



HAL
open science

Analyse isogéométrique multiéchelle à précision contrôlée en mécanique des structures

Alexandre Chemin

► **To cite this version:**

Alexandre Chemin. Analyse isogéométrique multiéchelle à précision contrôlée en mécanique des structures. Mécanique des structures [physics.class-ph]. INSA de Lyon, 2015. Français. NNT : 2015ISAL0098 . tel-01367583

HAL Id: tel-01367583

<https://theses.hal.science/tel-01367583>

Submitted on 9 Mar 2017

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.

THÈSE

Analyse isogéométrique multiéchelle à précision contrôlée en mécanique des structures.

– Applications à la statique et à la dynamique transitoire –

Présentée devant

l'Institut National des Sciences Appliquées de Lyon

pour obtenir

le GRADE DE DOCTEUR

École doctorale :

Mécanique, Énergétique, Génie Civil, Acoustique

Spécialité :

MÉCANIQUE - GÉNIE MÉCANIQUE - GÉNIE CIVIL

par

Alexandre CHEMIN

Agrégé de mécanique

Thèse soutenue le 9 novembre 2015 devant la Commission d'examen

Jury

AMINE AMMAR	Professeur	Rapporteur
DAVID RYCKELYNCK	Professeur	Rapporteur
JEAN-FRANÇOIS REMACLE	Professeur	Examineur
RENÉ DE BORST	Professeur	Examineur
MICHEL RAOUS	Professeur	Examineur
JEAN-CHARLES PASSIEUX	Maître de Conférences	Examineur
ANTHONY GRAVOUIL	Professeur	Directeur de thèse
THOMAS ELGUEDJ	Maître de Conférences HDR	Co-Directeur de thèse

LaMCoS - UMR CNRS 5259 - INSA de Lyon
20, avenue Albert Einstein, 69621 Villeurbanne Cedex (FRANCE)

**INSA Direction de la Recherche - Ecoles Doctorales - Quadriennal
2007-2010**

SIGLE	ECOLE DOCTORALE	NOM ET COORDONNEES DU RESPONSABLE
CHIMIE	<u>CHIMIE DE LYON</u> http://sakura.cpe.fr/ED206 M. Jean Marc LANCELIN Insa : R. GOURDON	M. Jean Marc LANCELIN Université Claude Bernard Lyon 1 Bât CPE 43 bd du 11 novembre 1918 69622 VILLEURBANNE Cedex Tél : 04.72.43 13 95 Fax : lancelin@hikari.cpe.fr
E.E.A.	<u>ELECTRONIQUE,</u> <u>ELECTROTECHNIQUE, AUTOMATIQUE</u> http://www.insa-lyon.fr/eea M. Alain NICOLAS Insa : C. PLOSSU ede2a@insa-lyon.fr Secrétariat : M. LABOUNE AM. 64.43 - Fax : 64.54	M. Alain NICOLAS Ecole Centrale de Lyon Bâtiment H9 36 avenue Guy de Collongue 69134 ECULLY Tél : 04.72.18 60 97 Fax : 04 78 43 37 17 eea@ec-lyon.fr Secrétariat : M.C. HAVGOUDOUKIAN
E2M2	<u>EVOLUTION, ECOSYSTEME,</u> <u>MICROBIOLOGIE, MODELISATION</u> http://biomserv.univ-lyon1.fr/E2M2 M. Jean-Pierre FLANDROIS Insa : H. CHARLES	M. Jean-Pierre FLANDROIS CNRS UMR 5558 Université Claude Bernard Lyon 1 Bât G. Mendel 43 bd du 11 novembre 1918 69622 VILLEURBANNE Cédex Tél : 04.26 23 59 50 Fax 04 26 23 59 49 06 07 53 89 13 e2m2@biomserv.univ-lyon1.fr
EDISS	<u>INTERDISCIPLINAIRE SCIENCES-</u> <u>SANTE</u> Sec : Safia Boudjema M. Didier REVEL Insa : M. LAGARDE	M. Didier REVEL Hôpital Cardiologique de Lyon Bâtiment Central 28 Avenue Doyen Lépine 69500 BRON Tél : 04.72.68 49 09 Fax :04 72 35 49 16 Didier.revel@creatis.uni-lyon1.fr
INFOMATHS	<u>INFORMATIQUE ET</u> <u>MATHEMATIQUES</u> http://infomaths.univ-lyon1.fr M. Alain MILLE	M. Alain MILLE Université Claude Bernard Lyon 1 LIRIS - INFOMATHS Bâtiment Nautibus 43 bd du 11 novembre 1918 69622 VILLEURBANNE Cedex Tél : 04.72. 44 82 94 Fax 04 72 43 13 10 infomaths@bat710.univ-lyon1.fr - alain.mille@liris.cnrs.fr
Matériaux	<u>MATERIAUX DE LYON</u> M. Jean Marc PELLETIER Secrétariat : C. BERNAVON 83.85	M. Jean Marc PELLETIER INSA de Lyon MATEIS Bâtiment Blaise Pascal 7 avenue Jean Capelle 69621 VILLEURBANNE Cédex Tél : 04.72.43 83 18 Fax 04 72 43 85 28 Jean-marc.Pelletier@insa-lyon.fr
MEGA	<u>MECANIQUE, ENERGETIQUE, GENIE</u> <u>CIVIL, ACOUSTIQUE</u> M. Jean Louis GUYADER Secrétariat : M. LABOUNE PM : 71.70 -Fax : 87.12	M. Jean Louis GUYADER INSA de Lyon Laboratoire de Vibrations et Acoustique Bâtiment Antoine de Saint Exupéry 25 bis avenue Jean Capelle 69621 VILLEURBANNE Cedex Tél :04.72.18.71.70 Fax : 04 72 43 72 37 mega@lva.insa-lyon.fr
ScSo	<u>ScSo*</u> M. OBADIA Lionel Insa : J.Y. TOUSSAINT	M. OBADIA Lionel Université Lyon 2 86 rue Pasteur 69365 LYON Cedex 07 Tél : 04.78.77.23.88 Fax : 04.37.28.04.48 Lionel.Obadia@univ-lyon2.fr

*ScSo : Histoire, Géographie, Aménagement, Urbanisme, Archéologie, Science politique, Sociologie, Anthropologie

Résumé

L'analyse isogéométrique pour la résolution de problèmes de la mécanique du solide suscite de vifs intérêts depuis une dizaine d'année. En effet, cette méthode de discrétisation autorise la description exacte des géométries étudiées permettant ainsi de supprimer les erreurs dues à une mauvaise description du domaine spatial étudié. Cependant elle pose un problème théorique de propagation de raffinement lors de la localisation de maillage. Des méthodes pour contourner ce problème ont été proposées dans la littérature mais complexifient grandement la mise en œuvre de cette stratégie de résolution.

Cette thèse propose une stratégie de raffinement localisé adaptatif en espace pour les problèmes de statique et en espace temps pour les problèmes de dynamique transitoire dans le cadre de l'analyse isogéométrique. Pour cela une méthode de localisation pour l'analyse isogéométrique en statique basée sur une résolution multigrille est tout d'abord développée pour des problèmes en deux dimensions. Elle présente l'avantage de contourner la problématique de propagation de raffinement de maillage due à l'analyse isogéométrique tout en étant plus simple à mettre en œuvre que les méthodes déjà existantes. De plus, l'utilisation de l'analyse isogéométrique permet de simplifier les procédures de raffinement lors de l'adaptation de maillage qui peuvent être complexes lors de l'utilisation d'éléments finis classiques.

Une méthode de raffinement adaptatif espace temps basée sur une résolution multigrille est ensuite développée pour des problèmes en une dimension. Une étude sur la structure des opérateurs est proposée afin de choisir un intégrateur temporel adapté. Les performances de cette stratégie sont mises en évidence, puis une modification de la méthode de résolution est proposée afin de diminuer significativement les coûts de calculs associées à cette résolution. La méthode de raffinement adaptatif espace temps est appliquée à quelques exemples académiques afin de valider son bon comportement lors de la localisation.

MOTS CLÉS: Analyse isogéométrique, dynamique transitoire, raffinement adaptatif espace temps, résolution multigrille.

Table des matières

Table des matières	i
Table des figures	iii
Liste des tableaux	vii
Introduction	1
1 État de l’art	5
1.1 Analyse isogéoméométrie	6
1.1.1 Principe	6
1.1.2 Fonctions B-Splines – NURBS	7
1.1.3 Méthode de Galerkin avec fonctions de base NURBS	12
1.1.4 Raffinement de maillage NURBS par insertion de nœuds	15
1.2 Techniques de résolution multigrille	20
1.2.1 Principe	21
1.2.2 Opérateurs de changement d’échelle	25
1.2.3 Localisation	26
1.2.4 Résolution Full-Multigrid	28
1.2.5 Intérêt du couplage de l’analyse isogéoméométrie et des stratégies de résolution multigrille	30
1.3 Raffinement adaptatif spatial et temporel	31
1.4 Objectifs	32
2 Résolution multigrille localisée appliquée à l’analyse isogéoméométrie en sta- tique linéaire	33
2.1 Principe – intérêt de la méthode	34
2.2 Outils développés pour une résolution full-multigrid localisée dans le cadre de l’analyse isogéoméométrie	34
2.2.1 Opérateur de prolongement	34
2.2.2 Indicateur d’erreur	40
2.3 Validation numérique de la méthode	45
2.3.1 Comparaison entre la résolution full-multigrid globale et localisée	46
2.3.2 Performances de la résolution full multigrid localisé	48

2.4	Synthèse	55
3	Résolution multigrille localisée en espace temps pour la dynamique transi-	
	toire	57
3.1	Objectif	58
3.2	Stratégie de résolution full-multigrid espace temps	58
	3.2.1 Problème de référence	58
	3.2.2 Choix de l'espace d'approximation	59
	3.2.3 Étude des performances	61
3.3	Résolution full-multigrid espace temps appliquée au Time-Discontinuous	
	Galerkin	65
	3.3.1 Schéma d'intégration Time-Discontinuous Galerkin	65
	3.3.2 Algorithme de résolution full-multigrid localisé avec TDG	69
	3.3.3 Résultats	77
	3.3.4 Algorithme de résolution full-multigrid localisé par tranches de	
	temps	82
3.4	Synthèse	88
4	Bilan – perspectives	89
A	Représentation des transformées de Fourier discrètes.	93
B	Solution analytique de la plaque infinie en tractions avec un trou elliptique.	95
	Bibliographie	97

Table des figures

1	La résolution du problème sur le maillage grossier donne une solution \mathbf{u}_g . Un maillage localisé plus fin est créé. Afin de pouvoir résoudre le problème sur ce maillage localisé, des conditions aux limites de type Dirichlet provenant de \mathbf{u}_g sont imposées sur la frontière entre le maillage fin localisé et le maillage grossier (cette frontière est représentée en rouge sur la figure)	2
1.1	Flèche isoparamétrique pour l'analyse éléments finis (Finite Element Analysis –FEA–).	6
1.2	Flèche isoparamétrique pour l'analyse isogéométrique.	7
1.3	Relation entre les différents espaces mis en jeu. S est la paramétrisation de la géométrie, R_{ij} les fonctions NURBS utilisées et B_{ij} les coefficients vectoriels qui leurs sont associés.	8
1.4	Exemple de fonctions de base NURBS de degré 4 pour un vecteur de nœuds ouvert $\Xi = \{0, 0, 0, 0, 0, 1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 5\}$. La continuité au passage des frontières d'éléments est une conséquence directe de l'ordre polynomial des fonctions et de la multiplicité des nœuds correspondants, d'après [COT 09].	9
1.5	Paramétrisation d'une courbe B-splines, d'après [COT 09].	10
1.6	Espace paramétrique 2D défini par le produit tensoriel de deux vecteurs de nœuds.	10
1.7	Exemple de paramétrisation pour un quart de disque troué.	12
1.8	Problème statique de référence.	12
1.9	Problème de poutre en traction en une dimension.	14
1.10	Remplissage de la matrice de rigidité pour une discrétisation éléments finis et une discrétisation isogéométrique de degré 2 à nombre de degré de liberté équivalent. Les valeurs rouges correspondent aux coefficients de la matrice de rigidité non nuls.	14
1.11	Raffinement h d'une courbe paramétrée par des fonctions de base de degré 2, d'après [COT 09].	16
1.12	Raffinement p d'une courbe paramétrée par des fonctions de base de degré 2 [HUG 05].	18
1.13	Exemple de raffinement d'un élément pour une structure NURBS 2D.	19
1.14	Maillage IGA localisé obtenu avec différentes méthodes de localisation.	20

1.15	Problème de référence.	21
1.16	Transformée de Fourier discrète du champs de déplacement suivant \vec{x} . . .	22
1.17	Transformée de Fourier discrète du champs de déplacement suivant \vec{y} . . .	23
1.18	Algorithme de Correction Scheme.	24
1.19	Efficacité de la prise en compte de la correction provenant de la grille grossière.	25
1.20	Cycle multi grille pour trois grilles.	25
1.21	Relations entre les quantités primates et duales pour deux grilles.	26
1.22	Raffinement global MG pour trois grilles, d'après [RAN 08].	27
1.23	Raffinement local MG pour trois grilles, d'après [RAN 08].	27
1.24	Maillage sur lequel s'exprime \mathbf{u}^Ω , d'après [RAN 08].	28
1.25	Résolution FMG sur quatre grilles avec des V-cycles.	29
1.26	Algorithme de raffinement automatique basée sur une résolution FMG. . .	29
1.27	Principe de raffinement local à l'aide d'une résolution multigrille locali- sée en analyse isogéométrique.	30
2.1	Raffinement pour des fonctions de base éléments finis de degré 1 et des fonctions de base NURBS de degré 2.	35
2.2	Définition des domaines Ω , Ω_c et Ω_f	36
2.3	Procédure de raffinement.	36
2.4	Définition des domaines Ω_c , Ω_f et de la frontière interne Γ dans le cas de la localisation.	37
2.5	Opération d'extraction.	38
2.6	Méthode de construction d'un opérateur de prolongement localisé.	38
2.7	Exemples de localisation en une et deux dimensions.	39
2.8	Création des opérateurs de prolongement durant la procédure de raffine- ment localisé.	39
2.9	Définition des domaines pour un ensemble de quatre grilles $\{\Omega_0, \Omega_1, \Omega_2, \Omega_3\}$. .	41
2.10	Création de la zone de localisation pour des fonctions de base NURBS 1D quadratiques.	44
2.11	Procédure de raffinement pour un maillage NURBS avec $p = q = 2$	45
2.12	Données du problème pour la plaque infinie en traction avec un trou el- liptique.	46
2.13	Maillages initial et finaux pour les résolution FMG IGA globale et localisée. .	47
2.14	Comparaison de la convergence et des performances entre les résolutions FMG IGA globale et localisée.	48
2.15	Paramètres des problèmes étudiés.	49
2.16	Maillages initiaux et finaux obtenus avec une résolution FMG IGA localisée. .	50
2.17	Convergence et performances de la résolution FMG IGA localisée pour le problèmes de la membrane de Cook.	52
2.18	Convergence et performances de la résolution FMG IGA localisée pour le problèmes du coin en L.	53

2.19	Convergence and performances of the FMG IGA algorithm on the Cook's Membrane problem.	54
3.1	Problème de référence.	58
3.2	Définition d'une grille espace temps.	60
3.3	Grilles espace temps grossière et fine et équations matricielles associées. La grille grossière est composée de 4 éléments en espace et 10 éléments en temps, la grille fine de 8 éléments en espace et 20 éléments en temps. Soit respectivement des problèmes à 80 et 320 degrés de liberté après application des conditions aux limites.	62
3.4	Performances de la résolution sur deux grilles.	63
3.5	Performances de la résolution sur deux grilles après adimensionnement du problème.	63
3.6	Évolution d'une quantité α en fonction du temps dans le cadre du TDG $P1 - P1$	66
3.7	Définition d'une grille espace temps localisée.	70
3.8	Gestion des inconnues en vitesse sur les frontières interne dans le cas de la localisation.	70
3.9	Schématisation de l'opérations de lissage sur une grille espace temps.	71
3.10	Grilles espace temps dans le cas d'un raffinement global.	72
3.11	Exemple de discrétisation espace temps NURBS équivalente avec les vecteur de nœuds correspondant pour une discrétisation NURBS de degré 1 en espace et l'utilisation d'un intégrateur TDG. Les doubles lignes horizontales symbolisent les discontinuités autorisée par la discrétisation temporelle.	72
3.12	Construction de l'opérateur de prolongement espace temps global \mathcal{P}_g^{XT}	73
3.13	Construction de l'opérateur d'extraction espace temps \mathcal{P}_e^{XT}	73
3.14	Construction de l'opérateur de prolongement espace temps localisé \mathcal{P}_{loc}^{XT}	73
3.15	Principe de résolution sur deux grilles espace temps.	75
3.16	Problème de référence.	77
3.17	Maillage initial et final de la résolution.	78
3.18	Performances de la résolution FMG espace temps localisée.	78
3.19	Performances de la résolution sur deux grilles.	79
3.20	Transformée de Fourier discrète de l'erreur sur le champ de vitesse. En abscisse se trouvent les fréquences associées à la dimension temporelle et en ordonnée les fréquences associées à la dimension spatiale.	81
3.21	Comparaison des principes de résolution multigrille espace temps et multigrille par tranches de temps. On rappelle que e_T^l correspond à l'élément l de la discrétisation temporelle sur la grille fine.	83
3.22	Grilles grossière et fine utilisées pour une résolution multigrille par tranches de temps. Les deux grilles possèdent la même discrétisation en temps.	83
3.23	Comparaison des performances de la résolution FMG espace temps localisée et de la résolution FMG par tranches de temps localisée.	85

Table des figures

3.24	Chargements imposés.	86
3.25	Maillages initial et finaux obtenus pour les trois types de chargement. . .	87
4.1	Les éléments en rouge sur le maillage de gauche représentent une liste d'éléments à raffiner répartis suivant une diagonale de l'espace paramétrique. Le domaine délimité en bleu est le plus petit convexe les contenant. Appliquer la procédure de raffinement localisé développée au chapitre 2 mène à la création du maillage fin de droite. L'ensemble de l'espace est donc raffiné.	91
A.1	Représentation adoptée pour la transformée de Fourier discrète du champ a	94
B.1	Plaque infinie en traction avec un trou elliptique.	95

Liste des tableaux

3.1	Structure de l'opérateur espace temps pour le problème sans la composante hyperbolique.	64
3.2	Structure de l'opérateur espace temps pour le problème avec la composante hyperbolique.	64
3.3	Ordre de grandeur du nombre d'itérations à convergence.	65

Introduction

Lors de la conception de produits industriels, les étapes de modélisation des structures et de simulation numérique prennent une part de plus en plus importante tandis que la part dédiée à la réalisation de campagnes d'essais sur prototypes est diminuée. En effet, ces derniers peuvent se révéler extrêmement coûteux du fait de l'instrumentation ou lorsque l'essai mène à la ruine du prototype.

Les erreurs commises lors des étapes de modélisation et de simulation numérique ne permettent pas pour l'instant de s'affranchir de la validation expérimentale. La gestion de ces erreurs est donc un enjeu primordial faisant l'objet de nombreuses recherches.

Nous nous intéressons ici à des problèmes de la mécanique du solide modélisés par la théorie des milieux continus dans le cadre de l'élasticité linéaire et de la dynamique transitoire. Ces problèmes mènent à la résolution d'un système d'équations aux dérivées partielles pour lesquelles on ne sait pas déterminer les solutions analytiques dans le cas général. Ces équations sont donc discrétisées afin d'être résolues numériquement. Les travaux présentés ici concernent le contrôle de l'erreur commise lors de cette discrétisation. On considèrera donc par la suite la géométrie obtenue par Conception Assistée par Ordinateur (CAO) comme étant la géométrie exacte du solide et les solutions des équations provenant de la théorie des milieux continus comme étant les solutions exactes du problème considéré.

La méthode la plus répandue pour discrétiser ce type d'équations dans le cadre de la mécanique du solide est la méthode des éléments finis. Elle se base sur une formulation faible du problème considéré et postule un espace de recherche des fonctions solutions généré par des polynômes de Lagrange. En général, les formulations éléments finis se basent sur l'utilisation d'éléments isoparamétriques, ce qui mène à une approximation de la géométrie provenant de la CAO du solide considéré. Le problème n'est donc pas résolu sur la géométrie exacte du solide mais sur une géométrie approximée. Il s'agit d'une source d'introduction d'erreur entre la modélisation du problème et sa discrétisation. Le concept d'analyse isogéométrique (IsoGeometric Analysis – IGA –) introduit par [HUG 05] propose de conserver la même stratégie de discrétisation, mais de générer l'espace de recherche des solutions avec les fonctions utilisées pour construire la CAO du solide considéré. Ainsi, l'utilisation d'éléments isoparamétriques ne mène plus à une approximation de la géométrie CAO. Cette géométrie est toujours exactement représentée et ce même lors du raffinement de l'espace de recherche des solutions.

Cependant, l'analyse isogéométrique présente l'inconvénient de ne pas autoriser de raffinement seulement sur une portion localisée du solide. Cela est dû à la structure de

l'espace de recherche des solutions qui a une structure de produit tensoriel.

Les travaux présentés dans ce manuscrit traitent de l'élaboration d'une méthode de résolution à raffinement adaptatif automatique en espace et en temps dans le cadre de l'analyse isogéométrique, pour des problèmes d'élasticité linéaire en dynamique transitoire. Ils s'inscrivent dans la continuité des travaux de Pauline Cavin [CAV 05] et d'Ewen Biotteau [BIO 10]. [CAV 05] propose une stratégie de raffinement adaptatif automatique en espace et en temps avec l'utilisation de plusieurs niveaux de discrétisation (pouvant être localisés) du domaine espace temps. Les solutions sur les discrétisations fines localisées sont calculées à partir des solutions provenant des niveaux les plus grossiers (voir figure 1).

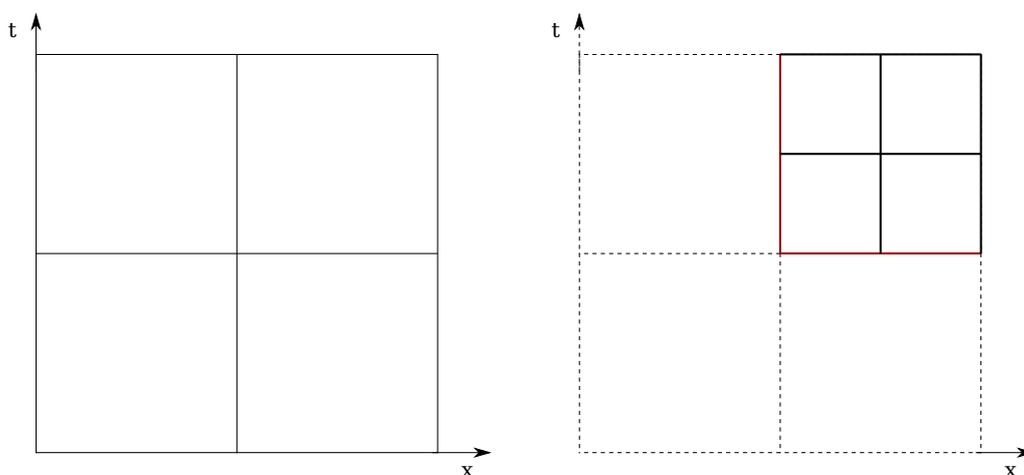


FIGURE 1: La résolution du problème sur le maillage grossier donne une solution \mathbf{u}_g . Un maillage localisé plus fin est créé. Afin de pouvoir résoudre le problème sur ce maillage localisé, des conditions aux limites de type Dirichlet provenant de \mathbf{u}_g sont imposées sur la frontière entre le maillage fin localisé et le maillage grossier (cette frontière est représentée en rouge sur la figure)

Cette approche fournit d'excellents résultats pour des problèmes linéaires mais le fait que les solutions sur les discrétisations grossières ne dépendent pas des solutions obtenues sur les niveaux de discrétisation plus fins peut poser problème lors de la prise en compte de non linéarités. [BIO 10] propose une stratégie de raffinement adaptatif en espace pour des problèmes de dynamique transitoire avec prise en compte de non linéarités matériau. Son étude de l'influence de la finesse de la discrétisation en temps sur la qualité des solutions obtenues mène à la conclusion qu'une adaptabilité en temps est nécessaire.

On propose ici une stratégie de résolution avec raffinement adaptatif automatique en espace et en temps basée sur une résolution multigrille pour des problèmes d'élasticité linéaire en dynamique transitoire. Cette méthode permet l'obtention des solutions sur plusieurs niveaux de discrétisations (pouvant être localisés). L'utilisation de techniques multigrille permet d'obtenir des solutions sur les niveaux de discrétisation grossiers dépendant des solutions obtenues sur les discrétisations fines localisées.

Ces travaux se divisent en deux parties :

- l'élaboration d'une technique de raffinement localisé pour l'IGA basée sur une résolution multigrille pour des problèmes d'élasticité linéaire en statique,
- la création d'une stratégie de résolution à raffinement adaptatif automatique en espace et en temps basée sur une résolution multigrille pour des problèmes d'élasticité linéaire en dynamique transitoire.

Pour cela, on présentera tout d'abord le principe de l'analyse isogéométrique et les méthodes de raffinement localisé existantes, puis les techniques de résolution multigrille pour des problèmes linéaires et enfin un état de l'art des techniques de raffinement adaptatif en espace et en temps.

Dans un second temps, la méthode de raffinement localisé basée sur une résolution multigrille pour des problèmes de statique dans le cadre de l'IGA est exposée. Il s'agit d'une stratégie de résolution à raffinement adaptatif automatique en espace et à précision contrôlée. Cette méthode s'avère plus simple à mettre en œuvre dans un code de calcul isogéométrique déjà existant que les autres stratégies de raffinement localisé déjà existantes.

Enfin, cette méthode de résolution est généralisée au raffinement localisé d'un domaine espace temps pour des problèmes de dynamique transitoire. Une étude de la modification de la nature des équations aux dérivées partielles sur les performances de la résolution est présentée et sera utilisée pour le choix d'un intégrateur temporel adapté. Les performances de ce principe de résolution sont étudiées sur un problème académique, puis une amélioration de la méthode est proposée afin de diminuer les coûts de calculs tout en obtenant une solution de qualité identique.

Chapitre 1

État de l'art

Sommaire

1.1	Analyse isogéoméométrie	6
1.1.1	Principe	6
1.1.2	Fonctions B-Splines – NURBS	7
1.1.3	Méthode de Galerkin avec fonctions de base NURBS	12
1.1.4	Raffinement de maillage NURBS par insertion de nœuds	15
1.2	Techniques de résolution multigrille	20
1.2.1	Principe	21
1.2.2	Opérateurs de changement d'échelle	25
1.2.3	Localisation	26
1.2.4	Résolution Full-Multigrid	28
1.2.5	Intérêt du couplage de l'analyse isogéoméométrie et des stratégies de résolution multigrille	30
1.3	Raffinement adaptatif spatial et temporel	31
1.4	Objectifs	32

1.1 Analyse isogéométrique

1.1.1 Principe

Nous nous intéressons ici à la résolution de problèmes de mécanique du solide. La méthode la plus répandue pour la résolution de ce type de problème consiste à définir la géométrie des solides considérés à l'aide d'outils de conception assistée par ordinateur (CAO), puis de le résoudre à l'aide d'une analyse éléments finis isoparamétriques. Dans la suite, nous considérerons la géométrie fournie par la CAO comme étant la géométrie exacte des solides considérés. Le principe de cette méthode est de choisir un espace d'approximation pour rechercher les solutions au problème étudié, et d'utiliser cet espace pour approximer la géométrie exacte du système étudié (figure 1.1).

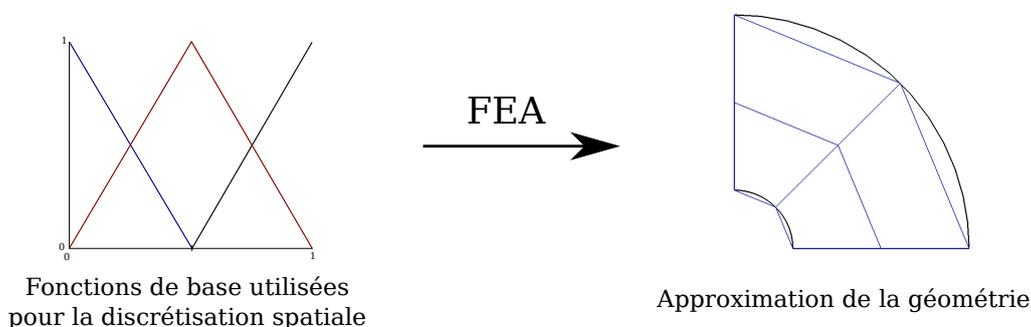


FIGURE 1.1: Flèche isoparamétrique pour l'analyse éléments finis (Finite Element Analysis –FEA–).

On peut noter que l'approximation de la géométrie exacte peut fournir des résultats erronés. On peut mentionner par exemple les problèmes de flambement de coques [GEE 05], les problèmes de couches limites pour des problèmes d'aérodynamique et d'hydrodynamiques [BAR 99, ESK 06] ou bien les problèmes de glissement entre deux solides [LAU 02, WRI 06] qui sont très sensibles aux imperfections géométriques. On peut aussi noter que les résolutions à raffinement automatique de maillage sont aussi confrontées au problème de l'approximation de la géométrie par la méthode des éléments finis. En effet elles nécessitent un retour à la géométrie exacte lors du raffinement et peuvent mener à des étapes complexes de projections entre maillage et géométrie exacte [ADA 00, BIB 13].

L'idée principale de l'analyse isogéométrique (IGA), introduite par [HUG 05, COT 09] et très largement développée par la suite (voir par exemple [BAZ 06b, BAZ 06a, COT 06, COT 07, ZHA 07, GÓM 08]), est d'inverser la flèche isoparamétrique, c'est à dire de construire l'espace de recherche des solutions à partir des fonctions utilisées pour définir la géométrie exacte du problème (figure 1.2).

De nombreuses technologies CAO peuvent être utilisées en analyse isogéométrique (Isogeometric Analysis –IGA–). La plus répandue en conception mécanique est l'utilisation de fonctions B-splines rationnelles non uniformes (Non-Uniform Rational B-Splines

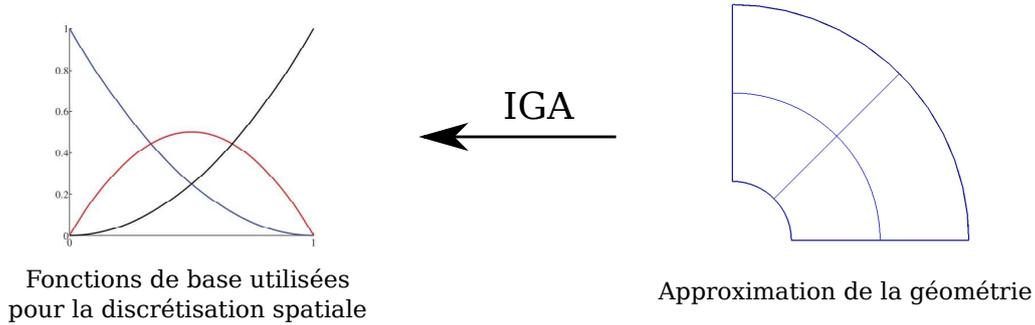


FIGURE 1.2: Flèche isoparamétrique pour l'analyse isogéométrique.

–NURBS–) [PIE 12, FAR 02, COH 01, ROG 00]. Leur avantage est de pouvoir représenter exactement toutes les sections coniques : cercles, sphères, cylindres, ellipsoïdes, etc. Elle présente l'avantage d'autoriser le raffinement par insertion de nœuds (raffinement h) ou par élévation d'ordre (raffinement p) et de présenter une continuité C^{p-1} où p est l'ordre des fonctions NURBS utilisées.

Ces fonctions permettent une description paramétrique d'un domaine. Elles sont définies dans un espace appelé espace paramétrique ; la géométrie est définie dans un espace nommé espace physique. La paramétrisation de la géométrie est obtenue par un morphisme non linéaire entre l'espace paramétrique et l'espace physique défini à l'aide des fonctions NURBS. S'il est nécessaire d'effectuer des intégrations lors du processus de résolution, elles sont effectuées élément par élément sur un élément de référence obtenu par une transformation linéaire à partir de l'espace paramétrique. Ces différents espaces mis en jeu sont représentés figure 1.3.

La définition des fonctions NURBS, le principe de raffinement h et leur utilisation pour la résolution de problème de mécanique du solide dans le cadre de l'élasticité linéaire en statique sont détaillés dans les sections suivantes.

1.1.2 Fonctions B-Splines – NURBS

Définitions des fonctions B-splines 1D Les fonctions de base NURBS sont une généralisation des fonctions B-splines. Ce sont des fonctions polynômiales par morceaux dont le degré de continuité est imposé. Les fonctions B-splines 1D sont construites à partir d'un vecteur de nœuds $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$ qui représente un jeu de coordonnées dans l'espace paramétrique (voir figure 1.3) où p est l'ordre polynomial des fonctions B-splines et n le nombre de fonctions de base. Elles sont définies sur l'espace paramétrique Ξ , à valeur dans $[0, 1]$, et calculées de la manière suivante :

$$\begin{cases} N_{i,0}(\xi) = \begin{cases} 1 & \text{si } \xi_i \leq \xi < \xi_{i+1} \\ 0 & \text{sinon} \end{cases}, \\ N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi), \end{cases} \quad (1.1)$$

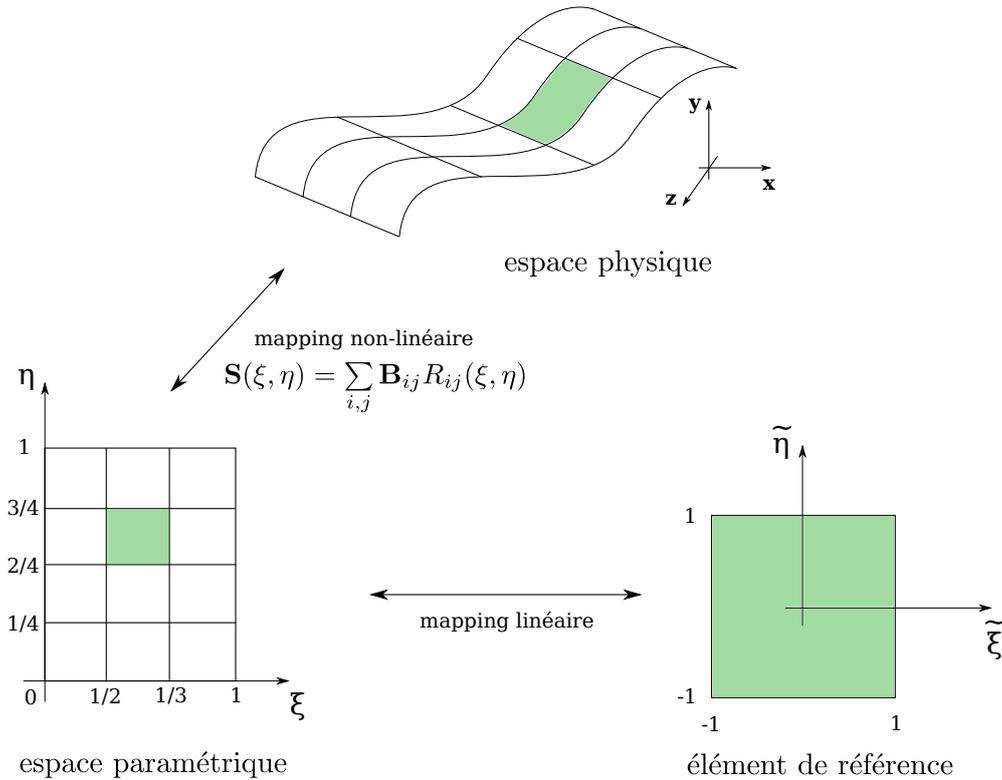


FIGURE 1.3: Relation entre les différents espaces mis en jeu. S est la paramétrisation de la géométrie, R_{ij} les fonctions NURBS utilisées et B_{ij} les coefficients vectoriels qui leurs sont associés.

où $\xi_i \in \mathbb{R}$ est le $i^{\text{ème}}$ nœud du vecteur de nœuds Ξ . Il est possible de répéter plusieurs fois un nœud dans le vecteur de nœuds Ξ . On notera γ_i la multiplicité du nœud ξ_i . Les fonctions B-splines non nulles en ξ_i présenteront une continuité $C^{p-\gamma_i}$ au passage de ce nœud. Pour le vecteur de nœuds $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$, on associe $n + p$ éléments $e_i = [\xi_i, \xi_{i+1}]$, $i \in [1, n + p]$. Un élément e_i peut donc être de mesure nulle si $\xi_i = \xi_{i+1}$. On appelle vecteur de nœuds ouvert un vecteur de nœuds présentant une multiplicité de $p + 1$ pour le premier et le dernier nœud de celui-ci. Les fonctions de base B-splines générées par un vecteur de nœuds ouvert sont interpolantes sur les extrémités de l'espace paramétrique. La figure 1.4 présente un exemple de la dépendance de la continuité au passage des frontières d'éléments à la multiplicité des nœuds correspondants, pour des fonctions de base de degré 4.

Il est important de noter que le support des fonctions de base B-splines peut être plus de deux éléments, contrairement aux fonctions de base éléments finis. En prenant en compte les éléments de mesure nulle, le support d'une fonction de base B-splines de

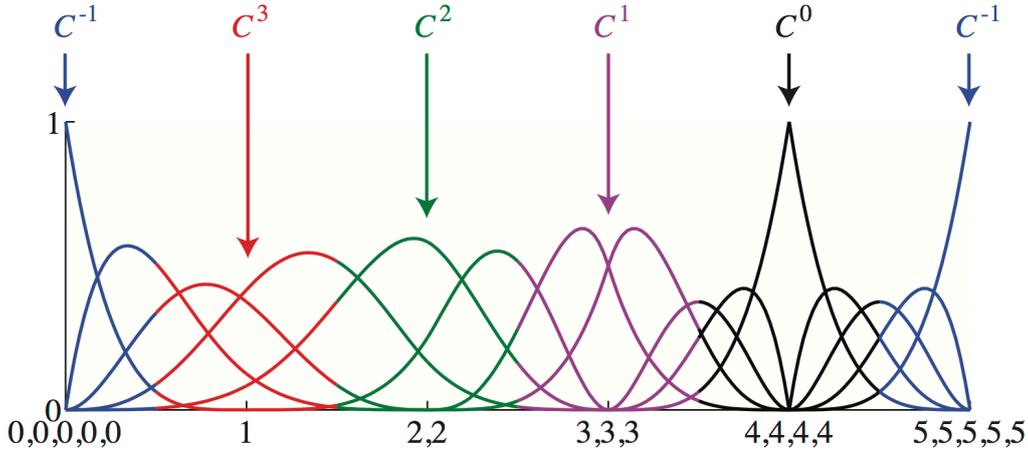


FIGURE 1.4: Exemple de fonctions de base NURBS de degré 4 pour un vecteur de nœuds ouvert $\Xi = \{0, 0, 0, 0, 0, 1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 5\}$. La continuité au passage des frontières d'éléments est une conséquence directe de l'ordre polynomial des fonctions et de la multiplicité des nœuds correspondants, d'après [COT 09].

degré p est de $p + 1$ éléments. Dans l'exemple présenté figure 1.4, toutes les fonctions B-splines ont donc pour support 5 éléments.

Mapping entre l'espace paramétrique et l'espace physique Les fonctions de base sont définies sur l'espace paramétrique et à valeurs dans $[0, 1]$. La paramétrisation de la courbe réelle de la structure dans l'espace physique C est effectuée par une combinaison linéaire des fonctions de base B-splines :

$$C(\xi) = \sum_{i=1}^n N_{i,p}(\xi) \mathbf{B}_i.$$

Les coefficient vectoriels \mathbf{B}_i associés aux fonctions de base $N_{i,p}$ sont appelés points de contrôle. Ils correspondent aux valeurs nodales pour une discrétisation éléments finis classique, à la différence que les points de contrôle ne sont pas nécessairement interpolants. Les points de contrôle n'ont donc pas forcément d'interprétation physique concrète car n'étant pas nécessairement sur la géométrie.

On donne un exemple de définition d'une géométrie 1D à l'aide de fonctions de base B-splines figure 1.5.

Surfaces et volumes B-splines Pour définir une surface B-spline, on se donne deux vecteurs de nœuds $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$ et $\mathcal{H} = \{\eta_1, \eta_2, \dots, \eta_{m+q+1}\}$ où p et q sont respectivement les ordres polynomiaux des fonctions de base B-splines 1D associées et n et m le nombre de ces fonctions de base. L'espace paramétrique 2D est défini par produit tensoriel des deux vecteurs de nœuds Ξ et \mathcal{H} (voir figure 1.6).

Les fonctions de base B-splines 2D sont donc définies par produit tensoriel des deux jeux de fonctions de base 1D dans chacune des directions paramétriques ξ et η et la

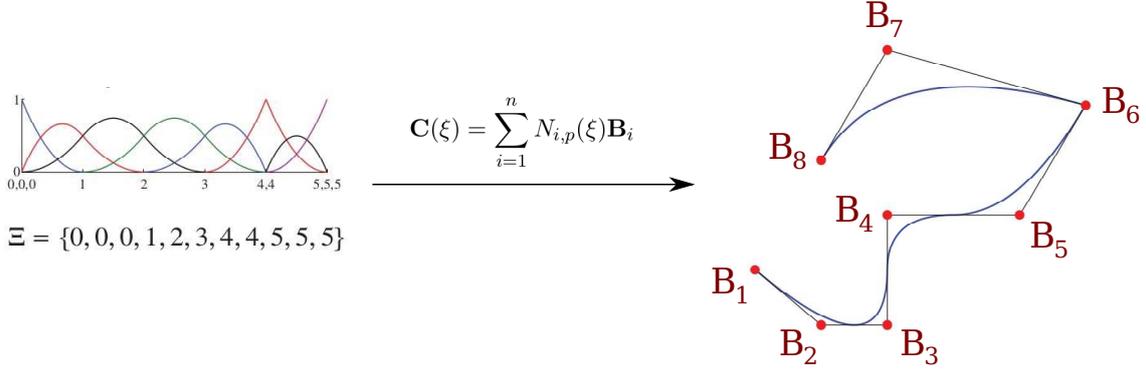


FIGURE 1.5: Paramétrisation d'une courbe B-splines, d'après [COT 09].

