega) - L(\Omega^\times)$ for the least square criterion on the arch problem of Section 12.2.4 and initialization with the shape of Figure 12.20.

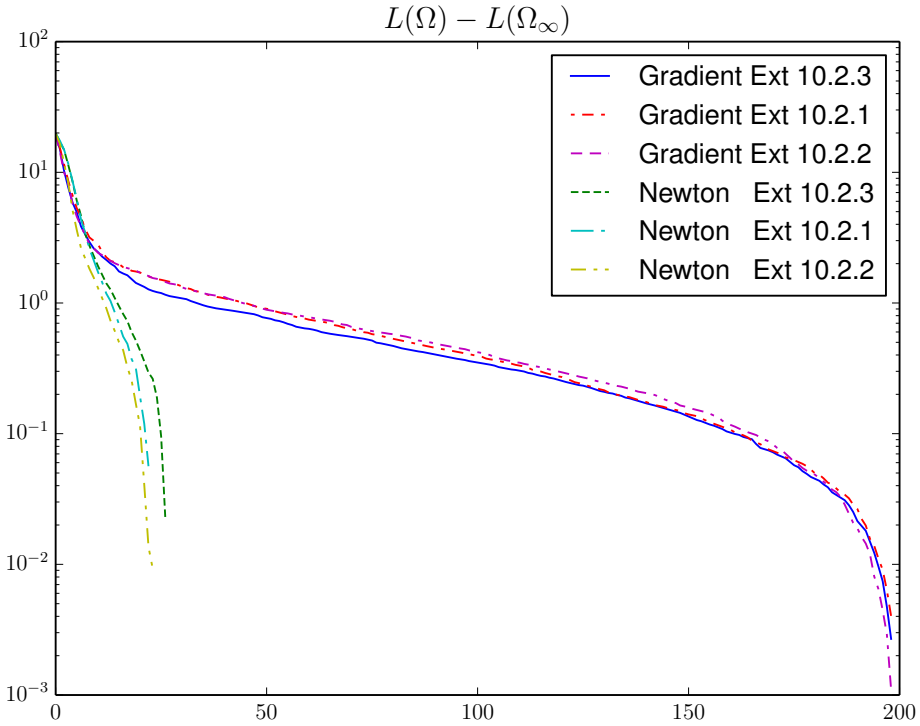


Figure 12.24: Convergence of $L(\Omega) - L(\Omega_\infty)$ for the least square criterion on the arch problem of Section 12.2.4 and initialization with the shape of Figure 12.20.

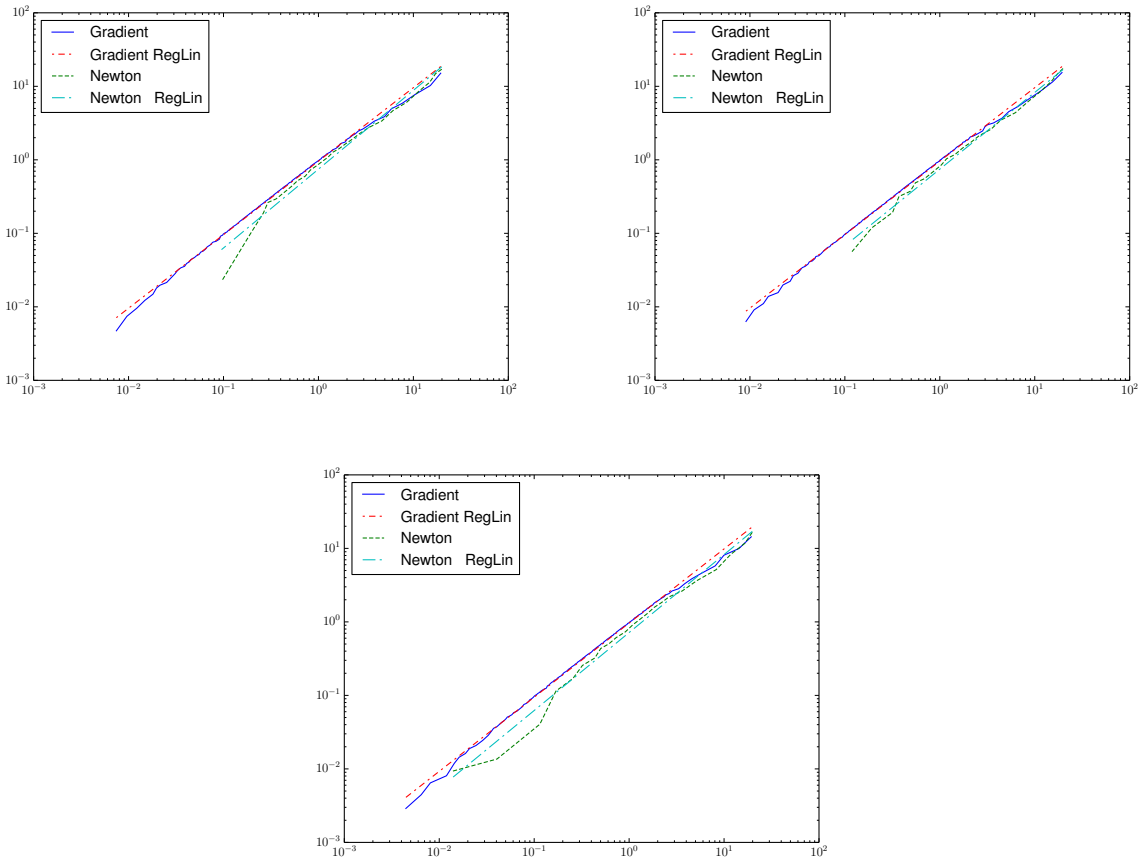


Figure 12.25: Linear regressions of the error with the best shape for the least square criterion on the arch problem of Section 12.2.4 and initialization with the shape of Figure 12.20. The extension is made with Method 10.2.3 (top-left), Method 10.2.1 (top-right), and Method 10.2.2 (bottom). Evolution of $L(\Omega_{n+1})-L(\Omega_\infty)$ with respect to $L(\Omega_n)-L(\Omega_\infty)$.

	Gradient	Newton
Extension 10.2.3	1.00	1.07
Extension 10.2.1	1.00	1.04
Extension 10.2.2	1.01	1.06

Table 12.8: Slopes of the linear regressions of Figure 12.25.

convergence rate. Secondly, the linear rate of convergence for the Newton method may come from the non-existence of optimal shapes. The second-order rate of convergence is indeed based on the assumption that the first-order optimality conditions are satisfied. When this condition is not satisfied - which is the case when there is no existence of an optimal shape - the second-order rate of convergence can naturally not be observed. Then we could also say that the low rate of convergence is due to the *optimize-then-discretize* approach. There are always optimal shapes for the discrete models, but in general they do not satisfy the first-order optimality condition, which could also explain the non-quadratic rate of convergence.

It appears that the observations on the rate of convergence are always the same in all the mechanical examples we consider here : both Newton's and gradient method have a linear rate of convergence. Thus, we will plot the rates of convergence only for this example.

Least square criterion - Initialization without holes

Figure 12.26: Initial shape for the arch.

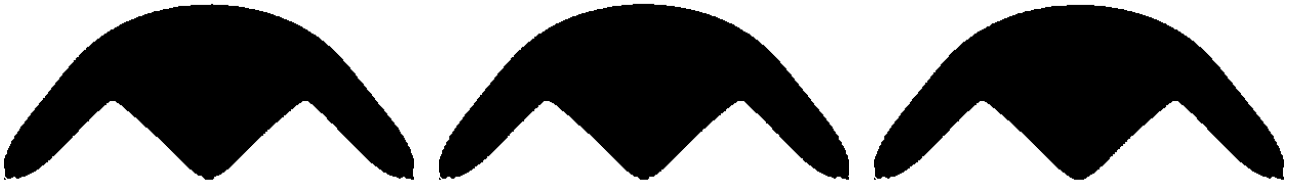


Figure 12.27: Final domains for the gradient algorithm for the least square criterion on the arch problem of Section 12.2.4 and initialization with the shape of Figure 12.26 with, from left to right, extension Method 10.2.3, Method 10.2.1 and Method 10.2.2.

Secondly we consider the same example but the initialization differs since it has no holes. All final shapes are displayed in Figure 12.27 and Figure 12.28 The final values of the Lagrangian and the number of iterations needed to get convergence are transcripted in Table 12.9. The best shape $\Omega^\times$ is obtained with the Newton method and the extension of Method 10.2.1. This case is particular since the latter method requires the largest number of iterations to converge. In Figure 12.29 and Figure 12.30 we respectively plot the convergence of $L(\Omega) - L(\Omega^\times)$ and $L(\Omega) - L(\Omega_\infty)$. We can see in Figure 12.29 that at the twentieth iteration, the current shape is already better that all final shapes of the other methods. If we leave this method apart, the Newton method converges to shapes that give nearly the optimal objective in fewer iterations than the gradient method. Of course, the final convergence of $L(\Omega)$ to $L(\Omega_\infty)$ is very fast but it is a lure since, by definition, $L(\Omega_\infty)$ is the limit value of $L(\Omega)$. Nevertheless, the convergence histories of Figure 12.29 and Figure 12.30 are good measures of convergence speed for the early iterations.

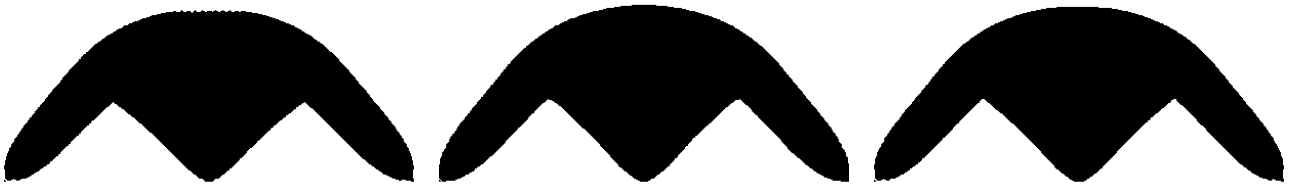


Figure 12.28: Final domains for the Newton algorithm for the least square criterion on the arch problem of Section 12.2.4 and initialization with the shape of Figure 12.26 with, from left to right, extension Method 10.2.3, Method 10.2.1, and Method 10.2.2.

		Gradient	Newton
Extension 10.2.3	Lagrangian	18.21257000	18.24514840
	Iterations	48	18
	FE	482	79
	Gradient ratio	3.963×10^{-2}	1.006×10^{-2}
Extension 10.2.1	Lagrangian	18.13401440	17.99718920
	Iterations	64	156
	FE	620	1072
	Gradient ratio	3.367×10^{-2}	5.717×10^{-4}
Extension 10.2.2	Lagrangian	18.22399400	18.21113520
	Iterations	41	30
	FE	387	175
	Gradient ratio	3.710×10^{-2}	1.228×10^{-3}

Table 12.9: Performance of each algorithm for the least square criterion on the arch problem of Section 12.2.4 and initialization with the shape of Figure 12.26.

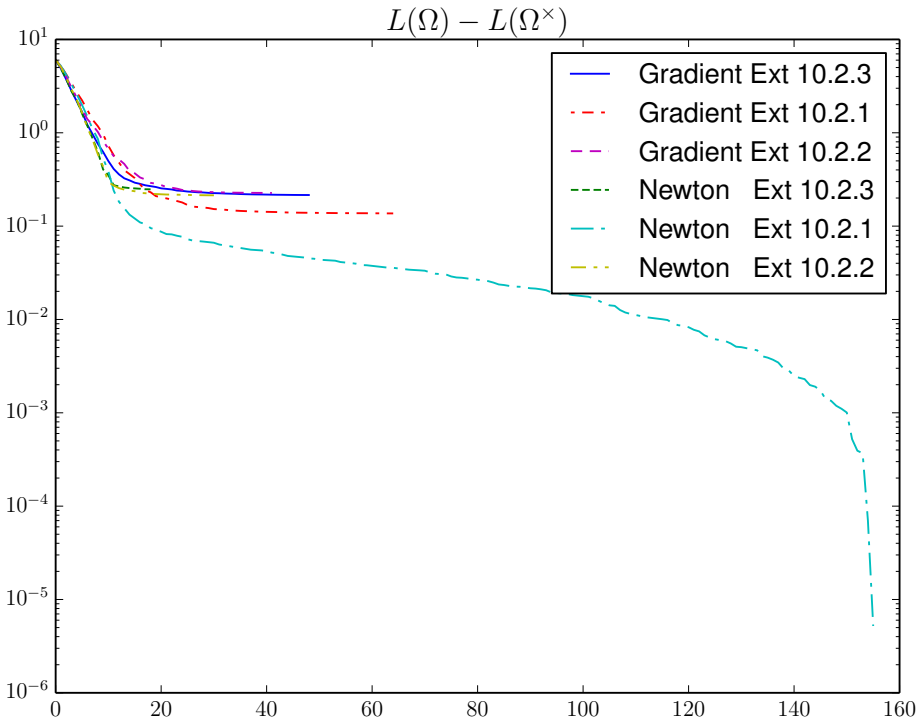


Figure 12.29: Convergence of $L(\Omega) - L(\Omega^\times)$ for the least square criterion on the arch problem of Section 12.2.4 and initialization with the shape of Figure 12.26.

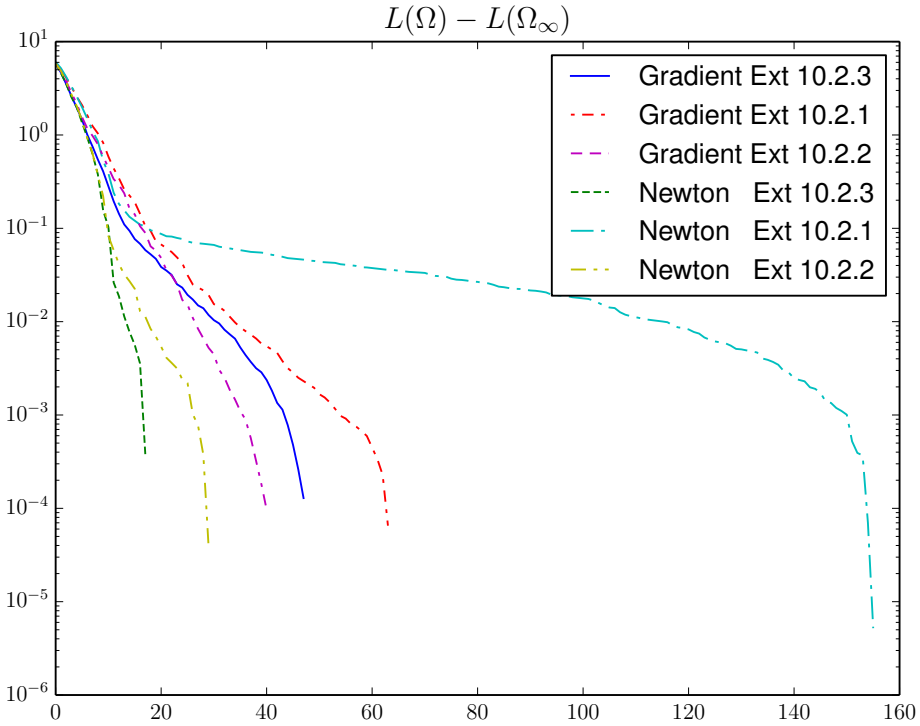


Figure 12.30: Convergence of $L(\Omega) - L(\Omega_\infty)$ for the least square criterion on the arch problem of Section 12.2.4 and initialization with the shape of Figure 12.26.

Compliance - Initialization without holes

Next we consider the compliance for the arch, for an initialization without holes. All final shapes are plotted in Figure 12.31 and Figure 12.32. In Table 12.10 we can see the final values of the Lagrangian and the number of iterations needed to converge. We also plot the convergence of $L(\Omega) - L(\Omega^{\times})$ and $L(\Omega) - L(\Omega_{\infty})$ respectively in Figure 12.33 and Figure 12.34.

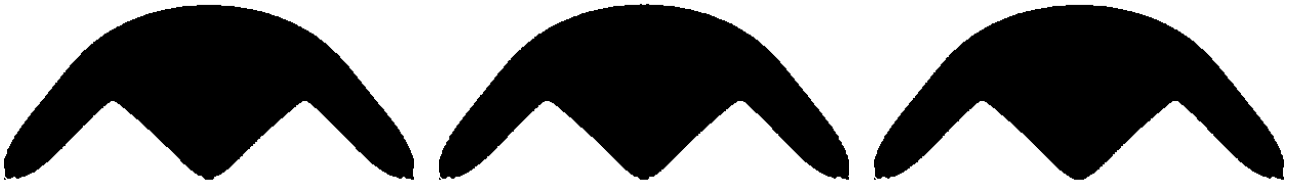


Figure 12.31: Final domains for the gradient algorithm for the compliance criterion on the arch problem of Section 12.2.4 and initialization with the shape of Figure 12.26 with, from left to right, extension Method 10.2.3, Method 10.2.1 and Method 10.2.2.

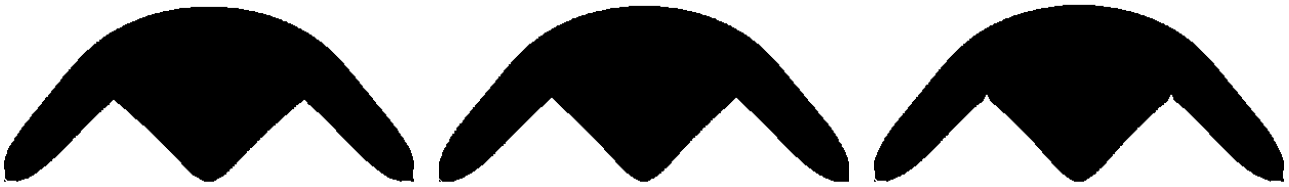


Figure 12.32: Final domains for the Newton algorithm for the compliance criterion on the arch problem of Section 12.2.4 and initialization with the shape of Figure 12.26 with, from left to right, extension Method 10.2.3, Method 10.2.1, and Method 10.2.2.

The extension 10.2.1 leads to the best shapes for both Newton's and gradient method, but it has a great cost in terms of iterations. Excepting these two results, the best shape is obtained by the Newton method with extension 10.2.2. In this case there is not much difference - for the Newton method - between the extensions 10.2.3 and 10.2.2.

We can also mention the number of finite element computations, corresponding to `FE` in Table 12.10. For the Newton method, there is much less finite element computations. This means that the line search for the Newton method is much less time consuming. Thus, the time step is closer to 1 which is the ideal time step for an exact Newton's method.

		Gradient	Newton
Extension 10.2.3	Lagrangian	18.21268880	18.21041280
	Iterations	55	18
	FE	574	88
	Gradient ratio	4.358×10^{-2}	1.908×10^{-4}
Extension 10.2.1	Lagrangian	18.13557280	17.96468880
	Iterations	60	35
	FE	575	90
	Gradient ratio	3.239×10^{-2}	3.025×10^{-4}
Extension 10.2.2	Lagrangian	18.22403080	18.19544320
	Iterations	39	21
	FE	353	105
	Gradient ratio	3.783×10^{-2}	4.065×10^{-4}

Table 12.10: Performance of each algorithm for the compliance criterion on the arch problem of Section 12.2.4 and initialization with the shape of Figure 12.26.

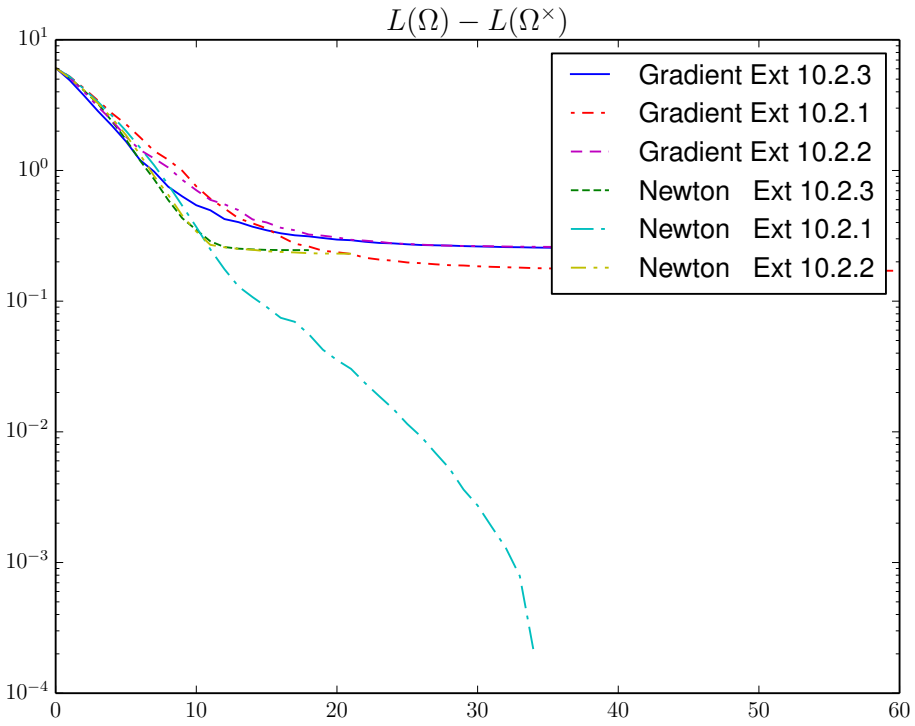


Figure 12.33: Convergence of $L(\Omega) - L(\Omega^\times)$ for the compliance criterion on the arch problem of Section 12.2.4 and initialization with the shape of Figure 12.26.

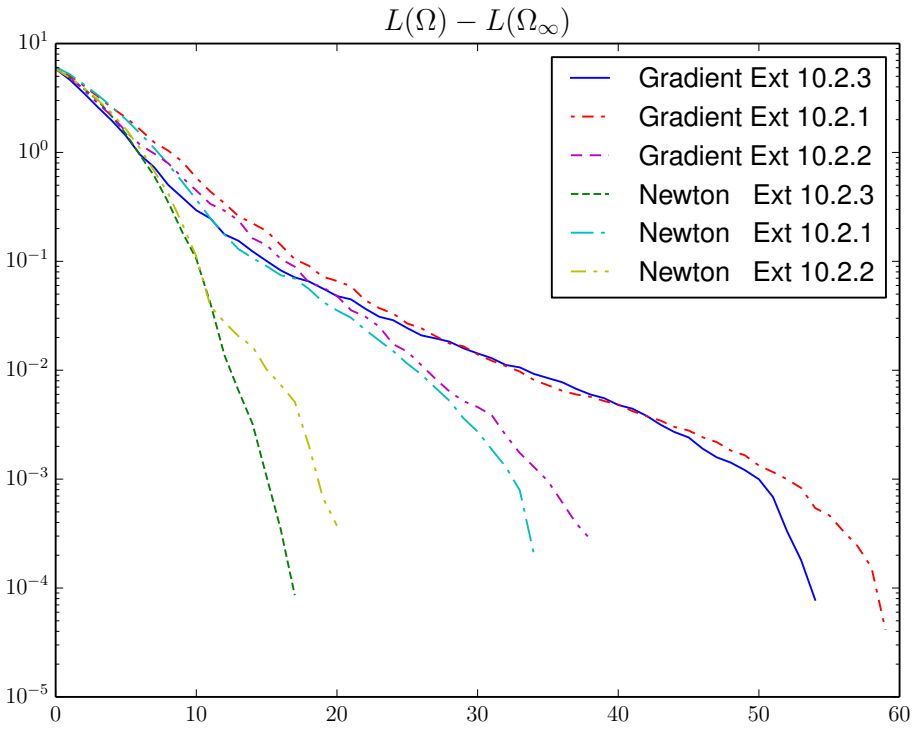


Figure 12.34: Convergence of $L(\Omega) - L(\Omega_\infty)$ for the compliance criterion on the arch problem of Section 12.2.4 and initialization with the shape of Figure 12.26.

Compliance - Initialization with holes

We still minimize the compliance but with an initialization with holes like in Figure 12.20. In this example, we take the opportunity to report, not only the total number of iterations and finite element analyses, but also the total number of linear system solves. In the case of the gradient algorithm, each finite element analysis corresponds to one and only one linear system to solve (for compliance minimization there is no adjoint equation). However, for Newton's algorithm, at each iteration we compute the Hessian matrix, which requires to solve as many equations (12.7) as the number of degrees of freedom on the boundary $\partial\Omega$. Solving (12.7) amounts to solve a linear system with the same rigidity matrix as the state equation (12.5). Therefore, if the rigidity matrix is factorized once per iteration, the additional cost of the Newton algorithm is just a large number of back-and-forth substitutions. Factorizing the rigidity matrix, and thus minimizing the overhead of Newton's algorithm is possible in 2-d but is clearly a problem in 3-d. We recall that, for the moment, we do not try to minimize the CPU and memory costs of Newton's algorithm.

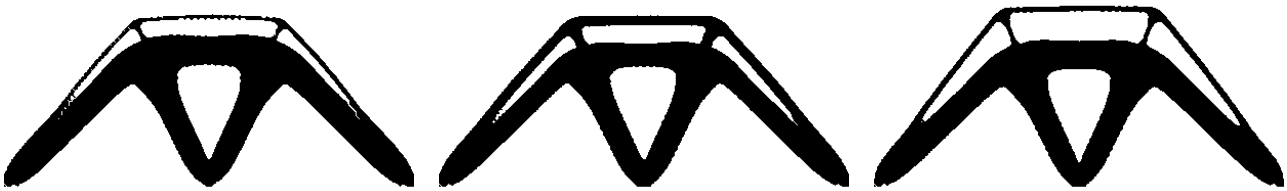


Figure 12.35: Final domains for the gradient algorithm for the compliance criterion on the arch problem of Section 12.2.4 and initialization with the shape of Figure 12.20 with, from left to right, extension Method 10.2.3, Method 10.2.1 and Method 10.2.2.

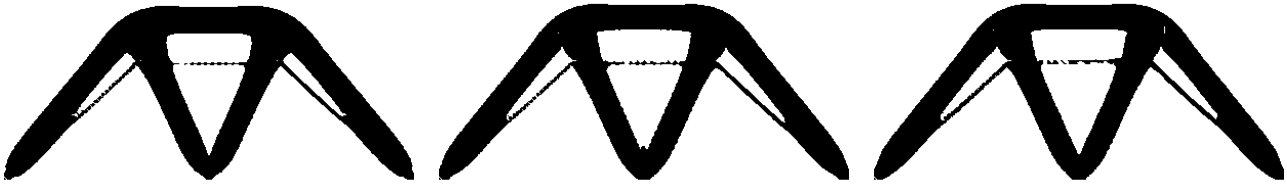


Figure 12.36: Final domains for the Newton algorithm for the compliance criterion on the arch problem of Section 12.2.4 and initialization with the shape of Figure 12.20 with, from left to right, extension Method 10.2.3, Method 10.2.1, and Method 10.2.2.

The final shapes are plotted in Figure 12.35 and Figure 12.36, while the performance of the different algorithms are detailed in Table 12.11. We also plot the convergence of $L(\Omega) - L(\Omega^\times)$ and $L(\Omega) - L(\Omega_\infty)$ respectively in Figure 12.37 and Figure 12.38. The behavior of the different optimization methods is almost the same as for the minimization of the displacement criterion in the case of the arch. Even if the best shape is obtained with gradient method and extension Method 10.2.3, the change of extension method seems to accelerate a bit the convergence. The Newton method converges anyway faster than the gradient method for this case, and whatever the choice of extension.

		Gradient	Newton
Extension 10.2.3	Lagrangian	26.22108415	26.26417000
	Iterations	181	34
	FE	2024	97
	FE total	2025	51894
	Gradient ratio	9.319×10^{-2}	6.547×10^{-3}
Extension 10.2.1	Lagrangian	26.31501400	26.29651600
	Iterations	190	28
	FE	1920	74
	FE total	1921	43270
	Gradient ratio	6.157×10^{-2}	1.239×10^{-2}
Extension 10.2.2	Lagrangian	26.38028800	26.34791200
	Iterations	133	26
	FE	1400	79
	FE total	1401	40556
	Gradient ratio	8.374×10^{-2}	1.340×10^{-2}

Table 12.11: Performance of each algorithm for the compliance criterion on the arch problem of Section 12.2.4 and initialization with the shape of Figure 12.20.

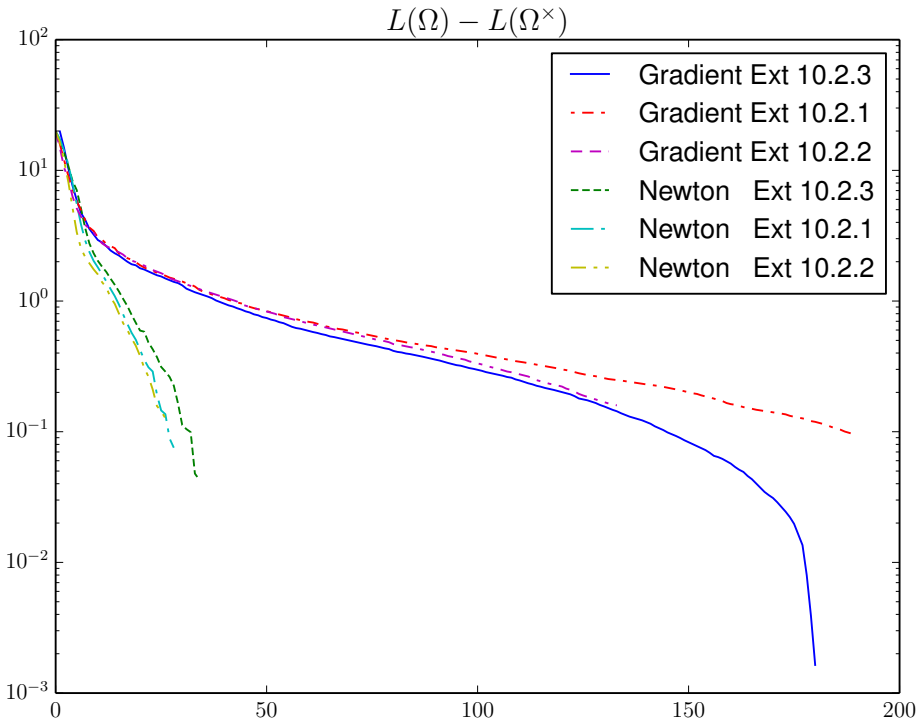


Figure 12.37: Convergence of $L(\Omega) - L(\Omega^\times)$ for the compliance criterion on the arch problem of Section 12.2.4 and initialization with the shape of Figure 12.20.

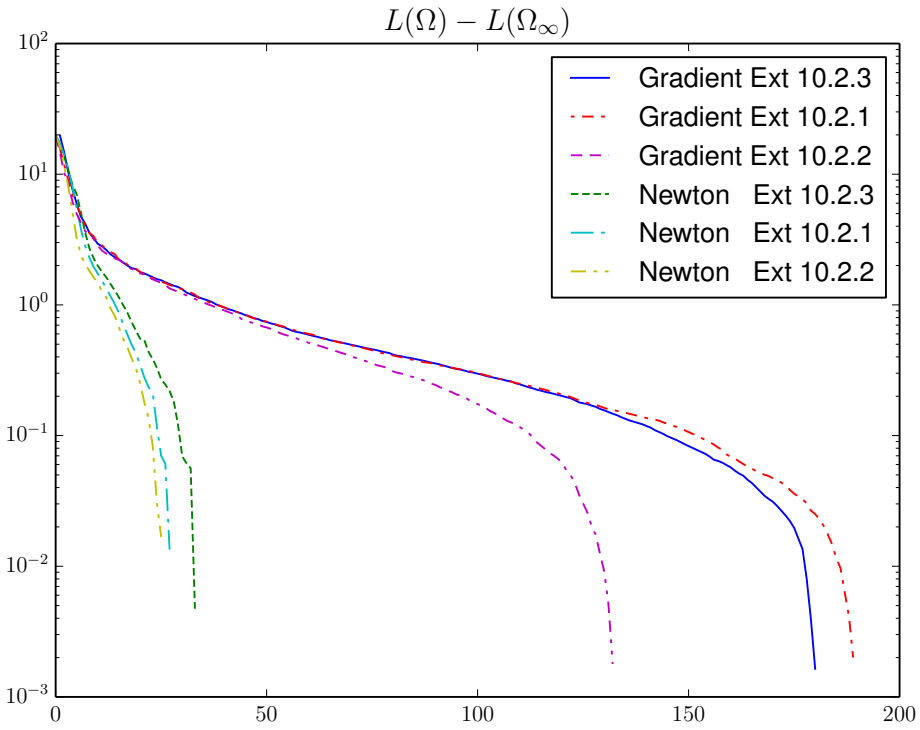


Figure 12.38: Convergence of $L(\Omega) - L(\Omega_\infty)$ for the compliance criterion on the arch problem of Section 12.2.4 and initialization with the shape of Figure 12.20.

Compliance - Initialization with holes and a finer mesh of size 200×100

To assess the mesh dependency, or scalability, of Newton's algorithm, compared to the gradient algorithm, we run the same test case as in Section 12.2.4, but on a finer mesh of size 200×100 (while the previous one was 120x60). Starting from the same initialization, the final domains are the same (but with different values of the Lagrangian).

