



Simulation numérique de l'usinage à l'échelle macroscopique : prise en compte d'une pièce déformable

Stéphanie Cohen Assouline

► To cite this version:

Stéphanie Cohen Assouline. Simulation numérique de l'usinage à l'échelle macroscopique : prise en compte d'une pièce déformable. Mécanique [physics.med-ph]. Arts et Métiers ParisTech, 2005. Français. NNT : 2005ENAM0028 . tel-00012006

HAL Id: tel-00012006

<https://pastel.hal.science/tel-00012006>

Submitted on 22 Mar 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Ecole Nationale Supérieure d'Arts et Métiers
Centre de Paris

THÈSE

présentée pour obtenir le titre de

DOCTEUR
de
L'ÉCOLE NATIONALE SUPÉRIEURE
D'ARTS ET MÉTIERS

Spécialité: Mécanique

par

Stéphanie COHEN ASSOULINE

**SIMULATION NUMERIQUE DE L'USINAGE A L'ECHELLE MACROSCOPIQUE :
PRISE EN COMPTE D'UNE PIECE DEFORMABLE**

Soutenue le jeudi 1^{er} décembre 2005 devant le jury composé de

MM.	P.	MARTIN	Professeur, LGIPM, ENSAM Metz	Président
	J.-Y.	HASCOËT	Professeur, IrCCyn, Ecole Centrale de Nantes	Rapporteur
	J.-F.	RIGAL	Professeur, LaMCOS, INSA de Lyon	Rapporteur
	J.-C.	LEON	Professeur, L3S, INPG	Examineur
	C.	LE CALVEZ	Chef de Département, Snecma	Invité
	P.	LORONG	Maître de Conférences, LMSP, ENSAM Paris	Codirecteur de thèse
	G.	COFFIGNAL	Professeur, LMSP, ENSAM Paris	Directeur de thèse

Remerciements

Je remercie les membres du jury qui ont accepté d'examiner ce travail et tout particulièrement les rapporteurs Messieurs Jean-Yves Hascoët et Jean-François Rigal qui m'ont apporté de précieuses remarques, ainsi que Monsieur Patrick Martin qui a accepté la présidence du jury.

Je tiens à remercier chaleureusement mes directeurs de thèse, Gérard et Philippe, qui se sont montrés toujours disponibles malgré un emploi du temps très chargé. Merci pour votre écoute, vos conseils et vos encouragements qui m'ont été précieux pour mener ce travail à son terme.

Je remercie également tous les membres du LMSP pour leur gentillesse et je souhaite un bon courage aux futurs docteurs.

Enfin, je remercie mon mari, mes parents, ma sœur, le reste de ma famille ainsi que mes amis qui m'ont soutenue jusqu'au bout.

Table des matières

Introduction	i
1 Présentation générale de l'étude	1
1.1 Rappels des objectifs visés	1
1.2 Modélisation du comportement dynamique du système POM	2
1.2.1 Principales hypothèses	2
1.2.2 Les modèles	2
1.2.3 Equation d'Equilibre Dynamique	8
1.2.4 Algorithme général de résolution	10
1.3 Exposé des différents modèles adoptés	10
1.3.1 Modèles EF	10
1.3.2 Modèles géométriques	11
1.3.3 Modèle de coupe	12
1.4 Conclusion	14
2 Définition de l'usage	15
2.1 Définition de l'enlèvement de matière	15
2.1.1 Configurations matérielle rC et déformée tC	16
2.1.2 Passage de rC à tC : application $\phi_{Pièce}(\bullet, t)$	16
2.1.3 Extension de l'application $\phi_{Pièce}(\bullet, t)$ en $\phi_{Pièce}^+(\bullet, t)$	17
2.1.4 Domaines volumiques balayé et usiné	17
2.2 Mise en œuvre avec les dexels	18
2.2.1 Exemples élémentaires de modèles dans rR	18
2.2.2 Paramètres influençant la précision	18
2.2.3 Configurations intermédiaires	20
2.3 Construction de $\phi_{Pièce}(\bullet, t)$	22
2.3.1 Maillage EF Coque	23
2.3.2 Pavage 2D	24
2.3.3 Passage à des éléments hexaédriques	24
2.4 Construction pratique de $\phi_{Pièce}^+$	28
2.5 Construction pratique de $(\phi_{Pièce}^+)^{-1}$	30
2.5.1 Première estimation de la position de référence	30
2.5.2 Correction itérative	30
2.6 Conclusion	31
3 Obtention de la surface usinée	35
3.1 Modèles géométriques	35
3.1.1 Description par discrétisation directe du domaine volumique (dixel)	35

3.1.2	Représentation des domaines volumiques par leur frontière (B-Rep) : description des surfaces	41
3.1.3	Présentation du module <i>GesDyn</i>	42
3.2	Modèle géométrique de la pièce	43
3.3	Modèles géométriques des parties coupantes de l'outil	44
3.3.1	Modèle géométrique de l'outil	44
3.3.2	Modèle géométrique du domaine volumique balayé par les faces de coupe de l'outil	46
3.4	Intersection Pièce/Outil	50
3.4.1	Démarche générale	50
3.4.2	Intersection d'axe-facette	50
3.4.3	Code matière	51
3.4.4	Première méthode d'intersection	51
3.4.5	Seconde méthode d'intersection	55
3.4.6	Progression par le voisinage	58
3.5	Calcul des efforts de coupe	60
3.5.1	Détermination du point M_i d'application d'effort	61
3.5.2	Récupération des informations nécessaires à la mise en place des lois de coupe	63
3.5.3	Détermination du repère local associé au point tM_i dans la configu- ration actuelle tC	64
3.6	Schéma de Newmark – Dynamique non linéaire	65
3.6.1	Cas particulier de l'usinage à l'échelle macroscopique	66
3.6.2	Résolution itérative	66
3.6.3	Processus de résolution sur un incrément : début	67
3.6.4	Processus de résolution sur un incrément : suite	68
3.6.5	Intégration de la première itération dans la boucle tant que	69
3.7	Conclusion	69
4	Validation et exemples	71
4.1	Choix des paramètres pour les simulations	71
4.1.1	Choix du pas de temps	72
4.1.2	Choix du nombre de dexels	73
4.1.3	Interpolation du volume balayé sur un pas de temps	73
4.2	Usinage d'une pièce rigide avec trajectoire d'outil rectiligne	74
4.3	Usinage d'une pièce rigide avec trajectoire d'outil complexe	78
4.3.1	Rappels des modèles géométriques de simulation de Grégoire Peigné	79
4.3.2	Comparaison des résultats obtenus	79
4.3.3	Validation de l'application $(\phi_{Pièce}^+)^{-1}$	81
4.4	Usinage d'une pièce rigide mobile selon une seule direction	81
4.4.1	Simulation <i>stable</i>	82
4.4.2	Simulation <i>instable</i>	84
4.5	Exemple d'un cas 3D d'une pièce déformable	87
4.6	Conclusion	89
5	Pièce ayant une rigidité et une masse évolutives	91
5.1	Prise en compte simplifiée de l'enlèvement de matière	92
5.1.1	Hypothèses de travail	92
5.1.2	La démarche	92

5.1.3	Validation de la démarche	93
5.2	Mise à jour approchée de la matrice $\hat{K}$	95
5.2.1	La démarche	96
5.2.2	Validation de la méthode	97
5.3	Conclusion	101
Conclusion et Perspectives		103
A	Pseudo Algorithmes	1
A.1	Algorithmes liés aux dexels	1
A.1.1	Détermination du dixel se trouvant juste avant une valeur z donnée	1
A.1.2	Création d'un dixel	2
A.1.3	Suppression d'un dixel	3
A.2	Algorithmes d'intersection	4
A.2.1	Première méthode d'intersection	4
A.2.2	Deuxième méthode d'intersection	5
A.3	Algorithmes de progression par le voisinage	6
A.3.1	Définition du voisinage	6
A.3.2	Mise en œuvre de la progression par le voisinage	7
A.3.3	Calcul d'une boîte englobante	9

Table des figures

1	Différentes échelles de l'étude de la coupe	ii
1.1	Fraisage	2
1.2	Modèles géométriques, outil effaceur de matière et usinage virtuel	4
1.3	Deux modèles géométriques possibles pour une pièce avec un trou	4
1.4	Représentation des voxels et des dexels	5
1.5	Représentation du modèle triple-nailboard [WZ04]	5
1.6	Décomposition de l'outil en outils élémentaires	6
1.7	Décomposition de l'outil en outils élémentaires	6
1.8	Modèle géométrique du domaine balayé par la face de coupe de l'outil au voisinage de l'arête de coupe	12
2.1	Différents domaines modélisés par des dexels (en 2D)	18
2.2	La zone à usiner est un parallélépipède rectangle	19
2.3	Exemple d'orientation des dexels en tournage	19
2.4	Passage de la configuration matérielle à la configuration relâchée	21
2.5	Passage de la configuration matérielle à la configuration déformée de la pièce à plusieurs étapes	21
2.6	Passage entre la configuration initiale et la configuration matérielle	22
2.7	Elément coque de référence à huit nœuds	23
2.8	Passage d'une configuration initiale à une configuration matérielle	24
2.9	Exemple d'un découpage du pavage en EF coque	25
2.10	Exemple d'un découpage d'un EF coque en un ensemble d'hexaèdres	26
2.11	Représentation des huit nœuds d'un EF coque et les points G	27
2.12	Modèle simplifié d'une plaque	28
2.13	Construction de $\phi_{Pi\grave{e}ce}^+$ - Maillage hexaédrique	29
2.14	Représentation de la boîte englobante en 2D	30
2.15	Configurations initiales et déformées de la pièce et de l'outil	33
3.1	Modèles Z-map simple et à plusieurs niveaux	36
3.2	Visualisation des dexels en solide rectangulaire	37
3.3	Représentation volumique et linéaire des dexels	38
3.4	Représentation d'une grille régulière	38
3.5	Les différentes entités	41
3.6	Entités décrivant le modèle géométrique des surfaces	42
3.7	Structure complémentaire des données topologiques	42
3.8	Orientation des entités et numérotation signée	43
3.9	Portion du pavage où sont disposés les dexels	44
3.10	Maillage d'une partie de la face de coupe	45

3.11	Différentes étapes de la construction du domaine balayé ${}^rV_{balayé}^i(t, t + \Delta t)$	46
3.12	Exemple avec $nbCoucheLiaison = 3$	48
3.13	Maillage entre deux contours	49
3.14	Différents codes matière d'un dixel	51
3.15	Représentation des trois modifications possibles d'un dixel après intersection	52
3.16	Représentation des différents codes matière extrémités	53
3.17	Code matière des <i>dexels fils</i> suivant la normale extérieure	54
3.18	Représentation des codes matières d'un dixel	54
3.19	Dexels non traités par la première méthode d'intersection	55
3.20	Mise à jour de la description de la pièce, les quatre cas possibles	57
3.21	Exemple de dexels <i>flottants</i> en 3D	58
3.22	Représentation des supports voisins	59
3.23	Représentation des différents dexels voisins	59
3.24	Représentation du volume balayé par une partie	61
3.25	Représentation du calcul du barycentre de chaque support	61
3.26	Représentation de l'application ϕ permettant de passer de la configuration matérielle à la configuration déformée	63
4.1	Calcul du pas de temps	72
4.2	Interpolation du volume balayé avec ou sans positions intermédiaires	74
4.3	Configuration d'un usinage rigide	75
4.4	Etat de surface théorique et avec $N_{tour} = 100$ et 1000	76
4.5	Etat de surface théorique et avec $N_{tour} = 1000, 2000, 3000$	76
4.6	Etat de surface avec $N_{tour} = 2000$ et $10\,000$ dexels, $1\,000$ dexels et 500 dexels	77
4.7	Représentation des composantes de l'effort de coupe avec un nombre de dexels = 1 ou 4	77
4.8	Evolution de la position horizontale du centre de la fraise en fonction de sa position verticale	78
4.9	Comparaison des positions des points de la surface avec les deux simulateurs	80
4.10	Comparaison des composantes de l'effort de coupe avec les deux simulateurs	80
4.11	Configuration d'usinage d'une pièce "flexible"	82
4.12	Déplacement de la pièce suivant l'axe des Z entre $[0, 15]$ et comparaisons sur deux zooms entre $[0, 3]$ et $[5, 6]$ avec les deux simulateurs : cas d'une simulation stable	83
4.13	Comparaison des positions des points de la surface avec les deux simulateurs : cas d'une simulation stable	83
4.14	Comparaison des composantes de l'effort de coupe avec les deux simulateurs : cas d'une simulation stable	84
4.15	Comparaison des positions des points de la surface avec les deux simulateurs : cas d'une simulation instable	84
4.16	Déplacement de la pièce suivant l'axe des Z entre $[0, 15]$ et comparaisons sur deux zooms entre $[0, 4]$ et $[4, 6]$ avec les deux simulateurs : cas d'une simulation instable	85
4.17	Représentation des positions des points de la surface avec deux pas de temps différent $N_{tour} = 1000$ et $N_{tour} = 1050$: cas d'une simulation instable	86
4.18	Représentation des positions des points de la surface avec deux pas de temps différent $N_{tour} = 1000$ et $N_{tour} = 3000$: cas d'une simulation instable	86
4.19	Définition du domaine usiné de la pièce	87

4.20	Exemple de l'état de surface simulée pour une pièce voile mince	88
5.1	Les dimensions de la pièce	93
5.2	La pièce en hexaèdres avec des éléments supprimés	94
5.3	Maillage complet de la pièce et des deux modèles	95
5.4	Evolution du déplacement au point P	100

Liste des tableaux

3.1	Passage entre le <i>dexel père</i> et les deux <i>dexels fils</i>	52
3.2	Code suivant la normale extérieure de la facette	54
3.3	Mise à jour de la description de la pièce, les quatre cas possibles	56
3.4	Discretisation, les quatre cas possibles	62
4.1	Relation entre N_{tour} et Δt	75
4.2	Caractéristiques modales	81
4.3	Caractéristiques dynamiques	81
5.1	Fréquences des trois premiers modes propres	94
5.2	Fréquences des cinq premiers modes propres	95
5.3	Erreur relative maximale (en %)	100
5.4	Nombre de factorisations et de descentes-montées par incrément et par étape	100
5.5	Exemple pour 100 étapes de 100 incréments	101
5.6	Gain estimé du coût pour 10000 incréments, par rapport à la <i>simulation 1</i>	101

Algorithmes en pseudo-code

1.1	Algorithme général de résolution	10
5.1	Réactualisation à chaque pas de temps Δt	98
5.2	Réactualisation avec l' <i>approche linéaire</i>	99
A.1	Algorithme permettant de déterminer le dixel se trouvant juste avant une valeur z donnée	1
A.2	Création d'un dixel	2
A.3	Elimination d'un dixel	3
A.4	Algorithme de la première méthode d'intersection	4
A.5	Algorithme de la deuxième méthode d'intersection	5
A.6	Détermination de la liste des dexels voisins d'un dixel	6
A.7	Progression par le voisinage : Initialisation	7
A.8	Progression par le voisinage	8
A.9	Progression par le voisinage : Conclusion	9
A.10	Création d'une boîte englobante	10

Introduction

A ce jour, l'enlèvement de matière par outil coupant occupe toujours une place prédominante parmi les procédés de mise en forme des matériaux. L'accroissement des performances des machines-outils, des outils de coupe et des logiciels de CFAO a œuvré à garder ce procédé compétitif grâce aux gains de productivité et de qualité. D'innombrables travaux ont été menés dans ce sens et les remarquables avancées dans les matériaux, les revêtements des outils coupants et les lubrifiants ont permis d'accroître considérablement les vitesses de coupe tout en améliorant la qualité de la surface usinée.

Cependant, l'utilisation rationnelle de cette technologie est encore fortement pénalisée par les lacunes dans la connaissance de la coupe, que ce soit au niveau mésoscopique où l'interaction fine entre l'outil et la pièce est étudiée (thermomécanique de la coupe, formation du copeau), aussi bien qu'au niveau macroscopique où l'étude porte plus sur la stabilité du procédé. Si bien que la coupe garde encore une grande part de mystère et exige de l'utilisateur un bon niveau de savoir faire et parfois d'empirisme pour exploiter au mieux les capacités des moyens de production.

Parmi les principales difficultés rencontrées, celles posées par les phénomènes vibratoires générés par la coupe occupent une place importante. Nous pouvons citer en premier le phénomène bien connu de broutement, dont les conditions d'apparition sont difficilement prévisibles et qui dégrade fortement la qualité de la surface usinée et accélère l'usure de certains éléments sensibles des moyens de production comme les outils de coupe ou les broches. Mais la nature discontinue et périodique de la coupe en fraisage est aussi la cause de vibrations systématiques de l'ensemble Pièce-Outil-Machine (POM). Ces dernières accroissent principalement le défaut de forme de la surface usinée, et ce d'autant plus que la fréquence d'excitation de la coupe (la fréquence de passage des dents de l'outil coupant) est proche d'une fréquence propre du système POM.

Pour réaliser ce travail nous avons pu nous appuyer sur les nombreuses études réalisées dans le domaine de la coupe dynamique. Ainsi, le choix de la simulation numérique comme outil d'étude a été motivé par des résultats prometteurs que cette technique émergente a montré pour étudier de tels problèmes. Le simulateur repose sur une modélisation, à présent classique, de la coupe à l'échelle macroscopique. Elle s'appuie sur une modélisation simplifiée de l'enlèvement de matière et des efforts de coupe, établie uniquement à partir de considérations géométriques, les phénomènes thermomécaniques n'étant pas pris en compte.

Approche multi-échelles

L'étude de l'usinage est souvent abordée à l'aide d'une approche multi-échelles, ceci permet de séparer les difficultés en limitant le nombre de phénomènes à prendre en compte

et la taille du modèle à une échelle donnée. Trois échelles d'analyse peuvent être distinguées : échelle *microscopique*, *mésoscopique* et *macroscopique* (figure 1). Sans entrer dans les détails, l'échelle *microscopique* est celui de l'étude des matériaux. Il s'agit de déduire certaines propriétés à partir de la structure fine du matériau. Ces propriétés sont, par exemple, les lois de comportement du matériau utilisées à l'échelle supérieure. L'échelle *mésoscopique* est en usinage celle de la description de la formation du copeau. Elle est basée sur une description thermomécanique faisant intervenir les différents phénomènes physiques et métallurgiques, mais à une échelle qui est celle de la mécanique des milieux continus. L'échelle *macroscopique* s'intéresse plus particulièrement au comportement global Pièce-Outil-Machine (POM). C'est dans les travaux à l'échelle macroscopique que s'inscrit ce travail de thèse qui vient compléter les investigations déjà réalisées au laboratoire LMSP par K. Dekelbab [Dek95], E. Beauchesne [Bea99] et A. Marty [Mar03]. Les travaux expérimentaux de François Lapujoulade [Lap03] fournissent par ailleurs un moyen de mettre en place des modèles de coupe adaptés pour des simulations à cette échelle, puis d'identifier les coefficients.

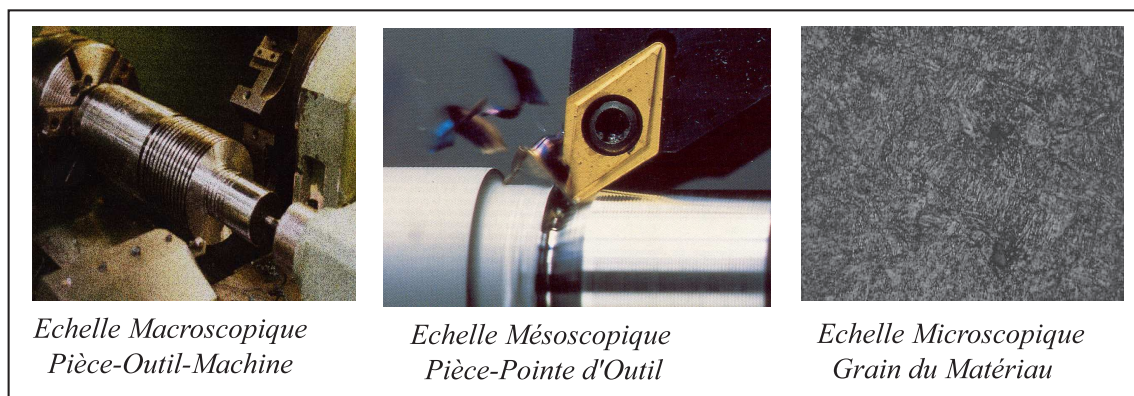


FIG. 1 – Différentes échelles de l'étude de la coupe

Approche périodique/ temporelle

Principalement deux approches visent à prévoir la stabilité d'usinage à l'échelle macroscopique. La première est une approche analytique et peut être qualifiée d'approche "périodique". Le principal objectif de cette approche est d'étudier la stabilité de l'usinage en supposant que le processus d'usinage est périodique. Un modèle linéarisé est construit pour permettre l'étude de la stabilité d'un cycle (une période, 1 cycle = 1 tour de broche) et sa stabilité théorique est étudiée [IS97, AES98, VDK01]. L'approche "périodique" mène à l'étude des lobes de stabilité qui donnent les zones où l'usinage est stable en fonction de la vitesse de rotation et de la profondeur de la coupe. Plusieurs logiciels sont utilisés dans l'industrie [EY01a, EY01b] avec cette approche. Les limites de cette approche se produisent souvent lors des exécutions de finissage où même les petites variations induisent une variation élevée de la force de coupe. Dans ce cas-ci, le modèle linéarisé n'est plus réaliste. D'ailleurs, il y a parfois des conditions stables, qui ne sont pas prévues parce que les non linéarités peuvent produire de l'instabilité aussi bien que de la stabilité.

La deuxième approche, identifiée comme une approche "temporelle", n'est pas fondée sur l'hypothèse de la périodicité. Elle est basée sur une étude temporelle (processus de

résolution *incrémental*), qui a un coût numérique plus élevé. L’histoire complète de l’évolution des faces de coupe de l’outil ainsi que de la surface de la pièce est simulée, pour n’importe quel type d’usinage [MA91, ABC⁺02]. Les critères de stabilité ne sont pas liés de façon analytique à cette approche, mais à l’évaluation de la qualité de la surface produite ou au niveau des vibrations de certaines parties du système. Une des principales difficultés de l’approche ”temporelle” est que la plus grande partie du coût numérique est liée à la prévision de l’évolution de la surface. Notre travail est basé sur une approche ”temporelle”.

Pièces déformables

L’usinage des parois minces se rencontre dans de nombreux domaines industriels : le secteur aéronautique et spatial, l’industrie des moules, les turbomachines, les capteurs, etc. Il existe deux grandes familles de pièces qui présentent des parois minces usinées. La première est celle des pièces réalisées dans la masse et qui se présentent à la fin comme une succession d’alvéoles. L’exemple le plus typique est celui des panneaux de voilure. La deuxième catégorie est celle des ailettes et des aubes qui entrent dans la composition des turbomachines. Il s’agit souvent de pièces très fines, fortement courbées et encastrées sur un seul côté. Dans tous les cas, ces parois sont déformables si on se situe loin des zones de maintien. L’action d’une fraise est susceptible d’entraîner de fortes vibrations. Le développement récent du fraisage à grande vitesse permet d’aller plus loin dans l’obtention de voiles très minces à des conditions économiques acceptables. S’il permet d’atteindre des points de fonctionnement stables très intéressants, il est aussi susceptible d’engendrer de très fortes instabilités qui vont jusqu’à la destruction de l’outil et de la pièce. Il est crucial de disposer de moyens qui autorisent la prédiction fiable des comportements vibratoires, d’autant plus que la mise au point sur machine est dangereuse et très coûteuse.

Le principal apport du travail présenté ici est la prise en compte de pièces dont la zone concernée par l’usinage est flexible. Cette présence de déformations dans la pièce a nécessité une définition plus précise de l’enlèvement de matière. Cette définition se fonde sur une configuration très particulière, une configuration dite matérielle, où la réalisation pratique de l’usinage est effectuée.

Par ailleurs, les pièces flexibles étant généralement très minces, l’enlèvement de matière peut modifier de façon significative le comportement dynamique de la pièce (perte de masse et de raideur). Pour être fidèle, une simulation doit alors prendre en compte cette modification de façon régulière en cours d’usinage. Nous montrons que cette prise en compte ne peut dans certains cas être évitée, et nous proposons une stratégie limitant les coûts numériques induits par cette prise en compte.

Un autre apport du travail réalisé est l’utilisation d’un modèle géométrique différent pour représenter le volume occupé par la pièce. Avec le modèle géométrique de la pièce retenu dans les thèses précédentes (modèles B-Rep analogue aux modèles utilisés en CAO), les très nombreuses intersections de polyèdres conduisent à des facettes ”dégénérées” dont la prise en compte dans les algorithmes d’intersection est loin d’être triviale et constitue actuellement un des points bloquants pour ce qui concerne la robustesse.

La nouvelle technique adoptée est celle des Z-buffer [Hoo86] où la pièce est modélisée par des dexels (ensemble de « bâtonnets » disposés sur une matrice régulière). Cette description permet de simplifier sensiblement les algorithmes d’intersection tout en les

rendant plus robustes. Cette nouvelle description nous a conduit à revoir l'interaction outil/pièce et notamment la démarche de calcul des efforts de coupe.

Pour présenter nos travaux nous avons décomposé ce document en cinq chapitres. Le chapitre 1 est une présentation générale de l'étude menée et de ses objectifs. Y sont rappelées les grandes lignes de l'approche temporelle utilisée ainsi que les hypothèses faites et les modèles adoptés. Dans le chapitre 2, nous revenons sur la notion d'enlèvement de matière en donnant une définition précise apte à décrire cet enlèvement lorsque la pièce se déforme dans la zone usinée. Sont en particulier abordés, dans cette partie, la définition de la configuration matérielle et des aspects pratiques liés à l'obtention de la surface usinée. Le chapitre 3 concerne la mise en place des dexels. Les différentes répercussions de cette mise en place y sont détaillées, notamment le calcul des efforts de coupe qui se base désormais sur la détermination d'une section moyenne de coupe. La bonne robustesse des algorithmes géométriques obtenus nous ayant enfin permis de réaliser des simulations complètes d'usinage nous consacrons le chapitre 4 à la validation de l'approche au travers d'exemples. Nous proposons en particulier une comparaison avec le simulateur développé par Grégoire Peigné durant sa thèse [Pei04] sur des cas relativement simples. Cette comparaison permet de valider le bon fonctionnement des algorithmes géométriques et dynamiques développés. Un exemple d'usinage de pièce flexible est donné en fin de chapitre. Le chapitre 5 porte sur la prise en compte de l'évolution progressive de la rigidité et de la masse de la pièce au cours de l'usinage. Ce chapitre est indépendant des chapitres précédents. Nous terminons par les conclusions et perspectives de ce travail.

Chapitre 1

Présentation générale de l'étude

Le LMSP (Laboratoire de Mécanique des Systèmes et des Procédés) a entrepris depuis plusieurs années un travail de recherche avec l'objectif de simuler numériquement l'usinage. C'est dans les travaux à l'échelle macroscopique que s'inscrit ce travail qui vient compléter les investigations déjà réalisées au LMSP par K. Dekelbab [Dek95], E. Beauchesne [Bea99] et A. Marty [Mar03]. A la suite de ces trois thèses la démonstration de la faisabilité de l'approche proposée a été validée : l'utilisation de deux modèles géométriques un pour l'outil, un pour la pièce, associés à un modèle éléments finis pour l'ensemble Pièce-Outil-Machine (POM) permettent une simulation de l'interaction outil-pièce. Ceci est réalisé en prenant en compte tous les aspects vibratoires du mouvement relatif via le concept d'outil effaceur de matière et du modèle d'efforts de coupe qui lui est associé. La démarche complète étant maîtrisée, il reste cependant de nombreux points à améliorer pour obtenir des algorithmes robustes et efficaces.

Ce chapitre présente le contexte de ce travail de thèse en commençant par donner les objectifs visés, puis les différents modèles nécessaires à la modélisation du comportement dynamique du système POM et enfin les différents modèles adoptés dans cette thèse pour résoudre les problèmes qui ont pu être rencontrés lors des différentes thèses sur le sujet.

1.1 Rappels des objectifs visés

L'objectif de ce travail est de réaliser un logiciel prédictif de simulation de l'usinage à l'échelle macroscopique avec une approche temporelle. Cet outil utilisera comme données des paramètres venant de la gamme d'usinage (géométrie de l'outil, matière usinée, opération, trajectoire des outils et des pièces par exemple) et issues de l'expérimentation (entre autres : loi de coupe, flexibilité et inertie de différents éléments non modélisables a priori).

Les objectifs de ce logiciel sont de permettre :

- la prédiction de l'état de surface (détails fins de la géométrie de la pièce, circularité des pièces, rugosité...),
- la prédiction de phénomènes d'instabilité vibratoire tel que le broutage,
- l'évaluation des différentes grandeurs liées à la coupe (efforts, déplacements, vitesses relatives).

On peut remarquer que les deux derniers objectifs peuvent être également atteints grâce à l'approche mésoscopique mais sur des portions très restreintes de la pièce et lorsque celle-ci sera opérationnelle.

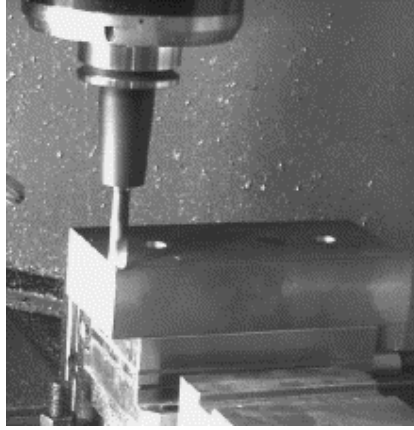


FIG. 1.1 – Fraisage

Pour atteindre ces objectifs, il est nécessaire de modéliser le comportement dynamique du système Pièce-Outil-Machine (POM).

1.2 Modélisation du comportement dynamique du système POM

1.2.1 Principales hypothèses

Le cadre de la modélisation est celui de petites déformations et déplacements venant se superposer à de grands déplacements connus (mouvement d'avance, de rotation de la broche ou de la pièce). Aussi le problème peut se ramener à un problème mécanique en petites perturbations au voisinage d'un point de fonctionnement.

A l'échelle macroscopique, l'outil est considéré comme un effaceur de matière. La matière est simplement effacée par le passage des faces de coupe de l'outil (il existe plusieurs faces de coupe dans le cas du fraisage : au moins une par dent). Nous ne tenons donc pas compte à cette échelle de l'évolution du copeau, ni de son interaction avec la pièce de façon fine.

Le système dynamique est constitué de l'ensemble Pièce-Outil-Machine (POM). Le système POM est étudié, chaque partie du système pouvant être considérée comme une partie déformable ou non. Le système POM a été largement décrit dans des travaux antérieurs [Dek95, Bea99] aussi nous ne rappellerons que les éléments les plus importants.

1.2.2 Les modèles

Pour étudier le comportement vibratoire, nous devons connaître les caractéristiques dynamiques du système POM et la relation permettant de déterminer la force de coupe à partir des conditions d'interaction outil - pièce. Afin de simuler les phénomènes présents, il est nécessaire de modéliser le système POM et certains des phénomènes physiques qui s'y produisent. Nous introduisons pour cela, principalement, quatre modèles :

- un modèle permettant le calcul de la force de coupe pour représenter l'interaction outil - pièce,
- un modèle géométrique décrivant la forme de l'outil,

1.2. MODÉLISATION DU COMPORTEMENT DYNAMIQUE DU SYSTÈME POM 3

- un modèle géométrique décrivant la surface de la pièce, cette dernière évoluant au cours de l'usinage,
- un modèle mécanique de la structure POM, pour simuler son comportement dynamique (vibration).

Modèle dynamique de la structure POM

Le modèle mécanique du système Pièce-Outil-Machine utilisé peut être un modèle simple masse-ressort ou un modèle complexe élément fini, aussi longtemps qu'il donne une bonne description de la réalité dynamique. La modélisation par éléments finis permet une discrétisation spatiale beaucoup plus fine et plus souple. Elle permet d'obtenir des modes de vibrations plus réalistes et d'aborder le cas où la pièce et/ou l'outil sont déformables dans la zone de travail. Les déformations de l'outil et de la pièce sont calculées en utilisant des modèles cinématiques construits par éléments finis (EF). Le reste de la cinématique du système POM peut aussi être discrétisé par éléments finis si cela est nécessaire.

Le modèle représentant le comportement dynamique du système POM est généralement défini par les trois matrices de masse $\underline{M}$, de raideur $\underline{K}$ et d'amortissement $\underline{C}$. Ces matrices sont constantes dans les cas simples, mais deviennent variables si l'enlèvement de matière et/ou le déplacement des chariots agit de façon significative sur la répartition des masses et des raideurs. Ce dernier cas est traité dans le chapitre 5 [ABC⁺02].

Modèles géométriques

Les modèles géométriques doivent permettre de situer à chaque instant t la position relative du volume occupé par la pièce et des volumes balayés, durant l'incrément de temps Δt , par les outils élémentaires. En cas d'interférence entre le volume balayé par l'outil et la pièce, l'intersection des deux volumes permet de déduire le pseudo-copeau qui doit être enlevé de la pièce. Cette opération est illustrée très schématiquement sur la figure 1.2. Cette intersection permet aussi d'isoler la partie ${}^{t+\Delta t}\partial\Omega_{coupe}$ de la surface de coupe en contact avec la pièce et d'en déduire, l'épaisseur de coupe instantanée à chaque itération effectuée durant l'incrément.

Les modèles géométriques utilisés dans ce domaine peuvent aller du plus simple, une série de points [Gu94, MA91], au plus complexe, description facettisée de la surface ou représentation de type Z-buffer ou dixel [MHK01, KCC00].

Modèles géométriques de la pièce

La géométrie de la pièce doit être approximée par un modèle géométrique évolutif et suffisamment fin pour décrire l'évolution de la frontière du domaine pièce $\partial\Omega_{pièce}$, à mesure que l'enlèvement de matière (l'effacement ici) se produit, depuis la forme initiale de la pièce jusqu'à sa forme finale. Trois grandes familles de représentations géométriques peuvent être dégagées :

- D'une part, le contour du volume (la frontière $\partial\Omega$ du domaine volumique Ω) peut être représenté par une liste de points en projection sur un plan.
- Les domaines volumiques peuvent avoir pour représentation leur frontière (surfaccique) : on parle alors de modèle B-Rep (paragraphe 3.2 [Mar03]).
- Enfin, le modèle géométrique de la pièce peut aussi être un modèle volumique, il est possible d'utiliser entre autres trois modèles : des voxels [RLHB04], des dexels

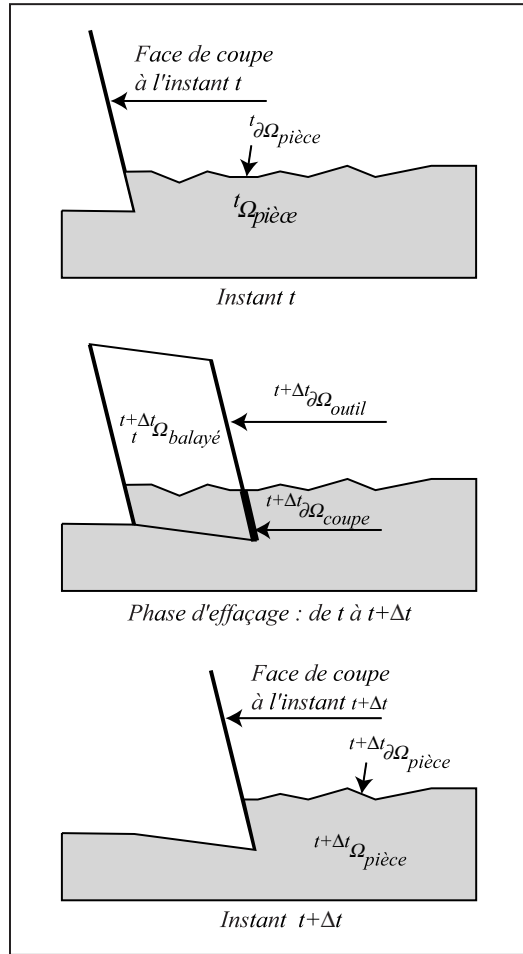


FIG. 1.2 – Modèles géométriques, outil effaceur de matière et usinage virtuel

(Z-buffer) [Hoo86] ou des triple-nailboard ("planches à clous dans les trois directions orthogonales") [WZ04].

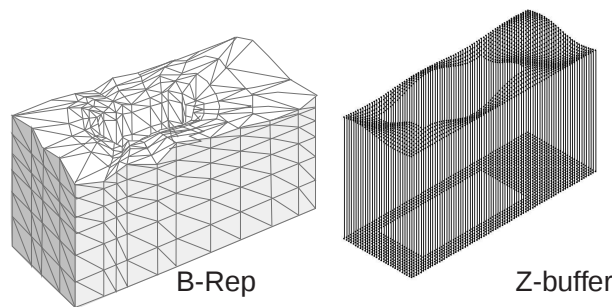


FIG. 1.3 – Deux modèles géométriques possibles pour une pièce avec un trou

Les voxels permettent de représenter un objet avec de nombreux petits cubes sur une grille régulière 3D (figure 1.4 à gauche). Les dexels sont représentés par de longues colonnes compactes sur une grille régulière (figure 1.4 à droite). L'utilisation du modèle volumique basé sur les dexels est plus avantageuse que les voxels car les dexels demandent moins d'espace mémoire et moins d'éléments pour le traitement du volume.

La pièce peut aussi être modélisée par un modèle de triple-nailboard, chaque nailboard ("planche à clous") est défini à partir d'un certain nombre de nails ("clous") virtuels

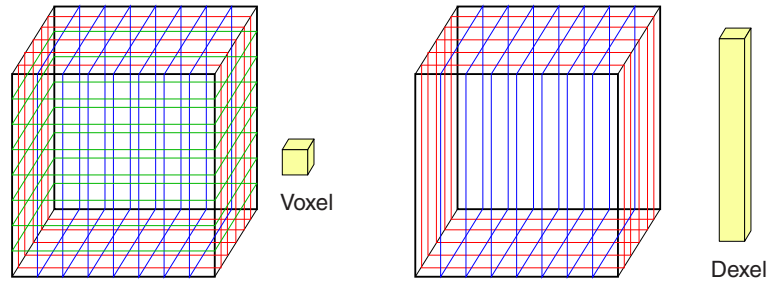


FIG. 1.4 – Représentation des voxels et des dexels

placés côte à côte et qui sont utilisés comme approximation de n'importe quelle géométrie de pièce. Le nombre de "clous" parallèles à chaque axe principal peut être choisi par l'utilisateur suivant les dimensions de la pièce et pour réaliser précisément la simulation désirée. Plus le nombre de "clous" introduits est important et plus la pièce est bien approximée. Mais cela entraîne un ralentissement de la vitesse de la simulation qui est proportionnel à la quantité de "clous".

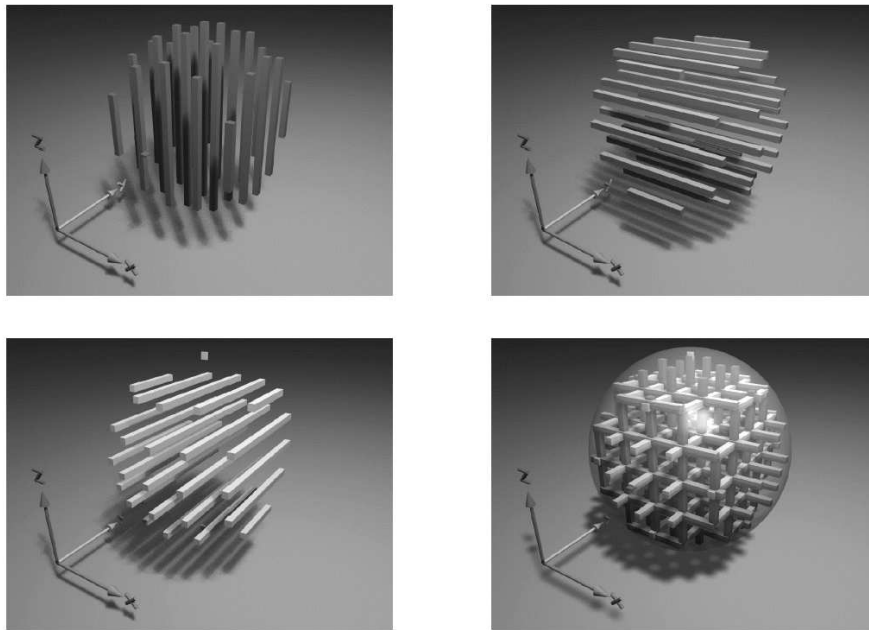


FIG. 1.5 – Représentation du modèle triple-nailboard [WZ04]

Pour modéliser des pièces tridimensionnelles, il est nécessaire d'assembler trois "plan-ches à clous" simples, chacune parallèle à l'un des axes orthogonaux (l'axe des X , des Y et des Z), ensemble dans un même modèle (triple-nailboard) comme le montre la figure 1.5.

La modélisation de simple nailboard, dans une seule direction principale, est équivalente à celle des dexels.

Modèles géométriques de l'outil

La modélisation géométrique de l'outil permet de générer la surface usinée et le calcul des grandeurs géométriques utilisées dans la loi de coupe. La géométrie de l'outil est

généralement constante tout au long de l'usinage si on ne fait pas intervenir l'usure de l'outil.

Plusieurs axes de recherches ont été creusés pour améliorer la modélisation, à commencer par la prise en compte de géométries d'outil plus complexes et réalistes. Cette problématique est complexe car la modélisation requiert d'une part un modèle de loi d'effort de coupe et d'autre part une description géométrique de l'outil compatible avec le principe de la simulation. La réponse a été trouvée dans les modélisations développées pour le calcul des efforts statiques de coupe. Ainsi, les simulations dédiées aux opérations de fraisage de profil se sont inspirées du modèle d'effort de coupe proposé par Kline et DeVor [DKZ80]. Dans ces travaux, les auteurs traitent les géométries d'outil complexes comme une somme d'outils élémentaires. Ainsi une fraise cylindrique avec un angle d'hélice se décompose suivant son axe comme un empilement d'outils élémentaires de hauteur élémentaire dz . Chaque arête de coupe est partagée en arête élémentaire de faible hauteur, sur laquelle il est possible de faire l'approximation d'un angle d'hélice nul. Ces arêtes sont paramétrées par un angle et un rayon. Le découpage de l'outil est illustré sur la figure 1.7.

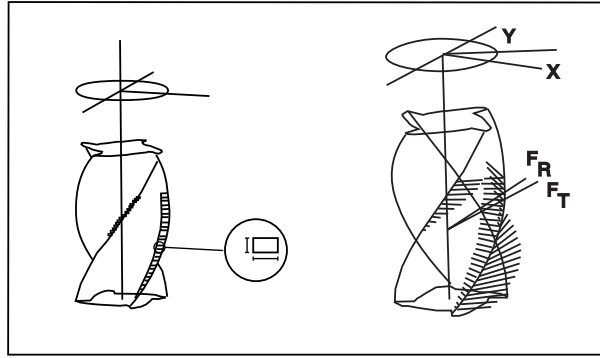


FIG. 1.6 – Décomposition de l'outil en outils élémentaires

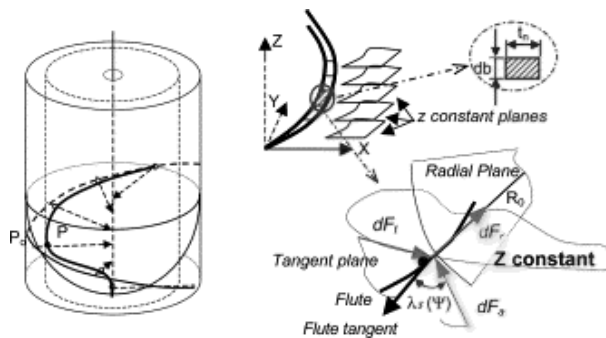


FIG. 1.7 – Décomposition de l'outil en outils élémentaires

Modèle permettant le calcul de la force de coupe

Toutes les études, que ce soit une recherche des limites de stabilité, ou une simulation des procédés de coupe, nécessitent un modèle permettant de calculer l'effort de coupe.

1.2. MODÉLISATION DU COMPORTEMENT DYNAMIQUE DU SYSTÈME POM 7

L'établissement de tels modèles étant, à lui seul, tout un domaine de recherche, nous ne détaillerons que les modèles qui pourront être utiles dans les simulations des procédés de coupe que nous envisageons dans ce travail.