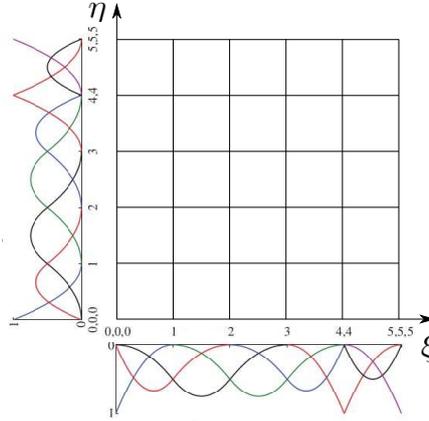


FIGURE 1.6: Espace paramétrique 2D défini par le produit tensoriel de deux vecteurs de nœuds.

surface B-splines définie par

$$\mathbf{S}(\xi, \eta) = \sum_{i=1}^n \sum_{j=1}^m \tilde{N}_{ij,pq}(\xi, \eta) \mathbf{B}_{i,j},$$

avec

$$\tilde{N}_{ij,pq}(\xi, \eta) = N_{i,p}(\xi) M_{j,q}(\eta).$$

Pour définir un volume B-spline, on ajoute un troisième vecteur de nœuds $\mathbf{Z} = \{\zeta_1, \zeta_2, \dots, \zeta_{l+r+1}\}$ avec l et r le nombre et l'ordre polynomial des fonctions de base B-splines associées. L'espace paramétrique et les fonctions de base B-splines 3D sont construites par produit tensoriel des trois vecteur de nœuds et des trois jeux de fonctions de base. Le volume B-splines est donc défini par

$$\mathbf{V}(\xi, \eta, \zeta) = \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^l \tilde{N}_{ijk,pqr}(\xi, \eta, \zeta) \mathbf{B}_{i,j,k},$$

avec

$$\tilde{N}_{ijk,pqr}(\xi, \eta, \zeta) = N_{i,p}(\xi) M_{j,q}(\eta) L_{k,r}(\zeta).$$

Fonctions de base NURBS Les fonctions B-splines permettent de représenter uniquement des géométries polynomiales. Les fonctions NURBS permettent la représentation notamment des coniques. Un objet NURBS de dimension d est obtenu par la projection d'un objet B-splines de dimension $d + 1$. Le détail de ces projections géométriques n'est pas détaillé ici. La méthodologie complète peut être trouvée dans les ouvrages de référence [FAR 99, HUG 05].

Le maillage NURBS dans l'espace paramétrique est défini de la même manière que pour les fonctions de base B-spline. On introduit le jeu de poids $W = \{w_{i,j,k} \in \mathbb{R}, i, j, k \in [1, n + p + 1] \otimes [1, m + q + 1] \otimes [1, l + r + 1]\}$. Les fonctions de base NURBS $R_{i,j,k}^{p,q,r}$ sont définies à partir des fonctions de base B-splines et du jeu de poids W de la manière suivante :

$$R_{i,j,k}^{p,q,r}(\xi, \eta, \zeta) = \frac{w_{i,j,k} N_{i,p}(\xi) M_{j,q}(\eta) L_{k,r}(\zeta)}{\sum_{\hat{i}=1}^n \sum_{\hat{j}=1}^m \sum_{\hat{k}=1}^l w_{\hat{i},\hat{j},\hat{k}} N_{\hat{i},p}(\xi) M_{\hat{j},q}(\eta) L_{\hat{k},r}(\zeta)} \quad (1.2)$$

Ces fonctions de base NURBS ont les mêmes propriétés de continuité et de taille de support que les fonctions de base B-splines utilisées pour les générer. Elles sont de plus non négatives, et forment une partition de l'unité du domaine de leur espace de définition. On remarquera que si tous les poids $w_{i,j,k}$ sont égaux alors $R_{i,j,k}^{p,q,r} = N_{i,p}(\xi) M_{j,q}(\eta) L_{k,r}$. Les fonctions de base B-splines sont donc un cas particulier des fonctions NURBS.

Un volume NURBS est défini par

$$\mathbf{V}(\xi, \eta, \zeta) = \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^l R_{i,j,k}^{p,q,r}(\xi, \eta, \zeta) \mathbf{B}_{i,j,k},$$

où les $\mathbf{B}_{i,j,k}$ sont les points de contrôle associés aux fonctions NURBS $R_{i,j,k}^{p,q,r}$.

On donne en figure 1.7 un exemple de paramétrisation NURBS en deux dimensions pour un quart de disque troué. La description de la géométrie est effectuée de manière exacte en utilisant un seul élément avec des fonctions de degré 2 dans la direction paramétrique ξ et de degré 1 dans la direction paramétrique η .

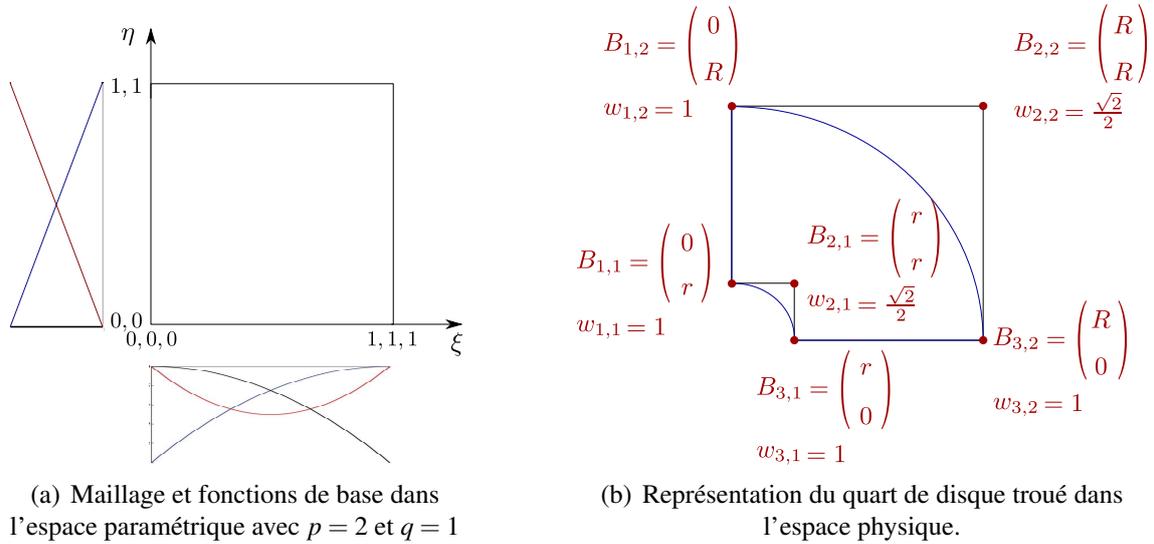


FIGURE 1.7: Exemple de paramétrisation pour un quart de disque troué.

1.1.3 Méthode de Galerkin avec fonctions de base NURBS

L'idée principale de l'analyse isogéométrique est d'utiliser les fonctions de base NURBS représentant la géométrie pour générer l'espace de recherche de la solution au problème de mécanique souhaité (voir section 1.1.1). L'idée est donc d'utiliser des éléments isoparamétriques pour effectuer une discrétisation de type Galerkin du problème. Nous allons ici détailler cette discrétisation pour un problème élastique linéaire en statique en l'absence d'efforts volumiques.

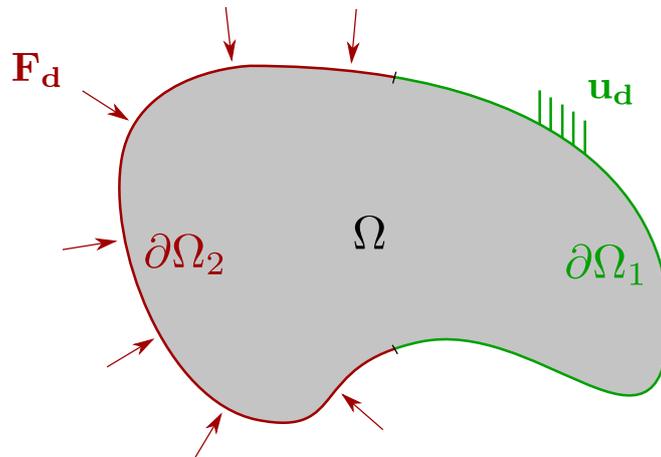


FIGURE 1.8: Problème statique de référence.

Les équations fortes associées au problème représenté figure 1.8 sont :

$$\begin{cases} \operatorname{div} \boldsymbol{\sigma}(\mathbf{u}(\mathbf{x})) = \mathbf{0}, & \forall \mathbf{x} \in \Omega, \\ \mathbf{u}(\mathbf{x}) = \mathbf{u}_d(\mathbf{x}), & \forall \mathbf{x} \in \partial\Omega_1, \\ \boldsymbol{\sigma}(\mathbf{u}(\mathbf{x})) \cdot \mathbf{n}(\mathbf{x}) = \mathbf{F}_d(\mathbf{x}), & \forall \mathbf{x} \in \partial\Omega_2. \end{cases} \quad (1.3)$$

où \mathbf{u} est le champ de déplacement, \mathbf{F}_d les efforts imposés sur $\partial\Omega_2$, \mathbf{u}_d les déplacements imposés sur $\partial\Omega_1$ et $\boldsymbol{\sigma}(\mathbf{u})$ le tenseur des contraintes associé au champ de déplacement \mathbf{u} .

On note \mathcal{U} l'espace fonctionnel associé au problème 1.3 à énergie finie et assurant les bonnes conditions de régularité $\mathcal{U} = \{\mathbf{u} \in H_1(\Omega) / \mathbf{u}|_{\partial\Omega_1} = \mathbf{u}_d\}$ et \mathcal{U}^* l'espace fonctionnel associé à \mathcal{U} cinématiquement admissible à zéro $\mathcal{U}^* = \{\mathbf{u}^* \in H_1(\Omega) / \mathbf{u}^*|_{\partial\Omega_1} = \mathbf{0}\}$.

La formulation faible associée aux équations 1.3 est donc : trouver $\mathbf{u} \in \mathcal{U}$ tel que, $\forall \mathbf{u}^* \in \mathcal{U}^*$,

$$\int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{u}^*) \boldsymbol{\sigma}(\mathbf{u}) \, d\Omega = \int_{\partial\Omega_2} \mathbf{u}^* \mathbf{F}_d \, d\partial\Omega \quad \forall \mathbf{u}^* \in \mathcal{U}^*. \quad (1.4)$$

Le domaine Ω provenant d'une paramétrisation NURBS et étant considéré comme étant la géométrie exacte, on peut l'écrire sous la forme

$$\Omega(\xi, \eta, \zeta) = \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^l R_{i,j,k}^{p,q,r}(\xi, \eta, \zeta) \mathbf{B}_{i,j,k}$$

où les fonctions $R_{i,j,k}^{p,q,r}$ sont les fonctions de base NURBS et $\mathbf{B}_{i,j,k}$ les points de contrôles associés à ces fonctions. Le concept d'éléments isoparamétriques consiste à rechercher le champ de déplacement \mathbf{u} sous la forme

$$\mathbf{u}(\xi, \eta, \zeta) = \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^l R_{i,j,k}^{p,q,r}(\xi, \eta, \zeta) \mathbf{U}_{i,j,k}.$$

En adoptant le même espace d'approximation pour le champ test \mathbf{u}^* et réinjectant les formes de \mathbf{u} et \mathbf{u}^* dans 1.4 il vient :

$$\begin{aligned} & \sum_{i^*=1}^n \sum_{j^*=1}^m \sum_{k^*=1}^l \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^l \mathbf{U}_{i^*,j^*,k^*}^* \left(\int_{\Omega} \boldsymbol{\varepsilon}(R_{i^*,j^*,k^*}^{p,q,r}(\mathbf{x})) \boldsymbol{\sigma}(R_{i,j,k}^{p,q,r}(\mathbf{x})) \, d\Omega \right) \mathbf{U}_{i,j,k} \\ & = \sum_{i^*=1}^n \sum_{j^*=1}^m \sum_{k^*=1}^l \mathbf{U}_{i^*,j^*,k^*}^* \left(\int_{\partial\Omega_2} R_{i^*,j^*,k^*}^{p,q,r}(\mathbf{x}) \mathbf{F}_d(\mathbf{x}) \, d\partial\Omega \right), \forall \mathbf{U}_{i^*,j^*,k^*}^*, \end{aligned} \quad (1.5)$$

que l'on peut mettre sous la forme $\mathbb{K}\mathbf{U} = \mathbf{F}$ ou \mathbb{K} et \mathbf{F} sont les classiques matrices de rigidité et efforts généralisés, et \mathbf{U} le vecteur contenant les variables de contrôle associées au champ de déplacement \mathbf{u} .

L'utilisation d'une discrétisation NURBS apporte quelques changements aux propriétés de la matrice de rigidité obtenue classiquement avec les éléments finis. Lors de l'utilisation de fonctions de base d'ordre supérieur ou égal à 2, la matrice de rigidité obtenue

avec des éléments isogéométriques est symétrique, définie positive et présente la même largeur de bande qu'en éléments finis. Cependant, la bande est plus dense. Les opérations de multiplication matrice - vecteur sont donc plus coûteuses, tout comme le stockage de la matrice de rigidité.

Dans un souci de simplicité, ces propriétés seront mises en évidence sur un exemple de discrétisation en une dimension : une poutre de module d'Young E , de longueur L et de section S encastée à une extrémité et soumise à un effort F à l'autre extrémité.

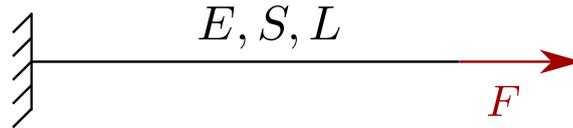
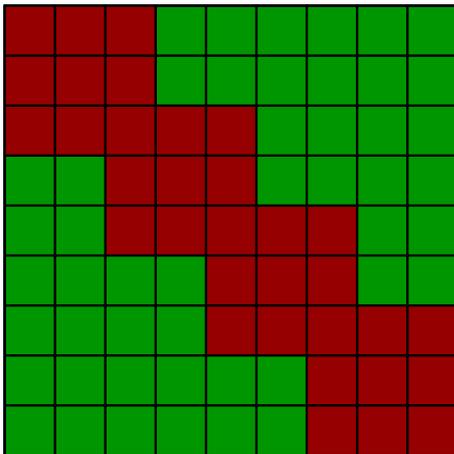
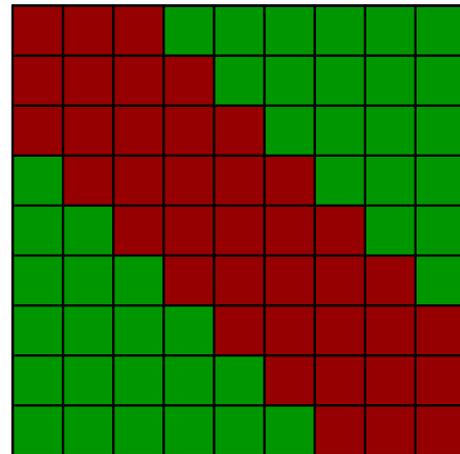


FIGURE 1.9: Problème de poutre en traction en une dimension.

Nous allons discrétiser le problème de poutre en traction présenté en figure 1.9 à l'aide d'éléments finis de degré 2 et d'éléments NURBS de degré 2 à nombre de degrés de libertés équivalents. Les matrices de rigidité obtenues (respectivement \mathbb{K}_{EF} et \mathbb{K}_{IGA}) ont la forme définie en figure 1.10.



(a) \mathbb{K}_{EF} pour une discrétisation éléments finis de degré 2 avec 9 degrés de liberté (4 éléments).



(b) \mathbb{K}_{IGA} pour une discrétisation isogéométrique de degré 2 et de continuité C^1 avec 9 degrés de liberté (7 éléments).

FIGURE 1.10: Remplissage de la matrice de rigidité pour une discrétisation éléments finis et une discrétisation isogéométrique de degré 2 à nombre de degré de liberté équivalent.

Les valeurs rouges correspondent aux coefficients de la matrice de rigidité non nuls.

En notant N_{DDL} le nombre de degrés de liberté de la discrétisation, la matrice de rigidité compte $4N_{DDL} - 3$ valeurs non nulles dans le cas éléments finis alors que dans le cas d'une discrétisation isogéométrique elle en compte $5N_{DDL} - 6$. Asymptotiquement, une matrice de rigidité résultant d'une discrétisation isogéométrique présente un remplissage

25% plus important qu'une matrice de rigidité provenant d'une discrétisation éléments finis à nombre de degrés de liberté équivalent.

Le stockage et les opérations classiques sur la matrice de rigidité sont donc significativement plus coûteuses. Cependant, les travaux effectués sur l'analyse isogéoméométrie montrent qu'à nombre de degrés de liberté égaux, la solution obtenue avec l'IGA est de meilleure qualité que la solution obtenue par FEA. À qualité de solution égale, une solution par IGA présente moins de degrés de liberté qu'une solution obtenue par FEA [COT 06, EVA 09, GRO 12]. Une comparaison objective de ces deux méthodes nécessite donc de prendre en compte le coût de chaque opération numérique. On peut également noter que les solveurs et préconditionneurs adaptés en FEA ne le sont pas forcément en IGA [COL 12, COL 13, VEI 13a, VEI 13b].

1.1.4 Raffinement de maillage NURBS par insertion de nœuds

Il existe trois façons différentes d'enrichir une base B-splines tout en laissant la géométrie représentée inchangée :

- le raffinement par insertion de nœuds dans l'espace paramétrique, qui se rapproche du raffinement h en éléments finis,
- le raffinement par élévation d'ordre des fonctions de base, qui se rapproche du raffinement p en éléments finis,
- le raffinement par insertion de nœuds et élévation d'ordre, nommé raffinement k qui ne possède pas d'équivalent en éléments finis.

Raffinement h (par insertion de nœuds) Soit $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$ et $\mathcal{B} = \{\mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_{n+p+1}\}^T$ le vecteur de nœuds initial et les points de contrôles définissant la courbe \mathbf{C} . On introduit $\bar{\Xi} = \{\bar{\xi}_1 = \xi_1, \bar{\xi}_2, \dots, \bar{\xi}_{n+m+p+1} = \xi_{n+p+1}\}$ un vecteur de nœuds étendu tel que $\Xi \subset \bar{\Xi}$. Afin de conserver la courbe \mathbf{C} inchangée, les nouveaux points de contrôle $\bar{\mathcal{B}} = \{\bar{\mathbf{B}}_1, \bar{\mathbf{B}}_2, \dots, \bar{\mathbf{B}}_{n+p+1}\}^T$ associés au vecteur de nœuds étendu $\bar{\Xi}$ sont obtenus par la combinaison linéaire des points de contrôle originaux \mathcal{B} suivante :

$$\bar{\mathcal{B}} = \mathbb{T}^p \mathcal{B}, \quad (1.6)$$

où

$$T_{ij}^0 = \begin{cases} 1, & \text{si } \bar{\xi}_i \in [\xi_j, \xi_{j+1}] \\ 0, & \text{sinon} \end{cases}, \quad (1.7)$$

et

$$T_{ij}^{q+1} = \frac{\bar{\xi}_{i+q} - \xi_j}{\xi_{j+q} - \xi_j} T_{ij}^q + \frac{\xi_{j+q+1} - \bar{\xi}_{i+q}}{\xi_{j+q+1} - \xi_{j+1}} T_{i,j+1}^q, \text{ pour } q = 0, 1, 2, \dots, p-1 \quad (1.8)$$

La figure 1.11 présente le raffinement par insertion de nœuds d'une courbe paramétrée initialement avec 5 éléments. La colonne de gauche représente les points de contrôle

initiaux, le maillage original dans l'espace physique et les fonctions de base initiales dans l'espace paramétrique. La colonne de droite représente ces mêmes quantités après un raffinement par insertion de nœuds où chaque élément a été coupé en deux dans l'espace paramétrique. La nouvelle courbe est géométriquement et paramétriquement identique à l'originale, mais les points de contrôle ont changé, et la base est plus riche.

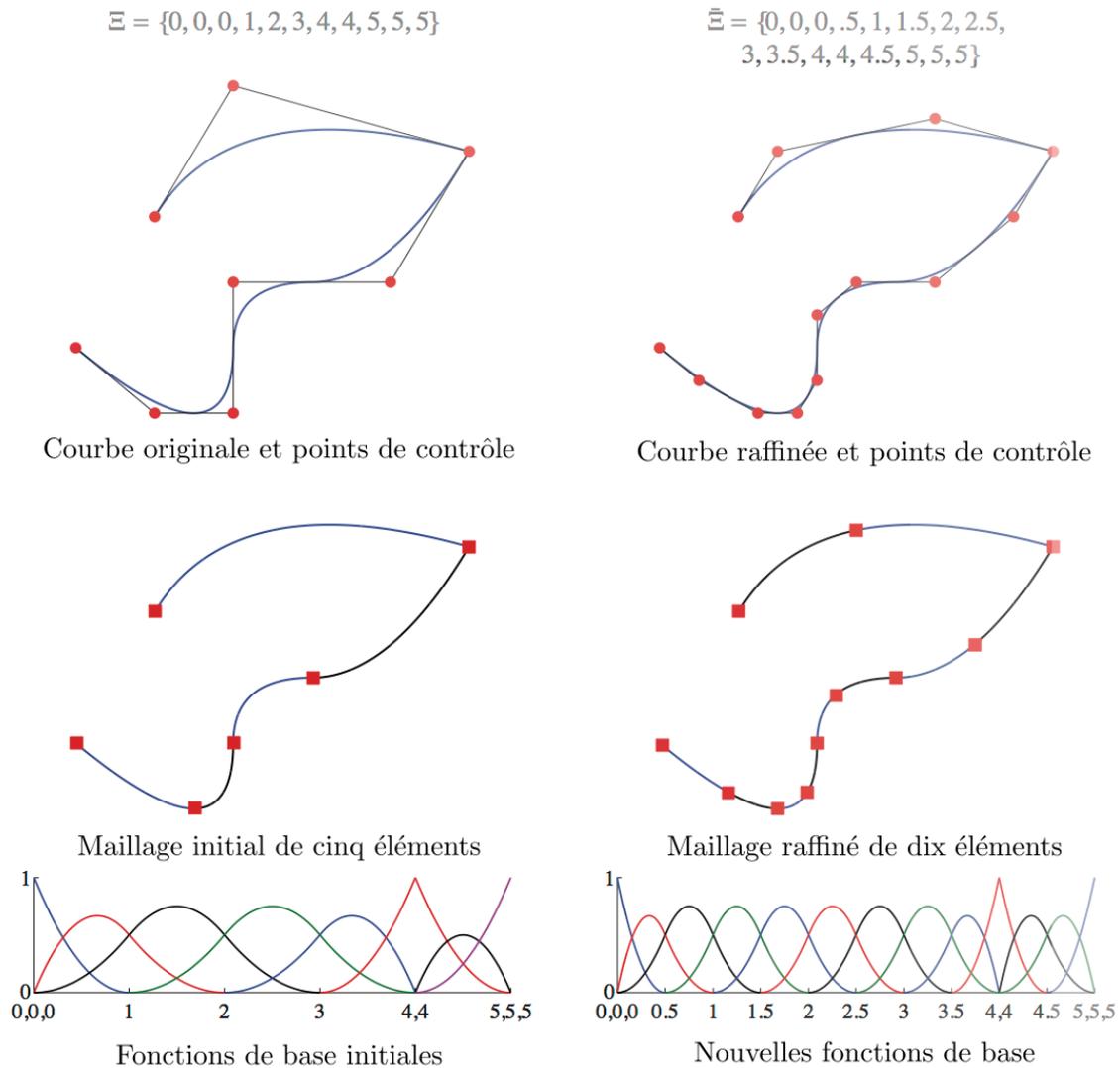


FIGURE 1.11: Raffinement h d'une courbe paramétrée par des fonctions de base de degré 2, d'après [COT 09].

Le raffinement par insertion de nœuds est très similaire au raffinement h en éléments finis en ce sens que les éléments existants sont découpés en plusieurs nouveaux éléments. Cependant la différence majeure entre le raffinement h en éléments finis et le raffinement h en analyse isogéométrique est le nombre de nouvelles fonctions de base créées, et la continuité des fonctions de base au passage des frontières des nouveaux éléments. En prenant

un exemple de structure une dimension, un raffinement h de fonctions de base NURBS introduit n_{elem} fonctions de base supplémentaires quelque soit le degré polynomial p des fonctions de base, où n_{elem} est le nombre d'éléments raffinés, alors qu'un raffinement h en éléments finis introduit $p \times n_{elem}$ fonctions de base supplémentaires.

Pour effectuer un raffinement h identique à celui en éléments finis, il faudrait insérer p fois chaque nouvelle valeur de nœuds (où p est l'ordre polynomial des fonctions de base) afin d'obtenir une continuité C^0 au passage de la nouvelle frontière.

Raffinement p (par élévation d'ordre) Le raffinement par élévation d'ordre consiste à augmenter l'ordre polynomial des fonctions de base utilisées pour paramétrer la géométrie. On rappelle que la continuité des fonctions de base au passage d'une frontière d'éléments est d'ordre $p - \gamma_i$. L'élévation du degré polynomial p implique donc l'augmentation de la multiplicité γ_i des nœuds afin de conserver les discontinuités des différentes dérivées présentes dans la courbe originale. Ainsi, durant le processus d'élévation d'ordre, la multiplicité de chacun des nœuds du vecteur de nœuds est augmentée mais aucune nouvelle valeur de nœud n'est ajoutée. Comme pour le processus de raffinement par insertion de nœuds, ni la géométrie ni la paramétrisation de la courbe ne sont changées lors d'un raffinement p . En dimension 1, un raffinement p d'une structure NURBS introduit n_{elem} fonctions de base supplémentaires, où n_{elem} est le nombre d'éléments de la paramétrisation, tout comme en éléments finis. Cependant, la continuité au passage de la frontière des éléments reste d'ordre $p - \gamma_i$, alors qu'en éléments finis les fonctions de base sont nécessairement C^0 au passage de ces frontières.

Le processus de raffinement par élévation d'ordre est basé sur une décomposition de Bézier de la courbe considérée. Ce processus ne sera pas exposé ici, on peut se référer à [ROG 00, FAR 99] pour de plus amples détails. De nombreux algorithmes existent afin de minimiser le coût de cette procédure et peuvent être trouvés dans [PIE 12]. Un exemple de raffinement p est proposé figure 1.12. Le nombre d'éléments initiaux est conservé. On constate bien l'augmentation de la multiplicité de chacun des nœuds dans l'espace paramétrique afin de conserver le degré de continuité présent avant le raffinement. La géométrie et la paramétrisation de la courbe \mathbf{C} restent inchangées.

Raffinement k Le raffinement k est une combinaison des deux raffinements présentés précédemment. On remarquera tout d'abord que la composition du raffinement h et du raffinement p n'est pas commutative. En effet, si un seul nœud $\bar{\xi}$ est inséré entre deux valeurs distinctes du vecteur de nœuds Ξ d'une courbe d'ordre p , la continuité des fonctions de base au passage de $\bar{\xi}$ sera C^{p-1} . Si l'on élève ensuite les fonctions de base à l'ordre q par un raffinement p , la continuité au passage de $\bar{\xi}$ sera toujours C^{p-1} , et toutes les fonctions de base seront d'ordre q . Si l'on inverse maintenant l'ordre de ces deux raffinements, la continuité au passage de $\bar{\xi}$ sera C^{q-1} et les fonctions de base seront de degré q . La procédure consistant à effectuer tout d'abord un raffinement p puis un raffinement h est nommée raffinement k .

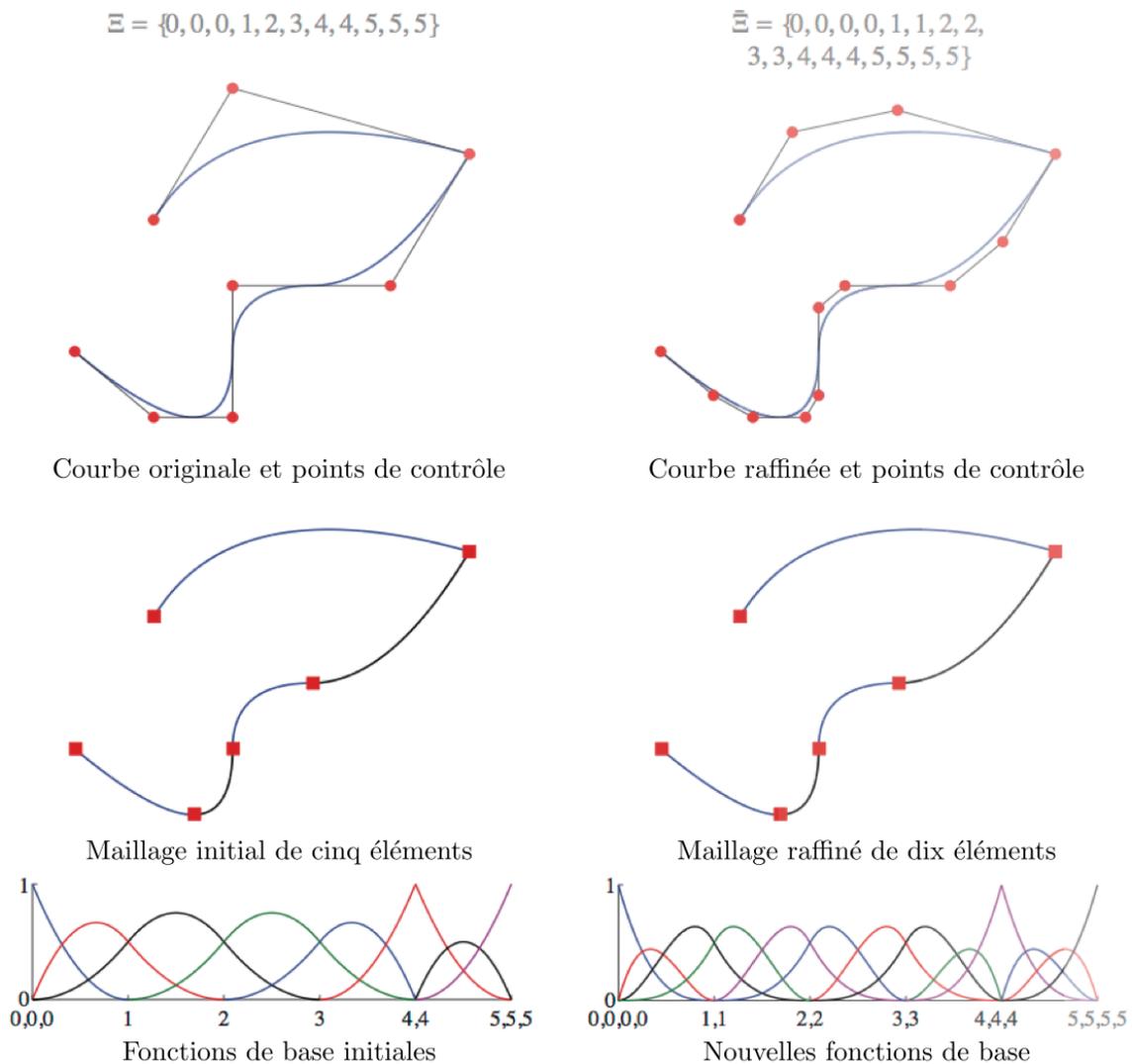


FIGURE 1.12: Raffinement p d'une courbe paramétrée par des fonctions de base de degré 2 [HUG 05].

Problématique de propagation de raffinement h dans une structure NURBS Dans la suite du manuscrit, nous nous intéresserons au raffinement h d'espaces paramétriques avec des fonctions de base NURBS. Nous allons ici souligner le problème de propagation de raffinement dans un espace NURBS de dimension supérieur ou égale à deux. On rappelle que les fonctions de base et l'espace paramétrique sont construits à l'aide d'un produit tensoriel pour une géométrie de dimension deux ou supérieure. La procédure de raffinement h consiste à insérer de nouveaux nœuds dans les différentes directions paramétriques. Il n'est donc pas possible de sous découper seulement un seul élément de l'espace paramétrique. Le sous découpage d'un éléments induit le sous découpage de tous les autres éléments obtenus par translation suivant les différentes directions paramétriques. On donne en figure 1.13 un exemple de cette problématique de propagation

de raffinement. On souhaite sous-découper l'élément rouge. Pour cela il faut insérer les nœuds $\xi = 0,25$ et $\eta = 0,25$ dans les vecteurs de nœuds Ξ et \mathcal{H} . Cela implique le sous-découpage des éléments verts.

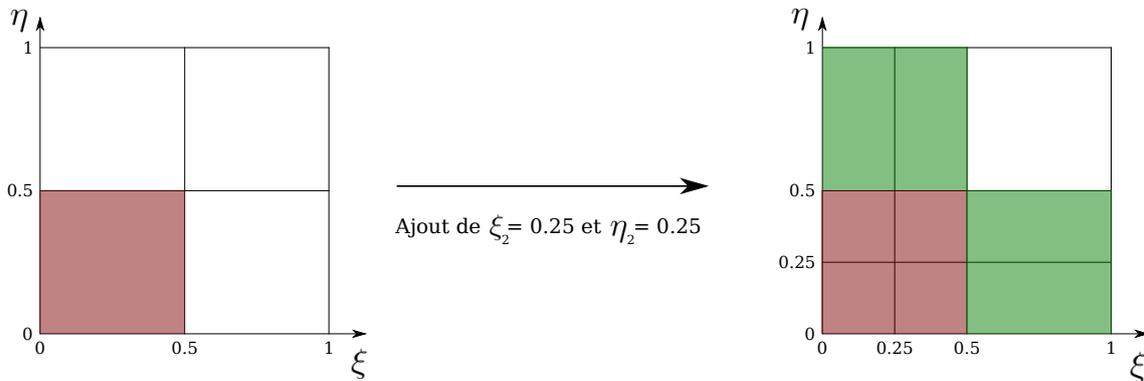


FIGURE 1.13: Exemple de raffinement d'un élément pour une structure NURBS 2D.

Cette problématique de propagation de raffinement est similaire à celle présente lors de l'utilisation de discrétisations basées sur les différences finies. Pour des problèmes en trois dimensions, cette propagation du raffinement dans les trois directions paramétriques conduit à la création d'un nombre de degrés de liberté non négligeable dans des portions de domaine où cela n'est pas nécessaire et conduit à des coûts de calculs prohibitifs.

Raffinement localisé en IGA La problématique de propagation de raffinement de maillage lors de l'utilisation de fonctions NURBS est l'objet de nombreux travaux.

Les technologies développées sont basées sur l'utilisation de fonctions de base dérivées des B-splines qui ne sont pas définies par un produit tensoriel global dans l'espace considéré. On peut notamment citer les fonctions B-splines hiérarchiques, les LRB-splines et T-splines.

Les fonctions de base T-splines [SED 03, BAZ 10, SCO 12] permettent l'utilisation de maillage présentant des jonctions en T, similaires à l'utilisation de nœuds flottants en éléments finis. L'utilisation de ce type de fonctions de base nécessite la prise de précaution lors du raffinement afin de conserver les propriétés d'indépendance linéaire de toutes les fonctions de base. Ces procédures de raffinement itératives peuvent donc se révéler très coûteuses. De plus il a été montré dans [DÖR 10] par exemple que la procédure de raffinement peut mener à un maillage beaucoup moins localisé que l'on pouvait l'espérer.

Les fonctions de base LR B-Splines [DOK 13] sont basées sur la séparabilité des fonctions B-splines en produit de plusieurs fonctions B-splines définies sur des vecteurs de nœuds locaux. L'utilisation de ce type de fonctions permet l'utilisation de maillage LR, que l'on obtient à partir d'un maillage tensoriel par insertion successive de lignes verticales et horizontales débutant sur un bord d'élément et finissant sur un autre bord d'élément. Les travaux menés par [JOH 14] semblent montrer un meilleur comportement lors de la création de maillage avec localisation que lors de l'utilisation de fonctions T-

Splines. Cependant l'indépendance linéaire des fonctions de base LR B-Splines n'est pas encore démontrée.

Les fonctions B-splines hiérarchiques [KRA 94, VUO 11] sont obtenues en remplaçant une fonction B-splines ayant comme support une portion \mathcal{S} du domaine considéré par des fonctions B-splines d'une discrétisation plus fine du domaine Ω , ayant support sur \mathcal{S} et s'annulant sur le bord de \mathcal{S} . Les systèmes linéaires à résoudre en adoptant ce genre d'approche se trouvent souvent très mal conditionnés. Les fonctions B-Splines hiérarchique tronquées (THB-Splines) [GIA 12] ont été développées pour résoudre les problèmes de conditionnement et diminuer le remplissage des matrices associées au système linéaire à résoudre lors de l'utilisation de ces fonctions de base. Cela nécessite de modifier localement un grand nombre de fonctions de base ce qui peut mener à des coûts de calculs importants lors du processus de raffinement.

On donne en figure 1.14 des exemples de maillages IGA localisés avec ces différentes méthodes.

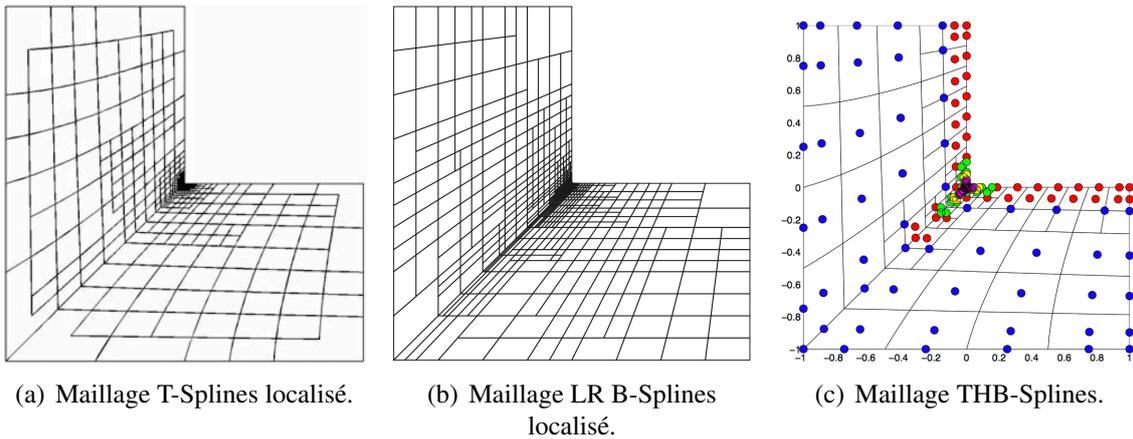


FIGURE 1.14: Maillage IGA localisé obtenu avec différentes méthodes de localisation.

Le coût du processus de raffinement et la complexité de l'implémentation des fonctions de base utilisées étant le principal inconvénient des méthodes de localisation citées ci-dessus, nous avons développé une stratégie de localisation en IGA en utilisant des fonctions NURBS basées sur une résolution multigrille localisée. En effet, l'utilisation de différents niveaux de discrétisation avec localisation permet de conserver une procédure de raffinement peu coûteuse et l'utilisation de fonctions NURBS permet une implémentation simple dans des codes de calcul FEA existants. Cette méthode est développée au chapitre 2.

1.2 Techniques de résolution multigrille

Les techniques de résolution multigrille ont été introduites pour résoudre efficacement des équations différentielles elliptiques dans le cadre des différences finies. Elles

utilisent plusieurs niveaux de discrétisation du problème afin d'accélérer la résolution. Un algorithme de résolution utilisant plusieurs niveaux de discrétisation a été présenté dans [FED 64] et a été développé par [BRA 77, BRA 11]. Les techniques de localisation ont été mises en place par [BAI 87].

Ces méthodes ont été adaptées aux problèmes de la mécanique du solide dans le cadre de la méthode des éléments finis par [PAR 90a, PAR 90b]. De nombreux algorithmes de résolution multigrille adaptés à différents types de problèmes de la mécanique du solide ont été développés. On peut notamment citer les travaux [KAC 93, BIO 10, WIE 99] pour les problèmes élastoplastiques, [RAN 09] pour la simulation de propagation de fissures, [FEN 00] pour la présence de non linéarité géométriques. L'algorithme du Full Approximation Scheme (FAS) est très largement utilisé pour la résolution de problèmes non linéaires. On peut en trouver une description complète dans [FIS 95, VEN 00].

Couplées à un indicateur ou estimateur d'erreur, les techniques de résolutions multigrille peuvent être utilisées comme un algorithme de résolution à raffinement adaptatif très efficace [CAV 05, BIO 10].

1.2.1 Principe

On s'intéresse ici à la résolution d'un système matriciel $\mathbb{K}\mathbf{U} = \mathbf{F}$ associé à un problème d'élasticité linéaire en statique dans le cadre de l'hypothèse de petits déplacements (voir équations 1.3).

Les techniques de résolution multigrille sont basées sur les propriétés de lissage des solveurs itératifs. En effet, les solveurs itératifs, tel que le gradient conjugué par exemple, sont très efficaces pour déterminer en très peu d'itérations les composantes en hautes fréquences spatiales de la solution. Le reste des efforts de calculs sont déployés pour calculer les composantes en basse fréquence spatiale de cette solution. Cette propriété des solveurs itératifs peut être mise en évidence avec l'exemple suivant :

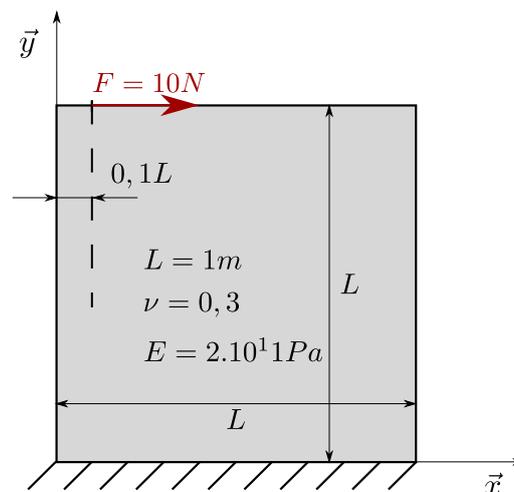


FIGURE 1.15: Problème de référence.

1. État de l'art

Le problème est discrétisé avec des éléments finis linéaires pour un maillage régulier comportant 4096 éléments quadrilatéraux. Le système matriciel associé de 8450 degrés de libertés est résolu à l'aide du solveur itératif gradient conjugué initialisé avec un champ de déplacement nul sur tout le domaine. Nous allons nous intéresser aux transformées de Fourier discrète (voir Annexe A) de l'erreur en déplacement suivant \vec{x} et suivant \vec{y} après 0, 1, 5 et 50 itérations.

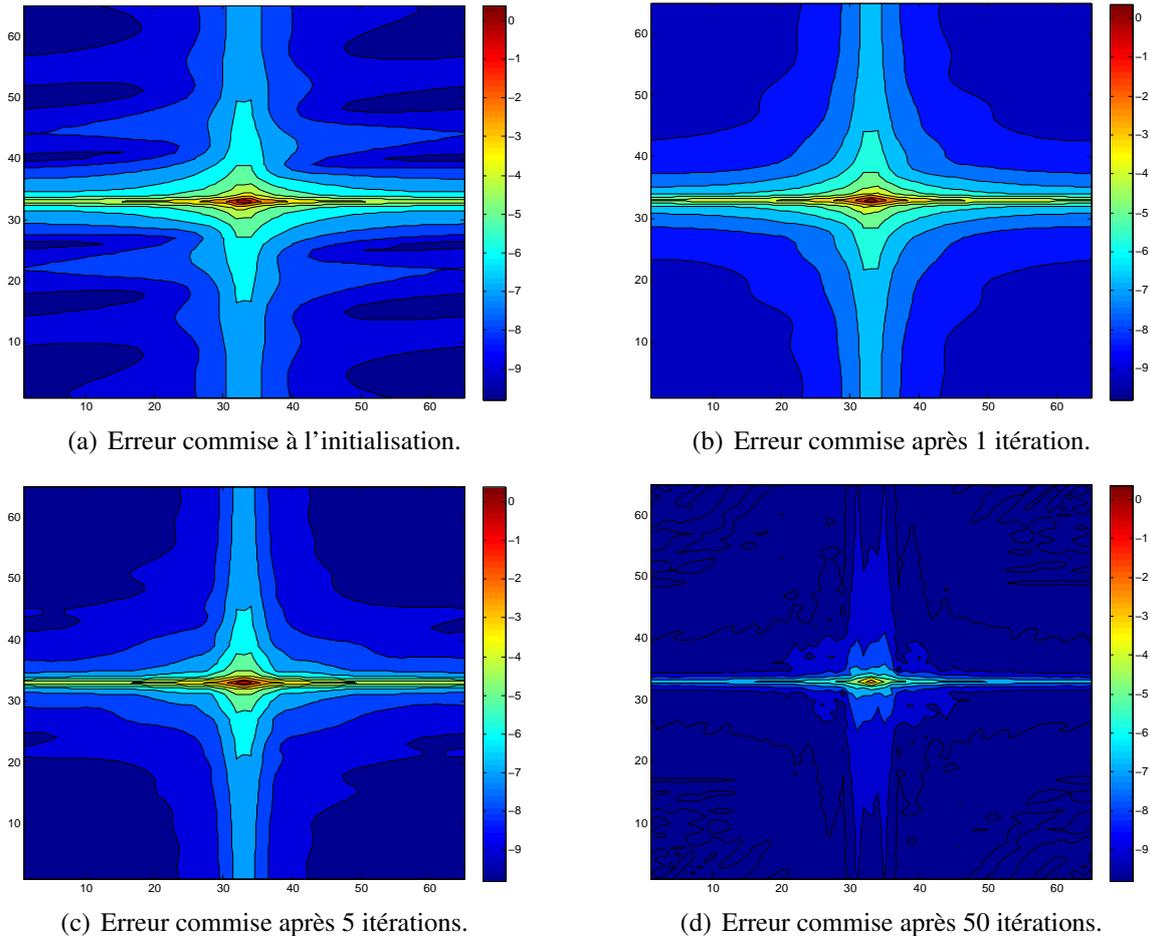


FIGURE 1.16: Transformée de Fourier discrète du champ de déplacement suivant \vec{x} .