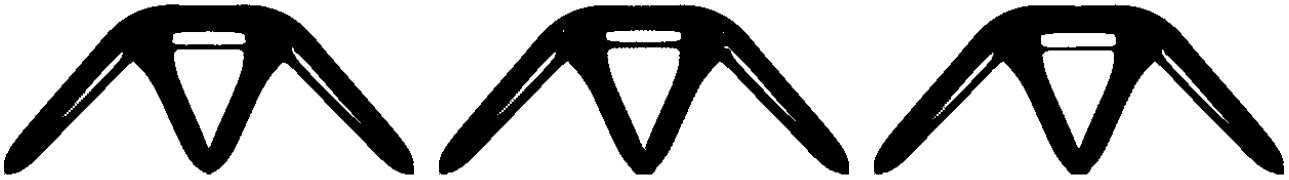


Figure 12.39: Final domains for the gradient algorithm for the compliance criterion on the arch problem of Section 12.2.4 and initialization with the shape of Figure 12.20 on a finer mesh of size 200×100 with, from left to right, extension Method 10.2.3, Method 10.2.1 and Method 10.2.2.

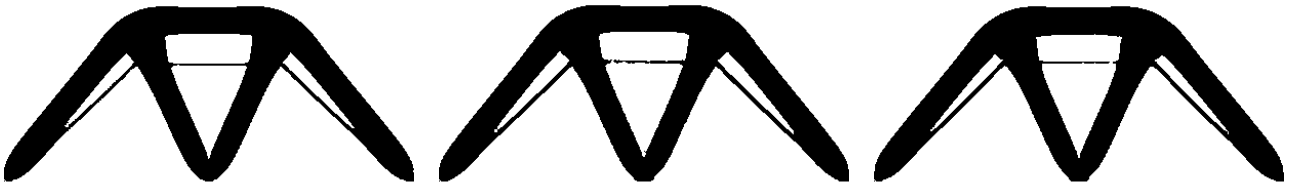


Figure 12.40: Final domains for the Newton algorithm for the compliance criterion on the arch problem of Section 12.2.4 and initialization with the shape of Figure 12.20 on a finer mesh of size 200×100 with, from left to right, extension Method 10.2.3, Method 10.2.1, and Method 10.2.2.

The final shapes are plotted in Figure 12.39 and Figure 12.40, while the performance of the different algorithms are detailed in Table 12.12. The convergence of $L(\Omega) - L(\Omega^\times)$ and $L(\Omega) - L(\Omega_\infty)$ are finally plotted in Figure 12.41 and Figure 12.42. Compared to the coarser mesh of the previous example, the number of iterations and finite element analyses has roughly doubled for the Newton method. Since the number of cells has doubled in each space direction too, the number of degrees of freedom on the boundary $\partial\Omega$ has roughly doubled. Therefore it is not a surprise that the number of linear system solves for Newton's algorithm has approximately quadrupled. In any case, it is clear that the cost of Newton's algorithm may turn out to be prohibitive for fine meshes, motivating the development of inexact (but cheaper) Newton's strategies. This example is also very interesting since the final shapes obtained with the Newton method are rather different from the one obtained with the gradient method. Even if the final values of the objective functions are lower with the gradient method, the Newton method converge always (for this example) much more rapidly.

		Gradient	Newton
Extension 10.2.3	Lagrangian	26.75750760	26.88574747
	Iterations	320	49
	FE	3500	95
	FE total	3501	127656
	Gradient ratio	4.711×10^{-2}	7.702×10^{-3}
Extension 10.2.1	Lagrangian	26.75589302	26.92230720
	Iterations	349	50
	FE	3373	92
	FE total	3374	133390
	Gradient ratio	4.026×10^{-2}	1.243×10^{-2}
Extension 10.2.2	Lagrangian	26.76955930	26.87542852
	Iterations	343	51
	FE	3808	120
	FE total	3809	136709
	Gradient ratio	3.748×10^{-2}	6.372×10^{-4}

Table 12.12: Performance of each algorithm for the compliance criterion on the arch problem of Section 12.2.4 and initialization with the shape of Figure 12.20 on a finer mesh of size 200×100.

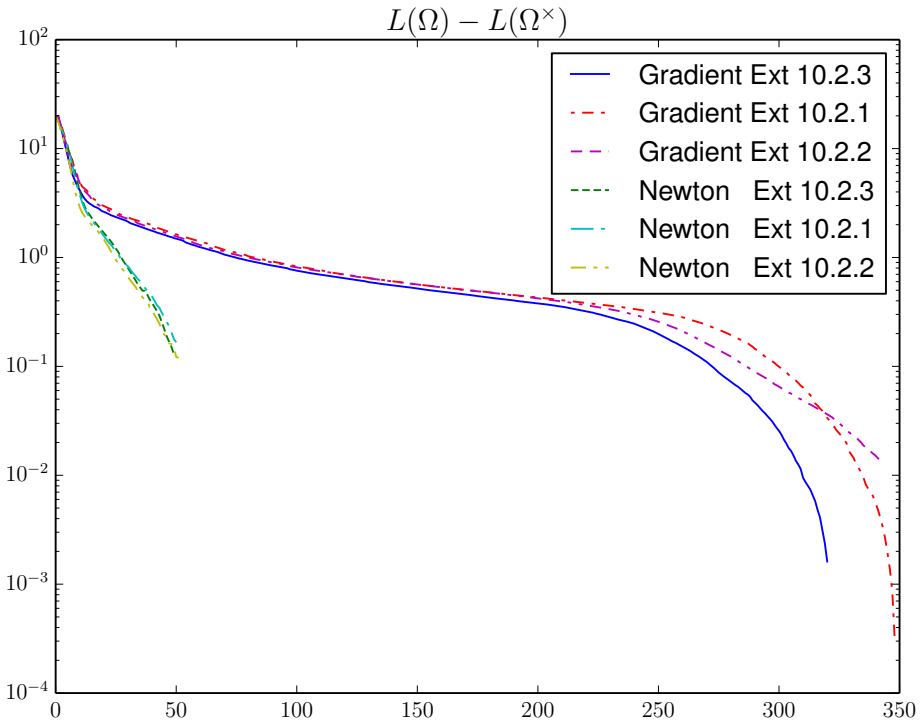


Figure 12.41: Convergence of $L(\Omega) - L(\Omega^\times)$ for the compliance criterion on the arch problem of Section 12.2.4 and initialization with the shape of Figure 12.20 on a finer mesh of size 200×100.

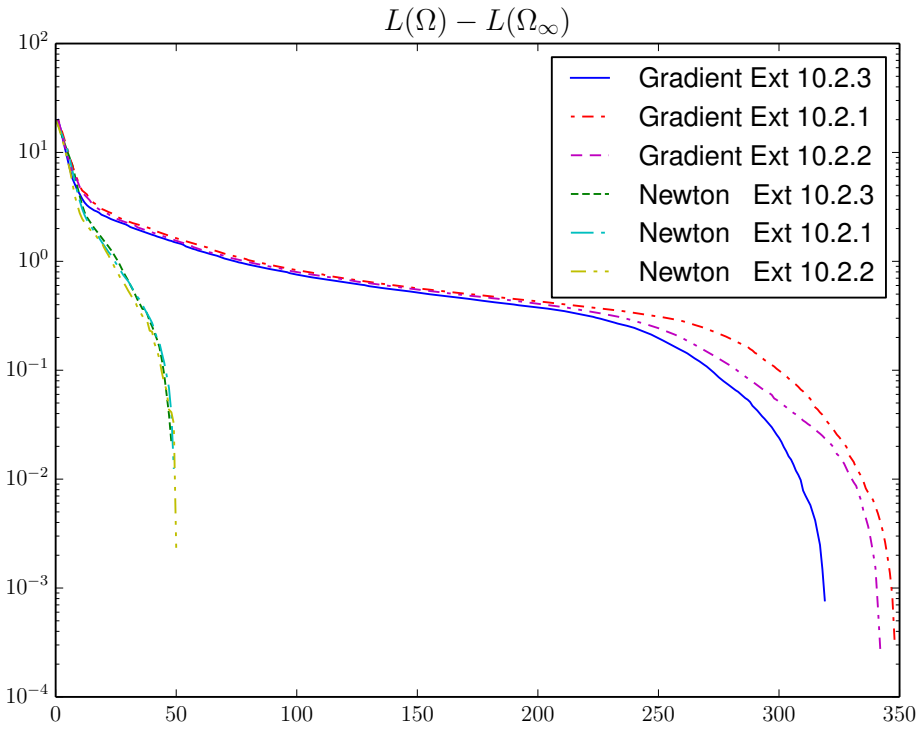


Figure 12.42: Convergence of $L(\Omega) - L(\Omega_\infty)$ for the compliance criterion on the arch problem of Section 12.2.4 and initialization with the shape of Figure 12.20 on a finer mesh of size 200×100.

12.2.5 Cantilever

Our second and last test case is a cantilever. The working domain and the mesh have the same size as in the case of the arch, i.e., 2×1 and 120×60. The value of h is also $\frac{1}{60}$. For both the compliance and the least square criteria, the applied force is a point load $(0, -0.1)$ applied at the middle of the right boundary. The Young modulus and the Poisson coefficient are still respectively $E = 1.0$ and $\nu = 0.3$.

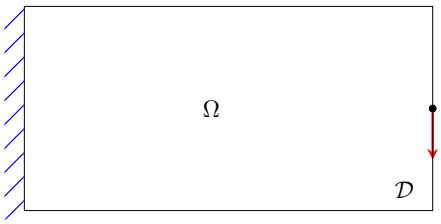


Figure 12.43: Cantilever

Least square criterion - Initialization with holes

We first consider the least square error on a cantilever for an initialization with holes (see Figure 12.44). The criterion is similar to the case of the arch, since we take the same target displacement, i.e $u_0 = (0, 0)$ on Γ_N. In the definition of the Lagrangian, the Lagrange multiplier is $\Lambda = 16$.

Figure 12.44: Initial shape for the cantilever.

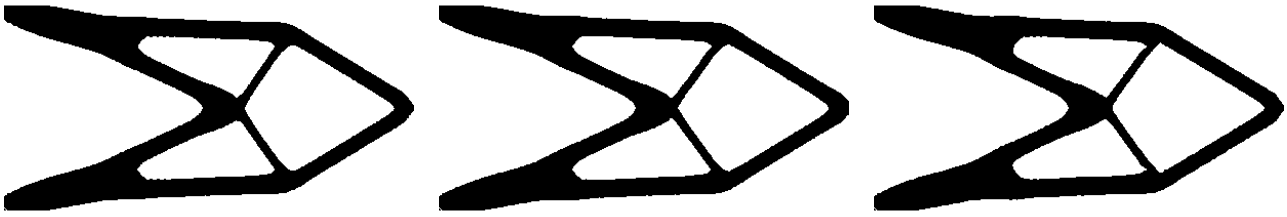


Figure 12.45: Final domains for the gradient algorithm for the least square criterion on the cantilever problem of Section 12.2.5 and initialization with the shape of Figure 12.44 with, from left to right, extension Method 10.2.3, Method 10.2.1 and Method 10.2.2.

The final shapes are plotted in Figure 12.45 and Figure 12.46, while the performance of the different algorithms are detailed in Table 12.13. The convergence of $L(\Omega) - L(\Omega^\times)$ and $L(\Omega) - L(\Omega_\infty)$ are respectively plotted in Figure 12.47 and Figure 12.48. The best shape is obtained with the Newton algorithm and extension Method 10.2.2. In this example,

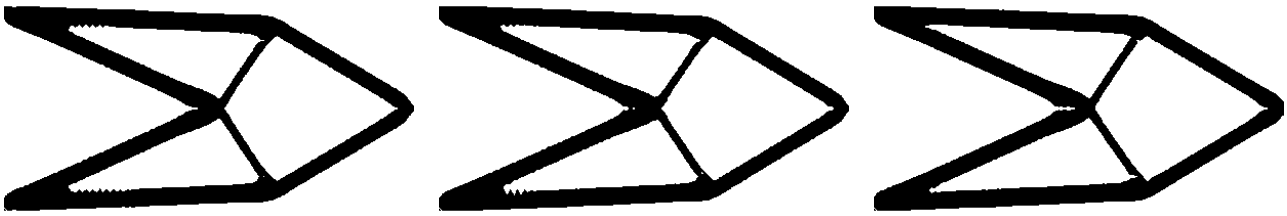


Figure 12.46: Final domains for the Newton algorithm for the least square criterion on the cantilever problem of Section 12.2.5 and initialization with the shape of Figure 12.44 with, from left to right, extension Method 10.2.3, Method 10.2.1, and Method 10.2.2.

		Gradient	Newton
Extension 10.2.3	Lagrangian	19.65491920	19.18934280
	Iterations	51	45
	FE	551	102
	Gradient ratio	9.179×10^{-2}	8.860×10^{-3}
Extension 10.2.1	Lagrangian	19.62798240	19.15956200
	Iterations	45	35
	FE	408	91
	Gradient ratio	1.010×10^{-1}	6.978×10^{-3}
Extension 10.2.2	Lagrangian	19.70648800	19.11006840
	Iterations	38	37
	FE	385	101
	Gradient ratio	1.163×10^{-1}	4.350×10^{-3}

Table 12.13: Performance of each algorithm for the least square criterion on the cantilever problem of Section 12.2.5 and initialization with the shape of Figure 12.44.

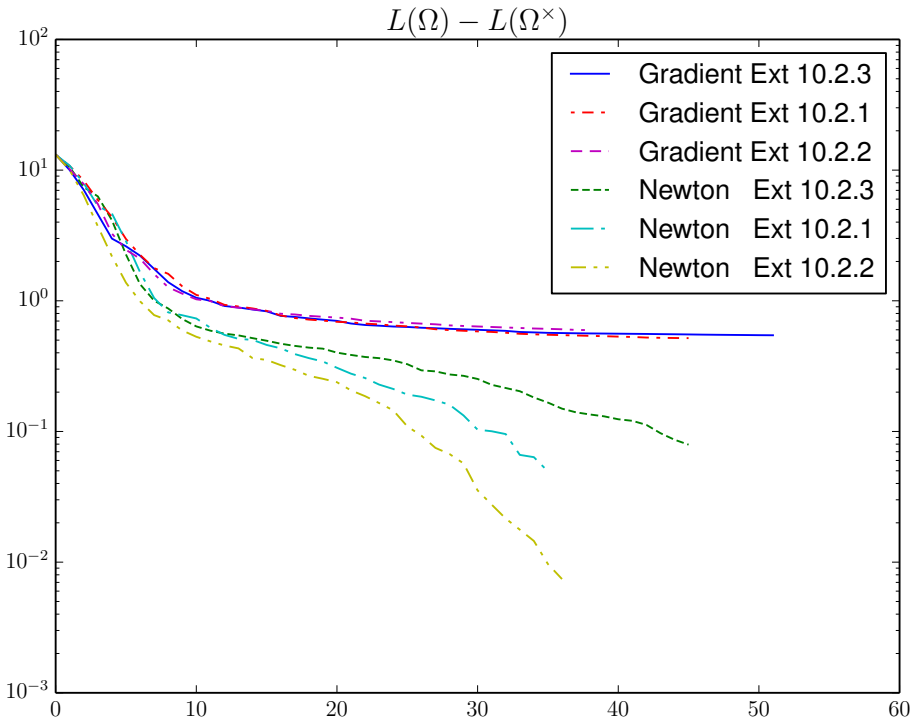


Figure 12.47: Convergence of $L(\Omega) - L(\Omega^\times)$ for the least square criterion on the cantilever problem of Section 12.2.5 and initialization with the shape of Figure 12.44.

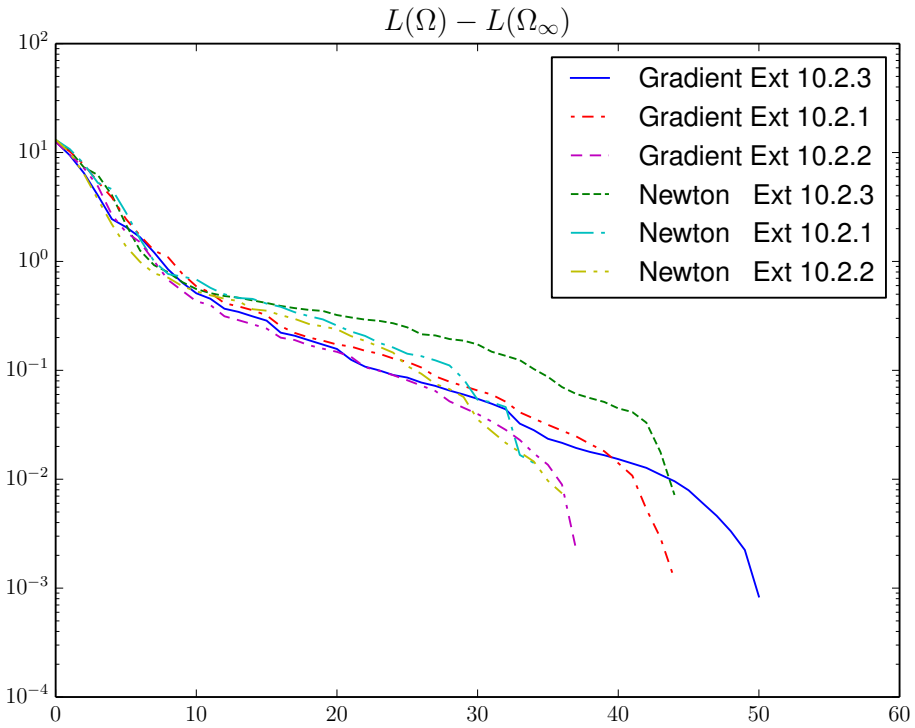


Figure 12.48: Convergence of $L(\Omega) - L(\Omega_\infty)$ for the least square criterion on the cantilever problem of Section 12.2.5 and initialization with the shape of Figure 12.44.

compared to the gradient method, the Newton method converges always more rapidly to a shape for which the final value of the objective function is smaller. We could also notice that, for the Newton method, the two extensions given by Method 10.2.1 and Method 10.2.2 lead to smaller objective values for the final shapes than with the extension of Method 10.2.3. Comparing the curves of the Newton method, we can also conclude that the improvement of the second-order method relies on the computation of the Hessian but also on the choice of the extension which has a great influence on the result.

Compliance - Initialization with holes

Now we consider the compliance for the cantilever with an initialization with holes (see Figure 12.44). In the definition of the Lagrangian, the Lagrange multiplier is $\Lambda = 1$.

Figure 12.49: Final domains for the gradient algorithm for the compliance criterion on the cantilever problem of Section 12.2.5 and initialization with the shape of Figure 12.44 with, from left to right, extension Method 10.2.3, Method 10.2.1 and Method 10.2.2.

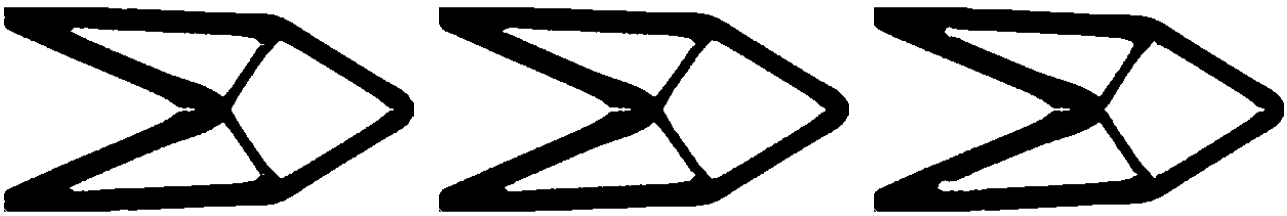


Figure 12.50: Final domains for the Newton algorithm for the compliance criterion on the cantilever problem of Section 12.2.5 and initialization with the shape of Figure 12.44 with, from left to right, extension Method 10.2.3, Method 10.2.1, and Method 10.2.2.

The final shapes are plotted in Figure 12.49 and Figure 12.50, while the performance of the different algorithms are detailed in Table 12.14. The convergence curves of $L(\Omega) - L(\Omega^\times)$ and $L(\Omega) - L(\Omega_\infty)$ are respectively plotted in Figure 12.51 and Figure 12.52. In this example, the Newton method performs quite well since for each extension method, the convergence needs fewer iterations and leads to a shape whose objective function is lower. We can also have a look at the number of evaluations - `FE` in Table 12.14 - and see that with the Newton method the number of finite element computation is much less important. We also recall that the number of finite element computations in the field `FE` of Table 12.14 corresponds to the number of different shapes on which the criterion is evaluated during the optimization process. For a mechanical criterion, this means that the stiffness matrix has to be assembled as many times.

Looking at the quantity denoted by `Gradient ratio` in Table 12.14, we can also notice a negative value for the Newton method with extension of Method 10.2.2. This means that the direction found by the quadratic subproblem (10.27) is not a descent direction. Since at the beginning, v^0 is always a descent direction (at least in practice), it

		Gradient	Newton
Extension 10.2.3	Lagrangian	1.56474350	1.53127160
	Iterations	50	41
	FE	544	86
	Gradient ratio	8.573×10^{-2}	8.504×10^{-3}
Extension 10.2.1	Lagrangian	1.56163040	1.52958940
	Iterations	50	46
	FE	467	89
	Gradient ratio	7.330×10^{-2}	6.576×10^{-3}
Extension 10.2.2	Lagrangian	1.57032030	1.53543090
	Iterations	54	25
	FE	648	71
	Gradient ratio	8.033×10^{-2}	-9.638×10^{-3}

Table 12.14: Performance of each algorithm for the compliance criterion on the cantilever problem of Section 12.2.5 and initialization with the shape of Figure 12.44.

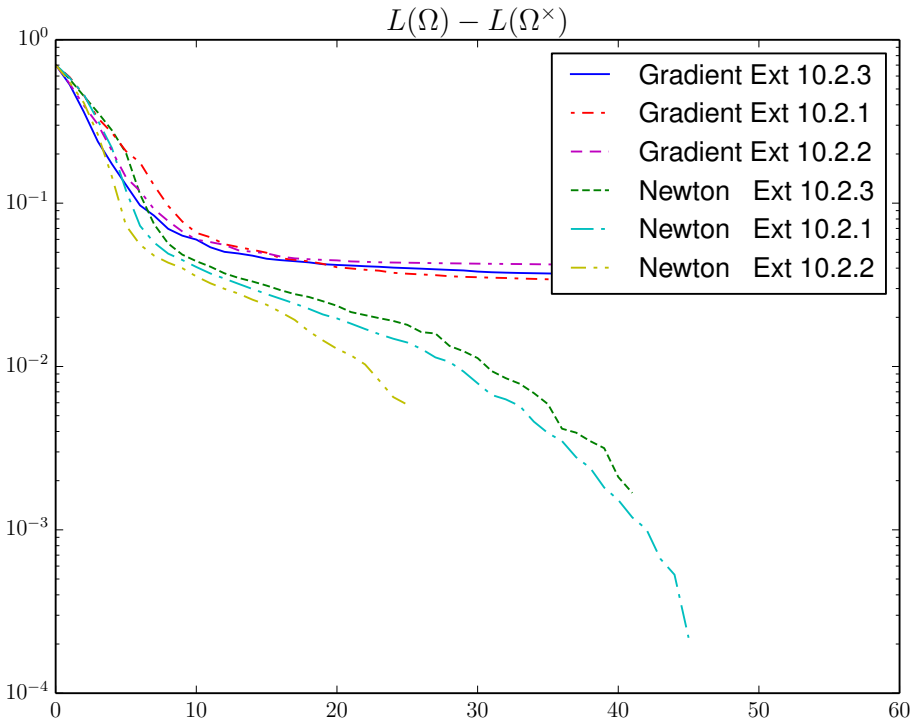


Figure 12.51: Convergence of $L(\Omega) - L(\Omega^\times)$ for the compliance criterion on the cantilever problem of Section 12.2.5 and initialization with the shape of Figure 12.44.

means that at this last iteration (like it is mentioned in Remark 10.3.2), the direction v^∞ is replaced by its opposite for the line search procedure. However, in this case, it appears that it does not permit to improve the current shape, meaning that the algorithm reached the best possible shape.

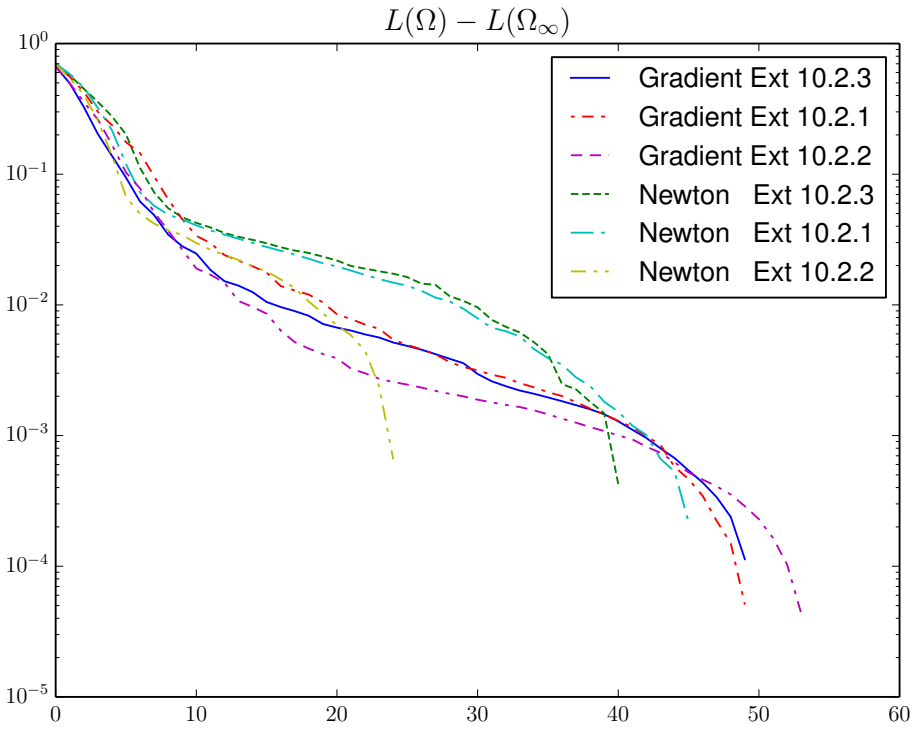


Figure 12.52: Convergence of $L(\Omega) - L(\Omega_\infty)$ for the compliance criterion on the cantilever problem of Section 12.2.5 and initialization with the shape of Figure 12.44.

12.3 Mechanical examples for approximation of the second-order shape derivative

The Newton method in shape optimization may be extremely time consuming. The computation of the second-order shape derivative of a criterion depending on a state equation (see Section 9.2 Section 12.2.2) indeed requires to solve as many linear systems as the number of nodes in the vicinity of the boundary. The time cost tends to be extremely important especially in three-dimensional cases (see Section 12.4). Thus, we are now concerned with the question of approximating the Hessian instead of computing it exactly. We will explore the different methods proposed in Section 9.2 on different examples. In that section different approximation methods were introduced : the BFGS-like and inexact Newton's method, the incomplete LU method, and the approximate CG-Newton method. It appears that the BFGS-like as well as the inexact Newton's method did perform well for the case of thickness optimization (see Section 9.3). However, for the geometric optimization we consider here, these methods stops at the first iterations of the optimization process whereas the shape can easily be still improved. The direction found by these methods can become very erratic (see Figure 12.53). Thus, even if numerically the first-order derivative in this direction is negative, this direction may not permit to decrease the objective function. Therefore, we will focus here on the last

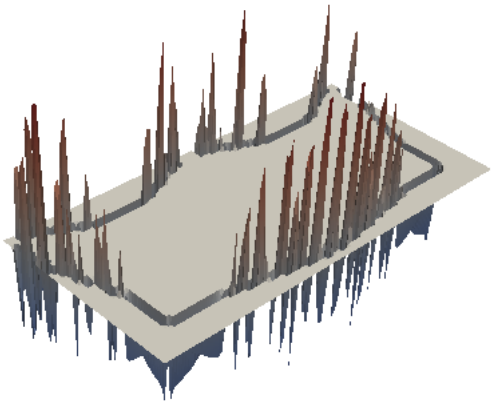


Figure 12.53: Descent direction at iteration 5 for the inexact Newton's method applied to the minimization of the compliance on the arch problem with initialization with the shape of Figure 12.26.

two methods. We will also consider another kind of approximation referred to as the *mass-lumping approximation*, that we start by describing.

12.3.1 Mass-lumping approximation

The mass-lumping method (see Section 5.2) is well-known for approximation of mass matrices. With a given mass matrix M is consists in building a diagonal matrix $\tilde{M}$ where the diagonal is given by the vector Me, with $e = (1, \cdots, 1)^T$. Here we will apply this method to the approximation of the Hessian matrix : we replace the matrix H of the second-order derivative by the diagonal matrix whose diagonal is given by He. The main advantage is to reduce significantly the computational cost. We detail it in the case of the compliance criterion. We recall that the second-order shape derivatives write

$$l_2(\theta,\xi) = 2\int_\Omega A\varepsilon(u'_\theta) : \varepsilon(u'_\xi) - \int_\Gamma (\theta\cdot\mathbf{n})\,(\xi\cdot\mathbf{n})\left(\mathcal{H}A\varepsilon(u):\varepsilon(u) + \partial_{\mathbf{n}}\Big(A\varepsilon(u):\varepsilon(u)\Big)\right),$$

where $\mathcal{H}$ is the mean curvature on the boundary and u'_θ is the solution of

$$\left\{\begin{array}{rcll} -\mathrm{div}\big(A\varepsilon(u'_\theta)\big) &=& 0 & \text{in } \Omega,\\ u'_\theta &=& 0 & \text{on } \Gamma_D,\\ A\varepsilon(u'_\theta)\mathbf{n} &=& 0 & \text{on } \Gamma_N,\\ A\varepsilon(u'_\theta)\mathbf{n} &=& \mathrm{div}_\Gamma\big((\theta\cdot\mathbf{n})\,A\varepsilon(u)\big) & \text{on } \Gamma. \end{array}\right. \tag{12.10}$$

It can also be written like

$$l_2(\theta,\xi) = -2\int_\Gamma (\theta\cdot\mathbf{n})\, A\varepsilon(u):\varepsilon(u'_\xi) - \int_\Gamma (\theta\cdot\mathbf{n})\,(\xi\cdot\mathbf{n})\Big(\mathcal{H}A\varepsilon(u):\varepsilon(u) + \partial_{\mathbf{n}}\Big(A\varepsilon(u):\varepsilon(u)\Big)\Big).$$

The second integral in this expression is not expensive to compute. To the contrary, the first integral implies to solve as many linear system with as the number of nodes in the vicinity of the boundary. With mass-lumping approximation, there is only one linear system to solve for computing u'_1. Then, the first integral of l_2 is replaced by

$$\int_\Gamma (\theta\cdot\mathbf{n})\,(\xi\cdot\mathbf{n})\, A\varepsilon(u):\varepsilon(u'_1).$$

There is a priori no reason for this approximation to be especially accurate, but it has at least two majors interest. First, this approximation of the Hessian is not time consuming since it requires only one linear system solve. Secondly, the resulting Hessian matrix is diagonal. Therefore, it eases the resolution of the sub-problem (10.27) for finding a descent direction.

12.3.2 Consistency of the approximations

Like in Section 11.1.2 and Section 11.3.2 we want first to evaluate the consistencies of the different approximations of the second-order shape derivative. We focus on the example of the arch introduced in Section 12.2.4, with the compliance (J_1) and the least square displacement (J_2) criterion.

For the approximate CG-Newton method, the direction v_γ is found by Algorithm 9.2.4. For the incomplete LU or mass-lumping method, the direction v_γ is the solution to (10.27) where the bilinear map $l_{p,2}$ is replaced by its approximation. The extension of the boundary field v_γ to the whole computational domain $\mathcal{D}$ is made with Method 10.2.3. Denoting by τ the final time of resolution of the Hamilton-Jacobi equation, we compare the following functions

$$\begin{aligned}
&\tau \mapsto J_i(\Omega_\tau),\\
&\tau \mapsto J_i(\Omega_0) + \tau {J_i}'(\Omega_0; v),\\
&\tau \mapsto J_i(\Omega_0) + \tau {J_i}'(\Omega_0; v) + \frac{1}{2}\tau^2 {J_i}''(\Omega_0; v, v),
\end{aligned} \tag{12.11}$$

where $\tilde{J}_i''$ is the approximation of the second-order shape derivative under consideration. Like in Section 11.3.2, we also plot the following curve for the consistency of the second-order approximation.

