



**HAL**  
open science

# Une approche à deux niveaux pour le calcul de structures haute performance : décomposition – maillage – résolution

Yannis El Gharbi

## ► To cite this version:

Yannis El Gharbi. Une approche à deux niveaux pour le calcul de structures haute performance : décomposition – maillage – résolution. Mécanique des solides [physics.class-ph]. Université Paris-Saclay, 2021. Français. NNT : 2021UPAST001 . tel-03166450

**HAL Id: tel-03166450**

**<https://theses.hal.science/tel-03166450>**

Submitted on 11 Mar 2021

**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.

Une approche à deux niveaux pour  
le calcul de structures haute  
performance : décomposition –  
maillage – résolution

*A two-level approach for high  
performance structural computations:  
substructuring – meshing – resolution*

**Thèse de doctorat de l'Université Paris-Saclay**

École doctorale n° 579, Sciences Mécaniques et Énergétiques,  
MATériaux et Géosciences (SMEMAG)  
Spécialité de doctorat : Mécanique des solides  
Unité de recherche : Université Paris-Saclay, ENS Paris-Saclay, CNRS, LMT -  
Laboratoire de Mécanique et Technologie, 91190, Gif-sur-Yvette, France  
Réfèrent : École Normale Supérieure Paris-Saclay

**Thèse présentée et soutenue à Paris-Saclay, le 19 janvier 2021, par**

**Yannis EL GHARBI**

**Composition du jury :**

<b>Daniel Rixen</b> Professeur, Université Technique de Munich	Président
<b>Pierre Kerfriden</b> Chargé de Recherche, Mines ParisTech <i>Senior Lecturer</i> , Université de Cardiff	Rapporteur & Examineur
<b>Youssef Mesri</b> Chargé de Recherche, Mines ParisTech	Rapporteur & Examineur
<b>Pierre-Alain Boucard</b> Professeur des Universités, ENS Paris-Saclay	Examineur
<b>François-Xavier Roux</b> Maître de Recherche, Onera Palaiseau Professeur Associé, UPMC	Examineur
<b>Robin Bouclier</b> Maître de Conférences, INSA Toulouse	Invité

**Direction de la thèse :**

<b>Pierre Gosselet</b> Chargé de Recherche CNRS, Université de Lille	Directeur de thèse
<b>Christophe Bovet</b> Docteur-Ingénieur, Onera Châtillon	Co-encadrant
<b>Augustin Parret-Fréaud</b> Docteur-Ingénieur, Safran Tech	Co-encadrant



*À mon grand-père,*



## REMERCIEMENTS

Je tiens à commencer ce manuscrit en remerciant toutes les personnes sans lesquelles la réalisation de ce travail n'aurait pas pu être possible. Ce n'est pas la partie la plus intéressante scientifiquement mais ce sera probablement celle qui sera lue par le plus grand nombre de personnes.

Je commence par remercier mon équipe d'encadrement : Pierre, Augustin et Christophe. Merci à tous les trois de m'avoir accompagné pendant toute cette période. Tout d'abord mon Directeur de thèse : Pierre a été pendant ces trois années (ou même quatre en comptant le stage) très disponible et d'un soutien indéfectible, surtout au début où c'était compliqué pour moi de garder la motivation pendant ces phases très (trop?) longues de débogage que j'ai traversées. En plus de cela, il a toujours été très accueillant allant même jusqu'à m'héberger chez lui à Lille pour nous permettre des séances de travail intenses. Christophe et Augustin, représentant le côté industriel de cette thèse, m'ont impressionné par leurs compétences et connaissances techniques et ils m'ont permis de mettre un pied dans le logiciel à portée industrielle qu'est Z-set. Merci à eux pour l'aide qu'ils ont pu me fournir quand je les ai sollicités.

Je voudrais ensuite remercier l'ensemble du jury pour l'intérêt qu'ils ont porté à mon travail. Cet intérêt s'est ressenti lors de la discussion qui a suivi la soutenance et qui a été très intéressante. Je remercie aussi Daniel Rixen de m'avoir fait l'honneur d'être le président ainsi qu'à Pierre Kerfriden et Youssef Mesri d'avoir pris de leur temps pour me lire et pour rapporter mon travail pendant ces fêtes de fin d'année.

Je remercie aussi ceux qui ont assisté à mes quelques présoutenances sans lesquelles ma présentation n'aurait pas eu la même allure : Manu, Oana, Nicolas, Adrien, Yoann, Maxence, Philippe, Léonard, Ronan et Clotilde.

Même si j'ai été assez peu présent physiquement au LMT (en raison de la proximité géographique entre Safran Tech et Versailles et de la présence épisodique de Pierre au labo) ces trois années de thèse sont passées très vite car un bon nombre de personnes rendent ce lieu agréable. Parmi ces personnes, l'équipe de gestionnaires très dynamique se distingue. Elle permet au laboratoire de tourner comme une horloge depuis tant d'années. Et je veux remercier particulièrement Lydia qui m'a été d'une grande aide pour l'organisation de la soutenance malgré un grand nombre d'imprévus dus à l'incertitude de nos nouveaux locaux. Ensuite, j'aimerais remercier Boubou qui était toujours là pour discuter et détendre l'atmosphère ainsi que Frisou dont la présence et l'aura nous manque à tous. Enfin, la majeure partie de mon temps, je l'ai bien évidemment passé avec les doctorants auxquels vont aussi mes remerciements pour les moments d'échanges et de détente, d'abord ceux avec qui j'ai partagé la légendaire salle 211 : Richou, Philippe, Florian, TD, Thibault, Pierre, Pierre-Jean puis ceux de la salle Satie : Maxence et Charles, ainsi que tous les autres qui travaillaient au CdC : pour les plus anciens (Paul, Maxime, Camille, Robin, Snach, Marie, Renaud, Bhimal, Paul-Baptiste, Arturo et ceux que j'oublie) ainsi que tous les plus jeunes. J'espère que les doctorants restants arriveront à s'approprier les nouveaux locaux du LMT et à en faire un endroit aussi convivial que l'était le labo à Cachan. Merci aussi à PAB (devenu récemment directeur du labo au commencement de ces temps obscurs de confinement et dans les nouveaux locaux de

l'ENS) d'avoir accepté de participer à mon jury au pied levé.

Je tiens aussi à remercier les membres du pôle Modélisation et Simulation de Safran Tech et en particulier Nicolas pour son accompagnement « présentatoire » pendant cette fin de thèse. Merci à Christian, pour son accueil dans son équipe CAM (Computational Applied Mathematics). Merci aussi aux collègues du bench de l'ambiance : Sébastien et sa délicatesse, Florian et ses blagues bien senties, Xavier (*a.k.a.* le remplaçant de Perle 😊) et Clément et son hubris. Et je remercie évidemment tous les autres doctorants toujours partants pour des escape games ou des bars à jeux/bières (même si l'organisation pouvait vite devenir chaotique) : Adrien, Maxence, Perle, Moubine, Loïc, Anthony, Camille, Thomas et Martin. Je tenais aussi à rendre hommage à Alexandros qui nous a quitté lors de ce premier confinement et dont j'ai particulièrement apprécié la compagnie et les conseils toujours pertinents.

Merci à tous ceux qui ont assisté à ma soutenance et qui, quasiment du début à la fin, n'ont probablement rien compris de ce que je racontais 😊. Malgré le fait que cela ait été fait en visio, j'ai quand même ressenti votre soutien à distance.

Je voulais aussi remercier ma famille de m'avoir soutenu et particulièrement dans la dernière ligne droite. Je tenais à remercier en particulier mon grand-père qui m'a donné envie de faire des sciences et m'a « enseigné » la curiosité. Papy, même si tu n'es plus là, ta passion continue de vivre en moi!

Enfin, merci Marie d'être toi et d'avoir été là dans tous les moments difficiles depuis tout ce temps (bientôt 9 ans!). Sans toi, je ne serais pas parvenu à écrire ce manuscrit pendant ces longues périodes à la maison. Merci de me supporter même quand mon humeur n'est pas au beau fixe pour me rendre la vie plus facile et agréable. Je t'aime.♡

Version numérique du manuscrit



<http://www.theses.fr/s189493>



<b>Remerciements</b>	<b>iii</b>
<b>Table des matières</b>	<b>vii</b>
<b>Liste des figures</b>	<b>ix</b>
<b>Liste des tableaux</b>	<b>xiii</b>
<b>Introduction</b>	<b>1</b>
<b>1 État de l'art</b>	<b>9</b>
1.1 Méthodes de décomposition de domaine sans recouvrement . . . . .	11
1.1.1 Notations générales pour la Décomposition de Domaine . . . . .	12
1.1.2 Formulation de problèmes condensés à l'interface . . . . .	16
1.1.3 Résolution des approches primale et duale . . . . .	24
1.2 Pré-traitement d'une simulation parallèle . . . . .	34
1.2.1 Stratégies de partitionnement . . . . .	34
1.2.2 Stratégies de maillage et de remaillage parallèle . . . . .	39
1.3 Méthodes de déformation de maillage . . . . .	47
1.3.1 Techniques de projection . . . . .	48
1.3.2 Déformation de maillage . . . . .	51
1.4 Conclusion partielle . . . . .	63
1.5 Références . . . . .	63
<b>2 Méthode de sous-structuration à deux niveaux</b>	<b>77</b>
2.1 Philosophie générale de la méthode à deux niveaux . . . . .	79
2.1.1 Maillage grossier . . . . .	80
2.1.2 Décomposition séquentielle . . . . .	81
2.1.3 Raffinement local . . . . .	83
2.2 Projection des différentes entités constitutives . . . . .	85
2.2.1 Projection des arêtes physiques $\mathcal{E}_{r,\phi}^{(s)}$ sur $\mathcal{E}_{\phi}^{(s)}$ . . . . .	85
2.2.2 Projection des arêtes de la frontière restantes . . . . .	86
2.2.3 Projection des surfaces . . . . .	86
2.3 Méthode de déformation de maillage hiérarchique . . . . .	87
2.3.1 Limitations de la méthode RBF . . . . .	87
2.3.2 Résolution par un solveur itératif . . . . .	90
2.3.3 Interpolation RBF avec adaptation hiérarchique de rayon . . . . .	93
2.4 Résultats numériques . . . . .	95
2.4.1 Influence de la décomposition . . . . .	96

2.4.2	Exemple homogène : Aube de turbine simplifiée . . . . .	97
2.4.3	Exemple hétérogène : volume simplifié de propergol solide . . . . .	105
2.5	Conclusion partielle . . . . .	108
2.6	Références . . . . .	109
<b>3</b>	<b>Impédance d'interface à deux niveaux</b>	<b>113</b>
3.1	Notion d'impédance d'interface optimale . . . . .	114
3.2	Approximation de l'impédance optimale par une approche à deux niveaux . . . . .	116
3.2.1	Techniques d'évaluation de l'impédance d'interface . . . . .	116
3.2.2	Approximation à deux niveaux . . . . .	119
3.2.3	Résultats numériques . . . . .	125
3.3	Formulations mixtes avancées . . . . .	127
3.3.1	FETI-2LM généralisée et introduction d'une seconde impédance d'interface . . . . .	127
3.3.2	Minimisation des sauts à l'interface . . . . .	131
3.3.3	Ajout d'une contrainte additionnelle . . . . .	132
3.4	Conclusion partielle . . . . .	134
3.5	Références . . . . .	134
	<b>Conclusions et Perspectives</b>	<b>139</b>
<b>A</b>	<b>Annexes</b>	<b>I</b>
A.1	Solveurs itératifs de Krylov . . . . .	I
A.1.1	Cas général . . . . .	I
A.1.2	Gradient Conjugué . . . . .	I
A.1.3	ORTHOMIN . . . . .	V
A.2	Étude paramétrique de la méthode RBF adaptative de la section 2.3.3 . . . . .	VII
A.3	Références . . . . .	X
<b>B</b>	<b>Liste des acronymes</b>	<b>XIII</b>
<b>C</b>	<b>Glossaire</b>	<b>XV</b>
<b>D</b>	<b>Notations</b>	<b>XVII</b>

1	Deux dernières familles de turboréacteurs pour l'aviation civile et commerciale développés par Safran sans leurs nacelles respectives (l'échelle entre les deux moteurs est à peu près respectée). . . . .	1
2	Pièces en composite tissé tridimensionnel équipant le moteur LEAP de Safran . . .	2
3	Mésostructure de composites tissés tridimensionnels . . . . .	3
4	Stratégie classique de préparation des données en vue d'une simulation parallèle par décomposition de domaine . . . . .	4
5	Stratégie envisagée dans ce manuscrit . . . . .	5
1.1	Domaine $\Omega$ après discrétisation . . . . .	12
1.2	Différents type de décomposition et définition des interfaces $\Gamma$ , $\Gamma^{(s)}$ et $\Gamma^{(i,j)}$ . . . . .	13
1.3	Différentes notations locales, de l'interface et opérateurs de trace et d'assemblage . . . . .	14
1.4	Types de découpages hétérogènes rencontrés dans un contexte sous-structuré . . . . .	29
1.5	Influence de la répartition des hétérogénéités dans une sous-structuration . . . . .	30
1.6	Sous-domaines avec des interfaces fractales . . . . .	31
1.7	Différents problèmes résolus dans le tableau 1.2 . . . . .	32
1.8	Partitionnement en 12 sous-domaines d'un maillage bidimensionnel issu du logiciel de partitionnement de graphe Metis . . . . .	35
1.9	Décompositions d'un maillage multigrille en 16 sous-domaines obtenus avec Metis et avec un algorithme à bulle. . . . .	37
1.10	Représentation schématique de l'algorithme de partitionnement multi-niveaux pour 4 partitions . . . . .	37
1.11	Exemple de l'étape d'expansion lors du passage du niveau $\lambda$ au niveau $\lambda - 1$ . . . . .	38
1.12	Illustration d'une triangulation de Delaunay . . . . .	39
1.13	Insertion d'un sommet par l'algorithme de Bowyer-Watson. . . . .	40
1.14	Première itération d'une méthode AFT . . . . .	40
1.15	Différents types de subdivisions d'un tétraèdre . . . . .	40
1.16	Génération de maillage à l'aide de la méthode SPH . . . . .	41
1.17	Insertion d'un nœud sur un cavité à l'interface entre 3 sous-domaines . . . . .	42
1.18	Illustration de la méthode de maillage par SPH en parallèle . . . . .	42
1.19	Stratégie de repartitionnement - remaillage . . . . .	43
1.20	Maillage obtenu avec la méthode de Laemmer . . . . .	44
1.21	Illustration de la méthode de maillage parallèle par partitionnement <i>Medial Axis</i> . . . . .	44
1.22	Principe général des méthodes utilisant un maillage grossier . . . . .	45
1.23	Décomposition d'une sphère par le méthode d'Ito et coll. . . . .	46
1.24	Remaillage parallèle par coloriage . . . . .	46
1.25	Remaillage parallèle par remaillage de l'interface sur le processus maître . . . . .	47
1.26	Différents niveaux de partitionnement d'un cube . . . . .	47

1.27	Méthode de <i>morphing</i> de maillage de coque . . . . .	48
1.28	Exemple de trois nœuds reliés au même quatrième nœud à l'aide de ressorts de traction/compression. . . . .	52
1.29	Fonction utilisée par la méthode de l'amortissement algébrique . . . . .	53
1.30	Tracé des différentes fonctions <i>RBF</i> . . . . .	56
1.31	Problème bidimensionnel de référence pour le test des fonction <i>RBF</i> . . . . .	57
1.32	Différents maillages testés . . . . .	58
1.33	Qualité <i>size-shape</i> du plus mauvais élément pour le maillage non-structuré triangulaire. . . . .	60
1.34	Qualité du plus mauvais élément pour le maillage non-structuré quadrangulaire. . . . .	61
1.35	Qualité <i>size-shape</i> du plus mauvais élément pour le maillage structuré triangulaire. . . . .	61
1.36	Qualité du plus mauvais élément pour le maillage structuré quadrangulaire. . . . .	62
2.1	Illustration des étapes parallèles de l'algorithme 2.1 appliquées à un sous-domaine inclus dans une sphère de la géométrie. . . . .	79
2.2	Définition des différents ensembles géométriques, chaque couleur représente une surface physique différente $\mathcal{F}_{c,\phi}^f$ . . . . .	82
2.3	Calcul de la normal d'une arête partagée entre deux éléments grossiers appartenant à des sous-domaines voisins . . . . .	83
2.4	Illustration du <i>h</i> -raffinement et du <i>p</i> -raffinement sur un élément unidimensionnel. . . . .	84
2.5	Schémas de <i>h</i> -raffinement utilisés . . . . .	85
2.6	Projection de l'arête $e_r \in \mathcal{E}_{r,\phi}$ (en traits pointillés noirs) sur l'arête correspondante $e_\phi \in \mathcal{E}_\phi$ (en trait plein bleu), $\ell_r$ et $\ell_\phi$ sont les abscisses curvilignes normalisées. Les lignes vertes correspondent aux segments où la condition $\ell_{r,k} \in [\ell_{\phi,K}, \ell_{\phi,K+1}]$ est satisfaite pour chaque nœud $\mathbf{x}_{r,m}$ de $e_r$ . . . . .	86
2.7	Cube unité . . . . .	88
2.8	Influence du nombre de nœuds de bord sur l'efficacité de la méthode de déformation par <i>RBF</i> pour un rayon d'influence $r = 0.5$ . . . . .	88
2.9	Remplissage de la matrice <i>RBF</i> $\Psi$ , pour un rayon d'influence $r = 0.5$ et un niveau de raffinement croissant . . . . .	89
2.10	Influence du rayon d'influence sur l'efficacité des <i>RBF</i> pour $n_b = 3\ 458$ . . . . .	89
2.11	Densité de la matrice <i>RBF</i> $\Psi$ , pour une taille de matrice fixée à $n_b = 3\ 458$ et différents rayons d'influence $r$ . . . . .	90
2.12	Pire qualité du maillage à $n_b = 13\ 826$ nœuds de bord pour différent critères d'arrêt du <i>CG</i> . . . . .	91
2.13	Comparaison de différentes stratégies de résolution du système (2.1) pour différentes tailles . . . . .	91
2.14	Influence du paramètre $\epsilon$ sur la qualité du préconditionneur <i>ILU</i> (0) d'un <i>PCG</i> , pour un problème de taille $n_b = 13\ 826$ . . . . .	92
2.15	Influence de $\epsilon$ sur l'efficacité du préconditionnement pour $n_b = 13\ 826$ nœuds pour un seuil de convergence $\eta = 10^{-5}$ . . . . .	92
2.16	Influence du paramètre de troncature $\mu$ sur la qualité du préconditionneur de $\Psi$ tronqué, pour un problème de taille $n_b = 3\ 458$ lors d'une résolution par <i>PCG</i> . . . . .	92
2.17	Influence de $\mu$ sur l'efficacité de solveurs itératifs préconditionnés pour $n_b = 3\ 458$ pour arriver à faire diminuer le résidu d'un facteur $10^5$ . . . . .	93
2.18	Illustration de la méthode <i>RBF</i> hiérarchique . . . . .	94
2.19	Influence du nombre de nœuds de bord sur l'efficacité des <i>RBF</i> pour un rayon d'influence variable avec différentes méthodes . . . . .	95
2.20	Motif initial du composite stratifié à 545 025 nœuds et 524 288 éléments et décompositions selon les trois types étudiés. . . . .	96

2.21	Structure en composite stratifié à 8 plis (a–e) et 9 plis (f) : influence du ratio d’hétérogénéité $\rho_E$ sur la convergence du solveur <i>FETI</i> pour les trois types de décomposition et un nombre croissant de sous-domaines (le fond gris de la figure (f) permet de distinguer le cas à 9 plis des cas à 8 plis). . . . .	98
2.22	Structure en composite stratifié à 8 plis : influence du ratio d’hétérogénéité $\rho_E$ sur le conditionnement $\kappa$ de l’opérateur <i>FETI</i> préconditionné pour les trois types de décomposition et un nombre de sous-domaines qui augmente. . . . .	99
2.23	Données initiales de la méthode . . . . .	100
2.24	Exemples de décompositions <i>Two-level</i> de l’aube de turbine . . . . .	101
2.25	Exemples de décompositions <i>Graph-based</i> de l’aube de turbine . . . . .	101
2.26	Temps de réalisation de la phase de déformation de maillage volumique dans chaque sous-domaine pour différentes méthodes <i>RBF</i> . . . . .	102
2.27	Temps du processus de maillage en fonction du nombre de sous-domaines pour un problème global de taille constante (approximativement 1.7 million de nœud) . . . . .	102
2.28	Temps de la génération de maillage et du partitionnement en séquentiel de l’aube avec approximativement 1.6 million de nœuds . . . . .	103
2.29	<i>Speedup</i> du processus de génération de maillage parallèle pour l’exemple de l’aube . . . . .	104
2.30	Nombre d’itérations de la méthode <i>FETI</i> pour une convergence relative de $10^{-6}$ pour le cas de l’aube de turbine et pour un nombre variable de sous-domaines . . . . .	105
2.31	Volume de propergol solide contenant $2 \times 2 \times 2$ inclusions, les différentes couleurs représentent les différentes couleurs; les éléments transparents représentent la matrice et les autres sont dans les inclusions . . . . .	105
2.32	Convergence de l’exemple du propergol solide pour une taille constante de problèmes locaux . . . . .	106
2.33	Temps de génération de maillage du propergol solide en fonction du nombre de sous-domaines pour un problème local de taille constante ( <i>extensibilité faible</i> ). . . . .	107
2.34	Temps de la génération de maillage pour le propergol solide pour une taille constante du problème global ( <i>extensibilité forte</i> , 34.6 millions de <i>DDLs</i> ) . . . . .	108
3.1	Discretisation d’une poutre 3D découpée en 4 sous-domaines cubiques soumise à un chargement de flexion à son extrémité . . . . .	115
3.2	Évolution de la solution au cours des itérations (gauche à droite, haut en bas). Le champ tracé est le champ en mm qui est dans la même direction que le déplacement imposé. . . . .	115
3.3	Convergence d’un solveur <i>FETI-2LM</i> lorsque $\mathbf{Q}_b^{(s)} = \mathbf{S}_p^{(\bar{s})}$ sur le cas de la figure 3.2 . . . . .	116
3.4	Impédance équivalente à deux ressorts en série de <i>NEGRELLO et coll. [2018]</i> . . . . .	118
3.5	Sous-structuration hiérarchique d’une poutre 3D en 4 sous-domaines avec son maillage grossier associé et le chargement considéré . . . . .	119
3.6	Distribution des données pour le sous-domaine $\Omega^{(3)}$ . . . . .	119
3.7	Fonctions de forme d’un élément unidimensionnel grossier (au dessus) auquel deux niveaux de raffinement hiérarchiques ont été appliqués (en dessous) . . . . .	120
3.8	Déformée associée à la solution initiale $\boldsymbol{\mu}_{b,0}^{(s)}$ pour $\mathcal{M}^{(s,\bar{s})} = \mathcal{J}^{(s,\bar{s})}$ et $\mathbf{Q}_b^{(s)} = \mathbf{K}_{bb}^{(\bar{s})}$ . . . . .	121
3.9	Déformée associée à la solution initiale $\boldsymbol{\mu}_{b,0}^{(s)}$ pour $\mathcal{M}^{(s,\bar{s})} = \mathbf{M}_T \mathcal{J}^{(s,\bar{s})} \mathbf{M}_{Gr}^{-1}$ et $\mathbf{Q}_b^{(s)} = \mathbf{K}_{bb}^{(\bar{s})}$ . . . . .	124
3.10	Exemple d’application avec un niveau de raffinement entre les maillages fins et grossiers . . . . .	125
3.11	Nombre d’itérations à convergence de la méthode mixte avec un solveur <i>GMRes</i> et différentes expressions pour l’impédance d’interface. . . . .	126
3.12	Solution selon la direction <i>y</i> , tracée pour l’initialisation sur les maillages fins et grossiers du sous-domaine numéro 3 (celui en jaune sur la figure 3.10), pour différentes impédances $\mathbf{Q}_b^{(s)}$ . . . . .	126

3.13	Solution selon la direction $y$ , tracée pour l'initialisation sur les maillages fins de tous les sous-domaines, pour différentes impédances $\mathbf{Q}_b^{(s)}$ . La solution représentée en filaire est la solution de référence. . . . .	127
3.14	Cube homogène constitué de quatre sous-domaines auquel un chargement de traction est appliqué. La figure de gauche représente la structure non déformée dont les couleurs permettent de distinguer les différents sous-domaines. La figure de droite représente la solution de référence déformée. . . . .	131
3.15	Valeurs propres de l'opérateur mixte . . . . .	132
A.1	Influence de $\alpha$ et $\beta$ sur le temps de résolution de (1.28) par un solveur direct et la qualité du maillage (1/4) . . . . .	VII
A.1	Influence de $\alpha$ et $\beta$ sur le temps de résolution de (1.28) par un solveur direct et la qualité du maillage (2/4) . . . . .	VIII
A.1	Influence de $\alpha$ et $\beta$ sur le temps de résolution de (1.28) par un solveur direct et la qualité du maillage (3/4) . . . . .	IX
A.1	Influence de $\alpha$ et $\beta$ sur le temps de résolution de (1.28) par un solveur direct et la qualité du maillage (4/4) . . . . .	X
A.2	Performances de la méthode pour des valeurs plus importantes de $\beta$ pour $\alpha = 0.5$ . . . . .	X

1.1	Nombre d'itérations nécessaires pour faire diminuer le résidu <i>FETI</i> d'un rapport $10^6$ pour des problèmes homogènes et hétérogènes (rapport de $10^5$ entre les modules de Young des deux constituants) représentés sur la figure 1.5 . . . . .	30
1.2	Évolution du nombre d'itérations à convergence pour une poutre homogène avec l'élanement variable . . . . .	31
1.3	Fonctions de base radiales dont la représentation est donnée sur l'intervalle $[0, 2]$ sur la figure 1.30 . . . . .	55
1.4	Avantages, inconvénients et complexité algorithmique des techniques de déformation de maillage . . . . .	57
2.1	Temps du processus de maillage de l'aube pour un nombre croissant $N_s$ de sous-domaines. La première colonne « <i>Full Seq.</i> » est le temps de génération de maillage et de partitionnement par un processus classique à partir de la géométrie initiale .	103
2.2	Nombre d'itérations pour atteindre la convergence pour l'exemple du propergol solide . . . . .	107
2.3	Temps détaillés du processus de génération de maillage pour l'exemple du propergol solide pour une taille constante de problème local . . . . .	107
2.4	Temps détaillés de la génération de maillage pour le propergol solide pour une taille constante du problème global ( <i>extensibilité forte</i> , 34.6 millions de DDLs) . .	108
3.1	Historique de convergence du cas de la figure 3.2 . . . . .	116



## LISTE DES ALGORITHMES

1.1	Méthode <i>BDD</i> couplée avec un <i>CG</i> . . . . .	28
1.2	Méthode <i>FETI</i> couplée avec un <i>CG</i> . . . . .	29
1.3	<i>CG</i> de la méthode <i>S-FETI</i> . . . . .	33
1.4	Intersection rayon/triangle . . . . .	51
2.1	Méthode de sous-structuration à deux niveaux . . . . .	81
A.1	Gradient Conjugué . . . . .	II
A.2	Gradient Conjugué réorthogonalisé . . . . .	II
A.3	Gradient Conjugué préconditionné avec réorthogonalisation . . . . .	III
A.4	Gradient Conjugué contraint avec réorthogonalisation . . . . .	IV
A.5	Gradient Conjugué augmenté avec réorthogonalisation . . . . .	V
A.6	ORTHOMIN . . . . .	VI



L'industrie aéronautique est confrontée à de fortes contraintes économiques et environnementales qui poussent les entreprises à réduire la consommation de leurs moteurs ainsi que leurs rejets d'agents polluants dans l'atmosphère. Le groupe Safran (et notamment la société *Safran Aircraft Engines*), à travers l'évolution de ses moteurs depuis la fin des années 1970, a contribué à constamment diminuer leur consommation en carburant ainsi que leurs émissions de gaz à effet de serre ( $\text{CO}_2$ ,  $\text{NO}_x$ , ...). Les principaux leviers que les ingénieurs de Safran ont pu utiliser sont la réduction des masses embarquées ainsi que l'augmentation des performances (rendement, poussée, ...). Pour répondre à ces deux attentes, la conception de la nouvelle génération de moteurs Safran intègre de plus en plus de composants en matériaux composites. À titre d'exemple, le moteur LEAP illustré sur la figure 1b en fait usage dans plusieurs pièces de structure.

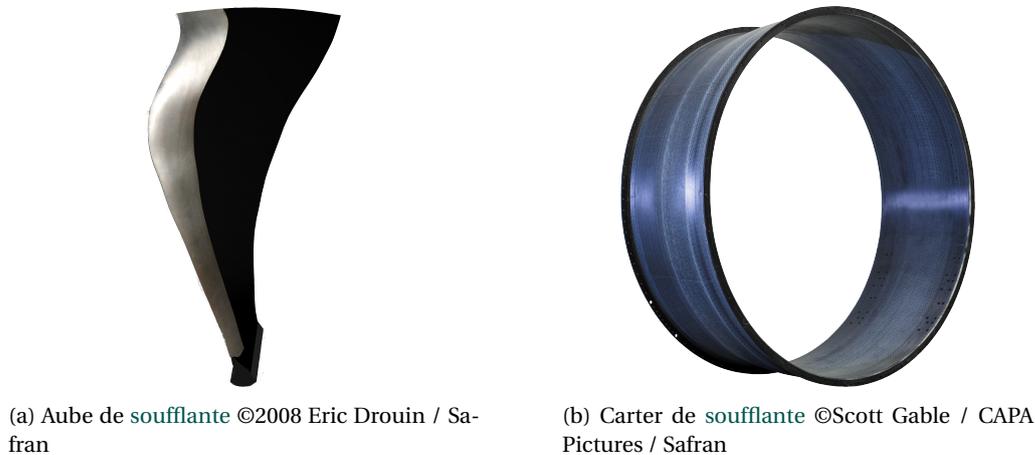


FIGURE 1 – Deux dernières familles de turbo-réacteurs pour l'aviation civile et commerciale développés par Safran sans leurs nacelles respectives (l'échelle entre les deux moteurs est à peu près respectée).

En effet, le passage de la génération précédente (*i.e.* la génération CFM56 illustrée par la figure 1a) à la génération actuelle s'est accompagné d'une révolution dans la conception du moteur. Les matériaux composites tissés tridimensionnels sont nouvellement intégrés dans différentes pièces de structures en remplacement de pièces métalliques. Les principaux éléments bénéficiant de ce nouveau matériau sont les aubes de soufflantes, ainsi que leur carter associé, présentés sur la figure 2. Pour une même pièce, celle constituée de matériaux composites peut présenter des gains de masse allant de 30% à 40% par rapport à son homologue métallique. Ainsi, il a été possible d'augmenter le diamètre de la soufflante par l'utilisation de composites à matrice organique, permettant d'augmenter le rendement du moteur, en passant d'un taux de dilution<sup>1</sup> de 6.6 : 1 à un taux de 11 : 1. L'augmentation des performances passe aussi par une augmentation des tem-

1. Rapport entre la masse d'air passant dans le flux secondaire et le flux primaire

pératures de fonctionnement dans les zones chaudes (chambre de combustion et turbine haute pression) qui atteignent des températures allant jusqu'à 1400 °C qui peuvent être supportées notamment par des matériaux composites à matrice céramique ou encore des alliages hautes performances refroidis par des perforations multiples.



(a) Aube de soufflante ©2008 Eric Drouin / Safran

(b) Carter de soufflante ©Scott Gable / CAPA Pictures / Safran

FIGURE 2 – Pièces en composite tissé tridimensionnel équipant le moteur LEAP de Safran

Cette nouvelle génération de moteurs est dotée de quelques composants conçus en matériaux composites tissés tridimensionnels tels que les aubes de soufflante (figure 2a), le carter de soufflante (figure 2b) ou encore l'anneau de turbine haute pression. Cette évolution du moteur a permis une diminution de 15% en consommation de carburant ainsi qu'un gain de masse de 450 kg et une diminution de 15% des émissions de CO<sub>2</sub> et 50% de NO<sub>x</sub>. Concernant les aubes de soufflante, qui sont à l'origine d'une portion significative de l'amélioration des performances, elles offrent une forme optimisée grâce à une conception s'appuyant sur des études numériques nécessitant de nombreuses simulations, évitant ainsi de réaliser un grand nombre d'essais coûteux.

Ce type de matériaux composites tissés tridimensionnels offre une large gamme de comportements macroscopiques qui dépendent fortement des caractéristiques du tissage choisi. Ainsi, lors des phases de conception, il serait intéressant de pouvoir prédire, à partir de la définition du tissage donnée par le concepteur, la réponse macroscopique de la structure à des chargements sévères qu'elle sera susceptible de subir tout au long de sa vie. C'est pourquoi le *virtual testing* tend à se développer dans l'industrie pour éviter le lancement de campagnes d'essais longues et onéreuses. Dans le secteur de l'aéronautique, il ne permet pas encore de pouvoir certifier des pièces aux yeux des autorités. Cependant dans le cas de dérogations de certaines pièces qui présenteraient des défauts, la simulation numérique peut être utilisée pour permettre la mise en service de celles-ci. Par ailleurs, la discrétisation d'une pièce telle que l'aube représentée sur la figure 2a, en présence d'une description explicite de tous les détails géométriques à l'échelle de la *mésosstructure*, conduit à des modèles numériques précis mais comportant un grand nombre d'inconnues (ou de degrés de liberté (DDLs)) dont le calcul est hors de portée des pratiques actuelles.

Deux types de stratégies permettent de réaliser la simulation de telles pièces. Les premières exploitent l'homogénéisation numérique par homogénéisation asymptotique initiée par exemple par SANCHEZ-PALENCIA [1974] ou encore PAPANICOLAU et coll. [1978] lorsque l'hypothèse de séparation d'échelle<sup>2</sup> est vérifiée. Plusieurs calculs sont effectués sur un Volume Élémentaire Représentatif (VER) (voir figure 3a) représentant la *mésosstructure* pour définir un comportement effectif moyen qui sera ensuite appliqué à toute la structure discrétisée de manière macroscopique sans prendre en compte les détails de *mésosstructure*. Lorsque cette hypothèse de séparation d'échelle n'est pas vérifiée (comme c'est le cas sur la figure 3), il faut modéliser entièrement toute la structure et réaliser le calcul directement sur un maillage comportant un grand nombre de DDLs ou envisager des approches multi-échelles telles que celles développées par FEYEL et CHABOCHE [2000] ou

2. Les dimensions caractéristiques de la *mésosstructure* sont négligeables devant la taille de la structure

LE et coll. [2020]. D'autres approches, permettant de ne modéliser finement qu'une sous-partie de la structure sont connues sous le nom de zoom structural (ou encore approche global-local) et ont été développées par exemple par BLANCHARD et coll. [2019]; GENDRE et coll. [2009]; WANGERMEZ et coll. [2020].

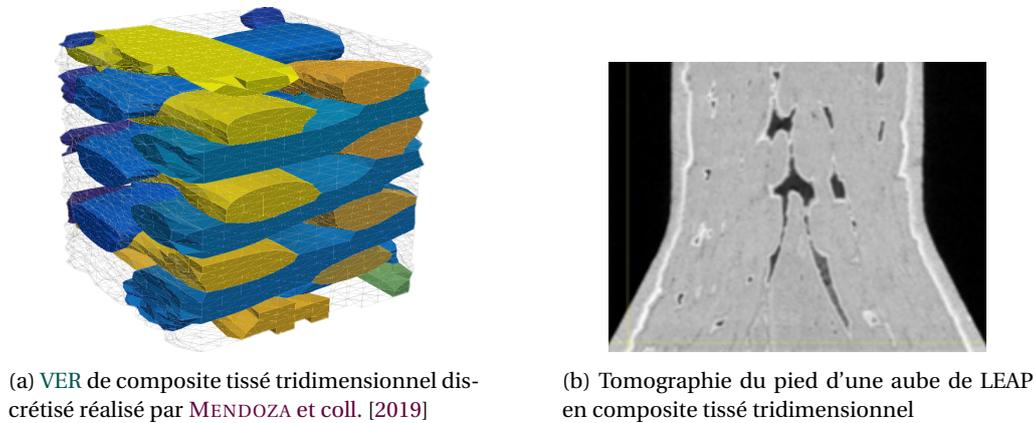


FIGURE 3 – Mésostructure de composites tissés tridimensionnels

Dans tous les cas, il faut être capable de générer le modèle numérique (*i.e.* le maillage) à partir d'une géométrie obtenue par Conception Assistée par Ordinateur (CAO) prenant en compte la structure mésoscopique, à l'échelle des torons. Ainsi, les modèles Éléments Finis (EF) à générer deviennent très volumineux et leur temps de génération et leur volume de stockage deviennent importants. Ensuite, une fois que ce maillage est généré, il faut pouvoir réaliser la simulation EF. Elle peut, elle aussi, devenir très longue et très volumineuse, ce qui n'est pas en accord avec les moyens mis à la disposition des concepteurs ni avec les temps de développement.

Pour pallier le problème de temps, la génération du maillage et le calcul peuvent être exécutés sous forme de différents sous-problèmes sur des machines différentes; on parle alors d'exécution en parallèle. Quant au problème lié à la mémoire, la distribution des données sur plusieurs unités de calcul possédant chacune leur propre système de mémoire locale permet d'éviter cet écueil.

Les machines permettant de réaliser ces tâches sont des machines dont l'architecture matérielle est distribuée. Ces machines, appelées *clusters* ou super-calculateurs, sont des machines regroupant un nombre important d'unités de calcul. Elles se divisent en plusieurs *nœuds de calcul* capables de communiquer avec les autres nœuds à l'aide d'un réseau haut débit. Ces *nœuds de calcul* sont eux-mêmes constitués de processeurs se partageant une même mémoire vive. Ces processeurs peuvent enfin être composés de plusieurs cœurs qui sont des unités de calcul indépendantes. Ainsi, l'accélération de l'exécution d'un programme peut passer simplement par son découpage en différentes instructions sur le grand nombre de cœurs du super-calculateur. Si les instructions sont indépendantes, il est possible de les exécuter simultanément sur chaque cœur pour en diviser le temps d'exécution d'autant plus que le nombre de cœurs disponibles est important.

En raison de cette architecture de calcul, deux modèles de programmation se sont développés pour paralléliser l'exécution d'un programme. Le premier est fondé sur le *multithreading* où les cœurs d'un même nœud se partagent les mêmes données en exécutant des actions sur des variables partagées dans la mémoire vive commune du nœud auquel ils appartiennent : on parle de parallélisme à mémoire partagée, dont l'interface la plus répandue est OpenMP. Le second modèle n'est plus limité au nombre de cœurs sur le nœud considéré mais est un parallélisme défini au niveau du *cluster* permettant d'utiliser le réseau d'interconnexion entre les différents *nœuds de calcul*. Ainsi chaque cœur est une unité de calcul indépendante dont la notion de variable partagée n'existe plus. Pour avoir accès à la valeur d'une variable d'un autre cœur il est nécessaire que les cœurs se synchronisent à l'aide de communications qui sont traitées, par exemple, par la bibliothèque standard de parallélisme *Message Passing Interface (MPI)*. Ainsi, chaque cœur possède

sa propre mémoire locale et on parle de parallélisme à mémoire distribuée.

Ce travail se place dans le cadre du deuxième modèle de programmation où les solveurs de décomposition de domaine sont plus adaptés. Les méthodes de décomposition de domaine permettent de résoudre numériquement et en parallèle de manière itérative des systèmes linéaires représentant par exemple des problèmes mécaniques dont un découpage du domaine de calcul est réalisé au préalable pour permettre sa distribution. Ces méthodes sont efficaces en raison d'un nombre limité de synchronisations nécessaires entre les cœurs et les communications consistent en des échanges limités en volume de données ainsi qu'en nombre de cœurs à mettre en relation. Pour limiter ces échanges, les communications sont en partie réalisées entre voisins uniquement (assemblages de vecteurs) ou entre tous les processus lorsque des quantités plus faibles de données (produits scalaires par exemple) sont mises en jeu.

Les capacités des algorithmes développés dans ce cadre parallèle sont évaluées en termes de passage à l'échelle aussi appelé extensibilité (*scalability* en anglais). On distingue deux types d'extensibilité : l'extensibilité faible et l'extensibilité forte. L'extensibilité forte consiste, à partir d'un problème global fixé, à faire varier le nombre de ressources allouées à sa résolution. Dans le cas optimal, le facteur d'accélération<sup>3</sup> doit augmenter d'autant que le nombre de ressources augmente. L'extensibilité faible consiste, quant à elle, à conserver des sous-problèmes de taille fixée tout en faisant varier le nombre de sous-problèmes totaux, conduisant à un problème global de taille variable. Ainsi, le nombre d'opérations locales reste le même. Dans le cas idéal, le temps d'exécution total du programme doit rester constante quel que soit le nombre de sous-problèmes du problème global.

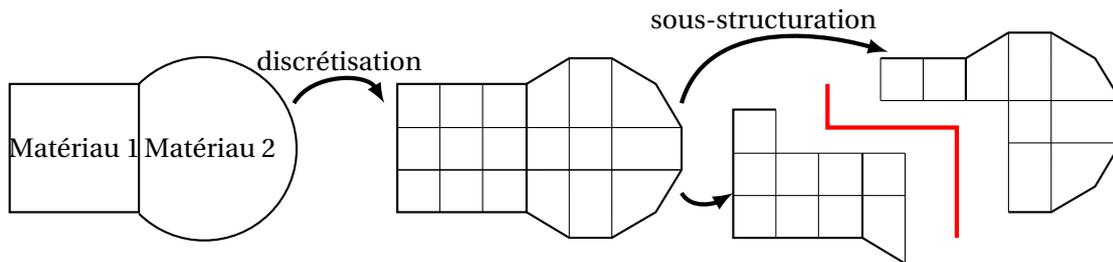


FIGURE 4 – Stratégie classique de préparation des données en vue d'une simulation parallèle par décomposition de domaine

Pour réaliser ces simulations dans un cadre parallèle, il faut obtenir une sous-structuration, c'est-à-dire un ensemble de plusieurs sous-maillages dont l'union représente une discrétisation de la structure globale. Cette sous-structuration est obtenue classiquement par la méthode illustrée par la figure 4. C'est une stratégie, qui, à partir d'une géométrie globale, produit une discrétisation globale de la structure à la finesse de calcul désirée par un procédé de génération de maillage séquentiel tel que les méthodes développées par DELAUNAY [1924]; LÖHNER [1988]. Ensuite, une méthode de partitionnement de maillage permet de découper le maillage en différents sous-maillages par des méthodes de partitionnement de graphe qui ont pu être développées par HENDRICKSON et LELAND [1995]; KARYPIS et KUMAR [1998]; PELLEGRINI et ROMAN [1996] par exemple. Ce partitionnement permet de générer une interface à laquelle sont attachées les données de connectivités entre les sous-domaines. Ces approches de partitionnement permettent de former des sous-domaines qui comportent approximativement le même nombre d'éléments et présentent une interface de taille minimale ou minimisant le volume de communication. Cette approche met en jeu un maillage global qui, lorsque la discrétisation est fine, est constitué d'un grand nombre de DDLs long à générer et volumineux en termes de mémoire (les composites tissés tridimensionnels rentrent dans ce cadre). En sus de la génération du maillage, son partitionnement aussi est onéreux. Ainsi, malgré l'existence de ces méthodes de calcul parallèle et en particulier les méthodes de décomposition de domaine qui sont plutôt robustes, la partie génération

3. Le rapport entre le temps d'exécution séquentiel et le temps d'exécution parallèle

de maillage en parallèle reste encore une étape difficile et critique dont une solution est proposée dans ce manuscrit.

Les sous-structurations issues d'une telle méthode de préparation des données conduisent à former des sous-domaines avec des interfaces irrégulières qui, en plus, peuvent être constitués de matériaux différents. Or la vitesse des méthodes de résolution par décomposition de domaine dépend fortement de la forme et de l'homogénéité des sous-domaines comme ont pu le montrer [GOSSELET et coll. \[2015\]](#); [KLAWONN et coll. \[2008\]](#); [RIXEN et FARHAT \[1997\]](#) et comme ce sera précisé dans le premier chapitre bibliographique. En sachant que la résolution se fait à l'aide de solveurs par décomposition de domaine, la solution proposée dans ce manuscrit va permettre, en plus de la parallélisation, de faciliter la convergence de ces solveurs.

L'objectif de ce manuscrit est de développer une approche de sous-structuration permettant d'améliorer la vitesse de résolution des méthodes de décomposition de domaine en jouant sur les leviers suivants :

- parallélisation de la génération de maillage;
- génération de sous-domaines dont la forme améliore la vitesse de convergence du solveur;
- exploitation de cette sous-structuration pour améliorer la convergence d'un solveur par décomposition de domaine.

Pour atteindre cet objectif, une méthodologie de sous-structuration qui se veut la plus parallèle possible a été envisagée suivant le schéma de la figure 5. Cette stratégie repose schématiquement

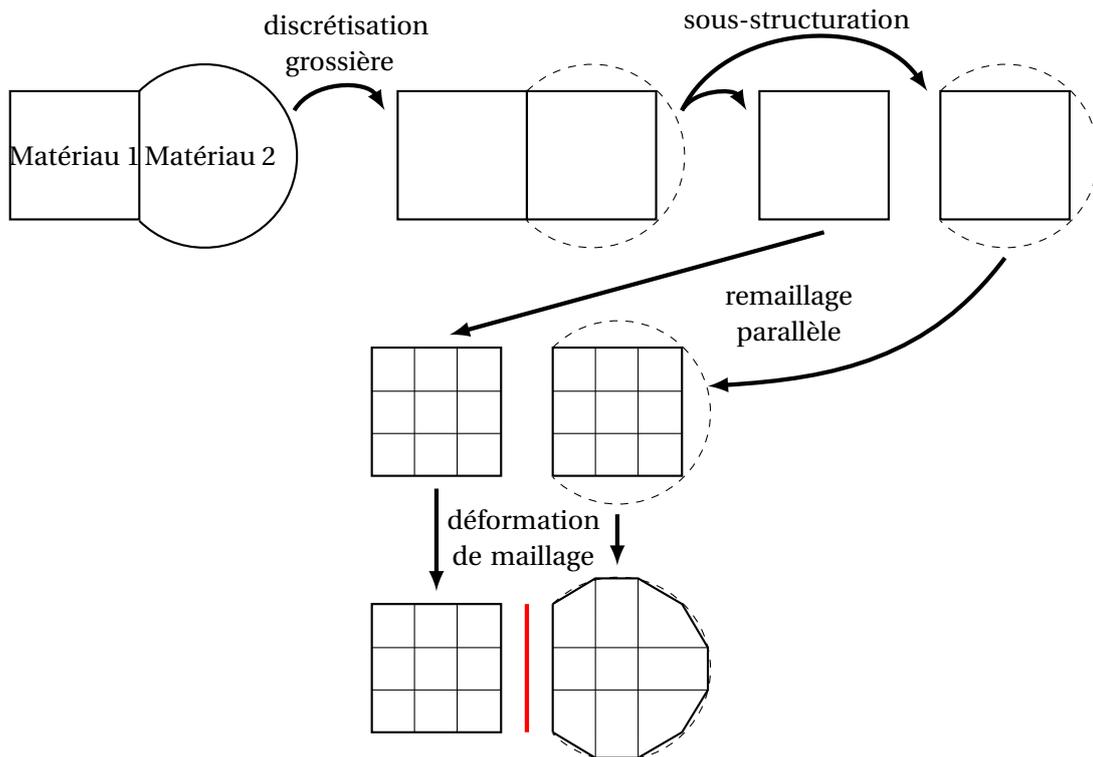


FIGURE 5 – Stratégie envisagée dans ce manuscrit

sur l'inversion des étapes de génération de maillage fin et de sous-structuration. À partir d'une géométrie, un maillage grossier est généré puis décomposé en sous-domaines. Suite à cela, les maillages sont raffinés en parallèle puis déformés pour respecter la géométrie. On parle ici de méthode de sous-structuration à deux niveaux car elle met en jeu un maillage fin et un maillage grossier permettant de définir la sous-structuration. Deux niveaux de maillages sont mis en jeu ici : le maillage fin final ainsi que le maillage grossier initial. Leur présence permet d'envisager l'utilisation du niveau grossier pour enrichir un solveur mixte tel que la méthode *Finite Element Tearing and Interconnecting (FETI) with two Lagrange Multipliers (FETI-2LM)* de [ROUX \[2009\]](#).

Ce manuscrit comporte trois chapitres permettant de réaliser cette sous-structuration et de l'exploiter. Le premier chapitre, essentiellement bibliographique, se penche sur trois thématiques distinctes. Les principes généraux des méthodes de décomposition de domaine sans recouvrement de type Schur sont abordés, puis une description de l'état de l'art lié à la préparation des données dans un contexte parallèle est dressée, et enfin une dernière section est consacrée aux approches de déformation de maillage. Ces trois aspects permettent de mettre en place la technique de sous-structuration décrite sur la figure 5 dans le second chapitre. La phase de déformation de maillage est ensuite développée car elle met en oeuvre, dans ce même chapitre, une approche hiérarchique originale. Les performances parallèles de la méthode y sont évaluées ainsi que les performances de la résolution équipée d'une telle décomposition par un solveur de décomposition de domaine. Finalement, dans le troisième chapitre du manuscrit, la décomposition est exploitée pour l'évaluation d'une approximation du paramètre d'impédance d'interface optimal d'un solveur de décomposition de domaine mixte. Une nouvelle formulation de ce solveur a aussi été développée pour permettre son extensibilité.

## Références

- BLANCHARD, M., O. ALLIX, P. GOSSELET et G. DESMEURE. 2019, «Space/time global/local non-invasive coupling strategy: Application to viscoplastic structures», *Finite Elements in Analysis and Design*, vol. 156, doi : [10.1016/j.finel.2019.01.003](https://doi.org/10.1016/j.finel.2019.01.003), p. 1–12, ISSN 0168-874X. URL <http://www.sciencedirect.com/science/article/pii/S0168874X18303810>. 3
- DELAUNAY, B. 1924, «Sur la sphère vide», dans *Proceedings of the International Mathematical Congress held in Toronto*, vol. 1, Toronto, p. 695–700. 4
- FEYEL, F. et J.-L. CHABOCHE. 2000, «FE2 multiscale approach for modelling the elastoviscoplastic behaviour of long fibre SiC/Ti composite materials», *Computer Methods in Applied Mechanics and Engineering*, vol. 183, n° 3, doi : [10.1016/S0045-7825\(99\)00224-8](https://doi.org/10.1016/S0045-7825(99)00224-8), p. 309–330, ISSN 0045-7825. URL <http://www.sciencedirect.com/science/article/pii/S0045782599002248>. 2
- GENDRE, L., O. ALLIX, P. GOSSELET et F. COMTE. 2009, «Non-intrusive and exact global/local techniques for structural problems with local plasticity», *Computational Mechanics*, vol. 44, n° 2, doi : [10.1007/s00466-009-0372-9](https://doi.org/10.1007/s00466-009-0372-9), p. 233–245, ISSN 1432-0924. URL <https://doi.org/10.1007/s00466-009-0372-9>. 3
- GOSSELET, P., D. J. RIXEN, F.-X. ROUX et N. SPILLANE. 2015, «Simultaneous FETI and block FETI: Robust domain decomposition with multiple search directions», *International Journal for Numerical Methods in Engineering*, vol. 104, n° 10, doi : [10.1002/nme.4946](https://doi.org/10.1002/nme.4946), p. 905–927. 5
- HENDRICKSON, B. et R. LELAND. 1995, «A multilevel algorithm for partitioning graphs», dans *Proceedings of the 1995 ACM/IEEE conference on Supercomputing (CDROM) - Supercomputing '95*, ACM Press, San Diego, California, United States, ISBN 978-0-89791-816-9, p. 28–es, doi : [10.1145/224170.224228](https://doi.org/10.1145/224170.224228). URL <http://portal.acm.org/citation.cfm?doid=224170.224228>. 4
- KARYPIS, G. et V. KUMAR. 1998, «A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs», *SIAM Journal on Scientific Computing*, vol. 20, n° 1, doi : [10.1137/S1064827595287997](https://doi.org/10.1137/S1064827595287997), p. 359–392, ISSN 1064-8275, 1095-7197. URL <http://epubs.siam.org/doi/10.1137/S1064827595287997>. 4
- KLAWONN, A., O. RHEINBACH et O. B. WIDLUND. 2008, «An Analysis of a FETI–DP Algorithm on Irregular Subdomains in the Plane», *SIAM Journal on Numerical Analysis*, vol. 46, n° 5, doi : [10.1137/070688675](https://doi.org/10.1137/070688675), p. 2484–2504, ISSN 0036-1429, 1095-7170. URL <http://epubs.siam.org/doi/10.1137/070688675>. 5

- LE, M. V., J. YVONNET, N. FELD et F. DETREZ. 2020, «The Coarse Mesh Condensation Multiscale Method for parallel computation of heterogeneous linear structures without scale separation», *Computer Methods in Applied Mechanics and Engineering*, vol. 363, doi : [10.1016/j.cma.2020.112877](https://doi.org/10.1016/j.cma.2020.112877), p. 112 877, ISSN 0045-7825. URL <http://www.sciencedirect.com/science/article/pii/S0045782520300591>. 3
- LÖHNER, R. 1988, «Interactive generation of unstructured grids for three dimensional problems», dans *Proceedings of the 2nd Intl. Conference on Numerical Grid Generation in Computational Fluid Mechanics*, Miami Beach, FL, United States, p. 685–698. 4
- MENDOZA, A., J. SCHNEIDER, E. PARRA et S. ROUX. 2019, «The correlation framework: Bridging the gap between modeling and analysis for 3D woven composites», *Composite Structures*, vol. 229, doi : [10.1016/j.compstruct.2019.111468](https://doi.org/10.1016/j.compstruct.2019.111468), p. 111 468, ISSN 0263-8223. URL <http://www.sciencedirect.com/science/article/pii/S0263822318346221>. 3
- PAPANICOLAU, G., A. BENSOUSSAN et J. L. LIONS. 1978, *Asymptotic Analysis for Periodic Structures.*, Elsevier, Burlington, ISBN 978-0-08-087526-2. URL [http://www.123library.org/book\\_details/?id=40707](http://www.123library.org/book_details/?id=40707), oCLC: 476216211. 2
- PELLEGRINI, F. et J. ROMAN. 1996, «Scotch: A software package for static mapping by dual recursive bipartitioning of process and architecture graphs», dans *High-Performance Computing and Networking*, vol. 1067, édité par G. Goos, J. Hartmanis, J. van Leeuwen, H. Liddell, A. Colbrook, B. Hertzberger et P. Sloot, Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN 978-3-540-61142-4 978-3-540-49955-8, p. 493–498, doi : [10.1007/3-540-61142-8\\_588](https://doi.org/10.1007/3-540-61142-8_588). URL [http://link.springer.com/10.1007/3-540-61142-8\\_588](http://link.springer.com/10.1007/3-540-61142-8_588). 4
- RIXEN, D. et C. FARHAT. 1997, «Preconditioning the FETI Method for Problems with Intra- and Inter-Subdomain Coefficient Jumps», dans *Ninth International Conference of Domain Decomposition Methods, 1997*. URL = <http://www.ddm.org/DD9>. 5
- ROUX, F.-X. 2009, «A FETI-2LM Method for Non-Matching Grids», dans *Domain Decomposition Methods in Science and Engineering XVIII*, vol. 70, édité par M. J. Gander, O. Widlund et M. Bercovier, Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN 978-3-642-02676-8 978-3-642-02677-5, p. 121–128, doi : [10.1007/978-3-642-02677-5\\_11](https://doi.org/10.1007/978-3-642-02677-5_11). URL [http://link.springer.com/10.1007/978-3-642-02677-5\\_11](http://link.springer.com/10.1007/978-3-642-02677-5_11), series Title: Lecture Notes in Computational Science and Engineering. 5
- SANCHEZ-PALENCIA, E. 1974, «Comportements local et macroscopique d'un type de milieux physiques heterogenes», *International Journal of Engineering Science*, vol. 12, n° 4, doi : [10.1016/0020-7225\(74\)90062-7](https://doi.org/10.1016/0020-7225(74)90062-7), p. 331–351, ISSN 0020-7225. URL <http://www.sciencedirect.com/science/article/pii/0020722574900627>. 2
- WANGERMEZ, M., O. ALLIX, P.-A. GUIDAULT, O. CIOBANU et C. REY. 2020, «Interface coupling method for the global-local analysis of heterogeneous models: A second-order homogenization-based strategy», *Computer Methods in Applied Mechanics and Engineering*, vol. 365, doi : [10.1016/j.cma.2020.113032](https://doi.org/10.1016/j.cma.2020.113032), p. 113 032, ISSN 0045-7825. URL <http://www.sciencedirect.com/science/article/pii/S0045782520302164>. 3



Ce chapitre a pour objectif de dresser un portrait de l'état de l'art actuel des différentes méthodes sur lesquelles s'appuie cette thèse, qu'elles soient parallèles ou qu'elles permettent les opérations nécessaires au développement de la démarche de la figure 5.

Ainsi, sont présentées les méthodes de décomposition de domaine sans recouvrement qui offrent un cadre naturellement favorable à la parallélisation et la distribution des calculs. Une présentation des différents types de condition d'interface menant aux approches primale, duale et mixte sera faite.

La discrétisation de la géométrie est une étape essentielle de la démarche de calcul d'une structure. Les méthodes parallèles pour la réaliser sont ensuite abordées, qu'elles s'appuient directement sur la géométrie ou sur un maillage déjà existant.

Enfin, les méthodes permettant de déformer un maillage sont abordées. La déformation d'un maillage s'effectue en deux étapes qui définiront les deux aspects bibliographiques évoqués dans la section 1.3. Une première étape consiste à déplacer les points de la surface du maillage alors que la seconde étape consiste à déplacer les nœuds restants (à l'intérieur du volume) en accord avec les déplacements du bord définis à l'étape précédente. Ces méthodes sont nécessaires pour réaliser la phase de déformation de maillage nécessaire à l'obtention du problème sous-structuré à deux niveaux, comme présenté sur la figure 5.

*« La bibliographie se fait après et non avant d'aborder un sujet de recherche. »*

Jean Perrin

<b>1.1 Méthodes de décomposition de domaine sans recouvrement</b>	<b>11</b>
1.1.1 Notations générales pour la Décomposition de Domaine	12
1.1.2 Formulation de problèmes condensés à l'interface	16
1.1.2.1 Équilibre local des sous-domaines	16
1.1.2.2 Approche primale	18
1.1.2.3 Approche duale	18
1.1.2.4 Approche mixte	20
1.1.2.5 Approches hybrides et recondensées	23
1.1.3 Résolution des approches primale et duale	24
1.1.3.1 Préconditionnement	24
1.1.3.2 Problèmes grossiers	26
1.1.3.3 Évolution du conditionnement pour divers problèmes	29
1.1.3.4 Méthodes avancées	31

1.1.3.5	Précisions sur la résolution des formulations mixtes . . . . .	33
<b>1.2</b>	<b>Pré-traitement d'une simulation parallèle . . . . .</b>	<b>34</b>
1.2.1	Stratégies de partitionnement . . . . .	34
1.2.1.1	Approche spectrale . . . . .	36
1.2.1.2	Approches combinatoires . . . . .	36
1.2.1.3	Approche multi-niveaux . . . . .	37
1.2.2	Stratégies de maillage et de remaillage parallèle . . . . .	39
1.2.2.1	Outils de maillage séquentiel . . . . .	39
1.2.2.2	Parallélisation des opérations élémentaires dans les algorithmes séquentiels . . . . .	41
1.2.2.3	« Orchestration » d'algorithmes séquentiels en tant que « boîtes noires » . . . . .	42
<b>1.3</b>	<b>Méthodes de déformation de maillage . . . . .</b>	<b>47</b>
1.3.1	Techniques de projection . . . . .	48
1.3.1.1	Paramétrisation sphérique . . . . .	48
1.3.1.2	Projection explicite le long d'une direction . . . . .	49
1.3.2	Déformation de maillage . . . . .	51
1.3.2.1	Techniques nécessitant une connectivité de maillage . . . . .	51
1.3.2.2	Techniques sans maillage ( <i>meshless</i> ) . . . . .	52
1.3.2.3	Comparaison de ces méthodes . . . . .	56
1.3.2.4	Comparaison des différents types de fonctions <i>RBF</i> . . . . .	56
<b>1.4</b>	<b>Conclusion partielle . . . . .</b>	<b>63</b>
<b>1.5</b>	<b>Références . . . . .</b>	<b>63</b>

## 1.1 Méthodes de décomposition de domaine sans recouvrement

Dans cette section, le point central de la discussion est la résolution numérique d'un système linéaire creux de la forme :

$$\mathbf{K}\mathbf{u} = \mathbf{f}. \quad (1.1)$$

Avec  $\mathbf{K}$  une matrice carrée creuse potentiellement symétrique et de grande taille issue de la discrétisation à une échelle fine, d'un problème de mécanique linéaire des structures (ou de problèmes de thermique linéaire,...),  $\mathbf{f}$  un second membre représentant des chargements imposés et  $\mathbf{u}$  un vecteur d'inconnues. Pour résoudre ces systèmes linéaires de grande taille, plusieurs solveurs sont utilisés :

- les solveurs directs (pivot de Gauss, factorisation  $LU$  ou de Cholesky...),
- les solveurs itératifs (méthodes de Krylov, point fixe...).

Dans ces deux cas, le calcul de l'inverse de  $\mathbf{K}$ , n'est jamais réalisé, car trop coûteux en termes de stockage et de nombre d'opérations algébriques nécessaires même si  $\mathbf{K}$  est un opérateur creux. Les solveurs creux directs comme *MULTifrontal Massively Parallel sparse direct Solver (MUMPS)* AMESTOY et DUFF [2000] ou *PARallel sparse DIrect Solver (PARDISO)* SCHENK et coll. [2001] exploitent le fait que certaines matrices sont simples à inverser : les matrices diagonales, les matrices triangulaires ou encore les matrices orthogonales. Ces solveurs vont donc ramener  $\mathbf{K}$  à un produit de matrices simples à inverser, c'est ce qui est appelé une factorisation de  $\mathbf{K}$ . Ils présentent une très bonne robustesse et donnent une solution en un nombre d'opérations connu ne dépendant que de la taille et de la densité de la matrice. Mais ces solveurs présentent une consommation mémoire élevée liée notamment à un stockage qui augmente très rapidement avec la taille des problèmes malgré une distribution des données sur un grand nombre de nœuds de calcul. En effet, même si  $\mathbf{K}$  est creuse, les matrices de sa factorisation peuvent être beaucoup plus denses. De plus, la parallélisation de la factorisation de  $\mathbf{K}$  n'est pas une tâche aisée et l'*extensibilité* de ces approches ne semble pas optimale. Ces méthodes peuvent afficher de bonnes performances jusqu'à quelques centaines de cœurs mais pas suffisamment au delà.

Les méthodes itératives, décrites entre autres par SAAD [2003] ou encore BARRETT et coll. [1994], quant à elles, utilisent des approximations successives permettant de se rapprocher de la solution du système linéaire (1.1) à chaque itération de l'algorithme. Seules les actions de la matrice sur un vecteur sont nécessaires, ainsi l'assemblage explicite peut être évité. De plus, elles permettent d'exploiter plus facilement le caractère creux des opérateurs et la qualité de la solution peut être contrôlée. Cependant, leur nombre d'itérations pour satisfaire le critère de convergence dépend de plusieurs facteurs : du *conditionnement* de  $\mathbf{K}$ , de l'estimation initiale, du critère d'arrêt et du choix d'un préconditionneur adéquat. Il est en effet possible d'ajouter un préconditionneur au système (1.1) permettant d'améliorer significativement le nombre d'itérations à convergence s'il est choisi judicieusement. Ces méthodes ne concurrencent pas encore les solveurs directs dans les codes de calcul industriels à cause de leur manque de robustesse. Mais leur implémentation devrait se développer pour la résolution de systèmes de grande taille en raison des limitations des architectures mémoires actuelles.

Une troisième classe de solveurs s'efforce de bénéficier de la robustesse des méthodes directes et de l'efficacité des méthodes itératives : ce sont les méthodes de décomposition de domaine. Dans ces méthodes, le problème posé sur  $\Omega$  est découpé en  $N_s$  sous-problèmes posés sur  $N_s$  plus petits domaines  $\Omega^{(s)}$  tels que

$$\Omega = \bigcup_s \bar{\Omega}^{(s)}.$$

Les méthodes de décomposition de domaines se distinguent en deux classes par le fait que les sous-domaines soient recouvrants ou non. La première classe regroupe les méthodes avec recouvrement telles que

$$\bigcap_s \Omega^{(s)} \neq \emptyset,$$

où  $\mathring{\Omega}^{(s)} = \Omega^{(s)} \setminus \partial\Omega^{(s)}$  avec  $\partial$  désignant le bord d'un domaine. Elles ont été introduites historiquement par SCHWARZ [1870] et développées par la suite dans un cadre discret pour la méthode des EF (on peut citer *Multiplicative Schwarz Method* CHAN et MATHEW [1994], *Additive Schwarz Method* DRYJA et WIDLUND [1989] et *Restricted Additive Schwarz Method* CAI et SARKIS [1999]). Ces méthodes permettent de résoudre le système linéaire (1.1) de manière itérative à l'aide de méthodes de point fixe dont la vitesse de convergence est d'autant meilleure que la taille du recouvrement est importante. Toutefois, un large recouvrement conduit aussi à de grandes quantités d'informations à échanger entre les sous-domaines, ce qui peut ajouter un surcoût temporel non négligeable à la résolution. C'est pourquoi des méthodes sans recouvrement, telles que :

$$\bigcap_s \Omega^{(s)} = \emptyset,$$

ont été développées pour limiter les échanges nécessaires dus au recouvrement.

En plus de cette limitation d'échanges, les méthodes sans recouvrement permettent de distinguer clairement des sous-structures qui peuvent s'apparenter à des sous-ensembles physiques. En effet, une structure est souvent composée de plusieurs pièces ou peut avoir des caractéristiques internes locales différentes (comportement matériau, non-linéarités). Il semble donc pertinent d'isoler ces caractéristiques dans des sous-domaines différents pour les traiter de manière isolée. Par la suite, les méthodes de décomposition de domaine sans recouvrement seront le point central du propos.

Parmi les méthodes sans recouvrement, on trouve *Optimized Schwarz* proposée par LIONS [1990] qui étend la philosophie des méthodes de recouvrement à l'aide de conditions de Robin bien choisies ainsi que les méthodes issues des idées de PRZEMIENIECKI [1963] sur la sous-structuration qui, naturellement, ne comporte pas de recouvrement.

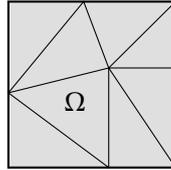


FIGURE 1.1 – Domaine  $\Omega$  après discrétisation

Les méthodes de décomposition de domaine sont aussi bien étudiées d'un point de vue continu. Cela permet de pouvoir considérer d'autres interactions entre sous-domaines et notamment pouvoir résoudre des problèmes où la discrétisation est différente entre les sous-domaines et pouvoir traiter des problèmes multi-physiques, multi-modèles (WANGERMEZ et coll. [2020]) ou encore des problèmes faisant apparaître des non-linéarités (BLANCHARD [2018]; NEGRELLO et coll. [2016]) Dans cette section, à la vue de la grande diversité du contenu bibliographique concernant les méthodes de décomposition de domaine sans recouvrement, les idées et concepts fondamentaux liés aux méthodes les plus classiques sans recouvrement de type Schur seront décrites simplement sans détailler les principes des méthodes de type Schwarz précédemment citées pour résoudre le système linéaire (1.1) où l'opérateur  $\mathbf{K}$  peut être issue de la discrétisation par la méthode des EF d'un problème de mécanique linéaire élastique (matrice de raideur du domaine  $\Omega$  de la figure 1.1) par la méthode des EF,  $\mathbf{f}$  le vecteur des efforts généralisés appliqués à la structure et  $\mathbf{u}$  le déplacement nodal inconnu sur la structure. Cette partie se penchera donc sur les aspects principaux qui seront utiles pour la suite. Les lecteurs qui voudraient de plus amples informations sur ces méthodes pourront se référer à GOSSELET et REY [2006], LE TALLEC [1994] et DOLEAN et coll. [2015].

### 1.1.1 Notations générales pour la Décomposition de Domaine

Dans cette partie, les notations et objets utiles aux méthodes de décomposition de domaine sans recouvrement sont introduits dans le cas d'un domaine  $\Omega$  décomposé en un nombre quelconque de sous-domaines  $N_s$ .

On notera par un exposant  $^{(s)}$  toute grandeur associée au sous-domaine  $\Omega^{(s)}$ . Cela représente une restriction au sous-domaine  $\Omega^{(s)}$  d'une grandeur définie sur  $\Omega$ . Cette sous-structuration du problème conduit à créer des séparations fictives (qui n'ont pas de réalité physique) entre les sous-domaines et qui seront appelés : l'interface. Elle peut être de plusieurs natures :

- l'interface entre les sous-domaines  $\Omega^{(i)}$  et  $\Omega^{(j)}$  notée  $\Gamma^{(i,j)} = \Gamma^{(j,i)} = \partial\Omega^{(i)} \cap \partial\Omega^{(j)}$ ;
- l'interface locale  $\Gamma^{(s)} = \cup_j \Gamma^{(s,j)}$ ;
- l'interface globale  $\Gamma = \cup_s \Gamma^{(s)}$ .

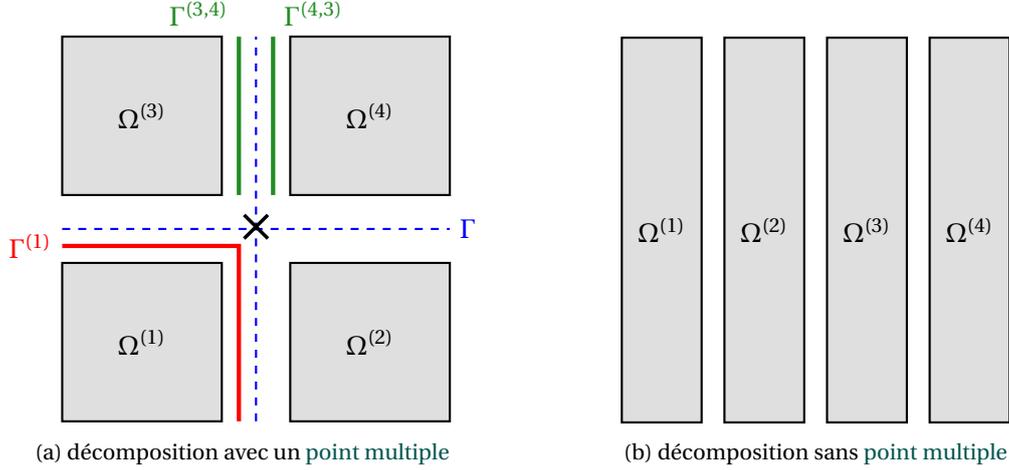


FIGURE 1.2 – Différents type de décomposition et définition des interfaces  $\Gamma$ ,  $\Gamma^{(s)}$  et  $\Gamma^{(i,j)}$

De plus, lorsque le graphe de connectivités<sup>1</sup> entre les sous-domaines peut former des cycles, la décomposition d'une structure en différents sous-domaines peut très souvent conduire à l'apparition de **points multiples** (voir figure 1.2) : ce sont des nœuds qui appartiennent à plus de deux sous-domaines. On appelle multiplicité d'un nœud ou d'un **DDL**, le nombre de sous-domaines partagés par ce point et un point multiple est ainsi un point dont la multiplicité est strictement supérieure à 2.

Sur la figure 1.3, comme cela a pu être illustré par **GOSSELET et REY [2006]**, ou dans les thèses de **PARRET-FRÉAUD [2011]** ou **NEGRELLO [2017]**, la construction d'opérateurs spécifiques aux méthodes de décomposition de domaine est détaillée sur un cas bidimensionnel à trois sous-domaines. Le lecteur pourra noter qu'il existe différentes numérotations (voir figure 1.3) :

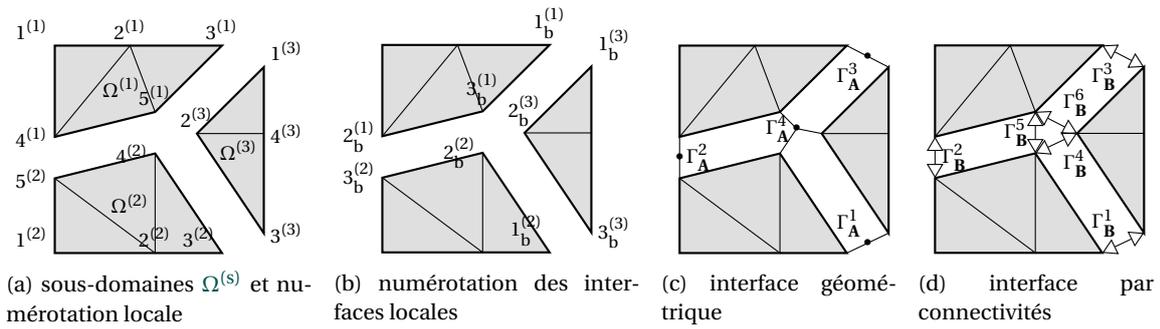
- une numérotation locale de chaque sous-domaine  $\Omega^{(s)}$  (1.3a) ;
- une numérotation de l'interface locale  $\Gamma^{(s)}$  (figure 1.3b) ;
- des numérotations de l'interface globale  $\Gamma$ , qui diffèrent aux **points multiples** :
  - une numérotation dite primale sur l'interface  $\Gamma_A$ , où tous les nœuds coïncidant de l'interface locale sont représentés par un seul nœud de l'interface primale (1.3c) ;
  - une numérotation dite duale sur l'interface  $\Gamma_B$ , où chaque connectivité est représentée par des nœuds d'interface locale deux à deux (1.3d).

Tout d'abord, en raison du fait qu'il est souhaitable de poser un problème sur l'interface globale  $\Gamma$ , il faut être capable de ramener une quantité définie sur tout  $\Omega^{(s)}$  à sa restriction sur  $\Gamma^{(s)}$ . Par exemple, si un champ  $\mathbf{x}$  est défini sur tout  $\Omega^{(s)}$ , sa restriction sur son bord  $\mathbf{x}_b^{(s)} = \mathbf{x}|_{\Gamma^{(s)}}$  peut être calculée grâce à l'opérateur de trace défini tel que :

$$\mathbf{x}_b^{(s)} = \mathbf{t}^{(s)} \mathbf{x}^{(s)}.$$

De plus, des opérations entre les sous-domaines sont nécessaires pour définir des continuités (ou des équilibres) aux interfaces, c'est pourquoi des opérateurs permettant de faire des sommes et des différences de quantités définies de part et d'autre de l'interface sont aussi introduits.

1. Le graphe dont les sommets sont les sous-domaines et les arêtes sont les connectivités entre les sous-domaines



$$\mathbf{t}^{(1)} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{A}^{(1)} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{B}^{(1)} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{t}^{(2)} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{A}^{(2)} = \begin{bmatrix} 1 & 0 & \mathbf{0} \\ 0 & 0 & \mathbf{1} \\ 0 & 0 & \mathbf{0} \\ 0 & 1 & \mathbf{0} \end{bmatrix}$$

$$\mathbf{B}^{(2)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ \mathbf{0} & \mathbf{-1} & \mathbf{0} \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{t}^{(3)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\mathbf{A}^{(3)} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\mathbf{B}^{(3)} = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 0 & -1 & 0 \end{bmatrix}$$

(e) opérateurs associés (les coefficients en gras sont les coefficient évoqués dans le texte)

FIGURE 1.3 – Différentes notations locales, de l'interface et opérateurs de trace et d'assemblage

Pour des efforts  $\lambda_b^{(s)}$  définis sur  $\Gamma^{(s)}$ , l'équilibre de ces efforts doit être imposé, c'est-à-dire que, sur une interface entre deux sous-domaines  $i$  et  $j$  (sans **point multiple**), l'on ait :  $\lambda_b^{(i)} + \lambda_b^{(j)} = 0$ . Pour faire cela dans un cadre général, il faut pouvoir assembler des quantités définies de part et d'autre de l'interface  $\Gamma$ . Cela peut être réalisé en sommant les quantités issues de chaque nœud correspondant par la relation :

$$\mathbf{x}_A = \sum_s \mathbf{A}^{(s)} \mathbf{x}_b^{(s)},$$

avec  $\mathbf{A}^{(s)}$  les opérateurs d'assemblage primal locaux définis sur la figure 1.3e et  $\mathbf{x}_A$  le vecteur de cette quantité assemblée sur l'interface primale. Pour construire de tels opérateurs permettant de passer d'une quantité définie sur  $\Gamma^{(s)}$  à une quantité définie sur l'interface primale  $\Gamma_A$ , il suffit, pour chaque colonne  $j$  (selon la numérotation de  $\Gamma^{(s)}$ ), la seule composante non nulle coïncide avec la numérotation du nœud correspondant sur l'interface globale. Par exemple, pour la construction de la troisième colonne de  $\mathbf{A}^{(2)}$  (elle correspond aux DDL du nœud  $3_b^{(2)}$ ) le nœud correspondant porte le deuxième numéro de l'interface primale  $\Gamma_A^2$ .

Quant à l'obtention de différences de part et d'autre de l'interface, qui est nécessaire pour permettre de définir les continuités des déplacements  $\mathbf{u}_b^{(s)}$  à l'interface, il faut pouvoir imposer des relations entre deux sous-domaines  $i$  et  $j$  du type  $\mathbf{u}_b^{(i)} - \mathbf{u}_b^{(j)} = 0$ . Ainsi, à l'instar des opérateurs d'assemblage primal, un assemblage dual peut aussi être défini :

$$\mathbf{x}_B = \sum_s \mathbf{B}^{(s)} \mathbf{x}^{(s)},$$

Avec  $\mathbf{B}^{(s)}$  les opérateurs d'assemblage dual locaux définis sur la figure 1.3e et  $\mathbf{x}_B$  le vecteur de cette quantité assemblée sur l'interface duale. De la même manière que les opérateurs d'assemblage primal, les colonnes des opérateurs d'assemblage dual  $\mathbf{B}^{(s)}$  sont construits par paires de 1 et  $-1$  entre les sous-domaines connectés. Par contre, l'interface duale est différente de l'interface primale. En effet, en raison des **points multiples**, il faut ici, considérer l'interface par connectivité à cause de l'opération de différence réalisée. Pour sa construction, il faut identifier chaque couple de nœud de part et d'autre de l'interface, puis reporter arbitrairement un 1 ou un  $-1$  aux nœuds concernés.

**Remarque 1.1 : Non unicité des  $\mathbf{B}^{(s)}$**

Le choix fait en figure 1.3 quant à la représentation de  $\mathbf{B}^{(s)}$  n'est pas le seul possible. En effet, ici, certaines relations sont redondantes. Cela ne pose pas de problème réel, mais il est possible de définir des relations non redondantes (en supprimant une des trois relation au niveau du **point multiple**) ou orthogonales comme suggéré par **PARK et coll. [1997]**.

Dans notre exemple de la figure 1.3, pour construire la cinquième ligne de tous les  $\mathbf{B}^{(s)}$ , il faut regarder quels nœuds de bord correspondent à  $\Gamma_B^5$ , ce qui correspond aux nœuds  $3_b^{(1)}$  et  $2_b^{(2)}$ . Pour le premier nœud ( $3_b^{(1)}$ ), un 1 est ajouté à la matrice  $\mathbf{B}^{(1)}$  à la cinquième ligne (pour  $\Gamma_B^5$ ) et en troisième position. Pour le second nœud, ( $2_b^{(2)}$ ), un  $-1$  est ajouté à la matrice  $\mathbf{B}^{(2)}$  à la cinquième ligne et en deuxième position tandis que le reste est mis à 0.

**Remarque 1.2 : Orthogonalité des opérateurs d'assemblage**

On peut noter que les opérateurs d'assemblages sont orthogonaux et que n'importe quels opérateurs  $(\mathbf{B}^{(s)})$  respectant la relation

$$\sum_s \mathbf{B}^{(s)} \mathbf{A}^{(s)T} = 0 \quad (1.2)$$

peuvent être choisis comme opérateurs d'assemblage dual. De plus, en raison de l'orthogonalité de ces opérateurs d'assemblage, il est possible d'écrire n'importe quel vecteur  $\mathbf{v}_b^{(s)}$  de bord local comme une somme de contributions de vecteurs uniques définis sur les interfaces primale et duale globales :

$$\mathbf{v}_b^{(s)} = \mathbf{A}^{(s)T} \mathbf{v}_A + \mathbf{B}^{(s)T} \mathbf{v}_B$$

Pour alléger les expressions algébriques, les notations de concaténation de quantités locales par blocs appelées notations « diamant » seront utilisées à partir d'ici. Ces concaténations se font selon des directions verticales, horizontales ou diagonales. Soit  $\mathbf{x}^{(s)}$  des vecteurs locaux et  $\mathbf{M}^{(s)}$  des matrices locales. Les trois types de notations diamant sont :

$$\begin{aligned} \mathbf{x}^\diamond &= \begin{bmatrix} \mathbf{x}^{(1)} \\ \vdots \\ \mathbf{x}^{(N_s)} \end{bmatrix}; \\ \mathbf{x}^{\diamond} &= [\mathbf{x}^{(1)} \quad \dots \quad \mathbf{x}^{(N_s)}]; \\ \mathbf{M}^{\diamond} &= \begin{bmatrix} \mathbf{M}^{(1)} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \mathbf{M}^{(N_s)} \end{bmatrix}. \end{aligned}$$

Ainsi, à titre d'exemple, l'expression (1.1.1) peut s'écrire sans le signe de sommation :

$$\mathbf{B}^\diamond \mathbf{A}^{\diamond T} = \mathbf{A}^\diamond \mathbf{B}^{\diamond T} = 0.$$

**Remarque 1.3 : Notations diamant**

En plus de la forme compacte de ces notations, le type d'opérations réalisées est plus facilement identifiable. En effet, les opérations réalisées localement et indépendamment sur chaque sous-domaine sont repérées par le signe  $\diamond$  tandis que les opérations d'assemblages sur l'interface et nécessitant des échanges le sont à l'aide de l'exposant  $\diamond$  (ou  $\diamond^T$ ).

### 1.1.2 Formulation de problèmes condensés à l'interface

À l'instar de l'indice  $\bullet_b$  introduit précédemment grâce à l'opérateur de trace, par la suite les nœuds (ou DDLs) internes aux sous-domaines seront notés avec un indice  $\bullet_i$ . Ainsi, la quantité  $\mathbf{x}$  définie uniquement sur cet intérieur est notée  $\mathbf{x}_i = \mathbf{x}|_{\Omega^{(s)} \setminus \Gamma^{(s)}}$ .

#### 1.1.2.1 Équilibre local des sous-domaines

Le partitionnement de la géométrie conduit aussi au partitionnement de l'équation d'équilibre globale (1.1) qui devient :

$$\mathbf{K}^{(s)} \mathbf{u}^{(s)} = \mathbf{f}^{(s)} + \boldsymbol{\lambda}^{(s)}, \quad \forall s \in \llbracket 1, N_s \rrbracket \quad (1.3a)$$

On note ici, l'apparition de multiplicateurs de Lagrange  $\boldsymbol{\lambda}^{(s)}$  qui rendent compte de l'interaction qui existe entre le sous-domaine  $\Omega^{(s)}$  et ses voisins. Ce sont des quantités nodales généralisées d'un effort surfacique qui sont nulles à l'intérieur du sous-domaine et ont une valeur non-nulle sur son interface.

En plus de l'équation d'équilibre, il faut ajouter les conditions de continuité des déplacements à l'interface et d'équilibre de ces interfaces. Elles correspondent, dans un cas à deux sous-domaines, aux équations  $\mathbf{u}_b^{(1)} - \mathbf{u}_b^{(2)} = 0$  et  $\boldsymbol{\lambda}_b^{(1)} + \boldsymbol{\lambda}_b^{(2)} = 0$  respectivement. La généralisation de ces équations pour  $N_s$  sous-domaines et les opérateurs d'assemblage de la section précédente permettent d'écrire deux relations sur les quantités d'interface :

$$\begin{aligned}\sum_s \mathbf{B}^{(s)} \mathbf{t}^{(s)} \mathbf{u}^{(s)} &= \mathbf{0}; \\ \sum_s \mathbf{A}^{(s)} \mathbf{t}^{(s)} \boldsymbol{\lambda}^{(s)} &= \mathbf{0}.\end{aligned}$$

On suppose qu'il est possible de renuméroter les DDLs, selon leur appartenance au bord ou l'intérieur du sous-domaine de manière à écrire n'importe quelle quantité par bloc. Ainsi, n'importe quel vecteur  $\mathbf{x}$  peut être mis sous la forme :

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_i \\ \mathbf{x}_b \end{bmatrix}.$$

L'équation d'équilibre (1.3a) peut ensuite être mise sous la forme séparée :

$$\begin{bmatrix} \mathbf{K}_{ii}^{(s)} & \mathbf{K}_{ib}^{(s)} \\ \mathbf{K}_{bi}^{(s)} & \mathbf{K}_{bb}^{(s)} \end{bmatrix} \begin{bmatrix} \mathbf{u}_i^{(s)} \\ \mathbf{u}_b^{(s)} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_i^{(s)} \\ \mathbf{f}_b^{(s)} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\lambda}_b^{(s)} \end{bmatrix}, \quad \forall s \in \llbracket 1, N_s \rrbracket. \quad (1.4)$$

En suivant le même type de démarche que celle de la condensation statique évoquée dans CRAIG et BAMPTON [1968], il est possible de condenser le problème sur un sous-ensemble de DDLs, en particulier ceux du bord. La première ligne du système (1.4) donne un relation liant  $\mathbf{u}_i^{(s)}$  et  $\mathbf{u}_b^{(s)}$  :

$$\mathbf{u}_i^{(s)} = \mathbf{K}_{ii}^{(s)-1} (\mathbf{f}_i^{(s)} - \mathbf{K}_{ib}^{(s)} \mathbf{u}_b^{(s)}), \quad \forall s \in \llbracket 1, N_s \rrbracket. \quad (1.5)$$

En réinjectant (1.5) dans la seconde ligne de l'équation (1.4), il est possible d'arriver à une équation de la forme :

$$\mathbf{S}_p^{(s)} \mathbf{u}_b^{(s)} = \mathbf{b}_p^{(s)} + \boldsymbol{\lambda}_b^{(s)}, \quad \forall s \in \llbracket 1, N_s \rrbracket,$$

avec

$$\begin{aligned}\mathbf{S}_p^{(s)} &= \mathbf{K}_{bb}^{(s)} - \mathbf{K}_{bi}^{(s)} \mathbf{K}_{ii}^{(s)-1} \mathbf{K}_{ib}^{(s)}, \\ \mathbf{b}_p^{(s)} &= \mathbf{f}_b^{(s)} - \mathbf{K}_{bi}^{(s)} \mathbf{K}_{ii}^{(s)-1} \mathbf{f}_i^{(s)}.\end{aligned}$$

L'opérateur  $\mathbf{S}_p^{(s)}$  est appelé le complément de Schur (ZHANG [2005]) primal local du sous-domaine  $\Omega^{(s)}$ . Cet opérateur prête son nom à toutes ces méthodes condensées sur l'interface entre les sous-domaines. Il permet de calculer des efforts de réaction à l'interface à partir d'un champ de déplacements donné. Cet opérateur existe bien car l'inverse de  $\mathbf{K}_{ii}$  existe aussi. En effet,  $\mathbf{K}_{ii}^{-1}$  correspond à un problème à conditions limites de Dirichlet nulles sur le bord : tous les mouvements de corps rigides sont bien bloqués et le problème admet donc une unique solution. Le second membre condensé  $\mathbf{b}_p^{(s)}$  représente la condensation des efforts intérieurs auxquels les efforts déjà imposés sont ajoutés directement sur l'interface.

Ainsi, le problème global sous-structuré en notation diamant se met sous la forme :

$$\begin{aligned}\mathbf{S}_p^\diamond \mathbf{u}_b^\diamond &= \mathbf{b}_p^\diamond + \boldsymbol{\lambda}_b^\diamond \\ \mathbf{B}^\diamond \mathbf{u}_b^\diamond &= \mathbf{0}\end{aligned} \quad (1.6a)$$

$$\mathbf{A}^\diamond \boldsymbol{\lambda}_b^\diamond = \mathbf{0} \quad (1.6b)$$

### 1.1.2.2 Approche primale

LE TALLEC et coll. [1991] introduit pour la première fois l'approche primale et sa version améliorant sa robustesse et son extensibilité a été développée quelques années plus tard sous le nom de *Balanced Domain Decomposition (BDD)* par LE TALLEC [1994] et MANDEL [1993].

