



**HAL**  
open science

# Méthodes de décomposition de domaine et méthodes d'accélération pour les problèmes multichamps en mécanique non-linéaire

Pierre Gosselet

► **To cite this version:**

Pierre Gosselet. Méthodes de décomposition de domaine et méthodes d'accélération pour les problèmes multichamps en mécanique non-linéaire. Mécanique [physics.med-ph]. Université Pierre et Marie Curie - Paris VI, 2003. Français. NNT : . tel-00277771

**HAL Id: tel-00277771**

**<https://theses.hal.science/tel-00277771>**

Submitted on 7 May 2008

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

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

# THÈSE DE DOCTORAT DE L'UNIVERSITÉ PARIS 6

Spécialité

## MÉCANIQUE

Présentée par

**Pierre GOSSELET**

pour obtenir le grade de :

**DOCTEUR DE L'UNIVERSITÉ PARIS 6**

## Méthodes de décomposition de domaine et méthodes d'accélération pour les problèmes multichamps en mécanique non-linéaire

soutenue le 11 décembre 2003 devant le jury composé de :

- Président :* Olivier Débordes, professeur à l'EGIM
- Rapporteurs :* Nicolas Moes, professeur à l'école centrale de Nantes  
Daniel Rixen, professeur à la faculté de Delft
- Examineurs :* Frédéric Feyel, chargé de recherche à l'ONERA  
Françoise Léné, professeur à l'université Paris 6
- Directeur de thèse :* Christian Rey, professeur à l'ENS de Cachan



---

# Remerciements

Je remercie toutes les personnes qui m'ont permis de préparer cette thèse dans les meilleures conditions, notamment :

- Les membres de mon jury pour l'intérêt qu'ils ont porté à mes recherches : Olivier Débordes qui a accepté d'être président, Nicolas Moes et Daniel Rixen qui ont été de très consciencieux rapporteurs, Frédéric Feyel qui a bien voulu être examinateur.
- Christian Rey, mon directeur de thèse, brillant, exigeant mais toujours amical. Son dynamisme, son accessibilité, son intégrité et son humanité ont rendu ces trois années productives et très agréables. J'espère le retrouver au travail pour de nombreuses collaborations et sur les terrains de squash pour de nouvelles parties acharnées.
- Françoise Léné, directrice du  $LM^2S$ , pour m'avoir accueilli dans ce laboratoire où je me suis senti très à l'aise pour travailler dans de bonnes conditions matérielles (merci Philippe) et dans la bonne humeur.
- L'ensemble des membres du  $LM^2S$  pour leur compétence et leur amitié, et plus particulièrement Vincent qui a relu ma thèse, ainsi que Boris toujours disponible, humble et avisé.
- Daniel Rixen et Frédéric Feyel avec qui j'ai eu le plaisir de collaborer durant ma thèse, en espérant avoir de nombreuses occasions de travailler avec eux, tant leur talent et leur gentillesse m'ont marqué.
- François-Xavier Roux dont j'ai eu la chance de pouvoir suivre un cours très formateur sur les techniques de calcul.
- Le LMT-Cachan où j'ai eu quelques occasions de travailler, je sais que je peux y trouver des gens de grande qualité.
- L'ENS de Cachan, le CNRS et l'université Pierre et Marie Curie qui ont successivement assuré le financement de cette thèse.
- Le CINES et le pôle informatique Paris-Sud dont j'ai pu utiliser les super-calculateurs.

Je remercie plus généralement les personnes qui m'accompagnent et sur lesquelles j'ai toujours pu compter, mes parents et ma soeur, mes amis géniaux et fidèles : Erika & Eric, Delphine & Nicolas, Sophie & Anthony, Claire, Scott, Juliette...



---

# Résumé

## MÉTHODES DE DÉCOMPOSITION DE DOMAINE ET MÉTHODES D'ACCÉLÉRATION POUR LES PROBLÈMES MULTICHAMPS EN MÉCANIQUE NON-LINÉAIRE

Nous développons des algorithmes parallèles pour la résolution de problèmes non-linéaires de grande taille. Les cadres d'application sont la simulation de matériaux hyperélastiques incompressibles en grandes déformations et l'étude des milieux poreux, dont les modélisations choisies font apparaître des inconnues en déplacement et en pression.

Nous retenons une stratégie éléments-finis associée à un solveur Newton-Raphson et une décomposition de domaine sans recouvrement combinée à un solveur de Krylov.

Nous proposons des améliorations pour adapter ces approches à nos problèmes, puis pour des cas plus exigeants nous définissons une nouvelle approche de décomposition de domaine, appelée approche hybride, permettant de mieux respecter la physique des phénomènes et unifiant les approches classiques. Nous proposons également des stratégies d'accélération du processus non-linéaire. Enfin un cadre orienté objet est exposé pour la mise en oeuvre de l'ensemble des méthodes proposées.

MOTS-CLEFS : décomposition de domaine sans recouvrement, Newton-Raphson, solveur itératif de Krylov, problèmes multichamps, calcul parallèle, programmation orientée objet, élastomères, poroélasticité



---

# Abstract

## DOMAIN DECOMPOSITION AND ACCELERATION METHODS FOR MULTIFIELDS PROBLEMS IN NONLINEAR MECHANICS

We develop herein parallel algorithms for the solution to large nonlinear problems. Applications deal with the simulation of hyperelastic incompressible material underlying large deformations and with the study of porous media. The chosen modelizations require both displacement and pressure unknown fields.

We use a finite element strategy associated with a Newton-Raphson solver and a non-overlapping domain decomposition associated with a Krylov iterative solver.

We propose improvements to adapt these approaches to our problems. For more complex cases we define a novel domain decomposition approach, called hybrid approach, which enables to be more respectful with the physics of the phenomena and which unifies the classical approaches. We also propose acceleration strategies for the nonlinear resolution. Eventually an object-oriented framework for the implementation of all the proposed methods is exposed.

KEY-WORDS : non-overlapping domain decomposition, Newton-Raphson, Krylov iterative solver, multifields problems, parallel computing, object-oriented programming, elastomer, poroelasticity





---

# Table des matières

<b>Remerciements</b>	<b>i</b>
<b>Résumé</b>	<b>iii</b>
<b>Abstract</b>	<b>v</b>
<b>Introduction</b>	<b>1</b>
<b>Notations</b>	<b>5</b>
<b>I Méthodes de décomposition de domaine sans recouvrement en élasticité linéaire</b>	<b>7</b>
<b>1 Sous-structuration</b>	<b>11</b>
1.1 Obtention des problèmes condensés . . . . .	12
1.2 Approches classiques . . . . .	16
1.3 Application à la méthode des éléments-finis . . . . .	17
<b>2 Solveurs itératifs de Krylov</b>	<b>25</b>
2.1 Principe des solveurs de Krylov . . . . .	26
2.2 Solveurs usuels . . . . .	27
2.3 Etude de la convergence, préconditionnement . . . . .	29
2.4 Etude de la convergence, spectre de Ritz . . . . .	29
2.5 Méthodes Krylov-augmenté, mise en oeuvre par projecteur . . . . .	31
<b>3 Résolution itérative du problème mécanique condensé à l'interface</b>	<b>33</b>
3.1 Préconditionneurs . . . . .	34
3.2 Problèmes grossiers . . . . .	35
3.3 Facteurs influençant les performances des méthodes de décomposition de domaine	37
3.4 Notation par blocs des méthodes de décomposition de domaine . . . . .	41

<b>II</b>	<b>Etude et amélioration des méthodes classiques de décomposition de domaine</b>	<b>45</b>
<b>4</b>	<b>Comparaison des approches primale et duale de décomposition de domaine</b>	<b>49</b>
4.1	Interprétation de l'initialisation dans l'approche duale . . . . .	51
4.2	Répartition des efforts dans l'approche duale . . . . .	51
4.3	Amélioration de l'initialisation de l'approche duale . . . . .	52
4.4	Validation . . . . .	53
4.5	Conclusions . . . . .	57
<b>5</b>	<b>Prise en compte de la (quasi)-incompressibilité</b>	<b>59</b>
5.1	Étude de l'élastomère . . . . .	60
5.2	Décomposition de domaine avec pression discontinue . . . . .	61
5.3	Validation et conclusions . . . . .	63
<b>III</b>	<b>Approche hybride de décomposition de domaine</b>	<b>65</b>
<b>6</b>	<b>Présentation de l'approche hybride</b>	<b>69</b>
6.1	Formulation hybride . . . . .	70
6.2	Préconditionnement du problème hybride . . . . .	74
6.3	Condition d'optimalité du préconditionneur . . . . .	74
6.4	Conclusion . . . . .	75
<b>7</b>	<b>Gestion des problèmes grossiers dans l'approche hybride</b>	<b>77</b>
7.1	Approche à un projecteur . . . . .	79
7.2	Approche à deux projecteurs . . . . .	79
7.3	Approche à trois projecteurs . . . . .	80
7.4	Conclusion . . . . .	81
<b>8</b>	<b>Une première validation de l'approche hybride</b>	<b>83</b>
8.1	Cisaillement d'un carré élastique . . . . .	84
8.2	Compactage d'un sol . . . . .	85
8.3	Conclusions sur l'approche hybride . . . . .	88
<b>IV</b>	<b>Accélération Krylov pour la multirésolution</b>	<b>91</b>
<b>9</b>	<b>Cadre de l'étude / algorithme de référence</b>	<b>95</b>
9.1	Linéarisation de Newton Raphson . . . . .	96
9.2	Algorithme Krylov-augmenté en multirésolution . . . . .	97

---

<b>10 Méthode GIRKS</b>	<b>99</b>
10.1 Rappel de la méthode . . . . .	100
10.2 Application au calcul de butée flexible par approche primale . . . . .	102
10.3 Conclusion . . . . .	103
<b>11 Méthode SRKS</b>	<b>105</b>
11.1 Principe de la méthode . . . . .	106
11.2 Cas du gradient conjugué avec préconditionneur positif . . . . .	106
11.3 Application au calcul du flambage d'une poutre composite par approche duale	107
11.4 Conclusion . . . . .	111
<b>V Mise en oeuvre orientée objet des méthodes et solveurs présentés</b>	<b>113</b>
<b>12 Mise en place du problème</b>	<b>117</b>
12.1 Organisation des données au sein des sous-domaines . . . . .	118
12.2 Définition de la formulation / mise en données . . . . .	119
<b>13 Organisation du code</b>	<b>121</b>
13.1 Solveurs . . . . .	122
13.2 Gestion de l'interface . . . . .	123
13.3 Problèmes grossiers . . . . .	124
13.4 L'objet PARALLEL_FORMULATION . . . . .	126
13.5 Possibilités d'extension . . . . .	128
<b>Conclusion</b>	<b>129</b>
<b>Bibliographie</b>	<b>133</b>
<b>A FETI-DP</b>	<b>143</b>
<b>B Butée flexible SNECMA-moteurs-de-fusées</b>	<b>145</b>
B.1 Présentation de la butée flexible . . . . .	145
B.2 Propriétés mécaniques . . . . .	147
B.3 Décompositions employées . . . . .	149