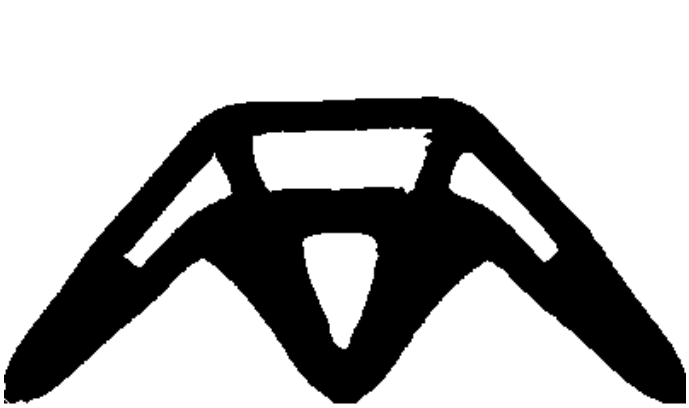
$$\tau \mapsto J_i(\Omega_\tau) - J_i(\Omega_0) - \tau {J_i}'(\Omega_0; v), \tag{12.12}$$

$$\tau \mapsto J_i(\Omega_\tau) - J_i(\Omega_0) - \tau {J_i}'(\Omega_0; v) - \frac{1}{2}\tau^2 \tilde{J}_i''(\Omega_0; v, v). \tag{12.13}$$

For both the compliance and least square displacement criteria, on the example of the arch, it appears that most of the time, the quadratic approximation of the objective in the direction taken by the chosen algorithm is rather consistent.

We start by the consistency of the second-order derivative of the compliance criterion. Figure 12.54 and Figure 12.55 correspond to the CG-Newton approximation; Figure 12.56 and Figure 12.57 correspond to the exact computation of the Hessian, Figure 12.58 and Figure 12.59 correspond to the LU-approximation and finally Figure 12.60 and Figure 12.61 correspond to the mass-lumping approximation. In these figures, we plot the curves (12.11) at different iterations of each algorithm. Looking at the consistency along the iterations, it is not possible to say that one method is more consistent that another. For each method, there are always shapes for which the method is consistent, and shapes for which the method is not really consistent.

The behavior of the approximations of the second-order derivative of the displacement criterion is more or less the same. However, sometimes it happens that the approximation of the second-order derivative has the wrong sign, like in Figure 12.62 for the displacement minimization. The potential lack of consistency may thus explain that on some examples in Section 12.3.3 or in Section 12.3.4 the approximated Newton's method with approximated Hessian may not be really efficient.



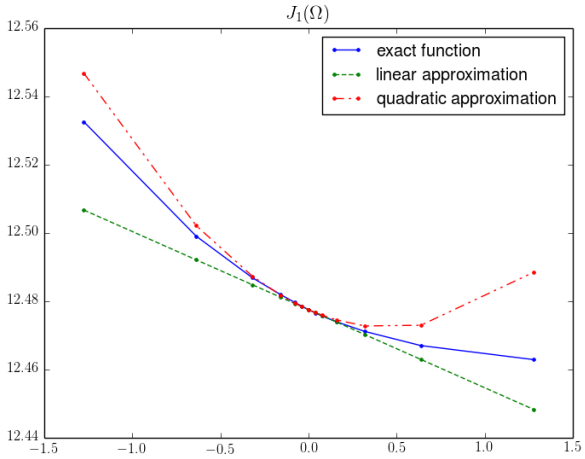
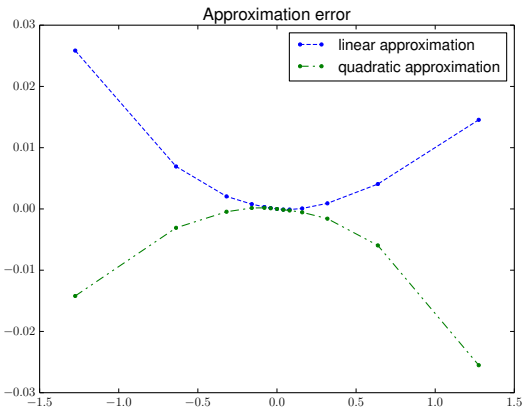


Figure 12.54: Shape at iteration 18 for the compliance minimization and consistency for the approximate CG-Newton method.



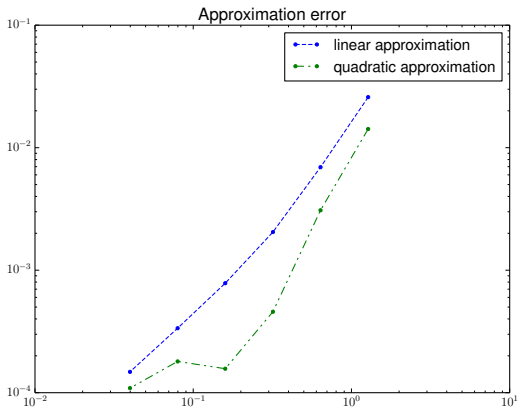


Figure 12.55: Plot of the curves (12.12) and (12.13) with respect to the time τ in linear (left) and log (right) scales for the iteration 18 with the CG-Newton method.

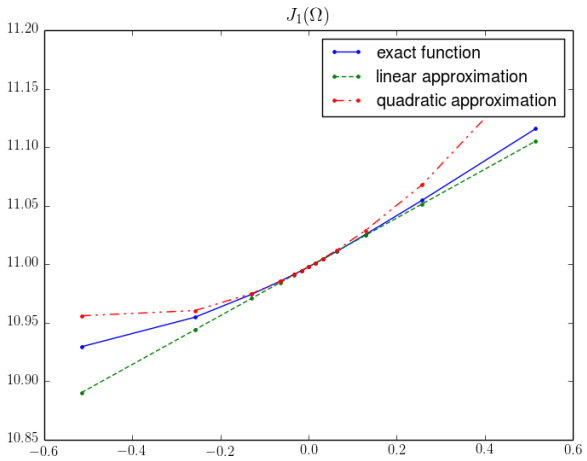
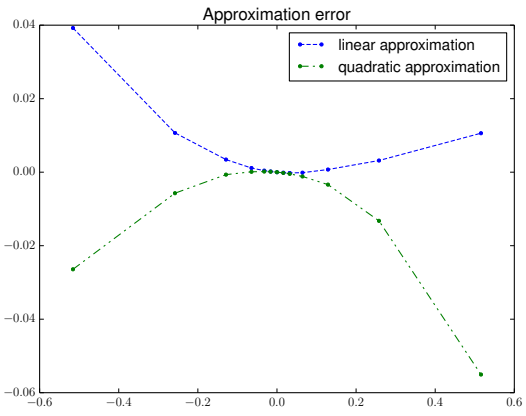


Figure 12.56: Shape at iteration 6 for the compliance minimization and consistency for the Newton method with exact Hessian.



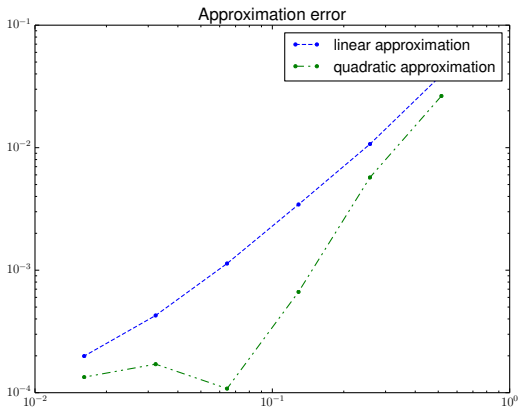


Figure 12.57: Plot of the curves (12.12) and (12.13) with respect to the time τ in linear (left) and log (right) scales for the iteration 6 with the Newton method.

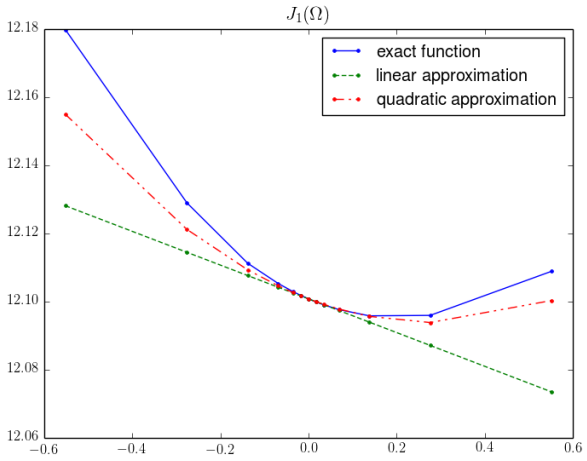
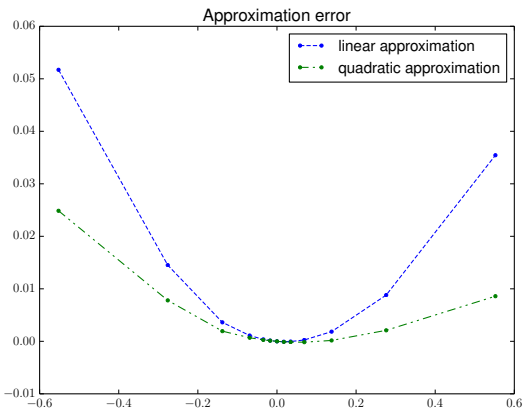


Figure 12.58: Shape at iteration 9 for the compliance minimization and consistency for the incomplete LU approximation on the Hessian.



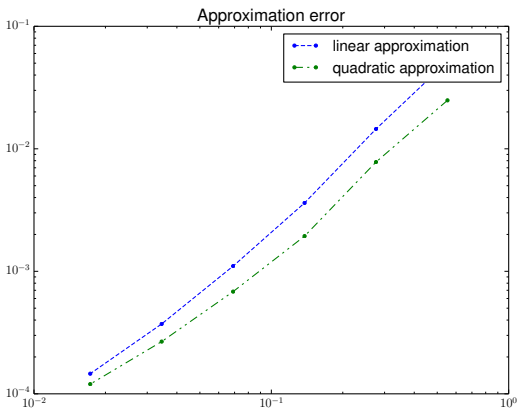
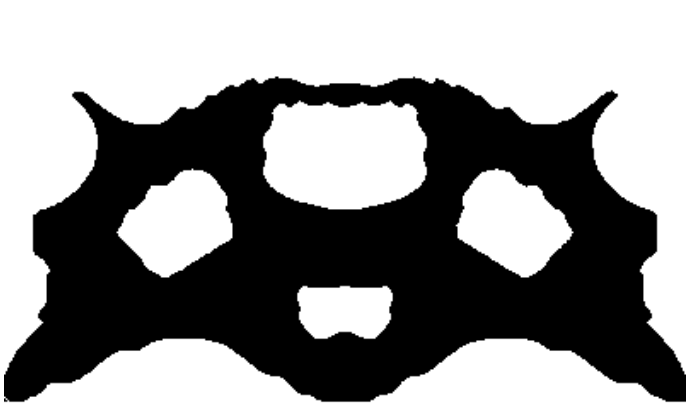


Figure 12.59: Plot of the curves (12.12) and (12.13) with respect to the time τ in linear (left) and log (right) scales for the iteration 9 with the incomplete LU approximation method.



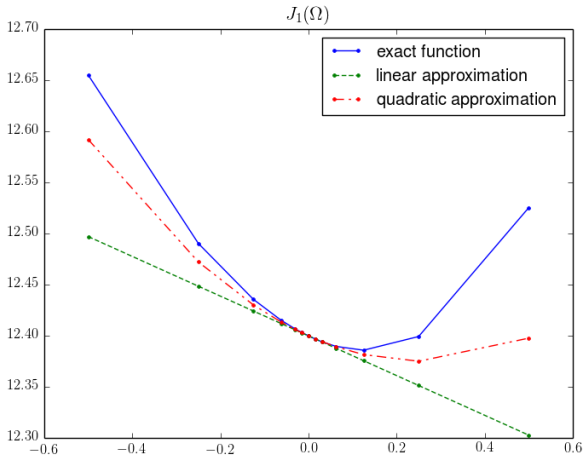
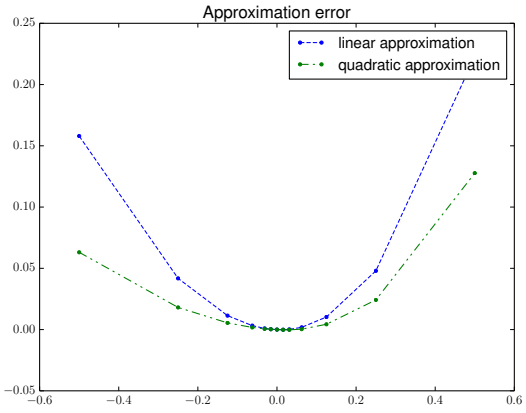


Figure 12.60: Shape at iteration 3 for the compliance minimization and consistency for the mass-lumping approximation on the Hessian.



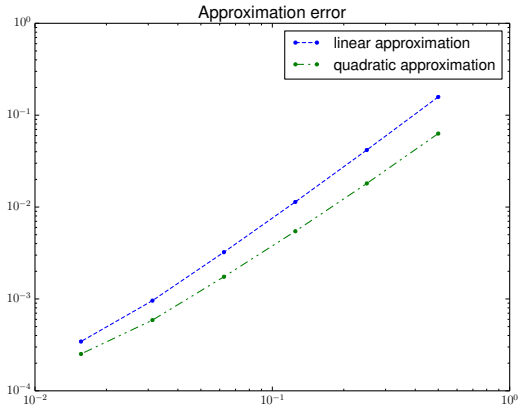


Figure 12.61: Plot of the curves (12.12) and (12.13) with respect to the time τ in linear (left) and log (right) scales for the iteration 3 with the mass-lumping approximation method.

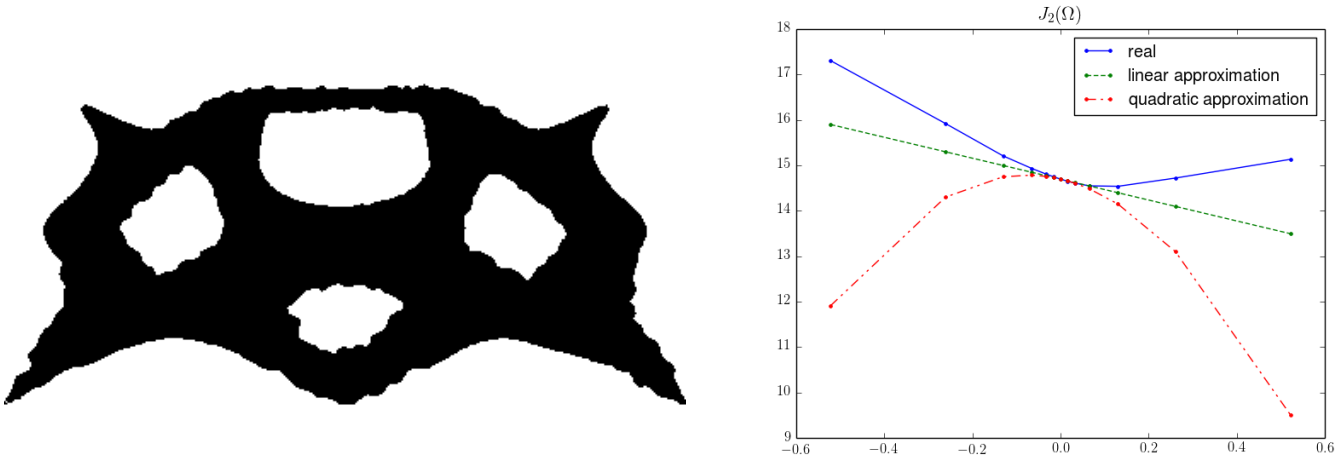


Figure 12.62: Shape at iteration 18 for the displacement minimization and consistency for the mass-lumping approximation on the Hessian.

12.3.3 Arch

Compliance - Initialization with holes

We first consider the same example as in Section 12.2.4 but with a Lagrange multiplier fixed to $\Lambda = 12$. We can first observe that the final shapes in Figure 12.63 and Figure 12.64 look all the same. The final values of the objective function in Table 12.15 are also similar.

We can also observe that all the second-order methods converge with fewer iterations than the gradient method, and that the Newton method is the one that needs the minimum number of iterations to converge. Thus, the approximation methods converge on this example with fewer iterations than the gradient method, and with more iterations than the Newton method. This is exactly the behavior we could expect from "good" approximation methods.

Figure 12.63: Final domains for the gradient (left) and Newton (right) algorithms for the compliance criterion on the arch problem of and initialization with holes.

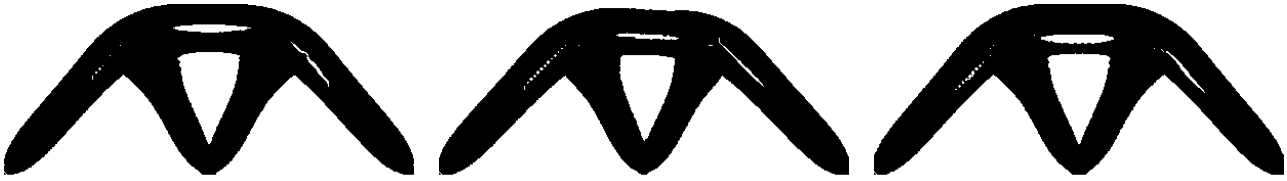


Figure 12.64: Final domains for the compliance criterion on the arch problem of and initialization with holes for the Newton algorithms with incomplete LU (left), mass-lumping (center) and CG-Newton (right) approximation.

Compliance - Initialization without holes

Now we take the example of the arch when the initial shape does not have any hole, like in Section 12.2.4. The Lagrange multiplier is fixed to $\Lambda = 6$. Here, all final shapes (displayed in Figure 12.67 and Figure 12.68) as well as the final values (reported in Table 12.16) of the objective function are similar. However, the behavior of the approximation methods is not exactly the same as in Section 12.3.3. We can indeed see that the incomplete LU approximation method converges even faster than the exact Newton method. The best final shape is obtained by the approximate CG-Newton method, but at the price of many iterations : it needs as many iterations than the gradient method. The convergence of $L(\Omega) - L(\Omega^{\times})$ as well as $L(\Omega) - L(\Omega_{\infty})$ are displayed in Figure 12.69 and Figure 12.70. In Figure 12.69 we can see that the objective value with this method is smaller than the one obtained with the gradient method only at the very last iterations. However, the speed of convergence for the approximate CG-Newton method (observed in this same figure) is rather important at the end of the optimization process. Thus, it could be interesting to first use the gradient at the beginning of the optimization and switch to a second-order approximate method for the last iterations.

	Iterations	Lagrangian	Gradient ratio	FE
Gradient	113	21.008393	4.339×10^{-2}	1082
Newton	30	21.016384	6.747×10^{-3}	77
CG-Newton	37	20.976707	2.289×10^{-2}	96
Mass-Lumping	85	21.035564	4.049×10^{-3}	251
LU	45	21.014296	2.166×10^{-3}	141

Table 12.15: Performance of each algorithm for the compliance criterion on the arch problem of and initialization with holes.

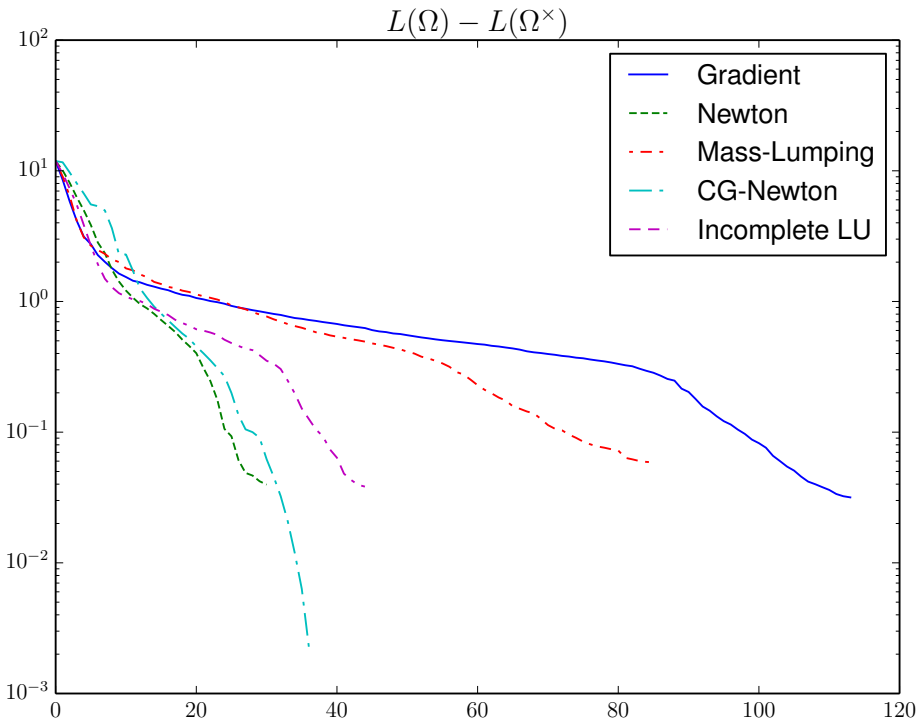


Figure 12.65: Convergence of $L(\Omega) - L(\Omega^\times)$ for the compliance criterion on the arch problem of and initialization with holes.

	Iterations	Lagrangian	Gradient ratio	FE
Gradient	32	16.248986	6.120×10^{-2}	283
Newton	15	16.250842	3.934×10^{-4}	77
CG-Newton	34	16.232746	4.657×10^{-2}	117
Mass-Lumping	60	16.261552	7.348×10^{-3}	301
LU	13	16.249782	8.036×10^{-4}	68

Table 12.16: Performance of each algorithm for the compliance criterion on the arch problem and initialization without holes.

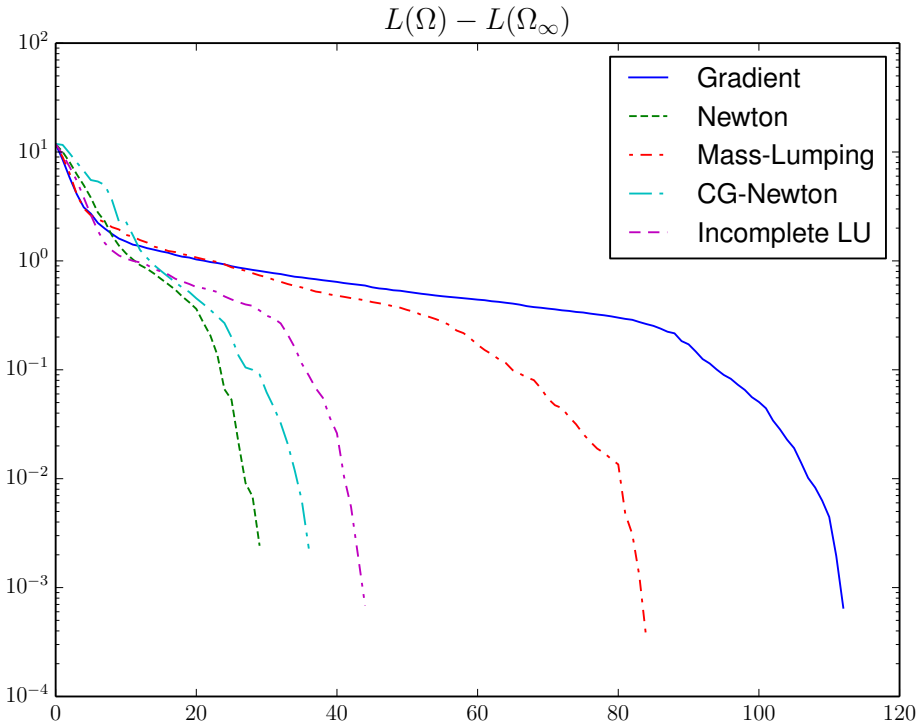


Figure 12.66: Convergence of $L(\Omega) - L(\Omega_\infty)$ for the compliance criterion on the arch problem of and initialization with holes.

Figure 12.67: Final domains for the gradient (left) and Newton (right) algorithms for the compliance criterion on the arch problem and initialization without holes.

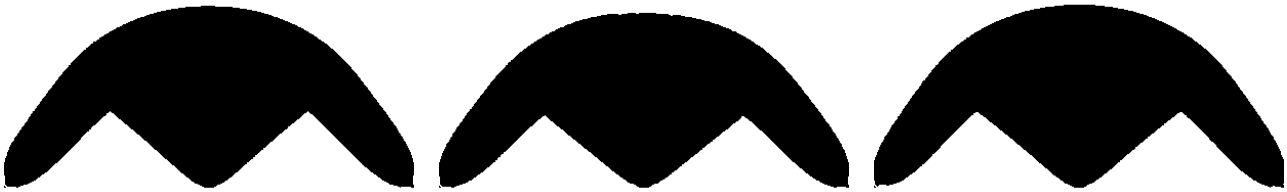


Figure 12.68: Final domains for the compliance criterion on the arch problem and initialization without holes for the Newton algorithms with incomplete LU (left), mass-lumping (center) and CG-Newton (right) approximation.

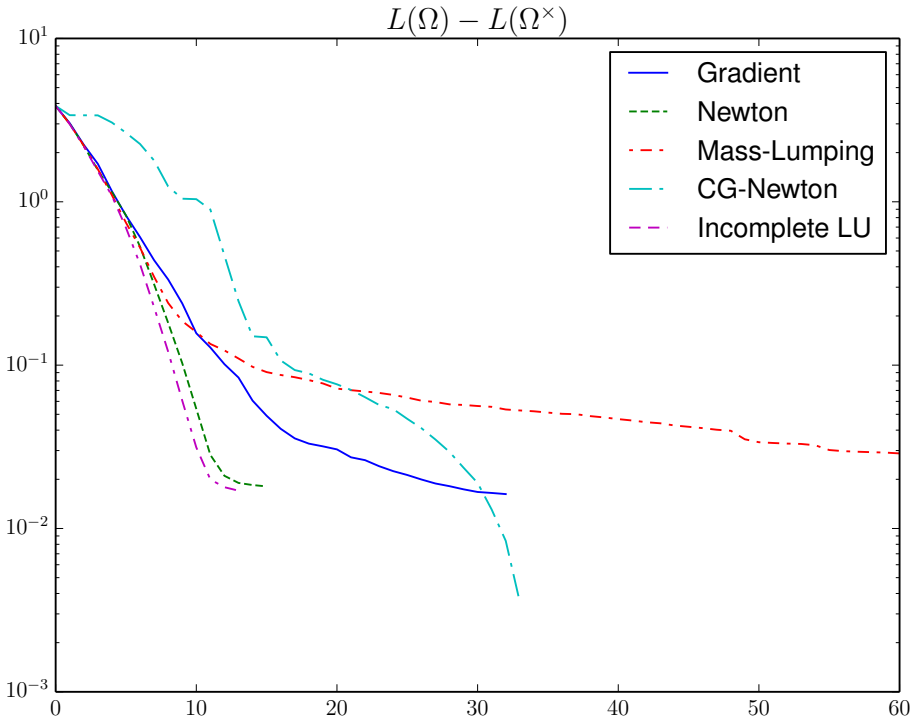


Figure 12.69: Convergence of $L(\Omega) - L(\Omega^\times)$ for the compliance criterion on the arch problem and initialization without holes.

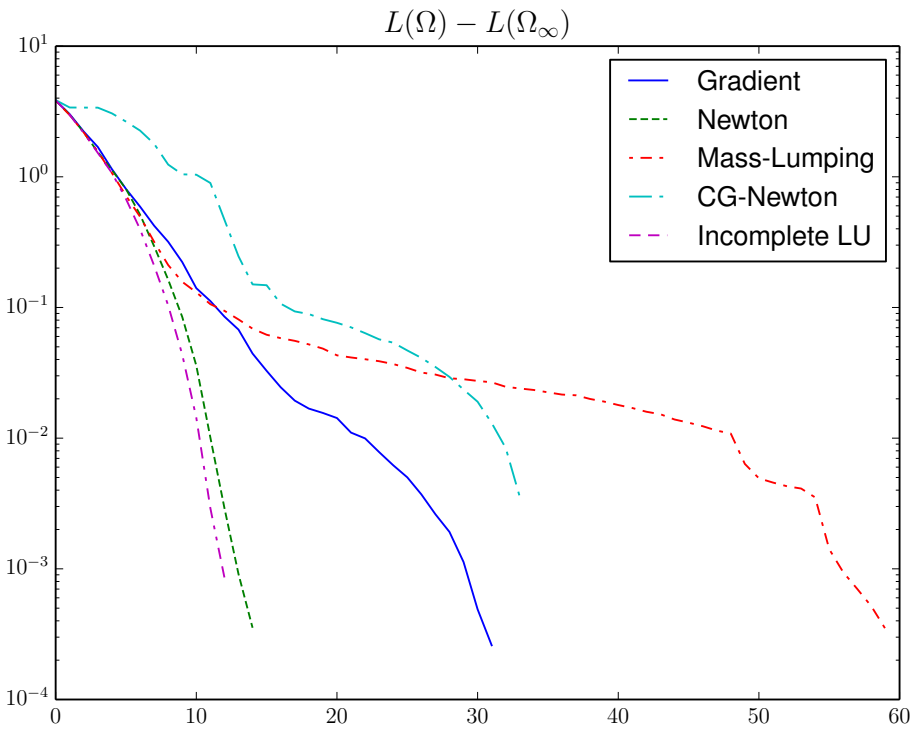


Figure 12.70: Convergence of $L(\Omega) - L(\Omega_\infty)$ for the compliance criterion on the arch problem and initialization without holes.

12.3.4 Cantilever

Compliance - Initialization with holes

In this last example, we take the case of the cantilever like in Section 12.2.5, and consider the compliance criterion. The Lagrange multiplier is fixed to $\Lambda = 0.5$. The final shapes are represented in Figure 12.71 and Figure 12.72, while the performance of the different algorithms are detailed in Table 12.17. The convergence curves of $L(\Omega) - L(\Omega^{\times})$ and $L(\Omega) - L(\Omega_{\infty})$ are respectively plotted in Figure 12.73 and Figure 12.74.

Figure 12.71: Final domains for the gradient (left) and Newton (right) algorithms for the compliance criterion on the cantilever problem and initialization with holes.

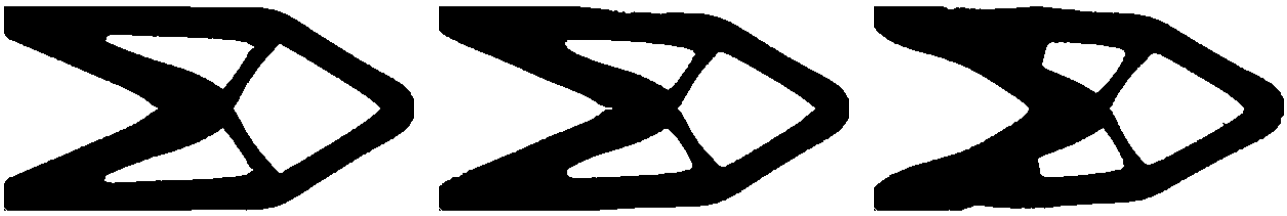


Figure 12.72: Final domains for the compliance criterion on the cantilever problem and initialization with holes for the Newton algorithms with incomplete LU (left), mass-lumping (center) and CG-Newton (right) approximation.

Here the Newton method with exact Hessian still converges with fewer iterations than the gradient method. Their resulting final shapes lead to similar values of the objective function. With the approximate CG-Newton method, the convergence is rather fast, but if we look at Figure 12.73, we observe that the speed is smaller than the gradient method. Moreover, the final shape has a greater value of the objective function. To the contrary, for the mass-lumping approximation, the convergence is rather slow, and the final shape is not as good as with the gradient method. Finally, the incomplete LU approximation leads to the best final shape (with the smallest value of the objective function), with fewer iterations than the gradient method.

	Iterations	Lagrangian	Gradient ratio	FE
Gradient	79	1.119020	6.048×10^{-2}	847
Newton	14	1.125227	-3.023×10^{-3}	58
CG-Newton	11	1.132158	5.422×10^{-1}	59
Mass-Lumping	103	1.120791	1.691×10^{-2}	480
LU	48	1.112262	5.381×10^{-3}	185

Table 12.17: Performance of each algorithm for the compliance criterion on the cantilever problem and initialization with holes.

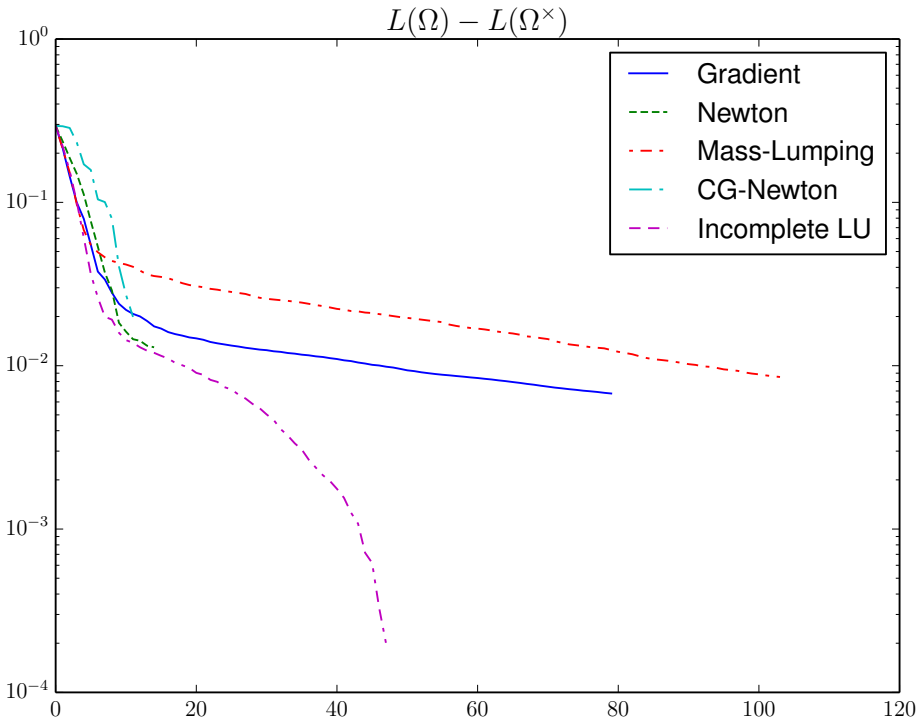


Figure 12.73: Convergence of $L(\Omega) - L(\Omega^\times)$ for the compliance criterion on the cantilever problem and initialization with holes.