La formulation primale s'attache à trouver un champ de déplacements  $\mathbf{u}_b^\diamond$  continu *a priori* qui permet d'équilibrer le champ d'inter-efforts  $\lambda_b^\diamond$ . Ce champ de déplacements prend la forme :

$$\mathbf{u}_b^\diamond = \mathbf{A}^\diamond \mathbf{u}_A$$

avec  $\mathbf{u}_A$  un champ inconnu défini sur l'interface primale  $\Gamma_A$ . Cette forme de déplacement d'interface impose bien sa continuité et le respect de l'équation (1.6a) en vertu de l'orthogonalité entre les opérateurs d'assemblage primal et dual apparaissant dans la remarque 1.1.1. Ainsi le système (1.6) devient :

$$\begin{aligned} \mathbf{S}_p^\diamond \mathbf{A}^\diamond \mathbf{u}_A &= \mathbf{b}_p^\diamond + \lambda_b^\diamond \\ \mathbf{A}^\diamond \lambda_b^\diamond &= 0 \end{aligned}$$

En multipliant, la première équation par  $\mathbf{A}^\diamond$ ,  $\lambda_b^\diamond$  peut disparaître à l'aide de la seconde équation pour arriver finalement à l'équation :

$$\mathbf{S}_p \mathbf{u}_A = \mathbf{b}_p \quad (1.8)$$

où

$$\begin{aligned} \mathbf{S}_p &= \mathbf{A}^\diamond \mathbf{S}_p^\diamond \mathbf{A}^\diamond \mathbf{A}^\diamond \mathbf{A}^\diamond \mathbf{A}^\diamond \\ \mathbf{b}_p &= \mathbf{A}^\diamond \mathbf{b}_p^\diamond \end{aligned}$$

Le résidu  $\mathbf{b}_p - \mathbf{S}_p \mathbf{u}_A = \mathbf{A}^\diamond \mathbf{u}_A^\diamond$  correspond au déséquilibre des inter-efforts à l'interface entre les sous-domaines. L'approche primale consiste donc à minimiser ce déséquilibre d'interface tout en supposant un champ de déplacement continu.

Le système (1.8) est un système symétrique défini positif (car l'opérateur  $\mathbf{S}_p$  hérite des propriétés de  $\mathbf{K}$ ) qui peut être résolu par n'importe quel solveur. En pratique, il est résolu à l'aide d'un Gradient Conjugué, ou *Conjugate Gradient (CG)* et l'opérateur  $\mathbf{S}_p$  n'est jamais assemblé explicitement.

### 1.1.2.3 Approche duale

L'approche duale a été introduite pour la première fois par FARHAT et ROUX [1991] à l'aide de l'algorithme *FETI*. Tout comme pour l'approche primale, des développements ont permis d'améliorer la robustesse et les performances de la méthode quelques années plus tard par FARHAT et coll. [1994]. Cette méthode a donné lieu à plusieurs variantes; on pourra citer par exemple *Total FETI (TFETI)* DOSTÁL et coll. [2006] et *Hybrid FETI (HFETI)* MERTA et coll. [2017] ou encore *Dual-Primal-FETI (FETI-DP)* FARHAT et coll. [2001]

À l'inverse de l'approche primale qui travaille avec une unique inconnue d'interface formulée sur les déplacements, l'approche duale quant à elle, s'attache à déterminer un champ d'inter-efforts  $\lambda_b^\diamond$  *a priori* équilibré à l'interface qui permet d'obtenir un champ de déplacements  $\mathbf{u}_b^\diamond$  continu. Ces multiplicateurs de Lagrange prennent la forme :

$$\lambda_b^\diamond = \mathbf{B}^\diamond \lambda_B$$

avec  $\lambda_B$  un champ inconnu défini sur l'interface duale  $\Gamma_B$ . Cette forme de champ d'inter-efforts d'interface a pour conséquence de respecter intrinsèquement l'équilibre des inter-efforts à l'interface de l'équation (1.6b), grâce à l'orthogonalité des opérateurs d'assemblage de l'équation (1.2).

De la même manière, le système (1.6) devient pour l'approche duale :

$$\mathbf{S}_p^\diamond \mathbf{u}_b^\diamond = \mathbf{b}_p^\diamond + \mathbf{B}^\diamond \boldsymbol{\lambda}_B, \quad (1.9a)$$

$$\mathbf{B}^\diamond \mathbf{u}_b^\diamond = 0. \quad (1.9b)$$

Pour éliminer  $\mathbf{u}_b^\diamond$ , il faut être capable d'inverser  $\mathbf{S}_p^\diamond$ . Or cet opérateur peut être singulier si un nombre trop faible de conditions aux limites de Dirichlet sont imposées aux sous-structures  $(\Omega^{(s)})_{1 \leq s \leq N_s}$ , qui sont ainsi qualifiées de « flottantes ». C'est pourquoi il va être nécessaire d'utiliser une inverse généralisée<sup>2</sup> (ou pseudo-inverse)  $\mathbf{S}_p^{\diamond+}$  de  $\mathbf{S}_p^\diamond$  pour résoudre l'équation (1.9a). Parmi les différentes inverses généralisées, on peut citer celle de Moore-Penrose [FILL et FISHKIND \[2000\]](#).

Pour que le système ait une solution, l'utilisation de la pseudo inverse impose que le second membre soit dans  $\text{Im}(\mathbf{S}_p^\diamond)$ , ce qui revient à ce que le second membre soit orthogonal à  $\text{Ker}(\mathbf{S}_p^\diamond)$ . Cela correspond à l'équation suivante :

$$\mathbf{R}_b^{\diamond T} \left( \mathbf{b}_p^\diamond + \mathbf{B}^\diamond \boldsymbol{\lambda}_B \right) = 0,$$

avec  $\mathbf{R}_b^\diamond = \text{Ker}(\mathbf{S}_p^\diamond)$  qui représente la trace sur l'interface de la base des modes de corps rigides ou de la base des modes à énergie nulle. D'un point de vue mécanique, le second membre ne doit pas faire travailler le noyau de l'opérateur. Grâce à cette pseudo-inverse, la résolution de (1.9a) se traduit par :

$$\mathbf{u}_b^\diamond = \mathbf{S}_p^{\diamond+} \left( \mathbf{b}_p^\diamond + \mathbf{B}^\diamond \boldsymbol{\lambda}_B \right) + \mathbf{R}_b^\diamond \boldsymbol{\alpha}^\diamond$$

où  $\boldsymbol{\alpha}^\diamond$ , représente l'amplitude de la contribution des mouvements de corps rigides au déplacement des sous-structures  $(\Omega^{(s)})_{1 \leq s \leq N_s}$ . En injectant cette équation dans l'équation de continuité de l'interface (1.9b), il est possible d'écrire :

$$\begin{bmatrix} \mathbf{S}_d & \mathbf{G}_d^\diamond \\ \mathbf{G}_d^{\diamond T} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}_B \\ \boldsymbol{\alpha}^\diamond \end{bmatrix} = \begin{bmatrix} \mathbf{b}_d \\ \mathbf{e}^\diamond \end{bmatrix}, \quad (1.10)$$

en notant :

$$\begin{aligned} \mathbf{S}_d &= \mathbf{B}^\diamond \mathbf{S}_p^{\diamond+} \mathbf{B}^{\diamond T} \\ \mathbf{G}_d^\diamond &= \mathbf{B}^\diamond \mathbf{R}_b^\diamond \\ \mathbf{b}_d &= -\mathbf{B}^\diamond \mathbf{S}_p^{\diamond+} \mathbf{b}_p^\diamond \\ \mathbf{e}^\diamond &= -\mathbf{R}_b^{\diamond T} \mathbf{b}_p^\diamond \end{aligned}$$

2. l'inverse généralisée d'une matrice  $\mathbf{M}$  notée  $\mathbf{M}^+$  est définie telle que :  $\forall \mathbf{x} \in \text{Im}(\mathbf{M}), \mathbf{M}\mathbf{M}^+\mathbf{x} = \mathbf{x}$

**Remarque 1.4 : Complément de Schur dual**

Pour faire le lien avec les notations qui apparaissent dans GOSSELET et REY [2006] sur l'approche duale, on peut introduire  $\mathbf{S}_d^\diamond = \mathbf{S}_p^{\diamond+}$  vérifiant qui est une concaténation le long de la diagonale du complément de Schur dual local  $\mathbf{S}_d^{(s)}$  vérifiant  $\mathbf{S}_d = \mathbf{B}^\diamond \mathbf{S}_d^\diamond \mathbf{B}^{\diamond T}$ . Cette matrice peut être exprimée à l'aide de la matrice de raideur initiale sachant que l'inverse d'une matrice par blocs  $\mathbf{K}$  symétrique vérifie :

$$\begin{bmatrix} \mathbf{K}_{ii} & \mathbf{K}_{ib} \\ \mathbf{K}_{bi} & \mathbf{K}_{bb} \end{bmatrix}^{-1} = \begin{bmatrix} (\mathbf{K}_{ii} - \mathbf{K}_{ib} \mathbf{K}_{bb}^{-1} \mathbf{K}_{bi})^{-1} & -\mathbf{K}_{ii}^{-1} \mathbf{K}_{ib} (\mathbf{K}_{bb} - \mathbf{K}_{bi} \mathbf{K}_{ii}^{-1} \mathbf{K}_{ib})^{-1} \\ -(\mathbf{K}_{bb} - \mathbf{K}_{bi} \mathbf{K}_{ii}^{-1} \mathbf{K}_{ib})^{-1} \mathbf{K}_{bi} \mathbf{K}_{ii}^{-1} & (\mathbf{K}_{bb} - \mathbf{K}_{bi} \mathbf{K}_{ii}^{-1} \mathbf{K}_{ib})^{-1} \end{bmatrix}$$

On notera que le bloc inférieur droit correspondant à l'inverse de  $\mathbf{K}$  restreinte à son bord est exactement l'expression du complément de Schur primal local. Ainsi l'expression de  $\mathbf{S}_d^\diamond$  est :

$$\mathbf{S}_d^\diamond = \mathbf{S}_p^{\diamond+} = (\mathbf{K}^{\diamond+})_{bb} = \mathbf{t}^\diamond \mathbf{K}^{\diamond+} \mathbf{t}^{\diamond T}$$

De même, il est possible d'introduire  $\mathbf{b}_d^\diamond = \mathbf{S}_p^{\diamond+} \mathbf{b}_p^\diamond$  le second membre condensé local dual. La partie inférieure gauche de l'inverse de la matrice par blocs donne aussi l'égalité suivante :

$$(\mathbf{K}^{\diamond+})_{bi} = -\mathbf{S}_p^{\diamond+} \mathbf{K}_{bi} \mathbf{K}_{ii}^{-1}$$

Ainsi, il est possible d'écrire :

$$\begin{aligned} \mathbf{b}_d^\diamond &= \mathbf{S}_p^{\diamond+} \mathbf{b}_p^\diamond \\ &= \mathbf{S}_p^{\diamond+} \mathbf{f}_b^\diamond - \mathbf{S}_p^{\diamond+} \mathbf{K}_{bi} \mathbf{K}_{ii}^{-1} \mathbf{f}_i^\diamond \\ &= (\mathbf{K}^{\diamond+})_{bb} \mathbf{f}_b^\diamond + (\mathbf{K}^{\diamond+})_{bi} \mathbf{f}_i^\diamond \\ &= (\mathbf{K}^{\diamond+} \mathbf{f}^\diamond)_b \\ &= \mathbf{t}^\diamond \mathbf{K}^{\diamond+} \mathbf{f}^\diamond \end{aligned}$$

Le système linéaire de l'approche duale, comme comme celui de l'approche primale, conduit à la formulation d'un système linéaire symétrique défini positif posé sur l'interface dont la résolution sera réalisée par un solveur itératif de Krylov (voir Annexe A.1) de type CG SAAD [1997]. Ici, le système est un point selle dont la résolution sera réalisée à l'aide de techniques de projection comme ce sera expliqué dans la section 1.1.3.2.

#### 1.1.2.4 Approche mixte

MANDEL et TEZAUER [1996] ou encore KLAUWONN et WIDLUND [2001] ont montré que le choix du type d'approche n'influe pas sur la convergence du solveur car les systèmes primal et dual présentent le même conditionnement. D'autres méthodes ne travaillent plus sur une unique inconnue d'interface en déplacements ou en inter-efforts, mais sur une combinaison linéaire de ces quantités d'interface. Ces méthodes dont les propriétés de convergence sont potentiellement meilleures avec un choix judicieux de cette condition d'interface mixte font écho à la méthode de Schwarz optimisée de LIONS [1990], évoquée en introduction de cette section, où un choix optimal de conditions d'interface de Robin permettait d'atteindre la convergence même à recouvrement nul. Ces aspects de conditions d'interface mixte (ou conditions de Robin) se retrouvent aussi dans plusieurs autres algorithmes de décomposition de domaine tels que la méthode *Large Time Increments (LaTin)* de LADEVÈZE [1999] et sa version parallèle LADEVÈZE et coll. [2007] ou les méthodes de type *FETI-2LM* de ROUX [2009] dont des concepts similaires seront développés ici. Le système sous-structuré (1.6) est toujours le point de départ. Une nouvelle inconnue d'interface homogène

à des inter-efforts est introduite :  $\boldsymbol{\mu}_b^\diamond$ . Elle est une combinaison linéaire de  $\boldsymbol{\lambda}_b^\diamond$  et  $\mathbf{u}_b^\diamond$  telle que :

$$\boldsymbol{\mu}_b^\diamond = \boldsymbol{\lambda}_b^\diamond + \mathbf{Q}_b^\diamond \mathbf{u}_b^\diamond. \quad (1.11)$$

Le paramètre  $\mathbf{Q}_b^\diamond$  correspond à une rigidité d'interface qui est aussi appelée impédance d'interface. En effet, imposer cette relation entre les quantités primales et duales d'interface s'apparente à relier l'interface et les sous-domaines  $(\Omega^{(s)})_{1 \leq s \leq N_s}$  par des ressorts de rigidité  $\mathbf{Q}_b^{(s)}$ . Après réécriture de l'équation (1.3a) en notation par blocs et en éliminant  $\boldsymbol{\lambda}^\diamond$  à l'aide de l'équation (1.11), il en découle :

$$\begin{aligned} \mathbf{K}^\diamond \mathbf{u}^\diamond &= \mathbf{f}^\diamond + \mathbf{t}^{\diamond T} (\boldsymbol{\mu}_b^\diamond - \mathbf{Q}_b^\diamond \mathbf{u}_b^\diamond) \\ \Leftrightarrow (\mathbf{K}^\diamond + \mathbf{t}^{\diamond T} \mathbf{Q}_b^\diamond \mathbf{t}^\diamond) \mathbf{u}^\diamond &= \mathbf{f}^\diamond + \mathbf{t}^{\diamond T} \boldsymbol{\mu}_b^\diamond \\ \Leftrightarrow \mathbf{u}^\diamond &= (\mathbf{K}^\diamond + \mathbf{t}^{\diamond T} \mathbf{Q}_b^\diamond \mathbf{t}^\diamond)^{-1} (\mathbf{f}^\diamond + \mathbf{t}^{\diamond T} \boldsymbol{\mu}_b^\diamond) \end{aligned} \quad (1.12)$$

Cette expression est valable uniquement si l'ajout de  $\mathbf{t}^{\diamond T} \mathbf{Q}_b^\diamond \mathbf{t}^\diamond$  à  $\mathbf{K}^\diamond$  rend le problème inversible.  $\mathbf{K}^\diamond$  étant une matrice singulière (certaines sous-structures peuvent ne pas comporter suffisamment de conditions limites de Dirichlet), une condition pour définir la formulation mixte est que  $\mathbf{Q}_b^\diamond$  permette de bloquer les mouvements de corps rigides des sous-structures  $(\Omega^{(s)})_{1 \leq s \leq N_s}$  qui seraient flottantes sans son adjonction.

Il peut en découler plusieurs formulations dépendant de l'expression de la condition de raccord faisant intervenir à la fois  $\mathbf{u}_b^\diamond$  et  $\boldsymbol{\lambda}_b^\diamond$  entre les sous-domaines.

**Première formulation** En réécrivant les conditions d'interface (1.6a) et (1.6b) sous une forme ne faisant intervenir uniquement l'opérateur d'assemblage primal  $\mathbf{A}^\diamond$  (et sous réserve que  $\mathbf{A}^\diamond \mathbf{Q}_b^\diamond \mathbf{A}^{\diamond T}$  soit inversible), l'expression suivante est obtenue :

$$\mathbf{A}^{\diamond T} (\mathbf{A}^\diamond \mathbf{Q}_b^\diamond \mathbf{A}^{\diamond T})^{-1} \mathbf{A}^\diamond \boldsymbol{\mu}_b^\diamond - \mathbf{u}_b^\diamond = 0. \quad (1.13)$$

On peut vérifier qu'en multipliant à gauche (1.13) par  $\mathbf{A}^\diamond \mathbf{Q}_b^\diamond$ , l'équation (1.6b) est bien obtenue et qu'en multipliant à gauche par  $\mathbf{B}^\diamond$ , (1.6a) apparaît :

$$\begin{aligned} \mathbf{A}^\diamond \mathbf{Q}_b^\diamond \left[ \mathbf{A}^{\diamond T} (\mathbf{A}^\diamond \mathbf{Q}_b^\diamond \mathbf{A}^{\diamond T})^{-1} \mathbf{A}^\diamond \boldsymbol{\mu}_b^\diamond - \mathbf{u}_b^\diamond \right] &= (\mathbf{A}^\diamond \mathbf{Q}_b^\diamond \mathbf{A}^{\diamond T}) (\mathbf{A}^{\diamond T} (\mathbf{A}^\diamond \mathbf{Q}_b^\diamond \mathbf{A}^{\diamond T})^{-1} \mathbf{A}^\diamond \boldsymbol{\mu}_b^\diamond - \mathbf{A}^\diamond \mathbf{Q}_b^\diamond \mathbf{u}_b^\diamond) \\ &= \mathbf{A}^\diamond (\boldsymbol{\mu}_b^\diamond - \mathbf{Q}_b^\diamond \mathbf{u}_b^\diamond) \\ &= \mathbf{A}^\diamond \boldsymbol{\lambda}_b^\diamond = 0 \\ \mathbf{B}^\diamond \left[ \mathbf{A}^{\diamond T} (\mathbf{A}^\diamond \mathbf{Q}_b^\diamond \mathbf{A}^{\diamond T})^{-1} \mathbf{A}^\diamond \boldsymbol{\mu}_b^\diamond - \mathbf{u}_b^\diamond \right] &= \underbrace{\mathbf{B}^\diamond \mathbf{A}^{\diamond T}}_{=0} (\mathbf{A}^\diamond \mathbf{Q}_b^\diamond \mathbf{A}^{\diamond T})^{-1} \mathbf{A}^\diamond \boldsymbol{\mu}_b^\diamond - \mathbf{B}^\diamond \mathbf{u}_b^\diamond \\ &= -\mathbf{B}^\diamond \mathbf{u}_b^\diamond = 0 \end{aligned}$$

Il est possible d'injecter le déplacement  $\mathbf{u}^\diamond$  de l'équation (1.12) dans la condition d'interface (1.13) et d'obtenir une équation ne dépendant que de l'inconnue mixte d'interface  $\boldsymbol{\mu}_b^\diamond$  :

$$\begin{aligned} \mathbf{A}^{\diamond T} (\mathbf{A}^\diamond \mathbf{Q}_b^\diamond \mathbf{A}^{\diamond T})^{-1} \mathbf{A}^\diamond \boldsymbol{\mu}_b^\diamond - \mathbf{t}^\diamond (\mathbf{K}^\diamond + \mathbf{t}^{\diamond T} \mathbf{Q}_b^\diamond \mathbf{t}^\diamond)^{-1} (\mathbf{f}^\diamond + \mathbf{t}^{\diamond T} \boldsymbol{\mu}_b^\diamond) &= 0 \\ \left[ \mathbf{A}^{\diamond T} (\mathbf{A}^\diamond \mathbf{Q}_b^\diamond \mathbf{A}^{\diamond T})^{-1} \mathbf{A}^\diamond - \mathbf{t}^\diamond (\mathbf{K}^\diamond + \mathbf{t}^{\diamond T} \mathbf{Q}_b^\diamond \mathbf{t}^\diamond)^{-1} \mathbf{t}^{\diamond T} \right] \boldsymbol{\mu}_b^\diamond &= \mathbf{t}^\diamond (\mathbf{K}^\diamond + \mathbf{t}^{\diamond T} \mathbf{Q}_b^\diamond \mathbf{t}^\diamond)^{-1} (\mathbf{f}^\diamond) \end{aligned} \quad (1.14)$$

En introduisant un complément de Schur mixte  $\mathbf{S}_m^\diamond$  et un second membre condensé mixte  $\mathbf{b}_m^\diamond$  définis tels que :

$$\begin{aligned}\mathbf{S}_m^\diamond &= \mathbf{t}^\diamond \left( \mathbf{K}^\diamond + \mathbf{t}^{\diamond T} \mathbf{Q}_b^\diamond \mathbf{t}^\diamond \right)^{-1} \mathbf{t}^{\diamond T} = \left( \mathbf{S}_p^\diamond + \mathbf{Q}_b^\diamond \right)^{-1} \\ \mathbf{b}_m^\diamond &= \mathbf{t}^\diamond \left( \mathbf{K}^\diamond + \mathbf{t}^{\diamond T} \mathbf{Q}_b^\diamond \mathbf{t}^\diamond \right)^{-1} \mathbf{f}^\diamond = \mathbf{S}_m^\diamond \mathbf{b}_p^\diamond,\end{aligned}$$

l'équation (1.14) devient :

$$\left[ \mathbf{A}^{\diamond T} \left( \mathbf{A}^\diamond \mathbf{Q}_b^\diamond \mathbf{A}^{\diamond T} \right)^{-1} \mathbf{A}^\diamond - \mathbf{S}_m^\diamond \right] \boldsymbol{\mu}_b^\diamond = \mathbf{b}_m^\diamond. \quad (1.15)$$

$\mathbf{S}_m^\diamond$  et  $\mathbf{b}_m^\diamond$  sont bien évidemment des concaténations des opérateurs locaux  $\mathbf{S}_m^{(s)}$  et des seconds membres locaux  $\mathbf{b}_m^{(s)}$ . Ce problème est formulé sur l'interface locale  $\Gamma^{(s)}$  et non pas sur  $\Gamma_A$  ou  $\Gamma_B$  comme c'est le cas pour les approches primale et duale. En effet, on ne recherche pas une inconnue assemblée mais une inconnue non assemblée et donc différente de part et d'autre de l'interface. C'est la formulation mise en avant par [NEGRELLO et coll. \[2016\]](#) et [NEGRELLO et coll. \[2018\]](#) qui mentionne le fait que ces opérateurs sont les mêmes que ceux utilisés dans la méthode *FETI-2LM* de [ROUX et coll. \[2003\]](#).

**Remarque 1.5 : Lien entre quantités mixtes et primales**

Le lien entre ces opérateurs est explicité par [NEGRELLO et coll. \[2018\]](#). En réécrivant l'équation (1.12) de manière condensée, il est possible de trouver le déplacement de bord de la forme :

$$\mathbf{u}_b^\diamond = \left( \mathbf{S}_p^\diamond + \mathbf{Q}_b^\diamond \right)^{-1} \left( \mathbf{b}_p^\diamond + \boldsymbol{\mu}_b^\diamond \right)$$

En injectant cette forme de  $\mathbf{u}_b$  dans (1.13), le lien entre  $\mathbf{S}_m$  et  $\mathbf{S}_p$  est évident :

$$\left[ \mathbf{A}^{\diamond T} \left( \mathbf{A}^\diamond \mathbf{Q}_b^\diamond \mathbf{A}^{\diamond T} \right)^{-1} \mathbf{A}^\diamond - \left( \mathbf{S}_p^\diamond + \mathbf{Q}_b^\diamond \right)^{-1} \right] \boldsymbol{\mu}_b^\diamond = \left( \mathbf{S}_p^\diamond + \mathbf{Q}_b^\diamond \right)^{-1} \mathbf{b}_p^\diamond,$$

et on retrouve bien ce qui a été annoncé en comparant cette équation avec l'équation (1.14)

**Seconde formulation** La réécriture des conditions d'interface (1.6a) est aussi possible si  $\mathbf{B}^\diamond \mathbf{Q}_b^\diamond \mathbf{B}^{\diamond T}$  est inversible dans  $\text{Im}(\mathbf{B}^\diamond)$ . En effet, il est possible de lier ces conditions par la formule suivante :

$$\mathbf{A}^{\diamond T} \mathbf{A}^\diamond \boldsymbol{\lambda}_b^\diamond + \mathbf{Q}_b^\diamond \mathbf{B}^{\diamond T} \mathbf{B}^\diamond \mathbf{u}_b^\diamond = 0 \quad (1.16)$$

comme le suggèrent [GOSSELET et REY \[2006\]](#) ou encore [LOISEL \[2013\]](#). Il faut vérifier que cette équation est bien équivalente aux conditions d'interface des équations (1.6a) et (1.6b). À partir de l'équation (1.16) et en la multipliant à gauche par  $\mathbf{B}^\diamond$ , il est possible d'éliminer la dépendance en  $\boldsymbol{\lambda}_b^\diamond$  :

$$\begin{aligned}\mathbf{B}^\diamond \left[ \mathbf{A}^{\diamond T} \mathbf{A}^\diamond \boldsymbol{\lambda}_b^\diamond + \mathbf{Q}_b^\diamond \mathbf{B}^{\diamond T} \mathbf{B}^\diamond \mathbf{u}_b^\diamond \right] &= \overbrace{\mathbf{B}^\diamond \mathbf{A}^{\diamond T} \mathbf{A}^\diamond \boldsymbol{\lambda}_b^\diamond}^{=0} + \mathbf{B}^\diamond \mathbf{Q}_b^\diamond \mathbf{B}^{\diamond T} \mathbf{B}^\diamond \mathbf{u}_b^\diamond \\ &= \mathbf{B}^\diamond \mathbf{Q}_b^\diamond \mathbf{B}^{\diamond T} \mathbf{B}^\diamond \mathbf{u}_b^\diamond = 0 \\ &\Leftrightarrow \mathbf{B}^\diamond \mathbf{u}_b^\diamond = 0.\end{aligned}$$

L'équivalence est vraie si et seulement si  $\mathbf{B}^\diamond \mathbf{Q}_b^\diamond \mathbf{B}^{\diamond T}$  est inversible dans  $\text{Im}(\mathbf{B}^\diamond)$ . Enfin, en utilisant cette condition de continuité à l'interface et en l'injectant dans (1.16), et après multiplication

gauche par  $\mathbf{A}^\diamond$ , il est possible d'obtenir :

$$\begin{aligned} \mathbf{A}^\diamond \left[ \mathbf{A}^{\diamond T} \mathbf{A}^\diamond \boldsymbol{\lambda}_b^\diamond + \mathbf{Q}_b^\diamond \mathbf{B}^{\diamond T} \mathbf{B}^\diamond \mathbf{u}_b^\diamond \right] &= \mathbf{A}^\diamond \mathbf{A}^{\diamond T} \mathbf{A}^\diamond \boldsymbol{\lambda}_b^\diamond + \mathbf{A}^\diamond \mathbf{Q}_b^\diamond \mathbf{B}^{\diamond T} \underbrace{\mathbf{B}^\diamond \mathbf{u}_b^\diamond}_{=0} \\ &= \mathbf{A}^\diamond \mathbf{A}^{\diamond T} \mathbf{A}^\diamond \boldsymbol{\lambda}_b^\diamond = 0 \\ &\Leftrightarrow \mathbf{A}^\diamond \boldsymbol{\lambda}_b^\diamond = 0. \end{aligned}$$

Ainsi, la condition (1.16) est bien équivalente aux équations (1.6a) et (1.6b).

À partir de (1.16) et en injectant  $\boldsymbol{\lambda}_b^\diamond$  de l'équation (1.11),

$$\mathbf{A}^{\diamond T} \mathbf{A}^\diamond \boldsymbol{\mu}_b^\diamond + \left( \mathbf{Q}_b^\diamond \mathbf{B}^{\diamond T} \mathbf{B}^\diamond - \mathbf{A}^{\diamond T} \mathbf{A}^\diamond \mathbf{Q}_b^\diamond \right) \mathbf{u}_b^\diamond = 0$$

puis en remplaçant  $\mathbf{u}_b^\diamond$  grâce à l'équation (1.12), une formulation mixte différente de la précédente est obtenue :

$$\begin{aligned} \left[ \mathbf{A}^{\diamond T} \mathbf{A}^\diamond + \left( \mathbf{Q}_b^\diamond \mathbf{B}^{\diamond T} \mathbf{B}^\diamond - \mathbf{A}^{\diamond T} \mathbf{A}^\diamond \mathbf{Q}_b^\diamond \right) \mathbf{t}^\diamond \left( \mathbf{K}^\diamond + \mathbf{t}^{\diamond T} \mathbf{Q}_b^\diamond \mathbf{t}^\diamond \right)^{-1} \mathbf{t}^{\diamond T} \right] \boldsymbol{\mu}_b^\diamond \\ = - \left( \mathbf{Q}_b^\diamond \mathbf{B}^{\diamond T} \mathbf{B}^\diamond - \mathbf{A}^{\diamond T} \mathbf{A}^\diamond \mathbf{Q}_b^\diamond \right) \mathbf{t}^\diamond \left( \mathbf{K}^\diamond + \mathbf{t}^{\diamond T} \mathbf{Q}_b^\diamond \mathbf{t}^\diamond \right)^{-1} \mathbf{f}^\diamond. \end{aligned}$$

Avec les notations mixtes introduites précédemment, cette équation se met sous la forme :

$$\left[ \mathbf{A}^{\diamond T} \mathbf{A}^\diamond + \left( \mathbf{Q}_b^\diamond \mathbf{B}^{\diamond T} \mathbf{B}^\diamond - \mathbf{A}^{\diamond T} \mathbf{A}^\diamond \mathbf{Q}_b^\diamond \right) \mathbf{S}_m^\diamond \right] \boldsymbol{\mu}_b^\diamond = - \left( \mathbf{Q}_b^\diamond \mathbf{B}^{\diamond T} \mathbf{B}^\diamond - \mathbf{A}^{\diamond T} \mathbf{A}^\diamond \mathbf{Q}_b^\diamond \right) \mathbf{b}_m^\diamond,$$

et peut être simplifiée en se rappelant que  $\mathbf{S}_m^\diamond = \left( \mathbf{S}_p^\diamond + \mathbf{Q}_b^\diamond \right)^{-1}$  :

$$\left( \mathbf{A}^{\diamond T} \mathbf{A}^\diamond \mathbf{S}_p^\diamond + \mathbf{Q}_b^\diamond \mathbf{B}^{\diamond T} \mathbf{B}^\diamond \right) \mathbf{S}_m^\diamond \boldsymbol{\mu}_b^\diamond = \left( \mathbf{A}^{\diamond T} \mathbf{A}^\diamond \mathbf{Q}_b^\diamond - \mathbf{Q}_b^\diamond \mathbf{B}^{\diamond T} \mathbf{B}^\diamond \right) \mathbf{b}_m^\diamond, \quad (1.17)$$

### 1.1.2.5 Approches hybrides et recondensées

D'autres types d'approches existent : les méthodes hybrides où un certain nombre de DDLs d'interface sont choisis comme primaux tandis que les autres sont choisis comme étant duaux.

Une première classe de méthodes constitue une unification des approches primale et duale ; elles sont appelées des méthodes hybrides. En effet, leur principe réside dans le fait que la recherche d'un champ de déplacement  $\mathbf{u}_p$  continu sur une partie des DDLs d'interface est effectuée en même temps que la recherche d'un champ d'inter-efforts équilibré  $\boldsymbol{\lambda}_d$  sur les DDLs d'interface restants et est exprimé par la relation (1.18b). C'est une méthode hybride qui a été développée par GOSSELET [2003]; GOSSELET et coll. [2004]. Cette méthode permet de considérer la résolution de problèmes multichamps tels que des problèmes en milieu poreux ou la résolution de l'équation de Stokes réalisée par TOULOU GOUSSOU et ROUX [2014]. Un complément de Schur hybride non-symétrique noté  $\mathbf{H}^\diamond$  peut ainsi être défini et le problème hybride peut se mettre sous la forme :

$$\begin{bmatrix} \mathbf{H}_{pp}^\diamond & \mathbf{H}_{pd}^\diamond \\ \mathbf{H}_{dp}^\diamond & \mathbf{H}_{dd}^\diamond \end{bmatrix} \begin{bmatrix} \mathbf{u}_p^\diamond \\ \boldsymbol{\lambda}_d^\diamond \end{bmatrix} = \begin{bmatrix} \mathbf{b}_p^\diamond \\ \mathbf{b}_d^\diamond \end{bmatrix} + \begin{bmatrix} \boldsymbol{\lambda}_p^\diamond \\ \mathbf{u}_d^\diamond \end{bmatrix} \quad (1.18a)$$

$$\begin{bmatrix} \mathbf{B}_p^\diamond & 0 \\ 0 & \mathbf{A}_d^\diamond \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}_p^\diamond \\ \mathbf{u}_d^\diamond \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{u}_p^\diamond \\ \boldsymbol{\lambda}_d^\diamond \end{bmatrix} = \begin{bmatrix} \mathbf{B}_p^{\diamond T} & 0 \\ 0 & \mathbf{A}_d^{\diamond T} \end{bmatrix} \begin{bmatrix} \mathbf{u}_p \\ \boldsymbol{\lambda}_d \end{bmatrix} \quad (1.18b)$$

Une autre classe de méthodes consiste à recondenser le système (1.18a) sur les DDLs primaux ou duaux. En particulier, en choisissant intelligemment les DDLs primaux, il est possible de bloquer tous les mouvements de corps rigides qui imposent l'utilisation de la pseudo-inverse de l'approche duale; c'est la méthode appelée *FETI-DP* et a pu être développée par LESOINNE et PIERSON [1999] et FARHAT et coll. [2000] ou encore FARHAT et coll. [2001]. Son homologue primale est la méthode *BDD by Constrained (BDD-C)* de DOHRMANN [2003].

### 1.1.3 Résolution des approches primale et duale

Les précédents systèmes conduisent à formuler des problèmes posés sur l'interface globale  $\Gamma_A$  ou  $\Gamma_B$  (pour les approches primale, duale, hybride et recondensées) ou sur l'interface locale  $\Gamma^{(s)}$  (pour l'approche mixte). Dans ces deux cas, l'assemblage explicite d'opérateurs locaux à un niveau global n'est pas accessible et donc l'utilisation de solveurs directs n'est pas envisageable en raison d'un volume de données trop important à échanger. C'est la raison pour laquelle ces problèmes à l'interface sont résolus à l'aide de solveurs itératifs tels que ceux de Krylov. En effet, ils nécessitent uniquement des produits scalaires (vecteur-vecteur) ou des produits matrice-vecteur qui ont la particularité d'être facilement parallélisables grâce à la structure additive des opérateurs. De plus, les sous domaines peuvent être utilisés comme des boîtes noires permettant de réaliser les résolutions locales indépendantes. En effet, seules les quantités d'interface varient et les seules quantités d'intérêt sont les actions de ces opérateurs sur la quantité d'interface.

Par la suite, seuls les aspects liés aux approches primale et duale seront mis en avant.

#### 1.1.3.1 Préconditionnement

Les méthodes de décomposition de domaine font usage d'un solveur itératif dont les performances sont un point critique de ces méthodes.

Or, celles-ci sont fortement liées aux propriétés spectrales du complément de Schur (primal ou dual). L'utilisation de préconditionneurs (permettant d'améliorer le *conditionnement* du problème), a donc pour but d'améliorer significativement le taux de convergence de ces solveurs itératifs. Il peut être aussi signalé que la résolution du problème d'interface ((1.8) ou (1.10)), en comparaison du problème global (1.1), est plus simple. D'une part, grâce à la taille réduite du problème à l'interface et le développement de préconditionneurs efficaces, mais surtout du fait d'un *conditionnement* plus favorable. En effet, d'après ROUX [2003], le complément de Schur (primal ou dual) a la particularité d'avoir un *conditionnement* – qui varie en  $1/h$  – plus faible que le *conditionnement* de la matrice de raideur globale  $\mathbf{K}$  qui, lui, varie en  $1/h^2$  avec  $h$  la taille caractéristique des éléments de la discrétisation de  $\Omega$ .

Les solveurs de Krylov possèdent des propriétés mathématiques de convergence permettant d'évaluer l'influence du *conditionnement* sur la vitesse de convergence. Par exemple, pour le *CG*, appliqué à la résolution du système  $\mathbf{K}\mathbf{x} = \mathbf{b}$ , où  $\mathbf{x}_k$  est la solution à l'itération  $k$  :

$$\|\mathbf{x} - \mathbf{x}_k\|_{\mathbf{K}} \leq 2 \left( \frac{\sqrt{\kappa(\mathbf{K})} - 1}{\sqrt{\kappa(\mathbf{K})} + 1} \right)^k \|\mathbf{x} - \mathbf{x}_0\|_{\mathbf{K}},$$

avec  $\kappa(\bullet)$ , le *conditionnement* d'un opérateur. L'intérêt de trouver un bon préconditionneur à  $\mathbf{K}$  est donc bien visible. En effet, si  $\kappa(\mathbf{K})$  est proche de 1, alors,  $\mathbf{x}_k$  approxime bien  $\mathbf{x}$ , même lorsque  $k$  est faible. Préconditionner un système revient à multiplier ce système par une matrice  $\mathbf{M}$ , qui est appelée « préconditionneur », telle que :

$$\kappa(\mathbf{MK}) \ll \kappa(\mathbf{K}).$$

Il faut donc pouvoir trouver une matrice  $\mathbf{M}$  qui soit proche de l'inverse de  $\mathbf{K}$  ( $\mathbf{M} \approx \mathbf{K}^{-1}$ ) si l'on souhaite pouvoir faire converger le solveur rapidement. La difficulté reste de trouver une bonne approximation de l'inverse de  $\mathbf{K}$  à un coût raisonnable. On modifie donc le système initial pour résoudre le système suivant :

$$\mathbf{MK}\mathbf{x} = \mathbf{M}\mathbf{b}$$

qui est, en principe, plus simple à résoudre itérativement du fait de son meilleur **conditionnement**.

Dans le contexte des méthodes de décomposition de domaine primale et duale exposées ici, un bon préconditionneur du problème d'interface peut être obtenu à moindre coût. En effet, la dualité qui existe entre les deux approches et la relation entre les compléments de Schur locaux va permettre de définir des préconditionneurs de bonne qualité. Cette relation, comme cela a pu être vu dans la section 1.1.2.3, étant :

$$\mathbf{S}_d^{(s)} = \mathbf{S}_p^{(s)+},$$

permet de trouver une bonne approximation de l'inverse de l'opérateur de Schur global.

**Approche duale** On rappelle ici, que la résolution du problème de point selle (1.10) par un solveur itératif nécessite de trouver un préconditionneur optimal pour l'opérateur  $\mathbf{S}_d$  qui est un assemblage de contributions locales  $\mathbf{S}_p^{(s)+}$  par les opérateurs  $\mathbf{B}^{(s)}$ . Pour construire un préconditionneur  $\tilde{\mathbf{S}}_d^{-1}$  de  $\tilde{\mathbf{S}}_d$ , il suffit de faire un assemblage pondéré des inverses de  $\mathbf{S}_p^{(s)+}$ , soit un assemblage pondéré de  $\mathbf{S}_p^{(s)}$  appelé préconditionneur de Dirichlet et correspondant à :

$$\tilde{\mathbf{S}}_d^{-1} = \tilde{\mathbf{B}} \diamond \mathbf{S}_p \diamond \tilde{\mathbf{B}}^T, \quad (1.19)$$

avec  $\tilde{\mathbf{B}} \diamond$  l'opérateur de pondération (ou de *scaling*) dual. Cet opérateur peut être défini en utilisant un opérateur symétrique défini positif  $\mathbf{M} \diamond$  de la manière décrite par Klawonn et Widlund [2001] :

$$\tilde{\mathbf{B}} \diamond = \left( \mathbf{B} \diamond \mathbf{M} \diamond^{-1} \mathbf{B} \diamond^T \right)^+ \mathbf{B} \diamond \mathbf{M} \diamond^{-1}.$$

Cette forme de  $\tilde{\mathbf{B}} \diamond$  est une inverse généralisée de  $\mathbf{B}$  satisfaisant donc la relation :

$$\begin{cases} \mathbf{B} \diamond \tilde{\mathbf{B}} \diamond^T \mathbf{B} \diamond = \mathbf{B} \diamond \\ \tilde{\mathbf{B}} \diamond \mathbf{B} \diamond^T \tilde{\mathbf{B}} \diamond = \tilde{\mathbf{B}} \diamond \end{cases}.$$

L'opérateur  $\mathbf{M} \diamond$  utilisé dans ce type de préconditionneur permet de rendre compte d'éventuelles hétérogénéités qui pourraient exister entre les sous-domaines. Différents choix pour cet opérateur ont été donnés par Rixen et Farhat [1999] :

- $\mathbf{M} \diamond = \mathbf{1} \diamond$  pour les structures homogènes;
- $\mathbf{M} \diamond = \text{diag}(\mathbf{K}_{bb}) \diamond$  pour les structures hétérogènes compressibles;
- $\mathbf{M} \diamond = \boldsymbol{\mu} \diamond$  pour les structures hétérogènes incompressibles (où  $\boldsymbol{\mu} \diamond$  est la matrice diagonale contenant les modules de cisaillement des DDLs d'interface, introduite par Gosselet et coll. [2002]).

#### Remarque 1.6

Pour rendre le calcul du préconditionneur moins coûteux, il est possible de modifier l'opérateur  $\mathbf{S}_p \diamond$  en des quantités plus simples à calculer, ne nécessitant pas de calcul d'inverse :

- $\mathbf{S}_p \diamond \approx \mathbf{K}_{bb} \diamond$  qui conduit au préconditionneur *lumped*;
- $\mathbf{S}_p \diamond \approx \text{diag}(\mathbf{K}_{bb}) \diamond$  qui conduit au préconditionneur *superlumped*.

**Approche primale** À l'instar de l'approche duale, un préconditionneur au problème primal (1.8) est recherché, qui est une approximation de  $\mathbf{S}_p$ , il sera noté  $\tilde{\mathbf{S}}_p^{-1}$ . Cet opérateur est un assemblage de quantités locales  $\mathbf{S}_p^{(s)}$  assemblées à l'aide des opérateurs  $\mathbf{A}^{(s)}$ . Le préconditionneur optimal

est aussi un assemblage pondéré de contributions duales appelé préconditionneur Neumann-Neumann et défini tel que :

$$\tilde{\mathbf{S}}_p^{-1} = \tilde{\mathbf{A}}^\diamond \mathbf{S}_d^\diamond \tilde{\mathbf{A}}^{\diamond T}. \quad (1.20)$$

Les opérateurs de pondération primaux sont définis de manière similaire à leurs homologues duaux par [KLAWONN et WIDLUND \[2001\]](#) :

$$\tilde{\mathbf{A}}^\diamond = \left( \mathbf{A}^\diamond \mathbf{M}^\diamond \mathbf{A}^\diamond T \right)^{-1} \mathbf{A}^\diamond \mathbf{M}^\diamond,$$

et respectent bien la propriété :

$$\mathbf{A}^\diamond \tilde{\mathbf{A}}^{\diamond T} = \mathbf{1}_{\Gamma_A}.$$

Ainsi, l'opérateur  $\mathbf{M}^\diamond$  est défini exactement de la même façon que pour le préconditionneur de Dirichlet. Par contre, contrairement à ce qui peut être fait dans la remarque 1.6 pour l'approche duale, le préconditionneur Neumann-Neumann ne peut être allégé de la même façon.

Les opérateurs d'assemblages précédemment introduits vérifient alors une propriété de supplémentarité de [RIXEN et FARHAT \[1999\]](#) :

$$\tilde{\mathbf{A}}^{\diamond T} \mathbf{A}^\diamond + \tilde{\mathbf{B}}^{\diamond T} \mathbf{B}^\diamond = \mathbf{1}^\diamond. \quad (1.21)$$

#### Remarque 1.7

Le préconditionneur introduit dans l'équation (1.20) consiste à résoudre des problèmes de Neumann de part et d'autre de l'interface, c'est pour cela qu'il est appelé préconditionneur Neumann-Neumann. Il en est de même pour celui de l'approche duale de l'équation (1.19) qui est appelé préconditionneur de Dirichlet.

### 1.1.3.2 Problèmes grossiers

Des problèmes grossiers obligatoires ou optionnels font partie intégrante des méthodes de décomposition de domaine. Dans le cas de l'approche duale, il n'est pas possible de s'en passer car ils permettent de régulariser le problème en évitant d'exciter ces modes rigides. De plus, leur adjonction permet de transmettre dès l'initialisation du solveur des informations de type « grande longueur d'onde » au sein de la structure. Ainsi, un domaine loin des conditions aux limites peut en avoir connaissance sans attendre qu'elles soient transmises de proche en proche au cours des itérations. Ces problèmes grossiers sont fondés sur l'utilisation des modes rigides des structures flottantes, et donc sur la base  $\mathbf{R}_b^\diamond$  du noyau de  $\mathbf{S}_p^\diamond$ . Ces problèmes grossiers permettent ainsi une forte accélération de la transmission de l'information conduisant à une *extensibilité* accrue de la méthode.

**Approche duale** La seconde ligne du système (1.10), qui est une équation d'admissibilité de la solution  $\lambda_B$  ne peut être omise. En pratique, cette contrainte est mise en œuvre à l'aide d'une stratégie de projection et d'initialisation au sein d'une méthode de type Krylov contraint. Une projection est ainsi réalisée à l'aide du projecteur  $\mathbf{P}_d$  sur l'espace orthogonal à  $\mathbf{Vect}(\mathbf{G}_d^\diamond)$  et l'initialisation consiste à capter la partie de la solution liée au second membre  $\mathbf{e}^\diamond$ . L'inter-effort est donc recherché de la forme  $\lambda_B = \lambda_{B,0} + \mathbf{P}_d \tilde{\lambda}_B$  de manière à ce que  $\mathbf{G}_d^{\diamond T} \lambda_{B,0} = \mathbf{e}^\diamond$  et  $\mathbf{G}_d^{\diamond T} \mathbf{P}_d = 0$ .

Ces conditions sont respectées en choisissant un couple projecteur-initialisation de la forme :

$$\begin{cases} \mathbf{P}_d &= \mathbf{1} - \mathbf{Q}_d \mathbf{G}_d^\diamond \left( \mathbf{G}_d^{\diamond T} \mathbf{Q}_d \mathbf{G}_d^\diamond \right)^{-1} \mathbf{G}_d^{\diamond T} \\ \lambda_{B,0} &= \mathbf{Q}_d \mathbf{G}_d^\diamond \left( \mathbf{G}_d^{\diamond T} \mathbf{Q}_d \mathbf{G}_d^\diamond \right)^{-1} \mathbf{e}^\diamond \end{cases}, \quad (1.22)$$

avec  $\mathbf{Q}_d$  une matrice symétrique définie positive. Pour calculer l'amplitude des modes rigides  $\alpha^\diamond$ , il faut ensuite repartir de la première ligne du système (1.10) et isoler  $\alpha^\diamond$ , donnant l'équation :

$$\alpha^\diamond = \left( \mathbf{G}_d^{\diamond T} \mathbf{Q}_d \mathbf{G}_d^\diamond \right)^{-1} \mathbf{G}_d^{\diamond T} \mathbf{Q}_d (\mathbf{b}_d - \mathbf{S}_d \lambda_B).$$

Le problème résolu à l'aide du projecteur peut se mettre sous la forme :

$$\mathbf{P}_d^T \mathbf{S}_d \mathbf{P}_d \lambda_b = \mathbf{P}_d^T (\mathbf{b}_d - \mathbf{S}_d \lambda_{b0}).$$

C'est ce problème qui sera résolu itérativement à l'aide du solveur de Krylov comme décrit dans l'algorithme A.4.

**Approche primale** Pour l'approche primale, il n'existe pas de problème grossier naturel ressortissant de la formulation du problème à l'interface contrairement à l'approche duale mais un problème grossier optionnel qui intervient au niveau de l'étape de préconditionnement peut y être adjoint (MANDEL [1993], LE TALLEC [1994], GOSSELET et REY [2006]). En effet, il faut s'assurer que les vecteurs qui sont appliqués à  $\tilde{\mathbf{S}}_d^{-1}$  soient bien orthogonaux à l'espace des modes rigides pondérés (il y a un produit entre  $\mathbf{S}_d^\diamond \tilde{\mathbf{A}}^{\diamond T}$  et un vecteur). Cette étape de préconditionnement se fait à la ligne du calcul du résidu préconditionné dans l'algorithme A.3 représenté par le produit  $\tilde{\mathbf{S}}_d^{-1} \mathbf{r}_i$  avec  $\mathbf{r}_i$  le résidu du problème (1.8). Ainsi, une condition d'orthogonalité avec l'espace des modes rigides pondérés  $\tilde{\mathbf{G}}_p^\diamond = \tilde{\mathbf{A}}^\diamond \mathbf{R}_b^\diamond$  est nécessaire pour permettre ce préconditionnement :

$$\tilde{\mathbf{G}}_p^{\diamond T} \mathbf{r}_i = 0.$$

En procédant de la même façon que pour l'approche duale, en introduisant un projecteur  $\mathbf{P}_p$  et en choisissant un champ solution de la forme :  $\mathbf{u}_A = \mathbf{u}_{A,0} + \mathbf{P}_p \tilde{\mathbf{u}}_{Ab}$ , des conditions sur ce projecteur et cette initialisation peuvent être trouvées.

$$\tilde{\mathbf{G}}_p^{\diamond T} \mathbf{r}_i = \tilde{\mathbf{G}}_p^{\diamond T} (\mathbf{b}_p - \mathbf{S}_p \mathbf{u}_{A,0} - \mathbf{S}_p \mathbf{P}_p \tilde{\mathbf{u}}_A) = 0$$

Cela conduit aux conditions suivantes :

$$\begin{cases} \tilde{\mathbf{G}}_p^{\diamond T} \mathbf{r}_0 = 0 \\ \tilde{\mathbf{G}}_p^{\diamond T} \mathbf{S}_p \mathbf{P}_p = 0 \end{cases},$$

qui, reformulées donnent les expressions :

$$\begin{cases} \mathbf{u}_{A,0} = \left( \tilde{\mathbf{G}}_p^{\diamond T} \mathbf{S}_p \tilde{\mathbf{G}}_p^\diamond \right)^{-1} \tilde{\mathbf{G}}_p^{\diamond T} \mathbf{S}_p \mathbf{b}_p \\ \mathbf{P}_p = \mathbf{1} - \tilde{\mathbf{G}}_p^\diamond \left( \tilde{\mathbf{G}}_p^{\diamond T} \mathbf{S}_p \tilde{\mathbf{G}}_p^\diamond \right)^{-1} \tilde{\mathbf{G}}_p^{\diamond T} \mathbf{S}_p \end{cases}.$$

Ainsi, la valeur de  $\tilde{\mathbf{u}}_A$  peut être obtenue en résolvant itérativement le problème projeté suivant :

$$\begin{aligned} \mathbf{S}_p \mathbf{P}_p \tilde{\mathbf{u}}_A &= \mathbf{b}_p - \mathbf{S}_p \mathbf{u}_{A,0} \\ \mathbf{P}_p^T \mathbf{S}_p \mathbf{P}_p \tilde{\mathbf{u}}_A &= \mathbf{P}_p^T (\mathbf{b}_p - \mathbf{S}_p \mathbf{u}_{A,0}) \\ \mathbf{P}_p^T \mathbf{S}_p \mathbf{P}_p \tilde{\mathbf{u}}_A &= \mathbf{P}_p^T \mathbf{b}_p. \end{aligned}$$

La résolution de cette équation se réalise en pratique à l'aide d'un solveur itératif de Krylov augmenté comme décrit dans l'algorithme A.5.

**Intérêt du préconditionnement et de l'espace grossier** Grâce à tout cela et comme cela a été montré mathématiquement par FARHAT et coll. [1994] ou encore Klawonn et Widlund [2001] ces méthodes ont des propriétés d'extensibilité optimale lorsque le problème est équipé de ces meilleurs préconditionneurs et de leur problème grossier car le conditionnement  $\kappa$  prend la forme :

$$\kappa \approx C \left( 1 + \log \left( \frac{H}{h} \right) \right)^2, \quad (1.23)$$

où  $h$  est la taille des éléments,  $H$  la taille du sous-domaine et  $C$  une constante dépendant de la forme des sous-domaines, du matériau, ... Ainsi, ce conditionnement ne dépend pas du nombre de sous-domaines et l'extensibilité de la résolution est assurée. L'approche primale munie de son préconditionneur et de son problème grossier est connue sous l'appellation *BDD*. De la même façon; l'approche duale équipée de son préconditionneur ainsi que de son problème grossier porte le nom de *FETI*. Ils consistent en des algorithmes de *CG* augmenté (ou contraint) préconditionnés et projetés. Il faut aussi noter que dans ces algorithmes une réorthogonalisation totale des  $\mathbf{d}_i$  est effectuée malgré la propriété d'orthogonalité de  $\mathbf{d}_{i+1}$  avec  $\mathbf{d}_i$  dans un *CG*. Cette réorthogonalisation totale s'explique par le fait que, numériquement, l'orthogonalité par rapport à toutes les directions de descentes précédentes a tendance à être perdue au cours des itérations.

---

**Algorithme 1.1 :** Méthode *BDD* couplée avec un *CG*

---

**Initialisation :**

On pose un projecteur :  $\mathbf{P}_p = \mathbf{1} - \tilde{\mathbf{G}}_p \left( \tilde{\mathbf{G}}_p^T \mathbf{S}_p \tilde{\mathbf{G}}_p \right)^{-1} \tilde{\mathbf{G}}_p^T \mathbf{S}_p$  ;

On initialise la solution :  $\mathbf{u}_{A,0} = \tilde{\mathbf{G}}_p \left( \tilde{\mathbf{G}}_p^T \mathbf{S}_p \tilde{\mathbf{G}}_p \right)^{-1} \tilde{\mathbf{G}}_p^T \mathbf{b}_p$  ;

On initialise le résidu :  $\mathbf{r}_0 = \mathbf{P}_p^T (\mathbf{b}_p - \mathbf{S}_p \mathbf{u}_{A,0})$  ;

On préconditionne le résidu :  $\mathbf{z}_0 = \tilde{\mathbf{S}}_p^{-1} \mathbf{r}_0$  ;

On pose :  $\mathbf{d}_0 = \mathbf{z}_0$  ;

**tant que**  $\frac{\mathbf{r}_i^T \mathbf{z}_i}{\mathbf{r}_0^T \mathbf{z}_0} > \varepsilon$  **faire**

$\mathbf{p}_i = \mathbf{P}_p^T \mathbf{S}_p \mathbf{d}_i$  ;

    Calcul du pas optimal :  $\alpha_i = \frac{\mathbf{r}_i^T \mathbf{z}_i}{\mathbf{d}_i^T \mathbf{p}_i}$  ;

    Calcul de la nouvelle solution :  $\mathbf{u}_{A,i+1} = \mathbf{u}_{A,i} + \alpha_i \mathbf{d}_i$  ;

    Calcul du nouveau résidu :  $\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i \mathbf{p}_i$  ;

    Calcul du résidu préconditionné :  $\mathbf{z}_{i+1} = \mathbf{P}_p \tilde{\mathbf{S}}_p^{-1} \mathbf{r}_{i+1}$  ;

**pour**  $0 \leq j \leq i$  **faire** Réorthogonalisation de la base de recherche

$\beta_i^j = -\frac{\mathbf{z}_{i+1}^T \mathbf{p}_j}{\mathbf{d}_j^T \mathbf{p}_j}$  ;

**fin**

    Calcul de la nouvelle direction de recherche :  $\mathbf{d}_{i+1} = \mathbf{z}_{i+1} + \sum_{j < i} \beta_i^j \mathbf{d}_j$  ;

**fin**

**Calcul de la solution convergée dans les sous-domaines :**

$\mathbf{u}_A = \mathbf{u}_{A,0} + \mathbf{P}_p \mathbf{u}_{A,i}$  ;

$\mathbf{u}_b^\diamond = \mathbf{A}^\diamond \mathbf{u}_A$  ;

$\mathbf{u}_i^\diamond = \mathbf{K}_{ii}^\diamond^{-1} \left( \mathbf{f}_i^\diamond - \mathbf{K}_{ib}^\diamond \mathbf{u}_b^\diamond \right)$  ;

---

**Algorithme 1.2 : Méthode *FETI* couplée avec un *CG***
**Initialisation :**

 On pose un projecteur :  $\mathbf{P}_d = \mathbf{1} - \mathbf{Q}_d \mathbf{G}_d (\mathbf{G}_d^T \mathbf{Q}_d \mathbf{G}_d)^{-1} \mathbf{G}_d^T$  ;

 On initialise la solution :  $\lambda_{bB,0} = \mathbf{Q}_d \mathbf{G}_d (\mathbf{G}_d^T \mathbf{Q}_d \mathbf{G}_d)^{-1} \mathbf{e}^\diamond$  ;

 On initialise le résidu :  $\mathbf{r}_0 = \mathbf{P}_d^T (\mathbf{b}_d - \mathbf{S}_d \lambda_{B,0})$  ;

 On préconditionne le résidu :  $\mathbf{z}_0 = \tilde{\mathbf{S}}_d^{-1} \mathbf{r}_0$  ;

 On pose :  $\mathbf{d}_0 = \mathbf{z}_0$  ;

**tant que**  $\frac{\mathbf{r}_i^T \mathbf{r}_i}{\mathbf{r}_0^T \mathbf{r}_0} > \varepsilon$  **faire**
 $\mathbf{p}_i = \mathbf{P}_d^T \mathbf{S}_d \mathbf{d}_i$  ;

 Calcul du pas optimal :  $\alpha_i = \frac{\mathbf{r}_i^T \mathbf{z}_i}{\mathbf{d}_i^T \mathbf{p}_i}$  ;

 Calcul de la nouvelle solution :  $\lambda_{B,i+1} = \lambda_{B,i} + \alpha_i \mathbf{d}_i$  ;

 Calcul du nouveau résidu :  $\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i \mathbf{p}_i$  ;

 Calcul du résidu préconditionné :  $\mathbf{z}_{i+1} = \mathbf{P}_d \tilde{\mathbf{S}}_d^{-1} \mathbf{r}_{i+1}$  ;

**pour**  $0 \leq j \leq i$  **faire** réorthogonalisation de la base de recherche

 $\beta_i^j = -\frac{\mathbf{z}_{i+1}^T \mathbf{p}_j}{\mathbf{d}_j^T \mathbf{p}_j}$  ;

**fin**

 Calcul de la nouvelle direction de recherche :  $\mathbf{d}_{i+1} = \mathbf{z}_{i+1} + \sum_{j \leq i} \beta_i^j \mathbf{d}_j$  ;

**fin**
**Calcul de la solution convergée puis du déplacement :**
 $\lambda_B = \lambda_{B,0} + \mathbf{P}_d \lambda_{B,i}$  ;

 $\alpha^\diamond = (\mathbf{G}_d^T \mathbf{Q}_d \mathbf{G}_d)^{-1} \mathbf{G}_d^T \mathbf{Q}_d \mathbf{r}_i$  ;

 $\mathbf{u}^\diamond = \mathbf{K}^\diamond + \left( \mathbf{f}^\diamond + \mathbf{t}^\diamond \mathbf{B}^\diamond \lambda_B \right) + \mathbf{R}^\diamond \alpha^\diamond$  ;

**1.1.3.3 Évolution du conditionnement pour divers problèmes**

Ces solveurs, bien que très efficaces sur des cas simples, se heurtent à de plus grandes difficultés numériques lorsqu'il s'agit de résoudre des problèmes moins académiques et notamment dans les cas où la structure complète est hétérogène. Dans le cas général, les sous-domaines générés, permettant de formuler un problème à l'interface, n'ont pas forcément une forme régulière et compacte. Dans cette section, l'influence néfaste de ces paramètres sur la convergence va être discutée au regard de ce qui existe dans la littérature et les solutions qui ont été apportées pour la résolution de ces problèmes seront exposées.

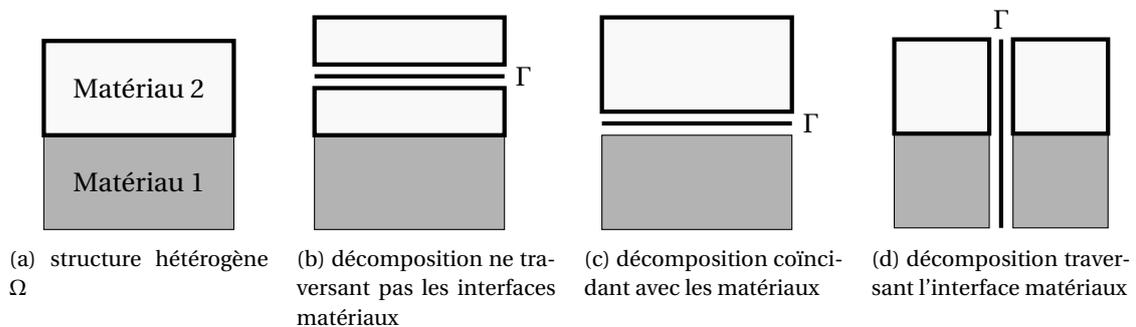
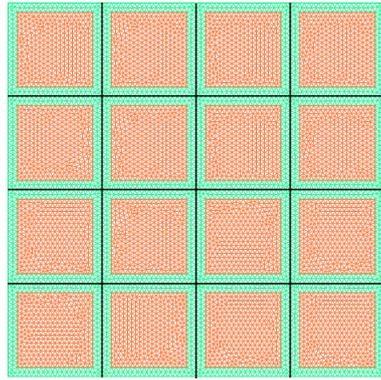
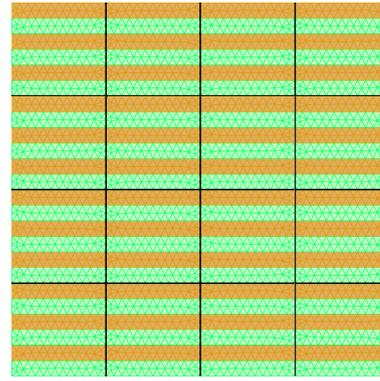


FIGURE 1.4 – Types de découpages hétérogènes rencontrés dans un contexte sous-structuré (reproduite de GOSSELET et REY [2006])

**Problèmes hétérogènes** Lors du partitionnement d'une structure hétérogène  $\Omega$ , comportant plusieurs matériaux différents, ses constituants se retrouvent distribués sur des sous-domaines



(a) hétérogénéités à l'intérieur des sous-domaines



(b) hétérogénéités traversant l'interface

FIGURE 1.5 – Influence de la répartition des hétérogénéités dans une sous-structuration (extrait de GOSSELET et coll. [2015])

Configuration	homogène	hétérogène $10^5$
Inclusions (figure 1.5a)	15	13
Composite (figure 1.5b)	14	89

TABLEAU 1.1 – Nombre d'itérations nécessaires pour faire diminuer le résidu FETI d'un rapport  $10^6$  pour des problèmes hétérogènes (extrait de GOSSELET et coll. [2015])

$\Omega^{(s)}$  distincts. Le placement des interfaces peut se faire de plusieurs façons (voir figure 1.4) : des interfaces entre sous-domaines ne coupant pas d'interface physique entre matériaux (figure 1.4b), une interface coïncidant avec cette interface matériau (figure 1.4c) ou encore une interface traversant complètement une interface entre matériaux (figure 1.4d).

Le résultat (1.23), prouvé par KLAUWONN et WIDLUND [2001], suppose que chaque sous-domaine est homogène et qu'il n'existe pas de saut de coefficient matériau à l'intérieur des sous-domaines (cas de la figure 1.4c). C'est bien ce qui a pu être observé par différents auteurs comme PECHSTEIN et SCHEICHL [2011], RIXEN et FARHAT [1999] ou encore KLAUWONN et RHEINBACH [2007] pour *FETI-DP* où l'utilisation de sous-domaines hétérogènes conduit à une détérioration forte de la vitesse de convergence. Cette constatation est d'autant plus vraie lorsque l'hétérogénéité traverse une interface comme c'est le cas sur la figure 1.4d. Pour illustrer l'influence de la position des hétérogénéités vis-à-vis des interfaces, des essais numériques menés par GOSSELET et coll. [2015] sont repris ici et reproduits sur la figure 1.5 et le tableau 1.1 où l'hétérogénéité est représenté par une variation de module de Young entre les différents matériaux d'un facteur  $10^5$ . Sur ce tableau, le nombre d'itérations à convergence est multiplié par 6 environ uniquement à cause des hétérogénéités traversant l'interface en comparaison de la même résolution mais avec un matériau homogène. Par comparaison, une structure comme celle de la figure 1.5a, où les hétérogénéités restent confinées à l'intérieur des sous-domaines, conserve un nombre d'itérations à convergence stable.

**Interfaces irrégulières** Le fait que l'interface entre les sous-domaines soit irrégulière est un autre facteur de détérioration du conditionnement. En effet, KLAUWONN et coll. [2008] montrent qu'avec des interfaces fractales entre les sous-domaines (voir figure 1.6), le conditionnement du problème d'interface préconditionné est de plus en plus mauvais lorsque la longueur de l'interface augmente (ou que l'interface est de plus en plus irrégulière). Cela peut être interprété mécaniquement par le fait que des sous-domaines deviennent artificiellement plus souples aux endroits où ils ont une épaisseur plus petite comme l'a montré GOSSELET [2017]. D'autres auteurs comme MANDEL et SOUSEDÍK [2007], ont développé une méthode pour améliorer le mauvais conditionnement dû à des interfaces irrégulières en ajoutant des DDLs grossiers automatiquement aux endroits problématiques. Ce type de problème n'est pas seulement un problème rencontré dans les méthodes

sans recouvrement de type Schur; il existe aussi pour des méthodes de type Schwarz comme cela a pu être illustré par DOHRMANN et coll. [2008]; EIKELAND et coll. [2017]. Dans le cadre de la méthode *FETI*, des essais numériques menées par GOSSELET et coll. [2015] ont montré que les interfaces irrégulières générées par un logiciel de partitionnement de graphe de manière automatique conduisait à un nombre d'itération à convergence 3 fois supérieur à une décomposition régulière appliquée à une structure du type poutre décomposée en 9 tronçons (même problème que celui de la figure 1.7a mais avec des sous-domaines de formes différentes).

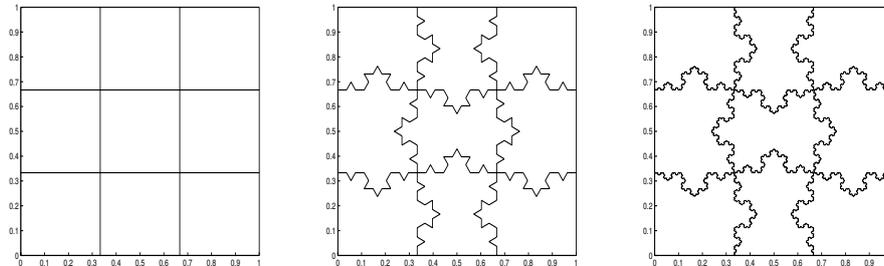


FIGURE 1.6 – Sous-domaines avec des interfaces fractales (extrait de K LAWONN et coll. [2008])

**Sous-domaines élancés** Une autre caractéristique géométrique qui pénalise le conditionnement des méthodes de décomposition de domaine est l'élancement ou le rapport d'aspect (*aspect ratio* en anglais) des sous-domaines. En effet, comme le montre GOSSELET et coll. [2015], ou EIKELAND et coll. [2017] pour les méthodes de Schwarz, lorsque les sous-structures sont élancées, le conditionnement du problème se détériore. Le tableau 1.2 présente le nombre d'itérations à convergence pour une structure homogène de type poutre qui a été dilatée dans la direction transverse d'un facteur allant de 1/5 à 10. On observe que lorsque les interfaces sont proches (relativement à la taille du sous-domaine), le nombre d'itérations à convergence augmente assez fortement.

élancement	1/5	1	5	10
# itérations <i>FETI</i>	5	6	17	29

TABEAU 1.2 – Évolution du nombre d'itérations à convergence pour une poutre homogène avec l'élancement variable(extrait de GOSSELET et coll. [2015]). Problèmes de la figure 1.7.

### 1.1.3.4 Méthodes avancées

Les baisses de performance observées dans la partie précédente sont dues à une stratégie de préconditionnement non adaptée qui ne parvient pas à prendre en compte correctement les caractéristiques particulières du problème d'interface. Ces caractéristiques géométriques modifient les propriétés spectrales de l'opérateur préconditionné. En effet, l'assemblage d'informations locales dans le préconditionnement – et notamment les pondérations  $\tilde{\mathbf{A}}^{\diamond}$  et  $\tilde{\mathbf{B}}^{\diamond}$  qui apparaissent dans les préconditionneurs des approches primale (1.20) et duale (1.19) – ne permettent pas d'approximer comme il le faudrait ces informations globales. Une idée émergente a été de modifier ces propriétés spectrales en supprimant la partie du spectre de l'opérateur préconditionné qui cause un mauvais conditionnement. Certaines méthodes sont basées sur un analyse du système *a priori*, en ajoutant un coût de calcul tandis que d'autres permettent de capter cette partie du spectre à la volée.

**Méthodes *a priori*** La première classe de méthodes s'attache à trouver les modes locaux de la structure pénalisant le conditionnement du solveur. Ces méthodes sont regroupées sous le nom de *Generalized Eigenproblems in the Overlaps (GenEO)* et ont été développées par SPILLANE et coll.

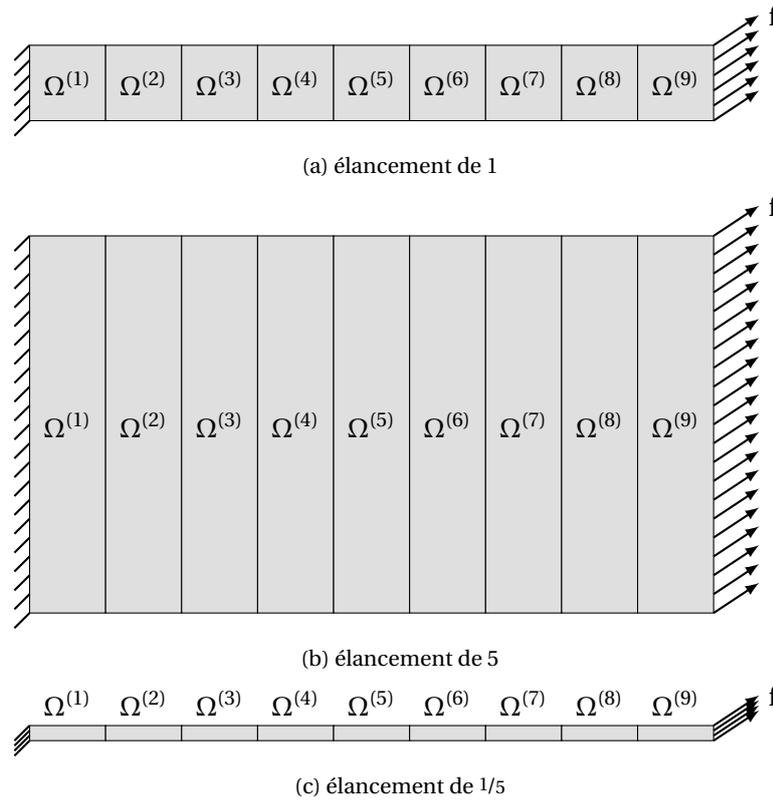


FIGURE 1.7 – Différents problèmes résolus dans le tableau 1.2

[2011]. Cette méthode a montré qu'il est possible de calculer ces modes à l'aide de problèmes aux valeurs propres généralisées formulés sur l'interface de chaque sous-domaine.

Par exemple, pour *FETI*, SPILLANE et RIXEN [2013] ont montré qu'il convient de résoudre des problèmes aux valeurs propres locaux, où il faut trouver les couples  $(\Lambda_k^{(s)}, \mathbf{v}_k^{(s)})$  où les  $\mathbf{v}_k^{(s)}$  sont les  $k$ -ièmes vecteurs propres et  $\Lambda_k^{(s)}$  les valeurs propres associées à ces  $k$ -ièmes vecteurs propres. Ce sont les éléments propres du problème aux valeurs propres généralisées suivant formulé sur l'interface locale :

$$\mathbf{S}_p^{(s)} \mathbf{v}_k^{(s)} = \Lambda_k^{(s)} \left( \mathbf{B}^{(s)T} \mathbf{S}_d^{-1} \mathbf{B}^{(s)} \right) \mathbf{v}_k^{(s)}.$$

Une fois les modes propres problématiques détectés, ils sont ôtés de la résolution grâce à des techniques d'augmentation lors de la résolution (de la même façon que dans *BDD*). Ainsi, un second niveau de projection est ajouté pour *FETI* tel que formulé dans les algorithmes de type Krylov-augmenté présentés en Annexe A.1.2.4, où l'espace d'augmentation est l'espace représenté par l'image de l'opérateur  $\mathbf{C}$  défini tel que chaque colonne de  $\mathbf{C}$  soit de la forme :

$$\mathbf{C}_k = \mathbf{P}_d \mathbf{S}_p \mathbf{B} \diamond \mathbf{v}_k \diamond.$$

Cette méthode porte le nom de *FETI-GenEO*.

L'utilisation de ces espaces grossiers *GenEO* n'est pas restreinte aux méthodes de Schur. En effet, cette méthode était initialement conçue par SPILLANE et coll. [2014] pour les méthodes avec recouvrement et a pu être utilisé par d'autres auteurs pour des méthode mixtes. Par exemple, on pourra citer les travaux d'OUMAZIZ et coll. [2020] où l'espace grossier de la méthode *LaTin* – consistant en les modes rigides et les modes d'extensions et de cisaillements – sont remplacés par des modes *GenEO* plus adaptés, ce qui permet de réduire significativement le nombre d'itérations *LaTin* nécessaires pour atteindre le même niveau de convergence.

**Méthodes « à la volée »** GOSSELET et coll. [2015] propose une généralisation du solveur à directions de recherches simultanées *Simultaneous FETI (S-FETI)* développé durant la thèse de RIXEN

[1997] pour 2 sous-domaines. Dans cette méthode, le préconditionnement se fait localement et la recombinaison des directions de recherche s'effectue automatiquement par l'algorithme de *CG* 1.3 pour trouver une nouvelle direction plus optimale qu'avec un préconditionnement classique. Ici, à l'étape de préconditionnement, les résidus ne sont plus sommés sur chaque sous-domaine, mais des blocs de résidus préconditionnés  $\mathbf{Z}_i$  sont formés. Grâce à cela, le solveur devient très robuste par rapport aux caractéristiques problématiques citées à la section précédente.

---

**Algorithme 1.3 :** *CG* de la méthode *S-FETI*


---

**Initialisation :**

On pose un projecteur :  $\mathbf{P}_d = \mathbf{1} - \mathbf{Q}_d \mathbf{G}_d (\mathbf{G}_d^T \mathbf{Q}_d \mathbf{G}_d)^{-1} \mathbf{G}_d^T$  ;

On initialise la solution :  $\lambda_{\mathbf{B},0} = \mathbf{Q}_d \mathbf{G}_d (\mathbf{G}_d^T \mathbf{Q}_d \mathbf{G}_d)^{-1} \mathbf{e}^\diamond$  ;

On initialise le résidu :  $\mathbf{r}_0 = \mathbf{P}_d^T (\mathbf{b}_d - \mathbf{S}_d \lambda_{\mathbf{B},0})$  ;

On multi-préconditionne le résidu :  $\mathbf{Z}_0 = [\dots, \tilde{\mathbf{B}}^{(s)} \mathbf{S}_p^{(s)} \tilde{\mathbf{B}}^{(s)T} \mathbf{r}_0, \dots]$  ;

On pose :  $\mathbf{D}_0 = \mathbf{Z}_0$  ;

**tant que**  $\frac{\mathbf{r}_i^T \mathbf{Z}_i \mathbf{1}}{\mathbf{r}_0^T \mathbf{Z}_0 \mathbf{1}} > \varepsilon$  **faire**

$\mathbf{Q}_i = \mathbf{P}_d^T \mathbf{S}_d \mathbf{D}_i$  ;

$\Delta_i = \mathbf{D}_i^T \mathbf{Q}_i$  ;

$\mathbf{Y}_i = \mathbf{Z}_i^T \mathbf{r}_i$  ;

    Calcul de la nouvelle solution :  $\lambda_{\mathbf{B},i+1} = \lambda_{\mathbf{B},i} + \mathbf{D}_i \Delta_i^+ \mathbf{Y}_i$  ;

    Calcul du nouveau résidu :  $\mathbf{r}_{i+1} = \mathbf{r}_i - \mathbf{Q}_i \Delta_i^+ \mathbf{Y}_i$  ;

    Calcul du résidu préconditionné :  $\mathbf{Z}_{i+1} = [\dots, \mathbf{P}_d \tilde{\mathbf{B}}^{(s)} \mathbf{S}_p^{(s)} \tilde{\mathbf{B}}^{(s)T} \mathbf{r}_{i+1}, \dots]$  ;

**pour**  $0 \leq j \leq i$  **faire** Réorthogonalisation de la base de recherche

        |  $\Phi_i^j = -\mathbf{Q}_j^T \mathbf{Z}_{i+1}$  ;

**fin**

    Calcul de la nouvelle direction de recherche :  $\mathbf{D}_{i+1} = \mathbf{Z}_{i+1} + \sum_{j \leq i} \mathbf{D}_j \Delta_j^+ \Phi_i^j$  ;

**fin**

**Calcul de la solution convergée puis du déplacement :**

$\lambda_{\mathbf{B}} = \lambda_{\mathbf{B},0} + \mathbf{P}_d \lambda_{\mathbf{B},i}$  ;

$\alpha^\diamond = (\mathbf{G}_d^T \mathbf{Q}_d \mathbf{G}_d)^{-1} \mathbf{G}_d^T \mathbf{Q}_d \mathbf{r}_i$  ;

$\mathbf{u}^\diamond = \mathbf{K}^{\diamond+} \left( \mathbf{f}^\diamond + \mathbf{t}^{\diamond+T} \mathbf{B}^{\diamond+T} \lambda_{\mathbf{B}} \right) + \mathbf{R}^\diamond \alpha^\diamond$  ;

---

Par contre cette approche reste coûteuse en termes de quantité de données générées, car à chaque itération il faut stocker autant de directions de recherche qu'il y a de sous-domaines. C'est pourquoi, pour réduire la quantité d'information à stocker, des algorithmes de *CG* Multipréconditionné Adaptatif ont été développés par SPILLANE [2016]. Cette idée a donné lieu au développement de la version adaptative de *S-FETI*, nommée *Adaptive MultiPreconditioned FETI (AMPFETI)* par BOVET et coll. [2017]. Ici, il s'agit de déterminer quelles directions de recherche générées ont une contribution significative pour être considérés séparément. Si ce n'est pas le cas, ces directions sont sommées de la même façon que dans la méthode *FETI* classique. Ainsi, la génération à chaque itération d'un nouveau nombre de directions de recherche inférieur compris entre 1 et  $N_s$  est possible.

### 1.1.3.5 Précisions sur la résolution des formulations mixtes

Cette partie a montré ce qu'il était nécessaire de réaliser pour permettre de résoudre itérativement les problèmes d'interfaces globaux des méthodes *FETI* (1.10) et *BDD* (1.8). Mais qu'en est-il de la résolution des systèmes issus de l'approche mixte ?

Le premier système (1.15) est un système qui est symétrique. Il peut-être résolu à l'aide du

même type de solveur que celui des approches primale et duale, c'est-à-dire un *CG*. Par contre, à l'inverse de ce qui est fait dans *FETI* et *BDD*, la définition d'un préconditionneur à ce problème n'est pas aussi directe que pour ces approches. En effet, l'opérateur de ce problème est un opérateur auquel à un effort appliqué, renvoie un déplacement. Pour trouver un préconditionneur à un tel opérateur, il faut trouver son opérateur dual auquel à partir d'un déplacement d'interface, renvoie un inter-effort. Malheureusement, il n'y a pas de stratégie de préconditionnement optimale pour cette approche à moins de se ramener à une formulation (*BDD* ou *FETI*) possédant des préconditionneurs comme a pu le faire *NEGRELLO et coll.* [2016] dans le cadre non linéaire.

Le second système (1.17), quant à lui, est non symétrique mais il a la particularité d'être homogène, c'est à dire qu'à partir d'un effort d'interface, il renvoie un autre effort d'interface. Ce problème-ci semble *a priori* bien mieux conditionné que le précédent car son conditionnement ne dépend plus que du choix de  $\mathbf{Q}_b$ . On pourrait envisager d'utiliser des solveurs basés sur la stationnarité. Par contre, du fait de la non-symétrie de l'opérateur, il n'est plus possible d'utiliser le *CG* en tant que solveur de Krylov, mais il faut utiliser des solveurs non-symétriques tels qu'*ORTHOMIN* de *VINSOME* [1976] (voir algorithme A.6) ou la méthode très connue de *SAAD et SCHULTZ* [1986] : *GMRES*.

Le choix de  $\mathbf{Q}_b$  étant critique pour la bonne convergence de la méthode, nous verrons au Chapitre 3 comment choisir cette impédance pour garantir une convergence rapide du solveur.

## 1.2 Pré-traitement d'une simulation parallèle

Pour réaliser une simulation en parallèle ou non, il faut être capable de générer la mise en données. Mais celle-ci est une étape qui peut devenir coûteuse dans le cas d'une discrétisation fine de la structure dont la simulation est désirée. Malheureusement, bien souvent dans l'industrie cette phase de préparation des données est réalisée avec des outils de maillage qui demeurent séquentiels et non distribués. Ainsi, la précision des maillages générés est limitée par les temps importants et les volumes importants de mémoire nécessaires qui découlent de cette génération de maillage. C'est pourquoi, il semble intéressant de rendre parallèle non seulement la simulation mais aussi la génération du maillage de la structure à simuler. Dans ces deux cas de figure (calcul et maillage), il est nécessaire de distribuer les données pour en disposer sur des machines parallèles grâce au partitionnement.

C'est pourquoi, nous allons voir dans un premier temps ce qui existe en termes de partitionnement de maillage, puis les différentes approches qui peuvent être envisagées pour générer un maillage à un niveau de raffinement souhaité en parallèle.

### 1.2.1 Stratégies de partitionnement

Le partitionnement de maillage est une étape essentielle pour la réalisation de calculs parallèles. Cette sous-structuration est réalisée à l'aide de méthodes de partitionnement de graphe. En effet, un maillage peut être ramené à un graphe de connectivité où les différentes entités du maillage peuvent être considérées en tant que sommet ou arêtes du graphe. On ne s'intéressera ici qu'au partitionnement au sens des sommets du graphe. Le problème de  $N_s$ -partitionnement d'un graphe consiste à venir découper un graphe non-orienté  $G = (V, E)$  avec  $V$  l'ensemble des sommets et  $E$  l'ensemble des arêtes, en un ensemble de  $N_s$  sous-ensembles disjoints de sommets de  $V$  qui sera noté  $(\pi_i)_{i \in \llbracket 1; N_s \rrbracket}$ .

#### Remarque 1.8

Dans le cadre d'une sous-structuration sans recouvrement géométrique, un graphe à considérer peut être le graphe tel que ses sommets  $V$  correspondent aux éléments du maillage et les arêtes  $E$  sont les connectivités des éléments entre eux; en 2D cela représente toutes les arêtes du maillage, et en 3D toutes les faces du maillage ou toutes les arêtes peuvent être choisies dans l'ensemble  $E$ .

Ces graphes peuvent être valués, c'est-à-dire qu'il est possible d'associer des poids à leurs nœuds et arêtes. Ainsi, la définition des applications qui à chaque sommet et à chaque arête associent leur poids strictement positif peuvent être définies. Elles sont notées  $w_V$  et  $w_E$  respectivement. On appelle aussi séparateur  $S$ , l'ensemble des arêtes dont les extrémités sont dans deux partitions différentes.

Le problème de  $N_S$ -partitionnement du graphe non-orienté  $G$  introduit par GAREY et coll. [1976] correspond à trouver une partition de  $G$  et un séparateur  $S$  tels que :

1. Le poids de chaque partie soit équivalent, il faut donc minimiser la différence entre chaque couple de parties :

$$\forall (i, j) \in \llbracket 1; N_S \rrbracket^2, |w_V(\pi_i) - w_V(\pi_j)| \text{ est minimal,}$$

c'est la condition qui conduit à l'équilibrage de charge de la partition.

2. Le poids de l'interface (aussi appelé coût de coupe ou *edgcut* en anglais) doit être le plus petit possible, c'est à dire que :

$$\sum_{x \in S} w_E(x) \text{ est minimal.}$$

La première contrainte permet à la décomposition d'être *in fine* équilibrée, tandis que la seconde permet de minimiser la taille de l'interface et donc les communications entre les partitions. La seconde contrainte du problème de partitionnement est appelée la fonction coût; c'est elle dont la minimisation sera faite par optimisation. Elle peut aussi être modifiée pour minimiser le nombre de partitions voisines aux sommets de bord. Soient  $N_{\text{adj}}(v)$  le nombre de sous-domaines adjacents au sommet  $v$  et  $V_b$  l'ensemble des nœuds définis sur le bord, la contrainte de minimisation du volume total de communications peut être remplacée en réduisant le nombre total de voisins aux sommets. Cette contrainte est remplacée par :

$$\sum_{v \in V_b} N_{\text{adj}}(v) \text{ est minimal.}$$

Cette formulation conduit à obtenir un minimum de voisins par sous-domaines sans s'occuper de minimiser la taille de l'interface.

L'obtention d'un tel partitionnement ne peut être réalisé de manière optimale car le problème de partitionnement est un problème NP-complet comme cela a pu être montré par GAREY et coll. [1976]. Ainsi, plusieurs approches ont été développées pour permettre d'approximer cet optimum

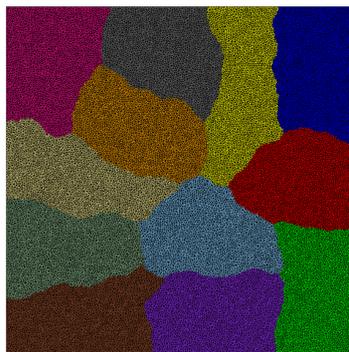


FIGURE 1.8 – Partitionnement en 12 sous-domaines d'un maillage bidimensionnel issu du logiciel de partitionnement de graphe Metis

à faible coût. Ces approches sont classées par CHEVALIER [2007] en 3 catégories qui seront décrites par la suite :

- les approches spectrales;
- les approches combinatoires;
- les approches multi-niveaux.

### 1.2.1.1 Approche spectrale

La première approche s'intéresse à la matrice de Laplace du graphe  $G$  dans le cas où il est non-valué. Pour sa construction,  $A = (a_{ij})_{1 \leq i, j \leq |V|}$  correspond à la matrice d'adjacence définie telle que  $\forall (i, j) \in \llbracket 1; |V| \rrbracket^2$ ,  $a_{ij} = 1$  si  $(i, j) \in E$  et vaut 0 sinon. De plus,  $D$  est la matrice diagonale contenant le nombre de voisins qu'a chaque sommet. Ainsi, La matrice de Laplace  $L$  du graphe est définie par  $L = D - A$ . Cette matrice est semi-définie positive et possède une unique valeur propre nulle dans le cas d'un graphe connexe. Le vecteur propre  $X_2$  associé à la seconde plus petite valeur propre est appelé « vecteur de Fiedler » introduit par [FIEDLER \[1975\]](#) et permet de définir un bipartitionnement. En effet, ce vecteur permet de répartir les sommets sur la droite réelle en fonction de leur valeur dans  $X_2$ . Ainsi, à l'aide d'un critère de séparation unidimensionnel (choix d'un scalaire qui sépare les sommets en 2 sous-ensembles), il est possible de trouver un séparateur pour le graphe  $G$ .

Cette méthode a montré une bonne robustesse d'après [BARNARD et SIMON \[1994\]](#) mais est onéreuse et difficile à mettre en œuvre numériquement car il faut parvenir à trouver le vecteur propre associé à la plus petite valeur propre non nulle d'une matrice symétrique creuse de la taille du graphe complet.

### 1.2.1.2 Approches combinatoires

Avec ce type d'approches, des modifications des partitions suivant une certaine heuristique vont être réalisées. Des algorithmes itératifs locaux ou des algorithmes d'optimisation globale peuvent être distingués parmi la quantité d'algorithmes existants.

Malgré la bonne qualité des partitionnements obtenus par les algorithmes d'optimisation globale, ils restent peu utilisés car un grand nombre de paramètre entre souvent en jeu et leur temps d'exécution sont assez longs. C'est pourquoi les algorithmes itératifs locaux sont plus courants.

**Algorithmes itératifs locaux** Les algorithmes locaux consistent à modifier localement chaque partie. La façon dont sont échangés les sommets définit les différents algorithmes.

Une première catégorie consiste à utiliser une partition initiale équilibrée (le nombre de sommets dans chaque partie est équivalent) et à échanger des sommets au cours d'itérations si l'échange contribue à faire diminuer effectivement la fonction coût. C'est la méthode utilisée dans l'algorithme de [KERNIGHAN et LIN \[1970\]](#).

L'algorithme de [FIDUCCIA et MATTHEYSES \[1982\]](#) est une amélioration du précédent qui permet un partitionnement plus rapide en relâchant faiblement la contrainte d'équilibre de la partition en faisant des transferts de sommets et non plus des échanges.