---

# Liste des figures

1	Décomposition du problème de Schwarz . . . . .	9
1.1	Problème modèle . . . . .	12
1.2	Décomposition en deux sous-domaines . . . . .	12
1.3	Définition de l'interface pour une décomposition en $N$ sous-domaines . . . . .	13
1.4	Décomposition conforme en trois sous-structures . . . . .	18
1.5	Numérotations locales, numérotations de l'interface, opérateurs de trace et assemblage . . . . .	19
1.6	Suppression des redondances dans l'approche duale . . . . .	22
3.1	Types d'hétérogénéité . . . . .	40
4.1	Cube hétérogène décomposé . . . . .	50
4.2	Convergence des approches primale et duale (pb. fig. 4.1) . . . . .	50
4.3	Cube hétérogène (échiquier, fig. 4.1) : convergence pour les répartitions classiques	54
4.4	Cube hétérogène (échiquier, fig. 4.1) : convergence avec la nouvelle initialisation	54
4.5	Cube avec feuillets hétérogènes . . . . .	55
4.6	Cube incliné avec feuillets hétérogènes . . . . .	55
4.7	Poutre composite à $3 \times 3 \times 3$ sous-structures . . . . .	56
4.8	Performances pour la flexion de la poutre composite . . . . .	57
5.1	Décomposition de la butée flexible en 6 sous-domaines . . . . .	63
6.1	Opérateurs de trace et d'assemblage . . . . .	70
8.1	Chargement en cisaillement . . . . .	84
8.2	Définition des ensembles de degrés de liberté . . . . .	84
8.3	Conditions aux limites sur un massif poreux . . . . .	86
8.4	Essai de compactage de sol . . . . .	87
8.5	Evolution du résidu GMRes . . . . .	89
9.1	Algorithme Newton-Raphson . . . . .	96

10.1 Performances de GIRKS au cours du processus non-linéaire . . . . .	103
11.1 Flambage d'une poutre composite . . . . .	108
11.2 Convergence des valeurs de Ritz . . . . .	109
11.3 Action de SRKS : nombre de contraintes . . . . .	109
11.4 Action de SRKS : nombre d' itérations . . . . .	109
11.5 Action de SRKS : temps CPU . . . . .	109
11.6 Spectre de Ritz sans contrainte . . . . .	110
11.7 Spectre de Ritz avec réutilisation sélective . . . . .	110
11.8 Evolution du résidu sans contrainte . . . . .	110
11.9 Evolution du résidu avec réutilisation sélective . . . . .	110
12.1 Données issues de la décomposition . . . . .	118
12.2 Classe gérant les échanges . . . . .	119
12.3 Relation solveur / formulation . . . . .	120
13.1 Organisation des solveurs . . . . .	122
13.2 Classe vecteur d'interface . . . . .	123
13.3 Les opérateurs d'interface . . . . .	125
13.4 Les problèmes grossiers . . . . .	126
13.5 La formulation parallèle . . . . .	127
B.1 Position de la butée flexible . . . . .	145
B.2 Discrétisation de la butée . . . . .	146
B.3 Rigidité et angle de braquage . . . . .	147
B.4 Flambage d'une armature en acier . . . . .	148
B.5 Evolution au cours de la compression . . . . .	148
B.6 Décompositions en calottes sphériques . . . . .	149
B.7 Décomposition mixte 3axi-6rota . . . . .	149
B.8 Allure des sous-domaines . . . . .	150
B.9 Allure des sous-domaines . . . . .	150

---

# Liste des tableaux

4.1	Performances des répartitions pour le cube fig. 4.1 . . . . .	55
4.2	Performances des répartitions sur les problèmes fig. 4.5 et 4.6 . . . . .	56
5.1	Performances des différents <i>scalings</i> sur la butée flexible décomposée en 6 ca- lottes (élément $Q_2 - P_1$ ) . . . . .	63
8.1	Convergence de l'approche hybride pour diverses configurations . . . . .	85
8.2	Performances de l'approche hybride sur le problème de compactage . . . . .	88
10.1	Performances numériques calcul Parallèle/Séquentiel (sans flambage) . . . . .	102
10.2	Performances numériques calcul Parallèle/Séquentiel (flambage) . . . . .	103
11.1	Symétrisation du gradient conjugué préconditionné . . . . .	107





---

# Liste des algorithmes

I.1	Procédure d'Arnoldi . . . . .	26
I.2	GMRes . . . . .	27
I.3	Gradient conjugué . . . . .	28
I.4	Gradient conjugué réorthogonalisé . . . . .	28
I.5	Approche primale 2 niveaux . . . . .	37
I.6	Approche duale 2 niveaux . . . . .	38
IV.1	GIRKS pour gradient conjugué projeté . . . . .	101



---

# Introduction

**E**NONCÉE en 1965, la loi de Moore prévoyait un doublement des capacités de calcul des ordinateurs tout les 18 mois. Cette prédiction, basée sur une prévision de l'augmentation de la capacité de gravure des circuits intégrés, est toujours d'actualité pour les machines individuelles, malgré le défi technologique croissant qu'elle représente. Cependant, on observe que les calculateurs scientifiques annoncent des performances jusqu'à 25 fois supérieures à cette loi. Ce surfacteur est principalement dû au parallélisme massif des nouvelles architectures où des très nombreux processeurs collaborent (plusieurs milliers sur les ASCI américains), ce qui conduit à des machines incroyables comme le "simulateur de terre japonais" évalué à 35 000 milliards d'opérations par seconde (35 TFlops/s) sur le benchmark LINPARK.

Ainsi au niveau des architectures, la collaboration entre de très nombreux processeurs est maintenant aisée à obtenir grâce à des techniques onéreuses mais largement maîtrisées. Par contre obtenir la collaboration entre les algorithmes au sein des codes de calcul est loin d'être aussi facile. En effet, un algorithme candidat au parallélisme doit permettre de décomposer un problème "global" en un ensemble de sous-problèmes aussi indépendants que possibles et nécessitant peu de synchronisations, chacune de ces étapes de communication devant consister en des échanges limités tant en taille de données qu'en nombre de processeurs à mettre en relation.

La notion de décomposition, dans le cadre de la résolution d'une équation aux dérivées partielles  $\mathcal{L}u = f$  posée sur un domaine  $\Omega$  traduisant la modélisation continue d'un phénomène physique, renvoie à trois grandes classes de techniques de résolution :

- les décompositions d'opérateur  $\mathcal{L} = \sum_i \mathcal{L}_i$  qui conduisent typiquement à des méthodes de résolution par directions de recherche alternées (algorithmes de type Uzawa),
- les décompositions d'espace fonctionnel  $u = \sum_i u_i \Phi_i$  dont sont membres les méthodes éléments-finis et les méthodes modales,
- les décompositions de domaine  $\Omega = \bigcup_i \Omega^{(i)}$ , entre autres notons celles de Schwarz (avec ou sans recouvrement) et de Schur (sans recouvrement).

Pour les deux premières classes, des choix judicieux permettent de découpler complètement la résolution de certains sous-problèmes entraînant une parallélisation extrême de la charge de calcul (par exemple, un processeur par mode propre d'une structure). Cependant dans ces deux cas, lors des synchronisations la taille de l'information à transmettre est celle du domaine complet, et certains pré-et-post-*processings* nécessitent des calculs sur l'ensemble du domaine. Par contre, les décompositions de domaine Schur-Krylov consistent à résoudre le problème condensé sur l'interface entre les sous-domaines par un solveur itératif de Krylov, les calculs sur chaque sous-domaine peuvent être réalisés en parallèle et nécessitent d'échanger des données de la taille de l'interface entre sous-domaines voisins. Dans certains cas une information globale doit également être transmise (échange *all to all*) cependant sa taille reste très petite devant les autres dimensions du problème.

La stratégie de résolution que nous développons pour le calcul non-linéaire de structure fait appel à l'ensemble des méthodes de décomposition. Ainsi nous combinons décomposition

d'espace fonctionnel par éléments-finis pour discrétiser le problème en espace, décomposition conforme du domaine (*ie* du maillage), puis décomposition d'opérateur (algorithme de Newton pour la linéarisation) au niveau des sous-structures (nous considérons des interfaces parfaites ce qui se traduit par des conditions linéaires de raccord entre les sous-structures). Notons de plus que le préconditionnement du problème linéaire condensé à l'interface fait appel de nouveau à la décomposition d'opérateur (décomposition de l'opérateur inverse). La résolution par décomposition de domaine sous-entend donc l'utilisation de nombreuses autres stratégies, c'est cependant de cet aspect là uniquement qu'est issu la parallélisation du code de calcul.

Notons que cette stratégie Newton-Krylov-Schur bénéficie d'un cadre théorique robuste. En effet elle combine les propriétés d'extensibilité des décompositions de Schur sur les problèmes linéaires (la dépendance du conditionnement envers les paramètres de discrétisation ne varie que selon le carré d'un logarithme) et la convergence quadratique du solveur non-linéaire de Newton-Raphson.

De plus, les décompositions de domaine sans recouvrement s'interprètent facilement d'un point de vue mécanique, elles s'intègrent même naturellement à un processus de conception de mécanisme où des pièces d'origines différentes sont assemblées entre elles (chaque pièce, dont les propriétés sont données par un service de conception spécifique, étant alors par exemple liée à un sous-domaine). Une analyse des méthodes de décomposition de domaine permet donc d'en faire un outil mathématique respectueux de la modélisation des problèmes physiques, ce qui conduit à obtenir des formulations (et préconditionneurs) mécaniquement consistants, et dans le cadre des problèmes multi-échelles à donner la possibilité d'adapter la stratégie de calcul à chacun des niveaux.

Historiquement, nos travaux sont issus d'un problème d'origine industrielle : l'étude par éléments-finis d'une butée flexible SNECMA-moteurs-de-fusées (décrite dans l'annexe B). La simulation de cette structure à composants élastomériques quasi-incompressibles fortement non-linéaires conduit à l'utilisation d'un algorithme de Newton pour obtenir une séquence de systèmes non-linéaires à matrices non-invariantes. La grande taille de ces problèmes et la grande largeur de bande des matrices de rigidité conduisent à envisager l'utilisation de méthodes Schur-Krylov pour la résolution des systèmes linéaires. Plus précisément nous nous sommes intéressés dans un premier temps aux deux principales méthodes, l'approche primale et l'approche duale, conceptuellement très similaires et pourtant conduisant à des processus de résolution parfois très différents. Etant donné qu'une séquence de systèmes linéaires doit être résolue, un objectif est de définir une stratégie de résolution exploitant l'information numérique générée au cours des résolutions. De plus la prise en compte de l'incompressibilité est réalisé via l'introduction d'un multiplicateur de Lagrange qui s'interprète comme une pression hydrostatique. L'adaptation des méthodes de décomposition de domaine à des problèmes multichamps puis multiphysique s'est alors imposé naturellement comme un objectif majeur de nos travaux. Finalement après avoir travaillé sur les bases du code MODULEF (INRIA) programmé en FORTRAN 77, nous avons souhaité profiter des possibilités offertes par les langages récents aussi nous avons entamé une réflexion sur la mise en place des méthodes de décomposition de domaine au sein du code orienté objet ZeBuLon (ONERA, Mines de Paris, NW-Numerics).

De là nos travaux s'organisent autour de cinq axes chacun détaillé dans une partie.