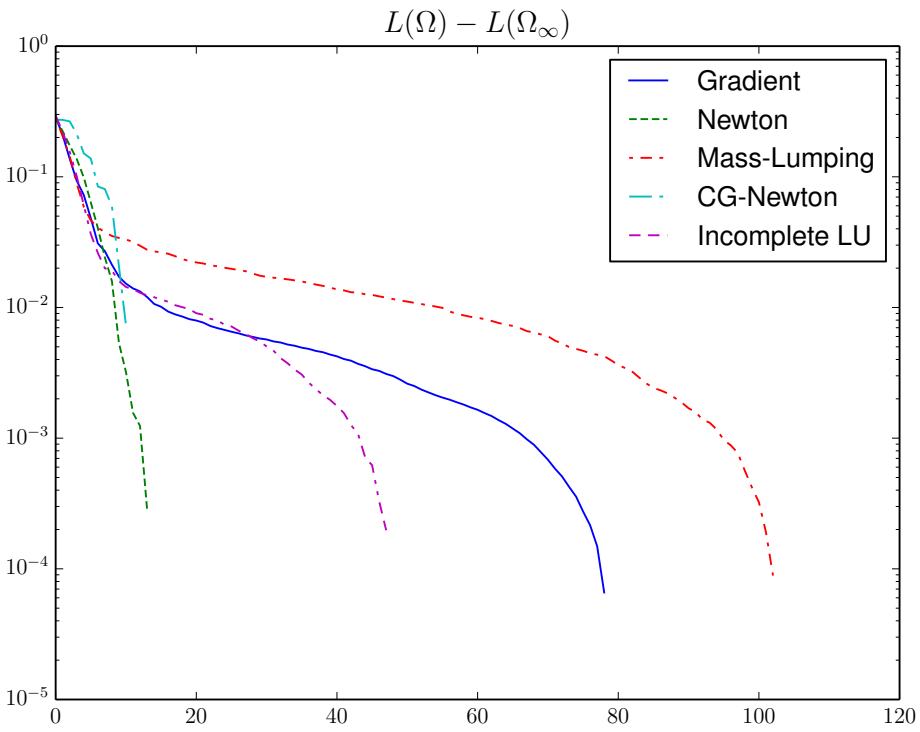


Figure 12.74: Convergence of $L(\Omega) - L(\Omega_\infty)$ for the compliance criterion on the cantilever problem and initialization with holes.

12.4 Numerical experiments in 3d

After considering the Newton method for two-dimensional examples, we would like to see if the method could also be efficient for three-dimensional cases. To that end, we will consider the cantilever example, the same as in Section 12.2.5 but in a three-dimensional version. These few 3-d numerical tests have been done thanks to a work with G. Michailidis, A. Faure and M. Albertelli that we thank for.

The method for computing boundary integrals (see Chapter 11) that was used in the previous section has not been implemented for three-dimensional cases. Therefore, we were only able to use approximations of the Dirac mass $\delta_{\partial\Omega}$ for computing these integrals. Thus, we have encountered the same consistency problems as already described. As a consequence, the directions found by the different algorithms did not always permit a decrease of the objective function. In such a case, the line search procedure leads to the termination of the optimization process whereas the shape may not be optimal at all.

For example, for the minimization of the compliance with the gradient method on the cantilever example, we plot the final shape in Figure 12.75. In this example, the mesh is of size $60 \times 30 \times 30$. We can observe that this shape could still be further optimized. This could be explained by either the coarsity of the mesh or by the lack of consistency of the numerical approach.

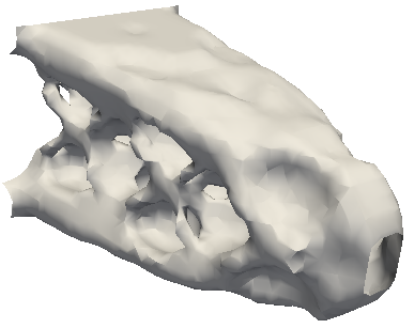


Figure 12.75: Cantilever

We also tried to use the Newton method on this same example. The final result of the optimization process does not reflect the efficiency of the Newton method because of the lack of consistency. However, it is still interesting to implement it to evaluate the computational cost of the evaluation of the exact second-order derivatives. In this example where the mesh is rather small, the computation of the second-order derivative takes about a couple of hours (on a computer with 12 cores with frequency 2.9GHz, and with 192Go of memory). Moreover, since the resulting Hessian is not sparse, the resolution of the quadratic problem (10.27) takes also a lot of time, still about an hour.

Our implementation can certainly be improved, in particular with parallelization of the code. The computation of the second-order derivative indeed implies to solve many times linear systems involving the same matrix. Thus, the parallelization of the code could really decrease the time cost of this computation.

However, if the mesh is to be refined for larger problems, the use of the exact second-order derivative seems to be out of reach. Thus, it again rises the question of approximating the Hessian in an efficient and consistent way.

12.5 Concluding remarks

In this chapter, we provide different numerical experiments on the proposed second-order optimization method. The analysis of the results leads to some general remarks :

PDE-free example

- The first numerical example is concerned with a PDE-free case when the objective function depends on the shape only through the integration domain. In this example, the second-order method outperforms the gradient-like method when the shape remains regular.

- When singularities develop on the boundary of the shape, the quadratic rate of convergence may be completely lost.

- However, even if the shape has singularities, changing the method of extension of the velocity from the boundary to the entire computational domain also improves the speed of convergence for both the gradient and Newton methods.

Accuracy and consistency

- It is important to use a redistanciation process in the level-set method. However, when it is used in the resolution of the advection equation, it may create artificial local optima which can cause the stopping of the optimization algorithm. This is mainly due to the fact that the redistanciation process may change the position of the boundary. Thus, it seems important to find either numerical schemes that keep the boundary unchanged - at least for distance functions - or to use redistanciation-free methods for the advection.

- The consistent and accurate computation of the derivatives is necessary for obtaining high rates of convergence. As it will be seen in Chapter 15, the accuracy required is higher on the first-order derivative than on the second-order one : in the particular example that is presented there, the error on the first-order derivative has to converge quadratically to zero. Thus, it means that the first step to get a quadratic rate of convergence is to have an accurate and consistent computation of the first-order derivatives.

Two-dimensional mechanical examples

- For the two-dimensional mechanical examples, the general behavior is not that pronounced. For all the presented examples, the Newton method converges more rapidly than the gradient method in the following sense : for a given same number of iterations the value of the objective function is lower with the Newton method than with the gradient method. Sometimes, the Newton method takes more iterations to converge, but in that case, the final value of the objective function is much smaller.

- However, even if the Newton method converges more rapidly than the gradient method, the rates of convergence remain always linear.

- This linear rate of convergence of the Newton method might come from the irregularity of the final shapes. In the present examples, when the shape is irregular, the second-order derivative is computed by cutting off the value of the curvature. This may not be the best way to use a second-order information when the shape is not regular. Thus, the question of using the second-order derivative on irregular shapes should be further investigated.

- The choice of extension between Method 10.2.1, Method 10.2.2 or Method 10.2.3 improves for most cases the convergence of the Newton method. Method 10.2.1 seems to be the one that finds the best shapes but at the cost of many iterations. Method 10.2.2 seems to offer a good compromise between the final value of the objective function and the number of iterations needed to reach convergence. However, the choice of extension method does not seem to have a major impact on the convergence of the gradient method.

Approximation of the second-order derivative

- We considered three different approximations of the second-order derivative of the compliance. On the three examples presented, only the incomplete LU approximation method seems to outperform the gradient method. The two other methods may sometimes be more efficient than the gradient method, but they do not seem as reliable as the incomplete LU approximation.

  However, let us mention different pros and cons of these three methods.

- There are two main advantages of the mass-lumping approximation : it is absolutely not expensive since it requires only the solution of one boundary-value problem, and secondly the resulting Hessian matrix is diagonal, meaning that the computation of a descent direction is also not expensive. However, in the way it was implemented and used, it does not seem to give efficient convergence results.

- The CG-Newton approximation seems to have better convergence properties than the mass-lumping approximation. This method has the advantage not to assemble the full Hessian of the criterion. From a discrete point of view, it only requires matrix-vector product computations : it is a matrix-free method. However, it is not clear how to use this method in the case when there are constraints in the optimization problem.

- The incomplete-LU approximation seems to perform quite well on the different examples. It has the advantage to keep the same structure as the exact second-order derivative : only the inverse of the stiffness matrix is approximated. However, the resulting Hessian matrix is still full, meaning that the resolution of the quadratic subproblem for computing a descent direction may remain expensive.

- Thus the question of approximating the second-order derivative of a mechanical criterion requires also further investigations.

Three-dimensional case

- The main conclusion of the three-dimensional numerical experiments is that the use of an exact second-order derivative of mechanical criteria is, for the moment, out of reach in three-dimensional cases. It would require new ingredients from high performance computing to become viable in terms of CPU and memory cost.

Part IV
Numerical analyses

"J'ai beaucoup travaillé.
Quiconque travaillera
comme moi pourra
faire ce que j'ai fait."
J.-S. Bach

Chapter 13

Full discretization of a shape optimization algorithm

Contents

The main goal of this chapter is to analyze a full discretization of a shape optimization algorithm with the level-set method. For the sake of clarity, the discussion focuses on the two-dimensional case ($d = 2$). The algorithm to be described here is a classic gradient method applied to a basic optimization problem : the minimization of the compliance of a structure with a fixed penalization on the volume. We detail the computation of boundary integrals (especially for the derivatives of the criteria) with the approximation of a Dirac mass $\delta_{\partial\Omega}$ (see Section 11.1). This is not the most consistent and accurate way of computing boundary integrals, but this is the most simple (see Chapter 11). For other criteria, all needed ingredients are given with the simple example here. It is indeed explained how to compute solutions to boundary-value problems, as well as boundary and volume integrals, which should suffice to compute any relevant criteria.

At first we recall the main background for optimization of the compliance under a volume constraint, and introduce some useful notation. Then we list all steps of the shape optimization algorithm we consider, and detail all corresponding numerical computations.

13.1 A simple case

13.1.1 Model problem

We consider the framework of the *elastic model* described in Section 2.1.5 that we recall briefly. Let $\Omega \subset \mathbb{R}^d$ be a bounded open set. Its boundary $\partial\Omega$ is assumed to be regular, and Γ_D, Γ_N, Γ is a partition of it : $\partial\Omega = \Gamma_D \cup \Gamma_N \cup \Gamma$. The two boundary parts Γ_D and Γ_N are fixed whereas Γ is the part to be optimized. We introduce a working domain $\mathcal{D} = [0,1]^2$ and impose that every admissible shape Ω is included in $\mathcal{D}$: $\Omega \subset \mathcal{D}$. For a given $g \in H^2(\mathcal{D};\mathbb{R}^d)$ such that $g = 0$ on Γ, we consider the following boundary-value problem, with unknown $u \in H^1(\Omega;\mathbb{R}^d)$.

$$\left\{\begin{array}{rcll} -\mathrm{div}\big(A\varepsilon(u)\big) & = & 0 & \Omega, \\ u & = & 0 & \Gamma_D, \\ A\varepsilon(u)\mathbf{n} & = & g & \Gamma_N, \\ A\varepsilon(u)\mathbf{n} & = & 0 & \Gamma, \end{array}\right. \tag{13.1}$$

where

$$A\xi = 2\mu\xi + \lambda \mathrm{Tr}(\xi)\mathrm{Id}, \quad \varepsilon(u) = \frac{1}{2}\Big(\nabla u + \nabla u^T\Big).$$

In order to compute u we introduce the variational space $\mathcal{V} = \{w \in H^1(\Omega;\mathbb{R}^d) \mid u = 0 \text{ on } \Gamma_D\}$. The variational formulation of (13.1) is then to find $u \in \mathcal{V}$ such that

$$\forall w \in \mathcal{V},\ \int_\Omega A\varepsilon(u) : \varepsilon(w) = \int_{\Gamma_N} g \cdot w.$$

The compliance and volume respectively read

$$J(\Omega) = \int_\Omega A\varepsilon(u) : \varepsilon(u), \quad V(\Omega) = \int_\Omega 1.$$

The shape derivatives in the direction $\theta \in C^{1,\infty}\left(\mathbb{R}^d;\mathbb{R}^d\right)$ are

$$J'(\Omega;\theta) = -\int_\Gamma A\varepsilon(u) : \varepsilon(u)\,(\theta\cdot\mathbf{n}), \quad V'(\Omega;\theta) = \int_\Gamma (\theta\cdot\mathbf{n}).$$

We denote the shape gradients of the compliance and the volume by

$$j_J = -A\varepsilon(u) : \varepsilon(u), \quad j_V = 1. \tag{13.2}$$

For $\Lambda > 0$ a fixed Lagrange multiplier, the aim is to minimize

$$L(\Omega) = J(\Omega) + \Lambda V(\Omega).$$

The shape Ω is represented via a level-set function $\phi : \mathcal{D} \to \mathbb{R}$ such that

$$\begin{cases} \phi(x) < 0 & x \in \Omega, \\ \phi(x) = 0 & x \in \partial\Omega, \\ \phi(x) > 0 & x \in \mathcal{D}\backslash(\Omega \cap \partial\Omega). \end{cases}$$

13.1.2 Notations

Let $N_x, N_y \in \mathbb{N}^*$ and $\Delta x = \dfrac{1}{N_x}$, $\Delta y = \dfrac{1}{N_y}$. The square $\mathcal{D} = [0,1]^2$ is endowed with a Cartesian grid $\mathcal{Q}_h$ with steps Δx in the x-direction and Δy in the y-direction. For a numerical quantity ϕ defined at the vertices of the grid, and for any $i, j \in \mathbb{Z}$, we denote by $\phi_{i,j}$ the value assigned to the node $x_{ij} := (i\Delta x, j\Delta y)$. We also recall the following finite differences quantities

$$D^{+x}_{i,j}\phi = \frac{\phi_{i+1,j} - \phi_{i,j}}{\Delta x}, \quad D^{-x}_{i,j}\phi = \frac{\phi_{i,j} - \phi_{i-1,j}}{\Delta x},$$
$$D^{+y}_{i,j}\phi = \frac{\phi_{i,j+1} - \phi_{i,j}}{\Delta y}, \quad D^{-y}_{i,j}\phi = \frac{\phi_{i,j} - \phi_{i,j-1}}{\Delta y},$$
$$D^{\pm x}_{i,j}\phi = \frac{\phi_{i+1,j} - \phi_{i-1,j}}{2\Delta x}, \quad D^{\pm y}_{i,j}\phi = \frac{\phi_{i,j+1} - \phi_{i,j-1}}{2\Delta y}.$$

and

$$\nabla^+_{i,j}\phi = \left[\max\Big(\max(D^{-x}_{i,j}\phi,0); -\min(D^{+x}_{i,j}\phi,0)\Big)^2 + \max\Big(\max(D^{-y}_{i,j}\phi,0); -\min(D^{+y}_{i,j}\phi,0)\Big)\right]^{\frac{1}{2}},$$
$$\nabla^-_{i,j}\phi = \left[\max\Big(\max(D^{+x}_{i,j}\phi,0); -\min(D^{-x}_{i,j}\phi,0)\Big)^2 + \max\Big(\max(D^{+y}_{i,j}\phi,0); -\min(D^{-y}_{i,j}\phi,0)\Big)\right]^{\frac{1}{2}}.$$

13.2 Algorithm

The level-set function ϕ is from now on the variable to be optimized. Starting from a function ϕ^p we describe the step $\phi^p \to \phi^{p+1}$.

Algorithm 13.2.1. *Optimization algorithm*

1. *Initialization : set $\phi_{i,j} = \phi^p_{i,j}$ and choose a threshold $\eta > 0$.*
2. *Redistanciation : apply Algorithm 13.2.2 to ϕ (see Section 13.2.1).*
3. *Computation of the density function (see Section 13.2.2).*
4. *Evaluation of the criterion L^h and its gradients ∇L^h, (see Section 13.2.3).*
5. *Computation of a descent direction $\tilde{W}$ (see Section 13.2.4).*
6. *Line search (see Section 13.2.6).*

13.2.1 Redistanciation of the level-set

The objective of the redistanciation is to make the level-set function close to a signed distance function. The level-set ϕ^p is replaced by $\phi(T,\cdot)$ with $T=\frac{5}{2}\Delta x$, the solution to

$$\begin{cases} \partial_t\phi(t,x)+s(\phi)(t,x)\big(\,|\nabla_x\phi(t,x)|-1\big)=0, \\ \phi(0,x)=\phi^p(x), \end{cases} \tag{13.3}$$

with s being the sign function. The numerical scheme for this step is the same as in Section 4.2.2 :

Algorithm 13.2.2. *Redistanciation algorithm*

1. *Initialization : set* $\phi^0_{i,j}=\phi^p_{i,j}$ *and* $s_{i,j}=\dfrac{\phi^n_{i,j}}{\big(\phi^n_{i,j}\big)^2+0.5(\Delta x^2+\Delta y^2)N_{i,j}+10^{-8}(\Delta x^2+\Delta y^2)}$

2. *For* n *in* $0,\cdots,4$

   - $N_{i,j}=\dfrac{1}{2}\Big(\big(D^{+x}_{i,j}\phi^n\big)^2+\big(D^{-x}_{i,j}\phi^n\big)^2+\big(D^{+y}_{i,j}\phi^n\big)^2+\big(D^{-y}_{i,j}\phi^n\big)^2\Big)$,
   - $dt=\dfrac{1}{2}\min\big(\Delta x,\Delta y\big)$,
   - $\phi^{n+1}_{i,j}=\phi^n_{i,j}-dt\Big(\max(s_{i,j},0)\nabla^+_{i,j}\phi^n+\min(s_{i,j},0)\nabla^-_{i,j}\phi^n-s_{i,j}\Big)$.
   - *Set* $n:=n+1$.

3. *Set* $\phi_{i,j}=\phi^4_{i,j}$.

We can notice that the approximation of the sign function s corresponds to the discretization of $s_{2,\varepsilon}$ in Section 11.1. We could also have chosen another approximation for the sign function. In this algorithm, the time-step dt is chosen as $dt=\frac{1}{2}\min(\Delta x,\Delta y)$ in order to ensure that the CFL condition is satisfied. The loop corresponding to step 2 of this algorithm is only performed with $n\in[\![0,4]\!]$ for numerical reasons. The redistanciation is used for the resolution of the Hamilton-Jacobi advection equation. It prevents the level-set from being too steep or too flat in the vicinity of the boundary. Finding the stationary solution of (13.3) is time consuming and not necessary, since the first iteration of the loop (step 2) correspond to the redistanciation in the neighborhood of of the boundary. Finally, taking $n\in[\![0,4]\!]$ corresponds to take a final time $T=5dt=\frac{5}{2}\Delta x$ for (13.3).

13.2.2 Density function, Dirac mass on the boundary

We denote $K\in\mathcal{Q}_h$ an element of the Cartesian grid and introduce the finite element space $\mathbb{Q}_1\big(\mathcal{Q}_h;\mathbb{R}^2\big)$ of dimension N_h. We introduce a density function ρ defined on each element of the mesh, that plays the role of the characteristic function of Ω. It is computed with the help of ϕ on each element by

$$\rho_K=\begin{cases} 1 & \text{if } K\subset\Omega, \\ 0 & \text{if } K\subset\mathcal{D}\backslash\Omega, \\ \dfrac{|K\cap\Omega|}{|K|} & \text{else.} \end{cases}$$

We consider the approximation of the sign function given by $s_{2,\varepsilon}$ defined by (11.5), with $\varepsilon=10^{-8}(\Delta x^2+\Delta y^2)$ (we could also have chosen $s_{1,\varepsilon}$). Thus, the discrete sign function reads

$$s_{i,j}=\frac{\phi_{i,j}}{\phi^2_{i,j}+0.5(\Delta x^2+\Delta y^2)N_{i,j}+10^{-8}(\Delta x^2+\Delta y^2)}, \tag{13.4}$$

where

$$N_{i,j}=\frac{1}{2}\Big(\big(D^{+x}_{i,j}\phi\big)^2+\big(D^{-x}_{i,j}\phi\big)^2+\big(D^{+y}_{i,j}\phi\big)^2+\big(D^{-y}_{i,j}\phi\big)^2\Big).$$

The Dirac function is approximated by $\delta_{2,\varepsilon}=\frac{1}{2}|\nabla s_{2,\varepsilon}|$, and therefore we take

$$\delta_{i,j}=\frac{1}{2}\Big(\,\big(D^{\pm x}_{i,j}s\big)^2+\big(D^{\pm y}_{i,j}s\big)^2\,\Big)^{\frac{1}{2}}.$$

13.2.3 Evaluation of the criteria and their shape gradients

Volume

The volume of the shape and its shape derivative are defined by

$$V(\Omega) = \int_\Omega 1, \quad V'(\Omega;\theta) = \int_{\partial\Omega} \theta \cdot \mathbf{n}.$$

The discrete approximation of the volume is

$$V^h = \sum_{K \in \mathcal{Q}_h} \rho_K \Delta x \Delta y.$$

The discrete approximation of the shape gradient - $\nabla V^h \in \mathbb{R}^{N_h}$ - is defined by

$$\forall i,j \in \mathbb{Z},\ \nabla V^h_{i,j} = 1.$$

Compliance

With u solution to (13.1), the compliance is defined by

$$J(\Omega) = \int_\Omega A\varepsilon(u) : \varepsilon(u), \quad J'(\Omega;\theta) = -\int_\Gamma (\theta \cdot \mathbf{n})\ A\varepsilon(u) : \varepsilon(u).$$

We do exactly like in Section 8.2. With $A_{\mathcal{Q}_h}$ being the elementary stiffness matrix of $\mathbb{Q}_1$, and G the components of the source term in $\mathbb{Q}_1$, we search $U \in \mathbb{R}^{N_h}$ - the components of the interpolation of u in $\mathbb{Q}_1$ - such that for all $P \in \mathbb{R}^{N_h}$

$$\sum_{K \in \mathcal{Q}_h} \rho_K {U_{|K}}^T A_{\mathcal{Q}_h} P_{|K} = G^T P.$$

Once U is computed the compliance is obtained with

$$J^h = \sum_{K \in \mathcal{Q}_h} \rho_K {U_{|K}}^T A_{\mathcal{Q}_h} U_{|K}.$$

The numerical shape gradient of the compliance, $\nabla J^h \in \mathbb{R}^{N_h}$, is computed with

$$\nabla J^h_{i,j} = -\frac{1}{n_{ij}} \sum_{K \supset x_{i,j}} \rho_K {U_{|K}}^T A_{\mathcal{Q}_h} U_{|K},$$

where $n_{i,j}$ is the number of elements containing $x_{i,j}$.

Lagrangian

We obviously have

$$L^h = J^h + \Lambda V^h, \quad \nabla L^h_{i,j} = \nabla J^h_{i,j} + \Lambda \nabla V^h_{i,j}.$$

13.2.4 Descent direction

Here we aim at computing a descent direction $\tilde{W} \in \mathbb{R}^{N_h}$. We start with the definition of the set of nodes of the boundary.

Boundary

Let B_Γ be the set of nodes that are in an element crossed by the boundary. A node is then in B_Γ if it is in a cell where the sign of the level-set function changes on the four vertices. We can approximate it like :

$$B_\Gamma = \left\{ x_{i,j} \,|\, \exists K \in \mathcal{Q}_h,\ x_{i,j} \in K \text{ and } \left| \sum_{x_{k,l} \in K} s_{k,l} \right| \neq 4 \right\},$$

where s is defined by (13.4).

Trace on the boundary

Now we compute a direction $W \in \mathbb{R}^{N_h}$ that is set to zero outside B_Γ. The numerical resolution of the gradient equation (10.28) consists in finding $W \in \mathbb{R}^{N_h}$ such that $W_{i,j} = 0$ if $x_{i,j} \notin B_\Gamma$ and that is solution to

$$\min_{|W_{i,j}| \leq v_M} \frac{1}{2} \sum_{x_{i,j} \in B_\Gamma} \delta_{i,j} W^2_{i,j} \Delta x \Delta y + \sum_{x_{i,j} \in B_\Gamma} \delta_{i,j} \nabla L^h_{i,j} W_{i,j} \Delta x \Delta y. \tag{13.5}$$

It now remains to compute an extension of W from B_Γ to the whole computational domain. We detail here the discrete version of the extensions presented in Section 10.2.

Extension with Method 10.2.1

We want to find $\tilde{W} \in \mathbb{R}^{N_h}$ such that $\partial_{\mathbf{n}}\tilde{W} = 1$ and $\tilde{W} = W$ on B_Γ. This consists in solving

$$\begin{cases} \partial_t \tilde{w}(t,x) + s(\phi)(t,x)\big(\partial_{\mathbf{n}}\tilde{w}(t,x) - 1\big) = 0, \\ \tilde{w}(0,x) = w(x). \end{cases} \tag{13.6}$$

From a numerical point of view, we do

Algorithm 13.2.3. *Extension*

1. *Initialization : set* $W^0_{i,j} = W_{i,j}$, $s_{i,j} = \dfrac{\phi^n_{i,j}}{\big(\phi^n_{i,j}\big)^2 + 0.5(\Delta x^2 + \Delta y^2)N_{i,j} + 10^{-8}(\Delta x^2 + \Delta y^2)}$

   and $S_{i,j} = \left(\big(\nabla^{\pm x}_{i,j} s\big)^2 + \big(\nabla^{\pm y}_{i,j} s\big)^2 + 10^{-3}\right)^{\frac{1}{2}}$.

2. *For* n *in* $0, \cdots, 20$

   - $N_{i,j} = \dfrac{1}{2}\Big(\big(D^{+x}_{i,j}\phi\big)^2 + \big(D^{-x}_{i,j}\phi\big)^2 + \big(D^{+y}_{i,j}\phi\big)^2 + \big(D^{-y}_{i,j}\phi\big)^2\Big)$,
   - $dt = \dfrac{1}{2}\min\big(\Delta x, \Delta y\big)$
   - *If* $x_{i,j} \notin B_\Gamma$ *set*

$$W^{n+1}_{i,j} = W^n_{i,j} - dt\left(\frac{\min(D^{\pm x}_{i,j}\phi_{i,j}, 0)}{S_{i,j}} D^{+x}_{i,j} W^n + \frac{\max(D^{\pm x}_{i,j}\phi_{i,j}, 0)}{S_{i,j}} D^{-x}_{i,j} W^n \right.$$
$$\left. + \frac{\min(D^{\pm y}_{i,j}\phi_{i,j}, 0)}{S_{i,j}} D^{+y}_{i,j} W^n + \frac{\max(D^{\pm y}_{i,j}\phi_{i,j}, 0)}{S_{i,j}} D^{-y}_{i,j} W^n - s_{i,j}\right).$$

   - *Set* $n := n + 1$.

3. *Set* $\tilde{W}_{i,j} = \tilde{W}^{20}_{i,j}$.

In this algorithm, the loop at step 2 is performed with $n \in [\![0, 20]\!]$. This amounts to extend the velocity W only in a neighborhood of the boundary. The extended velocity will be used for the resolution of the advection equation (see Section 13.2.5). In the advection, there is a bound on the number of iterations that can be performed. This is suggested by the fact that the boundary of the shape Ω^{p+1} given by ϕ^{p+1} should stay in a neighborhood of the boundary of Ω^p. The propagation of the boundary (or zero level-set) is directed by the values of the extended velocity on this front. Thus, only the values of the velocity in this neighborhood matter for the advection. Therefore, there is no need to extend the velocity in the whole computational domain, which would be time consuming.

Extension with Method 10.2.3

For a scalar quantity v_γ defined on the boundary $\partial\Omega$, the idea is to find v such that for all $w \in H^1(\mathbb{R}^2; \mathbb{R})$

$$\eta \int_{\mathbb{R}^2} \nabla v \nabla w + \int_{\partial\Omega} vw = \int_{\partial\Omega} v_\gamma w.$$

From a numerical point of view, we take $\eta = 4\Delta x \Delta y$ in order to ensure that the regularization by the Laplace operator will be efficient at least in a neighborhood of the boundary of size $2\sqrt{\Delta x \Delta y}$. We also respectively denote M_Δ, M the matrix of the Laplace operator and the mass matrix in $\mathbb{Q}_1(\mathcal{Q}_h, \mathbb{R}^2)$, i.e for v, w in $\mathbb{Q}_1(\mathcal{Q}_h, \mathbb{R}^2)$ with components $W_1, W_2 \in \mathbb{R}^{N_h}$

$$W_1{}^T M_\Delta W_2 = \int_{\mathbb{R}^2} \nabla w_1 \nabla w_2, \quad W_1{}^T M W_2 = \int_{\mathbb{R}^2} w_1 w_2.$$

Now let assume that we have $W \in \mathbb{R}^{N_h}$ solution to (13.5). With the help of the approximated Dirac function (see Section 13.2.2) we build a diagonal matrix D such that for $m = i\,N_x + j$, $D_{m,m} = \sqrt{\delta_{i,j}}$. Now we are looking for $\tilde{W} \in \mathbb{R}^{N_h}$ such that for all $P \in \mathbb{R}^{N_h}$,

$$P^T\Big(\eta M_\Delta + D\,M\,D\Big)\tilde{W} = P^T D\,M\,D\,W.$$

13.2.5 Advection through the Hamilton-Jacobi equation

The update of the shape in the direction $\tilde{w}$ is performed through the Hamilton-Jacobi equation, that is

$$\begin{cases} \partial_t \phi(t,x) + \tilde{w}(x)\,|\nabla_x \phi(t,x)| = 0, \\ \phi(0,x) = \phi^p(x). \end{cases} \tag{13.7}$$

At first we compute the time and time step parameters for the transport equation. The parameter ΔT is a priori given.

Algorithm 13.2.4. *Hamilton-Jacobi transport equation*

1. *Initialization : set* $\phi^0_{i,j} = \phi_{i,j}$.
2. *Set* $dt_0 = \dfrac{1}{2}\min\big(\Delta x, \Delta y\big)$ *and* $N = \left\lceil \dfrac{\max(\tilde{W})\Delta T}{dt_0} \right\rceil$.
3. *Set* $dt = \dfrac{\Delta T}{N}$ *and* $N = \min(N, 10)$.
4. *For* n *in* $0, \cdots, N$
   - *if* $n \equiv 0 \pmod 5$, *apply Algorithm 13.2.2 to* ϕ.
   - $\phi^{n+1}_{i,j} = \phi^n_{i,j} + dt\Big(\max(\tilde{W}_{i,j}, 0)\nabla^+_{i,j}\phi + \min(\tilde{W}_{i,j}, 0)\nabla^-_{i,j}\phi\Big)$.
   - *Set* $n := n + 1$.
5. *Apply Algorithm 13.2.2 to* ϕ.

13.2.6 Line search

We describe the line search procedure in a direction $\tilde{W}$.

Algorithm 13.2.5. *Line search*

1. *Set* $\Delta T = 1$.
2. *Let* ϕ_T *be the result of the transport equation (see Section 13.2.5) for* ΔT, W, ϕ.
3. *Compute the density function* ρ_T *for* ϕ_T *(see Section 13.2.2).*
4. *Evaluate the criterion* L^h_T *for* ρ_T *(see Section 13.2.3).*
5. *If* $\Delta T \leq \eta$ *terminate the optimization. Else If* $L^h_T > L^h + 10^{-3}\Delta T \sum_{i,j\in\mathbb{Z}} \nabla L^h_{i,j} W_{i,j}\delta_{i,j}$, *set* $\Delta T := \dfrac{\Delta T}{2}$ *and go to 2. Else set* $\phi = \phi_T$ *and stop.*

Chapter 14

Analysis of numerical schemes for the level-set method

Contents

The numerical scheme (see Section 4.2.2 or also Algorithm 13.2.4) used for solving the Hamilton-Jacobi equation is well-known for being very diffusive. The eikonal equation can be solved with the same scheme or with another method, namely the fast-marching method. In view of the Newton method for optimization we aim in this section at analyzing the numerical effect of these schemes on an optimization method. At first we recall the scheme for solving the Hamilton-Jacobi and the eikonal equation in 1D. The choice to work in dimension 1 allows us to make all calculations possible explicit. It appears that even with a basic PDE-free example, a gradient method for shape optimization may not have the expected behavior and may not converge. Secondly we consider the case of the fast-marching method and present an example to show that computing distances is not necessary a stable procedure, meaning distance functions are not necessarily fixed points of the method.