On constate que les composantes hautes fréquences de l'erreur sont corrigées très rapidement contrairement aux composantes basses fréquences.

Une façon d'exploiter cette propriété est d'effectuer la résolution du problème 1.15 en utilisant deux discrétisations différentes du problème : une discrétisation fine $\mathbb{K}_f \mathbf{U}_f = \mathbf{F}_f$ et une grossière $\mathbb{K}_c \mathbf{U}_c = \mathbf{F}_c$.

Cette méthode de résolution, le Correction Scheme (CS) est une méthode de résolution itérative sur deux grilles. L'objectif est de résoudre l'équation d'équilibre $\mathbb{K}_f \mathbf{U}_f = \mathbf{F}_f$ sur la grille fine. Un cycle de résolution sur deux grilles donne :

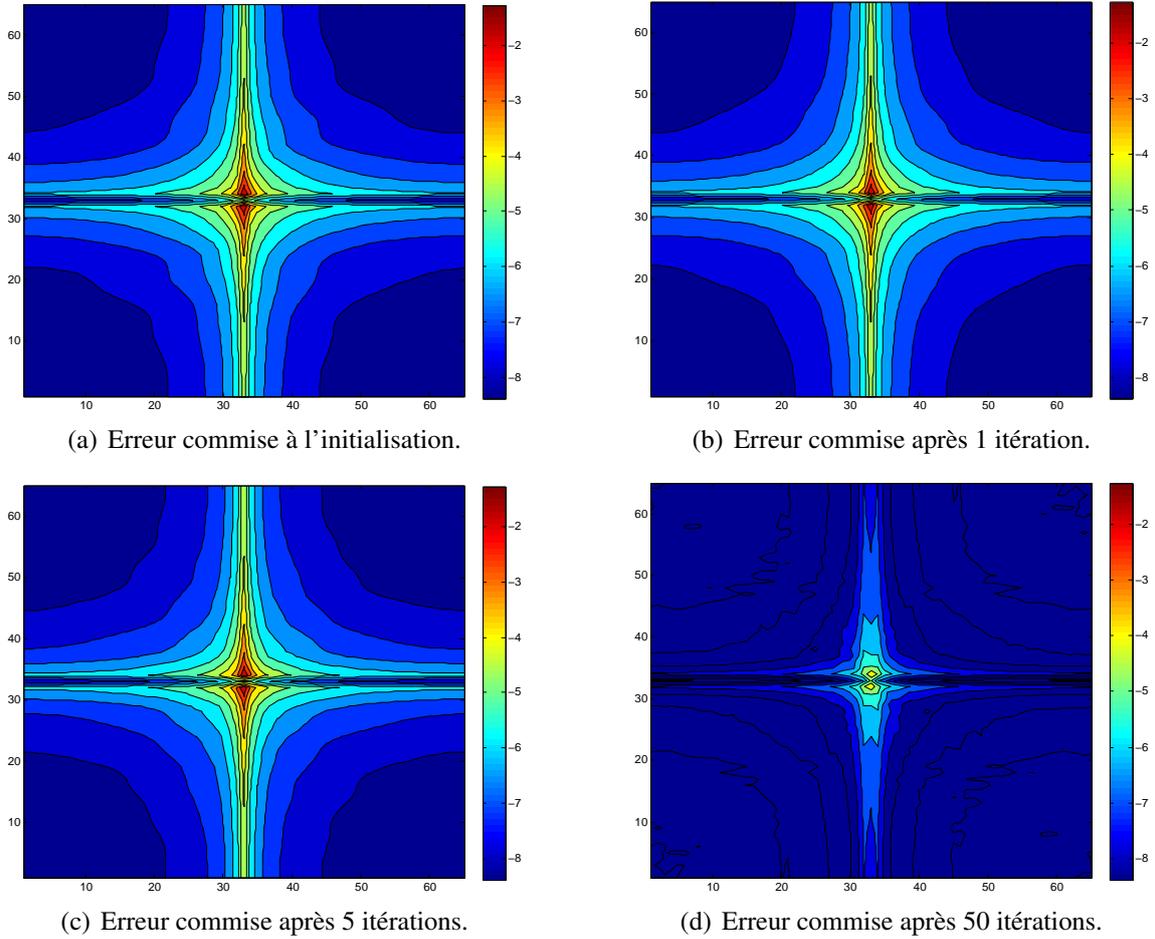


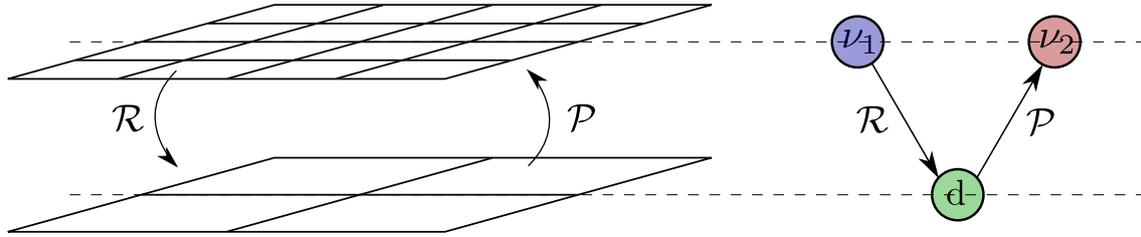
FIGURE 1.17: Transformée de Fourier discrète du champs de déplacement suivant \vec{y} .

1. effectuer ν_1 itérations de lisseur initialisé avec la solution \mathbf{U}_f^0 sur le problème fin. On obtient une solution $\bar{\mathbf{U}}_f^0$.
2. En supposant que le lisseur présente bien les propriétés exposées précédemment, la différence \mathbf{E}_f^0 entre $\bar{\mathbf{U}}_f^0$ et la solution exacte sur grille fine \mathbf{U}_f ($\mathbf{E}_f^0 = \mathbf{U}_f - \bar{\mathbf{U}}_f^0$) est une erreur en basses fréquences. \mathbf{E}_f^0 peut donc être calculé à partir de la discrétisation grossière du problème.
3. Le résidu $\mathbf{R}_f^0 = \mathbf{F}_f - \mathbb{K}_f \bar{\mathbf{U}}_f^0$ est donc transféré sur la grille grossière à l'aide d'un opérateur de restriction \mathcal{R} et la correction basses fréquences \mathbf{E}_c^0 est calculés sur la grille grossière à l'aide d'une résolution directe $\mathbb{K}_c \mathbf{E}_c^0 = \mathcal{R}(\mathbf{R}_f^0)$.
4. Cette correction basse fréquence est ensuite transférée sur la grille fine à l'aide d'un opérateur de prolongement \mathcal{P} et est utilisée pour corriger la solution sur grille fine $\bar{\mathbf{U}}_f^0 : \hat{\mathbf{U}}_f^0 = \bar{\mathbf{U}}_f^0 + \mathcal{P}(\mathbf{E}_c^0)$.
5. Enfin, ν_2 itérations de lisseur initialisé avec $\hat{\mathbf{U}}_f^0$ sont effectuées sur le problème fin

1. État de l'art

afin de corriger les erreurs hautes fréquences introduites par l'opération de prolongement. Cela nous donne la solution \mathbf{U}_f^1 au problème fin.

Ce cycle de résolution multigrille est répété jusqu'à convergence de la solution. L'algorithme CS est donné figure 1.18.



Algorithm 1: Correction Scheme

```

 $k \leftarrow 0; \mathbf{U}_f^0 \leftarrow \mathbf{0};$ 
tant que  $res > \varepsilon$  faire
     $\bar{\mathbf{U}}_f^k \leftarrow \text{lissage}(\mathbb{K}_f, \mathbf{U}_f^k, \mathbf{F}_f, \mathbf{v}_1);$ 
     $\mathbf{R}_f^k \leftarrow \mathbf{F}_f - \mathbb{K}_f \bar{\mathbf{U}}_f^k;$ 
     $\mathbf{E}_c^k \leftarrow \text{résolution}(\mathbb{K}_c \mathbf{E}_c^k = \mathcal{R}(\mathbf{R}_f^k));$ 
     $\hat{\mathbf{U}}_f^k \leftarrow \bar{\mathbf{U}}_f^k + \mathcal{P}(\mathbf{E}_c^k);$ 
     $\mathbf{U}_f^{k+1} \leftarrow \text{lissage}(\mathbb{K}_f, \hat{\mathbf{U}}_f^k, \mathbf{F}_f, \mathbf{v}_2);$ 
     $res \leftarrow \text{critère}(\mathbb{K}_f, \mathbf{U}_f^{k+1}, \mathbf{F}_f);$ 
     $k \leftarrow k + 1;$ 
 $\mathbf{U}_f \leftarrow \mathbf{U}_f^k;$ 

```

FIGURE 1.18: Algorithme de Correction Scheme.

L'efficacité de cette résolution sur deux grilles peut être mise en évidence en traçant l'évolution de la norme du résidu \mathbf{R}_f^k en fonction du nombre de cycles multigrille avec et sans prise en compte de la correction provenant de la grille grossière lors de la résolution sur deux grilles du problème 1.15 (Figure 1.19). Avec la prise en compte de cette correction, le problème est résolu avec une précision de 1.10^{-10} en 6 itérations multigrille contre environ 2000 sans prise en compte de la correction.

Cet algorithme peut être généralisé à N grilles. En effet, si la résolution directe sur la grille grossière est trop coûteuse, il est possible d'appliquer le Correction Scheme de manière récursive avec un nombre de cycle multigrille limité γ . La figure 1.20 présente un cycle multigrille pour trois grilles avec $\gamma = 1$ (V-cycles) et $\gamma = 2$ (W-cycles).

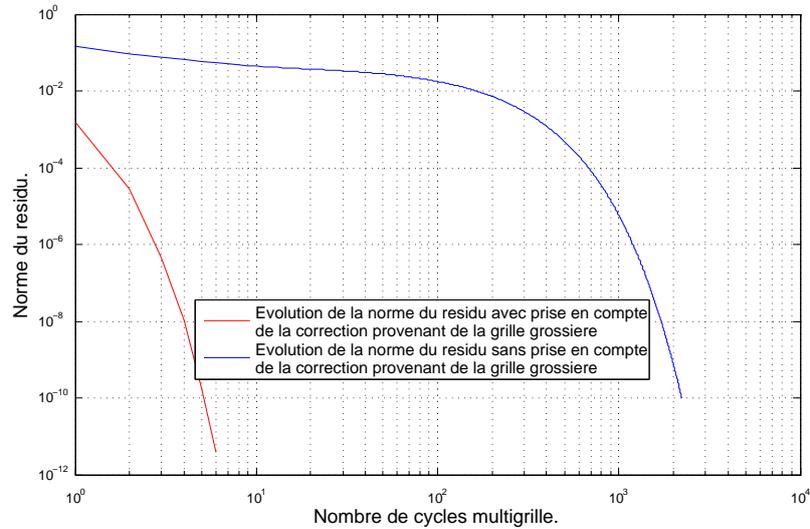


FIGURE 1.19: Efficacité de la prise en compte de la correction provenant de la grille grossière.

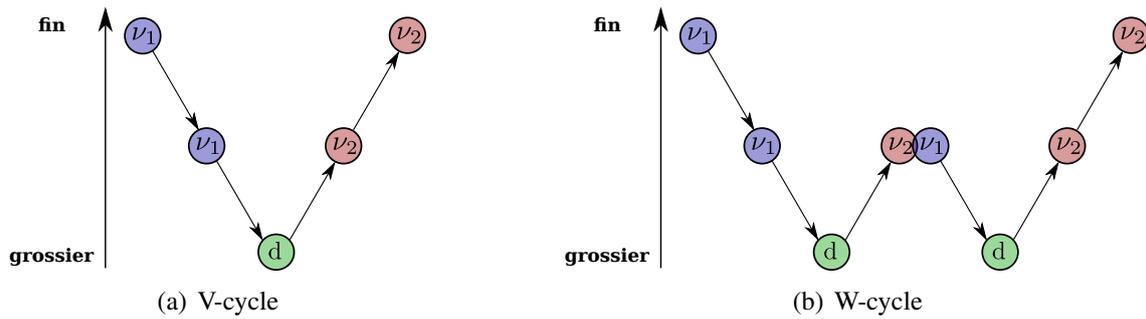


FIGURE 1.20: Cycle multi grille pour trois grilles.

1.2.2 Opérateurs de changement d'échelle

Dans la section précédente, les opérateurs de prolongement \mathcal{P} et de restriction \mathcal{R} agissant respectivement sur les quantités primales (U) et les quantités duales (R) ont été introduits mais n'ont pas été définis. La figure 1.21 représente les relations entre les différentes quantités en jeu dans le cas d'une résolution sur deux grilles.

Une condition que l'on peut écrire sur ces opérateurs est la conservation de l'énergie de déformation pour n'importe quel couple (U_c, F_f) :

$$\begin{aligned}
 \langle \mathbf{U}_c, \mathbf{F}_c \rangle &= \langle \mathbf{U}_f, \mathbf{F}_f \rangle, \quad \forall (\mathbf{U}_c, \mathbf{F}_f) \\
 \Leftrightarrow \mathbf{U}_c^T \mathcal{R} \mathbf{F}_f &= \mathbf{U}_c^T \mathcal{P}^T \mathbf{F}_f, \quad \forall (\mathbf{U}_c, \mathbf{F}_f) \\
 \Leftrightarrow \mathcal{R} &= \mathcal{P}^T
 \end{aligned} \tag{1.9}$$

Cette condition impose une relation entre \mathcal{P} et \mathcal{R} , soit $\mathcal{R} = \mathcal{P}^T$. Pour un opérateur de prolongement \mathcal{P} donné l'opérateur de restriction \mathcal{R} est donc imposé. En général, dans le

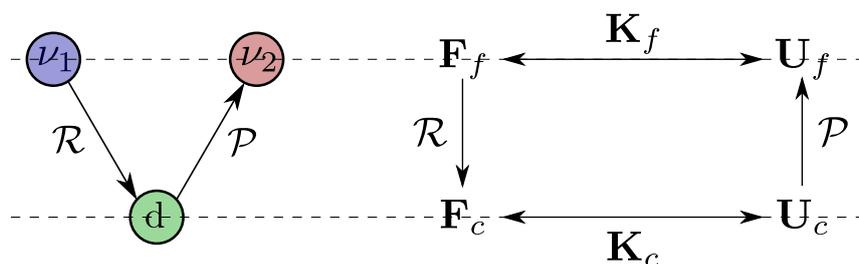


FIGURE 1.21: Relations entre les quantités primates et duales pour deux grilles.

cas d'une résolution éléments finis, l'opération de prolongement est construite par interpolation.

Remarque : Il est possible d'imposer un autre type de conditions : quel que soit U_c , un cycle de prolongement, résolution sur grille fine, restriction et résolution sur grille grossière doit garder U_c inchangé.

$$\begin{aligned}
 \mathbb{K}_c \mathbf{U}_c &= \mathbf{F}_c, \quad \forall \mathbf{U}_c \\
 \Leftrightarrow \mathbb{K}_c \mathbf{U}_c &= \mathcal{R} \mathbb{K}_f \mathcal{P} \mathbf{U}_c, \quad \forall \mathbf{U}_c \\
 \Leftrightarrow \mathbb{K}_c &= \mathcal{R} \mathbb{K}_f \mathcal{P}
 \end{aligned} \tag{1.10}$$

Cela donne une condition entre les matrices de rigidité des deux grilles (équation 1.10). Les techniques de résolution multigrille utilisant ce type d'approche sont les techniques multigrille algébriques [RUG 87, BRI 00, ADA 02]. Ce type d'approche ne sera pas utilisé dans les travaux présentés dans les chapitres 2 et 3. En effet, les techniques multigrille seront utilisées ici pour effectuer du raffinement adaptatif ce qui implique d'ajouter des grilles fines au problème au cours du calcul. Cela nécessiterait de recalculer toutes les matrices de rigidité à chacun de ces ajouts ce qui représente des coûts de calculs non négligeables.

1.2.3 Localisation

Les méthodes présentées précédemment travaillent sur plusieurs niveaux de discrétisations tous définis sur l'ensemble du domaine considéré. Les quantités d'intérêt (le déplacement aux nœuds dans le cas d'un problème de statique élastique linéaire) sont calculées dans des zones ayant une même finesse de discrétisation sur différents niveaux de raffinement. Un exemple pour trois niveaux de grilles est donné figure 1.22. Dans ce cas, une résolution multigrille mène au calcul des quantités d'intérêt dans la zone bleue qui possède la même finesse de discrétisation pour les trois niveaux de grilles, et dans la zone rouge qui possède la même finesse de discrétisation sur les deux dernières grilles. Il est possible de diminuer le temps de calcul en évitant d'effectuer les calculs dans les zones bleues et rouge sur toutes les grilles. C'est le principe de la localisation dans les méthodes de résolution multigrille.

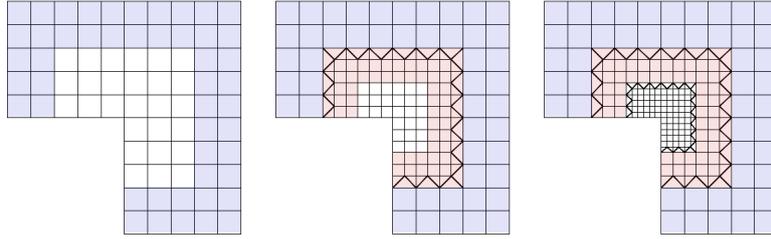


FIGURE 1.22: Raffinement global MG pour trois grilles, d'après [RAN 08].

L'idée est de ne conserver sur une grille que les zones où le raffinement est plus fin que sur la grille précédente (voir figure 1.23). L'entièreté du domaine n'est donc plus discrétisée sur tous les niveaux de grilles. La localisation implique l'apparition de frontières internes entre les différentes grilles. Dans l'exemple présenté figure 1.23, la frontière interne Γ_{AB} entre les grilles Ω_A et Ω_B est $\Gamma_{AB} = \Omega_A \cap \partial\Omega_B \setminus \partial\Omega_A$ et la frontière interne Γ_{BC} entre la grille Ω_B et Ω_C est $\Gamma_{BC} = \Omega_B \cap \partial\Omega_C \setminus \partial\Omega_B$. Pour pouvoir résoudre le problème sur une grille localisée, les conditions aux limites imposées sur les frontières internes seront des conditions aux limites de Dirichlet provenant de la grille précédente. Pour notre exemple, en notant u^{Ω_k} la solution en déplacement sur le domaine Ω_k , on souhaite que u^{Ω_A} , u^{Ω_B} et u^{Ω_C} vérifient :

$$\begin{cases} \mathbf{u}_{|\Gamma_{AB}}^{\Omega_B} = \mathbf{u}_{|\Gamma_{AB}}^{\Omega_A}, \\ \mathbf{u}_{|\Gamma_{BC}}^{\Omega_C} = \mathbf{u}_{|\Gamma_{BC}}^{\Omega_B}. \end{cases} \quad (1.11)$$

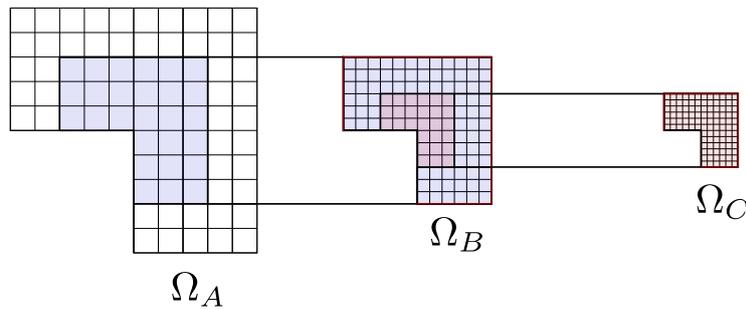


FIGURE 1.23: Raffinement local MG pour trois grilles, d'après [RAN 08].

La solution en déplacement u^Ω sur l'ensemble du domaine Ω sera donc définie de la manière suivante :

$$\begin{cases} \mathbf{u}^\Omega = \mathbf{u}^{\Omega_A} & \text{sur } \Omega_A \setminus \Omega_B, \\ \mathbf{u}^\Omega = \mathbf{u}^{\Omega_B} & \text{sur } \Omega_B \setminus \Omega_C, \\ \mathbf{u}^\Omega = \mathbf{u}^{\Omega_C} & \text{sur } \Omega_C. \end{cases} \quad (1.12)$$

La solution \mathbf{u}^Ω sera donc définie sur le maillage représenté figure 1.24.

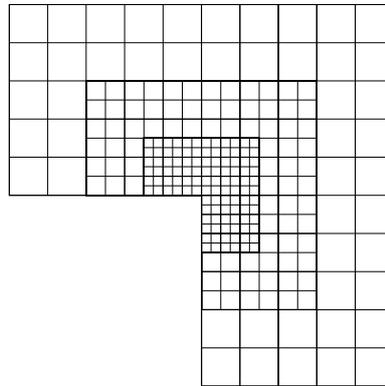


FIGURE 1.24: Maillage sur lequel s'exprime \mathbf{u}^Ω , d'après [RAN 08].

1.2.4 Résolution Full-Multigrid

Le nombre d'itérations à convergence d'une résolution multigrille dépend de la qualité de la solution d'initialisation U_f^0 . Une manière d'éviter cette dépendance est d'utiliser l'algorithme de résolution Full-Multigrid (FMG). Pour une résolution sur deux grilles, il s'agit d'utiliser comme solution d'initialisation sur la grille fine la solution provenant de la résolution du problème sur la grille grossière $U_f^0 = \mathcal{P}(U_c)$ avec $\mathbb{K}_c U_c = \mathbf{F}_c$. La généralisation de l'algorithme FMG à N grilles est représenté figure 1.25. Les solutions encadrées sur la figure 1.25 sont les solutions à convergence sur un niveau de discrétisation donné.

Cette méthode de résolution est conçue pour des problèmes où les différents niveaux de discrétisation sont connus, et notamment le maillage fin. Il est possible de coupler l'algorithme de résolution FMG à un indicateur d'erreur afin de construire un algorithme de résolution à raffinement adaptatif. Dans ce cas, seule la grille la plus fine est connue et les différents niveaux de discrétisation sont construits au fur et à mesure de l'avancement du calcul à l'aide des informations sur la précision de la solution données par l'indicateur d'erreur. Le principe de cette résolution à raffinement adaptatif est représenté figure 1.2.4.

On présentera dans la section suivante l'intérêt du couplage d'une résolution FMG à raffinement adaptatif avec une discrétisation isogéométrique du problème.

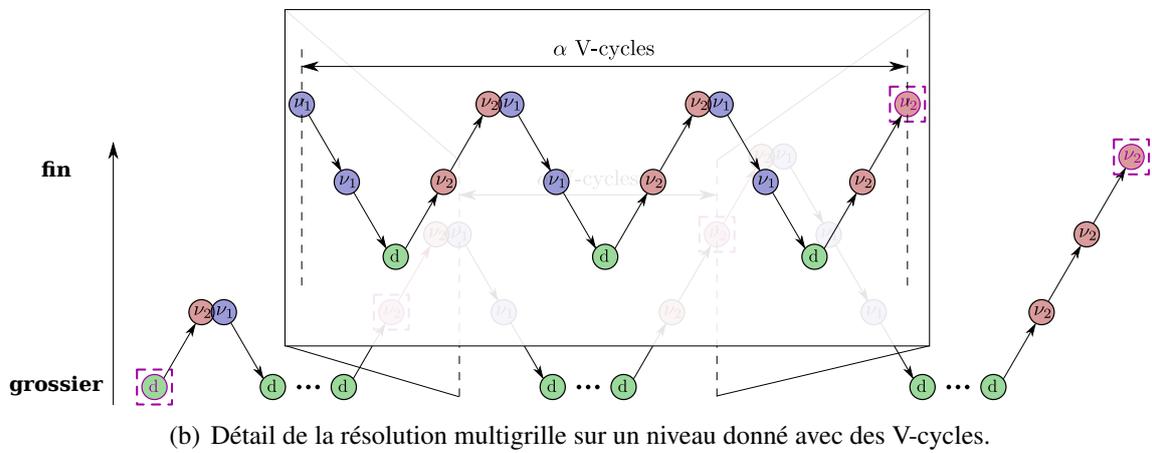
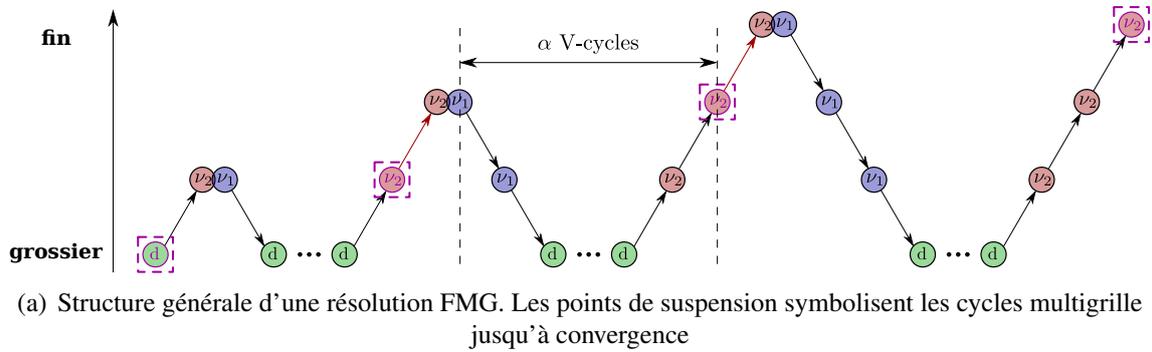


FIGURE 1.25: Résolution FMG sur quatre grilles avec des V-cycles.

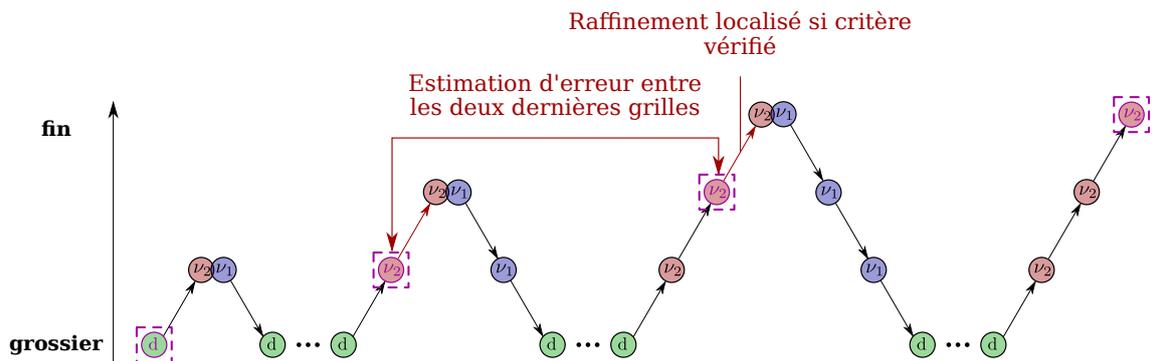


FIGURE 1.26: Algorithme de raffinement automatique basée sur une résolution FMG.

1.2.5 Intérêt du couplage de l'analyse isogéométrique et des stratégies de résolution multigrille

L'utilisation d'une discrétisation isogéométrique implique des problèmes de propagation de raffinement de maillages dus à la structure en produit tensoriel de l'espace d'approximation utilisé, détaillés en section 1.1.4. L'utilisation de stratégies multigrille localisées pour décrire la solution sur une portion du domaine considéré permet d'effectuer un raffinement localisé en IGA en contournant ces problèmes de propagations de raffinement (Figure 1.27).

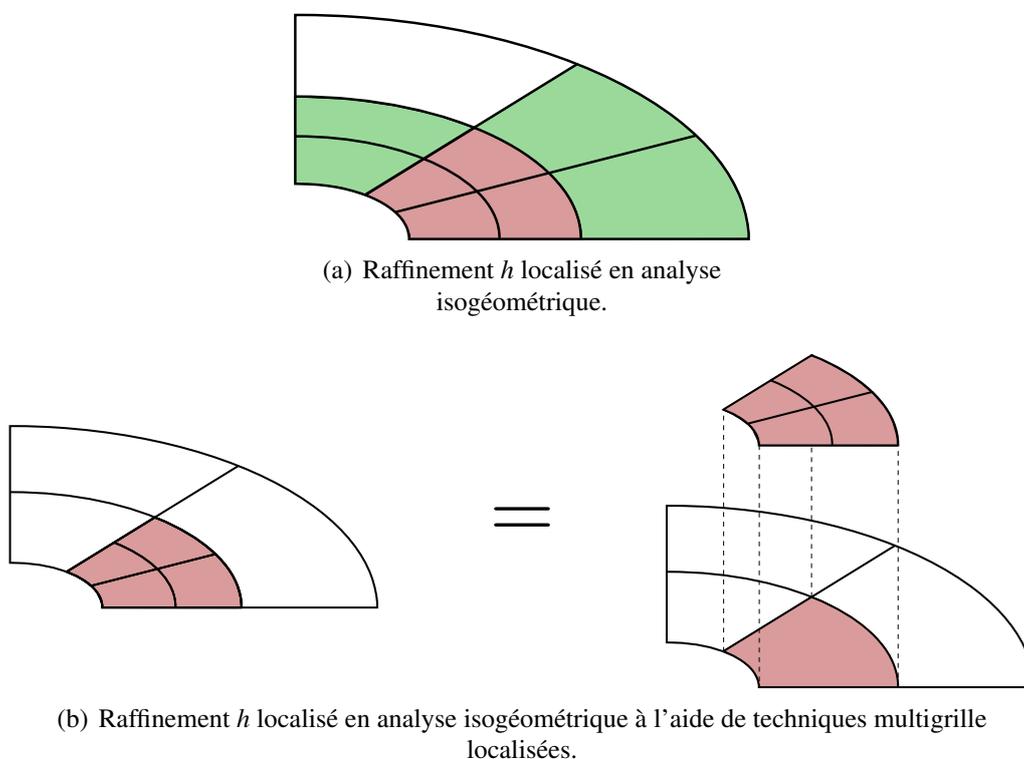


FIGURE 1.27: Principe de raffinement local à l'aide d'une résolution multigrille localisée en analyse isogéométrique.

Les travaux présentés dans le chapitre 2 proposent une méthode de raffinement adaptatif localisé en analyse isogéométrique basée sur une résolution FMG. Les méthodes de résolution FMG étant adaptées aux stratégies de raffinement adaptatif, les outils mis en place pour le raffinement adaptatif seront généralisés pour la résolution de problème de dynamique transitoire avec un adaptativité en espace et en temps.

En section suivante sont présentés quelques stratégies de raffinement adaptatif spatial et temporel existantes.

1.3 Raffinement adaptatif spatial et temporel

On s'intéresse ici à l'utilisation de méthodes à raffinement adaptatif en espace (telle que la résolution FMG couplée à un indicateur d'erreur, présentée en section 1.2.4) dans le cadre de la dynamique transitoire.

Pour les problèmes de dynamique transitoire, la dimension spatiale est classiquement traitée à l'aide de la méthode des éléments finis et la dimension temporelle avec un schéma d'intégration donné. Ces deux dimensions sont donc généralement traitées séparément. Dans le cas des schémas d'intégration temporelle explicites, le pas de temps utilisé est directement lié à la discrétisation spatiale du problème. Plus la discrétisation spatiale est fine, plus le pas de temps utilisé se doit d'être faible afin d'assurer la stabilité du schéma d'intégration [HUG 78].

L'utilisation de méthodes de raffinement adaptatif en espace implique la création de maillages de plus en plus fins dont la taille de maille finale n'est pas connue. Dans le cadre de la dynamique transitoire, le pas de temps du schéma d'intégration est donc diminué à chaque raffinement et mène à des coûts de calculs prohibitifs si le pas de temps final est trop faible. Adapter le pas de temps aux différentes zones de discrétisation spatiale devient donc nécessaire. Ces méthodes consistant à définir un pas de temps différent suivant les degrés de liberté en espace sont les méthodes de sous-cyclage [DAN 98b, SMO 88, SMO 00] et sont particulièrement adaptées pour des problèmes de couplage de modèles. Les problèmes de stabilité liés à ces méthodes sont l'objet de nombreuses études, par exemple [SMO 96, SMO 98, DAN 98a].

Ce type de problèmes peut aussi être résolu à l'aide de techniques de décomposition de domaines. On peut notamment citer la méthode FETI (Finite Element Tearing and Interconnecting method) pour laquelle chaque sous domaine peut posséder une discrétisation spatiale et une discrétisation temporelle différentes de celles des autres sous domaines [FAR 91]. La méthode FETI est par la suite adaptée aux problèmes de dynamique non linéaire [GRA 00, GRA 01, GRA 03, COM 01]. Il y est montré que la stabilité de ces méthodes est assurée si la continuité des vitesses est assurée aux interfaces entre les différents sous domaines.

Ces méthodes de décomposition de domaine sont très efficaces car elles permettent d'adapter la discrétisation temporelle à la discrétisation spatiale des différents sous domaines. Cependant, elles présentent l'inconvénient de nécessiter une intervention de l'utilisateur pour la création et la détermination des caractéristiques des sous domaines.

La STAR-method (Space Time Adaptive Refinement method) [CAV 05, CAV 06] propose une stratégie de raffinement adaptatif en espace et en temps, basée sur la création de domaines espace temps localisés. Elle présente l'avantage de ne nécessiter aucune intervention de l'utilisateur pour la création de ces domaines et la détermination de leurs caractéristiques de discrétisation spatiale et temporelle.

Un autre type d'approches pour la résolution de problèmes de dynamique transitoire est la formulation éléments finis espace temps [HUG 88, HUL 90, HUG 96]. Ce type d'approche se prête bien aux méthodes de raffinement adaptatif (voir par exemple [ABE 04, ABE 06]). Cependant l'inconvénient de telles méthodes est la complexité de la

génération de maillages espaces temps cohérents, notamment afin de respecter le principe de causalité [ERI 05].

Des méthodes à raffinement adaptatifs espace temps dans le cas d'une discrétisation spatiale effectuée par la méthode des éléments finis et l'intégration temporelle par un schéma de Galerkin discontinu en temps sont proposées dans [WIB 99, EKE 01] et [AUB 99, TIE 03]. Elles utilisent des indicateurs d'erreur a posteriori et nécessitent donc un calcul complet sur tout le domaine spatial à chaque pas de temps pour chaque étape de raffinement. On peut finalement citer les travaux [YUE 05] présentant une stratégie de raffinement adaptatif en espace et en temps basée sur une méthode de superposition de maillages [FIS 92, FIS 95], proche de la STAR-method mise en évidence précédemment.

1.4 Objectifs

Les travaux présentés dans ce manuscrit ont pour objectif de mettre au point une méthode à raffinement adaptatif espace temps pour des problèmes d'élasticité linéaire en dynamique transitoire dans le cadre de l'analyse isogéométrique. L'utilisation de l'analyse isogéométrique permet de supprimer l'erreur introduite par l'approximation de la géométrie, et la méthode à raffinement adaptatif a pour but de permettre le contrôle de l'erreur commise lors de la discrétisation et d'atteindre un niveau d'erreur donné.

Pour cela, une méthode de localisation en espace basée sur une résolution multigrille pour une discrétisation IGA est tout d'abord mise en place au chapitre 2. Cette méthode de résolution en espace est ensuite généralisée à des domaines espace temps au chapitre 3.

Chapitre 2

Résolution multigrille localisée appliquée à l'analyse isogéométrique en statique linéaire

Sommaire

2.1	Principe – intérêt de la méthode	34
2.2	Outils développés pour une résolution full-multigrid localisée dans le cadre de l'analyse isogéométrique	34
2.2.1	Opérateur de prolongement	34
2.2.2	Indicateur d'erreur	40
2.3	Validation numérique de la méthode	45
2.3.1	Comparaison entre la résolution full-multigrid globale et localisée	46
2.3.2	Performances de la résolution full multigrid localisé	48
2.4	Synthèse	55

2.1 Principe – intérêt de la méthode

L'algorithme Full-Multigrid appliqué aux éléments finis et combiné à un indicateur d'erreur a déjà été développé comme solveur à raffinement adaptatif [RAN 09, CAV 05, BIO 10]. La méthode présentée ici permet d'effectuer du raffinement h localisé dans le cadre de l'IGA. Il est montré en section 1.1 que la structure tensorielle de la définition des fonctions de base en analyse isogéométrique implique une propagation du raffinement h dans tout le domaine. On remarquera aussi que la méthode de résolution FMG à raffinement adaptatif appliquée aux éléments finis pose problème pour le raffinement des frontières courbes du domaine considéré. En effet, il est nécessaire de conserver les informations liées à la géométrie CAO du domaine et de passer par des étapes de projection qui peuvent être délicates suivant la complexité des géométries ([ADA 00] and [BIB 13]).

Coupler l'analyse isogéométrique à une résolution multigrille présente donc a priori deux avantages majeurs :

- l'utilisation de techniques multigrille permet de contourner ce problème de propagation de raffinement et d'effectuer un raffinement h localisé du domaine,
- l'utilisation de l'analyse isogéométrique permet de conserver une description exacte de la géométrie et donc de contourner le problème de raffinement de frontières courbes.

La méthode développée et exposée ici est donc un principe de résolution Full-multigrid appliqué à l'analyse isogéométrique combinée à un indicateur d'erreur afin d'effectuer un raffinement adaptatif pour respecter un critère de précision imposé par l'utilisateur. Il faut pour cela définir de nouveaux opérateurs de changement d'échelle adaptés à l'IGA et développer un indicateur d'erreur. Ces outils seront présentés en section 2.2. Les performances de cette méthode de résolution seront présentés sur différents exemples en section 2.3.

2.2 Outils développés pour une résolution full-multigrid localisée dans le cadre de l'analyse isogéométrique

Pour mettre en place un algorithme de résolution Full-Multigrid localisé dans le cadre de l'analyse isogéométrique, il est nécessaire de définir les opérateurs de changement d'échelle qui seront utilisés, ainsi que l'indicateur d'erreur qui pilotera le raffinement automatique. Dans cette section seront détaillées la construction de l'opérateur de prolongement, la définition de l'indicateur d'erreur global caractérisant la convergence de l'algorithme et la construction de l'indicateur d'erreur local indiquant les éléments du maillage à raffiner.

2.2.1 Opérateur de prolongement

Pour transférer l'information entre deux grilles hiérarchiques successives, une grille fine Ω_f et une grille grossière Ω_c , il faut définir un opérateur de prolongement \mathcal{P} et un

opérateur de restriction \mathcal{R} . Leur construction dans le cas d'une résolution multigrilles éléments finis à déjà été détaillée en section 1.2, notamment le fait que $\mathcal{R} = \mathcal{P}^T$. En notant \mathbf{U}_f les variables de contrôle du déplacement sur la grille fine, et \mathbf{U}_c les variables de contrôle du déplacement sur la grille grossière, on souhaite que \mathcal{P} vérifie :

$$\mathbf{U}_f = \mathcal{P}(\mathbf{U}_c). \quad (2.1)$$

Le cas des fonctions de base NURBS de degré 1 correspond aux fonctions de base éléments finis de degré 1. L'opérateur de prolongement utilisé pour des éléments finis de degré 1 est classiquement basé sur l'interpolation. Dans ce cas, le calcul des nouveaux points de contrôle peut être effectué élément par élément et l'opérateur de prolongement peut être construit a priori pour un élément et une règle de raffinement donnée (voir Figure 2.1).

Pour des fonctions de base NURBS de degré supérieur, le processus de raffinement h d'un élément dépend de ses voisins. L'opérateur de prolongement ne peut donc pas être connu a priori et dépend des éléments à raffiner (voir Figure 2.1). Sa construction est facilitée par le fait que les éléments isogéométriques sont isoparamétriques. Le processus de raffinement du maillage fournit le lien entre les points de contrôle des deux grilles. On peut donc utiliser ce lien pour construire l'opérateur de prolongement.

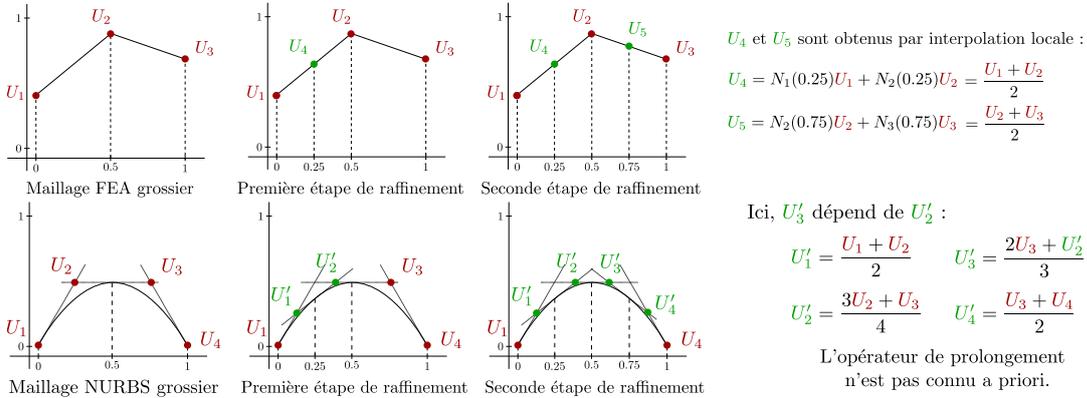


FIGURE 2.1: Raffinement pour des fonctions de base éléments finis de degré 1 et des fonctions de base NURBS de degré 2.

Soit Ω_c la grille grossière provenant d'une discrétisation NURBS du domaine Ω et Ω_f la grille fine provenant d'un raffinement h de Ω_c (Figure 2.2). On pose \mathbf{Q}_c et \mathbf{Q}_f respectivement les points de contrôle associés à Ω_c et Ω_f . Si \mathcal{P} vérifie $\mathbf{Q}_f = \mathcal{P}(\mathbf{Q}_c)$, alors il vérifiera $\mathbf{U}_f = \mathcal{P}(\mathbf{U}_c)$. L'opérateur de prolongement choisi est obtenu sous forme matricielle durant la procédure de raffinement de la géométrie NURBS présentée dans [COH 01].