Une des principales difficultés de la modélisation du fraisage réside dans la définition des lois de coupe car la construction du modèle d'effort repose sur la compréhension du mécanisme de formation du copeau qui est ici délicate à appréhender du fait de la géométrie complexe des fraises et de la cinématique de leur trajectoire. Enfin, l'identification des paramètres des lois de coupe reste un problème majeur en fraisage car les chocs causés par la coupe discontinue requièrent l'utilisation de moyens de mesures à la dynamique particulièrement maîtrisée afin de ne pas fausser les mesures.

Pour ces raisons, la majorité des études sur la dynamique de la coupe utilise une méthode de modélisation mécanique développée pour l'étude statique du fraisage et qui voient le processus d'usinage comme une combinaison des ingrédients suivants : la relation entre épaisseur de copeau et effort de coupe comme le propose Koengsberger et al. [KS61], Sabberwal [Sab62], Marttelloti [Mar41, Mar45], la géométrie des outils et la géométrie du processus de coupe.

Dans le contexte de la simulation numérique à l'échelle macroscopique, les interactions fines entre la pièce et l'outil au niveau de l'arête de coupe sont exprimées par des lois de coupe simplifiées.

Une des toutes premières simulations numériques de la coupe en modélisant le phénomène de régénération de la surface fut réalisée par Thusty et Ismail en 1981 [TI81].

Simulation de Smith et Thusty :

Les efforts sont vus comme la somme d'une force tangentielle et d'une force radiale exercées par la pièce sur l'outil. Ces efforts de coupe sont exprimés linéairement en fonction de la section de copeau, (produit de la hauteur de la passe b et de l'épaisseur h) et des coefficients spécifiques de coupe K_s et K_r :

$$F_T = K_s b h \quad (1.1)$$

$$F_r = K_r F_T \quad (1.2)$$

La difficulté du calcul des efforts de coupe ainsi modélisés réside dans la détermination de l'épaisseur du copeau (non déformé) h , qui est fonction non seulement de la position de l'outil par rapport à la pièce, mais aussi du profil de la surface usinée.

Simulation de Kline et DeVor :

Le point intéressant de la modélisation retenue par Kline [DKZ80] consiste à découper l'outil en tranches suivant l'axe de rotation (figure 1.7 à gauche). Ce découpage permet d'affiner la modélisation de la répartition de l'effort de coupe sur l'outil, en définissant pour chaque arête de coupe se trouvant dans chaque tranche, la partie correspondante de l'effort de coupe (figure 1.7 à droite).

L'effort de coupe total $\vec{F}_c$ exercé sur l'outil est la somme des efforts de coupe élémentaires $\vec{F}_{c_{i,k}}$ exercés sur chaque arête élémentaire en prise avec la matière. Les indices i et k réfèrent respectivement à l'arête et au plan :

$$\vec{F}_c = \sum_i \sum_k \vec{F}_{c_{i,k}} \quad (1.3)$$

Les efforts élémentaires sont vus comme la somme d'une composante radiale $F_{r,i,k}$ et d'une composante tangentielle $F_{t,i,k}$ appliquées sur l'arête. La composante axiale est ici négligée car elle n'a aucune utilité. Ces composantes sont exprimées en fonction de l'épaisseur de copeau élémentaire $h_{i,k}$ et de coefficients spécifiques K_t et K_r :

$$F_{t,i,k} = K_t dz h_{i,k} \quad (1.4)$$

$$F_{r,i,k} = K_r F_{t,i,k} \quad (1.5)$$

où K_t est une fonction puissance de l'avance par dent f :

$$K_t = f^p \quad (1.6)$$

L'épaisseur du copeau élémentaire $h_{i,k}$ est approchée, en supposant l'outil parfait (décalage angulaire constant d'une arête à l'autre et rayon identique sur un même plan) par :

$$h_{i,k} = f \sin \beta_{i,k} \quad (1.7)$$

où $\beta_{i,k}$ désigne la position angulaire de la dent par rapport à la direction d'avance. Enfin, l'engagement de la fraise dans la pièce est défini par un secteur angulaire constant sur lequel se fait l'intégration des efforts de coupe.

Simulation de Takata :

Le même principe du calcul d'effort de coupe a été utilisé par Takata [TTIS89]. Toutefois, Takata ne calcule l'effort qu'en cas de détection du contact outil-pièce grâce à un modèle géométrique basé sur le principe des Z-buffers.

Bien que tous ces auteurs aient développé ce modèle pour le calcul d'effort dans le cas d'un système pièce-outil rigide, la modélisation s'est avérée être applicable à l'étude dynamique de la coupe. Elle offre en effet une relative simplicité de description géométrique en 2D et une modélisation des efforts élémentaires directement transposable à la coupe dynamique.

Ces différents modèles, géométriques, dynamiques ainsi que les efforts de coupe, permettent d'aboutir à l'équation d'équilibre dynamique (EED) du système POM que nous allons à présent rappeler.

1.2.3 Equation d'Equilibre Dynamique

Lorsque l'on s'intéresse à une portion de l'opération d'usinage, pour laquelle il est possible de considérer que les caractéristiques de rigidité, de masse et d'amortissement sont constantes, l'équation (1.8) est l'équation différentielle matricielle du second ordre en temps qu'il faut résoudre.

$$\underline{\underline{M}} \ddot{\underline{q}} + \underline{\underline{C}}(\omega) \dot{\underline{q}} + \underline{\underline{K}} \underline{q} = \underline{Q}_{coupe}(\underline{q}, \dot{\underline{q}}, \partial \Omega_{coupe}) \quad (1.8)$$

Les matrices $\underline{\underline{M}}$ et $\underline{\underline{K}}$ sont respectivement les matrices de masse et de rigidité. La matrice $\underline{\underline{C}}$ prend en compte l'amortissement et les éventuels effets de Coriolis dus à une vitesse de rotation ω .

1.2. MODÉLISATION DU COMPORTEMENT DYNAMIQUE DU SYSTÈME POM 9

Le second membre $\underline{Q}_{coupe}$, provenant des efforts de coupe, est une fonction non linéaire qui prend en compte le caractère irréversible de l'enlèvement de matière ainsi que son histoire : $\partial\Omega_{coupe}$ dépend de la matière rencontrée par l'ensemble des faces de coupe à l'instant t . Or le domaine occupé par la matière dépend justement des passages précédents des faces de coupe, et par conséquent de la disposition de la matière qui reste.

Les composantes de la matrice colonne $\underline{q}$ sont les inconnues du problème. Elles comprennent les déplacements généralisés associés aux nœuds de la pièce, de l'outil et de la machine (degrés de liberté ou DDL).

La résolution pratique de l'équation non-linéaire (1.8) se fait grâce à un schéma d'intégration numérique qui s'appuie sur une discrétisation en temps de $\underline{q}$, par exemple le schéma de Newmark (paragraphe 3.6) [GR96, Bat82, Bat96]. Cette discrétisation en temps apporte, de la même façon que la discrétisation en espace liée à l'approche par éléments finis, une approximation des mouvements.

1.2.4 Algorithme général de résolution

AlgoGeneral

début

Initialisation :

Construction des matrices $\underline{\underline{M}}$, $\underline{\underline{C}}$ et $\underline{\underline{K}}$ du système POM

Construction du modèle géométrique de la pièce et de la topologie du modèle géométrique du volume balayé

Pour (chaque incrément de temps $t := t + \Delta t$)

Initialisation du vecteur généralisé des DDL du système POM :

${}^{t+\Delta t}\underline{q}_{POM} := {}^t\underline{q}_{POM}$

$\text{équilibre} := \text{faux}$

Tant que ($\text{équilibre} \neq \text{vrai}$)

Restitution du modèle géométrique obtenu à t après équilibre

Détermination de la frontière du volume balayé par l'outil pendant Δt

Détermination de la frontière du volume balayé dans la configuration matérielle

Si (Un risque d'intersection entre la pièce et la frontière du volume balayé existe) **alors**

Calcul des intersections

Calcul des efforts de coupe

fin Si

Résolution de l'EED (Newmark + Newton Raphson)

Correction du vecteur ${}^{t+\Delta t}\underline{q}_{POM}$ en fonction de ce déséquilibre

fin Tant que

Réactualisation du modèle géométrique de la pièce

Validation du nouveau modèle géométrique de la pièce à l'instant $t + \Delta t$

fin Pour

fin

Alg. 1.1: Algorithme général de résolution

1.3 Exposé des différents modèles adoptés

Après avoir exposé les différents modèles possibles, nous allons maintenant présenter les modèles adoptés dans notre étude.

1.3.1 Modèles EF

Modèle EF de l'outil

Le modèle EF de l'outil utilise plutôt des éléments de type poutre pour modéliser le corps de l'outil en tournage ou l'arbre de la broche en fraisage, y compris la fraise

lorsqu'elle est élançée et susceptible de se déformer durant l'usinage (par exemple une fraise hélicoïdale) [Dek95]. Des parties indéformables mais mobiles sont rajoutées pour simuler les parties massives telles que les plaquettes de coupe rapportées et la partie les supportant.

Modèles EF de la pièce

La cinématique de la pièce déformable impose plutôt une discrétisation par des éléments finis tridimensionnels ou par des éléments s'appuyant sur la théorie des coques suivant la géométrie de la pièce. Etant donné que nous nous concentrons sur l'étude des pièces minces, la modélisation repose sur une approche classique par éléments finis en utilisant des éléments de type plaque ou coque (voir chapitre 3 ou annexe pour plus d'information sur l'élément coque utilisé).

1.3.2 Modèles géométriques

Le suivi de l'interaction outil-pièce et de l'évolution du domaine pièce impose le choix de deux modèles géométriques : l'un pour décrire le domaine occupé par l'outil $\Omega_{outil}(t)$, plus exactement pour décrire le volume balayé par ses faces de coupe, l'autre pour décrire le domaine pièce $\Omega_{pièce}(t)$ et l'enlèvement de matière qui est un des objectifs de la simulation.

Ces modèles géométriques seront développés plus en détail dans le chapitre 3.

Modèle géométrique de l'outil

Le domaine occupé par l'outil $\Omega_{outil}(t)$ est décrit par sa frontière $\partial\Omega_{outil}$. Cette dernière est approximée par un modèle B-Rep polyédrique constitué de facettes triangulaires planes. Ce modèle est défini dans un référentiel oR par la donnée d'une configuration non déformée de l'outil ${}^r\Omega_{outil}$ à laquelle est associé le vecteur déplacement de chacun des sommets du polyèdre. En fraisage, oR est un référentiel tournant.

Comme l'usure de l'outil n'est pas actuellement prise en compte dans notre modèle, le domaine matériel ${}^r\Omega_{outil}$ occupé par l'outil ne change pas et il en est de même de toute sa frontière. En particulier la géométrie de chaque face de coupe de l'outil est inchangée. De plus, les mouvements des différents repères permettant de définir la cinématique de l'outil sont connus.

Le modèle géométrique de l'outil, ou de chaque partie active d'outil élémentaire, permet de construire, entre t et $t + \Delta t$, un modèle géométrique du volume balayé par cet outil élémentaire entre ces deux instants. Pour le cas d'une simulation tridimensionnelle, le modèle géométrique du volume balayé est surfacique. Il est construit en utilisant des facettes triangulaires planes d'un modèle B-Rep.

Modèle géométrique de la pièce

Avec le modèle géométrique de la pièce retenue dans les travaux antérieurs (modèles de type B-Rep analogue aux modèles de CAO) [Mar03], le problème de très nombreuses intersections de polyèdres conduit à des facettes "dégénérées" dont la prise en compte dans l'algorithme d'intersection est loin d'être triviale et constitue actuellement un des points bloquants pour ce qui concerne la robustesse de l'algorithme d'intersection.

Nous avons choisi, dans ce travail, de contourner ces difficultés en changeant radicalement la modélisation géométrique de la pièce. La partie à usiner de la pièce a ainsi été modélisée par un modèle volumique utilisant des dexels. Cette méthode augmente la robustesse de l'algorithme d'intersection, mais l'inconvénient de ce modèle géométrique est qu'il ne permet de modéliser que des opérations d'usinage relativement simples sur des pièces de forme simple.

Pour ne plus rencontrer ce problème, il serait possible de créer un modèle avec des dexels suivant les trois directions orthogonales comme le modèle triple-nailboard. Nous n'avons pas retenu actuellement cette possibilité.

1.3.3 Modèle de coupe

L'évaluation des efforts de coupe repose sur une structure de modèle des efforts de coupe qui découle des choix effectués pour modéliser la géométrie de l'outil, et plus particulièrement des parties de face de coupe et de face en dépouille qui sont directement concernées par l'usinage simulé. Cette évaluation doit tenir compte aussi de la façon de réaliser le modèle géométrique de la pièce (ici des dexels) et de calculer l'interaction entre le volume balayé une partie et le domaine pièce.

Démarche adoptée avec le modèle géométrique de l'outil

Le modèle retenu pour l'outil, décomposition en plusieurs arêtes élémentaires appelées *parties*, entraîne l'utilisation de la modélisation de Kline et DeVor avec un découpage de l'effort de coupe qui devient la somme des efforts élémentaires exercés sur chaque arête élémentaire en prise avec la matière.

Un point M_i d'application d'effort, qui peut être vu comme un point d'intégration numérique des efforts de coupe, est associé à chaque partie i d'arête de coupe. Sur la figure 1.8, ce point M_i est représenté sur l'arête de coupe elle même.

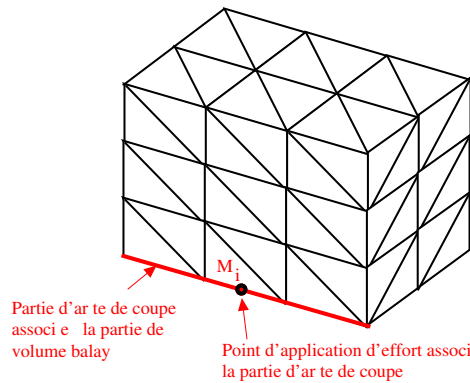


FIG. 1.8 – Modèle géométrique du domaine balayé par la face de coupe de l'outil au voisinage de l'arête de coupe

Loi de l'effort de coupe

La multiplicité et la diversité des lois de coupe qu'on rencontre dans la littérature confirment qu'aucun modèle général du calcul de l'effort de coupe qui soit admis par tous n'est parfaitement établi.

Parmi la variété des lois de coupe proposées, la loi la plus adaptée aux simulations numériques peut sembler être, à première vue, celle où l'on utilise des formules préétablies pour chaque procédé de coupe. Cependant, ces formules liées à une géométrie figée de l'outil, ne tiennent pas compte, en général, de la position réelle de l'outil par rapport à la pièce, ni de la vitesse relative réelle outil - pièce dans la zone d'usinage. Ceci n'est pas compatible avec l'approche retenue dans notre étude. En effet, l'examen du problème que nous traitons montre qu'on est en présence d'une coupe tridimensionnelle avec plusieurs arêtes de coupe. De plus, comme on vise l'utilisation de la simulation pour explorer des conditions d'usinage à la limite de la stabilité, il faut impérativement utiliser une loi de coupe dynamique.

En résumé, on est en présence d'une coupe dynamique en 3D, qui est très peu examinée dans la littérature. Cette ambiguïté dans le choix de la loi de coupe implique la définition d'un modèle de calcul du vecteur effort généralisé sous une forme aussi modulable que possible, de façon à pouvoir intégrer la plus grande variété possible de lois dans les simulations, ne serait-ce que pour les évaluer lorsqu'elles seront précisées par d'autres chercheurs.

Dans la solution adoptée dans ce travail, nous associons à chaque volume balayé par une partie i les trois entités suivantes :

1. Une loi de coupe qui permet de calculer la résultante $\overrightarrow{F_i^{coupe}}$ des efforts de coupe, pour chaque volume balayé par une partie i , à partir de certains paramètres de la coupe. La loi la plus simple consiste à définir dans un repère de mouvement connu chacune des composantes des efforts de coupe par une formule de type :

$$F_i^{coupe\ j} = k_j \frac{b_i}{h_0} \left(\frac{h_i}{h_0} \right)^{\alpha_j} \quad (1.9)$$

avec $F_i^{coupe\ j}$ la j -ème composante dans l'espace de F_i^{coupe} .

b_i la profondeur de passe du volume balayé par la partie i .

h_i l'épaisseur de coupe du volume balayé par la partie i .

h_0 l'épaisseur de coupe de référence.

k_j et α_j des constantes de coupe caractérisant cette loi.

2. Un point tM_i où s'applique l'effort de coupe $\overrightarrow{F_i^{coupe}}$ défini par la loi de coupe.
3. L'approche classique consiste à définir le vecteur force dans un repère local connu a priori qui lié à la configuration non déformée de l'outil.

Démarche adoptée avec les dexels

Devant les nombreuses difficultés techniques (voir entre autres la thèse d'Audrey Marty [Mar03]) qui sont liées au calcul itératif, auxquelles se rajoutent celles liées à l'anisotropie dans la représentation du modèle géométrique de la pièce par des dexels, une approche de type un peu différent est utilisée pour régulariser l'évaluation mathématique des aires de contact face de coupe / pièce.

Une aire moyenne équivalente peut être définie à partir du volume des dexels coupés pendant l'incrément de temps par chaque partie i et du déplacement moyen de la face de coupe de la partie ou de la partie elle-même (par exemple calculé au point M_i ou au barycentre du domaine balayé de la partie). C'est cette aire moyenne équivalente qui est

utilisée en lieu et place de S_i . Contrairement à la section réelle du copeau, cette section équivalente varie de façon continue. Cette approche est bien adaptée aux résolutions numériques pas à pas que nous utilisons.

1.4 Conclusion

Nous avons, dans ce chapitre, présenté les objectifs visés ainsi que le contexte de ce travail. Un exposé des différents modèles nécessaires à la description du comportement dynamique du système POM a été fait en précisant ceux que nous allons utiliser.

En particulier, afin de résoudre les problèmes rencontrés lors des travaux précédents du LMSP, nous avons utilisé un nouveau modèle géométrique pour représenter le domaine de la pièce. Ce modèle qui utilise des dexels nous a conduit à appréhender différemment la notion d'enlèvement de matière pour une pièce déformable (chapitre 2) ainsi qu'à mettre en place un algorithme d'intersection robuste (chapitre 3). Cette façon de procéder permet ensuite de mettre en place une nouvelle approche du calcul des efforts de coupe (chapitre 3).

Chapitre 2

Usinage en présence de pièces flexibles : définition et mise en œuvre

La simulation numérique de l'usinage à l'échelle macroscopique s'intéresse encore très peu à la simulation de pièces réellement flexibles dans la zone d'usinage avec prise en compte du mouvement fin de l'outil (en fraisage : flexibilité + rotation à vitesse non infinie + détail des faces de coupe + vibrations) en ayant pour objectif les descriptions fines : 1) des comportements mécaniques statique et dynamique du système Pièce-Outil-Machine pendant l'usinage 2) de la géométrie finale de la pièce et de sa surface usinée.

Dans ce cadre, il apparaît en effet quelques difficultés liées d'une part à la définition correcte de l'enlèvement de matière à mesure que l'usinage s'effectue, d'autre part à la mise en place d'une démarche concrète et générale pour sa prise en compte efficace au sein d'un simulateur s'appuyant sur des calculs de mécanique, qu'ils soient statiques ou dynamiques. La flexibilité de l'outil ne pose pas de problème particulier : seule la pièce réellement flexible (dont le comportement ne peut pas être assimilé à celui d'un mouvement de corps rigide) pose un problème délicat de description du mouvement relatif outil-pièce. Or c'est ce mouvement qui est à l'origine de l'enlèvement de matière et des efforts qui en résultent.

Nous décrivons dans ce chapitre une méthode générale qui peut être utilisée pour construire un tel simulateur. Cette méthode s'appuie sur une configuration matérielle dans laquelle nous définissons l'enlèvement de matière de façon précise (paragraphe 2.1). Ceci nous conduit à définir une application $\phi_{Pièce}^+(\bullet, t)^{-1}$ permettant de préciser les domaines balayés par l'outil dans la configuration matérielle. En réalité, c'est l'application $\phi_{Pièce}^+(\bullet, t)$ qui est définie grâce à un maillage éléments finis réalisé avec des hexaèdres réguliers (paragraphe 2.4). Ce type de maillage est retenu car bien adapté à la modélisation géométrique de la pièce par des dexels.

Le calcul éléments finis mécanique est quant à lui réalisé en utilisant des éléments coques. Un transfert de champ est réalisé entre ces deux maillages (paragraphe 2.3).

2.1 Définition de l'enlèvement de matière

En raison du mouvement simultané de l'outil et de la pièce, il n'est pas évident de définir correctement l'enlèvement de matière. Pour y arriver, il est nécessaire de choisir une configuration où la pièce ne se déforme pas. Cette configuration de référence que nous notons " C est une configuration "matérielle" dans laquelle les particules matérielles de la pièce initiale restent fixes $\forall t \in [0, T]$ même après avoir été usinées. Ceci revient à définir un système de coordonnées curvilignes attaché à la matière initiale. Nous pouvons alors

représenter ${}^rV_{balayé}$, le domaine 3D, produit par le déplacement des faces de coupe de l'outil dans cette configuration. Pour un intervalle de temps $[t_1, t_2]$, le volume usiné est alors défini dans la configuration rC par la relation suivante :

$${}^rV_{usiné}(t_1, t_2) = {}^r\Omega_{Pièce}(t_1) \cap {}^rV_{balayé}(t_1, t_2) \quad (2.1)$$

${}^r\Omega_{Pièce}(t_1)$ est le sous ensemble des points matérielles de la pièce qui sont encore présents après les enlèvements de matière effectués jusqu'à l'instant t_1 .

Nous allons définir plus en détail cette relation dans les paragraphes suivants.

2.1.1 Configurations matérielle rC et déformée tC

Configuration actuelle de la pièce déformée tC

A chaque instant t de l'intervalle d'étude $[0, T]$, la configuration déformée de la pièce est notée tC . La pièce est déformée suite à son bridage et par les différents contacts avec les faces de coupe de l'outil qui la déforment et la font vibrer.

Pour observer la pièce dans son état réel (c'est-à-dire déformé) à chaque instant, nous introduisons un référentiel mobile de E^3 , noté tR .

Configuration matérielle rC

La configuration de référence dite matérielle ${}^rC(t)$ est une configuration donnée, fixe qui permet de définir correctement l'enlèvement de matière. Cette configuration est paramétrée par le temps t car l'enlèvement de matière modifie la frontière du domaine occupé par la pièce ou, plus précisément, diminue la quantité de matière constituant la pièce, et par suite le domaine occupé par la pièce. En absence d'usinage, la quantité de matière reste constante et cette configuration est indépendante du temps.

Lors du processus d'usinage, les différentes configurations de la pièce, ${}^rC(t)$, sont observées dans un référentiel fixe noté rR d'un espace géométrique "matériel" rE_3 .

Il est possible de choisir comme référentiel matériel, celui où la pièce n'est pas déformée et est vierge de toute contrainte, ou tout autre référentiel relié à ce dernier par une application bijective.

Les coordonnées de tout point de la pièce dans le référentiel rR sont, par définition, les coordonnées matérielles de ce point. L'utilisation des coordonnées matérielles est classique en mécanique des milieux continus, lorsque l'on s'intéresse aux grandes transformations.

De par la définition de l'usinage, nous avons :

$$\forall t_2 > t_1 \quad {}^r\Omega_{Pièce}(t_2) \subset {}^r\Omega_{Pièce}(t_1) \quad (2.2)$$

2.1.2 Passage de rC à tC : application $\phi_{Pièce}(\bullet, t)$

Pour suivre les déformations de la partie usinée de la pièce, nous devons seulement définir, à chaque instant t , une application bijective $\phi_{Pièce}(\bullet, t)$ permettant de passer de la configuration rC de référence matérielle à la configuration tC actuelle de la pièce déformée. Cette application définie de façon unique est paramétrée par le temps t . Elle est inconnue $\forall t > 0$. Elle est continue, et différentiable par morceaux. L'image par $\phi_{Pièce}(\bullet, t)$ d'un point rM de la pièce dans la configuration rC est le point tM dans la configuration tC . De

même les images de ${}^r\Omega_{Pièce}(t)$ et de sa frontière $\partial_r\Omega_{pièce}(t)$ sont respectivement le domaine $\Omega_{Pièce}(t)$ et sa frontière $\partial\Omega_{pièce}(t)$.

$$\forall t, \forall {}^rM \in {}^r\Omega_{Pièce}(t), \quad \exists ! {}^tM \in \Omega_{Pièce}, \quad {}^tM = \phi_{Pièce}({}^rM, t) \quad (2.3)$$

2.1.3 Extension de l'application $\phi_{Pièce}(\bullet, t)$ en $\phi_{Pièce}^+(\bullet, t)$

Pour identifier le volume enlevé par les faces de coupe de l'outil entre deux instants t_1 et t_2 , il faut être capable de positionner ces faces de coupe vis-à-vis de la matière de la pièce à chaque instant $t \in [t_1, t_2]$. Un endroit idéal pour suivre le parcours des faces de coupe de l'outil est le référentiel rR où est définie la configuration de référence matérielle. Dans ce référentiel, l'image de la pièce étant *fixe*, l'image du volume enlevé est simplement le volume qui est balayé par les parties des faces de coupe de l'outil se trouvant dans la matière, une fois ramenées dans ce référentiel rR . En effet, le mouvement relatif de l'outil par rapport à la pièce étant non nul, les images des points de l'outil sont mobiles dans ce référentiel.

Une partie seulement de l'ensemble Γ_{coupe} des faces de coupe de l'outil se trouve "immergée" dans le domaine occupé par la pièce $\Omega_{pièce}$. Ne sachant pas, a priori, quelles sont les faces de coupe partiellement immergées, ni quelle partie de chacune d'elles est concernée, il est nécessaire de définir un prolongement $\phi_{Pièce}^+(\bullet, t)$ de $\phi_{Pièce}(\bullet, t)$ tel que $\phi_{Pièce}^+(\bullet, t)$ soit une application bijective, continue et différentiable par morceaux, avec comme seule condition nécessaire que :

$$\forall t, \quad \forall {}^rM \in {}^r\Omega_{Pièce}, \quad \phi_{Pièce}^+({}^rM, t) = \phi_{Pièce}({}^rM, t) \quad (2.4)$$

Cette application n'est pas unique et peut être choisie de la façon la plus commode possible pour la mise en œuvre informatique.

2.1.4 Domaines volumiques balayé et usiné

Pour déterminer le domaine volumique balayé par les parties coupantes de l'outil, il faut être capable, $\forall t \in [0, T]$, de positionner tous les points des parties coupantes de l'outil dans la configuration matérielle rC .

Comme pour le domaine occupé par la pièce $\Omega_{Pièce}$, les faces de coupe de l'outil Γ_{coupe} peuvent être observées dans les référentiels rR et tR , ce qui nous permet d'écrire la relation suivante :

$${}^r\Gamma_{coupe}(t) = (\phi_{Pièce}^+)^{-1}(\Gamma_{coupe}(t), t) \quad (2.5)$$

Le domaine volumique balayé par les faces de coupe de l'outil entre deux instants t_1 et t_2 est défini dans le référentiel rR par la relation suivante :

$${}^rV_{balayé}(t_1, t_2) = \underset{t \in [t_1, t_2]}{Balayage} [{}^r\Gamma_{coupe}(t)] \quad (2.6)$$

La fonction *Balayage* est construite en s'appuyant sur une approximation de la trajectoire, entre les deux instants t_1 et t_2 . Dans ce travail, nous adoptons une interpolation linéaire mais il est possible de choisir une autre interpolation, ce qui est indispensable si le pas de temps entre t_1 et t_2 est important [Pei04].

Le domaine volumique usiné et le domaine occupé par la pièce à l'instant t_2 se déduisent de façon immédiate de l'équation (2.6), toujours dans le référentiel matériel rR :

$${}^rV_{usin\acute{e}}(t_1, t_2) = {}^r\Omega_{Pi\grave{e}ce}(t_1) \cap {}^rV_{balay\acute{e}}(t_1, t_2) \quad (2.1')$$

$${}^r\Omega_{Pi\grave{e}ce}(t_2) = {}^r\Omega_{Pi\grave{e}ce}(t_1) - {}^rV_{usin\acute{e}}(t_1, t_2) \quad (2.7)$$

ou encore

$${}^r\Omega_{Pi\grave{e}ce}(t_2) = {}^r\Omega_{Pi\grave{e}ce}(0) - {}^rV_{usin\acute{e}}(0, t_2) \quad (2.8)$$

La mise en œuvre pratique des applications $\phi_{Pi\grave{e}ce}(\bullet, t)$, $\phi_{Pi\grave{e}ce}^+(\bullet, t)$ et de son inverse est détaillée dans les paragraphes suivants.

2.2 Mise en œuvre avec les dexels

Pour simplifier les calculs d'intersection et obtenir des algorithmes robustes, nous utilisons un modèle géométrique s'appuyant sur des dexels. Ce modèle est construit à partir d'une grille régulière de points situés dans un plan, par exemple le plan (X, Y) , et de segments disposés perpendiculairement à ce plan, par exemple l'axe des Z .

La configuration de référence rC est la seule configuration où les algorithmes d'intersection doivent être définis de façon efficace. Dans la configuration rC , les dexels doivent être parallèles entre eux et régulièrement espacés. Sans la description géométrique utilisant des dexels, de nombreux choix seraient possibles pour déterminer la configuration de référence matérielle rC .

2.2.1 Exemples élémentaires de modèles dans rR

Les dexels peuvent modéliser des pièces de forme complexe, ayant des trous ou ayant des surfaces courbes, ondulées. La figure 2.1 montre quelques exemples en 2D de pièces modélisées par des dexels.

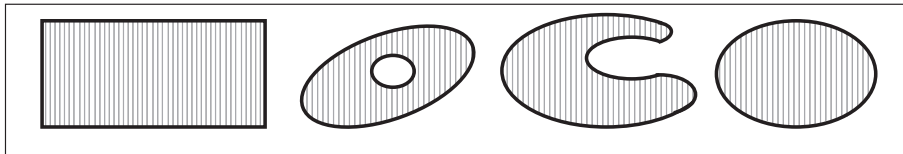


FIG. 2.1 – Différents domaines modélisés par des dexels (en 2D)

Dans la pratique, seule la partie de la pièce qui va être usinée sera modélisée par des dexels. Cela va permettre de modéliser un domaine restreint moins complexe du point de vue géométrique. La figure 2.2 illustre le cas d'un domaine restreint de forme parallélépipédique rectangle.

2.2.2 Paramètres influençant la précision

La précision d'une méthode utilisant des dexels dépend de deux paramètres, d'une part de la densité de dexels sur la grille et d'autre part du choix de l'orientation des dexels.

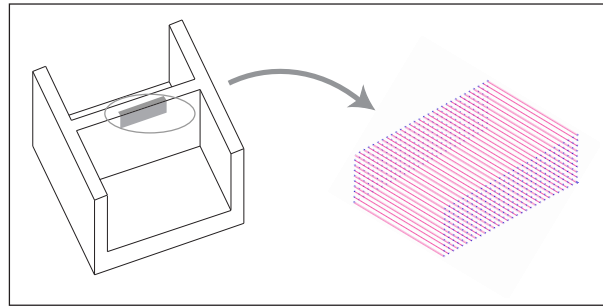


FIG. 2.2 – La zone à usiner est un parallélépipède rectangle

Pas de la grille

Le premier paramètre dépend de l'espacement entre deux dexels parallèles et donc du pas de la grille suivant l'axe des X et des Y . Plus le pas de la grille est faible et plus le domaine occupé par la pièce est bien approximé, mais cela augmente le nombre de dexels pour modéliser la pièce. Ce nombre entraîne une augmentation des calculs numériques et donc une diminution de la vitesse de la simulation. Un compromis est donc à trouver.

Orientation des dexels

Le deuxième paramètre de précision est le choix de l'orientation des dexels. Cette orientation dépend du type d'usinage (fraisage ou tournage) et de la géométrie de la partie de la pièce à usiner.

Si l'on veut par exemple usiner une pièce cylindrique en tournage, il est évident que le fait de disposer tous les dexels suivant la même direction (figure 2.3 à gauche) ne permettra pas d'obtenir la même précision sur toute la surface de la pièce.

Il serait plus judicieux pour obtenir une précision identique sur toute la surface de la zone usinée de diriger les dexels perpendiculairement au mouvement global des faces de coupe de l'outil. La figure 2.3 à droite illustre dans un cas de tournage pour une pièce cylindrique, une orientation homogène, localement, des dexels par rapport à la surface de la pièce.

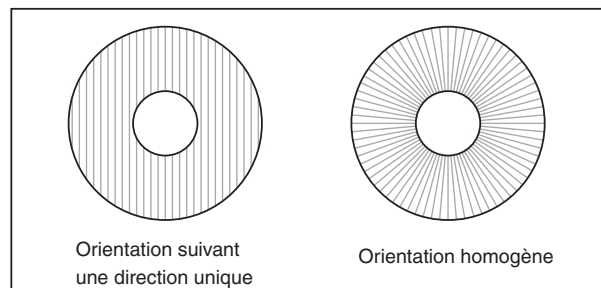


FIG. 2.3 – Exemple d'orientation des dexels en tournage

Cette description de dexels non parallèles pour modéliser la pièce ou une partie seulement de la pièce, à l'instant $t = 0$, est appropriée pour un type d'usinage donné mais

n'est pas réalisable avec les dexels tel qu'ils ont été définis précédemment. Il faut donc introduire une nouvelle application permettant de passer de la configuration matérielle où tous les dexels sont parallèles à une configuration de la pièce non déformée à l'instant $t = 0$ où les dexels ont l'orientation désirée par l'utilisateur.

2.2.3 Configurations intermédiaires

Pour passer de la configuration de référence rC à la configuration déformée actuelle tC , il est possible de définir des configurations intermédiaires ainsi que les applications qui les permettent de passer de l'une à l'autre.

Nous avons distingué en tout quatre configurations :

- La configuration de référence ${}^rC(t)$ dite matérielle. C'est une configuration abstraite : elle ne représente pas la géométrie réelle de la pièce cette dernière est représentée par un ensemble de dexels tous parallèles entre eux. Comme nous l'avons déjà dit cette configuration est fixe et c'est la seule configuration où nous pouvons définir correctement l'enlèvement de matière avec les dexels.
- Nous introduisons une configuration dite relâchée ${}^0\widehat{C}(t)$ où la pièce n'est pas déformée. Il est possible de considérer des contraintes résiduelles dans cette configuration. Ceci n'est pas pris en compte ici : $\sigma_0 = 0$. Il s'agit de la configuration initiale de la pièce avant bridage.
- Dans la configuration initiale 0C , à $t = 0$, la pièce est déformée uniquement suite à un bridage. Cette configuration est indépendante du temps car le processus d'enlèvement de matière n'a pas commencé à $t = 0$.
- Dans la configuration actuelle tC , en plus du bridage l'action de l'outil déforme également la pièce. Il est possible de suivre dans cette configuration et à chaque instant t la déformée de la pièce et l'évolution de l'enlèvement de matière.

Les différentes configurations ${}^rC(t)$ sont observées dans le référentiel fixe rR de l'espace "matériel".

Les autres configurations ${}^0\widehat{C}(t)$, 0C et tC sont observées dans un référentiel mobile tR où tout mouvement de corps rigide de la pièce est interdit. Ce référentiel correspond, par exemple, à un positionnement isostatique de la pièce sur la table de la machine ou à tout autre système de positionnement isostatique. Ceci permet d'observer la pièce en faisant abstraction de tout mouvement d'ensemble parasite (rotation en tournage, translation en fraisage ou mouvement plus complexe sur des machines à commande numérique par exemple).

Sur la figure 2.5, colonne de droite, un exemple, schématique bidimensionnel, d'évolution de la configuration tC déformée de la pièce (observée dans le référentiel tR) à différentes étapes est représenté. Les trois étapes illustrent les principales situations lors de la simulation de l'usinage par outil effaceur de matière. A l'étape initiale $t = 0$, la pièce est déformée suite à un bridage. A l'instant t_1 , par exemple, en plus du bridage l'action de l'outil déforme également la pièce, il est à noter qu'une partie du domaine occupé par la pièce a disparu, enfin, à $t_2 > t_1$, nous nous positionnons à un instant plus avancé dans le processus d'usinage. Les déformations ne sont pas les mêmes qu'à l'instant t_1 et la quantité de matière enlevée a augmenté.

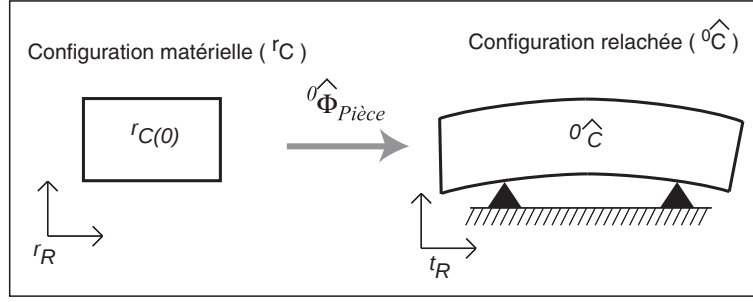


FIG. 2.4 – Passage de la configuration matérielle à la configuration relâchée

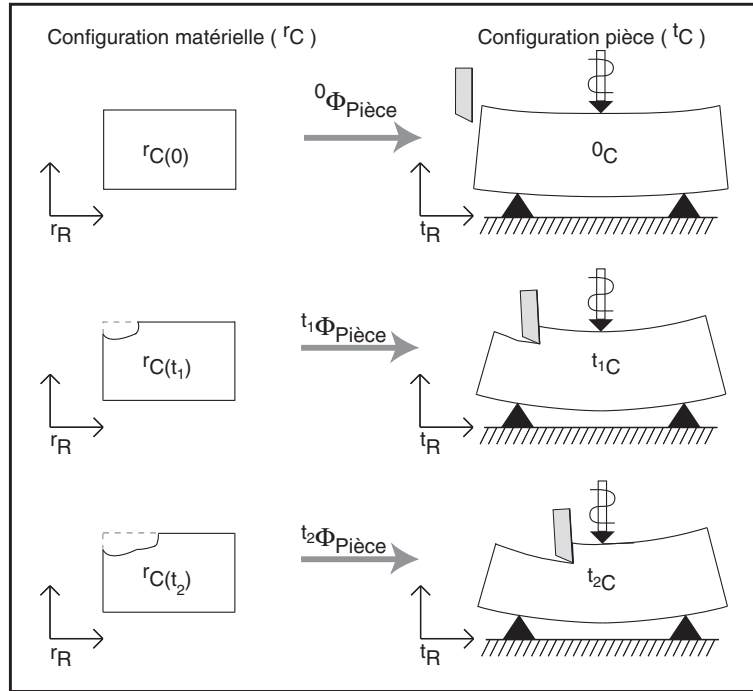


FIG. 2.5 – Passage de la configuration matérielle à la configuration déformée de la pièce à plusieurs étapes

La colonne de gauche de la figure 2.5 représente l'évolution de la configuration rC (observée dans le référentiel rR) aux mêmes étapes que pour la configuration tC . A l'étape initiale $t = 0$, l'enlèvement de matière n'a pas commencé, pour les deux étapes suivantes, nous pouvons suivre l'évolution du contour de la pièce lors de l'enlèvement de matière. Dans ce cas schématique bidimensionnel, contour et frontière du domaine occupé par la pièce sont confondus.

Passage entre rC et ${}^0\hat{C}$

Nous avons besoin d'avoir une certaine maîtrise de la disposition des dexels par rapport à la configuration relâchée ${}^0\hat{C}$ à l'instant $t = 0$. Dans la configuration ${}^0\hat{C}$, les dexels ne sont pas forcément orientés suivant la même direction.

La transformation permettant de passer de la configuration matérielle où une partie de la pièce est représentée par un ensemble de dexels tous parallèles entre eux à la confi-

guration relâchée de la pièce non déformée et vierge de toute contrainte est définie une fois pour toute avant l'usinage de façon unique par une application bijective ${}^0\widehat{\Phi}_{Pièce}$.

$$\forall t, \forall {}^rM \in {}^r\Omega_{Pièce}(t), \quad \exists! {}^0\widehat{M} \in {}^0\widehat{\Omega}_{Pièce}, \quad {}^0\widehat{M} = {}^0\widehat{\Phi}_{Pièce}({}^rM) \quad (2.9)$$

Le domaine occupé par la pièce à l'instant t dans le référentiel rR est noté ${}^r\Omega_{Pièce}(t)$ et celui occupé dans tR est noté ${}^0\widehat{\Omega}_{Pièce}(t)$.

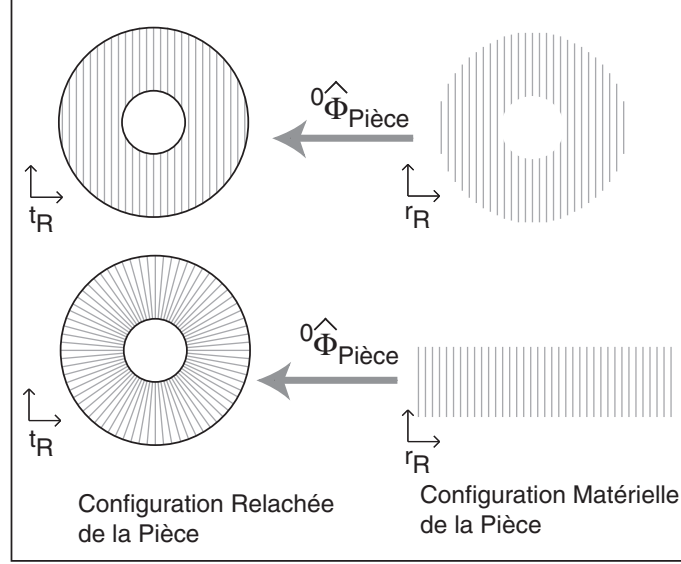


FIG. 2.6 – Passage entre la configuration initiale et la configuration matérielle

Si l'on reprend l'exemple précédent de la figure 2.3, lorsque les dexels sont orientés suivant la même direction, l'application bijective ${}^0\widehat{\Phi}_{Pièce}$ peut être l'identité ou bien une homothétie. L'application ${}^0\widehat{\Phi}_{Pièce}$ et surtout son inverse ont, dans ce cas, l'avantage de s'obtenir grâce à des calculs simples et rapides.

Si les dexels ne sont pas tous orientés suivant une même direction, l'application ${}^0\widehat{\Phi}_{Pièce}$ ainsi que son inverse sont plus complexes à déterminer.

2.3 Construction de $\phi_{Pièce}(\bullet, t)$

Les configurations étant définies précédemment, il est possible de déterminer les différentes étapes de la construction de l'application bijective $\phi_{Pièce}(\bullet, t)$ permettant de passer de la configuration rC de référence matérielle qui peut être abstraite à la configuration tC actuelle de la pièce déformée.

Notre construction de l'application bijective $\phi_{Pièce}(\bullet, t)$ s'appuie sur l'utilisation :

1. d'éléments finis coques pour simuler la cinématique de pièces constituées de voiles minces,
2. de la notion de pavage pour définir le domaine restreint de la pièce qui nous intéresse pour la simulation,
3. d'éléments finis hexaédriques pour construire la bijection $\phi_{Pièce}$ car les éléments coques ne permettent pas d'obtenir une méthode générale pour définir la bijection.

2.3.1 Maillage EF Coque

Du point de vue modélisation mécanique, dès que nous voulons usiner des pièces constituées de voiles minces, l'utilisation d'éléments finis coque ou plaque vient naturellement.

Nous rappelons ici les principales caractéristiques de l'élément quadrilatère isoparamétrique à huit nœuds de coque à double courbure proposé par Olivier Polit [Pol92]. Il a été construit pour être le plus simple et le plus efficace possible. Cet élément fini performant possède les propriétés suivantes :

- utilisation des degrés de liberté (DDL) de type ingénieur : les deux déplacements membranaires, le déplacement transversal, les deux rotations et éventuellement, la rotation autour de la normale à la surface moyenne (cas de structures possédant des discontinuités géométriques),
- élément utilisable dans le domaine allant des coques minces aux semi-épaisses,
- absence de verrouillage en membrane et en cisaillement transverse excluant l'utilisation de coefficients correcteurs et d'artifices de type intégration réduite ou sélective,
- absence de modes à énergie de déformation nulle autres que les modes associés à des mouvements de corps rigide,
- précision des calculs en déplacements, en contraintes et en efforts généralisés,
- bonne rapidité de convergence (précision avec peu de DDL) et absence d'oscillations (convergence monotone),
- élément facile à implémenter dans un code de calcul,
- possibilité de modélisation des structures coques à géométrie complexe (métriques et courbures non constants).