14.1 The one-dimensional transport equation

At first we recall the scheme for the resolution of the transport equation in the one-dimensional case.

14.1.1 The Hamilton-Jacobi transport equation

In this section we discuss the numerical analysis of the discretization scheme for the following Hamilton-Jacobi equation in space dimension one :

$$\begin{cases} \partial_t \phi + v|\nabla \phi| = 0 & \text{on } [0,T] \times [0,1], \\ \phi(0,\cdot) = \phi_0 & \text{on } \mathbb{R}, \end{cases} \tag{14.1}$$

with periodic boundary conditions, where the scalar field v and the initial condition ϕ_0 are given.

Notation. *Let $N \in \mathbb{N}^\star$ and $h = \frac{1}{\mathbb{N}}$. The segment $[0,1]$ is endowed with a Cartesian grid with steps h. For a numerical quantity ϕ defined at the vertices of the grid and $i \in [\![0, N]\!]$, ϕ_i denotes the value assigned to the node $x_i := ih$. In this context we introduce the respective upwind and downwind difference quantities :*

$$D_i^+ \phi = \frac{\phi_{i+1} - \phi_i}{h}, \quad D_i^- \phi = \frac{\phi_i - \phi_{i-1}}{h}. \tag{14.2}$$

For solving the Cauchy problem, a usual approximation consists in solving the following explicit finite difference scheme

$$\begin{cases} \forall n \in \mathbb{N}, i \in \mathbb{Z},\ \phi_i^{n+1} = \phi_i^n - \Delta t \left(\max(v_i;0) \nabla_i^+ \phi^n + \min(v_i;0) \nabla_i^- \phi^n \right), \\ \forall i \in \mathbb{Z},\ \phi_i^0 = \phi_0(ih). \end{cases} \tag{14.3}$$

The upwind and downwind discretizations of $|\nabla\phi|$, denoted respectively by $\nabla_i^+\phi$ and $\nabla_i^-\phi$, are defined by

$$\nabla_i^+\phi = \max\left(\max(D_i^-\phi;0); -\min(D_i^+\phi;0)\right), \tag{14.4}$$
$$\nabla_i^-\phi = \max\left(\max(D_i^+\phi;0); -\min(D_i^-\phi;0)\right), \tag{14.5}$$
$$\tag{14.6}$$

if $i \in [\![1, N-1]\!]$ and

$$\nabla_0^+\phi = -\min(D_0^+\phi;0), \tag{14.7}$$
$$\nabla_N^-\phi = -\min(D_N^-\phi;0). \tag{14.8}$$
$$\tag{14.9}$$

Now for $i \in [\![1, N-1]\!]$ we detail in Figure 14.1 all the possible configurations for ϕ_{i-1}, ϕ_i, ϕ_{i+1}, $\nabla_i^+\phi$ and $\nabla_i^-\phi$:

14.1.2 The eikonal equation

When the level-set function is expected to be a signed distance function, the resolution of the Hamilton-Jacobi transport equation needs to be coupled with a redistanciation procedure. This is described in Algorithm 13.2.4. The redistanciation of the level-set prevents it from being either too flat or too steep which could be source of numerical errors.

In the one-dimensional case, the scheme for the redistanciation algorithm 13.2.2 reads

$$\begin{cases} \forall n \in \mathbb{N}, i \in \mathbb{Z},\ \phi_i^{n+1} = \phi_i^n - \Delta t\left(\max(s_i;0)\nabla_i^+\phi^n + \min(s_i;0)\nabla_i^-\phi^n - s_i\right), \\ \forall i \in \mathbb{Z},\ \phi_i^0 = \phi_0(ih), \end{cases} \tag{14.10}$$

where s_i is the numerical quantity corresponding to the sign of ϕ_i.

14.2 A PDE-free 1-D example

Let $\mathbb{R} \ni x \mapsto f(x) \in \mathbb{R}$ be a regular function equal to $+\infty$ at ∞, and for Ω a bounded open set of $\mathbb{R}$,

$$E(\Omega) = \int_\Omega f(x)\,dx \tag{14.11}$$

that we aim at minimizing. The minimum solution is obviously the set of negative values of f. We recall that the shape derivative of f reads, for θ smooth enough

$$E'(\Omega;\theta) = \int_{\partial\Omega} (\theta\cdot\mathbf{n})\, f.$$

The direction θ defined by $\theta_{|\partial\Omega} = -f_{|\partial\Omega}$ is theoretically always a descent direction.

14.2.1 Effect of the redistanciation

At first let us consider the effect of the redistanciation process on the level-set. Let us assume that the interface $\partial\Omega$ is in the cell $[ih, (i+1)h]$ with $i \in [\![1, N-1]\!]$, and, without loss of generality that $\phi_i < 0$ and $\phi_{i+1} > 0$. We also assume that the discretization of the sign function is exact, more precisely that $s_i = -1$ and $s_{i+1} = 1$ (this is not exactly the case if we take the approximation used in Algorithm 13.2.2). Let $n \in \mathbb{N}$ and ϕ^n be one iterate of Algorithm 13.2.2. We have for all $j \in [\![0, N]\!]$

$$\phi_j^{n+1} = \phi_j^n - dt\left(\max(s_j;0)\nabla_j^+\phi + \min(s_j;0)\nabla_j^-\phi - s_j\right)$$

Proposition 14.2.1. *Let dt be such that $h \geq 2\,dt$. Assume that for $n \in \mathbb{N}$, ϕ^n satisfies*

$$\forall j \geq l,\ \phi_j^n \geq \phi_l^n.$$

Then, the same inequalities hold for ϕ^{n+1}. Moreover we have the relations

$$\phi_{i+1}^{n+1} + \phi_i^{n+1} = \phi_{i+1}^n + \phi_i^n, \tag{14.12}$$
$$D_i^+\phi^{n+1} - 1 = \left(D_i^+\phi^n - 1\right)\left(1 - 2\frac{dt}{h}\right). \tag{14.13}$$

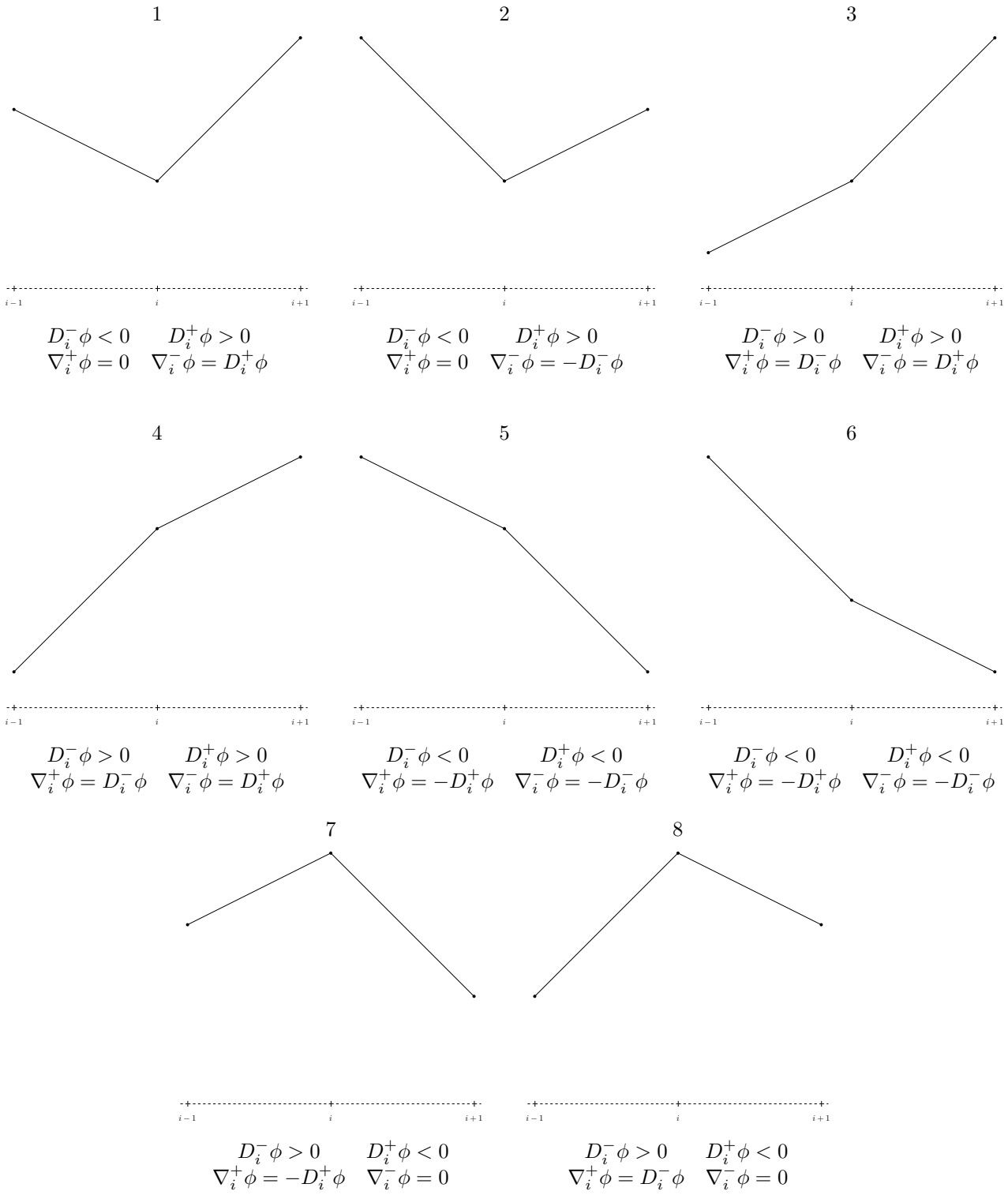


Figure 14.1: All possible different configurations for $\nabla_i^+\phi$ and $\nabla_i^-\phi$ in 1-D.

Proof. If ϕ^n is non-decreasing, the finite difference quantities $\nabla_i^+\phi$, $\nabla_i^-\phi$ are given by cases 3 or 4 of Figure 14.1. Let $j \in \mathbb{N}$ such that $s_j > 0$. We therefore have

$$\phi_j^{n+1} = \phi_j^n - dt\left(D_j^-\phi^n - 1\right),$$
$$\phi_{j+1}^{n+1} = \phi_{j+1}^n - dt\left(D_{j+1}^-\phi^n - 1\right),$$

and

$$\begin{aligned}\phi_{j+1}^{n+1} - \phi_j^{n+1} &= \phi_{j+1}^n - \phi_j^n - dt\left(D_{j+1}^- \phi^n - D_j^- \phi^n\right),\\ &= \left(\phi_{j+1}^n - \phi_j^n\right) - \frac{dt}{h}\left((\phi_{j+1}^n - \phi_j^n) - (\phi_j^n - \phi_{j-1}^n)\right),\\ &= \left(\phi_{j+1}^n - \phi_j^n\right)\left(1 - \frac{dt}{h}\right) + \frac{dt}{h}(\phi_j^n - \phi_{j-1}^n).\end{aligned}$$

Since dt satisfies the CFL condition $h \geq 2\,dt$ and ϕ^n is non-decreasing, then ϕ^{n+1} is also non-decreasing for all j such that $\phi_j^n \geq 0$. For all j such that $\phi_j^n \leq 0$ we can do exactly the same. It remains to look at $i \in \mathbb{N}$ such that $\phi_i^n < 0 < \phi_{i+1}^n$. We have

$$\begin{aligned}\phi_i^{n+1} &= \phi_i^n - dt\left(s_i \nabla_i^- \phi^n - s_i\right),\\ \phi_{i+1}^{n+1} &= \phi_{i+1}^n - dt\left(s_{i+1} \nabla_{i+1}^+ \phi^n - s_{i+1}\right).\end{aligned}$$

The finite difference quantities $\nabla_i^+ \phi^n$, $\nabla_{i+1}^+ \phi^n$ are still given by cases 3 or 4 of Figure 14.1. Therefore

$$\begin{aligned}\phi_i^{n+1} &= \phi_i^n + dt\left(D_i^+ \phi^n - 1\right),\\ \phi_{i+1}^{n+1} &= \phi_{i+1}^n - dt\left(D_{i+1}^- \phi^n - 1\right),\end{aligned}$$

which leads to

$$\begin{aligned}\phi_{i+1}^{n+1} - \phi_i^{n+1} &= \left(\phi_{i+1}^n - \phi_i^n\right) + dt\left(2 - D_{i+1}^- \phi^n - D_i^+ \phi^n\right),\\ \phi_{i+1}^{n+1} + \phi_i^{n+1} &= \left(\phi_{i+1}^n + \phi_i^n\right) + dt\left(D_{i+1}^- \phi^n - D_i^+ \phi^n\right).\end{aligned}$$

Since

$$D_{i+1}^- \phi^n = D_i^+ \phi^n = \frac{\phi_{i+1}^n - \phi_i^n}{h},$$

this leads to

$$\phi_{i+1}^{n+1} - \phi_i^{n+1} = \left(\phi_{i+1}^n - \phi_i^n\right) + 2dt\left(1 - D_i^+ \phi^n\right), \tag{14.14}$$

$$\phi_{i+1}^{n+1} + \phi_i^{n+1} = \phi_{i+1}^n + \phi_i^n. \tag{14.15}$$

Equation (14.14) can be also rewritten as

$$\begin{aligned}D_i^+ \phi^{n+1} - 1 &= \left(D_i^+ \phi^n - 1\right)\left(1 - 2\frac{dt}{h}\right),\\ D_i^+ \phi^{n+1} &= D_i^+ \phi^n \left(1 - 2\frac{dt}{h}\right) + 2\frac{dt}{h}.\end{aligned}$$

Since dt satisfies the CFL condition $h \geq 2dt$ and ϕ^n is non-decreasing we obtain that ϕ^{n+1} is also non-decreasing. □

This first statement of Proposition 14.2.1 indicates that if we start from a non-decreasing function ϕ^0, for every $n \in \mathbb{N}$ the function ϕ^n will also be non-decreasing. This property implies that the finite difference quantities $\nabla_i^+ \phi$, $\nabla_i^- \phi$ correspond to cases 3 or 4 of Figure 14.1. Therefore the intermediate result established in the previous proof, that is for i such that $\phi_i^n < 0 < \phi_{i+1}^n$

$$\phi_{i+1}^{n+1} + \phi_i^{n+1} = \phi_{i+1}^n + \phi_i^n,$$

remains true at each step $n \in \mathbb{N}$. This equality means that in the cell containing the zero level-set, the conserved quantity is not the position of the interface but the mean value $\phi_{i+1}^n + \phi_i^n$. In other words, the transport equation for solving the eikonal equation does not keep the interface unchanged in the redistanciation procedure.

14.2.2 The advection equation

Now we analyze the numerical scheme for the advection equation (Algorithm 13.2.4) coupled with the redistanciation algorithm. Let $z \in]0,1[$, ϕ^0 and v, two numerical quantities defined at the vertices of the mesh by

$$\forall j \in \mathbb{N},\ \phi_j^0 = x_j - z, \quad v_j = -f(x_j).$$

Let $\Omega_0 \subset [0,1]$ the (relatively to $[0,1]$) bounded open set defined as the negative set of ϕ^0, that is $\Omega_0 = [0, z[$. The level-set ϕ^0 is a signed distance function to the boundary of Ω_0. Given the shape derivative of the objective function

J, v is a discretization of a descent direction for J. Advecting in the direction given by v corresponds to solving the following Hamilton-Jacobi transport equation

$$\begin{cases} \partial_t \phi + v|\nabla \phi| = 0 & \text{on } [0,T]\times\mathbb{R}, \\ \phi(0,\cdot) = \phi_0 & \text{on } \mathbb{R}. \end{cases} \tag{14.16}$$

We recall the discretization scheme for the resolution of this equation. In order to ease the numerical calculation, we changed a little the fourth step of this algorithm compared to Algorithm 13.2.4. Indeed, the redistanciation algorithm is now applied each time, whereas it was supposed to be applied only one every five times. This ensures that at each step the level-set keeps being a distance function, but this is more consuming in terms of CPU time. The CPU time consideration is the only reason why the redistanciation process is not be applied at each step in practice. We also assume that the parameter ΔT is a priori given. Its value does not change the numerical analysis.

Let i be such that $\phi_i^0 < 0 < \phi_{i+1}^0$. Since ϕ^0 is a distance function, let $\alpha^0 \in [0,1]$ be such that

$$\phi_i^0 = -\alpha^0 h, \quad \phi_{i+1}^0 = (1-\alpha^0)h.$$

Let us now recall the structure of the numerical algorithm for solving the advection equation.

Algorithm 14.2.2. *Advection equation*

1. *Initialization : set* $\phi_{i,j}^0 = \phi_{i,j}$.
2. *Set* $dt_0 = \dfrac{1}{2}h$ *and* $N = \left\lceil \dfrac{\max(v)\Delta T}{dt_0} \right\rceil$.
3. *Set* $dt = \dfrac{\Delta T}{N}$ *and* $N = \min(N, 10)$.
4. *For* n *in* $0, \cdots, N$
   (a) *Apply Algorithm 13.2.2 to* ϕ.
   (b) $\phi_i^{n+1} = \phi_i^n - dt\Big(\max(v_i,0)\nabla_i^+\phi + \min(v_i,0)\nabla_i^-\phi\Big)$.
5. *Apply Algorithm 13.2.2 to* ϕ.

Let us first consider the first iteration in the loop (step 4) of this algorithm. With Proposition 14.2.1 it is straightforward to see that the first redistanciation does not change the level-set. Thus, for the first iteration of this algorithm, we can skip step 4a. We denote by $\phi^{1/2}$ the level-set after step 4b and $\phi^{1,k}$ the level-set after step 4a (of the next iteration) where $k \in \mathbb{N}$ is the number of iterations in the redistanciation process. We have

$$\begin{aligned} \phi_i^{1/2} &= \phi_i^0 - dt\, v_i, \\ \phi_{i+1}^{1/2} &= \phi_{i+1}^0 - dt\, v_{i+1}. \end{aligned}$$

Therefore

$$\begin{aligned} \phi_{i+1}^{1/2} - \phi_i^{1/2} &= h - dt\,(v_{i+1} - v_i), \\ \phi_{i+1}^{1/2} + \phi_i^{1/2} &= (1-2\alpha^0)h - dt\,(v_{i+1} + v_i). \end{aligned}$$

Now comes the redistanciation algorithm. The relations given by Proposition 14.2.1 lead to

$$\begin{aligned} \phi_{i+1}^{1,k} + \phi_i^{1,k} &= \phi_{i+1}^{1/2} + \phi_i^{1/2}, \\ \frac{\phi_{i+1}^{1,k} - \phi_i^{1,k}}{h} - 1 &= \left(\frac{\phi_{i+1}^{1/2} - \phi_i^{1/2}}{h} - 1\right)\left(1 - \frac{2dt}{h}\right)^k, \end{aligned}$$

that is

$$\begin{aligned} \phi_{i+1}^{1,k} + \phi_i^{1,k} &= (1-2\alpha^0)h - dt\,(v_{i+1} + v_i), \\ \phi_{i+1}^{1,k} - \phi_i^{1,k} &= h - dt(v_{i+1} + v_i)\left(1 - \frac{2dt}{h}\right)^k. \end{aligned}$$

Therefore we conclude that

$$\begin{aligned} \phi_i^{1,k} &= -\alpha^0 h - \frac{dt}{2}(v_{i+1} + v_i) + \frac{dt}{2}(v_{i+1} - v_i)\left(1 - \frac{2dt}{h}\right)^k, \\ \phi_{i+1}^{1,k} &= (1-\alpha^0)h - \frac{dt}{2}(v_{i+1} + v_i) - \frac{dt}{2}(v_{i+1} - v_i)\left(1 - \frac{2dt}{h}\right)^k. \end{aligned}$$

When $dt = \frac{h}{2}$ this leads to

$$\phi_i^{1,k} = -\alpha^0 h - \frac{h}{4}(v_{i+1} + v_i),$$
$$\phi_{i+1}^{1,k} = (1 - \alpha^0)h - \frac{h}{4}(v_{i+1} + v_i).$$

We could also have assumed that the redistanciation was performed with $k = \infty$ which means that the gradient of the level-set is 1. This would have given the same result for $\phi_i^{1,k}$ and $\phi_i^{1,k}$. Therefore, after one iteration, the parameter α^0 is updated as

$$\alpha^1 = \alpha^0 + \frac{1}{4}(v_{i+1} + v_i).$$

Now let us assume that the optimal position of the interface is in the cell $[ih, (i+1)h]$. In particular we have $f(x_i) < 0 < f(x_{i+1})$. The direction given by the gradient - namely $v_i = -f(x_i)$ - is a descent direction and does not depend on the level-set. Therefore, after k iterations of the gradient algorithm, the parameter α becomes

$$\alpha^k = \alpha^0 + \frac{k}{4}(v_{i+1} + v_i). \tag{14.17}$$

This implies that the zero level-set is attracted by one of the points ih or $(i+1)h$ depending on whether $v_{i+1} + v_i$ is positive or negative. If after an iteration, the zero level-set stays in another cell, it will soon go back to the optimal cell since the mean value of the direction would not be zero. And it would again be attracted by the same edge of the cell.

14.2.3 Numerical example

This is exactly what we can observe in practice. For example, we take the segment $[0, 1]$ and a mesh parameter $h = 10^{-3}$. We take $z^\star = 0.4003$ and $z_0 = 0.4004$. We also consider $f(x) = x - z^\star$ and the objective function E defined by (14.11). We start from an initial domain Ω_0 defined by the initial level-set function : $\phi_0(x) = x - z_0$. For the redistanciation algorithm as well as the Hamilton-Jacobi transport equation, we consider periodic boundary conditions for the level-set. We take $v : [0, 1] \to \mathbb{R}$ defined by $\forall x \in [0, 1], v(x) = -f(x)$. Thus we have $E'(\Omega; v) < 0$, whenever Ω is not a critical point for E.

We can first apply the algorithm described in Section 13.2. In that case the position of the interface, denoted by z, converges to $z^\star$, which is indeed the expected result. However, we can notice that the stopping criterion is always the second step of the line search algorithm Algorithm 13.2.5, meaning that this procedure does not find a time step that allows to decrease the objective function. In Section 14.2.2, we saw that the descent direction given by the first-order derivative doesn't seem to converges to zero, and that implies that the position of the interface - at z - tends to be attracted by a node of the mesh. This can be observed on this example. To that end, we skip the line search in the algorithm and always take the same small time step : $\Delta T = 20h$ in the spirit of the gradient algorithm with a fixed step (see Section 1.2.1).

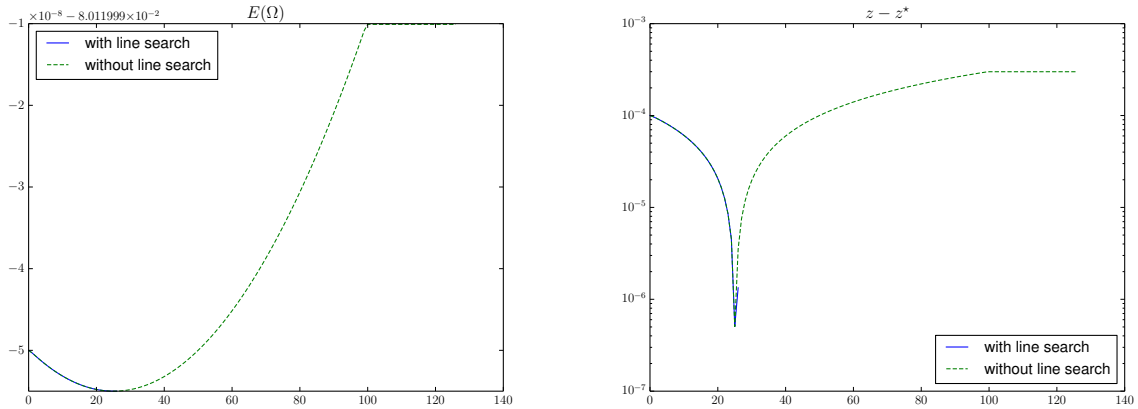


Figure 14.2: Evolution of the objective function (left) and error with the theoretical optimal solution (right).

We can first observe on Figure 14.2 that the algorithm with the line search procedure converges to the theoretical solution. We can also see that when there is no line search, the objective function tends to increase at the end of the optimization, and that the position of the interface z does not converge to the theoretical solution $z^\star$. It appears theoretically that advecting the level-set with the velocity $v = -f$ makes the interface z be attracted by a node of the mesh. Thus we also plot the evolution of

$$z - h \left\lfloor \frac{z^\star}{h} \right\rfloor.$$

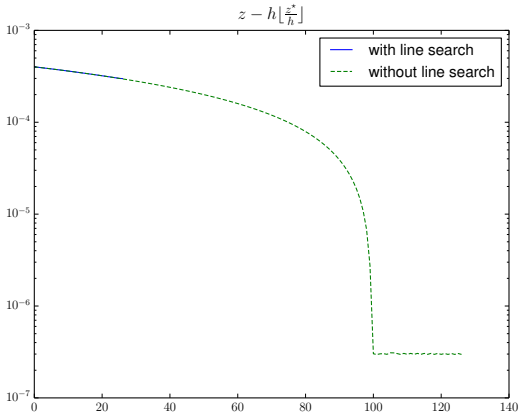


Figure 14.3: Convergence to a node of the mesh.

In the case where there is no line search we can indeed observe on Figure 14.3 that the solution z converges to $h\left\lfloor \dfrac{z^\star}{h} \right\rfloor$ which is one of the node of the cell containing $z^\star$. This also explains that we cannot really hope for the first-order shape derivative to vanish, and why this is always the line search algorithm that provides the practical stopping criterion.

Finally we could also make a remark on how the descent direction v is computed. Let us recall that in most practical case, the first-order shape derivative of a function E can be represented by a function $j \in L^1(\partial\Omega;\mathbb{R})$, with

$$\forall \theta \in C^{1,\infty}(\mathbb{R}^d;\mathbb{R}),\ E'(\Omega;\theta) = \int_{\partial\Omega} (\theta \cdot \mathbf{n})\, j.$$

In such a case, the function j can be called the shape gradient. For the present example, the restriction of $\mathbb{R} \ni x \mapsto f(x)$ to the boundary of the shape is the shape gradient. The choice $\forall j \in \mathbb{N}$, $v(x_j) = -f(x_j)$ seemed therefore natural. However, with (14.17) and Figure 14.3 we observe that the interface converges to a node of the mesh instead of the theoretical optimal solution. The error of order h is not a real problem about the optimality of the final shape. However, it shows that this approach for computing shape derivative is not really consistent. Moreover, the relation (14.17) depends only on the values of v at the nodes x_i and x_{i+1} where i is such that the interface is in the cell $[ih,(i+1)h]$. Thus it does not depend on the choice of extension for the descent direction v, and it also does not depend on an approximation by a Dirac function. This last observation could also be in favor of the method presented in Section 11.2 for computing boundary integrals, and therefore shape derivatives.

14.3 The fast-marching method

The fast-marching method is another alternative for solving the eikonal equation (see Section 4.3.2) on a Cartesian grid. It is a priori known for preserving the position of the boundary of the shape. It turns out that this property is true in the one-dimensional case, but not exactly in higher-dimensional spaces.

Let us consider the two-dimensional case. The main idea is to use a propagating front from the initial boundary. At each step of the propagation, the nearest external value to the front is updated by solving the following upwind equation

$$\max\left(D_{i,j}^{+x}\phi;0\right)^2 + \max\left(D_{i,j}^{-x}\phi;0\right)^2 + \max\left(D_{i,j}^{+y}\phi;0\right)^2 + \max\left(D_{i,j}^{-y}\phi;0\right)^2 = 1.$$

It appears that the procedure is well defined in the sense that there is always a unique solution to this equation (see [109] for example). The numerical trouble comes from the fact that the fast-marching procedure is not a projection. Precisely, starting from a level-set function ϕ^0, either solution to the eikonal equation or not, we denote ϕ^1 the solution to the eikonal equation with the fast-marching method initialized with ϕ^0. We apply again the fast-marching method to ϕ^1 and denote the result by ϕ^2. It appears that there is no reason why ϕ^2 should be the same that ϕ^1. The issue comes from the initialization of the algorithm. It consists in computing the 'exact' distance to the boundary assuming that the level-set is linear in each cell. We explain this with the help of an example.

Let ϕ^0 be a level-set function with

$$\phi^0(A) = -\frac{2h}{3}, \quad \phi^0(B) = \frac{h}{3}, \quad \phi^0(C) = -\frac{2h}{3},$$

$$\phi^0(D) = \frac{h}{3}, \qquad \phi^0(E) = \frac{h}{3},$$

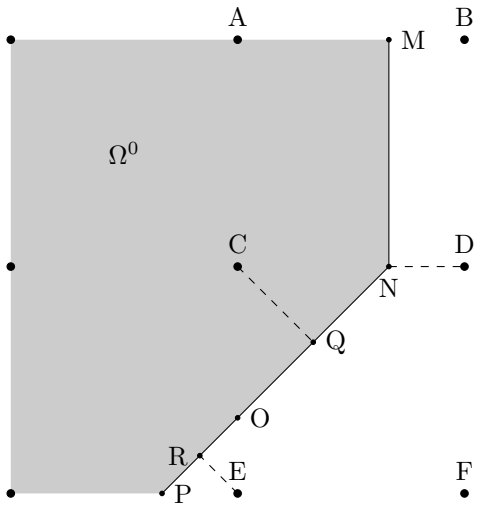


Figure 14.4: Example for the non-conservation of the boundary by the fast-marching method.

where the position of the points are given by Figure 14.4. Assuming that ϕ^0 is linear in each direction, we write

$$\frac{CO}{h} = -\frac{\phi^0(C)}{\phi^0(E) - \phi^0(C)} = \frac{2}{3}, \quad \frac{CN}{h} = -\frac{\phi^0(C)}{\phi^0(D) - \phi^0(C)} = \frac{2}{3}.$$

This leads to a domain Ω^0 such as represented by Figure 14.4. Now we start computing the solution ϕ^1 to the eikonal equation for Ω^0. The value assigned to the node C - namely $\phi^1(C)$ - is the distance CQ, i.e the distance of C to the boundary. This leads to

$$\phi^1(C) = -\frac{2\sqrt{2}}{3}h.$$

Similarly, the values assigned to A, B, D and E are

$$\phi^1(A) = -\frac{2}{3}h, \quad \phi^1(B) = \frac{h}{3}, \quad \phi^1(D) = \frac{h}{3}, \quad \phi^1(E) = \frac{\sqrt{2}}{3}h.$$

The upwind propagation of the front will not change the value of ϕ^1 at these nodes. Now we do the same computation starting from ϕ^1. This leads to

$$\frac{CO}{h} = -\frac{\phi^1(C)}{\phi^1(E) - \phi^1(C)} = \frac{2}{3}, \quad \frac{CN}{h} = -\frac{\phi^1(C)}{\phi^1(D) - \phi^1(C)} = \frac{2\sqrt{2}}{2\sqrt{2}+1}.$$

Therefore, the distance CQ turns out to be

$$CQ = \frac{2\sqrt{2}h}{\sqrt{18 + (2\sqrt{2}+1)^2}},$$

which is the value that is assigned at C for ϕ^2.

This elementary example shows that the fast-marching method is not always a projection. The problem is that for computing a distance to the boundary, the level-set is assumed to be linear in each cell. Even if it is true, this property is not satisfied by the first computed distance ϕ^1. Indeed for the domain Ω^0 represented in Figure 14.4, if ϕ^1 is the exact distance to the boundary, it is not linear on the segment $[C, D]$. Therefore computing again a distance function from ϕ^1 can not lead to the same result.

In the one-dimensional case this problem does not occur. The signed distance function is obviously piecewise linear on the whole domain, therefore the fast-marching method would be exact. However, this does not have a great interest, since once the position of the boundary is known - let say at z - the distance to the boundary has the very simple expression : $d_\Omega(x) = |x - z|$.

Chapter 15

A one-dimensional optimization case

Contents

In order to perform a complete numerical analysis of a shape optimization problem, we consider a one-dimensional case. A shape is defined as an open subset of the space $\mathbb{R}^d$ ($d \in \{1, 2, 3\}$). When $d = 1$, a convex open set of $\mathbb{R}$ is an interval and is uniquely determined by the two bounds of the interval. Contrary to topology optimization, geometric optimization considers a shape with a fixed topology. In the one-dimensional case, when the shape is a convex bounded set with one connected component, geometric optimization reduces to parametric optimization where the unknowns are the two boundaries of the interval. Although this case is somehow degenerate for geometric optimization (there is no question about the regularity of the boundary of the shape), the similarity with the parametric case allows one to thoroughly analyze the numerical schemes. The simplicity of the case makes the computations easier, and most of them, can be performed explicitly.