La première partie présente une partie de l'état de l'art des méthodes de décomposition de domaine Schur-Krylov. Dans le premier chapitre nous présentons le principe du partitionnement d'un domaine en sous-structures et de la description de l'interface entre les sous-

structures, puis comment un problème d'élasticité linéaire peut être condensé sur l'interface notamment dans le cadre des deux méthodes les plus classiques : l'approche primale et l'approche duale de décomposition de domaine sans recouvrement. La résolution du problème condensé est conduite par un solveur itératif de Krylov dont les principales propriétés sont rappelées dans le deuxième chapitre. La combinaison des décompositions de Schur avec les solveurs de Krylov entraîne la mise en place de préconditionneurs et problèmes grossiers classiques qui sont décrits dans le troisième chapitre.

La seconde partie présente des améliorations pouvant être apportées aux méthodes classiques de décomposition de domaine. La première (décrite dans le chapitre 4) est issue de la comparaison des approches primale et duale de décomposition de domaine dans le cadre de problèmes fortement hétérogènes (et possédant éventuellement des géométries complexes), elle permet de combler une lacune de l'approche duale en définissant une nouvelle initialisation qui apporte une plus grande information au processus de résolution. La seconde (chapitre 5) s'applique à une classe de problèmes multichamps où la seconde inconnue (typiquement la pression au sein d'un élastomère) n'est pas continue entre les sous-domaines. Suivant le traitement appliqué à cette dernière, elle peut perturber les préconditionneurs classiques en surévaluant la rigidité des interfaces ; nous proposons deux nouvelles stratégies pour restaurer les bonnes propriétés des préconditionneurs des approches classiques pour ces problèmes.

L'étude menée sur les approches classiques nous a conduit à ressentir leur forte connexion, et naturellement à envisager leur unification. Sachant qu'aucun résultat théorique ne permet d'affirmer leur équivalence, nous proposons dans la troisième partie une unification algorithmique des approches primale et duale de décomposition de domaine, la formulation résultante est appelée "approche hybride" de décomposition de domaine. Au sein de cette dernière, les approches classiques (avec l'ensemble de leurs préconditionneurs et problèmes grossiers) ne sont que des cas particuliers. L'approche hybride est présentée dans le chapitre 6, le chapitre 7 revient sur le détail pratique de la mise en oeuvre des problèmes grossiers. Une première validation est proposée dans le chapitre 8 tout d'abord sur un problème simple d'élasticité linéaire ensuite sur un problème multiphysique de l'étude des milieux poreux où nous montrons que l'approche hybride permet d'avoir une approche plus respectueuse de la physique des phénomènes et conduit à de meilleures performances que les méthodes classiques.

La quatrième partie traite des accélérations Krylov dans le cadre de la résolution d'une séquence de systèmes linéaires à matrices non-invariantes. Ces systèmes sont issus de la linéarisation par algorithmes de Newton-Raphson de problèmes fortement non-linéaires. Le chapitre 9 présente rapidement la méthode Newton-Raphson puis les algorithmes classiques d'accélération Krylov adaptés au cas de matrices invariantes. Le chapitre 10 rappelle la méthode GIRKS (*Generalized Iterative Reuse of Krylov Subspaces*) pour les séquences de systèmes linéaires à matrices non-invariantes puis le valide dans le cas des approches de décomposition de domaine en présence de sous-structures flottantes sur un problème industriel complexe, le flambage d'une butée flexible dont on donne une description de la géométrie et des principales propriétés mécaniques en annexe B. Un nouvel algorithme d'accélération, SRKS (*Selective Reuse of Krylov Subspaces*), également adaptée aux matrices non-invariantes, est présenté dans le chapitre 11. Là où GIRKS réutilise l'ensemble de l'information précédemment générée avec un solveur approché, SRKS utilise un solveur exact sur une sélection adroite de l'information. SRKS est validé sur le calcul du flambage d'une poutre composite.

La cinquième partie est quant-à-elle relative à la mise en oeuvre au sein d'un code orienté objet de l'ensemble des méthodes et solveurs présentées : il s'agit d'exploiter les possibilités offertes par les nouveaux paradigmes de programmation pour développer un code riche,

élégant et ouvert à de nouvelles extensions. Le chapitre 12 présente les données à générer durant la décomposition d'un domaine pour aboutir dans le chapitre 13 à la mise en place de formulations et de solveurs efficaces. Le principe de base est d'affecter à chaque degré de liberté un traitement individuel, nous montrons comment ce principe permet de développer l'approche hybride puis d'étendre le code à la majorité des approches de décomposition de domaine sans recouvrement.

---

# Notations

## Tenseurs et algèbre linéaire

- $\bar{a}$  : tenseur du premier ordre
- $\bar{\bar{a}}$  : tenseur du deuxième ordre
- $\mathbf{a}$  : tenseur du quatrième ordre
- $\bar{a}\bar{b}$  : produit tensoriel contracté
- $\bar{\bar{a}}\bar{\bar{b}}$  : produit tensoriel doublement contracté
- $\bar{\bar{a}}^T$  : transposé d'un tenseur du second ordre, d'une matrice ou d'un vecteur

## Grandeurs mécaniques

- $\bar{u}$  : champ de déplacement
- $\bar{\bar{\epsilon}}$  : tenseur des déformations linéarisées
- $\bar{\bar{\sigma}}$  : tenseur des contraintes de Cauchy
- $\mathbf{a}$  : tenseur d'élasticité (ou de rigidité drainée pour les milieu poreux)
- $m$  : apport de masse fluide
- $\bar{M}$  : flux de masse fluide
- $p$  : champ de pression
- $\lambda, \mu$  coefficients de Lamé

## Décompositions de domaine

- $^{(s)}$  : grandeur relative au sous-domaine  $s$
- $\Omega$  : domaine de  $\mathbb{R}^n$  ( $n$  est la dimension de l'espace)
- $\Upsilon$  : interface entre les sous-domaines
- $\underline{x}$  : la grandeur  $x$  est associée à une description par connectivité de l'interface
- $\lambda$  : intereffort (effort dû aux sous-structures voisines)
- $x_i$  : degré de liberté interne
- $x_b$  : degré de liberté de l'interface
- $x_p$  : degré de liberté primal sur l'interface
- $x_d$  : degré de liberté dual sur l'interface

## Solveurs de Krylov

- $x$  : champ inconnu
- $r$  : résidu
- $z$  : résidu préconditionné
- $w$  : direction de descente
- $v$  : vecteur de la base d'Arnoldi
- $y$  : vecteur de Ritz
- $\theta$  : valeur de Ritz
- Notation majuscule :  $M_i = (m_0, \dots, m_{i-1})$



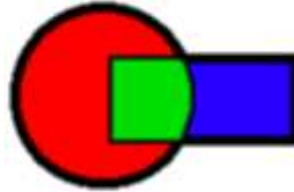


Première partie

Méthodes de décomposition de  
domaine sans recouvrement en  
élasticité linéaire



On attribue généralement la paternité des méthodes de décomposition de domaine à Hermann Schwarz qui au XIX<sup>ème</sup> siècle a proposé un premier algorithme de résolution d'une équation aux dérivées partielles en séparant le domaine en deux sous-domaines (avec recouvrement) de géométrie simple, laissant à la postérité ce schéma devenu célèbre :



*Figure 1* — Décomposition du problème de Schwarz

Le développement des méthodes de calcul est profondément lié à l'évolution de leur support, aussi les méthodes de décomposition de domaine ont connu un véritable essor récemment, accompagnant l'avènement des architectures parallèles pour les calculateurs scientifiques et industriels. Notamment, durant la dernière décennie des progrès importants ont été réalisés tant pour la création de nouvelles méthodes, pour l'obtention de résultats théoriques puissants que pour la mise en oeuvre efficace au sein de codes de calcul. On peut également remarquer la montée en puissance des méthodes sans recouvrement, et des méthodes algébriques (qui s'interprètent directement à partir d'un système linéaire).

En calcul de structure, les méthodes de décomposition de domaine sans recouvrement forment un outil privilégié. Outre les aspects purement liés aux performances de calcul sur une structure complète, elles s'intègrent naturellement au calcul de structure assemblée, avec par exemple un sous-domaine par pièce constituant le mécanisme (les caractéristiques de chaque pièce pouvant être fournies par des services de conception différents).

Cette partie présente les bases des méthodes de décomposition de domaine dans le cas de l'élasticité linéaire. Dans le premier chapitre nous montrons comment reformuler un problème par décomposition de domaine des points de vue continu et discret. Dans le second chapitre nous présentons les méthodes classiques pour la résolution de ce problème (solveurs de Krylov). Le troisième chapitre étudie alors les diverses optimisations issues de la conjonction des formulations et solveurs étudiés.



Ce chapitre introduit le principe des méthodes de décomposition de domaine sans recouvrement sur le cas simple de l'élasticité linéaire. Il s'agit, après avoir réalisé une partition du domaine, de réécrire un problème mécanique sur l'interface entre les sous-structures.

Le travail réalisé dans cette thèse se situe au niveau discret (après application de la méthode des éléments-finis) cependant l'obtention de résultats théoriques nécessite de connaître les propriétés des opérateurs au niveau continu. De manière à proposer une présentation assez complète des méthodes de décomposition de domaine, nous avons donc décidé d'évoquer rapidement les formulations d'un point de vue continu puis de présenter plus précisément le point de vue discret.

Parmi les approches de décomposition de domaine, deux servent de base à nos études, les approches dites primale et duale de décomposition de domaine, qui sont sans doute les plus communément utilisées. Evoquées au niveau continu, elles font l'objet d'une étude plus approfondie au niveau discret.

## 1.1 Obtention des problèmes condensés

### 1.1.1 Problème modèle

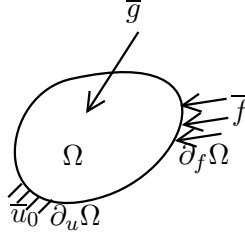


Figure 1.1 — Problème modèle

Considérons un domaine  $\Omega$  de  $\mathbb{R}^n$  ( $n=2$  ou  $3$ ) sur lequel est posé un problème classique d'élasticité linéaire (voir figure 1.1) : déplacement imposé  $\bar{u}_0$  sur une partie  $\partial_u\Omega$  de la frontière, effort  $\bar{f}$  imposé sur la partie complémentaire  $\partial_f\Omega$ , effort volumique  $\bar{g}$  imposé sur  $\Omega$ , tenseur d'élasticité  $\mathbf{a}$  [42, 91]. Le système est régi par les équations suivantes :

$$\begin{cases} \overline{\operatorname{div}}(\bar{\sigma}) + \bar{g} = \bar{0} & \text{dans } \Omega \\ \bar{\sigma} = \mathbf{a} : \bar{\varepsilon}(\bar{u}) & \text{dans } \Omega \\ \bar{\varepsilon}(\bar{u}) = \frac{1}{2} (\overline{\operatorname{grad}}(\bar{u}) + \overline{\operatorname{grad}}(\bar{u})^T) & \text{dans } \Omega \\ \bar{\sigma} \cdot \bar{n} = \bar{f} & \text{sur } \partial_f\Omega \\ \bar{u} = \bar{u}_0 & \text{sur } \partial_u\Omega \end{cases} \quad (1.1)$$

Pour que le problème soit bien posé, nous supposons  $\operatorname{mes}(\partial_u\Omega) > 0$ . Nous nous plaçons également dans le cadre où le tenseur  $\mathbf{a}$  définit une forme bilinéaire symétrique définie coercive sur l'espace des tenseurs symétriques d'ordre 2. Sous ces conditions le problème (1.1) possède une unique solution [21].