L'élément de référence utilisé est un carré biunitaire sur lequel est défini un repère appelé réduit : $(\xi, \eta) \in [-1, 1] \times [-1, 1]$. Cet élément possède 8 nœuds dont la numérotation est donnée sur la figure 2.7.

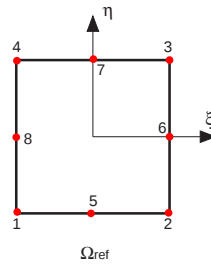


FIG. 2.7 – Élément coque de référence à huit nœuds

Les fonctions de base sont définies par la base canonique notée $\mathcal{B}$ associée à l'espace polynomial :

$$\mathcal{B} = \langle 1 \quad \xi \quad \eta \quad \xi^2 \quad \xi\eta \quad \eta^2 \quad \xi\eta^2 \quad \xi^2\eta \rangle$$

L'espace des degrés de liberté correspond à l'ensemble des formes linéaires associées, obtenues à partir des valeurs de la fonction considérée en chaque nœud. Cet élément est caractérisé par une interpolation identique de la géométrie et des déplacements. Cela permet de calculer les matrices de masse $\underline{\underline{M}}$ et de rigidité $\underline{\underline{K}}$ de toute la pièce et les 6

DDL de la pièce $\underline{q}_{Coque}^{Pièce}$. Le calcul de la matrice de rigidité $\underline{K}$ qui est un peu particulier est rappelé dans l'annexe.

Nous avons mis en place cet élément dans *Nessy* pour assurer les calculs mécaniques des pièces élancées sachant que les autres éléments finis disponibles dans *Nessy* (hexaèdres, poutres) ne permettaient pas de modéliser ces pièces.

2.3.2 Pavage 2D

Afin de maîtriser le stockage des entités associées au modèle géométrique, seule la partie de la pièce dont nous voulons simuler l'usinage est prise en considération. Cette partie est ici supposée modélisée par des éléments coque. C'est donc uniquement dans ce domaine que les dexels vont être inclus. L'ensemble de la pièce est constitué d'un maillage EF structuré ou pas. Nous regroupons dans un pavage 2D rectangulaire régulier les éléments coques appartenant à la partie à usiner de la pièce. Ce pavage est un maillage EF structuré de rC . La cinématique des nœuds de l'image de ce pavage par ${}^0\hat{\Phi}_{Pièce}$ est esclave de la cinématique EF du maillage de calcul de la pièce.

L'intérêt d'avoir un maillage régulier est de trouver facilement à quel élément le point tM appartient et de connaître les coordonnées réduites du point tM dans cet élément.

Le pavage de ${}^0\hat{C}$, image du pavage régulier de rC par ${}^0\hat{\Phi}_{Pièce}$, peut être disposé n'importe où dans le modèle EF (dans sa partie coque).

Nous notons $\underline{q}_{pavage}$ la matrice colonne qui contient l'ensemble des DDL des nœuds appartenant au pavage. La matrice colonne $\underline{q}_{pavage}$ est extraite de façon très simple de la matrice colonne $\underline{q}$ qui est connue après chaque étape de résolution de l'équation d'équilibre dynamique (équation (1.8)). $\underline{q}_{pavage}$ est donc connue après chaque itération.

Rien ne s'oppose, dans une version ultérieure de *Nessy*, de pouvoir modifier la liste des éléments qui composent le pavage au fur et à mesure que l'usinage progresse.

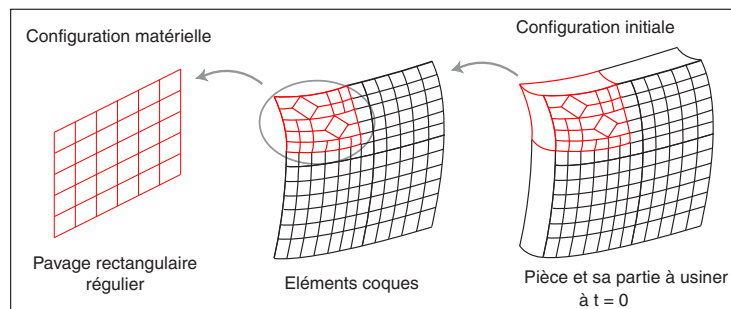


FIG. 2.8 – Passage d'une configuration initiale à une configuration matérielle

2.3.3 Passage à des éléments hexaédriques

Pour des raisons de commodité, nous introduisons un maillage 3D constitué d'éléments finis hexaédriques, entièrement lié au maillage éléments finis coque du pavage 2D, pour définir l'application bijective $\phi_{Pièce}(\bullet, t)$. Les éléments retenus [Bat96] sont des éléments hexaédriques isoparamétriques classiques de type *Hexa8*.

Ce maillage régulier en hexaèdres est construit dans la configuration matérielle rC dans laquelle le domaine représenté par les dexels est inclus.

Un des avantages de choisir un maillage 3D hexaédrique est de pouvoir appliquer la démarche à des pièces massives.

L'objectif est de transférer les déplacements généralisés des nœuds des EF coque aux nœuds des EF hexaèdre de façon à assurer le lien cinématique entre le maillage 3D en hexaèdres et le pavage 2D de coques.

Raison du choix d'un maillage en hexaèdres

Le choix d'un maillage en hexaèdres, en s'attachant à ce qu'il soit régulier pour le pavage, a été retenu pour accélérer la procédure de détermination de l'élément e dans lequel se trouve un point connaissant ses coordonnées x_i . En choisissant l'origine du repère au sommet du parallélépipède associé à la pièce qui assure que tous les points de ce domaine ont des coordonnées positives ou nulles, une simple division entière de chacune des coordonnées x_i par $\frac{l_i}{n_i}$ donne la rangée de l'élément dans cette direction. Une numérotation croissante des éléments, analogue à celle retenue pour l'adressage des tableaux à 3 indices dans les langages de programmation tels que *Fortran* ou *C* assure l'efficacité de la méthode. Enfin, un filtrage complémentaire, pour le cas où le point se trouve dans les couches extérieures, complète le dispositif.

Mise en œuvre pratique

Découpage du pavage

Le pavage est composé d'un ensemble d'EF coque disposé sur un maillage régulier. Le nombre d'éléments coque dans ce pavage est de :

$$nbElemPavage = nbCoucheCoqueX \times nbCoucheCoqueY$$

où $nbCoucheCoqueX$ est le nombre de couches d'éléments de coque suivant l'axe des X et $nbCoucheCoqueY$ est le nombre de couches d'éléments de coque suivant l'axe des Y du pavage.

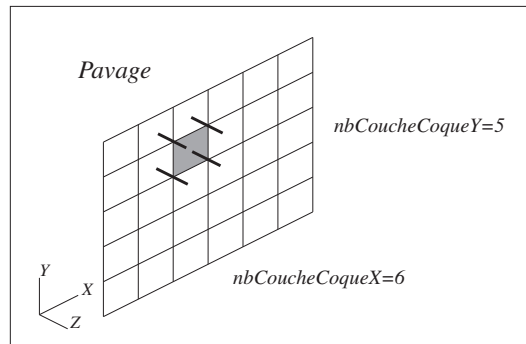


FIG. 2.9 – Exemple d'un découpage du pavage en EF coque

Chaque élément fini coque est décomposé en plusieurs éléments finis hexaédriques à 8 nœuds. Le découpage d'un élément coque se fait de la manière suivante :

- α couches d'hexaèdres suivant l'axe des X ,

- β couches d'hexaèdres suivant l'axe des Y ,
- γ couches d'hexaèdres suivant l'axe des Z .

Les paramètres α , β et γ sont des entiers choisis par l'utilisateur.

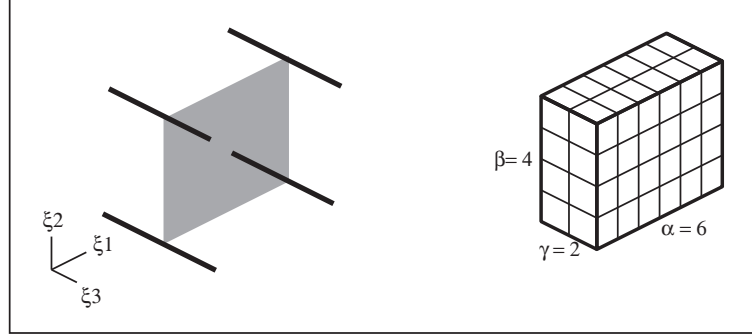


FIG. 2.10 – Exemple d'un découpage d'un EF coque en un ensemble d'hexaèdres

Le nombre d'éléments hexaédriques que comporte le domaine parallélépipédique représentant le pavage est :

$$nbElemHexaPavage = n_1^p \cdot n_2^p \cdot n_3^p$$

avec

$$n_1^p = \alpha \quad nbCoucheCoqueX \quad (2.10)$$

$$n_2^p = \beta \quad nbCoucheCoqueY \quad (2.11)$$

$$n_3^p = \gamma \quad (2.12)$$

Première étape

Dans un premier temps, nous devons calculer les déplacements généralisés $(\vec{U}, \vec{\Theta})$ des points G provenant de l'intersection du découpage en hexaèdres avec la surface de référence des éléments coques. Cette intersection forme une grille régulière sur la surface de référence de l'élément coque. Les coordonnées de ces points G sont (ξ_1, ξ_2) avec $\xi_3 = 0$. Nous devons connaître la position relative de ces points G par rapport aux nœuds d'un EF coque. Pour cela nous nous positionnons dans l'élément isoparamétrique coque de référence Ω_{ref} , de coordonnées $(\xi_1, \xi_2) \in [-1, 1] \times [-1, 1]$.

Nous obtenons alors très simplement les coordonnées réduites de tous les points G car entre deux points G , nous avons un écart de $\frac{2}{\alpha}$ suivant l'axe ξ_1 et $\frac{2}{\beta}$ suivant l'axe ξ_2 .

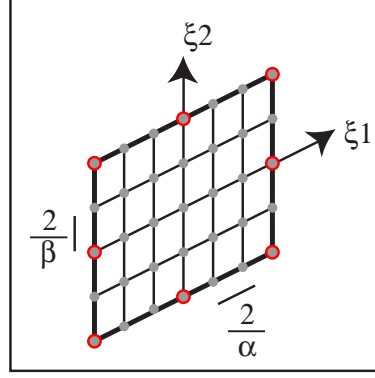
A partir du couple de coordonnées réduites (ξ_1, ξ_2) de chaque point G et des fonctions de base des EF coque, il est possible d'obtenir les déplacements généralisés du segment traversant associé au point G par la relation suivante :

$$\underline{U}_{geneCoq}(\xi_1, \xi_2) = \underline{N}_{elem}(\xi_1, \xi_2) \cdot \underline{U}_{elem} \quad (2.13)$$

* $\underline{U}_{geneCoq}$ regroupe les six déplacements généralisés du segment traversant associé au point G (ξ_1, ξ_2) ,

* $\underline{N}_{elem}$ regroupe les fonctions de base de l'élément coque à huit nœuds,

* $\underline{U}_{elem}$ regroupe les six DDL de chacun des huit nœuds de l'élément coque.

FIG. 2.11 – Représentation des huit nœuds d'un EF coque et les points G

Les fonctions de base pour un élément coque à huit nœuds $\Psi(i)$ où i précise le numéro du nœud sont les suivantes :

$$\Psi(1) = -\frac{(1 - \xi_1)}{2} \frac{(1 - \xi_2)}{2} (1 + \xi_1 + \xi_2) \quad (2.14)$$

$$\Psi(2) = -\frac{(1 + \xi_1)}{2} \frac{(1 - \xi_2)}{2} (1 - \xi_1 + \xi_2) \quad (2.15)$$

$$\Psi(3) = -\frac{(1 + \xi_1)}{2} \frac{(1 + \xi_2)}{2} (1 - \xi_1 - \xi_2) \quad (2.16)$$

$$\Psi(4) = -\frac{(1 - \xi_1)}{2} \frac{(1 + \xi_2)}{2} (1 + \xi_1 - \xi_2) \quad (2.17)$$

$$\Psi(5) = (1 - \xi_2) \frac{(1 - \xi_1^2)}{2} \quad (2.18)$$

$$\Psi(6) = (1 + \xi_1) \frac{(1 - \xi_2^2)}{2} \quad (2.19)$$

$$\Psi(7) = (1 + \xi_2) \frac{(1 - \xi_1^2)}{2} \quad (2.20)$$

$$\Psi(8) = (1 - \xi_1) \frac{(1 - \xi_2^2)}{2} \quad (2.21)$$

Seconde étape

Ici nous nous intéressons au transfert du champs de déplacement décrit par le pavage $2D$ en éléments coques à son approximation par le maillage $3D$ hexaédrique destiné aux calculs d'intersection.

Nous devons donc calculer les trois déplacements de tous les nœuds M des hexaèdres. Ces nœuds M appartiennent au pavage en éléments coques qui est un domaine parallépipédique rectangle d'épaisseur h .

Il est facile d'identifier l'élément coque contenant le nœud M et de déterminer les coordonnées réduites (ξ_1, ξ_2, ξ_3) du nœud M dans cet élément .

Grâce à l'hypothèse cinématique de base du segment traversant faite en HPP, dans le modèle coque, nous avons la relation suivante :

$$\vec{U}(M, t) = \vec{U}(G, t) + \vec{\Theta}(G, t) \wedge \overrightarrow{GM} \quad (2.22)$$

* Le point G est un point de la surface de référence, il a comme coordonnées $(\xi_1, \xi_2, 0)$.

- * Le point M se trouve sur la normale à la surface de référence en G tel que $\overrightarrow{GM} = \xi_3 \vec{n}$ définisse le segment traversant.
- * $\vec{U}(G, t)$ est le déplacement du point G à l'instant t .
- * $\vec{\Theta}(G, t)$ est la rotation du segment traversant associé au point G à l'instant t .
- * $\vec{U}(M, t)$ est le déplacement du point M à l'instant t .

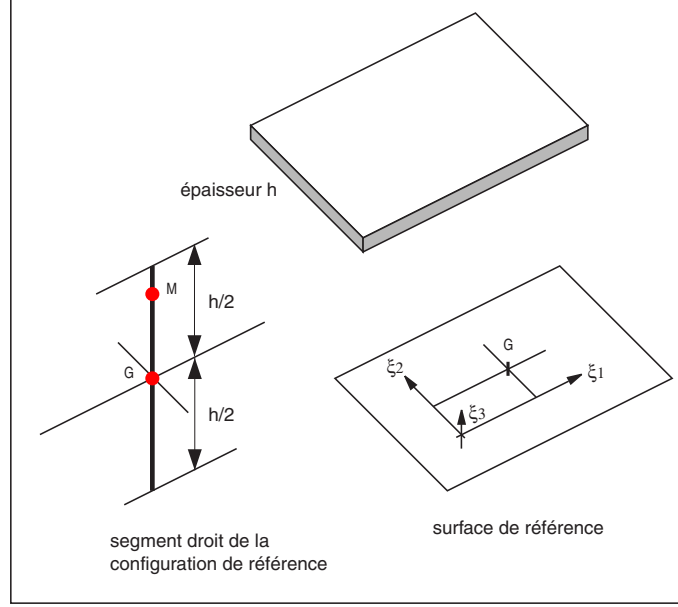


FIG. 2.12 – Modèle simplifié d'une plaque

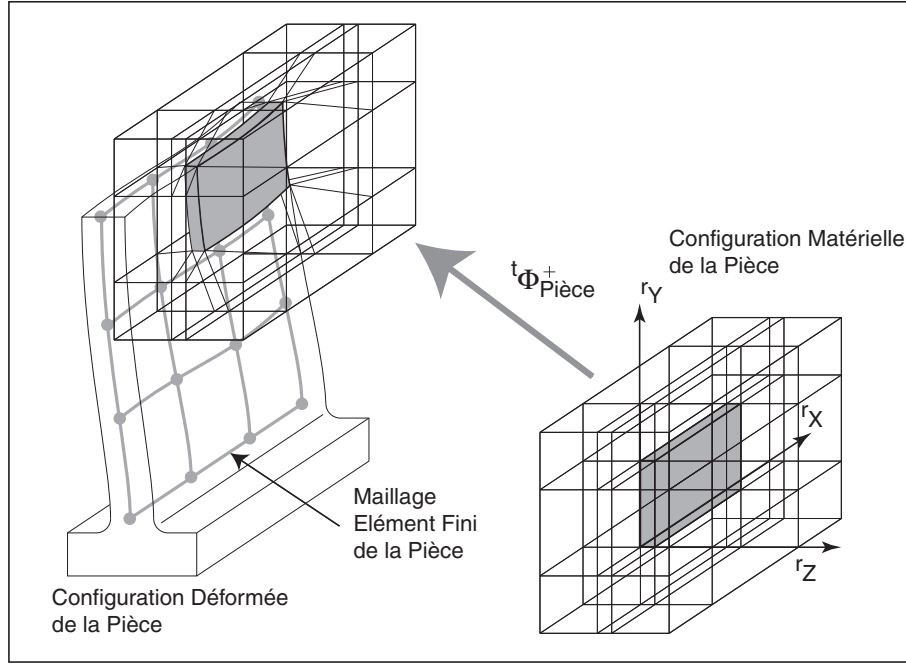
Nous avons précisé les différentes étapes permettant de transférer les champs de déplacements des nœuds des éléments coques aux nœuds des éléments hexaèdres définissant le maillage 3D régulier. Ceci peut se traduire à l'aide d'une matrice de passage cinématique $\underline{G}_{Hexa}^{Pavage}$ qui est indépendante du temps.

$$\underline{q}_{Hexa}(t) = \underline{G}_{Hexa}^{Pavage} \cdot \underline{q}_{Pavage}(t) \quad (2.23)$$

- * la colonne $\underline{q}_{Hexa}$ contient les déplacements des $n_{nœuds}^p = (n_1^p + 1) \times (n_2^p + 1) \times (n_3^p + 1)$ nœuds sommets des hexaèdres du maillage 3D régulier possédant chacun 3 paramètres cinématiques (DDL) correspondant chacun à sa translation suivant une des directions i . En définitive, ces $3 \times n_{nœuds}^p$ paramètres cinématiques sont regroupés dans la matrice colonne $\underline{q}_{Hexa}$.
- * la colonne $\underline{q}_{Pavage}$ contient les déplacements généralisés des $n_{nœuds}^{coque} = (nbCoucheCoqueX + 1) \times (nbCoucheCoqueY + 1)$ nœuds des éléments coques du pavage 2D (6 DDL par nœud).

2.4 Construction pratique de $\phi_{Pièce}^+$

Pour prendre correctement en compte les faces de coupe de l'outil, nous avons besoin d'étendre le domaine de l'application $\phi_{Pièce}(\bullet, t)$. Pour cela nous décidons de plonger le

FIG. 2.13 – Construction de $\phi_{Pièce}^+$ - Maillage hexaédrique

pavage représenté par un maillage 3D régulier d'hexaèdres dans une boîte englobante qui est un domaine parallélépipédique englobant.

La connaissance de $\underline{q}_{Hexa}$ et d'une interpolation classique sur le maillage 3D des hexaèdres vont permettre de définir complètement l'application $\phi_{Pièce}^+(\bullet, t)$ qui est l'extension de l'application $\phi_{Pièce}(\bullet, t)$.

Nous adoptons pour des raisons de commodité de calcul une application $\phi_{Pièce}^+(\bullet, t)$ telle que $\phi_{Pièce}^+(\bullet, 0)$ soit l'application identité. Dans ces conditions, nous avons :

$$\Omega_{Pièce}(0) = {}^r\Omega_{Pièce}(0) \quad \text{et} \quad M(0) = {}^rM \quad (2.24)$$

Nous utilisons donc ici un maillage 3D régulier de forme parallélépipédique rectangle, dont les arêtes sont parallèles aux axes d'un système de coordonnées cartésiennes ortho-normées, et dont la longueur suivant l'axe rX_i est notée l_i .

Le maillage de la boîte englobante est constitué de deux parties : l'une centrale est constituée d'un maillage régulier en parallélépipèdes identiques discrétisant le pavage, l'autre vient entourer cette partie centrale, de façon à former une boîte englobante, en rajoutant deux couches d'éléments dans chaque direction. Dans chacune des directions, les deux couches supplémentaires sont disposées de part et d'autre du domaine ${}^r\Omega_{Pièce}$ et permettent de définir la partie utile du prolongement $\phi_{Pièce}^+$ de $\phi_{Pièce}$. Elles ont respectivement pour épaisseurs l_{iav} et l_{iap} de sorte que la dimension L_i du parallélépipède englobant suivant l'axe rX_i est donnée par :

$$L_i = l_{iav} + l_i + l_{iap} \quad (2.25)$$

Le maillage utilisé pour discrétiser la boîte englobante comporte $n_i^p + 2$ éléments suivant chaque axe rX_i . Tous les sommets d'éléments appartenant aux faces du domaine englobant sont fixes : leurs déplacements sont nuls. La transformation $\phi_{Pièce}^+$ ainsi définie se réduit donc à l'identité à l'extérieur de ce domaine englobant.

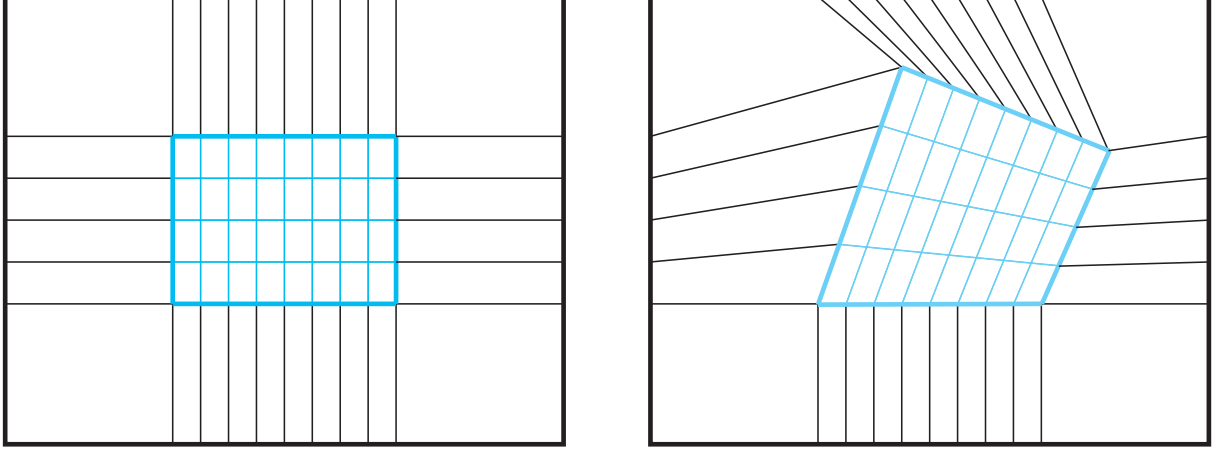


FIG. 2.14 – Représentation de la boîte englobante en 2D

L'application $\phi_{Pièce}^+$ est complètement définie pour tout point dont la position de référence est située à l'intérieur de la boîte englobante de la pièce et si la boîte est suffisamment grande, il n'y a aucun point utile à considérer hors de celle-ci. A l'extérieur de cette boîte, sa position est inchangée.

2.5 Construction pratique de $(\phi_{Pièce}^+)^{-1}$

La définition de $\phi_{Pièce}^+(\bullet, t)$ par une approche de type Eléments Finis fournit une démarche simple, systématique et efficace. Par contre, la définition de la transformation inverse de $\phi_{Pièce}^+$ n'est pas immédiate et nécessite un calcul itératif. Cette construction est pourtant indispensable pour trouver la position de chaque point des faces de coupe de l'outil, dans le référentiel matière, sachant que cette position est connue, dans la configuration déformée de l'outil et dans le référentiel pièce.

2.5.1 Première estimation de la position de référence

Pour déterminer la première estimation des coordonnées des points des faces de coupe dans la configuration de référence, on suppose par exemple [Bea99] que ces points ont les mêmes coordonnées qu'à l'instant précédent. Nous notons cette première estimation ${}^rM^{(0)}$.

2.5.2 Correction itérative

Choisissons un point M , connu de la configuration déformée de la pièce, dont la position de référence rM est inconnue. La détermination de rM se fait grâce à la résolution de l'équation :

$${}^rM = \phi_{Pièce}^{-1}(M, t) \quad (2.26)$$

L'application $\phi_{Pièce}(\bullet, t)$ est définie élément par élément et, pour un élément donné, est construite à partir des fonctions de base de cet élément. Généralement ces fonctions de base sont des fonctions non-linéaires des coordonnées réduites du point rM , de sorte que la résolution de (2.26) conduit à la résolution d'un système non-linéaire.

Dans le cas non-linéaire, $\phi_{Pièce}^{-1}(\bullet, t)$ ne pouvant pas être calculée directement, une méthode de Newton a été choisie pour la résolution. Comme en Mécanique des Milieux Continus, nous utilisons ${}^t\mathbf{F}$, l'application linéaire tangente associée à $\phi_{Pièce}(\bullet, t)$, dont l'expression est :

$${}^t\mathbf{F}({}^rM, t) = \mathbf{Grad} \left(\overrightarrow{\phi_{Pièce}}({}^rM, t) \right) \quad (2.27)$$

Pour chaque point M de la configuration déformée, une estimation de la position de référence ${}^rM^{(0)}$ a été choisie (paragraphe 2.5.1). Cette position va être utilisée comme point de départ de la méthode de Newton. $M^{(0)} = \phi_{Pièce}({}^rM^{(0)}, t)$ est calculé par application directe de $\phi_{Pièce}(\bullet, t)$.

Le point trouvé est alors comparé à M en calculant la distance entre M et $M^{(0)}$. Lorsque la norme de cette distance $\|{}^r\overrightarrow{\delta M}^{(0)}\|$ est inférieure à une précision *pdfilon* donnée, ${}^rM^{(0)}$ est le point solution. Si le point proposé n'est pas le point solution, on calcule un vecteur ${}^r\overrightarrow{\delta M}^{(1)}$ corrigeant la position de ${}^rM^{(0)}$ à l'aide de l'application linéaire tangente calculée en ${}^rM^{(0)}$:

$${}^r\overrightarrow{\delta M}^{(1)} = {}^t\mathbf{F}^{-1}({}^rM^{(0)}, t) \cdot \overrightarrow{\delta M}^{(0)} \quad (2.28)$$

et permettant de déduire une nouvelle estimation de la position recherchée ${}^rM^{(1)}$:

$$\overrightarrow{O^rM}^{(1)} = \overrightarrow{O^rM}^{(0)} + {}^r\overrightarrow{\delta M}^{(1)}$$

Une nouvelle erreur est ensuite calculée permettant de comparer la position du point $M^{(1)}$ à M et ainsi de suite. Pour l'itération i , nous avons ainsi

$$M^{(i)} = \phi_{Pièce}({}^rM^{(i)}, t)$$

et le processus est ensuite répété jusqu'à ce que l'on obtienne une solution ${}^rM^{(i)}$ telle que :

$$\|\overrightarrow{MM}^{(i)}\| = \|\overrightarrow{\delta M}^{(i)}\| \leq \epsilon$$

2.6 Conclusion

Dans ce chapitre, nous avons défini la notion d'enlèvement de matière en en donnant une définition précise, et apte à décrire correctement cet enlèvement, lorsque la pièce se déforme dans la zone usinée.

Dans une première partie, les différentes configurations introduites pour définir l'enlèvement de matière ont été présentées. L'utilisation du modèle géométrique s'appuyant sur des dexels impose à la configuration matérielle une géométrie spécifique. Dans le cas d'une pièce de géométrie complexe, cette géométrie impose la construction d'une application ${}^0\hat{\Phi}_{Pièce}$. Dans ce travail nous nous sommes limités au cas où ${}^0\hat{\Phi}_{Pièce}$ est une homothétie. Dans ce cas, la démarche complète permettant l'obtention de la surface usinée et la méthode pour assurer les passages entre les diverses configurations introduites ont été décrites.

Dans sa configuration matérielle, la partie usinée de la pièce est représentée par un ensemble de dexels tous parallèles entre eux. La transformation ${}^0\hat{\Phi}_{Pièce}$ permettant de passer

de cette configuration à la configuration relâchée de la pièce permet a priori de modéliser différentes opérations d'usinage sur des portions de pièces de formes plus ou moins complexes. Seule la définition de l'application bijective ${}^0\widehat{\Phi}_{Pièce}$ présente des difficultés si la pièce est de forme complexe et si l'étendue de la zone usinée est grande.

Les différentes configurations étant définies, il reste à présenter les deux modèles géométriques nécessaires pour décrire l'enlèvement de matière et pour obtenir la surface usinée (chapitre 3).

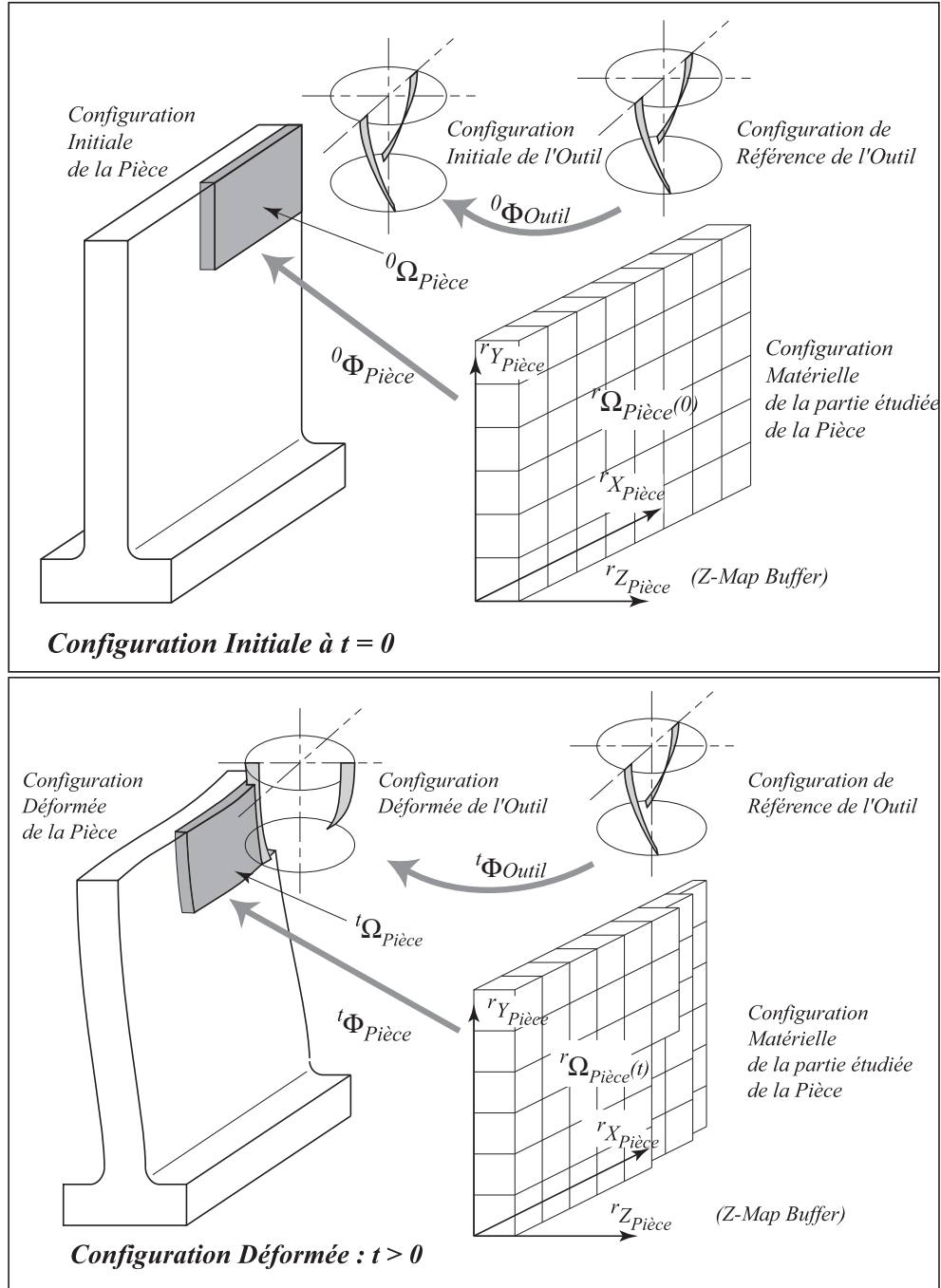


FIG. 2.15 – Configurations initiales et déformées de la pièce et de l'outil

Chapitre 3

Obtention de la surface usinée

L'obtention de la surface usinée passe par la description des modèles géométriques utilisés pour la pièce et pour les faces de coupe de l'outil, par la définition de l'interaction outil/pièce et par la mise place d'un algorithme pour traiter l'équation d'équilibre dynamique du système. Ce chapitre fait le point sur ces différents aspects en portant une attention particulière à l'introduction du nouveau modèle géométrique utilisant des dexels. Des détails pratiques sur la mise en place dans le logiciel *Nessy* sont également donnés.

Dans un premier temps, nous décrivons la modélisation géométrique de la pièce, en précisant la notion de dixel, et celle des domaines balayés par les faces de coupe de l'outil. Suit une description détaillée des algorithmes utilisés pour modéliser l'enlèvement de matière. Dans un second temps, nous précisons la méthode mise en place pour le calcul des efforts de coupe en liaison avec le modèle basé sur des dexels. Enfin la méthode de résolution de l'équation d'équilibre dynamique du système Pièce-Outil-Machine complet est présentée.

3.1 Modèles géométriques utilisés pour la description des volumes et des surfaces

Nous utilisons deux méthodes différentes pour représenter les domaines volumiques. Pour la pièce, le volume est discrétisé directement à l'aide de "dexels". Pour l'outil et les domaines balayés par ses faces de coupe, une approche de type B-Rep est retenue.

3.1.1 Description par discrétisation directe du domaine volumique (dixel)

Définition d'un Z-map simple : modélisation par projection sur une grille

Le principe de cette méthode est simple. La surface usinée est projetée sur une surface et la distance entre les deux est conservée en tous les points d'une grille appartenant à la surface de projection. Cette méthode est souvent appelée méthode du Z-buffer ou Z-map simple. Le cas le plus simple est celui où la surface de projection est un plan et où la projection se fait orthogonalement à ce plan (figure 3.1 à droite). Si la grille est du type rectangulaire, la mémorisation de la surface se fait à l'aide d'une simple matrice. La mise à jour de la surface est très simple. Il suffit de comparer localement les valeurs qui

correspondent à la surface et celles qui proviennent de la surface engendrée par l'arête de coupe lors du dernier pas de temps. Les calculs sont très rapides.

Cette méthode a été développée par Wang [WW86a, WW86b], Van Hook [Hoo86], Atherton et al [AEF87] et Menon et Robinson [MR92].

A chaque point d'une grille régulière de m lignes $\times$ n colonnes est associée une valeur z . Le tableau $z[m, n]$ contenant la valeur z est appelé Z-map simple. Comme le montre la figure 3.1, un Z-map est une représentation discrète de forme spéciale où les hauteurs des points de la grille sont stockées dans un tableau à deux dimensions.

Nous pouvons définir à chaque point de cette grille un support. Un support est défini comme une droite issue d'un point de la grille et parallèle à la direction de la projection.

La précision d'une méthode Z-map dépend de deux paramètres, d'une part du pas de la grille et d'autre part du choix de la direction de la projection. L'espacement entre les supports sur la surface de la pièce détermine la précision de la simulation.

Si la simplicité de la mémorisation de la surface, la rapidité de sa mise à jour et la rapidité du calcul des conditions de travail qui en découlent sont les grands avantages de cette méthode, celle-ci a, en contrepartie, des restrictions d'emplois nombreuses qui concernent la forme de la surface. La plus contraignante est, qu'à un point de la grille, ne doit correspondre qu'un point de la surface.

Définition d'un Z-map à plusieurs niveaux

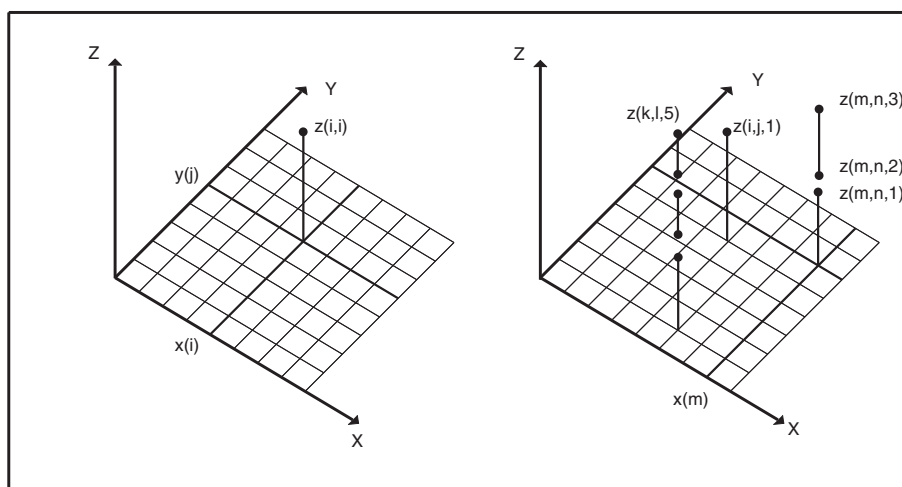


FIG. 3.1 – Modèles Z-map simple et à plusieurs niveaux

Pour remédier à ce point bloquant, Wang et Wang [WW86a] ont développé une méthode qui permet à chaque point de la grille de lui faire correspondre plusieurs valeurs, c'est une « extension » de la méthode de Z-map simple que l'on nommera un Z-map à plusieurs niveaux (figure 3.1 à gauche). Cela permet à cette méthode d'être utilisée, par exemple, lors d'une simulation 5-axes et de représenter des pièces avec par exemple des trous où il peut souvent y avoir plus d'une valeur de Z à conserver sur un même support.

Nous nous intéressons au cas le plus simple où la surface de projection est un plan, par exemple le plan (X, Y) et où la projection se fait orthogonalement à ce plan, par exemple l'axe Z .

Avec la méthode de Z-map à plusieurs niveaux, à chaque point d'une grille régulière de m lignes $\times$ n colonnes est associée une valeur z et un entier k qui indique la position du point sur le support (m, n) . La hauteur de chaque point est conservée dans un tableau à trois dimensions : $z[m, n, k]$ (figure 3.1 à gauche).

Définition d'un dixel

Basé sur des conventions d'appellation autour du pixel, avec par exemple le *voxel* pour l'élément de volume, Van Hook [Hoo86] a développé un modèle de Z-map à plusieurs niveaux où chaque élément de profondeur est un *dexel*. Chaque dixel peut être représenté comme un solide à base rectangulaire disposé parallèlement à l'axe des Z (figure 3.2).

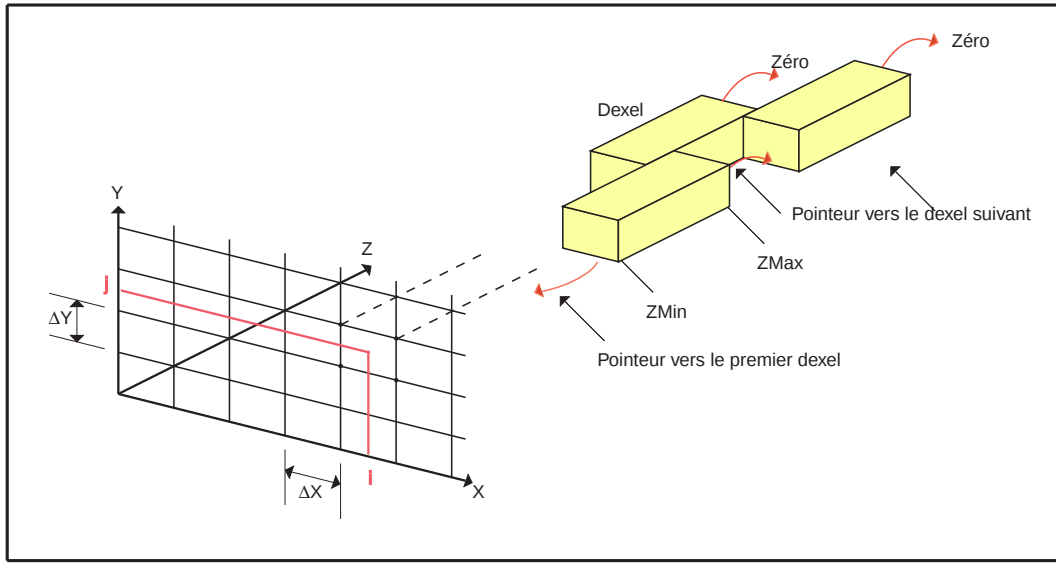


FIG. 3.2 – Visualisation des dexels en solide rectangulaire

La première fonction d'un dixel est de représenter un modèle volumique de forme parallélépipède rectangle. Pour les représentations simplifiées, un dixel peut être modélisé par un segment droit parallèle à l'axe des Z , appartenant au support issu du point (i, j) et tel que $z \in [zMin, zMax]$ (figure 3.3).

Un dixel est défini dans l'espace par les coordonnées de son support dans le plan (X, Y) et par son étendue suivant l'axe des Z : $[zMin, zMax]$.

A partir du couple (i, j) définissant un support, il est immédiat d'avoir la position du support du dixel dans le plan (X_{Grille}, Y_{Grille}) puisque la grille est régulière. Pour cela il suffit de connaître les pas de cette grille, ΔX suivant l'axe des X_{Grille} et ΔY suivant l'axe des Y_{Grille} ainsi que l'origine de la grille (X_0, Y_0) . Les $nbMaxSupportX$ et $nbMaxSupportY$ représentent le nombre maximal de supports suivant chaque direction.

$$\forall i \in [1, nbMaxSupportX], \quad X = X_0 + (i - \frac{1}{2}) * \Delta X \quad (3.1)$$

$$\forall j \in [1, nbMaxSupportY], \quad Y = Y_0 + (j - \frac{1}{2}) * \Delta Y \quad (3.2)$$

La définition précédente des dexels permet de modéliser un domaine volumique quelconque à partir de la description d'un ensemble de dexels. Pour des raisons de cohérence

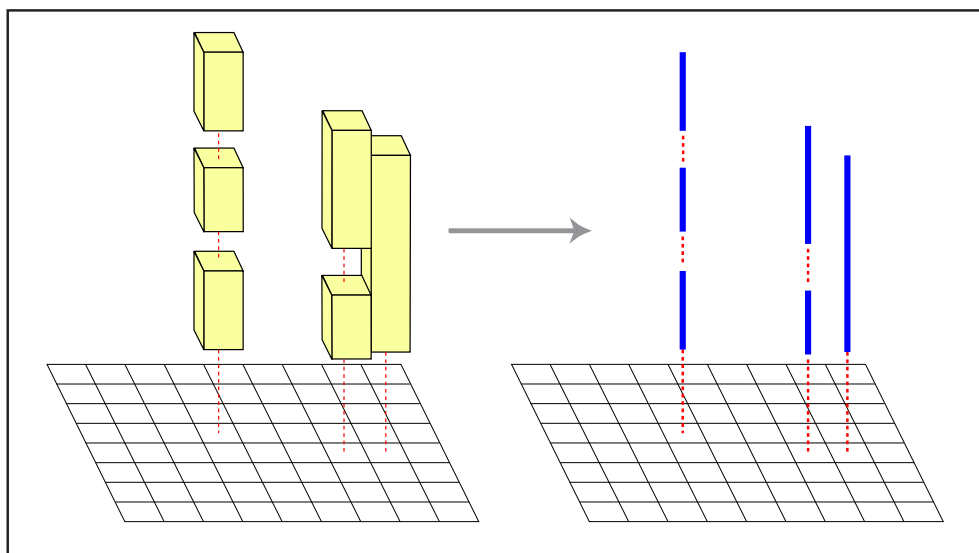


FIG. 3.3 – Représentation volumique et linéaire des dexels

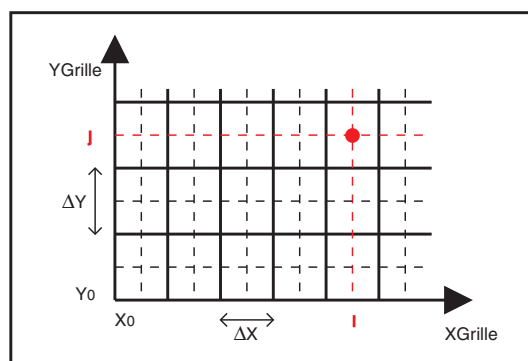


FIG. 3.4 – Représentation d'une grille régulière

évidentes, on interdit le chevauchement entre deux dexels appartenant à un même support.