Dans le cas d'une géométrie 2D, le raffinement h est effectué en deux étapes. Un raffinement h est réalisé dans la direction paramétrique η donnant une grille intermédiaire,

2. Résolution multigrille localisée appliquée à l'analyse isogéométrique en statique linéaire

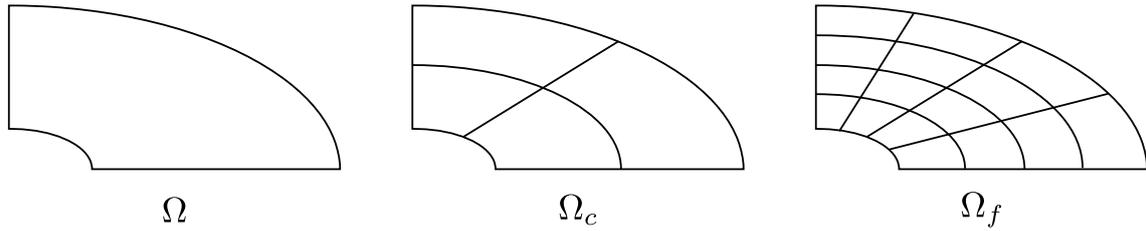


FIGURE 2.2: Définition des domaines Ω , Ω_c et Ω_f .

puis un raffinement h suivant la deuxième direction paramétrique ξ sur la grille intermédiaire est réalisé. Deux opérateurs de prolongement intermédiaires \mathcal{P}_η et \mathcal{P}_ξ sont définis. Ils sont respectivement les opérateurs de prolongement dans les directions η et ξ (Figure 2.3). Finalement nous obtenons :

$$\mathcal{P} = \mathcal{P}_\xi \circ \mathcal{P}_\eta. \quad (2.2)$$

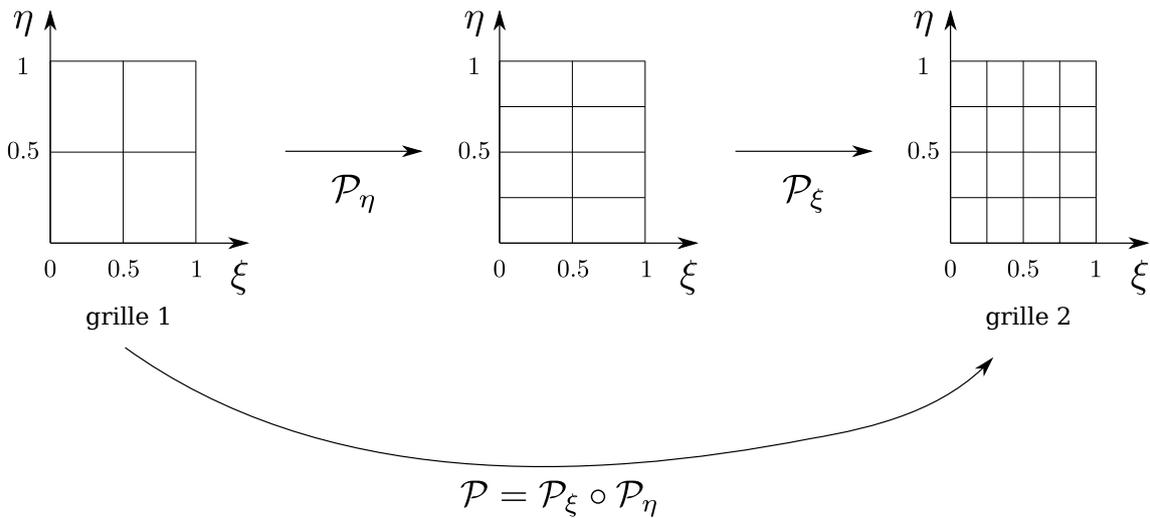


FIGURE 2.3: Procédure de raffinement.

L'utilisation de la décomposition $(\mathcal{P}_\xi, \mathcal{P}_\eta)$ de l'opérateur \mathcal{P} permet un important gain de mémoire, surtout dans le cas d'un problème en trois dimensions. La possibilité d'une telle décomposition est une conséquence directe de la structure produit tensoriel d'une discrétisation NURBS (voir section 1.1).

La construction de l'opérateur de prolongation présentée précédemment est réalisable dans le cas où Ω_c et Ω_f sont deux discrétisations du même domaine. Nous avons vu en section 1.2 que les résolutions multigrilles localisées sont plus efficaces que les résolutions multigrilles globales. Il nous faut donc définir un opérateur de prolongement local entre deux grilles Ω_c et Ω_f ne couvrant pas les mêmes portions du domaine Ω , telles que $\Omega_f \subset \Omega_c$.

Afin de définir cet opérateur de prolongement local, une adaptation de la méthode décrite précédemment est nécessaire. Il faut que cet opérateur permette l'extraction d'une zone de Ω_c et qu'il assure le même degré de continuité de la solution au passage de la frontière interne Γ entre Ω_c et Ω_f (Figure 2.4).

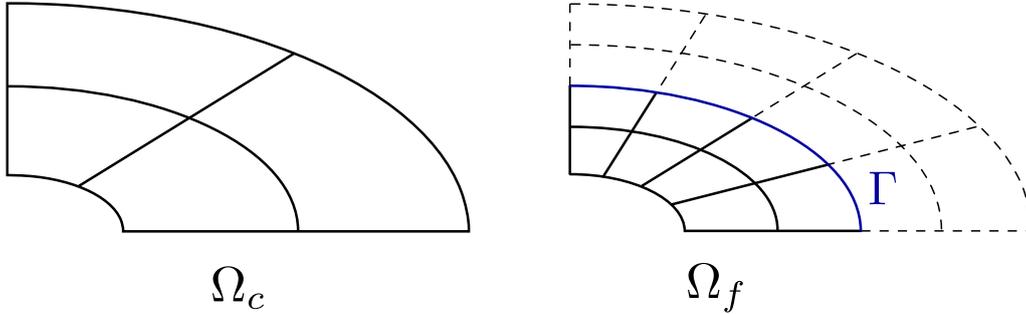


FIGURE 2.4: Définition des domaines Ω_c , Ω_f et de la frontière interne Γ dans le cas de la localisation.

Pour cela, nous allons introduire l'opérateur d'extraction \mathcal{P}_e . Le fonctionnement de cet opérateur est détaillé en figure 2.5. Il s'agit d'un exemple 1D avec des fonctions de base B-splines quadratiques. La zone du domaine à extraire est l'intervalle $[0.375, 0.875]$. Les points de contrôle associés aux fonctions de base en pointillés noirs sont supprimés, les points de contrôle associés aux fonctions de base rouges seront imposés de façon forte afin d'assurer le bon degré de continuité au passage de l'interface entre Ω_c et Ω_f et les points de contrôle associés aux fonctions de base bleues seront les inconnues du problème.

La construction de l'opérateur de prolongement local est détaillé en figure 2.6. Les éléments à raffiner sont colorés en rouge. L'opérateur de prolongement global \mathcal{P}_g entre le deux premières grilles est construit avec la méthode décrite précédemment. Ensuite, le maillage NURBS du plus petit convexe contenant les éléments à raffiner est extrait à l'aide de l'opérateur d'extraction \mathcal{P}_e défini précédemment. L'opérateur de prolongement local \mathcal{P}_{loc} est obtenu par composition de ces deux opérateurs : $\mathcal{P}_{loc} = \mathcal{P}_e \circ \mathcal{P}_g$. On peut noter que chacun de ces opérateurs peut être décomposé suivant les deux directions paramétriques $\mathcal{P}_i = \mathcal{P}_\xi^i \circ \mathcal{P}_\eta^i$, $i \in [e, g]$.

Ce raffinement local pour maillage NURBS permet d'imposer les conditions décrites en section 1.2.3 de façon forte. De plus, la discrétisation de cette équation mène à $\mathbf{U}_\Gamma^f = \mathcal{P}_{loc}(\mathbf{U}_\Gamma^c)$, où Γ est l'interface interne entre Ω_c et Ω_f . L'opérateur assure donc la conservation du degré de continuité au passage de la frontière interne (en bleu sur la figure 2.6). La figure 2.7 est un exemple de création de grilles fines localisées pour un problème unidimensionnel et un problème bidimensionnel.

Dans l'algorithme développé, plusieurs zones du maillage NURBS peuvent être raffinées localement. Un opérateur de prolongement local est construit pour chacune de ces zones. La figure 2.8 détaille la mise en oeuvre de ce procédé.

Nous avons vu en section 1.2 que l'opérateur de restriction \mathcal{R} est défini à partir de l'opérateur de prolongement \mathcal{P} et que $\mathcal{R} = \mathcal{P}^T$. Tous les opérateurs de transfert d'in-

2. Résolution multigrille localisée appliquée à l'analyse isogéométrique en statique linéaire

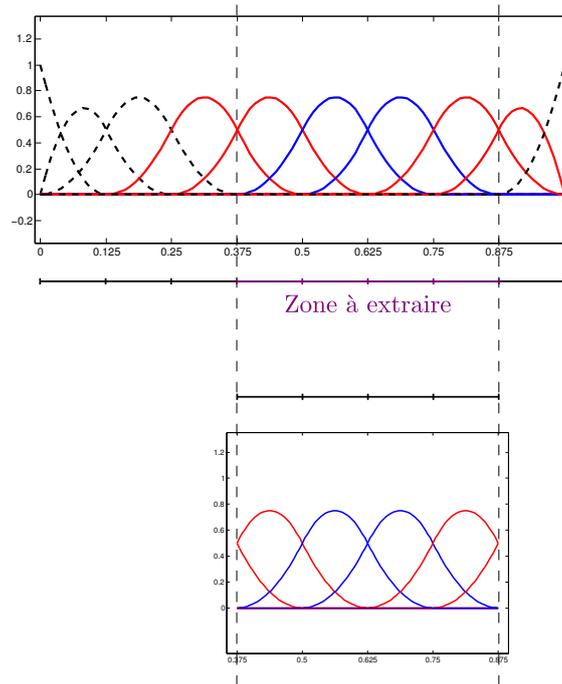


FIGURE 2.5: Opération d'extraction.

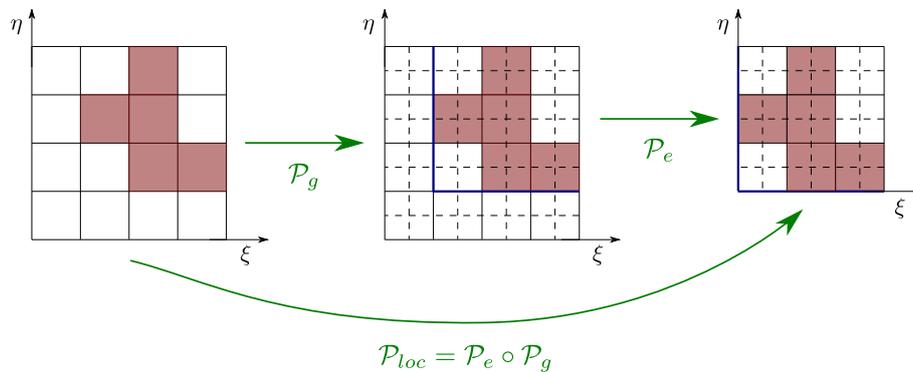


FIGURE 2.6: Méthode de construction d'un opérateur de prolongement localisé.

formations entre deux grilles successives sont donc définis dans le cas d'une résolution multigrilles pour des discrétisations NURBS avec localisation.

Il nous faut maintenant définir un indicateur d'erreur global pour caractériser la convergence de l'algorithme et un indicateur d'erreur local afin de déterminer les zones du domaine à raffiner afin d'avoir un algorithme Full-multigrid localisé pour des discrétisations NURBS opérationnel.

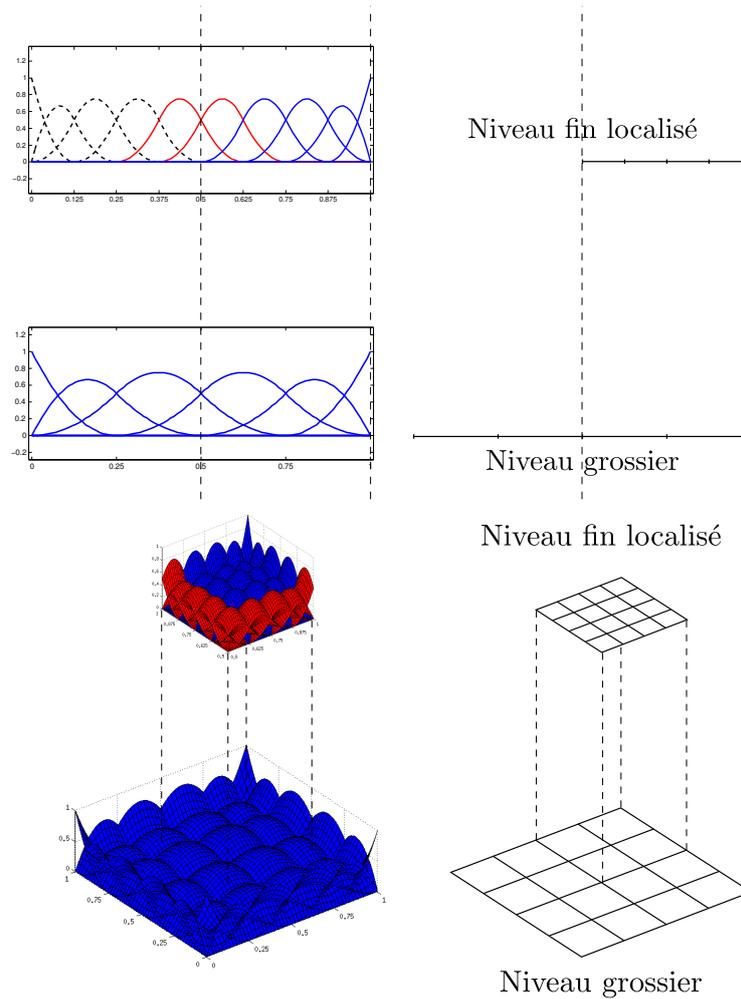


FIGURE 2.7: Exemples de localisation en une et deux dimensions.

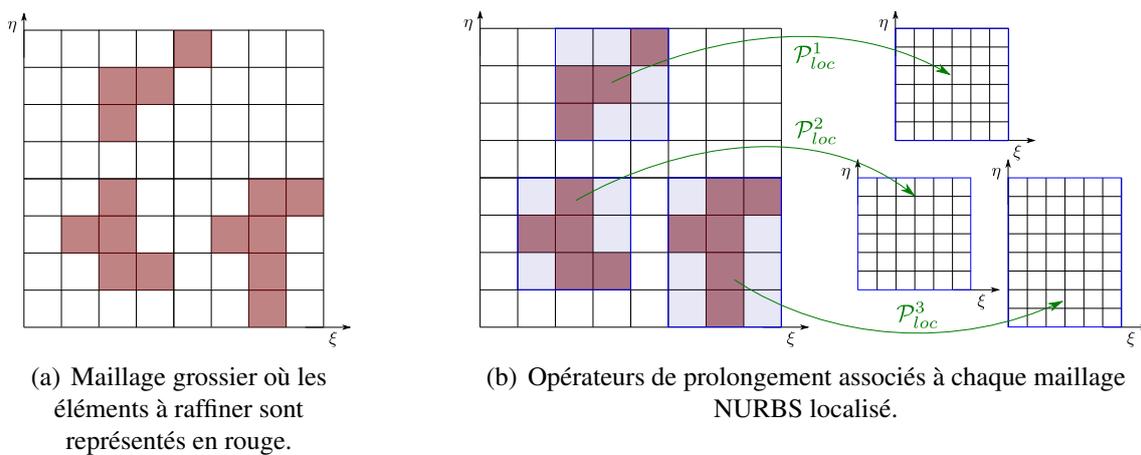


FIGURE 2.8: Création des opérateurs de prolongement durant la procédure de raffinement localisé.

2.2.2 Indicateur d'erreur

Pour un algorithme full-multigrid avec raffinement automatique, il est nécessaire d'avoir une information sur l'erreur entre la solution convergée sur la grille fine et la solution exacte du problème afin de construire une grille plus fine dans les zones où la solution n'est pas assez précise. En pratique, l'information qui nous intéresse n'est pas la valeur exacte de l'erreur commise mais les portions du domaine à raffiner. Nous allons donc utiliser un indicateur d'erreur qui est moins coûteux qu'un estimateur d'erreur. Pour un ensemble de grilles $\{\Omega_1, \Omega_2, \dots, \Omega_k, \Omega_{k+1}\}$ donné, l'indicateur d'erreur de la solution sur la grille Ω_k est calculé à l'aide des solutions obtenues sur les grilles Ω_k et Ω_{k+1} . Pour plus de clarté, une quantité α calculée à partir de la solution converge sur la grille Ω_i sera notée α_i .

Plusieurs types d'indicateurs d'erreurs en espace pour des algorithmes FMG localisés ont déjà été mis à l'épreuve [CAV 05] :

- indicateur d'erreur en déplacement : $\frac{\|\mathbf{u}_{k+1} - \mathbf{u}_k\|_{L^2}}{\|\mathbf{u}_0\|_{L^2}}$,
- indicateur d'erreur en contrainte : $\frac{VM(\sigma_{k+1} - \sigma_k)}{\sigma_0}$,
- indicateur d'erreur en densité d'énergie : $\sqrt{\frac{e_{k+1} - e_k}{e_0}}$,
- indicateur d'erreur en énergie : $\sqrt{\frac{\int_{\Omega} |e_{k+1} - e_k| d\Omega}{\int_{\Omega} e_0 d\Omega}}$,

où \mathbf{u} est le champ de déplacement, σ le tenseur des contraintes associé au champs de déplacement \mathbf{u} ($\sigma = \mathbf{C}\varepsilon$, avec \mathbf{C} le tenseur de Hooke), $VM(\sigma) = \frac{1}{\sqrt{2}} \sqrt{(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2}$ l'opérateur de Von Mises (avec σ_i les contraintes principales) et $e = \mathbf{C}\varepsilon$: ε la densité d'énergie de déformation élastique.

Il est montré dans [CAV 05] que les indicateurs d'erreur ayant le meilleur comportement avec des solutions singulières sont les indicateurs d'erreur en déplacement et en énergie. De plus, dans le cas de problèmes utilisant une discrétisation NURBS, calculer l'indicateur d'erreur en énergie connaissant les matrices de rigidité non assemblées est moins coûteux que de calculer l'indicateur d'erreur en déplacement. En effet, contrairement aux éléments finis, les points de contrôles ne sont pas interpolants (voir section 1.1.2) ce qui implique une réévaluation de toutes les fonctions de base pour déterminer les déplacement en un point du domaine. Nous utiliserons donc comme indicateur d'erreur global un indicateur d'erreur en énergie. En effet, un indicateur d'erreur en énergie tolère la présence d'éventuelles singularités dans la solution (fissures, etc.) [CAV 05]. Afin de mettre en place une stratégie de raffinement automatique, il est nécessaire également d'avoir à notre disposition un indicateur d'erreur local.

Les sections suivantes présentent l'indicateur d'erreur global en énergie et l'indicateur d'erreur local dérivant de celui-ci.

2.2.2.1 Indicateur d'erreur global

Soit un ensemble de grilles $\{\Omega_0, \dots, \Omega_k, \Omega_{k+1}\}$, on définit :

$$\begin{cases} \mathcal{D}^i &= \Omega_i \setminus \Omega_{i+1}, \forall i \in \llbracket 0; k \rrbracket, \\ \mathcal{D}^{k+1} &= \Omega_{k+1}, \\ \overline{\mathcal{D}}^i &= \Omega_i \cap \Omega_{i+1}, \forall i \in \llbracket 0; k \rrbracket, \\ \overline{\mathcal{D}}^{k+1} &= \{\emptyset\}. \end{cases} \quad (2.3)$$

Avec ces notations, il vient :

$$\begin{cases} \bigcup_{i \in \llbracket 0; k+1 \rrbracket} \mathcal{D}_i &= \Omega, \\ \int_{\mathcal{D}_i \cap \mathcal{D}_j} d\Omega &= 0, \forall i \neq j \setminus (i, j) \in \llbracket 0; k+1 \rrbracket^2. \\ \overline{\mathcal{D}}^k &= \mathcal{D}^{k+1} \end{cases} \quad (2.4)$$

La figure 2.9 illustre ces définitions pour un ensemble de quatre grilles.

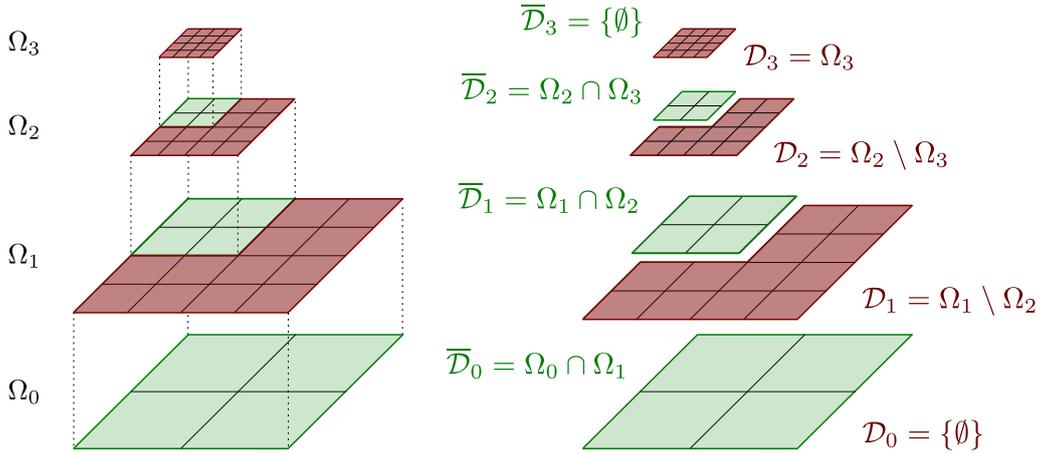


FIGURE 2.9: Définition des domaines pour un ensemble de quatre grilles $\{\Omega_0, \Omega_1, \Omega_2, \Omega_3\}$.

L'indicateur d'erreur en énergie de déformation est utilisé comme indicateur d'erreur global. A cause de la localisation, la quantité $e_i, i \in \llbracket 0; k+1 \rrbracket$ ne peut être définie sur tout le domaine Ω . L'indicateur d'erreur en énergie de déformation η est redéfini de la manière suivante :

$$\eta^2 = \sum_{0 \leq i \leq k+1} \frac{\int_{\mathcal{D}_i} |e_i - e_{i-1}| d\Omega}{\int_{\Omega_0} e_0 d\Omega} \quad (2.5)$$

L'indicateur d'erreur doit vérifier $\eta^2 \leq \varepsilon^2$, où ε est le critère de précision sur toute la structure imposé par l'utilisateur.

Pour chaque grille Ω_i on définit :

$$\left\{ \begin{array}{l} \eta_i^2 = \frac{\int_{\mathcal{D}_i} |e_{i+1} - e_i| d\Omega}{\int_{\Omega} e_0 d\Omega}, \\ \varepsilon_0^2 = \varepsilon^2, \\ \varepsilon_i^2 = \varepsilon_{i-1}^2 - \frac{\int_{\mathcal{D}_i} |e_i - e_{i-1}| d\Omega}{\int_{\Omega} e_0 d\Omega}, \forall i \in \llbracket 1; k \rrbracket. \end{array} \right. \quad (2.6)$$

L'algorithme proposé s'arrête lorsque $\eta_k^2 \leq \varepsilon_k^2$. En effet, si cette condition est vérifiée, le critère de précision global sera aussi vérifié :

$$\begin{aligned} \eta_k^2 &= \eta^2 - \sum_{1 \leq i \leq k} \frac{\int_{\mathcal{D}_i} |e_i - e_{i-1}| d\Omega}{\int_{\Omega} e_0 d\Omega} \leq \varepsilon_k^2 \\ \Leftrightarrow \eta^2 &\leq \varepsilon_k^2 + \sum_{1 \leq i \leq k} \frac{\int_{\mathcal{D}_i} |e_i - e_{i-1}| d\Omega}{\int_{\Omega} e_0 d\Omega} = \varepsilon^2. \end{aligned} \quad (2.7)$$

2.2.2.2 Critère de raffinement local

Pour l'ensemble des grilles $\{\Omega_0, \dots, \Omega_k, \Omega_{k+1}\}$, la grille Ω_{k+2} est construite si le critère d'arrêt $\eta_k^2 \leq \varepsilon_k^2$ n'est pas vérifié. Pour construire la grille Ω_{k+2} , il est nécessaire de définir un critère de raffinement local afin d'identifier les éléments de la grille Ω_{k+1} à raffiner.

L'indicateur d'erreur est calculé sur la grille Ω_k (section 2.2.2). Le critère de raffinement local est appliqué sur la grille Ω_{k+1} . La façon de choisir les éléments de Ω_{k+1} à raffiner est détaillée dans ce qui suit.

On pose :

$$\begin{aligned} \Omega_{\mathcal{D}_k}^e &: \text{élément de la grille } \Omega_k \text{ appartenant au domaine } \overline{\mathcal{D}_k}, \\ (\eta_k^e)^2 &= \frac{\int_{\Omega_{\mathcal{D}_k}^e} |e_{k+1} - e_k| d\Omega}{\int_{\Omega} e_0 d\Omega} \times \frac{\int_{\overline{\mathcal{D}_k}} d\Omega}{\int_{\Omega_{\mathcal{D}_k}^e} d\Omega}. \end{aligned} \quad (2.8)$$

Le critère de raffinement local choisi est de sélectionner tous les éléments $\Omega_{\mathcal{D}_k}^e$ qui ne vérifient pas la condition $(\eta_k^e)^2 \leq \varepsilon_k^2$ pour un raffinement futur. En effet, si tous les éléments de $\overline{\mathcal{D}_k}$ vérifient ce critère, alors :

$$(\eta_k^e)^2 = \frac{\int_{\Omega_{\mathcal{D}_k}^e} |e_{k+1} - e_k| d\Omega}{\int_{\Omega} e_0 d\Omega} \times \frac{\int_{\overline{\mathcal{D}_k}} d\Omega}{\int_{\Omega_{\mathcal{D}_k}^e} d\Omega} \leq \varepsilon_k^2, \quad \forall \Omega_{\mathcal{D}_k}^e \subset \overline{\mathcal{D}_k}. \quad (2.9)$$

Par sommation de cette relation sur tous les $\Omega_{\mathcal{D}_k}^e$, on obtient :

$$\sum_{\Omega_{\mathcal{D}_k}^e \subset \overline{\mathcal{D}_k}} \frac{\int_{\Omega_{\mathcal{D}_k}^e} |e_{k+1} - e_k| d\Omega}{\int_{\Omega} e_0 d\Omega} \leq \varepsilon_k^2 \times \sum_{\Omega_{\mathcal{D}_k}^e \subset \overline{\mathcal{D}_k}} \frac{\int_{\Omega_{\mathcal{D}_k}^e} d\Omega}{\int_{\overline{\mathcal{D}_k}} d\Omega}. \quad (2.10)$$

Sachant que $\mathcal{D}_{k+1} = \overline{\mathcal{D}_k}$, il reste :

$$\left\{ \begin{array}{l} \frac{\int_{\overline{\mathcal{D}_k}} |e_{k+1} - e_k| d\Omega}{\int_{\Omega} e_0 d\Omega} = \frac{\int_{\mathcal{D}_{k+1}} |e_{k+1} - e_k| d\Omega}{\int_{\Omega} e_0 d\Omega} = \eta_k^2, \\ \sum_{\Omega_{\mathcal{D}_k}^e \subset \overline{\mathcal{D}_k}} \frac{\int_{\Omega_{\mathcal{D}_k}^e} d\Omega}{\int_{\overline{\mathcal{D}_k}} d\Omega} = 1. \end{array} \right. \quad (2.11)$$

ce qui impose :

$$\eta_k^2 \leq \varepsilon_k^2.$$

Le respect du critère local $(\eta_i^e)^2 \leq \varepsilon_i^2$ pour tout élément $\Omega_{\mathcal{D}_i}^e$ de $\overline{\mathcal{D}_i}$, $i \in \llbracket 1; k \rrbracket$ mène à la vérification du critère d'arrêt global présenté en section 2.2.2.1, et donc à la convergence de l'algorithme FMG IGA localisé.

La section suivante présente la stratégie de sélection des éléments de Ω_{k+1} à raffiner si le critère d'arrêt de l'algorithme n'est pas vérifié.

2.2.2.3 Stratégie de raffinement local

Les éléments de la grille Ω_k ne respectant pas le critère local nous indique les zones pour lesquelles la solution n'est pas assez précise. On souhaite donc que les fonctions de base ayant support sur la zone à raffiner ne soient pas associées à des points de contrôle correspondant à une condition aux limite de frontière interne (voir figure 2.5). La figure 2.10 illustre cette remarque avec un exemple en une dimension pour des fonctions de base NURBS quadratiques. L'éléments en rouge sur la grille grossière correspond à la zone où la solution n'est pas considérée suffisamment précise. Les éléments de la grille fine correspondant à cette zone sont en rouge également. Les fonctions ayant support sur cette zone doivent être associés à des points de contrôle dont la valeur ne sera pas imposée

2. Résolution multigrille localisée appliquée à l'analyse isogéométrique en statique linéaire

par la solution provenant de la grille fine. Les éléments représentés en vert sur la grille fine sont donc ajoutés à la zone de localisation.

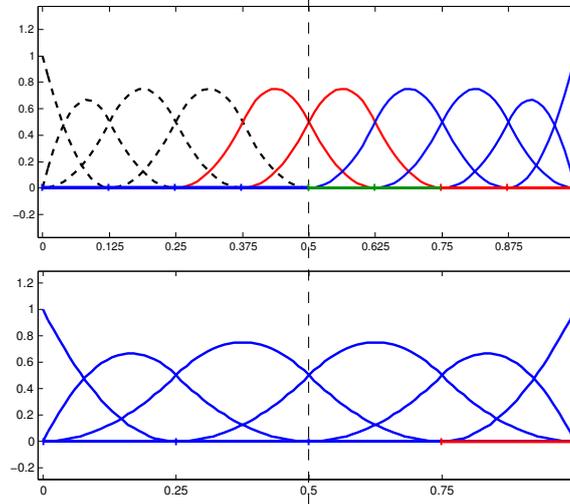


FIGURE 2.10: Création de la zone de localisation pour des fonctions de base NURBS 1D quadratiques.

Dans le cadre de l'analyse isogéométrique, les fonctions de base sont non nulles sur $p + 1$ éléments dans la direction paramétrique ξ et sur $q + 1$ éléments dans la direction paramétrique η . Les p éléments voisins dans la direction paramétrique ξ et les q éléments voisins dans la direction paramétrique η sont donc aussi ajoutés à la liste des éléments de Ω_k à raffiner.

Un exemple d'application de cette stratégie de raffinement appliquée pour un ensemble de grille $(\Omega_1, \Omega_2, \dots, \Omega_k, \Omega_{k+1})$ pour des fonctions quadratiques dans les directions ξ et η est donné en figure 2.11.

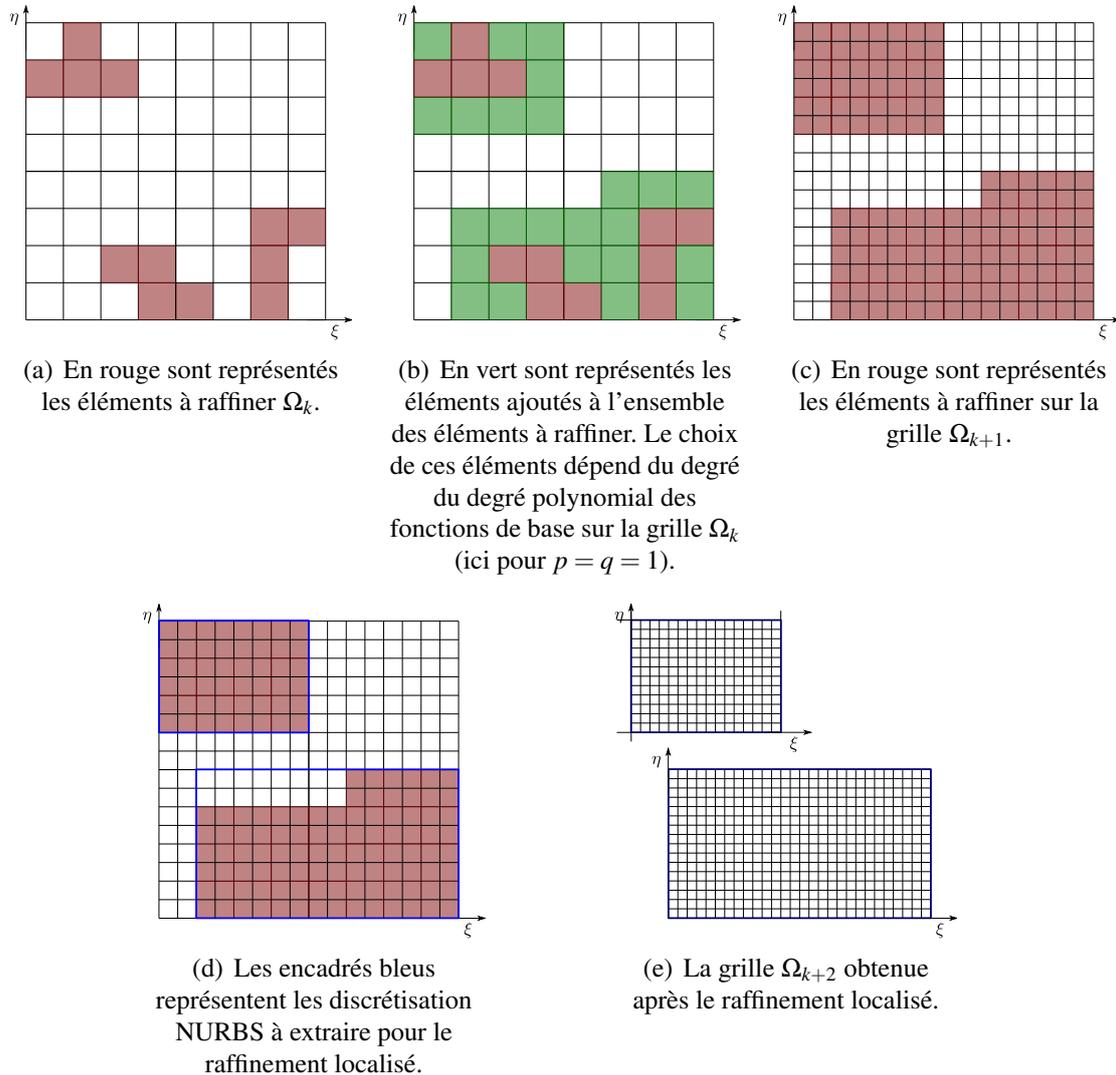


FIGURE 2.11: Procédure de raffinement pour un maillage NURBS avec $p = q = 2$.

2.3 Validation numérique de la méthode

Les résultats présentés ici ont été obtenus dans le cadre de problèmes statiques 2D en élasticité linéaire. Le solveur itératif utilisé comme lisseur dans l'algorithme FMG est un gradient conjugué préconditionné avec un préconditionneur Gauss-Seidel symétrique. L'indicateur d'erreur est un indicateur d'erreur en énergie (section 2.2.2). Les nombres de relaxations v_1 et v_2 (section 1.2) sont fixés à 3.

Nous allons présenter les résultats de simulation sur trois problèmes classiques : la plaque infinie en traction avec un trou elliptique, la membrane de Cook dans le cas compressible et le coin en L. La plaque trouée elliptique à l'avantage d'avoir une solution analytique (Annexe B) et permet donc d'avoir accès à l'erreur entre la solution exacte et la solution calculée. La membrane de Cook et le coin en L ont l'avantage de présenter des

solutions singulières ce qui permettra de tester le comportement de l'indicateur d'erreur en énergie vis à vis de solutions singulières.

2.3.1 Comparaison entre la résolution full-multigrid globale et localisée

Cette partie a pour but de mettre en évidence l'intérêt d'un algorithme full-multigrid localisé par rapport à un algorithme full-multigrid global dans le cas d'une discrétisation NURBS. Le problème que nous allons résoudre ici est le cas d'une plaque infinie en traction avec un trou elliptique. Le problème est résolu en 2D dans le cadre de l'élasticité linéaire en déformations planes. Étant donné les symétries du problème, nous allons restreindre la résolution sur un quart du domaine et imposer les conditions de Neumann provenant de la solution analytique sur le bord rouge du domaine (Figure 2.12).

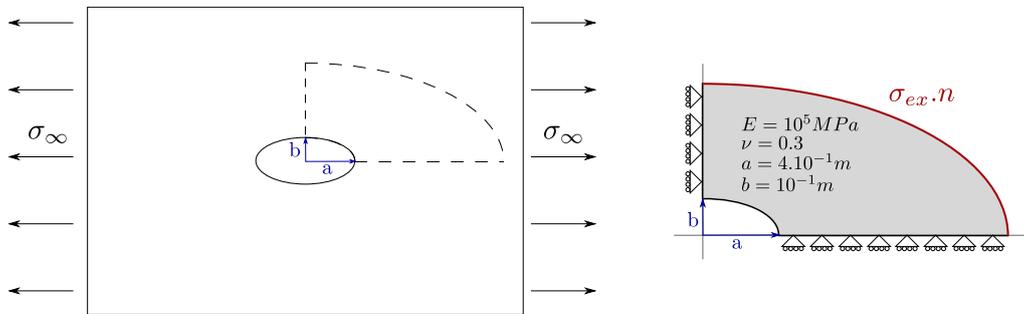


FIGURE 2.12: Données du problème pour la plaque infinie en traction avec un trou elliptique.

Les quantités qui nous intéresseront pour évaluer les performances de la localisation dans l'algorithme multigrilles sont :

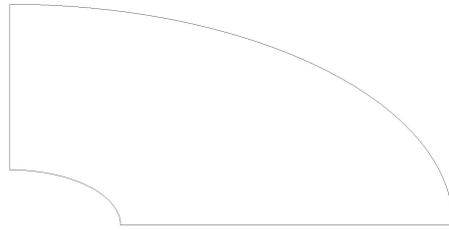
- l'évolution de l'erreur énergétique exacte sur tout le domaine (calculée à partir de la solution analytique) en fonction du nombre total de degrés de liberté. Pour un ensemble de grilles $\{\Omega_1, \Omega_2, \dots, \Omega_k, \Omega_{k+1}\}$ donné, le nombre total de degré de liberté

$$n_{dof} \text{ vaut : } n_{dof} = \sum_{i=1}^{k+1} dof(\Omega_i).$$

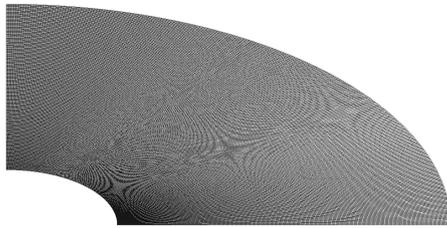
- Le nombre de cycles multigrilles nécessaires à la résolution sur chaque niveau de grille, qui est directement lié au coût de calcul pour chaque résolution.

La géométrie utilisée pour initialiser l'algorithme est la géométrie provenant directement de la CAO (Figure 2.13(a)). Le maillage final obtenu avec localisation est donné Figure 2.13(b) et le maillage final obtenu sans localisation est donné Figure 2.13(c).

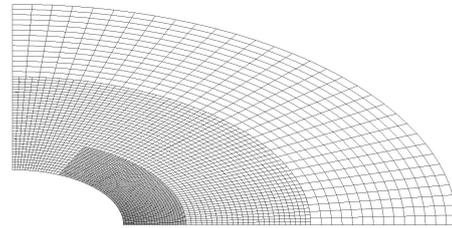
On peut voir sur la figure 2.14(a) que l'algorithme avec localisation fournit une solution plus précise avec moins de degré de libertés que l'algorithme sans localisation. On rappelle que cela est dû au fait que la structure de produit tensoriel des maillages NURBS implique une propagation du raffinement dans toute la structure et donc l'ajout de degrés de liberté qui n'améliore pas beaucoup la qualité de la solution obtenue. On peut aussi



(a) Maillage initial provenant de la CAO.



(b) Maillage final obtenu avec une résolution FMG IGA globale.



(c) Maillage final obtenu avec une résolution FMG IGA localisée.

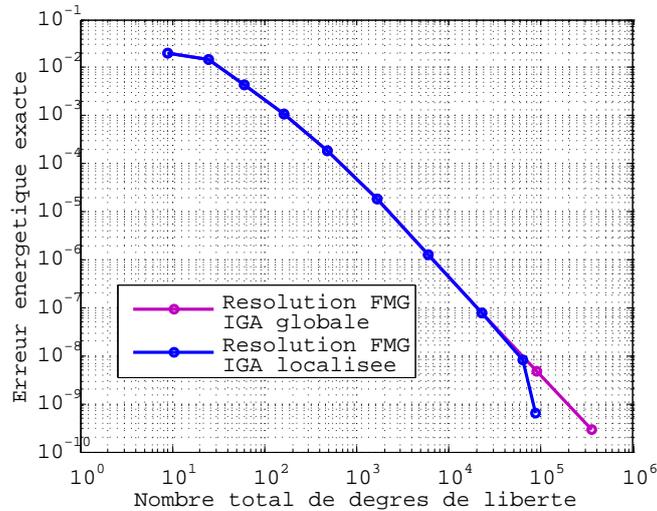
FIGURE 2.13: Maillages initial et finaux pour les résolutions FMG IGA globale et localisée.

remarquer sur la Figure 2.14(b) que le nombre de cycles multigrilles jusqu'à convergence diminue une fois que la localisation a commencé.

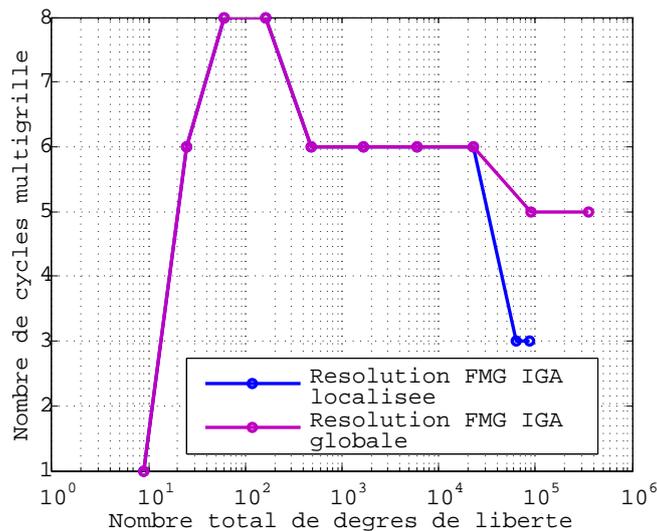
Les résultats obtenus sur cet exemple numérique valident bien l'intérêt de la localisation dans l'algorithme multigrille présenté pour une discrétisation NURBS.

Nous allons maintenant nous intéresser aux performances de l'algorithme FMG IGA localisé pour la recherche de solutions présentant des singularités.

2. Résolution multigrille localisée appliquée à l'analyse isogéométrique en statique linéaire



(a) Comparaison de l'évolution de l'erreur exacte pour les résolutions FMG IGA globale et localisée.



(b) Comparaison de l'évolution du nombre d'itérations multigrille à convergence pour chaque niveau de raffinement, pour les résolutions FMG IGA globale et localisée.

FIGURE 2.14: Comparaison de la convergence et des performances entre les résolutions FMG IGA globale et localisée.

2.3.2 Performances de la résolution full multigrid localisé

Nous allons nous intéresser ici à deux exemples numériques, toujours en 2D dans le cadre de l'élasticité linéaire : la membrane de Cook et le coin en L. Nous pouvons nous attendre à un taux de convergence moins élevé que dans l'exemple de la section précédente et à un plus fort impact de la localisation.