In this chapter we will consider a shape optimization problem where the shape is an interval and can be parametrized by a single parameter z. The minimization of a compliance or a least square criteria such as in Chapter 12 would lead to a too much simple problem. Thus, we will focus on the least square displacement criterion with a non-zero target displacement u_0. First we will start by describing the continuous problem, and its explicit optimal solution. Then we will give the exact solution of a discretized version of this problem, and give some numerical analysis on the error with the continuous optimal solution. Secondly, we will consider the derivatives of these problems in a parametric and a shape variation (see Section 2.2.3) frameworks and see that, in this particular case, they are the same. Then we will focus on the analysis of the rates of convergence of a Newton method for the continuous model, the *discretize-then-optimize* and *optimize-then-discretize* approaches (see Section 1.3). We will see in particular that the continuous model converges in one iteration (since the problem is quadratic), the *discretize-then-optimize* approach has a quadratic convergence whereas the *optimize-then-discretize* approach is only linear when it converges. We will also analyze the impact on the rate of convergence of errors on the first and second-order derivatives. Finally we will make some numerical experiments to confirm the theoretical results obtained.

15.1 The continuous problem

Let us consider $\Omega =]0,1[$ and $z \in [0,1]$. We denote $\Omega_1 :=]0,z[$ and $\Omega_2 :=]z,1[$. Let also $\alpha_1, \alpha_2 \in \mathbb{R}_+^\star$ and define $[0,1] \ni x \mapsto \alpha^z(x)$ with

$$\alpha^z(x) = \begin{cases} \alpha_1 & \text{if } x \leq z, \\ \alpha_2 & \text{otherwise.} \end{cases} \tag{15.1}$$

Notation. *For any function ψ defined on Ω, we denote by ψ_i its restriction to Ω_i, i.e $\psi_i := \psi_{|\Omega_i}$. We also denote by $[\psi]$ the jump of ψ at z : $[\psi] := \psi_1(z^-) - \psi_2(z^+)$.*

We also introduce

$$\gamma^z = \frac{1}{\alpha^z}, \quad \gamma_1 = \frac{1}{\alpha_1}, \quad \gamma_2 = \frac{1}{\alpha_2}, \quad [\gamma] = \gamma_1 - \gamma_2. \tag{15.2}$$

Let now $u^z \in H^1(]0,1[;\mathbb{R})$ be the solution to the following problem

$$\begin{cases} -d_x\big(\alpha^z d_x u^z\big) &= 0 \quad \text{in } \Omega, \\ \big(\alpha^z d_x u^z\big)(1) &= 1, \\ u^z(0) &= 0, \end{cases} \tag{15.3}$$

where we use the notation $d_x u^z(x) = \dfrac{du^z}{dx}(x)$.

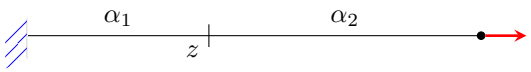


Figure 15.1: Boundary conditions for (15.3).

Given a target displacement $u_0 > 0$ we consider the objective function J defined by

$$J(z) = |u^z(1) - u_0|^2. \tag{15.4}$$

and the optimization problem

$$\min_{z \in [0,1]} J(z). \tag{15.5}$$

Proposition 15.1.1. *There exists a unique solution u^z to (15.3) in $H^1(]0,1[;\mathbb{R})$. It is given by*

$$u^z(x) = \begin{cases} \gamma_1 x & \text{if } x \leq z, \\ \gamma_2 + [\gamma] x & \text{if } x > z, \end{cases} \tag{15.6}$$

where γ_2 and $[\gamma]$ are defined by (15.2). Therefore, the objective function reads

$$J(z) = \Big(\gamma_2 + [\gamma] z - u_0\Big)^2, \tag{15.7}$$

and the solution to (15.5) is obtained at $z^\star$ with

$$z^\star = \frac{u_0 - \gamma_2}{[\gamma]}. \tag{15.8}$$

Proposition 15.1.2. *With $\mathcal{V} = \big\{v \in H^1(]0,1[;\mathbb{R}) \,|\, v(0) = 0\big\}$, the variational formulation of (15.3) reads*

$$\begin{cases} \text{find } u^z \in \mathcal{V}, \text{ such that} \\ \forall v \in \mathcal{V}, \ \displaystyle\int_0^1 \alpha^z \, d_x u^z \, d_x v = v(1). \end{cases} \tag{15.9}$$

15.2 The discretized problem

Let $N \in \mathbb{N}^\star$ and $h = \frac{1}{\mathbb{N}}$. The segment $[0,1]$ is endowed with a Cartesian grid with steps h.

Notation. *For $x \in \mathbb{R}$ we denote by $\lfloor x \rfloor$ the unique element of $\mathbb{Z}$ such that $\lfloor x \rfloor \leq x < \lfloor x \rfloor + 1$. We also define $\lceil x \rceil$ by the unique element of $\mathbb{Z}$ such that $\lceil x \rceil - 1 < x \leq \lceil x \rceil$.*

For $z \in [0,1]$ let $\eta_h^z = \frac{z}{h} - \left\lfloor \frac{z}{h} \right\rfloor$. Let also $\langle \alpha_h^z \rangle$, α_h^z be defined by $\langle \alpha_h^z \rangle := \eta_h^z \alpha_1 + (1 - \eta_h^z)\alpha_2$ and

$$\alpha_h^z(x) = \begin{cases} \alpha^z(x) & \text{if } x \leq h \left\lfloor \frac{z}{h} \right\rfloor \text{ or } x > h \left\lceil \frac{z}{h} \right\rceil, \\ \langle \alpha_h^z \rangle & \text{otherwise.} \end{cases} \tag{15.10}$$

This choice for α_h^z corresponds to a $\mathbb{P}_0$ interpolation of α^z. We can either notice that $\alpha_h^z(x) = \alpha^z(x)$ whenever x is outside the grid cell containing z. In the cell containing z, we take the convex combination given by the position of z in this cell, since by definition of η_h^z

$$z = (1 - \eta_h^z) h \left\lfloor \frac{z}{h} \right\rfloor + \eta_h^z h \left\lceil \frac{z}{h} \right\rceil.$$

We can also check that, when z is on a vertex of the grid, $\eta_h^z = 0$ and

$$\alpha_h^z(x) = \begin{cases} \alpha_1 & \text{if } x \leq z \\ \alpha_2 & \text{else.} \end{cases}$$

Notation. *For a numerical quantity ϕ defined at the vertices of the grid and $i \in [\![0, N]\!]$, ϕ_i denotes the value assigned to the node $x_i := ih$. Similarly, for a $\mathbb{P}_0$ quantity ψ and $i \in [\![1, N]\!]$, ψ_i denotes the value assigned in the cell $](i-1)h, ih]$. In the case of α_h^z it reads $\alpha_{h,i}^z = \alpha_h^z(ih)$.*

We introduce a finite element variational space $\mathcal{V}_h = \{v_h \in \mathbb{P}_1([0,1]; \mathbb{R}) \,|\, v_h(0) = 0\}$. The solution to (15.3) in $\mathbb{P}_1(\mathcal{V}_h; \mathbb{R})$ is also solution to the following variational formulation

$$\begin{cases} \text{find } u_h^z \in \mathcal{V}_h, \text{ such that} \\ \forall v_h \in \mathcal{V}_h, \ \displaystyle\int_0^1 \alpha_h^z \, d_x u_h^z \, d_x v_h = v_h(1). \end{cases} \tag{15.11}$$

The target displacement u_0 being the same as in the continuous case, we consider the objective function J_h defined by

$$J(z) = |u_h^z(1) - u_0|^2. \tag{15.12}$$

and the optimization problem

$$\min_{z \in [0,1]} J_h(z). \tag{15.13}$$

Proposition 15.2.1. *There exists a unique solution $u_h^z \in \mathcal{V}_h$ solution to (15.11). It satisfies*

$$u_{h,0}^z = 0, \tag{15.14}$$

$$u_{h,i}^z = i\gamma_1 h, \quad \forall i \leq \left\lfloor \frac{z}{h} \right\rfloor, \tag{15.15}$$

$$u_{h,i}^z = u_{h,N}^z - \gamma_2 h \, (N - i), \quad \forall i \geq \left\lceil \frac{z}{h} \right\rceil, \tag{15.16}$$

$$u_{h,N}^z = \gamma_2 + [\gamma] z + h \left(\frac{1}{\langle \alpha_h^z \rangle} - \left(\eta_h^z \gamma_1 + (1 - \eta_h^z)\gamma_2 \right) \right). \tag{15.17}$$

Therefore, the discretized objective function reads

$$J_h(z) = \left(\gamma_2 + [\gamma] z + h \left(\frac{1}{\langle \alpha_h^z \rangle} - \left(\eta_h^z \gamma_1 + (1 - \eta_h^z)\gamma_2 \right) \right) - u_0 \right)^2.$$

Proof. The existence and uniqueness result for (15.11) is given by the Lax-Milgram theorem on $\mathcal{V}_h$. In order to get the explicit expression of the solution we detail the computation of the linear system. Let (ψ_j) be the basis functions of $\mathcal{V}_h$. For $i \in [\![0, N]\!]$ and $j \notin \{i-1, i, i+1\}$ we get

$$\int_0^1 (\psi_i)^2 = \frac{2}{h} \quad \text{if } 1 \leq i \leq N-1, \quad \text{and } \int_0^1 (\psi_N)^2 = \frac{1}{h},$$

$$\int_0^1 \psi_i \psi_{i+1} = -\frac{1}{h} \quad \text{if } i < N,$$

$$\int_0^1 \psi_i \psi_{i-1} = -\frac{1}{h} \quad \text{if } i > 0,$$

$$\int_0^1 \psi_i \psi_j = 0.$$

Let u_h^z be the solution to (15.11). For $v_h \in \mathcal{V}_h$ we have

$$\int_0^1 \alpha_h^z \, d_x u_h^z \, d_x v_h = v_{h,N}.$$

With $u_h^z(0) = v_h(0) = 0$ it reads

$$\frac{1}{h}\Big(\alpha_{h,1}^z \, u_{h,1}^z \, v_{h,1}^z + \sum_{i=2}^{N} \alpha_{h,i}^z \big(u_{h,i}^z \, v_{h,i}^z + u_{h,i-1}^z \, v_{h,i-1}^z - u_{h,i}^z \, v_{h,i-1}^z - u_{h,i-1}^z \, v_{h,i}^z\big)\Big) = v_{h,N}.$$

Being true for every $v_h \in \mathcal{V}_h$, it leads to the following system

$$\left\{ \begin{array}{lcll} (\alpha_{h,1}^z + \alpha_{h,2}^z) u_{h,1}^z - \alpha_{h,2}^z u_{h,2}^z & = & 0, & \\ (\alpha_{h,i}^z + \alpha_{h,i+1}^z) u_{h,i}^z - \alpha_{h,i}^z u_{h,i-1}^z - \alpha_{h,i+1}^z u_{h,i+1}^z & = & 0, & \forall i,\, 1 < i < N-1 \\ \alpha_{h,N}^z (u_{h,N}^z - u_{h,N-1}^z) & = & h. & \end{array} \right. \tag{15.18}$$

When α_h^z is given by (15.10), (15.16) is straightforwardly obtained by recursion. A recursion also gives

$$\forall i,\, i+1 \leq \left\lfloor \frac{z}{h} \right\rfloor, \quad u_{h,i}^z = \frac{i}{\left\lfloor \frac{z}{h} \right\rfloor} u_{h,\left\lfloor \frac{z}{h} \right\rfloor}^z. \tag{15.19}$$

For $j = \left\lceil \frac{z}{h} \right\rceil$ we can also write the middle equation of (15.18) with $i = j$ and $i = j-1$. Combined with (15.19) and (15.16) it leads to

$$\left\{ \begin{array}{lcl} \left(\dfrac{\alpha_{h,j-1}^z}{j-1} + \alpha_{h,j}^z \right) u_{h,j-1}^z - \alpha_{h,j}^z u_{h,N}^z & = & -\alpha_{h,j}^z (N-j)\gamma_2 h, \\ -\alpha_{h,j}^z u_{h,j-1}^z + \alpha_{h,j}^z u_{h,N}^z & = & \alpha_{h,j}^z (N-j)\gamma_2 h + \alpha_{h,j+1}^z \gamma_2 h \end{array} \right. \tag{15.20}$$

Now it remains to recall that

$$\eta_h^z = z - h \left\lfloor \frac{z}{h} \right\rfloor, \quad \alpha_{h,\left\lceil \frac{z}{h} \right\rceil}^z = \eta_h^z \alpha_1 + (1 - \eta_h^z)\alpha_2, \quad Nh = 1,$$

to get the final result. □

Remark 15.2.2. *One can observe that the solution to (15.11) can be written as*

$$u_{h,i}^z = h \sum_{k=1}^{i} \frac{1}{\alpha_{h,i}^z}, \quad \forall i \leq N. \tag{15.21}$$

Moreover, this does not depend on the fact that $\alpha_{h,i}^z$ is constant for $i < \lfloor \frac{z}{h} \rfloor$ or $i \geq \lceil \frac{z}{h} \rceil$.

Proposition 15.2.3. *There exists a unique solution $z_h^\star$ to (15.13). It reads*

$$z_h^\star = z_0 + \eta_h^\star h, \tag{15.22}$$

where

$$\eta_h^\star = \frac{1}{[\alpha]} \left(\frac{1}{\gamma_2 + [\gamma] Z} - \alpha_2 \right), \quad Z = \frac{z^\star}{h} - \left\lfloor \frac{z^\star}{h} \right\rfloor, \quad z_0 = h \left\lfloor \frac{z^\star}{h} \right\rfloor,$$

and $z^\star$ is given by (15.8).

Proof. Let $z_h^\star$ be defined by $z_h^\star = z_0 + \eta_h^\star h$ with

$$\eta_h^\star = \frac{1}{[\alpha]} \left(\frac{1}{\gamma_2 + [\gamma] Z} - \alpha_2 \right), \quad Z = \frac{z^\star}{h} - \left\lfloor \frac{z^\star}{h} \right\rfloor, \quad z_0 = h \left\lfloor \frac{z^\star}{h} \right\rfloor,$$

and $z^\star = \dfrac{u_0 - \gamma_2}{[\gamma]}$. We can easily check that $\eta_h^z \in [0,1]$. Thus, one can write

$$\begin{aligned} u_h^{z_h^\star}(1) - u_0 &= \gamma_2 + [\gamma] z_h^\star + h \left(\frac{1}{\left\langle \alpha_h^{z_h^\star} \right\rangle} - \big(\eta_h^\star \gamma_1 + (1 - \eta_h^\star)\gamma_2\big) \right) - u_0, \\ &= \gamma_2 + [\gamma] z_0 - u_0 + h \eta_h^\star [\gamma] + h \left(\frac{1}{\left\langle \alpha_h^{z_h^\star} \right\rangle} - \big(\eta_h^\star \gamma_1 + (1 - \eta_h^\star)\gamma_2\big) \right). \end{aligned}$$

Since $Z = \dfrac{z^\star}{h} - \dfrac{z_0}{h}$ and $z^\star = \dfrac{u_0 - \gamma_2}{[\gamma]}$, we obtain

$$u_h^{z_h^\star}(1) - u_0 = -h[\gamma]Z + \frac{h}{\eta_h^\star[\alpha] + \alpha_2} - h\gamma_2. \tag{15.23}$$

The definition of $\eta_h^\star$ by $\eta_h^\star = \dfrac{1}{[\alpha]}\left(\dfrac{1}{\gamma_2 + [\gamma]Z} - \alpha_2\right)$ leads to

$$(\eta_h^\star[\alpha] + \alpha_2)(\gamma_2 + [\gamma]Z) = 1.$$

As a result, we get

$$u_h^{z_h^\star}(1) - u_0 = 0.$$

Therefore, $z_h^\star$ so defined is a minimum of J_h and $J_h(z_h^\star) = 0$. As a consequence any minimum satisfies $J_h(z) = 0$, which is the same as $u_h^z(1) = u_0$ (J_h is straightforwardly non-negative).

Let now y be minimum of J_h. We have then $u_h^y(1) = u_0$. Denoting $\hat{y} = h\left\lfloor \dfrac{y}{h} \right\rfloor$, we have $y = \hat{y} + \eta_h^y h$ and

$$u_h^y(1) - u_0 = \gamma_2 + [\gamma^y]\hat{y} - u_0 + h\eta_h^y[\gamma^y] + h\left(\frac{1}{\langle \alpha_h^y \rangle} - \left(\eta_h^y\gamma_1 + (1-\eta_h^y)\gamma_2\right)\right).$$

Thus, we obtain

$$u_h^y(1) - u_0 = [\gamma^y](\hat{y} - z^\star) + \frac{h}{\eta_h^y[\alpha_h^y] + \alpha_2} - h\gamma_2.$$

Since $u_h^y(1) = u_0$, we have

$$\begin{aligned} z^\star - \hat{y} &= \frac{h}{[\gamma^y]}\left(\frac{1}{\eta_h^y[\alpha_h^y] + \alpha_2} - \gamma_2\right) \\ &= -\frac{h\gamma_2\eta_h^y}{\eta_h^y[\alpha_h^y] + \alpha_2}\frac{[\alpha_h^y]}{[\gamma_h^y]}. \end{aligned}$$

But, since $\dfrac{[\alpha_h^y]}{[\gamma_h^y]} = -\alpha_1\alpha_2$ we obtain

$$z^\star - \hat{y} = \frac{h\alpha_1\eta_h^y}{\eta_h^y[\alpha_h^y] + \alpha_2} = h\frac{\alpha_1\eta_h^y}{\eta_h^y\alpha_1 + (1-\eta_h^y)\alpha_2}.$$

Therefore, $z^\star - \hat{y} \in [0, h]$. As a consequence we have $\hat{y} = z_0$. Finally we can then write the difference between $u_h^y(1)$ and u_0 like in (15.23) which gives

$$-h[\gamma^y]Z + \frac{h}{\eta_h^y[\alpha^y] + \alpha_2} - h\gamma_2 = 0.$$

Since $[\alpha^y] = \alpha_1 - \alpha_2$ is independent of y we obtain that

$$\eta_h^y = \eta_h^z = \eta_h^\star,$$

which proves that $y = z$ which is the unique minimum of J_h. □

Both the numerical and continuous optimization problems have unique solutions. Since their explicit expressions are given by Proposition 15.1.1 and Proposition 15.2.3 it is possible to get the approximation error due to the discretization.

Proposition 15.2.4. *Let $z^\star$ and $z_h^\star$ be the unique solutions to (15.5) and (15.13), that are respectively given by (15.8) and (15.22). Let $\varepsilon_h = z_h^\star - z^\star$. Then $\varepsilon_h = O(h)$. More precisely*

$$|\varepsilon_h| \leq \left|\frac{\sqrt{\alpha_2} - \sqrt{\alpha_1}}{\sqrt{\alpha_2} + \sqrt{\alpha_1}}\right| h,$$

and this bound is optimal.

Proof. With $Z = \dfrac{z^\star}{h} - \left\lfloor \dfrac{z^\star}{h} \right\rfloor$, the approximation error ε_h reads

$$\begin{aligned} \varepsilon_h &= h(\eta_h^\star - Z), \\ &= \frac{h}{[\alpha]}\left(\frac{1}{\gamma_2 + [\gamma]Z} - \alpha_2 - [\alpha]Z\right). \end{aligned}$$

Let $\Delta : [0,1] \ni Z \mapsto \frac{1}{[\alpha]}\left(\frac{1}{\gamma_2+[\gamma]Z} - \alpha_2 - [\alpha]Z\right)$. It is straightforward to see that $\Delta(0) = \Delta(1) = 0$. Studying the variations of Δ, its extremum is attained for $Z_w = \frac{\sqrt{\alpha_1}}{\sqrt{\alpha_1}+\sqrt{\alpha_2}}$, and

$$\Delta(Z_w) = \frac{\sqrt{\alpha_2}-\sqrt{\alpha_1}}{\sqrt{\alpha_2}+\sqrt{\alpha_1}},$$

which leads to the expected result. □

15.3 Differentiation

Before solving iteratively the optimization problems (15.5), (15.13) mentioned in the previous section we need to specify in which framework the optimization takes place, and consider the corresponding derivatives. At first we consider the parametric case where the variable is $z \in [0,1]$ and secondly we focus on the shape variation framework where the variable is the domain Ω_1.

Notation. *The notation for the different frameworks for differentiation is to be specified. Let $[0,1] \ni z \mapsto v(z) \in \mathbb{R}$ assumed to be smooth. For the parametric case of differentiation, when the variable is z we denote the first and second derivatives of v by $d_z v$ and $d^2_{zz} v$. When we focus on the shape variation framework, for θ, ξ in $W_0^{1,\infty}([0,1])$, we denote the first shape derivatives at 0 of v in the direction θ by $v'(\Omega_1;\theta)$ or v'_θ and the second-order shape derivative in directions θ, ξ by $v''(\Omega_1;\theta,\xi)$.*

15.3.1 Parametric framework

In the parametric case, the optimization variable is $z \in [0,1]$. Therefore the derivatives needed for an optimization process are the usual scalar derivatives in $\mathbb{R}$ of a scalar function.

Continuous problem

Here we aim at minimizing $[0,1] \ni z \mapsto J(z)$ where J is defined by Proposition 15.1.1 :

$$J(z) = \Big(\gamma_2 + [\gamma]z - u_0\Big)^2.$$

Proposition 15.3.1. *The function $[0,1] \ni z \mapsto J(z)$ is twice differentiable in $[0,1]$ with*

$$d_z J(z) = 2[\gamma]\Big(\gamma_2 + [\gamma]z - u_0\Big), \tag{15.24}$$

$$d^2_{zz} J(z) = 2[\gamma]^2. \tag{15.25}$$

Discretized problem

In the discretized case, the objective is to minimize $[0,1] \ni z \mapsto J_h(z)$ where J_h is defined by Proposition 15.2.1 :

$$J_h(z) = \left(\gamma_2 + [\gamma]z + h\left(\frac{1}{\langle\alpha_h^z\rangle} - \big(\eta_h^z\gamma_1 + (1-\eta_h^z)\gamma_2\big)\right) - u_0\right)^2,$$

with $\eta_h^z = \frac{z}{h} - \left\lfloor \frac{z}{h} \right\rfloor$ and $\langle\alpha_h^z\rangle := \eta_h^z\alpha_1 + (1-\eta_h^z)\alpha_2$.

Proposition 15.3.2. *Assume that $\eta_h^z \neq 0$. The function $[0,1] \ni z \mapsto \eta_h^z$ is differentiable with $d_z\eta_h^z = \frac{1}{h}$. Therefore, the function $[0,1] \ni z \mapsto J_h(z)$ is twice differentiable with*

$$d_z J_h(z) = -2\left([\gamma](z-z^\star) + \frac{h}{\eta_h^z[\alpha]+\alpha_2} - h\eta_h^z[\gamma] - h\gamma_2\right)\frac{[\alpha]}{\big(\eta_h^z[\alpha]+\alpha_2\big)^2}, \tag{15.26}$$

$$d^2_{zz} J_h(z) = 2\frac{[\alpha]^2}{\big(\eta_h^z[\alpha]+\alpha_2\big)^4} + 4\left([\gamma]\frac{z-z^\star}{h} + \frac{1}{\eta_h^z[\alpha]+\alpha_2} - \eta_h^z[\gamma] - \gamma_2\right)\frac{[\alpha]^2}{\big(\eta_h^z[\alpha]+\alpha_2\big)^3}. \tag{15.27}$$

When $\left\lfloor \frac{z}{h} \right\rfloor = \left\lfloor \frac{z^\star}{h} \right\rfloor$ one has with the definition of Proposition 15.2.3 for Z and $\eta_h^\star$

$$z - z^\star = h(\eta_h^z - Z), \quad \gamma_2 + [\gamma]Z = \frac{1}{\eta_h^\star[\alpha]+\alpha_2}, \quad Z = \frac{z^\star}{h} - \left\lfloor \frac{z^\star}{h} \right\rfloor.$$

Therefore, for all $z \in]0,1[$ *such that* $\left\lfloor \frac{z}{h} \right\rfloor = \left\lfloor \frac{z^\star}{h} \right\rfloor$, *we have*

$$u_h^z(1) - u_0 = [\gamma](z - z^\star) + \frac{h}{\eta_h^z[\alpha] + \alpha_2} - h\eta_h^z[\gamma] - h\gamma_2, \tag{15.28}$$

$$= h\left(\frac{1}{\eta_h^z[\alpha] + \alpha_2} - \frac{1}{\eta_h^\star[\alpha] + \alpha_2}\right), \tag{15.29}$$

and

$$d_z J_h(z) = -2h\left(\frac{1}{\eta_h^z[\alpha] + \alpha_2} - \frac{1}{\eta_h^\star[\alpha] + \alpha_2}\right)\frac{[\alpha]}{\left(\eta_h^z[\alpha] + \alpha_2\right)^2}, \tag{15.30}$$

$$d_{zz}^2 J_h(z) = 2\frac{[\alpha]^2}{\left(\eta_h^z[\alpha] + \alpha_2\right)^4} + 4\left(\frac{1}{\eta_h^z[\alpha] + \alpha_2} - \frac{1}{\eta_h^\star[\alpha] + \alpha_2}\right)\frac{[\alpha]^2}{\left(\eta_h^z[\alpha] + \alpha_2\right)^3}. \tag{15.31}$$

15.3.2 Shape variation framework

The main idea in this section is to compute derivatives following the Hadamard method for shape differentiation (Section 2.2.3). Taking Ω_1 as a reference domain, a variation of it is considered of the form $\Omega^{1,\theta} := (\mathrm{Id} + \theta)(\Omega_1)$, where Id is the identity operator from $\mathbb{R}$ to $\mathbb{R}$ and θ is a scalar field on $\mathbb{R}$. There are two main points to notice. First, due to the non-continuity of the parameter α^z through the interface between Ω_1 and Ω_2, u is not differentiable at z. Therefore, there are shape differentiability results for the Lagrangian derivatives of u but not its Eulerian derivatives. The existence of the Lagrangian or Eulerian derivatives when the coefficients of the differential operator have jumps has already been studied [10, 99]. The one-dimensional case considered here is simpler but the differentiability results are similar.

Differentiation of the mechanical problem

We consider identity perturbations with $\theta \in W_0^{1,\infty}([0,1])$ small. With $\Omega_{1,\theta}$ defined by $\Omega_{1,\theta} := (\mathrm{Id} + \theta)(\Omega_1)$ we can also define $\Omega_{2,\theta}$ by $\Omega_{2,\theta} :=]0,1[\setminus\overline{\Omega_{1,\theta}}$. We introduce $[0,1] \ni x \mapsto \alpha_\theta(x)$ such that $\alpha_\theta = \alpha^z \circ (\mathrm{Id} + \theta)$. When $\theta = 0$, we have $\alpha_\theta = \alpha^z$. Let also $\mathcal{V} := \left\{v \in H^1(]0,1[) \mid v(0) = 0\right\}$, u_θ be the solution to

$$\begin{cases} \text{find } u_\theta \in \mathcal{V}, \text{ such that} \\ \forall v \in \mathcal{V}, \displaystyle\int_0^1 \alpha_\theta \, d_x u_\theta \, d_x v = v(1), \end{cases} \tag{15.32}$$

and $U_\theta = u_\theta \circ (\mathrm{Id} + \theta)$. The derivatives (under existence assumption) of u_θ are the Eulerian derivatives whereas those of U_θ are the Lagrangian ones. It appears that even if α_θ is not continuous at the interface between $\Omega_{1,\theta}$ and $\Omega_{2,\theta}$, U_θ is differentiable with respect to θ. This is not the case of u_θ due to the discontinuity of $d_x u_\theta$ through the interface. If u_θ were of class C^1 with respect to θ, writing the derivative at $\theta = 0$ would lead to $u'_\theta = U'_\theta - d_x u\, \theta$. This implies that the shape differentiability of u_θ is strongly related to the regularity of u_θ as the solution to (15.3) (see also [72, Remarque 5.3.6] for more details).

Proposition 15.3.3. *Let* $\theta \in W_0^{1,\infty}([0,1])$ *and* u_θ *the solution to (15.32). Let also* $U_\theta = u_\theta \circ (\mathrm{Id} + \theta)$. *Then* U_θ *is differentiable with respect to* θ *and its derivative at* 0 *in the direction* θ, *denoted by* U'_θ, *is solution to*

$$\begin{cases} \text{find } U'_\theta \in \mathcal{V}, \text{ such that} \\ \forall v \in \mathcal{V}, \displaystyle\int_0^1 \alpha^z \, d_x U'_\theta \, d_x v = \int_0^1 \alpha^z d_x u \, d_x v \, d_x \theta, \end{cases} \tag{15.33}$$

Furthermore, for $\theta \in W_0^{1,\infty}([0,1])$ *and* $(U'_\theta)_\xi = U'_\theta \circ (\mathrm{Id} + \xi)$, $(U'_\theta)_\xi$ *is differentiable with respect to* ξ *and its derivative at* 0 *in the direction* ξ *- denoted as* $(U'_\theta)'_\xi$ *- is solution to*

$$\begin{cases} \text{find } (U'_\theta)'_\xi \in \mathcal{V}, \text{ such that} \\ \forall v \in \mathcal{V}, \displaystyle\int_0^1 \alpha^z \, d_x (U'_\theta)'_\xi \, d_x v = \int_0^1 \alpha^z d_x U'_\theta \, d_x v \, d_x \xi \\ \qquad\qquad + \displaystyle\int_0^1 \alpha^z d_x U'_\xi \, d_x v \, d_x \theta - 2\int_0^1 \alpha^z d_x u \, d_x v \, d_x \theta \, d_x \xi. \end{cases}$$

Proof. The main argument in this proof is the implicit function theorem. By assumption, u_θ satisfies

$$\forall v \in \mathcal{V}, \int_{\Omega_{1,\theta} \cup \Omega_{2,\theta}} \alpha_\theta \, d_x u_\theta \, d_x v = v(1).$$

We also denote $V = v \circ (\mathrm{Id} + \theta)$. After a change of variable this leads to

$$\forall V \in \mathcal{V},\ \int_{\Omega_1 \cup \Omega_2} \alpha^z\, d_x U_\theta (\mathrm{Id} + d_x\theta)^{-1}\, d_x V (\mathrm{Id} + d_x\theta)^{-1} |\det(\mathrm{Id} + d_x\theta)| = V(1).$$

Applying the implicit function theorem (see [72, Théorème 5.5.1] for more details), we deduce that U_θ is differentiable with respect to θ. Writing the first-order Taylor expansion with respect to θ leads to

$$\begin{aligned} U_\theta &= u + U'_\theta + o(\theta), \\ (\mathrm{Id} + d_x\theta)^{-1} &= \mathrm{Id} - d_x\theta + o(\theta), \\ |\det(\mathrm{Id} + d_x\theta)| &= \mathrm{Id} + d_x\theta, \end{aligned}$$

and

$$V(1) = \int_0^1 \alpha^z d_x u\, d_x V + \int_0^1 \alpha^z d_x U'_\theta\, d_x V - \int_0^1 \alpha^z d_x u\, d_x V\, d_x\theta + o(\theta).$$

The function u being solution to (15.3) we get the expected result. The proof for the differentiability of U'_θ is the same. □

Remark 15.3.4. *Since α^z is constant in each domain Ω_1 and Ω_2, u_θ is regular in each domain. The derivative of u jumps at the interface, but elsewhere u_θ is smooth (which can be checked with the explicit formula for u_θ). Therefore, taking the derivative with respect to θ at $\theta = 0$, the relation $u'_\theta = U'_\theta - d_x u\, \theta$ indicates that the Eulerian derivative exists at least at 1, and*

$$u'_\theta(1) = U'_\theta(1) - d_x u(1)\, \theta(1),$$

which, since $\theta(1) = 0$ leads to

$$u'_\theta(1) = U'_\theta(1). \tag{15.34}$$

Lemma 15.3.5. *According to Proposition 15.3.3, for $\theta \in W_0^{1,\infty}([0,1])$, u_θ is differentiable with respect to θ. The Lagrangian derivative U'_θ satisfies also*

$$\alpha^z d_x U'_\theta = \alpha^z d_x u\ d_x\theta.$$

Proof. Integrating by part the variational formulation for U'_θ (15.33) and taking the test function v with compact support in $[0,1]$ leads to

$$d_x\big(\alpha^z d_x U'_\theta\big) = d_x\big(\alpha^z d_x u\ d_x\theta\big).$$

Choosing a function v vanishing at $z, 0$ or 1 gives the result. □

Differentiation of the objective function

Now we can consider the objective function J and its shape differentiability.

Proposition 15.3.6. *The function J defined by $J(\Omega_1) = (u(1) - u_0)^2$ is shape-differentiable. For $\theta \in W_0^{1,\infty}([0,1])$ its shape derivative in the direction θ reads*

$$J'(\Omega_1; \theta) = [\alpha^z\, d_x u\, d_x p\, \theta], \tag{15.35}$$

where p is solution to

$$\left\{ \begin{array}{rcll} d_x\big(\alpha^z d_x p\big) & = & 0 & \text{in } \Omega, \\ \big(\alpha^z d_x p\big)(1) & = & 2(u(1) - u_0), & \\ p(0) & = & 0. & \end{array} \right. \tag{15.36}$$

The use of an adjoint state is not really necessary in this one-dimensional example. Indeed, one can easily notice that the adjoint state satisfies $p = 2(u(1) - u_0)u$. We nevertheless introduce the adjoint state in order to keep the similarity with the way of computing shape derivatives in higher dimensions.