Si $dexel_1 \in support(i, j)$ et $dexel_2 \in support(i, j)$ alors :

$$[zMin, zMax]_{dexel_1} \cap [zMin, zMax]_{dexel_2} = \emptyset \quad (3.3)$$

Mise en œuvre pratique

Afin de mettre en œuvre de façon pratique et efficace les algorithmes utilisant la notion de dixel, il est indispensable de rajouter des pointeurs ([Hoo86], figure 3.2) et la notion d'écart minimal entre dexels.

Pointeurs

Deux pointeurs sont destinés à connaître les positions relatives des dexels d'un même support. Avec deux pointeurs par dixel, il est possible de connaître parfaitement la position relative des dexels sur un support. Un des deux pointeurs du dixel permet de connaître le premier dixel du support auquel il appartient, ce pointeur sera noté *versPremier*. Le second pointeur, *versSuivant*, donne le dixel se trouvant juste au-dessus de lui sur le support. Si le dixel est le premier sur le support alors son pointeur *versPremier*

donnera le dixel lui même. Si le dixel est le dernier sur le support alors son pointeur *versSuivant* sera nul.

A partir du premier dixel se trouvant sur un support, il est facile de parcourir tous les dexels du support grâce au pointeur *versSuivant* qui permet de "remonter" le long du support.

Ecart minimal entre deux dexels

Pour éviter le chevauchement entre deux dexels d'un même support, nous imposons un écart minimal entre deux dexels appartenant à un même support. Cet écart est une marge de sécurité pour que deux dexels ne puissent pas se chevaucher à la suite des erreurs d'arrondi provenant des calculs numériques. Soit deux dexels, dex_{el_1} et dex_{el_2} , appartenant à un même support.

Si le dex_{el_1} se trouve en-dessous de dex_{el_2} alors :

$$(zMin)_{dex_{el_2}} - (zMax)_{dex_{el_1}} \geq \varepsilon_{dixel} \quad (3.4)$$

Mise en œuvre pratique dans *GesDyn*

Nous décrivons ici de façon succincte une partie des développements informatiques que nous avons réalisés au sein de *GesDyn* pour gérer les dexels.

Description des différents tableaux

D'un point de vue informatique, tous les premiers dexels d'un support (i, j) sont rangés dans une matrice *MatricePremierDixel* de dimension $nbMaxSupportX$ et $nbMaxSupportY$. Il est alors facile de connaître le premier dixel, *dexelEstLePremier*, d'un support (i, j) de la manière suivante :

$$MatricePremierDixel(i, j) = dexelEstLePremier \quad (3.5)$$

S'il n'y a pas de dixel sur un support (i, j) alors

$$MatricePremierDixel(i, j) = 0 \quad (3.6)$$

Il a été aussi introduit un tableau *DonneDixelSuivant* qui permet de donner tous les pointeurs *versSuivant* tel que :

$$DonneDixelSuivant(dixel) = dexelAprès \quad (3.7)$$

où *dixel* et *dixelAprès* appartiennent au même support (i, j) .

Si le dixel est le dernier sur le support alors :

$$DonneDixelSuivant(dexelEstLeDernier) = 0 \quad (3.8)$$

Il a été aussi nécessaire d'introduire une fonction permettant de connaître le dixel se trouvant juste avant une certaine valeur numérique z d'un support (i, j) . Cette fonction *DonneDixelJusteAvantZ* donne le *dixelAvant* se trouvant avant la valeur z donnée, elle est calculée grâce à la matrice *MatricePremierDixel* et au tableau *DonneDixelSuivant*.

$$DonneDixelJusteAvantZ(i, j, z) = dexelAvantZ \quad (3.9)$$

Les différentes étapes du calcul permettant de déterminer le dixel se trouvant juste avant une valeur z donnée sont présentées dans le pseudo-code A.1.

Création d'un dixel

Avec la représentation adoptée pour les dexels, la création d'un nouveau dixel correspond à une insertion dans la liste des dexels associés à son support. A chaque insertion d'un dixel dans cette liste, il faut actualiser les pointeurs des dexels d'un support entre eux.

Les pointeurs à actualiser sont :

- la matrice des premiers dexels sur un support (i, j) : *MatricePremierDixel*,
- le tableau donnant le dixel suivant sur un support (i, j) : *DonneDixelSuivant*.

En plus des pointeurs, à chaque fois qu'un dixel est créé, deux tableaux sont remplis, il s'agit de tableaux permettant de connaître soit le couple (i, j) soit l'étendue du dixel $[zMin, zMax]$ à partir du dixel. Ces tableaux sont indispensables pour déterminer la position et l'étendue d'un dixel donné.

$$IetJduDixel(1, dixel) = i \quad (3.10)$$

$$IetJduDixel(2, dixel) = j \quad (3.11)$$

$$EtendueZduDixel(1, dixel) = zMin \quad (3.12)$$

$$EtendueZduDixel(2, dixel) = zMax \quad (3.13)$$

Chaque dixel est mémorisé dans une liste par son numéro.

Les différentes étapes du calcul permettant de créer un dixel, connaissant son couple (i, j) ainsi que son étendue $[zMin, zMax]$, sont présentées dans le pseudo-code A.2.

Suppression provisoire d'un dixel

La suppression provisoire permet de remettre en place rapidement le domaine usiné dans un état où il se trouvait précédemment. Ceci est nécessaire pour mettre en place notre démarche itérative.

A chaque suppression d'un dixel sur un support, il faut modifier les pointeurs des autres dexels appartenant au même support. Le pseudo-algorithme A.3 montre le changement à apporter suivant la position du dixel sur le support c'est-à-dire suivant que le dixel se trouve :

- le seul sur le support,
- le premier sur le support,
- le dernier sur le support ou
- entre deux dexels.

Lors d'une suppression provisoire d'un dixel, ce dernier existe encore car son numéro n'a pas été «rendu » mais n'est plus utilisable.

Les différentes étapes du calcul permettant de supprimer provisoirement un dixel sont présentées dans le pseudo-code A.3.

Suppression définitive d'un dixel

Lorsque nous voulons supprimer de façon définitive un dixel (après convergence du schéma itératif), il faut effectuer les mêmes étapes que pour la suppression provisoire mais il faut dans ce cas « rendre » le numéro alloué du dixel et le retirer de la liste de tous les dexels.

3.1.2 Représentation des domaines volumiques par leur frontière (B-Rep) : description des surfaces

Description topologique

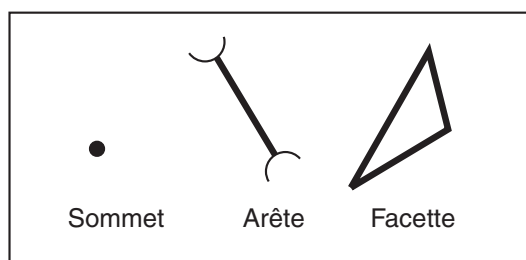


FIG. 3.5 – Les différentes entités

La description des surfaces qui est mise en place dans *GesDyn* est basée sur une représentation topologique. Les entités élémentaires, dans cette description sont les sommets, les arêtes et les facettes.

- les sommets sont des points,
- les arêtes sont des segments droits,
- les facettes sont des triangles plans.

Structure de données

Il existe des relations de connectivité ou d'adjacence entre les différentes entités. La description des surfaces est basée sur ces relations :

- un sommet est défini par ses coordonnées dans l'espace,
- une arête est définie par sa connexion à deux sommets,
- une facette est définie par sa connexion à trois arêtes.

D'un point de vue informatique, les données de description des surfaces sont regroupées dans trois principaux tableaux de la façon suivante :

- la liste des coordonnées des sommets : à un numéro de sommet, elle fait correspondre les coordonnées du sommet dans l'espace,
- la liste des extrémités des arêtes : à un numéro d'arête, elle fait correspondre la liste des numéros des 2 sommets auxquels elle est connectée (figure 3.6 à droite),
- la liste des arêtes des facettes : à un numéro de facette, elle fait correspondre la liste des numéros des 3 arêtes auxquelles elle est connectée (figure 3.6 à gauche).

En plus de ces trois tableaux, deux tableaux supplémentaires sont introduits dans *GesDyn* pour accéder rapidement aux entités topologiques de la structure de données. Il s'agit :

- de la liste des pointeurs arête vers facettes : à un numéro d'arête, elle fait correspondre les deux facettes qui lui sont connectées (figure 3.7 à droite),

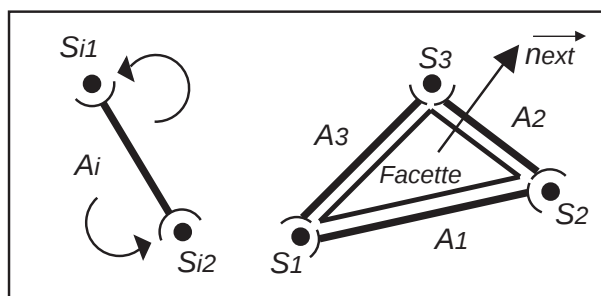


FIG. 3.6 – Entités décrivant le modèle géométrique des surfaces

- d’une liste de pointeurs sommet vers facette : à un numéro de sommet, elle fait correspondre une des facettes dont une des arêtes est connectée à ce sommet (figure 3.7 à gauche).

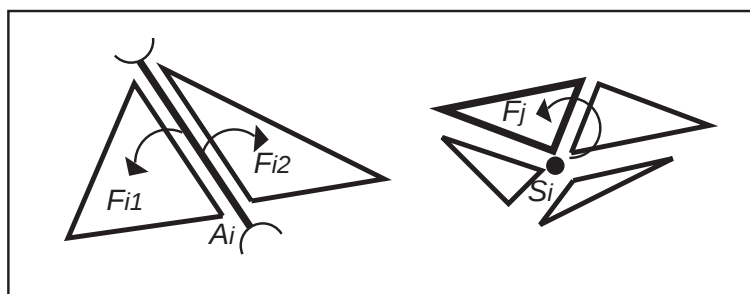


FIG. 3.7 – Structure complémentaire des données topologiques

Orientation des facettes et des arêtes

Il existe une orientation pour chaque facette ou arête. Cette orientation est cohérente avec l’orientation des facettes ou des arêtes adjacentes. Autrement dit, la liste des trois arêtes de chaque facette est *ordonnée*. Cet ordre est choisi de manière à ce que la normale définie selon le sens direct (ou trigonométrique) soit celle *extérieure à la matière*.

Les arêtes sont également orientées du premier sommet vers le second sommet auxquels elle est connectée.

Par convention et pour augmenter l’efficacité des algorithmes, les signes des numéros des arêtes auxquelles une facette est connectée, et mémorisés dans la liste des arêtes définissant les facettes, sont positifs si l’arête est orientée dans le sens de définition de la facette, négatif sinon.

Sur l’exemple donné figure 3.8, l’orientation de la facette **f** est définie par la liste ordonnée $\{b, c, a\}$, qui devient $\{b - c - a\}$ si on intègre la prise en compte de l’orientation des trois arêtes *a*, *b* et *c* vis-à-vis de l’orientation de la facette.

3.1.3 Présentation du module *GesDyn*

La structure de données associée à la description volumique et à la description topologique de surface a été mise en place dans le module de description et de gestion *GesDyn* des domaines volumiques et de leurs frontières. Ce module permet de gérer la géométrie dans un espace de référence défini par l’utilisateur [LAC00, CL03].

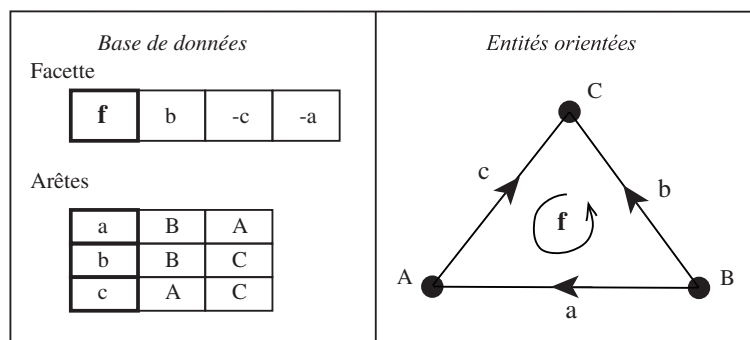


FIG. 3.8 – Orientation des entités et numérotation signée

Il gère donc les différentes entités géométriques décrites précédemment : les dexels, les sommets, les arêtes et les facettes.

L'accès à la structure de données du volume et de la surface se fait par l'intermédiaire de fonctions élémentaires permettant par exemple :

- la création d'une entité,
- l'élimination d'une entité du modèle (l'entité existe encore mais n'est plus utilisable),
- la suppression d'une entité,
- le changement des coordonnées d'un sommet par exemple ...

Par ailleurs, des fonctionnalités de tracé des différentes entités, pendant les calculs, sont disponibles et indispensables pour la mise au point.

3.2 Modèle géométrique de la pièce

La géométrie de la pièce doit être représentée par un modèle évolutif et suffisamment fin pour décrire l'évolution de la frontière du domaine pièce au fur et à mesure que l'enlèvement de matière, ou plus exactement ici l'effacement de matière, se produit.

La géométrie de la pièce est représentée par un modèle constitué de dexels. Nous cherchons à représenter la surface finale de la pièce usinée et ce modèle volumique permet d'obtenir une description suffisamment fine de la surface de la pièce. Il suffit pour cela d'avoir une densité de dexels assez importante pour obtenir la finesse recherchée de la surface de la pièce.

Les dexels ne sont pas disposés sur toute l'étendue du pavage $2D$ dans l'espace matériel rR . Ils sont disposés sur une partie rectangulaire du pavage : figure 3.9. Les dimensions sont indiquées en fraction des longueurs du pavage LX et LY suivant X et suivant Y dans cet espace. Ceci permet de disposer d'une densité de dexels assez importante uniquement à l'endroit où la pièce va être usinée.

Cette disposition de dexels sur le pavage pourra être évolutive suivant l'avancement de l'outil sur la pièce mais ceci n'est pas pris en compte pour le moment par *Nessy*.

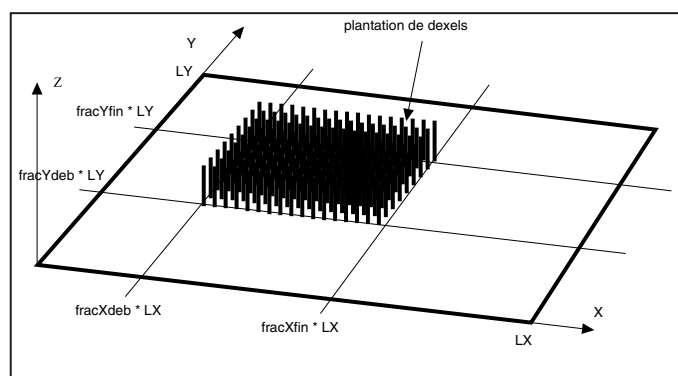


FIG. 3.9 – Portion du pavage où sont disposés les dexels

3.3 Modèles géométriques des parties coupantes de l'outil

3.3.1 Modèle géométrique de l'outil

Un outil, une fraise par exemple, est constitué d'un corps qui est placé dans la broche de la machine (en fraisage) ou dans le porte-outil (en tournage) et de parties actives qui sont les zones où l'outil entre en contact avec la pièce et retire de la matière. En tournage, une seule partie active apparaît alors qu'en fraisage plusieurs parties actives associées à plusieurs outils élémentaires, ou dents, sont présentes.

Les parties actives de l'outil sont composées des faces de coupe et des faces en dépouille. Les faces en dépouille sont adjacentes aux faces de coupe mais ne sont pas des parties coupantes de l'outil.

Pour le moment, seules les faces de coupe de l'outil sont prises en comptes dans le modèle géométrique de l'outil. Pour les faces en dépouille, le travail à effectuer serait le même que celui fait pour les faces de coupe.

Comme l'usure de l'outil n'est pas prise en compte, le domaine matériel occupé par l'outil ne change pas et il en est de même de toute sa frontière. En particulier la géométrie de chaque face de coupe de l'outil est inchangée .

Nous simplifions la géométrie en négligeant entre autre le rayon du bec de l'outil.

Les faces de coupe de l'outil sont modélisées par une portion de modèle B-Rep polyédrique correspondant à la description du paragraphe (...). Une face de coupe est donc constituée d'un ensemble de facettes triangulaires planes, d'arêtes et de sommets.

Décomposition des faces de coupe de l'outil en parties

Afin de mettre en place un calcul efficace de l'enlèvement de matière et des efforts de coupe, les faces de coupe de l'outil sont découpées en plusieurs parties. Chaque *partie* peut être considérée comme un outil élémentaire. Ceci est d'ailleurs bien adapté au problème des outils avec inserts ou plaquettes, encore que chaque insert doive être, le cas échéant, décomposé en plusieurs parties.

Si la face de coupe et la face en dépouille sont prises en compte alors une *partie* est composée des listes suivantes :

- d'une liste de facettes décrivant une *partie* de la face de coupe,
- d'une liste de facettes décrivant une *partie* de la face en dépouille,
- d'une liste d'arêtes décrivant le contour de la partie,
- ainsi qu'une liste d'arête de coupe qui délimite la frontière entre la face de coupe et la face en dépouille.

Exemple d'une fraise hélicoïdale

Nous avons choisi de modéliser une fraise hélicoïdale. La création des faces de coupe de la fraise est réalisée grâce à un maillage automatique. Ce maillage a été introduit dans *Nessy*, il a été conçu pour être le plus modulable possible. Tous les paramètres géométriques de la fraise sont des données que l'utilisateur peut modifier. Il est donc possible de changer la longueur, le rayon ou l'angle d'hélice de la fraise...

Pour respecter la géométrie de la fraise, et en particulier la courbure de l'hélice, il est nécessaire pour représenter une *partie* d'une fraise de plus ou moins de facettes triangulaires planes. Ce nombre de facettes dépend du découpage d'une partie suivant sa longueur et son épaisseur. La figure 3.10 montre un exemple de découpage d'une partie de l'ensemble des faces de coupe de l'outil (l'angle d'hélice est nul sur cette figure).

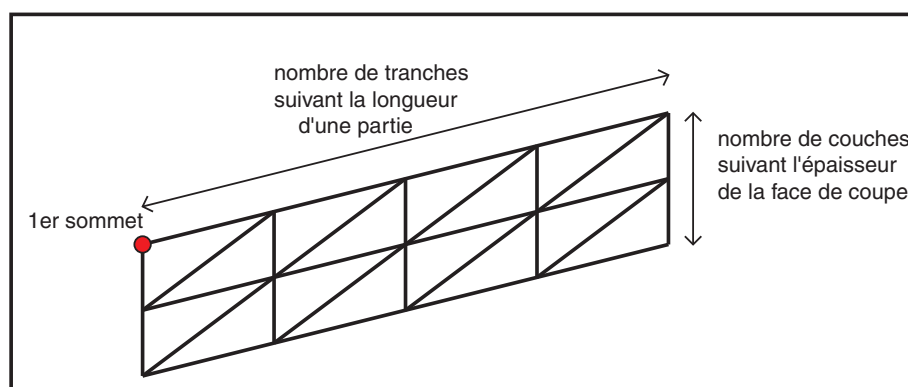


FIG. 3.10 – Maillage d'une partie de la face de coupe

La démarche adoptée pour réaliser le maillage régulier d'une partie i est la suivante :

- le calcul des coordonnées du premier sommet, d'après les données géométriques,
- le calcul des coordonnées des autres sommets, d'après la géométrie et le découpage,
- la création du premier sommet et des autres sommets,
- la création de toutes les arêtes (chaque arête étant connectée à deux sommets),
- la création de toutes les facettes (chaque facette étant connectée à trois arêtes).

Après avoir créé toutes ces entités, il faut en conserver la liste. Nous conservons la liste des facettes et la liste des arêtes du contour de chaque partie i de l'ensemble des faces de coupe.

Le nombre de facettes, $nbFacettePartie$, ainsi que le nombre d'arêtes du contour, $nbArêteCntrPartie$, par partie sont fonction des deux paramètres de découpage d'une partie, c'est-à-dire du nombre de tranches sur la longueur de la partie i , $nbTranche$, et du nombre de couches suivant l'épaisseur de la partie i , $nbCouche$. On a alors :

$$nbFacettePartie = 2 * nbTranche * nbCouche$$

$$nbArêteCntrPartie = 2 * (nbTranche + nbCouche)$$

La liste des arêtes de coupe d'une partie se déduit de la liste des arêtes du contour.

La géométrie et la topologie initiale d'une face de coupe d'une partie sont définies. On dispose alors d'un maillage fixe et régulier ainsi que d'une liste des différents éléments du modèle B-Rep (facettes triangulaires, arêtes et sommets) déterminée une fois pour toute.

3.3.2 Modèle géométrique du domaine volumique balayé par les faces de coupe de l'outil

Entre deux instants t et $t + \Delta t$, l'ensemble des points d'une partie i des faces de coupe de l'outil engendre un domaine volumique noté ${}^R V_{balayé}^i(t, t + \Delta t)$ dans le référentiel de référence matériel ${}^R R$.

Schématiquement, pour une partie i , la face avant du domaine balayé dans ${}^R R$ correspond à la face de coupe ${}^R \Gamma_{coupe}^i(t + \Delta t)$ à l'instant $t + \Delta t$. La face arrière du domaine balayé correspond à cette même face, mais telle qu'elle a été sauvegardée une fois l'équilibre atteint à l'instant t : ${}^R \Gamma_{coupe}^i(t)$. Une surface de liaison ${}^R \Gamma_{liaison}^i(t, t + \Delta t)$ est nécessaire entre ces deux faces de coupe pour compléter la frontière du domaine balayé. Cette frontière est fermée. ${}^R \Gamma_{liaison}^i(t, t + \Delta t)$ est, elle aussi, matérialisée par des facettes triangulaires, l'ensemble de ces faces de liaison est noté ${}^R \Gamma_{liaison}(t, t + \Delta t)$.

En utilisant l'exemple de la fraise hélicoïdale, la figure 3.11 montre les diverses phases de la construction du volume balayé ${}^R V_{balayé}^i(t, t + \Delta t)$ par une partie des faces de coupe, dans le référentiel ${}^R R$. Les faces de coupe ${}^R \Gamma_{coupe}^i(t)$ associées à chaque partie i à l'instant t sont représentées dans la partie gauche de la figure, les positions de ces mêmes faces de coupe aux instants t et $t + \Delta t$ sont montrées au milieu, et un exemple des domaines balayés par chaque partie de l'outil entre t et $t + \Delta t$ est donné dans la partie droite.

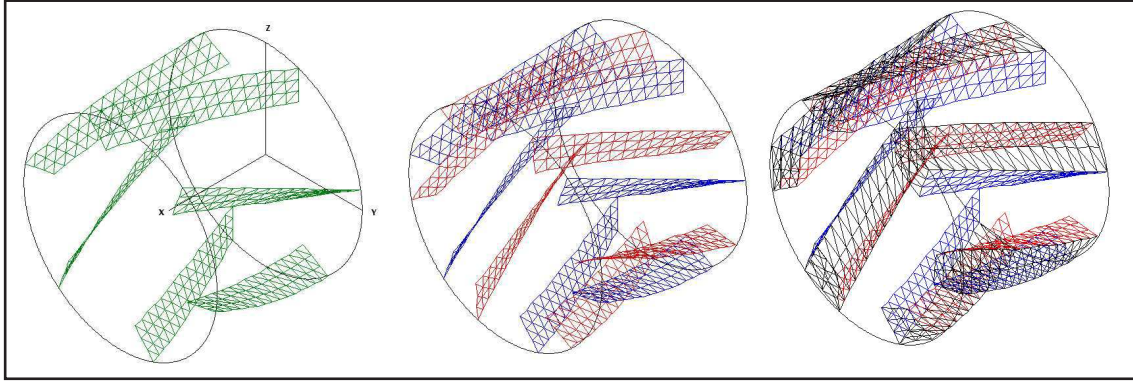


FIG. 3.11 – Différentes étapes de la construction du domaine balayé ${}^R V_{balayé}^i(t, t + \Delta t)$

Pour construire le domaine balayé, plusieurs étapes sont nécessaires, elles sont décrites ci-dessous. Nous nous appuyons sur l'exemple d'une fraise hélicoïdale, mais la démarche est complètement générale.

Etape initiale : topologie du domaine balayé pour une partie i de la face de coupe de l'outil

La première étape consiste à déterminer, une fois pour toute, la topologie du domaine balayé par chacune des parties i de la face de coupe de l'outil. Le domaine balayé par une

partie i est constitué de plusieurs listes de facettes :

- une liste de facettes représentant la face de coupe Γ_1 d'une partie i à l'instant t_1 , $ListeFacette\Gamma_1$,
- une liste de facettes représentant la face de coupe Γ_2 d'une partie i à l'instant $t_2 = t_1 + \Delta t$, $ListeFacette\Gamma_2$,
- une liste de facettes, $ListeFacette\Gamma_3$, représentant le domaine Γ_3 d'une partie i qui est la surface de liaison entre ces deux faces. Cette surface de liaison Γ_3 peut être décomposée en plusieurs couches, $nbCoucheLiaison$.

On peut représenter schématiquement, ces trois domaines surfaciques de la manière suivante :

- Γ_1 comme la face arrière,
- Γ_2 comme la face avant,
- Γ_3 comme le domaine de liaison entre Γ_1 et Γ_2 .

La géométrie et la topologie d'une partie i de la face de coupe de l'outil, Γ_1 , ont été définies dans le paragraphe précédent. Nous allons développer les différentes étapes permettant de déterminer la topologie du domaine balayé et donc la création des deux listes $ListeFacette\Gamma_2$ et $ListeFacette\Gamma_3$ à partir de $ListeFacette\Gamma_1$.

Topologie de la face de coupe d'une partie i à l'instant t_2

A partir de la liste des facettes $ListeFacette\Gamma_1$, de dimension $nbFacettePartie$, et des tableaux de connectivité, la construction de la liste des arêtes $ListeArête\Gamma_1$, de dimension $nbArêtePartie$ et de la liste des sommets $ListeSommet\Gamma_1$, de dimension $nbSommetPartie$ est immédiate.

$$nbFacettePartie = 2 * nbTranche * nbCouche \quad (3.14)$$

$$nbArêtePartie = 3 * nbTranche * nbCouche + nbTranche + nbCouche \quad (3.15)$$

$$nbSommetPartie = (nbTranche + 1) * (nbCouche + 1) \quad (3.16)$$

Nous disposons alors de toutes les listes définissant la topologie de la face de coupe à l'instant t_1 d'une partie i . Il faut maintenant déterminer la topologie de la face de coupe à l'instant t_2 de la partie i .

Pour cela, il suffit d'effectuer les étapes suivantes :

- la création des sommets de Γ_2 et le stockage de leurs numéros dans la liste $ListeSommet\Gamma_2$, de dimension $nbSommetPartie$,
- la création des arêtes de Γ_2 et le stockage de leurs numéros dans la liste $ListeArête\Gamma_2$, de dimension $nbArêtePartie$,
- la création des facettes de Γ_2 et le stockage de leurs numéros dans la liste $ListeFacette\Gamma_2$, de dimension $nbFacettePartie$.

La topologie de la face de coupe à l'instant t_2 d'une partie i est entièrement définie et fixée. Seules les coordonnées des sommets $Coor\Gamma_1$ et $Coor\Gamma_2$ seront calculées et modifiées au cours de la simulation.

Topologie du domaine de liaison entre les deux faces de coupe d'une partie i

Nous construisons le domaine volumique en créant une surface de liaison Γ_3 entre les deux faces de coupe, celle à l'instant t_1 et celle à l'instant t_2 . Cette surface est construite en

s'appuyant sur une approximation de la trajectoire, entre deux instants t_1 et $t_2 = t_1 + \Delta t$, des sommets des facettes de la face de coupe d'une partie i .

Dans ce travail, nous adoptons une interpolation linéaire mais il est possible de choisir une autre interpolation, ce qui est indispensable si le pas de temps Δt est important.

Pour créer cette surface de liaison, nous n'avons besoin que des contours des faces de coupe d'une partie. Un contour se limite à une liste d'arêtes. Dans notre approche, nous avons choisi de travailler avec un contour fermé.

Lorsque l'on construit le maillage d'une partie d'une face de coupe Γ_1 , la liste des arêtes du contour que l'on nommera *ListeArêteCntr* Γ_1 de dimension *nbArêteCntrPartie* est conservée. A partir de cette liste, il est aisé d'obtenir la liste des sommets du contour de Γ_1 , *ListeSommetCntr* Γ_1 grâce aux tables de connectivité. Comme le contour est fermé, le nombre d'arêtes est égal au nombre de sommets sur un contour :

$$nbSommetCntrPartie = nbArêteCntrPartie \quad (3.17)$$

La topologie des faces de coupe d'une partie i est la même pour Γ_1 et Γ_2 , seules les coordonnées des sommets sont différentes. Il en est donc de même pour la topologie de leur contour.

Afin de construire le domaine Γ_3 , il faut dans un premier temps, créer des contours intermédiaires. Le nombre de contours intermédiaires est directement lié au nombre de couches de liaison, *nbCoucheLiaison*, que l'on s'est fixé :

$$nbContourInter = nbCoucheLiaison - 1 \quad (3.18)$$

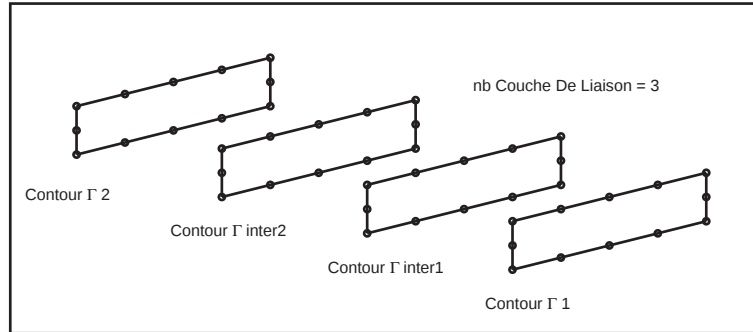


FIG. 3.12 – Exemple avec *nbCoucheLiaison* = 3

Pour déterminer la topologie des contours intermédiaires, il suffit d'effectuer les étapes suivantes :

- la création des sommets du contour intermédiaire j , où j est un indice variant de $[1, nbContourInter]$, de Γ_3 et le stockage de leurs numéros dans la liste *ListeSommetCntr* $Inter_j$ qui est de dimension *nbSommetCntrPartie*,
- la création des arêtes du contour intermédiaire j , où j est un indice variant de $[1, nbContourInter]$, de Γ_3 et le stockage de leurs numéros dans la liste *ListeArêteCntr* $Inter_j$ qui est de dimension *nbArêteCntrPartie*,

La figure 3.12 montre un exemple avec *nbCoucheLiaison* = 3.

La démarche à suivre maintenant est de relier ces différents contours pour former un domaine fermé constitué de facettes triangulaires planes. Il faut dans un premier temps créer des arêtes reliant les contours puis créer les facettes triangulaires planes (figure 3.13).

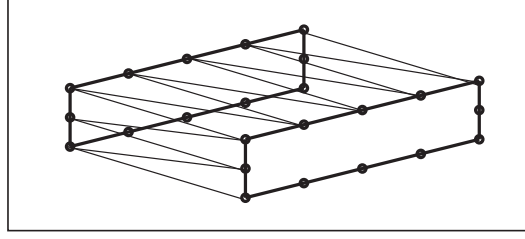


FIG. 3.13 – Maillage entre deux contours

Deuxième étape : actualisation de la géométrie du domaine balayé

A chaque début d'incrément, il faut actualiser la géométrie du domaine balayé en redéfinissant les coordonnées des sommets de Γ_1 et à chaque itération, il faut repositionner les sommets de Γ_2 et de Γ_3 . L'étape précédente, l'étape initiale, a permis de déterminer la topologie d'une partie i des faces de coupe de Γ_1 et de Γ_2 ainsi que la surface de liaison Γ_3 entre ces deux faces de coupe.

Nous allons développer maintenant les différentes étapes permettant de définir le domaine volumique balayé par les faces de coupe de l'outil à l'incrément $t_2 = t_1 + \Delta t$.

Construire le nouveau Γ_1 à partir de l'ancien Γ_2

A l'incrément précédent, à l'instant t_1 , lorsque l'équilibre est atteint, les coordonnées $Coor\Gamma_2$ des sommets de la face de coupe Γ_2 *avant* sont conservées.

Au nouvel incrément, à l'instant t_2 , il faut remplacer les coordonnées des sommets de la face de coupe Γ_1 par ceux de Γ_2 . Pour cela, les coordonnées de chaque sommet de Γ_2 sont associées au sommet homologue de Γ_1 .

Mettre le nouveau Γ_2 à la bonne position

Nous savons ramener à chaque instant, la face de coupe de l'outil dans le référentiel de référence matériel abstrait rR par l'application $(\phi_{Pièce}^+(\bullet, t))^{-1}$, où le mouvement des arêtes de coupe par rapport à la matière est parfaitement défini. Il faut alors ramener les coordonnées de la face de coupe à l'instant t_2 dans le référentiel rR avec l'application bijective $(\phi_{Pièce}^+(\bullet, t_2))^{-1}$. Ces nouvelles coordonnées devront être conservées pour l'incrément suivant.

Mettre Γ_3 à la bonne position

Les coordonnées des sommets de Γ_1 , $Coor\Gamma_1$, et de Γ_2 , $Coor\Gamma_2$, sont à ce stade définies dans le référentiel de référence matériel rR . Il reste à calculer les coordonnées de Γ_3 grâce à celles de Γ_1 et de Γ_2 . Pour chaque contour intermédiaire de Γ_3 , les coordonnées de chaque sommet de chaque contour j sont définies par la relation (symbolique) suivante :

$$CoorInter_j = Coor\Gamma_1 + j \frac{Coor\Gamma_2 - Coor\Gamma_1}{nbCoucheLiaison} \quad (3.19)$$

3.4 Intersection Pièce/Outil

3.4.1 Démarche générale

Comme nous l'avons déjà précisé, un des objectifs de ce travail est de pouvoir prédire l'état de la surface usinée ce qui nécessite de suivre son évolution tout au long de l'usinage de la pièce.

Dans le chapitre 2, nous avons défini comment l'outil *efface* la matière durant un pas de temps Δt . Le problème à résoudre est celui de l'opération booléenne d'intersection entre le domaine volumique balayé par les parties des faces de coupe de l'outil et la pièce modélisée par des dexels.

Nous nous plaçons ici dans le référentiel matériel rC où tous les dexels sont disposés suivant l'axe des Z sur une matrice régulière du plan (X, Y) .

La frontière de chaque partie du domaine balayé est décrite par un ensemble de facettes triangulaires planes. Nous devons alors détecter toutes les intersections entre les dexels et ces facettes triangulaires.

Nous allons dans un premier temps présenter la mise en œuvre du calcul d'intersection entre une facette triangulaire et un dixel. Puis il a été nécessaire d'introduire un code matière qui caractérise les extrémités des dexels.

Deux méthodes d'intersection ont été développées. La première méthode présentée à rencontrer quelques problèmes avec la présence de dexels *flottants*. Nous avons alors contourner ce problème par une deuxième méthode qui est actuellement utilisée.

3.4.2 Intersection dixel-facette

Le nombre de couples dixel-facette à traiter est limité par un système de tri : seules les zones où une intersection peut apparaître sont traitées.

Ce tri se base sur des domaines parallélépipédiques englobants (boîtes englobantes Alg.A.10). Ce tri est fait pour chaque volume balayé par partie ${}^rV_{balayé}^i$ et ensuite pour chaque facette.

Le calcul d'intersection effectué revient à réaliser l'intersection entre un plan et une droite parallèle à l'axe des Z . Le point trouvé est exprimé en coordonnées barycentriques pour la facette et en coordonnées réduites pour le dixel. Ce calcul s'effectue en utilisant comme inconnues les coordonnées barycentriques dans la facette et sur le dixel et nécessite la résolution du problème élémentaire suivant où M est le point d'intersection considéré :

Trouver $\lambda_1, \lambda_2, \lambda_3, \mu_1$ et μ_2 , tels que :

$$\overrightarrow{OM} = \lambda_1 \overrightarrow{OM_1} + \lambda_2 \overrightarrow{OM_2} + \lambda_3 \overrightarrow{OM_3} \quad (3.20)$$

et

$$\overrightarrow{OM} = \mu_1 \overrightarrow{OM_4} + \mu_2 \overrightarrow{OM_5} \quad (3.21)$$

où M_1, M_2, M_3 sont les sommets de la facette et M_4, M_5 sont les extrémités du dixel, en tenant compte des relations suivantes : $\lambda_1 + \lambda_2 + \lambda_3 = 1$ et $\mu_1 + \mu_2 = 1$

Si l'un des λ_i ou des μ_i est strictement supérieur à 1, alors le point se trouve à l'extérieur de la facette ou du dixel. Lorsque ce n'est pas le cas, nous obtenons le point d'intersection.

Ce nouveau point doit permettre d'effectuer la mise à jour de la description de la pièce c'est-à-dire de créer ou de supprimer des dexels. Deux méthodes ont été envisagées pour la mise à jour de la description de la pièce en dexels après une opération booléenne d'intersection.

3.4.3 Code matière

Un code matière a été introduit pour chaque dixel afin de traduire les "conditions limites" du dixel, c'est-à-dire si une extrémité d'un dixel est solidaire du reste du domaine pièce alors elle est dite *fixe* ou bien une extrémité est *libre*. Ce code est une donnée du problème pour la pièce à l'instant $t = 0$.

Chaque dixel possède un code matière pour ses deux extrémités. Ces deux codes extrémités permettent de définir quatre types de dexels : soit un dixel est *fixe* car ses deux extrémités sont *fixes*, soit un dixel est *fixe* à une seule de ses extrémité, l'autre étant *libre*, soit le dixel est complètement *libre*. La figure 3.14 illustre ces quatre cas, le trait horizontal à l'extrémité représente une extrémité *fixe* et la flèche une extrémité *libre*.

Si un dixel est *libre*, alors la matière qu'il représente peut disparaître partiellement ou entièrement après une intersection. Alors que pour un dixel qui est *fixe* à une ou deux de ses extrémités, la matière qu'il représente ne pourra jamais être supprimée entièrement. Une extrémité *fixe* ne pourra être modifiée au cours des algorithmes d'intersection : il restera toujours une portion de matière et un dixel.

Fixe	I	↑	↓
Libre	↕		

FIG. 3.14 – Différents codes matière d'un dixel

Si cela n'est pas le cas, cela entraînera un arrêt de la simulation et il faudra alors revoir la description du domaine de la pièce dont la taille est insuffisante (ou les vibrations sont exagérées). Le fait de vouloir usiner une extrémité correspond à l'usinage du dispositif de fixation de la pièce.

La figure 3.15 illustre, suivant le code matière des dexels, les modifications que pourra subir un dixel après une intersection.

- *Cas 1* : le dixel est *fixe* à ses deux extrémités. Si une intersection coupe le dixel au milieu alors ce dixel sera remplacé par deux dexels laissant une partie sans matière.
- *Cas 2* : le dixel est *fixe* à son extrémité supérieure, alors seule la partie inférieure du dixel peut être raccourcie après une intersection.
- *Cas 3* : le dixel est *fixe* à son extrémité inférieure, alors seule la partie supérieure du dixel peut être raccourcie après une intersection.

3.4.4 Première méthode d'intersection

La première idée pour effectuer l'intersection entre le modèle dixel et le volume balayé par une partie a été de traiter un par un chaque couple dixel-facette sans se soucier des

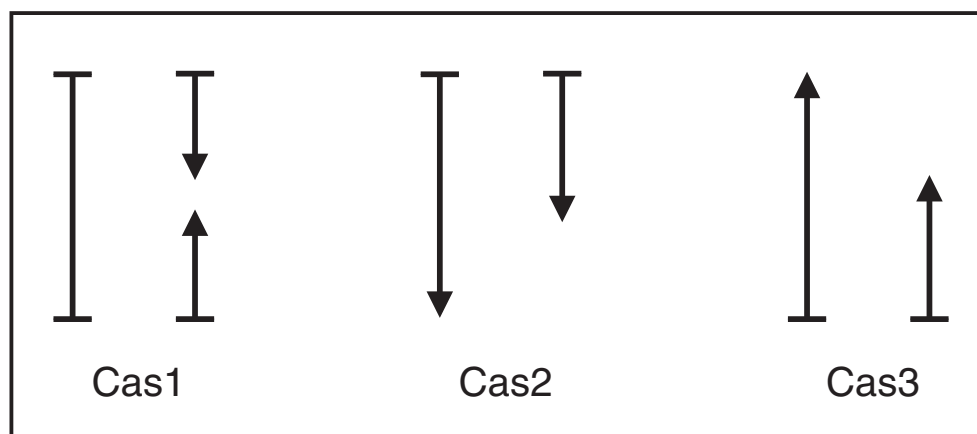


FIG. 3.15 – Représentation des trois modifications possibles d'un dixel après intersection

autres couples dixel-facette voisins et d'introduire un code matière supplémentaire pour les dexels, ce code matière dépendra de la normale extérieure de la facette.

Première étape : intersection dixel-facette

Pour un dixel, si une intersection a lieu avec une facette triangulaire, nous notons alors $zInter$ la valeur suivant l'axe des Z du point d'intersection. Chaque intersection est ainsi identifiée par la valeur $zInter$ et par le dixel intersecté noté *dixel père*. Ce *dixel père* appartient au support (i, j) , son étendue est $[zMin, zMax]$ et le dixel se trouvant juste au-dessus de lui sur le même support est le *dixelSuivant*. Le *dixelSuivant* peut être nul si le *dixel père* est le dernier sur le support.

La démarche à suivre est la suivante :

1. supprimer provisoirement le *dixel père*,
2. créer deux nouveaux dexels : *dixel fils du haut* et *dixel fils du bas*.

Les deux nouveaux dexels récupèrent les informations de leur *dixel père*. Ils appartiennent tous les deux au support (i, j) . Le *dixel fils du bas* a pour extrémité inférieure $zMin$ et pour extrémité supérieure $zInter - \varepsilon_{dixel}$ et le *dixel fils du haut* a pour extrémité inférieure $zInter + \varepsilon_{dixel}$ et pour extrémité supérieure $zMax$. Les informations sur les codes matières extrémités du *dixel père* sont aussi transmises aux deux dexels fils. Le code extrémité inférieure du *dixel fils du bas* est celui du *dixel père* et le code extrémité supérieure du *dixel fils du haut* est celui du *dixel père*.

	Extrémité inf	Extrémité sup	<i>versSuivant</i>
<i>dixel père</i>	$zMin$	$zMax$	<i>dixelSuivant</i>
<i>dixel fils du bas</i>	$zMin$	$zInter - \varepsilon_{dixel}$	<i>dixel fils du haut</i>
<i>dixel fils du haut</i>	$zInter + \varepsilon_{dixel}$	$zMax$	<i>dixelSuivant</i>

TAB. 3.1 – Passage entre le *dixel père* et les deux *dexels fils*

Les pointeurs sont aussi réactualisés, si le *dixel père* est le premier sur le support c'est-à-dire si :

$$MatricePremierDixel(i, j) = dixel\ père \quad (3.22)$$

il faut alors changer le terme de la matrice en (i, j) par :

$$\text{MatricePremierDexel}(i, j) = \text{dexel fils du bas} \quad (3.23)$$

Le pointeur *versSuivant* est modifié afin d'avoir le *dexel fils du haut* compris entre le *dexel fils du bas* et le *dexelSuivant*.

La création des deux nouveaux dexels assure bien la cohérence des pointeurs et le non-chevauchement. Entre les deux *dexels fils*, nous avons imposé un écart de $2 \varepsilon_{\text{dexel}}$ pour être sûr de ne plus détecter d'intersection avec les dexels du même support à la cote z_{Inter} .

Deuxième étape : mise à jour de la description de la pièce

Le domaine volumique balayé par une partie des faces de coupe de l'outil peut être considéré comme un domaine fermé. Nous avons besoin de savoir parmi les deux nouveaux dexels créés, *dexel fils du haut* et *dexel fils du bas*, celui qui appartient, complètement ou partiellement, à ce domaine fermé.