Nous nous intéresserons tout d'abord à la comparaison des performances entre l'al-

gorithme FMG IGA localisé et l'algorithme FMG FEA localisé. Puis nous étudierons le comportement de l'algorithme FMG IGA localisé lorsque le degré des fonctions de base NURBS augmente. L'intérêt de l'utilisation de fonctions de base quadratiques présentant une continuité au passage des frontières d'éléments plus importante que C^0 (comme c'est le cas en éléments finis de degré 2) sera mis en évidence ici. Les quantités utilisées pour comparer les performances des deux algorithmes sont :

- la convergence de l'indicateur d'erreur en fonction du nombre total de degrés de liberté,
- le nombre de cycle multigrilles à convergence pour chaque niveau de résolution.

Les données relatives aux problèmes traités sont présentées Figure 2.15.

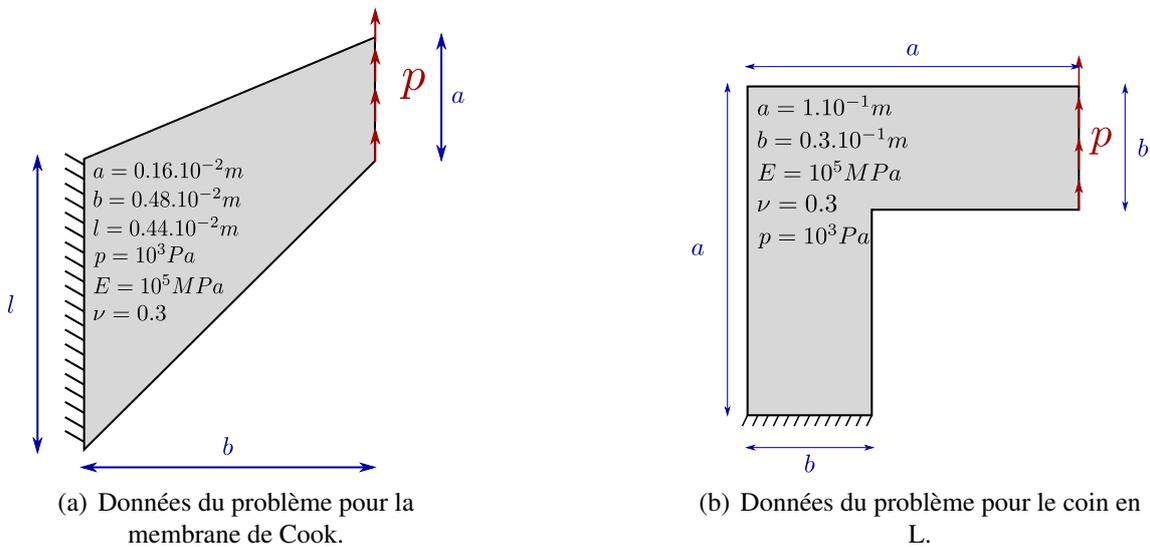


FIGURE 2.15: Paramètres des problèmes étudiés.

Les maillages initiaux et finaux pour les résolutions FMG IGA localisé et FMG FEA localisé pour ces deux problèmes sont donnés Figure 2.16. On remarque que l'utilisation de fonctions de base NURBS C^1 au passage des frontières d'éléments mène à des zones localisées plus étendues que lors de l'utilisation de fonctions de base C^0 . Cela est dû au fait que les fonctions de base C^1 ont comme support trois éléments (voir section 1.1.1) alors que les fonctions C^0 ont comme support un seul élément. Cependant il est important de noter que le fait d'utiliser plus d'éléments avec une discrétisation C^1 qu'avec une discrétisation C^0 ne signifie pas l'utilisation de plus de degrés de liberté. En notant $n_{elem}^{C^1}$, $n_{ddl}^{C^1}$, $n_{elem}^{C^0}$ et $n_{ddl}^{C^0}$ respectivement le nombre d'éléments et le nombre de degrés de liberté d'une discrétisation C^1 et d'une discrétisation C^0 et en supposant un raffinement homogène en deux dimensions, on a :

$$\begin{cases} n_{ddl}^{C^1} \sim n_{elem}^{C^1} & , \text{lorsque } n_{elem}^{C^1} \rightarrow \infty \\ n_{ddl}^{C^0} \sim 4 \times n_{elem}^{C^0} & , \text{lorsque } n_{elem}^{C^0} \rightarrow \infty \end{cases} \quad (2.12)$$

2. Résolution multigrille localisée appliquée à l'analyse isogéométrique en statique linéaire

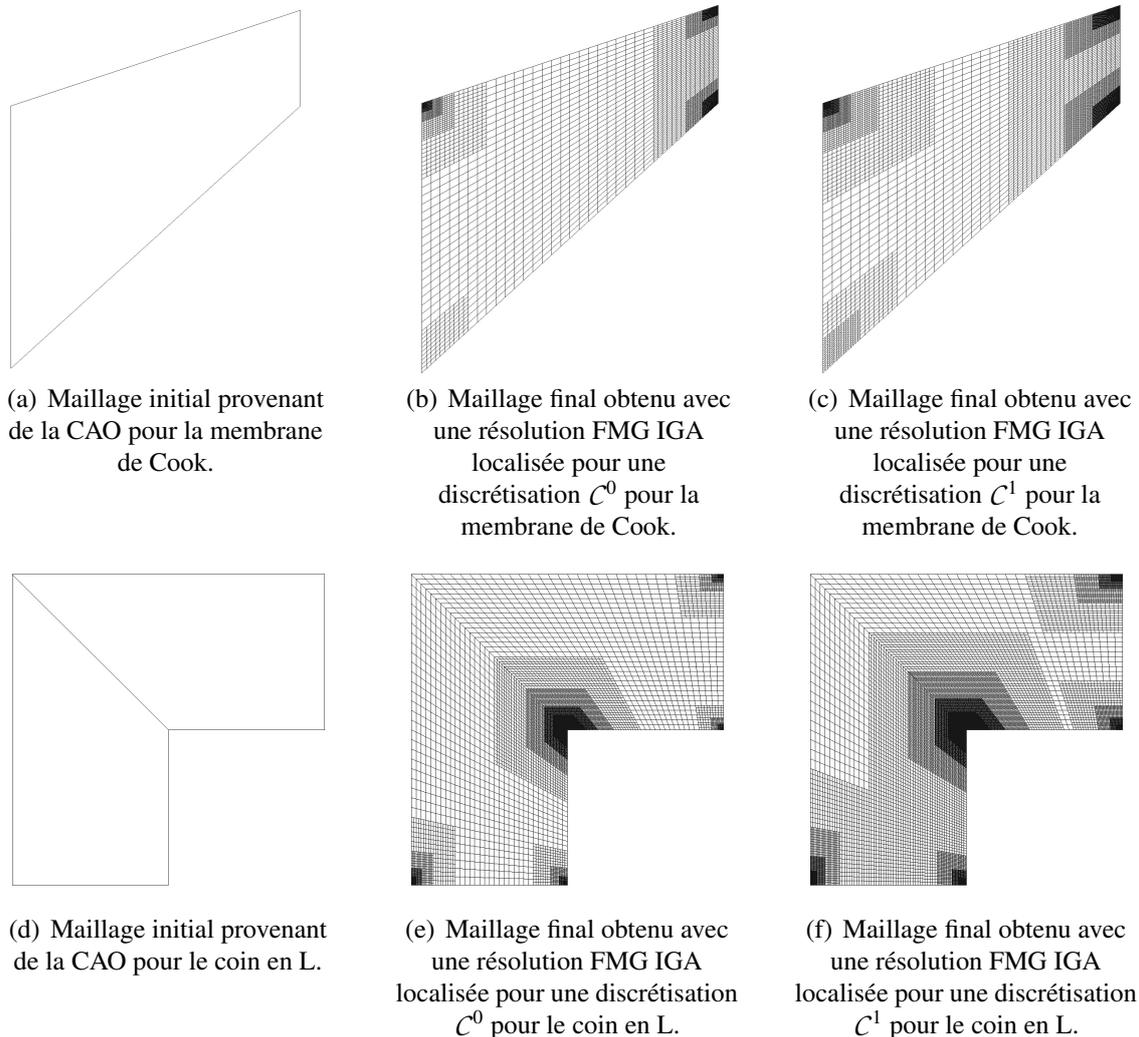


FIGURE 2.16: Maillages initiaux et finaux obtenus avec une résolution FMG IGA localisée.

La figure 2.17(a) présente l'évolution de l'indicateur d'erreur en fonction du nombre total de degrés de liberté dans les cas d'une résolution FMG localisé appliquée aux éléments finis et d'une résolution FMG localisé appliquée à l'analyse isogéométrique pour le problème de la membrane de Cook. On distingue deux comportements différents de l'évolution de l'indicateur d'erreur avant et après la localisation. Le dernier calcul avant localisation est symbolisé sur les tracés par un marqueur carré. Avant localisation, la résolution avec une discrétisation C^0 (type éléments finis) nécessite en moyenne trois fois plus de degrés de liberté que la discrétisation C^1 pour obtenir un niveau d'erreur équivalent. Après localisation, la discrétisation C^0 nécessite quatre fois plus de degrés de liberté que la discrétisation C^1 . De plus la résolution FMG localisé appliquée à l'IGA possède un jeu de quinze grilles à convergence alors que la résolution FMG localisé appliquée aux

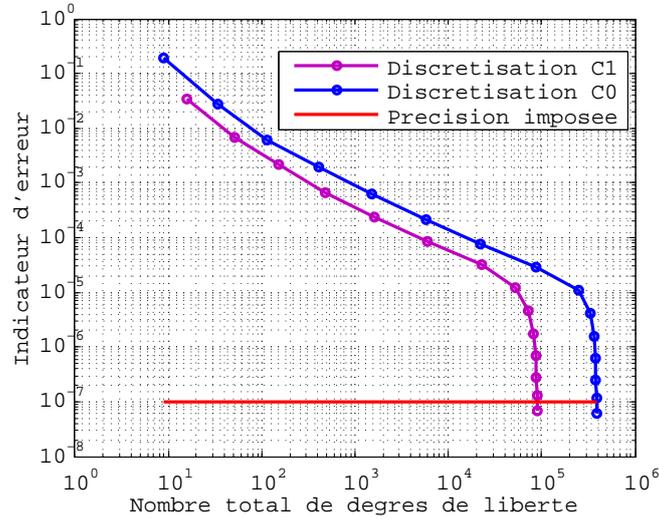
éléments finis en possède seize. Le stockage de la solution est donc plus coûteux pour une discrétisation C^0 que pour une discrétisation C^1 . Pour ce qui est de la différence de coûts de calcul, elle est assez difficile à quantifier. En effet, la discrétisation C^1 possède moins de degrés de liberté à convergence que la discrétisation C^0 mais un cycle multigrilles est plus coûteux en raison de l'utilisation de matrices de rigidité plus denses et le nombre de cycles multigrilles n'est pas connu à l'avance et est très dépendant du problème traité. Pour les exemples traités ici les temps de calculs sont approximativement identiques.

La figure 2.17(b) présente le nombre de cycles multigrilles à convergence pour chaque niveau de raffinement pour les deux types de discrétisation pour la résolution du problème de la membrane de Cook. On distingue encore les zones avant et après localisation sur ces deux courbes. Le dernier calcul avant localisation est symbolisé par un marqueur carré sur les deux tracés. La début de la localisation mène dans les deux cas à une diminution du nombre de cycles multigrilles. Ce comportement à déjà été mis en évidence (voir section 2.3.1 lors de la comparaison d'une résolution FMG globale et locale) et persiste pour la détermination de solution plus singulières. On remarque tout de même la présence d'un pic dans le nombre de cycles multigrilles après la localisation pour la discrétisation C^1 qui n'est pas présent pour la discrétisation C^0 . La résolution du problème du coin en L nous montrera que ce pic n'est pas un comportement spécifique à l'utilisation de l'IGA et doit dépendre du problème résolu.

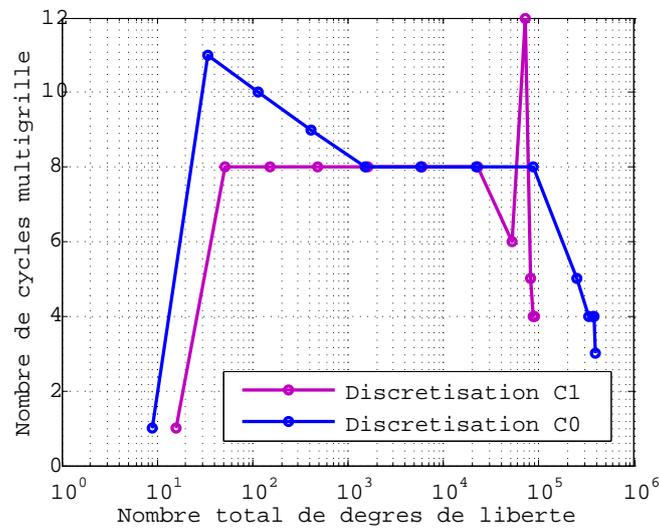
la figure 2.18(a) présente la convergence de l'indicateur d'erreur pour le problème du coin en L pour les deux types de discrétisation C^0 et C^1 . Les conventions de représentation sont les mêmes que pour le cas de la membrane de Cook figure 2.17(a). Avant la localisation, la résolution avec une discrétisation C^0 nécessite en moyenne 3 fois plus de degrés de libertés qu'avec une discrétisation C^1 et 2,5 fois plus après la localisation. De plus la résolution avec la discrétisation C^0 mène à la création d'une grille supplémentaire qu'avec la discrétisation C^1 . La figure 2.18(b) présente le nombre de cycles multigrilles à convergence pour les deux types de discrétisations à chaque niveau de raffinement. Le nombre de cycle multigrille effectué après la localisation est globalement moins important après la localisation pour les deux types de discrétisation. On note la présence d'un pic après la localisation pour les deux discrétisations. Cela nous montre que le pic relevé dans le cas de la membrane de Cook pour la discrétisation C^1 et qui était absent pour la discrétisation C^0 n'est pas spécifique à la continuité des fonctions de base au passage des frontières des éléments mais est problème dépendant. Ce n'est donc pas un comportement systématique lié à l'utilisation de l'analyse isogéométrique. Comme pour la résolution du problème de la membrane de Cook, les temps de calculs pour la résolution du problème du coin en L avec les deux discrétisations sont équivalents.

En conclusion, l'utilisation d'une discrétisation C^1 fournit une solution avec une meilleure continuité et moins de degrés de libertés (donc plus facile à stocker) pour un coût de calcul équivalent à une résolution de type éléments finis.

2. Résolution multigrille localisée appliquée à l'analyse isogéométrique en statique linéaire

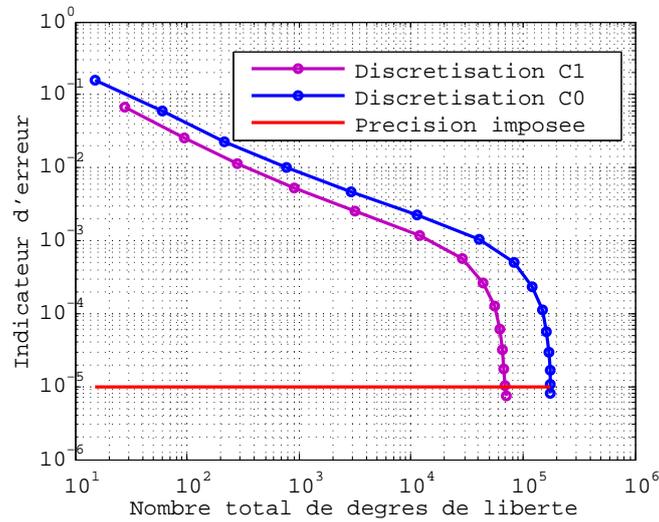


(a) Évolution de l'indicateur d'erreur en énergie en fonctions du nombre total de degrés de liberté pour une discrétisation C^0 et une discrétisation C^1 .

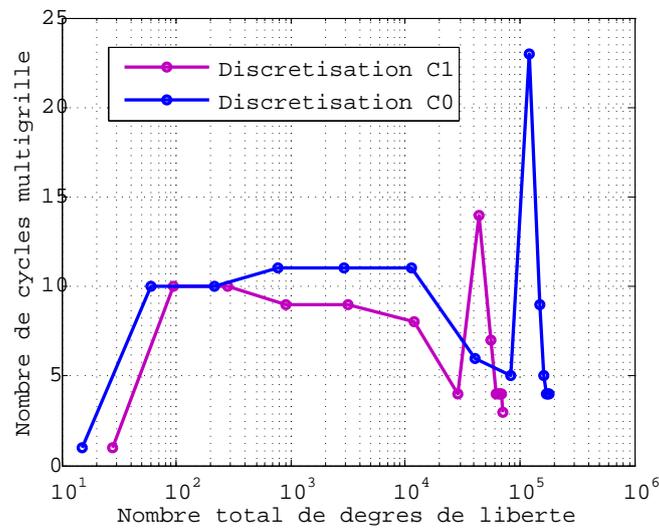


(b) Nombre d'itérations multigrille à convergence pour chaque niveau de raffinement dans le cas d'une discrétisation C^0 et d'une discrétisation C^1 .

FIGURE 2.17: Convergence et performances de la résolution FMG IGA localisée pour le problèmes de la membrane de Cook.



(a) Évolution de l'indicateur d'erreur en énergie en fonctions du nombre total de degrés de liberté pour une discrétisation C^0 et une discrétisation C^1 .

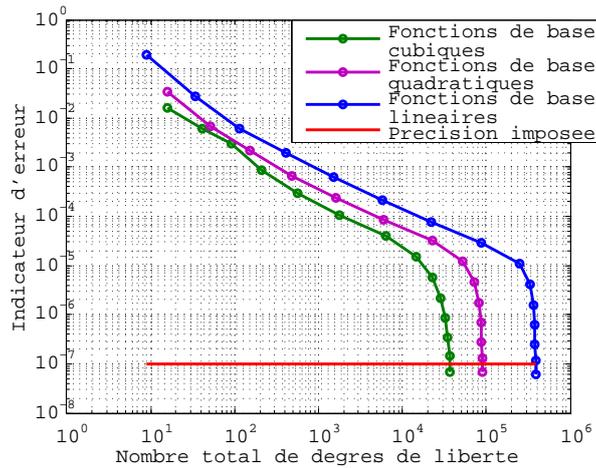


(b) Nombre d'itérations multigrille à convergence pour chaque niveau de raffinement dans le cas d'une discrétisation C^0 et d'une discrétisation C^1 .

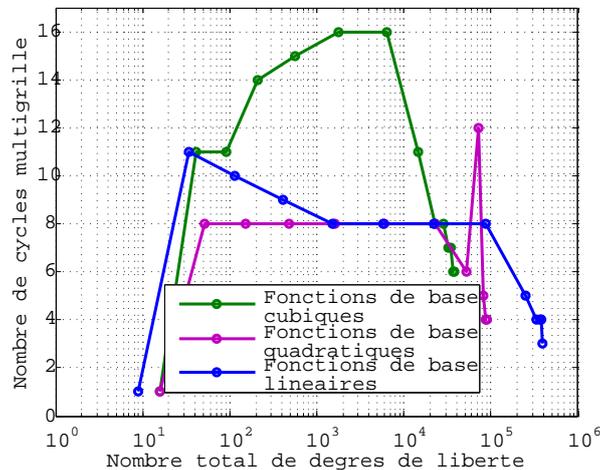
FIGURE 2.18: Convergence et performances de la résolution FMG IGA localisée pour le problèmes du coin en L.

2. Résolution multigrille localisée appliquée à l'analyse isogéométrique en statique linéaire

Nous allons maintenant nous intéresser à l'impact de l'élévation d'ordre des fonctions de base NURBS utilisées pour la résolution sur l'algorithme FMG IGA localisé. Nous allons résoudre le problème de la membrane de Cook avec des fonctions de base NURBS linéaires, quadratiques et cubiques respectivement C^0 , C^1 et C^2 au passage des frontières d'éléments. La figure 2.19 montre, comme on pouvait s'y attendre, que le nombre de degrés de liberté nécessaires pour atteindre le critère de précision imposé par l'utilisateur décroît lorsque le degré des fonctions de base NURBS augmente.



(a) Évolution de l'indicateur d'erreur en énergie en fonction du nombre total de degrés de liberté pour des fonctions de bases linéaires, quadratiques et cubiques.



(b) Nombre d'itérations multigrille à convergence pour chaque niveau de raffinement pour des fonctions de bases linéaires, quadratiques et cubiques.

FIGURE 2.19: Convergence and performances of the FMG IGA algorithm on the Cook's Membrane problem.

2.4 Synthèse

L'algorithme FMG IGA localisé présenté ici permet d'effectuer un raffinement h localisé en évitant les problèmes de propagation de raffinement dans les structures NURBS, tout en conservant la continuité au passage des frontières internes entre les différentes grilles. Cet algorithme offre donc une alternative aux techniques de raffinement en analyse isogéométrique telles que les B-splines hiérarchiques ([VUO 11, KRA 94]) ou les T-splines ([BAZ 10, DÖR 10]), qui nécessitent un effort d'implémentation assez important.

De plus, l'utilisation d'une discrétisation NURBS du domaine étudié avec un algorithme FMG à raffinement automatique permet de contourner la difficulté du raffinement de frontières courbes fréquent en éléments finis, et qui peut être assez complexe à résoudre. Dans notre cas, toutes les informations liées à la géométrie sont déjà contenues dans l'espace de recherche des solutions. Il n'est donc pas nécessaire de conserver des informations supplémentaires sur la géométrie du domaine étudié.

Enfin, la structure en produit tensoriel des fonctions de base NURBS utilisées ici permet de décomposer les opérations de prolongement et de restriction suivant chaque direction paramétrique. En pratique, cela se traduit par le fait que, contrairement lors d'une résolution multigrille appliquée aux éléments finis pour des maillages non structurés, il n'est pas nécessaire de stocker complètement ces opérateurs. Seuls les opérateurs de prolongement suivant chacune des directions paramétriques sont stockés, et l'opération de prolongement est effectuée par leur composition. Cela représente un gain de mémoire non négligeable pour des discrétisations présentant de nombreux degrés de liberté.

2. Résolution multigrille localisée appliquée à l'analyse isogéométrique en statique linéaire

Chapitre 3

Résolution multigrille localisée en espace temps pour la dynamique transitoire

Sommaire

3.1	Objectif	58
3.2	Stratégie de résolution full-multigrid espace temps	58
3.2.1	Problème de référence	58
3.2.2	Choix de l'espace d'approximation	59
3.2.3	Étude des performances	61
3.3	Résolution full-multigrid espace temps appliquée au Time-Discontinuous Galerkin	65
3.3.1	Schéma d'intégration Time-Discontinuous Galerkin	65
3.3.2	Algorithme de résolution full-multigrid localisé avec TDG	69
3.3.3	Résultats	77
3.3.4	Algorithme de résolution full-multigrid localisé par tranches de temps	82
3.4	Synthèse	88

3.1 Objectif

En section 1.3 sont exposées quelques méthodes de raffinement adaptatifs pour des problèmes de dynamique transitoire appliquées à l'élasticité linéaire en petites déformations.

L'application de l'algorithme de résolution Full-Multigrid avec raffinement adaptatif présenté précédemment à un domaine espace temps permet de proposer une stratégie alternative pour la résolution de ce type de problèmes avec un contrôle sur la précision de la solution obtenue. Dans cette partie seront présentés l'étude de faisabilité de cette technique de résolution, les adaptations de l'algorithme de résolution présenté en partie 2 qui en découlent et l'analyse des performances de cette stratégie de résolution.

3.2 Stratégie de résolution full-multigrid espace temps

3.2.1 Problème de référence

Nous nous intéressons à la résolution de problèmes de dynamique transitoire en élasticité linéaire représenté figure 3.1.

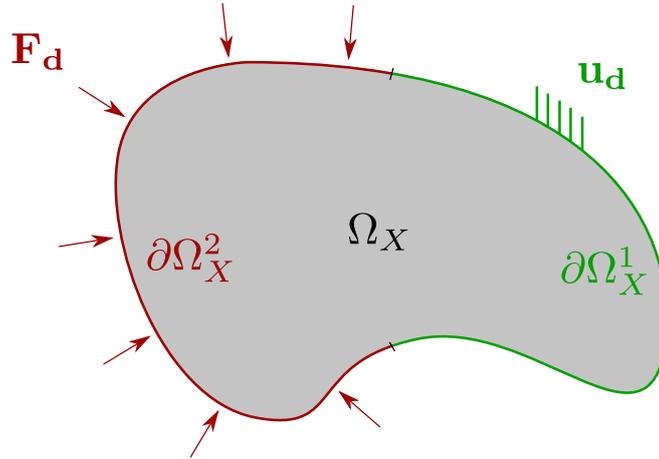


FIGURE 3.1: Problème de référence.

Nous souhaitons donc trouver le champ de déplacement \mathbf{u} vérifiant sur un intervalle de temps $\Omega_T = [t_0, t_N]$ et sur un domaine Ω_X le système d'équation suivant :

$$\left\{ \begin{array}{l} \operatorname{div} \boldsymbol{\sigma}(\mathbf{u}(\mathbf{x}, t)) = \rho \ddot{\mathbf{u}}(\mathbf{x}, t), \quad \forall (\mathbf{x}, t) \in \Omega_X \times \Omega_T, \\ \mathbf{u}(\mathbf{x}, t_0) = \mathbf{u}_0(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega_X, \\ \dot{\mathbf{u}}(\mathbf{x}, t_0) = \mathbf{v}_0(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega_X, \\ \mathbf{u}(\mathbf{x}, t) = \mathbf{u}_d(\mathbf{x}, t), \quad \forall (\mathbf{x}, t) \in \partial\Omega_X^1 \times \Omega_T, \\ \boldsymbol{\sigma}(\mathbf{u}(\mathbf{x}, t)) \cdot \mathbf{n}(\mathbf{x}, t) = \mathbf{F}_d(\mathbf{x}, t), \quad \forall (\mathbf{x}, t) \in \partial\Omega_X^2 \times \Omega_T. \end{array} \right. \quad (3.1)$$

avec $\boldsymbol{\sigma} = \mathbf{C}\boldsymbol{\varepsilon}$ et \mathbf{C} le tenseur de Hooke. Ce système d'équation peut être réécrit en faisant intervenir deux champs inconnus, le déplacement \mathbf{u} et la vitesse \mathbf{v} , de la manière suivante :

$$\left\{ \begin{array}{l} \operatorname{div} \boldsymbol{\sigma}(\mathbf{u}(\mathbf{x}, t)) = \rho \dot{\mathbf{v}}(\mathbf{x}, t), \quad \forall (\mathbf{x}, t) \in \Omega_X \times \Omega_T, \\ \mathbf{v}(\mathbf{x}, t) = \dot{\mathbf{u}}(\mathbf{x}, t), \quad \forall (\mathbf{x}, t) \in \Omega_X \times \Omega_T, \\ \mathbf{v}(\mathbf{x}, t_0) = \mathbf{v}_0(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega_X, \\ \mathbf{u}(\mathbf{x}, t_0) = \mathbf{u}_0(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega_X, \\ \mathbf{u}(\mathbf{x}, t) = \mathbf{u}_d(\mathbf{x}, t), \quad \forall (\mathbf{x}, t) \in \partial\Omega_X^1 \times \Omega_T, \\ \mathbf{v}(\mathbf{x}, t) = \mathbf{v}_d(\mathbf{x}, t), \quad \forall (\mathbf{x}, t) \in \partial\Omega_X^1 \times \Omega_T, \\ \boldsymbol{\sigma}(\mathbf{u}(\mathbf{x}, t)) \cdot \mathbf{n}(\mathbf{x}, t) = \mathbf{F}_d(\mathbf{x}, t), \quad \forall (\mathbf{x}, t) \in \partial\Omega_X^2 \times \Omega_T. \end{array} \right. \quad (3.2)$$

L'intérêt d'une telle réécriture du problème est de faire apparaître le champ de vitesse. En effet, il est montré dans [GRA 00, CAV 05] que les conditions aux limites sur les bords des grilles localisées en espace temps doivent être appliquées en vitesse afin de conserver l'énergie totale du système au cours du temps. De plus, la formulation faible de 3.2 permet de coupler de manière faible les champs \mathbf{u} et \mathbf{v} et ainsi de permettre une plus grande liberté sur les espaces de recherche de ces champs inconnus. En effet, la forme 3.1 fait intervenir la dérivée seconde en temps de \mathbf{u} et impose donc que l'espace de recherche de \mathbf{u} soit au moins quadratique par morceaux, alors que la forme 3.2 permet de chercher \mathbf{u} et \mathbf{v} dans des espaces linéaires par morceaux par exemple. Cette souplesse sur les degrés des fonctions de base des espaces de recherche est importante pour l'étude de la faisabilité d'une résolution Full-Multigrid espace temps car cela permet d'évacuer le paramètre de l'ordre des fonctions de bases. De plus, la forme 3.2 est la base de construction de formulations faibles en temps pour la construction d'intégrateurs temporels modernes [GRA 15].

3.2.2 Choix de l'espace d'approximation

Afin de résoudre le problème 3.2 avec l'algorithme Full-Multigrid à raffinement adaptatif nous allons raisonner sur un domaine espace temps. On définit $\Omega = \Omega_X \times \Omega_T$ le domaine espace temps. On pose :

$$\left\{ \begin{array}{l} \mathcal{U} = \{\mathbf{u} \in H_1(\Omega)\} / \mathbf{u}|_{\partial\Omega_X^1 \times \Omega_T} = \mathbf{u}_d \text{ et } \mathbf{u}|_{\Omega_X \times t_0} = \mathbf{u}_0, \\ \mathcal{U}^* = \{\mathbf{u}^* \in H_1(\Omega)\} / \mathbf{u}^*|_{\partial\Omega_X^1 \times \Omega_T} = \mathbf{0} \text{ et } \mathbf{u}^*|_{\Omega_X \times t_0} = \mathbf{0}, \\ \mathcal{V} = \{\mathbf{v} \in H_1(\Omega)\} / \mathbf{v}|_{\partial\Omega_X^1 \times \Omega_T} = \mathbf{v}_d \text{ et } \mathbf{v}|_{\Omega_X \times t_0} = \mathbf{v}_0, \\ \mathcal{V}^* = \{\mathbf{v}^* \in H_1(\Omega)\} / \mathbf{v}^*|_{\partial\Omega_X^1 \times \Omega_T} = \mathbf{0} \text{ et } \mathbf{v}^*|_{\Omega_X \times t_0} = \mathbf{0}. \end{array} \right. \quad (3.3)$$

La formulation faible du problème 3.2 donne : trouver $(\mathbf{u}, \mathbf{v}) \in \mathcal{U} \times \mathcal{V}$ tels que $\forall (\mathbf{u}^*, \mathbf{v}^*) \in \mathcal{U}^* \times \mathcal{V}^*$,

$$\begin{aligned} & \int_{\Omega} \mathbf{v}^*(\mathbf{x}, t) \rho \dot{\mathbf{v}}(\mathbf{x}, t) + \varepsilon(\mathbf{v}^*(\mathbf{x}, t)) \boldsymbol{\sigma}(\mathbf{u}(\mathbf{x}, t)) \, d\Omega \\ & + \int_{\Omega} \varepsilon(\mathbf{u}^*(\mathbf{x}, t)) \boldsymbol{\sigma}(\dot{\mathbf{u}}(\mathbf{x}, t) - \mathbf{v}(\mathbf{x}, t)) \, d\Omega = \int_{\Omega_T} \int_{\partial\Omega_X(t)} \mathbf{v}^*(\mathbf{x}, t) \mathbf{F}_d(\mathbf{x}, t) \, d\partial\Omega_X \, d\Omega_T. \end{aligned} \quad (3.4)$$

Une discrétisation de type Galerkin du problème faible 3.4 mène à la définition d'une grille espace temps discrétisée Ω_{XT} vue comme un ensemble d'éléments espace temps.

3. Résolution multigrille localisée en espace temps pour la dynamique transitoire

Un élément espace temps e^i est défini comme le produit tensoriel d'un élément spatial et d'un élément temporel : $e^i = e_T^l \otimes e_X^j$ (Figure 3.2).

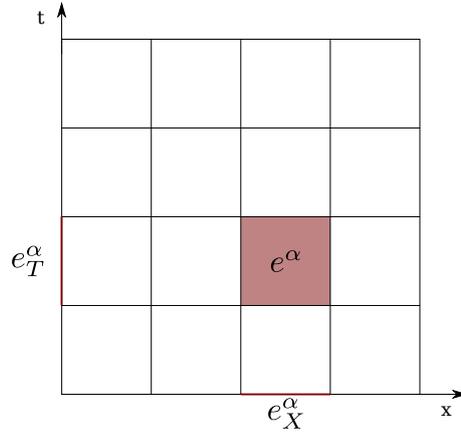


FIGURE 3.2: Définition d'une grille espace temps.

On a alors :

$$\left\{ \begin{array}{l} \Omega_X = \bigcup_{j=1}^{N_X} e_X^j, \\ \Omega_T = \bigcup_{l=1}^{N_T} e_T^l, \\ \Omega_{XT} = \Omega_T \otimes \Omega_X = \bigcup_{i=1}^N e^i. \end{array} \right. \quad (3.5)$$

Nous chercherons les champs de déplacement \mathbf{u} et de vitesse \mathbf{v} sous la forme :

$$\left\{ \begin{array}{l} \mathbf{u}(\mathbf{x}, t) = \sum_{i=1}^{N_X} \sum_{j=1}^{N_T} \phi_i^u(x) \psi_j^u(t) \mathbf{u}_{ij} \quad \forall (\mathbf{x}, t) \in \Omega, \\ \mathbf{v}(\mathbf{x}, t) = \sum_{i=1}^{N_X} \sum_{j=1}^{N_T} \phi_i^v(x) \psi_j^v(t) \mathbf{v}_{ij}, \quad \forall (\mathbf{x}, t) \in \Omega. \end{array} \right. \quad (3.6)$$

A priori on choisira $\phi_i^u = \phi_i^v = \phi_i$ et $\psi_j^u = \psi_j^v = \psi_j$.

Ce choix d'espace de recherche pour les champs \mathbf{u} et \mathbf{v} mène à la résolution du système matriciel :

$$\mathbb{A}\mathbf{X} = \mathbf{B}, \quad (3.7)$$

avec

$$\left\{ \begin{array}{l} \mathbb{A} = \begin{pmatrix} \mathbb{N}_1 \otimes \mathbb{K} & -\mathbb{N}_2 \otimes \mathbb{K} \\ \mathbb{N}_2 \otimes \mathbb{K} & \mathbb{N}_1 \otimes \mathbb{M} \end{pmatrix}, \\ (N_1)_{jl} = \int_{\Omega_T} \psi_j(t) \psi_l(t) d\Omega_T, \\ (N_2)_{jl} = \int_{\Omega_T} \psi_j(t) \psi_l(t) d\Omega_T, \\ (K)_{ik} = \int_{\Omega_X} \varepsilon(\phi_i(x)) \sigma(\phi_k(x)) d\Omega_X, \\ (M)_{ik} = \int_{\Omega_X} \phi_i(x) \rho \phi_k(x) d\Omega_X. \end{array} \right. \quad (3.8)$$

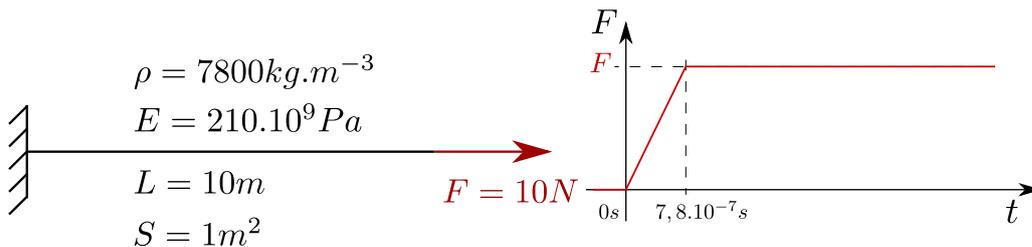
Une résolution multigrille est basée sur les propriétés de lissage d'un solveur itératif (voir section 1.2.1). Pour la résolution de problèmes en mécanique du solide à l'aide d'une stratégie multigrille, le lisseur le plus utilisé est le gradient conjugué. Pour le problème de dynamique transitoire discrétisé que nous souhaitons résoudre, l'opérateur espace temps \mathbb{A} n'est ni symétrique, ni défini positif. Le gradient conjugué ne peut donc pas être utilisé. La forme de l'opérateur \mathbb{A} nous oriente donc vers l'utilisation d'un lisseur de type de minimisation du résidu généralisé (Generalized Minimal RESidual –GMRES–).

La section suivante présente l'étude des performances d'une résolution multigrille pour le problème associé à l'équation 3.7 en fonction de la structure de l'opérateur espace temps \mathbb{A} . Cette étude a pour but de permettre de faire un choix sur l'intégrateur temporel à utiliser pour une résolution FMG localisée en dynamique transitoire.

Remarque : Les travaux effectués dans la section suivante sont réalisés avec un opérateur espace temps \mathbb{A} assemblé sur tout le domaine $\Omega_X \times \Omega_T$. Le stockage de cet opérateur nécessite énormément d'espace mémoire et n'est pas réalisable en pratique. Une fois l'étude de faisabilité terminée et le choix de l'intégrateur temporel effectué, un opérateur espace temps assemblé sur l'ensemble du domaine ne sera jamais utilisé.

3.2.3 Étude des performances

Afin d'étudier l'intérêt d'une résolution multigrille pour résoudre les problèmes matriciels (équation 3.7), nous allons nous intéresser à une résolution sur deux grilles globales pour un problème de traction en une dimension sur l'intervalle de temps $[0s; 7,8 \cdot 10^{-6}s]$ à conditions initiales nulles.



3. Résolution multigrille localisée en espace temps pour la dynamique transitoire

Les fonctions de base de l'espace de recherche des champs \mathbf{u} et \mathbf{v} (équation 3.6) seront des fonctions éléments finis linéaires. Ces fonctions correspondant à une discrétisation NURBS de degré 1, les opérateurs de prolongement \mathcal{P} et de restriction \mathcal{R} seront construits de la même manière que pour les problèmes de statique élastique linéaire (voir section 2.2.1). Pour l'opération de lissage, 5 itérations de GMRES seront effectuées sur l'opérateur espace temps sur grille fine.

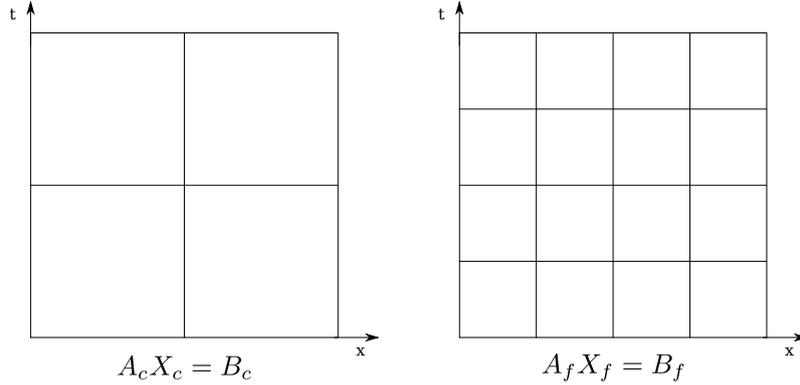


FIGURE 3.3: Grilles espace temps grossière et fine et équations matricielles associées. La grille grossière est composée de 4 éléments en espace et 10 éléments en temps, la grille fine de 8 éléments en espace et 20 éléments en temps. Soit respectivement des problèmes à 80 et 320 degrés de liberté après application des conditions aux limites.

Algorithm 2: Résolution sur deux grilles en espace temps.

```

 $k \leftarrow 0$  ;  $\mathbf{X}_f^0 \leftarrow \mathbf{0}$  ;
tant que  $res > \varepsilon$  faire
     $\bar{\mathbf{X}}_f^k \leftarrow GMRES(\mathbb{A}_f, \mathbf{X}_f^k, \mathbf{B}_f, 5)$  ;
     $\mathbf{R}_f^k \leftarrow \mathbf{B}_f - \mathbb{A}_f \bar{\mathbf{X}}_f^k$  ;
     $\mathbf{E}_c^k \leftarrow \text{résolution\_directe}(\mathbb{A}_c \mathbf{E}_c^k = \mathcal{R}(\mathbf{R}_f^k))$  ;
     $\hat{\mathbf{X}}_f^k \leftarrow \bar{\mathbf{X}}_f^k + \mathcal{P}(\mathbf{E}_c^k)$  ;
     $\mathbf{X}_f^{k+1} \leftarrow GMRES(\mathbb{A}_f, \hat{\mathbf{X}}_f^k, \mathbf{B}_f, 5)$  ;
     $res \leftarrow \text{critère}(\mathbb{A}_f, \mathbf{X}_f^{k+1}, \mathbf{B}_f)$  ;
     $k \leftarrow k + 1$  ;
 $\mathbf{X}_f \leftarrow \mathbf{X}_f^k$  ;

```

Afin d'évaluer les performances de la résolution multigrille, nous allons comparer l'évolution de la norme du résidu de l'équation 3.7 avec et sans prise en compte de la correction provenant de la grille grossière (résolution décrite par l'algorithme 2 avec $\mathcal{P} = 0$). Si une résolution multigrille est efficace pour ce problème, la prise en compte de la correction devrait faire diminuer fortement le nombre d'itérations à convergence comme dans le cas d'un problème de statique linéaire (voir section 1.2.1, figure 1.19).

La figure 3.4 montre que la résolution itérative ne converge pas que l'on prenne en

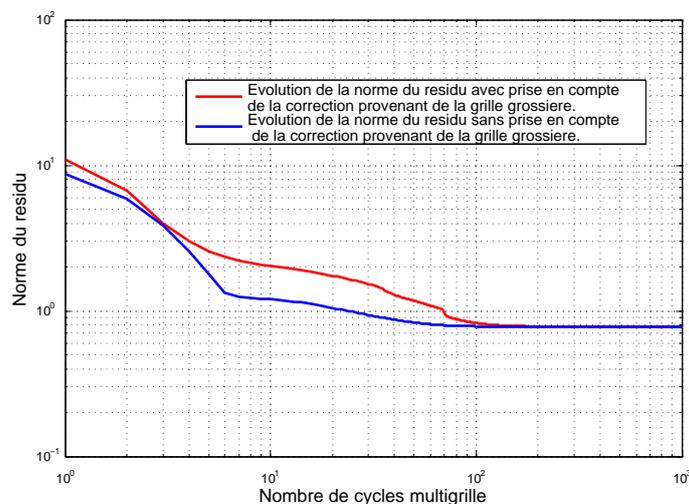


FIGURE 3.4: Performances de la résolution sur deux grilles.

compte ou non la correction provenant de la grille grossière. Cela est dû au fait que le lisseur utilisé (GMRES) n'est pas préconditionné. Le choix d'un préconditionneur pour une résolution multigrille espace temps pour la dynamique transitoire adapté à tout type de problèmes n'est pas traité ici. Dans la suite de ce chapitre, seuls des problèmes en une dimension spatiale sont traités. Le préconditionnement de ces problèmes est effectué par un adimensionnement des équations à résoudre. La figure 3.5 représente l'évolution de la norme du résidu après adimensionnement du problème.