Afin d'introduire les décompositions de domaine sans recouvrement, nous abordons d'abord le principe sur une décomposition en deux sous-domaines, les sous-structurations plus complexes, bien que basées sur exactement les mêmes principes, nécessitent une plus grande attention, notamment sur la description des interfaces, et sont traitées dans un deuxième temps.

### 1.1.2 Décomposition en deux sous-domaines

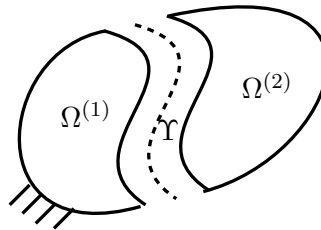


Figure 1.2 — Décomposition en deux sous-domaines

Considérons donc une partition de  $\Omega$  en 2 sous-structures  $\Omega^{(1)}$  et  $\Omega^{(2)}$ . On définit alors l'interface  $\Upsilon$  entre les sous-structures (figure 1.2) :

$$\Upsilon = \partial\Omega^{(1)} \cap \partial\Omega^{(2)} \quad (1.2)$$

Le système (1.1) est posé sur  $\Omega$ , on écrit ses restrictions sur  $\Omega^{(1)}$  et  $\Omega^{(2)}$  :

$$s = 1 \text{ ou } 2, \begin{cases} \overline{\text{div}}(\overline{\sigma}^{(s)}) + \overline{g}^{(s)} = \overline{0} & \text{dans } \Omega^{(s)} \\ \overline{\sigma}^{(s)} = \mathbf{a}^{(s)} : \overline{\varepsilon}(\overline{u}^{(s)}) & \text{dans } \Omega^{(s)} \\ \overline{\varepsilon}(\overline{u}^{(s)}) = \frac{1}{2} \left( \overline{\text{grad}}(\overline{u}^{(s)}) + \overline{\text{grad}}(\overline{u}^{(s)})^T \right) & \text{dans } \Omega^{(s)} \\ \overline{\sigma}^{(s)} \cdot \overline{n}^{(s)} = \overline{f}^{(s)} & \text{sur } \partial_f \Omega \cap \partial \Omega^{(s)} \\ \overline{u}^{(s)} = \overline{u}_0^{(s)} & \text{sur } \partial_u \Omega \cap \partial \Omega^{(s)} \end{cases} \quad (1.3)$$

raccordées par des conditions à l'interface, continuité du déplacement

$$\overline{u}^{(1)} = \overline{u}^{(2)} \quad \text{sur } \Upsilon \quad (1.4)$$

et équilibre des efforts (principe de l'action-réaction)

$$\overline{\sigma}^{(1)} \overline{n}^{(1)} + \overline{\sigma}^{(2)} \overline{n}^{(2)} = \overline{0} \quad \text{sur } \Upsilon \quad (1.5)$$

Naturellement, le système (1.3,1.4,1.5) est strictement équivalent au problème global (1.1).

### 1.1.3 Décomposition en $N$ sous-domaines

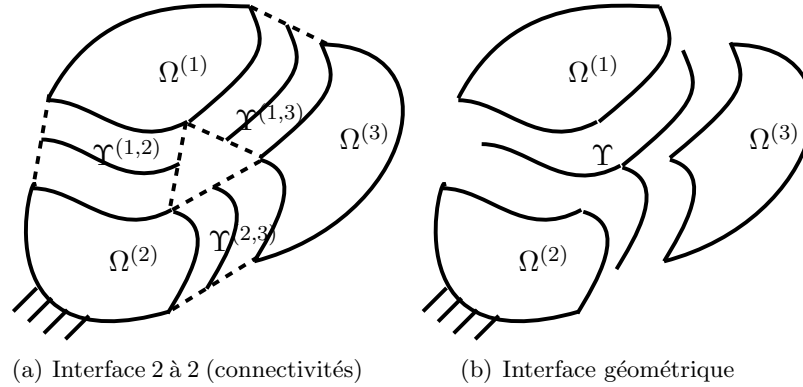


Figure 1.3 — Définition de l'interface pour une décomposition en  $N$  sous-domaines

Considérons une partition de  $\Omega$  en  $N$  sous-structures  $\Omega^{(s)}$ . On définit alors les interfaces entre les sous-structures :

$$\begin{cases} \Upsilon^{(i,j)} = \Upsilon^{(j,i)} = \partial \Omega^{(i)} \cap \partial \Omega^{(j)} \\ \Upsilon^{(s)} = \bigcup_j \Upsilon^{(s,j)} \\ \Upsilon = \bigcup_s \Upsilon^{(s)} \end{cases} \quad (1.6)$$

L'utilisation de plus de 2 sous-domaines conduit (sauf dans le cas de décompositions dites "bandes") à l'apparition de points multiples ou *cross-points* (noeuds appartenant à plus de 2 sous-structures). Ces points multiples entraînent la coexistence de 2 descriptions de l'interface (voir figure 1.3) : l'interface dite géométrique  $\Upsilon$  et l'interface des connectivités formée de l'ensemble des interfaces entre paires de sous-domaines  $(\Upsilon^{(i,j)})_{1 \leq i < j \leq N}$ .

Par ailleurs on remarque que les sous-domaines ne sont plus en contact avec l'intégralité de l'interface (par exemple  $\Omega^{(1)}$  ne rencontre pas  $\Upsilon^{(2,3)}$ ) ce qui implique de savoir positionner l'interface d'un sous-domaine par rapport aux interfaces assemblées.



Afin de mieux gérer les points de l'interface et notamment les points multiples, il est utile d'associer à chaque point de  $\Upsilon$  une fonction  $\iota$  donnant la liste des sous-structures dont l'interface contient ce point.

$$\begin{aligned} \iota : \Upsilon &\rightarrow \mathcal{P}(\mathbb{N}) \\ x &\mapsto \iota(x) = \{s \in \llbracket 1, N \rrbracket / x \in \Upsilon^{(s)}\} \end{aligned} \quad (1.7)$$

Considérons l'interface uniquement d'un point de vue géométrique à partir d'intersections et d'unions de frontières de sous-domaines. C'est sur cet ensemble que l'on doit rechercher le déplacement de l'interface, cette description conduit à l'espace fonctionnel de trace suivant (pour la définition des espaces de trace le lecteur peut se référer à [15]) :

$$\mathcal{Y}_\Upsilon = u_0 + H_{00}^{1/2}(\Upsilon) = u_0 + H_{00}^{1/2} \left( \bigcup_{1 \leq i < j \leq N} \Upsilon^{(i,j)} \right) \quad (1.8)$$

Considérons maintenant l'interface comme le lieu de connection entre paires de sous-domaines. C'est dans cet ensemble qu'évolue l'effort de réaction entre deux sous-structures, cette définition conduit à l'espace fonctionnel suivant :

$$\underline{\mathcal{Y}}_\Upsilon = \bigotimes_{1 \leq i < j \leq N} H^{1/2}(\Upsilon^{(i,j)}) \quad (1.9)$$

Notons que cette description fait apparaître des redondances au niveau des points multiples : en effet soit  $x \in \Upsilon$  si  $\iota(x) = \{1, 2, 3\}$ ,  $x$  appartient à  $\Upsilon^{(1,2)}$ ,  $\Upsilon^{(2,3)}$  et  $\Upsilon^{(1,3)}$ , alors que seulement deux connectivités suffisent à déclarer l'appartenance de  $x$  aux sous-domaines (1) (2) et (3). Dans le cas général d'un point  $m$ -multiple, cette description fait apparaître  $\binom{m}{2}$  relations quand seulement  $(m-1)$  sont nécessaires et suffisantes. Nous verrons lors de la partie algébrique qu'il est possible de définir une description non redondante des connectivités.

Maintenant que la difficulté liée à l'apparition des points multiples est explicitée (nous passons cependant sous silence les problèmes liés aux interfaces ponctuelles, donc de mesure nulle, ce problème disparaît après discrétisation puisqu'on ne s'intéresse alors qu'à des relations point-à-point), il convient de s'intéresser aux passages entre les différents espaces. La première opération est une opération de trace, elle permet de passer du sous-domaine à son interface [15, 59] :

$$\begin{aligned} \text{Tr}^{(s)} : H^1(\Omega^{(s)}) &\rightarrow H^{1/2}(\Upsilon^{(s)}) \\ \bar{v}^{(s)} &\mapsto \bar{v}|_{\Upsilon^{(s)}} \end{aligned} \quad (1.10)$$

Dans la nomenclature des décompositions de domaine, on distingue les aspects "globaux" (définis sur l'ensemble de la structure  $\Omega$  ou de l'interface  $\Upsilon$ ) des aspects "locaux" (définis séparément sur chacun des sous-domaines  $\Omega^{(s)}$  ou des interfaces  $\Upsilon^{(s)}$ ). La trace est une opération locale. Les autres opérations à définir permettent de passer des grandeurs locales aux grandeurs globales (et réciproquement), ceux sont des opérations d'assemblage (souvent appelées projection) / localisation (restriction).

L'opération la plus simple est l'injection canonique de  $\Upsilon^{(s)}$  dans  $\Upsilon$  :

$$\begin{aligned} A^{(s)} : \mathcal{Y}_{\Upsilon^{(s)}} &\rightarrow \mathcal{Y}_\Upsilon \\ \bar{v}^{(s)} &\mapsto \begin{cases} A^{(s)}(\bar{v}^{(s)}) = \bar{v}^{(s)} \text{ sur } \Upsilon^{(s)} \\ A^{(s)}(\bar{v}^{(s)}) = 0 \text{ ailleurs} \end{cases} \end{aligned} \quad (1.11)$$

naturellement

$$\begin{aligned} A^{(s)-1} : \mathcal{Y}_\Upsilon &\rightarrow \mathcal{Y}_{\Upsilon^{(s)}} \\ \bar{v} &\mapsto A^{(s)-1}(\bar{v}) = \bar{v}|_{\Upsilon^{(s)}} \end{aligned} \quad (1.12)$$

Si on utilise une description de l'interface par connectivité, on fait appel à une projection plus sophistiquée :

$$\begin{aligned} \underline{A}^{(s)} : \mathcal{Y}_{\Upsilon^{(s)}} &\rightarrow \underline{\mathcal{Y}}_\Upsilon \\ \bar{v}^{(s)} &\mapsto \underline{A}^{(s)}(\bar{v}^{(s)}) = \left(0, \dots, \epsilon_j^s \bar{v}|_{\Upsilon^{(s,j)}}, \dots, 0\right) \end{aligned} \quad (1.13)$$

pour tout voisin ( $j$ ) de ( $s$ )

où  $\epsilon_j^s$  vaut 1 si  $s > j$  et  $(-1)$  si  $s < j$  (l'objectif étant que  $\epsilon_j^s + \epsilon_s^j = 0$ ). L'opération réciproque est définie de la manière suivante :

$$\begin{aligned} \underline{A}^{(s)-1} : \underline{\mathcal{Y}}_\Upsilon &\rightarrow \mathcal{Y}_{\Upsilon^{(s)}} \\ \underline{\bar{v}} &\mapsto \underline{A}^{(s)-1}(\underline{\bar{v}}) : x \mapsto \sum_{j \in \iota(x)} \epsilon_j^s \bar{v}^{(s,j)}(x) \end{aligned} \quad (1.14)$$