Avec les deux codes matière extrémités existants, définis précédemment, *fixe* ou *libre*, nous manquons de renseignements sur la "position" du dexel par rapport à la matière c'est-à-dire par rapport au volume balayé par une partie. Pour cela un troisième code matière extrémité est ici nécessaire pour les extrémités des dexels. Ce code va permettre de se repérer par rapport à la matière.

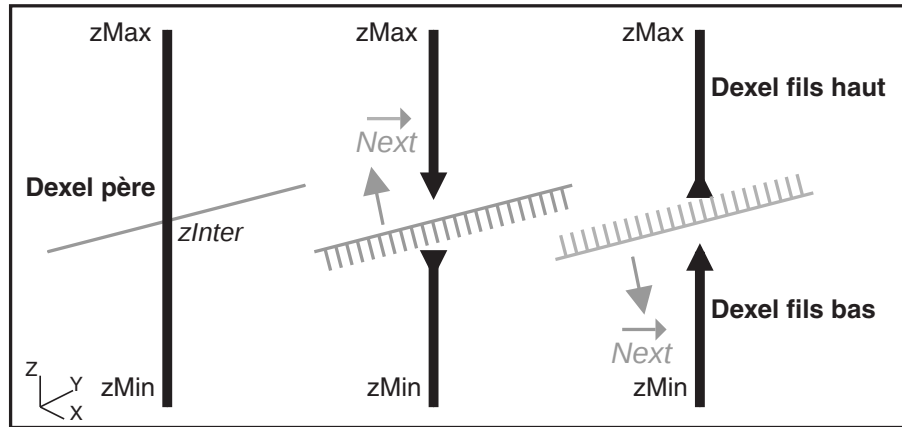
Le code matière extrémité *fixe* ne change pas. Le code matière extrémité *libre* et le troisième code vont être utiles pour définir le sens de la matière. L'un va pointer *vers la matière* et l'autre va se trouver *dans la matière*. Les trois codes extrémités sont représentés sur la figure 3.16 pour les deux extrémités d'un dexel.

	Fixe	VersMatière	DansMatière
Extrémité Supérieure			
Extrémité Inférieure			

FIG. 3.16 – Représentation des différents codes matière extrémités

L'orientation des facettes triangulaires (paragraphe 3.1.2) permet de déterminer la normale extérieure $\vec{N}_{ext}$ de chaque facette. Cette normale extérieure permet de savoir où se trouve "la matière". Suivant la direction de la normale extérieure, le code matière extrémité des deux dexels fils varie.

La figure 3.17 et le tableau 3.2 illustrent les deux cas possibles suivant l'orientation de la normale extérieure ou plus exactement suivant le signe du produit scalaire de la normale extérieure avec le vecteur unitaire de l'axe des Z , $\vec{e}_Z : (\vec{N}_{ext} \cdot \vec{e}_Z)$.

FIG. 3.17 – Code matière des *dexels fils* suivant la normale extérieure

	$\vec{N}_{ext} \cdot \vec{e}_z > 0$	$\vec{N}_{ext} \cdot \vec{e}_z < 0$
Extrémité supérieure du <i>dixel fils</i> du bas	<i>dans la matière</i>	<i>vers la matière</i>
Extrémité inférieure du <i>dixel fils</i> du haut	<i>vers la matière</i>	<i>dans la matière</i>

TAB. 3.2 – Code suivant la normale extérieure de la facette

Les deux nouveaux dexels fils sont complètement définis, nous connaissons leur position, c'est-à-dire leur support, leur étendue, leurs pointeurs vis-à-vis du premier dixel sur le support et du dixel suivant, ainsi que le code matière de leurs deux extrémités.

Lorsque toutes les intersections entre les facettes du domaine volumique d'une partie et les dexels sont traitées, il est possible suivant le code matière des deux extrémités d'un dixel de savoir parmi les nouveaux dexels créés ceux qui restent et ceux qui doivent disparaître.

La figure 3.18 rassemble les différents codes matière que peut prendre un dixel avec les trois codes matière extrémité :

- soit un dixel est *fixe* et ce dixel doit rester ;
- soit un dixel est *libre* et dans ce cas il faudra un traitement supplémentaire (une progression par le voisinage) pour déterminer si ce dixel reste ou s'il doit être supprimé ;
- soit un dixel a ses deux extrémités qui pointent *dans la matière* cela correspond à un dixel se trouvant à *l'intérieur* du volume balayé par une partie, ce dixel doit alors être supprimé.

Fixe	I	↑	↓
Libre	↕		
A supprimer	⌋	↑	⌋

FIG. 3.18 – Représentation des codes matières d'un dixel

Les différentes étapes du calcul d'intersection de cette première méthode sont présentées dans le pseudo-code A.4.

Défauts de la première méthode

Cette première méthode traite tous les dexels qui ont une intersection avec les facettes du volume balayé par une partie. Le problème rencontré par cette première méthode est qu'elle ne permet pas de prendre en compte facilement et efficacement les dexels se trouvant à l'intérieur du volume balayé par une partie. Ces dexels doivent normalement disparaître et la méthode pour les éliminer doit être rapide. Or dans cette approche, il n'est pas immédiat de considérer les dexels qui sont strictement à l'intérieur du volume balayé dès le début du calcul.

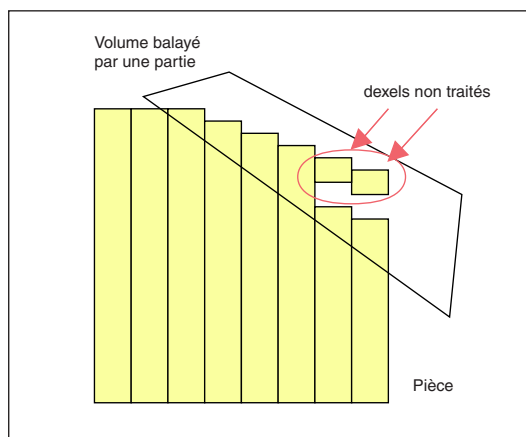


FIG. 3.19 – Dexels non traités par la première méthode d'intersection

3.4.5 Seconde méthode d'intersection

Afin de contourner les défauts de la première méthode, nous avons introduit une seconde méthode qui a pour but de déterminer et supprimer, partiellement ou complètement, tous les dexels se trouvant à l'intérieur du domaine volumique d'une partie. Cette détermination doit être rapide et efficace.

Première étape : intersection support-facette

Dans un premier temps, le problème se ramène à un calcul d'intersection entre un volume élémentaire et des segments de droite, le premier représente le domaine volumique balayé par une partie des faces de coupe de l'outil et le second les supports des dexels. Nous rappelons qu'un support est une droite parallèle à l'axe des Z et passant par un point (i, j) de la grille.

Une boîte englobante du domaine volumique d'une partie est calculée, cette boîte permet de faire un tri en déterminant l'ensemble des couples support (i, j) susceptibles d'avoir une intersection avec les facettes avec $i \in [iMin, iMax]$ et $j \in [jMin, jMax]$.

Il faut vérifier s'il existe une intersection entre un support (i, j) et toutes les facettes du domaine volumique d'une partie. Dès qu'une intersection est trouvée, la valeur de l'intersection suivant l'axe des Z , $zInter$, est conservée (stockée). Sur un même support,

plusieurs intersections peuvent être détectées, seule la plus petite valeur de z_{Inter} ainsi que la plus grande valeur de z_{Inter} seront conservées.

- la plus petite valeur de z_{Inter} sera stockée dans $minSupport$,
- la plus grande valeur de z_{Inter} sera stockée dans $maxSupport$.

Si une seule intersection a été trouvée sur un support alors $minSupport = maxSupport$.

A ce stade, deux valeurs $minSupport$ et $maxSupport$ ont été stockées pour chaque support (i, j) ayant eu une intersection avec au moins une facette du domaine volumique d'une partie.

Seconde étape : mise à jour de la description de la pièce

La deuxième étape consiste à déterminer quels sont les dexels compris entre les deux valeurs $minSupport$ et $maxSupport$ de chaque support. Ces dexels que nous nommerons les *dexels pères* devront être supprimés partiellement ou complètement car nous sommes sûr qu'ils appartiennent au moins en partie au domaine volumique d'une partie qui est considéré comme un domaine *convexe*.

Il est très facile grâce à la mise en œuvre des dexels, de connaître les dexels compris entre $minSupport$ et $maxSupport$ pour chaque support.

Les quatre cas possibles de positionnement d'un dixel d'étendue $[zMin, zMax]$ en fonction des deux valeurs $minSupport$ et $maxSupport$ sont présentés sur la figure 3.20 à droite et sur le tableau 3.3 à droite.

- Cas 1 : les deux valeurs $minSupport$ et $maxSupport$ coupent le *dixel père* en deux points,
- Cas 2 : seule la valeur $maxSupport$ coupe le *dixel père* en un point,
- Cas 3 : seule la valeur $minSupport$ coupe le *dixel père* en un point,
- Cas 4 : le *dixel père* n'est pas intersecté mais est compris entre les deux valeurs $minSupport$ et $maxSupport$.

Sur la figure 3.20, le dixel est représenté par un trait vertical, ses extrémités sont $zMin$ et $zMax$ et les deux valeurs $minSupport$ et $maxSupport$ sont représentées par deux traits horizontaux.

	Condition	Action
<i>Cas1</i>	$zMin \leq minSupport \leq maxSupport \leq zMax$	Création de deux nouveaux dexels
<i>Cas2</i>	$zMin \leq minSupport \leq zMax \leq maxSupport$	Création d'un nouveau dixel
<i>Cas3</i>	$minSupport \leq zMin \leq maxSupport \leq zMax$	Création d'un nouveau dixel
<i>Cas4</i>	$minSupport \leq zMin \leq zMax \leq maxSupport$	Rien

TAB. 3.3 – Mise à jour de la description de la pièce, les quatre cas possibles

Dès qu'un dixel se trouve dans l'un des quatre cas, le dixel est supprimé, il faut alors selon le cas créer un ou deux nouveaux dexels ou ne rien faire. Les conséquences pour le dixel intersecté, noté *dixel père*, appartenant au support (i, j) et d'étendue $[zMin, zMax]$ sont les suivantes :

- Cas 1 : le *dixel père* est intersecté en deux points, il sera remplacé par deux dexels, l'un d'étendue $[zMin, minSupport - \varepsilon]$ et le second d'étendue $[maxSupport +$

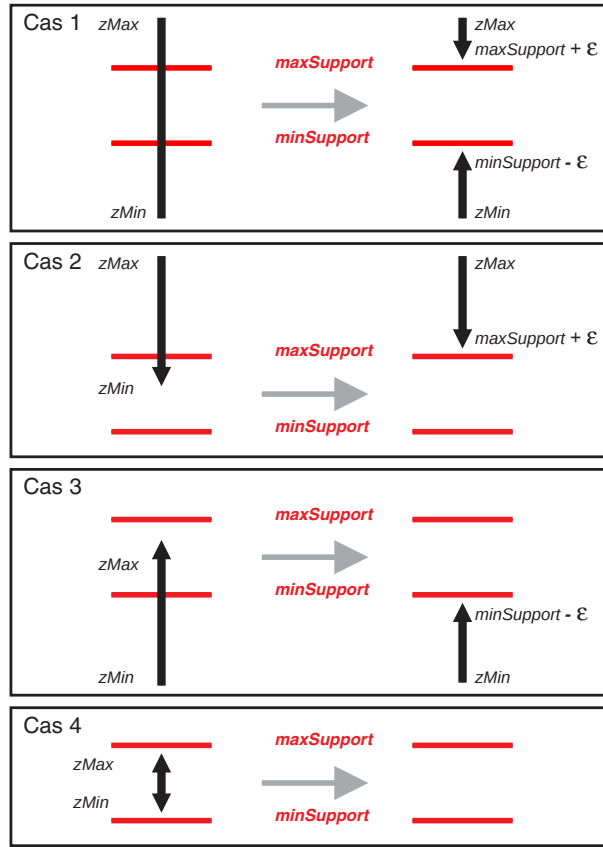


FIG. 3.20 – Mise à jour de la description de la pièce, les quatre cas possibles

$\epsilon, zMax]$ appartenant tous les deux au même support (i, j) . Le code matière des extrémités du *drexel père*, *fixe* ou *libre*, est transmis aux deux nouveaux dexels c'est-à-dire que le code de l'extrémité inférieure du nouveau dixel du bas est celui du *drexel père* ainsi que le code de l'extrémité supérieure du nouveau dixel du haut est celui du *drexel père*. Le code de l'extrémité supérieure du dixel du bas et celui de l'extrémité inférieure du dixel du haut sont quant à eux mis à *libres*.

- Cas 2 : le *drexel père* est intersecté en un point, en $maxSupport$, il sera remplacé par un dixel d'étendue $[maxSupport + \epsilon, zMax]$ appartenant au même support (i, j) . Ce nouveau dixel a pour code extrémité supérieure celui du *drexel père* et le code de l'extrémité inférieure est *libre*.
- Cas 3 : le *drexel père* est intersecté en un point, en $minSupport$, il sera remplacé par un dixel d'étendue $[zMin, minSupport - \epsilon]$ appartenant au même support (i, j) . Ce nouveau dixel a pour code extrémité inférieure celui du *drexel père* et le code de l'extrémité supérieure est *libre*.
- Cas 4 : le *drexel père* n'est pas intersecté, aucun dixel ne sera créé.

Il faut ensuite assurer la cohérence avec les pointeurs lorsqu'un dixel est créé ou supprimé (même provisoirement).

Cette méthode permet de traiter tous les dexels susceptibles d'intersecter le volume balayé par une partie. Nous avons donc au final deux catégories de dexels : des dexels *fixes* qui

restent et des dexels dits *libres*. Il faut à présent procéder à un traitement supplémentaire afin de savoir si les dexels *libres* doivent rester ou être supprimés.

Les différentes étapes du calcul d'intersection de cette deuxième méthode sont présentées dans le pseudo-code A.5.

3.4.6 Progression par le voisinage

Utilité de cet algorithme

L'algorithme d'intersection introduit la création de nouveaux dexels qui peuvent être *libres* ou *fixes*, d'après le code matière des dexels. Un problème se pose avec les dexels *libres* car ils peuvent être attachés à la matière ou pas. Quand les dexels *libres* ne sont pas attachés à la matière, les dexels sont dits *flottants*. Les dexels *flottants* doivent disparaître car ils n'ont pas lieu d'exister. On dit qu'un dixel est attaché à la matière quand un de ses voisins, au sens large, est *fixe*.

Si nous appliquons l'algorithme d'intersection dans l'exemple 3D de la figure 3.21 en supposant que les dexels sont encastrés à leur extrémité inférieure, nous observons qu'après avoir retiré la partie des dexels compris dans le domaine balayé par une partie, il reste en haut à droite des dexels *flottants*. Ces dexels doivent disparaître. Nous avons donc mis en place un programme qui doit prendre en compte ces dexels *flottants* pour les supprimer.

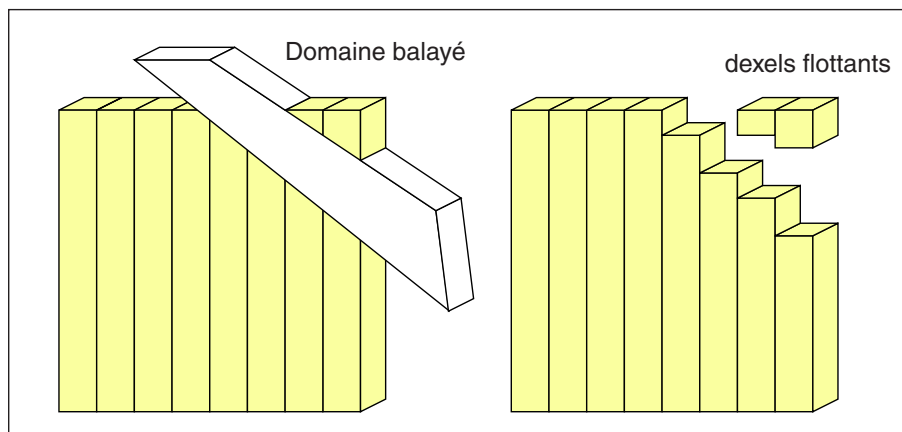


FIG. 3.21 – Exemple de dexels *flottants* en 3D

Nous allons dans un premier temps donner la définition du voisinage et ensuite développer les différentes étapes permettant la progression par le voisinage pour déterminer et supprimer les dexels libres *flottants*.

Définition du voisinage

Pour définir un dixel voisin, nous avons besoin de vérifier deux critères. Le premier est que le dixel voisin doit appartenir à un support dit voisin. Le deuxième repose sur l'étendue du dixel voisin.

Si le dixel considéré appartient au support (i, j) alors le dixel voisin est défini comme étant le dixel qui appartient à un des quatre supports voisins $(iVoisin, jVoisin)$ (figure 3.22) suivants :

- $(i - 1, j)$,
- $(i + 1, j)$,
- $(i, j - 1)$,
- $(i, j + 1)$.

Nous n'avons considéré que ces quatre supports dans la définition des dexels voisins.

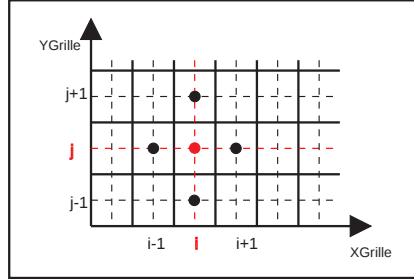


FIG. 3.22 – Représentation des supports voisins

Soit un dixel dex_1 d'étendue $[zMin, zMax]$ et un autre dixel dex_2 d'étendue $[zMinVoisin, zMaxVoisin]$, si l'intersection des étendues est non nul alors le dex_2 est un voisin du dex_1 et si et seulement si ils appartiennent à des supports dits voisins définis précédemment. Les quatre cas, illustrés sur la figure 3.23, sont les suivants :

- $zMinVoisin \leq zMin \leq zMax \leq zMaxVoisin$,
- $zMinVoisin \leq zMin \leq zMaxVoisin \leq zMax$,
- $zMin \leq zMinVoisin \leq zMax \leq zMaxVoisin$,
- $zMin \leq zMinVoisin \leq zMaxVoisin \leq zMax$.

Si $[(SupportVoisin) \text{ et } ([zMin, zMax]_1 \cap [zMin, zMax]_2 \neq \emptyset)]$

Alors les deux dexels dex_1 et dex_2 sont voisins

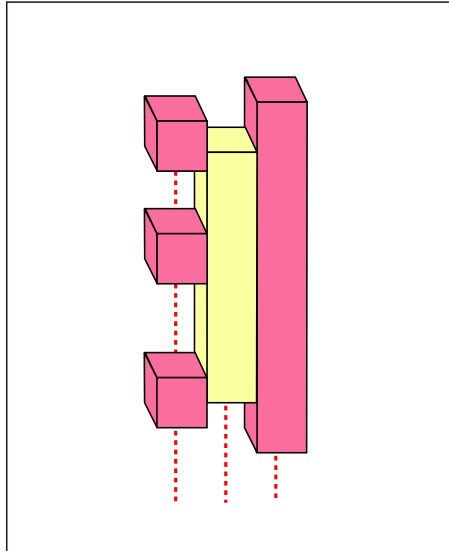


FIG. 3.23 – Représentation des différents dexels voisins

Les différentes étapes du calcul permettant de déterminer la liste des dexels voisins d'un dixel sont présentées dans le pseudo-code A.6.

Mise en œuvre de la progression par le voisinage

La première étape de ce programme consiste à déterminer parmi tous les dexels décrivant la partie de la pièce à usiner ceux qui sont *libres*. Ces dexels sont nommés *dexelLibre*, leurs numéros sont stockés dans la liste *ListeDexelLibre*.

Nous introduisons également un tableau de travail *EtatDexel* qui va permettre de suivre l'évolution de chaque *dexelLibre* au cours de cet algorithme. Pour chaque dexel, le tableau de travail *EtatDexel* peut prendre une des trois valeurs :

- *inconnu*
- *fixeNonTraité*
- *fixeTraité*

Tous les dexels sont initialisés à l'état *inconnu*.

Une première phase d'initialisation est présentée dans l'algorithme Alg.A.7, l'algorithme de progression par le voisinage est quant à lui détaillé dans Alg. A.8. La dernière phase de conclusion est présentée dans l'algorithme Alg.A.9.

Lorsque tous les dexels libres sont traités, il suffit de vérifier s'il reste des dexels *inconnus*. Si c'est le cas, ces dexels sont des dexels *flottants* et ils doivent être supprimés.

3.5 Calcul des efforts de coupe

Dans la solution adoptée dans ce travail, nous associons à chaque volume balayé par une partie i les trois entités suivantes :

1. Une loi de coupe qui permet de calculer la résultante $\overrightarrow{F_i^{coupe}}$ des efforts de coupe, pour chaque volume balayé par une partie i , à partir de certains paramètres de la coupe. La loi la plus simple consiste à définir dans un repère de mouvement connu chacune des composantes des efforts de coupe par une formule de type :

$$F_i^{coupe\ j} = k_j \frac{b_i}{h_0} \left(\frac{h_i}{h_0} \right)^{\alpha_j} \quad (3.24)$$

avec $F_i^{coupe\ j}$ la j -ème composante dans l'espace de F_i^{coupe} .

b_i la profondeur de passe du volume balayé par la partie i .

h_i l'épaisseur de coupe du volume balayé par la partie i .

h_0 l'épaisseur de coupe de référence.

k_j et α_j des constantes de coupe caractérisant cette loi.

2. Un point tM_i où s'applique l'effort de coupe $\overrightarrow{F_i^{coupe}}$ défini par la loi de coupe.
3. L'approche classique consiste à définir le vecteur force dans un repère local connu a priori qui lié à la configuration non déformée de l'outil.

Nous allons dans les paragraphes suivants déterminer les inconnues qui sont le point M_i où s'applique l'effort de coupe, le repère local ainsi que les paramètres de la loi de coupe.

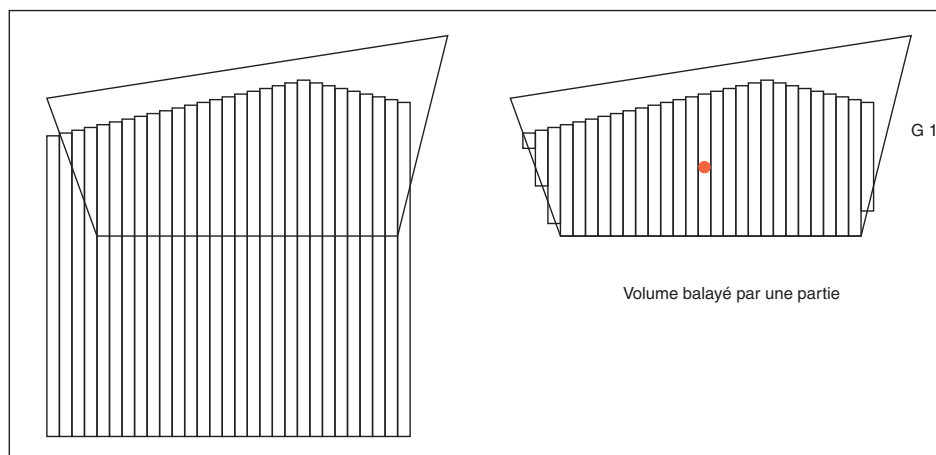


FIG. 3.24 – Représentation du volume balayé par une partie

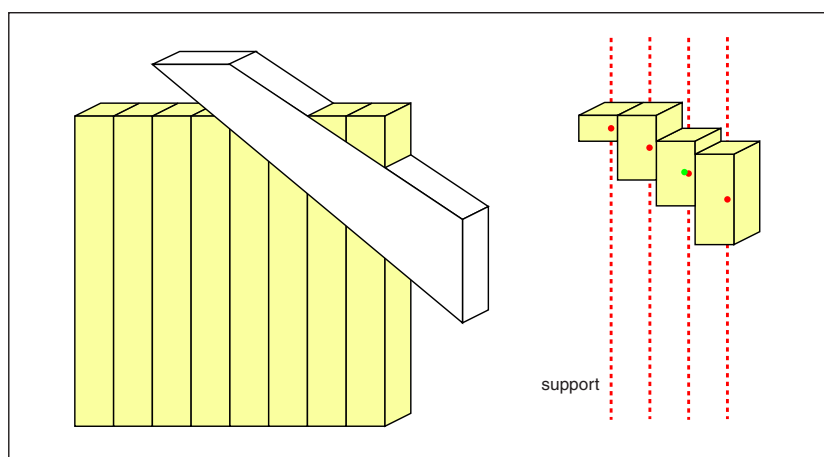


FIG. 3.25 – Représentation du calcul du barycentre de chaque support

3.5.1 Détermination du point M_i d'application d'effort

A partir de l'algorithme d'intersection, nous connaissons le volume de dexels compris dans le volume balayé par une partie qui est retiré (suivant les quatre cas décrits dans le paragraphe 3.4.5 sur la mise à jour de la pièce dans la seconde méthode d'intersection). Nous pouvons alors en déduire pour chaque volume balayé par une partie i , le barycentre de ce volume de dexels supprimés, nous le nommons G_i .

Le point G_i

Pour chaque support j , nous pouvons calculer la longueur du dixel à supprimer suivant les quatre cas décrits sur la figure 3.20 et le tableau 3.3. Ces différentes longueurs l_j sont regroupées dans le tableau 3.4 suivant les quatre cas. Sur un même support, plusieurs dexels peuvent être intersectés par le volume balayé par une partie, il faut alors cumuler toutes les longueurs coupées. L_j est donc la somme de toutes les longueurs coupées sur chaque support j de dixel.

$$L_j = \sum_{j=1} l_j \quad (3.25)$$

La base de chaque dixel est identique, elle est égale au produit des espacements entre chaque support et donc au produit des pas de la grille ΔX et ΔY . Le volume V_j se déduit immédiatement de la manière suivante :

$$V_j = L_j \cdot (\Delta X \cdot \Delta Y) \quad (3.26)$$

De même, il est possible de déterminer le barycentre de chaque dixel coupé lors de l'intersection avec le volume balayé par une partie. Pour chaque support j , le barycentre est le point G_j^s .

$$\overrightarrow{OG_j^s} = \begin{pmatrix} OGx_j^s \\ OGy_j^s \\ OGz_j^s \end{pmatrix} \quad (3.27)$$

OGx_j^s et OGy_j^s se déduisent des équations (3.1) et (3.2) qui permettent d'obtenir les coordonnées réelles à partir de l'indice du support.

Si sur un même support j , il y a plusieurs dexels coupés alors OGz_j^s dépend des différentes coordonnées suivant l'axe des Z des barycentres g_j de chaque dixel coupé. La coordonnée Z du barycentre de chaque dixel coupé, notée Ogz_j , est regroupée dans le tableau 3.4 suivant les quatre conditions mises dans le tableau 3.3.

$$OGz_j = \frac{\sum_{j=1} V_j \times Ogz_j}{\sum_{j=1} V_j} \quad (3.28)$$

	Longueur l_j	Ogz_j
<i>Cas1</i>	$maxSupport - minSupport$	$\frac{minSupport + maxSupport}{2}$
<i>Cas2</i>	$maxSupport - zMin$	$\frac{zMin + maxSupport}{2}$
<i>Cas3</i>	$zMax - minSupport$	$\frac{minSupport + zMax}{2}$
<i>Cas4</i>	$zMax - zMin$	$\frac{zMin + zMax}{2}$

TAB. 3.4 – Discrétisation, les quatre cas possibles

Le barycentre G_i du volume des dexels coupés pendant l'incrément de temps par chaque partie i est calculé grâce au volume V_j représentant le volume de dexels coupés par le support j et au point G_j barycentre du dixel coupé par le support j .

$$\overrightarrow{OG_i} = \frac{\sum_{j=1} V_j \times \overrightarrow{OG_j^s}}{\sum_{j=1} V_j} \quad (3.29)$$

Les coordonnées réduites du point G_i

Après l'obtention du barycentre G_i d'une partie i , nous avons besoin de "projeter" ce point sur la face de coupe et puis sur l'arête de coupe de la partie i . Le point Gf_i est le "projeté" de G_i sur la face de coupe de la partie i et le point Ga_i celui sur l'arête de coupe de la partie i .

Pour ce faire, l'utilisation d'une application qui permette de passer d'un hexaèdre quelconque de R^3 à un hexaèdre de référence de dimension $[-1, 1] * [-1, 1] * [-1, 1]$ a été retenue. L'hexaèdre quelconque représente le volume balayé par une partie et l'hexaèdre

élémentaire va permettre de déterminer les coordonnées réduites (ξ_1, ξ_2, ξ_3) du barycentre G_i dans la partie.

De la même façon que lors du passage d'une configuration déformée à une configuration matérielle, nous avons choisi d'utiliser la méthode de Newton qui fait intervenir une application linéaire tangente moyenne.

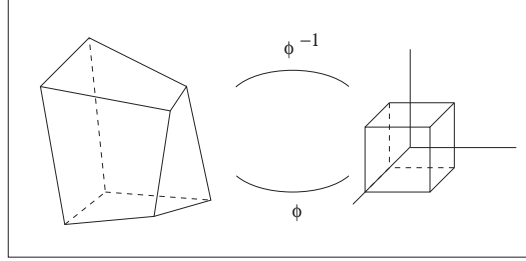


FIG. 3.26 – Représentation de l'application ϕ permettant de passer de la configuration matérielle à la configuration déformée

Les points Gf_i et Ga_i

Les coordonnées réduites du point Gf_i sont les mêmes que celles du point G_i mais une des coordonnées réduites est mise à -1 pour se trouver sur la face de coupe avant, Γ_2 . Dans notre cas, nous avons $\xi_3 = -1$, ce qui revient à passer dans un domaine de R^2 . De même, les coordonnées réduites du point Ga_i sont $(-1, \xi_2, -1)$.

Nous déterminons les coordonnées réelles des points Gf_i et Ga_i aisément grâce aux fonctions de base de l'élément hexaédrique à huit nœuds.

Le point Ga_i devient le point d'application de l'effort. Il pourra aussi être nommé dans la suite le point M_i .

L'arête de coupe où se trouve le point M_i

Nous cherchons à savoir sur quelle arête se trouve le point Ga_i . Pour cela, il faut chercher les deux sommets qui encadrent le point Ga_i . Nous nommerons ces deux points S_1 et S_2 . Grâce à la coordonnées réduite ξ_2 qui varie entre $[-1, 1]$ et au découpage d'une partie de face de coupe, c'est-à-dire du nombre de *tranches*, il est très rapide avec une division entière de retrouver les deux sommets S_1 et S_2 qui encadrent le point Ga_i et par la suite de connaître le numéro de l'arête de coupe où se trouve le point Ga_i .

3.5.2 Récupération des informations nécessaires à la mise en place des lois de coupe

Dans la mise en œuvre effectuée pour ce travail, nous nous sommes limités à la détermination de valeurs équivalentes h_i et b_i pour alimenter le modèle de la loi de coupe. D'autres approches pourront être testées plus tard.

La détermination des deux points Gf_i et Ga_i permet de déduire une épaisseur de coupe h_i pour chaque partie i :

$$h_i = 2\|Ga_iGf_i\| \quad (3.30)$$

Le vecteur unitaire $\vec{e}_{h_i}$ appartient à la face de coupe Γ_2 de la partie i :

$$\vec{e}_{h_i} = \frac{\overrightarrow{Ga_i G f_i}}{\|\overrightarrow{Ga_i G f_i}\|} \quad (3.31)$$

La largeur moyenne d'une partie peut être définie par la distance entre le point Ga_i qui est le "projeté" du point G_i sur la face de coupe Γ_2 et le "projeté" du point G_i sur la face de coupe Γ_1 que l'on nommera $Ga_i^{\Gamma_1}$. Cette distance correspond au pas de temps choisi. Les coordonnées du point $Ga_i^{\Gamma_1}$ se déduisent de la même manière que celles du point Ga_i avec pour coordonnées réduites de $Ga_i^{\Gamma_1}$ $(-1, \xi_2, +1)$.

$$l_i = \|\overrightarrow{Ga_i Ga_i^{\Gamma_1}}\| \quad (3.32)$$

Il est possible de définir un vecteur unitaire suivant la direction du vecteur $\overrightarrow{Ga_i Ga_i^{\Gamma_1}}$ dans la configuration matérielle rC :

$$\vec{e}_{l_i} = \frac{\overrightarrow{Ga_i Ga_i^{\Gamma_1}}}{l_i} \quad (3.33)$$

La largeur de coupe pour chaque partie est représentée par le scalaire b_i que nous déduisons du volume. Nous connaissons le vecteur unitaire de l'arête de coupe.

$$\vec{e}_{b_i} = \frac{\overrightarrow{S_1 S_2}}{\|\overrightarrow{S_1 S_2}\|} \quad (3.34)$$

Le volume V_i représente le volume de dexels compris dans le volume balayé par une partie i qui va être supprimé. Ce volume peut aussi être défini grâce au

$$V_i = \sum_{j=1} V_j = l_i \cdot b_i \cdot h_i \cdot \vec{e}_b (\vec{e}_l \wedge \vec{e}_h) \quad (3.35)$$

Nous posons :

$$c_i = \vec{e}_b (\vec{e}_l \wedge \vec{e}_h) \quad (3.36)$$

Nous avons donc la profondeur de passe b_i qui sera utile pour la loi de coupe :

$$b_i = \frac{V_i}{l_i \cdot h_i \cdot c_i} \quad (3.37)$$

3.5.3 Détermination du repère local associé au point tM_i dans la configuration actuelle tC

Dans l'approche que nous avons retenue, la loi de coupe est exprimée dans le repère mobile liée au mouvement réel de la face de coupe.

La loi de coupe est exprimée dans un repère spécifique dépendant du mouvement relatif outil-pièce. A chaque itération de chaque pas de temps, nous déterminons les trois axes du repère local défini sur chaque volume balayé par une partie et dans lequel sont calculées les composantes des efforts locaux $\overrightarrow{F_i^{coupe}}$. Nous définissons autant de repères que de points

d'intégration utilisés. Nous abordons à présent les différentes étapes de construction de l'un de ces repères.

Un vecteur, appelé "vecteur arête" et noté $\overrightarrow{e_{arête}}$, est défini dans la configuration actuelle tC . Pour un point d'intégration tM_i calculé à partir de son image M_i , ce vecteur est de direction identique à l'arête de coupe (supposée être un segment droit) sur laquelle se trouve le point d'intégration dans tC . Pour cela, il suffit de calculer les images des deux sommets extrémités de l'arête de coupe S_1 et S_2 dans la configuration actuelle, tS_1 et tS_2 , pour déterminer le vecteur unitaire $\overrightarrow{e_{arête}}$:

$$\overrightarrow{e_{arête}} = \frac{\overrightarrow{{}^tS_1 {}^tS_2}}{\|{}^tS_1 {}^tS_2\|} \quad (3.38)$$

Le deuxième vecteur considéré pour définir le repère local est la normale extérieure à la face de coupe, notée $\overrightarrow{n_{ext}}$, dans la configuration actuelle tC . L'arête de coupe appartient à deux facettes triangulaires. Une des facettes appartient à la liste des facettes de la face de coupe Γ_2 tandis que l'autre appartient au domaine englobant Γ_3 . La facette qui appartient à la *ListeFacette* Γ_2 sera nommée la facette de coupe.

Les trois sommets de la facette de coupe tS_1 , tS_2 et tS_3 dans la configuration actuelle déformée tC permettent de définir le vecteur unitaire normale à la facette de coupe $\overrightarrow{n_{ext}}$:

$$\overrightarrow{n_{ext}} = \frac{\overrightarrow{N_{ext}}}{\|\overrightarrow{N_{ext}}\|} \quad (3.39)$$

avec

$$\overrightarrow{N_{ext}} = \overrightarrow{{}^tS_1 {}^tS_2} \wedge \overrightarrow{{}^tS_1 {}^tS_3} \quad (3.40)$$

Le troisième vecteur de base, appelé "vecteur face" et noté $\overrightarrow{e_{face}}$, est défini de façon à compléter le repère local pour en faire un repère orthonormé direct de R^3 .

$$\overrightarrow{e_{face}} = \overrightarrow{e_{arête}} \wedge \overrightarrow{n_{ext}} \quad (3.41)$$

Nous définissons ainsi un repère local $\{{}^tM_i, \overrightarrow{e_{face}}, \overrightarrow{e_{arête}}, \overrightarrow{n_{ext}}\}$ lié au mouvement réel de l'outil et de la pièce pour chaque point d'intégration tM_i .

3.6 Algorithme implicite avec schéma de Newmark en dynamique non linéaire

L'équation d'équilibre EF à l'instant $t + \Delta t$ s'écrit, au niveau des DDL finaux $\{q_f\}$:

$$\left[\widehat{K}_{ff} \right] \cdot {}^{t+\Delta t} \{q_f\} = {}^{t+\Delta t} \{Q_f\} + {}^t \{\bar{\bar{Q}}_f\} \quad (3.42)$$

Ceci se réécrit aussi

$${}^{t+\Delta t} \{\psi_f\} = \{0\} \quad (3.43)$$

avec

$${}^{t+\Delta t} \{\psi_f\} \triangleq {}^{t+\Delta t} \{Q_f\} + {}^t \{\bar{\bar{Q}}_f\} - \left[\widehat{K}_{ff} \right] \cdot {}^{t+\Delta t} \{q_f\} \quad (3.44)$$

3.6.1 Cas particulier de l'usinage à l'échelle macroscopique

Dans le cas de l'usinage à l'échelle macroscopique, le terme ${}^{t+\Delta t}\{Q_f\}$ dépend en réalité de la valeur des DDL globaux ${}^{t+\Delta t}\{q_g\}$ dont une partie seulement ${}^{t+\Delta t}\{q_c\}$ est connue (mouvement d'avance et de rotation imposés), l'autre ${}^{t+\Delta t}\{q_i\}$ est inconnue, mais dépend uniquement de la partie ${}^{t+\Delta t}\{q_f\}$ des DDL conservés pour la résolution finale ce qui fait que l'équation est implicite en ${}^{t+\Delta t}\{q_f\}$.

On a :

$${}^{t+\Delta t}\{q_g\} \leftrightarrow \begin{Bmatrix} {}^{t+\Delta t}\{q_c\} \\ {}^{t+\Delta t}\{q_i\} \end{Bmatrix} \quad (3.45)$$

et par suite

$${}^{t+\Delta t}\{Q_f\} = \overline{Q}_f({}^{t+\Delta t}\{q_g\}) = Q_f(t + \Delta t, {}^{t+\Delta t}\{q_f\}) \quad (3.46)$$

3.6.2 Résolution itérative

Une résolution itérative doit donc être mise en place sur chaque incrément de façon à obtenir une solution précise. Cette résolution consiste à chercher ${}^{t+\Delta t}\{q_f\}$ tel que le résidu ${}^{t+\Delta t}\{\psi_f\}$ soit nul, ou plus exactement, compte tenu du caractère itératif de la démarche, de trouver ${}^{t+\Delta t}\{q_f\}$ tel que $\|{}^{t+\Delta t}\{\psi_f\}\|_{ad-hoc}$ soit suffisamment petite. La norme du résidu, $\|{}^{t+\Delta t}\{\psi_f\}\|_{ad-hoc}$ tient compte du caractère hétérogène des dimensions des composantes de $\{q_f\}$ (translations d'une part et rotations d'autre part).

Le schéma itératif consiste à rechercher une suite $\{q_f\}^{(i)} \triangleq {}^{t+\Delta t}\{q_f\}^{(i)}$ qui converge vers ${}^{t+\Delta t}\{q_f\}$ lorsque i croît.

Nous introduisons ainsi :

$$\{\delta_f\}^{(i)} \triangleq {}^{t+\Delta t}\{q_f\}^{(i)} - {}^{t+\Delta t}\{q_f\}^{(i-1)} \quad (3.47)$$

$$\{\Delta_f\}^{(i)} \triangleq \sum_{k=1}^i \{\delta_f\}^{(k)} \quad (3.48)$$

et

$$\{\Delta q_f\}^{(i)} = \{\delta_f\}^{(i)} + \{\Delta_f\}^{(i-1)} \quad (3.49)$$

en adoptant

$$\{\delta_f\}^{(0)} \triangleq \{0\} \quad (3.50)$$

$$\{\Delta q_f\}^{(0)} \triangleq \{0\} \quad (3.51)$$

par ailleurs, pour simplifier, nous notons aussi

$$\{q_f\}^{(0)} \triangleq {}^t\{q_f\} \quad (3.52)$$

$$\{q_f\}^{(i)} \triangleq {}^{t+\Delta t}\{q_f\}^{(i)} \quad (3.53)$$

$$\{Q_f\}^{(i)} \triangleq {}^{t+\Delta t}\{Q_f\}^{(i)} \quad (3.54)$$

$$\{\Psi_f\}^{(i)} \triangleq {}^{t+\Delta t}\{\Psi_f\}^{(i)} \quad (3.55)$$

de sorte que

$$\{q_f\}^{(i)} = \{q_f\}^{(0)} + \{\Delta_f\}^{(i)} \quad (3.56)$$

$$\{q_f\}^{(i)} = \{q_f\}^{(i-1)} + \{\delta_f\}^{(i)} \quad (3.57)$$

$$\{q_f\}^{(i)} = \{q_f\}^{(0)} + \{\Delta_f\}^{(i-1)} + \{\delta_f\}^{(i)} \quad (3.58)$$

et par suite (3.80) devient

$$\{\Psi_f\}^{(i)} = \{Q_f\}^{(i)} + {}^t\{\bar{\bar{Q}}_f\} - [\hat{K}_{ff}] \cdot \{q_f\}^{(i)} \quad (3.59)$$

autrement dit

$$\{\Psi_f\}^{(i)} = \{Q_f\}^{(i)} + {}^t\{\bar{\bar{Q}}_f\} - [\hat{K}_{ff}] \cdot \{q_f\}^{(i-1)} - [\hat{K}_{ff}] \cdot \{\delta_f\}^{(i)} \quad (3.60)$$

ou encore

$$\{\Psi_f\}^{(i)} = \{Q_f\}^{(i)} - \{Q_f\}^{(i-1)} + \{Q_f\}^{(i-1)} + {}^t\{\bar{\bar{Q}}_f\} - [\hat{K}_{ff}] \cdot \{q_f\}^{(i-1)} - [\hat{K}_{ff}] \cdot \{\delta_f\}^{(i)} \quad (3.61)$$

et, en introduisant

$$\{\Delta Q_f\}^{(i)} \triangleq \{Q_f\}^{(i)} - \{Q_f\}^{(i-1)} \quad (3.62)$$

il vient :

$$\{\Psi_f\}^{(i)} = \{\Delta Q_f\}^{(i)} + \{Q_f\}^{(i-1)} + {}^t\{\bar{\bar{Q}}_f\} - [\hat{K}_{ff}] \cdot \{q_f\}^{(i-1)} - [\hat{K}_{ff}] \cdot \{\delta_f\}^{(i)} \quad (3.63)$$

et en utilisant (3.60)

$$\{\Psi_f\}^{(i)} = \{\Delta Q_f\}^{(i)} + \{\Psi_f\}^{(i-1)} - [\hat{K}_{ff}] \cdot \{\delta_f\}^{(i)} \quad (3.64)$$

Dans la mesure où l'on cherche à obtenir $\{\Psi_f\}^{(i)} = \{0\}$, en supposant que $\{\Delta Q_f\}^{(i)}$ est petit, on résout :

$$[\hat{K}_{ff}] \cdot \{\delta_f\}^{(i)} = \{\Psi_f\}^{(i-1)} \quad (3.65)$$

de sorte qu'une fois (3.80) résolue, nous avons directement pour $i \geq 1$:

$$\{\Psi_f\}^{(i)} = \{\Delta Q_f\}^{(i)} \quad (si \quad i \geq 1) \quad (3.66)$$

3.6.3 Processus de résolution sur un incrément : début

En l'absence d'une réflexion plus approfondie, les premiers calculs à effectuer pour l'incrément peuvent se mettre sous la forme suivante :

- ★ Récupérer ${}^{t+\Delta t}\{q_c\}$ les DDL connus,
- ★ Calculer ${}^t\{\bar{\bar{Q}}_f\}$
- ★ Calculer $\{F_f\}$ avec

$$\{F_f\} \triangleq {}^t\{\bar{\bar{Q}}_f\} - [\hat{K}_{ff}] \cdot \{q_f\}^{(0)} \quad (3.67)$$

- ★ Calculer $\{q_i\}^{(0)} \triangleq {}^{t+\Delta t}\{q_i\}^{(0)}$ à partir de $\{q_f\}^{(0)}$, en déduire

$$\{q_g\}^{(0)} \leftrightarrow \left\{ \begin{array}{c} {}^{t+\Delta t}\{q_c\} \\ \{q_i\}^{(0)} \end{array} \right\} \quad (3.68)$$

- ★ Calcul d'intersection en utilisant $\{q_g\}^{(0)}$ puis déduction des efforts $\{Q_f\}^{(0)}$
- ★ Calculer

$$\{\tilde{\Psi}_f\}^{(0)} = \{Q_f\}^{(0)} + {}^t\{\bar{Q}_f\} - [\hat{K}_{ff}] \cdot \{q_f\}^{(0)} \quad (3.69)$$

autrement dit, calculer

$$\{\tilde{\Psi}_f\}^{(0)} = \{Q_f\}^{(0)} + \{F_f\} \quad (3.70)$$

- ★ Résoudre

$$[\hat{K}_{ff}] \cdot \{\delta_f\}^{(1)} = \{\tilde{\Psi}_f\}^{(0)} \quad (3.71)$$

En réalité, l'examen de cette procédure d'amorçage des itérations, peut être améliorée pour simplifier la programmation. En effet, dans le cas où l'outil n'est pas en contact avec la pièce, autrement dit, dans le cas où les efforts de coupe (ou de contact) ${}^{t+\Delta t}\{Q_f\} = \{0\}$, la solution ${}^{t+\Delta t}\{q_f\}$ pour l'incrément est donnée par la résolution de

$$[\hat{K}_{ff}] \cdot {}^{t+\Delta t}\{q_f\} = {}^t\{\bar{Q}_f\} \quad (3.72)$$

Si l'on note par convention,

$$\{\Psi_f\}^{(0)} \triangleq {}^t\{\bar{Q}_f\} - [\hat{K}_{ff}] \cdot {}^t\{q_f\} \quad (3.73)$$

autrement dit,

$$\{\Psi_f\}^{(0)} = \{F_f\} \quad (3.74)$$

On obtient ainsi immédiatement, avec les conventions retenues :

$$[\hat{K}_{ff}] \cdot \{\delta_f\}^{(1)} = \{\Psi_f\}^{(0)} \quad (3.75)$$

et :

$${}^{t+\Delta t}\{q_f\}^{(1)} = {}^t\{q_f\} + \{\delta_f\}^{(1)} = \{q_f\}^{(0)} + \{\delta_f\}^{(1)} \quad (3.76)$$

et par suite,

$$\{\Psi_f\}^{(1)} = \{Q_f\}^{(1)} + {}^t\{\bar{Q}_f\} - [\hat{K}_{ff}] \cdot \{q_f\}^{(0)} - [\hat{K}_{ff}] \cdot \{\delta_f\}^{(1)} \quad (3.77)$$

$$\{\Psi_f\}^{(1)} = \{Q_f\}^{(1)} \quad (3.78)$$

et dans le cas particulier où les efforts de coupe sont nuls, on a immédiatement $\{\Psi_f\}^{(1)} = \{0\}$ et donc la solution est obtenue dès la première résolution de système et c'est tout simplement ${}^{t+\Delta t}\{q_f\}^{(1)}$.