Proof. The differentiability of the objective function J sums up to the existence of the Eulerian derivative of u at 1. Since u is smooth in the neighborhood of 1 one has for $\theta \in W_0^{1,\infty}([0,1])$,

$$u'_\theta(1) = U'_\theta(1).$$

Therefore

$$J'(\Omega_1; \theta) = 2(u(1) - u_0) U'_\theta(1).$$

The adjoint state p solution to (15.36) is solution of the following variational formulation

$$\begin{cases} \text{find } p \in \mathcal{V}, \text{ such that} \\ \forall v \in \mathcal{V}, \ \displaystyle\int_0^1 \alpha^z \, d_x p \, d_x v = 2(u(1) - u_0)v(1). \end{cases}$$

Therefore, with $v = U'_\theta$, one has

$$J'(\Omega_1; \theta) = \int_0^1 \alpha^z \ d_x p \ d_x U'_\theta.$$

The variational formulation for U'_θ leads to

$$J'(\Omega_1; \theta) = \int_0^1 \alpha^z d_x u \ d_x p \ d_x \theta.$$

Since u is smooth and α^z constant on Ω_1 and Ω_2, one has $d^2_{xx} u = 0$ and

$$\int_0^1 \alpha^z d_x u \ d_x p \ d_x \theta = \int_{\Omega_1 \cup \Omega_2} \alpha^z d_x p \ d_x(\theta d_x u).$$

Integrating by parts with $d_x(\alpha^z d_x p) = 0$ and $\theta(1) = \theta(0) = 0$, this leads to

$$\begin{aligned} J'(\Omega_1; \theta) &= [\alpha d_x p \ d_x u \ \theta]_0^z + [\alpha d_x p \ d_x u \ \theta]_z^1, \\ &= [\alpha d_x p \ d_x u \ \theta]. \end{aligned}$$

□

Lemma 15.3.7. *The shape derivative of the objective function can also be rewritten as*

$$J'(\Omega_1; \theta) = (\alpha d_x u(z))(\alpha d_x p(z))[\gamma]\theta(z).$$

Proof. We have

$$[\alpha d_x p \ d_x u \ \theta] = \alpha_1 d_x p(z^-) \ d_x u(z^-)\theta(z^-) - \alpha_2 d_x p(z^+) \ d_x u(z^+)\theta(z^+).$$

Since $\alpha d_x p$, $\alpha d_x u$ and θ do not jump at z this leads to

$$\begin{aligned} [\alpha d_x p \ d_x u \ \theta] &= \theta(z)(\alpha d_x p)(z) \ \Big(d_x u(z^-) - d_x u(z^+) \Big) \\ &= \theta(z)(\alpha d_x p)(z) \ \left(\frac{1}{\alpha_1} \Big(\alpha_1 d_x u(z^-) - \alpha_2 d_x u(z^+) \Big) + \alpha_2 d_x u(z^+) \Big(\frac{1}{\alpha_1} - \frac{1}{\alpha_2} \Big) \right) \\ &= \theta(z)(\alpha d_x p)(z) \ (\alpha d_x p)(z) \ [\gamma]. \end{aligned}$$

□

It is interesting to compare here the results for the shape derivatives with some other results we can find in the literature for higher-dimensional cases. In [99], Pantz considers the heat equation in $\mathbb{R}^2$ with discontinuous conductivity (D). The domain Ω is composed of two domains Ω_1 and Ω_2 separated by an interface Γ such that Ω_1 is an inside domain, i.e $\partial\Omega_1 = \Gamma$ and $\Gamma \cap \partial\Omega = \emptyset$. The partial differential equation reads

$$\begin{cases} -\mathrm{div}\big(D^i \nabla u\big) &= f & \text{in } \Omega^i (i = 1, 2), \\ D^2 \nabla_x u \, \mathbf{n} &= g & \text{on } \Gamma_N, \\ u &= 0 & \text{on } \Gamma_D. \end{cases}$$

For the criterion $J(\Omega) = \displaystyle\int_\omega |u - u_0|^2$, the shape derivative in the direction $\theta \in W^{1,\infty}(\Omega)$ reads (the convention for the jump is taken in the other sense in the paper)

$$J'(\Omega; \theta) = -[D] \int_\Gamma \nabla_\Gamma u \nabla_\Gamma p \ \ (\theta \cdot \mathbf{n}) + [D^{-1}] \int_\Gamma D\partial_{\mathbf{n}} u \, D\partial_{\mathbf{n}} p \, (\theta \cdot \mathbf{n}) \, .$$

Since in one dimension there is no tangential component, this formula is consistent with ours.

Proposition 15.3.8. *The function J defined by $J(\Omega_1) = (u(1) - u_0)^2$ is twice shape-differentiable. For $\theta, \xi \in W_0^{1,\infty}([0,1])$, the second-order shape derivative in the direction θ reads*

$$J''(\Omega_1; \theta, \xi) = 2 \ [\alpha^z \, d_x u \, d_x p \, \theta] \ [\alpha^z \, d_x u \, d_x p \, \xi]. \tag{15.37}$$

where p is solution to (15.36).

Proof. Let $\theta, \xi \in W_0^{1;\infty}([0,1])$. The objective function J is shape-differentiable by Proposition 15.3.6 and we have

$$J'(\Omega_1;\theta) = 2(u(1)-u_0)U'_\theta(1).$$

The second-order differentiability is also given by the shape differentiability of u and U'_θ stated in Proposition 15.3.3. Therefore we have

$$J''(\Omega_1;\theta,\xi) = 2(u(1)-u_0)(U'_\theta)'_\xi(1) + 2U'_\theta(1)U'_\xi(1).$$

Similarly to the proof of Proposition 15.3.6, one can prove that $U'_\theta(1) = [\alpha^z\, d_x u\, d_x p\, \theta]$, and therefore

$$2\, U'_\theta(1)U'_\xi(1) = 2\,[\alpha^z\, d_x u\, d_x p\, \theta]\;[\alpha^z\, d_x u\, d_x p\, \xi].$$

It remains to prove that the first part of J'' vanishes. The variational formulation for the adjoint state p ensures that

$$2(u(1)-u_0)(U'_\theta)'_\xi(1) = \int_0^1 \alpha^z\, d_x p\, d_x(U'_\theta)'_\xi.$$

With the help of the variational formulation for $(U'_\theta)'_\xi$, we get

$$2(u(1)-u_0)(U'_\theta)'_\xi(1) = \int_0^1 \alpha^z d_x U'_\theta\, d_x v\, d_x\xi + \int_0^1 \alpha^z d_x U'_\xi\, d_x v\, d_x\theta - 2\int_0^1 \alpha^z d_x u\, d_x v\, d_x\theta\, d_x\xi,$$

which vanishes by virtue of Lemma 15.3.5. □

Lemma 15.3.9. *Similarly to Lemma 15.3.7, the second-order shape derivative may be rewritten as*

$$J''(\Omega_1;\theta,\xi) = 2(\alpha^z d_x u)\;(\alpha^z d_x p)[\gamma]^2\;\theta(z)\xi(z).$$

Remark 15.3.10. *Theoretically the solution u to (15.3) and the adjoint state p satisfy $\alpha d_x u = 1$ and $p = 2(u(1)-u_0)$ everywhere. Therefore the shape derivatives of the objective function read*

$$\begin{aligned} J'(\Omega_1;\theta) &= 2(u(1)-u_0)[\gamma]\theta(z), \\ J''(\Omega_1;\theta,\xi) &= 2[\gamma]^2\;\theta(z)\xi(z). \end{aligned}$$

These are the exact same formulas as in the parametric case.

Remark 15.3.11. *In the one-dimensional case, there is associativity on the variation of a shape by diffeomorphisms of the form* $\mathrm{Id}+\theta$. *Thus, contrary to higher-dimensional cases (see Remark 3.2.14), the second-order derivative is given by the derivative of the first-order derivative.*

Discretization of the shape derivatives

In this section we detail the numerical approximation of the shape derivatives in view of the next section where we focus on the numerical analysis of an optimization method. At first we check that the property $\alpha^z d_x u = 1$ is numerically true.

Lemma 15.3.12. *Let u_h^z be the solution to (15.11). We have everywhere*

$$\alpha_h^z d_x u_h^z = 1.$$

Proof. We have by Proposition 15.2.1, $u_h^z \in \mathbb{P}_1$ and

$$\begin{aligned} u_{h,0}^z &= 0, \\ u_{h,i}^z &= i\gamma_1 h, \quad \forall i \le \left\lfloor \frac{z}{h} \right\rfloor, \\ u_{h,i}^z &= u_{h,N}^z - \gamma_2 h\,(N-i), \quad \forall i \ge \left\lceil \frac{z}{h} \right\rceil, \\ u_{h,N}^z &= \gamma_2 + [\gamma]z + h\left(\frac{1}{\langle\alpha_h^z\rangle} - \left(\eta_h^z\gamma_1 + (1-\eta_h^z)\gamma_2\right)\right). \end{aligned}$$

Therefore, $d_x u^h$ is constant in each cell and, for each cell $](i-1)h, ih[$ with $i \le \left\lfloor \frac{z}{h} \right\rfloor$ or $i > \left\lceil \frac{z}{h} \right\rceil$, $\alpha^z d_x u_h^z = \alpha^z\,\gamma^z = 1$. It remains to check the formula for the cell $](i-1)h, ih[$ with $i = \left\lceil \frac{z}{h} \right\rceil$. We have

$$\begin{aligned} u_{h,i-1}^z &= (i-1)\gamma_1 h, \\ u_{h,i}^z &= u_{h,N}^z - (N-i)\gamma_2 h. \end{aligned}$$

Therefore, with $d_x u_h(z) = \dfrac{u_{h,i} - u_{h,i-1}}{h}$ one has

$$h d_x u_h(z) = \gamma_2 + [\gamma] z + h \left(\frac{1}{\langle \alpha_h^z \rangle} - \left(\eta_h^z \gamma_1 + (1 - \eta_h^z) \gamma_2 \right) \right) - (N - i)\, h\, \gamma_2 - (i-1)\, h\, \gamma_1.$$

Since $N\, h = 1$ and $z = h\eta_h^z + h(i-1)$, this leads to

$$\begin{aligned} h d_x u_h(z) &= [\gamma] z + h \left(\frac{1}{\langle \alpha_h^z \rangle} - \left(\eta_h^z [\gamma] + \gamma_2 \right) \right) - i\, h\, [\gamma] + h\, \gamma_1 \\ &= -h[\gamma] + \frac{h}{\langle \alpha_h^z \rangle} + h[\gamma] \\ &= \frac{h}{\langle \alpha_h^z \rangle}, \end{aligned}$$

and

$$\begin{aligned} \langle \alpha_h^z \rangle d_x u_h(z) &= 1, \\ \alpha_h^z(z) d_x u_h(z) &= 1. \end{aligned}$$

□

Therefore the discretization of the shape derivatives gives rise to

$$\begin{aligned} J'(\Omega_1; \theta) \quad &\simeq 2(u_h(1) - u_0)[\gamma] \theta(z), \\ J''(\Omega_1; \theta, \xi) \quad &\simeq 2[\gamma]^2\, \theta(z) \xi(z). \end{aligned} \tag{15.38}$$

15.4 Optimization with the Newton method

Starting from an interface point z^0 ($n \in \mathbb{N}$) the optimization process consists in building a sequence z^n ($n \in \mathbb{N}$) that minimizes the objective function, either J or J_h. The main objective of this section is to specify, for $n \in \mathbb{N}$ the relation between z^n and z^{n+1}.

Algorithm 15.4.1. *Newton's Algorithm*

1. *Set $n = 0$. Choose $z^0 \in [0,1]$ and a convergence threshold $\varepsilon > 0$*
2. *Compute a descent direction dz^n.*
3. *Set $z^{n+1} = z^n + dz^n$.*
4. *Set $n := n + 1$. If $|z^n - z^\star| \leq \varepsilon$ (or $|z^n - z_h^\star| \leq \varepsilon$ for the discretized version) stop. Else go to step 2.*

The Newton method is well known to converge quadratically under regularity conditions of the objective [92, Theorem 3.7]. It means that we have an estimate (with $C > 0$)

$$|z^{n+1} - z^\star| \leq C |z^n - z^\star|^2.$$

15.4.1 Parametric framework

When the variable is $z^n \in [0,1]$ the Newton method consists in making a step dz^n such that

$$d_{zz}^2 J(z^n) \cdot dz^n = -d_z J(z^n), \tag{15.39}$$

with a unitary time step. In the one-dimensional example considered here, the second-order derivative of J with respect to J is $2[\gamma]$ which does not vanish. Therefore there is no singularity of the Hessian. This leads to the relation

$$z^{n+1} = z^n - \frac{d_z J(z^n)}{d_{zz}^2 J(z^n)}.$$

15.4.2 Shape variation framework

In the shape variation framework, we use the level-set method. The shape Ω_1 is represented by a level-set function. Let $n > 0$ and $z^n \in [0,1]$. We consider ϕ^n as the signed distance function to $[0, z^n]$ (which corresponds to the domain Ω_1), that is $\phi^n(x) = x - z^n$ for $x \in [0,1]$. In this context the relation between z^n and z^{n+1} is not a priori straightforward, since it is defined by the evolution of the level-set : $\phi^n \to \phi^{n+1}$. The Newton method amounts to computing a direction θ^n such that

$$\forall \xi \in W_0^{1,\infty}([0,1]),\ J''(\Omega_1;\theta^n,\xi) = -J'(\Omega_1;\xi).$$

The shape derivative formulas are a priori defined for $\theta, \xi \in W_0^{1,\infty}([0,1])$ but one can notice that they depend only on $\theta(z)$ and $\xi(z)$. Therefore the Newton equation for the shape variation does not have a unique solution. This problem disappears if we restrict the equation to the value of θ, ξ at z. In that case, the Newton equation is exactly the same as (15.39) (with $\theta^n = dz^n$). Since the second-order derivative factor $(2[\gamma])$ does not vanish, there is a unique solution in $\mathbb{R}$. The solution θ^n to the Newton equation at z^n has to be extended (see Section 10.2) to be used in the Hamilton-Jacobi equation. Let v^n be a constant extension of θ^n : $\forall x \in [0,1],\ v^n(x) = \theta^n$. Let then ϕ be the solution to the Hamilton-Jacobi equation

$$\begin{cases} \partial_t \phi(t,x) + v^n(x)|\nabla \phi(t,x)| = 0, \\ \phi(0,x) = \phi^n. \end{cases} \tag{15.40}$$

The update of the level-set is defined as $\phi^{n+1} = \phi(1,\cdot)$. Since ϕ^n is assumed to be a signed distance function the relation between ϕ^n and ϕ^{n+1} becomes here : $\phi^{n+1} = \phi^n - v^n$. Then z^{n+1} is defined as the zero of the updated level-set ϕ^{n+1}. We have obviously $z^{n+1} = z^n + \theta^n$.

Thus, for the continuous problem, both parametric and shape variation frameworks coincide. The derivatives as well as the update of the variables correspond to the same operations.

15.5 Error analysis

We start by analyzing the continuous problem, with exact derivatives. Since the objective function is quadratic, we can check that the Newton method converges in one iteration. Secondly we consider the rates of convergence for the *optimize-then-discretize* and *discretize-then-optimize* approaches. Since for this particular problem, both parametric and shape variation approaches coincide we will consider here only the case of the parametric approach.

15.5.1 Continuous framework

Proposition 15.5.1. *The Newton method for solving (15.5) converges in one iteration.*

Proof. For $n \in \mathbb{N}$ and $z^n \in [0,1]$ the next iterate z^{n+1} is computed as

$$\begin{aligned} z^{n+1} &= z^n - \frac{d_z J(z^n)}{d^2_{zz} J(z^n)}, \\ &= z^n - \frac{2(u^{z^n}(1) - u_0)[\gamma]}{2[\gamma]^2}, \\ &= z^n - \frac{u^{z^n}(1) - u_0}{[\gamma]}. \end{aligned}$$

The solution u^{z^n} to (15.3) with $z = z^n$ satisfies

$$u^{z^n}(1) = \gamma_2 + z^n[\gamma],$$

and the optimal solution $z^\star$ is defined by

$$z^\star = \frac{u_0 - \gamma_2}{[\gamma]}.$$

Therefore

$$\begin{aligned} z^{n+1} &= z^n - \frac{\gamma_2 + z^n[\gamma] - u_0}{[\gamma]}, \\ &= z^n - \frac{z^n[\gamma] - z^\star[\gamma]}{[\gamma]}, \\ &= z^\star. \end{aligned}$$

As a result, the algorithm converges in only one iteration. □

15.5.2 Optimize-then-discretize approach

The optimize-then-discretize approach consists in computing the derivatives of the continuous problem and to compute the discretization of these derivatives. The continuous and discretized optimal solution $z^\star$ and $z_h^\star$ are respectively defined by (see Proposition 15.1.1 and Proposition 15.2.3)

$$z^\star = \frac{u_0 - \gamma_2}{[\gamma]}, \quad z_h^\star = z_0 + h\eta_h^\star, \tag{15.41}$$

where

$$\eta_h^\star = \frac{1}{[\alpha]}\left(\frac{1}{\gamma_2 + [\gamma]Z} - \alpha_2\right), \quad Z = \frac{z^\star}{h} - \left\lfloor \frac{z^\star}{h} \right\rfloor, \quad z_0 = h\left\lfloor \frac{z^\star}{h} \right\rfloor.$$

The update of z^n with the Newton method is given by

$$\begin{aligned} z^{n+1} &= z^n + \theta^n, \\ &= z^n - \frac{u_h^z(1) - u_0}{[\gamma]}. \end{aligned}$$

Proposition 15.5.2. *With the* optimize-then-discretize *approach, the evolution of the error satisfies*

$$\eta^{n+1} - \eta_h^\star = (\eta^n - \eta_h^\star)\left(1 - \frac{\alpha_1\,\alpha_2}{(\eta_h^\star[\alpha] + \alpha_2)(\eta^n[\alpha] + \alpha_2)}\right). \tag{15.42}$$

Proof. After discretization, we assume that z^n is close to the optimal continuous solution, and that $\left\lfloor \frac{z^n}{h} \right\rfloor = \left\lfloor \frac{z^\star}{h} \right\rfloor$. Therefore, (15.29) is valid :

$$u_h^n(1) - u_0 = h\left(\frac{1}{\eta^n[\alpha] + \alpha_2} - \frac{1}{\eta_h^\star[\alpha] + \alpha_2}\right),$$

which means that

$$[\gamma](z^{n+1} - z^n) = h\left(\frac{1}{\eta_h^\star[\alpha] + \alpha_2} - \frac{1}{\eta^n[\alpha] + \alpha_2}\right).$$

Writing $z^n = z_0 + \eta^n h$ with $z_0 = h\left\lfloor \frac{z^\star}{h} \right\rfloor$, this leads to

$$\begin{aligned} [\gamma](\eta^{n+1} - \eta^n) &= \left(\frac{1}{\eta_h^\star[\alpha] + \alpha_2} - \frac{1}{\eta^n[\alpha] + \alpha_2}\right), \\ [\gamma](\eta^{n+1} - \eta_h^\star) &= [\gamma](\eta^n - \eta^\star) + \frac{(\eta^n - \eta_h^\star)[\alpha]}{(\eta_h^\star[\alpha] + \alpha_2)(\eta^n[\alpha] + \alpha_2)}, \\ \eta^{n+1} - \eta_h^\star &= (\eta^n - \eta_h^\star)\left(1 + \frac{\frac{[\alpha]}{[\gamma]}}{(\eta_h^\star[\alpha] + \alpha_2)(\eta^n[\alpha] + \alpha_2)}\right). \end{aligned}$$

Since the jump ratio between α and γ is

$$\frac{[\alpha]}{[\gamma]} = \frac{\alpha_1 - \alpha_2}{\gamma_1 - \gamma_2} = -\alpha_1\,\alpha_2,$$

we finally get

$$\eta^{n+1} - \eta_h^\star = (\eta^n - \eta_h^\star)\left(1 - \frac{\alpha_1\,\alpha_2}{(\eta_h^\star[\alpha] + \alpha_2)(\eta^n[\alpha] + \alpha_2)}\right). \tag{15.43}$$

□

Remark 15.5.3. *At first, with $\alpha_1, \alpha_2 \in \mathbb{R}_+$ there is a priori no reason why we should have*

$$\left|1 - \frac{\alpha_1\,\alpha_2}{(\eta_h^\star[\alpha] + \alpha_2)(\eta^n[\alpha] + \alpha_2)}\right| = O(\eta^n - \eta^\star), \tag{15.44}$$

which would ensure a second-order convergence rate as soon as $|\eta^n - \eta^\star| \ll 1$. In fact, the left-hand side has no reason to converge to zero so that the expected convergence is at most linear. Moreover there is also no reason why this term should be between -1 and 1, which would ensure that the sequence $|\eta^n - \eta^\star|$ converges. Indeed, if $(\eta_h^n[\alpha] + \alpha_2) = (\eta_h^\star[\alpha] + \alpha_2) = \alpha_1$ and $\alpha_1 < \frac{\alpha_2}{2}$ we have

$$\left|1 - \frac{\alpha_1\,\alpha_2}{(\eta_h^\star[\alpha] + \alpha_2)(\eta^n[\alpha] + \alpha_2)}\right| > 1.$$

In such a case, the algorithm may even not converge to the optimal solution.

15.5.3 Discretize-then-optimize approach

Contrary to the previous approach, the main idea here is to compute the discretized model first and then to deduce the derivatives of the discrete model. We have

$$d_z J_h(z^n) = -2h\left(\frac{1}{\eta^n[\alpha]+\alpha_2} - \frac{1}{\eta_h^\star[\alpha]+\alpha_2}\right)\frac{[\alpha]}{\left(\eta^n[\alpha]+\alpha_2\right)^2},$$
$$d_{zz}^2 J_h(z^n) = 2\frac{[\alpha]^2}{\left(\eta^n[\alpha]+\alpha_2\right)^4} + 4\left(\frac{1}{\eta^n[\alpha]+\alpha_2} - \frac{1}{\eta_h^\star[\alpha]+\alpha_2}\right)\frac{[\alpha]^2}{\left(\eta^n[\alpha]+\alpha_2\right)^3}.$$

In order to ease the reading we denote in this section

$$\alpha^n = \eta^n[\alpha]+\alpha_2, \quad \alpha^\star = \eta_h^\star[\alpha]+\alpha_2.$$

Proposition 15.5.4. *The* discretize-then-optimize *approach has a second-order rate of convergence.*

Proof. The Newton equation gives

$$dz^n = \frac{h\left(\frac{1}{\alpha^n}-\frac{1}{\alpha^\star}\right)\frac{[\alpha]}{(\alpha^n)^2}}{\frac{[\alpha]^2}{(\alpha^n)^4} + 2\left(\frac{1}{\alpha^n}-\frac{1}{\alpha^\star}\right)\frac{[\alpha]^2}{(\alpha^n)^3}}.$$

With $z^{n+1}-z^n = h(\eta^{n+1}-\eta^n) = dz^n$, this reads

$$\eta^{n+1}-\eta^n = \frac{\left(\frac{1}{\alpha^n}-\frac{1}{\alpha^\star}\right)(\alpha^n)^2}{[\alpha] + 2\left(\frac{1}{\alpha^n}-\frac{1}{\alpha^\star}\right)[\alpha](\alpha^n)}.$$

We also have

$$\begin{aligned}\frac{1}{\alpha^n}-\frac{1}{\alpha^\star} &= \frac{1}{\eta^n[\alpha]+\alpha_2} - \frac{1}{\eta_h^\star[\alpha]+\alpha_2}\\ &= (\eta^\star-\eta^n)\frac{[\alpha]}{\alpha^n\,\alpha^\star}.\end{aligned}$$

Therefore

$$\begin{aligned}\eta^{n+1}-\eta^n &= \frac{(\eta^\star-\eta^n)\frac{[\alpha]}{\alpha^\star}\alpha^n}{[\alpha] + 2(\eta^\star-\eta^n)\frac{[\alpha]^2}{\alpha^\star}}\\ &= (\eta^\star-\eta^n)\frac{\frac{\alpha^n}{\alpha^\star}}{1+2(\eta^\star-\eta^n)\frac{[\alpha]}{\alpha^\star}}.\end{aligned}$$

Consequently

$$\eta^{n+1}-\eta^\star = (\eta^n-\eta^\star)\left(1 - \frac{\frac{\alpha^n}{\alpha^\star}}{1+2(\eta^\star-\eta^n)\frac{[\alpha]}{\alpha^\star}}\right).$$

If $\eta^\star-\eta^n$ is small, we can write

$$1 - \frac{\frac{\alpha^n}{\alpha^\star}}{1+2(\eta^\star-\eta^n)\frac{[\alpha]}{\alpha^\star}} = 1 - \frac{\alpha^n}{\alpha^\star} + 2\frac{\alpha^n}{\alpha^\star}\frac{[\alpha]}{\alpha^\star}(\eta^\star-\eta^n) + o(\eta^\star-\eta^n).$$

We also have

$$\begin{aligned}1-\frac{\alpha^n}{\alpha^\star} &= 1 - \frac{\eta^n[\alpha]+\alpha_2}{\eta^\star[\alpha]+\alpha_2}\\ &= \frac{[\alpha^z]}{\alpha^\star}(\eta^\star-\eta^n).\end{aligned}$$

Therefore

$$\eta^{n+1}-\eta^\star = -(\eta^n-\eta^\star)^2\frac{[\alpha^z]}{\alpha^\star}\left(1+2\frac{\alpha^n}{\alpha^\star}\right) + o(\eta^n-\eta^\star)^2.$$

□

In the *discretize-then-optimize* framework, the convergence is then of order 2. This is totally expected since the optimization takes place in a parametric case where the Newton method is known to converge quadratically.

15.5.4 Impact of numerical approximation on the convergence rate

In the *optimize-then-discretize* approach we saw that due to approximation errors the convergence was only first-order, even with the Newton method. This is due to the approximation of the state u in the first-order derivative. In the particular example presented above, it appears that there is no approximation error on the second-order derivative. In this paragraph we focus on the continuous problem and assume that there is respectively a numerical error δ_1 and δ_2 on the computation of the first and second-order derivatives of the criterion. In other words, we have

$$(d_z J)_h = d_z J + \delta_1, \quad (d^2_{zz} J)_h = d^2_{zz} J + \delta_2.$$

Therefore, starting from z^n with $n \in \mathbb{N}$ the next iterate is computed as

$$\begin{aligned} z^{n+1} &= z^n - \frac{(d_z J(z^n))_h}{(d^2_{zz} J(z^n))_h}, \\ &= z^n - \frac{(u^{z^n}(1) - u_0)[\gamma] + \delta_1(z^n)}{[\gamma]^2 + \delta_2(z^n)}. \end{aligned}$$

Since $u^{z^n}(1) = \gamma_2 + z^n[\gamma]$ and $z^\star = \dfrac{u_0 - \gamma_2}{[\gamma]}$ this leads to

$$\begin{aligned} z^{n+1} &= z^n - \frac{(z^n - z^\star)[\gamma] + \delta_1(z^n)}{[\gamma]^2 + \delta_2(z^n)}, \\ &= z^n - (z^n - z^\star)\frac{[\gamma]}{[\gamma]^2 + \delta_2(z^n)} - \frac{\delta_1(z^n)}{[\gamma]^2 + \delta_2(z^n)}, \\ &= z^\star + (z^n - z^\star)\frac{\delta_2}{[\gamma]^2 + \delta_2(z^n)} - \frac{\delta_1(z^n)}{[\gamma]^2 + \delta_2(z^n)}. \end{aligned}$$

Therefore

$$z^{n+1} - z^\star = (z^n - z^\star)\frac{\delta_2(z^n)}{[\gamma]^2 + \delta_2(z^n)} - \frac{\delta_1(z^n)}{[\gamma]^2 + \delta_2(z^n)}.$$

First, let assume that the gradient is exactly computed, i.e $\delta_1 = 0$. We get

$$z^{n+1} - z^\star = (z^n - z^\star)\frac{\delta_2(z^n)}{[\gamma]^2 + \delta_2(z^n)}.$$

Therefore, the second-order convergence rate is conserved if and only if $\delta_2(z^n) = O(z^n - z^\star)$. On the other hand, when $\delta_2(z^n) = 0$, the convergence is quadratic if and only if $\delta_1(z^n) = O(z^n - z^\star)^2$. In particular, we can observe that it is much more important to have a precise computation of the gradient than of the second-order derivative to get quadratic convergence.

15.6 Numerical experiments

We would like now to see if these theoretical results can be observed in numerical practice. We take

$$\alpha_1 = 0.5,\ \alpha_2 = 2,\ N = 10^4,\ h = 10^{-4},\ z_0 = \frac{1}{3}.$$

We will compare the continuous, *optimize-then-discretize* and *discretize-then-optimize* approaches. The continuous approach consists in minimizing the function J defined by (15.7) where the dependency with respect to z is explicit. The *optimize-then-discretize* minimizes $J_h = (u_h(1) - u_0)^2$ with the derivatives given by (15.38). Finally, the *discretize-then-optimize* approach minimizes also $J_h = (u_h(1) - u_0)^2$ but with the derivatives given by Proposition 15.3.1.

15.6.1 Case when $z^\star = z_h^\star$

First, we take $u_0 = 0.5$. It appears that the theoretical optimal solutions of the continuous model and of the discretized model, that is respectively $z^\star$ and $z_h^\star$ coincide. We run Algorithm 15.4.1 for the three approaches. In Figure 15.2 we plot the evolution of the error $z - z^\star$ along the iterations. We also plot the evolution of the error $z^{n+1} - z^\star$ at the iteration $n+1$ with respect to the error $z^n - z^\star$ at iteration n, like in Section 12.1.2. In the caption of Figure 15.2, the *optimize-then-discretize* and *discretize-then-optimize* approaches are respectively referred to as `OptDisc` and `DiscOpt`.

In accordance with the theoretical results obtained in Section 15.5, we can observe that the continuous approach converges in one iteration. The *discretize-then-optimize* approach has also a quadratic rate of convergence whereas the *optimize-then-discretize* approach converges only linearly. In Table 15.1, we can indeed see that the slope of the linear regression are respectively about two and one for these two approaches.

15.6.2 Case when $z^\star \neq z_h^\star$

According to Proposition 15.2.4, the error between $z^\star$ and $z_h^\star$ should be only of order h. We also mentioned in Remark 15.5.3 that the *optimize-then-discretize* approach may even not converge in some cases. We would like thus find u_0 such that

$$\left|1 - \frac{\alpha_1\,\alpha_2}{(\eta_h^\star[\alpha] + \alpha_2)(\eta^n[\alpha] + \alpha_2)}\right| > 1,$$

at least when n is sufficiently large. We then look for $z^\star$ such that

$$\frac{\alpha_1\,\alpha_2}{(\eta_h^\star[\alpha] + \alpha_2)^2} > 2. \tag{15.45}$$

Since $\dfrac{1}{\eta_h^\star[\alpha] + \alpha_2} = 2[\gamma] + \gamma_2$, taking $Z > \dfrac{\sqrt{2\alpha_1\alpha_2} - \alpha_1}{\alpha_2 - \alpha_1} =: Z_0$ (we recall that $Z = \frac{z^\star}{h} - \lfloor \frac{z^\star}{h} \rfloor$) ensures that (15.45) is satisfied (we can check that $Z_0 \in [0, 1]$). Thus, we take $Z = Z_0 + \dfrac{1}{2}(1 - Z_0)$. We then take $u_0 = (100 + Z)h[\gamma] + \gamma_2$ so that $z^\star = \dfrac{u_0 - \gamma_2}{[\gamma]}$ and $Z = \dfrac{z^\star}{h} - \left\lfloor \dfrac{z^\star}{h} \right\rfloor$. Thus (15.45) is satisfied, and we can observe on Figure 15.3 that the *optimize-then-discretize* approach does not converge. On the other hand, the continuous approach still converges in one iteration to the theoretical continuous optimal solution $z^\star$. Similarly, the behavior of the *discretize-then-optimize* approaches is not modified. It still converges to the theoretical discrete solution $z_h^\star$ with a quadratic, or at least superlinear, rate : the slope of the linear regression on the error is 1.72.

In that particular example, we have $Z \neq 0$ and $Z \neq 1$, so that in accordance with the proof of Proposition 15.2.4, the error between $z^\star$ and $z_h^\star$ may not be zero. Indeed, we have

$$z^\star - z_h^\star = 1.38 \times 10^{-5},$$

which is consistent with the fact that the error is of the order of h.