Revenons maintenant au problème global (1.1). Avant d'écrire le problème décomposé, on choisit d'introduire  $\bar{\lambda}^{(s)}$  qui représente les efforts de réaction appliqués sur le sous-domaine ( $s$ ) par ses voisins. Le système décomposé devient alors :

$$\forall s \leq N \begin{cases} \overline{\text{div}}(\bar{\sigma}^{(s)}) + \bar{g}^{(s)} = \bar{0} & \text{dans } \Omega^{(s)} \\ \bar{\sigma}^{(s)} = \mathbf{a}^{(s)} : \bar{\varepsilon}^{(s)} & \text{dans } \Omega^{(s)} \\ \bar{\varepsilon}^{(s)} = \frac{1}{2} \left( \overline{\text{grad}}(\bar{u}^{(s)}) + \overline{\text{grad}}(\bar{u}^{(s)})^T \right) & \text{dans } \Omega^{(s)} \\ \bar{\sigma}^{(s)} \cdot \bar{n}^{(s)} = \bar{f}^{(s)} & \text{sur } \partial_f \Omega \cap \partial \Omega^{(s)} \\ \bar{\sigma}^{(s)} \cdot \bar{n}^{(s)} = \bar{\lambda}^{(s)} & \text{sur } \Upsilon^{(s)} \\ \bar{u}^{(s)} = \bar{u}_0^{(s)} & \text{sur } \partial_u \Omega \cap \partial \Omega^{(s)} \end{cases} \quad (1.15a)$$

$$\forall (i, j) \leq N, \bar{u}^{(i)} = \bar{u}^{(j)} \text{ sur } \Upsilon^{(i,j)} \Leftrightarrow \sum A^{(s)} \text{Tr}^{(s)} \bar{u}^{(s)} = 0 \quad (1.15b)$$

$$\sum_s A^{(s)} \bar{\lambda}^{(s)} = \bar{0} \text{ sur } \Upsilon \quad (1.15c)$$

Le système peut être résolu de plusieurs manières, notamment :

- en considérant (1.15) comme un problème d'équilibre sous contrainte en  $u^{(s)}$  et  $\lambda^{(s)}$ , ce qui conduit à la formulation à trois champs [9, 69, 83] ;
- en choisissant de travailler sur le champ de déplacement de l'interface  $\bar{u} \in \mathcal{Y}_\Upsilon$  et en vérifiant *a priori* la continuité à l'interface (1.15b)  $\forall s \leq N, \text{Tr}^{(s)}(u^{(s)}) = A^{(s)-1} \bar{u}$ . On obtient alors le système (1.15a, 1.15c) posé en  $\bar{u}$ . Cette approche est la base de la méthode primale de décomposition de domaine [100, 101, 60, 97, 61, 102, 99] qui est détaillée dans la prochaine section ;
- en choisissant de travailler sur le champ d'intereffort à l'interface  $\bar{\lambda} \in \underline{\mathcal{Y}}_\Upsilon^*$  et en vérifiant *a priori* l'équilibre à l'interface (1.15c)  $\forall s \leq N, \lambda^{(s)} = \underline{A}^{(s)-1} \bar{\lambda}$ . On obtient alors le système (1.15a, 1.15b) posé en  $\bar{\lambda}$ . Cette approche est la base de la méthode duale de décomposition de domaine [37, 25, 38, 39, 35, 62, 6] qui est également détaillée dans la prochaine section ;
- en choisissant de séparer l'interface en deux sous-ensembles complémentaires  $\Upsilon_p$  et  $\Upsilon_d$  ( $\Upsilon_p \cup \Upsilon_d = \Upsilon, \Upsilon_p \cap \Upsilon_d = \emptyset$ ), et en introduisant  $(\bar{u}_p, \bar{\lambda}_d) \in \mathcal{Y}_{\Upsilon_p} \times \underline{\mathcal{Y}}_{\Upsilon_d}$  inconnues vérifiant *a priori* la continuité du déplacement de part et d'autre de  $\Upsilon_p$  et l'équilibre sur

$\Upsilon_d$ . On recherche alors  $(\bar{\mathbf{u}}_p, \bar{\boldsymbol{\lambda}}_d)$  de manière à garantir la continuité du déplacement de part et d'autre de  $\Upsilon_d$  et l'équilibre des efforts sur  $\Upsilon_p$ . Suivant la stratégie de résolution, cette méthode conduit à l'approche FETI-DP [52, 31, 63, 30] (voir annexe A) ou à l'approche hybride de décomposition de domaine [45, 47, 48] détaillée dans la partie III;

- en remplaçant les conditions d'interface (1.15b, 1.15c) par une combinaison linéaire équivalente, définissant ainsi des conditions mixtes (*i.e.* conditions de Robin) entre les sous-structures. Cette méthode permet de définir des conditions complexes à l'interface (par exemple raideur d'interface, contact, frottement) et conduit à des approches de type lagrangien augmenté [43], LaTIn [54] ou FETI-2-champs [93, 92].

## 1.2 Approches classiques

Dans cette section nous abordons du point de vue variationnel continu les principales (ou du moins les plus utiles pour la compréhension des travaux futurs) méthodes de décomposition de domaine.

### 1.2.1 Approche primale

Comme décrite dans la section précédente, l'approche primale consiste à rechercher le champ de déplacement  $\bar{\mathbf{u}}$  de l'interface. Considérons la formulation variationnelle associée à l'équilibre local des sous-structures [22] :

$$\left\{ \begin{array}{l} \text{Trouver } \bar{\mathbf{u}}^{(s)} \in V^{(s)} / \forall \bar{\mathbf{v}}^{(s)} \in V^{(s)}, a^{(s)}(\bar{\mathbf{u}}^{(s)}, \bar{\mathbf{v}}^{(s)} - \bar{\mathbf{u}}^{(s)}) = l^{(s)}(\bar{\mathbf{v}}^{(s)} - \bar{\mathbf{u}}^{(s)}) \\ V^{(s)} = \left\{ \bar{\mathbf{v}}^{(s)} \in H^1(\Omega^{(s)})^n, \bar{\mathbf{v}}^{(s)} = \bar{\mathbf{u}}_0 \text{ sur } \partial_u \Omega \cap \partial \Omega^{(s)}, \text{Tr}^{(s)} \bar{\mathbf{v}}^{(s)} = A^{(s)-1} \bar{\mathbf{u}} \right\} \end{array} \right. \quad (1.16)$$

Avec

$$\begin{aligned} a^{(s)}(\bar{\mathbf{u}}^{(s)}, \bar{\mathbf{v}}^{(s)}) &= \int_{\Omega^{(s)}} \mathbf{a}^{(s)} : \bar{\boldsymbol{\varepsilon}}(\bar{\mathbf{u}}^{(s)}) : \bar{\boldsymbol{\varepsilon}}(\bar{\mathbf{v}}^{(s)}) \\ l^{(s)}(\bar{\mathbf{v}}^{(s)}) &= \int_{\Omega^{(s)}} \bar{\mathbf{g}}^{(s)} \cdot \bar{\mathbf{v}}^{(s)} + \int_{\partial_f \Omega \cap \partial \Omega^{(s)}} \bar{\mathbf{f}}^{(s)} \cdot \bar{\mathbf{v}}^{(s)} \end{aligned}$$

Soit  $\bar{\mathbf{z}}^{(s)}(\bar{\mathbf{u}}, \bar{\mathbf{f}}^{(s)}, \bar{\mathbf{g}}^{(s)}) \in V^{(s)}$  la solution de (1.16), on introduit alors les opérateurs locaux de Steklov-Poincaré  $S^{(s)}$  et le second membre local  $\bar{L}^{(s)1}$  :

$$\langle S^{(s)} A^{(s)-1} \bar{\mathbf{u}}, \bar{\boldsymbol{\mu}} - \bar{\mathbf{u}}_0 \rangle = a^{(s)} \left( \bar{\mathbf{z}}^{(s)}(\bar{\mathbf{u}}, \bar{\mathbf{0}}, \bar{\mathbf{0}}), \text{Tr}^{(s)-1}(\bar{\boldsymbol{\mu}} - \bar{\mathbf{u}}_0) \right), \forall \bar{\boldsymbol{\mu}} \in \mathcal{Y}_{\Upsilon^{(s)}} \quad (1.17)$$

$$\langle \bar{L}^{(s)}, \bar{\boldsymbol{\mu}} - \bar{\mathbf{u}}_0 \rangle = l^{(s)}(\text{Tr}^{(s)-1}(\bar{\boldsymbol{\mu}} - \bar{\mathbf{u}}_0)) - a^{(s)} \left( \bar{\mathbf{z}}^{(s)}(\bar{\mathbf{0}}, \bar{\mathbf{f}}^{(s)}, \bar{\mathbf{g}}^{(s)}), \text{Tr}^{(s)-1}(\bar{\boldsymbol{\mu}} - \bar{\mathbf{u}}_0) \right), \forall \bar{\boldsymbol{\mu}} \in \mathcal{Y}_{\Upsilon^{(s)}} \quad (1.18)$$

L'opérateur local de Steklov-Poincaré est donc l'opérateur qui associe au déplacement imposé  $\bar{\mathbf{u}}$  sur  $\Upsilon^{(s)}$ , l'effort résultant (*ie* la réaction) sur  $\Upsilon^{(s)}$ . L'équilibre à l'interface (1.15c) donne alors le système suivant en  $\bar{\mathbf{u}}$  :

$$\left( \sum_s A^{(s)} S^{(s)} A^{(s)-1} \right) \bar{\mathbf{u}} = \left( \sum_s A^{(s)} \bar{L}^{(s)} \right) \quad (1.19)$$

---

<sup>1</sup>on note  $\langle \bar{\mathbf{a}}, \bar{\mathbf{b}} \rangle = \int_{\Upsilon^{(s)}} \bar{\mathbf{a}} \cdot \bar{\mathbf{b}}$

### 1.2.2 Approche duale

Cette approche consiste à rechercher le champ d'intereffort sur les interfaces entre les sous-structures, elle est traditionnellement introduite à partir de considérations énergétiques. Le champ de déplacement solution de (1.1) minimise l'énergie potentielle du système ce qui implique la minimisation locale de l'énergie potentielle  $I^{(s)}$  des sous-structures :

$$I^{(s)}(\bar{v}^{(s)}) = \frac{1}{2} \int_{\Omega^{(s)}} \mathbf{a}^{(s)} : \bar{\varepsilon}(\bar{v}^{(s)}) : \bar{\varepsilon}(\bar{v}^{(s)}) - \int_{\Omega^{(s)}} \bar{g}^{(s)} \cdot \bar{v}^{(s)} - \int_{\partial_f \Omega \cap \partial \Omega^{(s)}} \bar{f}^{(s)} \cdot \bar{v}^{(s)} \text{ sur } V^{(s)} \quad (1.20)$$

sous la contrainte de continuité des déplacements à l'interface :

$$\sum_s \underline{A}^{(s)} \text{Tr}^{(s)} \bar{u}^{(s)} = 0 \text{ dans } \underline{\mathcal{Y}}_{\Gamma} \quad (1.21)$$

Cette contrainte est dualisée par l'introduction du multiplicateur de Lagrange  $\bar{\mu} \in \underline{\mathcal{Y}}_{\Gamma}^*$  et la recherche de la solution du problème se ramène à la recherche du point selle du lagrangien  $\mathcal{L}(\bar{v}^{(1)}, \dots, \bar{v}^{(N)}, \bar{\mu})$ <sup>2</sup> :

$$\mathcal{L}(\bar{v}^{(1)}, \dots, \bar{v}^{(N)}, \bar{\mu}) = \sum_s I^{(s)}(\bar{v}^{(s)}) - \int \bar{\mu} \cdot \sum_s \underline{A}^{(s)} \text{Tr}^{(s)} \bar{u}^{(s)} \quad (1.22)$$

L'extremum  $(\bar{u}^{(1)}, \dots, \bar{u}^{(N)}, \bar{\lambda}) \in V^{(1)} \times \dots \times V^{(N)} \times \underline{\mathcal{Y}}_{\Gamma}^*$  de ce lagrangien vérifie le système suivant :

$$\forall (\bar{v}^{(s)}, \bar{\mu}) \in V^{(s)} \times \underline{\mathcal{Y}}_{\Gamma}^*, \begin{cases} a^{(s)}(\bar{u}^{(s)}, \bar{v}^{(s)}) = l^{(s)}(\bar{v}^{(s)}) + \int \bar{\lambda} \cdot \underline{A}^{(s)} \text{Tr}^{(s)} \bar{v}^{(s)} \\ \int \bar{\mu} \cdot \sum_s \underline{A}^{(s)} \text{Tr}^{(s)} \bar{u}^{(s)} = 0 \end{cases} \quad (1.23)$$

## 1.3 Application à la méthode des éléments-finis

On choisit de résoudre le problème de structure par une approche éléments-finis classique : les espaces admissibles sont approchés par des sous-espaces dont une base est formée par des fonctions polynômiales à support compact [12, 17, 5, 106].