3.6.4 Processus de résolution sur un incrément : suite

Pour un incrément, l'itération courante étant notée i , la démarche à suivre est ainsi résumée par :

- Tant que $\|\{\Psi_f\}^{(i-1)}\| > \text{toler}$

- ★ Résoudre

$$[\hat{K}_{ff}] \cdot \{\delta_f\}^{(i)} = \{\Psi_f\}^{(i-1)} \quad (3.79)$$

★ Améliorer

$$\{q_f\}^{(i)} = \{q_f\}^{(i-1)} + \{\delta_f\}^{(i)} \quad (3.80)$$

★ Calculer $\{q_i\}^{(i)} \triangleq {}^{t+\Delta t}\{q_i\}^i$ à partir de $\{q_f\}^{(i)}$, en déduire

$$\{q_g\}^i \leftrightarrow \left\{ \begin{array}{c} {}^{t+\Delta t}\{q_c\} \\ \{q_i\}^{(i)} \end{array} \right\} \quad (3.81)$$

★ Calcul d'intersection en utilisant $\{q_g\}^i$ puis déduction des efforts $\{Q_f\}^{(i)}$

★ Calculer

$$\{\Psi_f\}^{(i)} = \{\Delta Q_f\}^{(i)} \triangleq \{Q_f\}^{(i)} - \{Q_f\}^{(i-1)} \quad (3.82)$$

★ Passer à l'itération suivante : $i := i + 1$

3.6.5 Intégration de la première itération dans la boucle tant que

Si l'on applique la démarche du paragraphe 3.6.4 pour la première itération, en tenant compte des observations faites à la fin du paragraphe 3.6.3, il suffit d'initialiser le processus en adoptant :

$$\{Q_f\}^{(0)} \triangleq \{0\} \quad (3.83)$$

La première itération se particularise de la façon suivante :

○ a priori, $\|\{\Psi_f\}^{(0)}\| > \text{toler}$

★ Résoudre

$$\left[\widehat{K}_{ff} \right] \cdot \{\delta_f\}^{(1)} = \{\Psi_f\}^{(0)} \quad (3.84)$$

★ Améliorer

$$\{q_f\}^{(1)} = \{q_f\}^{(0)} + \{\delta_f\}^{(1)} \quad (3.85)$$

★ Calculer $\{q_i\}^{(1)} = {}^{t+\Delta t}\{q_i\}^{(1)}$ à partir de $\{q_f\}^{(1)}$, en déduire

$$\{q_g\}^{(1)} \leftrightarrow \left\{ \begin{array}{c} {}^{t+\Delta t}\{q_c\} \\ \{q_i\}^{(1)} \end{array} \right\} \quad (3.86)$$

★ Calcul d'intersection en utilisant $\{q_g\}^{(1)}$ puis déduction des efforts $\{Q_f\}^{(1)}$

★ Calculer

$$\{\Psi_f\}^{(1)} = \{\Delta Q_f\}^{(1)} = \{Q_f\}^{(1)} - \{Q_f\}^{(0)} = \{Q_f\}^{(1)} \quad (3.87)$$

★ Passer à l'itération suivante : $i := i + 1$

3.7 Conclusion

Dans ce chapitre, nous avons décrit les deux modèles géométriques nécessaires pour obtenir la surface usinée, l'un pour décrire la forme du domaine balayé par les faces de coupe de l'outil l'autre pour représenter la surface de la pièce qui évolue au cours de l'usinage. La démarche pratique de mise en place de ces deux modèles géométriques dans le logiciel *Nessy* a été détaillée.

Les répercussions du choix d'un modèle géométrique s'appuyant sur des dexels ont été détaillées.

Elles ont conduit à une description des algorithmes utilisés pour modéliser l'enlèvement de matière. Ces algorithmes se sont révélés être, comme attendu, d'une grande robustesse. De plus, le choix d'un modèle basé sur des dexels a permis une autre méthode de calcul des efforts de coupe qui se base désormais sur la détermination d'une section moyenne de coupe dont le calcul est aussi très robuste. Cette section équivalente moyenne varie de façon continue, ce qui la rend particulièrement bien adaptée aux résolutions numériques pas à pas que nous utilisons.

Enfin, la méthode retenue pour la résolution de l'équation d'équilibre dynamique du système Pièce-Outil-Machine a été décrite.

Chapitre 4

Validation et exemples

Dans ce chapitre, il s'agit, au travers d'exemples, de valider l'approche proposée puis de montrer son potentiel sur un exemple de pièce flexible.

Pour le premier point, une grande partie de la validation se fait grâce à une comparaison entre des simulations réalisées avec *Nessy*, logiciel au sein duquel nous avons réalisé nos travaux de programmation, et des simulations réalisées par Grégoire Peigné pour sa thèse [Pei04].

Nous commençons ce chapitre par l'exposé de la démarche de choix des paramètres propres à nos simulations. Nous présentons ensuite des exemples selon un ordre de complexité croissante. Le premier a pour support une pièce rigide et fixe usinée par une fraise ayant, en plus de sa rotation propre, un simple mouvement rectiligne. Il s'agit d'un exemple purement géométrique.

Pour le second exemple, seul le mouvement de la fraise se complique un peu. En plus de la translation précédente, est appliqué un mouvement d'oscillation afin de se rapprocher d'un cas réel d'usinage avec vibrations. Pour l'exemple suivant, la pièce possède un degré de liberté de translation, une masse propre, et est liée au bâti par l'intermédiaire d'une liaison raideur/amortisseur. Pour cet exemple deux simulations sont effectuées et commentées : une dite *stable* où le mouvement de la pièce se stabilise autour d'un mouvement périodique et une dite *instable* où le mouvement de la pièce est plus complexe. Enfin, un dernier exemple traite un cas de pièce réellement flexible pour lequel, au fur et à mesure de l'avancement de la fraise, le comportement dynamique de la pièce évolue suite notamment au déplacement du point d'application des efforts de coupe.

Nous tenons à préciser qu'aucune des simulations n'est comparée avec un usinage réel. Cet aspect fait partie des perspectives évoquées en conclusion.

4.1 Choix des paramètres pour les simulations

Pour réaliser une simulation il faut, après avoir identifié les caractéristiques de l'exemple à traiter, choisir un pas de temps pour le schéma d'intégration temporel et un nombre de dexels pour décrire la géométrie de la zone usinée de la pièce.

Comme nous allons le voir ces choix ne sont pas arbitraires et peuvent être liés l'un l'autre.

4.1.1 Choix du pas de temps

Il faut, dans un premier temps, se fixer la fréquence la plus élevée que l'on souhaite observer au travers de la simulation. Ceci revient à prendre en compte les caractéristiques dynamiques du système POM. Nous notons f_{obs} la fréquence choisie. Dans le cas d'une pièce flexible, f_{obs} est typiquement une des premières fréquences propres de la pièce si les modes associés ont une influence directe sur la forme de la pièce dans la zone usinée. Si la broche, ou toute autre partie de la machine, est également flexible le choix de f_{obs} doit en tenir compte. Sachant que pour pouvoir observer correctement un cycle il faut environ 10 pas de temps on obtient une estimation de la valeur maximale pour le pas de temps donnée par :

$$\Delta t \leq \Delta t_{obs} \quad \text{avec} \quad \Delta t_{obs} = \frac{1}{10} \cdot \frac{1}{f_{obs}} \quad (4.1)$$

Dans un cas où f_{obs} est la première fréquence propre de la pièce, en choisissant un pas égal au dixième de la période de ce mode, dix avances incrémentales sont nécessaires à la dent pour que la pièce effectue un "aller et retour" (fig 4.1). Vue de la pointe de la dent, cela signifie que deux minimums consécutifs de matière enlevée sont atteints tous les dix pas de temps.

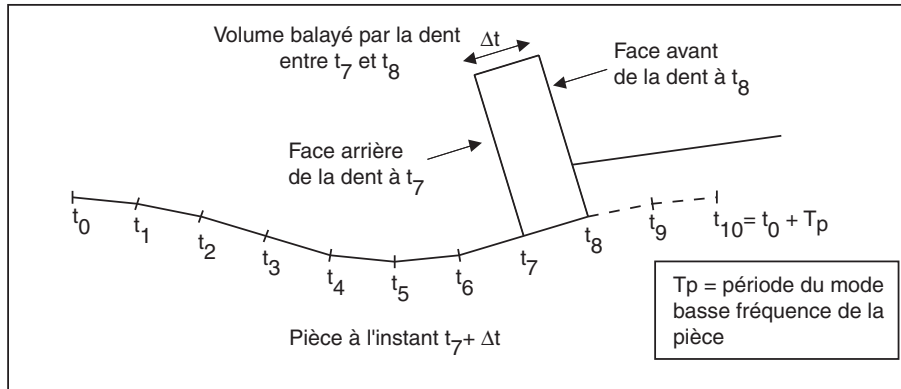


FIG. 4.1 – Calcul du pas de temps

Par ailleurs une estimation correcte des efforts de coupe à chaque pas de temps nécessite que le déplacement d'une face de coupe vis à vis de la matière ne soit pas trop important. Ainsi, que ce soit en fraisage ou en tournage, il est nécessaire d'avoir un nombre suffisant de pas de temps pour un tour de broche. Pour les fraises comportant plusieurs dents il est également impératif que les volumes balayés par les différentes faces de coupe ne se chevauchent pas au cours d'un même pas de temps. Par exemple, avec une fraise comportant 10 dents, la fraise doit faire moins d'un dixième de tour au cours d'un même incrément. Si on note N_{tour} le nombre minimal de pas de temps par tour de broche et Ω_{broche} la vitesse de rotation de cette dernière (en tours par minute), on obtient la contrainte :

$$\Delta t \leq \Delta t_{tour} \quad \text{avec} \quad \Delta t_{tour} = \frac{1}{N_{tour}} \cdot \frac{60}{\Omega_{broche}} \quad (4.2)$$

Δt et Δt_{tour} sont exprimés en secondes.

Au final, nous devons avoir :

$$\Delta t \leq \min(\Delta t_{tour}, \Delta t_{obs}) \quad (4.3)$$

4.1.2 Choix du nombre de dexels

Le choix du nombre de dexels, ou de façon plus pratique la taille ΔD des dexels (pour une grille homogène : $\Delta X = \Delta Y = \Delta D$), doit répondre à deux contraintes.

La première est de pouvoir représenter correctement la surface usinée si l'usinage reste stable. Ainsi pour représenter "correctement" le sillon laissé entre le passage successif de deux faces de coupe, il faut un nombre n_{sillon}^{dexels} de dexels suffisant (environ 5). En fraisage, pour une avance f (en millimètres par tour et par dent) on doit vérifier :

$$\Delta D \leq \Delta D_{sillon} \quad \text{avec} \quad \Delta D_{sillon} = \frac{f}{n_{sillon}^{dexels}} \quad (4.4)$$

où ΔD est exprimé en millimètres.

La seconde est d'avoir suffisamment de dexels coupés lors d'un incrément afin de minorer, toujours au cas où l'usinage reste stable, les fluctuations de l'effort de coupe dues à l'aspect discret de la représentation par dexels. Le nombre de dexels coupés étant nécessairement entier, même pour un cas d'un usinage parfaitement régulier il est possible d'observer des variations de l'effort de coupe. Le fait que le calcul des efforts de coupe passent par un calcul de volume tend déjà à limiter (*cf* paragraphe 3.5) ces fluctuations. En notant n_{coupe}^{dexels} le nombre de dexels que l'on s'impose de couper sur un pas de temps (dans le sens de la vitesse de coupe) on obtient la seconde contrainte :

$$\Delta D \leq \Delta D_{coupe} \quad \text{avec} \quad \Delta D_{coupe} = \Delta t \cdot \frac{V_{dent/pièce}}{n_{coupe}^{dexels}} \quad (4.5)$$

où $V_{dent/pièce}$ est une estimation de la valeur moyenne de la vitesse relative des dents par rapport à la matière de la pièce. Comme le montreront les simulations numériques, une valeur correcte pour n_{coupe}^{dexels} est 2 ou 3.

Au final, nous devons avoir :

$$\Delta D \leq \min(\Delta D_{sillon}, \Delta D_{coupe}) \quad (4.6)$$

4.1.3 Interpolation du volume balayé sur un pas de temps

Sur un intervalle de temps $[t, t + \Delta t]$, le volume balayé par une face de coupe de l'outil est construit à partir de la position de la face de coupe à l'instant t , de la position de la face de coupe à l'instant $t + \Delta t$ et éventuellement à partir d'un certain nombre de positions intermédiaires (cette construction se fait dans la configuration matérielle). L'intérêt d'ajouter ces positions intermédiaires est d'améliorer, si on le juge nécessaire, la description du volume balayé. Cette fonctionnalité peut s'avérer particulièrement intéressante si, une fois les conditions (4.3) et (4.6) respectées, la distance parcourue par une face de coupe durant un pas de temps demeure grande vis à vis de la taille d'un sillon (figure 4.2).

Soit $n_{sillon}^{\Delta t}$ le nombre de positions de faces de coupe minimale qu'on se fixe pour décrire un sillon (typiquement $n_{sillon}^{\Delta t} = 3$). Le nombre d'intervalles entre les positions successives

d'une même face sur un pas de temps $n_{intervalle}^{\Delta t}$ (nombre total de position de faces de coupe - 1) est donné par :

$$n_{intervalle}^{\Delta t} \geq \max \left(1, \frac{1}{f} \cdot \frac{V_{dent/pièce}}{n_{sillon}^{\Delta t}} \right) \quad (4.7)$$

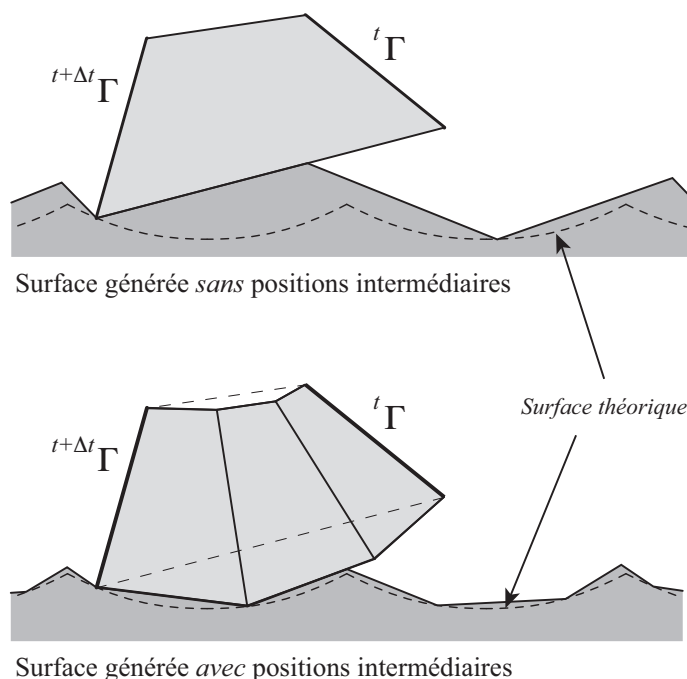


FIG. 4.2 – Interpolation du volume balayé avec ou sans positions intermédiaires

4.2 Usinage d'une pièce rigide avec trajectoire d'outil rectiligne

Pour cette première étude, la pièce est rigide et immobile. L'outil suit une consigne de déplacement à vitesse constante suivant la direction d'avance (figure 4.3). Il s'agit d'un cas purement cinématique pour lequel l'état de la surface usinée peut être déterminé analytiquement. Ceci va nous permettre de valider le logiciel et le choix de paramètres d'entrée (la taille des dexels et le pas de temps) sur des aspects purement géométriques.

Caractéristiques de l'usinage

Les caractéristiques de l'usinage sont les suivantes :

- diamètre de l'outil (fraise) : 10 mm
- nombre de dents : 1,
- angle d'hélice : 0°,
- conditions de coupe : vitesse d'avance $f = 0,1 \text{ mm/dent/tour}$, vitesse de rotation $\Omega = 27000 \text{ tour/minute}$,
- engagement radial : 2,5 mm,
- pièce rectangulaire de longueur 10 mm, de largeur 0,1 mm et de hauteur 4 mm.

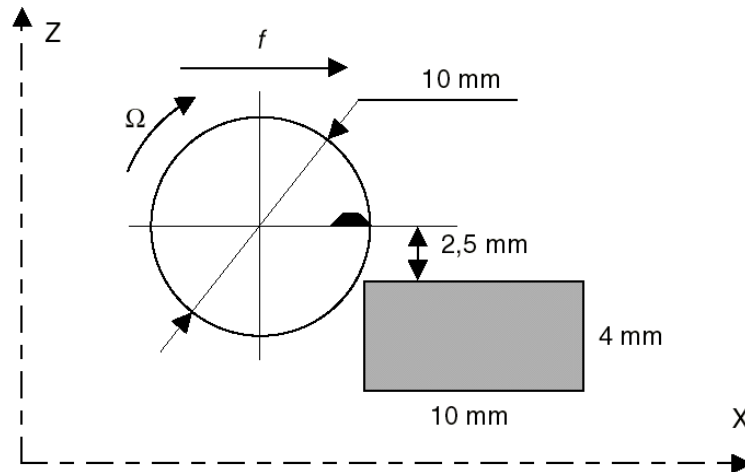


FIG. 4.3 – Configuration d'un usinage rigide

L'angle d'hélice de la dent de la fraise étant nul, le problème est invariant pour toute translation selon Y . Les dexels étant disposés parallèles à Z , une unique rangée de dexels selon Y est suffisante.

Par ailleurs, pour réaliser l'usinage complet de la pièce, la distance à parcourir par la fraise est de l'ordre de $2,5 \cdot \frac{\sqrt{3}}{2} + 10 \approx 12,2 \text{ mm}$ (entrée dans la matière + longueur de la pièce) ce qui correspond à 122 tour de fraise ($f = 0,1 \text{ mm/dent/tour}$). Enfin le nombre de sillons laissés sur la pièce est de $\frac{10}{f} = 100$.

Nous allons successivement étudier l'influence du pas de temps (à nombre de dexels fixé) puis l'influence du nombre de dexels (à pas de temps fixé) sur la description de la surface usinée.

Influence du pas de temps sur l'état de surface

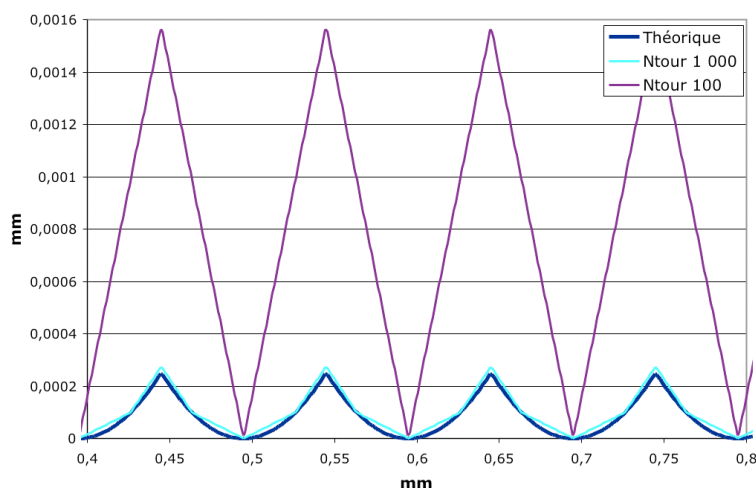
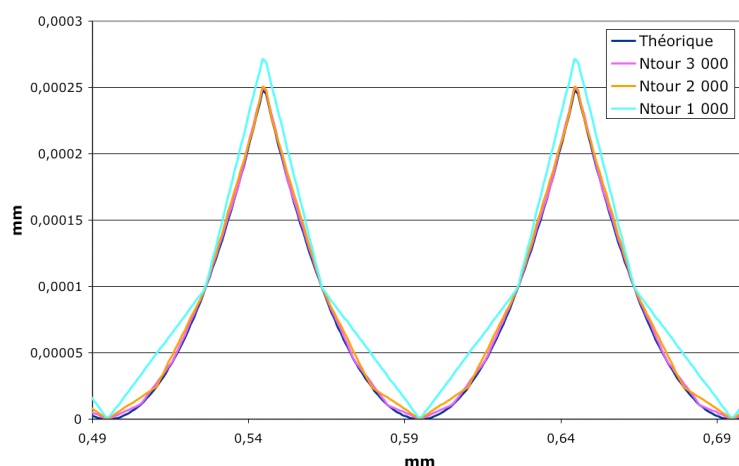
Afin d'avoir une description très fine de la surface usinée nous fixons $n_{\text{sillon}}^{\text{dexels}}$ à 100 ce qui donne 10 000 dexels pour décrire la pièce selon X .

Sur la figure 4.4 sont comparés les états de surfaces obtenus pour N_{tour} prenant les valeurs 100 et 1000 à l'état de surface théorique (N_{tour} représente le nombre d'incrément par tour).

Pour $N_{\text{tour}} = 100$, ce qui correspond à $\Delta t = 2,222 \cdot 10^{-5} \text{ s}$, le nombre d'incrément pour décrire correctement l'état de surface est insuffisant (majoration de l'écart crête/creux d'un facteur 5,5). Par contre pour $N_{\text{tour}} = 1000$, qui correspond à 4 incrément par sillon et à $\Delta t = 2,222 \cdot 10^{-6} \text{ s}$, on est nettement plus proche de la surface théorique.

N_{tour}	Δt en s
100	$2,222 \cdot 10^{-5}$
1000	$2,222 \cdot 10^{-6}$
2000	$1,111 \cdot 10^{-6}$
3000	$0,741 \cdot 10^{-6}$

TAB. 4.1 – Relation entre N_{tour} et Δt

FIG. 4.4 – Etat de surface théorique et avec $N_{tour} = 100$ et 1000FIG. 4.5 – Etat de surface théorique et avec $N_{tour} = 1000, 2000, 3000$

Il est à noter que nous n'avons pas ici divisé le pas de temps en plusieurs intervalles comme le suggère le paragraphe 4.1.3. Le choix du pas de temps pour cet exemple repose uniquement sur des aspects géométriques, la conditions (4.1) est inopérante ici.

Comme le montre la figure 4.5, l'augmentation du nombre d'incrémentes par tour améliore sensiblement la description de l'état de surface.

Influence du nombre de dexels sur l'état de surface

Pour toutes les simulations faites dans ce paragraphe, nous avons 2000 incrémentes par tour, ce qui correspond à un pas de temps $\Delta t = 1,111 \cdot 10^{-6} s$. La trajectoire simulée de la dent est ainsi relativement proche de la trajectoire théorique (cf figure 4.5). L'exemple traité étant purement cinématique cela nous permet de choisir librement le nombre de dexels. Le calcul de l'effort de coupe étant sans importance, la condition (4.5) n'a pas à être prise en compte.

Nous avons fait des simulations avec différents nombres de dexels : 10 000, 1 000 et 500 dexels.

Comme le montre la figure 4.6, seule la simulation avec 500 dexels donne un état

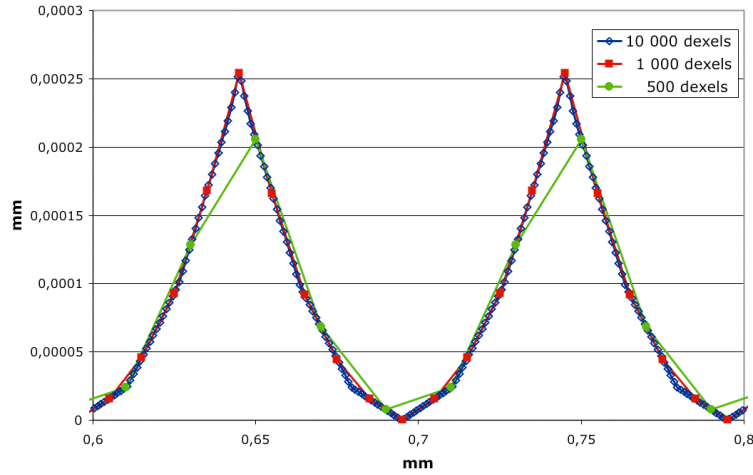


FIG. 4.6 – Etat de surface avec $N_{tour} = 2000$ et 10 000 dexels, 1 000 dexels et 500 dexels

de surface assez éloigné de la solution de référence. L'erreur obtenue peut aller jusqu'à 30%, si, toujours avec 500 dexels, le pic des sillons se situe exactement entre deux dexels successifs.

Les simulations avec 1 000 et 10 000 dexels permettent de représenter l'état de surface avec des erreurs maximales de l'ordre de 15% pour 1 000 dexels et inférieure à 1% pour 10 000 dexels.

Influence du nombre de dexels sur l'effort de coupe

La figure 4.7 représente les composantes des efforts de coupe pour deux simulations. Pour la première, une unique rangée de 500 dexels est disposée pour décrire l'ensemble de la pièce alors que pour la seconde simulation 5 000 dexels sont utilisés (toujours sur une seule rangée). Pour le cas avec 5 000 dexels, le nombre de dexels coupés, sur un pas de temps, est égal à $n_{coupe}^{dexels} \approx 4$. Par contre avec 500 dexels pour représenter la pièce, $n_{coupe}^{dexels} \approx 1$. On observe, pour ce dernier cas, des variations de l'effort de coupe.

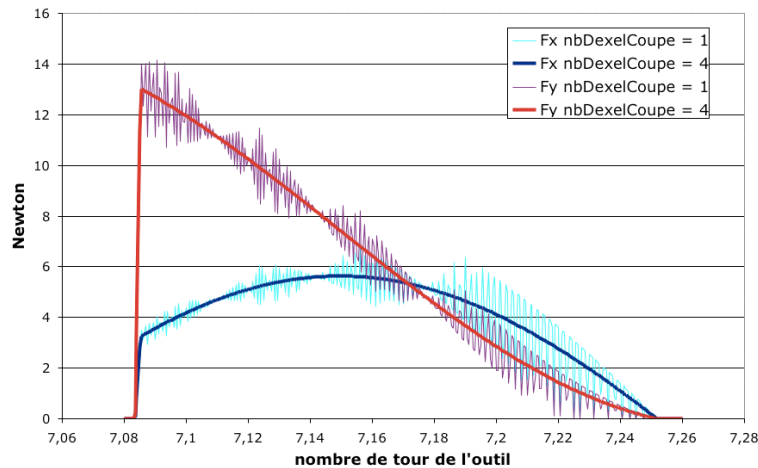


FIG. 4.7 – Représentation des composantes de l'effort de coupe avec un nombre de dexels = 1 ou 4

Un nombre de dexels coupé durant un pas de temps $n_{coupe}^{dexels} \approx 3$ est généralement suffisant.

Il est à noter que pour ces deux simulations nous nous trouvons dans un cas très particulier car une unique rangée de dexels est coupée. Pour un cas avec plusieurs rangés de dexels, et si la fraise possède un angle d'hélice non nul, $n_{coupe}^{dexels} \approx 1$ permet déjà d'observer une relative stabilité de l'effort de coupe.

4.3 Usinage d'une pièce rigide avec trajectoire d'outil complexe

L'exemple traité ici a été proposé par Grégoire Peigné et a été décrit en détail dans sa thèse [Pei04]. Par rapport à l'exemple traité dans le paragraphe précédent (paragraphe 4.2), seule la trajectoire de l'outil est modifiée. Est imposée, en plus de la vitesse d'avance constante f selon X , une consigne de déplacement "complexe" selon Z :

$$z = 0,5 \sin(17,3 x) + 0,12 \sin(13,3 x) \quad (4.8)$$

Ce mouvement est similaire à celui pouvant résulter des vibrations induites par des efforts de coupe pour un exemple réel d'usinage.

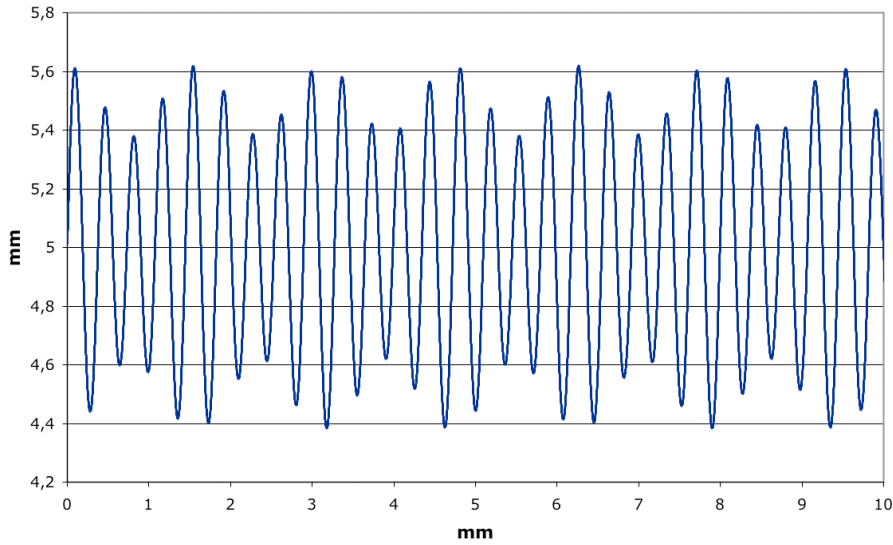


FIG. 4.8 – Evolution de la position horizontale du centre de la fraise en fonction de sa position verticale

Les fréquences de la consigne vibratoire sont égales à 17,3 et 13,3 fois la fréquence de rotation. Ces ratios sont choisis non entiers pour reproduire un déphasage dans le passage successif des dents tel qu'il existe dans un usinage instable.

L'objet est ici, pour une cinématique "complexe", de comparer les résultats obtenus avec *Nessy* avec ceux obtenus par Grégoire Peigné. Avant de faire cette comparaison nous décrivons brièvement, dans le paragraphe suivant, le modèle de simulation utilisé par Grégoire Peigné. Viennent ensuite les comparaisons à proprement parler. Cette étude comparative nous a également donné l'occasion de vérifier le bon fonctionnement du calcul de l'inverse de la fonction $(\phi_{Pièce}^+)$ (paragraphe 4.3.3).

4.3.1 Rappels des modèles géométriques de simulation de Grégoire Peigné

Le simulateur de Grégoire Peigné a été développé en vue de l'étude conjointe de la stabilité de la coupe et de la prédiction des défauts de la surface usinée.

La description géométrique de cette surface se fait dans un référentiel cartésien. Afin de réduire les coûts de calculs, deux simplifications ont été faites :

- la première permet de s'affranchir de la modélisation de la surface usinée d'une pièce de type voile mince en choisissant une pièce indéformable dans la zone d'usinage, c'est-à-dire qu'elle se comporte localement comme un solide rigide ;
- la seconde consiste à transformer la géométrie tridimensionnelle qui induit des calculs d'intersection outil-pièce très complexes en un empilement de tranches. A chaque tranche correspond une représentation géométrique 2D plane décrite par son contour. Toutes les tranches sont orthogonales à une même direction z .

Modèle géométrique de l'outil :

En accord avec le principe d'empilement précédent l'outil est représenté par un empilement d'outils élémentaires d'épaisseur Δz faible.

Représentation initiale d'une tranche :

Le contour de chaque tranche, à l'instant initial, est représenté par un ensemble de lignes brisées (dans le plan de la tranche). La densité des sommets dépend de la courbure locale de la surface et de la précision souhaitée sur le calcul des efforts de coupe et de la représentation de la surface finale.

Représentation en cours d'usinage d'une tranche :

Sur un incrément, le volume balayé par les faces de coupe de l'outil se réduit ici à un ensemble de surfaces pour chacune des tranches. La trajectoire d'un point de l'outil, vis à vis d'une tranche, est interpolée de façon quadratique grâce aux 3 positions successives du point : position à la fin de l'incrément actuel et à la fin des deux incréments précédents. Il est à noter, que seule la portion parcourue au cours de l'incrément actuel est utilisée. La description du contour d'une tranche, à un instant t , se compose ainsi d'un ensemble de segments droits et de portions de paraboles.

4.3.2 Comparaison des résultats obtenus

Pour cet exemple l'état de surface théorique n'est pas disponible. Pour comparer notre simulation avec celle de Grégoire Peigné, il a été nécessaire que ce dernier utilise un découpage très fin permettant d'avoir des erreurs d'interpolation négligeables.

De notre côté nous avons choisi $\Delta t = 2,222.10^{-6}$ s (soit $N_{tour} = 1000$) et 10 000 dexels suivant l'axe des X pour le modèle géométrique de la pièce (soit $n_{sillon}^{dexels} = 100$; dexels disposés sur une seule rangée selon Y) ce qui correspond également à une description fine de la surface usinée (cf paragraphe 4.2).

Une vue en détail de la surface simulée est donnée dans la figure 4.9. Une très grande proximité des deux simulations est observée.

Le graphe des efforts (figure 4.10) permet de mettre en évidence la validation de notre calcul de la section de copeau qui est une section moyenne équivalente (présentée dans

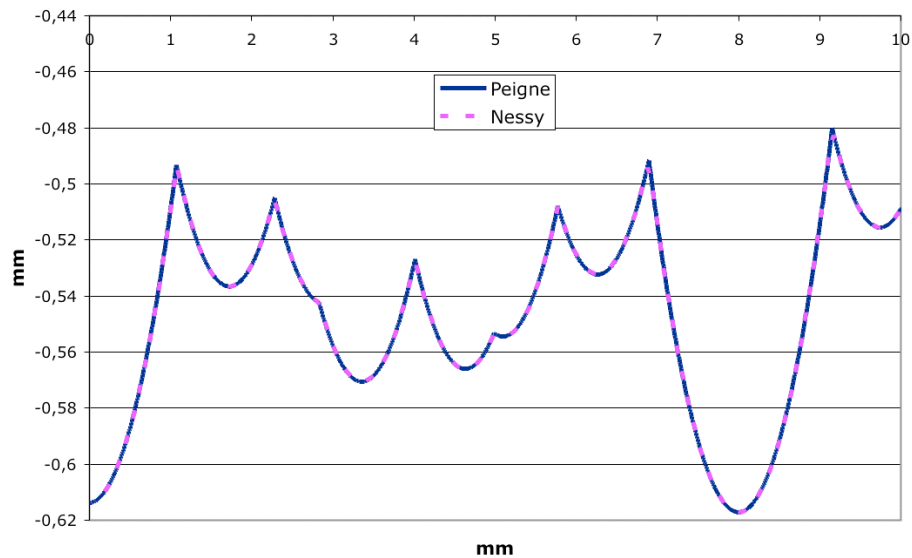


FIG. 4.9 – Comparaison des positions des points de la surface avec les deux simulateurs

le chapitre précédent 3.5). Grégoire Peigné utilise section instantanée ce qui explique les légères différences observées.

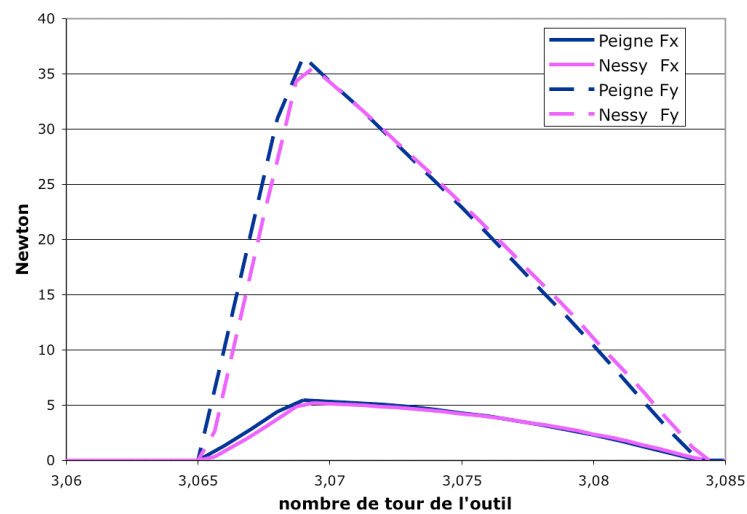


FIG. 4.10 – Comparaison des composantes de l'effort de coupe avec les deux simulateurs

Il ressort de cette étude cinématique que notre modélisation permet un calcul précis de la section du copeau, et par conséquent de l'effort de coupe, et une bonne représentation de la surface usinée quel que soit le pas angulaire utilisé et le régime vibratoire de la coupe.

Cette étude a permis de valider les aspects géométriques tels que l'intersection entre le volume balayé et la pièce et aussi le calcul de la section moyenne pour les efforts de coupe.

4.3.3 Validation de l'application $(\phi_{Pièce}^+)^{-1}$

Pour effectuer une première validation du calcul de $(\phi_{Pièce}^+)^{-1}$ nous avons imposé le mouvement à la pièce et non plus à l'outil avec l'opposé de la fonction oscillante donnée précédemment 4.8.

L'état de surface obtenu avec cette simulation a bien été identique à celui obtenu avec le mouvement oscillant de l'outil (fig 4.9).

4.4 Usinage d'une pièce rigide mobile selon une seule direction

L'usinage simulé ici est le fraisage d'une pièce rigide ayant un unique degré de liberté perpendiculaire à l'avance (figure 4.11). La pièce ne possède pas de flexibilité suivant la direction d'avance. L'outil utilisé est également rigide.

Cet exemple a été traité par Grégoire Peigné et décrit en détail dans sa thèse [Pei04]. Grégoire Peigné qualifie cette pièce de "pièce flexible" par allusion à une pièce pouvant être globalement flexible bien que rigide dans la zone concernée par l'usinage.

Après avoir décrit le modèle de simulation, une première comparaison avec les travaux de Grégoire Peigné est faite dans le cas d'un usinage où les deux simulateurs ont donné des résultats très voisins. Il s'agit d'un cas de simulation *stable*. Suit une seconde comparaison dans le cas d'un usinage pour lequel les résultats dépendent des paramètres choisis pour la simulation. Il s'agit d'un exemple de simulation *instable*.

Description du modèle de simulation

Modèle mécanique de la pièce :

Le comportement dynamique de la pièce est modélisé par un système de masse-ressort amorti. L'équation scalaire de mouvement est la suivante :

$$m_z \ddot{z} + c_z \dot{z} + k_z z = F_{cz} \quad (4.9)$$

Les valeurs des paramètres modaux de la pièce "flexible" utilisées pour les simulations sont :

	K (N/m)	ξ	f_n (Hz)
Direction z	$2,6 \cdot 10^6$	0,02	600

TAB. 4.2 – Caractéristiques modales

Ce qui donne après calcul :

k_z (N/m)	c_z (N/m/s)	m_z (kg)
$2,6 \cdot 10^6$	27,56	0,18

TAB. 4.3 – Caractéristiques dynamiques

Modèle d'effort de coupe :

L'effort de coupe est exprimé par sa composante tangentielle et sa composante radiale toutes deux fonctions linéaires de l'épaisseur du copeau h :

$$F_T = K_T \cdot b \cdot h \quad (4.10)$$

$$F_r = k_r \cdot F_T \quad (4.11)$$

où b est l'engagement radial en mm .

Les valeurs retenues pour les simulations sont : $K_T = 1500 \text{ Mpa}$ et $k_r = 0,3$.

Données géométriques et cinématiques :

Les caractéristiques de l'usinage sont les identiques à celles des exemples précédents. Seule la profondeur radial de coupe change. Elle est de 1 mm pour la simulation *stable* et de 3 mm pour la simulation *instable*.

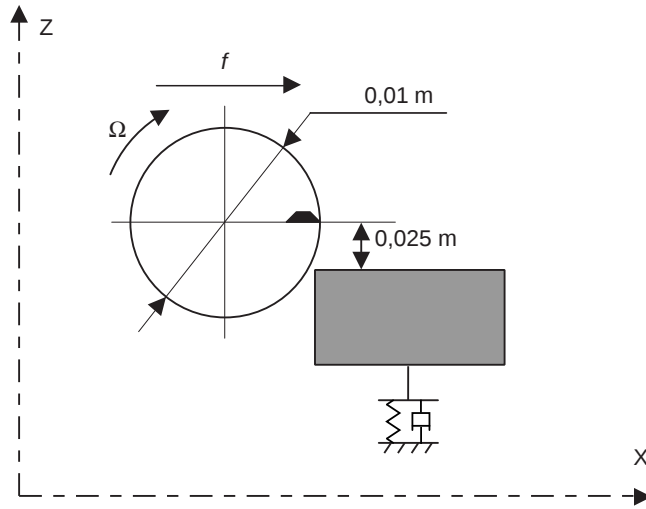


FIG. 4.11 – Configuration d'usinage d'une pièce "flexible"

Notre objectif étant dans l'immédiat de démontrer la faisabilité de notre approche, nous avons repris les valeurs utilisées par Grégoire Peigné sans chercher à les rattacher à une expérimentation précise. Pour une utilisation et une validation réelle de notre simulateur, une campagne d'essais devra être menée et devront être utilisées des lois et des paramètres de coupe physiquement représentatifs.

4.4.1 Simulation *stable*

Sur la figure 4.12 sont reproduites différentes représentations du déplacement selon Z de la pièce en fonction de l'avancement, selon X , du centre de l'outil. Les résultats obtenus avec *Nessy*, correspondent à ceux obtenus par Grégoire Peigné ce qui explique la présence d'une seule courbe par graphe.