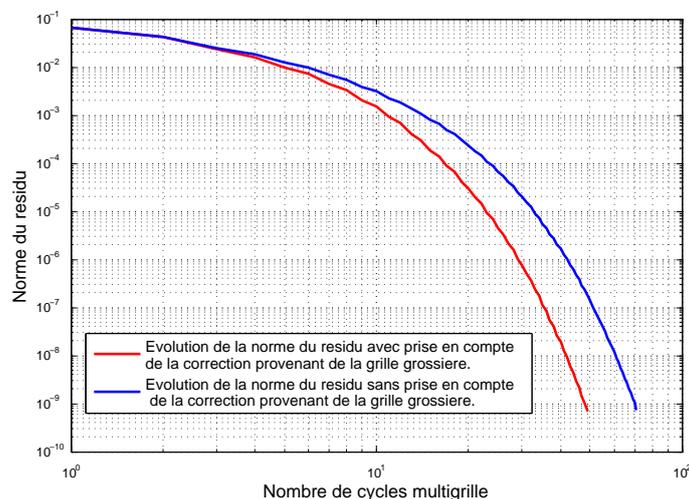


FIGURE 3.5: Performances de la résolution sur deux grilles après adimensionnement du problème.

On constate que la résolution sur deux grilles du problème 3.8 n'est pas efficace. Tout d'abord, la prise en compte de la correction provenant de la grille grossière ne diminue pas assez fortement le nombre de cycle multigrille à convergence : 49 cycles multigrille

3. Résolution multigrille localisée en espace temps pour la dynamique transitoire

à convergence lorsque l'on prend en compte la correction provenant de la grille grossière contre 70 lorsque l'on ne prend pas en compte cette correction. De plus, on constate que la résolution sans prise en compte de la correction sur grille grossière, correspondant donc à une résolution itérative avec le lisseur réinitialisé toutes les 5 itérations nécessite $70 \times 2 \times 5 = 700$ itérations de GMRES au total à convergence pour un problème à 320 degrés de liberté sur la grille fine.

Nous allons donc mettre en évidence les caractéristiques de la structure de l'opérateur A diminuant les performances du lisseur afin de faire un choix de discrétisation et d'intégrateur temporel adapté. On rappelle la structure de l'opérateur espace temps :

$$\mathbb{A} = \begin{pmatrix} \mathbb{N}_1 \otimes \mathbb{K} & -\mathbb{N}_2 \otimes \mathbb{K} \\ \mathbb{N}_2 \otimes \mathbb{K} & \mathbb{N}_1 \otimes \mathbb{M} \end{pmatrix}. \quad (3.9)$$

Les trois caractéristiques essentielles de la structure de l'opérateur \mathbb{A} sont :

- la non symétrie, due à la non symétrie de \mathbb{N}_1 (3.8) et au terme $-\mathbb{N}_2 \otimes \mathbb{K}$,
- le couplage des champs \mathbf{u} et \mathbf{v} ,
- la partie hyperbolique de l'équation différentielle 3.2 représentée par le terme $\mathbb{N}_1 \otimes \mathbb{M}$.

Nous allons donc nous intéresser au nombre d'itérations à convergence pour diverses structures de \mathbb{A} . Ce travail est effectué sur des opérateurs symétriques ou non symétriques, avec les champs \mathbf{u} et \mathbf{v} couplés ou découplés et avec la présence ou non de la composante hyperbolique du problème. Cela nous mène à huit structures de A différentes (voir tableaux 3.1 et 3.2).

Elliptique	Non symétrique	Symétrique
Couplé	$\mathbb{A}^{QS} = \begin{pmatrix} \mathbb{N}_1 \otimes \mathbb{K} & -\mathbb{N}_2 \otimes \mathbb{K} \\ \mathbb{N}_2 \otimes \mathbb{K} & \mathbf{0} \end{pmatrix}$	$\mathbb{A}_s^{QS} = \begin{pmatrix} \mathbb{N}_2 \otimes \mathbb{K} & \mathbb{N}_2 \otimes \mathbb{K} \\ \mathbb{N}_2 \otimes \mathbb{K} & \mathbf{0} \end{pmatrix}$
Découplé	$\mathbb{A}_d^{QS} = \begin{pmatrix} \mathbf{0} & -\mathbb{N}_2 \otimes \mathbb{K} \\ \mathbb{N}_2 \otimes \mathbb{K} & \mathbf{0} \end{pmatrix}$	$\mathbb{A}_{ds}^{QS} = \begin{pmatrix} \mathbf{0} & \mathbb{N}_2 \otimes \mathbb{K} \\ \mathbb{N}_2 \otimes \mathbb{K} & \mathbf{0} \end{pmatrix}$

TABLE 3.1: Structure de l'opérateur espace temps pour le problème sans la composante hyperbolique.

Hyperbolique	Non symétrique	Symétrique
Couplé	$\mathbb{A} = \begin{pmatrix} \mathbb{N}_1 \otimes \mathbb{K} & -\mathbb{N}_2 \otimes \mathbb{K} \\ \mathbb{N}_2 \otimes \mathbb{K} & \mathbb{N}_1 \otimes \mathbb{M} \end{pmatrix}$	$\mathbb{A}_s = \begin{pmatrix} \mathbb{N}_2 \otimes \mathbb{K} & \mathbb{N}_2 \otimes \mathbb{K} \\ \mathbb{N}_2 \otimes \mathbb{K} & \mathbb{N}_2 \otimes \mathbb{M} \end{pmatrix}$
Découplé	$\mathbb{A}_d = \begin{pmatrix} \mathbf{0} & -\mathbb{N}_2 \otimes \mathbb{K} \\ \mathbb{N}_2 \otimes \mathbb{K} & \mathbb{N}_1 \otimes \mathbb{M} \end{pmatrix}$	$\mathbb{A}_{ds} = \begin{pmatrix} \mathbf{0} & \mathbb{N}_2 \otimes \mathbb{K} \\ \mathbb{N}_2 \otimes \mathbb{K} & \mathbb{N}_2 \otimes \mathbb{M} \end{pmatrix}$

TABLE 3.2: Structure de l'opérateur espace temps pour le problème avec la composante hyperbolique.

10^1	10^2	10^3
\mathbb{A}_{ds}^{QS}	\mathbb{A}_d^{QS}	\mathbb{A}^{QS}
\mathbb{A}_{ds}	\mathbb{A}_d	
	\mathbb{A}_s^{QS}	
	\mathbb{A}_s	
	\mathbb{A}	

TABLE 3.3: Ordre de grandeur du nombre d'itérations à convergence.

Les résultats présentés dans ce cas particulier pour les différentes formes de l'opérateur espace temps \mathbb{A} (tableau 3.3) nous montrent que la composante hyperbolique du problème n'influence pas la rapidité de convergence de la résolution multigrille. Les caractéristiques prédominantes dans la perte de performances de la résolution multigrilles sont le couplage des champs \mathbf{u} et \mathbf{v} et la non symétrie de l'opérateur.

On rappelle que l'utilisation d'un intégrateur temporel faisant intervenir le champ de vitesses \mathbf{v} est très intéressant dans notre cas étant donné que les conditions aux limites à imposer sur les grilles localisées devront se faire sur le champ de vitesse (voir section 3.2.1).

L'intégrateur choisi pour la suite sera un intégrateur type Galerkin discontinu en temps (Time Discontinuous Galerkin –TDG–) à deux champs. En effet, cet type d'intégrateur fait intervenir les champs \mathbf{u} et \mathbf{v} , mais le couplage est résolu de façon directe et les opérateurs intervenants dans la résolution sont symétriques définis positifs. Ce choix nous permettra donc de contourner les pertes de performances dues au couplage et à la non symétrie de l'opérateurs \mathbb{A} .

3.3 Résolution full-multigrid espace temps appliquée au Time-Discontinuous Galerkin

3.3.1 Schéma d'intégration Time-Discontinuous Galerkin

On considère l'intervalle de temps $\Omega_T = [t_0, t_N[$ discrétisé en N éléments temporels e_T^l tels que $e_T^l = [t_{l-1}, t_l[$, $l \in [1, N]$. On pose $\Delta t_l = t_l - t_{l-1}$. Le schéma d'intégration TDG est basé sur une discrétisation de type Galerkin des équations fortes à deux champs 3.2, avec pour particularité d'autoriser les discontinuités aux frontières des intervalles de temps e_T^l . Il est possible d'utiliser des fonctions de base de degré i pour le traitement de la dimension temporelle du champ de déplacement \mathbf{u} et de degré j pour le traitement de la dimension temporelle du champ de vitesse \mathbf{v} . Un tel intégrateur est nommé TDG $P_i - P_j$. La méthode TDG $P_1 - P_1$ est la plus utilisée pour les méthodes à raffinement adaptatif (on peut se référer notamment à [AUB 99, LI 98, TIE 03]). Ce sera donc celle utilisée dans la suite de ces travaux.

Les champs de déplacement \mathbf{u} et de vitesse \mathbf{v} sont recherchés sous la forme :

$$\begin{cases} \mathbf{u}(\mathbf{x}, t) = \sum_{i=1}^{\hat{N}_X} \sum_{j=1}^N \left[\tilde{\phi}_i(\mathbf{x}) \psi_j^g(t) \mathbf{u}_{ij}^g + \tilde{\phi}_i(\mathbf{x}) \psi_j^d(t) \mathbf{u}_{ij}^d \right], & \forall (\mathbf{x}, t) \in \Omega_X \otimes \Omega_T, \\ \mathbf{v}(\mathbf{x}, t) = \sum_{i=1}^{\hat{N}_X} \sum_{j=1}^N \left[\hat{\phi}_i(\mathbf{x}) \psi_j^g(t) \mathbf{v}_{ij}^g + \hat{\phi}_i(\mathbf{x}) \psi_j^d(t) \mathbf{v}_{ij}^d \right], & \forall (\mathbf{x}, t) \in \Omega_X \otimes \Omega_T, \end{cases} \quad (3.10)$$

avec

$$\begin{cases} \psi_j^g(t) = \frac{t_j - t}{t_j - t_{j-1}} & \text{si } t \in e_T^j \\ = 0 & \text{sinon,} \\ \psi_j^d(t) = \frac{t - t_{j-1}}{t_j - t_{j-1}} & \text{si } t \in e_T^j \\ = 0 & \text{sinon.} \end{cases} \quad (3.11)$$

et où les $\tilde{\phi}_i$ sont les fonctions de bases en espace utilisées pour construire l'espace de recherche de \mathbf{u} et les $\hat{\phi}_i$ sont les fonctions de bases en espace utilisées pour construire l'espace de recherche de \mathbf{v} . L'égalité $\mathbf{v} = \dot{\mathbf{u}}$ étant vérifiée de façon faible, il n'y a aucune condition à respecter sur le degré des fonctions de bases choisies pour construire ce deux espaces. Dans la suite nous choisirons d'utiliser les mêmes fonctions de base NURBS quadratiques pour \mathbf{u} et \mathbf{v} en espace.

L'expression des champs \mathbf{u} et \mathbf{v} autorise une discontinuité à chaque piquet de temps t_l (voir Figure 3.6). Par souci de lisibilité, on introduit les notations suivantes :

$$\begin{cases} \alpha(\mathbf{x}, t_l^+) = \lim_{\varepsilon \rightarrow 0^+} \alpha(\mathbf{x}, t_l + \varepsilon), \\ \alpha(\mathbf{x}, t_l^-) = \lim_{\varepsilon \rightarrow 0^-} \alpha(\mathbf{x}, t_l + \varepsilon). \end{cases} \quad (3.12)$$

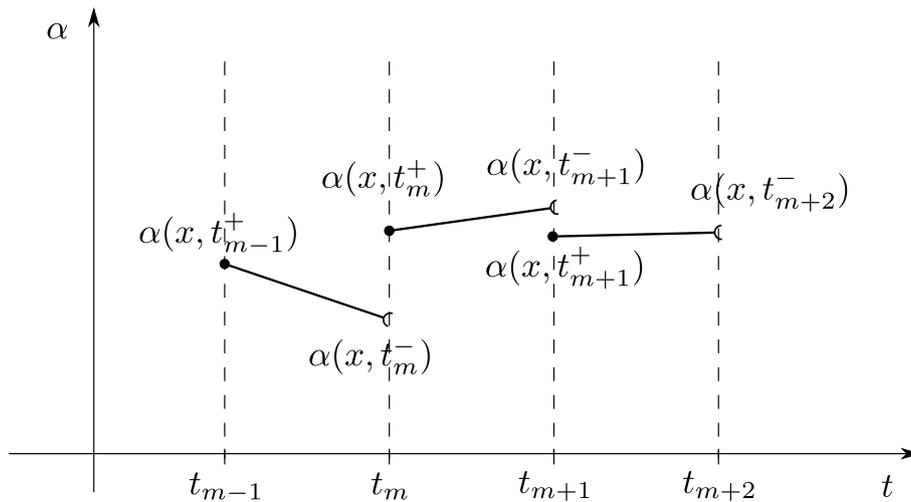


FIGURE 3.6: Évolution d'une quantité α en fonction du temps dans le cadre du TDG $P1 - P1$.

On définit le saut au passage de t_l d'une quantité $\alpha(x, t)$ de la façon suivante :

$$\begin{aligned} [[\alpha(\mathbf{x}, t_l)]] &= \sum_{i=1}^{N_X} \left[\phi_i(\mathbf{x}) \alpha_{il}^d - \phi_i(\mathbf{x}) \alpha_{il+1}^g \right] \\ &= \alpha(\mathbf{x}, t_l^+) - \alpha(\mathbf{x}, t_l^-). \end{aligned} \quad (3.13)$$

La formulation faible de 3.2 sur l'élément temporel e_T^l est :

$$\begin{aligned} &\int_{\Omega_X \otimes e_T^l} \mathbf{v}^*(\mathbf{x}, t) \rho \dot{\mathbf{v}}(\mathbf{x}, t) + \varepsilon(\mathbf{v}^*(\mathbf{x}, t)) \sigma(\mathbf{u}(\mathbf{x}, t)) \, d\Omega_X \, dt \\ &+ \int_{\Omega_X \otimes e_T^l} \varepsilon(\mathbf{u}^*(\mathbf{x}, t)) \sigma(\dot{\mathbf{u}}(\mathbf{x}, t) - \mathbf{v}(\mathbf{x}, t)) \, d\Omega_X \, dt = \int_{e_T^l} \int_{\partial\Omega_X(t)} \mathbf{v}^*(\mathbf{x}, t) \mathbf{F}(\mathbf{x}, t) \, d\partial\Omega_X \, dt. \end{aligned} \quad (3.14)$$

L'espace de recherche des champs \mathbf{u} et \mathbf{v} autorisant une discontinuité aux piquets de temps t_l , deux termes assurant la continuité de façon faible au passage de la frontière d'un élément temporel sont ajoutés :

$$\begin{aligned} &\int_{\Omega_X \otimes e_T^l} \mathbf{v}^*(\mathbf{x}, t) \rho \dot{\mathbf{v}}(\mathbf{x}, t) + \varepsilon(\mathbf{v}^*(\mathbf{x}, t)) \sigma(\mathbf{u}(\mathbf{x}, t)) \, d\Omega_X \, dt \\ &+ \int_{\Omega_X \otimes e_T^l} \varepsilon(\mathbf{u}^*(\mathbf{x}, t)) \sigma(\dot{\mathbf{u}}(\mathbf{x}, t) - \mathbf{v}(\mathbf{x}, t)) \, d\Omega_X \, dt \\ &+ \int_{\Omega_X} \mathbf{v}^*(\mathbf{x}, t_{l-1}^+) \rho [[\mathbf{v}(\mathbf{x}, t_{l-1})]] \, d\Omega_X \\ &+ \int_{\Omega_X} \sigma(\mathbf{u}^*(\mathbf{x}, t_{l-1}^+)) [[\varepsilon(\mathbf{u}(\mathbf{x}, t_{l-1}))]] \, d\Omega_X = \int_{e_T^l} \int_{\partial\Omega_X(t)} \mathbf{v}^*(\mathbf{x}, t) \mathbf{F}_d(\mathbf{x}, t) \, d\partial\Omega_X \, dt. \end{aligned} \quad (3.15)$$

On conserve les notations définies en section 3.2.2 pour la matrice de rigidité \mathbb{K}_l et de masse \mathbb{M}_l associés à l'intervalle de temps e_T^l . On pose \mathbf{U}_l^g , \mathbf{U}_l^d , \mathbf{V}_l^g et \mathbf{V}_l^d tels que :

$$\begin{cases} (\mathbf{U}_l^g)_i &= u_{il}^g, \\ (\mathbf{U}_l^d)_i &= u_{il}^d, \\ (\mathbf{V}_l^g)_i &= v_{il}^g, \\ (\mathbf{V}_l^d)_i &= v_{il}^d. \end{cases} \quad (3.16)$$

Remarque : On peut remarquer que dans le cas d'une discrétisation interpolante en espace, on a $u_{il}^g = u(x_i, t_{l-1}^+)$, $u_{il}^d = u(x_i, t_l^-)$, $v_{il}^g = v(x_i, t_{l-1}^+)$ et $v_{il}^d = v(x_i, t_l^-)$. Dans le cas d'une discrétisation isogéométrique en espace, cette égalité ne sera généralement pas vérifiée. Les vecteurs d'inconnues \mathbf{U}_l^g , \mathbf{U}_l^d , \mathbf{V}_l^g , et \mathbf{V}_l^d contiendront les variables de contrôle de la description isogéométrique des champs recherchés.

En injectant 3.10 dans 3.15 il vient :

$$\begin{bmatrix} \frac{1}{2}\mathbb{K}_l & \frac{1}{2}\mathbb{K}_l & -\frac{1}{3}\Delta t_l \mathbb{K}_l & -\frac{1}{6}\Delta t_l \mathbb{K}_l \\ -\frac{1}{2}\mathbb{K}_l & \frac{1}{2}\mathbb{K}_l & -\frac{1}{6}\Delta t_l \mathbb{K}_l & -\frac{1}{3}\Delta t_l \mathbb{K}_l \\ \frac{1}{3}\Delta t_l \mathbb{K}_l & \frac{1}{6}\Delta t_l \mathbb{K}_l & \frac{1}{2}\mathbb{M}_l & \frac{1}{2}\mathbb{M}_l \\ \frac{1}{6}\Delta t_l \mathbb{K}_l & \frac{1}{3}\Delta t_l \mathbb{K}_l & -\frac{1}{2}\mathbb{M}_l & \frac{1}{2}\mathbb{M}_l \end{bmatrix} \begin{bmatrix} \mathbf{U}_l^g \\ \mathbf{U}_l^d \\ \mathbf{V}_l^g \\ \mathbf{V}_l^d \end{bmatrix} = \begin{bmatrix} \mathbb{K}_l \mathbf{U}_{l-1}^d \\ \mathbf{0} \\ \mathbf{F}_l^g + \mathbb{M}_l \mathbf{V}_{l-1}^d \\ \mathbf{F}_l^d \end{bmatrix}, \quad (3.17)$$

où \mathbf{F}_l^g et \mathbf{F}_l^d sont tels que $(\mathbf{F}_l^g)_i = \int_{e_T^l} \int_{\partial\Omega_X(t)} \psi_i^g(t) \phi_i(\mathbf{x}) \mathbf{F}(\mathbf{x}, t) d\partial\Omega_X dt$ et $(\mathbf{F}_l^d)_i = \int_{e_T^l} \int_{\partial\Omega_X(t)} \psi_i^d(t) \phi_i(\mathbf{x}) \mathbf{F}(\mathbf{x}, t) d\partial\Omega_X dt$.

L'équation matricielle 3.17 peut être réécrite de manière à découpler le calcul des champs \mathbf{u} et \mathbf{v} et ainsi réduire la taille du système matriciel à résoudre. En effet, le système 3.17 est équivalent à :

$$\begin{bmatrix} \mathbb{K}_l & 0 & -\frac{1}{3}\Delta t_l \mathbb{K}_l & -\frac{1}{6}\Delta t_l \mathbb{K}_l \\ 0 & \mathbb{K}_l & -\frac{1}{6}\Delta t_l \mathbb{K}_l & -\frac{1}{3}\Delta t_l \mathbb{K}_l \\ 0 & 0 & \mathbb{M}_l^* & \frac{2}{3}\mathbb{M}_l \\ 0 & 0 & \frac{1}{3}\Delta t_l^2 \mathbb{K}_l & \mathbb{M}_l^* \end{bmatrix} \begin{bmatrix} \mathbf{U}_l^g \\ \mathbf{U}_l^d \\ \mathbf{V}_l^g \\ \mathbf{V}_l^d \end{bmatrix} = \begin{bmatrix} \mathbb{K}_l \mathbf{U}_{l-1}^d \\ \mathbb{K}_l \mathbf{U}_{l-1}^d \\ \mathbf{F}_l^{g*} \\ \mathbf{F}_l^{d*} \end{bmatrix}, \quad (3.18)$$

où

$$\begin{cases} \mathbb{M}_l^* &= \mathbb{M}_l + \frac{1}{6}\Delta t_l^2 \mathbb{K}_l, \\ \mathbf{F}_l^{g*} &= \frac{5}{3}(\mathbf{F}_l^g + \mathbb{M}_l \mathbf{V}_{l-1}^d) - \frac{1}{3}\mathbf{F}_l^d - \frac{2}{3}\Delta t_l \mathbb{K}_l \mathbf{U}_{l-1}^d, \\ \mathbf{F}_l^{d*} &= \mathbf{F}_l^g + \mathbf{F}_l^d + \mathbb{M}_l \mathbf{V}_{l-1}^d - \Delta t_l \mathbb{K}_l \mathbf{U}_{l-1}^d. \end{cases} \quad (3.19)$$

On peut donc résoudre le problème en vitesse

$$\begin{bmatrix} \mathbb{M}_l^* & \frac{2}{3}\mathbb{M}_l \\ \frac{1}{3}\Delta t_l^2 \mathbb{K}_l & \mathbb{M}_l^* \end{bmatrix} \begin{bmatrix} \mathbf{V}_l^g \\ \mathbf{V}_l^d \end{bmatrix} = \begin{bmatrix} \mathbf{F}_l^{g*} \\ \mathbf{F}_l^{d*} \end{bmatrix}, \quad (3.20)$$

puis en déduire les inconnues en déplacement sur l'intervalle de temps e_T^l

$$\begin{cases} \mathbf{U}_l^g &= \mathbf{U}_{l-1}^d + \frac{1}{6}\Delta t_l \mathbf{V}_l^g - \frac{1}{6}\Delta t_l \mathbf{V}_l^d \\ \mathbf{U}_l^d &= \mathbf{U}_{l-1}^d + \frac{1}{2}\Delta t_l \mathbf{V}_l^g + \frac{1}{2}\Delta t_l \mathbf{V}_l^d \end{cases}. \quad (3.21)$$

Remarque : Il est possible de symétriser le système associé au problème en vitesse en permutant par bloc les lignes de 3.22 :

$$\begin{bmatrix} \frac{1}{3}\Delta t_l^2 \mathbb{K}_l & \mathbb{M}_l^* \\ \mathbb{M}_l^* & \frac{2}{3}\mathbb{M}_l \end{bmatrix} \begin{bmatrix} \mathbf{V}_l^g \\ \mathbf{V}_l^d \end{bmatrix} = \begin{bmatrix} \mathbf{F}_l^{d*} \\ \mathbf{F}_l^{g*} \end{bmatrix}. \quad (3.22)$$

L'algorithme de résolution TDG sur l'ensemble de l'intervalle de temps Ω_T est décrit en Algorithme 3.

Algorithme 3: Résolution TDG.

Initialisation($\mathbf{U}_0^d, \mathbf{V}_0^d$);

pour $l = 1 \rightarrow N$ **faire**

$$\left[\begin{array}{l} \mathbf{F}_l^{g*} \leftarrow \frac{5}{3}(\mathbf{F}_l^g + \mathbb{M}_l \mathbf{V}_{l-1}^d) - \frac{1}{3}\mathbf{F}_l^d - \frac{2}{3}\Delta t_l \mathbb{K}_l \mathbf{U}_{l-1}^d ; \\ \mathbf{F}_l^{d*} \leftarrow \mathbf{F}_l^g + \mathbf{F}_l^d + \mathbb{M}_l \mathbf{V}_{l-1}^d - \Delta t_l \mathbb{K}_l \mathbf{U}_{l-1}^d ; \\ \left[\begin{array}{c} \mathbf{V}_l^g \\ \mathbf{V}_l^d \end{array} \right] \leftarrow \text{résolution} \left(\left[\begin{array}{cc} \frac{1}{3}\Delta t_l^2 \mathbb{K}_l & \mathbb{M}_l^* \\ \mathbb{M}_l^* & \frac{2}{3}\mathbb{M}_l \end{array} \right] \left[\begin{array}{c} \mathbf{V}_l^g \\ \mathbf{V}_l^d \end{array} \right] = \left[\begin{array}{c} \mathbf{F}_l^{d*} \\ \mathbf{F}_l^{g*} \end{array} \right] \right) ; \\ \mathbf{U}_l^g \leftarrow \mathbf{U}_{l-1}^d + \frac{1}{6}\Delta t_l \mathbf{V}_l^g - \frac{1}{6}\Delta t_l \mathbf{V}_l^d ; \\ \mathbf{U}_l^d \leftarrow \mathbf{U}_{l-1}^d + \frac{1}{2}\Delta t_l \mathbf{V}_l^g + \frac{1}{2}\Delta t_l \mathbf{V}_l^d ; \end{array} \right.$$

3.3.2 Algorithme de résolution full-multigrid localisé avec TDG

Dans cette section est détaillé le principe de résolution Full-Multigrid localisé en espace temps. Les résultats présentés précédemment conduisent à l'utilisation du schéma d'intégration temporel TDG. Il faut pour cela définir une grille espace temps localisée, détailler le traitement des conditions aux limites sur les grilles localisées, les opérateurs de prolongement et de restriction et le lisseur utilisé.

Grille espace temps localisée

Cette définition de grille espace temps n'est adaptée que dans le cas où le domaine spatial Ω_X ne dépend pas de l'intervalle de temps considéré. Cependant l'objectif ici est de pouvoir effectuer une résolution FMG spatiotemporelle localisée. Il faut donc définir une grille espace temps pour le cas où le domaine spatial Ω_X dépend de l'intervalle de temps considéré.

Pour définir une grille espace temps localisée Ω_{XT}^{loc} , nous allons tout d'abord associer à chaque élément temporel e_T^l un domaine spatial Ω_X^l tel que $\Omega_X^l \subset \Omega_X$, Ω_X^l pouvant être de mesure nulle. Le domaine Ω_{XT}^{loc} est défini de la façon suivante :

$$\Omega_{XT}^{loc} = \bigcup_{l=1}^{N_T} e_T^l \otimes \Omega_X^l. \quad (3.23)$$

En notant $\mathcal{J}_X^l = \{j/e_X^j \subset \Omega_X^l\}$, on peut réécrire l'équation 3.25 sous la forme :

$$\Omega_{XT}^{loc} = \bigcup_{l=1}^{N_T} \bigcup_{j \in \mathcal{J}_X^l} e_T^l \otimes e_X^j. \quad (3.24)$$

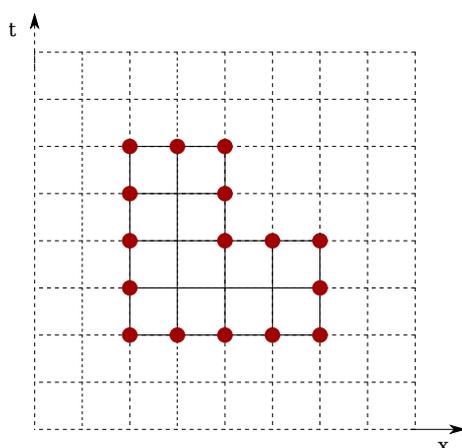


FIGURE 3.7: Définition d'une grille espace temps localisée.

Conditions aux limites dans le cas de la localisation

Pour pouvoir résoudre le problème de dynamique transitoire sur une grille espace temps localisée, il faut imposer des conditions aux limites sur les frontières internes créées entre deux niveaux de discrétisation (voir section 1.2.3). Il est montré dans [CAV 06, GRA 00, GRA 15] que les conditions aux limites sur ces frontières doivent être imposées en vitesse afin de conserver l'énergie totale du système au cours du temps. La figure 3.8 met en évidence tous les cas pouvant se présenter lors de la création d'une grille localisée et la manière dont les conditions aux limites sont traitées.

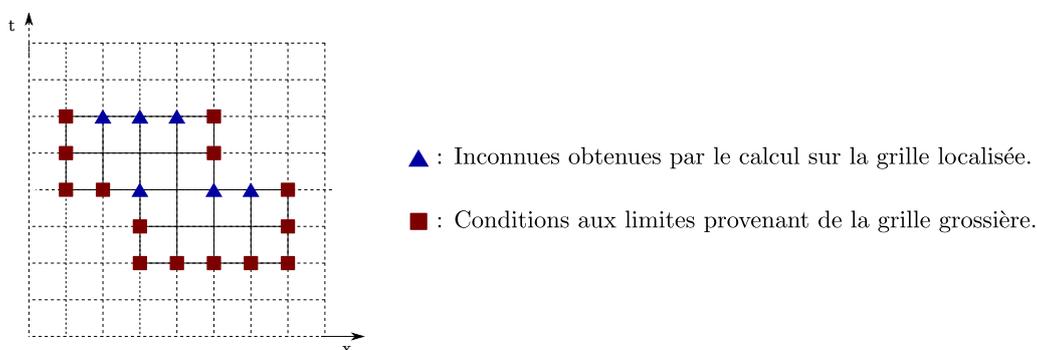


FIGURE 3.8: Gestion des inconnues en vitesse sur les frontières interne dans le cas de la localisation.

Lisseur

Pour chaque intervalle de temps e_T^l , l'opérateur espace temps associé est de la forme $\mathbb{A}_l = \begin{bmatrix} \frac{1}{3}\Delta t_l^2 \mathbb{K}_l & \mathbb{M}_l^* \\ \mathbb{M}_l^* & \frac{2}{3}\mathbb{M}_l \end{bmatrix}$. Cet opérateur est symétrique on peut donc utiliser un gradient conjugué comme lisseur. La stratégie de lissage est un lissage incrémental en temps. Elle consiste donc à effectuer deux ou trois itérations de gradient conjugué sur le système

matriciel associé au premier intervalle de temps e_T^1 , et à partir de la solution obtenue calculer le second membre associé au pas de temps suivant et recommencer l'opération pour le deuxième intervalle de temps e_T^2 . On répète ensuite cette opération jusqu'à avoir couvert l'ensemble des intervalles de temps. Cette opération est décrite Algorithme 4 et schématisée figure 3.9.

Algorithme 4: Opération de lissage sur une grille espace temps.

Initialisation($\mathbf{U}_0^d, \mathbf{V}_0^d$);
pour $l = 1 \rightarrow N$ **faire**
 $\mathbf{F}_l^{g*} \leftarrow \frac{5}{3}(\mathbf{F}_l^g + \mathbb{M}_l \mathbf{V}_{l-1}^d) - \frac{1}{3}\mathbf{F}_l^d - \frac{2}{3}\Delta t_l \mathbb{K}_l \mathbf{U}_{l-1}^d$;
 $\mathbf{F}_l^{d*} \leftarrow \mathbf{F}_l^g + \mathbf{F}_l^d + \mathbb{M}_l \mathbf{V}_{l-1}^d - \Delta t_l \mathbb{K}_l \mathbf{U}_{l-1}^d$;
 $\begin{bmatrix} \mathbf{V}_l^g \\ \mathbf{V}_l^d \end{bmatrix} \leftarrow GC \left(\mathbb{A}_l \begin{bmatrix} \mathbf{V}_l^g \\ \mathbf{V}_l^d \end{bmatrix} = \begin{bmatrix} \mathbf{F}_l^{d*} \\ \mathbf{F}_l^{g*} \end{bmatrix}, 3 \right)$;
 $\mathbf{U}_l^g \leftarrow \mathbf{U}_{l-1}^d + \frac{1}{6}\Delta t_l \mathbf{V}_l^g - \frac{1}{6}\Delta t_l \mathbf{V}_l^d$;
 $\mathbf{U}_l^d \leftarrow \mathbf{U}_{l-1}^d + \frac{1}{2}\Delta t_l \mathbf{V}_l^g + \frac{1}{2}\Delta t_l \mathbf{V}_l^d$;

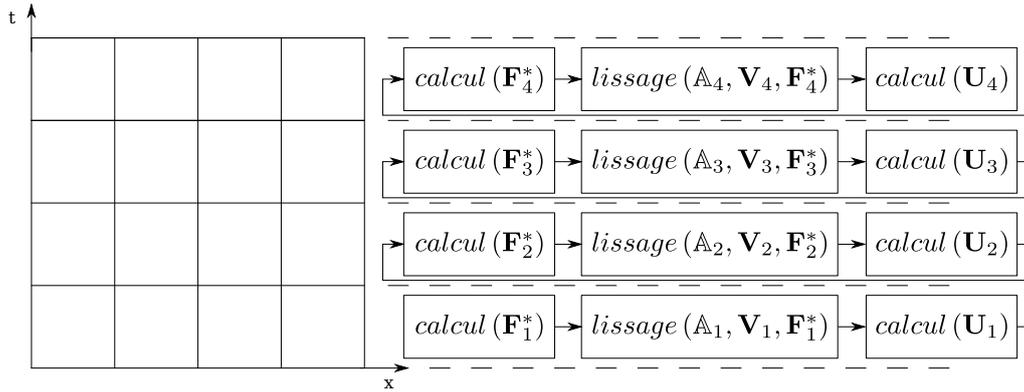


FIGURE 3.9: Schématisation de l'opérations de lissage sur une grille espace temps.

Opérateurs de prolongement et de restriction

La construction de l'opérateur de prolongement entre deux grilles espace temps sur lesquelles sont utilisées le TDG est effectuée en deux étapes. On construit tout d'abord un opérateur de prolongement pour un raffinement global du domaine, puis on utilise un opérateur d'extraction en cas de localisation.

On considère la grille fine espace temps (Figure 3.10(b)) provenant d'un raffinement global de la grille espace temps grossière (Figure 3.10(a)). On rappelle que la discrétisation spatiale est effectuée dans le cadre de l'analyse isogéométrique et que l'intégration temporelle est effectuée avec le TDG.

Les fonctions de base en temps sont linéaires par morceaux et autorisent les discontinuités au passage des frontières d'éléments (Figure 3.6). Il s'agit d'un cas particulier de fonctions de base NURBS de degré 1 associées à un vecteur de nœuds de la forme

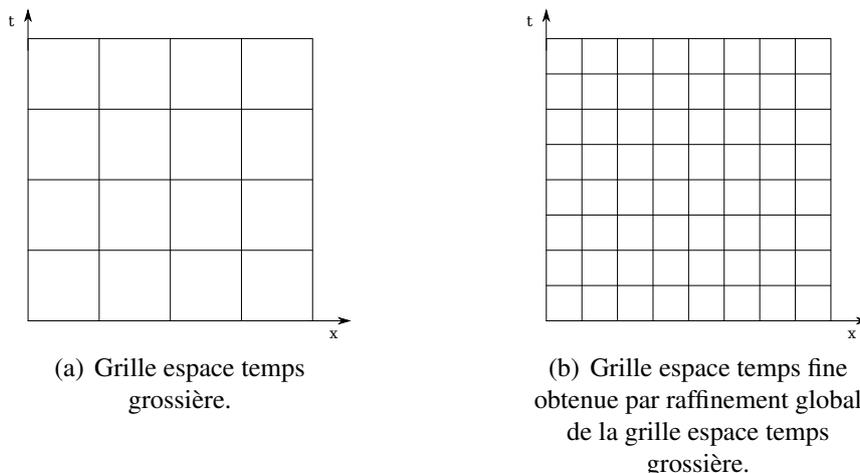


FIGURE 3.10: Grilles espace temps dans le cas d'un raffinement global.

$\Xi = \{\xi_0, \xi_0, \xi_1, \xi_1, \dots, \xi_{n-1}, \xi_{n-1}, \xi_n, \xi_n\}$. La discrétisation espace temps mise en place est ramenée à une discrétisation isogéométrique pour laquelle on impose une forme particulière au vecteur de nœuds dans la direction temporelle (figure 3.11).

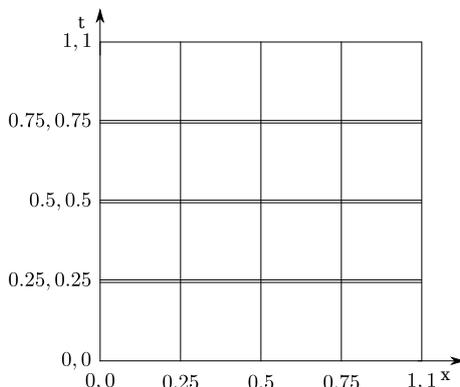


FIGURE 3.11: Exemple de discrétisation espace temps NURBS équivalente avec les vecteur de nœuds correspondant pour une discrétisation NURBS de degré 1 en espace et l'utilisation d'un intégrateur TDG. Les doubles lignes horizontales symbolisent les discontinuités autorisée par la discrétisation temporelle.

On peut donc construire un opérateur de prolongement espace temps global \mathcal{P}_g^{XT} entre la grille espace temps grossière et la grille espace temps fine de la même façon que pour une discrétisation NURBS en espace (section 2.2.1). L'opérateur de prolongement espace temps global \mathcal{P}_g^{XT} est obtenu par composition de l'opérateur de prolongement global en temps \mathcal{P}_T et l'opérateur de prolongement global en espace \mathcal{P}_X (figure 3.12) :

$$\mathcal{P}_g^{XT} = \mathcal{P}_T \circ \mathcal{P}_X. \quad (3.25)$$

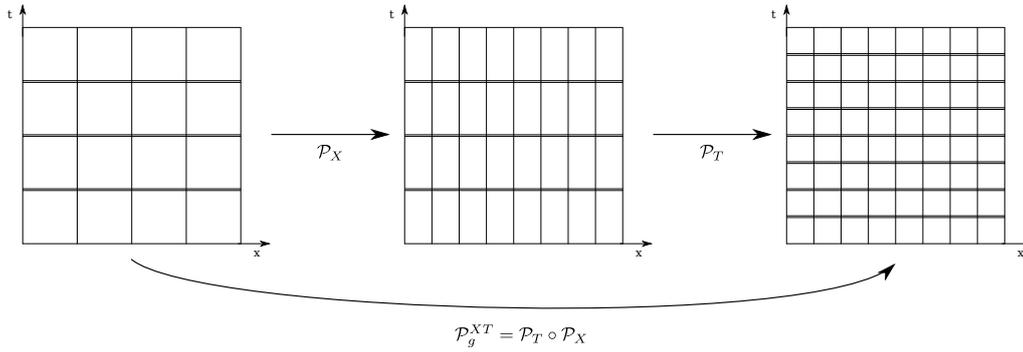


FIGURE 3.12: Construction de l'opérateur de prolongement espace temps global \mathcal{P}_g^{XT} .

Il reste maintenant à définir un opérateur d'extraction espace temps \mathcal{P}_e^{XT} afin de créer un opérateur de prolongement espace temps localisé \mathcal{P}_{loc}^{XT} . Pour cela, les intervalles de temps e_T^l sont traités séparément. Pour chaque intervalle de temps, un opérateur d'extraction spatial \mathcal{P}_e^l est défini en suivant le même raisonnement qu'en section 2.2.1 (figure 2.6). Cette opération nous donne l'opérateur d'extraction espace temps \mathcal{P}_e^{XT} (figure 3.13).

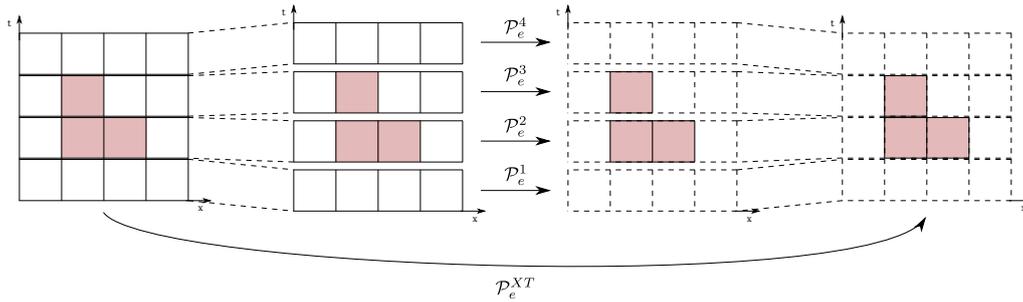


FIGURE 3.13: Construction de l'opérateur d'extraction espace temps \mathcal{P}_e^{XT} .

L'opérateur de prolongement localisé est donc la composition des deux opérateurs \mathcal{P}_g^{XT} et \mathcal{P}_e^{XT} : $\mathcal{P}_{loc}^{XT} = \mathcal{P}_e^{XT} \circ \mathcal{P}_g^{XT}$ (figure 3.14).

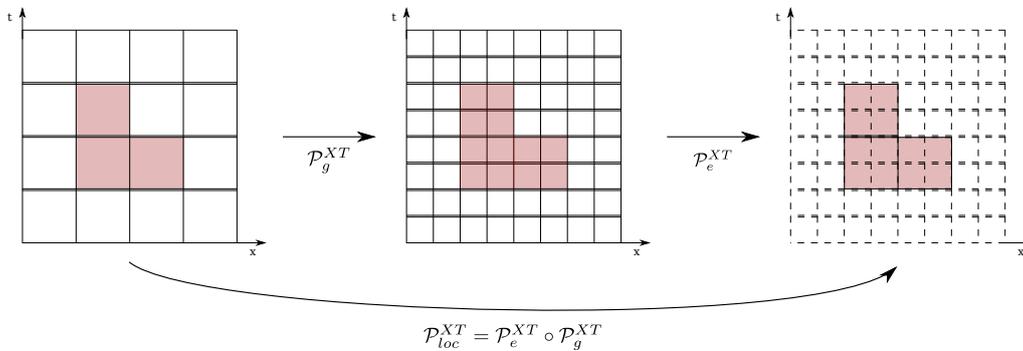


FIGURE 3.14: Construction de l'opérateur de prolongement espace temps localisé \mathcal{P}_{loc}^{XT} .

Indicateur d'erreur

On se donne un ensemble de grilles espace temps $\{\Omega_1, \Omega_2, \dots, \Omega_k, \Omega_{k+1}\}$. L'indicateur d'erreur de la solution sur la grille Ω_k est calculé à l'aide des solutions obtenues sur les grilles Ω_k et Ω_{k+1} . Il faut définir un indicateur d'erreur espace temps afin de caractériser la convergence de la résolution et de construire une grille plus fine si la solution obtenue n'est pas assez précise. [TIE 03, CAV 05] montrent qu'un indicateur d'erreur en énergie de déformation et en énergie cinétique est bien adapté aux problèmes de raffinement adaptatifs pour des problèmes de dynamique transitoire.

De la même manière qu'en section 2.2.2, l'indicateur d'erreur global est utilisé pour caractériser la convergence de l'algorithme et l'indicateur d'erreur local qui en découle sert à construire une grille plus fine si nécessaire. Avec $\Omega = \Omega_x \times \Omega_T$, l'indicateur d'erreur global est le suivant :

$$\sqrt{\frac{\int_{\Omega} |e_{k+1} - e_k| d\Omega}{\int_{\Omega} e_0 d\Omega}},$$

avec $e = \frac{1}{2} \mathbf{C} \boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{u}) + \frac{1}{2} \dot{\mathbf{v}} \rho \dot{\mathbf{v}}$.

L'indicateur d'erreur local et la présence de grilles localisées est géré de façon similaire au raisonnement développé en section 2.2.2.