Du point de vue décomposition de domaine, on se restreint au cas des décompositions orientées éléments (chaque élément appartient à une et une seule sous-structure) conformes avec la discrétisation éléments-finis. Cela implique trois conditions [81] :

- les noeuds correspondent de part et d'autre de l'interface ;
- les espaces d'approximation sont les mêmes de part et d'autre de l'interface ;
- les modèles (barre, poutre, coque, *etc*) sont les mêmes de part et d'autre de l'interface.

Sous ces conditions les raccords entre les sous-structures s'écrivent simplement comme des égalités de noeuds. Dans le cas non-conforme une approche classique consiste à définir des éléments de frontière (*mortar*) sur lesquels la continuité des déplacements et l'équilibre des efforts sont vérifiés dans un sens faible [2, 3, 94].

L'obtention des formulations décomposées peut se réaliser très simplement par discrétisation des formulations variationnelles condensées. Cependant, les méthodes étudiées ici ont l'avantage d'être purement algébriques (applicables à n'importe quel système matriciel) aussi nous choisissons de reprendre rapidement l'obtention des problèmes d'interface à partir du système linéaire global.

<sup>2</sup>on note  $\int \bar{x} \cdot \bar{y} = \sum_{i < j} \int_{\Gamma^{(i,j)}} \bar{x}^{(i,j)} \bar{y}^{(i,j)}$

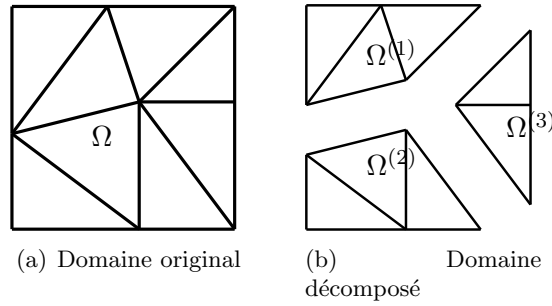


Figure 1.4 — Décomposition conforme en trois sous-structures

Le système global (1.1) s'écrit après discrétisation :

$$Ku = f \quad (1.24)$$

où  $K$  est la matrice de rigidité globale du problème,  $u$  le vecteur des déplacements inconnus,  $f$  le vecteur des efforts généralisés. Sous les hypothèses formulées précédemment,  $K$  est symétrique définie positive.

Ce système est équivalent au système décomposé suivant :

$$K^{(s)}u^{(s)} = f^{(s)} + \lambda^{(s)} \quad (1.25a)$$

$$\sum_s \underline{A}^{(s)} t^{(s)} u^{(s)} = 0 \quad (1.25b)$$

$$\sum_s A^{(s)} t^{(s)} \lambda^{(s)} = 0 \quad (1.25c)$$

où  $\lambda^{(s)}$  est la réaction imposée par ses voisines à la sous-structure ( $s$ ) ( $\lambda^{(s)}$  n'est donc pas nulle que sur les degrés de liberté d'interface),  $t^{(s)}$  est la discrétisation de l'opérateur  $\text{Tr}^{(s)}$ , on a conservé l'écriture  $A^{(s)}$  et  $\underline{A}^{(s)}$  pour leur version discrétisée (voir figure 1.5 pour un exemple complet). La ligne (1.25a) correspond à l'équilibre local des sous-structures, la ligne (1.25c) à l'équilibre des interefforts, la ligne (1.25b) à la continuité du champ de déplacement.

Notons quelques propriétés classiques dont celle d'orthogonalité entre les descriptions d'interface :

$$\sum_s \underline{A}^{(s)} A^{(s)T} = 0 \quad (1.26)$$

$$A^{(s)T} A^{(s)} = I_{\Upsilon^{(s)}} \quad (1.27)$$

$$\underline{A}^{(s)T} \underline{A}^{(s)} = \text{diag}(\text{multiplicité} - 1)_{\Upsilon^{(s)}} \quad (1.28)$$

$$A^{(s)} A^{(s)T} = \begin{cases} I & \text{sur } \Upsilon^{(s)} \\ 0 & \text{ailleurs} \end{cases} \quad (1.29)$$

Dans la suite on choisit de séparer les degrés de liberté des sous structures : indice  $i$  pour les degrés de liberté internes ( $\in \Omega^{(s)} \setminus \Upsilon^{(s)}$ ),  $b$  pour les degrés de liberté sur l'interface (*boundary*,  $\in \Upsilon^{(s)}$ ). On a notamment  $t^{(s)}v^{(s)} = v_b^{(s)}$ .

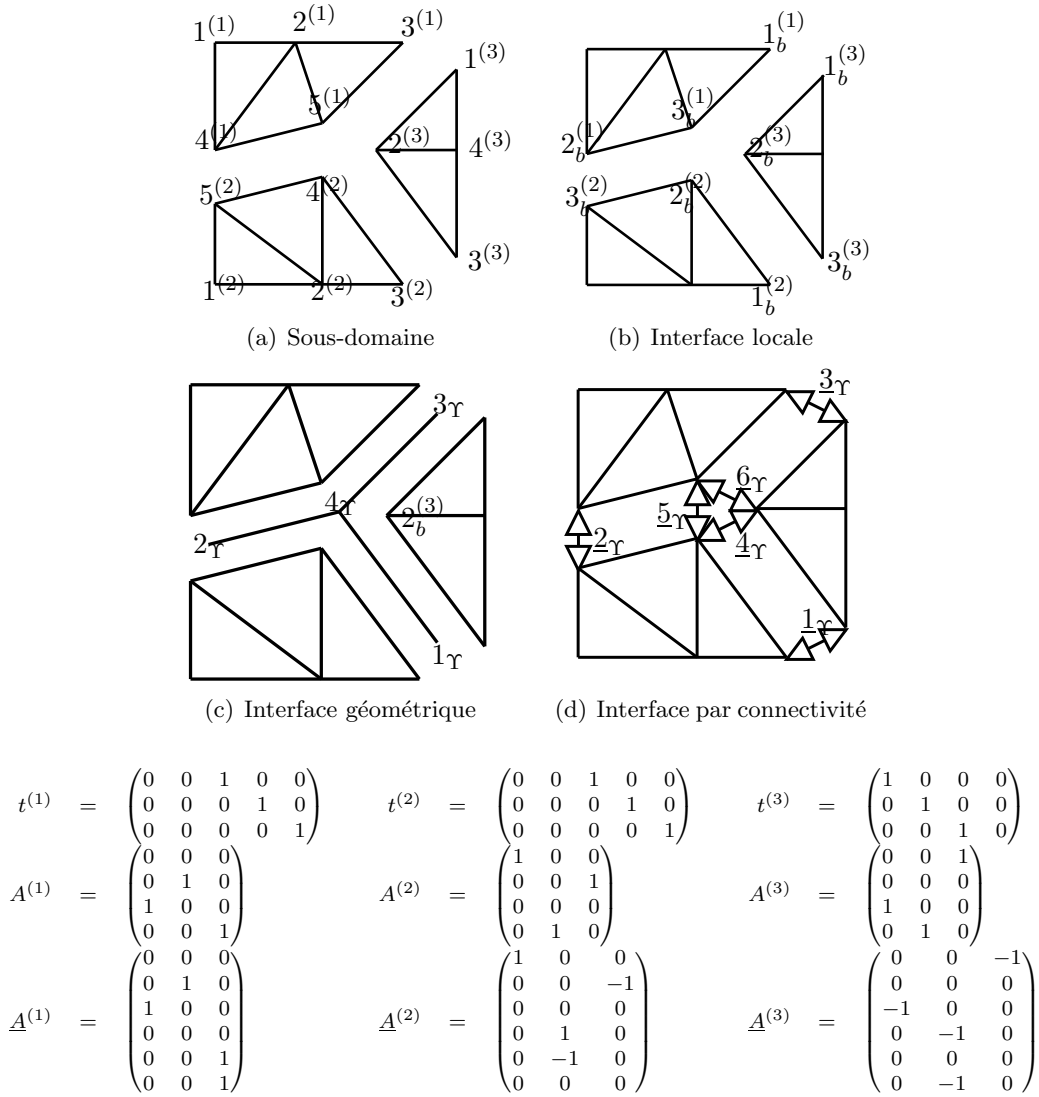


Figure 1.5 — Numérotations locales, numérotations de l'interface, opérateurs de trace et assemblage

### 1.3.1 Approche primale

On réécrit le système (1.24)

$$\begin{pmatrix} K_{ii}^{(1)} & 0 & \dots & 0 & K_{ib}^{(1)} A^{(1)T} \\ 0 & \ddots & & \vdots & \vdots \\ \vdots & & \ddots & 0 & \vdots \\ 0 & \dots & 0 & K_{ii}^{(N)} & K_{ib}^{(N)} A^{(N)T} \\ A^{(1)} K_{bi}^{(1)} & \dots & \dots & A^{(N)} K_{bi}^{(N)} & \sum_s A^{(s)} K_{bb}^{(s)} A^{(s)T} \end{pmatrix} \begin{pmatrix} u_i^{(1)} \\ \vdots \\ \vdots \\ u_i^{(N)} \\ \mathbf{u}_b \end{pmatrix} = \begin{pmatrix} f_i^{(1)} \\ \vdots \\ \vdots \\ f_i^{(N)} \\ \sum_s A^{(s)} f_b^{(s)} \end{pmatrix} \quad (1.30)$$

De manière à vérifier automatiquement la continuité des déplacements à l'interface, on a naturellement introduit une unique inconnue de déplacement à l'interface  $\mathbf{u}_b$ , on a  $u_b^{(s)} = A^{(s)T} \mathbf{u}_b$ . Les réactions se sont naturellement éliminées lors du calcul de la dernière ligne.