Il existe aussi des algorithmes de diffusion appelés algorithmes « à bulles » utilisés par exemple par [MEYERHENKE et coll. \[2006\]](#); [MEYERHENKE et SCHAMBERGER \[2006\]](#). À partir d'une répartition aléatoire de  $N_S$  nœuds servant de graine initiale, chaque sous-domaine s'élargit de proche en proche jusqu'à ce que chaque sommet soit assigné à une partie. Ensuite, une nouvelle graine est choisie comme étant le centre de chacune des parties et la croissance des bulles est réitérée. Ce processus est répété jusqu'à ce qu'il n'y ait plus de variation dans les graines initiales. Cela conduit à former des sous-domaines avec une forme globale régulière comme le montre la figure 1.9. Cette approche ne minimise pas la fonction coût et le calcul des nouvelles graines est assez coûteux mais cela permet d'obtenir des partitions ayant parfois de meilleurs résultats que des méthodes plus classiques, comme celles que nous allons décrire maintenant.

**Algorithmes d'optimisation globale** Les autres heuristiques d'optimisation cherchent à trouver le minimum global de la fonction coût. Elles ont toutes en commun un nombre de paramètres difficile à estimer *a priori* pour le problème de partitionnement. On pourra citer des méthodes telles que les algorithmes évolutionnistes développés par exemple par [CHOCKALINGAM et ARUNKUMAR \[1995\]](#), le recuit simulé de [ČERNÝ \[1985\]](#), la recherche tabou de [BATTITI et BERTOSSI \[1999\]](#), les Gradients Aléatoires Adaptatifs de [AREIBI et YANG \[2004\]](#), les méthodes de diffusion statistique

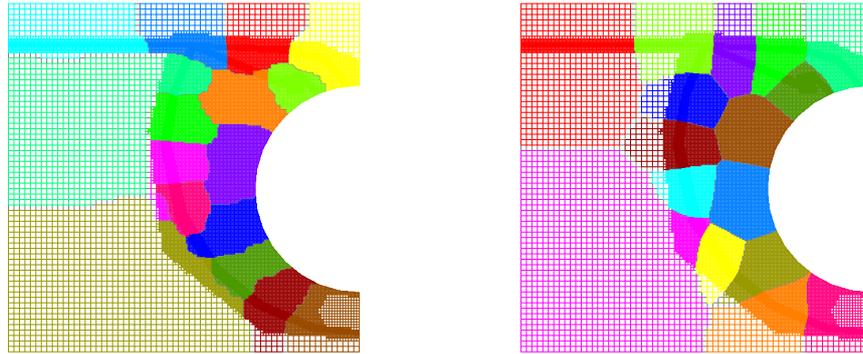


FIGURE 1.9 – Décompositions d'un maillage multigrille en 16 sous-domaines obtenus avec Metis (à gauche) et avec un algorithme à bulle (à droite). Metis produit une décomposition dont la fonction coût vaut 1226 arêtes et l'interface est constituée de 2082 nœuds alors que l'algorithme à bulle conduit à 1168 arêtes coupées et une interface de taille 1795 (figures issues de MEYERHENKE et coll. [2006]).

par TAO et ZHAO [1993] ou encore les méthodes de colonies de fourmis développées par KOROŠEC et coll. [2004].

### 1.2.1.3 Approche multi-niveaux

Toutes les méthodes précédemment citées deviennent très coûteuses lorsque la taille du graphe est importante, c'est pourquoi des méthodes de type multi-niveaux ont été développées par BARNARD et SIMON [1994], HENDRICKSON et LELAND [1995] ou encore KARYPIS et KUMAR [1998]. En effet, ces méthodes, à l'aide d'un algorithme de contraction, permettent de réduire la taille du graphe et ainsi de faciliter l'application de n'importe quelle des méthodes susmentionnées. Ces approches comportent trois phases (voir figure 1.10) :

1. une phase de contraction qui va permettre de réduire la taille du graphe;
2. une phase de partitionnement qui découpe le plus petit graphe;
3. et une phase d'expansion, où la partition du petit graphe est projetée sur le plus grand graphe.

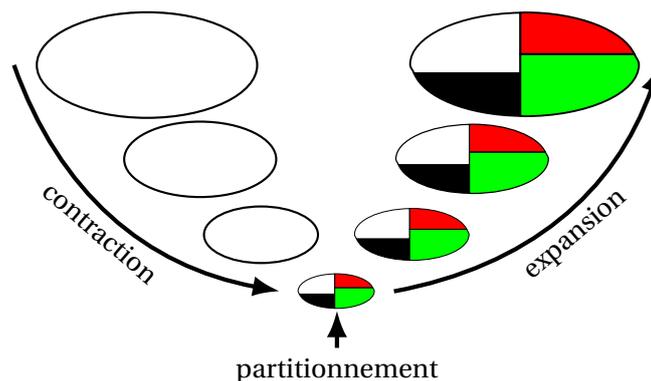


FIGURE 1.10 – Représentation schématique de l'algorithme de partitionnement multi-niveaux pour 4 partitions (reproduite de CHEVALIER [2007])

**La contraction** L'objectif de l'étape de contraction est de réduire la taille du graphe pour permettre de trouver l'optimal du partitionnement facilement. De plus, la suite de graphes ( $G_\lambda$ ) construits ont des topologies « proches » de celles du graphe initial  $G$  en ayant un nombre de sommets décroissant quand le niveau de contraction augmente.

Pour réaliser cette contraction, les nœuds sont regroupés par paires. Historiquement, ces paires étaient choisies aléatoirement comme le faisaient HENDRICKSON et LELAND [1995] par

exemple. Mais la méthode qui fait consensus consiste à appairer les arêtes dont la somme du poids des extrémités est le plus élevé, c'est la méthode *Heavy Edge Matching (HEM)* développée par KARYPIS et KUMAR [1998]. En effet, elle semble mieux préserver la topologie du graphe. Il existe aussi des méthodes qui ne regroupent plus seulement les sommets deux à deux mais bien des groupes de sommets qui peuvent être sélectionnés par des méthodes dites « probabilistes » telle que celle développée par RON et coll. [2005]. Cette réduction de la taille du graphe se fait jusqu'à atteindre un graphe ne faisant plus qu'une centaine de sommets environ.

**Le partitionnement du graphe réduit** Comme mentionné précédemment, un partitionnement tel que ceux décrits dans la section 1.2.1.2 sur les approches combinatoires est appliqué sur le graphe contracté afin de trouver un partitionnement proche de l'optimum.

**L'expansion** Cette phase tente de conserver l'optimum sur les graphes de plus haut niveau lors de la ré-augmentation de la taille du graphe; elle est illustrée sur la figure 1.11. La première étape de cette expansion est appelée phase de projection (voir figure 1.11b), c'est l'opération inverse de l'étape de contraction où la taille du graphe va augmenter. Une fois le graphe de plus grande taille obtenu, des retouches locales des partitions sont réalisées par échanges locaux de sommets à l'aide des algorithmes de KERNIGHAN et LIN [1970] ou de FIDUCCIA et MATTHEYSES [1982] par exemple : c'est ce qui est appelé l'étape de raffinement (voir figure 1.11c).

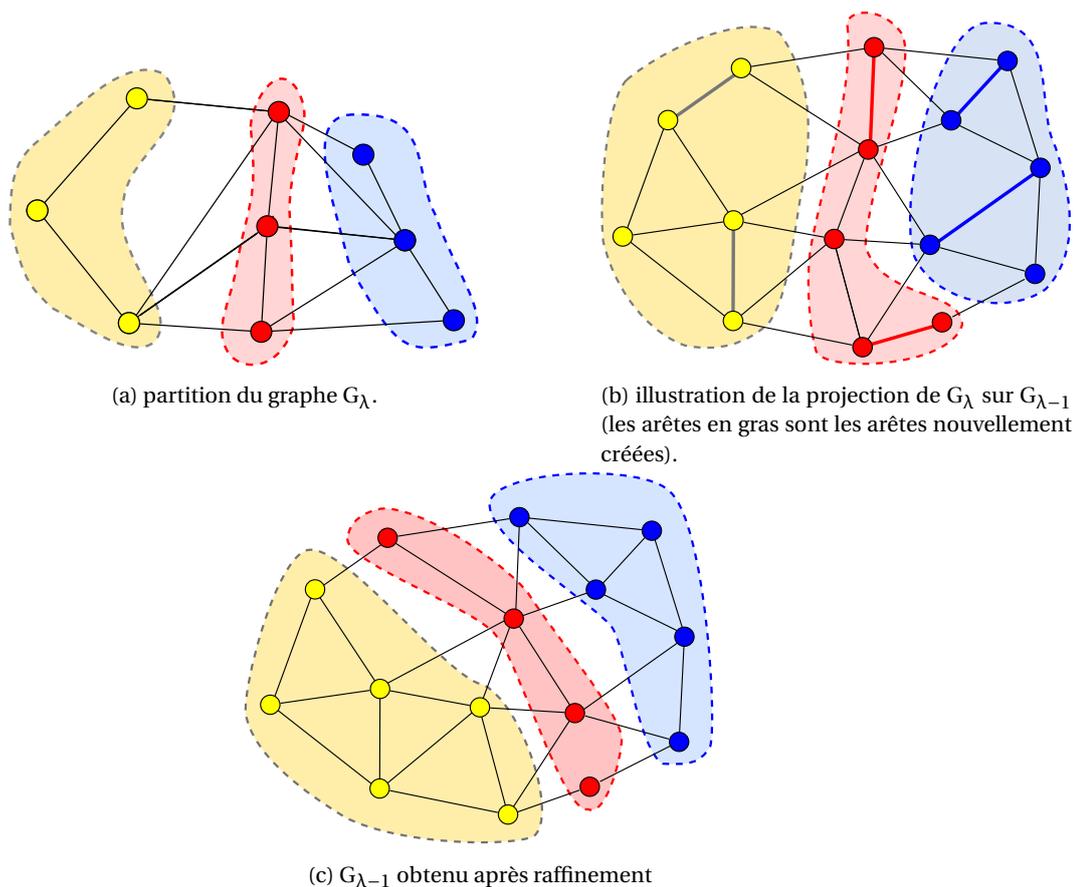


FIGURE 1.11 – Exemple de l'étape d'expansion lors du passage du niveau  $\lambda$  au niveau  $\lambda - 1$  (Reproduction de CHEVALIER [2007]). Les différentes couleurs représentent les différentes partitions.

En raison de la rapidité d'exécution des méthodes multi-niveaux, ce sont les méthodes qui sont le plus implémentées dans les logiciels de partitionnement de maillage tels que Metis, Scotch ou Chaco.

## 1.2.2 Stratégies de maillage et de remaillage parallèle

Les logiciels de génération de maillage séquentiels sont limités à cause de temps de génération de maillage et de l'occupation mémoire associée qui deviennent beaucoup trop important lorsqu'il s'agit de générer des maillages de très grande taille. Il semble donc essentiel que ces méthodes soient développées dans un cadre à mémoire distribuée à l'aide de communications inter-processus *MPI* pour éviter cet écueil.

Dans cette sous-section, les méthodes de maillage et de remaillage seront considérées indifféremment. En effet, la différence majeure entre ces deux processus est le fait qu'il existe déjà lors d'une étape de remaillage, un maillage dont la taille caractéristique des éléments doit être adaptée pour réduire l'erreur de discrétisation après un calcul *EF* par exemple. De plus, lors du remaillage il faut être capable de transférer les solutions obtenues sur le nouveau maillage. Différentes stratégies ont pu être développées pour atteindre cet objectif avec plus ou moins de facilité.

### 1.2.2.1 Outils de maillage séquentiel

Avant de voir comment la génération de maillage peut être rendue parallèle, il faut s'attacher à savoir quelles sont les méthodes de maillage séquentielles existantes.

**Méthode de Delaunay** La méthode la plus répandue car assurant la génération d'éléments peu distordus est la méthode de triangulation ou de tétrahédrisation de *DELAUNAY* [1924, 1934]. Elle consiste, à partir d'un nuage de points, à construire une triangulation qui respecte la règle de la sphère vide. Autrement dit, une triangulation (respectivement une tétrahédrisation en 3D) est de Delaunay si et seulement si pour tout élément du maillage, le cercle (respectivement la sphère) circonscrit à chacun de ses éléments ne contient aucun autre sommet. La figure 1.12 montre deux triangulations d'un même nuage de points : dans le cas de gauche, le cercle circonscrit au triangle (ADC) contient le point B, ainsi par définition cette triangulation n'est pas de Delaunay ; par contre la triangulation de droite est bien de Delaunay car les deux cercles circonscrits laissent bien le quatrième sommet en leur extérieur.

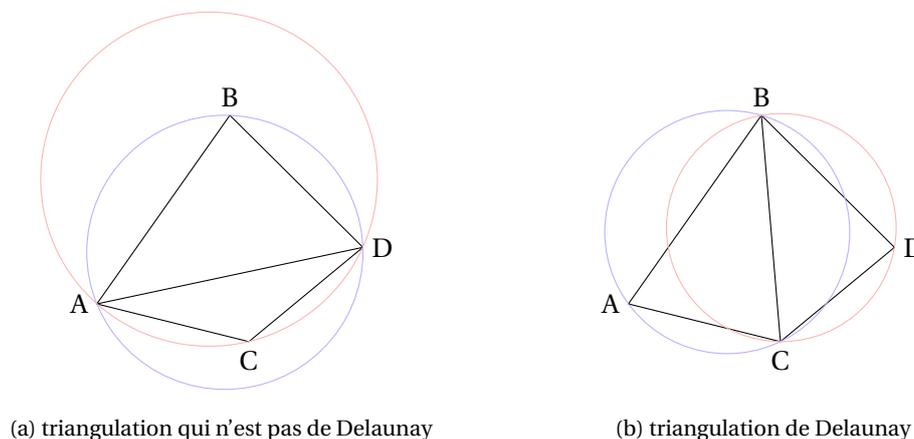


FIGURE 1.12 – Illustration d'une triangulation de Delaunay

Ici, avec l'algorithme de Bowyer-Watson créé par *BOWYER* [1981] et *WATSON* [1981] il est possible d'insérer un nouveau nœud dans le maillage ; l'algorithme se décline de la manière suivante et est décrit sur la figure 1.13 :

1. Trouver l'élément dans lequel insérer le nœud ;
  2. Trouver tous les éléments adjacents dont le cercle (respectivement la sphère) contient le nœud ;
  3. Supprimer tous les éléments sélectionnés (ce qui crée ce qui est appelé une cavité<sup>3</sup>) ;
- 
3. Une cavité d'un point est l'ensemble des triangles dont le cercle circonscrit contient ce point.

4. Connecter les bords de cette cavité avec le nœud à insérer.

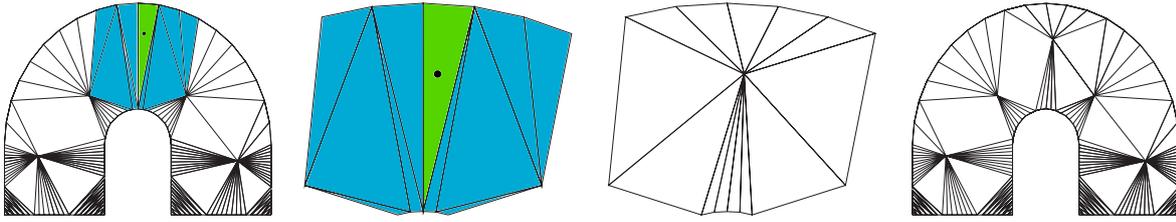


FIGURE 1.13 – Insertion d'un sommet par l'algorithme de Bowyer-Watson. L'élément en vert est l'élément dans lequel on retrouve le nœud à insérer et les éléments en bleu sont ceux dont le cercle circonscrit contient le nœud à insérer. (figure issue de CHRISOCHOIDES et NAVE [2003])

**Méthode par avancée de front** Une autre famille de méthodes développées par LÖHNER [1988] ou encore LO [1991] sont les méthodes frontales ou *Advancing Front Technique* (AFT) qui permettent de construire itérativement, à partir d'une triangulation de la surface de l'objet à mailler, une triangulation en ajoutant un front de nouveaux éléments à l'intérieur de la structure. La figure 1.14 illustre l'avancée du front après la première itération. Ce processus est réitéré jusqu'à ce que toute la structure soit maillée.

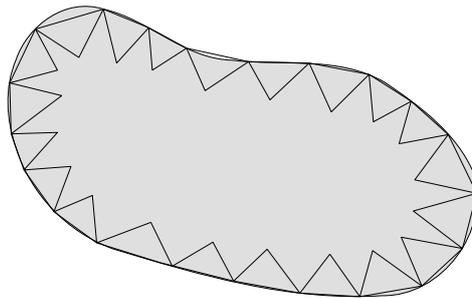


FIGURE 1.14 – Première itération d'une méthode AFT

**Raffinements imbriqués adaptatifs** Il existe une troisième famille de méthodes, celles qui raffinent un maillage grossier en divisant les éléments existants selon des règles prédéfinies (voir figure 1.15 pour certaines règles de subdivision d'un tétraèdre). Par exemple, MUTHUKRISHNAN et coll. [1995] utilise cela pour raffiner un maillage grossier en fonction de critères sur les arêtes les plus longues du maillage total.

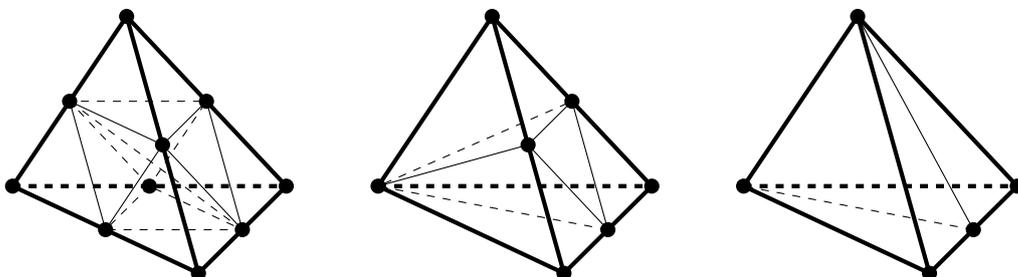


FIGURE 1.15 – Différents types de subdivisions d'un tétraèdre (reproduite de OLIKER et coll. [2000])

**méthode SPH** Plus récemment, une méthode de maillage développée par FU et coll. [2019] et fondée sur une approche de relaxation d'un fluide *Smoothed Particle Hydrodynamics* (SPH) à l'intérieur du domaine permet de distribuer des particules dans le volume et de générer ensuite un maillage à partir de la position de ces particules. Ce processus est illustré sur la figure 1.16. À partir

d'une distribution initiale de particules (figure 1.16a) et avec la résolution de proche en proche de l'équation de la SPH de MONAGHAN [1992], les particules peuvent être distribuées selon une densité donnée (figure 1.16b). Enfin une triangulation de Delaunay est appliquée au nuage de points ainsi obtenu (figure 1.16c). Cette méthode semble coûteuse car il faut résoudre un problème phy-

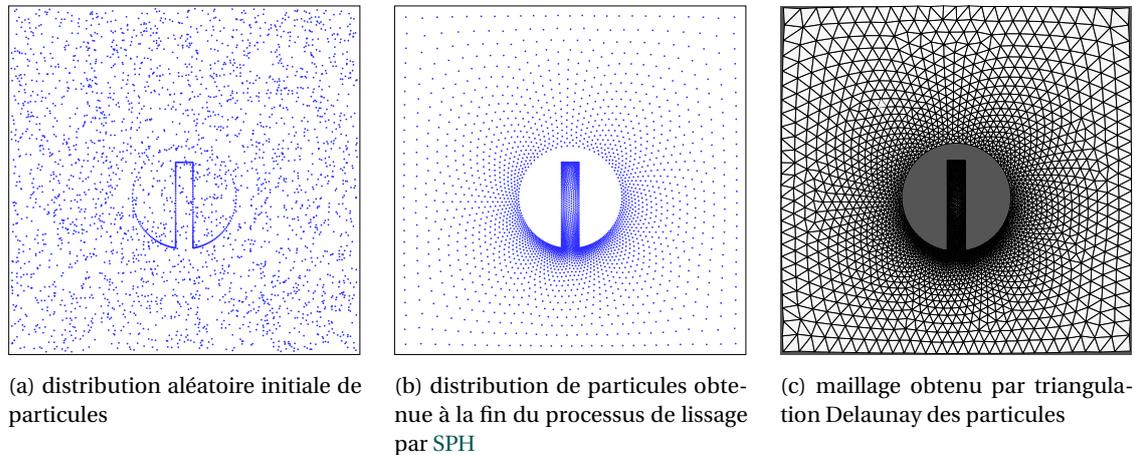


FIGURE 1.16 – Génération de maillage à l'aide de la méthode SPH

sique de grande dimension pour trouver la position des nœuds.

### 1.2.2.2 Parallélisation des opérations élémentaires dans les algorithmes séquentiels

**Méthodes de subdivision en parallèle** Les premiers travaux concernant le maillage parallèle ont été menés dans les années 1990 par CASTAÑOS et SAVAGE [1997], d'abord sur des maillages bidimensionnels en contexte de mémoire distribué. Il s'agit d'une méthode de subdivision d'éléments qui se fait indépendamment dans chaque sous-domaine. Ainsi, lorsqu'une arête qui est localisée sur l'interface est subdivisée, un algorithme de synchronisation s'assure que les deux sous-domaines créent le nœud au même endroit. La méthode a été étendue en 3D dans la fin des années 1990 par DE COUGNY et SHEPHARD [1999] ou encore OLIKER et coll. [2000].

**Triangulation de Delaunay parallèle** La parallélisation de la triangulation de Delaunay a été imaginée par OKUSANYA et PERAIRE [1996] d'abord en 2D, puis en 3D par OKUSANYA [1997]. Ces travaux ont ensuite pu inspirer CHRISOCHOIDES et NAVE [2003]. L'algorithme de Bowyer-Watson a été adapté en parallèle par NAVE et coll. [2004], il est possible de générer un maillage en parallèle à partir d'un maillage initialement grossier par insertion de points.

Le problème se pose en parallèle lorsque le nœud à insérer se situe dans une cavité en contact avec l'interface. Le sous-domaine dans lequel est la cavité doit envoyer une requête à son voisin et lui transmettre les coordonnées du nœud à insérer pour que le voisin vérifie si le maillage reste de type Delaunay après l'insertion de ce nœud. Cela entraîne un grand nombre de communications et cela implique d'attendre le retour de l'information pour construire la cavité. Une fois la cavité créée puis triangulée, il faut distribuer les éléments sur les sous-domaines. La figure 1.17 illustre cette méthode. Lors de l'insertion d'un nœud P sur la figure 1.17a, le sous-domaine  $\Omega^{(0)}$ , crée la cavité locale. Comme cette cavité est en contact du bord, le sous-domaine voisin  $\Omega^{(1)}$ , dont l'élément au bord a bien son cercle circonscrit qui contient le point P, est interrogé et l'élément en question est ajouté à la cavité. La cavité étant maintenant au contact de l'interface avec  $\Omega^{(2)}$ , celui-ci est aussi interrogé et un de ses éléments fait aussi effectivement partie de la cavité. Ainsi, la cavité a été construite de proche en proche en faisant des communications de la coordonnées de P. Enfin, la cavité est triangulée en parallèle comme le montre la figure 1.17b.

**Parallélisation de la méthode AFT** Pour l'approche frontale, LÖHNER et coll. [1992] choisissent de mailler par leur méthode AFT tous les domaines en parallèle indépendamment sans mailler

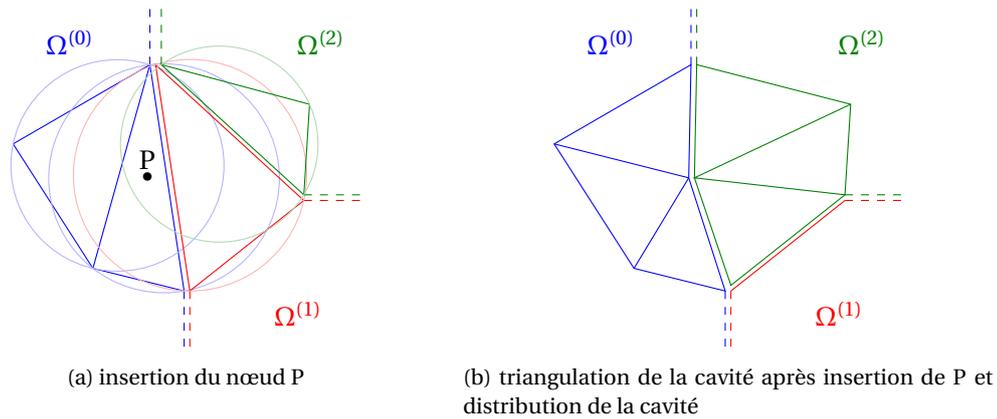


FIGURE 1.17 – Insertion d'un nœud sur un cavité à l'interface entre 3 sous-domaines

l'interface : un espace vide reste donc présent entre le maillage et l'interface. Ensuite, tous les fronts sont envoyés au processus maître qui va terminer le maillage de l'interface à l'intérieur de la surface fermée formée par tous ces fronts. Cette méthode semble ne pas être extensible car la taille possiblement importante de l'interface engendre un surcoût significatif pour le processus maître.

**Parallélisation de l'approche particulière par SPH** L'approche particulière a aussi pu bénéficier d'une parallélisation développée par Ji et coll. [2020]. Ici, des ensembles de particules sont générés en parallèle dans des zones distinctes de la structure. Une force de tension de surface répulsive est choisie pour permettre aux différents domaines de ne pas se mélanger. Les équations de la SPH sont résolues itérativement jusqu'à stagnation. Une fois cette stagnation atteinte, la tension de surface est réduite jusqu'à ce qu'elle s'annule et en atteignant la stagnation des particules à chaque pas de diminution des particules. Ces étapes sont illustrées sur le lapin de Stanford par Ji et coll. [2020] sur la figure 1.18. Une fois cela effectué, la tétrahédralisation est réalisée à l'aide de méthodes de type Delaunay de manière indépendante sur chaque sous-région.



FIGURE 1.18 – Illustration de la méthode de maillage par SPH en parallèle (figures extraites de Ji et coll. [2020])

### 1.2.2.3 « Orchestration » d'algorithmes séquentiels en tant que « boîtes noires »

En raison de la maturité des algorithmes et des logiciels de maillages séquentiels, il semble difficile de développer une nouvelle méthodologie spécifique aux architectures parallèles et qui serait une adaptation et donc une modification d'une méthode de maillage séquentiel. En effet,

CHRISOCHOIDES [2006] affirme que cela constitue un travail très important qui requiert des compétences multi-disciplinaires et nécessiterait de 10 à 15 ans pour que l'écosystème logicielle et algorithmique atteigne un tel niveau de maturité. L'idée découlant de ce constat est que, finalement, l'effort n'est pas à mettre sur la parallélisation d'opération élémentaires mais plutôt sur la bonne façon d'« orchestrer » un mailleur séquentiel pour lui permettre de réaliser efficacement et de manière extensible cette génération de maillage parallèle. À la différence de la partie précédente, le mailleur ici est vu comme une « boîte noire ».

**Repartitionnement** Les méthodes fondées sur le repartitionnement utilisant un mailleur en tant que « boîte noire » ont été développées par COUPEZ et coll. [2000] ou encore DIGONNET et coll. [2007, 2017]. Cette méthode consiste à remailler les éléments n'étant pas au contact de l'interface. C'est un processus itératif où l'interface est repositionnée à chaque itération du fait d'un poids de communication important mis sur les éléments à remailler. La figure 1.19 illustre ce procédé sur un cas bidimensionnel avec 6 sous-domaines.

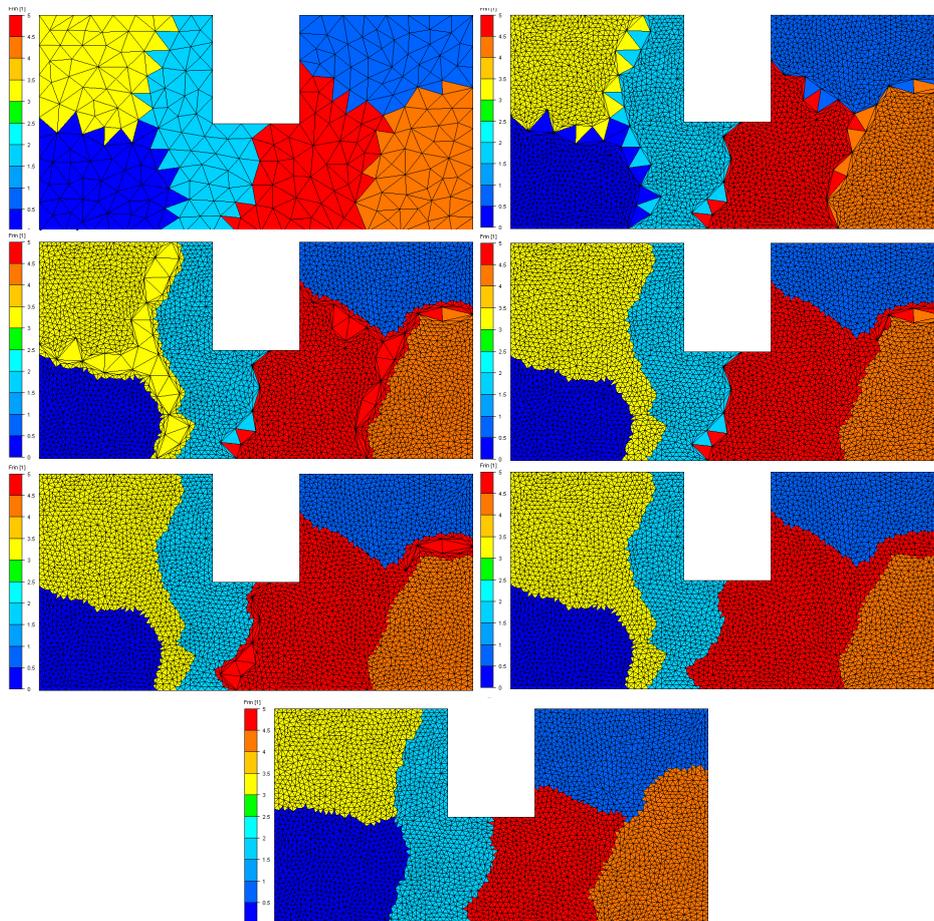


FIGURE 1.19 – Stratégie de repartitionnement - remaillage de DIGONNET et coll. [2007] (se lit dans le sens lexico-graphique). Le remaillage se fait uniquement sur les éléments non adjacents à l'interface (colonne de droite). Le remaillage de toute la structure est assuré par un déplacement itératif de l'interface (colonne de gauche)

CAVALLO et coll. [2005] choisit de ne pas faire un repartitionnement total mais seulement des migrations d'éléments : grâce à cela, il n'a besoin que de 2 étapes de remaillage pour garantir le remaillage de l'entièreté de la structure.

Dans ces deux cas, la non préservation de l'interface (et donc de la forme des sous-domaines) est un gros inconvénient dans un cadre d'adaptation de maillage. En effet, les champs solution calculés sur le maillage initial ont perdu leur localisation dans les sous-domaines car ils ont changé de géométrie. Des échanges importants sont nécessaires pour permettre le transfert de ces champs

sur le nouveau maillage fin.

**Partitionnement de la géométrie** LAEMMER [1997] propose une méthode de maillage parallèle où une géométrie définie par des primitives géométriques bidimensionnelles (*i.e.* des primitives utilisées dans la CAO) est bi-partitionnée récursivement. Ensuite, chaque sous-domaine construit ses nœuds de bord en utilisant exactement la même méthode de maillage de son bord; la conformité à l'interface est ainsi garantie. Et enfin une méthode de maillage quelconque est choisie pour mailler l'intérieur du volume. Cette méthode présente l'avantage de n'avoir que les échanges des indices des nœuds à l'interface à effectuer avec les voisins pour permettre de connaître la connectivité globale du maillage. Le résultat de cette technique est illustré sur la figure 1.20, où la taille des domaines est adaptée à la finesse du maillage : plus le maillage est fin, plus le sous-domaine a une taille caractéristique faible. GALTIER et GEORGE [1997] proposent le même type de méthode mais appliqué à des géométries discrètes (triangulation de la surface externe).

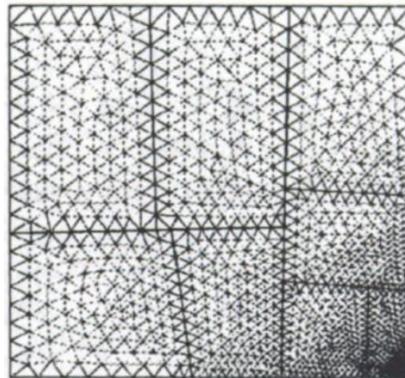


FIGURE 1.20 – Maillage obtenu avec la méthode de LAEMMER [1997], les 8 sous-domaines sont séparés par des lignes droites.

LINARDAKIS et CHRISOCHOIDES [2006, 2008] ont réussi à proposer une méthode avec un partitionnement de la géométrie qui rend indépendant la génération de maillage entre les sous-domaines. Pour ce faire, il suffit de construire le squelette de la structure (l'ensemble des points à égale distance des bords du domaine), c'est ce qui est représenté sur la figure 1.21(a) et approximé à l'aide de points de Voronoï d'une triangulation des nœuds du bord. Ensuite les séparateurs sont simplement des segments qui joignent un point de Voronoï et le bord (figure 1.21(b)). À partir de là, une discrétisation uniforme des nœuds de cette interface est réalisée en prétraitement et le maillage est distribué et généré en parallèle facilement. Cette méthode semble extensible mais elle n'est applicable qu'en 2D car l'étape de découpage de la structure (figure 1.21(b)) n'est pas généralisable telle quelle en 3D.

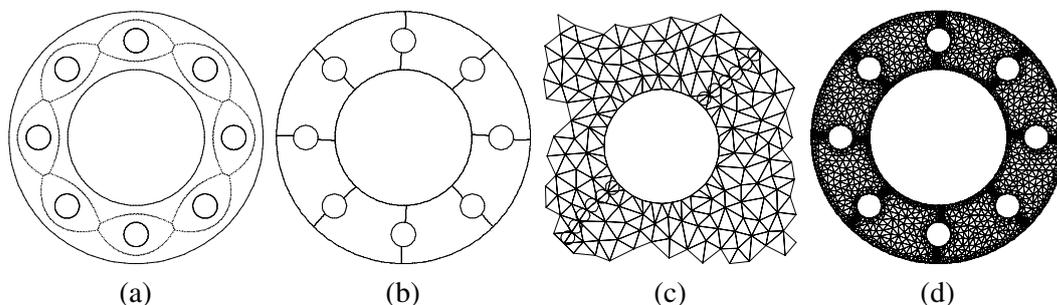


FIGURE 1.21 – Illustration de la méthode de maillage parallèle par partitionnement *Medial Axis* de LINARDAKIS et CHRISOCHOIDES [2006]. Le squelette de la structure est calculé (a), des séparateurs sont construits en suivant des lignes orthogonales au squelette (b), ces séparateurs sont discrétisés uniformément (c) (c'est représenté par les petits cercles sur (c)) et le maillage intérieur est ensuite réalisé en parallèle (d)

**Utilisation d'un maillage grossier initial** La majorité des méthodes présentées ici sont vues comme étant des méthodes de génération de maillage mais du fait de l'existence de ce maillage grossier initial, finalement, il est possible de se ramener aux mêmes problématiques que pour le remaillage. Par contre, ici, en raison de l'existence d'un maillage grossier, il faut avoir à disposition un algorithme qui permet, après raffinement, de ramener le bord  $\partial\Omega$  du maillage fin généré sur la géométrie-objectif. Globalement, ce type de méthode suit le schéma de la figure 1.22 où l'interface est maillée avant le volume. Ici, l'avantage est que la géométrie des sous-domaines reste la même tout au long du processus, ainsi, lors d'une adaptation de maillage, la projection de champs est simple à réaliser.

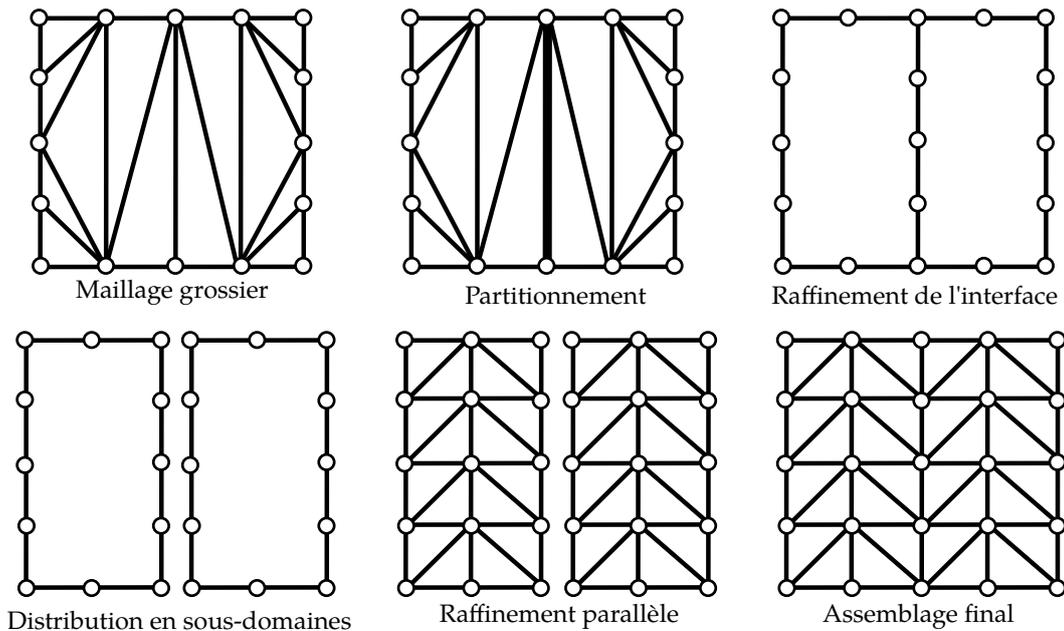


FIGURE 1.22 – Principe général des méthodes utilisant un maillage grossier (extrait de CHEN et coll. [2018])

ITO et coll. [2007] proposent une méthode qui consiste à subdiviser chaque arête de la géométrie selon une règle prédéfinie (chaque arête est découpée en un nombre d'arêtes de même longueur). Ensuite, toutes les faces sont maillées à l'aide d'un algorithme de Delaunay 2D. Ces opérations étant identiques de part et d'autre de l'interface, les nœuds des maillages coïncident. Il faut noter aussi que les nœuds qui sont sur le bord physique du domaine sont interpolés à l'aide de fonctions de forme quadratiques des triangles pour approcher la géométrie initiale. ITO et NAKAHASHI [2002] ont développé une méthode permettant cette interpolation à l'aide de l'information de la normale à la surface. Les maillages obtenus sont décrits sur la figure 1.23. Cette méthode semble cependant requérir un volume d'échange important pour reconstruire la connectivité à l'interface. Cela pénalise l'extensibilité de l'approche.

Ce problème est résolu par WANG et coll. [2017] qui, à l'aide d'un raffinement hiérarchique régulier et une indexation avancée aux interfaces, est capable sans aucun échange de générer le maillage et d'obtenir la connectivité à l'interface. Ici, comme avec ITO et coll. [2007], les nœuds sur les bords ne sont pas exactement sur la géométrie sous-jacente<sup>4</sup> car cette méthode ne la prend pas en compte.

CHEN et coll. [2018] utilisent les mêmes outils que précédemment mais, ils parviennent à faire des opérations sur la CAO pour ne plus l'approximer mais bien à pouvoir l'exploiter exactement.

Une autre approche a été développée par BOVET et coll. [2019a,b] et est contrainte de n'utiliser qu'un mailleur volumique, sans mailleur bidimensionnel, par opposition aux méthodes susmentionnées. Un coloriage est défini sur les domaines de telle sorte que deux sous-domaines adjacents soient de couleurs différentes. Ensuite, l'ensemble des couleurs est parcouru, et chaque portion d'interface de la couleur courante est remaillée sur une profondeur d'éléments définie

4. c'est une interpolation quadratique

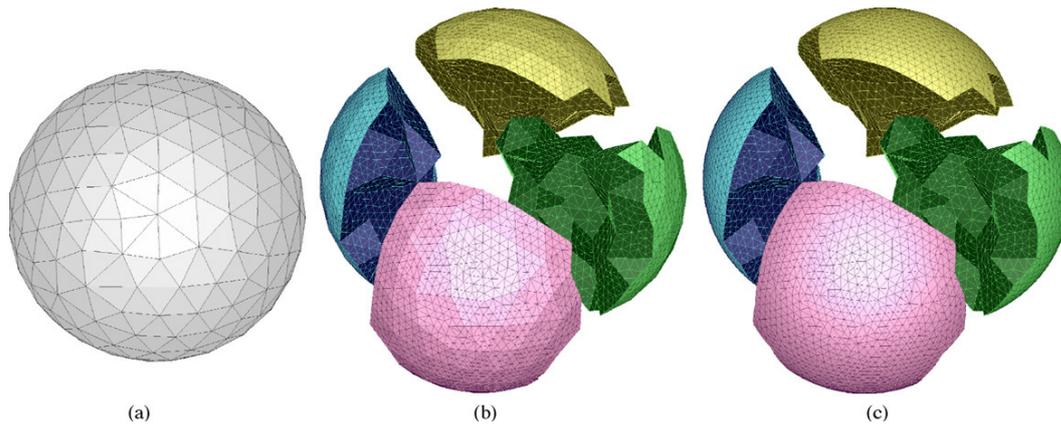


FIGURE 1.23 – Décomposition d'une sphère par le méthode d'ITO et coll. [2007] : (a) maillage grossier initial, (b) surface extraite du 4-partitionnement puis raffinée et (c) interpolation quadratique des nœuds situés sur l'extérieur du domaine.

par l'un des deux sous-domaine adjacent. Pour ce faire, les maillages volumiques de l'interface obtenus sont préalablement échangés et assemblés grâce à des opérations booléennes puis redécoupés et redistribués après remaillage, permettant ainsi de garantir la conformité à l'interface. Puis enfin, l'intérieur des volumes est remaillé en parallèle. Cette méthode est présentée sur la figure 1.24.

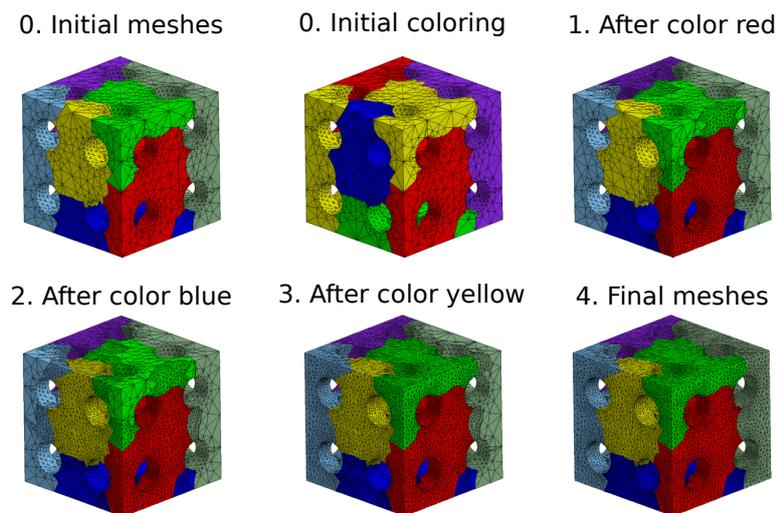


FIGURE 1.24 – Remaillage parallèle par coloriage issu de BOVET et coll. [2019a,b]

Une dernière méthode encore décrite par BOVET et coll. [2019a] consiste à remailler une interface volumique de manière séquentielle sur un seul processus. Après un partitionnement du maillage initial, les interfaces sont envoyées au processus maître qui effectue son remaillage, puis ces maillages volumiques sont renvoyés aux sous-domaines et un remaillage de l'intérieur est enfin réalisé. Tout cela est illustré sur la figure 1.25. Cette stratégie a pour inconvénient de ne pas être extensible lorsque la taille de l'interface devient importante car le processus maître devra remailler un maillage de taille de plus en plus importante lorsque le nombre de sous-domaines augmente.

LOSEILLE et coll. [2015] parviennent à éviter ce problème en partitionnant à nouveau ce maillage de l'interface en différents niveaux. Ces niveaux sont définis comme étant les interfaces du niveau immédiatement supérieur. Par exemple, le niveau 3, représenté sur la figure 1.26, est l'interface de l'interface des domaines volumiques, il correspond donc aux interfaces qui sont des lignes et non plus des surfaces comme au niveau 2. Ainsi, en remaillant successivement les différents niveaux partitionnés, il est possible de faire un remaillage parallèle qui soit extensible car la même charge de calcul est distribuée aux différents processus. Par contre, la méthodologie déve-

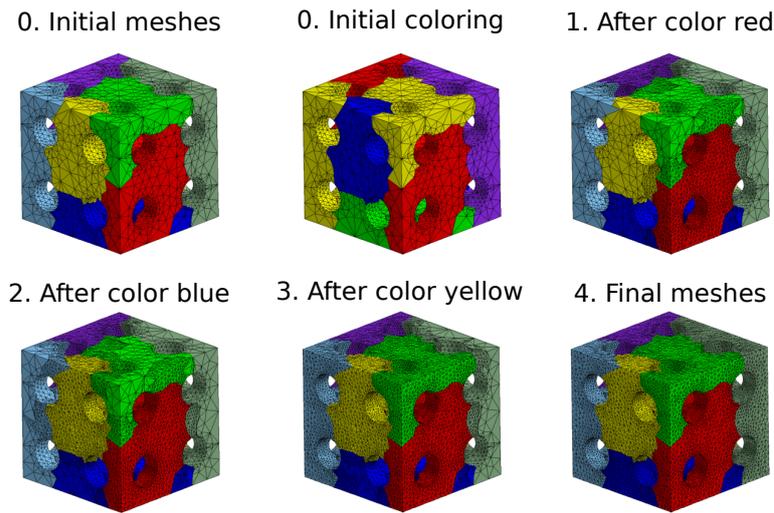


FIGURE 1.25 – Remaillage parallèle par remaillage de l'interface sur le processus maître issu de BOVET et coll. [2019a]

loppée ici ne permet pas de respecter la géométrie sous-jacente si elle est courbe.

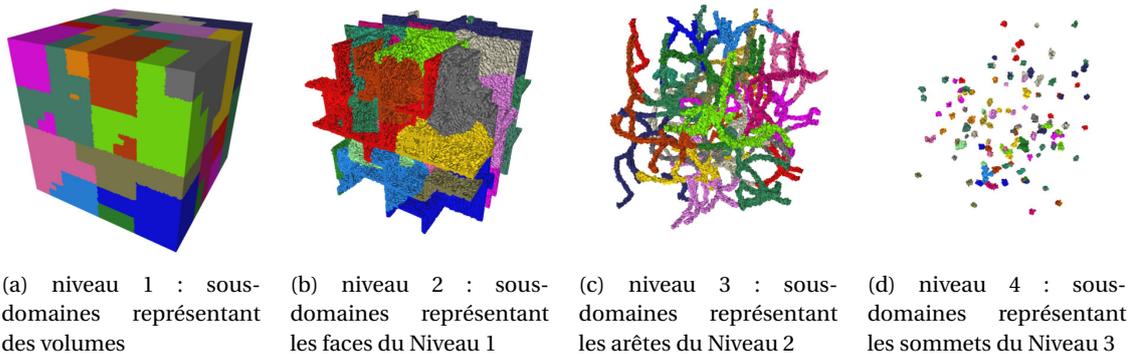


FIGURE 1.26 – Différents niveaux de partitionnement d'un cube (issu de LOSEILLE et coll. [2015])

### 1.3 Méthodes de déformation de maillage

Les méthodes de déformation de maillage se présentent comme un outil essentiel pour la limitation des erreurs géométriques lors du remaillage ou lors du maillage (fondé sur un maillage grossier initial); le remaillage ne prenant souvent pas en compte la géométrie sous-jacente (ou seulement au moyen de corrections approximatives locales), il conduit à des discrétisations qui peuvent être très éloignées de la géométrie réelle. Il faut donc, après raffinement, déplacer les nœuds nouvellement créés sur le bord du domaine et répartir les nœuds restant à l'intérieur du volume.

Dans toute cette section, et contrairement aux notations utilisées dans la Section 1.1 (concernant les méthodes de décomposition de domaine), le bord n'est plus une interface mais ce qui est entendu classiquement concernant le bord d'un domaine  $\partial\Omega$ . Les grandeurs qui seront définies sur ce bord seront indicées par la lettre  $b$ .

L'objectif des méthodes de déformation de maillage est qu'à partir d'un déplacement de bord connu  $\mathbf{u}(\mathbf{X}_b) = \mathbf{u}_b$ , il soit possible de calculer le déplacement  $\mathbf{u}(\mathbf{X})$  en toutes les positions  $\mathbf{X}$ . Les entités mises en jeu dans cette section sont :

- le maillage fin éloigné de la géométrie, noté  $\Omega_r$ ,
- la géométrie physique vers laquelle  $\Omega_r$  est déformé, noté  $\Omega_\phi$ .

Cette méthode consiste donc à projeter le maillage  $\Omega_r$  sur la géométrie  $\Omega_\phi$  en répartissant au mieux les nœuds pour créer des éléments de bonne qualité<sup>5</sup>. La déformation complète est réalisée en deux étapes qui seront les deux grands axes de cette section :

- la projection de la peau de  $\Omega_r$  sur  $\Omega_\phi$ ;
- la répartition des autres nœuds à l'intérieur de  $\Omega_r$ .

### 1.3.1 Techniques de projection

Comme évoqué précédemment, la première étape consiste à projeter la peau de  $\Omega_r$  sur celle de  $\Omega_\phi$ . Cela peut-être fait de manière explicite à l'aide d'une direction prédéterminée ou de manière automatique en paramétrisant les deux surfaces et en faisant une correspondance des nœuds sur cette paramétrisation sans avoir à faire de choix *a priori* sur la direction de projection.

#### 1.3.1.1 Paramétrisation sphérique

Cette méthode s'inscrit dans un cadre plus large que la projection d'un maillage sur un autre. En effet, elles permettent de « mélanger » deux maillages permettant d'en obtenir un maillage hybride, où le résultat obtenu est plus ou moins proche de l'un ou l'autre grâce à un paramètre de réglage scalaire. Ce paramètre permet à moindre coût – une fois la paramétrisation stockée – d'obtenir un maillage plus ou moins proche des maillages d'origine comme illustré dans le cadre *s.* de la figure 1.27. Ce type de méthode peut être utilisé dans des domaines tels que l'animation ou les effets spéciaux au cinéma en permettant des transitions douces entre différentes positions ou images. Cette méthode s'appuie sur les travaux réalisés par MOCANU [2012] durant sa thèse. Sa méthode, comme celles de AHN et coll. [2004]; ALEXA [2000]; ATHANASIADIS et coll. [2012]; KANAI et coll. [2000]; KENT et coll. [1992], est fondée sur une interpolation simple entre les deux modèles. La figure 1.27 représente la méthode de déformation de maillages de coques 3D fermées. La première ligne représente le maillage source et dans le formalisme de ce manuscrit, il est représenté par  $\Omega_r$  alors que la seconde ligne (le maillage cible) est le maillage sur lequel la projection est réalisée et correspond à  $\Omega_\phi$ . Pour des maillages décrivant des surfaces fermées et sans trous (une

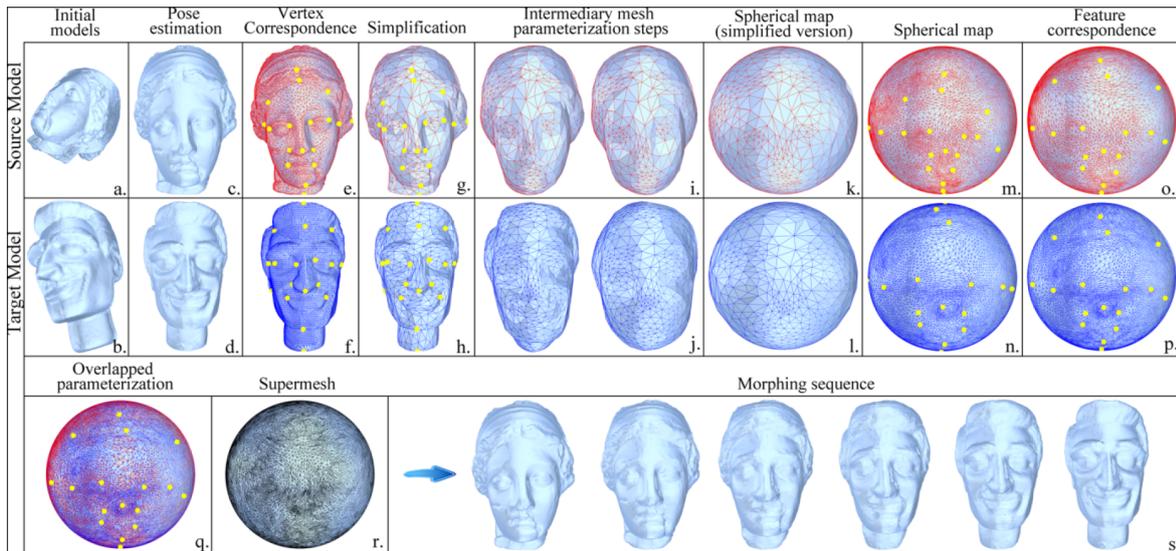


FIGURE 1.27 – Méthode de *morphing* de maillage de coque (extrait de MOCANU [2012])

méthode plus simple est aussi décrite pour les maillages décrivant des surfaces ouvertes (mais la philosophie reste la même), le processus est composé de plusieurs étapes :

5. voir section 1.3.2.4 pour des indicateurs de qualité de maillage

1. L'alignement des axes principaux d'inertie des modèles est réalisé pour que les points remarquables de  $\Omega_r$  et  $\Omega_\phi$  ne soient pas trop éloignés (c. et d. sur la figure 1.27). Ces points remarquables sont les points qui doivent coïncider à la fin du processus. Cette étape n'est pas nécessaire dans un contexte de remaillage car les maillages sont déjà alignés.
2. La correspondance entre les nœuds remarquables est faite par l'utilisateur. (e. et f. sur la figure 1.27). Pour le remaillage, cette étape pourrait-être automatique : les nœuds du maillage grossier étant ces nœuds remarquables.
3. Un déraffinement des deux maillages est effectué pour alléger les ressources de calcul nécessaires aux prochaines étapes (g. et h. sur la figure 1.27).
4. Les maillages sont rendus convexes aux endroits où la courbure Gaussienne est la plus importante (i. et j. sur la figure 1.27) à l'aide d'un lissage autour des nœuds avoisinants. La courbure Gaussienne est le produit des courbures principales. Cette stratégie est issue de [MOCANU et ZAHARIA \[2011\]](#).
5. Les maillages sont projetés sur un espace paramétrique commun : la sphère unité. Cela permet, à l'étape 7, de trouver la position des nœuds de  $\Omega_r$  dans les coordonnées barycentriques des éléments de  $\Omega_\phi$ . Ces mêmes maillages sont ensuite raffinés à nouveau (k., l., m. et n. sur la figure 1.27).
6. Les points remarquables de  $\Omega_r$  et  $\Omega_\phi$  sont ensuite mis en coïncidence par déformation du maillage (o. et p. sur la figure 1.27). Cela peut être réalisé à l'aide des techniques qui seront détaillées dans la partie 1.3.2.
7. Les coordonnées barycentriques  $(\alpha, \beta, \gamma)$  des nœuds  $\mathbf{X}_r^p$  de  $\Omega_r$  dans les triangles  $(\mathbf{X}_{0\phi}^p, \mathbf{X}_{1\phi}^p, \mathbf{X}_{2\phi}^p)$  de  $\Omega_\phi$  sont calculées telles que  $\mathbf{X}_r^p = \alpha\mathbf{X}_{0\phi}^p + \beta\mathbf{X}_{1\phi}^p + \gamma\mathbf{X}_{2\phi}^p$  (l'exposant  $p$  signifie que les nœuds sont dans l'espace paramétrique de la sphère unité).
8. La même interpolation est réalisée entre les maillages initiaux sur les nœuds non paramétriques (i.e. que la transformation  $\mathbf{X}_r = \alpha\mathbf{X}_{0\phi} + \beta\mathbf{X}_{1\phi} + \gamma\mathbf{X}_{2\phi}$  est réalisée pour tous les nœuds de  $\Omega_r$ ).

L'inconvénient de cette méthode réside dans le fait que rien ne garantit une coïncidence entre les faces physiques. Ainsi, une face sur laquelle des conditions aux limites seraient appliquées pourrait empiéter sur une face voisine qui n'a pas la même condition limite. En effet seuls les nœuds grossiers sont utilisés en tant que points remarquables ; les arêtes physiques ne le sont pas car les seules entités « marquables » sont des nœuds. De plus, dans un contexte parallèle, il n'y a aucune garantie que les nœuds coïncidant de part et d'autre d'une interface avant déformation le soient toujours après cette déformation.

### 1.3.1.2 Projection explicite le long d'une direction

Une autre façon de projeter un maillage sur un autre est de définir une règle pour choisir une direction (normale à la surface source, normale à la surface cible, ...) puis d'utiliser cette direction pour la projection. Celle-ci se fait grâce à une intersection du maillage cible et de la droite passant par un point donné du maillage source. Cette opération est répétée pour chaque nœud du maillage source  $\Omega_r$ . Ces méthodes sont très utilisées en *Computer Graphics* pour permettre la visualisation de rendus tridimensionnels.

On suppose cette direction  $\mathbf{d}$  connue. L'objectif de cette projection est qu'à partir d'un nœud  $\mathbf{X}_r$  de  $\Omega_r$ , on trouve ses nouvelles coordonnées dans le maillage  $\Omega_\phi$ , qui est un maillage composé de triangles. Un premier type d'algorithmes fondés sur la méthode de [BADOUEL \[1990\]](#) permettent de réaliser cette projection en calculant l'équation du plan de tous les triangles de  $\Omega_\phi$  et en trouvant l'intersection avec la direction  $(\mathbf{X}_r, \mathbf{d})$ . [MÖLLER et TRUMBORE \[1997\]](#) sont parvenus à éviter le calcul de ces plans de triangles en utilisant les coordonnées barycentriques du triangle, ce qui a permis de rendre cette projection plus efficace, comme illustré par les tests numériques de [SEGURA et FEITO \[2001\]](#). D'autre part, le stockage de tous les plans n'est plus nécessaire, réduisant ainsi l'empreinte mémoire associée.

La méthode de MÖLLER et TRUMBORE [1997] consiste à exprimer de manière paramétrique un point appartenant à  $(\mathbf{X}_r, \mathbf{d})$  par ce qui est appelé un rayon  $\mathbf{R}$  de paramètre  $\rho$  :

$$\mathbf{R}(\rho) = \mathbf{X}_r + \rho \mathbf{d} \quad (1.24)$$

et à exprimer aussi de manière paramétrique un point  $\mathbf{T}_j$  appartenant au  $j$ -ième triangle  $(\mathbf{X}_{0\phi}^j, \mathbf{X}_{1\phi}^j, \mathbf{X}_{2\phi}^j)_j$

$$\mathbf{T}_j(\alpha_j, \beta_j, \gamma_j) = \gamma_j \mathbf{X}_{0\phi}^j + \alpha_j \mathbf{X}_{1\phi}^j + \beta_j \mathbf{X}_{2\phi}^j, \quad (1.25)$$

où la lettre  $j$  désigne le triangle numéro  $j$ . Cette expression est celle d'un plan si  $\forall j, \gamma_j = 1 - \alpha_j - \beta_j$ . Il suffit de trouver enfin dans quel triangle se fait réellement l'intersection à l'aide des conditions d'admissibilité paramétriques du triangle suivantes :

$$\begin{cases} \alpha_j \geq 0 \\ \beta_j \geq 0 \\ \alpha_j + \beta_j \leq 1 \end{cases} \quad (1.26)$$

Pour calculer l'intersection, il suffit d'égaliser  $\mathbf{T}_j(\alpha_j, \beta_j)$  et  $\mathbf{R}(\rho)$  dans les deux équations (1.24) et (1.25) pour arriver à l'équation suivante :

$$\mathbf{X}_r + \rho \mathbf{d} = (1 - \alpha_j - \beta_j) \mathbf{X}_{0\phi}^j + \alpha_j \mathbf{X}_{1\phi}^j + \beta_j \mathbf{X}_{2\phi}^j.$$

Pour chaque triangle, il faut résoudre un problème à 3 équations et 3 inconnues :

$$\begin{bmatrix} -\mathbf{d} & \mathbf{X}_{1\phi}^j - \mathbf{X}_{0\phi}^j & \mathbf{X}_{2\phi}^j - \mathbf{X}_{0\phi}^j \end{bmatrix} \begin{bmatrix} \rho \\ \alpha_j \\ \beta_j \end{bmatrix} = \mathbf{X}_r - \mathbf{X}_{0\phi}^j.$$

Il faut résoudre ce système pour chaque triangle et trouver celui qui respecte les conditions (1.26). La résolution de ce système se fait algébriquement en inversant le système à l'aide de la règle de Cramer.

$$\begin{bmatrix} \rho \\ \alpha_j \\ \beta_j \end{bmatrix} = \frac{1}{\det(-\mathbf{d}, \mathbf{e}_1^j, \mathbf{e}_2^j)} \begin{bmatrix} \det(\mathbf{e}_3^j, \mathbf{e}_1^j, \mathbf{e}_2^j) \\ \det(-\mathbf{d}, \mathbf{e}_3^j, \mathbf{e}_2^j) \\ \det(-\mathbf{d}, \mathbf{e}_1^j, \mathbf{e}_3^j) \end{bmatrix};$$

où  $\mathbf{e}_1^j = \mathbf{X}_{1\phi}^j - \mathbf{X}_{0\phi}^j$ ,  $\mathbf{e}_2^j = \mathbf{X}_{2\phi}^j - \mathbf{X}_{0\phi}^j$  et  $\mathbf{e}_3^j = \mathbf{X}_r - \mathbf{X}_{0\phi}^j$ . On peut enfin exprimer tous ces déterminants à l'aide des relations entre déterminants et produits mixtes qui sont, pour rappel :

$$\det(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3) = -(\mathbf{e}_1 \wedge \mathbf{e}_3) \cdot \mathbf{e}_2 = -(\mathbf{e}_3 \wedge \mathbf{e}_2) \cdot \mathbf{e}_1 = -(\mathbf{e}_2 \wedge \mathbf{e}_1) \cdot \mathbf{e}_3, \quad \forall (\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3) \in (\mathbb{R}^3)^3$$

Ainsi, avec l'introduction des produits vectoriels  $\mathbf{p}^j = \mathbf{d} \wedge \mathbf{e}_2^j$  et  $\mathbf{q}^j = \mathbf{e}_3^j \wedge \mathbf{e}_2^j$ , pour calculer la position du nouveau point, il suffit de réaliser les opérations suivantes :

$$\begin{bmatrix} \rho \\ \alpha_j \\ \beta_j \end{bmatrix} = \frac{1}{(\mathbf{d} \wedge \mathbf{e}_2^j) \cdot \mathbf{e}_1^j} \begin{bmatrix} -(\mathbf{e}_1^j \wedge \mathbf{e}_3^j) \cdot \mathbf{e}_2^j \\ (\mathbf{d} \wedge \mathbf{e}_2^j) \cdot \mathbf{e}_3^j \\ (\mathbf{e}_3^j \wedge \mathbf{e}_1^j) \cdot \mathbf{d} \end{bmatrix} = \frac{1}{\mathbf{p}^j \cdot \mathbf{e}_1^j} \begin{bmatrix} \mathbf{q}^j \cdot \mathbf{e}_2^j \\ \mathbf{p}^j \cdot \mathbf{e}_3^j \\ \mathbf{q}^j \cdot \mathbf{d} \end{bmatrix}$$

et de vérifier les conditions d'admissibilité (1.26). Toute cette méthodologie est résumée dans l'algorithme 1.4.

Outre le fait que la méthode est la plus rapide et la moins gourmande en mémoire de la littérature, il est possible de regrouper toutes les projections ayant le même vecteur de direction de projection  $\mathbf{d}$  et de vectoriser les opérations, ce qui permet de faire des projections par blocs plus rapidement dans des langages interprétés.

---

**Algorithme 1.4 :** Intersection rayon/triangle de MÖLLER et TRUMBORE [1997]

---

**Données :** Une direction  $(\mathbf{X}_r, \mathbf{d})$  ;  
**pour** triangle  $j \in (\mathbf{X}_{0\phi}^j, \mathbf{X}_{1\phi}^j, \mathbf{X}_{2\phi}^j)_j$  **faire**  
     $\mathbf{e}_1 = \mathbf{X}_{1\phi}^j - \mathbf{X}_{0\phi}^j$  ;  
     $\mathbf{e}_2 = \mathbf{X}_{2\phi}^j - \mathbf{X}_{0\phi}^j$  ;  
     $\mathbf{e}_3 = \mathbf{X}_r - \mathbf{X}_{0\phi}^j$  ;  
     $\mathbf{p} = \mathbf{d} \wedge \mathbf{e}_2$  ;  
     $\mathbf{q} = \mathbf{e}_3 \wedge \mathbf{e}_2$  ;  
     $\Delta = \mathbf{p} \cdot \mathbf{e}_1$  ;  
     $\alpha_j = \frac{\mathbf{p} \cdot \mathbf{e}_3}{\Delta}$  ;  
     $\beta_j = \frac{\mathbf{q} \cdot \mathbf{d}}{\Delta}$  ;  
    **si**  $\alpha_j \geq 0$  **et**  $\beta_j \geq 0$  **et**  $(\alpha_j + \beta_j) \leq 1$  **alors**  
        | retourner  $\rho = \frac{\mathbf{q} \cdot \mathbf{e}_2}{\Delta}$   
    **fin**  
**fin**

---

### 1.3.2 Déformation de maillage

Une fois le déplacement  $\mathbf{u}(\mathbf{X}_b) = \mathbf{u}_b$  des nœuds de bord du domaine  $\Omega_r$  connu, il faut encore déplacer les nœuds intérieurs au domaine car ceux-ci n'ont encore subi aucun traitement. C'est pourquoi cette partie s'attachera à décrire les méthodes principales permettant de réaliser cette fonction et de les comparer. Une revue de ces méthodes a été rendue disponible par SELIM et KOMULLIL [2016]. Ces approches se classent en deux grandes familles qui se distinguent par l'exploitation ou non de la connectivité du maillage.

#### 1.3.2.1 Techniques nécessitant une connectivité de maillage

Ces méthodes nécessitent la connectivité entre les nœuds afin de décrire les  $n_a$  arêtes ou les  $n_e$  éléments (en fonction des approches).

**Analogie avec des ressorts** Dans ces méthodes, développées par BATINA [1990]; BLOM [2000] pour la déformation de maillage, chaque arête du maillage entre deux nœuds  $i$  et  $j$  est considérée en tant que ressort de traction-compression de raideur  $k_{ij} = k_{ji}$  et de longueur à vide correspondant à la distance entre les nœuds, soit  $\|\mathbf{X}_i - \mathbf{X}_j\|$ . Le nombre de nœuds connectés au nœud  $i$  par une arête est noté  $n_i$ . L'isolation statique des efforts  $\mathbf{F}_i$  s'appliquant au nœud  $i$  se traduit par :

$$\mathbf{F}_i = \sum_{j=1}^{n_i} k_{ij} (\mathbf{u}(\mathbf{X}_i) - \mathbf{u}(\mathbf{X}_j)).$$

Ces méthodes supposent que l'équilibre statique est atteint et peut finalement s'écrire :

$$\sum_{j=1}^{n_i} k_{ij} (\mathbf{u}(\mathbf{X}_i) - \mathbf{u}(\mathbf{X}_j)) = 0.$$

Dans l'exemple de la figure 1.28, le système à résoudre est :

$$\begin{bmatrix} k_{13} & 0 & -k_{13} & 0 \\ 0 & k_{23} & -k_{23} & 0 \\ -k_{13} & k_{23} & k_{13} + k_{23} + k_{34} & -k_{34} \\ 0 & 0 & -k_{34} & k_{34} \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{u}_3 \\ \mathbf{u}_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

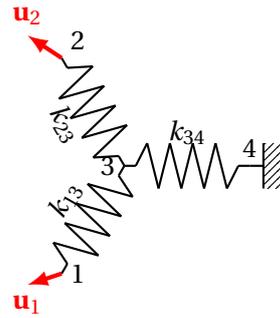


FIGURE 1.28 – Exemple de trois nœuds reliés au même quatrième nœud à l'aide de ressorts de traction/compression.

L'inconvénient de cette méthode est qu'elle peut conduire bien souvent à des retournements d'éléments. D'autres méthodes pour éviter cet écueil ajoutent des ressorts supplémentaires entre les nœuds et les faces de tout le **patch d'éléments** comme préconisé par **BOTTASSO et coll. [2005]**; **MARKOU et coll. [2007]**. D'autres auteurs comme **FARHAT et coll. [1998]**, ajoutent des ressorts de torsion à chaque sommet (*i.e.* entre deux arêtes), mais cela ajoute un surcoût de calcul important en raison de l'ajout de nouveaux **DDLs** angulaires.

**Problème d'élasticité linéaire « Retour élastique »** Ici, le problème revient à un problème **EF** où des conditions limites de Dirichlet  $\mathbf{u}(\mathbf{X}_b) = \mathbf{u}_b$  sont imposées sur tout le bord. Dans le but d'obtenir une bonne qualité de maillage, les coefficients matériau tels que le module d'Young  $E$  et le coefficient de Poisson  $\nu$  sont variables dans chaque élément. En effet, le choix de  $E$  et  $\nu$  plus raides dans certaines zones permet d'éviter des retournements d'éléments. Cela peut être fait en suivant plusieurs critères :

- **KARMAN [2007]** propose un critère sur  $E$  et  $\nu$  dépendant de la forme et de la position des éléments.
- Un critère dépendant de la qualité des éléments a été proposé par **FREITAG et KNUPP [2002]**.
- La procédure coûteuse de **YANG et MAVRIPLIS [2007]** permet, à l'aide d'un problème adjoint, d'optimiser les paramètres  $E$  et  $\nu$ .
- L'énergie potentielle d'une déformation antérieure permet de définir un critère sur  $E$  et  $\nu$  comme proposé par **HSU et coll. [2004]**.

**Problème en Laplacien** Une dernière approche est basée sur la résolution de l'équation de Laplace en imposant uniquement des conditions limites de Dirichlet :  $\Delta \mathbf{u} = 0$ . Cela revient à imposer  $E$  constant,  $\nu = 0$  et une cinématique incompressible ( $\text{div}(\mathbf{u}) = 0$ ) dans la méthode du paragraphe précédent. Rien ne permet d'affirmer que les éléments obtenus ne soient retournés ou présentent une bonne qualité.

### 1.3.2.2 Techniques sans maillage (*meshless*)

L'autre grande famille de méthode est celle fondée sur des approches ignorant la connectivité du maillage car généralement issues de l'interpolation sur des nuages de points. En effet, le déplacement de certains nœuds est interpolé à partir du déplacement connu d'autres nœuds. Ici, les déplacements obtenus n'ont, en général, pas de sens « physique » en comparaison des méthodes avec un maillage obtenu par la résolution d'un problème plus physique. Dans toute cette partie sera noté  $(\mathbf{X}_{bj})_{j \in \llbracket 1; n_b \rrbracket}$  l'ensemble des  $n_b$  nœuds dont les déplacements sont déjà connus et qui serviront à bâtir l'interpolation.

**TransFinite Interpolation (TFI)** La méthode dite *TFI* utilisée pour la déformation de maillages structurés par **BYUN et GURUSWAMY [1998]**, permet d'interpoler le déplacement des nœuds dont le déplacement est connu. Des courbes paramétriques sont définies sur l'espace paramétrique à l'aide des lignes du maillage structuré. Une pondération du déplacement peut ainsi être définie à l'aide de la distance sur ces courbes. Ainsi, le déplacement d'un nœud dans le volume situé sur la courbe paramétrique commençant sur le point  $\mathbf{X}_{b1}$  et se terminant en  $\mathbf{X}_{b2}$  se met sous la forme :

$$\mathbf{u}(\mathbf{X}) = \ell(\mathbf{X}) \mathbf{u}(\mathbf{X}_{b1}) + (1 - \ell(\mathbf{X})) \mathbf{u}(\mathbf{X}_{b2})$$

où  $\ell$  est l'abscisse curviligne normalisée de cette courbe paramétrique.

Cette méthode est efficace numériquement mais elle souffre de problèmes de robustesse car elle ne permet pas d'éviter le problème de retournement d'éléments. En plus, son utilisation est bornée au cas de maillages structurés où il est possible de définir des courbes paramétriques entre deux nœuds dont le déplacement est connu.

**Méthode de l'amortissement algébrique** Cette méthode, introduite par **ZHAO et FORHAD [2003]**, consiste à associer à chaque nœud à interpoler, le plus proche nœud dont le déplacement est connu. Ainsi, le déplacement d'un nœud à la position  $\mathbf{X}$  dépend uniquement du déplacement de son nœud associé  $\mathbf{X}_b$  et de sa distance par rapport à celui-ci :

$$\mathbf{u}(\mathbf{X}) = f(\|\mathbf{X} - \mathbf{X}_b\|) \mathbf{u}(\mathbf{X}_b),$$

avec  $f$  une fonction générale de la distance dont l'expression dépend de deux autres fonctions exponentielles amorties. L'expression de  $f$  est la suivante :

$$f(x) = \frac{f_2^2(x)}{f_1^2(x) + f_2^2(x)}.$$

Les deux fonctions exponentielles amorties sont définies telles que :

$$f_1(x) = \frac{e}{e-1} \left( 1 - \exp\left(-\frac{x}{d_{\max}}\right) \right),$$

$$f_2(x) = \frac{1}{1-e} \left( 1 - \exp\left(1 - \frac{x}{d_{\max}}\right) \right),$$

avec  $e = \exp(1)$  et  $d_{\max} = \max_{\substack{\mathbf{X} \in \Omega_r \\ \mathbf{X}_b \in \partial\Omega_r}} \|\mathbf{X} - \mathbf{X}_b\|$ . Comme le montre la figure 1.29, la fonction tend

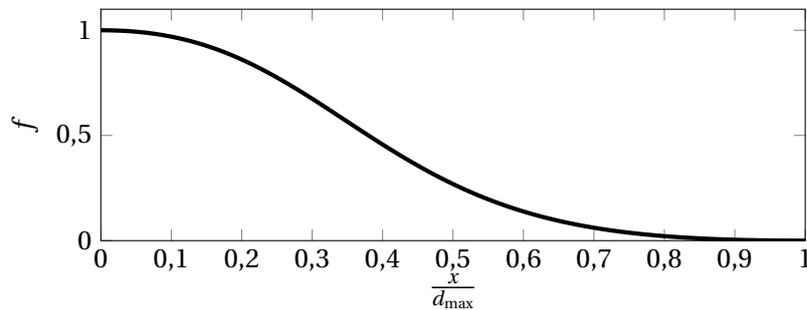


FIGURE 1.29 – Fonction utilisée par la méthode de l'amortissement algébrique

vers 1 lorsque  $x$  tend vers 0 et vers 0 lorsque  $x$  tend vers  $d_{\max}$ . Cela a pour conséquence de déformer fortement le maillage dans les zones proches des nœuds dont le déplacement est connu et de rigidifier le maillage dans les zones plus éloignées.

Cette méthode peut conduire à créer des éléments retournés quand l'amplitude des déplacements connus est importante. C'est pourquoi un lissage du maillage est souvent nécessaire pour améliorer sa qualité.

### Méthode de l'*Inverse Distance Weighting* ou pondération par l'inverse de la distance (*IDW*)

Comme pour la méthode précédente, l'*IDW* permet d'exprimer explicitement le déplacement interpolé dépendant des nœuds dont le déplacement est connu. WITTEVEEN [2010] a exprimé cette relation à l'aide d'une fonction de pondération  $w$  dépendant de l'inverse de la distance entre les nœuds. L'expression du déplacement se met sous la forme :

$$\mathbf{u}(\mathbf{X}) = \frac{\sum_{k=1}^{n_b} \mathbf{u}(\mathbf{X}_{bk}) w(\|\mathbf{X} - \mathbf{X}_{bk}\|)}{\sum_{k=1}^{n_b} w(\|\mathbf{X} - \mathbf{X}_{bk}\|)};$$

et la fonction de pondération est une fonction inverse à la puissance  $c$  de la forme  $w(x) = 1/x^c$ . Le paramètre  $c$  permet de moduler l'influence de la distance sur le déplacement. En effet, le déplacement connu d'un nœud aura une influence plus localisée autour de ce nœud si  $c$  a une valeur importante. En pratique, un choix de  $c$  égal à 2 donne des résultats satisfaisants d'après WITTEVEEN [2010].

**Interpolation par *Radial Basis Functions* ou fonctions de base radiales (*RBF*)** Cette méthode décrite par DE BOER et coll. [2007] et implémentée dans le logiciel de *Computational Fluid Dynamics (CFD)* ANSYS [2016] pour la déformation de maillage dans des simulations de *CFD*. Cette technique est issue d'une théorie plus large sur l'interpolation de données multivariées à l'aide de *RBFs* conçue par BUHMANN [2003]. Ici, chaque composante spatiale du champ déplacement est une variable à interpoler grâce aux déplacements connus. La formulation du champ  $\mathbf{u}$  interpolé pour chaque direction  $j \in \llbracket 1; d \rrbracket$  avec  $d$  la dimension de l'espace d'interpolation, se met sous la forme :

$$u_j(\mathbf{X}) = \sum_{k=1}^{n_b} a_k^j \psi(\|\mathbf{X} - \mathbf{X}_{bk}\|) + p^j(\mathbf{X}), \quad (1.27)$$

où  $\psi$  est une fonction de base radiale choisie parmi celles présentées dans le tableau 1.3 et  $p$  est un polynôme du premier ordre permettant de décrire exactement les mouvements de corps rigides.

#### Remarque 1.9

Les polynômes  $p^j$  ont des vertus régularisantes pour le problème d'interpolation lorsque la fonction  $\psi$  n'est pas *définie positive* (voir les définitions de BUHMANN [2000] ou de WENDLAND [1995]). Dans le cas où  $\psi$  est choisie comme étant une fonction du tableau 1.3, elle est définie positive et l'utilisation du polynôme  $p$  n'est plus obligatoire d'après BUHMANN [2000]. Récemment, plusieurs auteurs comme KEDWARD et coll. [2017]; XIE et LIU [2017] obtiennent de bons résultats sans utiliser ce polynôme. En sus, dans nos applications à des maillages qui, finalement, n'auront pas de déplacements de solides rigides, l'utilisation de ce polynôme peut être omise.

Les coefficients  $(a_k^j)_{1 \leq k \leq n_b}$  et  $(b_l^j)_{1 \leq l \leq d+1}$  du polynôme  $p^j$  sont déterminés par les conditions d'interpolation et d'admissibilité suivantes :

$$\begin{aligned} u_j(\mathbf{X}_b) &= u_{bj}, \quad \forall j \in \llbracket 1; d \rrbracket, \\ \sum_{k=1}^{n_b} a_k^j q^j(\mathbf{X}) &= 0, \quad \forall q \text{ polynôme de degré inférieur ou égal à } p. \end{aligned}$$

La seconde relation vient du fait que les coefficients de  $p$  sont recherchés comme étant dans l'espace orthogonal à celui des polynômes d'ordre inférieurs au degré de  $p$  d'après BUHMANN [2000]. Les conditions sus-mentionnées conduisent à la formulation du système linéaire *RBF* :

$$\begin{bmatrix} \Psi & \mathbf{P}_b \\ \mathbf{P}_b^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{a}^j \\ \mathbf{b}^j \end{bmatrix} = \begin{bmatrix} \mathbf{u}_{bj} \\ 0 \end{bmatrix}. \quad (1.28)$$

où  $\Psi$  est une matrice carrée symétrique conditionnellement définie positive<sup>6</sup> de taille  $n_b$  définie telle que  $\Psi_{ij} = \psi(\|\mathbf{X}_i - \mathbf{X}_{bj}\|)$  et  $\mathbf{P}_b$  une matrice rectangulaire de taille  $n_b \times (d+1)$  définie telle

6. La condition est que la fonction  $\psi$  soit définie positive.

que la  $i^{\text{ème}}$  ligne de  $\mathbf{P}_b$  soit  $P_{bi} = [1 \quad \mathbf{X}_i]$ . On notera que l'interpolation ne couple pas les dimensions de l'espace physique, si bien que la même matrice s'applique simultanément à chacune des  $d$  directions dans l'équation (1.28), seul le second membre change avec la direction de l'espace considérée.

Parmi les 14 fonctions du tableau 1.3, les huit premières sont des fonctions à support compact sur l'intervalle  $[0, 1]$ , c'est-à-dire qu'à partir d'un certain rayon d'influence  $r$ , la fonction devient nulle. Les fonctions  $\psi$  sont remises à l'échelle à l'aide de ce rayon d'influence. Ainsi, il est possible de faire le changement de variable pour les fonctions  $\psi$  de la forme  $\xi \equiv x/r$ . Ces fonctions sont donc définies telles que :

$$\psi(\xi) = \begin{cases} \psi(\xi) & \text{si } \xi \in [0, 1], \\ 0 & \text{sinon.} \end{cases}$$

L'utilisation de telles fonctions implique que seuls les nœuds à l'intérieur de la sphère de rayon  $R$  centrée sur un nœud  $\mathbf{X}_b$  sont affectés par le déplacement de ce nœud. De plus, ce choix de fonctions à support compact conduit à diminuer la largeur de bande de  $\Psi$  car les nœuds  $\mathbf{X}_b$  ne sont plus tous couplés si une valeur de  $R$  « raisonnablement » faible est choisie.

Parmi ces huit fonctions, les quatre premières sont les fonctions *Continuous Polynomials (CP)* : les polynômes de degrés minimaux qui sont  $\mathcal{C}^n$  continus avec  $n \in \{0, 2, 4, 6\}$ . Les quatre suivantes sont des séries de fonctions basées sur la théorie *Thin Plate Spline (TPS)* de DUCHON [1976, 1977]. De la même façon, ces fonctions *Continuous TPS (CTPS)* sont les formes minimales pour obtenir la  $\mathcal{C}^n$  continuité avec  $n \in \{0, 1, 2\}$ . Dans la théorie de WENDLAND [1996], deux fonctions  $\mathcal{C}^2$  continues sont possibles ; elles sont indicées par  $a$  et  $b$ . Les six autres fonctions, à support global, couvrent l'ensemble du domaine d'interpolation, ce qui conduit à des opérateurs  $\Psi$  denses dont la résolution peut devenir numériquement plus coûteuse. Parmi ces fonctions, *MultiQuadratic Biharmonics (MQB)* et *Inverse MQB (IMQB)* se voient attribuées un paramètre  $a$  qui contrôle la largeur des fonctions.

No.	Nom	$\psi(\xi)$
(a)	<i>CP</i> $\mathcal{C}^0$	$(1 - \xi)^2$
(b)	<i>CP</i> $\mathcal{C}^2$	$(1 - \xi)^4 (4\xi + 1)$
(c)	<i>CP</i> $\mathcal{C}^4$	$(1 - \xi)^6 (\frac{35}{3}\xi^2 + 6\xi + 1)$
(d)	<i>CP</i> $\mathcal{C}^6$	$(1 - \xi)^8 (32\xi^3 + 25\xi^2 + 8\xi + 1)$
(e)	<i>CTPS</i> $\mathcal{C}^0$	$(1 - \xi)^5$
(f)	<i>CTPS</i> $\mathcal{C}^1$	$1 + \frac{80}{3}\xi^2 - 40\xi^3 + 15\xi^4 - \frac{8}{3}\xi^5 + 20\xi^2 \log(\xi)$
(g)	<i>CTPS</i> $\mathcal{C}_a^2$	$1 - 30\xi^2 - 10\xi^3 + 45\xi^4 - 6\xi^5 - 60\xi^3 \log(\xi)$
(h)	<i>CTPS</i> $\mathcal{C}_b^2$	$1 - 20\xi^2 + 80\xi^3 - 45\xi^4 - 16\xi^5 + 60\xi^4 \log(\xi)$
(i)	<i>MQB</i>	$\sqrt{a^2 + \xi^2}$
(j)	<i>IMQB</i>	$\sqrt{\frac{1}{a^2 + \xi^2}}$
(k)	<i>Quadratic Biharmonics (QB)</i>	$1 + \xi^2$
(l)	<i>Inverse QB (IQB)</i>	$\frac{1}{1 + \xi^2}$
(m)	<i>TPS</i>	$\xi^2 \log(\xi)$
(n)	Gaussien	$e^{-\frac{\xi^2}{2}}$

TABLEAU 1.3 – Fonctions de base radiales dont la représentation est donnée sur l'intervalle  $[0, 2]$  sur la figure 1.30

Cette méthode utilisant des *RBFs* a l'avantage de produire un maillage de bonne qualité comme l'affirme SELIM et KOMULLIL [2016] en comparaison d'autres méthodes et notamment celles citées dans les paragraphes précédents.

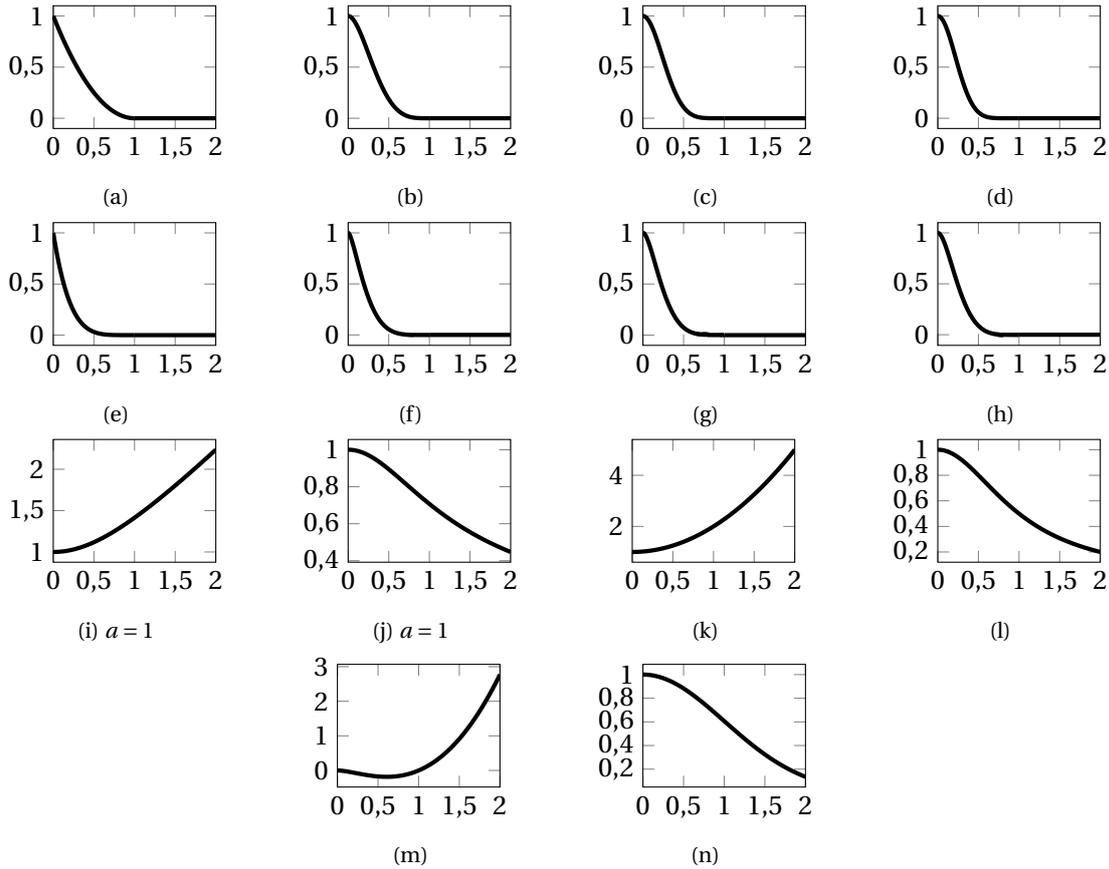


FIGURE 1.30 – Tracé des différentes fonctions *RBF*

### 1.3.2.3 Comparaison de ces méthodes

Les différentes méthodes sont comparées à l'aide du tableau 1.4 extrait de SELIM et KOMULLIL [2016]. Dans ce tableau,  $n_v$  désigne le nombre total de point à interpoler alors que  $n_a$  désigne le nombre d'arêtes présentes dans le maillage.

En général, les seules méthodes qui permettent l'obtention d'un maillage déformé de manière efficace et robuste sont les méthodes *IDW* et *RBF*. L'avantage d'*IDW* réside dans la complexité algorithmique linéaire de la méthode mais conduit à des éléments de moins bonne qualité que pour la méthode *RBF*. Par contre la méthode *RBF* implique de résoudre un système linéaire de taille  $n_b$  qui a une complexité cubique et semble coûteux. Mais ce système est de petite taille, car, en général  $n_b \ll n_v$  et il est possible d'exploiter le caractère creux de l'opérateur pour accélérer la résolution. Si  $n_b$  est du même ordre de grandeur que  $n_v$ , il faut avoir recours à des techniques d'optimisation de la méthode *RBF* comme ce sera développé dans le chapitre 2 pour se ramener à une complexité quasi-linéaire.