Le nombre de dexels choisi pour représenter la pièce est de 10 000 et le pas de temps est $\Delta t = 2,222 \cdot 10^{-6} \text{ s}$. Ce pas de temps est tel qu'il y ait 1000 incréments par oscillation de la pièce (durée d'une oscillation $\approx \frac{1}{600} \approx 1,67 \cdot 10^{-3}$). L'équation (4.1) nous donne un pas de temps $\Delta t_{obs} = 1,67 \cdot 10^{-4}$, ce qui est environ 100 fois plus grand que le pas de

temps choisi. Le critère géométrique est ici prépondérant par rapport aux caractéristiques dynamiques du système POM.

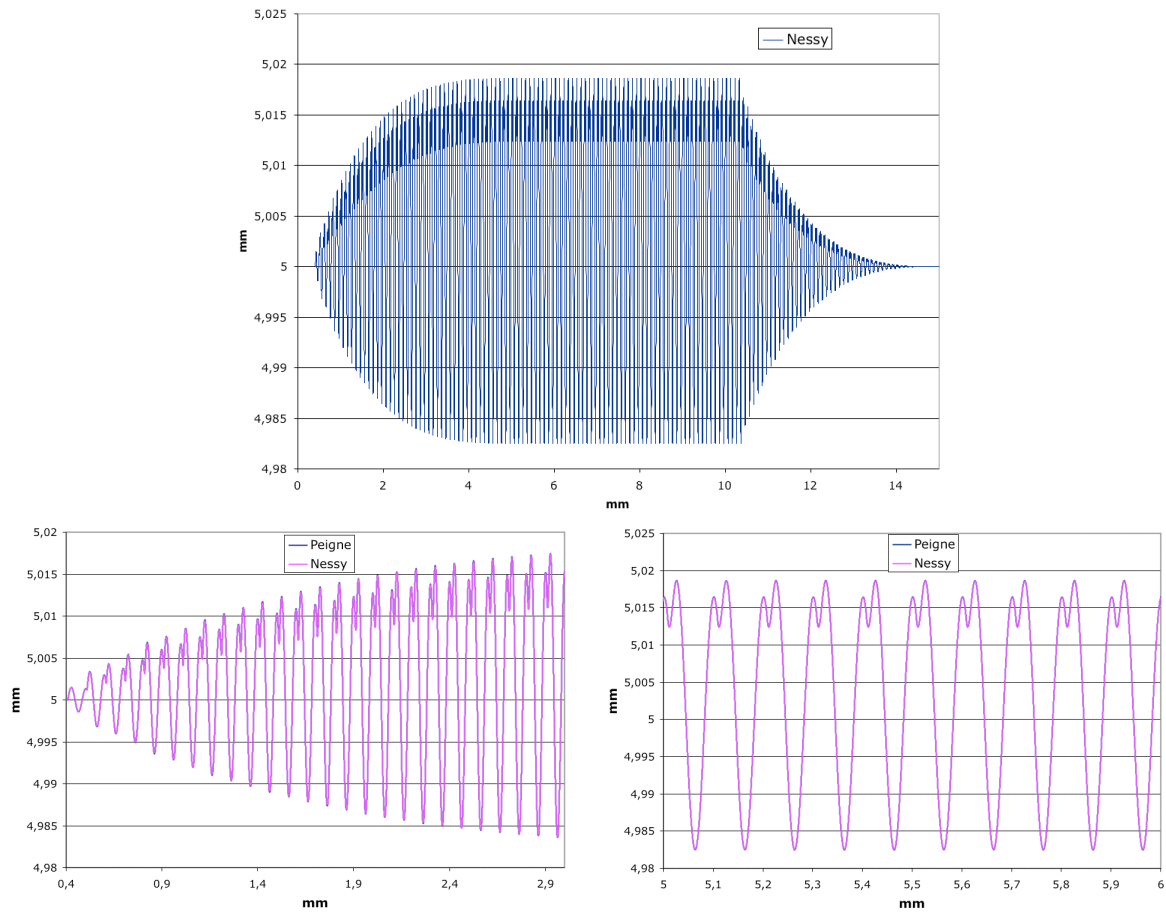


FIG. 4.12 – Déplacement de la pièce suivant l'axe des Z entre $[0, 15]$ et comparaisons sur deux zooms entre $[0, 3]$ et $[5, 6]$ avec les deux simulateurs : cas d'une simulation stable

La très grande régularité du résultat permet effectivement de qualifier cet usinage (ou encore cette simulation) de stable. Cette régularité se retrouve sur le profil de la pièce usinée (figure 4.13).

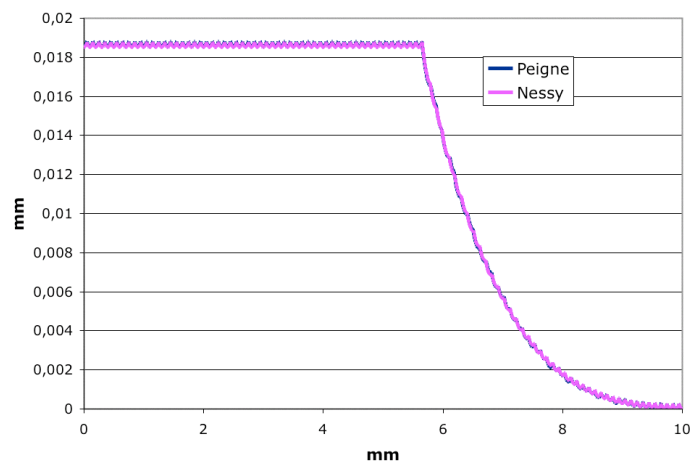


FIG. 4.13 – Comparaison des positions des points de la surface avec les deux simulateurs : cas d'une simulation stable

Pour ce qui est des efforts de coupe calculés (un extrait est donné figure 4.14), nous retrouvons les mêmes différences que celles observées dans le cas purement cinématique (paragraphe 4.3.2, figure 4.10).

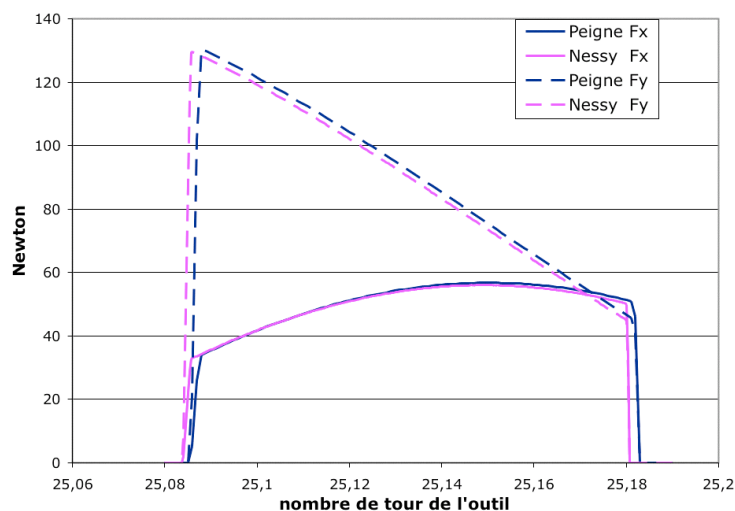


FIG. 4.14 – Comparaison des composantes de l'effort de coupe avec les deux simulateurs : cas d'une simulation stable

4.4.2 Simulation *instable*

Pour cet exemple, l'usinage est effectué avec une profondeur radiale de 3 mm au lieu de 1 mm dans le cas *stable*. Toutes les autres caractéristiques de l'usinage sont identiques.

Le nombre de dexels choisi pour représenter la pièce est de 10 000 et le pas de temps est $\Delta t = 2,222 \cdot 10^{-6} s$. Ce pas de temps est identique à celui utilisé par Grégoire Peigné lors de sa simulation.

Sur la figure 4.15 sont comparés les états de surface obtenus, avec le même pas de temps, par le simulateur de Grégoire Peigné et par notre simulateur *Nussy*. Ces deux courbes ont une allure assez proche mais ne se superposent pas.

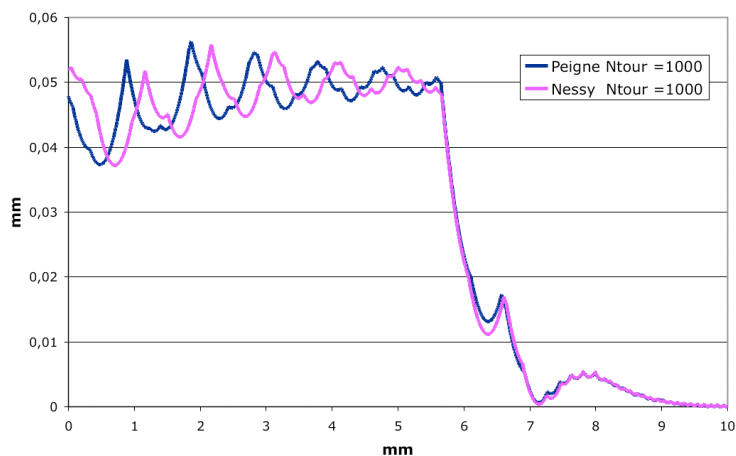


FIG. 4.15 – Comparaison des positions des points de la surface avec les deux simulateurs : cas d'une simulation instable

On peut remarquer que le début du profil de surface de la pièce résulte de 46 tours c'est-à-dire de 46 passages successifs de la dent (46 000 incréments).

Sur la figure 4.16 sont reproduites différentes représentations du déplacement selon Z de la pièce en fonction de l'avancement, selon X , du centre de l'outil. Les résultats obtenus avec *Nessy*, et par Grégoire Peigné ne se superposent qu'au début (jusqu'à 40 tours soit 40 passages successifs de la dent (40 000 incréments), ensuite ils sont différents. Cette différence permet de justifier l'écart existant entre les deux états de surface de la figure 4.15.

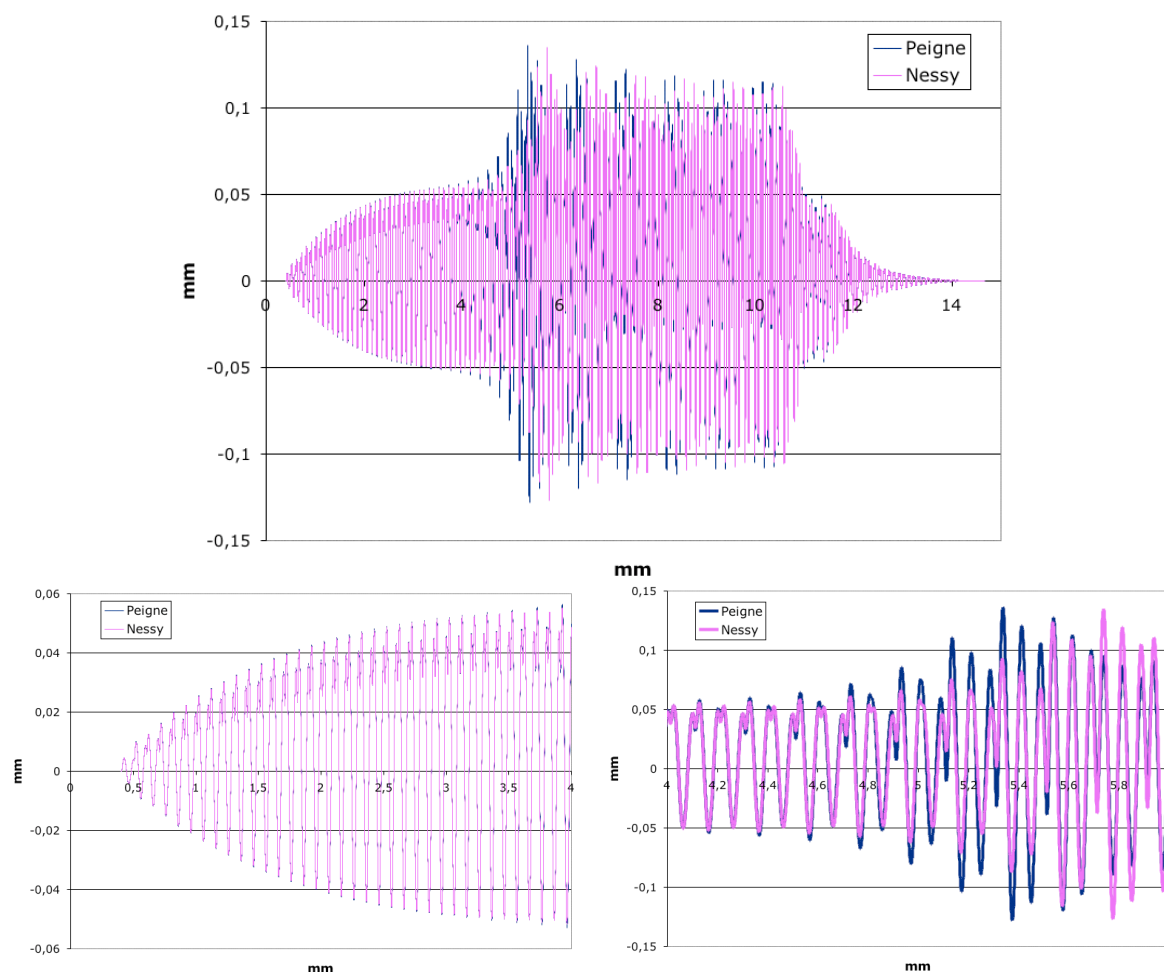


FIG. 4.16 – Déplacement de la pièce suivant l'axe des Z entre $[0, 15]$ et comparaisons sur deux zooms entre $[0, 4]$ et $[4, 6]$ avec les deux simulateurs : cas d'une simulation instable

Afin de vérifier si le nombre de dexels n'est pas à l'origine de cette différence nous avons réalisé une simulation avec 40 000 dexels au lieu de 10 000. Aucune différence notable n'a été observée entre les deux simulations. La finesse de description de la surface n'est donc pas responsable des différences observées sur la figure 4.15.

Nous avons ensuite décidé de modifier *légèrement* le pas de temps. Nous avons pris $\Delta t = 2,116 \cdot 10^{-6} s$ au lieu de $2,222 \cdot 10^{-6} s$ ce qui correspond à 1 050 incréments par tour au lieu de 1 000. Sur la figure 4.17 nous avons tracé les surfaces finales pour ces deux pas de temps. Les courbes ont des allures générales semblables mais sont différentes.

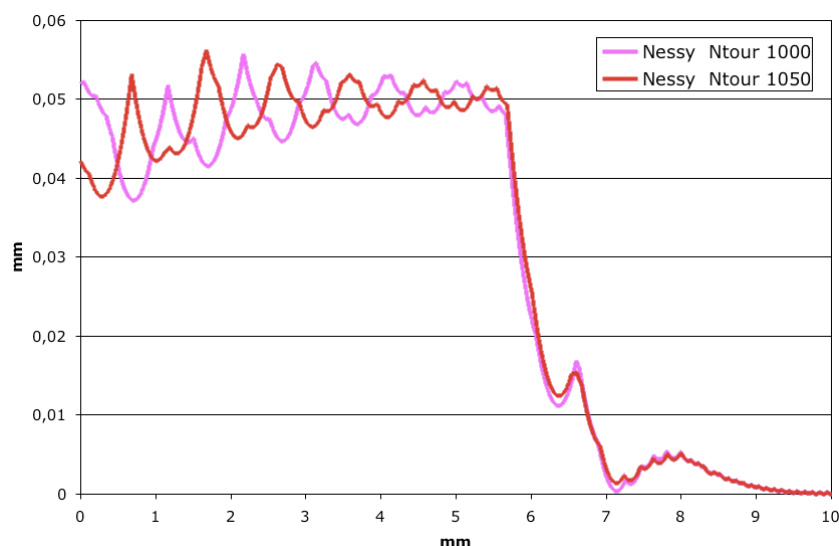


FIG. 4.17 – Représentation des positions des points de la surface avec deux pas de temps différent $N_{\text{tour}} = 1000$ et $N_{\text{tour}} = 1050$: cas d’une simulation instable

Pour tenter de mieux cerner les raisons de ces différences nous avons réalisé une simulation avec un pas de temps trois fois plus petit : $\Delta t = 7,407 \cdot 10^{-7} \text{ s}$. Nous avons également augmenté le nombre de dexels en proportion (30 000 dexels).

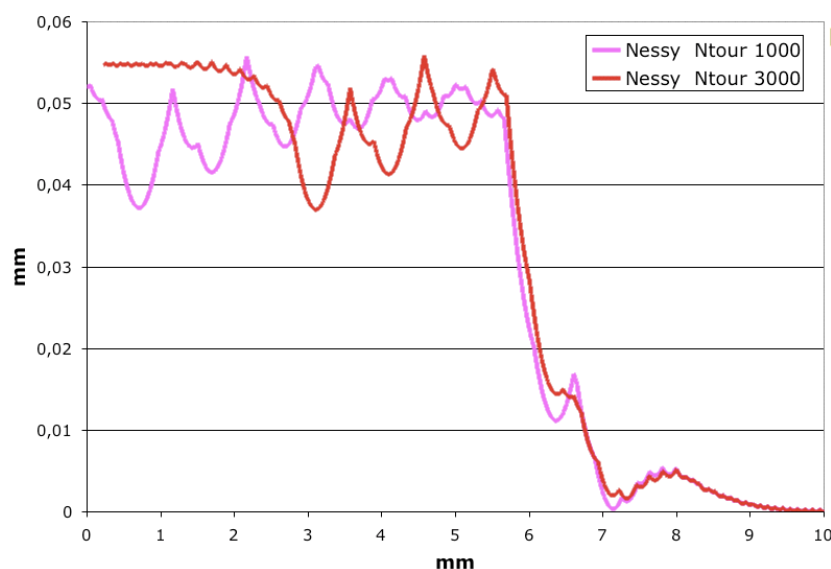


FIG. 4.18 – Représentation des positions des points de la surface avec deux pas de temps différent $N_{\text{tour}} = 1000$ et $N_{\text{tour}} = 3000$: cas d’une simulation instable

Le résultat sur l’état de surface est encore différent. Il semblerait que nous nous approchions d’un cas de simulation instable.

Il paraît raisonnable de penser que de toutes petites perturbations peuvent déclencher cette instabilité. Dans ces conditions, parmi les explications quant à la provenance de la différence de l’état de surface, nous pouvons avancer que :

- la première interaction outil-pièce n’a pas lieu exactement au même instant ce qui est liée, indirectement, au pas de temps Δt choisi ;

- le choix des paramètres pas de temps et nombre de dexels a une influence sur l'évaluation de l'état de surface ainsi que sur la détermination des efforts de coupe. Comme en plus les deux sont liés, l'étude n'est pas évidente ;
- le choix du schéma d'intégration numérique est peut être mal adapté à cette simulation.

La possibilité de mener à bien et efficacement une simulation complète ne se fait que depuis peu.

Il est très clair que ce point demande une étude plus approfondie qu'il reste à mener.

4.5 Exemple d'un cas 3D d'une pièce déformable

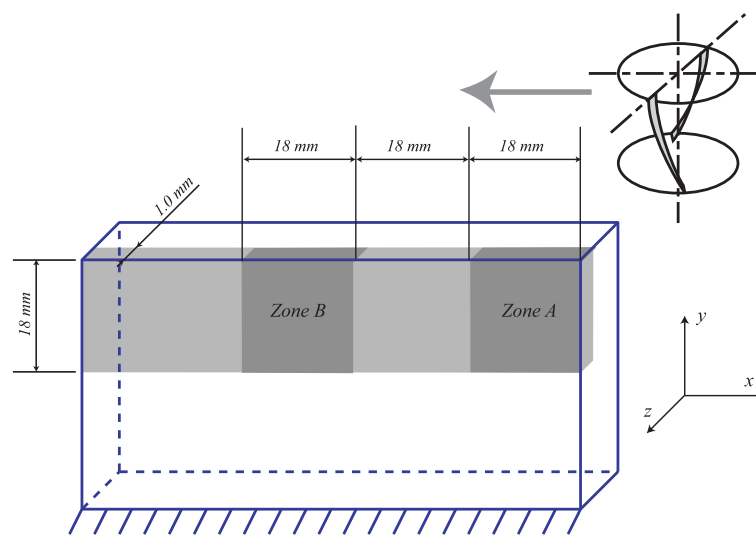


FIG. 4.19 – Définition du domaine usiné de la pièce

Nous présentons pour finir la simulation d'un usinage de voile mince. Cet exemple vise à montrer le potentiel de l'approche proposée lorsque la flexibilité de la pièce dans la zone usinée ne peut être négligée. Il s'agit d'un cas "virtuel" dont tous les paramètres sont réalistes. L'usinage réel n'a pas été réalisé et la validation expérimentale de cette simulation n'a pu être effectuée.

Les caractéristiques de l'usinage sont les suivantes :

- diamètre de l'outil (fraise hélicoïdale) : 12 mm
- nombre de dents : 3,
- angle d'hélice constant : 30°,
- conditions de coupe : vitesse d'avance $f = 1,8 \text{ mm/dent/tour}$, vitesse de rotation $\Omega = 43\,115 \text{ tour/minute}$,
- profondeur axiale de la coupe de 18 mm (le long de l'axe des Y),
- profondeur radiale de coupe fixée à 1 mm (dans la direction Z),
- pièce rectangulaire de longueur 100 mm, de largeur 40 mm et d'épaisseur 4 mm.

La figure 4.19 illustre le domaine de la pièce ainsi que le domaine usiné. La base de la pièce est encastrée.

L'avance par dent est de 1,8 mm par tour, ce qui donne une "rugosité" de 0,1 mm (distance crête/creux pour un sillon en l'absence de vibration).

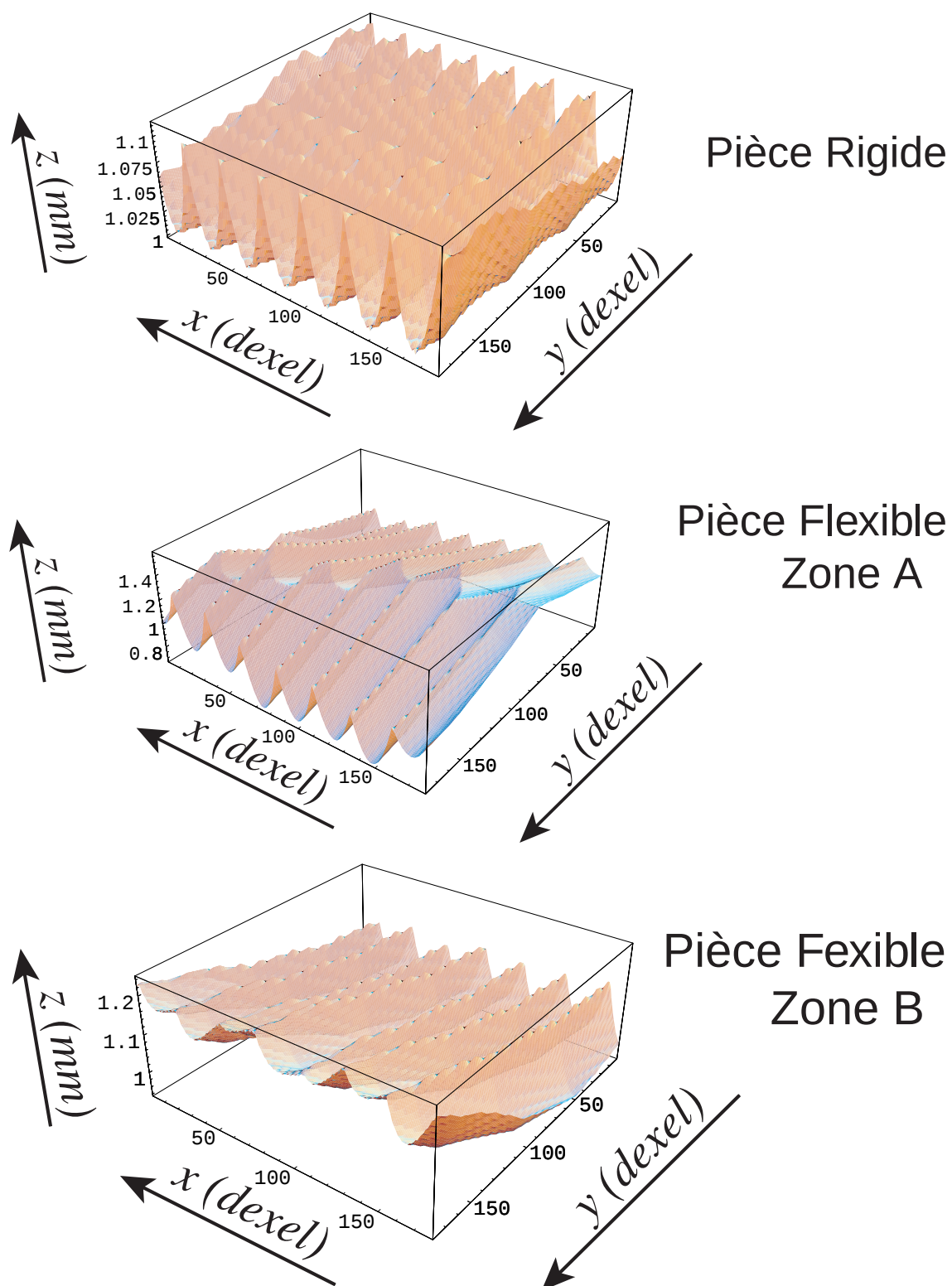


FIG. 4.20 – Exemple de l'état de surface simulée pour une pièce voile mince

Le matériau de cette pièce est un acier dont les caractéristiques sont les suivantes :

- module de Young $E = 2.10^{11} \text{MPa}$,
- masse volumique $\rho = 7800 \text{kgm}^{-3}$,
- coefficient de Poisson $\nu = 0,3$.

La pièce est modélisée avec un seul élément coque à huit nœuds. La fréquence du premier mode propre est de 2134Hz . La vitesse de rotation tend vers la fréquence de l'excitation (2155Hz) qui est très proche de la première fréquence propre de la pièce.

La matrice d'amortissement $\underline{\underline{C}}$ est choisie proportionnelle à la matrice de rigidité $\underline{\underline{K}}$: $\underline{\underline{C}} = \lambda \underline{\underline{K}}$. Le coefficient λ a été calculé pour obtenir un taux d'amortissement de $0,005$ pour le premier vecteur propre. Ceci conduit à l'expression suivante : $\lambda = 2(0,005/\omega) \approx 7,510^{-7}\text{s}$.

Pour modéliser la pièce, il a été nécessaire de mettre 1400 dexels le long de l'axe des X et 180 le long de l'axe des Y . La figure 4.20 montre, en haut, la surface usinée pour une pièce rigide. Les deux autres vues correspondent aux frontières usinées pour la zone A et la zone B (ces zones sont définies sur la figure 4.19) pour une pièce flexible.

Pour la zone A, la "rugosité" (distance crête/creux) est de l'ordre de $0,88\text{ mm}$, tandis que pour la zone B elle est de $0,30\text{ mm}$. Ces deux "rugosités" devant être comparées avec celle obtenue lorsque la pièce est supposée rigide : $0,10\text{ mm}$.

L'évolution de la "rugosité" entre la zone A et zone B provient de l'évolution du comportement vibratoire de la pièce en cours d'usinage. Ce dernier est lié à la flexibilité de la pièce conjugué au déplacement du lieu d'interaction outil/pièce.

4.6 Conclusion

La première remarque qui doit être faite est que nous disposons maintenant d'algorithmes d'intersection robustes, aucun problème d'intersection n'ayant interrompu les calculs : nous avons pu réaliser des simulations complètes d'usinage.

Les premiers exemples ont permis de valider de façon très fine les algorithmes géométriques et les algorithmes d'intégration temporels par comparaison avec les résultats analytiques ou issus du simulateur de Grégoire Peigné. Seules les évaluations des efforts de coupe sont très légèrement différentes. Dans les cas *stables*, les états de surface finaux sont identiques.

Les seuls écarts notables sont observés dans le cas d'une simulation *instable*. Les investigations que nous avons menées dans ce cas montrent clairement l'influence de certains paramètres sur le calcul numérique (pas de temps, instant du premier contact outil-pièce en particulier). Ceci n'est pas surprenant car le travail au voisinage d'une situation instable peut s'avérer extrêmement sensible à de petites variations des paramètres. Il reste donc encore à faire des investigations plus précises dans ce cas.

Néanmoins, on peut observer que les états de surface obtenus avec les diverses simulations sont très différents mais qu'ils ont des allures générales très voisines. Du point de vue de l'utilisateur, il se peut que ceci soit déjà bien suffisant.

La dernière simulation visait à montrer le potentiel de l'approche proposée lorsque la flexibilité de la pièce dans la zone usinée ne peut être négligée.

Nous n'avons pas à ce stade réalisé de comparaison avec des résultats expérimentaux. De nombreux travaux, en particulier ceux de François Lapujoulade [Lap03], montrent déjà la pertinence de l'approche dans le cas d'une pièce non déformable, voire déformable. Il n'existe pas de raison particulière pour que l'approche retenue modifie ces résultats : la seule chose que nous rajoutons est une prise en compte fine du comportement de la pièce permettant ainsi d'étendre les simulations à des pièces réellement flexibles, et ce par une méthode générale.

Chapitre 5

Pièce ayant une rigidité et une masse évolutives

Au cours de l'usinage, l'enlèvement de matière induit une modification de la rigidité et de la masse de la pièce. Cette modification peut engendrer une évolution significative du comportement dynamique de la pièce si cette dernière est très mince. C'est en particulier le cas pour les voiles minces où 2/3 de l'épaisseur peuvent être supprimés lors d'une passe d'ébauche. La prise en compte de l'évolution du comportement est alors nécessaire et doit souvent être associée à la prise en compte des déformations de la pièce dans la zone usinée (chapitre 2).

Une première possibilité pour prendre en compte, le plus fidèlement possible, l'évolution du domaine usiné est d'adapter le maillage éléments finis à chaque pas de temps. Cela n'est hélas pas envisageable pour plusieurs raisons. Il faut tout d'abord disposer d'un outil de maillage suffisamment fiable pour pouvoir réaliser la simulation sans arrêts intempestifs, et efficace pour ne pas accroître le temps de calcul de façon exagérée. Ensuite, une projection des quantités cinématiques (déplacements et vitesses) est à réaliser entre les maillages successifs ce qui génère, outre des coûts supplémentaires, une perte d'information. Enfin, l'évolution constante de la colonne de DDL $\underline{q}$ (nombre de valeurs et significations de ces dernières) rend la définition de $\phi_{Pièce}^+$ évolutive, et de ce fait plus délicate à réaliser.

Pour lever ces difficultés, une autre possibilité consiste à prendre en compte de façon simplifiée l'effet - et a priori - de l'enlèvement de matière. Pour cela nous proposons de fixer la géométrie du maillage éléments finis tout en modifiant les caractéristiques mécaniques (module d'Young, masse volumique) de certains éléments. Cette modification doit tendre à reproduire, au mieux, le comportement de la pièce à chaque instant. Le paragraphe 5.1 présente la mise en œuvre de cette approche.

La modification des caractéristiques mécaniques sur le maillage fixe conduit à des matrices de masse $\underline{\underline{M}}$ et de rigidité $\underline{\underline{K}}$ dont certaines valeurs sont en constante évolution. Dans le schéma d'intégration numérique temporel, (paragraphe 3.6), le calcul de $\underline{\underline{K}}$ et, surtout, de sa factorisation à chaque pas de temps dt représente un coût numérique important. Nous proposons, dans le paragraphe 5.2, une simplification pour le calcul de la factorisation de $\underline{\underline{K}}$. Cette simplification se base sur un calcul effectif de la factorisation tous les n pas de temps seulement.

5.1 Prise en compte simplifiée de l'enlèvement de matière

5.1.1 Hypothèses de travail

Dans toute la démarche proposée ici, nous supposons que les modifications de géométrie résultant des vibrations sont relativement faibles devant les modifications de géométrie qui seraient obtenues dans le cas d'un "usinage idéal" (typiquement épaisseur de l'ordre du millimètre, erreurs dues aux vibrations de l'ordre du 1/10 du millimètre). Cet usinage idéal est ici défini en faisant les hypothèses suivantes :

- la pièce et l'outil sont parfaitement rigides (ou ont des déformations "statiques" connues a priori),
- le mouvement de rotation relative outil-pièce est idéalisé en admettant que le mouvement de rotation relatif s'effectue à vitesse infinie (rotation de la pièce autour de son axe en tournage ou rotation de l'outil autour de son axe en fraisage ou en perçage),
- le mouvement d'avance est exactement égal à celui qui est imposé par la consigne fournie au dispositif de commande numérique des organes de la machine.

Il s'agit en réalité des hypothèses faites dans des logiciels de CAO.

5.1.2 La démarche

A chaque instant t , il est possible de construire une colonne $\underline{p}$ de valeurs définissant la position théorique outil/pièce (sans vibrations). Ceci permet de prévoir la géométrie théorique de la pièce à partir d'une fonction scalaire $h(t)$ déduite de l'histoire de $\underline{p}(\tau)$ pour $\tau \in [0, t]$.

Il est légitime de supposer, comme dit précédemment, que le volume réel enlevé est voisin du volume théorique précédent car les vibrations doivent demeurer petites vis à vis de l'épaisseur coupée. Ainsi, l'évolution du comportement dynamique de la pièce peut être pilotée par l'évolution *connue* - a priori - de la géométrie théorique de la pièce.

Notre approche consiste à introduire une fonction de présence de la matière $D(h(t), \underline{x})$ des éléments dans la zone usinée pour simuler l'affaiblissement local de la pièce lié à l'enlèvement de matière. $\underline{x}$ est la position d'un point appartenant à un des éléments dans la zone usinée et $h(t)$ la fonction d'histoire précédemment introduite.

La fonction de présence de la matière est connue de façon explicite en fonction de l'histoire $h(t)$. Elle est utilisée pour modifier les caractéristiques du matériau, comme par exemple le module d'Young E et la masse volumique ρ ; le coefficient de Poisson ν restant inchangé. Le choix d'une correction analogue pour le module d'Young et la masse volumique a été retenu pour ne pas générer de modes propres parasites. Cette correction est calculée en fonction de la fraction volumique usinée d'un élément.

$$E' = E(1 - D)$$

$$\rho' = \rho(1 - D)$$

$$\text{avec } 0 < D < D_{max} \quad \text{où } D_{max} = 0,95$$

D_{max} n'est pas choisi égal à 1 pour ne pas introduire de matrices singulières. Il ne doit pas être pris trop proche de 1 pour éviter d'éventuel problème de conditionnement des matrices.

L'hypothèse qui est faite et qui pourra être faite dans une simulation réelle d'usinage est que cette façon de procéder permet d'obtenir une estimation suffisamment précise des matrices de masse $\underline{\underline{M}}(h)$, de rigidité $\underline{\underline{K}}(h)$ et d'amortissement $\underline{\underline{C}}(h)$ pour représenter le comportement vibratoire de la pièce.

Pour des éléments de type plaque (ou coque), si l'usinage ne modifie que l'épaisseur des éléments il est possible d'introduire directement une évolution de l'épaisseur et de la position du segment traversant vis à vis du plan de référence pour traduire l'enlèvement théorique de matière. Il est, dans ce cas, inutile d'introduire une fonction de présence de la matière.

Nous présentons ci-dessous une validation de la démarche qui repose sur l'application de cette dernière à un problème simple suffisamment représentatif.

5.1.3 Validation de la démarche

Nous avons réalisé plusieurs simulations numériques qui reposent toutes sur la même pièce. Il s'agit d'un parallélépipède rectangle. Ses dimensions sont $100mm \times 40mm \times 4mm$. La base de la pièce est encastree. Le matériau est un acier dont les caractéristiques sont les suivantes :

$$E = 2.10^{11} MPa \quad (5.1)$$

$$\nu = 0,3 \quad (5.2)$$

$$\rho = 7800 kgm^{-3} \quad (5.3)$$

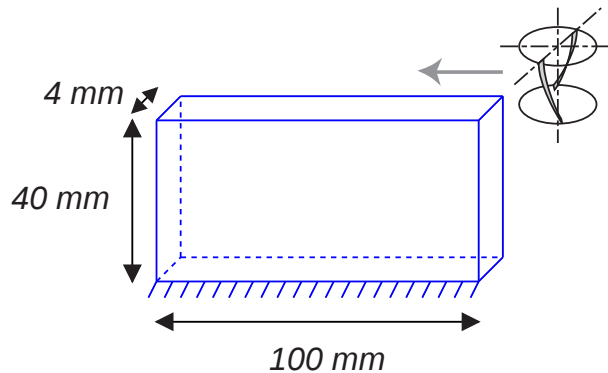


FIG. 5.1 – Les dimensions de la pièce

Une première série de simulations a pour objectif de vérifier que l'enlèvement de matière influe bien sur le comportement dynamique de la pièce. Dans une seconde série, nous montrons que l'introduction de la fonction de présence de la matière permet d'obtenir des résultats voisins à ceux obtenus avec enlèvement effectif d'éléments. Nous vérifions également que l'approche n'introduit pas de modes parasites.

Influence de l'enlèvement de matière sur le comportement dynamique de la pièce

Pour cette série de simulations, les modifications du comportement mécanique lors d'un usinage sont importantes. Les éléments de la zone usinée sont supprimés au fur et

à mesure de l'avancement de l'usinage. La moitié de l'épaisseur est retirée dans la partie haute de la pièce (fig. 5.2).

Le maillage retenu est assez fin. Le nombre d'éléments est de 640 (20 dans la longueur suivant l'axe des X , 8 dans la hauteur suivant l'axe des Y , 4 dans l'épaisseur suivant l'axe des Z). L'élément utilisé est un hexaèdre à huit nœuds.

Un calcul de modes propres de vibration a été fait pour la pièce encastree définie précédemment. Nous donnons dans le tableau 5.1.3 une comparaison des trois premières fréquences propres pour :

- une pièce vierge ($L_X = 0mm$),
- une pièce usinée sur la moitié de sa longueur ($L_X = 50mm$),
- une pièce usinée sur toute sa longueur ($L_X = 100mm$),

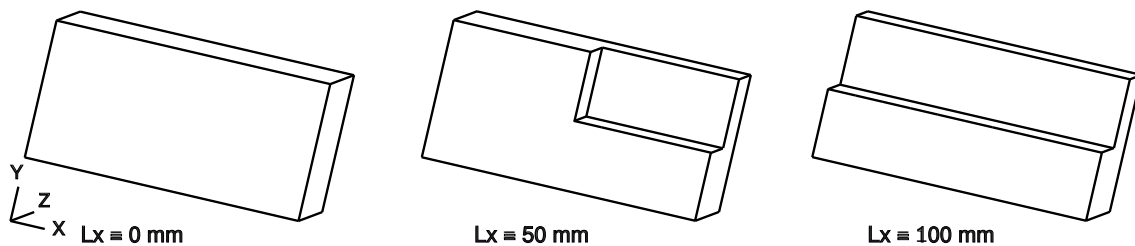


FIG. 5.2 – La pièce en hexaèdres avec des éléments supprimés

N° du mode	$L_X = 0mm$	$L_X = 50mm$	$L_X = 100mm$
1	2115, 4Hz	2233, 7Hz	2515, 9Hz
2	2842, 3Hz	2933, 6Hz	2889, 9Hz
3	4758, 3Hz	4361, 9Hz	4027Hz

TAB. 5.1 – Fréquences des trois premiers modes propres

Ces valeurs permettent de mettre en évidence une variation non négligeable (20%), mais néanmoins modérée, des fréquences en fonction de la grande partie usinée. Les modes propres de vibration ont une forme analogue (pas de croisement) en tout cas pour les trois premiers. Cette étude nous permet de conclure à une modification assez lente des caractéristiques mécaniques modales.

Validation de la méthode de la fonction de présence de la matière

Nous comparons ici, pour un usinage terminé, le comportement dynamique de la pièce d'un modèle dans lequel on retire les éléments "usinés" (*modèle 1*) avec un modèle où ces mêmes éléments sont simplement "endommagés" (*modèle 2*).

La partie à usiner doit être relativement importante pour pouvoir mettre en évidence des effets significatifs de l'usinage. Le volume usiné concerne la moitié de l'épaisseur, les 3/5 de la hauteur ainsi que la totalité de la longueur de la pièce.

Le maillage retenu est grossier. Nous le noterons MF, pour Maillage Fixé. Le nombre d'éléments est de 100 ($10 \times 5 \times 2$), ce qui permet d'avoir une première idée des résultats pour un maillage simple. L'élément utilisé dans le maillage est un hexaèdre à huit nœuds.

Pour récapituler, nous avons :

modèle 1 : les éléments considérés comme usinés sont retirés de la pièce. Nous avons alors ôté 30 éléments sur toute la longueur ($10 \times 3 \times 1$) sur les 100 éléments de la pièce.

modèle 2 : une fonction de présence de la matière est introduite pour la partie usinée. La valeur de la fonction de présence de la matière D est fixée à 0.9 ce qui donne, pour les éléments concernés, un module d'Young $E = 2.10^{10} MPa$ et une masse volumique $\rho = 780 kgm^{-3}$.

Sur la figure 5.3, nous avons représenté à gauche le maillage complet de la pièce, au centre le *modèle 1* et sur la partie droite, la partie grisée correspond aux éléments "endommagés" du *modèle 2*.

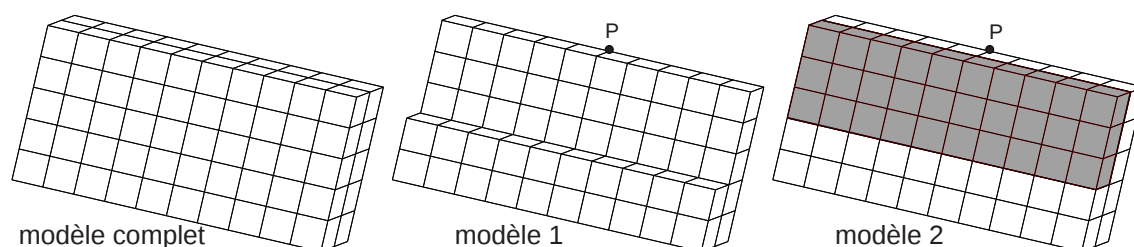


FIG. 5.3 – Maillage complet de la pièce et des deux modèles

Nous donnons dans le tableau 5.2, les cinq premières fréquences propres pour les deux modèles.

N° du mode	Modèle 1 en Hz	Modèle 2 en Hz	Ecart relatif en %
1	2580	2529,6	1,95
2	3221,6	3205,9	0,49
3	5063,3	5119,8	1,11
4	8272,1	8497,8	2,73
5	13087	13723	4,86

TAB. 5.2 – Fréquences des cinq premiers modes propres

Les fréquences propres sont voisines. L'écart relatif est inférieur à 5 %. Les modes ont une forme similaire. Il n'y a pas d'apparition de modes parasites en basse fréquence. On peut donc bien se servir de la fonction de présence de la matière comme un "effacement de matière". Le fait de diminuer la valeur du module d'Young et de la masse volumique de certains éléments est un moyen de rendre ces derniers comme usinés et de ne plus les considérer dans le calcul éléments finis.

5.2 Mise à jour approchée de la matrice $\hat{K}$

On se place ici dans le cas où l'enlèvement de matière est réalisé par introduction de la fonction de présence de la matière décrite dans le paragraphe 5.1. De plus, le cas étudié est celui du paragraphe 5.1.3 où il est possible de définir l'avancement de l'usinage, à chaque instant t , grâce à un unique scalaire $p(t)$. La position théorique outil/pièce $\underline{p}(t)$ et la fonction d'histoire $h(t)$ se déduisent de façon immédiate de $p(t)$.

5.2.1 La démarche

On rappelle brièvement le schéma d'intégration numérique (de Newmark, Δt constant) utilisé. La valeur de $\underline{q}$ à l'instant $t + \Delta t$ est ainsi donnée par :

$$\underline{\hat{K}}(p(t + \Delta t)) \quad {}^{t+\Delta t}\underline{q} = {}^{t+\Delta t}\underline{Q} + {}^t\bar{\bar{\underline{Q}}} \quad (5.4)$$

En posant :

$$\underline{q} = {}^{t+\Delta t}\underline{q} \quad ; \quad p = p(t + \Delta t) \quad ; \quad \underline{Q}_{total} = {}^{t+\Delta t}\underline{Q} + {}^t\bar{\bar{\underline{Q}}} \quad (5.5)$$

L'expression (5.4) peut se réécrire sous la forme suivante :

$$\underline{\hat{K}}(p) \quad \underline{q} = \underline{Q}_{total} \quad \text{ou encore} \quad \underline{q} = \underline{\hat{K}}^{-1}(p) \quad \underline{Q}_{total} \quad (5.6)$$

La méthode consiste à définir, a priori, en fonction des étapes d'usinage, des intervalles de temps $I_i =]t_{i-1}, t_i]$ où $t_i = t_{i-1} + n_i \Delta t$. Le paramètre n_i est choisi le plus grand possible en tenant compte de la plus ou moins grande variation des caractéristiques mécaniques due à l'enlèvement de matière théorique. Deux approches de calcul ont été retenues : une approche linéaire et une approche quadratique.