Pour condenser le problème à l'interface, on élimine les inconnues internes  $u_i^{(s)}$  :

$$\begin{aligned} K_{ii}^{(s)} u_i^{(s)} + K_{ib}^{(s)} A^{(s)T} \mathbf{u}_b &= f_i^{(s)} \\ \Rightarrow u_i^{(s)} &= K_{ii}^{(s)-1} \left( f_i^{(s)} - K_{ib}^{(s)} A^{(s)T} \mathbf{u}_b \right) \end{aligned} \quad (1.31)$$

L'inversion de  $K_{ii}^{(s)}$  correspond à la résolution d'un problème où l'interface est soumise à des conditions de Dirichlet (déplacement imposé  $A^{(s)T} \mathbf{u}_b$ ), il s'agit donc d'un problème bien posé et l'écriture précédente est licite.

Si on utilise l'expression (1.31) dans la dernière ligne de (1.30), on obtient

$$\sum_s A^{(s)} \left( K_{bb}^{(s)} - K_{bi}^{(s)} K_{ii}^{(s)-1} K_{ib}^{(s)} \right) A^{(s)T} \mathbf{u}_b = \sum_s A^{(s)} \left( f_b^{(s)} - K_{bi}^{(s)} K_{ii}^{(s)-1} f_i^{(s)} \right) \quad (1.32)$$

Soit avec les notations suivantes :

$$\begin{aligned} S_p^{(s)} &= \left( K_{bb}^{(s)} - K_{bi}^{(s)} K_{ii}^{(s)-1} K_{ib}^{(s)} \right) & S_p &= \sum_s A^{(s)} S_p^{(s)} A^{(s)T} \\ b_p^{(s)} &= \left( f_b^{(s)} - K_{bi}^{(s)} K_{ii}^{(s)-1} f_i^{(s)} \right) & b_p &= \sum_s A^{(s)} b_p^{(s)} \end{aligned}$$

$$S_p \mathbf{u}_b = b_p \quad (1.33)$$

qui est la version discrétisée de (1.19). La résolution de ce système permet d'obtenir  $\mathbf{u}_b$  et d'en déduire l'ensemble des  $u^{(s)}$  grâce à (1.31).

### 1.3.2 Approche duale ("FETI")

Pour cette approche, on considère directement l'équilibre local des sous-structures (1.25a). De manière à assurer automatiquement le principe d'action-réaction entre les sous-structures, on définit les réactions  $\lambda^{(s)}$  à partir d'une unique inconnue d'interface  $\underline{\lambda}_b$ , on a  $\lambda_b^{(s)} = \underline{A}^{(s)T} \underline{\lambda}_b$  (on rappelle  $\lambda_i^{(s)} = 0$ ). On a donc :

$$K^{(s)} u^{(s)} = f^{(s)} + t^{(s)T} \underline{A}^{(s)T} \underline{\lambda}_b \quad (1.34)$$

l'élimination de l'inconnue  $u^{(s)}$  passe par l'inversion de la matrice  $K^{(s)}$ . Or cette matrice correspond à l'équilibre du sous-domaine ( $s$ ) soumis à un effort imposé sur son interface, cette condition de Neumann peut conduire à un problème mal posé (sous-structure dite flottante, c'est à dire avec trop peu de conditions de Dirichlet sur son bord). On doit alors faire appel à la notion de matrice pseudo-inverse ou inverse généralisée  $K^{(s)+}$ .

Si  $A \in \mathbb{M}^{n \times n}$ , une pseudo inverse  $M^+ \in \mathbb{R}^{n \times n}$  est telle que [29]

$$\forall x \in \text{Im}(M), \quad MM^+x = x \quad (1.35)$$

Cette définition ne rend pas unique la pseudo-inverse, citons la pseudo-inverse de Moore-Penrose qui est unique et vérifie : [40]

$$\begin{aligned} MM^+M &= M & M^+MM^+ &= M^+ \\ (MM^+)^T &= MM^+ & (M^+M)^T &= M^+M \end{aligned} \quad (1.36)$$

On sépare alors le déplacement de la sous-structure  $(s)$  en une partie déformée, *ie* dans  $\text{Im}(K^{(s)})$ , et une partie rigide ou plus généralement à énergie nulle, *ie* dans  $\text{ker}(K^{(s)})$ . On fait donc apparaître une base  $R^{(s)}$  de  $\text{ker}(K^{(s)})$  et  $\alpha^{(s)}$  l'amplitude des déplacements de solide rigide associés.

$$u^{(s)} = K^{(s)+} \left( f^{(s)} + t^{(s)T} \underline{A}^{(s)T} \underline{\lambda}_b \right) + R^{(s)} \alpha^{(s)} \quad (1.37)$$

Naturellement cette expression est accompagnée de la condition explicitée dans (1.35), en notant que  $x \in \text{Im}(M) \Leftrightarrow x \perp \text{ker}(M^T)$  et en utilisant la symétrie de  $K^{(s)}$ , elle s'écrit :

$$R^{(s)T} \left( f^{(s)} + t^{(s)T} \underline{A}^{(s)T} \underline{\lambda}_b \right) = 0 \quad (1.38)$$

Mécaniquement, cette expression traduit le fait que le chargement ne doit pas exciter les déplacements de solide rigide.

En utilisant (1.37) dans (1.25b), on obtient le système suivant en  $\underline{\lambda}_b$  et  $\alpha^{(s)}$  :

$$\sum_s \underline{A}^{(s)} t^{(s)} K^{(s)+} t^{(s)T} \underline{A}^{(s)T} \underline{\lambda}_b + \sum_s \underline{A}^{(s)} t^{(s)} R^{(s)} \alpha^{(s)} = - \sum_s \underline{A}^{(s)} t^{(s)} K^{(s)+} f^{(s)} \quad (1.39)$$

En introduisant les notations suivantes :

$$\begin{aligned} S_d^{(s)} &= t^{(s)} K^{(s)+} t^{(s)T} & \underline{S}_d &= \sum_s \underline{A}^{(s)} S_d^{(s)} \underline{A}^{(s)T} \\ b_d^{(s)} &= t^{(s)} K^{(s)+} f^{(s)} & \underline{b}_d &= \sum_s \underline{A}^{(s)} b_d^{(s)} \\ \underline{G} &= \left( \dots, \underline{A}^{(s)} t^{(s)} R^{(s)}, \dots \right) & \alpha^T &= \left( \dots, \alpha^{(s)T}, \dots \right) & e^T &= \left( \dots, f^{(s)T} R^{(s)}, \dots \right) \end{aligned}$$

on obtient le système

$$\begin{pmatrix} \underline{S}_d & \underline{G} \\ \underline{G}^T & 0 \end{pmatrix} \begin{pmatrix} \underline{\lambda}_b \\ \alpha \end{pmatrix} = \begin{pmatrix} -\underline{b}_d \\ -e \end{pmatrix} \quad (1.40)$$

Remarquons que dans la littérature relative à l'approche duale,  $\underline{\lambda}_b$  est généralement pris avec un signe opposé ce qui conduit à une autre répartition des signes. Notre notation est cohérente avec celle retenue pour l'approche hybride, elle permet pour cette dernière de donner des interprétations simples aux opérateurs d'interface.

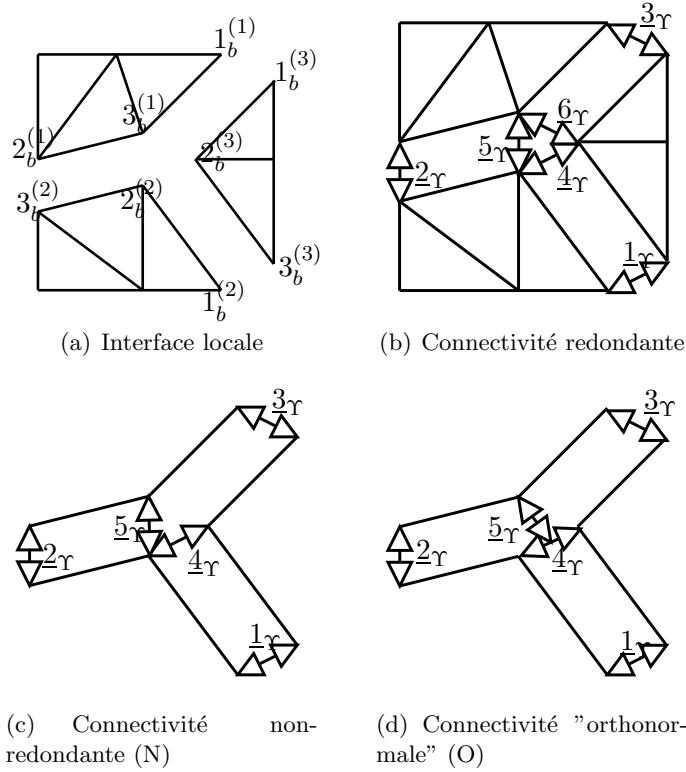
### Suppression des redondances

Il existe une possibilité très simple pour supprimer les redondances en modifiant la description de l'interface par connectivité : il suffit en effet qu'un point multiple soit connecté une et une seule fois à tous les sous-domaines. Cela conduit à la méthode FETI-A [69] et l'emploi de nouveaux types d'opérateur d'assemblage (voir figure 1.6), citons notamment le "non-redondant" et l'"orthonormal". Ces méthodes sont rarement plus intéressantes que l'approche duale classique. Un résultat important est l'équivalence entre l'usage de l'opérateur d'assemblage orthonormal et la formulation à trois champs dans le cas des maillages conformes [70]. On verra plus loin que l'usage de l'opérateur d'assemblage orthonormal est également équivalent à une variante *a priori* non-optimale de FETI [83].

#### 1.3.3 Etude des systèmes obtenus

Les systèmes condensés obtenus par approches primale et duale font intervenir des opérateurs d'interface  $S_p$  et  $\underline{S}_d$  définis comme un assemblage (somme avec projection sur





$$\begin{aligned}
 \underline{A}^{(1)} &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} & \underline{A}^{(2)} &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix} & \underline{A}^{(3)} &= \begin{pmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix} \\
 \underline{A}_N^{(1)} &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} & \underline{A}_N^{(2)} &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix} & \underline{A}_N^{(3)} &= \begin{pmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\
 \underline{A}_O^{(1)} &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{2}{\sqrt{6}} \end{pmatrix} & \underline{A}_O^{(2)} &= \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & -\frac{1}{\sqrt{6}} & 0 \end{pmatrix} & \underline{A}_O^{(3)} &= \begin{pmatrix} 0 & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 0 & 0 \\ -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & -\frac{1}{\sqrt{2}} & 0 \\ 0 & -\frac{1}{\sqrt{6}} & 0 \end{pmatrix}
 \end{aligned}$$

Figure 1.6 — Suppression des redondances dans l'approche duale

l'interface globale) de contributions locales  $S_p^{(s)}$  ou  $S_d^{(s)}$ . Ces opérateurs locaux sont appelés compléments de Schur (primal et dual). Notamment le complément de Schur primal est une discrétisation de l'opérateur de Steklov-Poincaré introduit en début de chapitre.