### 1.3.2.4 Comparaison des différents types de fonctions *RBF*

Dans la suite de ce travail, la méthode *RBF* a été retenue pour la déformation de maillage. Une étude est réalisée quant à l'influence du choix de la fonction de base *RBF* au regard de la qualité des éléments obtenus après déformation. Cette étude vise à corroborer ou à réfuter les conclusions qui ont pu être faites dans DE BOER et coll. [2007] ou encore RENDALL et ALLEN [2009] par leurs tests numériques quant à l'efficacité de la fonction (b) du tableau 1.3 :  $CP^{\mathcal{C}^2}$  par rapport aux autres fonctions *RBF*.

Le problème sur lequel la méthode est étudié consiste en un problème bidimensionnel carré de côté  $H$  illustré sur la figure 1.31. Ce carré est soumis à un déplacement sinusoïdal sur son arête

Méthode	Avantages	Inconvénients	Complexité algorithmique
Ressorts linéaires	Simple et facile à implémenter	Retournement d'éléments	$O(n_v^3)$
Ressorts de torsion	Robuste	Coûteux	$O(n_a^3 + n_v^3)$
Elasticité linéaire	Facile à implémenter dans un code EF	Choix de E et $\nu$ compliqué	$O(n_v^3)$
TFI	Simple et peu coûteux	Retournement d'éléments	$O(n_v)$
Amortissement algébrique	Simple et peu coûteux	Très rigide près des bords et très élastique ailleurs	$O(n_v)$
IDW	Simple et peu coûteux	Pas d'inconvénients majeurs	$O(n_v)$
RBF	Robuste et conserve une bonne qualité de maillage	Pas d'inconvénients majeurs	$O(n_b^3)$

TABEAU 1.4 – Avantages, inconvénients et complexité algorithmique des techniques de déformation de maillage (issu de SELIM et KOMULLIL [2016])

supérieure de la forme :

$$\mathbf{u}(x, y = H) = \frac{H}{5} \sin\left(2\pi \frac{x}{H}\right) \mathbf{e}_y,$$

tandis que le déplacement des autres arêtes est imposé à une valeur nulle. On choisit de tes-

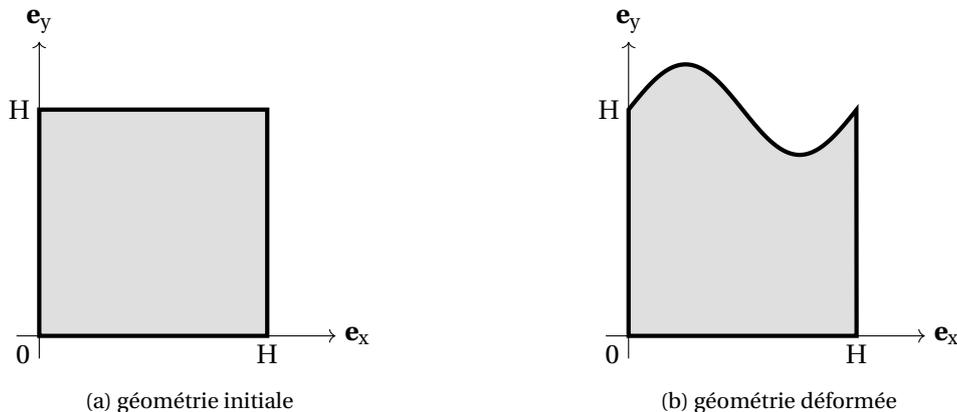


FIGURE 1.31 – Problème bidimensionnel de référence pour le test des fonction *RBF*

ter cette méthode sur cette géométrie qui sera discrétisée de plusieurs manières grâce au logiciel *Gmsh* développé par GEUZAIN et REMACLE [2009] comme le montre la figure 1.32 :

- un maillage triangulaire non-structuré (figure 1.32a) ;
- un maillage quadrangulaire non-structuré (figure 1.32b) ;
- un maillage triangulaire structuré (figure 1.32c) ;
- un maillage quadrangulaire structuré (figure 1.32d).

**Mesure de qualité de maillage** Pour étudier l'influence de  $\psi$  sur la qualité des éléments, une métrique a été introduite par KNUPP [2003] et utilisée par DE BOER et coll. [2007] pour ses tests. KNUPP [2003] introduit en deux dimensions trois indicateurs de qualité :

- un indicateur de taille noté  $f_{size}$  ;
- un indicateur de forme qui quantifie l'écart à l'équilatéralité (écart par rapport à un triangle équilatéral ou un carré en fonction du type d'élément) et sera noté  $f_{shape}$  ;

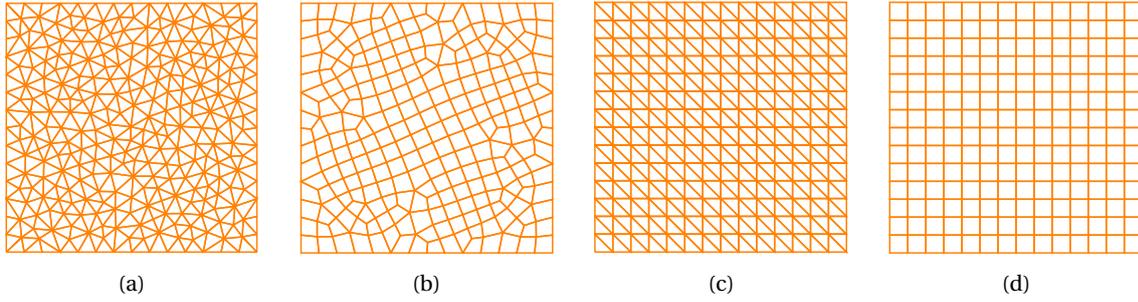


FIGURE 1.32 – Différents maillages testés

— un indicateur quantifiant la perte de parallélisme des arêtes en vis-à-vis dans un quadrangle noté  $f_{\text{skew}}$ .

Le premier indicateur  $f_{\text{size}}$  est défini de la manière suivante :

$$f_{\text{size}} = \min\left(\tau, \frac{1}{\tau}\right),$$

avec  $\tau = S/S_0$  dont  $S$  est la surface de l'élément et  $S_0$  celle de l'élément de référence dans la structure. Ainsi cet indicateur a une valeur proche de 1 lorsque les éléments ont une taille uniforme sur tout le maillage et devient faible si les éléments ont des tailles trop variables. Le second indicateur a une définition différenciée dans le cas d'un triangle ou d'un quadrangle. Considérons d'abord un triangle dont les trois sommets sont localisés en  $(\mathbf{X}_k)$  avec  $k \in \llbracket 0, 2 \rrbracket$ . Avec cette notation, trois matrices jacobiniennes au nœud  $k$  peuvent être définies :  $\mathbf{J}_k = [\mathbf{X}_{k+1} - \mathbf{X}_k \quad \mathbf{X}_{k+2} - \mathbf{X}_k]$ . Les indices sont pris modulo 3 de telle sorte que, par exemple pour  $k = 1$ ,  $k + 2$  devienne 0. À partir de ces matrices jacobiniennes, un tenseur symétrique peut aussi être défini comme étant  $\lambda_k = \mathbf{J}_k^T \mathbf{J}_k$ . On peut remarquer que  $J_k = \det(\mathbf{J}_k) = J$  est indépendant de  $k$ <sup>7</sup> et en déduire une expression de l'indicateur de forme indépendante du nœud auquel il est calculé :

$$f_{\text{shape}} = \frac{\sqrt{3}J}{\lambda_{11} + \lambda_{22} - \lambda_{12}}$$

Pour les quadrangles, quatre matrices  $\mathbf{J}_k$  peuvent être calculées de la même façon que précédemment mais avec  $k \in \llbracket 0, 3 \rrbracket$  et en prenant les indices au modulo 4, cet indicateur est exprimé par la relation :

$$f_{\text{shape}} = \frac{8}{\sum_{k=0}^3 \frac{\lambda_{11}^k + \lambda_{22}^k}{J_k}}$$

Ces indicateurs valent 1 lorsque l'élément se rapproche de l'élément régulier qui lui est associé (le triangle équilatéral ou le carré) et 0 lorsqu'ils sont dégénérés.

Le troisième indicateur  $f_{\text{skew}}$  est défini uniquement pour un quadrangle. En effet, il mesure le défaut de parallélisme entre deux arêtes en vis-à-vis et est défini par l'expression :

$$f_{\text{skew}} = \frac{4}{\sum_{k=0}^3 \frac{\sqrt{\lambda_{11}^k \lambda_{22}^k}}{J_k}}$$

Cet indicateur vaut 1 lorsque l'élément est un rectangle et il vaut 0 lorsque l'élément est dégénéré. À la différence de  $f_{\text{shape}}$ ,  $f_{\text{skew}}$  peut être maximal quel que soit le rapport d'aspect de l'élément, même pour des rectangles très élancés, d'où son intérêt. De la même façon que **KNUPP [2003]**, des indicateurs évaluant à la fois la taille et la forme (ou l'inclinaison des arêtes) peuvent être définis. Ils sont caractérisés par les produits des indicateurs précédemment introduits :

$$\begin{aligned} f_{\text{size-shape}} &= f_{\text{size}} f_{\text{shape}}, \\ f_{\text{size-skew}} &= f_{\text{size}} f_{\text{skew}}. \end{aligned}$$

7. Il représente le double de la surface du triangle.

**Essais numériques** Les indicateurs précédemment introduits pouvant devenir négatifs, les valeurs négatives seront seuillées à 0 dans un soucis de visualisation. Pour les fonctions à support compact, l'influence du rayon d'action est étudiée dans un domaine de variation allant de  $H/3$  à  $H$ . Les déformations sont comparées à une référence réalisée à l'aide d'une méthode que l'on a appelé « Retour Elastique » et qui correspond à une méthode de type *problème d'élasticité linéaire* vue à la section 1.3.2.1 où  $\nu = 0$  et  $E$  constant ont été choisis en tant que paramètres. Ce problème est simplement un repère d'une méthode naïve de déformation de maillage qui peut déjà obtenir des déformation convenables. Les résultats sont présentés pour chacun des quatre maillages sur les figures 1.33, 1.34, 1.35 et 1.36.

Pour les fonctions à support global, la fonction *TPS* semble donner les meilleurs résultats. Dans tous les cas tests, et pour tous les critères, cela s'avère vrai mis-à-part l'indicateur  $f_{\text{size-shape}}$  pour le maillage 1.32b où l'interpolation reste moins bonne que la déformation linéaire élastique comme le montre la figure 1.34.

Pour chaque test sur les fonctions à support compact, le même comportement peut être observé : pour de faibles valeurs de  $r$ , la qualité du maillage est la meilleure pour la fonction *CP*  $\mathcal{C}^0$ . Mais lorsque le rayon augmente, pour cette fonction, l'indicateur de qualité atteint un maximum puis diminue alors qu'avec toutes les autres fonctions, les indicateurs croissent. La fonction qui a la valeur de qualité la plus importante à partir d'une certaine valeur de  $r$  est la fonction *CP*  $\mathcal{C}^2$ . Finalement, le choix de fonctions à support compact ou non ne peut se faire sur le critère de qualité de maillage seul. En effet, la largeur de bande de  $\Psi$  est bien plus grande si la fonction *TPS* est choisie en comparaison de sa largeur de bande pour *CP*  $\mathcal{C}^2$  même lorsque  $r = H$ . Pour des raisons d'efficacité de calcul, *CP*  $\mathcal{C}^2$  semble donc être la fonction la plus robuste pour permettre de réaliser une déformation de maillage.

On pourra noter aussi qu'avec une amplitude de déplacement supérieure à  $H/5$  la méthode fondée sur l'élasticité linéaire conduit à un maillage déformé possédant des éléments retournés alors que pour certaines fonctions  $\psi$  le maillage conserve une bonne qualité.

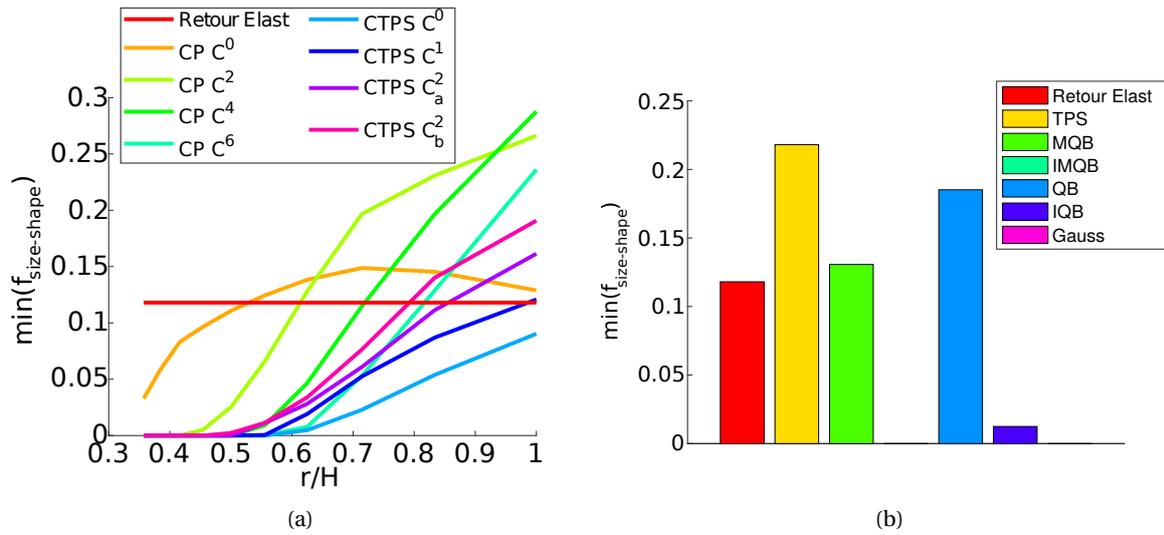


FIGURE 1.33 – Qualité *size-shape* du plus mauvais élément pour le maillage non-structuré triangulaire (figure 1.32a). La figure de gauche représente l'évolution de l'indicateur lorsque le rapport  $r/H$  évolue pour plusieurs RBF à support compact. La figure de droite représente la valeur de l'indicateur de qualité de maillage pour les différentes RBF à support global. Pour rappel, la qualité optimale est de 1 et un élément dégénéré à une qualité nulle.

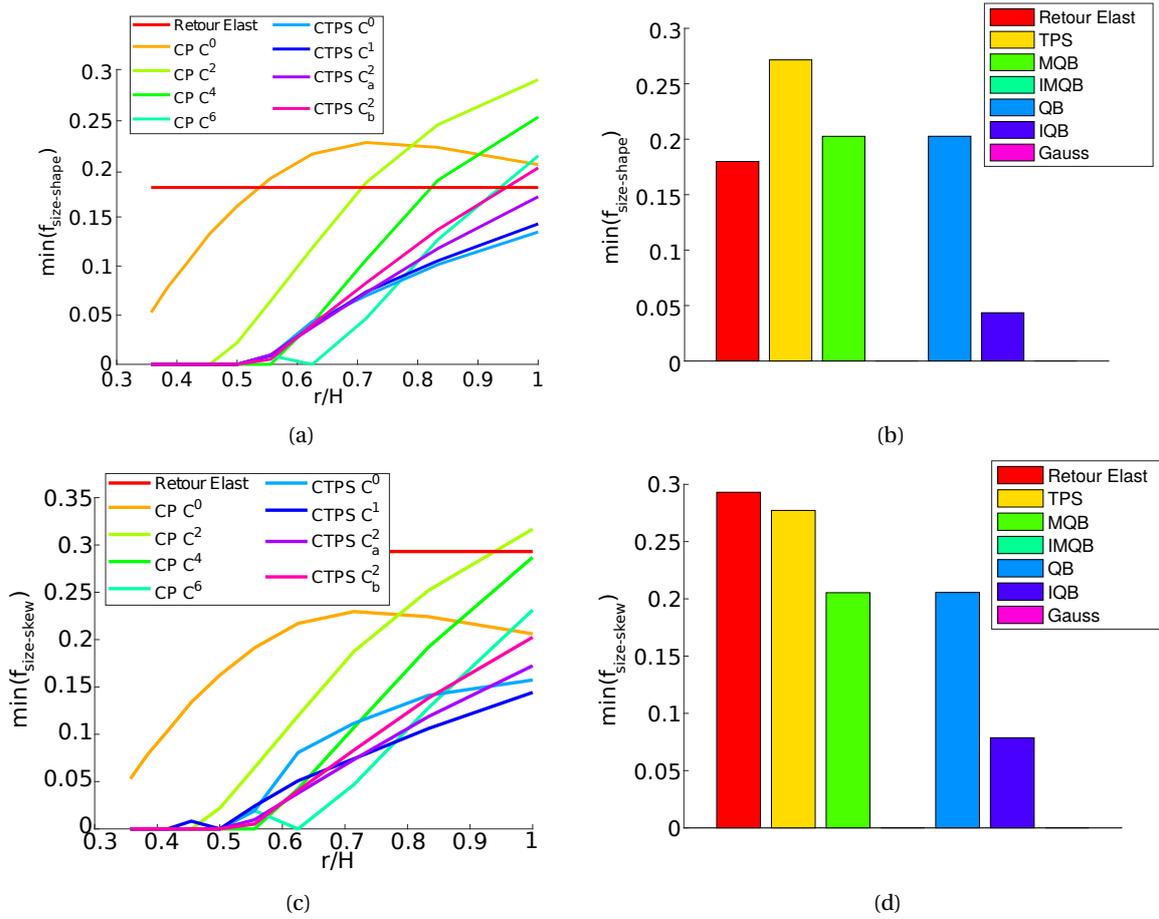


FIGURE 1.34 – Qualité du plus mauvais élément pour le maillage non-structuré quadrangulaire (figure (figure 1.32b)). Les figures de gauche représentent l'évolution de l'indicateur lorsque le rapport  $r/H$  évolue pour plusieurs RBF à support compact. Les figures de droite représentent la valeur de l'indicateur de qualité de maillage pour les différentes RBF à support global. Pour rappel, la qualité optimale est de 1 et un élément dégénéré à une qualité nulle.

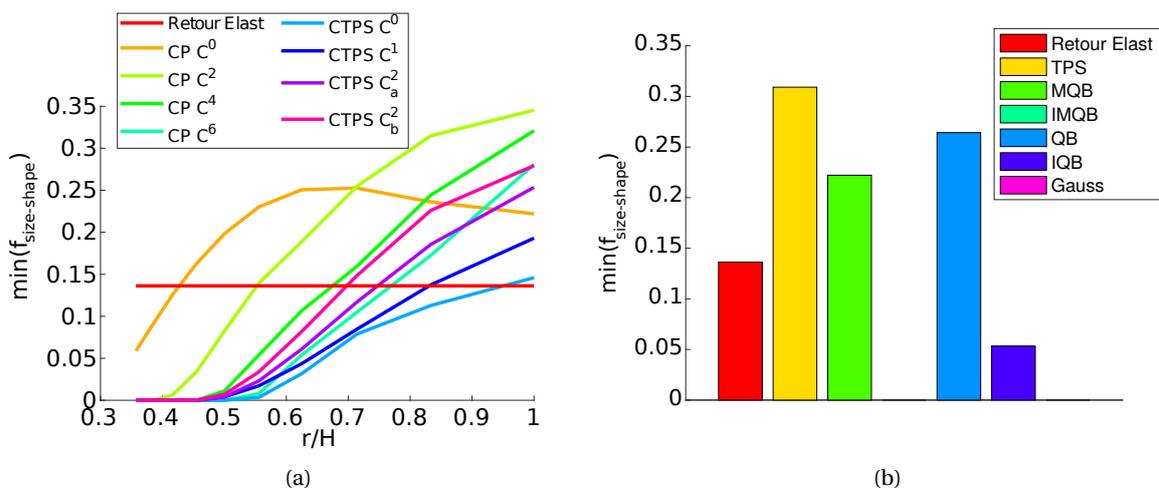


FIGURE 1.35 – Qualité *size-shape* du plus mauvais élément pour le maillage structuré triangulaire (figure (figure 1.32c)). La figure de gauche représente l'évolution de l'indicateur lorsque le rapport  $r/H$  évolue pour plusieurs RBF à support compact. La figure de droite représente la valeur de l'indicateur de qualité de maillage pour les différentes RBF à support global. Pour rappel, la qualité optimale est de 1 et un élément dégénéré à une qualité nulle.

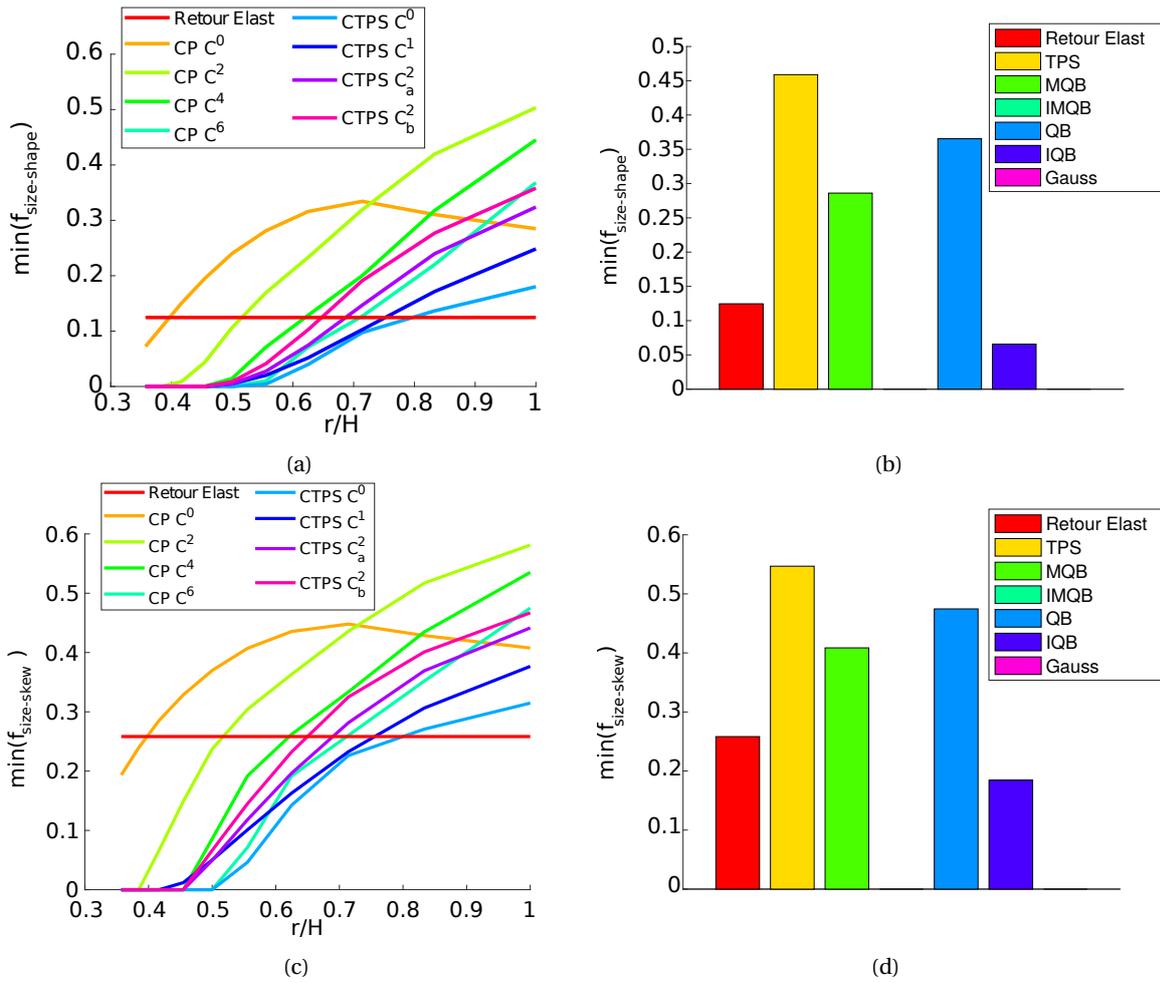


FIGURE 1.36 – Qualité du plus mauvais élément pour le maillage structuré quadrangulaire (figure 1.32d). Les figures de gauche représentent l'évolution de l'indicateur lorsque le rapport  $r/H$  évolue pour plusieurs RBF à support compact. Les figure de droite représentent la valeur de l'indicateur de qualité de maillage pour les différentes RBF à support global. Pour rappel, la qualité optimale est de 1 et un élément dégénéré à une qualité nulle.

## 1.4 Conclusion partielle

Dans ce chapitre, nous avons présenté les principales méthodes de décomposition de domaine sans recouvrement, qu'elles soient les plus classiques comme les approches primale et duale mais aussi d'autres approches hybrides. Nous avons aussi identifié que la formulation mixte de ROUX et coll. [2003] fait apparaître un paramètre critique pour la convergence du solveur qui est appelé l'impédance d'interface.

Dans un second volet de ce chapitre, nous avons vu que le pré-traitement parallèle était un champs de recherche important. Dans la communauté du partitionnement de graphe, il semble faire consensus que les approches multi-niveaux sont les plus efficaces comme le montre la quantité de logiciels qui en sont équipés exclusivement (Scotch, Metis et Chaco par exemple). Par contre, du côté du maillage en parallèle, si on veut pouvoir générer de manière efficace des maillages 3D sans faire trop d'échanges, il ne semble pas se dégager de réelle tendance même si un nombre important de travaux ont été réalisés pour la réutilisation de maillages séquentiels dans un contexte parallèle et que les travaux de LOSEILLE et coll. [2015] ou encore CHEN et coll. [2018] ont permis un remaillage efficace à grande échelle.

Enfin, nous avons abordé la thématique de la déformation de maillage qui se compose de deux étapes qui sont la projection de la surface externe du maillage et de la déformation du reste de la structure. Ici, il semble se dégager qu'une projection explicite de type MÖLLER et TRUMBORE [1997] permet de respecter les entités de la géométrie plus précisément que les méthodes de paramétrisations où la maîtrise de la zone de projection est mauvaise. Pour la déformation en elle-même, il semble se dégager que celle exploitant les RBFs est la technique la mieux adaptée aux petites déformations de maillages. Par contre, il faut inverser des problèmes relativement denses mais de faible taille. Nous allons voir par la suite ce qui a été développé dans cette thèse pour pallier ce problème.

## 1.5 Références

- AHN, M., S. LEE et H.-P. SEIDEL. 2004, «Connectivity transformation for mesh metamorphosis», dans *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, SGP '04, Association for Computing Machinery, Nice, France, ISBN 978-3-905673-13-5, p. 75–82, doi : [10.1145/1057432.1057443](https://doi.org/10.1145/1057432.1057443). URL <https://doi.org/10.1145/1057432.1057443>. 48
- ALEXA, M. 2000, «Merging polyhedral shapes with scattered features», *The Visual Computer*, vol. 16, n° 1, doi : [10.1007/PL00007211](https://doi.org/10.1007/PL00007211), p. 26–37, ISSN 1432-2315. URL <https://doi.org/10.1007/PL00007211>. 48
- AMESTOY, P. et I. DUFF. 2000, «MUMPS MULTifrontal Massively Parallel Solver Version 2.0», *Computer Methods in Applied Mechanics and Engineering*, vol. 184, p. 501–520. 11
- ANSYS. 2016, «ANSYS Fluent - CFD Software | ANSYS», URL <http://www.ansys.com/products/fluids/ansys-fluent>. 54
- AREIBI, S. et Z. YANG. 2004, «Effective memetic algorithms for VLSI design automation = genetic algorithms + local search + multi-level clustering», *Evolutionary Computation*, vol. 12, n° 3, doi : [10.1162/1063656041774947](https://doi.org/10.1162/1063656041774947), p. 327–353, ISSN 1063-6560. URL <https://doi.org/10.1162/1063656041774947>. 36
- ATHANASIADIS, T., I. FUDOS, C. NIKOU et V. STAMATI. 2012, «Feature-based 3D morphing based on geometrically constrained spherical parameterization», *Computer Aided Geometric Design*, vol. 29, n° 1, doi : <https://doi.org/10.1016/j.cagd.2011.09.004>. 48
- BADOUEL, D. 1990, «An efficient ray-polygon intersection», *Graphics Gems*, doi : [10.1016/B978-0-08-050753-8.50082-6](https://doi.org/10.1016/B978-0-08-050753-8.50082-6), p. 390–393. 49

- BARNARD, S. T. et H. D. SIMON. 1994, «Fast multilevel implementation of recursive spectral bisection for partitioning unstructured problems», *Concurrency: Practice and Experience*, vol. 6, n° 2, doi : [10.1002/cpe.4330060203](https://doi.org/10.1002/cpe.4330060203), p. 101–117, ISSN 10403108, 10969128. URL <http://doi.wiley.com/10.1002/cpe.4330060203>. 36, 37
- BARRETT, R., M. BERRY, T. F. CHAN, J. DEMMEL, J. DONATO, J. DONGARRA, V. EIJKHOUT, R. POZO, C. ROMINE et H. VAN DER VORST. 1994, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, Society for Industrial and Applied Mathematics, ISBN 978-0-89871-328-2 978-1-61197-153-8, doi : [10.1137/1.9781611971538](https://doi.org/10.1137/1.9781611971538). URL <http://epubs.siam.org/doi/book/10.1137/1.9781611971538>. 11
- BATINA, J. T. 1990, «Unsteady Euler airfoil solutions using unstructured dynamic meshes», *AIAA Journal*, vol. 28, n° 8, doi : [10.2514/3.25229](https://doi.org/10.2514/3.25229), p. 1381–1388, ISSN 0001-1452, 1533-385X. URL <http://arc.aiaa.org/doi/10.2514/3.25229>. 51
- BATTITI, R. et A. BERTOSSI. 1999, «Greedy, prohibition, and reactive heuristics for graph partitioning», *IEEE Transactions on Computers*, vol. 48, n° 4, doi : [10.1109/12.762522](https://doi.org/10.1109/12.762522), p. 361–385, ISSN 1557-9956. Conference Name: IEEE Transactions on Computers. 36
- BLANCHARD, M. 2018, *Méthode global/local non-intrusive pour les simulations cycliques non-linéaires*, phdthesis, Université Paris-Saclay. URL <https://tel.archives-ouvertes.fr/tel-01852574>. 12
- BLOM, F. J. 2000, «Considerations on the spring analogy», *International Journal for Numerical Methods in Fluids*, vol. 32, n° 6, doi : [10.1002/\(SICI\)1097-0363\(20000330\)32:6<647::AID-FLD979>3.0.CO;2-K](https://doi.org/10.1002/(SICI)1097-0363(20000330)32:6<647::AID-FLD979>3.0.CO;2-K), p. 647–668, ISSN 0271-2091, 1097-0363. 51
- BOTTASSO, C. L., D. DETOMI et R. SERRA. 2005, «The ball-vertex method: a new simple spring analogy method for unstructured dynamic meshes», *Computer Methods in Applied Mechanics and Engineering*, vol. 194, n° 39-41, doi : [10.1016/j.cma.2004.08.014](https://doi.org/10.1016/j.cma.2004.08.014), p. 4244–4264, ISSN 00457825. URL <http://linkinghub.elsevier.com/retrieve/pii/S0045782504005201>. 52
- BOVET, C., O. CIOBANU, A. PARRET-FREAUD et V. CHIARUTTINI. 2019a, «A parallel mesh refinement process tailored to domain decomposition methods», dans *Proceedings of the Sixth International Conference on Parallel, Distributed, GPU and Cloud Computing for Engineering*, Pollock Mihály Faculty of Engineering and Information Technology, University of Pécs, Hungary, p. 20, doi : [10.4203/ccp.112.20](https://doi.org/10.4203/ccp.112.20). URL <http://www.ctresources.info/ccp/paper.html?id=9287>. 45, 46, 47
- BOVET, C., O. CIOBANU et A. PARRET-FRÉAUD. 2019b, «Méthodes de remaillage parallèle pour le calcul intensif de structures industrielles», dans *14ème Colloque National en Calcul des Structures*, Giens, France, p. 9. URL <https://hal.archives-ouvertes.fr/hal-02306095>. 45, 46
- BOVET, C., A. PARRET-FRÉAUD, N. SPILLANE et P. GOSSELET. 2017, «Adaptive multipreconditioned FETI: Scalability results and robustness assessment», *Computers & Structures*, vol. 193, doi : [10.1016/j.compstruc.2017.07.010](https://doi.org/10.1016/j.compstruc.2017.07.010), p. 1–20, ISSN 00457949. URL <https://linkinghub.elsevier.com/retrieve/pii/S004579491730367X>. 33
- BOWYER, A. 1981, «Computing Dirichlet tessellations», *The Computer Journal*, vol. 24, n° 2, doi : [10.1093/comjnl/24.2.162](https://doi.org/10.1093/comjnl/24.2.162), p. 162–166, ISSN 0010-4620. URL <https://academic.oup.com/comjnl/article/24/2/162/338193>, publisher: Oxford Academic. 39
- BUHMANN, M. D. 2000, «Radial basis functions», *Acta Numerica*, vol. 9, doi : [10.1017/S0962492900000015](https://doi.org/10.1017/S0962492900000015), p. 1–38, ISSN 0962-4929, 1474-0508. URL [https://www.cambridge.org/core/product/identifiler/S0962492900000015/type/journal\\_article](https://www.cambridge.org/core/product/identifiler/S0962492900000015/type/journal_article). 54

- BUHMANN, M. D. 2003, *Radial basis functions: theory and implementations*, Cambridge University Press, Cambridge; New York, ISBN 9780511040207 9780511543241 9786610432295 9780511051241. URL <https://doi.org/10.1017/CB09780511543241>, oCLC: 56352083. 54
- BYUN, C. et G. GURUSWAMY. 1998, «A parallel, multi-block, moving grid method for aeroelastic applications on full aircraft», dans *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, American Institute of Aeronautics and Astronautics, p. 11, doi : [10.2514/6.1998-4782](https://doi.org/10.2514/6.1998-4782). URL <https://arc.aiaa.org/doi/abs/10.2514/6.1998-4782>, \_eprint: <https://arc.aiaa.org/doi/pdf/10.2514/6.1998-4782>. 53
- CAI, X.-C. et M. SARKIS. 1999, «A Restricted Additive Schwarz Preconditioner for General Sparse Linear Systems», *SIAM Journal on Scientific Computing*, vol. 21, n° 2, doi : [10.1137/S106482759732678X](https://doi.org/10.1137/S106482759732678X), p. 792–797, ISSN 1064-8275. URL <https://epubs.siam.org/doi/abs/10.1137/S106482759732678X>, publisher: Society for Industrial and Applied Mathematics. 12
- CASTAÑOS, J. et J. SAVAGE. 1997, «The Dynamic Adaptation of Parallel Mesh-Based Computation.», dans *Proceedings of the Eighth {SIAM} Conference on Parallel Processing for Scientific Computing*, SIAM, Hyatt Regency Minneapolis on Nicollet Mall Hotel, Minneapolis, Minnesota, USA,, p. 79. 41
- CAVALLO, P. A., N. SINHA et G. M. FELDMAN. 2005, «Parallel Unstructured Mesh Adaptation Method for Moving Body Applications», *AIAA Journal*, vol. 43, n° 9, doi : [10.2514/1.7818](https://doi.org/10.2514/1.7818), p. 1937–1945, ISSN 0001-1452. URL <https://doi.org/10.2514/1.7818>, publisher: American Institute of Aeronautics and Astronautics \_eprint: <https://doi.org/10.2514/1.7818>. 43
- CHAN, T. F. et T. P. MATHEW. 1994, «Domain decomposition algorithms», *Acta Numerica*, vol. 3, doi : [10.1017/S0962492900002427](https://doi.org/10.1017/S0962492900002427), p. 61–143, ISSN 0962-4929, 1474-0508. URL [https://www.cambridge.org/core/product/identifier/S0962492900002427/type/journal\\_article](https://www.cambridge.org/core/product/identifier/S0962492900002427/type/journal_article). 12
- CHEN, J., Z. XIAO, Y. ZHENG, J. ZOU, D. ZHAO et Y. YAO. 2018, «Scalable generation of large-scale unstructured meshes by a novel domain decomposition approach», *Advances in Engineering Software*, vol. 121, doi : [10.1016/j.advengsoft.2018.04.005](https://doi.org/10.1016/j.advengsoft.2018.04.005), p. 131–146, ISSN 0965-9978. URL <http://www.sciencedirect.com/science/article/pii/S0965997816306378>. 45, 63
- CHEVALIER, C. 2007, *Conception et mise en oeuvre d'outils efficaces pour le partitionnement et la distribution parallèles de problèmes numériques de très grande taille*, thesis, Bordeaux 1. URL <http://www.theses.fr/2007B0R13434>, publication Title: <http://www.theses.fr>. 35, 37, 38
- CHOCKALINGAM, T. et S. ARUNKUMAR. 1995, «Genetic algorithm based heuristics for the mapping problem», *Computers & Operations Research*, vol. 22, n° 1, doi : [10.1016/0305-0548\(94\)P2435-7](https://doi.org/10.1016/0305-0548(94)P2435-7), p. 55–64, ISSN 0305-0548. URL <http://www.sciencedirect.com/science/article/pii/0305054894P24357>. 36
- CHRISOCHOIDES, N. 2006, «Parallel Mesh Generation», dans *Numerical Solution of Partial Differential Equations on Parallel Computers*, édité par A. M. Bruaset et A. Tveito, Lecture Notes in Computational Science and Engineering, Springer, Berlin, Heidelberg, ISBN 978-3-540-31619-0, p. 237–264, doi : [10.1007/3-540-31619-1\\_7](https://doi.org/10.1007/3-540-31619-1_7). 43
- CHRISOCHOIDES, N. et D. NAVE. 2003, «Parallel Delaunay mesh generation kernel», *International Journal for Numerical Methods in Engineering*, vol. 58, n° 2, doi : [10.1002/nme.765](https://doi.org/10.1002/nme.765), p. 161–176, ISSN 1097-0207. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.765>, \_eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.765>. 40, 41

- COUPEZ, T., H. DIGONNET et R. DUCLOUX. 2000, «Parallel meshing and remeshing», *Applied Mathematical Modelling*, vol. 25, n° 2, doi : [10.1016/S0307-904X\(00\)00045-7](https://doi.org/10.1016/S0307-904X(00)00045-7), p. 153–175, ISSN 0307-904X. URL <http://www.sciencedirect.com/science/article/pii/S0307904X00000457>. 43
- CRAIG, R. R. et M. C. C. BAMPTON. 1968, «Coupling of substructures for dynamic analyses.», *AIAA Journal*, vol. 6, n° 7, doi : [10.2514/3.4741](https://doi.org/10.2514/3.4741), p. 1313–1319, ISSN 0001-1452, 1533-385X. URL <https://arc.aiaa.org/doi/10.2514/3.4741>. 17
- DE BOER, A., M. S. VAN DER SCHOOT et H. BIJL. 2007, «Mesh deformation based on radial basis function interpolation», *Computers & Structures*, vol. 85, n° 11-14, doi : [10.1016/j.compstruc.2007.01.013](https://doi.org/10.1016/j.compstruc.2007.01.013), p. 784–795, ISSN 00457949. URL <http://linkinghub.elsevier.com/retrieve/pii/S0045794907000223>. 54, 56, 57
- DE COUGNY, H. L. D. et M. S. SHEPHARD. 1999, «Parallel refinement and coarsening of tetrahedral meshes», *International Journal for Numerical Methods in Engineering*, vol. 46, n° 7, doi : [10.1002/\(SICI\)1097-0207\(19991110\)46:7<1101::AID-NME741>3.0.CO;2-E](https://doi.org/10.1002/(SICI)1097-0207(19991110)46:7<1101::AID-NME741>3.0.CO;2-E), p. 1101–1125, ISSN 1097-0207. 41
- DELAUNAY, B. 1924, «Sur la sphère vide», dans *Proceedings of the International Mathematical Congress held in Toronto*, vol. 1, Toronto, p. 695–700. 39
- DELAUNAY, B. N. 1934, «Sur la sphère vide», *Bulletin de l'Académie des Sciences de l'URSS. VII. Série*, vol. 1934, n° 6, p. 793–800. Publisher: Academy of Sciences of the Union of Soviet Socialist Republics - USSR (Akademiya Nauk SSSR). 39
- DIGONNET, H., M. BERNACKI, L. SILVA et T. COUPEZ. 2007, «Adaptation de maillage en parallèle, application à la simulation de la mise en forme des matériaux», *Congrès Français de Mécanique*. 43
- DIGONNET, H., T. COUPEZ, P. LAURE et L. SILVA. 2017, «Massively parallel anisotropic mesh adaptation», *International Journal of High Performance Computing Applications*, doi : [10.1177/1094342017693906](https://doi.org/10.1177/1094342017693906). URL <https://hal.archives-ouvertes.fr/hal-01487424>, publisher: SAGE Publications. 43
- DOHRMANN, C. R. 2003, «A Preconditioner for Substructuring Based on Constrained Energy Minimization», *SIAM Journal on Scientific Computing*, vol. 25, n° 1, doi : [10.1137/S1064827502412887](https://doi.org/10.1137/S1064827502412887), p. 246–258, ISSN 1064-8275. URL <https://epubs.siam.org/doi/abs/10.1137/S1064827502412887>, publisher: Society for Industrial and Applied Mathematics. 24
- DOHRMANN, C. R., A. KLAWONN et O. B. WIDLUND. 2008, «Domain Decomposition for Less Regular Subdomains: Overlapping Schwarz in Two Dimensions», *SIAM Journal on Numerical Analysis*, vol. 46, n° 4, doi : [10.1137/070685841](https://doi.org/10.1137/070685841), p. 2153–2168, ISSN 0036-1429. URL <https://epubs.siam.org/doi/10.1137/070685841>, publisher: Society for Industrial and Applied Mathematics. 31
- DOLEAN, V., P. JOLIVET et F. NATAF. 2015, *An Introduction to Domain Decomposition Methods*, Other Titles in Applied Mathematics, Society for Industrial and Applied Mathematics, ISBN 978-1-61197-405-8, doi : [10.1137/1.9781611974065](https://doi.org/10.1137/1.9781611974065). URL <https://epubs.siam.org/doi/book/10.1137/1.9781611974065>. 12
- DOSTÁL, Z., D. HORÁK et R. KUČERA. 2006, «Total FETI—an easier implementable variant of the FETI method for numerical solution of elliptic PDE», *Communications in Numerical Methods in Engineering*, vol. 22, n° 12, doi : [10.1002/cnm.881](https://doi.org/10.1002/cnm.881), p. 1155–1162, ISSN 1099-0887. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/cnm.881>, eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/cnm.881>. 18

- DRYJA, M. et O. B. WIDLUND. 1989, *Towards a unified theory of domain decomposition algorithms for elliptic problems*, New York University, Courant Institute of Mathematical Sciences, Division of Computer Science. 12
- DUCHON, J. 1976, «Interpolation des fonctions de deux variables suivant le principe de la flexion des plaques minces», *Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique*, vol. 10, n° R3, doi : [10.1051/m2an/197610R300051](https://doi.org/10.1051/m2an/197610R300051), p. 5–12, ISSN 0397-9342. URL <https://www.esaim-m2an.org/articles/m2an/abs/1976/03/m2an197610R300051/m2an197610R300051.html>, number: R3 Publisher: EDP Sciences. 55
- DUCHON, J. 1977, «Splines minimizing rotation-invariant semi-norms in Sobolev spaces», dans *Constructive Theory of Functions of Several Variables*, édité par W. Schempp et K. Zeller, Lecture Notes in Mathematics, Springer, Berlin, Heidelberg, ISBN 978-3-540-37496-1, p. 85–100, doi : [10.1007/BFb0086566](https://doi.org/10.1007/BFb0086566). 55
- EIKELAND, E., L. MARCINKOWSKI et T. RAHMAN. 2017, «A Study of the Effects of Irregular Subdomain Boundaries on Some Domain Decomposition Algorithms», dans *Domain Decomposition Methods in Science and Engineering XXIII*, édité par C.-O. Lee, X.-C. Cai, D. E. Keyes, H. H. Kim, A. Klawonn, E.-J. Park et O. B. Widlund, Lecture Notes in Computational Science and Engineering, Springer International Publishing, Cham, ISBN 978-3-319-52389-7, p. 295–302, doi : [10.1007/978-3-319-52389-7\\_30](https://doi.org/10.1007/978-3-319-52389-7_30). 31
- FARHAT, C., C. DEGAND, B. KOOBUS et M. LESOINNE. 1998, «Torsional springs for two-dimensional dynamic unstructured fluid meshes», *Computer Methods in Applied Mechanics and Engineering*, vol. 163, n° 1-4, doi : [10.1016/S0045-7825\(98\)00016-4](https://doi.org/10.1016/S0045-7825(98)00016-4), p. 231–245, ISSN 00457825. URL <http://linkinghub.elsevier.com/retrieve/pii/S0045782598000164>. 52
- FARHAT, C., M. LESOINNE, P. LETALLEC, K. PIERSON et D. RIXEN. 2001, «FETI-DP: a dual-primal unified FETI method—part I: A faster alternative to the two-level FETI method», *International Journal for Numerical Methods in Engineering*, vol. 50, n° 7, doi : [10.1002/nme.76](https://doi.org/10.1002/nme.76), p. 1523–1544, ISSN 1097-0207. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.76>, \_eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.76>. 18, 24
- FARHAT, C., M. LESOINNE et K. PIERSON. 2000, «A scalable dual-primal domain decomposition method», *Numerical Linear Algebra with Applications*, vol. 7, n° 7-8, doi : [10.1002/1099-1506\(200010/12\)7:7/8<687::AID-NLA219>3.0.CO;2-S](https://doi.org/10.1002/1099-1506(200010/12)7:7/8<687::AID-NLA219>3.0.CO;2-S), p. 687–714, ISSN 1070-5325, 1099-1506. 24
- FARHAT, C., J. MANDEL et F. X. ROUX. 1994, «Optimal convergence properties of the FETI domain decomposition method», *Computer Methods in Applied Mechanics and Engineering*, vol. 115, n° 3-4, doi : [10.1016/0045-7825\(94\)90068-X](https://doi.org/10.1016/0045-7825(94)90068-X), p. 365–385, ISSN 00457825. URL <http://linkinghub.elsevier.com/retrieve/pii/004578259490068X>. 18, 28
- FARHAT, C. et F.-X. ROUX. 1991, «A method of finite element tearing and interconnecting and its parallel solution algorithm», *International Journal for Numerical Methods in Engineering*, vol. 32, n° 6, doi : [10.1002/nme.1620320604](https://doi.org/10.1002/nme.1620320604), p. 1205–1227, ISSN 0029-5981, 1097-0207. URL <http://doi.wiley.com/10.1002/nme.1620320604>. 18
- FIDUCCIA, C. M. et R. M. MATTHEYSES. 1982, «A linear-time heuristic for improving network partitions», dans *Proceedings of the 19th Design Automation Conference, DAC '82*, IEEE Press, ISBN 978-0-89791-020-0, p. 175–181. 36, 38
- FIEDLER, M. 1975, «A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory», *Czechoslovak Mathematical Journal*, vol. 25, n° 4, p. 619–633, ISSN 0011-4642 (print). URL <https://dml.cz/handle/10338.dmlcz/101357>, publisher: Institute of Mathematics, Academy of Sciences of the Czech Republic. 36

- FILL, J. A. et D. E. FISHKIND. 2000, «The Moore–Penrose Generalized Inverse for Sums of Matrices», *SIAM Journal on Matrix Analysis and Applications*, vol. 21, n° 2, doi : [10.1137/S0895479897329692](https://doi.org/10.1137/S0895479897329692), p. 629–635, ISSN 0895-4798, 1095-7162. URL <http://epubs.siam.org/doi/10.1137/S0895479897329692>. 19
- FREITAG, L. A. et P. M. KNUPP. 2002, «Tetrahedral mesh improvement via optimization of the element condition number», *International Journal for Numerical Methods in Engineering*, vol. 53, n° 6, doi : [10.1002/nme.341](https://doi.org/10.1002/nme.341), p. 1377–1391, ISSN 0029-5981, 1097-0207. URL <http://doi.wiley.com/10.1002/nme.341>. 52
- FU, L., L. HAN, X. Y. HU et N. A. ADAMS. 2019, «An isotropic unstructured mesh generation method based on a fluid relaxation analogy», *Computer Methods in Applied Mechanics and Engineering*, vol. 350, doi : [10.1016/j.cma.2018.10.052](https://doi.org/10.1016/j.cma.2018.10.052), p. 396–431, ISSN 0045-7825. URL <http://www.sciencedirect.com/science/article/pii/S0045782519301227>. 40
- GALTIER, J. et P. L. GEORGE. 1997, «Prepartitioning as a way to mesh subdomains in parallel», dans *5th International Meshing Roundtable*, p. 107–122. 44
- GAREY, M. R., D. S. JOHNSON et L. STOCKMEYER. 1976, «Some simplified NP-complete graph problems», *Theoretical Computer Science*, vol. 1, n° 3, doi : [10.1016/0304-3975\(76\)90059-1](https://doi.org/10.1016/0304-3975(76)90059-1), p. 237–267, ISSN 0304-3975. URL <http://www.sciencedirect.com/science/article/pii/0304397576900591>. 35
- GEUZAINÉ, C. et J.-F. REMACLE. 2009, «Gmsh: A 3-D Finite Element Mesh Generator with Built-in Pre- and Post-Processing Facilities», *International Journal for Numerical Methods in Engineering*, vol. 79, doi : [10.1002/nme.2579](https://doi.org/10.1002/nme.2579), p. 1309–1331. 57
- GOSSELET, P. 2003, *Méthodes de décomposition de domaine et méthodes d'accélération pour les problèmes multichamps en mécanique non-linéaire*, Thèse, Paris 6. URL <http://www.theses.fr/2003PA066379>, publication Title: <http://www.theses.fr>. 23
- GOSSELET, P. 2017, *Extension du domaine de la décomposition*, Habilitation à diriger des recherches, ENS Cachan. URL <https://hal.archives-ouvertes.fr/tel-01472920>. 30
- GOSSELET, P., V. CHIARUTTINI, C. REY et F. FEYEL. 2004, «A monolithic strategy based on an hybrid domain decomposition method for multiphysic problems», *Revue Européenne des Éléments Finis*, vol. 13, n° 5-7, doi : [10.3166/reef.13.523-534](https://doi.org/10.3166/reef.13.523-534), p. 523–534, ISSN 1250-6559. URL <https://doi.org/10.3166/reef.13.523-534>, publisher: Taylor & Francis \_eprint: <https://doi.org/10.3166/reef.13.523-534>. 23
- GOSSELET, P. et C. REY. 2006, «Non-overlapping domain decomposition methods in structural mechanics», *Archives of Computational Methods in Engineering*, vol. 13, n° 4, doi : [10.1007/BF02905857](https://doi.org/10.1007/BF02905857), p. 515–572, ISSN 1134-3060, 1886-1784. URL <http://link.springer.com/10.1007/BF02905857>. 12, 13, 20, 22, 27, 29
- GOSSELET, P., C. REY, F. LÉNÉ et P. DASSET. 2002, «A domain decomposition method for quasi-incompressible formulations with discontinuous pressure field», *Revue Européenne des Éléments Finis*, vol. 11, n° 2-4, doi : [10.3166/reef.11.363-377](https://doi.org/10.3166/reef.11.363-377), p. 363–377, ISSN 1250-6559. URL <https://doi.org/10.3166/reef.11.363-377>, publisher: Taylor & Francis \_eprint: <https://doi.org/10.3166/reef.11.363-377>. 25
- GOSSELET, P., D. J. RIXEN, F.-X. ROUX et N. SPILLANE. 2015, «Simultaneous FETI and block FETI: Robust domain decomposition with multiple search directions», *International Journal for Numerical Methods in Engineering*, vol. 104, n° 10, doi : [10.1002/nme.4946](https://doi.org/10.1002/nme.4946), p. 905–927. 30, 31, 32

- HENDRICKSON, B. et R. LELAND. 1995, «A multilevel algorithm for partitioning graphs», dans *Proceedings of the 1995 ACM/IEEE conference on Supercomputing (CDROM) - Supercomputing '95*, ACM Press, San Diego, California, United States, ISBN 978-0-89791-816-9, p. 28–es, doi : [10.1145/224170.224228](https://doi.org/10.1145/224170.224228). URL <http://portal.acm.org/citation.cfm?doid=224170.224228>. 37
- HSU, S.-Y., C.-L. CHANG et J. SAMAREH. 2004, «A Simplified Mesh Deformation Method Using Commercial Structural Analysis Software», dans *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, American Institute of Aeronautics and Astronautics, Albany, New York, ISBN 978-1-62410-019-2, p. 33, doi : [10.2514/6.2004-4409](https://doi.org/10.2514/6.2004-4409). URL <http://arc.aiaa.org/doi/10.2514/6.2004-4409>. 52
- ITO, Y. et K. NAKAHASHI. 2002, «Direct Surface Triangulation Using Stereolithography Data», *AIAA Journal*, vol. 40, n° 3, doi : [10.2514/2.1672](https://doi.org/10.2514/2.1672), p. 490–496, ISSN 0001-1452. URL <https://arc.aiaa.org/doi/10.2514/2.1672>, publisher: American Institute of Aeronautics and Astronautics. 45
- ITO, Y., A. M. SHIH, A. K. ERUKALA, B. K. SONI, A. CHERNIKOV, N. P. CHRISOCHOIDES et K. NAKAHASHI. 2007, «Parallel unstructured mesh generation by an advancing front method», *Mathematics and Computers in Simulation*, vol. 75, n° 5, doi : [10.1016/j.matcom.2006.12.008](https://doi.org/10.1016/j.matcom.2006.12.008), p. 200–209, ISSN 0378-4754. URL <http://www.sciencedirect.com/science/article/pii/S0378475406003351>. 45, 46
- JI, Z., L. FU, X. HU et N. ADAMS. 2020, «A consistent parallel isotropic unstructured mesh generation method based on multi-phase SPH», *Computer Methods in Applied Mechanics and Engineering*, vol. 363, doi : [10.1016/j.cma.2020.112881](https://doi.org/10.1016/j.cma.2020.112881). 42
- KANAI, T., H. SUZUKI et F. KIMURA. 2000, «Metamorphosis of arbitrary triangular meshes», *IEEE Computer Graphics and Applications*, vol. 20, n° 2, doi : [10.1109/38.824544](https://doi.org/10.1109/38.824544), p. 62–75, ISSN 1558-1756. Conference Name: IEEE Computer Graphics and Applications. 48
- KARMAN, S. L. 2007, «Unstructured Viscous Layer Insertion Using Linear-Elastic Smoothing», *AIAA Journal*, vol. 45, n° 1, doi : [10.2514/1.23283](https://doi.org/10.2514/1.23283), p. 168–180, ISSN 0001-1452, 1533-385X. URL <http://arc.aiaa.org/doi/10.2514/1.23283>. 52
- KARYPIS, G. et V. KUMAR. 1998, «A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs», *SIAM Journal on Scientific Computing*, vol. 20, n° 1, doi : [10.1137/S1064827595287997](https://doi.org/10.1137/S1064827595287997), p. 359–392, ISSN 1064-8275, 1095-7197. URL <http://epubs.siam.org/doi/10.1137/S1064827595287997>. 37, 38
- KEDWARD, L., C. B. ALLEN et T. C. S. RENDALL. 2017, «Efficient and exact mesh deformation using multiscale RBF interpolation», *Journal of Computational Physics*, vol. 345, doi : [10.1016/j.jcp.2017.05.042](https://doi.org/10.1016/j.jcp.2017.05.042), p. 732–751, ISSN 0021-9991. URL <http://www.sciencedirect.com/science/article/pii/S0021999117304254>. 54
- KENT, J. R., W. E. CARLSON et R. E. PARENT. 1992, «Shape transformation for polyhedral objects», URL <https://doi.org/10.1145/142920.134007>. 48
- KERNIGHAN, B. W. et S. LIN. 1970, «An Efficient Heuristic Procedure for Partitioning Graphs», *Bell System Technical Journal*, vol. 49, n° 2, doi : [10.1002/j.1538-7305.1970.tb01770.x](https://doi.org/10.1002/j.1538-7305.1970.tb01770.x), p. 291–307, ISSN 0005-8580. URL <https://collaborate.princeton.edu/en/publications/an-efficient-heuristic-procedure-for-partitioning-graphs>, publisher: John Wiley and Sons Inc. 36, 38
- KLAWONN, A. et O. RHEINBACH. 2007, «Robust FETI-DP methods for heterogeneous three dimensional elasticity problems», *Computer Methods in Applied Mechanics and Engineering*, vol.

- 196, n° 8, doi : [10.1016/j.cma.2006.03.023](https://doi.org/10.1016/j.cma.2006.03.023), p. 1400–1414, ISSN 00457825. URL <http://linkinghub.elsevier.com/retrieve/pii/S004578250600260X>. 30
- KLAWONN, A., O. RHEINBACH et O. B. WIDLUND. 2008, «An Analysis of a FETI–DP Algorithm on Irregular Subdomains in the Plane», *SIAM Journal on Numerical Analysis*, vol. 46, n° 5, doi : [10.1137/070688675](https://doi.org/10.1137/070688675), p. 2484–2504, ISSN 0036-1429, 1095-7170. URL <http://epubs.siam.org/doi/10.1137/070688675>. 30, 31
- KLAWONN, A. et O. WIDLUND. 2001, «FETI and Neumann-Neumann iterative substructuring methods: Connections and new results», *Communications on Pure and Applied Mathematics*, vol. 54, n° 1, doi : [10.1002/1097-0312\(200101\)54:1<57::AID-CPA3>3.0.CO;2-D](https://doi.org/10.1002/1097-0312(200101)54:1<57::AID-CPA3>3.0.CO;2-D), p. 57–90, ISSN 0010-3640, 1097-0312. 20, 25, 26, 28, 30
- KNUPP, P. M. 2003, «Algebraic mesh quality metrics for unstructured initial meshes», *Finite Elements in Analysis and Design*, vol. 39, n° 3, doi : [10.1016/S0168-874X\(02\)00070-7](https://doi.org/10.1016/S0168-874X(02)00070-7), p. 217–241, ISSN 0168874X. URL <http://linkinghub.elsevier.com/retrieve/pii/S0168874X02000707>. 57, 58
- KOROŠEC, P., J. ŠILC et B. ROBIČ. 2004, «Solving the mesh-partitioning problem with an ant-colony algorithm», *Parallel Computing*, vol. 30, n° 5, doi : [10.1016/j.parco.2003.12.016](https://doi.org/10.1016/j.parco.2003.12.016), p. 785–801, ISSN 0167-8191. URL <http://www.sciencedirect.com/science/article/pii/S0167819104000432>. 37
- LADEVÈZE, P. 1999, *Nonlinear Computational Structural Mechanics: New Approaches and Non-Incremental Methods of Calculation*, Mechanical Engineering Series, Springer New York, New York, NY, ISBN 978-1-4612-7141-3 978-1-4612-1432-8, doi : [10.1007/978-1-4612-1432-8](https://doi.org/10.1007/978-1-4612-1432-8). URL <http://link.springer.com/10.1007/978-1-4612-1432-8>. 20
- LADEVÈZE, P., D. NÉRON et P. GOSSELET. 2007, «On a mixed and multiscale domain decomposition method», *Computer Methods in Applied Mechanics and Engineering*, vol. 196, n° 8, doi : [10.1016/j.cma.2006.05.014](https://doi.org/10.1016/j.cma.2006.05.014), p. 1526–1540, ISSN 0045-7825. URL <http://www.sciencedirect.com/science/article/pii/S0045782506002829>. 20
- LAEMMER, L. 1997, «Parallel mesh generation», dans *Solving Irregularly Structured Problems in Parallel*, édité par G. Bilardi, A. Ferreira, R. Lüling et J. Rolim, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, ISBN 978-3-540-69157-0, p. 1–12, doi : [10.1007/3-540-63138-0\\_1](https://doi.org/10.1007/3-540-63138-0_1). 44
- LE TALLEC, P. 1994, «Domain-decomposition methods in computational mechanics», *Computational Mechanics Advances*. 12, 18, 27
- LE TALLEC, P., Y. H. DE ROECK et M. VIDRASCU. 1991, «Domain decomposition methods for large linearly elliptic three-dimensional problems», *Journal of Computational and Applied Mathematics*, vol. 34, n° 1, doi : [10.1016/0377-0427\(91\)90150-I](https://doi.org/10.1016/0377-0427(91)90150-I), p. 93–117, ISSN 0377-0427. URL <http://www.sciencedirect.com/science/article/pii/037704279190150I>. 18
- LESOINNE, M. et K. PIERSON. 1999, «FETI-DP: An Efficient, Scalable and Unified Dual-Primal FETI Method», dans *12th International Conference on Domain Decomposition Methods*, édité par T. F. Chan, T. Kako, H. Kawarada et O. Pironneau, p. 8. 24
- LINARDAKIS, L. et N. CHRISOCHOIDES. 2006, «Delaunay Decoupling Method for Parallel Guaranteed Quality Planar Mesh Refinement», *SIAM J. Scientific Computing*, vol. 27, doi : [10.1137/030602812](https://doi.org/10.1137/030602812), p. 1394–1423. 44
- LINARDAKIS, L. et N. CHRISOCHOIDES. 2008, «Algorithm 870: A static geometric Medial Axis domain decomposition in 2D Euclidean space», *ACM Transactions on Mathematical Software*, vol. 34, n° 1, doi : [10.1145/1322436.1322440](https://doi.org/10.1145/1322436.1322440), p. 4:1–4:28, ISSN 0098-3500. URL <https://doi.org/10.1145/1322436.1322440>. 44

- LIONS, P.-L. 1990, «On the Schwarz alternating method III: a variant for nonoverlapping subdomains», dans *Third international symposium on domain decomposition methods for partial differential equations*, vol. 6, p. 202–223. 12, 20
- LO, S. H. 1991, «Volume discretization into tetrahedra—II. 3D triangulation by advancing front approach», *Computers & Structures*, vol. 39, n° 5, doi : [10.1016/0045-7949\(91\)90059-U](https://doi.org/10.1016/0045-7949(91)90059-U), p. 501–511, ISSN 0045-7949. URL <http://www.sciencedirect.com/science/article/pii/004579499190059U>. 40
- LOISEL, S. 2013, «Condition Number Estimates for the Nonoverlapping Optimized Schwarz Method and the 2-Lagrange Multiplier Method for General Domains and Cross Points», *SIAM Journal on Numerical Analysis*, vol. 51, n° 6, doi : [10.1137/100803316](https://doi.org/10.1137/100803316), p. 3062–3083, ISSN 0036-1429. URL <https://epubs.siam.org/doi/abs/10.1137/100803316>, publisher: Society for Industrial and Applied Mathematics. 22
- LOSEILLE, A., V. MENIER et F. ALAUZET. 2015, «Parallel Generation of Large-size Adapted Meshes», *Procedia Engineering*, vol. 124, doi : [10.1016/j.proeng.2015.10.122](https://doi.org/10.1016/j.proeng.2015.10.122), p. 57–69, ISSN 18777058. URL <https://linkinghub.elsevier.com/retrieve/pii/S1877705815032233>. 46, 47, 63
- LÖHNER, R. 1988, «Interactive generation of unstructured grids for three dimensional problems», dans *Proceedings of the 2nd Intl. Conference on Numerical Grid Generation in Computational Fluid Mechanics*, Miami Beach, FL, United States, p. 685–698. 40
- LÖHNER, R., J. CAMBEROS et M. MERRIAM. 1992, «Parallel unstructured grid generation», *Computer Methods in Applied Mechanics and Engineering*, vol. 95, n° 3, doi : [10.1016/0045-7825\(92\)90192-M](https://doi.org/10.1016/0045-7825(92)90192-M), p. 343–357, ISSN 0045-7825. URL <http://www.sciencedirect.com/science/article/pii/004578259290192M>. 41
- MANDEL, J. 1993, «Balancing domain decomposition», *Communications in Numerical Methods in Engineering*, vol. 9, n° 3, doi : [10.1002/cnm.1640090307](https://doi.org/10.1002/cnm.1640090307), p. 233–241, ISSN 10698299. URL <http://doi.wiley.com/10.1002/cnm.1640090307>. 18, 27
- MANDEL, J. et B. SOUSEDÍK. 2007, «Adaptive selection of face coarse degrees of freedom in the BDDC and the FETI-DP iterative substructuring methods», *Computer Methods in Applied Mechanics and Engineering*, vol. 196, n° 8, doi : [10.1016/j.cma.2006.03.010](https://doi.org/10.1016/j.cma.2006.03.010), p. 1389–1399, ISSN 00457825. URL <http://linkinghub.elsevier.com/retrieve/pii/S0045782506002593>. 30
- MANDEL, J. et R. TEZAUR. 1996, «Convergence of a substructuring method with Lagrange multipliers», *Numerische Mathematik*, vol. 73, n° 4, doi : [10.1007/s002110050201](https://doi.org/10.1007/s002110050201), p. 473–487, ISSN 0945-3245. URL <https://doi.org/10.1007/s002110050201>. 20
- MARKOU, G. A., Z. S. MOUROUTIS, D. C. CHARMPIS et M. PAPADRAKAKIS. 2007, «The ortho-semi-torsional (OST) spring analogy method for 3D mesh moving boundary problems», *Computer Methods in Applied Mechanics and Engineering*, vol. 196, n° 4-6, doi : [10.1016/j.cma.2006.04.009](https://doi.org/10.1016/j.cma.2006.04.009), p. 747–765, ISSN 00457825. URL <http://linkinghub.elsevier.com/retrieve/pii/S0045782506002295>. 52
- MERTA, M., L. RIHA, O. MECA, A. MARKOPOULOS, T. BRZOBOHATY, T. KOZUBEK et V. VONDRAK. 2017, «Intel Xeon Phi acceleration of Hybrid Total FETI solver», *Advances in Engineering Software*, vol. 112, doi : [10.1016/j.advengsoft.2017.05.001](https://doi.org/10.1016/j.advengsoft.2017.05.001), p. 124–135, ISSN 0965-9978. URL <http://www.sciencedirect.com/science/article/pii/S0965997816302745>. 18
- MEYERHENKE, H., B. MONIEN et S. SCHAMBERGER. 2006, «Accelerating shape optimizing load balancing for parallel FEM simulations by algebraic multigrid», dans *Proceedings 20th IEEE International Parallel Distributed Processing Symposium*, p. 10 pp.–, doi : [10.1109/IPDPS.2006.1639295](https://doi.org/10.1109/IPDPS.2006.1639295). ISSN: 1530-2075. 36, 37

- MEYERHENKE, H. et S. SCHAMBERGER. 2006, «A Parallel Shape Optimizing Load Balancer», dans *Euro-Par 2006 Parallel Processing*, édité par W. E. Nagel, W. V. Walter et W. Lehner, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, ISBN 978-3-540-37784-9, p. 232–242, doi : [10.1007/11823285\\_24](https://doi.org/10.1007/11823285_24). 36
- MOCANU, B. et T. ZAHARIA. 2011, «Direct Spherical Parameterization Based on Surface Curvature», dans *2011 Workshop on Digital Media and Digital Content Management*, p. 266–269, doi : [10.1109/DMDCM.2011.61](https://doi.org/10.1109/DMDCM.2011.61). 49
- MOCANU, B. C. 2012, *3D mesh morphing. (Métamorphose de maillage 3D)*, Thèse, Telecom & Management SudParis, Évry, Essonne, France. URL <https://tel.archives-ouvertes.fr/tel-00836048>. 48
- MONAGHAN, J. J. 1992, «Smoothed Particle Hydrodynamics», *Annual Review of Astronomy and Astrophysics*, vol. 30, n° 1, doi : [10.1146/annurev.aa.30.090192.002551](https://doi.org/10.1146/annurev.aa.30.090192.002551), p. 543–574. URL <https://doi.org/10.1146/annurev.aa.30.090192.002551>, \_eprint: <https://doi.org/10.1146/annurev.aa.30.090192.002551>. 41
- MUTHUKRISHNAN, S. N., P. S. SHIAKOLAS, R. V. NAMBIAR et K. L. LAWRENCE. 1995, «Simple algorithm for adaptive refinement of three-dimensional finite element tetrahedral meshes», *AIAA Journal*, vol. 33, n° 5, doi : [10.2514/3.12386](https://doi.org/10.2514/3.12386), p. 928–932, ISSN 0001-1452. URL <https://doi.org/10.2514/3.12386>, publisher: American Institute of Aeronautics and Astronautics \_eprint: <https://doi.org/10.2514/3.12386>. 40
- MÖLLER, T. et B. TRUMBORE. 1997, «Fast, Minimum Storage Ray-Triangle Intersection», *Journal of Graphics Tools*, vol. 2, n° 1, doi : [10.1080/10867651.1997.10487468](https://doi.org/10.1080/10867651.1997.10487468), p. 21–28, ISSN 1086-7651. URL <http://www.tandfonline.com/doi/abs/10.1080/10867651.1997.10487468>. 49, 50, 51, 63
- NAVE, D., N. CHRISOCHOIDES et L. P. CHEW. 2004, «Guaranteed-quality parallel Delaunay refinement for restricted polyhedral domains», *Computational Geometry*, vol. 28, n° 2, doi : [10.1016/j.comgeo.2004.03.009](https://doi.org/10.1016/j.comgeo.2004.03.009), p. 191–215, ISSN 0925-7721. URL <http://www.sciencedirect.com/science/article/pii/S0925772104000240>. 41
- NEGRELLO, C. 2017, *Méthodes fortement parallèles pour la simulation numérique en mécanique non linéaire des structures*, Thèse, Université Paris-Saclay (ComUE). URL <http://www.theses.fr/2017SACLN049>, publication Title: <http://www.theses.fr>. 13
- NEGRELLO, C., P. GOSSELET et C. REY. 2018, «A new impedance accounting for short and long range effects in mixed substructured formulations of nonlinear problems», *International Journal for Numerical Methods in Engineering*, vol. 114, n° 7, doi : [10.1002/nme.5758](https://doi.org/10.1002/nme.5758), p. 675–693. URL <https://hal.archives-ouvertes.fr/hal-01488615>. 22
- NEGRELLO, C., P. GOSSELET, C. REY et J. PEBREL. 2016, «Substructured formulations of nonlinear structure problems - influence of the interface condition», *International Journal for Numerical Methods in Engineering*, vol. 107, n° 13, doi : [10.1002/nme.5195](https://doi.org/10.1002/nme.5195), p. 1083–1105. 12, 22, 34
- OKUSANYA, T. 1997, «3D Parallel Unstructured Mesh Generation», dans *Proceedings of Fifth International Conference on Numerical Grid Generation in Computational Fluid Dynamics and Related Fields*, p. 8. 41
- OKUSANYA, T. et J. PERAIRE. 1996, «Parallel Unstructured Mesh Generation», dans *Proceedings of Numerical Grid Generation in Computational Field Simulations*, édité par B. K. Soni, Mississippi State University, p. 719–729. 41
- OLIKER, L., R. BISWAS et H. N. GABOW. 2000, «Parallel tetrahedral mesh adaptation with dynamic load balancing», *Parallel Computing*, vol. 26, n° 12, doi : [10.1016/S0167-8191\(00\)00047-8](https://doi.org/10.1016/S0167-8191(00)00047-8),

- p. 1583–1608, ISSN 0167-8191. URL <http://www.sciencedirect.com/science/article/pii/S0167819100000478>. 40, 41
- OUMAZIZ, P., P. GOSSELET, K. SAAVEDRA et N. TARDIEU. 2020, «Analysis, improvement and limits of the multiscale Latin method», URL <https://hal.archives-ouvertes.fr/hal-03090960>. 32
- PARK, K. C., M. R. JUSTINO et C. A. FELIPPA. 1997, «An algebraically partitioned FETI method for parallel structural analysis: algorithm description», *International Journal for Numerical Methods in Engineering*, vol. 40, n° 15, doi : [10.1002/\(SICI\)1097-0207\(19970815\)40:15<2717::AID-NME185>3.0.CO;2-B](https://doi.org/10.1002/(SICI)1097-0207(19970815)40:15<2717::AID-NME185>3.0.CO;2-B), p. 2717–2737, ISSN 1097-0207. 15
- PARRET-FRÉAUD, A. 2011, *Estimation d'erreur de discrétisation dans les calculs par décomposition de domaine*, thesis, Cachan, Ecole normale supérieure. URL <http://www.theses.fr/2011DENS0022>, publication Title: <http://www.theses.fr>. 13
- PECHSTEIN, C. et R. SCHEICHL. 2011, «Analysis of FETI methods for multiscale PDEs. Part II: interface variation», *Numerische Mathematik*, vol. 118, n° 3, doi : [10.1007/s00211-011-0359-2](https://doi.org/10.1007/s00211-011-0359-2), p. 485–529, ISSN 0029-599X, 0945-3245. URL <http://link.springer.com/10.1007/s00211-011-0359-2>. 30
- PRZEMIENIECKI, J. S. 1963, «Matrix structural analysis of substructures», *AIAA Journal*, vol. 1, n° 1, doi : [10.2514/3.1483](https://doi.org/10.2514/3.1483), p. 138–147, ISSN 0001-1452. URL <https://arc.aiaa.org/doi/10.2514/3.1483>, publisher: American Institute of Aeronautics and Astronautics. 12
- RENDALL, T. C. S. et C. B. ALLEN. 2009, «Efficient mesh motion using radial basis functions with data reduction algorithms», *Journal of Computational Physics*, vol. 228, n° 17, doi : [10.1016/j.jcp.2009.05.013](https://doi.org/10.1016/j.jcp.2009.05.013), p. 6231–6249, ISSN 0021-9991. URL <http://www.sciencedirect.com/science/article/pii/S0021999109002721>. 56
- RIXEN, D. 1997, *Substructuring and Dual Methods in Structural Analysis*, thèse de doctorat, Université de Liège, doi : [10.13140/RG.2.2.10951.91046](https://doi.org/10.13140/RG.2.2.10951.91046). 32
- RIXEN, D. J. et C. FARHAT. 1999, «A simple and efficient extension of a class of substructure based preconditioners to heterogeneous structural mechanics problems», *International Journal for Numerical Methods in Engineering*, vol. 44, n° 4, doi : [10.1002/\(SICI\)1097-0207\(19990210\)44:4<489::AID-NME514>3.0.CO;2-Z](https://doi.org/10.1002/(SICI)1097-0207(19990210)44:4<489::AID-NME514>3.0.CO;2-Z), p. 489–516. 25, 26, 30
- RON, D., S. WISHKO-STERN et A. BRANDT. 2005, «An algebraic multigrid based algorithm for bisectioning general graphs», cahier de recherche MCS05-01, Department of Computer Science and Applied Mathematics, the Weizmann Institute of Science. URL <http://www.wisdom.weizmann.ac.il/~dron/partition.pdf>. 38
- ROUX, F.-X. 2003, «Parallel solver based on domain decomposition», . 24
- ROUX, F.-X. 2009, «A FETI-2LM Method for Non-Matching Grids», dans *Domain Decomposition Methods in Science and Engineering XVIII*, vol. 70, édité par M. J. Gander, O. Widlund et M. Bercovier, Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN 978-3-642-02676-8 978-3-642-02677-5, p. 121–128, doi : [10.1007/978-3-642-02677-5\\_11](https://doi.org/10.1007/978-3-642-02677-5_11). URL [http://link.springer.com/10.1007/978-3-642-02677-5\\_11](http://link.springer.com/10.1007/978-3-642-02677-5_11), series Title: Lecture Notes in Computational Science and Engineering. 20
- ROUX, F.-X., F. MAGOULÈS, S. SALMON et L. SERIES. 2003, «Optimization of Interface Operator Based on Algebraic Approach», dans *Fourteenth International Conference on Domain Decomposition Methods*, Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates, p. 297–304. 22, 63

- SAAD, Y. 1997, «Analysis of Augmented Krylov Subspace Methods», *SIAM Journal on Matrix Analysis and Applications*, vol. 18, n° 2, doi : [10.1137/S0895479895294289](https://doi.org/10.1137/S0895479895294289), p. 435–449, ISSN 0895-4798, 1095-7162. URL <http://epubs.siam.org/doi/10.1137/S0895479895294289>. 20
- SAAD, Y. 2003, *Iterative Methods for Sparse Linear Systems*, Other Titles in Applied Mathematics, Society for Industrial and Applied Mathematics, ISBN 978-0-89871-534-7, doi : [10.1137/1.9780898718003](https://doi.org/10.1137/1.9780898718003). URL <https://epubs.siam.org/doi/book/10.1137/1.9780898718003>. 11
- SAAD, Y. et M. H. SCHULTZ. 1986, «GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems», *SIAM Journal on Scientific and Statistical Computing*, vol. 7, n° 3, doi : [10.1137/0907058](https://doi.org/10.1137/0907058), p. 856–869, ISSN 0196-5204. URL <https://epubs.siam.org/doi/abs/10.1137/0907058>, publisher: Society for Industrial and Applied Mathematics. 34
- SCHENK, O., K. GÄRTNER, W. FICHTNER et A. STRICKER. 2001, «PARDISO: a high-performance serial and parallel sparse linear solver in semiconductor device simulation», *Future Generation Computer Systems*, vol. 18, n° 1, doi : [10.1016/S0167-739X\(00\)00076-5](https://doi.org/10.1016/S0167-739X(00)00076-5), p. 69–78, ISSN 0167-739X. URL <http://www.sciencedirect.com/science/article/pii/S0167739X00000765>. 11
- SCHWARZ, H. A. 1870, «Über einen Grenzübergang durch alternierendes Verfahren», *Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich*, vol. 5, p. 272–286. 12
- SEGURA, R. et F. FEITO. 2001, «Algorithms to Test Ray-Triangle Intersection. Comparative Study.», dans *WSCG*, vol. 9, p. 76–81. 49
- SELIM, M. et K. KOMULLIL. 2016, «Mesh Deformation Approaches A Survey», *Journal of Physical Mathematics*, vol. 7, n° 2, doi : [10.4172/2090-0902.1000181](https://doi.org/10.4172/2090-0902.1000181), ISSN 20900902. URL <https://www.omicsonline.com/open-access/mesh-deformation-approaches\T1\textendasha-survey-2090-0902-1000181.php?aid=76056>. 51, 55, 56, 57
- SPILLANE, N. 2016, «An Adaptive MultiPreconditioned Conjugate Gradient Algorithm», *SIAM Journal on Scientific Computing*, vol. 38, n° 3, doi : [10.1137/15M1028534](https://doi.org/10.1137/15M1028534), p. A1896–A1918, ISSN 1064-8275. URL <https://epubs.siam.org/doi/10.1137/15M1028534>, publisher: Society for Industrial and Applied Mathematics. 33
- SPILLANE, N., V. DOLEAN, P. HAURET, F. NATAF, C. PECHSTEIN et R. SCHEICHL. 2011, «A robust two-level domain decomposition preconditioner for systems of PDEs», *Comptes Rendus Mathématique*, vol. 349, n° 23, doi : [10.1016/j.crma.2011.10.021](https://doi.org/10.1016/j.crma.2011.10.021), p. 1255–1259, ISSN 1631-073X. URL <http://www.sciencedirect.com/science/article/pii/S1631073X11003049>. 31
- SPILLANE, N., V. DOLEAN, P. HAURET, F. NATAF, C. PECHSTEIN et R. SCHEICHL. 2014, «Abstract robust coarse spaces for systems of PDEs via generalized eigenproblems in the overlaps», *Numerische Mathematik*, vol. 126, n° 4, doi : [10.1007/s00211-013-0576-y](https://doi.org/10.1007/s00211-013-0576-y), p. 741–770, ISSN 0029-599X. URL <https://researchportal.bath.ac.uk/en/publications/abstract-robust-coarse-spaces-for-systems-of-pdes-via-generalized>, publisher: Springer New York. 32
- SPILLANE, N. et D. J. RIXEN. 2013, «Automatic spectral coarse spaces for robust finite element tearing and interconnecting and balanced domain decomposition algorithms», *International Journal for Numerical Methods in Engineering*, vol. 95, n° 11, doi : [10.1002/nme.4534](https://doi.org/10.1002/nme.4534), p. 953–990, ISSN 1097-0207. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.4534>. 32
- TAO, L. et Y. ZHAO. 1993, «Multi-way graph partition by stochastic probe», *Computers & Operations Research*, vol. 20, n° 3, doi : [10.1016/0305-0548\(93\)90007-6](https://doi.org/10.1016/0305-0548(93)90007-6), p. 321–347, ISSN 0305-0548. URL <http://www.sciencedirect.com/science/article/pii/0305054893900076>. 37

- TOULOUGOUSSOU, A. B. et F.-X. ROUX. 2014, «Hybrid Dual-Primal FETI-Schur Complement Method for Stokes», dans *Domain Decomposition Methods in Science and Engineering XXI*, édité par J. Erhel, M. J. Gander, L. Halpern, G. Pichot, T. Sassi et O. Widlund, Lecture Notes in Computational Science and Engineering, Springer International Publishing, Cham, ISBN 978-3-319-05789-7, p. 899–907, doi : [10.1007/978-3-319-05789-7\\_87](https://doi.org/10.1007/978-3-319-05789-7_87). 23
- VINSOME, P. K. W. 1976, «Orthomin, an Iterative Method for Solving Sparse Sets of Simultaneous Linear Equations», dans *4th Symposium on Numerical Simulation of Reservoir Performance*, Society of Petroleum Engineers, ISBN 978-1-55563-751-4, p. 149–159, doi : [10.2118/5729-MS](https://doi.org/10.2118/5729-MS). URL <https://www.onepetro.org/conference-paper/SPE-5729-MS>. 34
- WANG, X.-Q., X.-L. JIN, D.-Z. KOU et J.-H. CHEN. 2017, «A Parallel Approach for the Generation of Unstructured Meshes with Billions of Elements on Distributed-Memory Supercomputers», *International Journal of Parallel Programming*, vol. 45, n° 3, doi : [10.1007/s10766-016-0452-3](https://doi.org/10.1007/s10766-016-0452-3), p. 680–710, ISSN 1573-7640. URL <https://doi.org/10.1007/s10766-016-0452-3>. 45
- WANGERMEZ, M., O. ALLIX, P.-A. GUIDAULT, O. CIOBANU et C. REY. 2020, «Interface coupling method for the global–local analysis of heterogeneous models: A second-order homogenization-based strategy», *Computer Methods in Applied Mechanics and Engineering*, vol. 365, doi : [10.1016/j.cma.2020.113032](https://doi.org/10.1016/j.cma.2020.113032), p. 113 032, ISSN 0045-7825. URL <http://www.sciencedirect.com/science/article/pii/S0045782520302164>. 12
- WATSON, D. F. 1981, «Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes», *The Computer Journal*, vol. 24, n° 2, doi : [10.1093/comjnl/24.2.167](https://doi.org/10.1093/comjnl/24.2.167), p. 167–172, ISSN 0010-4620. URL <https://academic.oup.com/comjnl/article/24/2/167/338200>, publisher: Oxford Academic. 39
- WENDLAND, H. 1995, «Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree», *Advances in Computational Mathematics*, vol. 4, n° 1, doi : [10.1007/BF02123482](https://doi.org/10.1007/BF02123482), p. 389–396, ISSN 1019-7168, 1572-9044. URL <https://link.springer.com/article/10.1007/BF02123482>. 54
- WENDLAND, H. 1996, *Konstruktion und Untersuchung radialer Basisfunktionen mit kompaktem Träger*, Thèse, [s.n.], S.I. OCLC: 68522203. 55
- WITTEVEEN, J. 2010, «Explicit and Robust Inverse Distance Weighting Mesh Deformation for CFD», dans *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, American Institute of Aeronautics and Astronautics, ISBN 978-1-60086-959-4, p. 1–10, doi : [10.2514/6.2010-165](https://doi.org/10.2514/6.2010-165). URL <http://arc.aiaa.org/doi/10.2514/6.2010-165>. 54
- XIE, L. et H. LIU. 2017, «Efficient mesh motion using radial basis functions with volume grid points reduction algorithm», *Journal of Computational Physics*, vol. 348, doi : [10.1016/j.jcp.2017.07.042](https://doi.org/10.1016/j.jcp.2017.07.042), p. 401–415, ISSN 0021-9991. URL <http://www.sciencedirect.com/science/article/pii/S0021999117305508>. 54
- YANG, Z. et D. J. MAVRIPLIS. 2007, «Mesh Deformation Strategy Optimized by the Adjoint Method on Unstructured Meshes», *AIAA Journal*, vol. 45, n° 12, doi : [10.2514/1.30592](https://doi.org/10.2514/1.30592), p. 2885–2896, ISSN 0001-1452. URL <https://doi.org/10.2514/1.30592>, publisher: American Institute of Aeronautics and Astronautics \_eprint: <https://doi.org/10.2514/1.30592>. 52
- ZHANG, F., éd.. 2005, *The Schur Complement and Its Applications*, Numerical Methods and Algorithms, Springer US, ISBN 978-0-387-24271-2, doi : [10.1007/b105056](https://doi.org/10.1007/b105056). URL <https://www.springer.com/gp/book/9780387242712>. 17

ZHAO, Y. et A. FORHAD. 2003, «A general method for simulation of fluid flows with moving and compliant boundaries on unstructured grids», *Computer Methods in Applied Mechanics and Engineering*, vol. 192, n° 39-40, doi : [10.1016/S0045-7825\(03\)00424-9](https://doi.org/10.1016/S0045-7825(03)00424-9), p. 4439–4466, ISSN 00457825. URL <http://linkinghub.elsevier.com/retrieve/pii/S0045782503004249>.  
53

ČERNÝ, V. 1985, «Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm», *Journal of Optimization Theory and Applications*, vol. 45, n° 1, doi : [10.1007/BF00940812](https://doi.org/10.1007/BF00940812), p. 41–51, ISSN 1573-2878. URL <https://doi.org/10.1007/BF00940812>.  
36

# CHAPITRE 2

## MÉTHODE DE SOUS-STRUCTURATION À DEUX NIVEAUX

Le chapitre précédent a introduit les méthodes de décomposition de domaine sans recouvrement les plus classiques. La section 1.1.3.3 a montré que la présence de certaines caractéristiques (hétérogénéités, irrégularité de l'interface et élancement des sous-domaines) avaient une influence néfaste sur la vitesse de convergence des solveurs itératifs utilisés pour résoudre le problème d'interface. Des solutions efficaces existent, décrites dans la section 1.1.3.4, elles induisent cependant un surcoût de calcul. Il semble donc judicieux d'anticiper et d'éviter l'apparition de ces caractéristiques néfastes.

L'objet de ce chapitre est de les éviter en se concentrant sur la phase de génération de maillage de la structure tout en permettant une parallélisation de cette phase en un minimum d'échanges entre les processus. La forme des sous-domaines de la sous-structuration est contrôlée afin qu'ils présentent une interface régulière et, dans le cas de problèmes multi-matériaux, de rendre les sous-domaines homogènes.

La méthode développée dans ce chapitre utilise comme point de départ un maillage grossier qui subira un raffinement, choisi ici comme hiérarchique. Hormis quelques cas particuliers, le maillage grossier présente un écart non négligeable à la CAO, écart conservé par le raffinement hiérarchique. Une phase de déformation de maillage est donc nécessaire pour garantir une bonne représentativité de la structure. Cette phase de déformation de maillage fait usage d'une interpolation par *RBF*. Une amélioration de l'efficacité de cette méthode utilisant la hiérarchie du maillage local pour adapter le rayon d'influence de la *RBF* est développée.

Le travail décrit ici a fait l'objet de communications internationales (EL GHARBI et coll. [2019a, 2020, 2021a]) et d'une communication nationale (EL GHARBI et coll. [2019b]). Ce travail a aussi donné lieu à la publication d'un article dans le journal international *Finite Elements in Analysis and Design (FEAD)* (EL GHARBI et coll. [2021b]).

« La simplicité est la sophistication suprême. »

Léonard De Vinci

<b>2.1 Philosophie générale de la méthode à deux niveaux</b> . . . . .	<b>79</b>
2.1.1 Maillage grossier . . . . .	80
2.1.2 Décomposition séquentielle . . . . .	81
2.1.3 Raffinement local . . . . .	83
<b>2.2 Projection des différentes entités constitutives</b> . . . . .	<b>85</b>
2.2.1 Projection des arêtes physiques $\mathcal{E}_{r,\phi}^{(s)}$ sur $\mathcal{E}_{\phi}^{(s)}$ . . . . .	85
2.2.2 Projection des arêtes de la frontière restantes . . . . .	86

2.2.3	Projection des surfaces . . . . .	86
<b>2.3</b>	<b>Méthode de déformation de maillage hiérarchique . . . . .</b>	<b>87</b>
2.3.1	Limitations de la méthode RBF . . . . .	87
2.3.2	Résolution par un solveur itératif . . . . .	90
2.3.2.1	Influence du critère d'arrêt . . . . .	90
2.3.2.2	Préconditionneur issu d'une factorisation incomplète . . . . .	91
2.3.2.3	Préconditionneur exploitant une troncature de $\Psi$ . . . . .	92
2.3.3	Interpolation RBF avec adaptation hiérarchique de rayon . . . . .	93
<b>2.4</b>	<b>Résultats numériques . . . . .</b>	<b>95</b>
2.4.1	Influence de la décomposition . . . . .	96
2.4.2	Exemple homogène : Aube de turbine simplifiée . . . . .	97
2.4.2.1	Description du problème . . . . .	100
2.4.2.2	Évaluation de la méthode de déformation de maillage hiérarchique . . . . .	100
2.4.2.3	Extensibilité forte . . . . .	102
2.4.3	Exemple hétérogène : volume simplifié de propergol solide . . . . .	105
2.4.3.1	Description du problème . . . . .	106
2.4.3.2	Extensibilité faible . . . . .	106
2.4.3.3	Extensibilité forte . . . . .	108
<b>2.5</b>	<b>Conclusion partielle . . . . .</b>	<b>108</b>
<b>2.6</b>	<b>Références . . . . .</b>	<b>109</b>

## 2.1 Philosophie générale de la méthode à deux niveaux

La méthode développée est décrite dans le cadre des méthodes de décomposition de domaine sans recouvrement. L'objectif est d'améliorer la vitesse de convergence des méthodes de type Schur (comme *FETI* ou *BDD*).