Approche linéaire : à chaque étape i , l'inverse de la matrice $\underline{\hat{K}}$ est approximée par la combinaison linéaire de deux matrices évaluées aux instants t_{i-1} et t_i .

$$\underline{\hat{K}}^{-1}(p)_{approché} = (1 - \eta) \underline{\hat{K}}^{-1}(p_{i-1}) + \eta \underline{\hat{K}}^{-1}(p_i) \quad (5.7)$$

Approche quadratique : à chaque étape i , l'inverse de la matrice $\underline{\hat{K}}$ est approximée par la combinaison de trois matrices évaluées aux instants t_{i-1} , $t_{i-\frac{1}{2}}$ et t_i .

$$\underline{\hat{K}}^{-1}(p)_{approché} = (1 - \eta)(1 - 2\eta) \underline{\hat{K}}^{-1}(p_{i-1}) + 4\eta(1 - \eta) \underline{\hat{K}}^{-1}(p_{i-\frac{1}{2}}) + \eta(2\eta - 1) \underline{\hat{K}}^{-1}(p_i) \quad (5.8)$$

Dans les deux expressions précédentes on a : $\eta = \frac{t-t_{i-1}}{n_i \Delta t}$ et $p_j = p(t_j)$.

En posant :

$$\underline{q}_{i-1} = \underline{\hat{K}}^{-1}(p_{i-1}) \underline{Q}_{total} \quad (5.9)$$

$$\underline{q}_{i-\frac{1}{2}} = \underline{\hat{K}}^{-1}(p_{i-\frac{1}{2}}) \underline{Q}_{total} \quad (5.10)$$

$$\underline{q}_i = \underline{\hat{K}}^{-1}(p_i) \underline{Q}_{total} \quad (5.11)$$

selon l'approche retenue, on obtient $\underline{q}_{approché}$ de façon évidente :

$$\underline{q}_{approché} = (1 - \eta) \underline{q}_{i-1} + \eta \underline{q}_i \quad (5.12)$$

ou

$$\underline{q}_{approché} = (1 - \eta)(1 - 2\eta) \underline{q}_{i-1} + 4\eta(1 - \eta) \underline{q}_{i-\frac{1}{2}} + \eta(2\eta - 1) \underline{q}_i \quad (5.13)$$

Il est à noter que les inverses des différentes matrices $\underline{\hat{K}}$ ne sont pas calculées. Elles sont uniquement décomposées (factorisation de Crout $\underline{\underline{L}} \cdot \underline{\underline{D}} \cdot \underline{\underline{L}}^T$).

5.2.2 Validation de la méthode

Le cadre de travail choisi pour la validation de cette méthode est le même que celui du paragraphe 5.1 pour la prise en compte simplifiée de l'enlèvement de matière. Nous cherchons ici à montrer qu'une utilisation de la forme interpolée de la factorisation de la matrice $\underline{\hat{K}}$ permet de bien suivre l'évolution du comportement dynamique de la pièce au fil de l'usinage. Contrairement à ce qui se passe pendant un usinage réel, la démarche de validation de la méthode est effectuée en utilisant un effort de coupe parfaitement connu : la force excitatrice apparaissant dans le vecteur $\underline{Q}(t)$ qui est retenue est de la forme $F(t) = \cos(\omega t)$. Cette force schématise l'excitation de l'outil sur la pièce. Elle est mobile, appliquée sur la partie supérieure de la pièce, et suit l'avancement de l'usinage.

Simulations

Afin de valider la démarche, nous avons réalisé trois simulations différentes :

- *Simulation 1* : les matrices EF sont recalculées après chaque pas de temps dt en fonction du paramètre $p(t)$.
- *Simulation 2* : les matrices EF sont recalculées à des pas de temps plus espacés, interpolation pour les pas de temps intermédiaires par l'*approche linéaire*.
- *Simulation 3* : les matrices EF sont recalculées à des pas de temps plus espacés, interpolation pour les pas de temps intermédiaires par l'*approche quadratique*.

La *simulation 1* est donc destinée à donner des résultats de référence pour les deux simulations suivantes.

La mise en œuvre des approches simplifiées est validée pour des étapes d'usinage plus ou moins longues.

Paramètres communs aux simulations

Avant de commencer les simulations, il faut fixer quelques paramètres, comme f la fréquence pour la force schématisant l'excitation de l'outil sur la pièce, le coefficient d'amortissement α , la vitesse d'avancement et le pas de temps Δt .

Nous choisissons $f = 3342,68Hz$ qui est la fréquence du premier mode propre de vibration de la pièce non encore usinée avec le maillage MF pour caractériser la force excitatrice ($\omega = 21002,66rad/s$).

Nous adoptons comme précédemment une matrice d'amortissement $\underline{C}$ proportionnelle à la matrice $\underline{K}$ permettant d'assurer un taux d'amortissement de 0,01 pour le premier mode propre de vibration de la pièce non usinée. Ceci nous donne $\alpha = 2(0,01/\omega) \sim 9.10^{-7}$.

La vitesse d'usinage (vitesse d'avancement) utilisée lors des simulations est de l'ordre de 1m/s ce qui n'est pas très représentatif d'une vitesse dans des conditions réelles qui est un peu au-dessous de 0.1m/s. Ce choix a été fait pour limiter le temps de calcul.

Pour parcourir la pièce sur toute sa longueur, nous avons besoin de 10000 incréments, pour un pas de temps Δt égal à 10^{-4} .

Pseudo-code des différentes simulations

Simulation 1

Cette première simulation, représentée par le pseudo-code Alg. 5.1, permet d'avoir des résultats de référence pour valider les deux approches : linéaire et quadratique.

Les différentes étapes avec l'endommagement pris en compte à chaque pas de temps Δt

début

Initialisations

$t_D := t_0$

Tant que ($t_D < t_{max}$)

$t_F := t_D + \Delta t$

$p_F := p(t_F)$

Calcul de $\underline{\underline{M}}(p_F)$, $\underline{\underline{K}}(p_F)$

Calcul de $\hat{\underline{\underline{K}}}(p_F)$

Décomposition de $\hat{\underline{\underline{K}}}(p_F)$ en $dec\hat{\underline{\underline{K}}}(p_F) := \underline{\underline{L}} * \underline{\underline{D}} * \underline{\underline{L}}^T$

Calcul de $\underline{\underline{Q}}_{total}(t_F)$

Application du schéma de Newark à t_F , on obtient $q_F := (dec\hat{\underline{\underline{K}}}(p_F))^{-1} \underline{\underline{Q}}_{total}(t_F)$

$t_D := t_F$

fin Tant que

fin

Alg. 5.1: Réactualisation à chaque pas de temps Δt

Simulation 2 et 3

Les grandes étapes de la programmation de la *simulation 2* sont décrites dans le pseudo-code Alg. 5.2. Pour la *simulation 3*, les étapes de la programmation sont très proches de celles de la *simulation 2*.

En effet, pour l'approche quadratique, il faut en plus : décomposer la matrice $\hat{\underline{\underline{K}}}(p_{i-\frac{1}{2}})$, stocker la décomposition de cette dernière et calculer $q_{i-\frac{1}{2}}$ (équation (5.10)).

Les différentes étapes avec l'endommagement pris en compte à chaque étape d'usinage Δt

début

Initialisations

$T_D := T_0$

Calcul de $\underline{\underline{M}}(p_F)$, $\underline{\underline{K}}(p_F)$

Calcul de $\underline{\underline{\hat{K}}}(p_F)$

Décomposition de $\underline{\underline{\hat{K}}}(p_F)$ en $dec\underline{\underline{\hat{K}}}(p_F) := \underline{\underline{L}}.\underline{\underline{D}}.\underline{\underline{L}}^T$

Tant que ($T_D < T_{max}$)

$T_F := T_D + \Delta T$

$p_F := p(T_F)$

$dec\underline{\underline{\hat{K}}}(p_F)$ devient $dec\underline{\underline{\hat{K}}}(p_D)$

$\underline{\underline{M}}(p_F)$ devient $\underline{\underline{M}}(p_D)$

$\underline{\underline{K}}(p_F)$ devient $\underline{\underline{K}}(p_D)$

Calcul de $\underline{\underline{M}}(p_F)$, $\underline{\underline{K}}(p_F)$

Calcul de $\underline{\underline{\hat{K}}}(p_F)$

Décomposition de $\underline{\underline{\hat{K}}}(p_F)$ en $dec\underline{\underline{\hat{K}}}(p_F)$

Tant que ($t_D < T_F$)

$t_F := t_D + \Delta t$

$p := p(t_F)$

Calcul de $\alpha := (p - p_D)/(p_F - p_D)$

Calcul de $\underline{Q}_{total}(t_F)$

Application du schéma de Newark à t_D , on obtient $q_D := (dec\underline{\underline{\hat{K}}}(p_D))^{-1} \underline{Q}_{total}$

Application du schéma de Newark à t_F , on obtient $q_F := (dec\underline{\underline{\hat{K}}}(p_F))^{-1} \underline{Q}_{total}$

Calcul de l'approche linéaire $q_P := (1 - \alpha)q_D + \alpha q_F$

$t_D := t_F$

fin Tant que

$T_D := T_F$

fin Tant que

fin

Alg. 5.2: Réactualisation avec l'approche linéaire

Résultats

Pour comparer les résultats des deux approches, linéaire et quadratique, par rapport aux résultats de la solution de référence, nous avons calculé une erreur relative $E(t)$ (équation (5.14)) en un point P de la pièce appartenant au maillage. Le point P est choisi dans la partie non usinée de la pièce, il est pris sur la partie supérieure de la pièce au milieu (figure 5.3). Nous avons une représentation de l'évolution des déplacements en ce

point sur la figure 5.4.

$$E(t) = \frac{|q_{Ref} - q(t)|}{q_{RefMax}} \quad (5.14)$$

Nbre d'étapes	Nbre d'incréments/étape	$E(t)$ Linéaire	$E(t)$ Quadratique
100	100	$1,975.10^{-2}$	$1,779.10^{-4}$
50	200	$1,077.10^{-1}$	$3,339.10^{-2}$
20	500	$4,945.10^{-1}$	$8,006.10^{-2}$
10	1000	1,660	$2,836.10^{-1}$
5	2000	5,260	1,747
1	10000	63,111	9,514

TAB. 5.3 – Erreur relative maximale (en %)

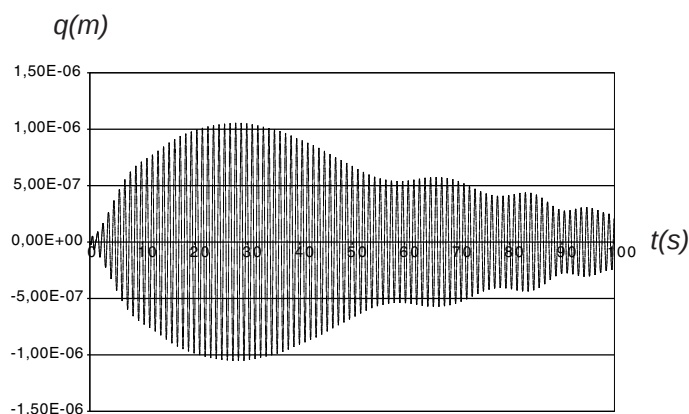


FIG. 5.4 – Evolution du déplacement au point P

Le tableau 5.3 regroupe les erreurs relatives maximales $E(t)$ au point P pour l'approche linéaire et quadratique. Que ce soit l'approche linéaire ou quadratique, à partir de 100 incréments par étape, nous obtenons des erreurs relatives faibles (2% en linéaire et 0,02% en quadratique) ce qui démontre l'intérêt de l'approche.

Il faut bien rappeler que l'intérêt majeur de l'interpolation de $\hat{K}$ est de diminuer, uniquement, le nombre de factorisations. Cette diminution des factorisations s'accompagne cependant d'une augmentation des descentes-montées. Le nombre de descente-montée double pour l'approche linéaire, et triple pour l'approche quadratique à chaque incrément. Le tableau 5.4 regroupe pour les trois simulations, le nombre de factorisations et de descente-montée par incrément et par étape.

	<i>Simulation 1</i>	<i>Simulation 2</i>	<i>Simulation 3</i>
Par étape		1 factorisation	2 factorisations
Par incrément	1 factorisation + 1 descente-montée	2 descentes-montées	3 descentes-montées

TAB. 5.4 – Nombre de factorisations et de descentes-montées par incrément et par étape

Il est possible de comparer le nombre de factorisations et de descentes-montées entre les trois simulations sur un cas précis. Pour 100 étapes de 100 incréments, nous avons :

<i>Simulation 1</i>	10000 factorisations + 10000 descentes-montées
<i>Simulation 2</i>	100 factorisations + 20000 descentes-montées
<i>Simulation 3</i>	200 factorisations + 30000 descentes-montées

TAB. 5.5 – Exemple pour 100 étapes de 100 incréments

Le temps de calcul est de huit heures pour la *simulation 1* sur un *PowerPCG3* (pour effectuer 10000 incréments). Pour 100 étapes de 100 incréments, le temps de calcul avec l'approche linéaire est d'environ de quarante-cinq minutes. On a réussi à réduire considérablement le temps de calcul, environ 10 fois plus rapide. Nous avons alors décidé d'augmenter le nombre de sous incréments tout en notant le temps de calcul nécessaire. Il nous est apparu qu'à partir de 50 étapes de 200 incréments, la simulation de l'usinage durait vingt-cinq minutes (soit environ 20 fois plus rapide que la *simulation 1*) et que pour un nombre d'incrément plus important par étape la durée du calcul ne diminuait plus. Cela signifie que les vingt-cinq minutes correspondent au coût estimé pour effectuer les deux descentes-montées dans le cas linéaire.

Le gain en précision de l'approche quadratique n'est, dans l'exemple, pas suffisant pour compenser les coûts des calculs supplémentaires.

Nombre d'étapes	Gain du coût Linéaire	Gain du coût Quadratique
100 ($\times 100$ incréments)	10,6	5,3
50 ($\times 200$ incréments)	19,2	9,6
20 ($\times 500$ incréments)	19,2	9,6

TAB. 5.6 – Gain estimé du coût pour 10000 incréments, par rapport à la *simulation 1*

Il est à noter que ces conclusions se basent sur des coûts numériques pour un modèle éléments finis de 495 DDL. Plus la taille du modèle EF est importante, plus les coûts liés à la décomposition deviennent prépondérants vis à vis de la descente-montée. L'approche de $\hat{K}$ linéarisée n'en sera que plus pertinente.

Dans une simulation réelle de l'usinage, il faudra aussi tenir compte du coût non négligeable des nombreuses intersections.

5.3 Conclusion

La simulation numérique de l'usinage en présence d'un important enlèvement de matière nécessite la prise en compte des variations de masse et de flexibilité. Ceci nécessite une actualisation des matrices de masse et de rigidité, même si nous retenons l'approche simplifiée que nous proposons ici et qui ne nécessite pas de remaillage. La connaissance a priori de la trajectoire relative théorique outil/pièce permet en effet d'anticiper et de prédire les modifications des matrices de masse et de rigidité entre deux intervalles de temps. Une fois cette observation faite, nous avons montré qu'il est possible de simplifier considérablement les calculs en effectuant une interpolation de l'inverse de la matrice apparaissant dans le schéma d'intégration numérique.

Une interpolation linéaire ou quadratique des matrices inverses correspondant à l'évolution prévisible du domaine pièce d'un instant à l'autre, qui se traduit tout simplement ensuite par la même interpolation des solutions de deux ou trois systèmes linéaires, a donc été imaginée, mise en place et testée.

La méthode, mise en place dans *Nessy* s'avère très prometteuse étant donné les diminutions des temps de calcul obtenues avec des résultats suffisamment précis. Cette mise en place s'est actuellement effectuée séparément de la simulation de l'usinage proprement dite.

La méthode sera très utile pour réaliser les simulations de pièces minces avec fort enlèvement de matière. Ceci n'a pas pu être effectué dans le temps imparti à ce travail mais pourra très facilement être intégré aux calculs présentés dans les chapitres précédents.

L'objectif des travaux à venir sera donc de réaliser des simulations d'usinage en utilisant notre stratégie de réactualisation des matrices, et de valider à la fois la qualité des résultats obtenus et les gains en temps de calcul.

Conclusion et Perspectives

L'objectif de notre travail était la mise en place d'une méthode de simulation de l'usinage à l'échelle macroscopique qui tienne compte des déformations de la pièce dans la zone usinée. Une première simulation autorisant cette fonctionnalité a été montrée. Pour pouvoir la mener, une nouvelle définition de l'enlèvement de matière a du être introduite ainsi qu'une nouvelle description géométrique du volume de la pièce. Cette nouvelle description a permis de mettre en place des algorithmes géométriques robustes et a conduit à l'utilisation d'une section de coupe moyenne pour le calcul des efforts de coupe. Une prise en compte simplifiée de l'altération de la masse et de la rigidité de la pièce en cours d'usinage a enfin été proposée.

Un important apport de ce travail est qu'il est désormais possible, grâce à l'introduction des dexels, de réaliser des simulations sans arrêts intempestifs. Une campagne d'essais systématiques pourra enfin être menée et, nous l'espérons, permettra de valider notre approche d'une façon plus décisive. Pour cette validation, une attention particulière devra être portée sur l'identification d'une loi de coupe appropriée, qui est l'une des données fondamentales du logiciel.

Il est d'ores et déjà possible d'envisager des perfectionnements à l'approche. Parmi ceux-ci nous pouvons citer :

- la prise en compte du talonnage,

Le talonnage est fréquemment rencontré lorsqu'il y a broutement. L'effort de contact entre la face de dépouille d'une dent et la pièce conduit à un effort tendant à les éloigner l'un l'autre bien. Cet effort de contact, qui peut être bien supérieur aux forces de coupe, joue alors un rôle majeur dans la dynamique de l'usinage. Dans un premier temps, la détection de la présence de talonnage peut déjà être intéressante.

- la possibilité de traiter des pièces de forme complexe,

Il s'agit ici principalement d'un problème d'entrée de donnée et d'infrastructure de logiciel car l'approche actuelle, dans son principe, peut s'adapter à toute forme de pièce.

- la simulation d'opérations de tournage.

Les dexels ont pour principal inconvénient de proposer une description homogène du domaine occupé par la pièce (densité constante des dexels). Il ne faut ainsi pas exclure un retour à une représentation B-rep basée sur une surface composée de facettes planes triangulaires lorsqu'il existera, enfin, des algorithmes géométriques suffisamment robustes.

Bibliographie

- [ABC⁺02] S. Assouline, E. Beauchesne, G. Coffignal, P. Lorong, and A. Marty. Simulation numérique de l’usinage à l’échelle macroscopique : modèles dynamiques de la pièce. *Mécanique et Industries*, 3 :389–402, 2002.
- [AEF87] P.R. Atherton, C. Earl, and C. Fred. A graphical simulation system for dynamic five-axis nc verification. In *SME*, pages 2.1–2.12, 1987.
- [AES98] F. Abrari, M.A. Elbestawi, and A.D. Spence. On the dynamics of ball end milling : modeling of cutting forces and stability analysis. *International Journal of Machine, Tools and Manufacture*, 38(3) :215–237, 1998.
- [Bat82] K.J. Bathe. *Finite Element Procedures in Engineering Analysis*. Prentice-Hall, Englewood Cliffs, 1982.
- [Bat96] K.J. Bathe. *Finite Element Procedures*. Prentice-Hall, 1996.
- [Bea99] Erwan Beauchesne. *Simulation numérique de l’usinage à l’échelle Macroscopique : prise en compte d’une pièce déformable*. PhD thesis, Ecole Nationale Supérieure d’Arts et Métiers - CER de Paris, 1999.
- [CL03] G. Coffignal and P. Lorong. Un logiciel éléments finis pour développer et capitaliser des travaux de recherche. In *6ème Colloque National en Calcul des Structures*, mai 2003.
- [Dek95] Khaled Dekelbab. *Modélisation et simulation du comportement dynamique de l’ensemble Pièce-Outil-Machine en usinage par outil coupant*. PhD thesis, Ecole Nationale Supérieure d’Arts et Métiers - CER de Paris, 1995.
- [DKZ80] R.E. DeVor, W.A. Kline, and W.J. Zdeblick. A mechanistic model for the force system in end milling with application to machining frame. In *8th North American Manufacturing Research Conference*, pages 297–303, 1980.
- [EY01a] S. Engin and Y. Altintas. Mechanics and dynamics of general milling cutters. part i : helical end mills. *International Journal of Machine, Tools and Manufacture*, 41 :2195–2212, 2001.
- [EY01b] S. Engin and Y. Altintas. Mechanics and dynamics of general milling cutters. part ii : inserted cutters. *International Journal of Machine, Tools and Manufacture*, 41 :2213–2231, 2001.
- [GR96] M. Géradin and D. Rixen. *Théorie des vibrations. Application à la dynamique des structures*. Masson, 1996.
- [Gu94] Fangming Gu. *Prediction of cutting forces and surface errors in face milling with generalized cutter and workpiece geometry*. PhD thesis, University of Illinois at Urbana- champaign, 1994.
- [Hoo86] T. Van Hook. Real-time shaded nc milling display. In *SIGGRAPH’86*, pages 15–35, 1986.

- [IS97] F. Ismail and E. Soliman. A new method for the identification of stability lobes in machining. *International Journal of Machine, Tools and Manufacture*, 37(6) :763–774, 1997.
- [KCC00] G.M. Kim, P.J. Cho, and C.N. Chu. Cutting force prediction for sculptured surface ball-end milling using z-map. *International Journal of Machine, Tools and Manufacture*, 40 :277–291, 2000.
- [KS61] F. Koenigsberger and A.J.P. Sabberwal. An investigation into the cutting force pulsations during milling operations. *International Journal of Machine Tool, Design and Research*, 1 :15–33, 1961.
- [LAC00] P. Lorong, F. Ali, and G. Coffignal. Research oriented software development platform for structural mechanics : a solution for distributed computing. In *Second International Conference on Engineering Computational Technology, Developments in engineering computational technology*, pages 93–100, 6-8 September 2000.
- [Lap03] François Lapujoulade. *Simulation des phénomènes vibratoires intervenant en usinage- Etat de recherches et perspectives*. PhD thesis, Université de Lyon, 2003.
- [MA91] D. Montgomery and Y. Altintas. Mechanism of cutting force and surface generation in dynamic milling. *ASME Trans.-Journal of Engineering for Industry*, 113 :160–168, May 1991.
- [Mar41] M.E. Martellotti. An analysis of the milling process. *ASME Trans.-Journal of Engineering for Industry*, 63 :667–700, 1941.
- [Mar45] M.E. Martellotti. An analysis of the milling process, partii. : Down milling. *ASME Trans.-Journal of Engineering for Industry*, 67 :233–251, 1945.
- [Mar03] Audrey Marty. *Simulation numérique de l’usinage par outil coupant à l’échelle Macroscopique : contribution à la définition géométrique de la surface usinée*. PhD thesis, Ecole Nationale Supérieure d’Arts et Métiers - CER de Paris, 2003.
- [MHK01] Y. Mizugaki, M. Hao, and K. Kikkawa. Geometric generating mechanism of machined surface by ball-nosed end milling. In *Annals of CIRP*, volume 50, 2001.
- [MR92] J.P. Menon and D.M. Robinson. High performance nc verification via massively parallel raycasting. In *Symposium on Concurrent Engineering, ASME Winter Annual Meeting*, November 1992.
- [Pei04] Grégoire Peigne. *Etude et simulation des effets dynamiques de la coupe sur la stabilité de la coupe et la qualité géométrique de la surface usinée : application au fraisage de profil*. PhD thesis, Institut National Polytechnique de Grenoble, 2004.
- [Pol92] Olivier Polit. *développement d’éléments finis de plaque semi-épaisse et de coque semi-épaisse à double courbure*. PhD thesis, Université de Paris VI, 1992.
- [RLHB04] S. Ratchev, S. Liu, W. Huang, and A.A. Becker. Milling error prediction and compensation in machining of low-rigidity parts. *International Journal of Machine, Tools and Manufacture*, 44 :1629–1641, 2004.
- [Sab62] A.J.P. Sabberwal. Chip section and cutting force during the milling operation. In *Annals of CIRP*, volume 10, pages 197–203, 1962.

- [TI81] J. Tlustý and F. Ismail. Basic non-linearity in machining chatter. In *Annals of CIRP*, volume 30, pages 299–304, 1981.
- [TTIS89] S. Takata, M.D. Tsai, M. Inui, and T. Sata. A cutting simulation system for machinability evaluation using a workpiece model. In *Annals of CIRP*, volume 38, pages 417–420, 1989.
- [VDK01] M. P. Volger, R. E. DeVor, and S. G. Kapoor. Nonlinear influence of effective lead angle in turning process stability. *ASME Trans.- Journal of Manufacturing Science and Engineering*, 124 :473–475, 2001.
- [WW86a] W.P. Wang and K.K. Wang. Geometric modeling for swept volume of moving solids. *IEEE Computer Graphics and Applications*, 6 :8–25, 1986.
- [WW86b] W.P. Wang and K.K. Wang. Real-time verification of multiaxis nc programs with raster graphics. In *IEEE International Conference on Robotics and Automation*, pages 166–237, 1986.
- [WZ04] K. Weinert and A. Zabel. Simulation based tool wear prediction in milling of sculptured surfaces. In *Annals of CIRP*, volume 53, pages 217–223, 2004.

Annexe A

Pseudo Algorithmes

A.1 Algorithmes liés aux dexels

A.1.1 Détermination du dexel se trouvant juste avant une valeur z donnée

DonneDexelJusteAvantZ
Entrée : i, j, z
début $dexel := MatricePremierDexel(i,j)$ $dexelAvantZ := nil$ Tant que ($dexel \neq nil$) $zMax := EtendueZduDexel(2,dexel)$ Si ($zMax \leq z$) alors $dexelAvantZ := dexel$ $dexel := DonneDexelSuivant(dexel)$ Sinon Quitter fin Si fin Tant que $DonneDexelJusteAvantZ := dexelAvantZ$ fin

Alg. A.1: Algorithme permettant de déterminer le dexel se trouvant juste avant une valeur z donnée

A.1.2 Création d'un dixel

CreerUnDixel
Entrée : $dixel, i, j, zMin, zMax$
début $IetJduDixel(1,dixel) := i$ $IetJduDixel(2,dixel) := j$ $EtendueZduDixel(1,dixel) := zMin$ $EtendueZduDixel(2,dixel) := zMax$ $dixelAvant := nil$ $dixelAprès := nil$ Si ($MatricePremierDixel(i,j) := nil$) alors $MatricePremierDixel(i,j) := dixel$ Sinon $dixelAvant := \text{DonneDixelJusteAvantZ}(i,j,zMin)$ Si ($dixelAvant \neq nil$) alors $dixelAprès := \text{DonneDixelSuivant}(dixelAvant)$ $\text{DonneDixelSuivant}(dixelAvant) := dixel$ Sinon $dixelAprès := \text{MatricePremierDixel}(i,j)$ $\text{MatricePremierDixel}(i,j) := dixel$ fin Si fin Si $\text{DonneDixelSuivant}(dixel) := dixelAprès$ fin

Alg. A.2: Création d'un dixel

A.1.3 Suppression d'un dixel

EliminerUnDixel**Entrée :** *dixel***début***i* := *IetJduDixel*(1,*dixel*)*j* := *IetJduDixel*(2,*dixel*)*zMin* := *EtendueZduDixel*(1,*dixel*)*dixelAvant* := *nil**dixelAprès* := *DonneDixelSuivant*(*dixel*)*dixelEstLePremier* := (*dixel* = *MatricePremierDixel*(*i,j*))*dixelEstLeDernier* := (*dixelAprès* := *nil*)*dixelEstSeul* := (*dixelEstLePremier* **et** *dixelEstLeDernier*)**Si** (*dixelEstSeul*) **alors***MatricePremierDixel*(*i,j*) := *nil***Sinon****Si** (*dixelEstLePremier*) **alors***MatricePremierDixel*(*i,j*) := *dixelAprès***Sinon***dixelAvant* := **DonneDixelJusteAvantZ** (*i,j,zMin*)*DonneDixelSuivant*(*dixelAvant*) := *dixelAprès***fin Si****fin Si****fin**

Alg. A.3: Elimination d'un dixel

A.2 Algorithmes d'intersection

A.2.1 Première méthode d'intersection

Intersection entre les facettes d'une partie et les dexels
Entrée : <i>ListeFacetteDuVolumeBalayé</i>

début

Pour (chaque facette de la *ListeFacetteDuVolumeBalayé* : *facette*)

Récupérer la Normale Extérieure de la *facette*

Récupérer les coordonnées des trois sommets de la *facette* :

CoorSommets(dim3,3)

CreerUneBoiteEnglobante <i>entrée</i> : <i>CoorSommets(dim3,3)</i> <i>sortie</i> : <i>BoiteEnglobante(dim3,2)</i>
--

Récupérer *iMin, iMax, jMin, jMax, kMin* et *kMax* de la *BoiteEnglobante* de la *facette*

Se limiter aux dexels appartenant à la *BoiteEnglobante* de la *facette* :

Pour (chaque indice du support suivant *X* compris entre[*iMin, iMax*] : *i*)

Pour (chaque indice du support suivant *Y* compris entre[*jMin, jMax*] : *j*)

dexel := *MatricePremierDexel(i,j)*

Tant que (*dexel* ≠ nil)

Si (*EtendueZduDexel(2,dexel)* ≤ *kMin*) **alors**

Quitter la condition Si

Sinon

Si (*EtendueZduDexel(1,dexel)* ≥ *kMax*) **alors**

Quitter la condition Si

Sinon

GereIntersectionFacetteDexel <i>entrée</i> : <i>facette, dexel</i> <i>sortie</i> : <i>zInter, intersectionTrouvée</i>
--

Si (*intersectionTrouvée* := vrai) **alors**

Supprimer provisoirement le *dexel* et créer deux nouveaux dexels

fin Si

fin Si

fin Si

fin Tant que

dexel := *DonneDexelSuivant(dexel)*

fin Pour

fin Pour

fin Pour

fin

A.2.2 Deuxième méthode d'intersection

Intersection entre les facettes d'une partie et les supports des dexels
Entrée : <i>ListeTriaDUnePartie</i>
Sortie : <i>minSupport, maxSupport</i>
début <i>nbSupportCoupe</i> := 0 Pour (chaque facette de la <i>ListeTriaDUnePartie</i> : <i>facette</i>) Pour (chaque support compris dans la boîte englobante de la <i>facette</i> : <i>support</i>) GereIntersectionFacetteSupport <i>entrée</i> : <i>facette, support</i> <i>sortie</i> : <i>zInter, intersectionTrouvee</i> Si (<i>intersectionTrouvee</i> := vrai) alors Si (<i>nbSupportCoupe</i> = 0) alors <i>nbSupportCoupe</i> := <i>nbSupportCoupe</i> + 1 <i>minSupport</i> := <i>zInter</i> <i>maxSupport</i> := <i>zInter</i> Sinon <i>minSupport</i> = MIN(<i>zInter, minSupport</i>) <i>maxSupport</i> = MAX(<i>zInter, maxSupport</i>) fin Si fin Si fin Pour fin

Alg. A.5: Algorithme de la deuxième méthode d'intersection

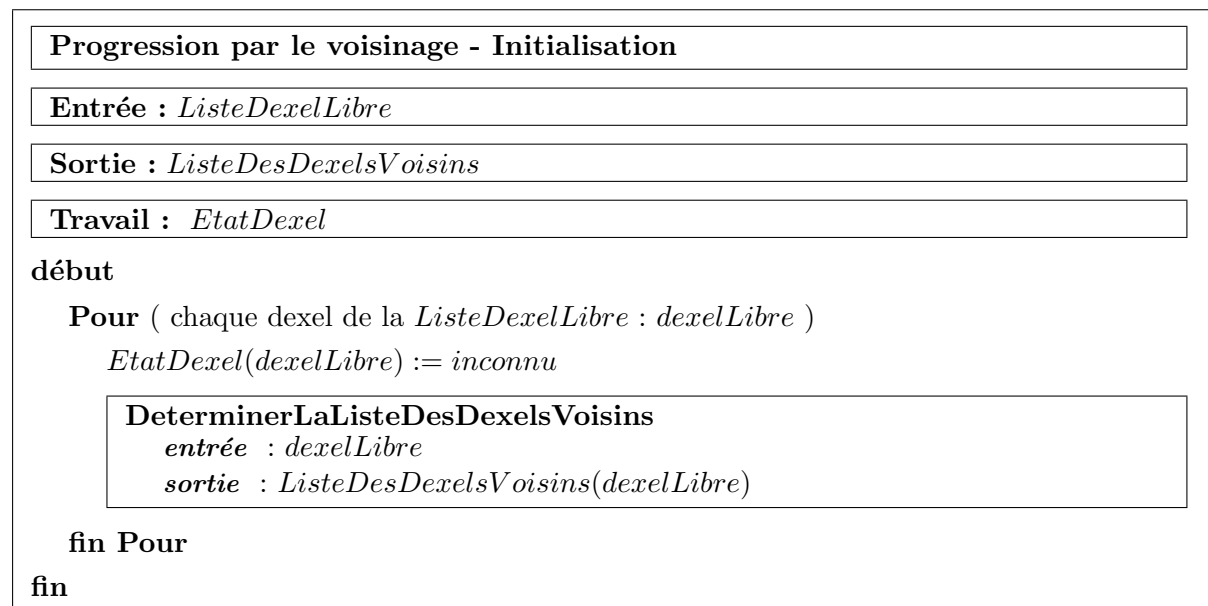
A.3 Algorithmes de progression par le voisinage

A.3.1 Définition du voisinage

DeterminerLaListeDesDexelsVoisins
Entrée : <i>dexel</i>
Sortie : <i>ListeDesDexelsVoisins(nbDexelVoisin)</i>
début $i := IetJduDexel(1, dexel)$ $j := IetJduDexel(2, dexel)$ $zMin := EtendueZduDexel(1, dexel)$ $zMax := EtendueZduDexel(2, dexel)$ $dexelVoisin := MatricePremierDexel(iVoisin, jVoisin)$ $zMinVoisin := EtendueZduDexel(1, dexelVoisin)$ $zMaxVoisin := EtendueZduDexel(2, dexelVoisin)$ $nbDexelVoisin := 0$ Tant que ($dexelVoisin \neq nil$) Si ($zMaxVoisin \leq zMin$) alors Quitter le bloc Si Sinon Si ($zMinVoisin \geq zMax$) alors Quitter le bloc Si Sinon $nbDexelVoisin := nbDexelVoisin + 1$ $ListeDesDexelsVoisins(nbDexelVoisin) := dexelVoisin$ fin Si fin Si $dexelVoisin := DonneDexelSuivant(dexelVoisin)$ fin Tant que fin

Alg. A.6: Détermination de la liste des dexels voisins d'un dexel

A.3.2 Mise en œuvre de la progression par le voisinage



Alg. A.7: Progression par le voisinage : Initialisation

Progression par le voisinage
Entrée : <i>ListeDexelLibre, ListeDesDexelsVoisins</i>
Travail : <i>EtatDexel</i>
début Tant que (<i>ilResteDesDexelsAtraiter = vrai</i>) <i>nbDexelChange</i> := 0 Pour (chaque dexel de la <i>ListeDexelLibre</i> : <i>dexelLibre</i>) Etape 1 Si (<i>EtatDexel(dexelLibre) = inconnu</i>) alors Si (il existe un voisin de <i>dexelLibre</i> qui n'appartient pas à la <i>ListeDexelLibre</i>) alors <i>EtatDexel(dexelLibre)</i> := <i>fixeNonTraite</i> <i>nbDexelChange</i> := <i>nbDexelChange</i> + 1 fin Si fin Si Etape 2 Si (<i>EtatDexel(dexelLibre) := fixeNonTraite</i>) alors Si (tous les dexels voisins du <i>dexelLibre</i> appartiennent à la <i>ListeDexels</i>) alors Pour (chaque dexel voisin du <i>dexelLibre</i> : <i>dexelVoisin</i>) Si (<i>EtatDexel(dexelVoisin) ≠ fixeTraite</i>) alors <i>EtatDexel(dexelVoisin)</i> := <i>fixeNonTraite</i> <i>nbDexelChange</i> := <i>nbDexelChange</i> + 1 fin Si fin Pour <i>EtatDexel(dexelLibre)</i> := <i>fixeTraite</i> fin Si fin Si fin Pour <i>ilResteDesDexelsAtraiter</i> := <i>nbDexelChange</i> ≠ 0 fin Tant que fin

Alg. A.8: Progression par le voisinage

Progression par le voisinage - Conclusion
Entrée : <i>ListeDexelLibre</i>
Travail : <i>EtatDexel</i>
début Pour (chaque dexel de la <i>ListeDexelLibre</i> : <i>dexelLibre</i>) Si (<i>EtatDexel</i> (<i>dexelLibre</i>) = <i>inconnu</i>) alors EliminerUnDexel <i>entrée</i> : <i>dexelLibre</i> fin Si fin Pour fin

Alg. A.9: Progression par le voisinage : Conclusion

A.3.3 Calcul d'une boîte englobante

Une liste de facettes triangulaires planes permet de décrire la frontière du volume balayé par partie ${}^rV_{balayé}^i$ (domaine quasi convexe).

Cette liste *ListeFacetteDuVolumeBalayeParPartie* des facettes du volume balayé par une partie est l'union des trois listes de facettes suivantes :

- *ListeFacette* Γ_1 ,
- *ListeFacette* Γ_2 et
- *ListeFacette* Γ_3 .

Dans un premier temps, nous devons récupérer les coordonnées de tous les sommets de cette liste afin de calculer une boîte englobante de cette liste de sommets.

Nous connaissons les dimensions de la pièce et plus exactement celles de la partie rectangulaire du pavage où sont positionnés les dexels. L'épaisseur du pavage est comprise entre *zMinDuPavage* et *zMaxDuPavage*.

Il est alors très facile de savoir si le volume balayé par une partie se trouve en dehors de la zone de dexels ou non, c'est-à-dire si la boîte se trouve au-dessus ou au-dessous du pavage suivant l'axe des *Z*.

$$BoiteEnglobante(3, 1) = hMin$$

$$BoiteEnglobante(3, 2) = hMax$$

$$Si \quad zMinDuPavage > hMax \quad ou \quad si \quad zMaxDuPavage < hMin$$

$$Alors \text{ pas d'intersection}$$

Il reste maintenant à transformer les valeurs (*réelles*) de la boîte englobante suivant les directions 1 et 2 en *entiers* afin d'obtenir les valeurs *iMin*, *iMax*, *jMin* et *jMax* en tenant compte du nombre de supports de dexels *nbMaxSupportX* et *nbMaxSupportY*. Ces quatre valeurs vont permettre de déterminer les différents couples supports (*i, j*) qui

sont susceptibles d'être en intersection avec les facettes du volume balayé par une partie avec $i \in [iMin, iMax]$ et $j \in [jMin, jMax]$.

Les différentes étapes du calcul de la boîte englobante, à partir d'une liste de coordonnées de points, sont présentées dans le pseudo-code A.10.

Le calcul de la boîte englobante a été effectué pour le volume balayé par une partie mais dans certains cas et si cela est nécessaire, il sera possible de calculer une boîte englobante pour chaque facette.

CreerUneBoiteEnglobante
Entrée : <i>CoordonneesDesPoints(dimEsp,nbPoints)</i>
Sortie : <i>BoiteEnglobante(dimEsp,2)</i>
début Initialisation : Pour (chaque dimension de la <i>dimEsp</i> : <i>dim</i>) <i>BoiteEnglobante(dim,1) := tresGrandeValPostiveGsD</i> <i>BoiteEnglobante(dim,2) := tresGrandeValNegativeGsD</i> fin Pour Pour (chaque point des <i>nbPoints</i> : <i>point</i>) Pour (chaque dimension de la <i>dimEsp</i> : <i>dim</i>) <i>coor := CoordonneesDesPoints(dim,point)</i> <i>BoiteEnglobante(dim,1) = MIN(BoiteEnglobante(dim,1),coor)</i> <i>BoiteEnglobante(dim,2) = MAX(BoiteEnglobante(dim,2),coor)</i> fin Pour fin Pour fin

Alg. A.10: Création d'une boîte englobante

SIMULATION NUMERIQUE DE L'USINAGE A L'ECHELLE MACROSCOPIQUE : PRISE EN COMPTE D'UNE PIECE DEFORMABLE

RESUME :

L'idée de mettre en place un simulateur relativement général de la partie mécanique de l'usinage à l'échelle macroscopique est à l'origine de ce travail.

Le principal apport de ce travail est la prise en compte de pièces dont la zone concernée par l'usinage est flexible. Cette présence de déformations dans la pièce a nécessité une définition précise de l'enlèvement de matière. Cette définition se fonde sur une configuration dite matérielle, où la réalisation pratique de la simulation de l'usinage est effectuée.

Par ailleurs, les pièces flexibles étant généralement très minces, l'enlèvement de matière peut modifier de façon significative le comportement dynamique de la pièce (perte de masse et de raideur). Pour être fidèle, une simulation doit alors prendre en compte cette modification de façon régulière en cours d'usinage. Nous montrons que cette prise en compte ne peut dans certains cas être évitée, et nous proposons une stratégie limitant les coûts numériques induits par cette prise en compte.

Un autre apport du travail réalisé est l'utilisation d'un modèle géométrique différent de celui des approches précédentes du LMSP, pour représenter le volume occupé par la pièce. Avec le modèle géométrique de la pièce retenu dans les travaux précédents du LMSP (modèle B-Rep analogue aux modèles utilisés en CAO), de très nombreuses intersections de polyèdres conduisent à des facettes "dégénérées" dont la prise en compte dans les algorithmes d'intersection est loin d'être triviale et constitue actuellement un des points bloquants pour ce qui concerne la robustesse. La nouvelle technique adoptée est celle des Z-buffer où la pièce est modélisée par des dexels (ensemble de « bâtonnets » disposés sur une matrice régulière). Cette description permet de simplifier sensiblement les algorithmes d'intersection tout en les rendant plus robustes. Cette nouvelle description nous a conduit à revoir la démarche de modélisation de l'interaction outil/pièce et notamment la démarche de calcul des efforts de coupe. Cette modélisation s'intègre dans le logiciel *Nessy* du LMSP. Par les résultats obtenus, nous démontrons la faisabilité de notre approche et son potentiel d'analyse.

Mots-clés : *simulation, usinage, pièce flexible, vibrations, dexels, algorithmes d'intersection*

NUMERICAL SIMULATION OF MACHINING AT MACROSCOPIC SCALE : TAKING INTO ACCOUNT THE WORKPIECE DEFORMATIONS

ABSTRACT

The idea to set general simulator of the mechanical part of machining has given its origin to this work.

The principal contribution of this work is the taking into account of workpieces for which the zone concerned with machining is flexible. This presence of deformations in the workpiece requires a precise definition of the removal of matter. This definition is based on a configuration known as material, where the practical realization of the simulation of machining is carried out.

In addition, the flexible workpiece being generally very thin, the removal of matter can modify to a significant degree the dynamic behaviour of the workpiece (loss of mass and stiffness). To be accurate, a simulation must then take into account this modification in a regular way in the course of machining. We show that this taking into account cannot be avoided in certain cases, and we propose a strategy limiting the numerical costs induced by this taking into account.

Another contribution of this work is the use of a different geometrical model, to represent the volume occupied by the workpiece than in previous studies of LMSP. With the geometric models of the workpiece used in previous works at LMSP (B-Rep model like the models used in CAO) many difficulties to achieve a sufficient reliability occurred, in particular when the tool is passing many times on the same place (this is often the case in milling). The new adopted technique is that of the Z-buffer where the workpiece is modelled by dexels (set of "sticks" laid out on a regular matrix). This description makes it possible to simplify appreciably the algorithms of intersection while making them more robust. This new description led us to re-examine the interaction tool/workpiece and in particular the step of calculation of the cutting forces.

This work is implemented in the software *Nessy*, developed at LMSP. Simulations show the feasibility of our approach and its potential of analysis.

Keywords: *simulation, machining, thin-walled pieces, vibrations, dexels, intersection algorithms*