15.6.3 Impact of numerical approximation

Finally we consider the impact of numerical errors on the first and second-order derivatives on the rate of convergence. In Section 15.5.4 we saw that the respective errors on the first and second-order derivatives should be respectively of second and first-order with respect to $z^n - z^\star$ in order to keep the quadratic rate of convergence. To that end, we consider two cases. For the first one, we add $0.1(z^n - z^\star)$ to the first-order derivative. This case is referred to as `Test 1`. For the second case, referred to as `Test 2`, we respectively add $0.1(z^n - z^\star)^2$ and $z^n - z^\star$ to the first and second-order derivatives. We now observe the convergence of the error with the optimal solution - $z^n - z^\star$ - and the evolution of the rate of convergence and the respective linear regressions. We display the slopes of the linear regressions in Table 15.3. Like it was theoretically anticipated, the addition of a first-order error on the first-order derivative (case of `Test 1`) makes the rate of convergence go down to first-order. To the contrary, adding errors of second and first-orders respectively to the first and second-order derivatives does not jeopardize the quadratic rate of convergence.

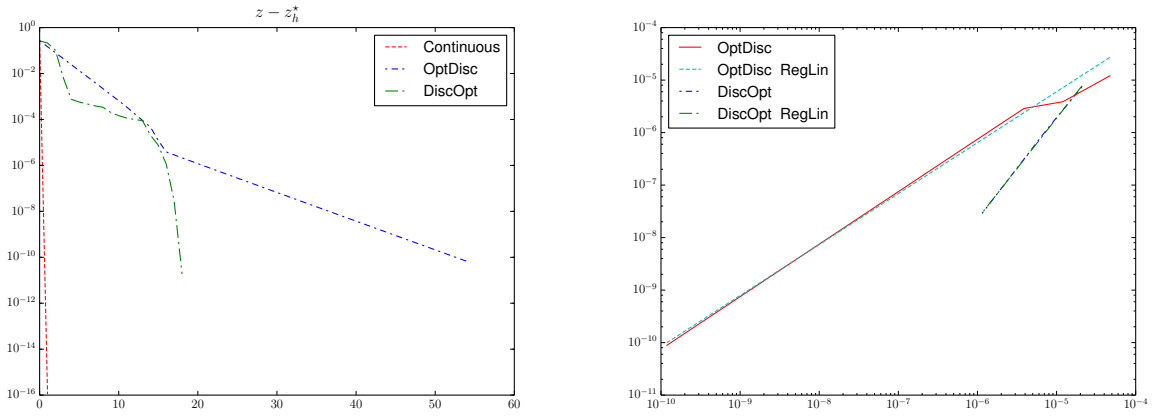


Figure 15.2: Evolution of the error to the optimal solution (left) and linear regression on this error (right).

	OptDisc	DiscOpt
Slope	0.97	1.92

Table 15.1: Slopes of the linear regression.

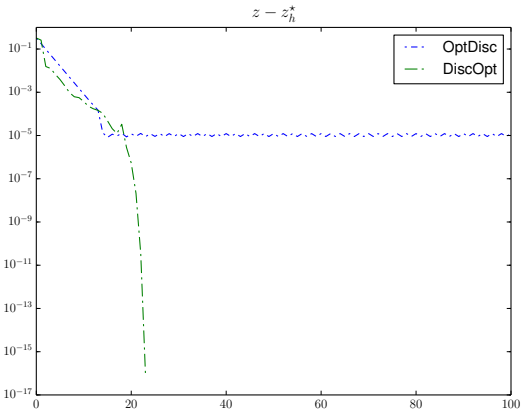


Figure 15.3: Convergence of the error with the optimal solution.

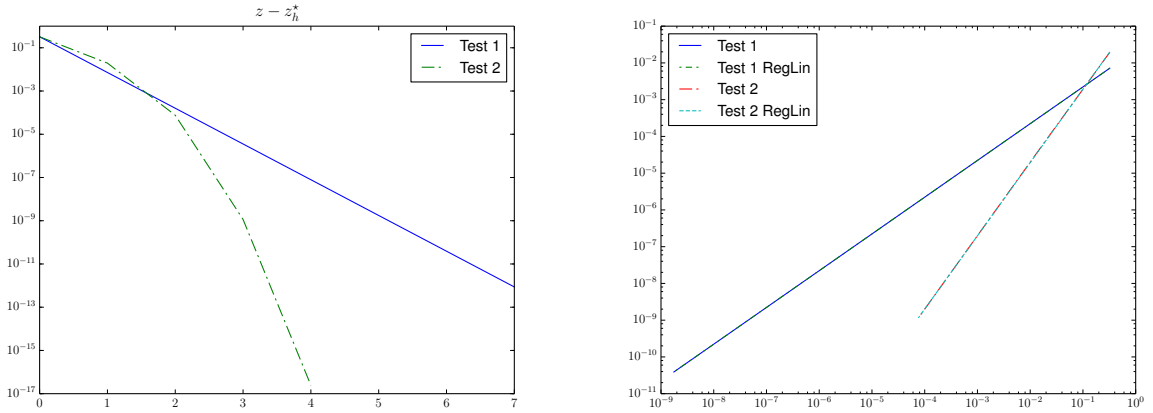


Figure 15.4: Convergence of the error with the optimal solution (left), and rates of convergence (right).

	Test 1	Test 2
Slope	1.00	1.99

Table 15.3: Slopes of the linear regressions.

Bibliography

[1] R. Abgrall. Numerical discretization of the first-order Hamilton-Jacobi equation on triangular meshes. *Communications on Pure and Applied Mathematics*, 49(12):1339–1373, 1996.

[2] R. Abgrall. Construction of simple, stable, and convergent high order schemes for steady first-order Hamilton-Jacobi equations. *SIAM Journal on Scientific Computing*, 31(4):2419–2446, 2009.

[3] D. Adalsteinsson and J.-A. Sethian. The fast construction of extension velocities in level-set methods. *Journal of Computational Physics*, 148(1):2–22, 1999.

[4] L. Afraites, M. Dambrine, and D. Kateb. On second-order shape optimization methods for electrical impedance tomography. *SIAM Journal of Control Optimization*, 47(3):1556–1590, 2007.

[5] G. Allaire. Homogenization and two-scale convergence. *SIAM Journal on Mathematical Analysis*, 23(6):1482–1518, 1992.

[6] G. Allaire. *Shape optimization by the homogenization method*, volume 146 of *Applied Mathematical Sciences*. Springer-Verlag New York, 2002.

[7] G. Allaire. *Conception optimale de structures*, volume 58 of *Mathématiques & Applications (Berlin) [Mathematics & Applications]*. Springer-Verlag, Berlin, 2007.

[8] G. Allaire. *Numerical analysis and optimization*. Numerical Mathematics and Scientific Computation. Oxford University Press, Oxford, 2007. An introduction to mathematical modelling and numerical simulation, Translated from the French by Alan Craig.

[9] G. Allaire, E. Cancès, and J.-L. Vié. Second-order shape derivatives along normal trajectories, governed by hamilton-jacobi equations. *Structural and Multidisciplinary Optimization*, pages 1–22, 2016.

[10] G. Allaire, C. Dapogny, G. Delgado, and G. Michailidis. Multi-phase structural optimization via a level-set method. *ESAIM : Control, Optimisation and Calculus of Variations*, 20(2):576–611, 2014.

[11] G. Allaire, C. Dapogny, and P. Frey. Shape optimization with a level-set based mesh evolution method. *Computer Methods in Applied Mechanics and Engineering*, 282:22–53, 2014.

[12] G. Allaire, F. Jouve, and G. Michailidis. Thickness control in structural optimization via a level-set method. working paper or preprint, April 2014.

[13] G. Allaire, F. Jouve, and G. Michailidis. Molding direction constraints in structural optimization via a level-set method. working paper or preprint, December 2015.

[14] G. Allaire, F. Jouve, and A.-M. Toader. A level-set method for shape optimization. *Comptes-Rendus Mathématiques de l'Académie des Sciences. Paris*, 334(12):1125–1130, 2002.

[15] G. Allaire, F. Jouve, and A.-M. Toader. Structural optimization using sensitivity analysis and a level-set method. *Journal of Computational Physics*, 194(1):363–393, February 2004.

[16] G. Allaire and S.-M. Kaber. *Numerical linear algebra*, volume 55 of *Texts in Applied Mathematics*. Springer, New York, 2008. Translated from the 2002 French original by Karim Trabelsi.

[17] S. Amstutz. A semismooth Newton method for topology optimization. *Nonlinear Analysis*, 73(6):1585–1595, 2010.

[18] G. Barles. Remarks on a flame propagation model. Research Report RR-0451, INRIA, 1985.

[19] J.-T. Beale. A proof that a discrete delta function is second-order accurate. *Journal of Computational Physics*, 227(4):2195–2197, 2008.

[20] M.-P. Bendsøe. Optimal shape design as a material distribution problem. *Structural Optimization*, 1(4):193–202, 1989.

[21] M.-P. Bendsøe and O. Sigmund. *Topology optimization*. Springer-Verlag, Berlin, 2003. Theory, methods and applications.

[22] C. Bernardi, Y. Maday, and F. Rapetti. *Discrétisations variationnelles de problèmes aux limites elliptiques*, chapter III. Number 45 in Mathématiques et Applications. 2004. Théorème 1.2.

[23] J.-T. Betts and S.-L. Campbell. *Mathematics for industry : challenges and frontiers. A process view : practice and theory*, chapter Discretize then Optimize, pages 140 – 157. SIAM, 2003.

[24] J.-F. Bonnans, J.-C. Gilbert, C. Lemaréchal, and C.-A. Sagastizábal. *Numerical optimization : theoretical and practical aspects*. Universitext. Springer-Verlag, Berlin, second edition, 2006. Theoretical and practical aspects.

[25] J.-F. Bonnans and A. Shapiro. *Perturbation analysis of optimization problems*. Springer Series in Operations Research. Springer-Verlag, New York, 2000.

[26] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, Cambridge, 2004.

[27] H. Brezis. *Analyse fonctionnelle*. Dunod, 2005. Theorem IX.3.

[28] C.-G. Broyden. The convergence of a class of double-rank minimization algorithms. II. The new algorithm. *Journal of the Institute of Mathematics and its Applications*, 6:222–231, 1970.

[29] C.-G. Broyden. The convergence of single-rank quasi-Newton methods. *Mathematics of Computation*, 24:365–382, 1970.

[30] D. Bucur and G. Buttazzo. *Variational methods in shape optimization problems*. Progress in Nonlinear Differential Equations and their Applications, 65. Birkhäuser Boston, Inc., Boston, MA, 2005.

[31] D. Bucur and J.-P. Zolésio. Anatomy of the shape Hessian via lie brackets. *Annali di Matematica Pura ed Applicata*, IV(CLIX):315–339, 1997.

[32] M. Burger. A framework for the construction of level-set methods for shape optimization and reconstruction. *Interfaces and Free Boundaries*, 5:301–329, 2003.

[33] G. Buttazzo and G. Dal Maso. An existence result for a class of shape optimization problems. *Archive for Rational Mechanics and Analysis*, 122(2):183–195, 1993.

[34] S.-L. Campbell, J.-P. Chancelier, and R. Nikoukhah. *Modeling and simulation in Scilab/Scicos*. Springer, New York, 2006.

[35] Jean Céa. *Lectures on optimization - theory and algorithms*, volume 53 of *Tata Institute of Fundamental Research Lectures on Mathematics and Physics*. Tata Institute of Fundamental Research, Bombay, 1978.

[36] A. Chambolle. A density result in two-dimensional linearized elasticity, and applications. *Archive for Rational Mechanics and Analysis*, 167(3):211–233, 2003.

[37] C.-M. Chen and V. Thomée. The lumped mass finite element method for a parabolic problem. *The ANZIAM Journal*, 26:329–354, 1 1985.

[38] Y.-G. Chen, Y. Giga, and S. Goto. Uniqueness and existence of viscosity solutions of generalized mean curvature flow equations. *Journal of Differential Geometry*, 33(3):749–786, 1991.

[39] M. Choulli. *Une introduction aux problèmes inverses elliptiques et paraboliques*, chapter I, page 22. Number 65 in Mathématiques & Applications. SMAI, 2009.

[40] P. W. Christensen and A. Klarbring. *An introduction to structural optimization*, volume 153 of *Solid Mechanics and its Applications*. Springer, New York, 2009.

[41] P.-G. Ciarlet. *The finite element method for elliptic problems*, volume 40 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2002. Reprint of the 1978 original [North-Holland, Amsterdam; MR0520174 (58 #25001)].

[42] W.-J. Cook, W.-H. Cunningham, W.-R. Pulleyblank, and A. Schrijver. *Combinatorial optimization*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, Inc., New York, 1998. A Wiley-Interscience Publication.

[43] M. Coquerelle and S. Glockner. A fourth-order accurate curvature computation in a level-set framework for two-phase flows subjected to surface tension forces. December 2014.

[44] R. Courant. Variational methods for the solution of problems of equilibrium and vibrations. *Bulletin of the American Mathematical Society*, 49:1–23, 1943.

[45] R. Courant, K. Friedrichs, and H. Lewy. Über die partiellen Differenzengleichungen der mathematischen Physik. *Mathematische Annalen*, 100(1):32–74, 1928. (English translation, with commentaries by Lax, P.B., Widlund, O.B., Parter, S.V., in IBM J. Res. Develop. 11 (1967)).

[46] M.-G. Crandall and P.-L. Lions. Viscosity solutions of Hamilton-Jacobi equations. *Transactions of the American Mathematical Society*, 277:1–42, 1983.

[47] M.-G. Crandall and P.-L. Lions. Two approximations of solutions of Hamilton-Jacobi equations. *Mathematics of Computation*, 43(167):1–19, 1984.

[48] M. Dambrine. On variations of the shape Hessian and sufficient conditions for the stability of critical shapes. *RACSAM. Revista de la Real Academia de Ciencias Exactas Físicas y Naturales. Serie A. Matemàticas*, 96(1):95–121, 2002.

[49] M. Dambrine and J. Lamboley. Stability in shape optimization with second variation, 2014. arXiv:1410.2586v1.

[50] M. Dambrine and M. Pierre. About stability of equilibrium shapes. *Mathematical Modeling and Numerical Analysis*, 34(4):811–834, 3 2000.

[51] M. Dambrine, J. Sokolowski, and A. Zochowski. On stability analysis in shape optimization : critical shape for Neumann problem. *Control and Cybernetics*, 32(3):503–528, 2003.

[52] C. Dapogny. *Shape optimization, level-set methods on unstructured meshes and mesh evolution.* PhD thesis, Université Paris VII - Denis Diderot; Université Pierre et Marie Curie, 2013.

[53] F. de Gournay. Velocity extension for the level-set method and multiple eigenvalues in shape optimization. *SIAM Journal on Control and Optimization*, 45(1):343–367, 2006.

[54] M.-C. Delfour and J.-P. Zolésio. Anatomy of the shape Hessian. *Annali di Matematica Pura ed Applicata*, CLIX(IV):315–339, 1991.

[55] M.-C. Delfour and J.-P. Zolésio. Velocity method and Lagrangian formulation for the computation of the shape Hessian. *SIAM Journal on Control and Optimization*, 29(6):1414–1442, 1991.

[56] M.-C. Delfour and J.-P. Zolésio. *Shapes and geometries*, volume 4 of *Advances in Design and Control*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2001. Analysis, differential calculus, and optimization.

[57] K. Dems and Z. Mróz. Variational approach to first and second-order sensitivity analysis of elastic structures. *International Journal for Numerical Methods in Engineering*, 21(4):637–661, 1985.

[58] Z. Ding. A proof of the trace theorem of Sobolev spaces on Lipschitz domains. *Proceedings of the American Mathematical Society*, 124(2):591–600, 1996.

[59] B. Engquist, Tornberg A.-K., and Tsai R. Discretization of Dirac delta functions in level-set methods. *Journal of Computational Physics*, 207(1):28–51, 2005.

[60] K. Eppler and H. Harbrecht. Compact gradient tracking in shape optimization. *Computational Optimization and Applications*, 39(3):297–318, April 2008.

[61] L.-C. Evans and J. Spruck. Motion of level-sets by mean curvature. i. *Journal of Differential Geometry*, 33(3):635–681, 1991.

[62] R. Fletcher. A new approach to variable metric algorithms. *The Computer Journal*, 13(3):317–322, 1970.

[63] M. Giaquinta and S. Hildebrandt. *Calculus of variations I*, volume 310 of *Grundlehren der mathematischen Wissenschaften*. Springer Berlin Heidelberg, 2004.

[64] Y. Giga. *Surface evolution equations*, volume 99 of *Monographs in Mathematics*. Birkhäuser Verlag, Basel, 2006. A level-set approach.

[65] D. Goldfarb. A family of variable-metric methods derived by variational means. *Mathematics of Computation*, 24:23–26, 1970.

[66] J. Hadamard. Mémoire sur le problème d'analyse relatif à l'équilibre des plaques élastiques encastrées. In *Oeuvres de Jacques Hadamard*, volume II, pages 515–641. C.N.R.S., 1907.

[67] R.-T. Haftka. Second-order sensitivity derivatives in structural analysis. *Journal of American Institute of Aeronautics and Astronautics*, 20(12):1765–1766, 1982.

[68] R.-T. Haftka and Z. Gürdal. *Elements of structural optimization*, volume 11 of *Solid Mechanics and its Applications*. Kluwer Academic Publishers Group, Dordrecht, third edition, 1992.

[69] R.-T. Haftka and Z. Mroz. First and second-order sensitivity analysis of linear and nonlinear structures. *Journal of American Institute of Aeronautics and Astronautics*, 24:1187–1192, 1986.

[70] E.-J. Haug. Second-order design sensitivity analysis of structural systems. *Journal of the American Institute of Aeronautics and Astronautics*, 19(8):1087–1088, 1981.

[71] F. Hecht. New development in freefem++. *Journal of Numerical Mathematics*, 20(3-4):251–265, 2012.

[72] A. Henrot and M. Pierre. *Variation et optimisation de formes*, volume 48. Springer, 2005.

[73] M. Hinze and A. Rösch. Discretization of optimal control problems. In Günter Leugering, Sebastian Engell, Andreas Griewank, Michael Hinze, Rolf Rannacher, Volker Schulz, Michael Ulbrich, and Stefan Ulbrich, editors, *Constrained Optimization and Optimal Control for Partial Differential Equations*, volume 160 of *International Series of Numerical Mathematics*, pages 391–430. Springer Basel, 2012.

[74] C. Hu and C.-W. Shu. A discontinuous Galerkin finite element method for Hamilton-Jacobi equations. *SIAM Journal on Scientific Computing*, 21(2):666–690 (electronic), 1999.

[75] A. F. Izmailov and M. V. Solodov. *Newton-type methods for optimization and variational problems*. Springer Series in Operations Research and Financial Engineering. Springer, Cham, 2014.

[76] C. Johnson and J.-C. Nédélec. On the coupling of boundary integral and finite element methods. *Mathematics of Computation*, 35(152):1063–1079, 1980.

[77] B. Karasözen and F. Yılmaz. Optimal boundary control of the unsteady Burgers equation with simultaneous space-time discretization. *Optimal Control Applications and Methods*, 35(4):423–434, 2014.

[78] H. Kasumba and K. Kunisch. On computation of the shape Hessian of the cost functional without shape sensitivity of the state variable. *Journal of Optimization Theory and Applications*, 162(3):779–804, 2014.

[79] B. Korte and J. Vygen. *Combinatorial optimization*, volume 21 of *Algorithms and Combinatorics*. Springer, Heidelberg, fifth edition, 2012. Theory and algorithms.

[80] G. Leugering, P. Benner, Engell S., A. Griewank, H. Harbrecht, M. Hinze, R. Rannacher, and S. Ulbrich, editors. *Trends in PDE constrained optimization*, volume 165 of *International Series of Numerical Mathematics*. Birkhäuser Basel, 2014.

[81] W. MacLean. *Strongly elliptic systems and boundary integral equations*. 2000.

[82] S. Meddahi, J. Valdés, O. Menéndez, and P. Pérez. On the coupling of boundary integral and mixed finite element methods. *Journal of Computational and Applied Mathematics*, 69(1):113–124, 1996.

[83] Georgios Michailidis. *Manufacturing constraints and multi-phase shape and topology optimisation via a level-set method*. PhD thesis, Ecole Polytechnique, 2014.

[84] C. Min and F. Gibou. Geometric integration over irregular domains with application to level-set methods. *Journal of Computational Physics*, 226(2):1432–1443, 2007.

[85] C. Min and F. Gibou. Robust second-order accurate discretizations of the multi-dimensional Heaviside and Dirac delta functions. *Journal of Computational Physics*, 227(22):9686–9695, 2008.

[86] B. Mohammadi and O. Pironneau. *Applied shape optimization for fluids*. Numerical Mathematics and Scientific Computation. Oxford University Press, Oxford, second edition, 2010.

[87] F. Murat and J. Simon. *Optimization techniques modeling and optimization in the service of man. Part II : proceedings, 7th IFIP conference Nice, september 8–12, 1975*, chapter Etude de problèmes d'optimal design, pages 54–62. Springer Berlin Heidelberg, Berlin, Heidelberg, 1976.

[88] F. Murat and J. Simon. Sur le contrôle par un domaine géométrique. Pré-publication du laboratoire d'analyse numérique, no 76015, Université Paris VI, 1976.

[89] F. Murat and L. Tartar. Calcul des variations et homogénéisation. In *Homogenization Methods : Theory and Applications in Physics (Bréau-sans-Nappe, 1983)*, volume 57 of *Collect. Dir. Études Rech. Élec. France*, pages 319–369. Eyrolles, Paris, 1985.

[90] F. Murat and L. Tartar. Optimality conditions and homogenization. In *Nonlinear Variational Problems (Isola d'Elba, 1983)*, volume 127 of *Res. Notes in Math.*, pages 1–8. Pitman, Boston, MA, 1985.

[91] G. Nguetseng. A general convergence result for a functional related to the theory of homogenization. *SIAM Journal on Mathematical Analysis*, 20(3):608–623, 1989.

[92] J. Nocedal and S.-J. Wright. *Numerical optimization*. Springer, 1999.

[93] A. A. Novotny and J. Sokołowski. *Topological derivatives in shape optimization*. Interaction of Mechanics and Mathematics. Springer, Heidelberg, 2013.

[94] A. Novruzi and M. Pierre. Structure of shape derivatives. *Journal of Evolution Equations*, 2:365–383, 2002.

[95] A. Novruzi and J.-R. Roche. Newton's method in shape optimisation : a three-dimensional case. *BIT*, 40(1):102–120, 2000.

[96] S. Osher and R.-P. Fedkiw. Level-set methods: an overview and some recent results. *Journal of Computational Physics*, 169(2):463–502, May 2001.

[97] S. Osher and J.-A. Sethian. Fronts propagating with curvature-dependent speed : algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.

[98] S.-J. Osher and F. Santosa. Level-set methods for optimization problems involving geometry and constraints I. Frequencies of a two-density inhomogeneous drum. *Journal of Computational Physics*, 171(1):272–288, July 2001.

[99] O. Pantz. Sensibilité de l'équation de la chaleur aux sauts de conductivité. *Comptes Rendus Mathématiques*, 341(5):333–337, 2005.

[100] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang. A PDE-based fast local level-set method. *Journal of Computational Physics*, 155(2):410–438, 1999.

[101] A. Potschka. *A direct method for parabolic PDE constrained optimization problems*. Advances in Numerical Mathematics. Wiesbaden : Springer Fachmedien Wiesbaden, 2014.

[102] J. Rauch. *Hyperbolic partial differential equations and geometrics optics*, volume 133 of *Graduate Studies in Mathematics*. American Mathematical Society, 2012.

[103] G.-R. Richter. An inverse problem for the steady state diffusion equation. *SIAM Journal on Applied Mathematics*, 41(2):210–221, October 1981.

[104] J.-R. Roche. Adaptive Newton-like method for shape optimization. *Control and Cybernetics*, 34(1):363–377, 2005.

[105] E. Rouy and A. Tourin. A viscosity solutions approach to shape-from-shading. *SIAM Journal on Numerical Analysis*, 29(3):867–884, 1992.

[106] G. Rozvany, editor. *Topology optimization in structural mechanics*, volume 374 of *CISM Courses and Lectures*. Springer-Verlag, Vienna, 1997.

[107] G. Russo and P. Smereka. A remark on computing distance functions. *Journal of Computational Physics*, 163(1):51–67, 2000.

[108] A. Schrijver. *Combinatorial optimization. Polyhedra and efficiency. Vol. C*, volume 24 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, 2003. Disjoint paths, hypergraphs, Chapters 70–83.

[109] J.-A. Sethian. A fast-marching level-set method for monotonically advancing fronts. In *Proceedings of the National Academy of Science*, pages 1591–1595, 1995.

[110] J.-A. Sethian. Theory, algorithms, and applications of level-set methods for propagating interfaces. *Acta Numerica*, 5:309–395, 1 1996.

[111] J.-A. Sethian. Advancing interfaces : level-set and fast-marching methods. In *Proceedings of the International Conference on Industrial and Applied Mathematics*, 1999.

[112] J.-A. Sethian. Fast-marching methods. *SIAM Review*, 41(2):199–235, June 1999.

[113] J.-A. Sethian. *Level-set methods and fast-marching methods*, volume 3 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, Cambridge, second edition, 1999. Evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science.

[114] J.-A. Sethian and J. Strain. Crystal growth and dendritic solidification. *Journal of Computational Physics*, 98(2):231–253, 1992.

[115] J.-A. Sethian and A. Wiegmann. Structural boundary design via level-set and immersed interface methods. *Journal of Computational Physics*, 163(2):489–528, September 2000.

[116] D.-F. Shanno. Conditioning of quasi-Newton methods for function minimization. *Mathematics of Computation*, 24:647–656, 1970.

[117] O. Sigmund. A 99 line topology optimization code written in Matlab. 2001.

[118] J. Simon. Differentiation with respect to the domain in boundary-value problems. *Numerical Functional Analysis and Optimization*, 2:649–687, 1980.

[119] J. Simon. Second variation for domain optimization problems. *Control and Estimation of Distributed Parameter Systems, F. Kappel, K. Kunish et W. Schappacher éd., International Series of Numerical Mathematics*, 91:361–378, 1989.

[120] P. Smereka. The numerical approximation of a delta function with application to level-set methods. *Journal of Computational Physics*, 211(1):77–90, 2006.

[121] J. Sokolowski and A. Zochowski. On the topological derivative in shape optimization. *SIAM Journal on Control and Optimization*, 37((4)):1251–1272, 1999.

[122] J. Sokolowski and J.-P. Zolésio. *Introduction to shape optimization*, volume 16 of *Springer Series in Computational Mathematics*. Springer Berlin Heidelberg, 1992.

[123] V. Šverák. On optimal shape design. *Journal de Mathématiques Pures et Appliquées (9)*, 72(6):537–551, 1993.

[124] V. Thomée. *Galerkin finite element method for parabolic problems*, volume 25 of *Springer Series in Computational Mathematics*. Springer, 1997.

[125] V. Thomée. From finite difference to finite element : a short history of numerical analysis of partial differential equations. *Journal of Computational and Applied Mathematics*, 128(1–2):1–54, 2001. Numerical Analysis 2000. Vol. VII: Partial Differential Equations.

[126] N.-P. van Dijk, M. Langelaar, and F. van Keulen. Explicit level-set-based topology optimization using an exact heaviside function and consistent sensitivity analysis. *International Journal for Numerical Methods in Engineering*, 91(1):67–97, 2012.

[127] F. van Keulen, R.-T. Haftka, and N.-H. Kim. Review of options for structural design sensitivity analysis. Part i : linear systems. *Computer Methods in Applied Mechanics and Engineering*, 194(30–33):3213–3243, 2005. Structural and Design Optimization.

[128] A. Wächter and L.-T. Biegler. On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.

[129] M.-Y. Wang, X Wang, and D. Guo. A level-set method for structural topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 192(1–2):227–246, 2003.

[130] S.-R. Wu. Lumped mass matrix in explicit finite element method for transient dynamics of elasticity. *Computer Methods in Applied Mechanics and Engineering*, 195(44–47):5983–5994, 2006.

[131] Y.-X. Yuan. A review of trust region algorithms for optimization. volume 99, pages 271–282. International Congress on Industrial and Applied Mathematics, Oxford University Press, 2000.

[132] M. Zhou and G. Rozvany. Second world congress on computational mechanics the COC algorithm. Part ii : topological, geometrical and generalized shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 89(1):309–336, 1991.

[133] O.-C. Zienkiewicz. *The finite element method in engineering science*. McGraw-Hill, London-New York-Düsseldorf, 1971.

[134] J.-P. Zolésio. The material derivative (or speed) method for shape optimization. In *Optimization of Distributed Parameter Structures, Vol. II (Iowa City, Iowa, 1980)*, volume 50 of *NATO Adv. Study Inst. Ser. E: Appl. Sci.*, pages 1089–1151. Nijhoff, The Hague, 1981.

Résumé

Le but de cette thèse est de définir une méthode d'optimisation de formes qui conjugue l'utilisation de la dérivée seconde de forme et la méthode des lignes de niveaux pour la représentation d'une forme. On considèrera d'abord deux cas plus simples : un cas d'optimisation paramétrique et un cas d'optimisation discrète.

Ce travail est divisé en quatre parties. La première contient le matériel nécessaire à la compréhension de l'ensemble de la thèse. Le premier chapitre rappelle des résultats généraux d'optimisation, et notamment le fait que les méthodes d'ordre deux ont une convergence quadratique sous certaines hypothèses. Le deuxième chapitre répertorie différentes modélisations pour l'optimisation de formes, et le troisième se concentre sur l'optimisation paramétrique puis l'optimisation géométrique. Les quatrième et cinquième chapitres introduisent respectivement la méthode des lignes de niveaux (level-set) et la méthode des éléments-finis.

La deuxième partie commence par les chapitres 6 et 7 qui détaillent des calculs de dérivée seconde dans le cas de l'optimisation paramétrique puis géométrique. Ces chapitres précisent aussi la structure et certaines propriétés de la dérivée seconde de forme. Le huitième chapitre traite du cas de l'optimisation discrète. Dans le neuvième chapitre on introduit différentes méthodes pour un calcul approché de la dérivée seconde, puis on définit un algorithme de second ordre dans un cadre général. Cela donne la possibilité de faire quelques premières simulations numériques dans le cas de l'optimisation paramétrique (Chapitre 6) et dans le cas de l'optimisation discrète (Chapitre 7).

La troisième partie est consacrée à l'optimisation géométrique. Le dixième chapitre définit une nouvelle notion de dérivée de forme qui prend en compte le fait que l'évolution des formes par la méthode des lignes de niveaux, grâce à la résolution d'une équation eikonale, se fait toujours selon la normale. Cela permet de définir aussi une méthode d'ordre deux pour l'optimisation. Le onzième chapitre détaille l'approximation d'intégrales de surface et le douzième chapitre est consacré à des exemples numériques.

La dernière partie concerne l'analyse numérique d'algorithmes d'optimisation de formes par la méthode des lignes de niveaux. Le Chapitre 13 détaille la version discrète d'un algorithme d'optimisation de formes. Le Chapitre 14 analyse les schémas numériques relatifs à la méthodes des lignes de niveaux. Enfin le dernier chapitre fait l'analyse numérique complète d'un exemple d'optimisation de formes en dimension un, avec une étude des vitesses de convergence.

Mots-clefs

optimisation par la méthode de Newton, algorithme d'optimisation de second ordre, optimisation de formes, level-set, dérivée de forme, approximation par des schémas numériques.

Abstract

The main purpose of this thesis is the definition of a shape optimization method which combines second-order differentiation with the representation of a shape by a level-set function. A second-order method is first designed for simple shape optimization problems : a thickness parametrization and a discrete optimization problem.

This work is divided in four parts. The first one is bibliographical and contains different necessary backgrounds for the rest of the work. Chapter 1 presents the classical results for general optimization and notably the quadratic rate of convergence of second-order methods in well-suited cases. Chapter 2 is a review of the different modelings for shape optimization while Chapter 3 details two particular modelings : the thickness parametrization and the geometric modeling. The level-set method is presented in Chapter 4 and Chapter 5 recalls the basics of the finite element method.

The second part opens with Chapter 6 and Chapter 7 which detail the calculation of second-order derivatives for the thickness parametrization and the geometric shape modeling. These chapters also focus on the particular structures of the second-order derivative. Then Chapter 8 is concerned with the computation of discrete derivatives for shape optimization. Finally Chapter 9 deals with different methods for approximating a second-order derivative and the definition of a second-order algorithm in a general modeling. It is also the occasion to make a few numerical experiments for the thickness (defined in Chapter 6) and the discrete (defined in Chapter 8) modelings.

Then, the third part is devoted to the geometric modeling for shape optimization. It starts with the definition of a new framework for shape differentiation in Chapter 10 and a resulting second-order method. This new framework for shape derivatives deals with normal evolutions of a shape given by an eikonal equation like in the level-set method. Chapter 11 is dedicated to the numerical computation of shape derivatives and Chapter 12 contains different numerical experiments.

Finally the last part of this work is about the numerical analysis of shape optimization algorithms based on the level-set method. Chapter 13 is concerned with a complete discretization of a shape optimization algorithm. Chapter 14 then analyses the numerical schemes for the level-set method, and the numerical error they may introduce. Finally Chapter 15 details completely a one-dimensional shape optimization example, with an error analysis on the rates of convergence.

Keywords

Newton optimization method, second-order optimization algorithm, shape optimization, level-set, shape derivative, numerical approximation schemes.