Comme cela a pu être vu dans la section 1.1.3.3, ce type de méthodes peut converger difficilement lorsque le problème sous-structuré présente des caractéristiques comme la présence d'hétérogénéités traversant les interfaces ou encore l'existence d'interfaces irrégulières. La méthode présentée dans ce chapitre a alors été développée afin de permettre d'éviter que le maillage sous-structuré ne présente ces caractéristiques.

Pour générer des sous-domaines homogènes, il semble possible de calibrer les poids des arêtes du graphe afin d'optimiser la position des interfaces par rapport à d'éventuelles hétérogénéités (HAVE et coll. [2013]). Cependant il paraît impossible de garantir un aspect géométrique régulier des interfaces. En effet, même des partitionnements par arbres binaires (tels que les arbres BSP ou KD), qui génèrent des sous-domaines confinés dans des zones bien définies spatialement, ne peuvent assurer de l'obtention d'interfaces régulières si le maillage n'est pas structuré. Dans le cas général, même un séparateur régulier n'existe pas dans le maillage en lui-même.

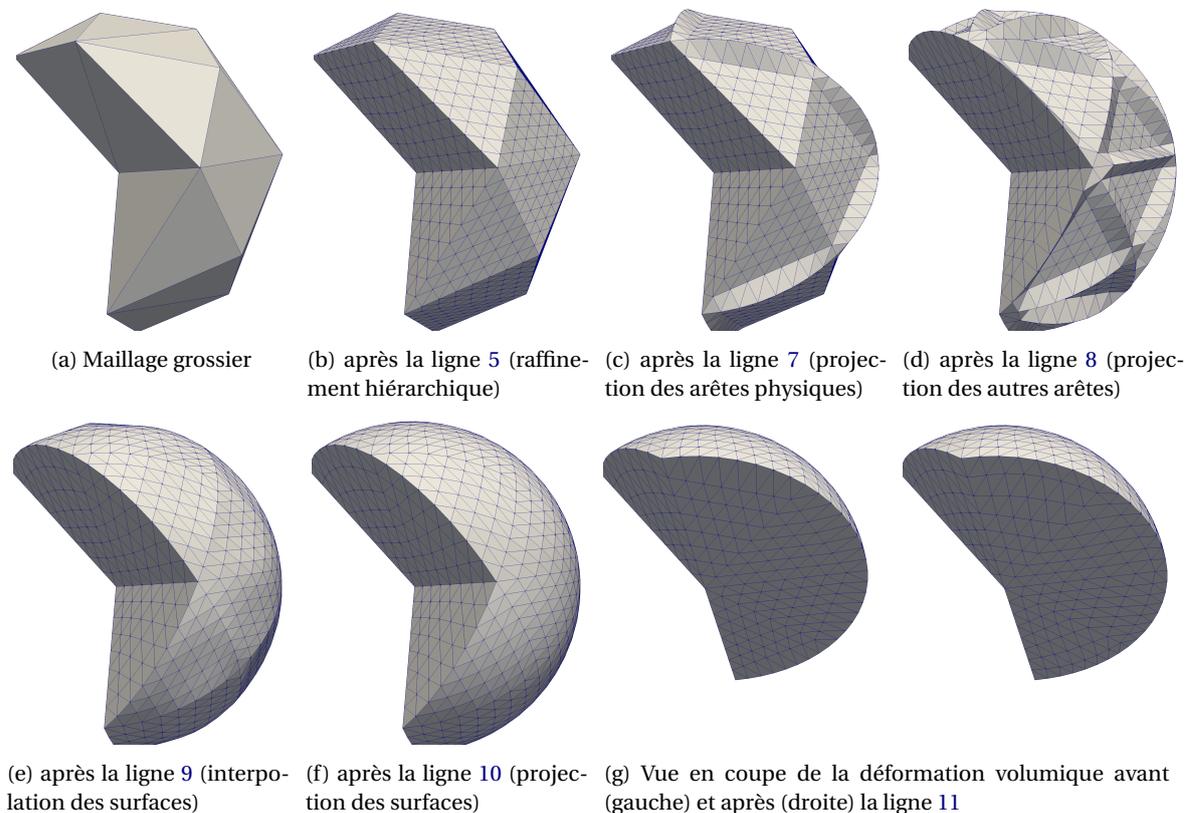


FIGURE 2.1 – Illustration des étapes parallèles de l'algorithme 2.1 appliqué à un sous-domaine inclus dans une sphère de la géométrie initiale. Ici, trois niveaux de raffinement hiérarchiques ont été utilisés pour générer le maillage final.

Il faut par conséquent s'assurer de l'existence de cette interface régulière avant même de discrétiser finement la structure. Les méthodes fondées sur un maillage grossier initial de la structure semblent les plus à même de produire cette interface régulière. C'est cette classe de méthodes (décrite dans le paragraphe 1.2.2.3) qui sera donc exploitée par la suite.

La méthode proposée dans ce chapitre, en plus de générer de la manière la plus parallèle possible un maillage comportant des interfaces régulières et possiblement adaptées aux hétérogénéités présentes dans le problème initial, permet d'améliorer le **conditionnement** du problème condensé à l'interface issu des méthodes de décomposition de domaine sans recouvrement.

Premièrement, à partir d'une géométrie sous la forme d'une CAO continue ou discrète, un maillage grossier est généré. Ce maillage grossier ne représente évidemment pas de manière précise la géométrie mais permet de définir simplement un partitionnement bien proportionné. Pour cela, les sous-domaines sont définis en tant qu'unions d'éléments grossiers et ne peuvent chevaucher plusieurs composants physiques de la structure, garantissant ainsi leur homogénéité. Ces sous-domaines sont ensuite distribués sur les différentes entités de calcul disponibles accompagnés des informations de CAO associées.

Par la suite, le maillage fin de la structure est réalisé en parallèle en conservant la contrainte de compatibilité à l'interface. Pour cette étape, les méthodes où la position des nouveaux nœuds est connue *a priori* (comme le raffinement hiérarchique) permettent en outre d'éviter les communications inter-processeurs. Finalement, l'application de techniques de déformation de maillage permet d'assurer que le maillage obtenu respecte la géométrie initiale.

Cette méthodologie est décrite dans l'algorithme 2.1 et illustrée par la figure 2.1. Un point clé de la méthode repose sur la description fine de toutes les entités géométriques pour la mise en données du problème mécanique à résoudre :

- la description des domaines physiques  $\left(\Omega_{\phi}^d\right)_d$ . Ce sont des composantes connexes qui sont, en général, associées à des propriétés matériaux constantes. Le nombre de domaines physiques sera noté  $N_{\phi}$ .
- l'ensemble des faces physiques  $\mathcal{F}_{\phi}$ . Les frontières des domaines physiques peuvent être décomposées en un ensemble de faces régulières par morceaux. Certaines d'entre elles sont le support de conditions aux limites telles qu'un chargement de type pression ou un déplacement imposé ou encore des conditions d'interface comme le contact avec ou sans frottement.
- l'ensemble des arêtes physiques  $\mathcal{E}_{\phi}$ . Ce sont les intersections des faces physiques. Elles jouent un rôle majeur dans la représentation de la géométrie et notamment pour rendre compte d'arêtes vives ou de primitives géométriques.

Ces informations peuvent être fournies dans plusieurs formats : des formats de type CAO comme le *Boundary REPresentation (B-REP)* de MÄNTYLÄ [1987], le *Standard for the Exchange of Product model data (STEP)* issu de la norme ISO/TC 184/SC 4 INDUSTRIAL DATA [2016] ou d'autres formats de géométrie spécifiques à un logiciel donné (comme le format CATPart de CATIA V5). Une autre alternative, choisie dans la méthode décrite ici, consiste à considérer un maillage fin de la peau (comme le format *STereoLithography (STL)* de 3D SYSTEMS, INC. [1988]) dont une désignation adaptée des faces et des arêtes aura pu être faite.

### 2.1.1 Maillage grossier

La première étape de la méthode, illustrée par la figure 2.2b et se déroulant à la ligne 1 de l'algorithme 2.1, consiste à générer un maillage grossier de la géométrie physique  $\left(\left(\Omega_{\phi}^d\right)_d, \mathcal{F}_{\phi}, \mathcal{E}_{\phi}\right)$ . Cette étape conduit à la construction d'ensembles discrets, éléments  $\mathcal{T}_c$ , faces  $\mathcal{F}_c$  et arêtes  $\mathcal{E}_c$ , possédant les propriétés suivantes :

- les éléments grossiers  $\mathcal{T}_c$  sont des polyèdres simples (en général des tétraèdres et parfois des hexaèdres). Chacun de ces éléments grossiers est associé à un unique domaine physique.
- les faces grossières, regroupées dans l'ensemble  $\mathcal{F}_c$ . Elles sont les frontières des éléments grossiers. On note  $\mathcal{F}_{c,\phi}$  le sous-ensemble des faces grossières associées aux faces physiques. Il faut noter que leurs sommets sont positionnés sur la face physique associée.
- les arêtes grossières, regroupées dans l'ensemble  $\mathcal{E}_c$ . Elles sont les arêtes de ces faces grossières. On note  $\mathcal{E}_{c,\phi}$  le sous-ensemble des arêtes grossières associées aux arêtes physiques  $\mathcal{E}_{\phi}$ , et  $\mathcal{E}_{c,\mathcal{F}} = \left(\partial\mathcal{F}_{c,\phi}\right) \setminus \mathcal{E}_{c,\phi}$  le sous-ensemble des arêtes grossières associées aux surfaces physiques mais pas à une arête physique. Celles-ci peuvent être les arêtes connectant deux faces grossières associées à la même surface physique. Il faut noter que, dans ces deux cas, les

**Algorithme 2.1** : Méthode de sous-structuration à deux niveaux

---

**Données :**

- $\left(\left(\Omega_{\phi}^d\right)_d, \mathcal{F}_{\phi}, \mathcal{E}_{\phi}\right)$  géométrie de la structure à simuler ;

**Entrées :**

- Nombre de sous-domaines cibles  $N_s$  ;
- Nombre de raffinements hiérarchiques  $N_r$  ;

**Début**

1 | génération du maillage grossier tridimensionnel de la géométrie :  $(\mathcal{T}_c, \mathcal{F}_c, \mathcal{E}_c)$  ;

2 | partitionnement du maillage en ses  $N_{\phi}$  différents matériaux  $(\mathcal{T}_c^d, \mathcal{F}_c^d, \mathcal{E}_c^d)_{d \in \llbracket 1, N_{\phi} \rrbracket}$  ;

**Pour**  $d \in \llbracket 1, N_{\phi} \rrbracket$  **faire**

3 |     partitionnement de chaque matériau en un nombre  $N_s^d$  de sous-domaines  
    conformes dépendant de  $N_s$  tel que  $\sum_{d=1}^{N_{\phi}} N_s^d = N_s$  ;

**Fin**

**Sortie intermédiaire :**

- maillage grossier partitionné  $(\mathcal{T}_c^{(s)}, \mathcal{F}_c^{(s)}, \mathcal{E}_c^{(s)})_{s \in \llbracket 1, N_s \rrbracket}$  ;

**Pour**  $s \in \llbracket 1, N_s \rrbracket$  **faire en parallèle** // étapes parallèles

4 |     restriction de  $\Omega_{\phi}$  dans le voisinage des faces  $\mathcal{F}_c^{(s)}$  du sous-domaine  $\Omega^{(s)}$  ;

5 |      $N_r$  raffinements hiérarchiques successifs du maillage du sous-domaine  
     $(\mathcal{T}_r^{(s)}, \mathcal{F}_r^{(s)}, \mathcal{E}_r^{(s)})$  ;

6 |     projection de  $(\mathcal{F}_r^{(s)})$  sur  $(\mathcal{F}_{\phi}^{(s)})$  :

7 |     – projection des arêtes physiques  $(\mathcal{E}_{r,\phi}^{(s)})$  à l'aide d'une méthode fondée sur  
    l'abscisse curviligne ;

8 |     – projection des arêtes restantes  $(\mathcal{E}_{r,\mathcal{F}}^{(s)})$  le long de leurs normales ;

9 |     – interpolation des surfaces  $(\mathcal{F}_r^{(s)})$  ;

10 |     – projection des surfaces  $(\mathcal{F}_{r,\phi}^{(s)})$  sur  $(\mathcal{F}_{\phi}^{(s)})$  le long de leurs normales ;

11 |     déformation de  $(\mathcal{T}_{r,\phi}^{(s)}, \mathcal{F}_{r,\phi}^{(s)}, \mathcal{E}_{r,\phi}^{(s)})$  conformément à la projection de  $(\mathcal{F}_{r,\phi}^{(s)})$  (lignes 6  
    à 10) ;

**Fin**

**Fin**

**Sortie :** maillage sous-structuré et raffiné  $(\mathcal{T}_{r,\phi}^{(s)}, \mathcal{F}_{r,\phi}^{(s)}, \mathcal{E}_{r,\phi}^{(s)})$  respectant la géométrie physique  
 $\left(\left(\Omega_{\phi}^d\right)_d, \mathcal{F}_{\phi}, \mathcal{E}_{\phi}\right)$  ;

---

sommets de ces arêtes sont déjà bien positionnés sur les faces physiques de la géométrie et ne nécessitent pas d'être déplacés.

À ce stade, il n'y a absolument aucun besoin de représenter la vraie géométrie correctement. Seules la topologie et les informations physiques nécessitent d'être sauvegardées au travers du maillage grossier.

### 2.1.2 Décomposition séquentielle

Durant cette étape, décrite par les lignes 2 et 3 de l'algorithme 2.1, l'utilisateur choisit le nombre total de sous-domaines  $N_s$  pour la sous-structuration. Ce nombre n'est pas limité au nombre d'unités de calcul disponible car de la sur-allocation peut être employée. Il est par contre encadré par des caractéristiques du maillage grossier tel que  $N_{\phi} \leq N_s \leq \#\mathcal{T}_c$ .

Chaque représentation grossière d'un domaine physique est ensuite divisée en sous-domaines de calcul  $(\Omega_c^{(s)})_s$  en ayant pour objectif d'équilibrer au mieux la charge entre les unités de cal-

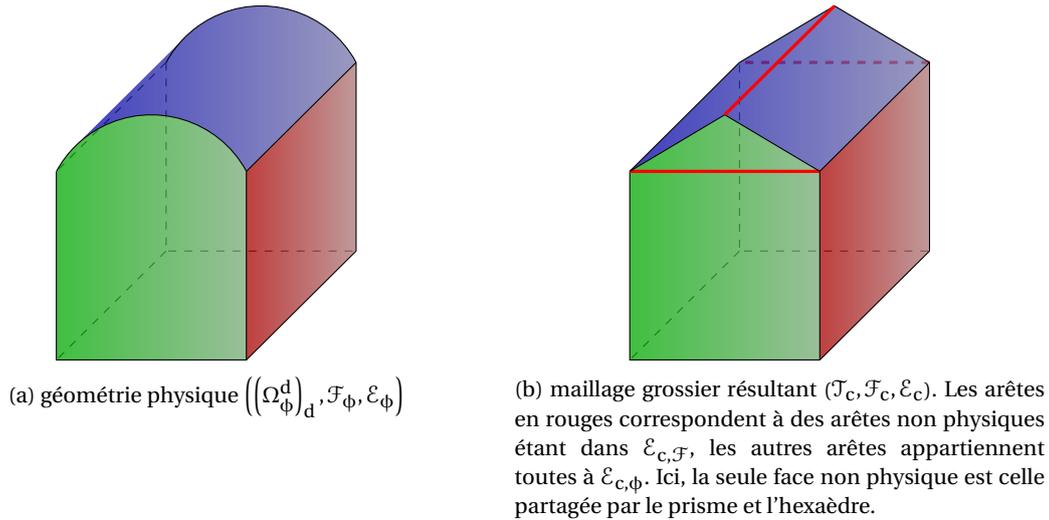


FIGURE 2.2 – Définition des différents ensembles géométriques, chaque couleur représente une surface physique différente  $\mathcal{F}_{c,\phi}^f$ .

cul. Cela peut être obtenu en choisissant un nombre sensiblement proche d'éléments grossiers dans chaque sous-domaine. Cette décomposition est réalisée par un logiciel de partitionnement de graphe tels que Scotch de PELLEGRINI et ROMAN [1996], Metis de KARYPIS et KUMAR [1998] ou encore Chaco de HENDRICKSON [2011], dont les principes théoriques ont été abordés dans la section 1.2.1. Dans le cas où la structure est hétérogène et composée de matériaux dont le coût de calcul est très différent, des poids peuvent être ajoutés sur les éléments lors du partitionnement pour équilibrer la charge de calcul. Par exemple, cela peut être utile pour les matériaux composites où les matériaux sont modélisés par un ensemble de fibres élastiques et de matrice visco-plastique non-linéaire.

#### Remarque 2.1 : Condition d'homogénéité des sous-domaines

Si le fait de préserver l'homogénéité des sous-domaines de calcul conduit à former un nombre important de sous-domaines de faible taille, la recombinaison de ces sous-domaines sous la condition de quasi monotonie décrite par PECHSTEIN et SCHEICHL [2013] est envisageable. L'idée est de construire les sous-domaines de calcul comme étant des chaînes d'éléments dont la raideur augmente. Cette construction permet de limiter la dégradation de la constante  $C$  qui contrôle l'extensibilité de la méthode de décomposition de domaine dans l'équation (1.23).

#### Remarque 2.2 : Préservation des normales

À ce stade, une information doit être conservée avant la distribution des données sur les différents processeurs. En effet, les arêtes grossières non-physiques associées aux faces physiques  $\mathcal{E}_{c,\mathcal{F}}$  doivent avoir la capacité de se projeter sur la surface physique. Comme ce ne sont pas des arêtes physiques, ces arêtes sont sur des parties « lisses » de la surface où un vecteur normal qui varie lentement peut être défini. Classiquement le vecteur normal est approximé par la moyenne des vecteurs normaux des deux faces adjacentes à l'arête. Dans le cas où ces faces physiques appartiennent à deux sous-domaines de calcul différents (comme illustré sur la figure 2.3), la construction du vecteur normal requiert une communication supplémentaire de petite taille. Pour l'éviter, les vecteurs normaux des arêtes de  $\mathcal{E}_{c,\mathcal{F}}$  sont calculés avant la décomposition.

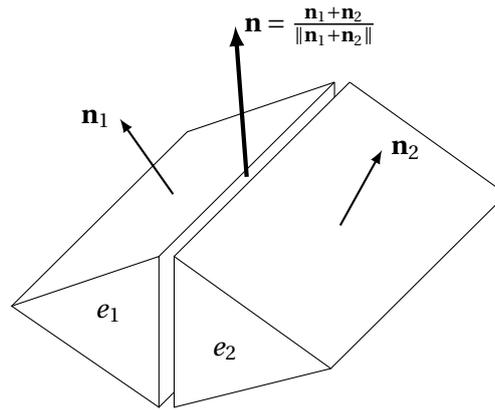


FIGURE 2.3 – Calcul de la normal d’une arête partagée entre deux éléments grossiers appartenant à des sous-domaines voisins  $e_1 \in \mathcal{T}_c^{(1)}$  et  $e_2 \in \mathcal{T}_c^{(2)}$

### Remarque 2.3 : Découpage de la géométrie sous-jacente

Pour réduire la taille de la zone sur laquelle chaque face physique  $f \in \mathcal{F}_{c,\phi}$  et chaque arête non-physique  $e \in \mathcal{E}_{c,\mathcal{F}}$  doit être projetée et ainsi de réduire le coût numérique de la projection, il est judicieux de couper le maillage fin autour de chacune de ces entités (ligne 4 de l’algorithme 2.1). Ce découpage est réalisé en utilisant l’espace délimité par les plans définis par les arêtes, qui sont les bords de la face ( $\partial f$ ), et leurs vecteurs normaux associés tels que définis dans la Remarque 2.2.

À la fin de cette étape, les données sont distribuées sur les  $N_s$  unités de calcul. Chaque sous-domaine connaît ainsi son maillage grossier, son maillage fin de la géométrie et tous les ensembles physiques qui ont été marqués. Le reste de l’algorithme 2.1 est réalisé en parallèle. Bien sûr, la connectivité entre les sous-domaines est connue et un **communicateur** fondé sur la topologie du graphe de connectivité des sous-domaines peut être créé. Ce **communicateur**, dit topologique, permet aux sous-domaines de connaître quels sont leurs voisins permettant ainsi de faciliter l’implantation des échanges entre voisins (`MPI_Neighbor_alltoall`) qui sont très courants dans les méthodes de décomposition de domaine, notamment dans les opérations d’assemblage. De plus, ce **communicateur** topologique permet une amélioration des performances lors des échanges ainsi qu’une meilleure lisibilité du code.

### 2.1.3 Raffinement local

Le maillage grossier construit précédemment a été conçu pour distribuer les données. Ainsi, un raffinement est nécessaire (ligne 5 de l’algorithme 2.1) pour réaliser un calcul à la précision souhaitée. Différentes stratégies sont détaillées ci-après :

**Raffinement hiérarchique.** Le  $h$ -raffinement et le  $p$ -raffinement peuvent être identiquement considérés (voir figure 2.4). Pour que la conformité à l’interface entre les sous-domaines soit assurée, il suffit que le même raffinement soit utilisé dans chaque sous-domaine (c’est l’option la plus simple pour s’assurer de la conformité). La qualité du maillage fin dépend de la qualité des éléments grossiers et du nombre  $N_r$  d’appels successifs au  $h$ -raffinement. Un avantage de cette approche est de générer des éléments grossiers raffinés de même topologie, ce qui est particulièrement bien adapté pour l’accélération de calculs à l’aide de GPU comme le montrent [LE GOUEZ et ETANCELIN \[2015\]](#).

**Maillage topologique.** En supposant qu’une taille d’élément est disponible à chaque nœud grossier et que le même algorithme déterministe de maillage est utilisé sur chaque sous-domaine, il est possible de remailler la pièce avec de faibles échanges. Pour ce faire, les sous-domaines remaillent d’abord leurs arêtes, puis leurs faces et enfin leurs volumes. Les interfaces seront alors bien conformes en raison de la nature déterministe du remaillage

de l'interface. Seuls des échanges d'indices sont nécessaires pour reconstruire les données d'interface.

**Approche maître-esclave.** C'est le type de techniques qu'ont pu développer ITO et coll. [2007], YILMAZ et OZTURAN [2015] ou BOVET et coll. [2019] dont la méthode est décrite sur la figure 1.25 où un coloriage permet de définir un ordre de remaillage et d'échanges de frontières.

**Interfaces non conformes.** Une dernière possibilité est que tous les sous-domaines se remaillent en parallèle indépendamment. Les maillages n'étant pas conformes aux interfaces entre les sous-domaines, une communication est nécessaire pour construire les opérateurs d'assemblage de type Mortar à l'interface comme DRYJA [2005]; DRYJA et PROSKUROWSKI [2005]; STEFANICA [2001] ont pu développer. Dans cette approche, le solveur requiert un traitement particulier.

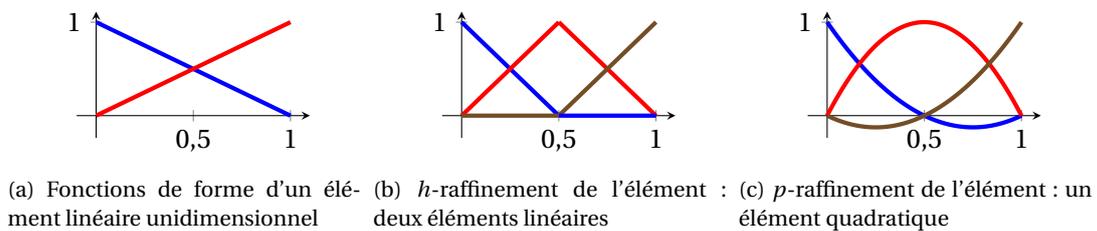
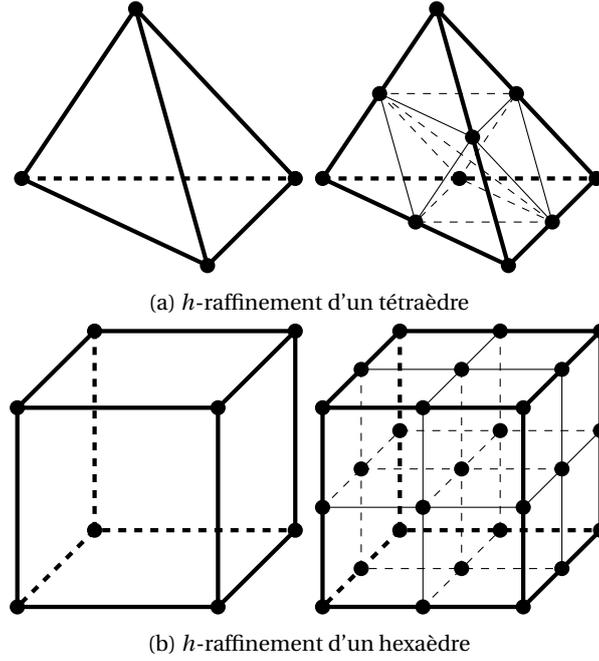


FIGURE 2.4 – Illustration du  $h$ -raffinement et du  $p$ -raffinement sur un élément unidimensionnel. Chaque fonction de forme a une couleur différente.

Quelle que soit la stratégie retenue, seule une très faible quantité d'information doit être échangée entre les sous-domaines voisins. Dans un souci de simplicité, une méthode fondée sur le recours à une ou plusieurs étapes successives de  $h$ -raffinement a été retenue par la suite. Par contre aucune exploitation de l'accélération par l'identité topologique des maillages grossiers raffinés ne sera faite par la suite. Le schéma de  $h$ -raffinement choisi est décrit sur la figure 2.5 où chaque arête est divisée en deux plus petites arêtes de mêmes tailles. Il faut noter que la qualité des maillages hexaédriques est conservée après une étape de  $h$ -raffinement alors qu'elle se dégrade pour les maillages tétraédriques.

On note  $\Omega_r^{(s)}$  (représenté figure 2.1b) la version raffinée du maillage  $\Omega_c^{(s)}$ . De plus, tous les ensembles existants sur  $\Omega_c^{(s)}$  et définis en 2.1.1 sont reportés sur  $\Omega_r^{(s)}$  en y incluant les nœuds nouvellement créés par le raffinement.


 FIGURE 2.5 – Schémas de  $h$ -raffinement utilisés

## 2.2 Projection des différentes entités constitutives

Après raffinement, le maillage  $(\Omega_r^{(s)})_s$  ne représente pas encore précisément la géométrie exacte décrite par  $\cup_d \partial\Omega_\phi^d$  lorsque celle-ci est courbe. Pour chaque sous-domaine  $\Omega_r^{(s)}$ , le maillage final est alors obtenu par une suite de projections sur la géométrie exacte qui ne nécessite aucune communication entre les sous-domaines. Dans la suite, seules des opérations locales seront effectuées et l'exposant  $(s)$  pourra être omis.

### 2.2.1 Projection des arêtes physiques $\mathcal{E}_{r,\phi}^{(s)}$ sur $\mathcal{E}_\phi^{(s)}$

Cette étape est décrite à la ligne 7 de l'algorithme 2.1. Chaque arête  $e_r \in \mathcal{E}_{r,\phi}^{(s)}$  est associée à une arête  $e_\phi \in \mathcal{E}_\phi^{(s)}$  qui doit être représentée correctement dans le maillage final. Il faut noter que, par construction, les nœuds aux extrémités des arêtes  $e_r$  et  $e_\phi$  associées coïncident déjà. Pour réaliser un projection adéquate, une solution simple consiste à faire une correspondance des abscisses curvilignes normalisées des deux arêtes.

Les arêtes sont ici considérées en tant que courbes orientées de manière similaire (même nœud de départ et même nœud d'arrivée). Soit  $(\mathbf{x}_{r,m})_{1 \leq m \leq M}$  la suite ordonnée des  $M$  nœuds de  $e_r$ . Une description identique  $(\mathbf{x}_{\phi,n})_{1 \leq n \leq N}$  est faite pour les  $N$  nœuds de  $e_\phi$ .

Soit  $\ell$  cette abscisse curviligne normalisée, elle peut être calculée sur  $e_r$  et  $e_\phi$  aux  $i^{\text{ème}}$  et  $j^{\text{ème}}$  nœuds, respectivement, par :

$$\ell_{r,i} = \frac{\sum_{m=1}^i \|\mathbf{x}_{r,m+1} - \mathbf{x}_{r,m}\|_2}{\sum_{m=1}^{M-1} \|\mathbf{x}_{r,m+1} - \mathbf{x}_{r,m}\|_2}, \quad \ell_{\phi,j} = \frac{\sum_{m=1}^j \|\mathbf{x}_{\phi,m+1} - \mathbf{x}_{\phi,m}\|_2}{\sum_{m=1}^{N-1} \|\mathbf{x}_{\phi,m+1} - \mathbf{x}_{\phi,m}\|_2}.$$

Les nœuds de  $e_r$  peuvent être déplacés sur  $e_\phi$  à l'abscisse curviligne correspondante comme le montre la figure 2.6 : pour le  $k^{\text{ème}}$  nœud de  $e_r$ ,  $K$  est défini comme étant le rang de  $e_\phi$  tel que  $\ell_{r,k} \in [\ell_{\phi,K}, \ell_{\phi,K+1}]$ . Ainsi, une interpolation linéaire permettant de trouver la nouvelle position de  $\mathbf{x}_{r,k}$ , notée  $\mathbf{x}_{r,k}^{\text{new}}$  peut être écrite :

$$\mathbf{x}_{r,k}^{\text{new}} = \alpha \mathbf{x}_{\phi,K} + (1 - \alpha) \mathbf{x}_{\phi,K+1}, \quad \text{avec } \alpha = \frac{\ell_{r,k} - \ell_{\phi,K+1}}{\ell_{\phi,K} - \ell_{\phi,K+1}}.$$

Le résultat de cette opération est illustré sur la figure 2.1c.

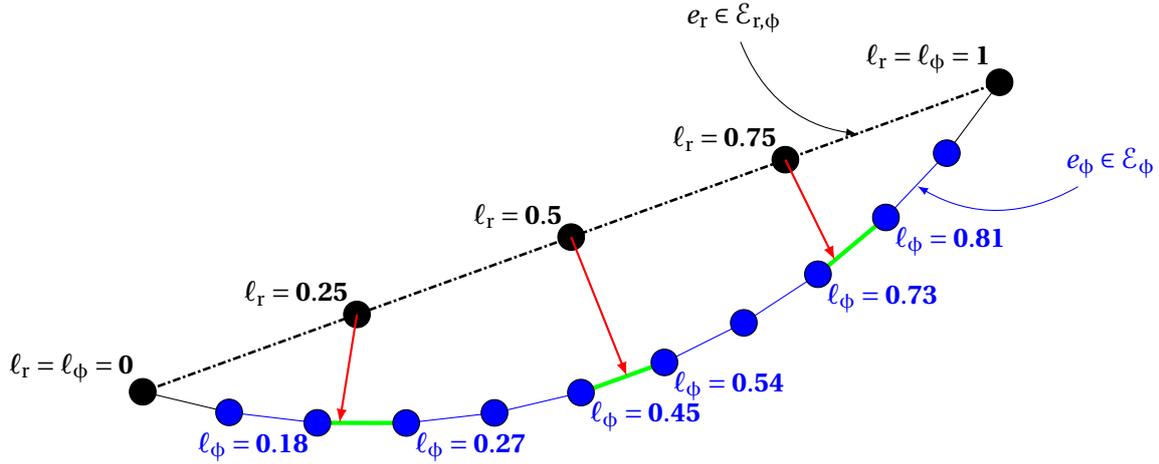


FIGURE 2.6 – Projection de l’arête  $e_r \in \mathcal{E}_{r,\phi}$  (en traits pointillés noirs) sur l’arête correspondante  $e_\phi \in \mathcal{E}_\phi$  (en trait plein bleu),  $l_r$  et  $l_\phi$  sont les abscisses curvilignes normalisées. Les lignes vertes correspondent aux segments où la condition  $l_{r,k} \in [l_{\phi,k}, l_{\phi,k+1}]$  est satisfaite pour chaque nœud  $\mathbf{x}_{r,m}$  de  $e_r$ .

## 2.2.2 Projection des arêtes de la frontière restantes

Les nœuds des arêtes restantes sur la frontière et qui appartiennent à l’ensemble  $\mathcal{E}_{r,\mathcal{F}} = (\mathcal{E}_r \cap (\partial\mathcal{F}_{c,\phi})) \setminus \mathcal{E}_{r,\phi}$  n’ont encore reçu aucun traitement; il reste donc à les projeter sur leur surface physique associée (ligne 8 de l’algorithme 2.1). Toutes les arêtes sont garanties de posséder un vecteur normal sans besoin supplémentaire de communication (voir remarque 2.2). Plus précisément, la direction  $\mathbf{d}_e$  est choisie comme étant la somme des vecteurs normaux des deux faces physiques adjacentes à l’arête. Par exemple, si  $e_r \in \mathcal{E}_{r,\mathcal{F}}$ , il existe  $(f_{e,1}, f_{e,2}) \in \mathcal{F}_{c,\phi}^2$  tel que  $e_r = \partial f_{e,1} \cap \partial f_{e,2}$ . Soient  $\mathbf{n}_1$  et  $\mathbf{n}_2$  les deux vecteurs normaux associés à  $f_{e,1}$  et  $f_{e,2}$  respectivement, il est possible de définir une direction moyenne  $\mathbf{d}_e = \frac{\mathbf{n}_1 + \mathbf{n}_2}{\|\mathbf{n}_1 + \mathbf{n}_2\|}$ .

Pour déplacer ces arêtes, l’algorithme d’intersection rayon-triangle de MÖLLER et TRUMBORE [1997] est utilisé. Cet algorithme a l’avantage d’éviter le calcul de l’équation des plans des triangles, ce qui représente une économie de mémoire importante. De plus, cette méthode semble être la plus rapide de la littérature d’après les auteurs. Cet algorithme détaillé dans la section 1.3.1.2 et l’algorithme 1.4, peut être vectorisé pour tous les triangles de  $(\Omega_\phi^d)_d$  et pour tous les nœuds de  $\mathcal{E}_{r,\mathcal{F}}$  afin d’accélérer son exécution. Le résultat de cette opération est illustré sur la figure 2.1d.

## 2.2.3 Projection des surfaces

À ce stade, l’ensemble des arêtes a été projeté. Par la suite, comme il faut obtenir des faces raffinées dont les nœuds coïncident avec la surface physique et sont répartis de manière homogène, une projection suffisamment régulière est nécessaire. Celle-ci est réalisée en deux étapes. Dans un premier temps, les surfaces sont déformées selon le déplacement de leurs arêtes à l’aide d’une interpolation sur chaque face  $f_r \in \mathcal{F}_{r,\phi}$ . Dans un second temps, une projection de tous les nœuds de la surface le long de leur normale est réalisée.

La première étape (l’interpolation de la ligne 9 de l’algorithme 2.1) ne prend en compte ni la topologie sous-jacente ni la géométrie physique; la seule information nécessaire est le déplacement  $\mathbf{u}_b$  de toutes les arêtes de bord de  $\mathcal{E}_r \cap (\partial\mathcal{F}_{c,\phi})$ . Cette interpolation est réalisée à l’aide de *Radial Basis Functions* ou fonctions de base radiales (*RBF*) comme décrit par DE BOER et coll. [2007] et détaillée dans la section 1.3.2.2. Ce choix a été motivé par la simplicité d’implémentation de cette méthode sans maillage et par la qualité des éléments déformés obtenus comme le montrent SELIM et KOMULLIL [2016]. La fonction retenue est CP  $\mathcal{C}^2$  en tant que *RBF*  $\psi$ , qui nous semble être la plus adaptée au regard de la littérature (DE BOER et coll. [2007]; SELIM et KOMULLIL [2016]) et de l’étude menée dans la section 1.3.2.4, à laquelle un rayon support de taille  $r = 5 \max(\|\mathbf{u}_b\|)$  est associé. Le résultat de cette première étape est illustré sur la figure 2.1e. Celle-ci a deux intérêts

majeurs :

- sur les faces physiques, elle permet de rapprocher les nœuds de la surface réelle.
- sur les faces non-physiques qui sont à l'interface entre les sous-domaines, elle permet de répartir les nœuds sur celle-ci lorsqu'une ou plusieurs arêtes ont été déplacées.

Ce dernier aspect peut être observé au passage de la figure 2.1d à la figure 2.1e. En effet, la surface visible interne à la sphère a une bien meilleure répartition des nœuds sur la figure 2.1e que sur la figure 2.1d.

Une fois les nœuds des faces interpolés conformément à leur déplacement d'arêtes, ils peuvent être projetés sur la surface physique grâce à une seconde étape (ligne 10 de l'algorithme 2.1). De manière similaire aux projections d'arêtes non-physiques de la section 2.2.2, un algorithme d'intersection rayon-triangle est utilisé pour projeter toutes les faces physiques. Cette projection est réalisée le long des normales aux faces grossières calculées dès l'initialisation de la méthode. Le résultat de cette opération est illustré sur la figure 2.1f.

## 2.3 Méthode de déformation de maillage hiérarchique

Une fois tous les nœuds du bord  $\partial\Omega^{(s)}$  projetés, il faut déplacer les nœuds internes au sous-domaine. Cela est réalisé à l'aide d'une méthode par interpolation *RBF*. Dans l'équation (2.1), seule la partie sans la matrice  $\mathbf{P}_b$  est considérée pour la direction  $j \in \llbracket 1, d \rrbracket$  :

$$\Psi \mathbf{a}^j = \mathbf{u}_{bj}. \quad (2.1)$$

l'article de BUHMANN [2000] montre que ce système est défini positif en raison du caractère défini positif des *RBFs* à support compact telles que la fonction CP  $\mathcal{C}^2$  utilisée ici. La présence de la matrice  $\mathbf{P}_b$  n'est nécessaire que lorsque la *RBF* choisie n'est pas définie positive et permet au système d'admettre une unique solution.

Dans cette section, les défauts des méthodes de déformation de maillage par *RBF* seront exposés puis des méthodes visant à les contourner seront détaillées. Les temps de calcul exposés sont mesurés sur un seul processeur d'une machine Intel Xeon E5-2680 v3 cadencé à 2,50 GHz.

### 2.3.1 Limitations de la méthode RBF

Un rayon d'influence *RBF*  $r$  faible rend la matrice  $\Psi$  de l'équation (2.1) plus creuse au détriment de la qualité des éléments produits. En effet, les déplacements du bord ne sont pas bien pris en compte à l'intérieur du sous-domaine lorsque  $r$  est faible. Il faut donc choisir un rayon d'influence suffisamment grand. Par contre, si ce rayon est grand, la largeur de bande de l'opérateur  $\Psi$  augmente et les temps de calcul ainsi que la mémoire occupée peuvent devenir prohibitifs, en particulier sur des problèmes tridimensionnels lors de l'utilisation de solveurs directs creux.

En outre, plus l'opérateur est de grande taille moins bon est le conditionnement de celui-ci et un solveur itératif tel que le *CG* convergera difficilement. Ce conditionnement se dégrade car, lorsque l'opérateur devient grand, l'indépendance linéaire entre les différentes colonnes de celui-ci diminue en raison de la proximité des nœuds les uns par rapport aux autres.

Un exemple de l'influence de la taille de la matrice  $\Psi$  sur les performances de la méthode *RBF* est réalisé sur un cube de côté  $H = 1$  unitaire et illustré sur la figure 2.7, auquel est appliqué sur la face supérieure un déplacement de la forme :

$$\mathbf{u}(x, y, z = H) = \frac{H}{10} \sin\left(\frac{\pi x}{H} + \pi\right) \sin\left(\frac{3\pi y}{H}\right) \mathbf{e}_z.$$

Ce cube est discrétisé à l'aide d'un maillage tétraédrique structuré et différents raffinements hiérarchiques successifs sont appliqués pour augmenter significativement le nombre de nœuds et

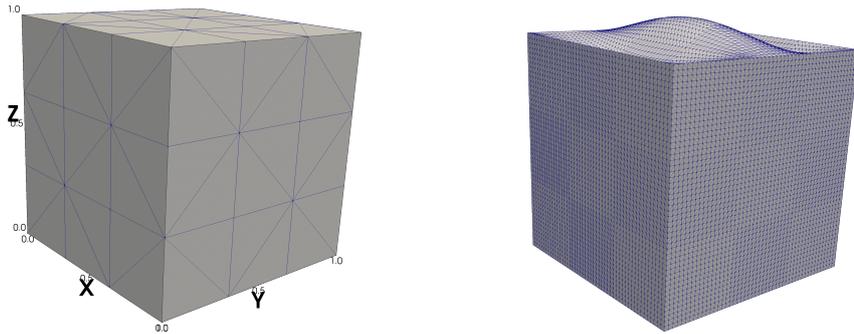
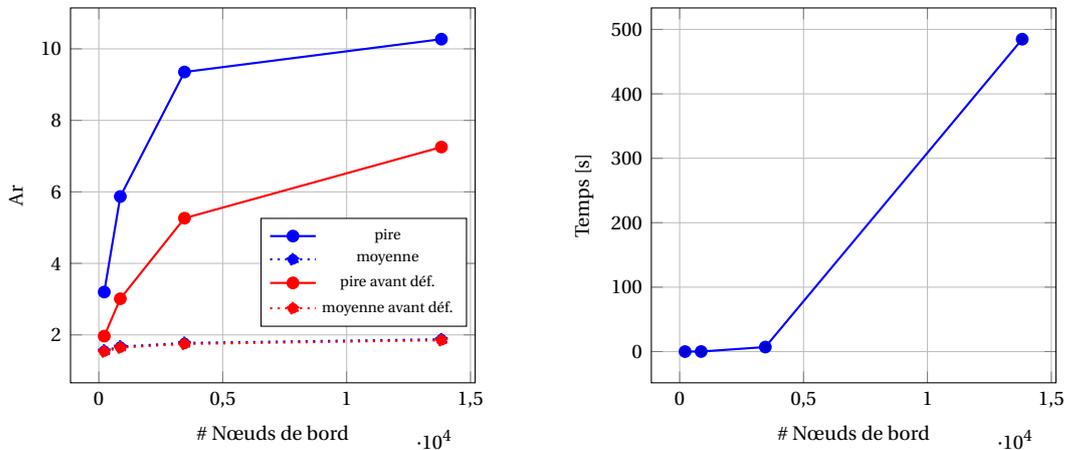


FIGURE 2.7 – Cube unité

d'éléments du maillage. Sur la figure 2.8, les points successifs représentés sur les courbes correspondent à des niveaux de raffinement différents. Pour évaluer la qualité de la déformation obtenue par la méthode *RBF*, un indicateur de qualité de type *aspect ratio* est utilisé. Sa définition, telle qu'implémentée dans Z-set, est :

$$Ar(E) = \frac{\sqrt{6}}{6} \max_{i,j} (d_{ij}) \frac{\sum_k A_k}{\det(J)},$$

où  $E$  est le tétraèdre considéré et  $d_{ij}$  est la distance entre les nœuds  $i$  et  $j$  de  $E$ ,  $A_k$  est la surface du triangle  $k \in [1..4]$  des faces de  $E$  et  $J$  est la matrice jacobienne de  $E$ . Cet indicateur varie entre 1 et  $+\infty$  : 1 étant la qualité optimale d'un tétraèdre régulier. Dans les figures rencontrées par la suite, on parlera de pire qualité et qualité moyenne comme étant respectivement les quantités  $\max_E Ar(E)$  et  $\frac{1}{\#J_r} \sum_E Ar(E)$ .



(a) pires qualités et qualités moyennes des maillages pour un nombre croissant de nœuds de bord avant et après déformation

(b) temps nécessaire pour résoudre le système *RBF* (2.1) pour différents nombres de nœuds de bord à l'aide d'une factorisation LU

FIGURE 2.8 – Influence du nombre de nœuds de bord sur l'efficacité de la méthode de déformation par *RBF* pour un rayon d'influence  $r = 0.5$

La figure 2.8 montre que lorsque le nombre de nœuds augmente, à rayon  $r$  fixé, le maillage perd en qualité. Elle montre également que le temps de résolution à l'aide d'un solveur direct creux pour réaliser la déformation devient très important. Cela n'est pas seulement dû à l'augmentation de la taille de  $\Psi$ , mais aussi à celle de la densité de coefficients non nuls dans  $\Psi$  comme la figure 2.9 permet de mettre en évidence. Cela influence directement l'étape de factorisation de l'opérateur.

L'impact de la variation du rayon d'influence est, quant à lui, illustré par la figure 2.10. La qualité des éléments obtenus devient très mauvaise lorsque  $r$  est faible, alors que le temps de résolution décroît. Par ailleurs, aucune amélioration n'est notable pour  $r \geq 0.5$ . Par contre, il convient de

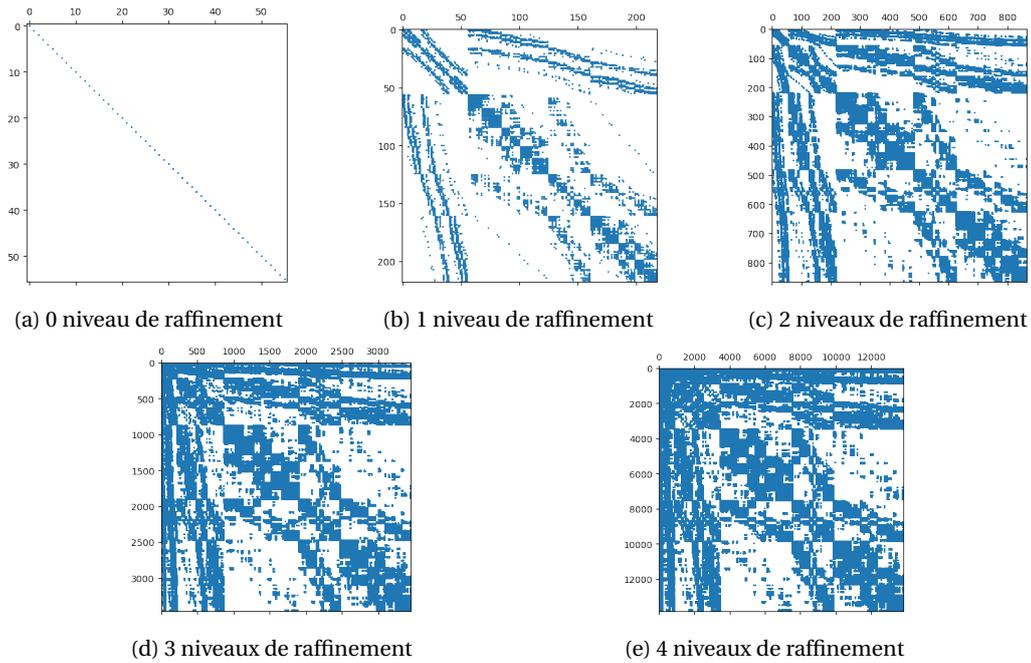


FIGURE 2.9 – Remplissage de la matrice *RBF*  $\Psi$ , pour un rayon d'influence  $r = 0.5$  et un niveau de raffinement croissant

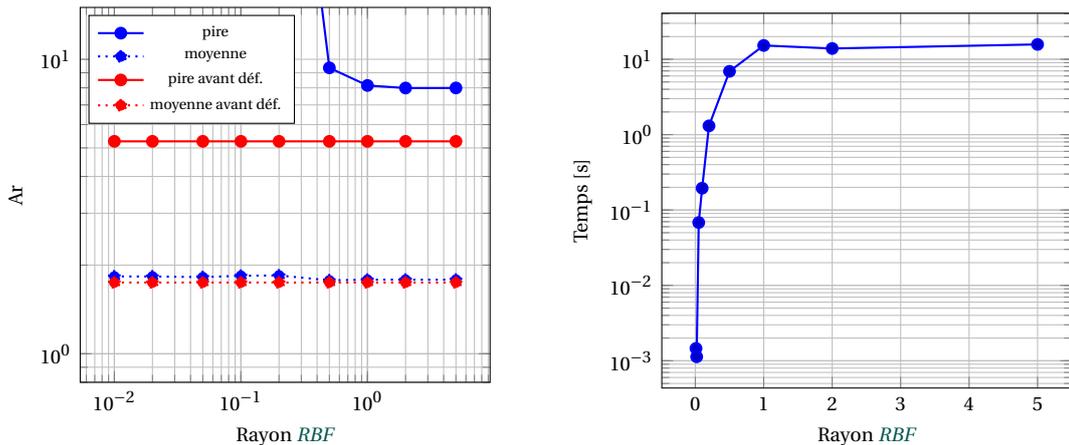


FIGURE 2.10 – Influence du rayon d'influence sur l'efficacité des *RBF* pour  $n_b = 3\,458$ .

ne pas choisir  $r$  trop grand car le temps de calcul peut vite devenir important en raison de l'augmentation de la densité de  $\Psi$ , comme l'illustre la figure 2.11. Cela est dû au fait que les solveurs directs creux perdent fortement en efficacité lorsque la densité de l'opérateur devient importante. De plus, l'opération finale de l'équation (1.27) permettant de calculer le champ déformé est aussi plus coûteuse en raison de l'efficacité du produit matrice-vecteur qui devient moindre lorsque  $\Psi$  est plus dense.

Pour réaliser la déformation de maillage de manière efficace, il est donc nécessaire de trouver un compromis entre temps de calcul et qualité de maillage. Les méthodes développées dans la section suivante ont pour objectif d'éviter de faire ce compromis.

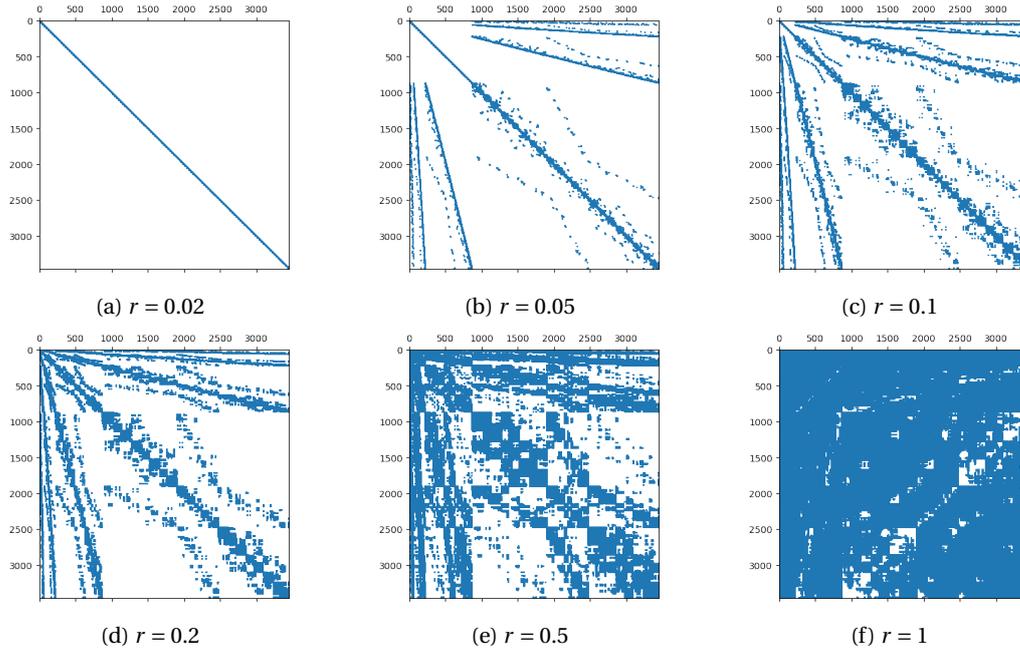


FIGURE 2.11 – Densité de la matrice  $RBF \Psi$ , pour une taille de matrice fixée à  $n_b = 3\,458$  et différents rayons d'influence  $r$ .

### 2.3.2 Résolution par un solveur itératif

Pour parvenir à réaliser la déformation de manière rapide avec un grand rayon  $r$ , une résolution itérative semble être une bonne alternative car elle est moins sensible au remplissage de  $\Psi$ . Le système (2.1) étant symétrique défini positif, un solveur itératif comme la méthode du  $CG$  peut être considéré. Son application à la place d'un solveur creux direct est étudiée dans cette partie.

#### 2.3.2.1 Influence du critère d'arrêt

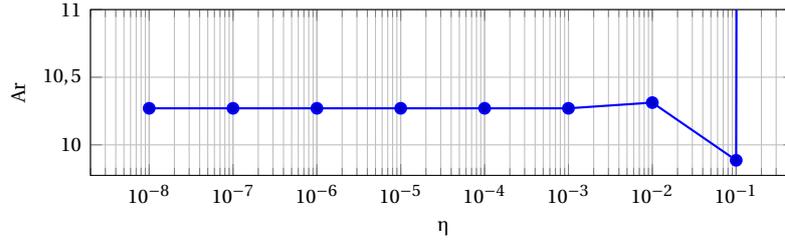
Une idée pour réduire le temps de calcul de ce  $CG$  est de choisir un critère relatif de convergence (avec un seuil  $\eta$ ) moins précis pour limiter le nombre d'itérations du solveur car il n'est pas nécessaire de trouver la solution exacte, mais une solution telle que le résidu  $\mathbf{r}_k = \Psi \mathbf{a}_k^j - \mathbf{u}_{bj}$  à l'itération  $k$  soit suffisamment faible pour que la qualité du maillage soit suffisante selon un critère donné. Ce critère d'arrêt relatif sur le résidu consiste à arrêter le solveur lorsque :

$$\frac{\|\mathbf{r}_k\|_2}{\|\mathbf{u}_b\|_2} = \frac{\sqrt{\mathbf{r}_k^T \mathbf{r}_k}}{\sqrt{\mathbf{u}_b^T \mathbf{u}_b}} < \eta$$

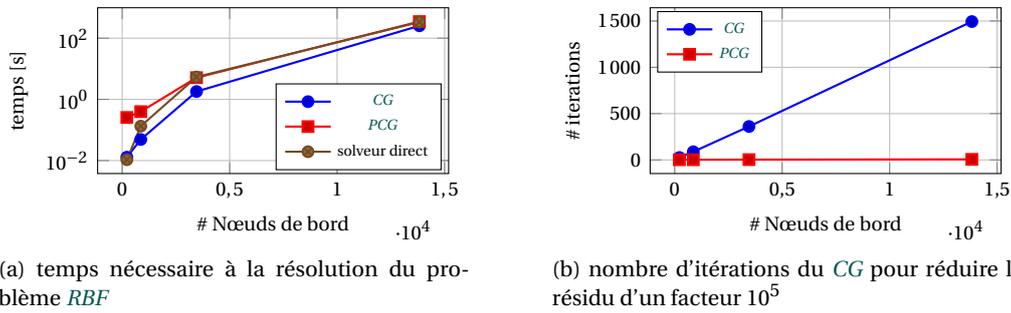
Le seuil de convergence du solveur peut donc être adapté en fonction de la qualité de maillage voulue.

La figure 2.12 montre que choisir une faible valeur pour le seuil de convergence du  $CG$  ne conduit pas à améliorer de façon substantielle la qualité du maillage déformé. Ainsi, un critère à  $\eta = 10^{-3}$  est largement suffisant, des valeurs plus faibles ne conduisant pas à améliorer la qualité du maillage, tout en augmentant le nombre d'itérations et donc le temps de calcul.

Une autre idée consiste à utiliser un algorithme de  $CG$  Préconditionné, ou *Preconditioned CG* ( $PCG$ ) permettant de réduire le conditionnement du système. En effet, le nombre d'itérations à convergence du solveur dépend directement du conditionnement de  $\Psi$ . Un choix de préconditionneur est donc primordial pour garantir une bonne efficacité de la méthode. La figure 2.13 montre que le préconditionnement du système  $RBF$  à l'aide d'une factorisation  $LU$  Incomplète ( $ILU$ ) (voir section 2.3.2.2) avec une approximation de  $\Psi$  très proche de la matrice initiale, conduit à un nombre d'itérations à convergence beaucoup plus faible quelle que soit la taille du système.


 FIGURE 2.12 – Pire qualité du maillage à  $n_b = 13\,826$  nœuds de bord pour différent critères d'arrêt du *CG*

Par contre le calcul de cette inverse est coûteux à tel point que le temps de calcul est même supérieur à celui du solveur direct. Il est donc nécessaire de trouver des méthodes pour calculer à moindre coût un préconditionneur approximant l'inverse de  $\Psi$  de manière plus creuse, permettant une résolution plus rapide.


 FIGURE 2.13 – Comparaison de différentes stratégies de résolution du système (2.1) pour différentes tailles; le préconditionnement est réalisé à l'aide de la factorisation incomplète *ILU* de  $\Psi$  avec  $\epsilon = 5 \cdot 10^{-5}$ .

### 2.3.2.2 Préconditionneur issu d'une factorisation incomplète

Un préconditionneur issu d'une factorisation LU incomplète *ILU(0)* avec un coefficient de tolérance a été choisi. Pour réaliser une factorisation LU incomplète avec une tolérance  $\epsilon$ , comme le font [JOVANOVIĆ et coll. \[1993\]](#), il faut dans un premier temps construire une matrice tronquée  $\Psi'$  définie telle que :

$$\Psi'_{ij} = \begin{cases} \Psi_{ij} & \text{si } i \neq j \text{ et } |\Psi_{ij}| > \epsilon \|\Psi_i\|_2 \\ \Psi_{ij} & \text{si } i = j \\ 0 & \text{sinon} \end{cases} .$$

où  $\Psi_i$  est la  $i^{\text{e}}$  ligne de  $\Psi$ . Ensuite, une factorisation incomplète de  $\Psi'$  est réalisée. Elle se met sous la forme :

$$\Psi' = \mathbf{L}\mathbf{U} - \mathbf{R},$$

où  $\mathbf{L}$  est une matrice triangulaire inférieure à coefficients unitaires sur la diagonale,  $\mathbf{U}$  est une matrice triangulaire supérieure. La matrice  $\mathbf{L} + \mathbf{U}$  a la même structure creuse que  $\Psi'$  dans le cas d'*ILU(0)*. Ainsi la matrice  $\mathbf{R}$  est un résidu des coefficients non-nuls retirés de  $\mathbf{L}$  et  $\mathbf{U}$ .

Ce préconditionneur est obtenu à l'aide de la bibliothèque SuperLU de [DEMME et coll. \[1999\]](#) interfacée avec le module SciPy de Python dans les résultats suivants. Sur la figure 2.13, la courbe *PCG* a été obtenue à l'aide d'un préconditionneur *ILU(0)* avec  $\epsilon = 5 \cdot 10^{-5}$ .

Les figures 2.15b et 2.14, montrent qu'à partir d'une certaine valeur de  $\epsilon$ , le préconditionneur est trop loin de l'inverse de  $\Psi$  pour permettre une résolution du système en un nombre faible d'itérations. Une procédure pour atteindre la convergence en un nombre faible d'itérations avec un coût faible du préconditionnement, serait de déterminer un  $\eta$  dépendant de  $\epsilon$  permettant une convergence du solveur et évitant les stagnations. En effet, même avec un critère de convergence

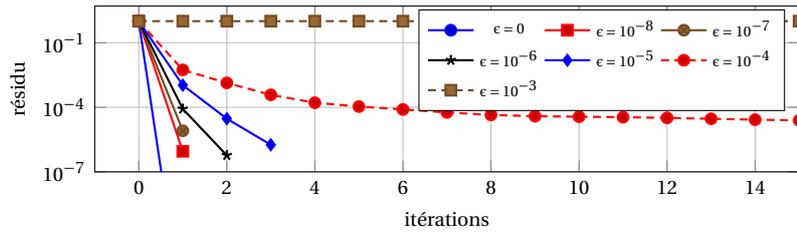
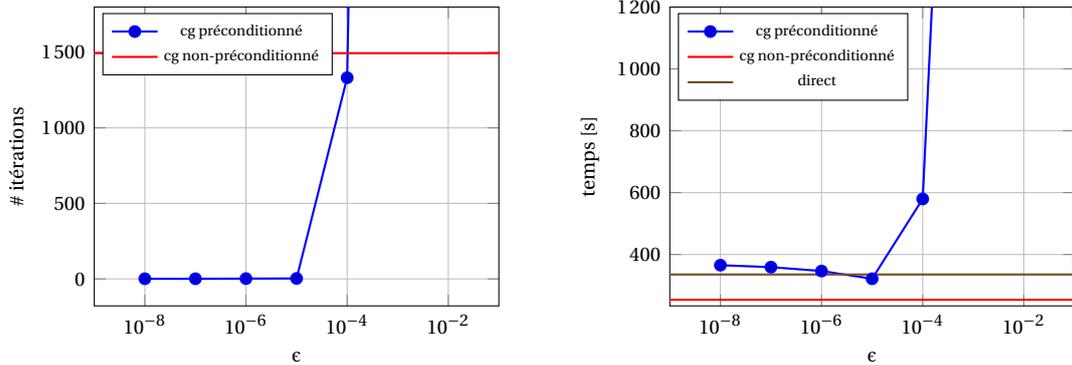


FIGURE 2.14 – Influence du paramètre  $\epsilon$  sur la qualité du préconditionneur  $ILU(0)$  d'un  $PCG$ , pour un problème de taille  $n_b = 13\,826$



(a) nombre d'itérations à convergence

(b) temps nécessaire pour résoudre le système  $RBF$

FIGURE 2.15 – Influence de  $\epsilon$  sur l'efficacité du préconditionnement pour  $n_b = 13\,826$  nœuds pour un seuil de convergence  $\eta = 10^{-5}$

de  $\eta = 10^{-3}$ , la solution donne des éléments de suffisamment bonne qualité (voir figure 2.12). Cependant ce type de procédure semble difficile à mettre en place en pratique.

### 2.3.2.3 Préconditionneur exploitant une troncature de $\Psi$

Une autre idée pour trouver une approximation de  $\Psi^{-1}$  est de prendre une inverse d'une approximation de  $\Psi$  notée  $\tilde{\Psi}$ . On choisit cette dernière comme une troncature de la matrice  $\Psi$ . Soit  $\mu \in [0, 1]$ , un coefficient réglant le niveau de troncature tel que :

$$\tilde{\Psi}_{ij} = \begin{cases} \Psi(\|\mathbf{x}_{b_i} - \mathbf{x}_{b_j}\|) & \text{si } \|\mathbf{x}_{b_i} - \mathbf{x}_{b_j}\| \leq \mu \cdot r, \\ 0 & \text{sinon.} \end{cases}$$

$\tilde{\Psi}$  étant plus creuse que  $\Psi$ , son inversion par un solveur direct est plus efficace. Lorsque  $\mu$  devient trop petit, le préconditionneur n'est plus positif et ne peut plus être utilisé pour un  $PCG$  car sa factorisation de Cholesky est compromise. Les figures 2.17a et 2.16 montrent qu'à partir de  $\mu < 0.9$ , la

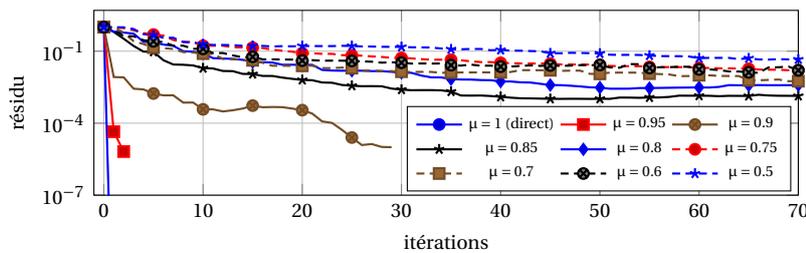


FIGURE 2.16 – Influence du paramètre de troncature  $\mu$  sur la qualité du préconditionneur de  $\Psi$  tronqué, pour un problème de taille  $n_b = 3\,458$  lors d'une résolution par  $PCG$

méthode de préconditionnement perd en efficacité à tel point qu'elle dégrade le conditionnement du système.

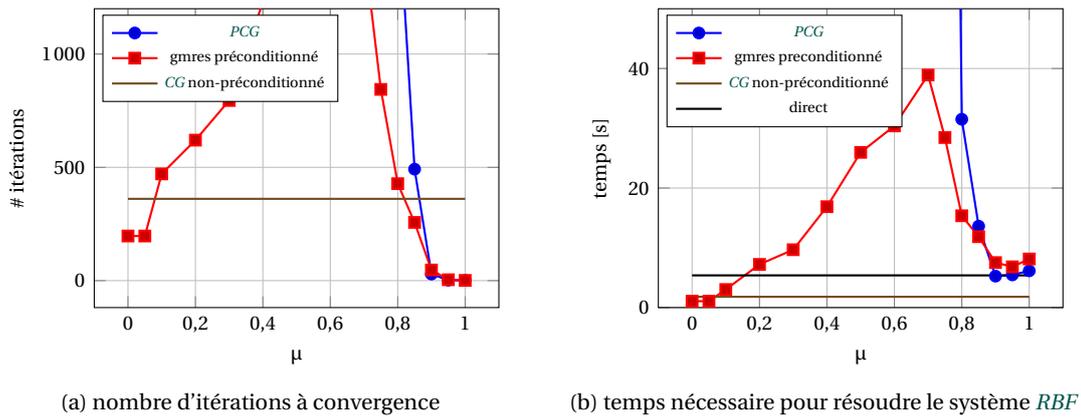


FIGURE 2.17 – Influence de  $\mu$  sur l'efficacité de solveurs itératifs préconditionnés pour  $n_b = 3458$  pour arriver à faire diminuer le résidu d'un facteur  $10^5$

De plus, le temps pour générer ce préconditionneur (voir figure 2.17b) est très important (de l'ordre de grandeur de la résolution directe lorsque  $\mu$  s'approche de 1). Cela montre que ce type de préconditionnement n'est pas adapté au problème *RBF*. En effet, la troncature ne permet pas de réduire significativement la largeur de bande du préconditionneur. Malgré un nombre d'itérations à convergence très faible lorsque  $\mu > 0.9$ , cela ne suffit pas à compenser le temps très important de la factorisation du préconditionneur qui reste de l'ordre de grandeur de la résolution directe du problème.

Lorsque  $\mu < 0.9$ , le *PCG* ne parvient pas à converger en raison de la perte du caractère positif du préconditionneur qui ne peut être utilisé avec un *CG*. C'est pourquoi, un solveur comme généralisation de la méthode de minimisation du résidu, ou *Generalized Minimal Residual (GMRes)*, permettant l'utilisation d'un préconditionneur non défini positif a permis de résoudre le système à l'aide d'un préconditionneur dont la troncature est plus prononcée (voir figure 2.17). On peut donc, en prenant un  $\mu$  proche de 0, arriver à une convergence en moins d'itérations et plus rapidement grâce à la factorisation rapide du préconditionneur. Ainsi, le préconditionnement du solveur GMRes avec l'inverse de l'opérateur tronqué fortement ( $\mu < 0.1$ ) permet une diminution du temps de résolution du problème par rapport au temps de résolution utilisant un *CG* non-préconditionné.

### 2.3.3 Interpolation RBF avec adaptation hiérarchique de rayon

Une seconde idée pour réduire le temps de calcul de l'interpolation par *RBF* est de réduire la taille du système en choisissant un sous-ensemble de nœuds à interpoler. Des algorithmes ont été proposés par [RENDALL et ALLEN \[2009\]](#) ou encore [NIU et coll. \[2017\]](#) pour permettre de sélectionner ces points de manière intelligente. Cependant, comme l'interpolation n'est plus exacte sur les nœuds du bord, cela ne permet pas de conserver la conformité à l'interface entre les sous-domaines.

Une autre idée consiste à ajouter des points de manière « multi-échelle » en réalisant des interpolations de maillages successives avec un ensemble de points de taille croissante et un rayon d'influence  $r$  variable dépendant de différents critères (erreur maximale commise, nombre de nœuds ajoutés,...). Cela permet de capturer les mouvements à grande échelle lorsqu'un faible nombre de nœuds est considéré. Lorsque le nombre de nœuds est faible, un rayon de grande taille est possible tout en permettant une résolution rapide du système (2.1). Puis lorsque de nouveaux nœuds sont ajoutés, les mouvements de plus faible « longueur d'onde » sont mieux représentés et, étant plus localisés, un plus faible rayon peut être utilisé pour les interpoler. Dans l'article de [WANG et coll. \[2018\]](#), un algorithme de ce type a été développé. Il permet de réaliser la déformation *RBF* en sélectionnant un ensemble initial comportant un faible nombre de nœuds et en ajoutant un seul nœud à chaque itération  $k$ . Accompagné de cet enrichissement de l'ensemble des nœuds,

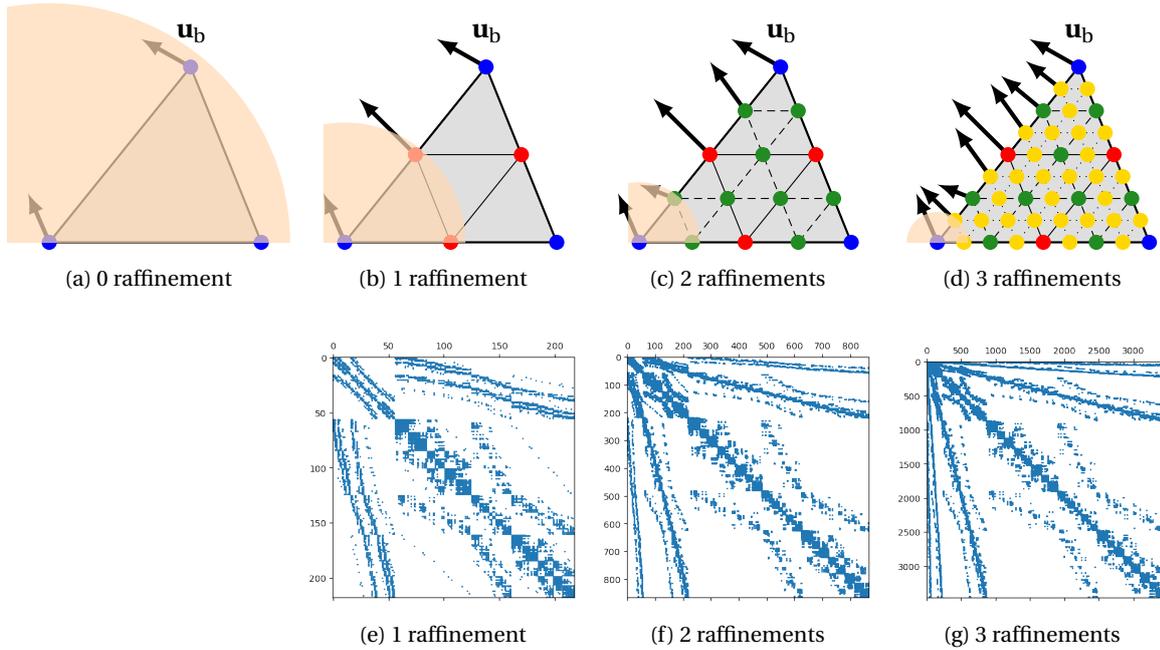


FIGURE 2.18 – Illustration de la méthode *RBF* hiérarchique. Ligne supérieure (a-d) : différents niveaux de raffinements avec le rayon d'influence du nœud en bas à gauche illustré par une portion de disque de couleur orange. Les différents niveaux de raffinement sont représentés par différentes couleurs de nœuds. Ligne inférieure (e-g) : Remplissage des matrices *RBF*  $\Psi^{(k)}$  du problème de la figure 2.7 pour différents niveaux de raffinements avec le rayon adapté.

une méthode d'adaptation du rayon a été définie à chaque itération selon la loi :

$$r^{(k)} = r_0 \cdot 10^{\alpha + \beta \lg \left( \frac{u_{b\max}^{(k)}}{u_{b\max}^{(0)}} \right)}. \quad (2.2)$$

$\alpha$  et  $\beta$  sont deux paramètres influençant la décroissance de  $r^{(k)}$  et dépendant de l'erreur maximale  $u_{b\max}^{(k)}$  commise par l'interpolation sur les nouveaux nœuds ajoutés à l'itération  $k$ . Dans le cas des maillages hiérarchiques imbriqués, à la différence de l'enrichissement choisi par [WANG et coll. \[2018\]](#) et [KEDWARD et coll. \[2017\]](#) qui est fondé sur l'erreur maximale d'interpolation, on propose de réaliser les différentes étapes de l'enrichissement *RBF* de manière hiérarchique : l'ajout de nœuds est réalisé de manière successive entre les différents niveaux de raffinement.

De plus, comme la distance entre les nœuds voisins est divisée par deux entre chaque niveau de raffinement hiérarchique, on propose naturellement un rayon adapté de la forme :

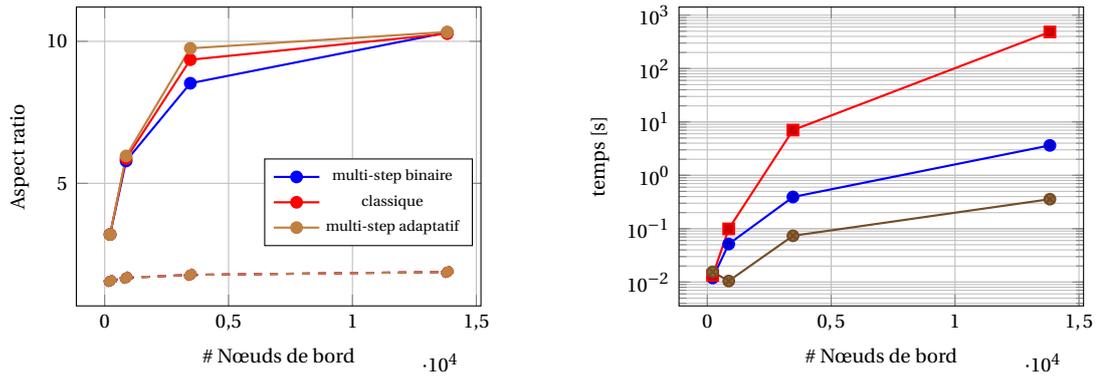
$$r^{(k)} = r_0 \cdot \frac{1}{2^{k-1}}, \quad (2.3)$$

dont l'évolution est représentée sur la figure 2.18.

Cette évolution du rayon sera appelée évolution « binaire ». L'avantage d'un tel choix de variation du rayon est que la largeur de bande reste la même à chaque étape hiérarchique comme on peut le voir sur la la ligne inférieure de la figure 2.18. Une idée similaire, qui permet de capturer les déformations de manière multi-échelle a été développée par [FLOATER et ISKE \[1996\]](#).

La figure 2.19 montre que le fait de choisir un rayon variable de la forme (2.3) permet d'obtenir des éléments de qualité comparable à celle obtenue avec un rayon fixe mais dans un temps bien inférieur (environ un facteur 130 pour la matrice la plus grande de la figure 2.19b). Cependant, en utilisant un rayon de la forme (2.2) et en choisissant  $\alpha = 0$  et  $\beta = 1.2$ , on arrive aussi à obtenir des éléments de qualité similaire à celle des deux autres méthodes en un temps encore plus faible.

Si un rayon de la forme (2.2) est sélectionné, il faut pouvoir déterminer les coefficients  $\alpha$  et  $\beta$  de décroissance de ce rayon pour réaliser l'interpolation de manière optimale (rapidement avec une bonne qualité d'éléments). Une étude paramétrique est réalisée en Annexe A.2, sur la figure A.1. En



(a) Pire qualité des éléments pour un nombre croissant de nœuds de bord (en traits pleins) et qualité moyenne (en pointillés)

(b) Temps nécessaire pour résoudre le système *RBF* par un solveur direct pour un nombre croissant de nœuds de bord

FIGURE 2.19 – Influence du nombre de nœuds de bord sur l'efficacité des *RBF* pour un rayon d'influence variable avec différentes méthodes

augmentant la valeur de  $\beta$ , le temps de calcul augmente alors que la qualité du maillage diminue. Par contre, lorsque  $\alpha$  augmente le temps de calcul augmente faiblement alors que la qualité des éléments s'améliore.

Finalement prendre  $\alpha$  et  $\beta$  grands permettrait de générer un maillage déformé de bonne qualité en un temps très faible d'après la figure A.2 et un bon choix pour ces paramètres peut être  $\alpha = 0.5$  et  $\beta = 2$ ; cela minimise le temps de calcul en ayant une qualité de maillage semblable à la qualité avec un rayon fixe et très grand.

Le choix de ces paramètres semble dépendre fortement du cas ciblé. C'est pourquoi la méthode binaire a été choisie dans les applications qui seront développées dans la section suivante, car plus systématique.

## 2.4 Résultats numériques

Dans cette section, les performances parallèles de la méthodologie proposée sont évaluées à la fois en termes d'*extensibilité* de la phase de génération de maillage mais aussi en termes de vitesse de convergence d'un solveur FETI. Pour ce dernier, la décomposition à deux niveaux est comparée à des décompositions classiquement fondées sur l'analyse du graphe de connectivité des éléments comme décrits dans la section 1.2.1, ou encore, des décompositions intuitives dans le cas de géométries triviales.

La méthode de sous-structuration à deux niveaux et de maillage en parallèle a été implémentée dans un contexte parallèle à mémoire distribuée en langage Python, en utilisant la suite logicielle *EF Z-set* commercialisée par *TRANSVALOR* [2019] et développée initialement par *BESSON et FOERCH* [1997] et *FEYEL* [2002]. Le code *Z-set* est utilisé pour son solveur *EF* et ses outils de manipulation de maillage, les communications entre les processus sont gérés par la bibliothèque standard *MPI*.

Cette section est découpée en trois parties, correspondant à trois exemples différents traitant d'abord de l'influence de la décomposition sur l'efficacité du solveur *FETI*, puis de l'évaluation des performances de la méthodologie développée précédemment sur un exemple homogène puis sur un exemple hétérogène.

Tous les résultats ont été obtenus sur un *cluster* composé de nœuds de  $2 \times 12$  cœurs dont les processeurs sont des Intel Xeon E5-2680 v3 cadencés à 2,50 GHz et connectés entre eux par un réseau InfiniBand Mellanox. Un seul cœur est associé à chaque sous-domaine afin de réaliser un parallélisme en *MPI* uniquement. Pour tous les calculs *EF*, un solveur *FETI* est utilisé avec le préconditionneur de Dirichlet le plus avancé décrit dans l'équation (1.19) comme cela est préconisé par

GOSSELET et REY [2006]. L'opérateur de *scaling* choisi est le *scaling* « rigidité »  $\mathbf{M}^\diamond = \text{diag}(\mathbf{K}_{bb})^\diamond$ . L'opérateur  $\mathbf{Q}_d$  du projecteur naturel (1.22) est choisi comme étant le préconditionneur de Dirichlet. Cette configuration devrait être la plus robuste possible sur des cas hétérogènes et en l'absence des techniques plus avancées qui ont été évoquées dans la section 1.1.3.4. Le critère de convergence  $\epsilon$  du solveur *FETI* est fixé à  $10^{-6}$ .

### 2.4.1 Influence de la décomposition

Premièrement, l'influence des hétérogénéités et de leur position vis-à-vis des interfaces entre les sous-domaines est discutée par rapport au taux de convergence d'un solveur *FETI*. Le problème étudié est un composite stratifié composé de huit plis linéaires élastiques et isotropes (voir figure 2.20a). Le module de Young alterne entre une valeur haute  $E_1$  et une plus faible  $E_2$ . Tous les plis partagent le même coefficient de Poisson  $\nu = 0.3$ . Plusieurs rapports d'hétérogénéités  $\rho_E = E_1/E_2$ , compris entre 1 et  $10^5$ , sont utilisés pour analyser les variations de la vitesse de convergence du solveur. Une des faces latérales est encastrée et un déplacement de traction et de flexion est imposé sur la face opposée.

Pour chaque valeur d'hétérogénéité  $\rho_E$ , une étude d'*extensibilité* faible est réalisée. Pour ce faire, un motif initial composé de 16 sous-domaines est répliqué dans les deux directions principales de la plaque. La taille des problèmes locaux reste ainsi constante entre les différents calculs (*i.e.* 32 768 éléments par sous-domaine quelle que soit la décomposition) alors que la taille du problème global augmente. Trois types de décompositions sont étudiés :

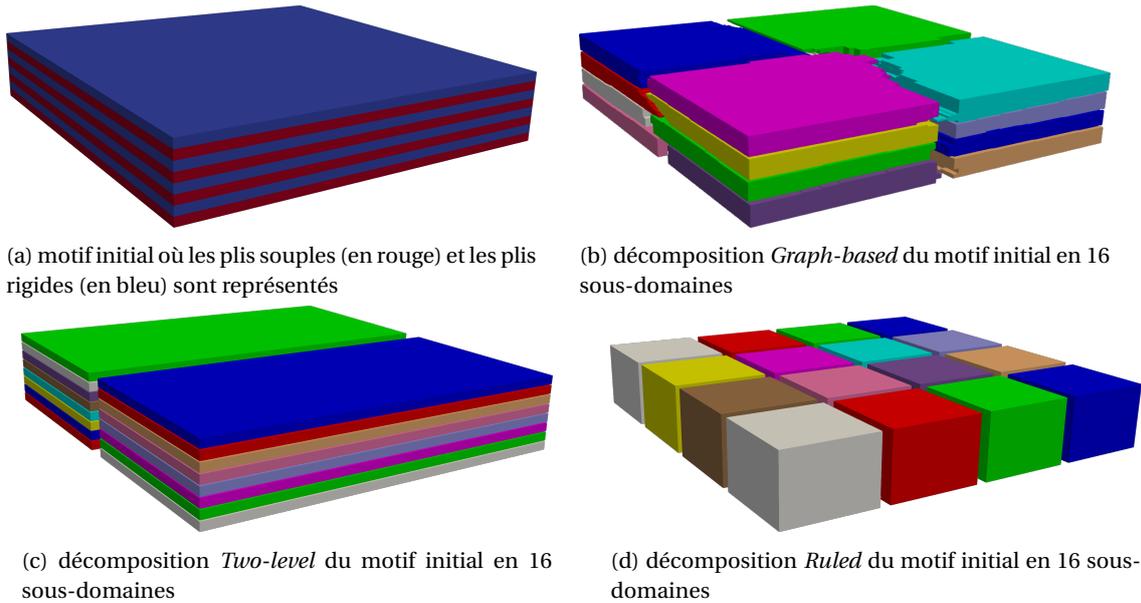


FIGURE 2.20 – Motif initial du composite stratifié à 545 025 nœuds et 524 288 éléments et décompositions selon les trois types étudiés.

- la décomposition dite *Graph-based* directement issue du partitionneur de graphe Metis, où les sous-domaines sont hétérogènes et comportent des interfaces irrégulières ;
- la décomposition *Two-level*, à deux niveaux, qui est le sujet de ce chapitre, où aucune étape de déformation n'est requise dans ce cas simple. Elle conduit à des sous-domaines homogènes avec une forme régulière qui, cependant, peuvent être très élancés ;
- la décomposition *Ruled*, qui comporte des sous-domaines de forme parallépipédique traversant complètement la structure, produisant ainsi des sous domaines réguliers mais hétérogènes.

Le lecteur notera que le choix de la décomposition ne conduit pas à des variations extrêmes de nombre de *DDLs* d'interface. Par exemple, le plus petit cas conduit à une interface constituée

de 100 000 DDLs pour la décomposition *Graph-based*, 112 000 pour la décomposition à deux niveaux et 147 000 pour la décomposition *Ruled*. Par ailleurs, la décomposition *Ruled* est utilisée ici uniquement dans un but de comparaison, mais cela est difficile, voire impossible de généraliser ce type de décomposition à des géométries réalistes.

Les résultats de convergence du solveur *FETI* sur ces différentes décompositions et pour différents rapports de modules de Young  $\rho_E$  sont présentés sur la figure 2.21. Les sous-figures (a–e) se rapportent à la géométrie à 8 plis, de 16 à 400 sous-domaines. Tout d’abord, la décomposition *Ruled* conduit à une convergence plus rapide que la décomposition *Graph-based*. Ces deux décompositions se comportent de la même façon lorsque le rapport d’hétérogénéité varie. La convergence du solveur *FETI* est assez peu sensible aux variations de faible rapport ( $\rho_E < 10^2$ ), mais elle est vraiment ralentie au delà de cette limite, avec une augmentation significative du nombre d’itérations à convergence. Ces phénomènes sont amplifiés par l’augmentation du nombre de sous-domaines. Au contraire, la décomposition à deux niveaux est assez insensible à l’accroissement du rapport d’hétérogénéité alors que ses performances se dégradent faiblement quand le rapport d’hétérogénéité tend vers 1 (comportement homogène). Ce comportement peut être attribué à l’élancement des sous-domaines qui pénalise la convergence globale. Cette caractéristique géométrique devient moins critique que le saut de coefficient matériau à l’interface pour des rapports importants. Cela explique les meilleures performances de la décomposition à deux niveaux dans le cas considéré. Ces observations sont confirmées par la figure 2.22, où le conditionnement  $\kappa$  de l’opérateur *FETI* est estimé par la valeur maximale de Ritz (le spectre étant minoré par 1). Il faut aussi noter que la comparaison entre les décompositions *Two-level* et *Graph-based* est discutable dans le cas purement homogène ( $\rho_E = 1$ ) car il n’y a plus de plis physiques dans ces problèmes. Par conséquent, en se passant de la contrainte supplémentaire due aux plis composites, la décomposition à deux niveaux générerait un découpage similaire à celui de la décomposition *Ruled* et permettrait ainsi une convergence en un nombre d’itérations bien inférieur à la décomposition *Graph-based*.

Dans le cas à 8 plis des figures 2.21(a–e), la décomposition *Ruled* est meilleure que les deux autres. En ajoutant simplement un pli, il est possible de dégrader fortement ses performances, comme illustré par la figure 2.21(f) pour le cas à 81 sous-domaines. En effet, dans ce cas, la décomposition *Ruled* (9 × 9 sous-domaines « dans le plan ») contient plus de « mauvais motifs d’hétérogénéité » ne respectant pas la contrainte de quasi-monotonie de PECHSTEIN et SCHEICHL [2013] (3 « canaux 3D rigides » dans un sous-domaine) qui sont connus pour pénaliser la convergence d’après la théorie *GenEO* de SPILLANE et RIXEN [2013]. Ainsi, la décomposition à deux niveaux devrait avoir de meilleures performances que les deux autres lorsque le nombre de plis augmente. Cependant, la configuration à 8 plis reste intéressante car elle montre clairement que l’approche à deux niveaux n’est pas appropriée à toutes les situations, surtout lorsque la structure considérée comporte de faibles hétérogénéités.

Cette étude a mis en évidence que, d’un point de vue de la convergence, la décomposition à deux niveaux n’a que peu d’intérêt dans le cas où la structure est quasi-homogène. Elle devient cependant intéressante en présence d’un rapport d’hétérogénéité suffisamment fort ( $\rho_E > 10^3$ ) et de nombreuses hétérogénéités. En comparant les décompositions *Ruled* et *Graph-based*, les interfaces régulières peuvent améliorer significativement la convergence (sauf lorsque le saut de coefficients matériau devient trop important).