Algorithme de résolution FMG localisé espace temps

Tous les outils pour effectuer une résolution FMG localisée en espace temps sont maintenant définis. Les données d'entrées sont une discrétisation grossière espace temps Ω_{XT}^1 du problème et les conditions aux limites. On pose \mathbf{X}^k et \mathbf{B}^k respectivement le vecteur d'inconnues généralisées et le vecteur d'efforts généralisés associés à la grille Ω_{XT}^k . On note $\mathcal{P}_{k-1 \rightarrow k}$ l'opérateur de prolongement de la grille Ω_{XT}^{k-1} vers la grille Ω_{XT}^k et $\mathcal{R}_{k \rightarrow k-1} = \mathcal{P}_{k-1 \rightarrow k}^T$ l'opérateur de restriction de la grille Ω_{XT}^k vers la grille Ω_{XT}^{k-1} .

Dans un souci de clarté, on présente tout d'abord le principe de résolution multigrille espace temps pour deux grilles Ω_{XT}^1 et Ω_{XT}^2 , où Ω_{XT}^1 est la grille grossière et Ω_{XT}^2 la grille fine. Le principe de résolution sur deux grilles espace temps est le suivant (Algorithme 5) :

1. on initialise la résolution avec une solution sur grille fine \mathbf{X}_{init}^2 .
2. Une opération de lissage (cf. Algorithme 4) est effectuée sur la grille fine Ω_{XT}^2 donnant $\bar{\mathbf{X}}^2$.
3. Le résidu \mathbf{R}^2 associé à $\bar{\mathbf{X}}^2$ est calculé et restreint sur la grille grossière Ω_{XT}^1 .
4. La correction \mathbf{E}^1 est calculée par résolution directe (Algorithme 3) sur la grille grossière Ω_{XT}^1 .
5. La correction \mathbf{E}^1 est prolongée sur la grille fine et utilisée pour corriger la solution $\bar{\mathbf{X}}^2$.
6. Une opération de lissage est effectuée sur la grille fine Ω_{XT}^2 donnant \mathbf{X}^2 .
7. Si le critère de convergence du cycle multigrille n'est pas vérifié, les opérations 2 à 6 sont répétées.

Algorithm 5: Résolution espace temps sur 2 grilles.

Données d'entrées : $\mathbf{X}^2 \leftarrow \mathbf{X}_{init}^2$; $\{\Omega_{XT}^1, \Omega_{XT}^2\}$;

tant que $res > \varepsilon$ **faire**

$\bar{\mathbf{X}}^2 \leftarrow lissage(\Omega_{XT}^2, \mathbf{X}^2)$;
 $\mathbf{R}^2 \leftarrow résidu(\Omega_{XT}^2, \bar{\mathbf{X}}^2)$;
 $\mathbf{E}^1 \leftarrow résolution_TDG(\Omega_{XT}^1, \mathcal{R}(\mathbf{R}^2))$;
 $\hat{\mathbf{X}}^2 \leftarrow \bar{\mathbf{X}}^2 + \mathcal{P}(\mathbf{E}^1)$;
 $\mathbf{X}^2 \leftarrow lissage(\Omega_{XT}^2, \hat{\mathbf{X}}^2)$;
 $res \leftarrow critère(\Omega_{XT}^2, \mathbf{X}^2)$;

La figure 3.15 représente de manière simplifiée le principe de résolution sur deux grilles espace temps.

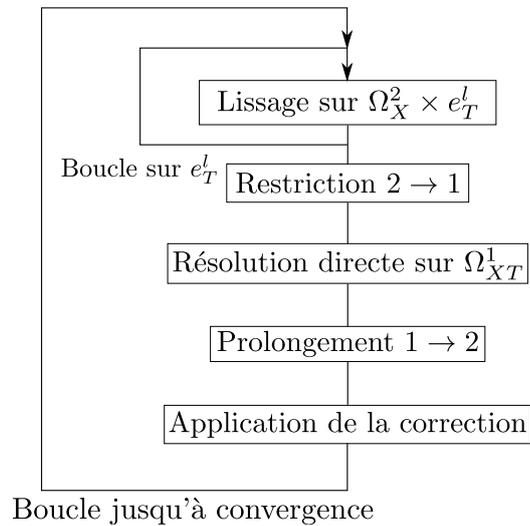


FIGURE 3.15: Principe de résolution sur deux grilles espace temps.

Comme pour une résolution multigrille en espace, cette résolution peut être appliquée récursivement à k grilles si la résolution directe du problème sur la grille grossière est trop

3. Résolution multigrille localisée en espace temps pour la dynamique transitoire

coûteuse. La résolution multigrille espace temps sur k grilles est décrite Algorithme 6.

Algorithme 6: Résolution multigrille espace temps.

Données d'entrées : \mathbf{X}^k ; $\{\Omega_{XT}^1, \Omega_{XT}^2, \dots, \Omega_{XT}^k\}$; \mathbf{B}^k ;
tant que $res > \varepsilon$ **faire**
 $\bar{\mathbf{X}}^k \leftarrow \text{lissage}(\Omega_{XT}^k, \mathbf{X}^k)$;
 $\mathbf{R}^k \leftarrow \text{résidu}(\Omega_{XT}^k, \bar{\mathbf{X}}^k)$;
 $\mathbf{E}^{k-1} \leftarrow \text{résolution_MG}(\mathbf{0}, \{\Omega_{XT}^1, \Omega_{XT}^2, \dots, \Omega_{XT}^{k-1}\}, \mathcal{R}_{k \rightarrow k-1}(\mathbf{R}^k))$;
 $\hat{\mathbf{X}}^k \leftarrow \bar{\mathbf{X}}^k + \mathcal{P}(\mathbf{E}^{k-1})$;
 $\mathbf{X}^k \leftarrow \text{lissage}(\Omega_{XT}^k, \hat{\mathbf{X}}^k)$;
 $res \leftarrow \text{critère}(\Omega_{XT}^k, \mathbf{X}^k, \Omega_{XT}^{k+1}, \mathbf{X}^{k+1})$;

Finalement, le principe de résolution FMG espace temps (décrit en Algorithme 7) est le suivant :

1. effectuer une résolution directe sur la grille grossière Ω_{XT}^1 .
2. Construire une grille plus fine Ω_{XT}^2 avec un raffinement de tous les éléments de Ω_{XT}^1 .
3. Déterminer la solution sur Ω_{XT}^2 à l'aide d'une résolution multigrille (Algorithme 6) avec le jeu de grilles espace temps $\{\Omega_{XT}^1, \Omega_{XT}^2\}$. La procédure de lissage est celle décrite Algorithme 4.
4. Évaluer l'erreur de la solution sur Ω_{XT}^1 à l'aide des solutions sur Ω_{XT}^1 et Ω_{XT}^2 .
5. Construire une grille plus fine Ω_{XT}^3 si le critère de convergence n'est pas respecté. Ω_{XT}^3 peut être une grille localisée.
6. Répéter le cycle résolution multigrille, évaluation d'erreur et construction d'une grille plus fine jusqu'à convergence.

Algorithme 7: Résolution FMG espace temps.

Données d'entrées : Ω_{XT}^1 ;
 $\mathbf{X}^1 \leftarrow \text{résolution_directe}(\Omega_{XT}^1)$;
 $\Omega_{XT}^2 \leftarrow \text{raffinement_homogène}(\Omega_{XT}^1)$;
 $\mathbf{X}^2 \leftarrow \text{résolution_MG}(\mathcal{P}_{1 \rightarrow 2}(\mathbf{X}^1), \{\Omega_{XT}^1, \Omega_{XT}^2\})$;
 $res \leftarrow \text{critère}(\Omega_{XT}^1, \mathbf{X}^1, \Omega_{XT}^2, \mathbf{X}^2)$;
 $k = 3$;
tant que $res > \varepsilon$ **faire**
 $\Omega_{XT}^k \leftarrow \text{raffinement}(\Omega_{XT}^{k-1})$;
 $\mathbf{X}^k \leftarrow \text{résolution_MG}(\{\Omega_{XT}^1, \Omega_{XT}^2, \dots, \Omega_{XT}^{k-1}\}, \mathcal{P}_{k-1 \rightarrow k}(\mathbf{X}^{k-1}))$;
 $res \leftarrow \text{critère}(\Omega_{XT}^{k-1}, \mathbf{X}^{k-1}, \Omega_{XT}^k, \mathbf{X}^k)$;
 $k \leftarrow k + 1$;

3.3.3 Résultats

Nous allons appliquer la méthode de résolution décrite en section précédente au problème 3.16 sur l'intervalle de temps $\Omega_T = [0s; 7,7 \cdot 10^{-3}s]$ afin d'évaluer les performances de cette stratégie de résolution.

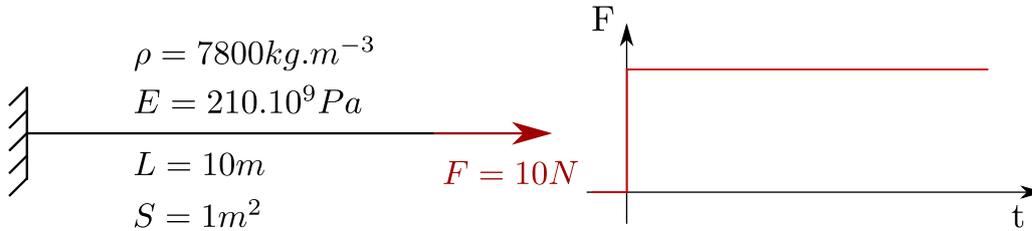


FIGURE 3.16: Problème de référence.

L'algorithme de résolution est initialisé avec le maillage grossier défini figure 3.17(a) contenant 3 éléments en espace et 12 en temps. Le critère de convergence global imposé est $\varepsilon = 1,6 \cdot 10^{-2}$. Le maillage final obtenu à convergence est représenté figure 3.17(b). Quelque soit l'instant considéré, le maillage spatial est toujours localisé autour du front d'onde. En se plaçant en un point $\mathbf{x} = \mathbf{x}_i$ fixe sur la poutre, on observe donc une évolution de la taille de maille locale. Cette taille de maille locale peut notamment décroître, sans qu'aucune procédure de déraffinement ne soit utilisée. On peut aussi noter que la largeur de la zone localisée augmente au cours du temps. Cela est dû à l'accumulation des erreurs au cours du temps impliquée par le schéma d'intégration choisi.

La figure 3.18(a) représente l'évolution de l'indicateur d'erreur global en fonction du nombre total de degrés de liberté pour la résolution FMG espace temps localisée, et pour une résolution FMG espace temps sans localisation (Pour laquelle tous les éléments sont raffinisés à chaque étape de raffinement). L'évolution de l'indicateur d'erreur pour ces deux résolutions diffère à partir de l'instant où commence la localisation. On constate que l'introduction de la localisation permet d'augmenter l'ordre de convergence de la résolution.

On peut remarquer sur la figure 3.18(b) que le nombre de cycles multigrille est assez élevé (de l'ordre de 10) lorsqu'on le compare au nombre de cycles multigrille à convergence pour les problèmes de statique lorsque la localisation a commencé (de l'ordre de 5 pour des fonctions NURBS quadratiques en espace, Figure 2.17(b), 2.19(b)). Cela met en évidence le fait que les corrections apportées à la solution sur la grille fine par les grilles plus grossières sont moins pertinentes qu'en statique. L'exemple suivant permet d'apporter une explication à ce phénomène.

3. Résolution multigrille localisée en espace temps pour la dynamique transitoire

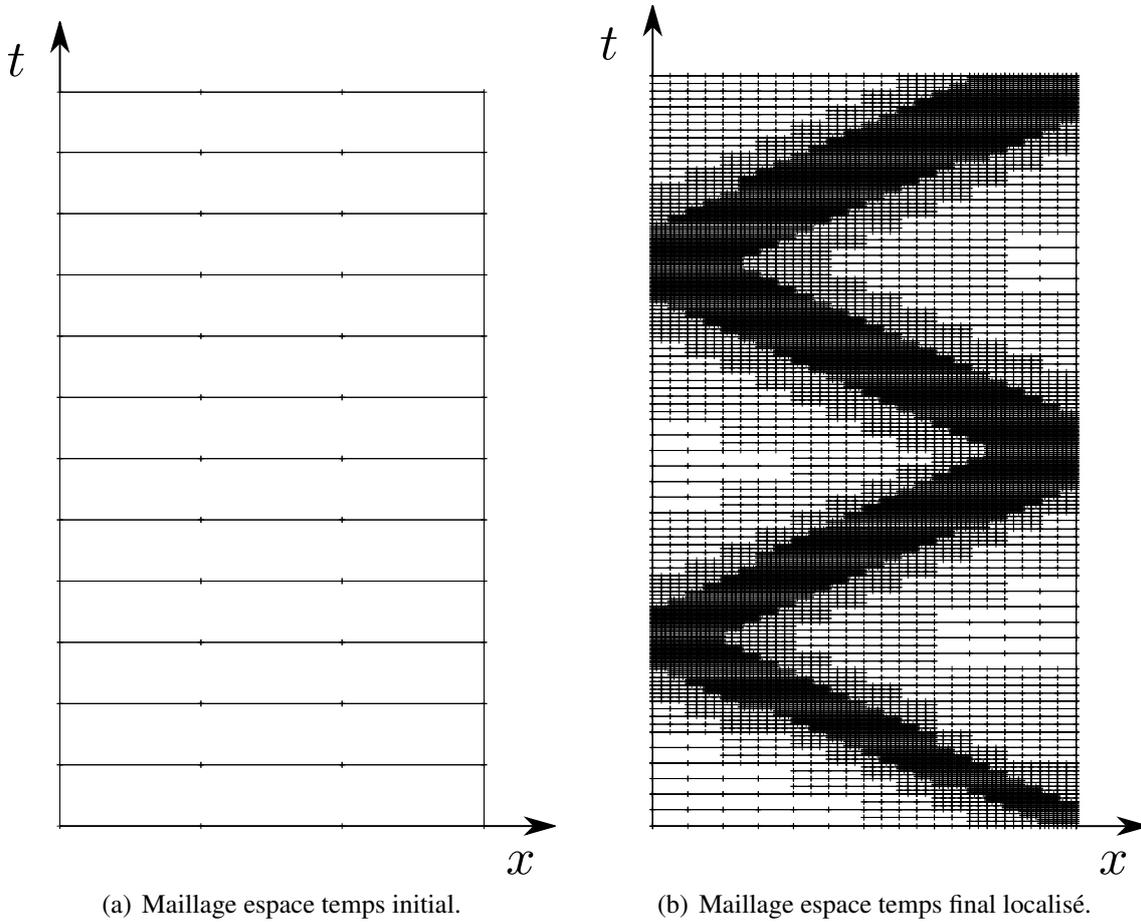


FIGURE 3.17: Maillage initial et final de la résolution.

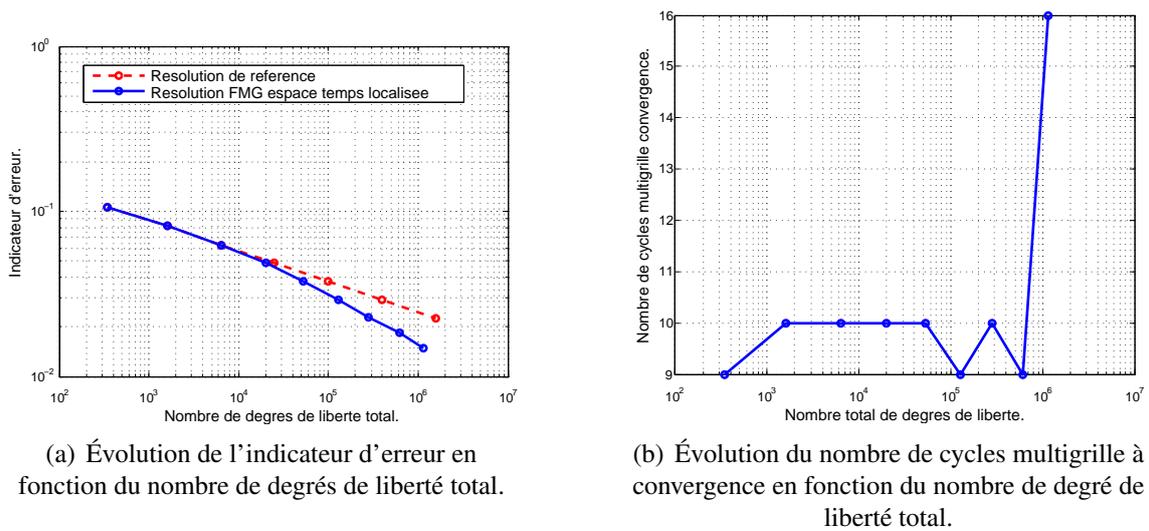


FIGURE 3.18: Performances de la résolution FMG espace temps localisée.

On effectue une résolution sur deux grilles du problème 3.16 avec une discrétisation grossière comptant 120 éléments en espace et 120 éléments en temps et une discrétisation fine comptant 240 éléments en espace et 240 éléments en temps (soit 480 inconnues en vitesse par intervalle de temps sur la grille fine). On s'intéressera à l'évolution de la norme du résidu en fonction du nombre de cycles multigrille avec la prise en compte de la correction provenant de la grille grossière et sans cette prise en compte (Figure 3.19).

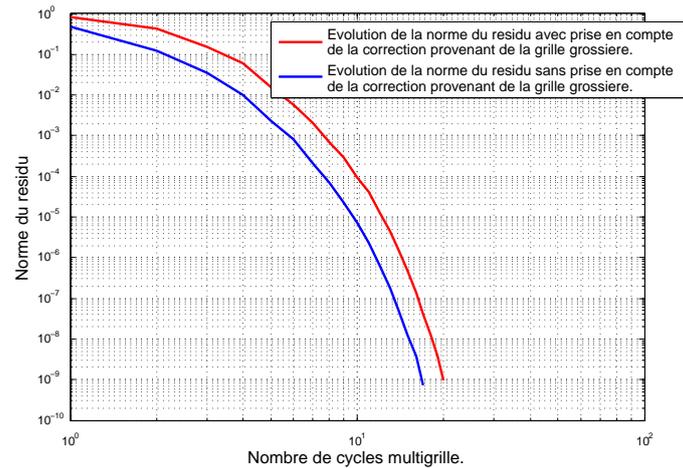


FIGURE 3.19: Performances de la résolution sur deux grilles.

La prise en compte de la correction provenant de la grille grossière entraîne une augmentation du nombre de cycles multigrille à convergence. Cela peut être dû au fait que l'étape de prolongement de la correction de la grille grossière vers la grille fine introduit une erreur associée à une moyenne fréquence suivant la direction temporelle qui n'était pas présente avant la correction. Ce phénomène est mis en évidence figure 3.20.

On y représente la transformée de Fourier discrète de l'erreur commise sur le champ de vitesse lors de la prise en compte de la correction provenant de la grille grossière sur les figures de la colonne de gauche, et sans la prise en compte de cette correction sur la colonne de droite. Les figures 3.20(a) et 3.20(b) représentent l'erreur après la première opération de lissage sur la grille fine pour ces deux cas.

La figure 3.20(c) correspond à l'erreur après prise en compte de la correction provenant de la grille grossière. On constate l'apparition de ligne d'erreur verticales d'abscisse 120 et 360. Ce sont des erreurs de moyenne fréquence en temps introduites par le prolongement de la correction calculée sur la grille grossière. La figure 3.20(d) est l'erreur sans prise en compte de cette correction est donc identique à l'erreur présentée figure 3.20(a) et 3.20(b).

La figure 3.20(f) décrit l'erreur après la seconde étape de lissage sans prise en compte de la correction provenant de la grille grossière. Enfin, la figure 3.20(e) représente l'erreur après la seconde étape de lissage avec prise en compte de cette correction. Les erreurs moyenne fréquence en temps y sont toujours présentes et ne sont donc pas rapidement corrigées par le lisseur.

3. Résolution multigrille localisée en espace temps pour la dynamique transitoire

Cette introduction d'erreurs moyennes fréquences en temps ralentit la convergence de la résolution et donc l'efficacité de la résolution sur grille fine.

L'aspect multigrille permet donc de déterminer une solution sur un ensemble de grilles, pouvant être localisées, sans avoir à traiter les incompatibilités de maillages et de permettre l'évolution des conditions aux limites sur les bords des grilles localisées. Cette dernière propriété est très intéressante pour le traitement de problèmes présentant des non linéarités. Cependant, cela se paye par un coût de calcul plus important qu'une résolution sans prise en compte de la correction provenant des grilles grossières.

Une évolution de la stratégie de résolution FMG localisée espace temps exploitant le potentiel d'accélération d'un solveur multigrille est présentée en section suivante.

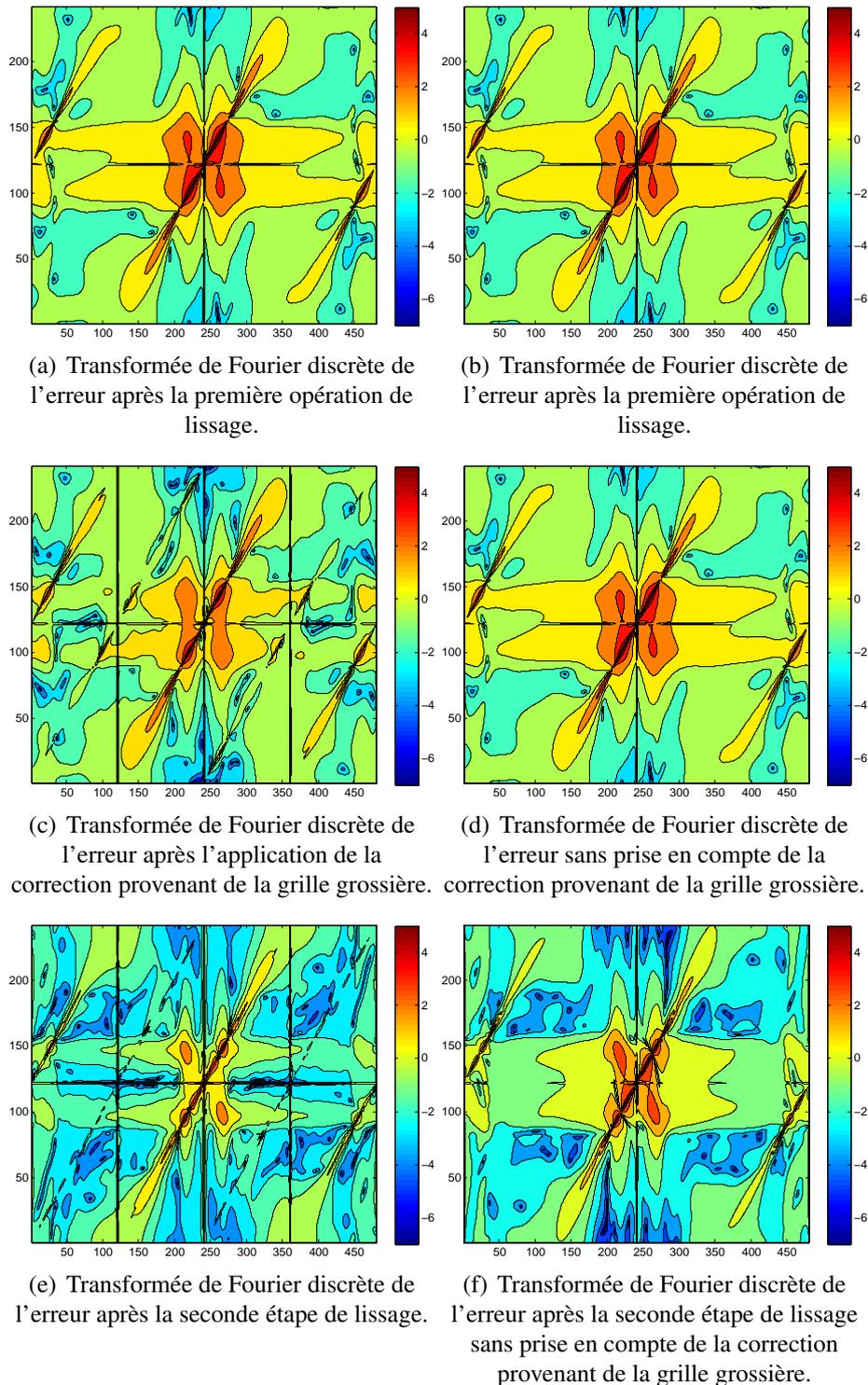


FIGURE 3.20: Transformée de Fourier discrète de l'erreur sur le champ de vitesse. En abscisse se trouvent les fréquences associées à la dimension temporelle et en ordonnée les fréquences associées à la dimension spatiale.

3.3.4 Algorithme de résolution full-multigrid localisé par tranches de temps

Cette stratégie de résolution est une amélioration de celle présentée en section 3.3.2, permettant de conserver l'avantage de l'évolution des conditions aux limites sur les grilles localisées tout en présentant de meilleures performances qu'une résolution sur grille fine sans prise en compte de correction provenant des grilles grossières.

On définit la tranche de temps l du niveau de discrétisation k comme étant le domaine $\Omega_X^k \times e_T^l$. La résolution du problème en vitesse à chaque intervalle de temps de la grille fine va être effectuée successivement par une résolution multigrille jusqu'à convergence sur chacune des tranches de temps. Pour plus de clarté, la résolution va être détaillée avec deux grilles espace temps sans localisation, puis généralisée à une résolution FMG.

On rappelle la résolution multigrille espace temps sur deux grilles mise en place précédemment Algorithme 8.

Algorithme 8: Résolution espace temps sur 2 grilles.

Données d'entrées : \mathbf{X}^2 ; $\{\Omega_{XT}^1, \Omega_{XT}^2\}$;
tant que $res > \varepsilon$ **faire**
 $\bar{\mathbf{X}}^2 \leftarrow lissage(\Omega_{XT}^2, \mathbf{X}^2)$;
 $\mathbf{R}^2 \leftarrow résidu(\Omega_{XT}^2, \bar{\mathbf{X}}^2)$;
 $\mathbf{E}^1 \leftarrow résolution_TDG(\Omega_{XT}^1, \mathcal{R}(\mathbf{R}^2))$;
 $\hat{\mathbf{X}}^2 \leftarrow \bar{\mathbf{X}}^2 + \mathcal{P}(\mathbf{E}^1)$;
 $\mathbf{X}^2 \leftarrow lissage(\Omega_{XT}^2, \hat{\mathbf{X}}^2)$;
 $res \leftarrow critère(\Omega_{XT}^2, \mathbf{X}^2)$;

Cette résolution est schématisée de manière simplifiée figure 3.21(a). Le principe de la résolution multigrille par tranches de temps sur deux grilles consiste à inverser la boucle sur les éléments en temps et la boucle sur les différents niveaux de discrétisation (Figure 3.21(b)). La méthode de résolution est décrite Algorithme 9, en conservant les notations définies en section 3.3.1. Les deux niveaux de discrétisation associés sont donnés figures 3.22.

La résolution FMG par tranches de temps qui en découle est donnée Algorithme 10.

L'algorithme de résolution FMG par tranches de temps est appliqué au problème 3.16 et les résultats obtenus sont comparés à ceux obtenus avec l'algorithme FMG espace temps en figure 3.23. Afin de comparer leurs performances, la précision de la solution obtenue sera quantifiée à l'aide de l'évolution de l'indicateur d'erreur, et le coût en calcul sera quantifié par le nombre de cycles multigrille effectués pour obtenir une solution convergée sur chaque niveau de discrétisation. En effet, le coût de calcul d'une résolution multigrille est de l'ordre de grandeur du coût du nombre de lissages effectués sur la grille la plus fine. Dans le cas de la résolution FMG espace temps localisée, on a

$$C_{XT} = 2 \times (v_1 + v_2) \times N \times n_{MG}^{XT},$$

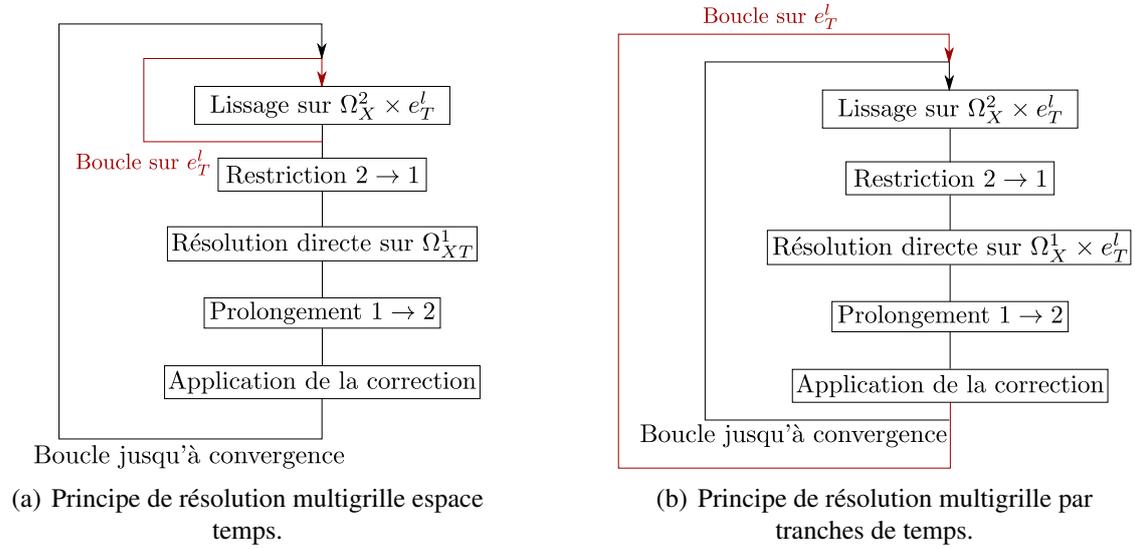


FIGURE 3.21: Comparaison des principes de résolution multigrille espace temps et multigrille par tranches de temps. On rappelle que e_T^l correspond à l'élément l de la discrétisation temporelle sur la grille fine.

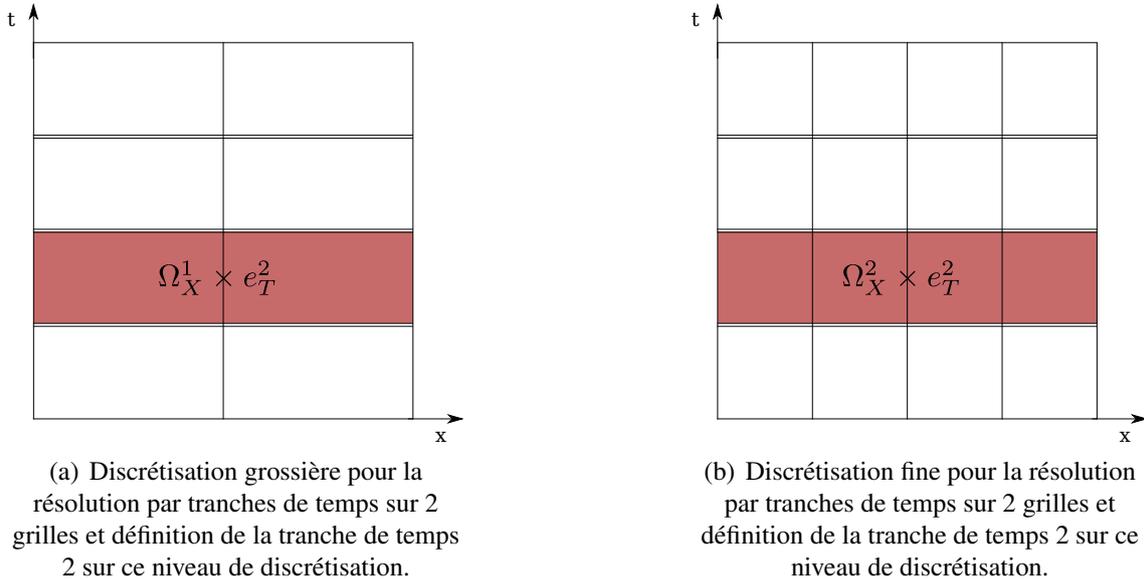


FIGURE 3.22: Grilles grossière et fine utilisées pour une résolution multigrille par tranches de temps. Les deux grilles possèdent la même discrétisation en temps.

où n_{MG}^{XT} est le nombre de cycles multigrille à convergence. Pour la résolution FMG par

3. Résolution multigrille localisée en espace temps pour la dynamique transitoire

Algorithm 9: Résolution multigrille par tranches de temps sur 2 grilles.

```

 $\mathbf{X}_l^2 \leftarrow \mathbf{X}_l^{init};$ 
pour  $l = 1 \rightarrow N$  faire
  tant que  $res > \varepsilon$  faire
     $\bar{\mathbf{X}}_l^2 \leftarrow \text{lissage}(\mathbb{A}_l^2, \mathbf{X}_l^2, \mathbf{B}_l^2, \mathbf{v}_1);$ 
     $\mathbf{R}_l^2 \leftarrow \mathbf{B}_l^2 - \mathbb{A}_l^2 \bar{\mathbf{X}}_l^2;$ 
     $\mathbf{E}_l^1 \leftarrow \text{résolution\_directe}(\mathbb{A}_l^1 \mathbf{E}_l^1 = \mathcal{R}(\mathbf{R}_l^2));$ 
     $\hat{\mathbf{X}}_l^2 \leftarrow \bar{\mathbf{X}}_l^2 + \mathcal{P}(\mathbf{E}_l^1);$ 
     $\mathbf{X}_l^2 \leftarrow \text{lissage}(\mathbb{A}_l^2, \hat{\mathbf{X}}_l^2, \mathbf{B}_l^2, \mathbf{v}_2);$ 
     $res \leftarrow \text{critère}(\mathbb{A}_l^2, \mathbf{X}_l^2, \mathbf{B}_l^2);$ 

```

Algorithm 10: Résolution FMG par tranches de temps.

```

Données d'entrées :  $\Omega_{XT}^1;$ 
 $\mathbf{X}^1 \leftarrow \text{résolution\_directe}(\Omega_{XT}^1);$ 
 $\Omega_{XT}^2 \leftarrow \text{raffinement\_homogène}(\Omega_{XT}^1);$ 
 $\Omega_{XT}^1 \leftarrow \text{raffinement\_homogène\_temporel}(\Omega_{XT}^1);$ 
 $\mathbf{X}^2 \leftarrow \text{résolution\_MG\_pseudo\_espace\_temps}(\mathcal{P}_{1 \rightarrow 2}(\mathbf{X}^1), \{\Omega_{XT}^1, \Omega_{XT}^2\});$ 
 $res \leftarrow \text{critère}(\Omega_{XT}^1, \mathbf{X}^1, \Omega_{XT}^2, \mathbf{X}^2);$ 
 $k = 3;$ 
tant que  $res > \varepsilon$  faire
   $\Omega_{XT}^k \leftarrow \text{raffinement}(\Omega_{XT}^{k-1});$ 
  pour  $i = 0 \rightarrow k - 1$  faire
     $\Omega_{XT}^i \leftarrow \text{raffinement\_homogène\_temporel}(\Omega_{XT}^i);$ 
   $\mathbf{X}^k \leftarrow$ 
   $\text{résolution\_MG\_pseudo\_espace\_temps}(\mathcal{P}_{k-1 \rightarrow k}(\mathbf{X}^{k-1}), \{\Omega_{XT}^1, \dots, \Omega_{XT}^k\});$ 
   $res \leftarrow \text{critère}(\Omega_{XT}^{k-1}, \mathbf{X}^{k-1}, \Omega_{XT}^k, \mathbf{X}^k);$ 

```

tranches de temps localisée, le nombre d'opération de lissage est de

$$C_{pXT} = \sum_{l=1}^N 2 \times (\mathbf{v}_1 + \mathbf{v}_2) \times n_{MG}^l,$$

où n_{MG}^l est le nombre de cycles multigrille à convergence sur l'intervalle de temps e_T^l . On peut donc avoir une idée du rapport des coûts de calcul des deux méthodes

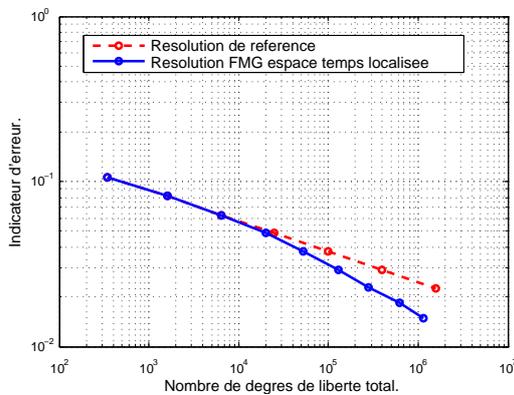
$$\eta = \frac{C_{pXT}}{C_{XT}} = \frac{\text{moyenne}(n_{MG}^l)}{n_{MG}^{XT}},$$

en comparant le nombre de cycles multigrille à convergence pour la résolution FMG espace temps et la moyenne du nombre de cycles multigrille à convergence sur chaque pas

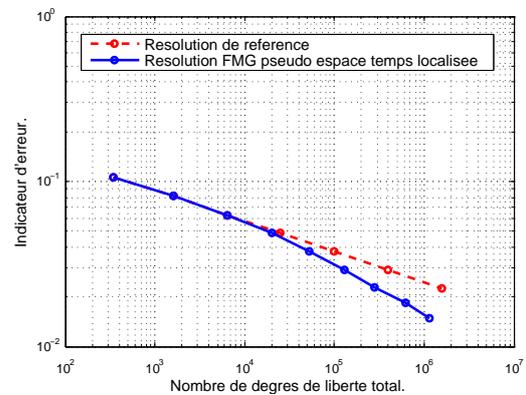
de temps pour la résolution FMG par tranches de temps.

On constate figure 3.25(a) et 3.25(b) que l'évolution de l'indicateur d'erreur est strictement identique pour les deux types de résolution car la procédure de raffinement et de localisation est identique dans les deux cas.

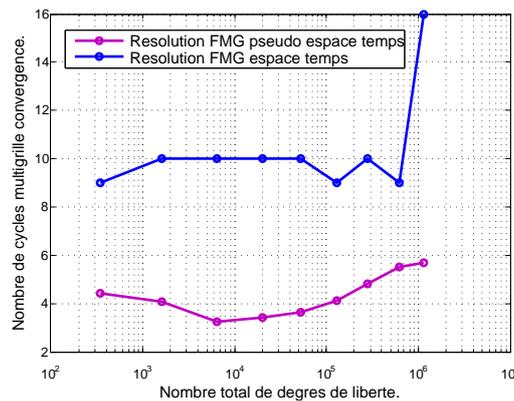
Figure 3.23(c) sont comparés le nombre de cycles multigrille à convergence pour la résolution FMG espace temps localisée et le nombre moyen de cycles multigrille à convergence pour chaque pas de temps pour la résolution FMG par tranches de temps. On constate que le nombre de cycles multigrille effectués jusqu'à convergence sur chacun des niveaux de discrétisation est beaucoup plus faible pour la résolution FMG par tranches de temps : le rapport η varie entre 2 et 3.



(a) Évolution de l'indicateur d'erreur en fonction du nombre de degrés de liberté total pour une résolution FMG espace temps localisée.



(b) Évolution de l'indicateur d'erreur en fonction du nombre de degrés de liberté total pour une résolution FMG par tranches de temps localisée.



(c) Évolution du nombre de cycles multigrille à convergence en fonction du nombre de degré de liberté total pour les deux types de résolution.

FIGURE 3.23: Comparaison des performances de la résolution FMG espace temps localisée et de la résolution FMG par tranches de temps localisée.

La résolution FMG par tranches de temps localisée proposée permet donc de diminuer

significativement les coûts de calcul pour obtenir une solution ayant un même niveau d'erreur que celle obtenue avec une résolution FMG espace temps localisée.

On donne figure 3.25 les maillages finaux obtenus pour les différents types de chargement décrits figure 3.24. Dans le cas de chargement présentant des variations violentes de l'effort F imposé à l'extrémité de la poutre (chargements représentés figures 3.25(a) et 3.25(b)), le procédé de raffinement adaptatif permet bien la localisation autour des fronts d'onde et mène à des maillages très localisés. Dans le cas d'un chargement plus progressif, le maillage final obtenu est beaucoup plus homogène sur l'ensemble du domaine espace temps.

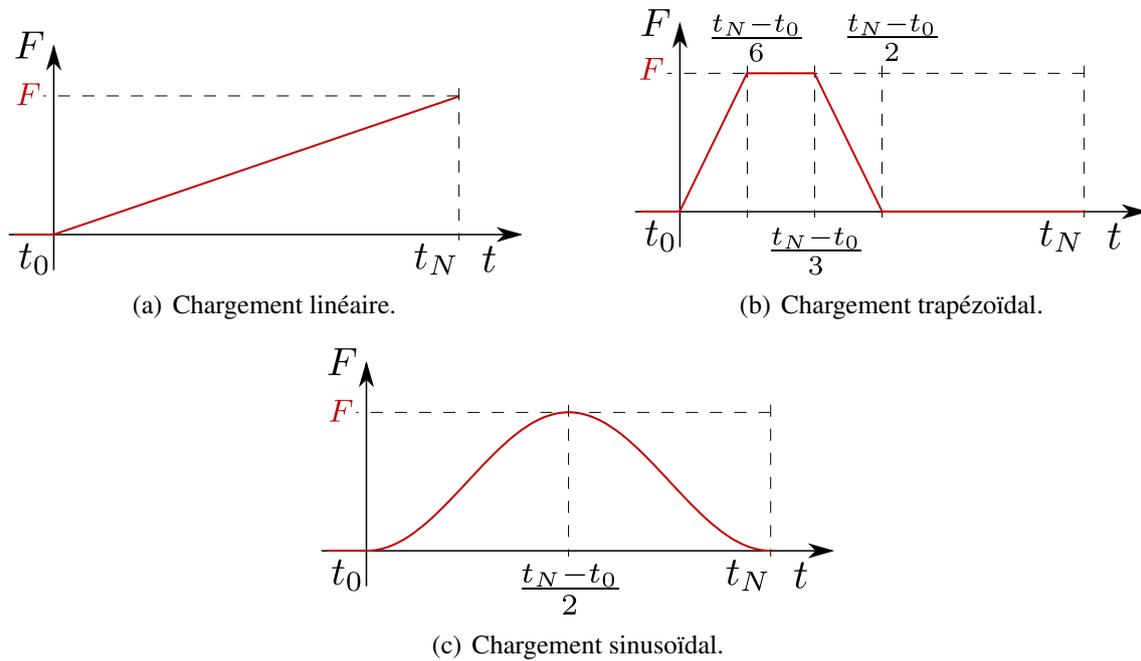
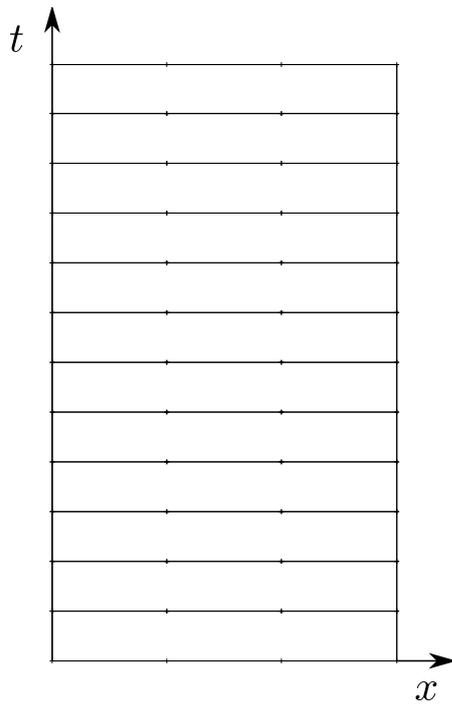
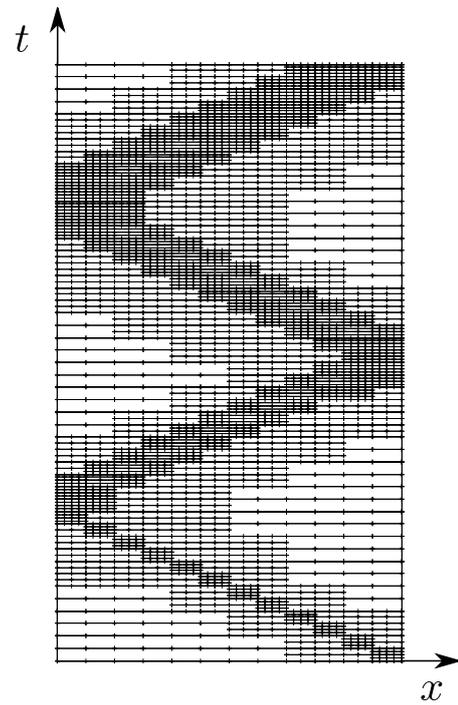


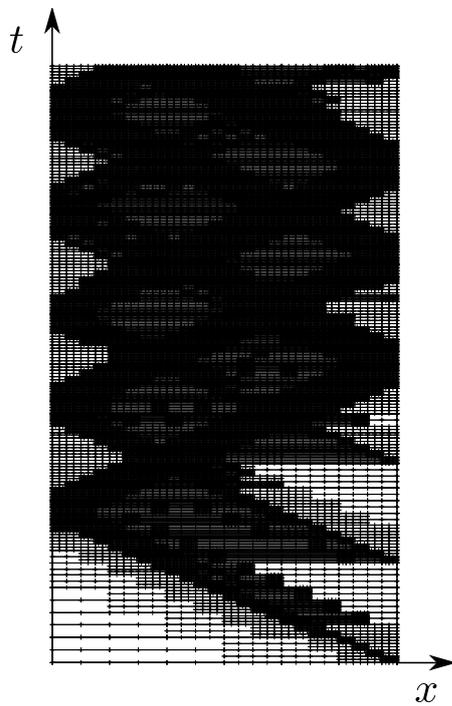
FIGURE 3.24: Chargements imposés.