#### 2.4.2 Exemple homogène : Aube de turbine simplifiée

La section précédente a montré le bon comportement de la méthode quand l’hétérogénéité devient importante en comparaison de la décomposition *Graph-based*, malgré des sous-domaines très élancés. Comme la précédente géométrie était constituée de surface planes, aucune déformation du maillage n’a dû être nécessaire à la génération du maillage. Ainsi, l’efficacité parallèle de cette génération n’a pas pu être démontrée. Les performances de la génération de maillage vont donc être démontrées dans cette section par une étude d’extensibilité forte dans le cas d’une aube de turbine d’un moteur d’avion constituée d’un matériau homogène dont la géométrie a été sim-

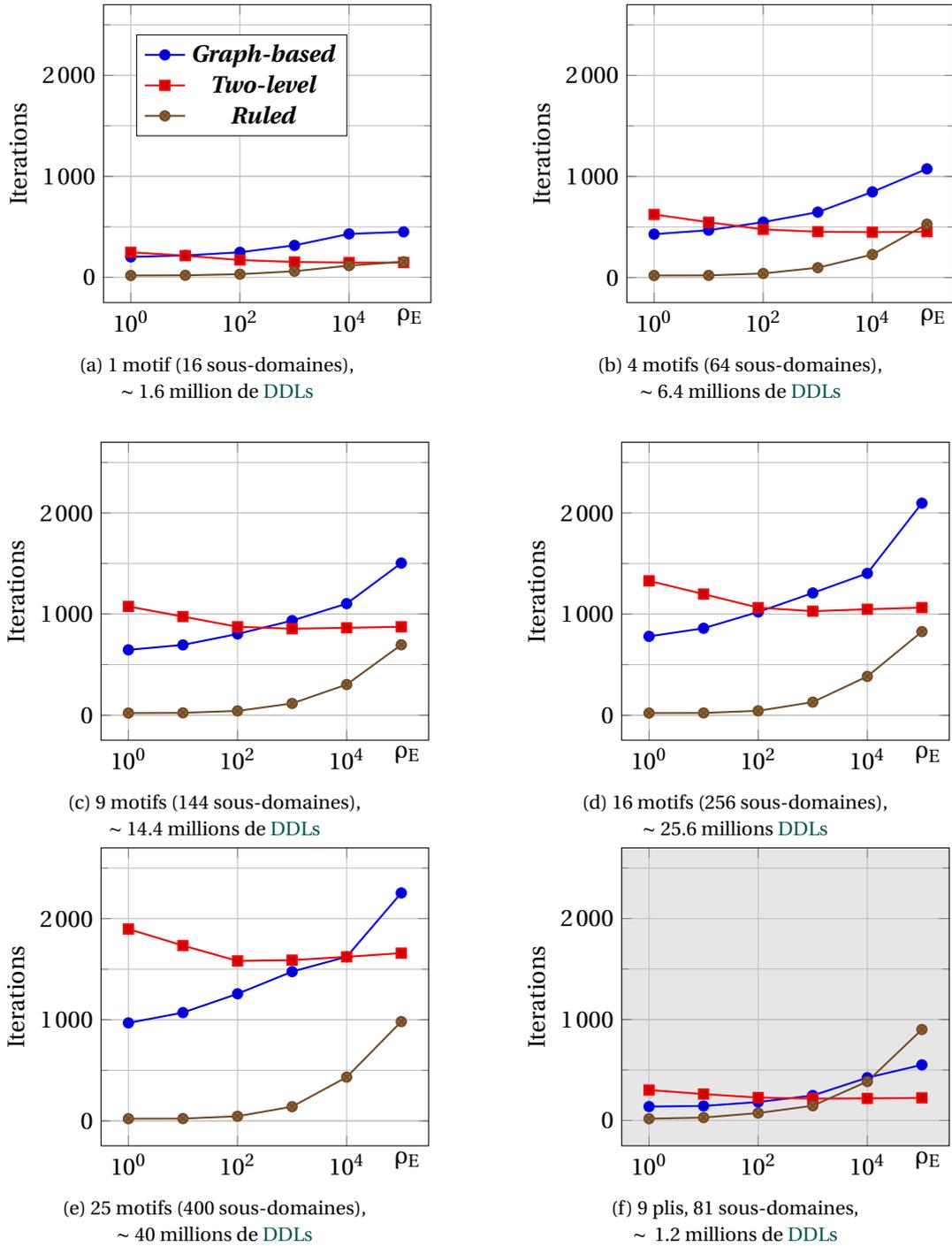
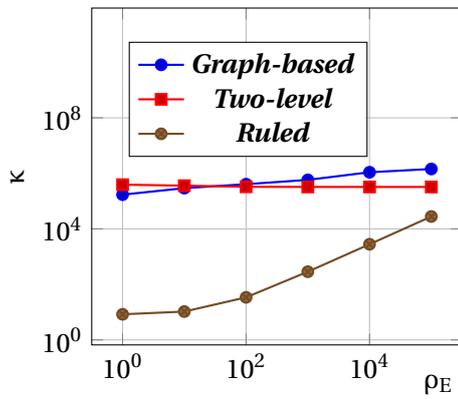
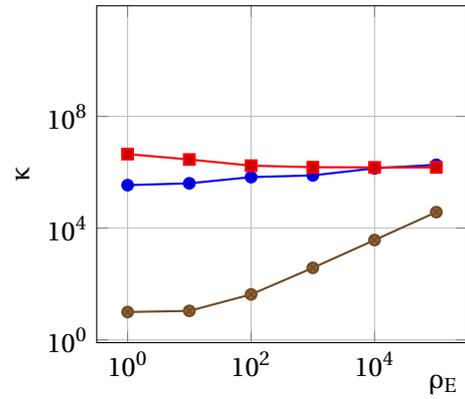


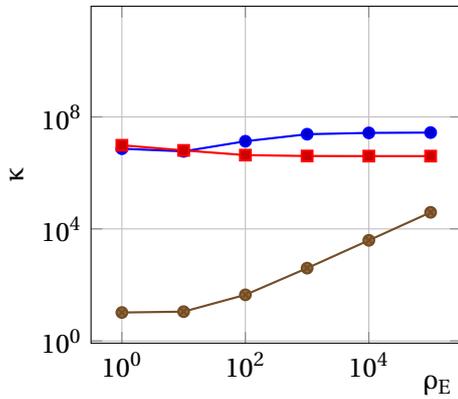
FIGURE 2.21 – Structure en composite stratifié à 8 plis (a–e) et 9 plis (f) : influence du ratio d’hétérogénéité  $\rho_E$  sur la convergence du solveur *FETI* pour les trois types de décomposition et un nombre croissant de sous-domaines (le fond gris de la figure (f) permet de distinguer le cas à 9 plis des cas à 8 plis).



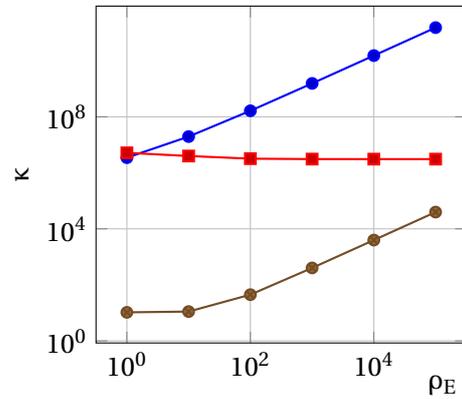
(a) 1 motif (16 sous-domaines),  
~ 1.6 million de DDLs



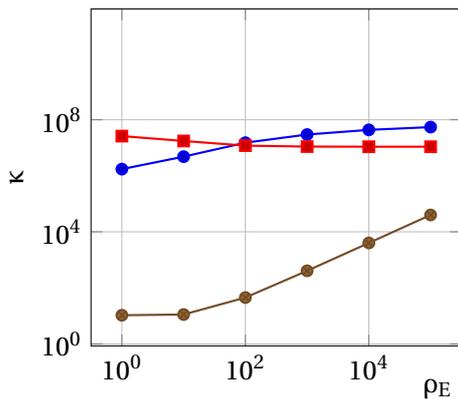
(b) 4 motifs (64 sous-domaines),  
~ 6.4 millions de DDLs



(c) 9 motifs (144 sous-domaines),  
~ 14.4 millions de DDLs



(d) 16 motifs (256 sous-domaines),  
~ 25.6 millions de DDLs



(e) 25 motifs (400 sous-domaines),  
~ 40 millions de DDLs

FIGURE 2.22 – Structure en composite stratifié à 8 plis : influence du ratio d'hétérogénéité  $\rho_E$  sur le conditionnement  $\kappa$  de l'opérateur *FETI* préconditionné pour les trois types de décomposition et un nombre de sous-domaines qui augmente.

plifiée évitant d'obtenir un maillage grossier initial ( $\mathcal{T}_c, \mathcal{F}_c, \mathcal{E}_c$ ) avec un trop grand nombre d'éléments (voir figure 2.23).

### 2.4.2.1 Description du problème

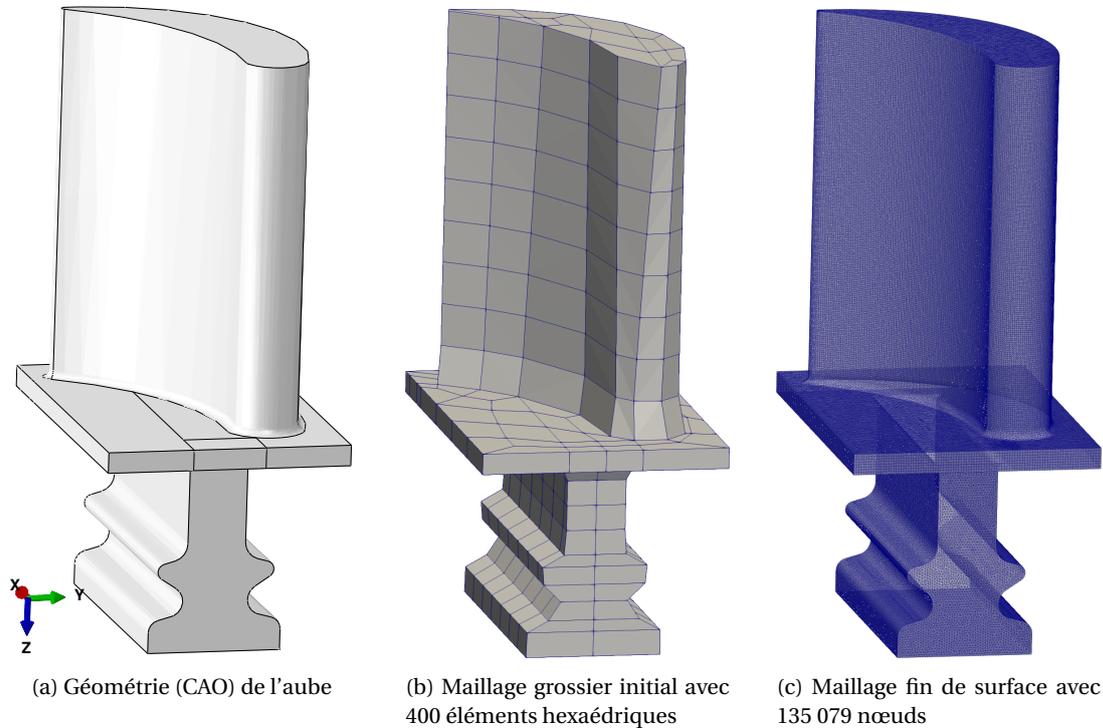


FIGURE 2.23 – Données initiales de la méthode

Le maillage grossier est représenté sur la figure 2.23b. Les maillages fins associés aux sous-domaines sont obtenus après quatre niveaux de raffinement hiérarchique, conduisant à approximativement 1.7 million de nœuds (5.1 millions de DDLs). Pour la décomposition à deux niveaux, le maillage grossier est divisé en 10, 25, 50, 100, 200 et 400 sous-domaines de calcul. Pour l'approche *Graph-based*, le maillage fin a été décomposé en autant de sous-domaines. Certaines des décompositions obtenues sont illustrées sur les figures 2.24 et 2.25. Notons que le maillage de calcul monolithique est identique pour une configuration donnée entre les deux décompositions, ainsi, seule la décomposition (la forme des sous-domaine) varie entre les méthodes.

### 2.4.2.2 Évaluation de la méthode de déformation de maillage hiérarchique

Avant de réaliser cette étude d'extensibilité forte, l'intérêt d'utiliser la méthode de déformation de maillage par *RBF* avec adaptation de rayon est illustrée au travers de l'exemple à 400 sous-domaines représenté sur la figure 2.24c. Dans ce cas, certains sous-domaines subissent une déformation car ils ont au moins une arête en commun avec le bord de la structure globale alors que d'autres sous-domaines ne sont pas du tout déformés. Les temps de calcul de la phase de déformation de maillage sont mesurés pour la méthode classique de déformation par *RBF* décrite à la section 1.3.2.2 et la méthode d'adaptation de rayon hiérarchique (qu'on appellera *RBF* hiérarchique) décrite dans la section 2.3.3 et exploitant un rayon de la forme (2.3). Pour l'approche classique, un rayon d'influence  $r$  égal à  $5 \max(\|\mathbf{u}_b\|)$  a été choisi tandis que pour l'approche hiérarchique seul  $r_0$  prend cette valeur. Sur ce cas, des accélérations pouvant aller jusqu'à une réduction de 98% du temps de calcul initial peuvent être observées sur la figure 2.26. Dans cette étude, la qualité des éléments hexaédriques n'a pas été calculée mais elle devrait être similaire pour les deux méthodes, comme cela a pu être illustré dans la section 2.3.3 pour des tétraèdres.

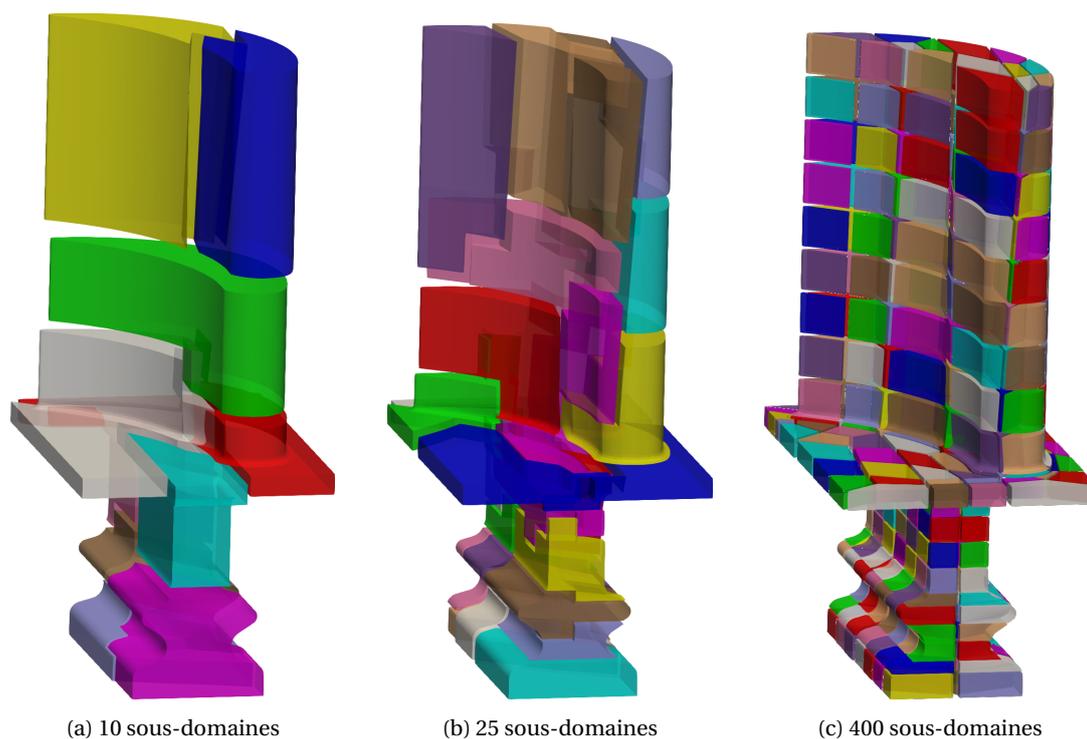


FIGURE 2.24 – Exemples de décompositions *Two-level* de l'aube de turbine

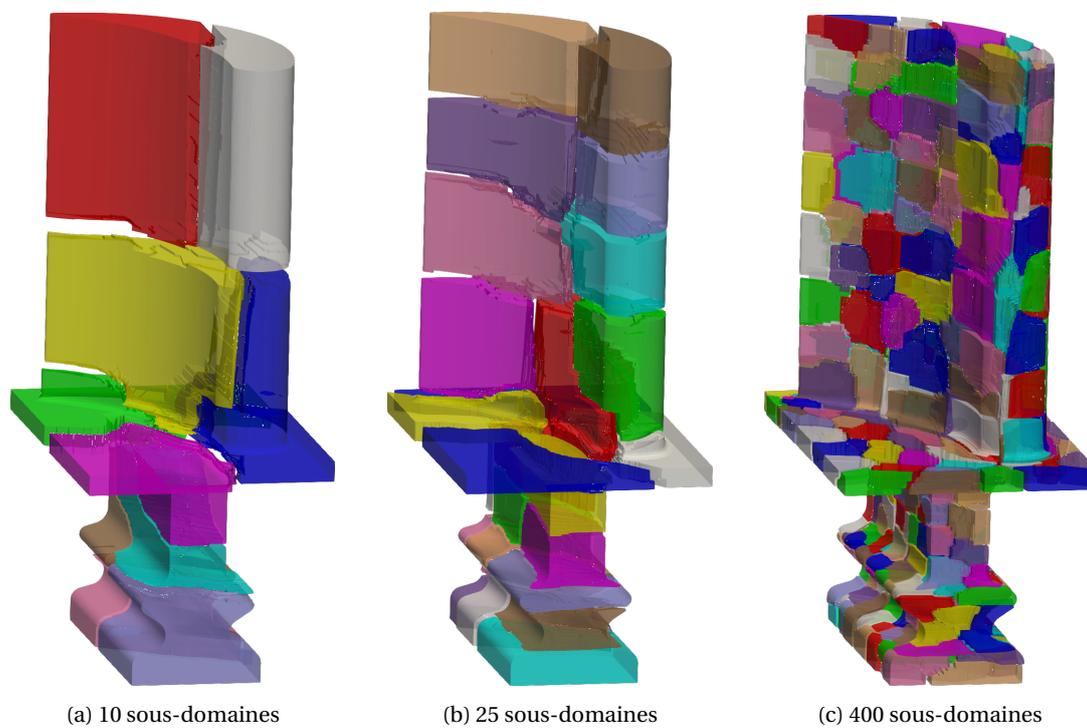


FIGURE 2.25 – Exemples de décompositions *Graph-based* de l'aube de turbine

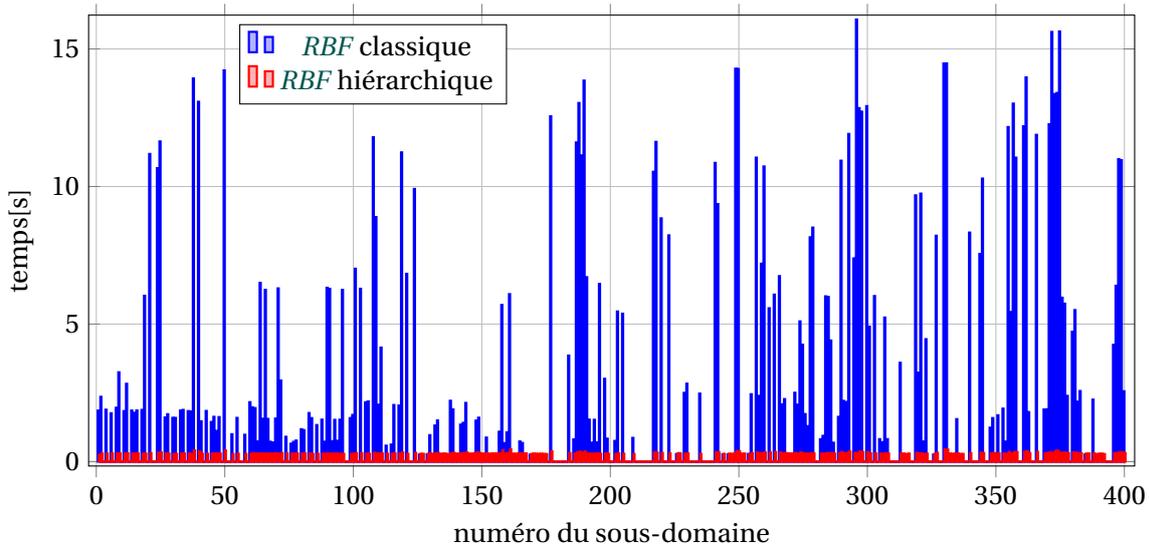


FIGURE 2.26 – Temps de réalisation de la phase de déformation de maillage volumique dans chaque sous-domaine pour différentes méthodes *RBF*

### 2.4.2.3 Extensibilité forte

Les deux stratégies sont évaluées à la fois en terme de traitement parallèle du maillage mais aussi en terme de performances du solveur.

Les temps nécessaires à la réalisation de l’algorithme 2.1 sont décrits pour chaque décomposition sur la figure 2.27 et le tableau 2.1. Le code développé est instrumenté de six mesures de temps (les numéros de lignes font référence à l’algorithme 2.1) :

- *Seq.* est le temps passé dans les étapes séquentielles (les lignes 1, 2 et 3) ;
- *Cutting* est le temps passé dans l’étape de restriction de la géométrie fine (ligne 4) ;
- *Refin.* et *Update* représentent, respectivement, les temps passés dans les étapes de raffinement réel du maillage et de la mise à jour des données d’interface (leur somme est le temps de la ligne 5) ;
- *Surface* représente le temps correspondant à la déformation du maillage de surface sur les bords des sous-domaines (lignes 6 à 10) ;
- *Volume* représente la durée de l’interpolation volumique du maillage par la méthode par *RBF*.

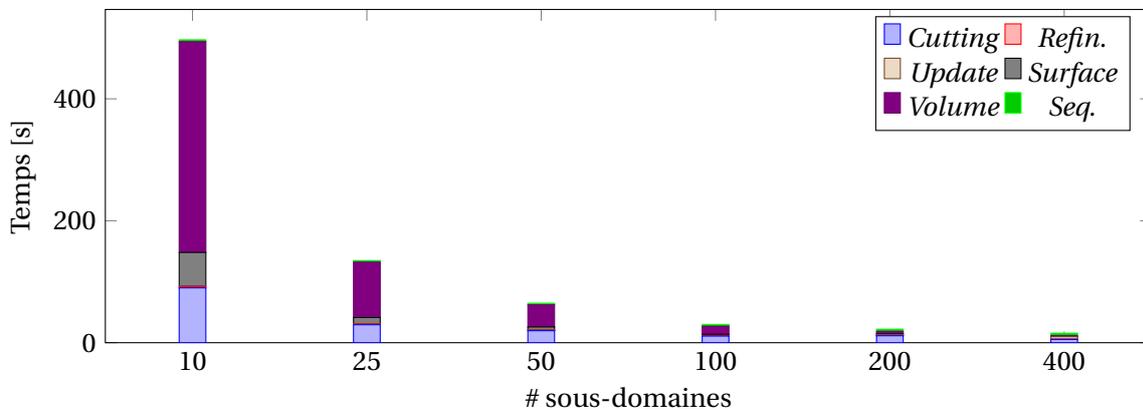


FIGURE 2.27 – Temps du processus de maillage en fonction du nombre de sous-domaines pour un problème global de taille constante (approximativement 1.7 million de nœud)

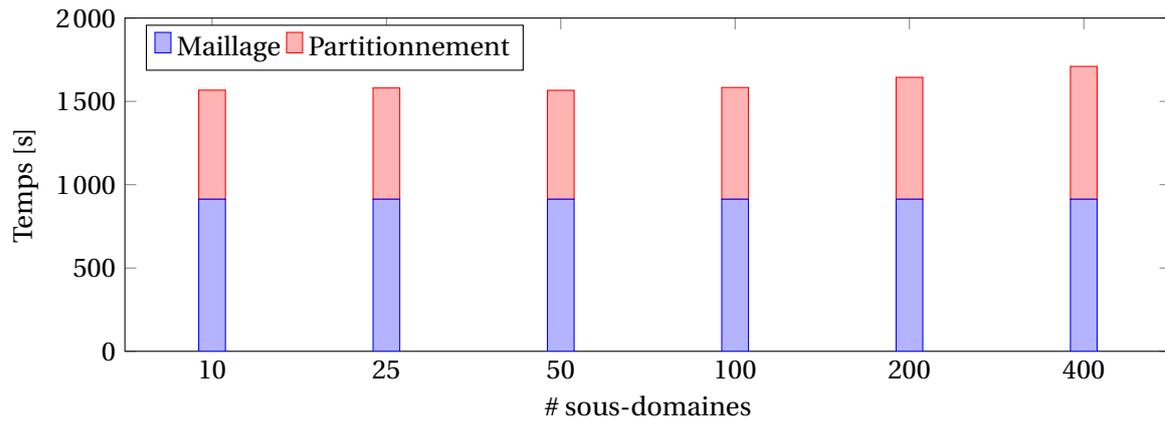


FIGURE 2.28 – Temps de la génération de maillage et du partitionnement en séquentiel de l’aube avec approximativement 1.6 million de nœuds

$N_s$	Temps [s]							
	<i>Full Seq.</i>	Total	<i>Seq.</i>	<i>Cutting</i>	<i>Refin.</i>	<i>Update</i>	<i>Surface</i>	<i>Volume</i>
10	1 567,5	496,86	0,61%	18,1%	0,44%	0,19%	11,06%	69,59%
25	1 580,52	134,8	1,94%	21,83%	0,81%	0,42%	7,55%	67,45%
50	1 566,53	64,81	4,05%	30,33%	1,08%	1,14%	7,23%	56,17%
100	1 583,47	30,02	9,7%	36,25%	1,33%	3,19%	5,2%	44,34%
200	1 644,28	22,21	15,73%	51,42%	1,92%	8,44%	3,65%	18,84%
400	1 709,77	15,69	26,94%	33,4%	4,48%	24,41%	1,84%	8,94%

TABLEAU 2.1 – Temps du processus de maillage de l’aube pour un nombre croissant  $N_s$  de sous-domaines. La première colonne « *Full Seq.* » est le temps de génération de maillage et de partitionnement par un processus classique à partir de la géométrie initiale

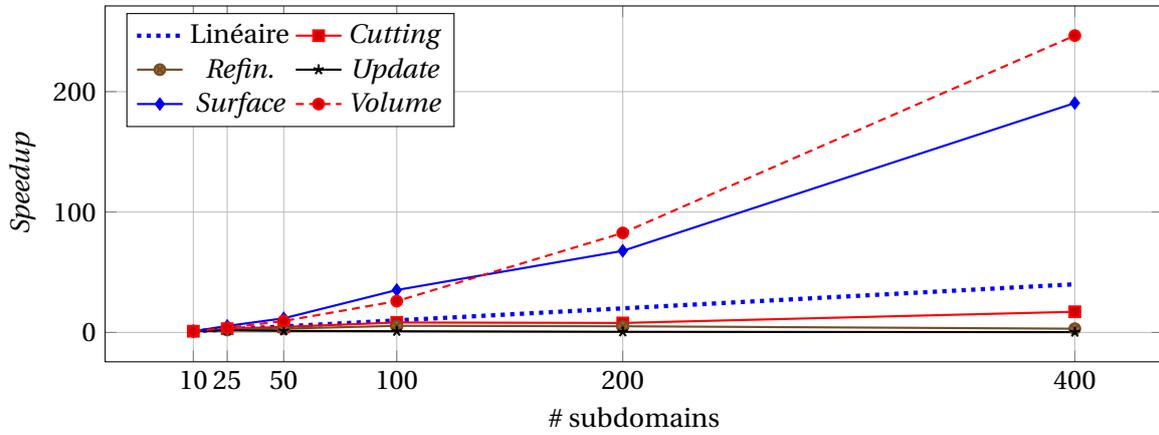
Les figures 2.29a et 2.29b illustrent le facteur d’accélération (*speedup*) pour les différentes mesures de temps. Ce *speedup* est calculé par rapport à la configuration à 10 sous-domaines selon la formule suivante :

$$speedup(N_s \text{ sd}) = \frac{\text{Temps}(10 \text{ sd})}{\text{Temps}(N_s \text{ sd})}$$

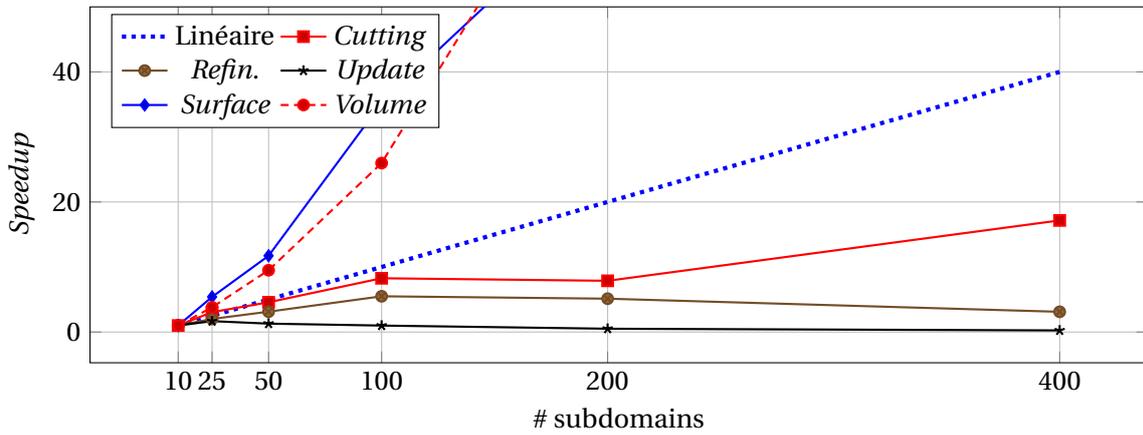
D’abord, le temps total diminue rapidement lorsque le nombre de sous-domaines augmente. Cependant, le temps passé dans les parties séquentielles du processus augmente toujours avec le nombre de sous-domaines comme le montre la dernière ligne du tableau 2.1, si bien qu’il en occupe une grande partie. Cela a pour conséquence de mener à une saturation de la courbe de *speedup* Total sur la figure 2.29c. Cette saturation peut être expliquée par les étapes de partitionnement des lignes 2 et 3 de l’algorithme 2.1, où une union de tous les éléments grossiers est réalisée pour générer les données d’interface. Les mesures de temps *Cutting* et *Surface*, qui représentent une part importante du temps total, ont des *speedups* presque linéaires contrairement à *Refin.* et *Update* qui mettent en jeu des communications *MPI* entre tous les voisins (Neighbor-All-to-All). Les *speedups* *Volume* et *Surface* sont super-linéaires, car un problème creux est résolu avec une complexité au moins quadratique.

Ensuite, le temps total du processus de génération de maillage séquentiel (maillage et découpage) est comparé entre l’approche à deux niveaux et une approche qualifiée de « classique », où le maillage fin est généré directement à partir de la définition CAO puis découpé. Pour les mesures de temps de l’approche classique, Abaqus 6.13-1 de SIMULIA [2013] est utilisé en tant que mailleur, et Z-set interfacé avec Scotch (PELLEGRINI et ROMAN [1996]) en tant que partitionneur. Les temps sont représentés sur la figure 2.28 et la tableau 2.1, où une très grande amélioration du temps de génération de maillage et de partitionnement peut être observée pour l’approche parallèle, même lorsque le nombre de sous-domaines est faible.

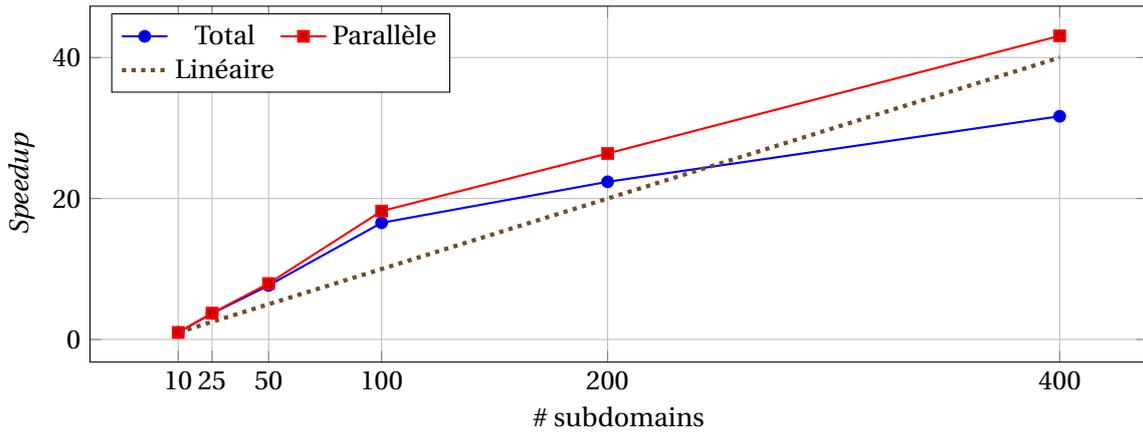
Finalement, pour étudier l’efficacité de la résolution *FETI*, des calculs *EF* sont menés sur les



(a) Speedups de toutes les mesures de temps individuels



(b) zoom sur la figure 2.29a



(c) Speedup des temps cumulés (Parallèle={Cutting, Refin., Update, Surface, Volume}, Total={Parallèle, Seq.})

FIGURE 2.29 – Speedup du processus de génération de maillage parallèle pour l'exemple de l'aube

différentes configurations générées. Le problème mécanique est linéaire élastique et isotrope dans les hypothèses des petites déformations, avec un module de Young de 200 GPa, un coefficient de Poisson de 0.3, un coefficient de dilatation thermique de  $12 \times 10^{-6} \text{ K}^{-1}$  et une masse volumique de  $8000 \text{ kg/m}^3$ . La pièce subit un effort centrifuge correspondant à une vitesse de rotation de  $3 \times 10^6 \text{ rad/s}$  autour d'un axe parallèle à l'axe Z (voir figure 2.23a) et placé à 100 mm du pied d'aube. Une déformation thermique homogène est générée par l'application d'un champ de température de 980 K ainsi qu'un effort de pression sur l'intrados de l'aube. Les déplacements de la face inférieure sont bloqués le long de l'axe Z et une arête sur cette face est encastree, ce qui permet d'éliminer tous les mouvements de corps rigide.

La figure 2.30 montre que la convergence du solveur *FETI* sur une décomposition résultant de la méthode mise en avant dans ce chapitre est meilleure qu'une décomposition *Graph-based* lorsque les sous-domaines sont composés d'un ou deux éléments grossiers (donc lorsque  $N_s$  est grand) ou quand la structure est décomposée en un nombre faible de sous-domaines. En effet, lorsque le nombre de sous-domaines est faible ou important, la régularité des interfaces permet une meilleure convergence pour la décomposition à deux niveaux. Cependant, dans le cas intermédiaire ( $10 < N_s < 200$ ), les sous-domaines sont trop élancés pour permettre une meilleure convergence que la méthode *Graph-based*.

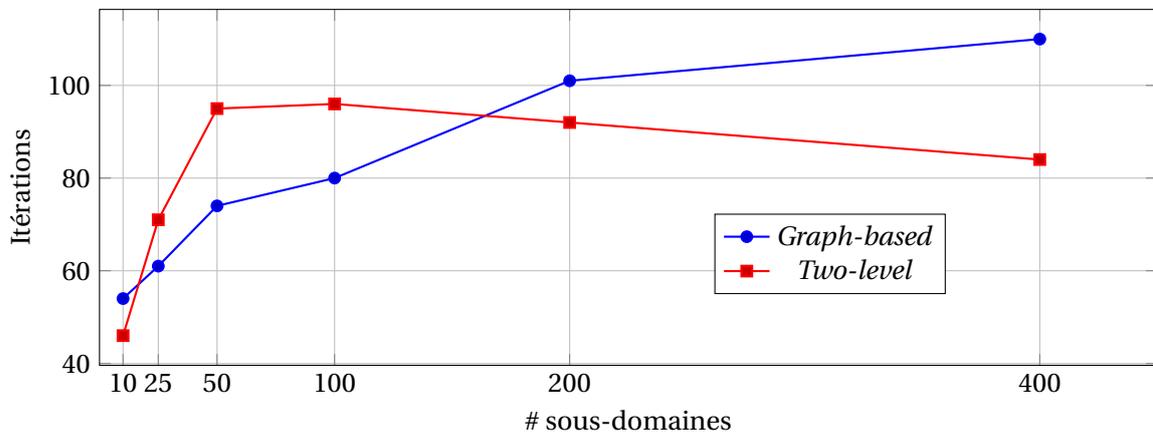


FIGURE 2.30 – Nombre d'itérations de la méthode *FETI* pour une convergence relative de  $10^{-6}$  pour le cas de l'aube de turbine et pour un nombre variable de sous-domaines

### 2.4.3 Exemple hétérogène : volume simplifié de propergol solide

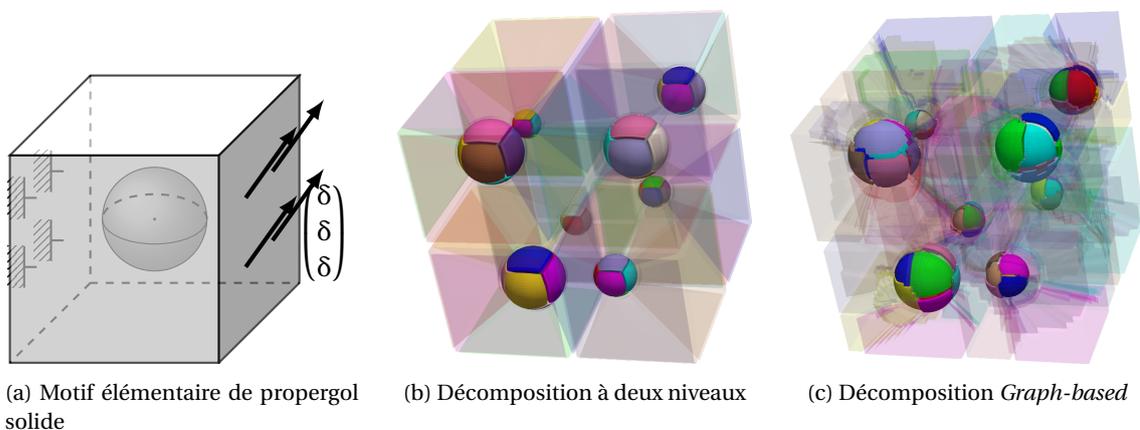


FIGURE 2.31 – Volume de propergol solide contenant  $2 \times 2 \times 2$  inclusions, les différentes couleurs représentent les différentes couleurs; les éléments transparents représentent la matrice et les autres sont dans les inclusions

Les deux sections précédentes ont montré d’abord un exemple hétérogène sans déformation de maillage puis une application homogène avec déformation de maillage. La méthode sera donc évaluée dans cette section sur un exemple hétérogène nécessitant une déformation du maillage.

### 2.4.3.1 Description du problème

L’exemple considéré est un modèle de propergol solide (voir figure 2.31) utilisé pour la propulsion de fusées. Ce modèle est obtenu par réplication du motif de la figure 2.31a où la position et le rayon des inclusions sont distribués aléatoirement selon une distribution uniforme. De plus, le rayon des inclusions est compris entre 25% et 75% de la taille du cube répliqué. Deux matériaux linéaires élastiques et isotropes composent la structure. Les inclusions sont composées d’un matériau dont le module de Young est  $E_1 = 69\,000\text{ MPa}$  et le coefficient de Poisson  $\nu_1 = 0.346$ . La matrice est quasi-incompressible avec un coefficient de Poisson  $\nu_2 = 0.499$  et un module de Young plus souple  $E_2 = 0,96\text{ MPa}$ , ce qui conduit à un rapport d’hétérogénéité  $\rho_E = E_1/E_2 = 7.2 \cdot 10^5$ . La structure cubique obtenue est encadrée sur une face et un déplacement dans les trois directions est imposé sur la face opposée, comme cela est schématisé sur la figure 2.31a. Comme la matrice est quasi-incompressible, des éléments mixtes à trois champs, décrits par ZIENKIEWICZ et coll. [2013], sont utilisés (les champs additionnels étant la pression et la variation volumique qui sont constants par éléments).

### 2.4.3.2 Extensibilité faible

Une étude d’extensibilité faible est réalisée. Le motif initial de la figure 2.31a est divisé en 13 sous-domaines et répliqué dans les trois directions de l’espace pour obtenir un ensemble de structures globales possédant un nombre total de  $13 \times N_c^3$  sous-domaines,  $N_c$  étant le nombre de réplifications dans une direction. La figure 2.32 et le tableau 2.2 montrent le nombre d’itérations à

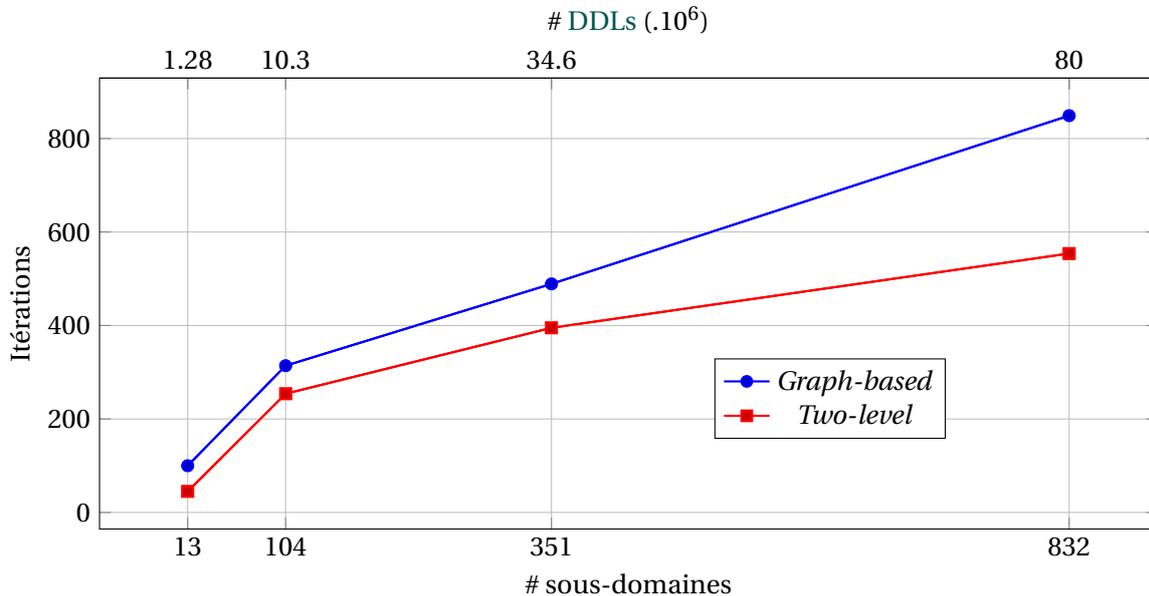


FIGURE 2.32 – Convergence de l’exemple du propergol solide pour une taille constante de problèmes locaux

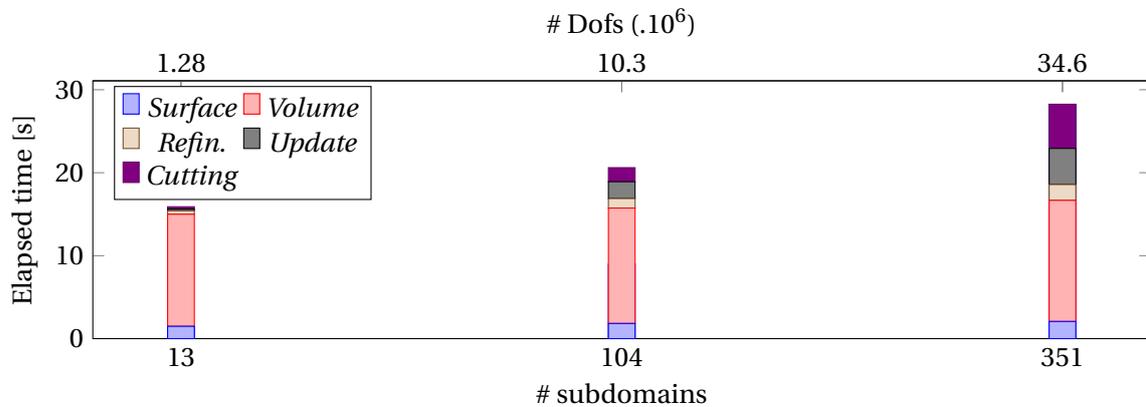
convergence du solveur *FETI* pour la méthode à deux niveaux et la méthode automatique *Graph-based* pour des valeurs de  $N_c$  comprises entre 1 et 4. La décomposition à deux niveaux converge plus rapidement que la décomposition *Graph-based*. Un gain de l’ordre de 20% à 55% en termes d’itérations peut être obtenu en fonction des cas par rapport à la décomposition *Graph-based*, ce gain augmente continûment avec la taille du problème.

Le temps du processus de génération de maillage est tracé sur la figure 2.33 et détaillé dans le tableau 2.3 pour un nombre  $N_c$  compris entre 1 et 3. Les résultats pour  $N_c = 4$  ont été ob-

tenu par sur-allocation des processus *MPI* sur les unités de calcul si bien que les temps mesurés, non pertinents, ne sont pas montrés sur les figures. Les mesures de temps *Surface* et *Volume* sont associés à la partie liée à la déformation de maillage. Elles représentent une partie significative du temps de calcul mais mettent en jeu des problèmes locaux de taille constante, dont les temps restent constants. Les temps mesurés *Refin.* et *Update* augmentent légèrement avec  $N_c$ . Cette augmentation peut être expliquée par le fait que des échanges *MPI* de données sont impliqués dans ces étapes. En effet, la latence *MPI* devrait augmenter à partir de  $N_c = 1$  (chaque processus *MPI* est exécuté sur un même nœud) jusqu'à  $N_c = 3$  (les 351 processus *MPI* sont distribués entre 15 nœuds, interconnectés par au moins deux niveaux de réseau Infiniband). Cette latence réseau est significative si on considère les ordres de grandeur des temps totaux que l'on voit dans le tableau 2.3. De plus, l'interface *mpi4py* pour le *MPI* peut aussi causer cette latence lorsque les échanges doivent être considérés entre les nœuds de calcul. Finalement, l'augmentation du compteur *Cutting* est due à l'augmentation de la taille du maillage 2D de la peau car la géométrie du problème total devient plus grande, rendant l'étape de découpage de ce maillage de peau (étape 4 de l'algorithme 2.1) plus onéreuse.

$N_s$	Itérations	
	<i>Graph-based</i>	<i>Two-level</i>
13	100	45
104	314	254
351	489	395
832	849	554

TABLEAU 2.2 – Nombre d'itérations pour atteindre la convergence pour l'exemple du propergol solide


 FIGURE 2.33 – Temps de génération de maillage du propergol solide en fonction du nombre de sous-domaines pour un problème local de taille constante (*extensibilité* faible).

$N_s$	Temps [s]					
	Total	<i>Cutting</i>	<i>Refin.</i>	<i>Update</i>	<i>Surface</i>	<i>Volume</i>
13	15,9	1,73%	2,66%	1,31%	9,36%	84,94%
104	20,61	8,22%	5,55%	9,86%	8,87%	67,5%
351	28,25	18,83%	6,74%	15,43%	7,32%	51,68%

TABLEAU 2.3 – Temps détaillés du processus de génération de maillage pour l'exemple du propergol solide pour une taille constante de problème local

### 2.4.3.3 Extensibilité forte

Une étude d'*extensibilité* forte de la génération de maillage seule a aussi été réalisée pour le cas  $N_c = 3$  avec approximativement 34.6 millions de DDLs. Les temps de génération des maillages sont illustrés sur la figure 2.34 et le tableau 2.4. Malheureusement, il n'est pas possible d'équilibrer de manière optimale la charge entre les sous-domaines pour les configurations à 108 et 54 sous-domaines. En effet, dans le cas à 108 sous-domaines, ils sont constitués de 3 ou 4 éléments grossiers. Dans le cas à 54 sous-domaines, chaque inclusion est un sous-domaine constitué de 7 éléments grossiers quand les sous-domaines dans la matrice n'en comportent que 6.

Les compteurs *Cutting* et *Surface* diminuent pour les mêmes raisons que le cas de l'aube de turbine. Cependant, entre les cas à 108 et 54 sous-domaines, les autres compteurs (*i.e.* *Refin.*, *Update* et *Volume*) restent à peu près constants car les sous-domaines qui demandent le plus d'opérations sont composés par le même nombre de faces grossières (celles dans l'ensemble  $\mathcal{F}_r^{(s)}$ ). C'est ce nombre de faces qui est déterminant quant à la rapidité de ces processus. À titre d'exemple, pour le compteur *Volume*, cela conduit à un opérateur *RBF*  $\Psi$  de la même taille entre les cas à 54 et 108 sous-domaines en raison du fait qu'ils possèdent tous les deux un nombre maximal de faces par sous-domaine égal. Ainsi leurs temps de calcul sont similaires.

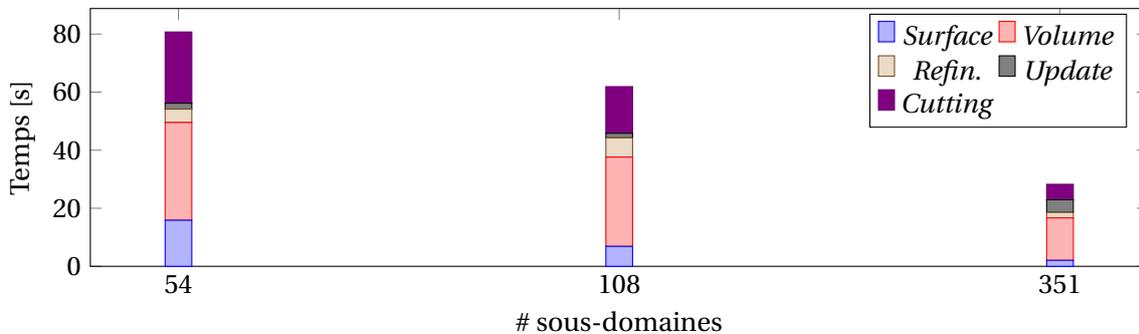


FIGURE 2.34 – Temps de la génération de maillage pour le propergol solide pour une taille constante du problème global (*extensibilité* forte, 34.6 millions de DDLs)

$N_s$	Temps [s]					
	Total	<i>Cutting</i>	<i>Refin.</i>	<i>Update</i>	<i>Surface</i>	<i>Volume</i>
54	80,71	30,4%	5,72%	2,53%	19,67%	41,69%
108	61,89	25,97%	10,72%	2,53%	11,12%	49,66%
351	28,25	18,83%	6,74%	15,43%	7,32%	51,68%

TABEAU 2.4 – Temps détaillés de la génération de maillage pour le propergol solide pour une taille constante du problème global (*extensibilité* forte, 34.6 millions de DDLs)

## 2.5 Conclusion partielle

Dans cette partie, une nouvelle approche de sous-structuration parallèle à deux niveaux pour les méthodes de décomposition de domaine sans recouvrement a été présentée. Son principe est, schématiquement, de permuter les étapes de découpage et de génération du maillage de calcul qui sont réalisées dans les approches classiques de préparation des données. Pour cela, le découpage en sous-domaines est réalisé au tout début du processus complet, en utilisant un maillage grossier qui tient compte de la distribution des différents matériaux. Ensuite seulement, un maillage conforme plus fin est généré en parallèle sur chacun des sous-domaines, puis une déformation du maillage final est réalisée pour assurer une représentation fidèle et précise de la géométrie d'entrée. Cette méthode produit des sous-domaines qui sont homogènes et qui possèdent une forme

régulière. Ces caractéristiques de sous-domaines sont particulièrement bien adaptées aux méthodes de décomposition de domaine telles que *FETI* ou *BDD* comme cela a pu être illustré dans la section 1.1.3.3. En effet, elles réduisent le **conditionnement** du système à résoudre, conduisant à une convergence plus rapide du solveur comparée à une approche classique de partitionnement fondée sur le découpage d'un graphe de connectivités. Aussi, une partie importante de la génération de maillage peut alors être réalisée en parallèle avec très peu d'échanges entre sous-domaines voisins, ce qui ajoute à l'intérêt de la méthode en contexte parallèle.

Les expériences numériques ont confirmé que, sur une partie des cas étudiés, la sous-structuration proposée améliore le taux de convergence des résolutions *FETI* sur des exemples aussi bien homogènes que fortement hétérogènes en comparaison d'une décomposition automatique. Cependant, dans quelques situations, la décomposition à deux niveaux peut conduire à une moins bonne convergence que l'approche classique. Ces situations correspondent à des configurations où, soit les sous-domaines sont composés de l'agrégation de trop d'éléments grossiers (illustré dans la section 2.4.2) conduisant à des formes géométriques trop irrégulières, soit les sous-domaines ont un élancement trop important (section 2.4.1) qui n'est pas compensé par un traitement optimal de la forte hétérogénéité. Concernant la génération de maillage en parallèle, la méthode démontre une bonne **extensibilité** forte et une **extensibilité** faible prometteuse qui gagnerait à être testée sur des configurations à grand nombre de processus (plusieurs milliers).

Ce travail ouvre principalement deux perspectives : la première est de partitionner directement la géométrie lorsqu'elle est disponible, par exemple dans un format *CAO*, s'inspirant d'approches décrites dans le paragraphe 1.2.2.3. Cela permet d'éviter d'avoir à discrétiser finement le maillage de peau et même d'éviter d'avoir recours à des méthodes de déformation de maillage. La seconde est d'explorer des méthodes alternatives de raffinement, comme suggéré dans la section 2.1.3, permettant d'obtenir des maillages de meilleure qualité et de s'affranchir de la qualité du maillage grossier initial. Cependant, le caractère hiérarchique du maillage est perdu, si bien que la méthode de déformation de maillage par *RBF* avec adaptation de maillage hiérarchique décrite dans la section 2.4.2.2 n'est plus accessible. Une stratégie d'enrichissement de l'ensemble des nœuds sélectionnés adaptée aux maillages non hiérarchiques doit être développée pour pouvoir utiliser de telles méthodes de raffinement local efficacement.

## 2.6 Références

- 3D SYSTEMS, INC. 1988, *StereoLithography Interface Specification*, 3D Systems, Inc. 80
- BESSON, J. et R. FOERCH. 1997, «Large scale object-oriented finite element code design», *Computer Methods in Applied Mechanics and Engineering*, vol. 142, n° 1-2, doi : [10.1016/S0045-7825\(96\)01124-3](https://doi.org/10.1016/S0045-7825(96)01124-3), p. 165–187, ISSN 00457825. URL <http://linkinghub.elsevier.com/retrieve/pii/S0045782596011243>. 95
- BOVET, C., O. CIOBANU, A. PARRET-FREAUD et V. CHIARUTTINI. 2019, «A parallel mesh refinement process tailored to domain decomposition methods», dans *Proceedings of the Sixth International Conference on Parallel, Distributed, GPU and Cloud Computing for Engineering*, Polack Mihály Faculty of Engineering and Information Technology, University of Pécs, Hungary, p. 20, doi : [10.4203/ccp.112.20](https://doi.org/10.4203/ccp.112.20). URL <http://www.ctresources.info/ccp/paper.html?id=9287>. 84
- BUHMANN, M. D. 2000, «Radial basis functions», *Acta Numerica*, vol. 9, doi : [10.1017/S0962492900000015](https://doi.org/10.1017/S0962492900000015), p. 1–38, ISSN 0962-4929, 1474-0508. URL [https://www.cambridge.org/core/product/identifier/S0962492900000015/type/journal\\_article](https://www.cambridge.org/core/product/identifier/S0962492900000015/type/journal_article). 87
- DE BOER, A., M. S. VAN DER SCHOOT et H. BIJL. 2007, «Mesh deformation based on radial basis function interpolation», *Computers & Structures*, vol. 85, n° 11-14, doi : [10.1016/j.compstruc.2007.01.013](https://doi.org/10.1016/j.compstruc.2007.01.013), p. 784–795, ISSN 00457949. URL <http://linkinghub.elsevier.com/retrieve/pii/S0045794907000223>. 86

- DEMMELE, J. W., S. C. EISENSTAT, J. R. GILBERT, X. S. LI et J. W. H. LIU. 1999, «A Supernodal Approach to Sparse Partial Pivoting», *SIAM Journal on Matrix Analysis and Applications*, vol. 20, n° 3, doi : [10.1137/S0895479895291765](https://doi.org/10.1137/S0895479895291765), p. 720–755, ISSN 0895-4798. URL <https://epubs.siam.org/doi/abs/10.1137/S0895479895291765>, publisher: Society for Industrial and Applied Mathematics. 91
- DRYJA, M. 2005, «A Neumann-Neumann algorithm for a mortar discretization of elliptic problems with discontinuous coefficients», *Numer. Math.*, vol. 99, n° 4, doi : [10.1007/s00211-004-0573-2](https://doi.org/10.1007/s00211-004-0573-2), p. 645–656, ISSN 0945-3245. URL <https://doi.org/10.1007/s00211-004-0573-2>. 84
- DRYJA, M. et W. PROSKUROWSKI. 2005, «A FETI-DP Method for the Mortar Discretization of Elliptic Problems with Discontinuous Coefficients», dans *Domain Decomposition Methods in Science and Engineering*, vol. 40, édité par T. J. Barth, M. Griebel, D. E. Keyes, R. M. Nieminen, D. Roose, T. Schlick, R. Kornhuber, R. Hoppe, J. Périaux, O. Pironneau, O. Widlund et J. Xu, Springer-Verlag, Berlin/Heidelberg, ISBN 978-3-540-22523-2, p. 345–352, doi : [10.1007/3-540-26825-1\\_34](https://doi.org/10.1007/3-540-26825-1_34). URL [http://link.springer.com/10.1007/3-540-26825-1\\_34](http://link.springer.com/10.1007/3-540-26825-1_34). 84
- EL GHARBI, Y., P. GOSSELET, A. PARRET-FRÉAUD et C. BOVET. 2019a, «Parallel preprocessing with subdomain's shape control for domain decomposition methods», dans *The Sixth International Conference on Parallel, Distributed, GPU and Cloud Computing for Engineering*, Pollack Mihály Faculty of Engineering and Information Technology, University of Pécs, Hungary, p. 21, doi : [10.4203/ccp.112.21](https://doi.org/10.4203/ccp.112.21). URL <http://www.ctresources.info/ccp/paper.html?id=9288>. 77
- EL GHARBI, Y., P. GOSSELET, A. PARRET-FRÉAUD et C. BOVET. 2019b, «Stratégie hiérarchique de sous-structuration et maillage pour la simulation HPC», dans *14ème colloque en calcul des structures*, Giens, France, p. 8. URL <https://hal.archives-ouvertes.fr/hal-02147196>. 77
- EL GHARBI, Y., P. GOSSELET, A. PARRET-FRÉAUD et C. BOVET. 2020, «A Two-level Substructuring method tailored to Schur based Domain Decomposition Methods», dans *26th International Domain Decomposition Conference*, Hong-Kong, China. URL <https://hal.archives-ouvertes.fr/hal-03054225>. 77
- EL GHARBI, Y., P. GOSSELET, A. PARRET-FRÉAUD et C. BOVET. 2021a, «Hierarchical substructuring and parallel mesh generation of heterogeneous structures for domain decomposition methods», dans *14th World Congress in Computational Mechanics and ECCOMAS Congress*, Paris, France. URL <https://hal.archives-ouvertes.fr/hal-03054211>. 77
- EL GHARBI, Y., A. PARRET-FRÉAUD, C. BOVET et P. GOSSELET. 2021b, «Two-level substructuring and parallel mesh generation for domain decomposition methods», *Finite Elements in Analysis and Design*, doi : [10.1016/j.finel.2020.103484](https://doi.org/10.1016/j.finel.2020.103484). URL <https://hal.archives-ouvertes.fr/hal-02881249>, publisher: Elsevier. 77
- FEYEL, F. 2002, «Some new technics regarding the parallelisation of ZéBuLoN, an object oriented finite element code for structural mechanics», *ESAIM: M2AN*, vol. 36, n° 5, doi : [10.1051/m2an:2002040](https://doi.org/10.1051/m2an:2002040), p. 923–935. URL <https://doi.org/10.1051/m2an:2002040>. 95
- FLOATER, M. S. et A. ISKE. 1996, «Multistep scattered data interpolation using compactly supported radial basis functions», *Journal of Computational and Applied Mathematics*, vol. 73, n° 1, doi : [10.1016/0377-0427\(96\)00035-0](https://doi.org/10.1016/0377-0427(96)00035-0), p. 65–78, ISSN 0377-0427. URL <http://www.sciencedirect.com/science/article/pii/0377042796000350>. 94
- GOSSELET, P. et C. REY. 2006, «Non-overlapping domain decomposition methods in structural mechanics», *Archives of Computational Methods in Engineering*, vol. 13, n° 4, doi : [10.1007/BF02905857](https://doi.org/10.1007/BF02905857), p. 515–572, ISSN 1134-3060, 1886-1784. URL <http://link.springer.com/10.1007/BF02905857>. 96

- HAVE, P., R. MASSON, F. NATAF, M. SZYDLARSKI, H. XIANG et T. ZHAO. 2013, «Algebraic Domain Decomposition Methods for Highly Heterogeneous Problems», *SIAM Journal on Scientific Computing*, vol. 35, n° 3, p. C284–C302. URL <https://hal.archives-ouvertes.fr/hal-00611997>, publisher: Society for Industrial and Applied Mathematics. 79
- HENDRICKSON, B. 2011, «Chaco», dans *Encyclopedia of Parallel Computing*, édité par D. Padua, Springer US, Boston, MA, ISBN 978-0-387-09766-4, p. 248–249, doi : 10.1007/978-0-387-09766-4\_310. URL [http://dx.doi.org/10.1007/978-0-387-09766-4\\_310](http://dx.doi.org/10.1007/978-0-387-09766-4_310). 82
- ISO/TC 184/SC 4 INDUSTRIAL DATA. 2016, «Industrial automation systems and integration – Product data representation and exchange – Part 21: Implementation methods: Clear text encoding of the exchange structure», ISO 10303-21, International Organization for Standardization, Geneva, CH. 80
- ITO, Y., A. M. SHIH, A. K. ERUKALA, B. K. SONI, A. CHERNIKOV, N. P. CHRISOCHOIDES et K. NAKAHASHI. 2007, «Parallel unstructured mesh generation by an advancing front method», *Mathematics and Computers in Simulation*, vol. 75, n° 5, doi : 10.1016/j.matcom.2006.12.008, p. 200–209, ISSN 0378-4754. URL <http://www.sciencedirect.com/science/article/pii/S0378475406003351>. 84
- JOVANOVIC, N., D. KEYES, K. GUM PRASAD et J. KANE. 1993, «Drop Tolerance ILU Preconditioners For Iterative Solution Techniques In Boundary Element Analysis», dans *Boundary Elements XV Vol 1 Fluid Flow and Computational Aspects*, vol. 1, p. 16, doi : 10.2495/BE930341. 91
- KARYPIS, G. et V. KUMAR. 1998, «A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs», *SIAM Journal on Scientific Computing*, vol. 20, n° 1, doi : 10.1137/S1064827595287997, p. 359–392, ISSN 1064-8275, 1095-7197. URL <http://epubs.siam.org/doi/10.1137/S1064827595287997>. 82
- KEDWARD, L., C. B. ALLEN et T. C. S. RENDALL. 2017, «Efficient and exact mesh deformation using multiscale RBF interpolation», *Journal of Computational Physics*, vol. 345, doi : 10.1016/j.jcp.2017.05.042, p. 732–751, ISSN 0021-9991. URL <http://www.sciencedirect.com/science/article/pii/S0021999117304254>. 94
- LE GOUEZ, J.-M. L. et J. M. ETANCELIN. 2015, «Numerical Properties and GPU Implementation of a High Order Finite Volume Scheme», dans *Turbulence and Interactions*, Springer, Cham, p. 177–184, doi : 10.1007/978-3-319-60387-2\_19. URL [https://link.springer.com/chapter/10.1007/978-3-319-60387-2\\_19](https://link.springer.com/chapter/10.1007/978-3-319-60387-2_19). 83
- MÄNTYLÄ, M. 1987, *An Introduction to Solid Modeling*, Computer Science Press, Inc., New York, NY, USA, ISBN 0-88175-108-1. 80
- MÖLLER, T. et B. TRUMBORE. 1997, «Fast, Minimum Storage Ray-Triangle Intersection», *Journal of Graphics Tools*, vol. 2, n° 1, doi : 10.1080/10867651.1997.10487468, p. 21–28, ISSN 1086-7651. URL <http://www.tandfonline.com/doi/abs/10.1080/10867651.1997.10487468>. 86
- NIU, J., J. LEI et J. HE. 2017, «Radial basis function mesh deformation based on dynamic control points», *Aerospace Science and Technology*, vol. 64, doi : 10.1016/j.ast.2017.01.022, p. 122–132, ISSN 1270-9638. URL <http://www.sciencedirect.com/science/article/pii/S1270963816309658>. 93
- PECHSTEIN, C. et R. SCHEICHL. 2013, «Weighted Poincaré inequalities», *IMA Journal of Numerical Analysis*, vol. 33, n° 2, doi : 10.1093/imanum/drs017, p. 652–686, ISSN 0272-4979. URL <https://academic.oup.com/imajna/article/33/2/652/653341>, publisher: Oxford Academic. 82, 97

- PELLEGRINI, F. et J. ROMAN. 1996, «Scotch: A software package for static mapping by dual recursive bipartitioning of process and architecture graphs», dans *High-Performance Computing and Networking*, vol. 1067, édité par G. Goos, J. Hartmanis, J. van Leeuwen, H. Liddell, A. Colbrook, B. Hertzberger et P. Sloot, Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN 978-3-540-61142-4 978-3-540-49955-8, p. 493–498, doi : [10.1007/3-540-61142-8\\_588](https://doi.org/10.1007/3-540-61142-8_588). URL [http://link.springer.com/10.1007/3-540-61142-8\\_588](http://link.springer.com/10.1007/3-540-61142-8_588). 82, 103
- RENDALL, T. C. S. et C. B. ALLEN. 2009, «Efficient mesh motion using radial basis functions with data reduction algorithms», *Journal of Computational Physics*, vol. 228, n° 17, doi : [10.1016/j.jcp.2009.05.013](https://doi.org/10.1016/j.jcp.2009.05.013), p. 6231–6249, ISSN 0021-9991. URL <http://www.sciencedirect.com/science/article/pii/S0021999109002721>. 93
- SELIM, M. et K. KOMULLIL. 2016, «Mesh Deformation Approaches A Survey», *Journal of Physical Mathematics*, vol. 7, n° 2, doi : [10.4172/2090-0902.1000181](https://doi.org/10.4172/2090-0902.1000181), ISSN 20900902. URL <https://www.omicsonline.com/open-access/mesh-deformation-approaches\T1\textendasha-survey-2090-0902-1000181.php?aid=76056>. 86
- SIMULIA. 2013, «Abaqus», URL <http://dsk.ippt.pan.pl/docs/abaqus/v6.13/books/usi/default.htm>. 103
- SPILLANE, N. et D. J. RIXEN. 2013, «Automatic spectral coarse spaces for robust finite element tearing and interconnecting and balanced domain decomposition algorithms», *International Journal for Numerical Methods in Engineering*, vol. 95, n° 11, doi : [10.1002/nme.4534](https://doi.org/10.1002/nme.4534), p. 953–990, ISSN 1097-0207. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.4534>. 97
- STEFANICA, D. 2001, «A Numerical Study of FETI Algorithms for Mortar Finite Element Methods», *SIAM Journal on Scientific Computing*, vol. 23, n° 4, doi : [10.1137/S1064827500378829](https://doi.org/10.1137/S1064827500378829), p. 1135–1160, ISSN 1064-8275, 1095-7197. URL <http://epubs.siam.org/doi/10.1137/S1064827500378829>. 84
- TRANSVALOR, S. A. 2019, «Z-set 9.0 user manual», URL <http://www.zset-software.com/>. 95
- WANG, G., X. CHEN et Z. LIU. 2018, «Mesh deformation on 3D complex configurations using multistep radial basis functions interpolation», *Chinese Journal of Aeronautics*, vol. 31, n° 4, doi : [10.1016/j.cja.2018.01.028](https://doi.org/10.1016/j.cja.2018.01.028), p. 660–671, ISSN 1000-9361. URL <http://www.sciencedirect.com/science/article/pii/S1000936118300669>. 93, 94
- YILMAZ, Y. et C. OZTURAN. 2015, «Using sequential NETGEN as a component for a parallel mesh generator», *Advances in Engineering Software*, vol. 84, n° C, doi : [10.1016/j.advengsoft.2014.12.013](https://doi.org/10.1016/j.advengsoft.2014.12.013), p. 3–12, ISSN 0965-9978. URL <https://doi.org/10.1016/j.advengsoft.2014.12.013>. 84
- ZIENKIEWICZ, O. C., R. L. TAYLOR et J. Z. ZHU. 2013, «Chapter 10 - Incompressible Problems, Mixed Methods, and Other Procedures of Solution», dans *The Finite Element Method: its Basis and Fundamentals (Seventh Edition)*, édité par O. C. Zienkiewicz, R. L. Taylor et J. Z. Zhu, Butterworth-Heinemann, Oxford, ISBN 978-1-85617-633-0, p. 315–359, doi : [10.1016/B978-1-85617-633-0.00010-1](https://doi.org/10.1016/B978-1-85617-633-0.00010-1). URL <http://www.sciencedirect.com/science/article/pii/B9781856176330000101>. 106

# CHAPITRE 3

## IMPÉDANCE D'INTERFACE À DEUX NIVEAUX

Le chapitre 1 a présenté des méthodes de décomposition de domaine mixtes dont l'un des paramètres est l'impédance d'interface  $\mathbf{Q}_b^\diamond$ . Cette impédance doit permettre de rendre compte de la réponse du reste de la structure pour permettre une convergence rapide de la méthode. Le chapitre 2 a permis de mettre en place une sous-structuration à deux niveaux où un maillage grossier global de faible résolution et un maillage fin entrent en jeu. La taille de ce maillage grossier étant négligeable devant la taille du maillage fin, il peut être distribué sur chacun des sous-domaines et est ainsi rendu accessible à l'ensemble des sous-domaines. L'idée de ce chapitre est ainsi d'exploiter ce maillage grossier pour obtenir une bonne estimation de  $\mathbf{Q}_b^\diamond$  permettant d'améliorer la convergence de ces solveurs en permettant une propagation d'informations lointaines aux sous-domaines.

*« Les détails font la perfection, et la perfection n'est pas un détail. »*

Léonard De Vinci

<b>3.1 Notion d'impédance d'interface optimale</b>	<b>114</b>
<b>3.2 Approximation de l'impédance optimale par une approche à deux niveaux</b>	<b>116</b>
3.2.1 Techniques d'évaluation de l'impédance d'interface	116
3.2.1.1 Évaluation et approximation des interactions courtes distances	117
3.2.1.2 Approximation des interactions longues distances	117
3.2.1.3 Approximation à deux échelles	118
3.2.2 Approximation à deux niveaux	119
3.2.2.1 Expression voulue	119
3.2.2.2 Couplage entre deux maillages imbriqués	120
3.2.2.3 Factorisation de l'opérateur local	124
3.2.3 Résultats numériques	125
<b>3.3 Formulations mixtes avancées</b>	<b>127</b>
3.3.1 FETI-2LM généralisée et introduction d'une seconde impédance d'interface	127
3.3.1.1 Mise en équation	127
3.3.1.2 Étude spectrale et estimation d'une bonne impédance	129
3.3.2 Minimisation des sauts à l'interface	131
3.3.3 Ajout d'une contrainte additionnelle	132
<b>3.4 Conclusion partielle</b>	<b>134</b>
<b>3.5 Références</b>	<b>134</b>

### 3.1 Notion d'impédance d'interface optimale

Dans ce chapitre sera considérée uniquement la seconde formulation du problème mixte (1.17). En effet, celle-ci semble plus efficace en raison de l'homogénéité de son opérateur, ce qui doit conduire à un **conditionnement** moindre. On rappelle sa formulation :

$$\left( \mathbf{A}^{\diamond T} \mathbf{A}^{\diamond} \mathbf{S}_p^{\diamond} + \mathbf{Q}_b^{\diamond} \mathbf{B}^{\diamond T} \mathbf{B}^{\diamond} \right) \mathbf{S}_m^{\diamond} \boldsymbol{\mu}_b^{\diamond} = \left( \mathbf{A}^{\diamond T} \mathbf{A}^{\diamond} \mathbf{Q}_b^{\diamond} - \mathbf{Q}_b^{\diamond} \mathbf{B}^{\diamond T} \mathbf{B}^{\diamond} \right) \mathbf{b}_m^{\diamond}, \quad (3.1)$$

et les expressions des quantités mixtes  $\mathbf{S}_m^{\diamond}$  et  $\mathbf{b}_m^{\diamond}$  sont :

$$\begin{aligned} \mathbf{S}_m^{\diamond} &= \mathbf{t}^{\diamond} \left( \mathbf{K}^{\diamond} + \mathbf{t}^{\diamond T} \mathbf{Q}_b^{\diamond} \mathbf{t}^{\diamond} \right)^{-1} \mathbf{t}^{\diamond T} = \left( \mathbf{S}_p^{\diamond} + \mathbf{Q}_b^{\diamond} \right)^{-1} \\ \mathbf{b}_m^{\diamond} &= \mathbf{t}^{\diamond} \left( \mathbf{K}^{\diamond} + \mathbf{t}^{\diamond T} \mathbf{Q}_b^{\diamond} \mathbf{t}^{\diamond} \right)^{-1} \mathbf{f}^{\diamond} = \mathbf{S}_m^{\diamond} \mathbf{b}_p^{\diamond}. \end{aligned}$$

Le paramètre  $\mathbf{Q}_b^{\diamond}$  introduit dans l'inconnue mixte (1.11) est homogène à une rigidité d'interface. C'est un paramètre de la méthode qu'il convient de choisir avec soin car il influe grandement sur la convergence du solveur itératif. Le résidu  $\mathbf{r}_j^{\diamond}$  correspond, à l'itération  $j$  et pour une solution  $\boldsymbol{\mu}_{b,j}^{\diamond}$ , à l'écart à la vérification de l'équation (3.1) et se traduit par :

$$\mathbf{r}_j^{\diamond} = \left( \mathbf{A}^{\diamond T} \mathbf{A}^{\diamond} \mathbf{Q}_b^{\diamond} - \mathbf{Q}_b^{\diamond} \mathbf{B}^{\diamond T} \mathbf{B}^{\diamond} \right) \mathbf{b}_m^{\diamond} - \left( \mathbf{A}^{\diamond T} \mathbf{A}^{\diamond} \mathbf{S}_p^{\diamond} + \mathbf{Q}_b^{\diamond} \mathbf{B}^{\diamond T} \mathbf{B}^{\diamond} \right) \mathbf{S}_m^{\diamond} \boldsymbol{\mu}_{b,j}^{\diamond}. \quad (3.2)$$

La norme de ce résidu est notée  $r_j = \|\mathbf{r}_j^{\diamond}\|_2$  et est la quantité suivie dans les graphes de convergence et sera appelée simplement « résidu ».

L'équilibre local condensé du sous-domaine  $\Omega^{(s)}$  dans le cas d'une inconnue mixte peut se met sous la forme :

$$\left( \mathbf{S}_p^{(s)} + \mathbf{Q}_b^{(s)} \right) \mathbf{u}_b^{(s)} = \mathbf{b}_p^{(s)} + \boldsymbol{\mu}_b^{(s)}. \quad (3.3)$$

De plus, l'équilibre du sous-domaine  $\Omega^{(s)}$  ainsi que celui de  $\Omega^{(\bar{s})} \equiv \Omega \setminus \Omega^{(s)}$  donnent :

$$\begin{cases} \mathbf{S}_p^{(s)} \mathbf{u}_b^{(s)} &= \mathbf{b}_p^{(s)} + \boldsymbol{\lambda}_b^{(s)} \\ \mathbf{S}_p^{(\bar{s})} \mathbf{u}_b^{(\bar{s})} &= \mathbf{b}_p^{(\bar{s})} + \boldsymbol{\lambda}_b^{(\bar{s})}. \end{cases}$$

Les conditions de raccords entre les sous-domaines  $\mathbf{B}^{\diamond} \mathbf{u}_b^{\diamond} = 0$  et  $\mathbf{A}^{\diamond} \boldsymbol{\lambda}_b^{\diamond} = 0$  s'écrivent dans le cas de ces deux sous-domaines :

$$\begin{cases} \boldsymbol{\lambda}_b^{(s)} + \boldsymbol{\lambda}_b^{(\bar{s})} &= 0 \\ \mathbf{u}_b^{(s)} - \mathbf{u}_b^{(\bar{s})} &= 0. \end{cases}$$

En sommant les deux équations locales d'équilibre et en utilisant les relations d'interface, l'équation d'équilibre globale suivante apparaît :

$$\left( \mathbf{S}_p^{(s)} + \mathbf{S}_p^{(\bar{s})} \right) \mathbf{u}_b^{(s)} = \mathbf{b}_p^{(s)} + \mathbf{b}_p^{(\bar{s})}. \quad (3.4)$$

L'équation d'équilibre locale mixte (3.3) devient équivalente à l'équation d'équilibre globale (3.4) en choisissant  $\mathbf{Q}_b^{(s)}$  comme étant le complément de Schur du sous-domaine complémentaire  $\Omega^{(\bar{s})}$  et en initialisant l'inconnue mixte par le second membre condensé du sous-domaine complémentaire. Concernant la matrice, ce choix s'écrit :

$$\mathbf{Q}_b^{(s)} = \mathbf{S}_p^{(\bar{s})},$$

où

$$\mathbf{S}_p^{(\bar{s})} = \mathbf{K}_{bb}^{(\bar{s})} - \mathbf{K}_{bi}^{(\bar{s})} \mathbf{K}_{ii}^{(\bar{s})}{}^{-1} \mathbf{K}_{ib}^{(\bar{s})}. \quad (3.5)$$

MAGOULÈS et coll. [2004] dans le cas de *FETI-2LM* et GANDER et KWOK [2011] dans le cas de méthodes de Schwarz ont montré qu'avec ce choix d'impédance d'interface, et dans le cas d'une décomposition dont le graphe de connectivité ne contient pas de cycles, la convergence est atteinte au plus en  $N_s - 1$  itérations. La figure 3.1 illustre un problème de ce type. C'est un exemple linéaire élastique dans les hypothèses des petites perturbations où le module de Young  $E$  vaut 1 GPa et le coefficient de Poisson  $\nu = 0.3$ . La géométrie est constituée de quatre cubes de côté 1 mm. Un déplacement de 1 mm est imposé dans la direction transverse de la poutre, en son extrémité. Le paramètre d'impédance d'interface est choisi comme étant celui de l'équation (3.1). Le solveur converge bien en  $N_s - 1 = 3$  itérations, ce qui se manifeste par une chute brusque du résidu à la troisième itération comme le montre le tableau 3.1 et la figure 3.3. En effet, le second membre se propage au sein de la structure au cours des itérations. La figure 3.2 illustre ce phénomène en donnant les déformés au cours des itérations d'un solveur *FETI-2LM*.

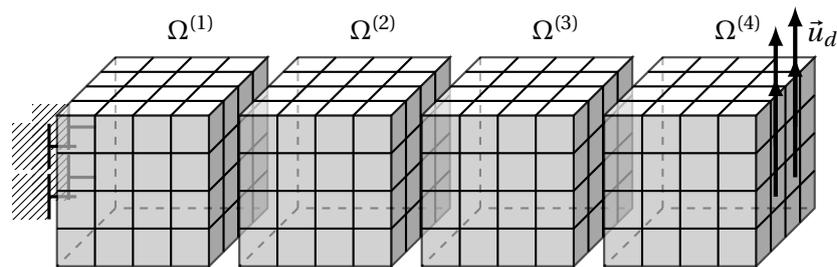


FIGURE 3.1 – Discretisation d'une poutre 3D découpée en 4 sous-domaines cubiques soumise à un chargement de flexion à son extrémité

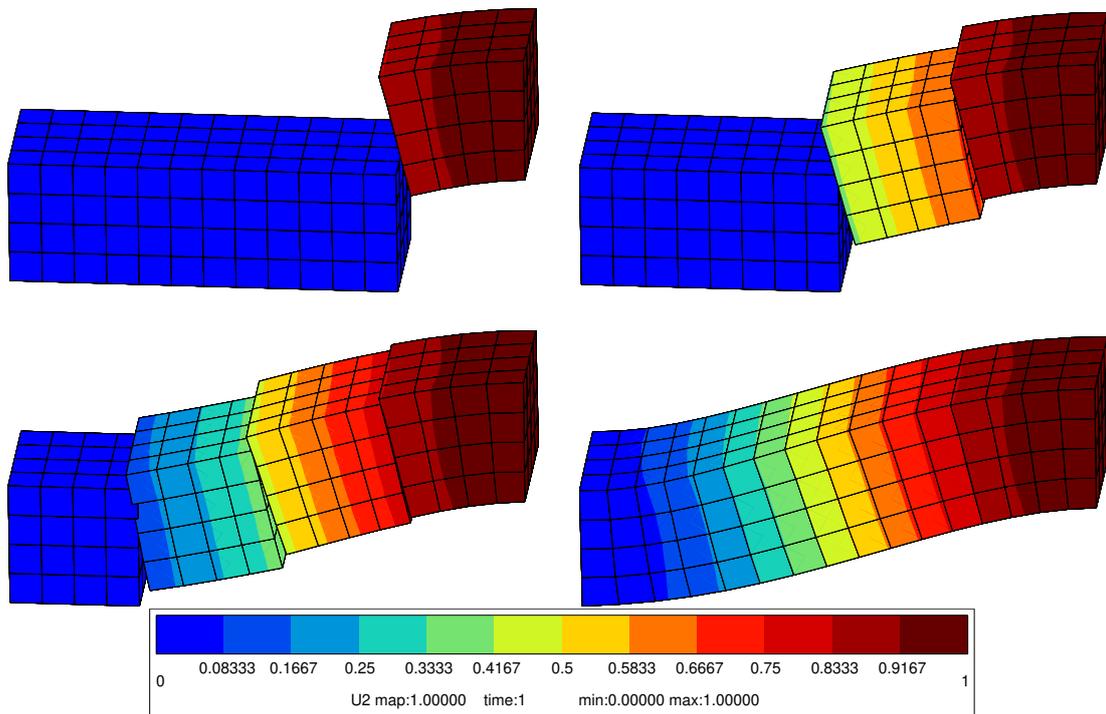


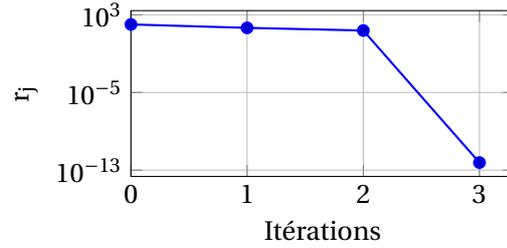
FIGURE 3.2 – Évolution de la solution au cours des itérations (gauche à droite, haut en bas). Le champ tracé est le champ en mm qui est dans la même direction que le déplacement imposé.

Par contre, dans le cas où la sous-structuration fait apparaître des points multiples, il n'y a pas de résultats théoriques qui montrent une convergence en un nombre maximal d'itération prédéterminé car la propagation du second membre est plus complexe d'après GANDER et KWOK [2011]. Cependant la pertinence du choix de  $\mathbf{Q}_b^{(s)} = \mathbf{S}_p^{(s)}$  ne semble pas être remis en question.

De plus, en comparant les équations (3.3) et (3.4), il est possible de remarquer une condition

Itération	Résidu
0	$1,02 \cdot 10^2$
1	$4,54 \cdot 10^1$
2	$2,42 \cdot 10^1$
3	$5,97 \cdot 10^{-13}$

TABLEAU 3.1 – Historique de convergence du cas de la figure 3.2


 FIGURE 3.3 – Convergence d'un solveur *FETI-2LM* lorsque  $\mathbf{Q}_b^{(s)} = \mathbf{S}_p^{(s)}$  sur le cas de la figure 3.2

supplémentaire pour la bonne convergence du solveur. En effet, en choisissant une initialisation de l'inconnue  $\boldsymbol{\mu}_{b,0}^{(s)}$  comme étant les efforts soumis à la sous-structure  $\Omega^{(s)}$  condensés de manière primale sur l'interface  $\Gamma_{ss}$ , le chargement se propage instantanément, dès l'initialisation. Donc en choisissant, en plus de l'impédance optimale  $\mathbf{Q}_b^{(s)} = \mathbf{S}_p^{(s)}$ , une initialisation de la forme  $\boldsymbol{\mu}_{b,0}^{(s)} = \mathbf{b}_p^{(s)}$ , le solveur converge dès l'initialisation avec un résidu de norme  $7.41 \cdot 10^{-13}$ , même en présence de points multiples.

L'impédance (équation (3.5)) et l'initialisation optimales sont de la forme :

$$\begin{cases} \mathbf{Q}_b^{(s)} &= \mathbf{K}_{bb}^{(s)} - \mathbf{K}_{bi}^{(s)} \mathbf{K}_{ii}^{(s)-1} \mathbf{K}_{ib}^{(s)} \\ \boldsymbol{\mu}_{b,0}^{(s)} &= \mathbf{f}_b^{(s)} - \mathbf{K}_{bi}^{(s)} \mathbf{K}_{ii}^{(s)-1} \mathbf{f}_i^{(s)} \end{cases}, \quad (3.6)$$

on peut remarquer qu'elles comprennent chacune, deux termes. Les deux premiers,  $\mathbf{K}_{bb}^{(s)}$  et  $\mathbf{f}_b^{(s)}$ , représentent les interactions locales, en contact avec le bord du sous-domaine  $\Omega^{(s)}$ , tandis que les deux suivants rendent compte des interactions plus lointaines.

Concernant l'impédance, le terme proche  $\mathbf{K}_{bb}^{(s)}$  est creux, bloc-diagonal et de même remplissage que  $\mathbf{K}_{bb}^{(s)}$ , alors que le terme lointain,  $\mathbf{K}_{bi}^{(s)} \mathbf{K}_{ii}^{(s)-1} \mathbf{K}_{ib}^{(s)}$ , est une matrice pleine car il dépend de la structure totale et couple tous les DDLs. Il faut noter que la présence du signe moins entre ces deux termes permet d'assouplir l'impédance locale à l'aide des interactions globales car l'utilisation du terme proche  $\mathbf{K}_{bb}^{(s)}$  revient à considérer le reste de la structure comme infiniment rigide.

Le terme lointain  $\mathbf{K}_{bi}^{(s)} \mathbf{K}_{ii}^{(s)-1} \mathbf{K}_{ib}^{(s)}$  est construit à l'aide de communications globales entre tous les sous-domaines. Son calcul n'est donc pas envisageable dans un contexte distribué pour conserver la localité des données. Des approximations de l'impédance optimale vont donc être nécessaires pour permettre une évaluation efficace de cette quantité.

## 3.2 Approximation de l'impédance optimale par une approche à deux niveaux

Cette section se penche sur quelques techniques qui ont été développées pour approximer l'impédance d'interface optimale qui est représentée par le complément de Schur de la partie de la structure complémentaire au sous-domaine considéré. Ensuite, l'approche exploitant la décomposition à deux niveaux est exposée.

### 3.2.1 Techniques d'évaluation de l'impédance d'interface

Pour estimer l'impédance d'interface optimale  $\mathbf{Q}_b^{(s)} = \mathbf{S}_p^{(s)}$ , il faut pouvoir approximer ce complément de Schur du complémentaire du sous-domaine. Plusieurs approches ont été développées et notamment des méthodes qui considèrent, soit les interactions proches du sous-domaine, soit les interactions plus lointaines.

### 3.2.1.1 Évaluation et approximation des interactions courtes distances

Le premier type d'approches consiste à calculer uniquement l'assemblage des compléments de Schur locaux des sous-domaines voisins pour chaque sous-domaine comme utilisé par [CRESTA et coll. \[2007\]](#) ou encore [HINOJOSA et coll. \[2014\]](#). Soit  $\text{neigh}(s)$  l'ensemble des indices des voisins de la sous-structure  $\Omega^{(s)}$ , l'approximation de cette impédance devient ainsi  $\tilde{\mathbf{S}}_p^{(s)}$  :

$$\mathbf{Q}_b^{(s)} \approx \tilde{\mathbf{S}}_p^{(s)} = \mathbf{A}^{(s)\text{T}} \left( \sum_{j \in \text{neigh}(s)} \mathbf{A}^{(j)} \mathbf{S}_p^{(j)} \mathbf{A}^{(j)\text{T}} \right) \mathbf{A}^{(s)}.$$

En comparaison de l'expression optimale, les communications sont réduites, par contre la matrice reste dense et le calcul explicite des compléments de Schur locaux doit encore être effectué car l'opérateur  $\mathbf{S}_m^{\diamond} = \mathbf{S}_p^{\diamond} + \mathbf{Q}_b^{\diamond}$  doit être factorisé pour les résolutions locales. Pour éviter cela, [MAGOULÈS et coll. \[2006\]](#) a considéré uniquement une condensation de la matrice de rigidité d'une bande d'éléments voisins d'une certaine épaisseur  $k$ . [OUMAZIZ et coll. \[2017\]](#) a adapté cette considération pour  $k = 1$ , dans un contexte non-intrusif, à une « semelle » correspondant à une bande d'un seul élément d'épaisseur à laquelle des conditions de Dirichlet nulles sont imposées sur sa frontière extérieure. Cette approximation, pour  $k = 1$ , revient à choisir la version *lumped* du complément de Schur local :

$$\tilde{\mathbf{S}}_{p,l}^{(s)} = \mathbf{A}^{(s)\text{T}} \left( \sum_{j \in \text{neigh}(s)} \mathbf{A}^{(j)} \mathbf{K}_{bb}^{(j)} \mathbf{A}^{(j)\text{T}} \right) \mathbf{A}^{(s)}. \quad (3.7)$$

Une autre possibilité consiste à choisir la version *super-lumped* du complément de Schur local conduisant à une quantité de communications moindre :

$$\tilde{\mathbf{S}}_{p,sl}^{(s)} = \mathbf{A}^{(s)\text{T}} \left( \sum_{j \in \text{neigh}(s)} \mathbf{A}^{(j)} \text{diag} \left( \mathbf{K}_{bb}^{(j)} \right) \mathbf{A}^{(j)\text{T}} \right) \mathbf{A}^{(s)}. \quad (3.8)$$

Ces deux dernières approximations sont peu chères à calculer car elles ne nécessitent pas de construire explicitement de nouveaux opérateurs car seules les matrices de rigidité, déjà stockées par chacun des sous-domaines, sont nécessaires. Par contre, elles ne peuvent rendre compte de phénomènes éloignés du sous-domaine (présence d'hétérogénéités, structure élancée, ...) et cette perte d'information peut être dommageable pour l'efficacité du solveur.

### 3.2.1.2 Approximation des interactions longues distances

Les techniques précédentes privilégient les informations locales en ignorant les effets longues distances. Dans le cadre général des méthodes de décomposition de domaine, les informations lointaines sont prises en compte à l'aide de méthodes multi-échelles telles que les problèmes grossiers des méthodes *FETI* de [FARHAT et coll. \[1994\]](#), *BDD* de [MANDEL \[1993\]](#) ou dans la version de la méthode *LaTIn* de [LADEVÈZE et NOUY \[2003\]](#) exploitant *a minima* la base des mouvements de corps rigides des sous-structures permettant de transmettre instantanément le chargement lointain à tous les sous-domaines. Inspirées des méthodes d'homogénéisation, des techniques de projection permettent de formuler une approximation de rang faible d'un opérateur en exploitant un problème « macroscopique » comme ont pu le faire [LADEVÈZE et DUREISSEIX \[1999\]](#), [GUIDAULT et coll. \[2008\]](#) ou encore [WANGERMEZ et coll. \[2020\]](#).

Soit  $\mathbf{U}$  une base orthonormale du sous-espace de l'espace dans lequel la solution est recherché, de manière générique, il est possible d'approximer un opérateur  $\mathbf{S}_p^{(s)}$  sous la forme :

$$\tilde{\mathbf{S}}_p^{(s)} \approx \mathbf{U} \left( \mathbf{U}^{\text{T}} \mathbf{S}_p^{(s)} \mathbf{U} \right) \mathbf{U}^{\text{T}}.$$

Le principe de Saint-Venant impose à  $\mathbf{U}$  de contenir au moins les mouvements de corps rigides des sous-structures auxquelles des modes de déformations affines (ou d'autres modes) peuvent être ajoutés.

### 3.2.1.3 Approximation à deux échelles

Dans le cadre des petits déplacements, utilisé en mécanique linéaire, les interactions longues distances ont un impact fort sur la réponse de la structure totale. Il semble donc peu judicieux de ne considérer que les interactions courte distance. Se limiter à l'une ou l'autre des stratégies ne peut pas permettre d'obtenir une information complète sur la réponse de la structure.

C'est pourquoi **GENDRE et coll. [2011]** (dans un schéma itératif global/local) et **NEGRELLO et coll. [2018]** (dans un contexte distribué) ont développé une stratégie consistant à combiner ces deux formulations.

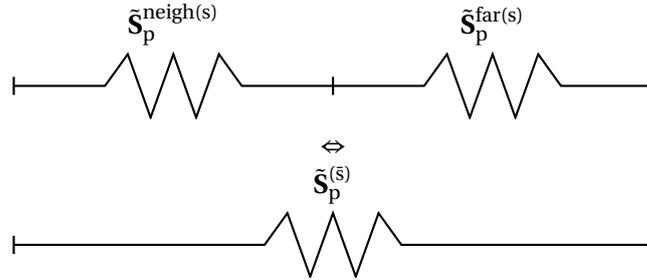


FIGURE 3.4 – Impédance équivalente à deux ressorts en série de **NEGRELLO et coll. [2018]**

Pour coupler les effets lointains et proches, **NEGRELLO et coll. [2018]** proposent une modélisation de cette impédance comme étant deux ressorts en série (voir figure 3.4) où chaque ressort a une rigidité associée à son comportement lointain ou proche. Ainsi, la souplesse de la structure complémentaire peut s'estimer à l'aide de la propriété d'additivité des souplesses :

$$\tilde{\mathbf{S}}_p^{(\tilde{s})^{-1}} = \tilde{\mathbf{S}}_p^{\text{neigh}(s)^{-1}} + \tilde{\mathbf{S}}_p^{\text{far}(s)^{-1}}$$

où  $\tilde{\mathbf{S}}_p^{\text{neigh}(s)}$  correspond à la rigidité des sous-domaines proches du sous-domaine  $\Omega^{(s)}$  sous forme de matrice creuse. *A contrario*,  $\tilde{\mathbf{S}}_p^{\text{far}(s)}$  représente la rigidité des sous-domaines loin de  $\Omega^{(s)}$  sous forme d'une approximation de rang faible. Plus précisément, la souplesse équivalente peut être choisie sous la forme :

$$\tilde{\mathbf{S}}_p^{(\tilde{s})^{-1}} = \mathbf{K}_{bb}^{\text{neigh}(s)^{-1}} + \mathbf{A}^{(s)T} \mathbf{V} \mathbf{F} \mathbf{V}^T \mathbf{A}^{(s)},$$

où  $\mathbf{K}_{bb}^{\text{neigh}(s)}$  peut être choisi comme étant l'une des deux expressions (3.7) ou (3.8),  $\mathbf{F}$  une matrice carrée de petite dimension  $m \times m$  et  $\mathbf{V}$  une base de vecteurs d'interface à  $m$  colonnes. Par la suite, la notation  $\mathbf{V}^{(s)} = \mathbf{A}^{(s)T} \mathbf{V}$  sera utilisée pour expliciter la contribution locale de la base  $\mathbf{V}$  sur  $\Omega^{(s)}$ .

À l'aide de la formule de Sherman-Morrison-Woodbury<sup>1</sup> (**HAGER [1989]**), l'approximation de l'impédance d'interface peut s'exprimer de la manière suivante :

$$\tilde{\mathbf{S}}_p^{(\tilde{s})} = \mathbf{K}_{bb}^{\text{neigh}(s)} - \mathbf{K}_{bb}^{\text{neigh}(s)} \mathbf{V}^{(s)} \left( \mathbf{F}^{-1} + \mathbf{V}^{(s)T} \mathbf{K}_{bb}^{\text{neigh}(s)} \mathbf{V}^{(s)} \right)^{-1} \mathbf{V}^{(s)T} \mathbf{K}_{bb}^{\text{neigh}(s)}.$$

Cette impédance correspond à une matrice creuse corrigée par un terme de rang faible, ce qui rappelle l'expression (3.6). Cette approximation n'ajoute que l'inversion de matrices de faibles dimensions  $m$ , à savoir  $\mathbf{F}$  et  $\left( \mathbf{F}^{-1} + \mathbf{V}^{(s)T} \mathbf{K}_{bb}^{\text{neigh}(s)} \mathbf{V}^{(s)} \right)$ . En pratique, **NEGRELLO et coll. [2018]** a choisi les opérateurs comme  $\mathbf{V} = \left[ \dots \left( \sum_{j \neq s} \mathbf{A}^{(j)} \mathbf{M}^{(j)} \mathbf{A}^{(j)T} \right)^{-1} \mathbf{A}^{(s)} \mathbf{M}^{(s)} \mathbf{R}_b^{(s)} \dots \right]_{j \neq s}$  et  $\mathbf{F} = \left( \mathbf{V}^T \mathbf{S}_p^{(\tilde{s})} \mathbf{V} \right)^{-1}$ .  $\mathbf{F}$  est un opérateur de rang faible dont le rang correspond à la somme du nombre de modes rigides de tous les sous-domaines mis à part  $\Omega^{(s)}$ , et  $\mathbf{V}$  l'assemblage des modes rigides excluant la contribution du sous-domaine  $\Omega^{(s)}$ . Cette impédance est complexe à mettre en œuvre du fait du calcul d'un problème grossier dans l'opérateur  $\mathbf{F}$ .

1. L'inverse d'une matrice  $\mathbf{M} = \mathbf{A} + \mathbf{UCV}$  s'écrit  $\mathbf{M}^{-1} = (\mathbf{A} + \mathbf{UCV})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1} \mathbf{U} (\mathbf{C}^{-1} + \mathbf{V} \mathbf{A}^{-1} \mathbf{U})^{-1} \mathbf{V} \mathbf{A}^{-1}$  sous réserve que  $\mathbf{A}$  et  $\mathbf{C}$  soient inversibles et que  $\mathbf{U}$  et  $\mathbf{V}$  soient rectangulaires et de dimensions cohérentes

### 3.2.2 Approximation à deux niveaux

En mettant de côté l'impédance de NEGRELLO et coll. [2018], les approximations développées dans la section précédente ne permettent pas, à la fois de tenir compte du comportement proche et du comportement distant. On se propose donc de développer dans cette section une méthode permettant de prendre en compte à la fois ces deux aspects pour un coût numérique faible.

#### 3.2.2.1 Expression voulue

En se plaçant dans le cadre de la sous-structuration à deux niveaux, illustré par la figure 3.5 qui a été développée dans le chapitre 2, l'information lointaine peut être obtenue à l'aide du maillage grossier de la structure auquel toutes les conditions limites auront été appliquées à l'identique du problème fin sous-structuré. Ce maillage grossier est distribué sur chacun des sous-domaines. Sur des structures où les bords ne sont pas des plans comme l'aube du chapitre précédent (voir figure 2.23), le maillage grossier peut être éloigné de la géométrie réelle. Cependant cela ne pose pas de problèmes majeurs *a priori* pour capter les effets lointains dans l'impédance.

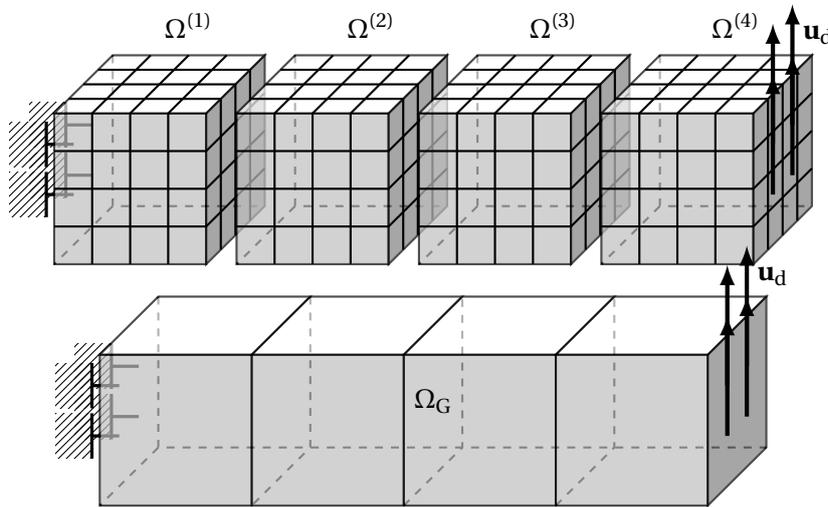


FIGURE 3.5 – Sous-structuration hiérarchique d'une poutre 3D en 4 sous-domaines avec son maillage grossier associé et le chargement considéré

Le coût important engendré par le calcul de l'interaction lointaine optimale peut ainsi être réduit en calculant un comportement grossier en tant qu'interaction lointaine. Il faut noter que l'approximation qui est visée doit permettre d'être exacte dans le cas où le maillage grossier  $\Omega_G$  et le maillage de calcul  $\Omega^{(s)}$  ont des tailles d'éléments égales et ainsi permettre une convergence immédiate dès l'initialisation. Finalement, chaque sous-domaine a la connaissance de son propre maillage  $\Omega^{(s)}$  ainsi que de l'approximation sur la structure grossière de son complémentaire comme le montre la figure 3.6.

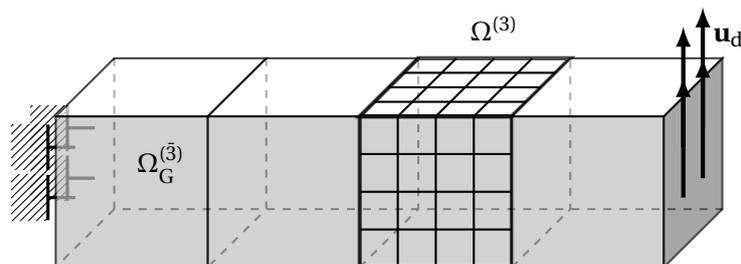


FIGURE 3.6 – Distribution des données pour le sous-domaine  $\Omega^{(3)}$

Ainsi, en notant  $\mathbf{K}_G^{(s)}$  la matrice de raideur grossière du complémentaire au domaine  $\Omega^{(s)}$ , il serait souhaitable de modifier les paramètres de l'égalité (3.6) pour obtenir une impédance à deux

niveaux de la forme :

$$\begin{cases} \mathbf{Q}_b^{(s)} &= \mathbf{K}_{bb}^{(s)} - \mathcal{M}^{(s,\bar{s})} \mathbf{K}_{G_{bi}}^{(s)} \mathbf{K}_{G_{ii}}^{(s)-1} \mathbf{K}_{G_{ib}}^{(s)} \mathcal{M}^{(s,\bar{s})T} \\ \boldsymbol{\mu}_{b,0}^{(s)} &= \mathbf{f}_b^{(s)} - \mathcal{M}^{(s,\bar{s})} \mathbf{K}_{G_{bi}}^{(s)} \mathbf{K}_{G_{ii}}^{(s)-1} \mathbf{f}_{G_i}^{(s)} \end{cases}, \quad (3.9)$$

où  $\mathcal{M}^{(s,\bar{s})}$  est un opérateur rectangulaire permettant de transférer des quantités grossières sur un maillage fin. Les quantités lointaines grossières seront notées :

$$\begin{cases} \tilde{\mathbf{S}}_{p_G}^{(s)} &= -\mathbf{K}_{G_{bi}}^{(s)} \mathbf{K}_{G_{ii}}^{(s)-1} \mathbf{K}_{G_{ib}}^{(s)} \\ \tilde{\mathbf{b}}_{p_G}^{(s)} &= -\mathbf{K}_{G_{bi}}^{(s)} \mathbf{K}_{G_{ii}}^{(s)-1} \mathbf{f}_{G_i}^{(s)} \end{cases}$$

### Remarque 3.1

Dans le cas où les maillages fins et grossiers sont identiques (*i.e.*  $\mathbf{K}_G^{(s)} = \mathbf{K}^{(s)}$  et  $\mathbf{f}_G^{(s)} = \mathbf{f}^{(s)}$ ), tous les sous-domaines connaissent entièrement la structure. Ainsi, le problème local est équivalent au problème global et la convergence est donc immédiate. D'un point de vue algébrique, il faut que les expressions (3.9) soient équivalentes aux expressions de l'équation (3.6). Pour cela, il suffit que l'opérateur  $\mathcal{M}^{(s,\bar{s})}$  soit égal à la matrice identité carrée  $\mathbf{I}_{bb}$ .

### 3.2.2.2 Couplage entre deux maillages imbriqués

**Interpolation cinématique** L'idée la plus simple pour coupler ces deux maillages est de réaliser une interpolation entre les DDLs des deux interfaces, cela est aussi appelé de la collocation (DUREISSEIX et BAVESTRELLO [2006]). Cette solution n'est pas adaptée ici car elle représente une condition forte sur les degrés de liberté de déplacements et ne permet pas de rendre compte des efforts qui transitent entre les sous-structures. Cette approche est tout de même présentée ici car elle permet de définir une interpolation entre ces maillages imbriqués facilitant les développements suivants. Cela correspond à trouver une matrice rectangulaire  $\mathcal{J}^{(s,\bar{s})}$  qui permet d'écrire une relation de passage entre les DDLs d'interface fins et grossiers :

$$\mathbf{u}_b^{(s)} = \mathcal{J}^{(s,\bar{s})} \mathbf{u}_{G_b}^{(s)} \quad (3.10)$$

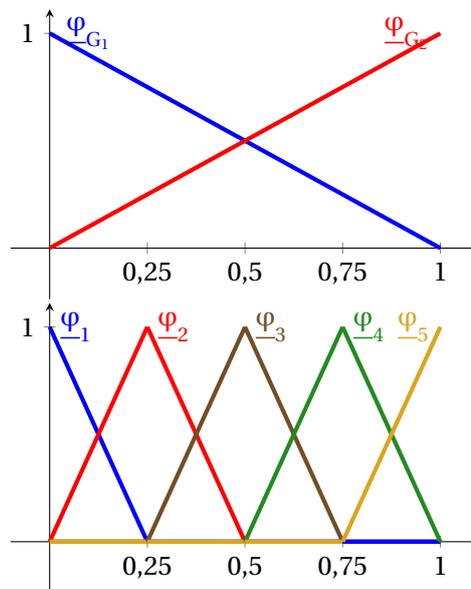


FIGURE 3.7 – Fonctions de forme d'un élément unidimensionnel grossier (au dessus) auquel deux niveaux de raffinement hiérarchiques ont été appliqués (en dessous)

Soient  $\underline{\mathbf{u}}_b^{(s)}(\mathbf{x})$  et  $\underline{\mathbf{u}}_{G_b}^{(\bar{s})}(\mathbf{x})$  les déplacements continus (pas définis uniquement aux nœuds mais en tout point de l'espace) fins et grossiers à l'interface à la position  $\mathbf{x}$ . Dans le cas de maillages imbriqués, l'interpolation impose l'égalité de ces déplacements à l'interface :

$$\underline{\mathbf{u}}_b^{(s)}(\mathbf{x}) = \underline{\mathbf{u}}_{G_b}^{(\bar{s})}(\mathbf{x}).$$

Ainsi, en notant  $\varphi_k$  (respectivement  $\varphi_{G_1}$ ) la fonction de forme EF de l'interface fine (respectivement grossière) associée au DDL  $k$  (respectivement  $l$ ), cette égalité devient :

$$\sum_{k=1}^{n_b} \varphi_k \underline{\mathbf{u}}_b^{(s)}[k] = \sum_{l=1}^{n_{G_b}} \varphi_{G_l} \underline{\mathbf{u}}_{G_b}^{(\bar{s})}[l].$$

En notant  $\Phi(\mathbf{x}) = [\varphi_0(\mathbf{x}) \ \dots \ \varphi_{n_b}(\mathbf{x})]$  et  $\Phi_G(\mathbf{x}) = [\varphi_{G_0}(\mathbf{x}) \ \dots \ \varphi_{G_{n_{G_b}}}(\mathbf{x})]$  et à l'aide de l'équation (3.10), la condition d'interpolation devient :

$$\begin{aligned} \Phi(\mathbf{x}) \underline{\mathbf{u}}_b^{(s)} &= \Phi_G(\mathbf{x}) \underline{\mathbf{u}}_{G_b}^{(\bar{s})}; \\ \Phi(\mathbf{x}) \mathcal{J}^{(s,\bar{s})} \underline{\mathbf{u}}_{G_b}^{(\bar{s})} &= \Phi_G(\mathbf{x}) \underline{\mathbf{u}}_{G_b}^{(\bar{s})}; \\ \Phi(\mathbf{x}) \mathcal{J}^{(s,\bar{s})} &= \Phi_G(\mathbf{x}). \end{aligned}$$

Les fonctions de formes sont illustrés dans un cas unidimensionnel sur la figure 3.7. Dans le cas illustré ici, l'opérateur  $\mathcal{J}^{(s,\bar{s})}$  est de la forme :

$$\mathcal{J}^{(s,\bar{s})} = \begin{bmatrix} 1 & 0 \\ 0.75 & 0.25 \\ 0.5 & 0.5 \\ 0.25 & 0.75 \\ 0 & 1 \end{bmatrix}.$$

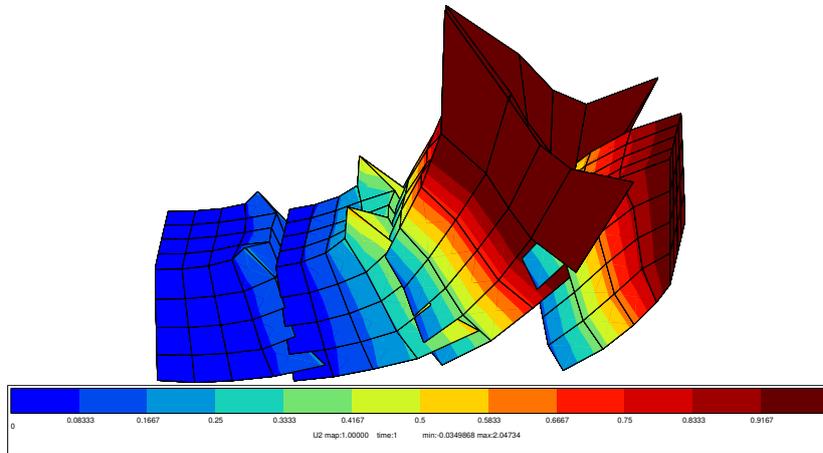


FIGURE 3.8 – Déformée associée à la solution initiale  $\underline{\mathbf{u}}_{b,0}^{(s)}$  pour  $\mathcal{M}^{(s,\bar{s})} = \mathcal{J}^{(s,\bar{s})}$  et  $\mathbf{Q}_b^{(s)} = \mathbf{K}_{bb}^{(\bar{s})}$ .

Cette interpolation est testée sur l'exemple de la figure 3.5. Son influence sur la solution initiale pour une impédance *lumpée* ( $\mathbf{Q}_b^{(s)} = \mathbf{K}_{bb}^{(\bar{s})}$ ) est illustrée sur la figure 3.8. Ce couplage semble surestimer les efforts transmis aux nœuds de coins. En effet, la déformée obtenue ne semble pas résulter d'une condensation appropriée du chargement. L'interpolation ne rend pas compte de la connectivité existant entre les DDL et permettant l'estimation des efforts de réaction. Ainsi, un autre couplage, permettant de transférer des efforts est nécessaire.

**Couplage mortier** Une autre approche consiste à minimiser l'énergie du saut de déplacement entre les deux maillages : c'est la méthode que l'on appelle *mortar* (« mortier » en français), développée dans le cadre du couplage surfacique entre deux domaines sans recouvrement par BERNARDI et coll. [1993] ou BEN BELGACEM [1999], ce qui permet d'étendre cette notion au contact

(POPP [2012] puis AKULA [2019]). Des travaux ont aussi été menés dans le cadre de méthodes de décomposition de domaine comme *FETI* par STEFANICA [2001, 2005] ou *FETI-2LM* par ROUX [2009]. Cela revient à imposer un saut de déplacement nul en moyenne entre les deux maillages.

Ainsi, au sein de la minimisation de l'énergie potentielle totale du système constitué des deux maillages (fin et grossier), la minimisation de l'énergie surfacique du saut de déplacement passe par l'introduction de multiplicateurs de Lagrange  $\underline{\eta}$  représentant les inter-efforts entre les deux maillages. Ils permettent de définir la quantité à minimiser  $W_g$  :

$$W_g = \int_{\Gamma^{(s,\bar{s})}} \underline{\eta} \left( \underline{\mathbf{u}}_b^{(s)} - \underline{\mathbf{u}}_{G_b}^{(\bar{s})} \right) d\Gamma.$$

La différenciation de cette quantité est nécessaire pour cette minimisation, et elle se met sous la forme suivante :

$$\delta W_g = \int_{\Gamma^{(s,\bar{s})}} \delta \underline{\eta} \left( \underline{\mathbf{u}}_b^{(s)} - \underline{\mathbf{u}}_{G_b}^{(\bar{s})} \right) d\Gamma + \int_{\Gamma^{(s,\bar{s})}} \underline{\eta} \left( \delta \underline{\mathbf{u}}_b^{(s)} - \delta \underline{\mathbf{u}}_{G_b}^{(\bar{s})} \right) d\Gamma. \quad (3.11)$$

Précédemment, une forme discrète des déplacements a été exprimée à l'aide de la méthode de Galerkin en utilisant des fonctions de formes EF différentes sur  $\Omega^{(s)}$  et  $\Omega^{(\bar{s})}$ . Il faut faire de même et définir une base de fonctions permettant de décrire de manière discrète les multiplicateurs de Lagrange. Les inter-efforts sont choisis comme étant exprimés dans la base grossière du domaine  $\Omega^{(\bar{s})}$  (l'autre choix conduisant à retrouver le même couplage par collocation que celui de la section 3.2.2.2). Ainsi, les multiplicateurs de Lagrange  $\underline{\eta}$  peuvent être exprimés en fonction d'un vecteur d'inconnues discrètes  $\boldsymbol{\eta}$ , par la relation :

$$\underline{\eta} = \Phi_G \boldsymbol{\eta}.$$

Et l'équation (3.11) peut se mettre sous la forme discrète suivante :

$$\begin{aligned} \delta W_g &= \int_{\Gamma^{(s,\bar{s})}} \delta \boldsymbol{\eta}^T \Phi_G^T \Phi_G \mathbf{u}_b^{(s)} d\Gamma - \int_{\Gamma^{(s,\bar{s})}} \delta \boldsymbol{\eta}^T \Phi_G^T \Phi_G \mathbf{u}_{G_b}^{(\bar{s})} d\Gamma \\ &+ \int_{\Gamma^{(s,\bar{s})}} \delta \mathbf{u}_b^{(s)T} \Phi^T \Phi_G \boldsymbol{\eta} d\Gamma - \int_{\Gamma^{(s,\bar{s})}} \delta \mathbf{u}_{G_b}^{(\bar{s})T} \Phi_G^T \Phi_G \boldsymbol{\eta} d\Gamma \\ &= \delta \boldsymbol{\eta}^T \int_{\Gamma^{(s,\bar{s})}} \Phi_G^T \Phi_G d\Gamma \mathbf{u}_b^{(s)} - \delta \boldsymbol{\eta}^T \int_{\Gamma^{(s,\bar{s})}} \Phi_G^T \Phi_G d\Gamma \mathbf{u}_{G_b}^{(\bar{s})} \\ &+ \delta \mathbf{u}_b^{(s)T} \int_{\Gamma^{(s,\bar{s})}} \Phi^T \Phi_G d\Gamma \boldsymbol{\eta} - \delta \mathbf{u}_{G_b}^{(\bar{s})T} \int_{\Gamma^{(s,\bar{s})}} \Phi_G^T \Phi_G d\Gamma \boldsymbol{\eta} \\ &= \delta \boldsymbol{\eta}^T \mathbf{C}_\Gamma \mathbf{u}_b^{(s)} - \delta \boldsymbol{\eta}^T \mathbf{M}_{G_\Gamma} \mathbf{u}_{G_b}^{(\bar{s})} + \delta \mathbf{u}_b^{(s)T} \mathbf{C}_\Gamma^T \boldsymbol{\eta} - \delta \mathbf{u}_{G_b}^{(\bar{s})T} \mathbf{M}_{G_\Gamma} \boldsymbol{\eta}, \end{aligned}$$

où  $\mathbf{M}_{G_\Gamma}$  est la matrice (symétrique) de masse grossière de l'interface, et  $\mathbf{C}_\Gamma$  une matrice de couplage rectangulaire, définies par :

$$\begin{cases} \mathbf{M}_{G_\Gamma} &= \int_{\Gamma^{(s,\bar{s})}} \Phi_G^T \Phi_G d\Gamma \\ \mathbf{C}_\Gamma &= \int_{\Gamma^{(s,\bar{s})}} \Phi_G^T \Phi d\Gamma = \mathcal{J}^{(s,\bar{s})T} \int_{\Gamma^{(s,\bar{s})}} \Phi^T \Phi d\Gamma = \mathcal{J}^{(s,\bar{s})T} \mathbf{M}_\Gamma \end{cases},$$

avec  $\mathbf{M}_\Gamma$  la matrice (symétrique) de masse de l'interface du maillage fin. Par construction, elle vérifie l'égalité  $\mathbf{M}_{G_\Gamma} = \mathcal{J}^{(s,\bar{s})T} \mathbf{M}_\Gamma \mathcal{J}^{(s,\bar{s})}$ . Ainsi, l'équation (3.11) est équivalente à l'équation matricielle :

$$\begin{aligned} \delta W_g &= \begin{bmatrix} \delta \mathbf{u}_b^{(s)T} & \delta \mathbf{u}_{G_b}^{(\bar{s})T} & \delta \boldsymbol{\eta}^T \end{bmatrix} \begin{bmatrix} \mathbf{M}_\Gamma \mathcal{J}^{(s,\bar{s})} \boldsymbol{\eta} \\ -\mathbf{M}_{G_\Gamma} \boldsymbol{\eta} \\ \mathcal{J}^{(s,\bar{s})T} \mathbf{M}_\Gamma \mathbf{u}_b^{(s)} - \mathbf{M}_{G_\Gamma} \mathbf{u}_{G_b}^{(\bar{s})} \end{bmatrix} \\ &= \begin{bmatrix} \delta \mathbf{u}_b^{(s)T} & \delta \mathbf{u}_{G_b}^{(\bar{s})T} & \delta \boldsymbol{\eta}^T \end{bmatrix} \begin{bmatrix} 0 & 0 & \mathbf{M}_\Gamma \mathcal{J}^{(s,\bar{s})} \\ 0 & 0 & -\mathbf{M}_{G_\Gamma} \\ \mathcal{J}^{(s,\bar{s})T} \mathbf{M}_\Gamma & -\mathbf{M}_{G_\Gamma} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_b^{(s)} \\ \mathbf{u}_{G_b}^{(\bar{s})} \\ \boldsymbol{\eta} \end{bmatrix}. \end{aligned}$$

que l'on peut ajouter dans l'équation d'équilibre du sous-domaine  $\Omega^{(s)}$  soumis à des conditions de Robin pour arriver au système matriciel suivant :

$$\begin{bmatrix} \mathbf{K}_{ii}^{(s)} & \mathbf{K}_{ib}^{(s)} & 0 & 0 & 0 \\ \mathbf{K}_{bi}^{(s)} & \mathbf{K}_{bb}^{(s)} + \mathbf{K}_{bb}^{(\bar{s})} & 0 & 0 & \mathbf{M}_\Gamma \mathcal{J}^{(s,\bar{s})} \\ 0 & 0 & \mathbf{K}_{Gi}^{(\bar{s})} & \mathbf{K}_{Gb}^{(\bar{s})} & 0 \\ 0 & 0 & \mathbf{K}_{Gbi}^{(\bar{s})} & 0 & -\mathbf{M}_{G\Gamma} \\ 0 & \mathcal{J}^{(s,\bar{s})T} \mathbf{M}_\Gamma & 0 & -\mathbf{M}_{G\Gamma} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_i^{(s)} \\ \mathbf{u}_b^{(s)} \\ \mathbf{u}_{Gi}^{(\bar{s})} \\ \mathbf{u}_{Gb}^{(\bar{s})} \\ \boldsymbol{\eta} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_i^{(s)} \\ \mathbf{f}_b^{(s)} + \mathbf{f}_b^{(\bar{s})} + \boldsymbol{\mu}_b^{(s)} \\ \mathbf{f}_{Gi}^{(\bar{s})} \\ 0 \\ 0 \end{bmatrix}. \quad (3.12)$$

Il est à noter, dans ce système, que les quantités de bord appartenant au complémentaire sont choisies en tant que quantités fines comme le montrent les termes ajoutés  $\mathbf{K}_{bb}^{(\bar{s})}$  et  $\mathbf{f}_b^{(\bar{s})}$  qui remplacent leurs homologues grossiers. En effet, la présence de  $\mathbf{K}_{bb}^{(\bar{s})}$  permet de rendre l'opérateur  $\mathbf{Q}_b^{(s)}$  de l'expression (3.9) inversible en comparaison de  $\mathbf{K}_{Gbb}^{(\bar{s})}$  ramenée sur le fin à l'aide de l'opérateur  $\mathcal{M}^{(s,\bar{s})}$ . En condensant ce système sur les DDLs de bord grossiers et fins (en éliminant les DDLs intérieurs), le système suivant peut être écrit :

$$\begin{bmatrix} \mathbf{S}_p^{(s)} + \mathbf{K}_{bb}^{(\bar{s})} & 0 & \mathbf{M}_\Gamma \mathcal{J}^{(s,\bar{s})} \\ 0 & \tilde{\mathbf{S}}_{pG}^{(\bar{s})} & -\mathbf{M}_{G\Gamma} \\ \mathcal{J}^{(s,\bar{s})T} \mathbf{M}_\Gamma & -\mathbf{M}_{G\Gamma} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_b^{(s)} \\ \mathbf{u}_{Gb}^{(\bar{s})} \\ \boldsymbol{\eta} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_p^{(s)} + \mathbf{f}_b^{(\bar{s})} + \boldsymbol{\mu}_b^{(s)} \\ \tilde{\mathbf{b}}_{pG}^{(\bar{s})} \\ 0 \end{bmatrix} \quad (3.13)$$

La dernière ligne du système (3.13) traduit la condition de couplage entre les deux maillages :

$$\mathbf{u}_{Gb}^{(\bar{s})} = \mathbf{M}_{G\Gamma}^{-1} \mathcal{J}^{(s,\bar{s})T} \mathbf{M}_\Gamma \mathbf{u}_b^{(s)}. \quad (3.14)$$

Injecter (3.14) dans la seconde ligne du système (3.13) conduit à :

$$\begin{aligned} \tilde{\mathbf{S}}_{pG}^{(\bar{s})} \mathbf{M}_{G\Gamma}^{-1} \mathcal{J}^{(s,\bar{s})T} \mathbf{M}_\Gamma \mathbf{u}_b^{(s)} - \mathbf{M}_{G\Gamma} \boldsymbol{\eta} &= \tilde{\mathbf{b}}_{pG}^{(\bar{s})} \\ \boldsymbol{\eta} &= \mathbf{M}_{G\Gamma}^{-1} \tilde{\mathbf{S}}_{pG}^{(\bar{s})} \mathbf{M}_{G\Gamma}^{-1} \mathcal{J}^{(s,\bar{s})T} \mathbf{M}_\Gamma \mathbf{u}_b^{(s)} - \mathbf{M}_{G\Gamma}^{-1} \tilde{\mathbf{b}}_{pG}^{(\bar{s})} \end{aligned} \quad (3.15)$$

Finalement, en injectant (3.15) dans la première ligne du système (3.13), l'équation d'équilibre du sous-domaine  $\Omega^{(s)}$  devient :

$$\left[ \mathbf{S}_p^{(s)} + \mathbf{K}_{bb}^{(\bar{s})} + \mathbf{M}_\Gamma \mathcal{J}^{(s,\bar{s})} \mathbf{M}_{G\Gamma}^{-1} \tilde{\mathbf{S}}_{pG}^{(\bar{s})} \mathbf{M}_{G\Gamma}^{-1} \mathcal{J}^{(s,\bar{s})T} \mathbf{M}_\Gamma \right] \mathbf{u}_b^{(s)} = \mathbf{b}_p^{(s)} + \mathbf{f}_b^{(\bar{s})} + \mathbf{M}_\Gamma \mathcal{J}^{(s,\bar{s})} \mathbf{M}_{G\Gamma}^{-1} \tilde{\mathbf{b}}_{pG}^{(\bar{s})} + \boldsymbol{\mu}_b^{(s)}$$

Par identification avec les équations (3.3) et (3.9), l'opérateur  $\mathcal{M}^{(s,\bar{s})}$  peut être calculé de la manière suivante :

$$\mathcal{M}^{(s,\bar{s})} = \mathbf{M}_\Gamma \mathcal{J}^{(s,\bar{s})} \mathbf{M}_{G\Gamma}^{-1} = \mathbf{M}_\Gamma \mathcal{J}^{(s,\bar{s})} \left( \mathcal{J}^{(s,\bar{s})T} \mathbf{M}_\Gamma \mathcal{J}^{(s,\bar{s})} \right)^{-1}$$

Dans l'exemple de la figure 3.7, cet opérateur est :

$$\mathcal{M}^{(s,\bar{s})} = \frac{1}{16} \begin{bmatrix} 7 & -3 \\ 10 & -2 \\ 4 & 4 \\ -2 & 10 \\ -3 & 7 \end{bmatrix}$$

Avec ce couplage, et dans les mêmes conditions que dans le paragraphe sur l'interpolation cinématique simple, l'initialisation est bien plus régulière comme le montre la figure 3.9

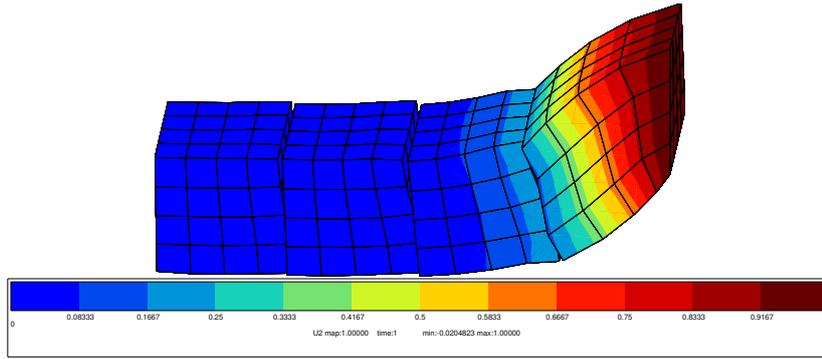


FIGURE 3.9 – Déformée associée à la solution initiale  $\boldsymbol{\mu}_{b,0}^{(s)}$  pour  $\mathcal{M}^{(s,\bar{s})} = \mathbf{M}_\Gamma \mathcal{J}^{(s,\bar{s})} \mathbf{M}_{G_\Gamma}^{-1}$  et  $\mathbf{Q}_b^{(s)} = \mathbf{K}_{bb}^{(\bar{s})}$ .

### Remarque 3.2 : Autre choix pour la matrice de raideur du complémentaire

Dans l'équation (3.12), l'opérateur  $\mathbf{K}^{(s)}$  a été modifié en ajoutant la raideur fine de l'interface du complémentaire  $\mathbf{K}_{bb}^{(\bar{s})}$ . Cela permet de rendre cette matrice inversible même lorsqu'un nombre insuffisant de conditions aux limites de Dirichlet sont imposées au sous domaine  $\Omega^{(s)}$ . Ainsi, il est nécessaire de retrancher cette rigidité à la matrice  $\mathbf{K}_G^{(s)}$  de rigidité grossière. Le choix fait ici pour retrancher cette quantité n'est pas unique; en effet, le système (3.12) peut aussi, par exemple, être mis sous la forme :

$$\begin{bmatrix} \mathbf{K}_{ii}^{(s)} & \mathbf{K}_{ib}^{(s)} & 0 & 0 & 0 \\ \mathbf{K}_{bi}^{(s)} & \mathbf{K}_{bb}^{(s)} + \mathbf{K}_{bb}^{(\bar{s})} & 0 & 0 & \mathbf{M}_\Gamma \mathcal{J}^{(s,\bar{s})} \\ 0 & 0 & \mathbf{K}_{Gii}^{(\bar{s})} & \mathbf{K}_{Gib}^{(\bar{s})} & 0 \\ 0 & 0 & \mathbf{K}_{Gbi}^{(\bar{s})} & \mathbf{K}_{Gbb}^{(\bar{s})} - \mathcal{J}^{(s,\bar{s})T} \mathbf{K}_{bb}^{(\bar{s})} \mathcal{J}^{(s,\bar{s})} & -\mathbf{M}_{G_\Gamma} \\ 0 & \mathcal{J}^{(s,\bar{s})T} \mathbf{M}_\Gamma & 0 & -\mathbf{M}_{G_\Gamma} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_i^{(s)} \\ \mathbf{u}_b^{(s)} \\ \mathbf{u}_{G_i}^{(\bar{s})} \\ \mathbf{u}_{G_b}^{(\bar{s})} \\ \boldsymbol{\eta} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_i^{(s)} \\ \mathbf{f}_b^{(s)} + \mathbf{f}_b^{(\bar{s})} + \boldsymbol{\mu}_b^{(s)} \\ \mathbf{f}_{G_i}^{(\bar{s})} \\ 0 \\ 0 \end{bmatrix},$$

où une interpolation de la matrice de raideur de l'interface fine complémentaire a été retranchée de celle de la matrice grossière. Ce choix conduit à l'obtention d'une impédance d'interface de la forme :

$$\mathbf{Q}_b^{(s)} = \mathbf{K}_{bb}^{(\bar{s})} + \mathcal{M}^{(s,\bar{s})} \left( \mathbf{S}_{p_G}^{(\bar{s})} - \mathcal{J}^{(s,\bar{s})T} \mathbf{K}_{bb}^{(\bar{s})} \mathcal{J}^{(s,\bar{s})} \right) \mathcal{M}^{(s,\bar{s})T},$$

où  $\mathbf{S}_{p_G}^{(\bar{s})}$  est le complément de Schur primal total de la structure grossière complémentaire et vaut  $\mathbf{S}_{p_G}^{(\bar{s})} = \mathbf{K}_{Gbb}^{(\bar{s})} - \mathbf{K}_{Gbi}^{(\bar{s})} \mathbf{K}_{Gii}^{(\bar{s})}^{-1} \mathbf{K}_{Gib}^{(\bar{s})}$ . Cette impédance possède une propriété importante concernant son interpolation sur le maillage grossier :

$$\mathcal{J}^{(s,\bar{s})} \mathbf{Q}_b^{(s)} \mathcal{J}^{(s,\bar{s})T} = \mathbf{S}_{p_G}^{(\bar{s})}$$

qui représente exactement le comportement de la structure complémentaire grossière.

### 3.2.2.3 Factorisation de l'opérateur local

L'opérateur local  $\mathbf{S}_m^{(s)} = \left( \mathbf{S}_p^{(s)} + \mathbf{Q}_b^{(s)} \right)^{-1}$  est un opérateur qui peut devenir coûteux à calculer en raison de la densité de  $\mathbf{S}_p^{(s)} + \mathbf{Q}_b^{(s)}$  mais à l'aide de quelques manipulations algébriques, il est possible de montrer que le coût d'inversion de  $\mathbf{S}_p^{(s)} + \mathbf{Q}_b^{(s)}$  n'est pas très élevé. En effet, à partir de la définition donnée de  $\mathbf{S}_m^{(s)}$ , il est possible d'écrire :

$$\begin{aligned}
 \left( \mathbf{S}_p^{(s)} + \mathbf{Q}_b^{(s)} \right)^{-1} &= \mathbf{t}^{(s)} \left( \mathbf{K}^{(s)} + \mathbf{t}^{(s)T} \mathbf{Q}_b^{(s)} \mathbf{t}^{(s)} \right)^{-1} \mathbf{t}^{(s)T} \\
 &= \mathbf{t}^{(s)} \left( \mathbf{K}^{(s)} + \mathbf{t}^{(s)T} \mathbf{K}_{bb}^{(s)} \mathbf{t}^{(s)} + \mathbf{t}^{(s)T} \mathcal{M}^{(s,\bar{s})} \tilde{\mathbf{S}}_{p_G}^{(s)} \mathcal{M}^{(s,\bar{s})T} \mathbf{t}^{(s)} \right)^{-1} \mathbf{t}^{(s)T} \\
 &= \mathbf{t}^{(s)} \left( \tilde{\mathbf{K}}^{(s)} + \tilde{\mathcal{M}}^{(s,\bar{s})} \tilde{\mathbf{S}}_{p_G}^{(s)} \tilde{\mathcal{M}}^{(s,\bar{s})T} \right)^{-1} \mathbf{t}^{(s)T}
 \end{aligned}$$

où  $\tilde{\mathbf{K}}^{(s)} = \mathbf{K}^{(s)} + \mathbf{t}^{(s)T} \mathbf{K}_{bb}^{(s)} \mathbf{t}^{(s)}$  est une matrice creuse de même profil que  $\mathbf{K}^{(s)}$  et  $\tilde{\mathcal{M}}^{(s,\bar{s})} = \mathbf{t}^{(s)T} \mathcal{M}^{(s,\bar{s})}$ . En utilisant la formule de Sherman-Morrison-Woodbury, l'expression de  $\mathbf{S}_m^{(s)}$  devient :

$$\left( \mathbf{S}_p^{(s)} + \mathbf{Q}_b^{(s)} \right)^{-1} = \mathbf{t}^{(s)} \left( \tilde{\mathbf{K}}^{(s)-1} - \tilde{\mathbf{K}}^{(s)-1} \tilde{\mathcal{M}}^{(s,\bar{s})} \left[ \tilde{\mathbf{S}}_{p_G}^{(s)-1} + \tilde{\mathcal{M}}^{(s,\bar{s})T} \tilde{\mathbf{K}}^{(s)-1} \tilde{\mathcal{M}}^{(s,\bar{s})} \right]^{-1} \tilde{\mathcal{M}}^{(s,\bar{s})} \tilde{\mathbf{K}}^{(s)-1} \right) \mathbf{t}^{(s)T}.$$

La factorisation de  $\tilde{\mathbf{K}}^{(s)}$  n'est pas plus coûteuse que celle de  $\mathbf{K}^{(s)}$  car ces deux matrices ont le même profil. L'ajout du terme grossier dans l'expression de  $\mathbf{Q}_b^{(s)}$  implique l'inversion de deux opérateurs de petite taille ainsi qu'une résolution locale à seconds membres multiples  $\tilde{\mathbf{K}}^{(s)-1} \tilde{\mathcal{M}}^{(s,\bar{s})}$  avec  $n_{G_b}$  vecteurs (pour rappel,  $n_{G_b}$  est le nombre de degrés de liberté sur l'interface grossière).

### 3.2.3 Résultats numériques

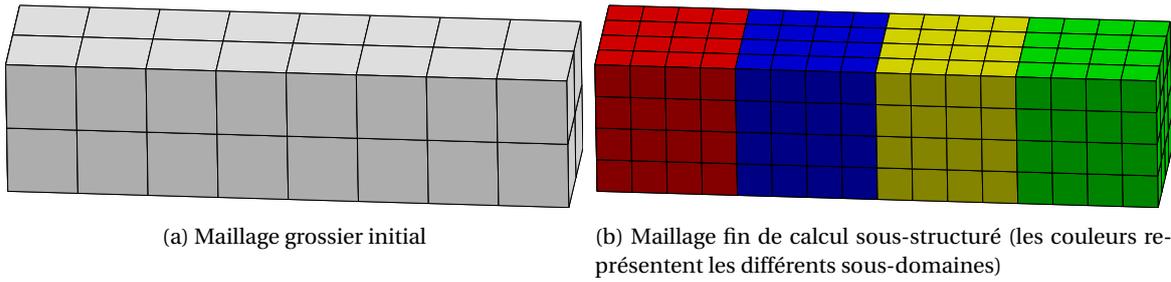


FIGURE 3.10 – Exemple d'application avec un niveau de raffinement entre les maillages fins et grossiers

L'efficacité des impédances choisies sera illustrée sur l'exemple poutre de la figure 3.10 avec les conditions limites de la figure 3.6 et un seul niveau de raffinement entre le maillage grossier et le maillage fin. Cet exemple comporte 4 sous-domaines de 64 éléments chacun. Comme précédemment, le cas étudié est linéaire et le matériau élastique isotrope et homogène. Les notations suivantes seront utilisées dans cette section :

$$\begin{aligned}
 \tilde{\mathbf{S}}_{p_1}^{(s)} &= \mathbf{K}_{bb}^{(s)} \\
 \tilde{\mathbf{S}}_p^{(s)} &= \mathbf{K}_{bb}^{(s)} - \mathcal{M}^{(s,\bar{s})} \mathbf{K}_{G_{bi}}^{(s)} \mathbf{K}_{G_{ii}}^{(s)-1} \mathbf{K}_{G_{ib}}^{(s)} \mathcal{M}^{(s,\bar{s})T} \\
 \tilde{\mathbf{S}}_{p_{neg}}^{(s)} &= \mathbf{K}_{bb}^{(s)} + \mathcal{M}^{(s,\bar{s})} \left( \mathbf{S}_{p_G}^{(s)} - \mathcal{J}^{(s,\bar{s})T} \mathbf{K}_{bb}^{(s)} \mathcal{J}^{(s,\bar{s})} \right) \mathcal{M}^{(s,\bar{s})T}
 \end{aligned}$$

Le problème résolu est celui de l'équation (3.1), qui pour rappel, prend cette forme :

$$\left( \mathbf{A}^{\diamond T} \mathbf{A}^{\diamond} \mathbf{S}_p^{\diamond} + \mathbf{Q}_b^{\diamond} \mathbf{B}^{\diamond T} \mathbf{B}^{\diamond} \right) \mathbf{S}_m^{\diamond} \boldsymbol{\mu}_b^{\diamond} = \left( \mathbf{A}^{\diamond T} \mathbf{A}^{\diamond} \mathbf{Q}_b^{\diamond} - \mathbf{Q}_b^{\diamond} \mathbf{B}^{\diamond T} \mathbf{B}^{\diamond} \right) \mathbf{b}_m^{\diamond},$$

où l'opérateur  $\mathbf{Q}_b^{(s)}$  sera choisi parmi les trois approximations exprimées plus haut :  $\tilde{\mathbf{S}}_{p_1}^{(s)}$ ,  $\tilde{\mathbf{S}}_p^{(s)}$  et  $\tilde{\mathbf{S}}_{p_{neg}}^{(s)}$ . Ce problème va être résolu par un algorithme *GMRes* qui va permettre de minimiser le résidu exprimé par l'équation (3.2) et qui est équivalente, à l'itération  $j$ , à l'expression :

$$\mathbf{r}_j^{\diamond} = \mathbf{A}^{\diamond T} \mathbf{A}^{\diamond} \boldsymbol{\lambda}_{b,j}^{\diamond} + \mathbf{Q}_b^{\diamond} \mathbf{B}^{\diamond T} \mathbf{B}^{\diamond} \mathbf{u}_{b,j}^{\diamond},$$

représentant une combinaison linéaire entre le saut de déplacement  $\mathbf{B}^\diamond \mathbf{u}_{b,j}^\diamond$  et le déséquilibre des inter-efforts  $\mathbf{A}^\diamond \lambda_{b,j}^\diamond$ . Le critère de convergence utilisé est un critère absolu sur la norme du résidu, l'algorithme sera stoppé lorsque :

$$\|\mathbf{r}_j^\diamond\| = \sqrt{\mathbf{r}_j^\diamond \mathbf{T} \mathbf{r}_j^\diamond} < \varepsilon,$$

avec  $\varepsilon$  un réel strictement positif. Dans les exemples traités ici,  $\varepsilon = 10^{-6}$ . Il faut noter que l'utilisa-

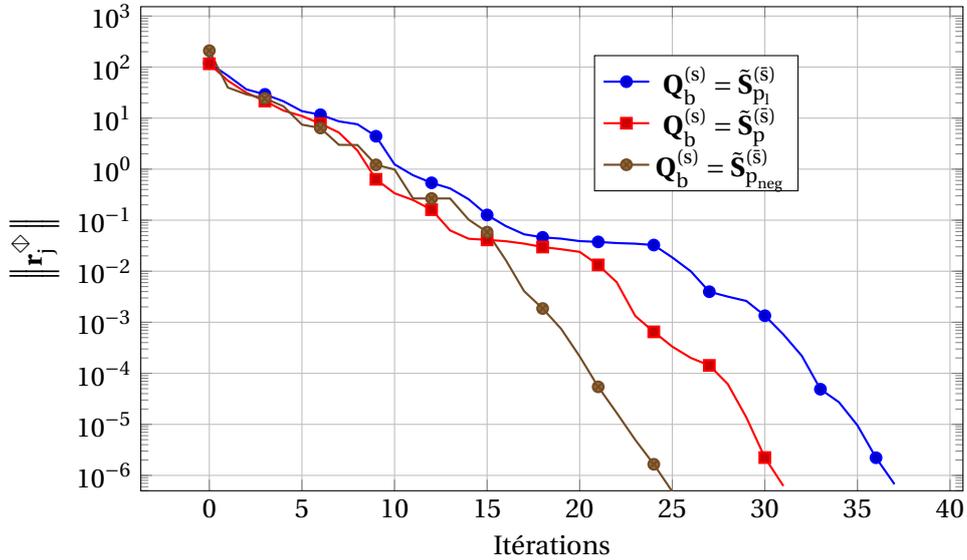


FIGURE 3.11 – Nombre d'itérations à convergence de la méthode mixte avec un solveur *GMRes* et différentes expressions pour l'impédance d'interface.

tion des impédances  $\tilde{\mathbf{S}}_p^{(s)}$  et  $\tilde{\mathbf{S}}_{pneg}^{(s)}$  conduit à une convergence instantanée dès l'initialisation si les maillages fins sont confondus avec le maillage grossier.

La figure 3.11 montre la convergence de la méthode pour les différentes impédances choisies avec une initialisation qualifiée d'optimale (voir équation (3.9) pour son expression). Le nombre d'itérations à convergence permis par l'approximation *lumped*  $\tilde{\mathbf{S}}_{p1}^{(s)}$  est de 37. Le fait d'ajouter le

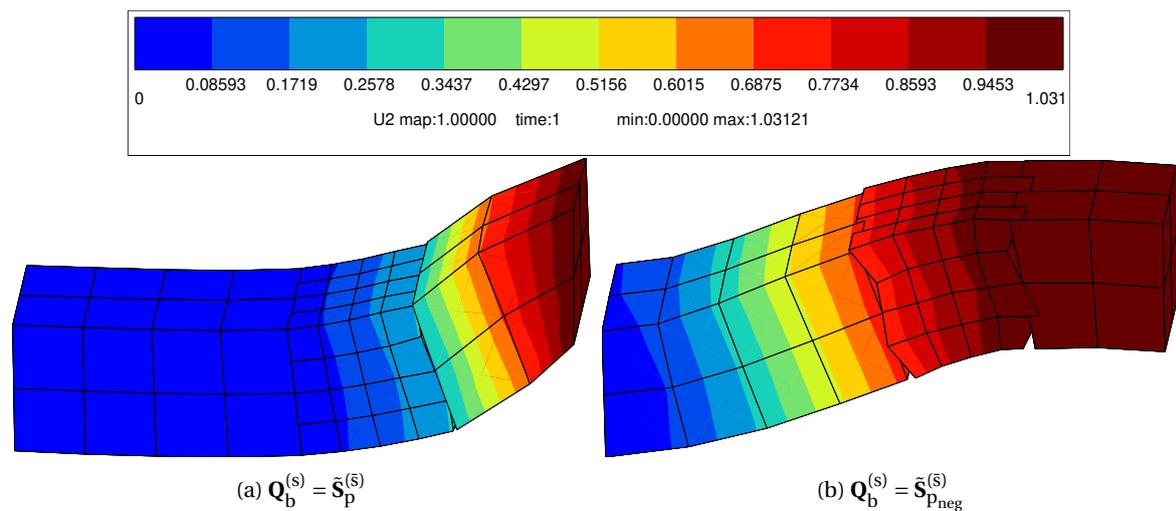


FIGURE 3.12 – Solution selon la direction  $y$ , tracée pour l'initialisation sur les maillages fins et grossiers du sous-domaine numéro 3 (celui en jaune sur la figure 3.10), pour différentes impédances  $\mathbf{Q}_b^{(s)}$

comportement plus lointain dans l'expression de l'impédance d'interface permet de gagner en nombre d'itérations. En effet, l'utilisation de  $\tilde{\mathbf{S}}_p^{(s)}$  permet au solveur de converger en 31 itérations. Cependant, le gain reste faible en raison de la non transmission des conditions limites au travers

du sous-domaine. En effet, la figure 3.12a montre que le maillage grossier de la partie à gauche, proche de l'encastrement semble avoir un comportement beaucoup plus rigide qu'en réalité. Ce problème est résolu avec l'utilisation de la seconde impédance à deux niveaux  $\tilde{\mathbf{S}}_{\text{pneg}}^{(s)}$ , où le comportement global de la structure est bien mieux retranscrit (voir figure 3.12b) et fait encore gagner en termes d'itérations à convergence : seulement 25 itérations sont nécessaires. La figure 3.13 montre

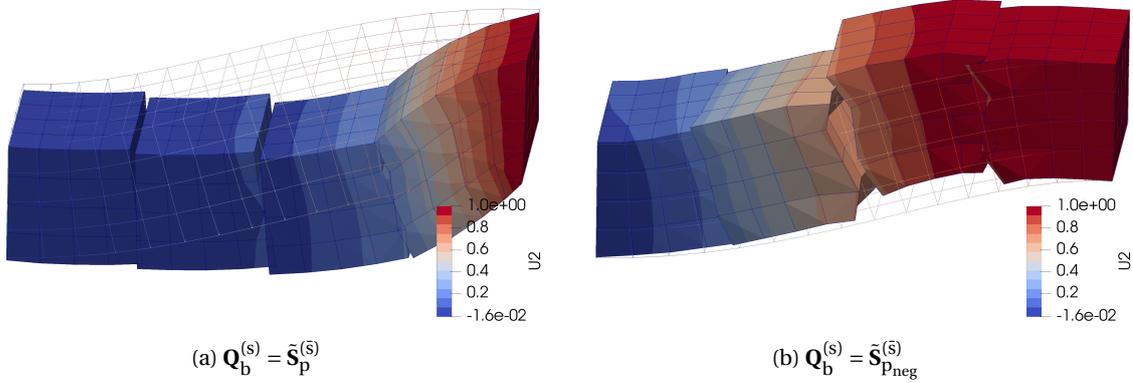


FIGURE 3.13 – Solution selon la direction  $y$ , tracée pour l'initialisation sur les maillages fins de tous les sous-domaines, pour différentes impédances  $\mathbf{Q}_b^{(s)}$ . La solution représentée en filaire est la solution de référence.

aussi que l'impédance  $\tilde{\mathbf{S}}_{\text{pneg}}^{(s)}$  permet, dès l'initialisation, d'obtenir une solution bien plus proche de la solution de référence en comparaison de l'impédance  $\tilde{\mathbf{S}}_p^{(s)}$ .

### 3.3 Formulations mixtes avancées

La formulation « classique » de *FETI-2LM* (equation (1.17)) peut aussi être modifiée; il est possible d'y adjoindre des notions issues de la théorie du preconditionnement des approches primale et duale. Cela permet d'ajouter des paramètres à la méthode pour en optimiser la convergence. Les développements de cette section sont des pistes de réflexion dont une réalisation pratique n'a pas pu être réalisée pour le moment.

#### 3.3.1 FETI-2LM généralisée et introduction d'une seconde impédance d'interface

##### 3.3.1.1 Mise en équation

Le paramètre  $\mathbf{Q}_b^\diamond$  est introduit à la fois dans la définition de l'inconnue mixte  $\mu_b^\diamond$  et dans la définition du résidu de l'équation (1.16) (dans la suite, ce résidu sera également appelé « saut à l'interface »). Or, dans ces deux équations, il ne joue pas le même rôle. Il est alors possible de les définir différemment dans ces deux équations. Pour rappel, le résidu (1.16) fait intervenir les deux opérateurs d'assemblage  $\mathbf{A}^\diamond$  et  $\mathbf{B}^\diamond$  ainsi que l'opérateur de rigidité d'interface  $\mathbf{Q}_b^\diamond$ . Il faut noter que satisfaire l'équation (1.16) revient à satisfaire les conditions d'équilibre et de continuité à l'interface. Dans cette section, on propose d'introduire un nouveau résidu qui permettrait d'améliorer la convergence en améliorant le conditionnement de l'opérateur global. Ainsi, il est possible d'utiliser les opérateurs de *scaling* utilisés dans le preconditionnement des approches primale et duale. Pour rappel, ces opérateurs primal  $\tilde{\mathbf{A}}^\diamond$  et dual  $\tilde{\mathbf{B}}^\diamond$  se mettent sous la forme :

$$\begin{aligned}\tilde{\mathbf{A}}^\diamond &= \left( \mathbf{A}^\diamond \mathbf{M}^\diamond \mathbf{A}^{\diamond T} \right)^{-1} \mathbf{A}^\diamond \mathbf{M}^\diamond, \\ \tilde{\mathbf{B}}^\diamond &= \left( \mathbf{B}^\diamond \mathbf{M}^{\diamond -1} \mathbf{B}^{\diamond T} \right)^+ \mathbf{B}^\diamond \mathbf{M}^{\diamond -1}.\end{aligned}$$

Avec cette forme, la propriété d'orthogonalité des opérateurs d'assemblage suivante :

$$\tilde{\mathbf{B}}^\diamond \tilde{\mathbf{A}}^{\diamond T} = 0 = \tilde{\mathbf{A}}^\diamond \tilde{\mathbf{B}}^{\diamond T}, \quad (3.16)$$

est conservée. Les opérateurs  $\mathbf{M}^\diamond$  entrant en jeu dans la définition de  $\tilde{\mathbf{A}}^\diamond$  et  $\tilde{\mathbf{B}}^\diamond$  peuvent être différents mais la condition d'orthogonalité (3.16) est perdue.

En plus de l'utilisation des ces opérateurs de *scaling*, il est possible de modifier l'opérateur homogène à une raideur par une seconde impédance d'interface, notée  $\mathbf{X}_b^\diamond$ . Ainsi, l'introduction d'une nouvelle condition de saut d'interface peut être obtenue de la manière suivante :

$$\tilde{\mathbf{A}}^{\diamond T} \mathbf{A}^\diamond \boldsymbol{\lambda}_b^\diamond + \mathbf{X}_b^\diamond \tilde{\mathbf{B}}^{\diamond T} \mathbf{B}^\diamond \mathbf{u}_b^\diamond = 0. \quad (3.17)$$

Cette équation est bien équivalente aux conditions de continuité des déplacements d'interface (1.6a) et d'équilibre des inter-efforts (1.6b). En effet, si on multiplie cette équation à gauche par  $\tilde{\mathbf{B}}^\diamond$ , on obtient bien :

$$\begin{aligned} \tilde{\mathbf{B}}^\diamond \left[ \tilde{\mathbf{A}}^{\diamond T} \mathbf{A}^\diamond \boldsymbol{\lambda}_b^\diamond + \mathbf{X}_b^\diamond \tilde{\mathbf{B}}^{\diamond T} \mathbf{B}^\diamond \mathbf{u}_b^\diamond \right] &= \overbrace{\tilde{\mathbf{B}}^\diamond \tilde{\mathbf{A}}^{\diamond T} \mathbf{A}^\diamond \boldsymbol{\lambda}_b^\diamond}^{=0} + \tilde{\mathbf{B}}^\diamond \mathbf{X}_b^\diamond \tilde{\mathbf{B}}^{\diamond T} \mathbf{B}^\diamond \mathbf{u}_b^\diamond \\ &= \tilde{\mathbf{B}}^\diamond \mathbf{X}_b^\diamond \tilde{\mathbf{B}}^{\diamond T} \mathbf{B}^\diamond \mathbf{u}_b^\diamond = 0, \\ &\Leftrightarrow \mathbf{B}^\diamond \mathbf{u}_b^\diamond = 0, \end{aligned}$$

qui correspond à la continuité des déplacements d'interface sous réserve de l'inversibilité de  $\tilde{\mathbf{B}}^\diamond \mathbf{X}_b^\diamond \tilde{\mathbf{B}}^{\diamond T}$  dans  $\text{Im}(\mathbf{B}^\diamond)$ . Et de l'équation (3.17), il ne reste plus que :

$$\begin{aligned} \tilde{\mathbf{A}}^{\diamond T} \mathbf{A}^\diamond \boldsymbol{\lambda}_b^\diamond &= 0 \\ \Rightarrow \mathbf{A}^\diamond \tilde{\mathbf{A}}^{\diamond T} \mathbf{A}^\diamond \boldsymbol{\lambda}_b^\diamond &= 0 \\ \Rightarrow \mathbf{A}^\diamond \boldsymbol{\lambda}_b^\diamond &= 0, \end{aligned}$$

qui permet de conclure quant à l'équivalence entre les équations (1.6a), (1.6b), et l'équation (3.17).

En suivant une démarche analogue à celle suivie dans la section 1.1.2.4, un autre problème d'interface peut être écrit :

$$\left[ \tilde{\mathbf{A}}^{\diamond T} \mathbf{A}^\diamond - \left( \tilde{\mathbf{A}}^{\diamond T} \mathbf{A}^\diamond \mathbf{Q}_b^\diamond - \mathbf{X}_b^\diamond \tilde{\mathbf{B}}^{\diamond T} \mathbf{B}^\diamond \right) \mathbf{S}_m^\diamond \right] \boldsymbol{\mu}_b^\diamond = \left( \tilde{\mathbf{A}}^{\diamond T} \mathbf{A}^\diamond \mathbf{Q}_b^\diamond - \mathbf{X}_b^\diamond \tilde{\mathbf{B}}^{\diamond T} \mathbf{B}^\diamond \right) \mathbf{b}_m^\diamond,$$

puis être simplifié de la manière suivante :

$$\left( \tilde{\mathbf{A}}^{\diamond T} \mathbf{A}^\diamond \mathbf{S}_p^\diamond + \mathbf{X}_b^\diamond \tilde{\mathbf{B}}^{\diamond T} \mathbf{B}^\diamond \right) \mathbf{S}_m^\diamond \boldsymbol{\mu}_b^\diamond = \left( \tilde{\mathbf{A}}^{\diamond T} \mathbf{A}^\diamond \mathbf{Q}_b^\diamond - \mathbf{X}_b^\diamond \tilde{\mathbf{B}}^{\diamond T} \mathbf{B}^\diamond \right) \mathbf{b}_m^\diamond. \quad (3.18)$$

Ce type d'approche rappelle les méthodes mixtes à deux paramètres (directions de montée et de descente), ce type d'approche est suivie par la méthode *LaTin* de LADEVÈZE [1985]; LADEVÈZE et DUREISSEIX [1999] par exemple. De la même façon que ce type de méthodes, il serait intéressant que le rayon spectral de  $\mathbf{I}^\diamond - \left( \tilde{\mathbf{A}}^{\diamond T} \mathbf{A}^\diamond \mathbf{S}_p^\diamond + \mathbf{X}_b^\diamond \tilde{\mathbf{B}}^{\diamond T} \mathbf{B}^\diamond \right) \mathbf{S}_m^\diamond$  soit le plus petit possible pour permettre l'utilisation de solveur de type point fixe, exploitant la stationnarité (et permettant aussi d'envisager l'exploitation de solveurs asynchrones). Par analogie avec le préconditionnement des approches primale et duale,  $\mathbf{X}_b^\diamond$  étant introduit uniquement dans l'expression du résidu (3.17), il peut être interprété comme une « impédance de préconditionnement ». Ce choix de résidu ne modifie pas la solution  $\boldsymbol{\mu}_b^\diamond$  à convergence mais modifie fortement le comportement itératif du solveur; il sert simplement à faire le lien entre des déplacements d'interface et des inter-efforts, permettant de lier les résidus d'interface (saut de déplacement, déséquilibre des inter-efforts) plus ou moins fortement. De la même manière que l'impédance d'interface et le préconditionneur de *BDD* et *FETI*, cette seconde impédance d'interface doit représenter un faible coût de calcul.

### 3.3.1.2 Étude spectrale et estimation d'une bonne impédance

Dans cette section, l'intérêt est porté sur le spectre de l'opérateur non-symétrique  $\left(\tilde{\mathbf{A}}^{\diamond T} \mathbf{A}^{\diamond} \mathbf{S}_p^{\diamond} + \mathbf{X}_b^{\diamond} \tilde{\mathbf{B}}^{\diamond T} \mathbf{B}^{\diamond}\right) \mathbf{S}_m^{\diamond}$  qui permet d'avoir une idée du conditionnement de l'opérateur<sup>2</sup>.

Dans un premier temps, on peut remarquer que si  $\mathbf{X}_b^{\diamond}$  reste constant alors, en choisissant un  $\mathbf{Q}_b^{\diamond}$  suffisamment rigide (*i.e.* que les valeurs propres de  $\mathbf{S}_m^{\diamond}$  sont très faibles), il est possible de réduire le rayon spectral de l'opérateur en deçà de 1, permettant d'envisager l'utilisation de solveurs fondés sur la stationnarité.

#### Remarque 3.3

L'étude de ce système ne semble pas évidente. Néanmoins, il est possible d'observer qu'en recherchant une condition pour que l'opérateur approxime l'identité, une condition sur  $\mathbf{X}_b^{\diamond}$  peut apparaître :

$$\begin{aligned} & \left(\tilde{\mathbf{A}}^{\diamond T} \mathbf{A}^{\diamond} \mathbf{S}_p^{\diamond} + \mathbf{X}_b^{\diamond} \tilde{\mathbf{B}}^{\diamond T} \mathbf{B}^{\diamond}\right) \mathbf{S}_m^{\diamond} \simeq \mathbf{1}^{\diamond} \\ \Leftrightarrow & \left(\tilde{\mathbf{A}}^{\diamond T} \mathbf{A}^{\diamond} \mathbf{S}_p^{\diamond} + \mathbf{X}_b^{\diamond} \tilde{\mathbf{B}}^{\diamond T} \mathbf{B}^{\diamond}\right) \simeq \mathbf{S}_p^{\diamond} + \mathbf{Q}_b^{\diamond} \\ \Rightarrow & \mathbf{B}^{\diamond} \mathbf{X}_b^{\diamond} \tilde{\mathbf{B}}^{\diamond T} \mathbf{B}^{\diamond} \simeq \mathbf{B}^{\diamond} \left(\mathbf{S}_p^{\diamond} + \mathbf{Q}_b^{\diamond}\right) \\ \Rightarrow & \mathbf{B}^{\diamond} \mathbf{X}_b^{\diamond} \tilde{\mathbf{B}}^{\diamond T} \simeq \mathbf{B}^{\diamond} \left(\mathbf{S}_p^{\diamond} + \mathbf{Q}_b^{\diamond}\right) \tilde{\mathbf{B}}^{\diamond T} \end{aligned}$$

Finalement, si  $\mathbf{X}_b^{\diamond} = \mathbf{S}_p^{\diamond} + \mathbf{Q}_b^{\diamond}$ , alors, l'égalité est vérifiée et la matrice se rapproche de l'identité. Ce résultat permet de dire que si  $\boldsymbol{\mu}_b^{\diamond} \in \text{Im}\left(\left(\mathbf{S}_p^{\diamond} + \mathbf{Q}_b^{\diamond}\right) \tilde{\mathbf{B}}^{\diamond T}\right)$  alors  $\left(\tilde{\mathbf{A}}^{\diamond T} \mathbf{A}^{\diamond} \mathbf{S}_p^{\diamond} + \mathbf{X}_b^{\diamond} \tilde{\mathbf{B}}^{\diamond T} \mathbf{B}^{\diamond}\right) \mathbf{S}_m^{\diamond} \boldsymbol{\mu}_b^{\diamond} = \boldsymbol{\mu}_b^{\diamond}$ .

Pour retrouver cela de manière plus rigoureuse, on réalise une étude aux valeurs propres du système. L'objectif est donc de trouver les valeurs propres  $\alpha$  et les vecteurs propres  $\boldsymbol{\mu}_e^{\diamond}$  associés tels que :

$$\left(\tilde{\mathbf{A}}^{\diamond T} \mathbf{A}^{\diamond} \mathbf{S}_p^{\diamond} + \mathbf{X}_b^{\diamond} \tilde{\mathbf{B}}^{\diamond T} \mathbf{B}^{\diamond}\right) \mathbf{S}_m^{\diamond} \boldsymbol{\mu}_e^{\diamond} = \alpha \boldsymbol{\mu}_e^{\diamond}.$$

Pour avoir la convergence la plus rapide possible il faut que l'écart entre la plus petite et la plus grande valeur propre de cet opérateur soit minimal. L'objectif est donc de voir quelles conditions sur  $\mathbf{X}_b$  rendent cet écart minimal. En posant le changement de variable  $\mathbf{y}_e^{\diamond} = \mathbf{S}_m^{\diamond} \boldsymbol{\mu}_e^{\diamond}$ , un problème aux valeurs propres généralisé peut être formulé sur les nouveaux vecteurs propres  $\mathbf{y}_e^{\diamond}$  :

$$\left(\tilde{\mathbf{A}}^{\diamond T} \mathbf{A}^{\diamond} \mathbf{S}_p^{\diamond} + \mathbf{X}_b^{\diamond} \tilde{\mathbf{B}}^{\diamond T} \mathbf{B}^{\diamond}\right) \mathbf{y}_e^{\diamond} = \alpha \mathbf{S}_m^{\diamond -1} \mathbf{y}_e^{\diamond}. \quad (3.19)$$

De plus, tout vecteur de déplacement défini sur l'interface locale peut s'écrire comme une somme d'un champ continu ainsi que d'un champ équilibré au sens des opérateurs de pondération.

$$\mathbf{y}_e^{\diamond} = \mathbf{S}_m^{\diamond} \tilde{\mathbf{A}}^{\diamond T} \mathbf{y}_A + \tilde{\mathbf{B}}^{\diamond T} \mathbf{y}_B,$$

où  $\mathbf{y}_A$  et  $\mathbf{y}_B$  sont des vecteurs uniques définis respectivement sur les interfaces globales  $\Gamma_A$  et  $\Gamma_B$ .

2. Le conditionnement d'un opérateur non-symétrique est borné par le rapport des valeurs propres maximale et minimale multipliée par le conditionnement de la matrice des vecteurs propres (BENZI et coll. [2005], section 9.2)

En multipliant l'équation (3.19) à gauche par  $\tilde{\mathbf{A}}^\diamond \mathbf{X}_b^{\diamond -1}$  et par  $\tilde{\mathbf{B}}^\diamond$ , deux égalités sont obtenues :

$$\begin{cases} \tilde{\mathbf{A}}^\diamond \mathbf{X}_b^{\diamond -1} \tilde{\mathbf{A}}^{\diamond T} \mathbf{A}^\diamond \mathbf{S}_p^\diamond \mathbf{y}_e^\diamond &= \alpha \tilde{\mathbf{A}}^\diamond \mathbf{X}_b^{\diamond -1} \mathbf{S}_m^{\diamond -1} \mathbf{y}_e^\diamond \\ \tilde{\mathbf{B}}^\diamond \mathbf{X}_b^\diamond \tilde{\mathbf{B}}^{\diamond T} \mathbf{B}^\diamond \mathbf{y}_e^\diamond &= \alpha \tilde{\mathbf{B}}^\diamond \mathbf{S}_m^{\diamond -1} \mathbf{y}_e^\diamond \end{cases},$$

$$\begin{cases} \tilde{\mathbf{A}}^\diamond \mathbf{X}_b^{\diamond -1} \tilde{\mathbf{A}}^{\diamond T} \mathbf{A}^\diamond \mathbf{S}_p^\diamond \left( \mathbf{S}_m^\diamond \tilde{\mathbf{A}}^{\diamond T} \mathbf{y}_A + \tilde{\mathbf{B}}^{\diamond T} \mathbf{y}_B \right) &= \alpha \tilde{\mathbf{A}}^\diamond \mathbf{X}_b^{\diamond -1} \tilde{\mathbf{A}}^{\diamond T} \mathbf{y}_A + \alpha \tilde{\mathbf{A}}^\diamond \mathbf{X}_b^{\diamond -1} \mathbf{S}_m^{\diamond -1} \tilde{\mathbf{B}}^{\diamond T} \mathbf{y}_B \\ \tilde{\mathbf{B}}^\diamond \mathbf{X}_b^\diamond \tilde{\mathbf{B}}^{\diamond T} \mathbf{B}^\diamond \mathbf{S}_m^\diamond \tilde{\mathbf{A}}^{\diamond T} \mathbf{y}_A + \tilde{\mathbf{B}}^\diamond \mathbf{X}_b^\diamond \tilde{\mathbf{B}}^{\diamond T} \mathbf{y}_B &= \alpha \tilde{\mathbf{B}}^\diamond \mathbf{S}_m^{\diamond -1} \tilde{\mathbf{B}}^{\diamond T} \mathbf{y}_B \end{cases}.$$

Pour simplifier l'étude de cet opérateur, on choisit en tant que valeur pour  $\mathbf{X}_b^\diamond$ , la valeur intuitive dans la remarque 3.3 et étant  $\mathbf{X}_b^\diamond = \mathbf{S}_p^\diamond + \mathbf{Q}_b^\diamond = \mathbf{S}_m^{\diamond -1}$ . Avec ce choix, on peut écrire :

$$\begin{cases} \mathbf{A}^\diamond \mathbf{S}_p^\diamond \left( \mathbf{S}_m^\diamond \tilde{\mathbf{A}}^{\diamond T} \mathbf{y}_A + \tilde{\mathbf{B}}^{\diamond T} \mathbf{y}_B \right) &= \alpha \mathbf{y}_A \\ \mathbf{B}^\diamond \mathbf{S}_m^\diamond \tilde{\mathbf{A}}^{\diamond T} \mathbf{y}_A + \mathbf{y}_B &= \alpha \mathbf{y}_B \end{cases}.$$

En injectant l'expression de  $\mathbf{y}_B$  dans la première équation, on arrive à l'équation posée uniquement en  $\mathbf{y}_A$  suivante :

$$\left[ (\alpha - 1) \mathbf{A}^\diamond \mathbf{S}_p^\diamond \mathbf{S}_m^\diamond \tilde{\mathbf{A}}^{\diamond T} + \mathbf{A}^\diamond \mathbf{S}_p^\diamond \tilde{\mathbf{B}}^{\diamond T} \mathbf{B}^\diamond \mathbf{S}_m^\diamond \tilde{\mathbf{A}}^{\diamond T} \right] \mathbf{y}_A = \alpha(\alpha - 1) \mathbf{y}_A$$

et avec l'équation (1.21) ( $\tilde{\mathbf{B}}^{\diamond T} \mathbf{B}^\diamond = \mathbf{1}^\diamond - \tilde{\mathbf{A}}^{\diamond T} \mathbf{A}^\diamond$ ) de complémentarité des opérateurs de *scaling*, cette équation peut se mettre sous la forme :

$$\begin{aligned} &\left[ \alpha \mathbf{A}^\diamond \mathbf{S}_p^\diamond \mathbf{S}_m^\diamond \tilde{\mathbf{A}}^{\diamond T} - \mathbf{A}^\diamond \mathbf{S}_p^\diamond \mathbf{A}^\diamond \tilde{\mathbf{A}}^{\diamond T} \mathbf{S}_m^\diamond \tilde{\mathbf{A}}^{\diamond T} \right] \mathbf{y}_A = \alpha(\alpha - 1) \mathbf{y}_A \\ \Leftrightarrow &\mathbf{A}^\diamond \mathbf{S}_p^\diamond \left( \alpha \mathbf{1}^\diamond - \mathbf{A}^{\diamond T} \tilde{\mathbf{A}}^\diamond \right) \mathbf{S}_m^\diamond \tilde{\mathbf{A}}^{\diamond T} \mathbf{y}_A = \alpha(\alpha - 1) \mathbf{y}_A \end{aligned}$$

#### Remarque 3.4

Cette équation est un problème aux valeurs propres du second ordre, qui n'est pas symétrique et admet comme solution des valeurs propres complexes. Sa résolution est difficile à réaliser, c'est pourquoi on se propose d'ajouter des hypothèses au problème. Si on pouvait choisir les matrices  $\mathbf{M}^\diamond$  comme étant égale à  $\mathbf{S}_m^{\diamond -1} = \mathbf{S}_p^\diamond + \mathbf{Q}^\diamond$ , le problème aux valeurs propre se simplifie de la manière suivante :

$$\begin{cases} 0 &= (\alpha - 1) \mathbf{y}_B \\ \mathbf{A}^\diamond \mathbf{S}_p^\diamond \mathbf{A}^\diamond \tilde{\mathbf{A}}^{\diamond T} \left( \mathbf{A}^\diamond \left( \mathbf{S}_p^\diamond + \mathbf{Q}_b^\diamond \right) \mathbf{A}^{\diamond T} \right)^{-1} \mathbf{y}_A &= \alpha \mathbf{y}_A \end{cases}.$$

Dans ces hypothèses, on peut montrer que ces équations admettent des solution lorsque  $\alpha = 1$  ou  $0 < \alpha < 1$ .

Une illustration du spectre des opérateurs mis en jeu dans ces formulations mixtes est donnée figure 3.15. Un solveur *GMRes* permet la résolution en parallèle de l'exemple de la figure 3.14, dont les valeurs de Ritz à convergence permettent d'estimer les valeurs propres de l'opérateur. Les systèmes résolus sont les systèmes (3.1) et (3.18). Ce problème est un cube de côté 1 mm dont une face est encadrée et un déplacement normal à la face opposé est imposé avec une amplitude de 1 mm. Ce cube est constitué d'un matériau homogène identique à celui de la figure 3.1. La disposition de ces sous-domaines fait apparaître des points multiples (douze points de multiplicité 3 et un point de multiplicité 4).

Pour le système (3.1), le rayon spectral de l'opérateur est d'environ 3.5 alors que celui de l'équation (3.18) est de l'ordre de 2. Le seul fait d'utiliser les opérateurs de *scaling*  $\tilde{\mathbf{A}}^\diamond$  et  $\tilde{\mathbf{B}}^\diamond$  à la place des opérateurs d'assemblage classiques  $\mathbf{A}^\diamond$  et  $\mathbf{B}^\diamond$  permet de réduire le rayon spectral ainsi que l'amplitudes des parties imaginaires des valeurs propres complexes. Par contre, l'utilisation de l'impédance de preconditionnement comme étant celle intuitée dans la remarque 3.3 (à savoir  $\mathbf{X}_b^\diamond = \mathbf{S}_p^\diamond + \mathbf{Q}_b^\diamond$ ) regroupe les valeurs propres à partie imaginaire non nulles proche de la même partie réelle ( $\approx 2$ ) pour ce cas test.

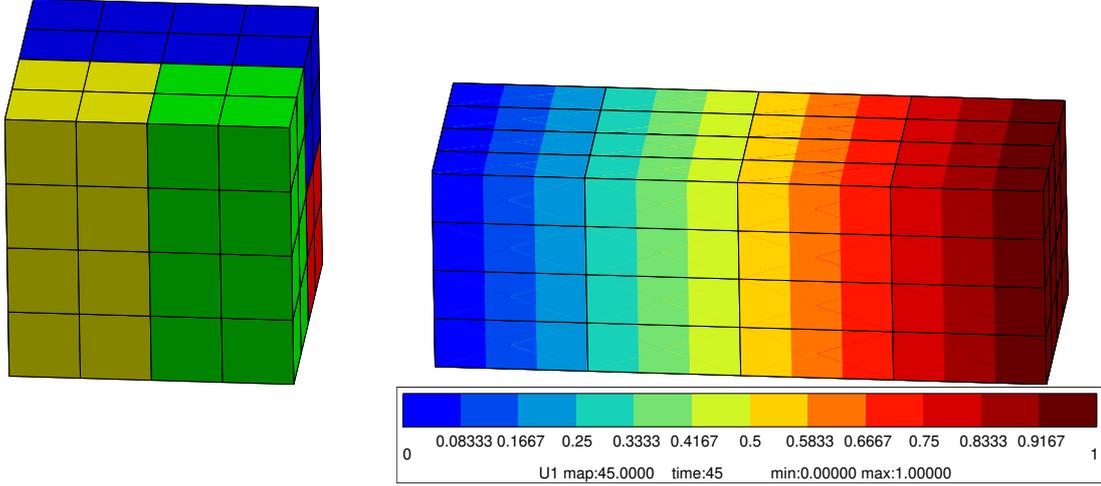


FIGURE 3.14 – Cube homogène constitué de quatre sous-domaines auquel un chargement de traction est appliqué. La figure de gauche représente la structure non déformée dont les couleurs permettent de distinguer les différents sous-domaines. La figure de droite représente la solution de référence déformée.

### 3.3.2 Minimisation des sauts à l'interface

L'objectif d'un solveur mixte est de minimiser à la fois le saut de déplacement et le déséquilibre des inter-efforts au cours des itérations. Cela revient à minimiser une fonctionnelle  $\mathcal{L}$ , fonction uniquement de  $\boldsymbol{\mu}_b^\diamond$ , de la forme :

$$\mathcal{L}(\boldsymbol{\mu}_b^\diamond) = \left\| \mathbf{B}^\diamond \mathbf{u}_b^\diamond \right\|_Z^2 + \left\| \mathbf{A}^\diamond \boldsymbol{\lambda}_b^\diamond \right\|_Y^2, \quad (3.20)$$

avec  $\mathbf{Y}$  et  $\mathbf{Z}$  des opérateurs de norme sur les interfaces sont choisis comme :

$$\mathbf{Y} = \left( \mathbf{A}^\diamond \mathbf{M}^\diamond \mathbf{A}^{\diamond T} \right)^{-1} \quad ; \quad \mathbf{Z} = \left( \mathbf{B}^\diamond \mathbf{M}^{\diamond -1} \mathbf{B}^{\diamond T} \right)^+.$$

La fonctionnelle de l'équation (3.20) peut être exprimée uniquement en fonction de  $\boldsymbol{\mu}_b^\diamond$  en réinjectant les expressions de  $\mathbf{u}_b^\diamond$  et  $\boldsymbol{\lambda}_b^\diamond$  :

$$\begin{aligned} \mathcal{L}(\boldsymbol{\mu}_b^\diamond) = & \left( \mathbf{S}_m^\diamond \boldsymbol{\mu}_b^\diamond + \mathbf{b}_m \right)^T \mathbf{B}^{\diamond T} \mathbf{Z} \mathbf{B}^\diamond \left( \mathbf{S}_m^\diamond \boldsymbol{\mu}_b^\diamond + \mathbf{b}_m \right) + \\ & \left( \mathbf{S}_p^\diamond \mathbf{S}_m^\diamond \boldsymbol{\mu}_b^\diamond - \mathbf{Q}^\diamond \mathbf{b}_m \right)^T \mathbf{A}^{\diamond T} \mathbf{Y} \mathbf{A}^\diamond \left( \mathbf{S}_p^\diamond \mathbf{S}_m^\diamond \boldsymbol{\mu}_b^\diamond - \mathbf{Q}^\diamond \mathbf{b}_m \right) \end{aligned}$$

En effectuant une minimisation de cette fonctionnelle par rapport à  $\boldsymbol{\mu}_b^\diamond$ , le résultat suivant apparaît :

$$d\boldsymbol{\mu}_b^{\diamond T} \left[ \mathbf{S}_m^\diamond \left( \mathbf{B}^{\diamond T} \mathbf{Z} \mathbf{B}^\diamond + \mathbf{S}_p^\diamond \mathbf{A}^{\diamond T} \mathbf{Y} \mathbf{A}^\diamond \mathbf{S}_p^\diamond \right) \mathbf{S}_m^\diamond \boldsymbol{\mu}_b^\diamond + \mathbf{S}_m^\diamond \left( \mathbf{B}^{\diamond T} \mathbf{Z} \mathbf{B}^\diamond - \mathbf{S}_p^\diamond \mathbf{A}^{\diamond T} \mathbf{Y} \mathbf{A}^\diamond \mathbf{Q}^\diamond \right) \mathbf{b}_m \right] = 0.$$

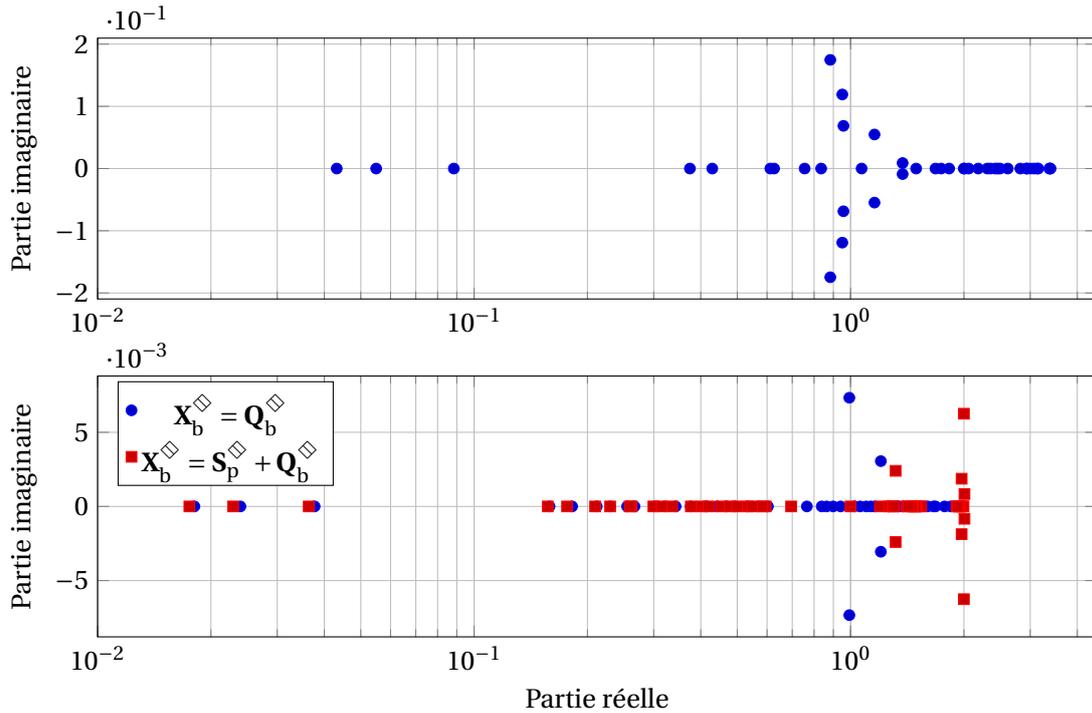


FIGURE 3.15 – Valeurs propres de l’opérateur de l’équation (3.1) (en haut) et de l’équation (3.18) (en bas) pour différentes valeurs de l’impédance de preconditionnement  $\mathbf{X}_b^\diamond$ , obtenues à convergence par les valeurs de Ritz d’un solveur de Krylov (*GMRes* ici). Le problème considéré est le problème illustré à la figure 3.14 avec une impédance d’interface  $\mathbf{Q}_b^{(s)} = \mathbf{K}_{bb}^{(s)}$ .