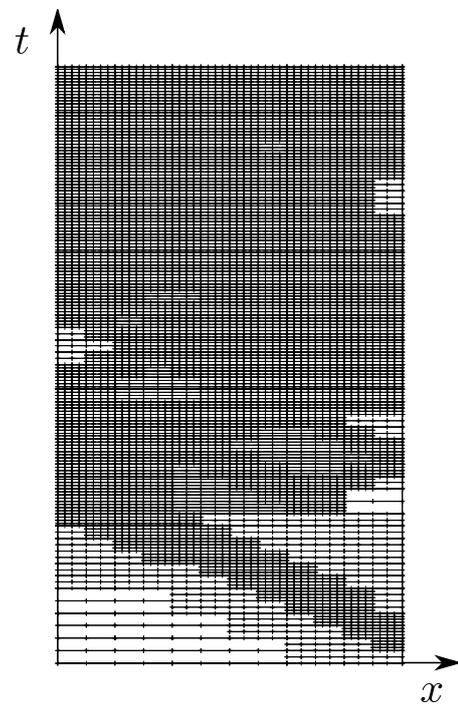
(a) Maillage initial.



(b) Chargement linéaire, critère énergétique global : $1,2 \cdot 10^{-3}$.



(c) Chargement trapézoïdal, critère énergétique global : $2,0 \cdot 10^{-3}$.



(d) Chargement sinusoïdal, critère énergétique global : $1,3 \cdot 10^{-3}$.

FIGURE 3.25: Maillages initial et finaux obtenus pour les trois types de chargement.

3.4 Synthèse

Dans ce chapitre a été étudié la possibilité d'une mise en œuvre d'une résolution multigrille espace temps pour des problèmes de dynamique transitoire dans le cadre de l'élasticité linéaire. L'étude du comportement de solveurs itératifs classiques sur une formulation à deux champs du problème de dynamique nous a mené à l'utilisation d'un intégrateur Galerkin discontinu en temps. L'étude des performances de l'algorithme FMG espace temps avec TDG a mis en évidence le manque de pertinence des corrections provenant des grilles grossières menant à des coûts de calcul trop élevés. Une adaptation de cette méthode de résolution utilisant pleinement le potentiel d'une résolution multigrille a donc été mise en place : la résolution FMG par tranches de temps. Les meilleures performances de ce dernier algorithme sur la résolution FMG espace temps pour obtenir la même solution ont été mises en évidence.

La méthode de résolution FMG par tranches de temps est une méthode à raffinement adaptatif automatique en espace et en temps permettant de ne pas réeffectuer une résolution directe à chaque nouveau raffinement, fournissant une solution définie sur un maillage dépendant du temps dont la taille de maille locale peut décroître au cours du temps sans utiliser de procédure de déraffinement (voir par exemple [COM 00]) et permettant l'évolution des conditions aux limites sur les frontières internes créées par les grilles espaces temps localisées.

Chapitre 4

Bilan – perspectives

Les apports des travaux présentés dans ce manuscrit concernent :

- la création d'une stratégie de raffinement localisé en espace pour les discrétisations isogéométriques,
- la création d'une stratégie de raffinement adaptatif en espace et en temps pour les problèmes d'élasticité linéaire en dynamique transitoire.

La stratégie de raffinement localisé proposée pour des discrétisations IGA est une méthode à raffinement adaptatif automatique en espace basée sur une résolution multigrille localisée. Un critère d'erreur à atteindre pour la solution sur l'ensemble du domaine est imposé par l'utilisateur. Un critère d'erreur local en est déduit et utilisé pour définir les zones du domaine à raffiner afin d'obtenir une solution atteignant la précision imposée. Les zones à raffiner sont regroupées pour former des domaines convexes afin de pouvoir être décrits indépendamment par des fonctions NURBS. Ces zones sont raffinées et utilisées comme discrétisations fines lors de la résolution multigrille. Le processus de raffinement est arrêté lorsque la solution atteint la précision souhaitée par l'utilisateur.

Cette méthode de raffinement adaptatif en espace pour discrétisation IGA (chapitre 2) présente les avantages suivants :

- l'implémentation dans des codes de calculs IGA préexistants est plus simple que l'implémentation des autres méthodes de raffinement localisé en IGA, présentées en section 1.1.4.
- Les opérateurs de prolongement et de restriction servant à la résolution multigrille peuvent être décomposés suivant chacune des directions paramétriques grâce à la structure en produit tensoriel des fonctions de bases NURBS utilisées. L'espace mémoire nécessaire à leur stockage est donc beaucoup moins important que celui nécessaire pour stocker les opérateurs de prolongement et de restriction pour une résolution multigrille avec des éléments finis classiques pour des maillages non structurés.
- L'utilisation d'une discrétisation spatiale IGA pour un algorithme de résolution à raffinement adaptatif contourne les problématiques de raffinement de frontières courbes se présentant lors de discrétisations éléments finis classiques. En effet, leur utilisation nécessite un retour à la définition CAO de la géométrie lors de la procé-

de raffinement des frontières courbes, et implique des étapes de projection qui peuvent se révéler délicates [ADA 00, BIB 13]. La description IGA représentant de manière exacte la géométrie provenant de la CAO, cette question ne se pose donc pas.

La méthode de raffinement adaptatif en espace et en temps pour les problèmes d'élasticité linéaire en dynamique transitoire repose sur la même stratégie de résolution que celle mise en place pour les problèmes d'élasticité linéaire en statique. Le temps est considéré comme une dimension spatiale, menant à la définition d'éléments espace temps. L'impact du changement de la nature des équations différentielles associées au problème, et donc du système matriciel associé, sur une résolution multigrille (section 3.2.3) nous mène à l'utilisation d'un intégrateur TDG pour la dimension temporelle.

Une procédure de lissage, les opérateurs de prolongement et de restriction et la résolution multigrille en découlant sont développés en adoptant une approche espace temps du problème. Le bon fonctionnement de cette méthode est mis en évidence section 3.3.2. Finalement, une amélioration de cette méthode de résolution menant à une solution identique pour moins d'efforts de calculs est présentée (section 3.3.4).

Les avantages de cette dernière méthode sont :

- la résolution mène à une solution respectant un critère d'erreur global espace temps imposé par l'utilisateur à l'aide d'un raffinement adaptatif automatique.
- L'utilisation de méthodes multigrille permet l'évolution des conditions aux limites sur les bords des domaines espace temps localisés. Cet aspect est très important dans le cas de présence non linéarités.
- L'ajout d'un niveau de raffinement plus fin ne nécessite pas de réeffectuer une résolution directe du problème. La résolution multigrille est initialisée avec la solution obtenue précédemment sur les niveaux de discrétisation plus grossiers.

Perspectives

Les travaux sur la localisation en espace pour des discrétisation IGA présentés ici ne permettent pas la localisation selon une diagonale dans l'espace paramétrique. Le parti pris ici consiste à pouvoir représenter les zones localisées par des fonctions de bases indépendantes pour chaque zone. Cela implique de regrouper les éléments à raffiner afin de former des entités contenues dans un domaine convexe dans l'espace paramétrique. Ce choix mène à un raffinement de l'ensemble du domaine lorsque les éléments à raffiner forment une diagonale dans l'espace paramétrique (voir figure 4.1).

Cela peut être amélioré en raffinant tout le domaine et en effectuant l'assemblage de la matrice de rigidité seulement sur les éléments à raffiner. Les opérateurs de prolongement et de restriction seront obtenus en ne conservant que les points de contrôles associés aux fonctions de base NURBS non nulles sur au moins un élément à raffiner. Cette démarche permettrait d'autoriser le raffinement d'entités diagonales.

Il serait également intéressant d'étendre cette stratégie aux comportements non linéaires. Dans ce cas, la résolution multigrille présentée devra être adaptée en utilisant une résolution du type Full Approximation Scheme (FAS). Ce type de résolution multigrille est adaptée aux problèmes non linéaires et est compatible avec une résolution FMG. Une

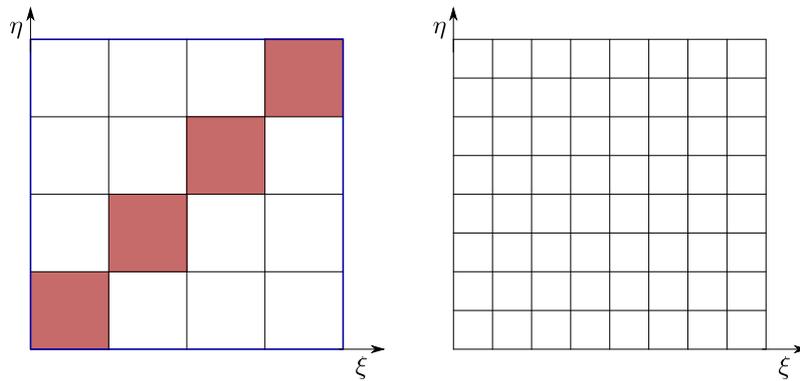


FIGURE 4.1: Les éléments en rouge sur le maillage de gauche représentent une liste d'éléments à raffiner répartis suivant une diagonale de l'espace paramétrique. Le domaine délimité en bleu est le plus petit convexe les contenant. Appliquer la procédure de raffinement localisé développée au chapitre 2 mène à la création du maillage fin de droite. L'ensemble de l'espace est donc raffiné.

résolution type FAS est donc compatible avec la méthode de raffinement adaptatif localisé pour l'IGA présentée.

Le principe de résolution à raffinement adaptatif en espace temps pour les problèmes d'élasticité linéaire en dynamique transitoire est présenté ici pour des problèmes 1D en espace. L'extension de la méthode aux problèmes 2D en espace pour vérifier le bon comportement de l'algorithme de résolution serait intéressant. Pour cela, il est nécessaire d'effectuer une étude sur le préconditionnement du lisseur. En effet, les résultats présentés ici sont obtenus après adimensionnement du problème. Cet adimensionnement est beaucoup moins facile à effectuer pour des problèmes quelconques 2D en espace. Il faut donc compenser cela par un préconditionnement adapté lors de l'opération de lissage.

Un des avantages de la méthode présentée ici est la prise en compte des solutions obtenues sur les grilles fines localisées pour faire évoluer les solutions sur les grilles plus grossières. Ceci n'a pas vraiment d'influence pour des problèmes linéaires mais est nécessaire lors de la présence de non linéarités. Il faut donc étendre cette stratégie de résolution pour pouvoir prendre en compte des non linéarités en remplaçant la résolution multigrille présentée par une résolution multigrille de type FAS. Comme pour la méthode de raffinement adaptatif en espace présentée précédemment, la structure FMG de la résolution à raffinement adaptatif en espace et en temps reste adaptée à l'utilisation d'une méthode de type FAS.

Enfin, la méthode de résolution multigrille à raffinement adaptatif en espace et en temps proposée ne permet pas d'effectuer un raffinement uniquement en espace ou uniquement en temps. Les raffinements se font en espace et en temps simultanément. Il serait intéressant d'adapter le critère pour permettre un choix entre un raffinement seulement en espace, un raffinement seulement en temps, ou alors de coupler les deux. Cela mènerait à l'obtention de solutions sur des maillages plus optimaux que ceux obtenus par la méthode présentée dans ce manuscrit.

Annexe A

Représentation des transformées de Fourier discrètes.

La transformée de Fourier discrète en deux dimension A d'un champs a défini sur une grille structurée $(x, y), x \in \llbracket 1, M \rrbracket$ et $y \in \llbracket 1, N \rrbracket$ est :

$$A(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} a(x, y) e^{-j2\pi(\frac{ux}{M} + \frac{vy}{N})},$$

avec $u \in \llbracket 1, M \rrbracket$ et $v \in \llbracket 1, N \rrbracket$. On a donc :

$$a(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} A(u, v) e^{j2\pi(\frac{ux}{M} + \frac{vy}{N})}.$$

En posant $\omega_x(u) = \frac{u}{M}$ et $\omega_y(v) = \frac{v}{N}$, il reste :

$$a(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} A(u, v) e^{j2\pi(x\omega_x(u) + y\omega_y(v))}.$$

Pour représenter cette transformée de Fourier discrète, le choix effectué dans ce manuscrit est de représenter le champs de valeurs A dans le plan (ω_x, ω_y) , $u \in \llbracket 1, M \rrbracket$ et $v \in \llbracket 1, N \rrbracket$. L'origine du repère est prise en $(\omega_x = 0, \omega_y = 0)$. La figure A.1 donne les détails de cette représentation.

Les valeurs de A représentées le long de l'axe des abscisses sont donc associées à des basses fréquences suivant la direction y , celles représentées le long de l'axe des ordonnées à des basses fréquences suivant la direction x et par conséquent celles représentées au centre de ce graphe correspondent donc à des basses fréquences suivant les directions x et y .

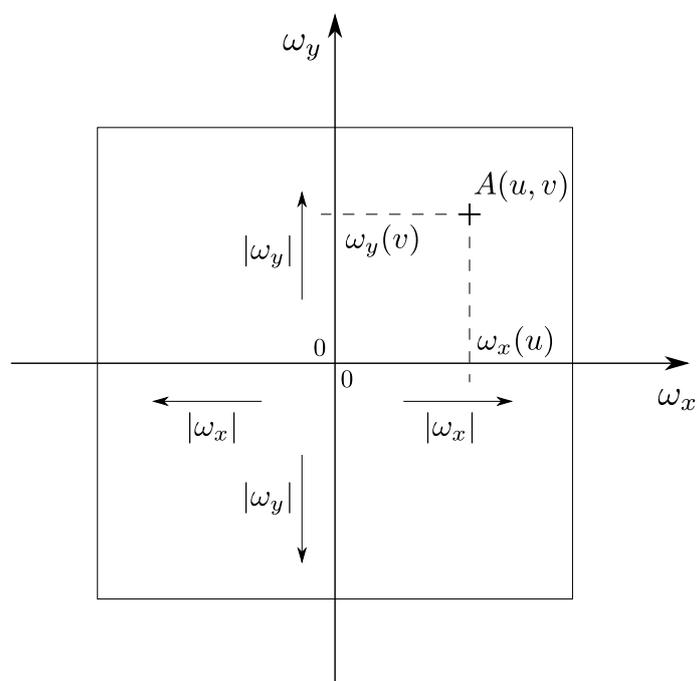


FIGURE A.1: Représentation adoptée pour la transformée de Fourier discrète du champ
a

Annexe B

Solution analytique de la plaque infinie en tractions avec un trou elliptique.

La solution analytique pour la plaque infinie avec un trou elliptique sous traction biaxiale est déterminée en élasticité linéaire dans [GAO 96] par exemple. Appliquée dans le cas d'une traction uniaxiale (figure B.1), on obtient la solution donnée en équation B.1, exprimée dans le système de coordonnées curvlinéaires résultant de la transformation $z = c \cos(\zeta)$, où $z = x + iy$ et $\zeta = \xi + i\eta$. En posant $c = \sqrt{a^2 - b^2}$ et $\xi_0 = \cosh^{-1} \left(\frac{a}{\sqrt{a^2 - b^2}} \right)$.

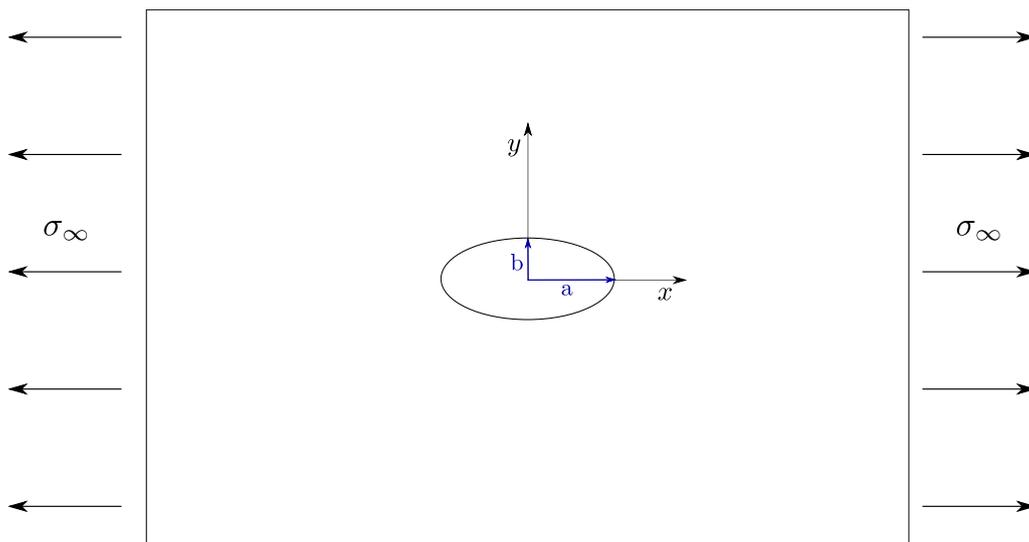


FIGURE B.1: Plaque infinie en traction avec un trou elliptique.

$$\left\{ \begin{array}{l}
 \sigma_{\xi\xi} = -\frac{1}{2}\sigma_{\infty}e^{2\xi_0} \\
 \quad + \frac{1}{2} \frac{\sigma_{\infty}e^{2\xi_0}}{\cosh(2\xi) - \cos(2\eta)} [\sinh(2\xi) + \cosh(2\xi_0) - \cos(2\eta) \cosh(2(\xi - \xi_0))] \\
 \quad + \frac{1}{2} \frac{\sigma_{\infty}e^{2\xi_0}}{(\cosh(2\xi) - \cos(2\eta))^2} \sinh(2\xi) [\cos(2\eta) - \cosh(2\xi_0)] \\
 \quad + \frac{1}{2} \frac{\sigma_{\infty}}{(\cosh(2\xi) - \cos(2\eta))^2} \sinh(2\xi) [\cosh(2\xi) - \cosh(2\xi_0)] \\
 \sigma_{\eta\eta} = -\frac{1}{2}\sigma_{\infty}e^{2\xi_0} \\
 \quad + \frac{1}{2} \frac{\sigma_{\infty}e^{2\xi_0}}{\cosh(2\xi) - \cos(2\eta)} [\sinh(2\xi) - \cosh(2\xi_0) + \cosh(2(\xi - \xi_0)) \cos(2\eta)] \\
 \quad - \frac{1}{2} \frac{\sigma_{\infty}e^{2\xi_0}}{(\cosh(2\xi) - \cos(2\eta))^2} \sinh(2\xi) [\cos(2\eta) - \cosh(2\xi_0)] \\
 \quad + \frac{1}{2} \frac{\sigma_{\infty}}{(\cosh(2\xi) - \cos(2\eta))^2} \sinh(2\xi) [\cosh(2\xi) + \cosh(2\xi_0) - 2\cos(2\eta)] \\
 \sigma_{\xi\eta} = \frac{1}{2} \frac{\sigma_{\infty}e^{2\xi_0}}{(\cosh(2\xi) - \cos(2\eta))^2} \sin(2\eta) \times \\
 \quad [\sinh(2(\xi - \xi_0))(\cos(2\xi) - \cos(2\eta)) - \cosh(2\xi_0) + \cosh(2\xi)] \\
 \quad - \frac{1}{2} \frac{\sigma_{\infty}}{(\cosh(2\xi) - \cos(2\eta))^2} \sin(2\eta) [\cosh(2\xi_0) - \cosh(2\xi)]
 \end{array} \right. \quad (B.1)$$

Bibliographie

- [ABE 04] ABEDI R., CHUNG S.-H., ERICKSON J., FAN Y., GARLAND M., GUOY D., HABER R., SULLIVAN J. M., THITE S., ZHOU Y.
Spacetime meshing with adaptive refinement and coarsening. *Proceedings of the twentieth annual symposium on Computational geometry* ACM, 2004, p. 300–309.
- [ABE 06] ABEDI R., HABER R. B., THITE S., ERICKSON J.
An h-adaptive spacetime-discontinuous Galerkin method for linear elastodynamics. *European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique*, vol. 15, n° 6, 2006, p. 619–642, Taylor & Francis.
- [ADA 00] ADAMS M. F., TAYLOR R.
Parallel multigrid solvers for 3D unstructured finite element problems in large deformation elasticity and plasticity. *International Journal for Numerical Methods in Engineering*, vol. 48, n° 8, 2000, p. 1241–1262, Citeseer.
- [ADA 02] ADAMS M.
Evaluation of three unstructured multigrid methods on 3D finite element problems in solid mechanics. *International Journal for Numerical Methods in Engineering*, vol. 55, n° 5, 2002, p. 519–534, Wiley Online Library.
- [AUB 99] AUBRY D., LUCAS D., TIE B.
Adaptive strategy for transient/coupled problems applications to thermoelasticity and elastodynamics. *Computer methods in applied mechanics and engineering*, vol. 176, n° 1, 1999, p. 41–50, Elsevier.
- [BAI 87] BAI D., BRANDT A.
Local mesh refinement multilevel techniques. *SIAM Journal on Scientific and Statistical Computing*, vol. 8, n° 2, 1987, p. 109–134, SIAM.
- [BAR 99] BARTH T. J.
Numerical methods for gasdynamic systems on unstructured meshes. *An introduction to recent developments in theory and numerics for conservation laws*, p. 195–285 Springer, 1999.
- [BAZ 06a] BAZILEVS Y., CALO V., ZHANG Y., HUGHES T. J.
Isogeometric fluid–structure interaction analysis with applications to arterial blood flow. *Computational Mechanics*, vol. 38, n° 4-5, 2006, p. 310–322, Springer.
- [BAZ 06b] BAZILEVS Y., BEIRAO DA VEIGA L., COTTRELL J. A., HUGHES T. J., SANGALLI G.

- Isogeometric analysis : approximation, stability and error estimates for h-refined meshes. *Mathematical Models and Methods in Applied Sciences*, vol. 16, n° 07, 2006, p. 1031–1090, World Scientific.
- [BAZ 10] BAZILEVS Y., CALO V. M., COTTRELL J. A., EVANS J. A., HUGHES T., LIPTON S., SCOTT M., SEDERBERG T.
Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering*, vol. 199, n° 5, 2010, p. 229–263, Elsevier.
- [BIB 13] BIBOULET N., GRAVOUIL A., DUREISSEIX D., LUBRECHT A., COMBESCURE A.
An efficient linear elastic FEM solver using automatic local grid refinement and accuracy control. *Finite Elements in Analysis and Design*, vol. 68, 2013, p. 28–38, Elsevier.
- [BIO 10] BIOTTEAU E., GRAVOUIL A., LUBRECHT A., COMBESCURE A.
Multigrid solver with automatic mesh refinement for transient elastoplastic dynamic problems. *International Journal for Numerical Methods in Engineering*, vol. 84, n° 8, 2010, p. 947–971, Wiley Online Library.
- [BRA 77] BRANDT A.
Multi-level adaptive solutions to boundary-value problems. *Mathematics of computation*, vol. 31, n° 138, 1977, p. 333–390.
- [BRA 11] BRANDT A., LIVNE O. E.
Multigrid techniques : 1984 guide with applications to fluid dynamics, vol. 67. SIAM, 2011.
- [BRI 00] BRIGGS W. L., MCCORMICK S. F. et al.
A multigrid tutorial. Siam, 2000.
- [CAV 05] CAVIN P., GRAVOUIL A., LUBRECHT A., COMBESCURE A.
Efficient FEM calculation with predefined precision through automatic grid refinement. *Finite Elements in Analysis and Design*, vol. 41, n° 11, 2005, p. 1043–1055, Elsevier.
- [CAV 06] CAVIN P.
Méthode éléments finis avec raffinement spatial et temporel adaptatif et automatique : "STAR-METHOD" (Space Time Automatic Refinement). Thèse de doctorat, Institut National des Sciences Appliquées de Lyon, Juin 2006.
- [COH 01] COHEN E., RIESENFELD R. F., ELBER G.
Geometric modelling with splines : An Introduction. A. K. Peters, 2001.
- [COL 12] COLLIER N., PARDO D., DALCIN L., PASZYNSKI M., CALO V. M.
The cost of continuity : a study of the performance of isogeometric finite elements using direct solvers. *Computer Methods in Applied Mechanics and Engineering*, vol. 213, 2012, p. 353–361, Elsevier.
- [COL 13] COLLIER N., DALCIN L., PARDO D., CALO V. M.
The cost of continuity : performance of iterative solvers on isogeometric finite ele-

- ments. *SIAM Journal on Scientific Computing*, vol. 35, n° 2, 2013, p. A767–A784, SIAM.
- [COM 00] COMBE J.-P.
Sur le contrôle des calculs en dynamique rapide : application aux problèmes d’impact. Thèse de doctorat, École Normale Supérieure de Cachan, Janvier 2000.
- [COM 01] COMBESCURE A., GRAVOUIL A.
A time–space multi-scale algorithm for transient structural nonlinear problems. *Mécanique & industries*, vol. 2, n° 1, 2001, p. 43–55, Elsevier.
- [COT 06] COTTRELL J., REALI A., BAZILEVS Y., HUGHES T.
Isogeometric analysis of structural vibrations. *Computer methods in applied mechanics and engineering*, vol. 195, n° 41, 2006, p. 5257–5296, Elsevier.
- [COT 07] COTTRELL J., HUGHES T., REALI A.
Studies of refinement and continuity in isogeometric structural analysis. *Computer methods in applied mechanics and engineering*, vol. 196, n° 41, 2007, p. 4160–4183, Elsevier.
- [COT 09] COTTRELL J. A., HUGHES T. J., BAZILEVS Y.
Isogeometric analysis : toward integration of CAD and FEA. John Wiley & Sons, 2009.
- [DAN 98a] DANIEL W. J. T.
A study of the stability of subcycling algorithms in structural dynamics. *Computer methods in applied mechanics and engineering*, vol. 156, n° 1, 1998, p. 1–13, Elsevier.
- [DAN 98b] DANIEL W.
Subcycling first-and second-order generalizations of the trapezoidal rule. *International journal for numerical methods in engineering*, vol. 42, n° 6, 1998, p. 1091–1119, Wiley Online Library.
- [DOK 13] DOKKEN T., LYCHE T., PETTERSEN K. F.
Polynomial splines over locally refined box-partitions. *Computer Aided Geometric Design*, vol. 30, n° 3, 2013, p. 331–356, Elsevier.
- [DÖR 10] DÖRFEL M. R., JÜTTLER B., SIMEON B.
Adaptive isogeometric analysis by local h-refinement with T-splines. *Computer methods in applied mechanics and engineering*, vol. 199, n° 5, 2010, p. 264–275, Elsevier.
- [EKE 01] EKEVID T., LI M. X., WIBERG N.-E.
Adaptive FEA of wave propagation induced by high-speed trains. *Computers & Structures*, vol. 79, n° 29, 2001, p. 2693–2704, Elsevier.
- [ERI 05] ERICKSON J., GUOY D., SULLIVAN J. M., ÜNGÖR A.
Building spacetime meshes over arbitrary spatial domains. *Engineering with Computers*, vol. 20, n° 4, 2005, p. 342–353, Springer.
- [ESK 06] ESKILSSON C., SHERWIN S. J.
Spectral/hp discontinuous Galerkin methods for modelling 2D Boussinesq equations. *Journal of Computational Physics*, vol. 212, n° 2, 2006, p. 566–589, Elsevier.

- [EVA 09] EVANS J. A., BAZILEVS Y., BABUŠKA I., HUGHES T. J.
n-Widths, sup–infs, and optimality ratios for the k-version of the isogeometric finite element method. *Computer Methods in Applied Mechanics and Engineering*, vol. 198, n° 21, 2009, p. 1726–1741, Elsevier.
- [FAR 91] FARHAT C., ROUX F.-X.
A method of finite element tearing and interconnecting and its parallel solution algorithm. *International Journal for Numerical Methods in Engineering*, vol. 32, n° 6, 1991, p. 1205–1227, Wiley Online Library.
- [FAR 99] FARIN G. E.
NURBS Curves and Surfaces : from Projective Geometry to Practical Use, Second Edition. A. K. Peters, Ltd., 1999.
- [FAR 02] FARIN G. E.
Curves and surfaces for CAGD : a practical guide. Morgan Kaufmann, 2002.
- [FED 64] FEDORENKO R. P.
The speed of convergence of one iterative process. *USSR Computational Mathematics and Mathematical Physics*, vol. 4, n° 3, 1964, p. 227–235, Elsevier.
- [FEN 00] FENG Y., PERIC D.
Coarse mesh evolution strategies in the Galerkin multigrid method with adaptive re-meshing for geometrically non-linear problems. *International Journal for Numerical Methods in Engineering*, vol. 49, n° 4, 2000, p. 547–571.
- [FIS 92] FISH J.
The s-version of the finite element method. *Computers & Structures*, vol. 43, n° 3, 1992, p. 539–547, Elsevier.
- [FIS 95] FISH J., PANDHEERADI M., BELSKY V.
An efficient multilevel solution scheme for large scale non-linear systems. *International Journal for Numerical Methods in Engineering*, vol. 38, n° 10, 1995, p. 1597–1610, Wiley Online Library.
- [GAO 96] GAO X.-L.
A general solution of an infinite elastic plate with an elliptic hole under biaxial loading. *International journal of pressure vessels and piping*, vol. 67, n° 1, 1996, p. 95–104, Elsevier.
- [GEE 05] GEE M., RAMM E., WALL W. A.
Parallel multilevel solution of nonlinear shell structures. *Computer methods in applied mechanics and engineering*, vol. 194, n° 21, 2005, p. 2513–2533, Elsevier.
- [GIA 12] GIANNELLI C., JÜTTLER B., SPELEERS H.
THB-splines : The truncated basis for hierarchical splines. *Computer Aided Geometric Design*, vol. 29, n° 7, 2012, p. 485–498, Elsevier.
- [GÓM 08] GÓMEZ H., CALO V. M., BAZILEVS Y., HUGHES T. J.
Isogeometric analysis of the Cahn–Hilliard phase-field model. *Computer methods in applied mechanics and engineering*, vol. 197, n° 49, 2008, p. 4333–4352, Elsevier.

- [GRA 00] GRAVOUIL A.
Méthode multi-échelles en temps et en espace avec décomposition de domaines pour la dynamique non-linéaire des structures. Thèse de doctorat, École Normale Supérieure de Cachan, 2000.
- [GRA 01] GRAVOUIL A., COMBESURE A.
Multi-time-step explicit–implicit method for non-linear structural dynamics. *International Journal for Numerical Methods in Engineering*, vol. 50, n° 1, 2001, p. 199–225, Wiley Online Library.
- [GRA 03] GRAVOUIL A., COMBESURE A.
Multi-time-step and two-scale domain decomposition method for non-linear structural dynamics. *International Journal for Numerical Methods in Engineering*, vol. 58, n° 10, 2003, p. 1545–1569, Wiley Online Library.
- [GRA 15] GRAVOUIL A., COMBESURE A., BRUN M.
Heterogeneous asynchronous time integrators for computational structural dynamics. *International Journal for Numerical Methods in Engineering*, vol. 102, n° 3-4, 2015, p. 202–232, Wiley Online Library.
- [GRO 12] GROSSMANN D., JÜTTLER B., SCHLUSNUS H., BARNER J., VUONG A.-V.
Isogeometric simulation of turbine blades for aircraft engines. *Computer Aided Geometric Design*, vol. 29, n° 7, 2012, p. 519–531, Elsevier.
- [HUG 78] HUGHES T., LIU W.
Implicit-explicit finite elements in transient analysis : stability theory. *Journal of applied Mechanics*, vol. 45, n° 2, 1978, p. 371–374, American Society of Mechanical Engineers.
- [HUG 88] HUGHES T. J., HULBERT G. M.
Space-time finite element methods for elastodynamics : formulations and error estimates. *Computer methods in applied mechanics and engineering*, vol. 66, n° 3, 1988, p. 339–363, Elsevier.
- [HUG 96] HUGHES T. J., STEWART J. R.
A space-time formulation for multiscale phenomena. *Journal of Computational and Applied Mathematics*, vol. 74, n° 1, 1996, p. 217–229, Elsevier.
- [HUG 05] HUGHES T. J., COTTRELL J. A., BAZILEVS Y.
Isogeometric analysis : CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer methods in applied mechanics and engineering*, vol. 194, n° 39, 2005, p. 4135–4195, Elsevier.
- [HUL 90] HULBERT G. M., HUGHES T. J.
Space-time finite element methods for second-order hyperbolic equations. *Computer Methods in Applied Mechanics and Engineering*, vol. 84, n° 3, 1990, p. 327–348, Elsevier.
- [JOH 14] JOHANNESSEN K. A., KVAMSDAL T., DOKKEN T.
Isogeometric analysis using LR B-splines. *Computer Methods in Applied Mechanics and Engineering*, vol. 269, 2014, p. 471–514, Elsevier.

- [KAC 93] KACOU S., PARSONS I.
A parallel multigrid method for history-dependent elastoplasticity computations. *Computer methods in applied mechanics and engineering*, vol. 108, n° 1, 1993, p. 1–21, Elsevier.
- [KRA 94] KRAFT R.
Hierarchical B-splines. Citeseer, 1994.
- [LAU 02] LAURSEN T. A.
Computational Contact and Impact Mechanics : Fundamentals of Modelling Interfacial Phenomena in Nonlinear Finite Element Analysis. Springer, Berlin, 2002.
- [LI 98] LI X., WIBERG N.-E.
Implementation and adaptivity of a space-time finite element method for structural dynamics. *Computer Methods in Applied Mechanics and Engineering*, vol. 156, n° 1, 1998, p. 211–229, Elsevier.
- [PAR 90a] PARSONS I., HALL J.
The multigrid method in solid mechanics : part I – algorithm description and behaviour. *International Journal for Numerical Methods in Engineering*, vol. 29, n° 4, 1990, p. 719–737, Wiley Online Library.
- [PAR 90b] PARSONS I., HALL J.
The multigrid method in solid mechanics : part II – practical applications. *International journal for numerical methods in engineering*, vol. 29, n° 4, 1990, p. 739–753, Wiley Online Library.
- [PIE 12] PIEGL L., TILLER W.
The NURBS book. Springer Science & Business Media, 2012.
- [RAN 08] RANNOU J.
Prise en compte d'effets d'échelle en mécanique de la rupture tridimensionnelle par une approche X-FEM multigrille localisée non-linéaire. Thèse de doctorat, Institut National des Sciences Appliquées de Lyon, Octobre 2008.
- [RAN 09] RANNOU J., GRAVOUIL A., BAIETTO-DUBOURG M.
A local multigrid X-FEM strategy for 3-D crack propagation. *International Journal for Numerical Methods in Engineering*, vol. 77, n° 4, 2009, p. 581–600, Wiley Online Library.
- [ROG 00] ROGERS D. F.
An introduction to NURBS : with historical perspective. Elsevier, 2000.
- [RUG 87] RUGE J., STÜBEN K.
Algebraic multigrid. *Multigrid methods*, vol. 3, 1987, p. 73–130, SIAM, Philadelphia.
- [SCO 12] SCOTT M., LI X., SEDERBERG T., HUGHES T.
Local refinement of analysis-suitable T-splines. *Computer Methods in Applied Mechanics and Engineering*, vol. 213, 2012, p. 206–222, Elsevier.
- [SED 03] SEDERBERG T. W., ZHENG J., BAKENOV A., NASRI A.
T-splines and T-NURCCs. *ACM transactions on graphics (TOG)*, vol. 22 ACM, 2003, p. 477–484.

- [SMO 88] SMOLINSKI P., BELYTSCHKO T., NEAL M.
Multi-time-step integration using nodal partitioning. *International Journal for Numerical Methods in Engineering*, vol. 26, n° 2, 1988, p. 349–359, Wiley Online Library.
- [SMO 96] SMOLINSKI P., SLEITH S., BELYTSCHKO T.
Stability of an explicit multi-time step integration algorithm for linear structural dynamics equations. *Computational Mechanics*, vol. 18, n° 3, 1996, p. 236–244, Springer.
- [SMO 98] SMOLINSKI P., WU Y.-S.
Stability of explicit subcycling time integration with linear interpolation for first-order finite element semidiscretizations. *Computer methods in applied mechanics and engineering*, vol. 151, n° 3, 1998, p. 311–324, Elsevier.
- [SMO 00] SMOLINSKI P., PALMER T.
Procedures for multi-time step integration of element-free Galerkin methods for diffusion problems. *Computers & Structures*, vol. 77, n° 2, 2000, p. 171–183, Elsevier.
- [TIE 03] TIE B., AUBRY D., BOULLARD A.
Adaptive computation for elastic wave propagation in plate/shell structures under moving loads. *Revue Européenne des Éléments*, vol. 12, n° 6, 2003, p. 717–736, Taylor & Francis.
- [VEI 13a] BEIRÃO DA VEIGA L., CHO D., PAVARINO L., SCACCHI S.
BDDC preconditioners for isogeometric analysis. *Mathematical Models and Methods in Applied Sciences*, vol. 23, n° 06, 2013, p. 1099–1142, World Scientific.
- [VEI 13b] DA VEIGA L. B., CHO D., PAVARINO L., SCACCHI S.
Isogeometric Schwarz preconditioners for linear elasticity systems. *Computer Methods in Applied Mechanics and Engineering*, vol. 253, 2013, p. 439–454, Elsevier.
- [VEN 00] VENNER C. H., LUBRECHT A. A.
Multi-level methods in lubrication, vol. 37. Elsevier, 2000.
- [VUO 11] VUONG A.-V., GIANNELLI C., JÜTTLER B., SIMEON B.
A hierarchical approach to adaptive local refinement in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, vol. 200, n° 49, 2011, p. 3554–3567, Elsevier.
- [WIB 99] WIBERG N.-E., LI X.
Adaptive finite element procedures for linear and non-linear dynamics. *International journal for numerical methods in engineering*, vol. 46, n° 10, 1999, p. 1781–1802, Wiley Online Library.
- [WIE 99] WIENERS C.
Multigrid methods for Prandtl-Reuss plasticity. *Numerical linear algebra with applications*, vol. 6, n° 6, 1999, p. 457–478, Citeseer.
- [WRI 06] WRIGGERS P., LAURSEN T. A.
Computational contact mechanics, vol. 30167. Springer, 2006.
- [YUE 05] YUE Z., ROBBINS D.
Adaptive superposition of finite element meshes in elastodynamic problems. *Internation-*

tional journal for numerical methods in engineering, vol. 63, n° 11, 2005, p. 1604–1635, Wiley Online Library.

- [ZHA 07] ZHANG Y., BAZILEVS Y., GOSWAMI S., BAJAJ C. L., HUGHES T. J.
Patient-specific vascular NURBS modeling for isogeometric analysis of blood flow.
Computer methods in applied mechanics and engineering, vol. 196, n° 29, 2007,
p. 2943–2959, Elsevier.

FOLIO ADMINISTRATIF

THÈSE SOUTENUE DEVANT L'INSTITUT NATIONAL DES SCIENCES APPLIQUÉES DE LYON

NOM : Chemin

DATE de SOUTENANCE : 9 novembre 2015

Prénoms : Alexandre, Paul, Philippe

TITRE : Analyse isogéométrique multiéchelle à précision contrôlée en mécanique des structures.
– Applications à la statique et à la dynamique transitoire –

NATURE : Doctorat

Numéro d'ordre : 2015-ISAL-0098

École doctorale : MEGA

Spécialité : Mécanique - Génie Mécanique - Génie Civil

Cote B.I.U. - Lyon : T 50/210/19 / et bis

CLASSE :

RÉSUMÉ :

L'analyse isogéométrique pour la résolution de problèmes de la mécanique du solide suscite de vifs intérêts depuis une dizaine d'année. En effet, cette méthode de discrétisation autorise la description exacte des géométries étudiées permettant ainsi de supprimer les erreurs dues à une mauvaise description du domaine spatial étudié. Cependant elle pose un problème théorique de propagation de raffinement lors de la localisation de maillage. Des méthodes pour contourner ce problème ont été proposées dans la littérature mais complexifient grandement la mise en œuvre de cette stratégie de résolution.

Cette thèse propose une stratégie de raffinement localisé adaptatif en espace pour les problèmes de statique et en espace temps pour les problèmes de dynamique transitoire dans le cadre de l'analyse isogéométrique. Pour cela une méthode de localisation pour l'analyse isogéométrique en statique basée sur une résolution multigrille est tout d'abord développée pour des problèmes en deux dimensions. Elle présente l'avantage de contourner la problématique de propagation de raffinement de maillage due à l'analyse isogéométrique tout en étant plus simple à mettre en œuvre que les méthodes déjà existantes. De plus, l'utilisation de l'analyse isogéométrique permet de simplifier les procédures de raffinement lors de l'adaptation de maillage qui peuvent être complexes lors de l'utilisation d'éléments finis classiques.

Une méthode de raffinement adaptatif espace temps basée sur une résolution multigrille est ensuite développée pour des problèmes en une dimension. Une étude sur la structure des opérateurs est proposée afin de choisir un intégrateur temporel adapté. Les performances de cette stratégies sont mises en évidence, puis une modification de la méthode de résolution est proposée afin de diminuer significativement les coûts de calculs associées à cette résolution. La méthode de raffinement adaptatif espace temps est appliquée à quelques exemples académiques afin de valider son bon comportement lors de la localisation.

MOTS-CLÉS : Analyse isogéométrique, dynamique transitoire, raffinement adaptatif espace temps, résolution multigrille.

Laboratoire(s) de recherche : Laboratoire de Mécanique des Contacts et des Structures
UMR CNRS 5259 - INSA de Lyon
18-20 rue des Sciences
69621 Villeurbanne Cedex FRANCE

Directeurs de thèse : Anthony Gravouil et Thomas Elguedj

Président du jury : le président

Composition du jury : AMINE AMMAR
DAVID RYCKELYNCK
JEAN-FRANÇOIS REMACLE
RENÉ DE BORST
MICHEL RAOUS
JEAN-CHARLES PASSIEUX
ANTHONY GRAVOUIL
THOMAS ELGUEDJ