Ainsi, en raison de l’inversibilité de  $\mathbf{S}_m^\diamond$  le système à résoudre devient :

$$\left( \mathbf{S}_p^\diamond \mathbf{A}^\diamond \mathbf{Y} \mathbf{A}^\diamond \mathbf{S}_p^\diamond + \mathbf{B}^\diamond \mathbf{Z} \mathbf{B}^\diamond \right) \mathbf{S}_m^\diamond \boldsymbol{\mu}_b^\diamond = \left( \mathbf{S}_p^\diamond \mathbf{A}^\diamond \mathbf{Y} \mathbf{A}^\diamond \mathbf{Q}^\diamond - \mathbf{B}^\diamond \mathbf{Z} \mathbf{B}^\diamond \right) \mathbf{b}_m^\diamond$$

On remarque que l’opérateur de ce système est le produit de 2 opérateurs symétriques définis positifs. En utilisant les définitions de  $\mathbf{Y}$  et  $\mathbf{Z}$ , il est possible de montrer que  $\mathbf{B}^\diamond \mathbf{Z} \mathbf{B}^\diamond = \mathbf{M}^\diamond \tilde{\mathbf{B}}^\diamond \mathbf{B}^\diamond$  et  $\mathbf{A}^\diamond \mathbf{Y} \mathbf{A}^\diamond = \mathbf{M}^\diamond \tilde{\mathbf{A}}^\diamond \mathbf{A}^\diamond$ , et le système précédent peut être réécrit de la manière suivante :

$$\left( \mathbf{S}_p^\diamond \mathbf{M}^\diamond \tilde{\mathbf{A}}^\diamond \mathbf{A}^\diamond \mathbf{S}_p^\diamond + \mathbf{M}^\diamond \tilde{\mathbf{B}}^\diamond \mathbf{B}^\diamond \right) \mathbf{S}_m^\diamond \boldsymbol{\mu}_b^\diamond = \left( \mathbf{S}_p^\diamond \mathbf{M}^\diamond \tilde{\mathbf{A}}^\diamond \mathbf{A}^\diamond \mathbf{Q}^\diamond - \mathbf{M}^\diamond \tilde{\mathbf{B}}^\diamond \mathbf{B}^\diamond \right) \mathbf{b}_m^\diamond$$

ou encore :

$$\left( \mathbf{M}^\diamond \tilde{\mathbf{S}}_p^\diamond \mathbf{M}^\diamond \tilde{\mathbf{A}}^\diamond \mathbf{A}^\diamond \mathbf{S}_p^\diamond + \tilde{\mathbf{B}}^\diamond \mathbf{B}^\diamond \right) \mathbf{S}_m^\diamond \boldsymbol{\mu}_b^\diamond = \left( \mathbf{M}^\diamond \tilde{\mathbf{S}}_p^\diamond \mathbf{M}^\diamond \tilde{\mathbf{A}}^\diamond \mathbf{A}^\diamond \mathbf{Q}^\diamond - \tilde{\mathbf{B}}^\diamond \mathbf{B}^\diamond \right) \mathbf{b}_m^\diamond$$

Ainsi, en comparant cette expression avec la formulation généralisée de *FETI-2LM* de la section précédente, nous remarquons que ce nouveau système est un cas particulier de la généralisation (3.18) lorsque  $\mathbf{X}_b^{\diamond -1} = \mathbf{M}^\diamond \tilde{\mathbf{S}}_p^\diamond \mathbf{M}^\diamond$ . La formulation mixte ainsi obtenue conserve le caractère symétrique du problème, caractéristique assez rare dans le cas des formulations mixtes.

### 3.3.3 Ajout d’une contrainte supplémentaire

L’avantage de la méthode précédente, outre le fait de manipuler des opérateurs symétriques, est qu’il est possible d’adjoindre des conditions s’apparentant à un problème grossier dans le Lagrangien (3.20). Cela permettrait de rendre la méthode extensible même dans le cas où le couple initialisation-impédance d’interface serait mal choisi. En effet, ce problème grossier permet la propagation instantanée des bonnes informations grâce à une recherche de la solution dans un

espace approprié. Pour cela, le travail du déséquilibre des inter-efforts  $\mathbf{A}^\diamond \boldsymbol{\lambda}^\diamond$  sur une base macroscopique d'interface  $\mathbf{G}_m^\diamond = \mathbf{A}^\diamond \mathbf{R}_b^\diamond$  (potentiellement les modes rigides des sous-structures auxquels des modes de déformations peuvent être ajoutés) sera considéré. Ainsi, la fonctionnelle à stationnariser devient :

$$\mathcal{L}(\boldsymbol{\mu}_b^\diamond, \boldsymbol{\gamma}^\diamond) = \|\mathbf{B}^\diamond \mathbf{u}_b^\diamond\|_Z^2 + \|\mathbf{A}^\diamond \boldsymbol{\lambda}_b^\diamond\|_Y^2 + \boldsymbol{\gamma}^{\diamond T} \mathbf{G}_m^{\diamond T} \mathbf{A}^\diamond \boldsymbol{\lambda}^\diamond.$$

La stationnarité de  $\mathcal{L}(\boldsymbol{\mu}_b^\diamond, \boldsymbol{\gamma}^\diamond)$  donne les expressions :

$$\begin{cases} 0 = d\boldsymbol{\mu}_b^{\diamond T} \left[ \mathbf{S}_m^\diamond \left( \mathbf{B}^{\diamond T} \mathbf{Z} \mathbf{B}^\diamond + \mathbf{S}_p^\diamond \mathbf{A}^{\diamond T} \mathbf{Y} \mathbf{A}^\diamond \mathbf{S}_p^\diamond \right) \mathbf{S}_m^\diamond \boldsymbol{\mu}_b^\diamond + \mathbf{S}_m^\diamond \left( \mathbf{B}^{\diamond T} \mathbf{Z} \mathbf{B}^\diamond - \mathbf{S}_p^\diamond \mathbf{A}^{\diamond T} \mathbf{Y} \mathbf{A}^\diamond \mathbf{Q}_b^\diamond \right) \mathbf{b}_m^\diamond \right] \\ \quad + \mathbf{S}_m^\diamond \mathbf{S}_p^\diamond \mathbf{A}^{\diamond T} \mathbf{G}_m^\diamond \boldsymbol{\gamma}^\diamond \\ 0 = d\boldsymbol{\gamma}^T \mathbf{G}_m^{\diamond T} \mathbf{A}^\diamond \left( \mathbf{S}_p^\diamond \mathbf{S}_m^\diamond \boldsymbol{\mu}_b^\diamond - \mathbf{Q}_b^\diamond \mathbf{b}_m^\diamond \right) \end{cases}$$

où  $\boldsymbol{\gamma}^\diamond$  est le vecteur d'amplitudes des modes  $\mathbf{R}_b^\diamond$ . Le système à résoudre devient finalement :

$$\begin{aligned} \mathbf{S}_m^\diamond \left( \mathbf{S}_p^\diamond \mathbf{M}^{\diamond -1} \tilde{\mathbf{A}}^{\diamond T} \mathbf{A}^\diamond \mathbf{S}_p^\diamond + \mathbf{M}^\diamond \tilde{\mathbf{B}}^{\diamond T} \mathbf{B}^\diamond \right) \mathbf{S}_m^\diamond \boldsymbol{\mu}_b^\diamond + \mathbf{S}_p^\diamond \mathbf{A}^{\diamond T} \mathbf{G}_m^\diamond \boldsymbol{\gamma}^\diamond \\ = \mathbf{S}_m^\diamond \left( \mathbf{S}_p^\diamond \mathbf{A}^{\diamond T} \mathbf{Y} \mathbf{A}^\diamond \mathbf{Q}_b^\diamond - \mathbf{B}^{\diamond T} \mathbf{Z} \mathbf{B}^\diamond \right) \mathbf{b}_m^\diamond \\ \mathbf{G}_m^{\diamond T} \mathbf{A}^\diamond \mathbf{S}_p^\diamond \mathbf{S}_m^\diamond \boldsymbol{\mu}_b^\diamond = \mathbf{G}_m^{\diamond T} \mathbf{A}^\diamond \mathbf{Q}_b^\diamond \mathbf{b}_m^\diamond \end{aligned}$$

et peut être écrit sous forme matricielle :

$$\begin{aligned} \begin{bmatrix} \mathbf{S}_m^\diamond \left( \mathbf{S}_p^\diamond \mathbf{M}^{\diamond -1} \tilde{\mathbf{A}}^{\diamond T} \mathbf{A}^\diamond \mathbf{S}_p^\diamond + \mathbf{M}^\diamond \tilde{\mathbf{B}}^{\diamond T} \mathbf{B}^\diamond \right) \mathbf{S}_m^\diamond & \mathbf{S}_m^\diamond \mathbf{S}_p^\diamond \mathbf{A}^{\diamond T} \mathbf{G}_m^\diamond \\ \mathbf{G}_m^{\diamond T} \mathbf{A}^\diamond \mathbf{S}_p^\diamond \mathbf{S}_m^\diamond & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\mu}_b^\diamond \\ \boldsymbol{\gamma}^\diamond \end{bmatrix} \\ = \begin{bmatrix} \mathbf{S}_m^\diamond \left( \mathbf{S}_p^\diamond \mathbf{M}^{\diamond -1} \tilde{\mathbf{A}}^{\diamond T} \mathbf{A}^\diamond \mathbf{Q}_b^\diamond - \mathbf{M}^\diamond \tilde{\mathbf{B}}^{\diamond T} \mathbf{B}^\diamond \right) \mathbf{b}_m^\diamond \\ \mathbf{G}_m^{\diamond T} \mathbf{A}^\diamond \mathbf{Q}_b^\diamond \mathbf{b}_m^\diamond \end{bmatrix} \\ \begin{bmatrix} \mathbf{S}_m^\diamond \left( \mathbf{S}_p^\diamond \mathbf{M}^{\diamond -1} \tilde{\mathbf{A}}^{\diamond T} \mathbf{A}^\diamond \mathbf{S}_p^\diamond + \mathbf{M}^\diamond \tilde{\mathbf{B}}^{\diamond T} \mathbf{B}^\diamond \right) \mathbf{S}_m^\diamond & \tilde{\mathbf{G}}_m^\diamond \\ \tilde{\mathbf{G}}_m^{\diamond T} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\mu}_b^\diamond \\ \boldsymbol{\gamma}^\diamond \end{bmatrix} \\ = \begin{bmatrix} \mathbf{S}_m^\diamond \left( \mathbf{S}_p^\diamond \mathbf{M}^{\diamond -1} \tilde{\mathbf{A}}^{\diamond T} \mathbf{A}^\diamond \mathbf{Q}_b^\diamond - \mathbf{M}^\diamond \tilde{\mathbf{B}}^{\diamond T} \mathbf{B}^\diamond \right) \mathbf{b}_m^\diamond \\ \mathbf{G}_m^{\diamond T} \mathbf{A}^\diamond \mathbf{Q}_b^\diamond \mathbf{b}_m^\diamond \end{bmatrix} \end{aligned}$$

qui est un système de point selle symétrique. Dans ce système la notation  $\tilde{\mathbf{G}}_m^\diamond = \mathbf{S}_m^\diamond \mathbf{S}_p^\diamond \mathbf{A}^{\diamond T} \mathbf{G}_m^\diamond$  a aussi été introduite.

Ce système rappelle, par sa forme, le système de point selle (1.10) *FETI* et il peut être résolu de la même façon à l'aide d'une méthode d'initialisation et de projection en décomposant l'inconnue  $\boldsymbol{\mu}_b^\diamond$  en deux termes :

$$\boldsymbol{\mu}_b^\diamond = \hat{\boldsymbol{\mu}}_{b,0}^\diamond + \mathbf{P}_m^\diamond \hat{\boldsymbol{\mu}}_b^\diamond$$

où  $\mathbf{P}_m^\diamond$  est un projecteur orthogonal à  $\mathbf{Vect}(\tilde{\mathbf{G}}_m^\diamond)$  défini tel que  $\mathbf{P}_m^\diamond \tilde{\mathbf{G}}_m^\diamond = 0 = \tilde{\mathbf{G}}_m^{\diamond T} \mathbf{P}_m^\diamond$  et qui conduit à choisir un projecteur de la forme :

$$\mathbf{P}_m^\diamond = \mathbf{1}^\diamond - \tilde{\mathbf{G}}_m^\diamond \left( \tilde{\mathbf{G}}_m^{\diamond T} \tilde{\mathbf{G}}_m^\diamond \right)^{-1} \tilde{\mathbf{G}}_m^{\diamond T}$$

ainsi qu'une initialisation  $\hat{\boldsymbol{\mu}}_{b,0}^\diamond$  satisfaisant la seconde ligne du système  $\tilde{\mathbf{G}}_m^\diamond \hat{\boldsymbol{\mu}}_{b,0}^\diamond = \mathbf{G}_m^\diamond \mathbf{A}^\diamond \mathbf{Q}_b^\diamond \mathbf{b}_m^\diamond$ . Elle est définie telle que :

$$\hat{\boldsymbol{\mu}}_{b,0}^\diamond = \tilde{\mathbf{G}}_m^\diamond \left( \tilde{\mathbf{G}}_m^\diamond \tilde{\mathbf{G}}_m^\diamond \right)^{-1} \mathbf{G}_m^\diamond \mathbf{A}^\diamond \mathbf{Q}_b^\diamond \mathbf{b}_m^\diamond.$$

Ainsi, le problème à résoudre devient :

$$\begin{aligned} \mathbf{P}_m^\diamond \mathbf{S}_m^\diamond \left( \mathbf{S}_p^\diamond \mathbf{M}^\diamond \mathbf{A}^\diamond \tilde{\mathbf{A}}^\diamond \mathbf{A}^\diamond \mathbf{S}_p^\diamond + \mathbf{M}^\diamond \tilde{\mathbf{B}}^\diamond \mathbf{B}^\diamond \right) \mathbf{S}_m^\diamond \mathbf{P}_m^\diamond \hat{\boldsymbol{\mu}}_b^\diamond \\ = \mathbf{P}_m^\diamond \mathbf{S}_m^\diamond \left( \mathbf{S}_p^\diamond \mathbf{M}^\diamond \mathbf{A}^\diamond \tilde{\mathbf{A}}^\diamond \mathbf{A}^\diamond \mathbf{Q}_b^\diamond - \mathbf{M}^\diamond \tilde{\mathbf{B}}^\diamond \mathbf{B}^\diamond \right) \mathbf{b}_m^\diamond \\ - \mathbf{P}_m^\diamond \mathbf{S}_m^\diamond \left( \mathbf{S}_p^\diamond \mathbf{M}^\diamond \mathbf{A}^\diamond \tilde{\mathbf{A}}^\diamond \mathbf{A}^\diamond \mathbf{S}_p^\diamond + \mathbf{M}^\diamond \tilde{\mathbf{B}}^\diamond \mathbf{B}^\diamond \right) \mathbf{S}_m^\diamond \hat{\boldsymbol{\mu}}_{b,0}^\diamond \end{aligned}$$

Le taux de convergence de la résolution de ce système devrait être invariant lorsque le nombre de domaines augmente grâce à la projection sur l'espace orthogonal à l'espace grossier.

### 3.4 Conclusion partielle

Dans ce chapitre, en accord avec la littérature, une impédance d'interface a pu être définie comme étant le complément de Schur de la partie complémentaire au sous-domaine considéré pour le solveur *FETI-2LM* de ROUX et coll. [2003]. De plus, une initialisation optimale a aussi permis de montrer une convergence instantanée du solveur. Cependant cette impédance est très coûteuse à calculer et des approximations rarement satisfaisantes ont été développées dans la littérature. C'est pourquoi, nous avons proposé une impédance à deux niveaux, économique car exploitant le maillage grossier de la structure complète généré lors de la sous-structuration à deux niveaux du chapitre 2. Pour cette approximation à deux niveaux, un couplage de type mortar a été mis en place pour permettre de faire le lien entre les deux niveaux présents (fin et grossier). Deux propositions différentes, de cette approximation fondée sur un couplage mortar, ont aussi été mis en place. Un exemple de type poutre a permis de montrer que la seconde approximation proposée se comporte mieux que la première.

Une formulation de *FETI-2LM* a aussi été mise en place en faisant apparaître une seconde impédance d'interface jouant le rôle de préconditionneur. Un lien fort entre cette formulation et une minimisation du saut des quantités d'interface a pu être établi permettant l'ajout d'un problème grossier. Malheureusement, pour des questions de temps, ces idées n'ont pu être implémentées pour évaluer leur efficacité au regard de la convergence du solveur *FETI-2LM* équipé de différents types d'impédance de la littérature et de celles proposées ici.

Concernant la formulation avec problème grossier, la nature symétrique des opérateurs en jeu devrait permettre d'exploiter la théorie *GenEO* pour calculer des modes macroscopiques garantissant l'extensibilité.

### 3.5 Références

- AKULA, B. R. 2019, *Extended mortar method for contact and mesh-tying applications*, phdthesis, PSL Research University. URL <https://pastel.archives-ouvertes.fr/tel-02200516>. 122
- BEN BELGACEM, F. B. 1999, «The Mortar finite element method with Lagrange multipliers», *Numerische Mathematik*, vol. 84, n° 2, doi : 10.1007/s002110050468, p. 173–197, ISSN 0029-599X, 0945-3245. URL <http://link.springer.com/10.1007/s002110050468>. 121
- BENZI, M., G. H. GOLUB et J. LIESEN. 2005, «Numerical solution of saddle point problems», *Acta Numerica*, vol. 14, doi : 10.1017/S0962492904000212, p. 1–137, ISSN 1474-0508, 0962-4929. URL <https://www.cambridge.org/core/journals/>

- [acta-numerica/article/numerical-solution-of-saddle-point-problems/2596C5D03B23AF89FE5A756891029B12](#), publisher: Cambridge University Press. 129
- BERNARDI, C., Y. MADAY et A. T. PATERA. 1993, «Domain Decomposition by the Mortar Element Method», dans *Asymptotic and Numerical Methods for Partial Differential Equations with Critical Parameters*, édité par H. G. Kaper, M. Garbey et G. W. Pieper, NATO ASI Series, Springer Netherlands, Dordrecht, ISBN 978-94-011-1810-1, p. 269–286, doi : [10.1007/978-94-011-1810-1\\_17](#). URL [https://doi.org/10.1007/978-94-011-1810-1\\_17](https://doi.org/10.1007/978-94-011-1810-1_17). 121
- CRESTA, P., O. ALLIX, C. REY et S. GUINARD. 2007, «Nonlinear localization strategies for domain decomposition methods: Application to post-buckling analyses», *Computer Methods in Applied Mechanics and Engineering*, vol. 196, n° 8, doi : [10.1016/j.cma.2006.03.013](#), p. 1436–1446, ISSN 0045-7825. URL <http://www.sciencedirect.com/science/article/pii/S0045782506002635>. 117
- DUREISSEIX, D. et H. BAVESTRELLO. 2006, «Information transfer between incompatible finite element meshes: Application to coupled thermo-viscoelasticity», *Computer Methods in Applied Mechanics and Engineering*, vol. 195, n° 44, doi : [10.1016/j.cma.2006.02.003](#), p. 6523–6541, ISSN 0045-7825. URL <http://www.sciencedirect.com/science/article/pii/S0045782506000673>. 120
- FARHAT, C., J. MANDEL et F. X. ROUX. 1994, «Optimal convergence properties of the FETI domain decomposition method», *Computer Methods in Applied Mechanics and Engineering*, vol. 115, n° 3-4, doi : [10.1016/0045-7825\(94\)90068-X](#), p. 365–385, ISSN 00457825. URL <http://linkinghub.elsevier.com/retrieve/pii/004578259490068X>. 117
- GANDER, M. J. et F. KWOK. 2011, «Optimal Interface Conditions for an Arbitrary Decomposition into Subdomains», dans *Domain Decomposition Methods in Science and Engineering XIX*, édité par Y. Huang, R. Kornhuber, O. Widlund et J. Xu, Lecture Notes in Computational Science and Engineering, Springer, Berlin, Heidelberg, ISBN 978-3-642-11304-8, p. 101–108, doi : [10.1007/978-3-642-11304-8\\_9](#). 115
- GENDRE, L., O. ALLIX et P. GOSSELET. 2011, «A two-scale approximation of the Schur complement and its use for non-intrusive coupling», *International Journal for Numerical Methods in Engineering*, vol. 87, n° 9, doi : [10.1002/nme.3142](#), p. 889–905, ISSN 1097-0207. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.3142>, \_eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.3142>. 118
- GUIDAULT, P. A., O. ALLIX, L. CHAMPANEY et C. CORNUAULT. 2008, «A multiscale extended finite element method for crack propagation», *Computer Methods in Applied Mechanics and Engineering*, vol. 197, n° 5, doi : [10.1016/j.cma.2007.07.023](#), p. 381–399, ISSN 0045-7825. URL <http://www.sciencedirect.com/science/article/pii/S0045782507003283>. 117
- HAGER, W. W. 1989, «Updating the Inverse of a Matrix», *SIAM Review*, vol. 31, n° 2, doi : [10.1137/1031049](#), p. 221–239, ISSN 0036-1445, 1095-7200. URL <http://epubs.siam.org/doi/10.1137/1031049>. 118
- HINOJOSA, J., O. ALLIX, P. A. GUIDAULT et P. CRESTA. 2014, «Domain decomposition methods with nonlinear localization for the buckling and post-buckling analyses of large structures», *Advances in Engineering Software*, vol. 70, doi : [10.1016/j.advengsoft.2013.12.010](#), p. 13–24, ISSN 0965-9978. URL <http://www.sciencedirect.com/science/article/pii/S096599781400012X>. 117
- LADEVÈZE, P. 1985, «Sur une famille d'algorithmes en mécanique des structures», *Sur une famille d'algorithmes en mécanique des structures*, vol. 300, n° 2, p. 41–44, ISSN 0750-7623. Place: Paris Publisher: Gauthier-Villars. 128

- LADEVÈZE, P. et D. DUREISSEIX. 1999, «Une nouvelle stratégie de calcul micro/macro en mécanique des structures», *Comptes Rendus de l'Académie des Sciences - Series IIB - Mechanics-Physics-Astronomy*, vol. 327, n° 12, doi : [10.1016/S1287-4620\(00\)88647-0](https://doi.org/10.1016/S1287-4620(00)88647-0), p. 1237–1244, ISSN 1287-4620. URL <http://www.sciencedirect.com/science/article/pii/S1287462000886470>. 117, 128
- LADEVÈZE, P. et A. NOUY. 2003, «On a Multiscale Computational Strategy with Time and Space Homogenization for Structural Mechanics», *Computer Methods in Applied Mechanics and Engineering*, vol. 192, n° 28-30, doi : [10.1016/S0045-7825\(03\)00341-4](https://doi.org/10.1016/S0045-7825(03)00341-4), p. 3061–3087. URL <https://hal.archives-ouvertes.fr/hal-00366648>, publisher: Elsevier. 117
- MAGOULÈS, F., F.-X. ROUX et S. SALMON. 2004, «Optimal Discrete Transmission Conditions for a Nonoverlapping Domain Decomposition Method for the Helmholtz Equation», *SIAM Journal on Scientific Computing*, vol. 25, n° 5, doi : [10.1137/S1064827502415351](https://doi.org/10.1137/S1064827502415351), p. 1497–1515, ISSN 1064-8275. URL <https://epubs.siam.org/doi/10.1137/S1064827502415351>. 115
- MAGOULÈS, F., F.-X. ROUX et L. SERIES. 2006, «Algebraic approximation of Dirichlet-to-Neumann maps for the equations of linear elasticity», *Computer Methods in Applied Mechanics and Engineering*, vol. 195, n° 29, doi : [10.1016/j.cma.2005.01.022](https://doi.org/10.1016/j.cma.2005.01.022), p. 3742–3759, ISSN 0045-7825. URL <http://www.sciencedirect.com/science/article/pii/S0045782505002562>. 117
- MANDEL, J. 1993, «Balancing domain decomposition», *Communications in Numerical Methods in Engineering*, vol. 9, n° 3, doi : [10.1002/cnm.1640090307](https://doi.org/10.1002/cnm.1640090307), p. 233–241, ISSN 10698299. URL <http://doi.wiley.com/10.1002/cnm.1640090307>. 117
- NEGRELLO, C., P. GOSSELET et C. REY. 2018, «A new impedance accounting for short and long range effects in mixed substructured formulations of nonlinear problems», *International Journal for Numerical Methods in Engineering*, vol. 114, n° 7, doi : [10.1002/nme.5758](https://doi.org/10.1002/nme.5758), p. 675–693. URL <https://hal.archives-ouvertes.fr/hal-01488615>. xi, 118, 119
- OUMAZIZ, P., P. GOSSELET, P.-A. BOUCARD et S. GUINARD. 2017, «A non-invasive implementation of a mixed domain decomposition method for frictional contact problems», *Computational Mechanics*, vol. 60, n° 5, doi : [10.1007/s00466-017-1444-x](https://doi.org/10.1007/s00466-017-1444-x), p. 797–812, ISSN 0178-7675, 1432-0924. URL <https://link.springer.com/article/10.1007/s00466-017-1444-x>. 117
- POPP, A. 2012, *Mortar Methods for Computational Contact Mechanics and General Interface Problems*, PhD Thesis, Universitätsbibliothek der TU München. 122
- ROUX, F.-X. 2009, «A FETI-2LM Method for Non-Matching Grids», dans *Domain Decomposition Methods in Science and Engineering XVIII*, vol. 70, édité par M. J. Gander, O. Widlund et M. Bercovier, Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN 978-3-642-02676-8 978-3-642-02677-5, p. 121–128, doi : [10.1007/978-3-642-02677-5\\_11](https://doi.org/10.1007/978-3-642-02677-5_11). URL [http://link.springer.com/10.1007/978-3-642-02677-5\\_11](http://link.springer.com/10.1007/978-3-642-02677-5_11), series Title: Lecture Notes in Computational Science and Engineering. 122
- ROUX, F.-X., F. MAGOULÈS, S. SALMON et L. SERIES. 2003, «Optimization of Interface Operator Based on Algebraic Approach», dans *Fourteenth International Conference on Domain Decomposition Methods*, Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates, p. 297–304. 134
- STEFANICA, D. 2001, «A Numerical Study of FETI Algorithms for Mortar Finite Element Methods», *SIAM Journal on Scientific Computing*, vol. 23, n° 4, doi : [10.1137/S1064827500378829](https://doi.org/10.1137/S1064827500378829), p. 1135–1160, ISSN 1064-8275, 1095-7197. URL <http://epubs.siam.org/doi/10.1137/S1064827500378829>. 122

- STEFANICA, D. 2005, «Parallel FETI algorithms for mortars», *Applied Numerical Mathematics*, vol. 54, n° 2, doi : [10.1016/j.apnum.2004.09.030](https://doi.org/10.1016/j.apnum.2004.09.030), p. 266–279, ISSN 01689274. URL <http://linkinghub.elsevier.com/retrieve/pii/S016892740400203X>. 122
- WANGERMEZ, M., O. ALLIX, P.-A. GUIDAULT, O. CIOBANU et C. REY. 2020, «Interface coupling method for the global-local analysis of heterogeneous models: A second-order homogenization-based strategy», *Computer Methods in Applied Mechanics and Engineering*, vol. 365, doi : [10.1016/j.cma.2020.113032](https://doi.org/10.1016/j.cma.2020.113032), p. 113 032, ISSN 0045-7825. URL <http://www.sciencedirect.com/science/article/pii/S0045782520302164>. 117



Ce manuscrit a cherché à répondre aux objectifs suivants :

- permettre une parallélisation de la génération de maillage;
- générer des sous-domaines dont la forme améliore la rapidité du solveur;
- exploiter cette sous-structuration pour améliorer la convergence d'un solveur par décomposition de domaine.

La démarche du schéma de la figure 5 permet de remplir les deux premiers objectifs alors que le dernier objectif n'est que partiellement rempli car seul un cadre théorique a pu être mis en place pour l'approximation à deux niveaux du paramètre d'impédance d'interface critique de la méthode *FETI-2LM*.

Dans un premier temps, les principales méthodes de décomposition de domaine sans recouvrement ont été explicitées, en insistant sur les approches primale, duale et mixte qui, pour les deux dernières, ont été utilisées dans ce travail. Dans ces trois cas, la résolution consiste à décomposer le problème global en plusieurs sous-problèmes définis à partir de la décomposition du domaine physique. Les différentes approches de décomposition de domaine se distinguent par les différents types de conditions entre les *DDLs* aux interfaces. La continuité des déplacements est imposée dans la méthode primale alors que l'équilibre des efforts est imposé dans la méthode duale. La méthode mixte quant à elle impose une condition mixte à l'interface appelée condition de Robin en faisant apparaître un opérateur de combinaison linéaire appelé impédance (ou rigidité) d'interface. Cet opérateur est un paramètre des approches mixtes dont un choix judicieux permet une meilleure convergence du solveur. Cette section a aussi permis de montrer que la vitesse de convergence de ces méthodes dépendait fortement de la forme des sous-domaines ainsi que de leur homogénéité. Les méthodes de *pre-processing* d'une simulation parallèle ont ensuite été évoquées pour montrer que les approches multi-niveaux de partitionnement font consensus et sont massivement utilisées dans des logiciels libres tels que Metis, Scotch ou encore Chaco. Pour la génération de maillage en parallèle, par contre, les méthodes les plus performantes exploitent toujours un maillage initial grossier permettant de distribuer les données dès le début du processus de maillage (qui s'apparente alors à du remaillage parallèle d'un maillage grossier initial). Pour réaliser l'étape de déformation de maillage, notre revue de la littérature ainsi que les tests numériques concluent que la projection de surface la plus efficace est un algorithme d'intersection rayon-maillage triangulaire optimisé. Et la déformation du maillage volumique réalisée à l'aide d'une méthode de déformation de maillage par *RBF* semble être la plus adaptée mais cette méthode fait intervenir des opérateurs denses dont la factorisation peut devenir coûteuse.

Dans un deuxième temps, la méthode de sous-structuration à deux niveaux a été explicitée. Son principe est, schématiquement, d'échanger les étapes de partitionnement et de génération du maillage qui sont réalisées dans les approches classiques de préparation des données. Pour cela, la décomposition en sous-domaines est réalisée dès le début du processus complet sur un maillage

grossier permettant la prise en compte de la distribution des différents matériaux. Ensuite seulement, un maillage conforme plus fin est généré en parallèle sur chacun des sous-domaines, puis une déformation du maillage final est réalisée pour assurer une représentation fidèle et précise de la géométrie à discrétiser. L'avantage ici est que les sous-domaines générés ont des interfaces régulières et sont composés d'un unique matériau. Ainsi, la convergence d'un solveur *FETI* sur cette décomposition est meilleure en présence d'hétérogénéités fortes sur les cas qui ont été testés en comparaison d'une approche plus systématique utilisant uniquement un logiciel de partitionnement.

Dans un troisième temps, une nouvelle approximation de l'impédance d'interface optimale du solveur mixte a pu être développée en tenant compte à la fois les interactions proches et les interactions lointaines. Le caractère à deux niveaux de la sous-structuration (existence d'un maillage grossier de la structure globale partagé par tous les sous-domaines) permet d'obtenir cette approximation pour un faible coût en conservant toutes les informations courtes et longues distances. Des formulations mixtes alternatives ont été proposées pour permettre l'adjonction d'un problème grossier permettant de rendre la méthode extensible. En effet, il n'existe pas de problème grossier naturel ici car l'opérateur local est inversible et les modes rigides n'apparaissent pas aussi naturellement que dans l'approche duale par exemple. Ces formulations font apparaître une seconde impédance d'interface jouant le rôle de préconditionneur dont l'interprétation physique n'est pas évidente.

Finalement, l'ensemble des développements présentés ont été implémentés en langage Python grâce à l'interface qu'offre la suite logicielle *EF Z-set/ZéBuLon* permettant d'être suffisamment intrusif pour utiliser *Z-set* en tant que solveur local dont les opérateurs peuvent être modifiés. Les outils de manipulation de maillage de *Z-set* ont aussi été très utiles pour le développement de la méthode de sous-structuration à deux niveaux. Le parallélisme est quant à lui géré par la bibliothèque *MPI* interfacé avec Python proposant le module *mpi4py*.

Ces travaux suscitent plusieurs perspectives dont trois axes peuvent être dégagés. La première piste de recherche peut être l'application de ces méthodes à d'autres cas; ce qui peut être vu comme des perspectives à court terme. La deuxième piste concerne l'adaptation de maillage et peut être vue comme une perspective à moyen terme. La troisième piste peut être liée à l'extension de la méthode mixte à des cadres plus large que les solveurs de Krylov.

Par rapport à la première piste concernant les cas d'applications, plusieurs cas peuvent être envisagés. Une application directe serait d'utiliser des géométries issues de composites tissés tridimensionnels en générant par exemple une structure de type *VER* à l'aide d'un logiciel tel que *TexGen*<sup>3</sup> pour avoir accès à la géométrie et au maillage de ce *VER*. Cela n'est pas évident en entreprise car, bien souvent, seuls les maillages des structures sont disponibles. Les maillages générés ne pouvant être très grossiers en raison des faibles épaisseurs de matrice, il faudrait pouvoir utiliser un grand nombre de sous-domaines pour leur garantir une forme régulière en ayant autant de sous-domaines que d'éléments grossiers. Si, un faible nombre de machines est disponible pour obtenir un grand nombre de sous-domaines, il peut être préférable de relâcher la contrainte d'homogénéité des sous-domaines en conservant seulement leur aspect régulier. Cette méthodologie peut aussi être appliquée à d'autres composants mécaniques de Safran tels que des aubes monocristallines ou polycristallines si le maillage et surtout la géométrie sont rendus disponibles. Une seconde application concerne le solveur mixte, où il faudrait vérifier que la présence d'hétérogénéités est bien prise en compte par l'impédance d'interface à deux échelles. En effet, les hétérogénéités éloignées d'un sous-domaines doivent être détectées et leur influence doit être intégrée instantanément à la résolution. Pour rendre compte de cela, un grand nombre de sous-domaines est nécessaire et le passage à l'échelle nécessite des optimisations de l'implémentation permettant de rendre les opérations locales plus efficaces.

Les perspectives à moyens termes concernant l'adaptation de maillage se présentent assez naturellement. Un indicateur d'erreur de discrétisation parallèle aurait un grand intérêt dans ce type de décomposition homogène. En effet, les travaux de *PARRET-FRÉAUD [2011]* ont montré

3. <http://texgen.sourceforge.net>

que son estimateur d'erreur parallèle était meilleur que son homologue séquentiel lorsque les sous-domaines sont homogènes. La sous-structuration développée dans ce manuscrit permettrait l'exploitation de cet estimateur d'erreur parallèle couplé à une méthode de remaillage parallèle conservant la forme des sous-domaines comme la méthode de BOVET et coll. [2019]. Par contre la propriété de maillage localement structuré serait perdue et les accélérations liées à l'utilisation d'accélérateur graphique (GPU) évoquées en section 2.1.3 ne pourraient pas être exploitées. On peut aussi envisager dans la phase de raffinement local de la stratégie de sous-structuration à deux niveaux d'utiliser cette méthode de remaillage parallèle ou encore la méthode appelée « maillage topologique » décrite brièvement dans la section 2.1.3.

Pour la troisième et dernière piste concernant l'extension de la méthode mixte à d'autres types de résolution. Certains choix pour les opérateurs d'impédance d'interface ( $\mathbf{Q}_b^\diamond$  et  $\mathbf{X}_b^\diamond$ ) conduisent à des systèmes linéaires de rayon spectral plus petit que 1. L'utilisation de solveurs de Krylov n'est alors plus indispensable et permettrait une résolution possiblement asynchrone du problème d'interface mixte à l'aide de solveurs itératifs de type point fixe nécessitant des opérateurs d'itérations contractants (dont le rayon spectral est inférieur à 1). Une extension au non-linéaire est aussi envisageable. En effet, les développements de solveurs non linéaires avec recondensation non-linéaire tels que ceux développés par NEGRELLO et coll. [2016] permettent de laisser s'exprimer la non-linéarité indépendamment dans chaque sous-domaine. L'introduction d'une condition d'interface proche de la physique permet de conserver une certaine cohérence entre les non-linéarités locales. L'introduction d'une impédance prenant en compte la rigidité globale à l'aide du maillage grossier de la structure permettrait aussi de considérer des non-linéarités plus globales telles que le flambement.

## Références

- BOVET, C., O. CIOBANU, A. PARRET-FREAUD et V. CHIARUTTINI. 2019, «A parallel mesh refinement process tailored to domain decomposition methods», dans *Proceedings of the Sixth International Conference on Parallel, Distributed, GPU and Cloud Computing for Engineering*, Pollock Mihály Faculty of Engineering and Information Technology, University of Pécs, Hungary, p. 20, doi : 10.4203/ccp.112.20. URL <http://www.ctresources.info/ccp/paper.html?id=9287>. 141
- NEGRELLO, C., P. GOSSELET, C. REY et J. PEBREL. 2016, «Substructured formulations of nonlinear structure problems - influence of the interface condition», *International Journal for Numerical Methods in Engineering*, vol. 107, n° 13, doi : 10.1002/nme.5195, p. 1083–1105. 141
- PARRET-FRÉAUD, A. 2011, *Estimation d'erreur de discrétisation dans les calculs par décomposition de domaine*, thesis, Cachan, Ecole normale supérieure. URL <http://www.theses.fr/2011DENS0022>, publication Title: <http://www.theses.fr>. 140



## A.1 Solveurs itératifs de Krylov

Le but de cette annexe est de présenter rapidement les principes et algorithmes relatifs aux solveurs de Krylov. Le lecteur pourra consulter [BARRETT et coll. \[1994\]](#), [SAAD \[1997\]](#) ou encore [SAAD \[2003\]](#) pour des explications plus détaillées.

Les méthodes de Krylov consistent à approximer la solution  $\mathbf{A}^{-1}\mathbf{b}$  du système  $\mathbf{Ax} = \mathbf{b}$  par un vecteur  $p(\mathbf{A})\mathbf{b}$  où  $p$  est un polynôme construit intelligemment. On considère par la suite que  $\mathbf{A}$  est une matrice carrée de taille  $n \times n$  et  $\mathbf{b}$  un vecteur de  $\text{Im}(\mathbf{A})$ . La solution à l'itération  $i$  est notée  $\mathbf{x}_i$  et est associée à un résidu  $\mathbf{r}_i = \mathbf{b} - \mathbf{Ax}_i = \mathbf{A}(\mathbf{x} - \mathbf{x}_i)$ . Une solution initiale  $\mathbf{x}_0$ , souvent prise égale à 0, est choisie. Les quantités à l'itération  $i$  seront notées avec un indice  $i$ .

### A.1.1 Cas général

Les solveurs de Krylov consistent à construire itérativement un « sous-espace de Krylov »  $\mathcal{K}_i(\mathbf{A}, \mathbf{r}_0)$  défini par :

$$\mathcal{K}_i(\mathbf{A}, \mathbf{r}_0) = \text{Vect}(\mathbf{r}_0, \dots, \mathbf{A}^{i-1}\mathbf{r}_0)$$

La recherche de la solution du système linéaire consiste à chercher  $\mathbf{x}_i$  sous les contraintes :

$$\begin{cases} \mathbf{x}_i \in \mathbf{x}_0 + \mathcal{K}_i(\mathbf{A}, \mathbf{r}_0) \\ \mathbf{r}_i \perp \mathcal{K}_i(\mathbf{A}, \mathbf{r}_0) \end{cases}$$

où le choix de la relation d'orthogonalité permet de définir les différentes approches.

### A.1.2 Gradient Conjugué

#### A.1.2.1 Algorithmes de base

Le *CG* est un cas particulier de solveur de Krylov où la relation d'orthogonalité choisie est l'orthogonalité au sens de la norme  $\|\cdot\|_{\mathbf{A}}$  qui présuppose que l'opérateur  $\mathbf{A}$  est symétrique défini positif. Il consiste en une projection orthogonale basée sur l'espace de Krylov  $\mathcal{K}_i(\mathbf{A}, \mathbf{r}_0)$ . Le principe de recherche est le suivant :

$$\begin{cases} \mathbf{x}_i \in \mathbf{x}_0 + \mathcal{K}_i(\mathbf{A}, \mathbf{r}_0) \\ \mathbf{r}_i \perp \mathcal{K}_i(\mathbf{A}, \mathbf{r}_0) \end{cases},$$

ce qui équivaut à trouver  $\mathbf{x}_i \in \mathbf{x}_0 + \mathcal{K}_i(\mathbf{A}, \mathbf{r}_0)$  minimisant  $\|\mathbf{x}_i - \mathbf{x}\|_{\mathbf{A}}$ .

Du fait, des propriétés de  $\mathbf{A}$ , et notamment de sa symétrie, des relations de conjugaison apparaissent, conduisant à l'algorithme [A.1](#). Dans cet algorithme, deux bases de  $\mathcal{K}_i(\mathbf{A}, \mathbf{r}_0)$  sont construites : la base des résidus qui est orthogonale et la base des directions de recherche qui,

**Algorithme A.1 : Gradient Conjugué****Initialisation :**Initialisation du résidu  $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$  ; $\mathbf{d}_0 = \mathbf{r}_0$  ;**tant que**  $\frac{\mathbf{r}_i^T \mathbf{r}_i}{\mathbf{r}_0^T \mathbf{r}_0} > \varepsilon$  **faire**    Calcul du pas optimal  $\alpha_i = \frac{\mathbf{r}_i^T \mathbf{r}_i}{\mathbf{d}_i^T \mathbf{A} \mathbf{d}_i}$  ;    Calcul de la nouvelle solution  $\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{d}_i$  ;    Calcul du nouveau résidu  $\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i \mathbf{A} \mathbf{d}_i$  ;

A-orthonormalisation de la direction de recherche par rapport à la précédente :

 $\beta_i = \frac{\mathbf{r}_{i+1}^T \mathbf{r}_{i+1}}{\mathbf{r}_i^T \mathbf{r}_i}$  ;    Calcul de la nouvelle direction de recherche :  $\mathbf{d}_{i+1} = \mathbf{r}_{i+1} + \beta_i \mathbf{d}_i$  ;**fin**

elle, est A-orthogonale. L'orthogonalisation effectuée dans l'algorithme A.1 implique une orthogonalité de  $\mathbf{d}_{i+1}$ , vis-à-vis de toutes les directions précédentes. Cependant, numériquement, cette orthogonalité est perdue au cours des itérations. Il est donc souvent nécessaire d'utiliser un orthogonalisation complète de la base des directions de descente, ce qui conduit à l'algorithme A.2

**Algorithme A.2 : Gradient Conjugué réorthogonalisé****Initialisation :**Initialisation du résidu  $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$  ; $\mathbf{d}_0 = \mathbf{r}_0$  ;**tant que**  $\frac{\mathbf{r}_i^T \mathbf{r}_i}{\mathbf{r}_0^T \mathbf{r}_0} > \varepsilon$  **faire**    Calcul du pas optimal  $\alpha_i = \frac{\mathbf{r}_i^T \mathbf{r}_i}{\mathbf{d}_i^T \mathbf{A} \mathbf{d}_i}$  ;    Calcul de la nouvelle solution  $\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{d}_i$  ;    Calcul du nouveau résidu  $\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i \mathbf{A} \mathbf{d}_i$  ;    **pour**  $0 \leq j \leq i$  **faire** Réorthogonalisation de la base de recherche         $\beta_i^j = -\frac{\mathbf{r}_{i+1}^T \mathbf{A} \mathbf{d}_j}{\mathbf{d}_j^T \mathbf{A} \mathbf{d}_j}$  ;    **fin**    Calcul de la nouvelle direction de recherche :  $\mathbf{d}_{i+1} = \mathbf{r}_{i+1} + \sum_{j \leq i} \beta_i^j \mathbf{d}_j$  ;**fin****A.1.2.2 CG préconditionné**

Comme c'est le cas dans les approches primale et duale de décomposition de domaine, il est parfois nécessaire de résoudre un problème comportant un préconditionneur  $\tilde{\mathbf{M}}^{-1}$  symétrique défini positif. Ainsi, le système à résoudre devient :

$$\tilde{\mathbf{M}}^{-1} \mathbf{A} \mathbf{x} = \tilde{\mathbf{M}}^{-1} \mathbf{b}.$$

Ce système n'étant pas symétrique, le recours à un produit scalaire adapté  $(\mathbf{x}, \mathbf{y})_{\tilde{\mathbf{M}}^{-1}} = \mathbf{x}^T \tilde{\mathbf{M}}^{-1} \mathbf{y}$  est nécessaire pour conserver le CG en tant que solveur. Sa mise en œuvre se fait directement dans l'algorithme du gradient conjugué (algorithme A.1 ou A.2) en multipliant le résidu par le préconditionneur  $\tilde{\mathbf{M}}^{-1}$  et cela revient à l'algorithme A.3.

**Algorithme A.3 : Gradient Conjugué préconditionné avec réorthogonalisation****Initialisation :**Initialisation du résidu  $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$  ;Préconditionnement du résidu  $\mathbf{z}_0 = \tilde{\mathbf{M}}^{-1}\mathbf{r}_0$  ; $\mathbf{d}_0 = \mathbf{z}_0$  ;**tant que**  $\frac{\mathbf{r}_i^T \mathbf{z}_i}{\mathbf{r}_0^T \mathbf{z}_0} > \varepsilon$  **faire**    Calcul du pas optimal  $\alpha_i = \frac{\mathbf{r}_i^T \mathbf{z}_i}{\mathbf{d}_i^T \mathbf{A} \mathbf{d}_i}$  ;    Calcul de la nouvelle solution  $\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{d}_i$  ;    Calcul du nouveau résidu  $\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i \mathbf{A} \mathbf{d}_i$  ;    Préconditionnement du résidu  $\mathbf{z}_{i+1} = \tilde{\mathbf{M}}^{-1} \mathbf{r}_{i+1}$  ;    **pour**  $0 \leq j \leq i$  **faire** Réorthogonalisation de la base de recherche         $\beta_i^j = -\frac{\mathbf{z}_{i+1}^T \mathbf{A} \mathbf{d}_j}{\mathbf{d}_j^T \mathbf{A} \mathbf{d}_j}$  ;    **fin**    Calcul de la nouvelle direction de recherche :  $\mathbf{d}_{i+1} = \mathbf{z}_{i+1} + \sum_{j \leq i} \beta_i^j \mathbf{d}_j$  ;**fin****A.1.2.3 Algorithmes de type Krylov contraint**

Parfois, la résolution d'un système peut s'interpréter comme un sous-problème auquel un condition sur la solution aurait été adjointe (comme c'est le cas dans l'approche duale). Ainsi, le problème qui est sous la forme d'un problème de point selle :

$$\begin{bmatrix} \mathbf{A} & \mathbf{G} \\ \mathbf{G}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{e} \end{bmatrix}$$

peut se résoudre classiquement en prenant une solution  $\mathbf{x}$  comme somme de deux quantité appartenent à des espaces supplémentaires. Une initialisation  $\mathbf{x}_0$  est choisie de telle sorte qu'elle satisfasse la contrainte et qu'elle assure que le reste de la solution est bien dans l'espace supplémentaire, de manière à ce que la solution  $\mathbf{x}$  vérifie :

$$\mathbf{G}^T (\mathbf{x} - \mathbf{x}_0) = 0. \quad (\text{A.1})$$

Cela se traduit par l'introduction d'un projecteur  $\mathbf{P}$  sur l'espace orthogonal à  $\mathbf{Vect}(\mathbf{G})$  pour pouvoir mettre la solution sous la forme :

$$\mathbf{x} = \mathbf{x}_0 + \mathbf{P}\tilde{\mathbf{x}}.$$

Les conditions qui découlent de l'équation (A.1), sont :

$$\mathbf{G}^T \mathbf{x}_0 = \mathbf{e},$$

$$\mathbf{G}^T \mathbf{P} = 0.$$

Cela conduit à trouver  $\mathbf{x}_0$  et  $\mathbf{P}$  de la forme :

$$\mathbf{x}_0 = \mathbf{Q} \mathbf{G} (\mathbf{G}^T \mathbf{Q} \mathbf{G})^{-1} \mathbf{e},$$

$$\mathbf{P} = \mathbf{1} - \mathbf{Q} \mathbf{G} (\mathbf{G}^T \mathbf{Q} \mathbf{G})^{-1} \mathbf{G}^T,$$

où  $\mathbf{Q}$  est une matrice symétrique définie positive qui permet à  $\mathbf{G}^T \mathbf{Q} \mathbf{G}$  d'être inversible. Ainsi, le système à résoudre itérativement peut s'écrire :

$$\mathbf{P}^T \mathbf{A} \mathbf{P} \tilde{\mathbf{x}} = \mathbf{P}^T (\mathbf{b} - \mathbf{A} \mathbf{x}_0).$$

Il est possible de calculer enfin  $\boldsymbol{\alpha}$  par la relation suivante :

$$\boldsymbol{\alpha} = (\mathbf{G}^T \mathbf{Q} \mathbf{G})^{-1} \mathbf{G}^T \mathbf{Q} (\mathbf{b} - \mathbf{A} \mathbf{x}).$$

Pour calculer  $\mathbf{x}$ , le système est résolu grâce à une méthode de Krylov contrainte, comme par exemple un gradient conjugué comme dans l'algorithme A.4.

**Algorithme A.4 :** Gradient Conjugué contraint avec réorthogonalisation**Initialisation :**Calcul du projecteur  $\mathbf{P} = \mathbf{1} - \mathbf{Q}\mathbf{G}(\mathbf{G}^T\mathbf{Q}\mathbf{G})^{-1}\mathbf{G}^T$  ;Initialisation de la solution :  $\mathbf{x}_0 = \mathbf{Q}\mathbf{G}(\mathbf{G}^T\mathbf{Q}\mathbf{G})^{-1}\mathbf{e}$  ;Initialisation du résidu  $\mathbf{r}_0 = -\mathbf{P}^T(\mathbf{b} - \mathbf{A}\mathbf{x}_0)$  ; $\mathbf{d}_0 = \mathbf{r}_0$  ;**tant que**  $\frac{\mathbf{r}_i^T\mathbf{r}_i}{\mathbf{r}_0^T\mathbf{r}_0} > \varepsilon$  **faire**     $\mathbf{p}_i = \mathbf{P}^T\mathbf{A}\mathbf{d}_i$  ;    Calcul du pas optimal  $\alpha_i = \frac{\mathbf{r}_i^T\mathbf{r}_i}{\mathbf{d}_i^T\mathbf{p}_i}$  ;    Calcul de la nouvelle solution  $\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i\mathbf{d}_i$  ;    Calcul du nouveau résidu  $\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i\mathbf{p}_i$  ;    **pour**  $0 \leq j \leq i$  **faire** Réorthogonalisation de la base de recherche         $\beta_i^j = -\frac{\mathbf{r}_{i+1}^T\mathbf{p}_j}{\mathbf{d}_j^T\mathbf{p}_j}$  ;    **fin**    Calcul de la nouvelle direction de recherche :  $\mathbf{d}_{i+1} = \mathbf{r}_{i+1} + \sum_{j \leq i} \beta_i^j \mathbf{d}_j$  ;**fin****Post-traitement :** $\boldsymbol{\alpha} = (\mathbf{G}^T\mathbf{Q}\mathbf{G})^{-1}\mathbf{G}^T\mathbf{Q}\mathbf{r}_i$ **A.1.2.4 Algorithmes de type Krylov augmenté**

Les méthodes de Krylov augmenté décrites par SAAD [1997] ou CHAPMAN et SAAD [1997] sont utilisées pour ajouter des contraintes optionnelles à la résolution d'un système linéaire. Le principe est de définir un sous espace  $\mathcal{C}$  de  $\mathbb{R}^n$  de dimension  $n_c$ , représenté par une matrice rectangulaire  $\mathbf{C}$  (tel que  $\text{Im}(\mathbf{C}) = \mathcal{C}$ ) où  $\mathbf{C}$  est supposée être une base de  $\mathcal{C}$ . On appelle espace de Krylov augmenté, l'espace défini tel que  $\tilde{\mathcal{K}}_i(\mathbf{A}, \mathbf{r}_0, \mathbf{C}) = \mathcal{K}_i(\mathbf{A}, \mathbf{r}_0) + \text{Im}(\mathbf{C})$ . Le principe de recherche reste le suivant :

$$\begin{cases} \mathbf{x}_i \in \mathbf{x}_0 + \tilde{\mathcal{K}}_i(\mathbf{A}, \mathbf{r}_0, \mathbf{C}) \\ \mathbf{r}_i \perp \tilde{\mathcal{K}}_i(\mathbf{A}, \mathbf{r}_0, \mathbf{C}) \end{cases}$$

Généralement les méthode de type Krylov-augmenté sont mises en œuvre par des algorithmes projetés. Le principe est de chercher la solution séparément dans deux sous-espaces supplémentaires  $\text{Im}(\mathbf{C})$  et  $\ker(\mathbf{C}^T\mathbf{A})$ . La partie de la solution dans  $\ker(\mathbf{C}^T\mathbf{A})$  est détectée à l'initialisation alors que l'autre partie qui est dans un sous-espace supplémentaire est cherchée itérativement : un projecteur  $\mathbf{P}$  sur  $\ker(\mathbf{C}^T\mathbf{A})$  assure la recherche dans cet espace. Ainsi le résidu  $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}$  respecte la condition  $\mathbf{C}^T\mathbf{r} = 0$ . Autrement dit,  $\mathbf{x}$  est recherché sous la forme :

$$\mathbf{x} = \mathbf{x}_0 + \mathbf{P}\tilde{\mathbf{x}}.$$

Ce qui revient à des conditions sur  $\mathbf{P}$  et  $\mathbf{x}_0$  :

$$\begin{aligned} \mathbf{C}^T\mathbf{r}_0 &= \mathbf{C}^T(\mathbf{b} - \mathbf{A}\mathbf{x}_0) = 0, \\ \mathbf{C}^T\mathbf{A}\mathbf{P} &= 0 \end{aligned}$$

conduisant à :

$$\begin{aligned} \mathbf{x}_0 &= \mathbf{C}(\mathbf{C}^T\mathbf{A}\mathbf{C})^{-1}\mathbf{C}^T\mathbf{b}, \\ \mathbf{P} &= \mathbf{1} - \mathbf{C}(\mathbf{C}^T\mathbf{A}\mathbf{C})^{-1}\mathbf{C}^T\mathbf{A} \end{aligned}$$

Le système projeté devient alors :

$$\mathbf{A}\mathbf{P}\tilde{\mathbf{x}} = \mathbf{P}^T(\mathbf{b} - \mathbf{A}\mathbf{x}_0)$$

**Algorithme A.5 : Gradient Conjugué augmenté avec réorthogonalisation****Initialisation :**Calcul du projecteur  $\mathbf{P} = \mathbf{1} - \mathbf{C}(\mathbf{C}^T \mathbf{A} \mathbf{C})^{-1} \mathbf{C}^T \mathbf{A}$  ;Initialisation de la solution :  $\mathbf{x}_0 = \mathbf{C}(\mathbf{C}^T \mathbf{A} \mathbf{C})^{-1} \mathbf{C}^T \mathbf{b}$  ;Initialisation du résidu  $\mathbf{r}_0 = \mathbf{P}^T (\mathbf{b} - \mathbf{A} \mathbf{x}_0)$  ; $\mathbf{d}_0 = \mathbf{r}_0$  ;**tant que**  $\frac{\mathbf{r}_i^T \mathbf{r}_i}{\mathbf{r}_0^T \mathbf{r}_0} > \varepsilon$  **faire**     $\mathbf{p}_i = \mathbf{P}^T \mathbf{A} \mathbf{d}_i$  ;    Calcul du pas optimal  $\alpha_i = \frac{\mathbf{r}_i^T \mathbf{r}_i}{\mathbf{d}_i^T \mathbf{p}_i}$  ;    Calcul de la nouvelle solution  $\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{d}_i$  ;    Calcul du nouveau résidu  $\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i \mathbf{p}_i$  ;    **pour**  $0 \leq j \leq i$  **faire** Réorthogonalisation de la base de recherche         $\beta_i^j = -\frac{\mathbf{r}_{i+1}^T \mathbf{p}_j}{\mathbf{d}_j^T \mathbf{p}_j}$  ;    **fin**    Calcul de la nouvelle direction de recherche :  $\mathbf{d}_{i+1} = \mathbf{r}_{i+1} + \sum_{j \leq i} \beta_i^j \mathbf{d}_j$  ;**fin**

et l'algorithme pour un gradient conjugué qui en découle est l'algorithme A.5.

L'adjonction de cette contrainte optionnelle sur le résidu  $\mathbf{r}$  permet d'améliorer le conditionnement du système d'après DOSTÁL [1988], mais cette efficacité dépend essentiellement du choix fait pour l'opérateur  $\mathbf{C}$ . Le choix de  $\mathbf{C}$  reste un problème ouvert mais, dans le cadre de la résolution des équations de la mécanique par les méthodes de Schur, ce choix peut être guidé par les modes rigides des sous-structures (comme dans l'algorithme 1.1 BDD) ou d'autres considérations telles que des modes propres de sous-structures pour la méthode FETI-GenEO introduite par SPILLANE et RIXEN [2013].

**A.1.3 ORTHOMIN**

Lorsque  $\mathbf{A}$  n'est pas symétrique, CG n'est plus applicable. D'autres solveurs de Krylov, ayant des propriétés d'orthogonalité différentes, ont dû être développés.

L'algorithme ORTHOMIN a été introduit pour la première fois par VINSOME [1976]. Dans cette méthode, les directions de recherche sont orthogonales 2 à 2 au sens de la norme  $\mathbf{A}^T \mathbf{A}$  et non plus au sens de la norme  $\mathbf{A}$  comme c'est le cas pour le CG. De plus, la minimisation réalisée ici est la minimisation de la quantité  $\|\mathbf{x} - \mathbf{x}_i\|_{\mathbf{A}^T \mathbf{A}}$ . L'algorithme qui en découle est l'algorithme A.6.

---

**Algorithme A.6 : ORTHOMIN**

---

**Initialisation :**Initialisation du résidu  $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$  ; $\mathbf{p}_0 = \mathbf{r}_0$  ; $\mathbf{d}_0 = \mathbf{A}\mathbf{p}_0$  ;**tant que**  $\frac{\mathbf{r}_i^T \mathbf{r}_i}{\mathbf{r}_0^T \mathbf{r}_0} > \varepsilon$  **faire**    Calcul du pas optimal  $\alpha_i = \frac{\mathbf{d}_i^T \mathbf{r}_i}{\mathbf{d}_i^T \mathbf{d}_i}$  ;    Calcul de la nouvelle solution  $\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{p}_i$  ;    Calcul du nouveau résidu  $\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i \mathbf{d}_i$  ;    **pour**  $0 \leq j \leq i$  **faire** Réorthogonalisation de la base de recherche        |  $\beta_i^j = -\frac{\mathbf{d}_j^T \mathbf{A}\mathbf{r}_{i+1}}{\mathbf{d}_j^T \mathbf{d}_j}$  ;    **fin**    Calcul de la nouvelle direction de recherche : 
$$\begin{cases} \mathbf{p}_{i+1} = \mathbf{r}_{i+1} + \sum_{j \leq i} \beta_i^j \mathbf{p}_j \\ \mathbf{d}_{i+1} = \mathbf{A}\mathbf{r}_{i+1} + \sum_{j \leq i} \beta_i^j \mathbf{d}_j \end{cases} ;$$
**fin**

---

## A.2 Étude paramétrique de la méthode RBF adaptative de la section 2.3.3

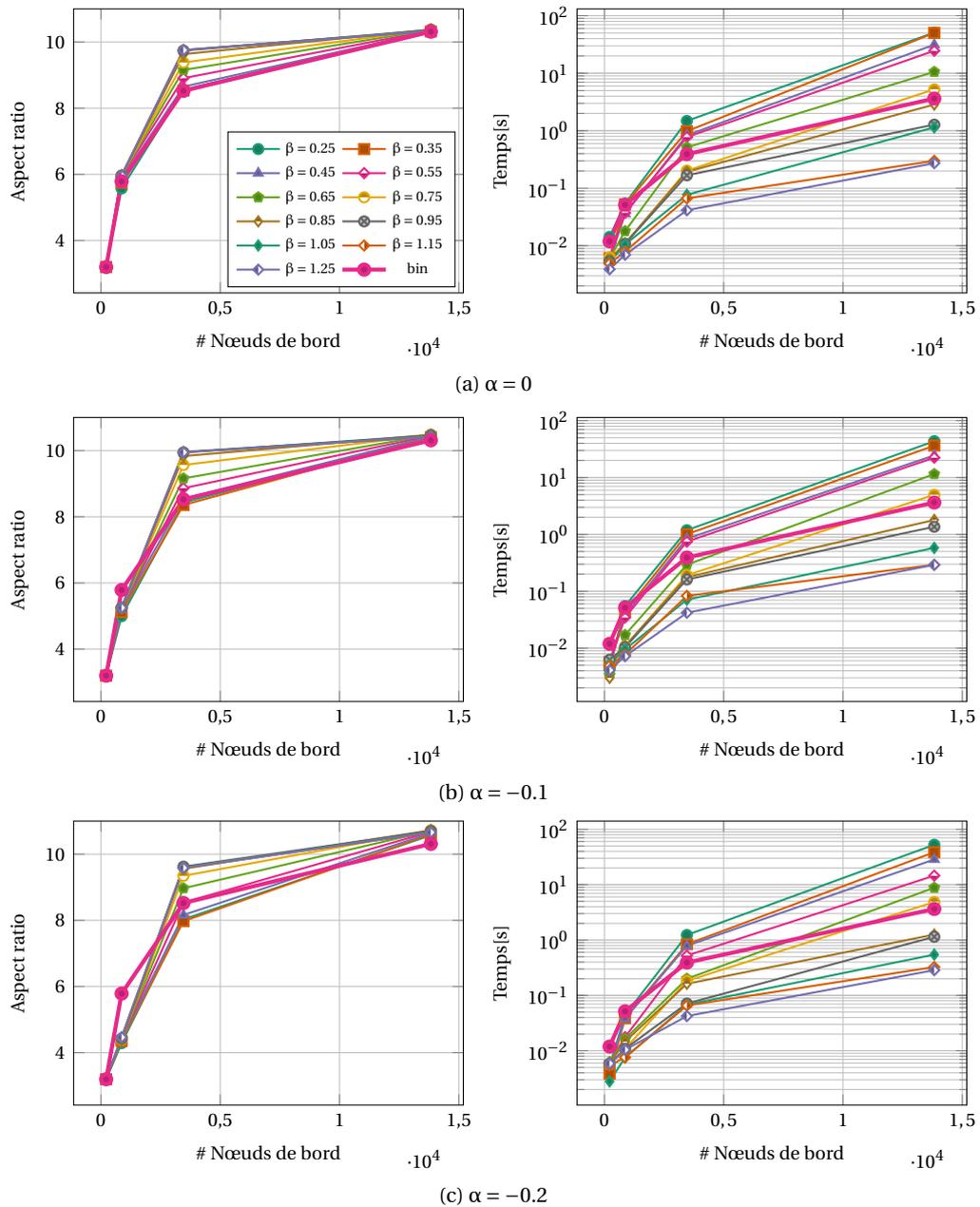


FIGURE A.1 – Influence de  $\alpha$  et  $\beta$  sur le temps de résolution de (1.28) par un solveur direct et la qualité du maillage (1/4)

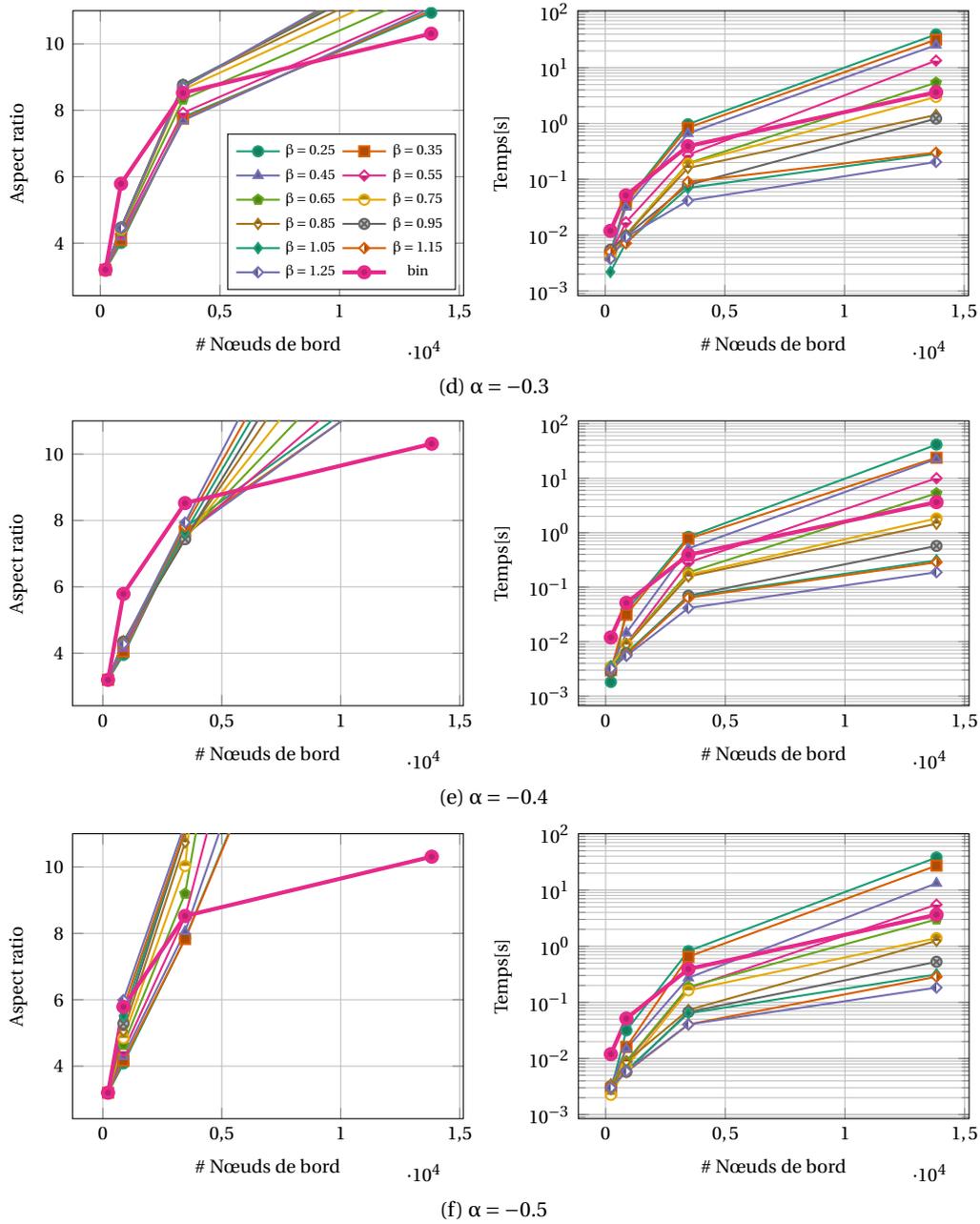


FIGURE A.1 – Influence de  $\alpha$  et  $\beta$  sur le temps de résolution de (1.28) par un solveur direct et la qualité du maillage (2/4)

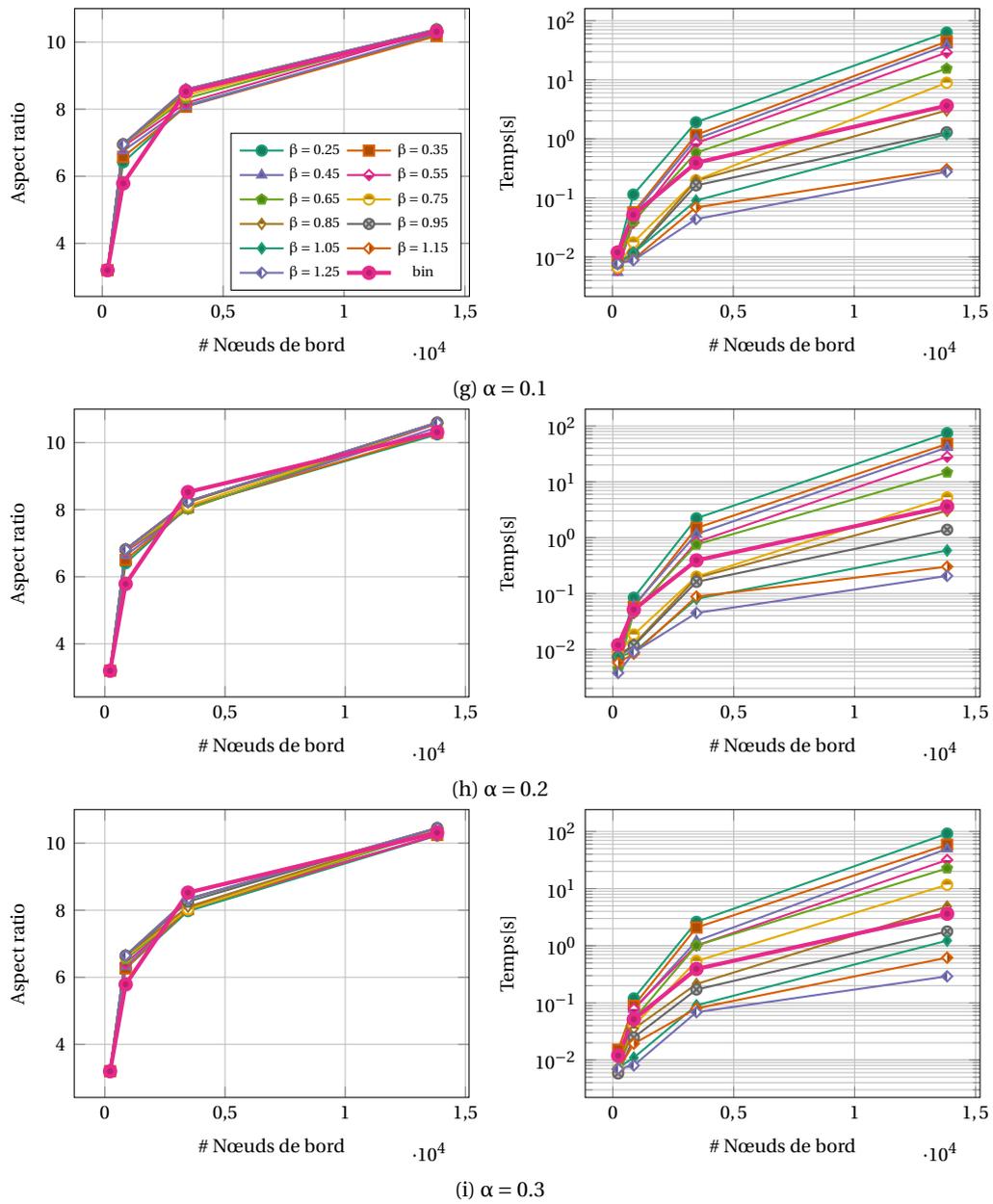


FIGURE A.1 – Influence de  $\alpha$  et  $\beta$  sur le temps de résolution de (1.28) par un solveur direct et la qualité du maillage (3/4)

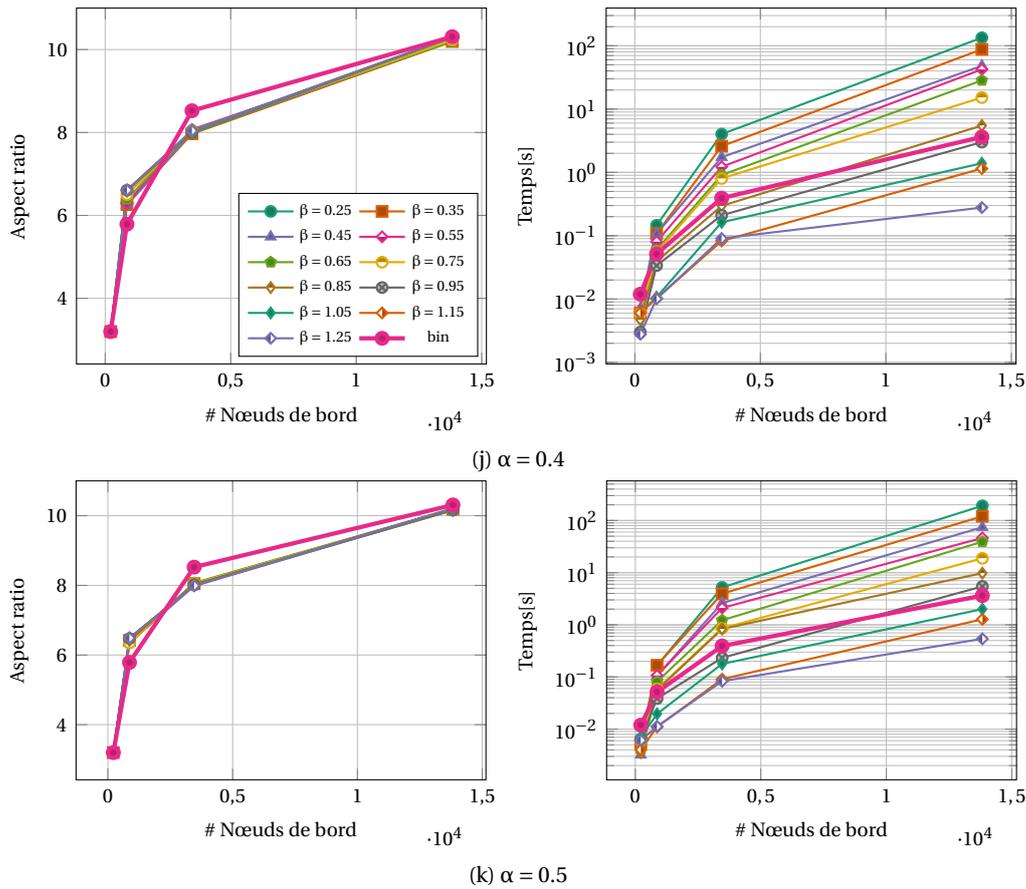


FIGURE A.1 – Influence de  $\alpha$  et  $\beta$  sur le temps de résolution de (1.28) par un solveur direct et la qualité du maillage (4/4)

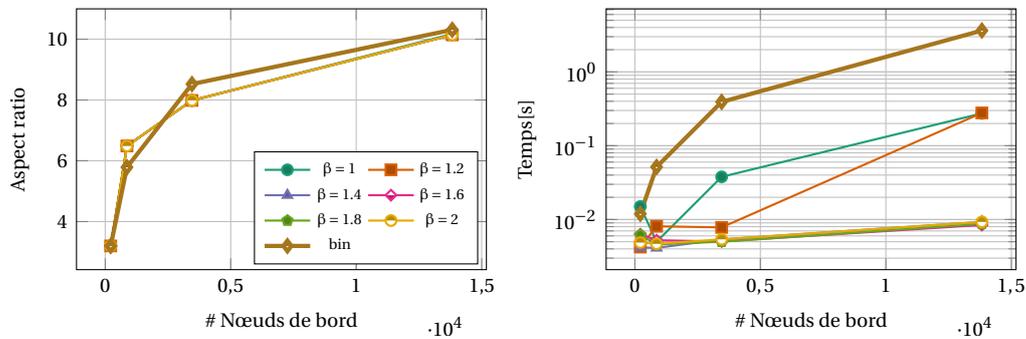


FIGURE A.2 – Performances de la méthode pour des valeurs plus importantes de  $\beta$  pour  $\alpha = 0.5$

### A.3 Références

BARRETT, R., M. BERRY, T. F. CHAN, J. DEMMEL, J. DONATO, J. DONGARRA, V. EIJKHOUT, R. POZO, C. ROMINE et H. VAN DER VORST. 1994, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, Society for Industrial and Applied Mathematics, ISBN 978-0-89871-328-2 978-1-61197-153-8, doi : [10.1137/1.9781611971538](https://doi.org/10.1137/1.9781611971538). URL <http://epubs.siam.org/doi/book/10.1137/1.9781611971538>. I

CHAPMAN, A. et Y. SAAD. 1997, «Deflated and Augmented Krylov Subspace Techniques», *Numerical Linear Algebra with Applications*, vol. 4, n° 1, doi : [10.1002/\(SICI\)1099-1506\(199701/02\)4:1<43::AID-NLA99>3.0.CO;2-Z](https://doi.org/10.1002/(SICI)1099-1506(199701/02)4:1<43::AID-NLA99>3.0.CO;2-Z), p. 43–66, ISSN 1099-1506. IV

- DOSTÁL, Z. 1988, «Conjugate gradient method with preconditioning by projector», *International Journal of Computer Mathematics*, vol. 23, n° 3-4, doi : [10.1080/00207168808803625](https://doi.org/10.1080/00207168808803625), p. 315–323, ISSN 0020-7160, 1029-0265. URL <http://www.tandfonline.com/doi/abs/10.1080/00207168808803625>. V
- SAAD, Y. 1997, «Analysis of Augmented Krylov Subspace Methods», *SIAM Journal on Matrix Analysis and Applications*, vol. 18, n° 2, doi : [10.1137/S0895479895294289](https://doi.org/10.1137/S0895479895294289), p. 435–449, ISSN 0895-4798, 1095-7162. URL <http://epubs.siam.org/doi/10.1137/S0895479895294289>. I, IV
- SAAD, Y. 2003, *Iterative Methods for Sparse Linear Systems*, Other Titles in Applied Mathematics, Society for Industrial and Applied Mathematics, ISBN 978-0-89871-534-7, doi : [10.1137/1.9780898718003](https://doi.org/10.1137/1.9780898718003). URL <https://epubs.siam.org/doi/book/10.1137/1.9780898718003>. I
- SPILLANE, N. et D. J. RIXEN. 2013, «Automatic spectral coarse spaces for robust finite element tearing and interconnecting and balanced domain decomposition algorithms», *International Journal for Numerical Methods in Engineering*, vol. 95, n° 11, doi : [10.1002/nme.4534](https://doi.org/10.1002/nme.4534), p. 953–990, ISSN 1097-0207. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.4534>. V
- VINSOME, P. K. W. 1976, «Orthomin, an Iterative Method for Solving Sparse Sets of Simultaneous Linear Equations», dans *4th Symposium on Numerical Simulation of Reservoir Performance*, Society of Petroleum Engineers, ISBN 978-1-55563-751-4, p. 149–159, doi : [10.2118/5729-MS](https://doi.org/10.2118/5729-MS). URL <https://www.onepetro.org/conference-paper/SPE-5729-MS>. V



# ANNEXE B

## LISTE DES ACRONYMES

- AFT** *Advancing Front Technique*. ix, 39, 40, 42
- AMPFETI** *Adaptive MultiPreconditioned FETI*. 33
- BDD** *Balanced Domain Decomposition*. xv, 18, 24, 28, 32–34, 79, 108, 117, 128, V, XIII
- BDD-C** *BDD by Constrained*. 24
- B-REP** *Boundary REPresentation*. 80
- CAO** *Conception Assistée par Ordinateur*. 3, 44, 45, 77, 80, 103, 109, XV
- CFD** *Computational Fluid Dynamics*. 54
- CG** *Gradient Conjugué, ou Conjugate Gradient*. x, xv, 18, 20, 24, 28, 29, 32–34, 87, 89–91, 93, I, II, V, XIV
- CP** *Continuous Polynomials*. 55, 56, 59
- CTPS** *Continuous TPS*. 55
- DDL** degrés de liberté. xi, xiii, 2, 4, 13, 15–17, 23–25, 30, 52, 96–98, 100, 107, 108, 116, 120, 121, 123, 139
- EF** *Éléments Finis*. 3, 12, 39, 52, 57, 95, 103, 121, 122, 140
- FEAD** *Finite Elements in Analysis and Design*. 77
- FETI** *Méthode de décomposition de domaine sans recouvrement duale*. xi, xiii, xv, 5, 18, 28–34, 79, 95–99, 103, 105, 106, 108, 109, 117, 122, 128, 133, 140, XIII, XIV
- FETI-2LM** *FETI with two Lagrange Multipliers*. xi, 5, 20, 22, 115, 116, 122, 127, 132, 134, 139
- FETI-DP** *Dual-Primal-FETI*. 18, 24, 30
- GenEO** *Generalized Eigenproblems in the Overlaps*. 31, 32, 99, 134, V
- GMRes** *généralisation de la méthode de minimisation du résidu, ou Generalized Minimal Residual*. xi, 93, 125, 126, 130, 132
- HEM** *Heavy Edge Matching*. 37
- HFETI** *Hybrid FETI*. 18
- IDW** *Inverse Distance Weighting* ou pondération par l'inverse de la distance. 54, 56
- ILU** *LU Incomplète*. x, 91
- IMQB** *Inverse MQB*. 55

**IQB** *Inverse QB*. 55

**LaTIn** *Large Time Increments*. 20, 32, 117, 128

**MPI** *Message Passing Interface*. 3, 38, 95, 103, 106, 107, 140

**MQB** *MultiQuadratic Biharmonics*. 55, XIII

**MUMPS** *MUltifrontal Massively Parallel sparse direct Solver*. 11

**PARDISO** *PARallel sparse DIrect SOLver*. 11

**PCG** *CG Préconditionné, ou Preconditioned CG*. x, 90–93

**QB** *Quadratic Biharmonics*. 55, XIII

**RBF** *Radial Basis Functions* ou fonctions de base radiales. x, xi, 54–57, 63, 77, 86–95, 100, 102, 108, 109, 139

**S-FETI** *Simultaneous FETI*. xv, 32, 33

**SPH** *Smoothed Particle Hydrodynamics*. ix, 40–42

**STEP** *STandard for the Exchange of Product model data*. 80

**STL** *STereoLithography*. 80

**TFETI** *Total FETI*. 18

**TFI** *TransFinite Interpolation*. 52

**TPS** *Thin Plate Spline*. 55, 59, XIII

**VER** *Volume Élémentaire Représentatif*. 2, 3, 140

- CATIA V5** Un logiciel de CAO créé au départ par la société Dassault Aviation pour ses propres besoins. 80
- cluster** Ensemble de plusieurs machines indépendantes appelés nœuds de calcul communiquant les uns avec les autres afin de dépasser les limitations d'une machine seule. 3, 95
- communicateur** Un communicateur est un ensemble de processus susceptibles de communiquer entre eux. 83
- conditionnement** Soit une matrice  $\mathbf{K}$  inversible, on appelle  $\kappa(\mathbf{K}) = \|\mathbf{K}^{-1}\|_2 \|\mathbf{K}\|_2$  son conditionnement. Plus simplement, c'est le rapport entre la plus grande et la plus petite valeur propre de  $\mathbf{K}$ . xi, 11, 20, 24, 25, 27, 28, 30, 31, 79, 87, 90, 98, 99, 108, 114, XVII
- extensibilité** Capacité qu'à une méthode parallèle à rester efficace lorsque la taille du problème à résoudre augmente ou lorsque le nombre de ressources disponibles augmente. xi, xiii, 11, 18, 26–28, 45, 82, 95, 96, 99, 100, 106–109
- Gmsh** Un logiciel de génération de maillage éléments finis tridimensionnel avec des fonctions de pré/post-traitement intégrées. 56
- mésosstructure** Structure à l'échelle des torons pour un matériaux composite tissé 3D. 2
- non-structuré** Se dit d'un maillage dont on ne peut pas connaître la connectivité uniquement avec les indices des nœuds. 58
- NP-complet** Un problème NP-complet est un problème dont on ne peut trouver la solution en un temps polynômial alors que l'on peut vérifier si une solution est plus optimale qu'une autre facilement. 35
- nœud de calcul** C'est une machine physique composée de plusieurs processeurs possédant sa propre mémoire vive. 3
- patch d'éléments** Ensemble d'éléments partageant un même nœud. 52
- point multiple** Un nœud du maillage qui appartient à plus de trois sous-domaines. 13, 15
- soufflante** Ensemble de grandes aubes permettant la propulsion de l'air à l'intérieur du moteur et permettant sa première compression. C'est l'hélice qu'un observateur voit à l'avant du moteur. 1, 2
- structuré** Se dit d'un maillage dont les nœuds sont définis uniquement par des indices. Ce type de maillage permet de trouver facilement les voisins d'un nœud sans avoir à faire une recherche sur toute la grille. 58
- toron** Ensemble de fibres formant un plus gros fil. 3, XV



# ANNEXE D

## NOTATIONS

- $\Omega^{(s)}$  Quantité liée au sous-domaine  $\Omega^{(s)}$ . 12
- $\diamond$  Quantité en notation par blocs : concaténation en colonne. 16
- $\diamond$  Quantité en notation par blocs : concaténation selon la diagonale. 16
- $\diamond$  Quantité en notation par blocs : concaténation en ligne. 16
  
- $\Gamma$  Interface globale entre les sous-domaines. ix, 13, XVII
- $\Gamma^{(s)}$  Interface locale : c'est une restriction de  $\Gamma$  au sous-domaine  $\Omega^{(s)}$ . ix, 13, 15, 22
- $\Gamma_A$  Interface globale primale. 13, 22
- $\Gamma_B$  Interface globale duale. 13, 22
- $\Gamma^{(i,j)}$  Interface entre les sous-domaines  $\Omega^{(i)}$  et  $\Omega^{(j)}$ . ix, 13
  
- $\mathbf{Q}_b^{(s)}$  Impédance d'interface. 21
  
- $\mathbf{K}$  Matrice de raideur de la structure complète  $\Omega$ . 12
- $\mathbf{K}^{(s)}$  Matrice de raideur liée au sous-domaine  $\Omega^{(s)}$ . 16
  
- $\Omega$  Structure complète discrétisée. ix, 11, 12, 24, 114, XVII
- $\Omega^{(s)}$  Sous-structure discrétisée numéro s. 11–14, 114, 116–119, 122, 123, XVII
- $\Omega^{(\bar{s})}$  Sous-structure discrétisée complémentaire au domaine  $\Omega^{(s)}$ . 114, 116, 122
  
- $\mathbf{S}_p^{(s)}$  Complément de Schur primal local de la sous-structure  $\Omega^{(s)}$ . 17
- $\mathbf{S}_d^{(s)}$  Complément de Schur dual local de la sous-structure  $\Omega^{(s)}$ . 20
- $\mathbf{S}_m^{(s)}$  Complément de Schur mixte local de la sous-structure  $\Omega^{(s)}$ . 22
  
- $\kappa(\bullet)$  Conditionnement d'une matrice. 24





**Titre :** Une approche à deux niveaux pour le calcul de structures haute performance : décomposition – maillage – résolution

**Mots clés :** décomposition de domaine, sous-structuration à deux niveaux, maillage parallèle, impédance d'interface à deux niveaux, structures hétérogènes

**Résumé :** La simulation numérique représente encore un aspect minoritaire de la certification de pièces critiques dans l'industrie. Pourtant, elle permettrait de réaliser de fortes économies lors de la conception, en évitant de réaliser des essais onéreux sur des pièces réelles.

En effet, lorsque le matériau est architecturé il existe des structures internes ayant un comportement mécanique radicalement différent en des zones distinctes de la structure, il devient difficile, voire impossible, de réaliser ces simulations en des temps raisonnables du fait du nombre important d'inconnues nécessaires à l'obtention d'une réponse fiable de la structure.

Pour obtenir cette réponse, l'utilisation de méthodes de résolution parallèle de problèmes de grande taille est nécessaire. Les méthodes de décomposition de domaine, qui font partie de cette catégorie, sont les méthodes qui sont explorées durant cette thèse. L'objectif est donc de rendre possible ces simulations à l'aide de ces méthodes.

En effet, la résolution du problème mais aussi le maillage de la structure deviennent coûteux et l'usage de méthodes parallèles devient indispensable. Pour cela, une méthode de sous-structuration à deux niveaux est proposée. Elle vise à produire en phase de préparation des données des sous-domaines réguliers et homogènes pouvant être maillés en parallèle. Par ailleurs, elle conduit à une forte réduction du conditionnement de problèmes à fortes hétérogénéités résolus par un solveur FETI. Une méthode de décomposition de domaine mixte avec impédance d'interface à deux niveaux adaptée à cette sous-structuration a ensuite pu être développée.

L'objectif à long terme est, ici, de traiter des problèmes de complexité quasi-industrielle tels que des calculs à l'échelle de la structure complète sur des matériaux multi-échelles comme les composites tissés tridimensionnels utilisés de plus en plus intensivement dans l'industrie aéronautique par exemple.

**Title:** A two-level approach for high performance structural computations: substructuring – meshing – resolution

**Keywords:** Domain Decomposition Methods, two-level substructuring, parallel meshing, two-level Robin conditions, heterogeneous structures

**Abstract:** Numerical simulations represent a minor part of the certification process for critical parts in the industry. However, it would result in significant cost savings during conception phases, avoiding expensive real tests.

Indeed, in cases of localized strong heterogeneities across all the structure, it becomes hard, if not impossible, to run successfully these simulations in reasonable times because of a too large number of unknowns needed for a reliable answer of the structure.

To obtain this answer, large scale parallel solving methods are necessary. Domain decomposition methods, which are part of it, are the ones investigated during this thesis.

The goal is to make these simulations possible thanks to domain decomposition methods. Indeed, the

resolution of the problem but also the meshing of the structure become expensive and the use of parallel methods becomes essential. For this purpose, a two-level substructuring method is proposed. It aims at producing, during the pre-processing step, regular-shaped and homogeneous subdomains possibly meshed in parallel. In addition, it allows to a significant reduction of the condition number for strongly heterogeneous problems solved by a FETI solver. A mixed domain decomposition method with a two-level Robin condition which is adapted to this decomposition could then be developed.

The long term objective is to deal with problems with a quasi-industrial complexity like computations at the global structural scale with multi-scale materials such as tridimensional woven composites which are used increasingly intensively in the aeronautical industry for instance.