Ces opérateurs ont une interprétation mécanique simple. En effet considérons l'équilibre d'une sous-structure pilotée sur son interface (on suppose par ailleurs que ce pilotage n'excite pas de mode rigide). On a

$$\begin{pmatrix} K_{ii}^{(s)} & K_{ib}^{(s)} \\ K_{bi}^{(s)} & K_{bb}^{(s)} \end{pmatrix} \begin{pmatrix} u_i^{(s)} \\ u_b^{(s)} \end{pmatrix} = \begin{pmatrix} 0 \\ \lambda_b^{(s)} \end{pmatrix} \quad (1.41)$$

Si l'on considère le pilotage en déplacement  $u_b$ , le calcul de la réaction  $\lambda_b$  donne

$$\lambda_b^{(s)} = S_p^{(s)} u_b^{(s)} \quad (1.42)$$

Cette analyse rejoint naturellement les observations faites sur l'opérateur de Stecklov-Poincaré. Si l'on considère le pilotage en effort  $\lambda_b$ , le calcul du déplacement résultant sur l'interface  $u_b$  donne

$$u_b^{(s)} = S_d^{(s)} \lambda_b^{(s)} \quad (1.43)$$

On voit donc qu'au niveau local les opérateurs de Schur primal et dual sont inverses sur le sous-espace des chargements déformants. On a donc :

$$S_d^{(s)} = S_p^{(s)+} \quad (1.44)$$

Si de plus  $K^{(s)+}$  est la pseudo-inverse de Moore-Penrose de  $K^{(s)}$  alors  $S_d^{(s)}$  est la pseudo-inverse de Moore-Penrose de  $S_p^{(s)}$ .

En ce qui concerne les seconds-membres, on a la relation suivante :

$$b_d^{(s)} = S_d^{(s)} b_p^{(s)} \quad (1.45)$$

On voit ici la forte connexion entre les approches primale et duale qui est la base de plusieurs apports de cette thèse (voir les parties II, III et V).

Du point de vue résolution numérique les systèmes condensés sont symétriques positifs, systématiquement définis dans le cas de l'approche primale, définis ou semi-définis dans le cas de l'approche duale suivant l'absence ou la présence de points multiples (cependant la solution existe et est unique).

Dans les deux cas l'opérateur d'interface est un assemblage de contributions locales. L'inversion de cet opérateur est une opération coûteuse (calcul explicite des opérateurs locaux, assemblage, inversion de l'opérateur assemblé) [104, 7]. Un solveur itératif ne requiert que des produits opérateur / vecteur, or cette opération est reportée au niveau local, les opérateurs locaux ne nécessitent pas d'être calculés explicitement : il suffit de "savoir faire" les multiplications, et en fin de calcul de réassembler les contributions locales de la solution. On comprend alors que traditionnellement l'utilisation d'un solveur itératif est retenue, et étant données les propriétés des opérateurs, celle d'un gradient conjugué. De plus par rapport à la résolution itérative du système global, la résolution itérative du problème condensé présente de nombreux autres avantages :

- le problème est de plus petite dimension ;
- les opérateurs de Schur primal et dual sont les discrétisations d'opérateurs de Stecklov-Poincaré qui sont des opérateurs différentiels du premier ordre alors qu'une matrice de rigidité est la discrétisation d'un opérateur différentiel du second ordre, soit  $h$  la taille caractéristique des éléments le conditionnement varie donc en  $1/h$  au lieu de  $1/h^2$  [86] ;
- enfin point fondamental, des préconditionneurs efficaces ont été développés.

Nous abandonnons ici temporairement l'étude des méthodes de décomposition de domaine afin de nous consacrer à celle des solveurs itératifs de manière à introduire les notions nécessaires à leur mise en place numérique efficace.



---

## Solveurs itératifs de Krylov

Les méthodes itératives de Krylov pour la résolution d'un système linéaire sont très populaires et ont été largement classifiées et étudiées. Elles appartiennent à la classe plus large des méthodes de projection et consistent à approcher un vecteur  $S^{-1}b$  (solution du système  $Sx = b$ ) par un vecteur  $p(S)b$  où  $p$  est un polynôme adroitement construit par la méthode choisie.

Ce chapitre a pour but de présenter les bases de ces méthodes et plus spécifiquement de deux des principaux solveurs utilisés en calcul scientifique. Nous proposons ensuite une étude assez poussée des facteurs influant la convergence et d'une technique d'accélération classique ce qui nous permettra dans la partie IV de définir des stratégies d'accélération pertinentes dans le cadre de la multirésolution.

Sauf citation supplémentaire, tous les résultats et algorithmes présentés ici se trouvent dans [89].

Pour ce chapitre nous considérons la résolution itérative du système  $Sx = b$ .  $S$  est une matrice  $n \times n$  et  $b$  un vecteur de  $\text{Im}(S)$ . La  $i^{\text{eme}}$  itération conduit à l'approximation  $x_i$  de la solution, le résidu associé est  $r_i = b - Sx_i = S(x - x_i)$ . L'initialisation est notée  $x_0$  (généralement  $x_0 = 0$ ). La base canonique (orthonormale) de  $\mathbb{R}^n$  est notée  $(e_1, \dots, e_n)$ .

## 2.1 Principe des solveurs de Krylov

Les solveurs de Krylov reposent sur la construction itérative d'un sous-espace de Krylov  $\mathcal{K}_m(S, r_0)$  défini de la manière suivante :

$$\mathcal{K}_m(S, r_0) = \text{Vect}(r_0, \dots, S^{m-1}r_0) \quad (2.1)$$

La résolution du système linéaire consiste alors à rechercher  $x_m$  sous les contraintes suivantes :

$$\begin{cases} x_m \in x_0 + \mathcal{K}_m(S, r_0) \\ r_m \perp \mathcal{K}_m(S, r_0) \end{cases} \quad (2.2)$$

où le choix de la relation d'orthogonalité permet de définir les différentes approches.

La base de ces méthodes demeure la procédure d'Arnoldi (algorithme I.1) qui permet de factoriser une matrice sous la forme d'Hessenberg. Elle correspond à une projection orthogonale.

---

### Algorithme I.1 Procédure d'Arnoldi

---

Vecteur initial  $v_0 / \|v_0\| = 1$

**pour**  $j = 0, \dots, m$  **faire**

Calculer  $h_{ij} = (Sv_i, v_j)$  pour  $0 \leq i \leq j$

Calculer  $w_j = Sv_j - \sum_{i=1}^j h_{ij}v_i$

$h_{(j+1)j} = \|w_j\|_2$

**si**  $h_{(j+1)j} = 0$  **alors**

stop

**sinon**

$v_{j+1} = w_j / h_{(j+1)j}$

**fin si**

**fin pour**

---

Si l'algorithme arrive à la  $(m-1)^{\text{eme}}$  itération, les vecteurs  $(v_i)$  forment une base orthonormale de  $\mathcal{K}_m(S, v_0)$ . En notant  $V_m = (v_0, \dots, v_{m-1})$ ,  $\bar{H}_m = (h_{ij})_{1 \leq i \leq m, 1 \leq j \leq m-1}$  on a :

$$SV_m = V_{m+1}\bar{H}_m \quad (2.3)$$

$$= V_m\bar{H}_m + w_m e_m^T \quad (2.4)$$

$$V_m^T SV_m = H_m \quad (2.5)$$

La matrice  $H_m$  est dite matrice Hessenberg supérieure (triangulaire supérieure + première sous-diagonale), elle joue un rôle fondamental dans l'analyse des performances des solveurs.

Notons que dans le cas où  $S$  est une matrice symétrique, la procédure d'Arnoldi se simplifie en procédure de Lanczos symétrique et que dans ce cas la matrice de Hessenberg est tridiagonale symétrique.

## 2.2 Solveurs usuels

Nous présentons ici les deux principaux solveurs de Krylov employés. Tout d'abord GMRes [90] qui est très proche de la procédure d'Arnoldi et convient à tout type de matrice puis le gradient conjugué qui n'est adapté qu'aux matrices symétriques définies positives.

Naturellement la résolution itérative d'un système linéaire sous-entend une tolérance assumée sur la précision de la solution obtenue, nous notons ainsi  $\varepsilon$  la valeur du critère de convergence.

### 2.2.1 GMRes

Soit  $S$  est une matrice quelconque telle que  $b \in \text{Im } S$ , l'algorithme GMRes (alg. I.2) consiste en une projection oblique basée sur la construction de l'espace de Krylov  $\mathcal{K}_m(S, v_0)$  avec  $v_0 = r_0 / \|r_0\|_2$ . Le principe de recherche est le suivant :

$$\begin{cases} x_m \in x_0 + \mathcal{K}_m(S, r_0) \\ r_m \perp S\mathcal{K}_m(S, r_0) \end{cases} \quad (2.6)$$

ce qui revient à trouver  $x_m \in x_0 + \mathcal{K}_m(S, r_0)$  minimisant  $\|r_m\|_2$ .

---

#### Algorithme I.2 GMRes

---

- 1: Calculer  $r_0 = b - Sx_0$ ,  $v_0 = r_0 / \|r_0\|_2$
  - 2: **pour**  $j = 0, \dots, m - 1$  **faire**
  - 3:   Calculer  $w_j = Sv_j$
  - 4:   **pour**  $i = 0, \dots, j$  **faire**
  - 5:      $h_{ij} = (v_i, w_j)$
  - 6:      $w_j = w_j - h_{ij}v_i$
  - 7:   **fin pour**
  - 8:    $h_{(j+1)j} = \|w_j\|_2$
  - 9:   **si**  $\|r_j\|_2 \leq \varepsilon$  **alors**
  - 10:     stop
  - 11:   **sinon**
  - 12:      $v_{j+1} = w_j / h_{(j+1)j}$
  - 13:   **fin si**
  - 14: **fin pour**
  - 15: Calculer  $y_m$  minimisant  $\| \|r_0\|_2 e_1 - \bar{H}_m y \|_2$  et poser  $x_m = x_0 + V_m y_m$
- 

La particularité de GMRes est de ne pas calculer l'approximation de la solution à chaque itération, on ne calcule que l'approximation finale  $x_m$ . Cependant une mise en oeuvre adroite du solveur permet d'avoir accès à chaque itération  $j$  à la valeur de la norme du résidu  $\|r_j\|_2$ .

Au niveau complexité de calcul, chaque itération consiste en une orthogonalisation complète de  $w_j$  vis à vis de  $\mathcal{K}_j$ ; une mise en oeuvre adroite du solveur permet de ramener le coût du calcul de la solution finale à une inversion de matrice triangulaire de dimension  $m$ .

L'algorithme GMRes(m) (ou GMRes avec redémarrage) consiste à arrêter le calcul avant convergence à un pas  $m$  fixé *a priori* et de relancer le calcul en initialisant par  $x_m$ . L'objectif est de minimiser les calculs d'orthogonalisation en limitant la taille de l'espace de Krylov [23]. Cette variante peut conduire à des stagnations pour des matrices non définies positives, par ailleurs les progrès des calculateurs en terme de puissance de calcul et de mémoire rendent cette tentative d'économie inutile dans le cadre des problèmes étudiés.

### 2.2.2 Gradient Conjugué

Soit  $S$  une matrice symétrique définie positive, l'algorithme du gradient conjugué consiste en une projection orthogonale basée sur l'espace de Krylov  $\mathcal{K}_m(S, r_0)$ . Le principe de recherche est le suivant :

$$\begin{cases} x_m \in x_0 + \mathcal{K}_m(S, r_0) \\ r_m \perp \mathcal{K}_m(S, r_0) \end{cases} \quad (2.7)$$

ce qui revient à trouver  $x_m \in x_0 + \mathcal{K}_m(S, r_0)$  minimisant  $\|x_m - x\|_S$ .

Du fait des propriétés de  $S$ , des relations de conjugaison (orthogonalité) apparaissent, conduisant à l'algorithme I.3.

---

#### Algorithme I.3 Gradient conjugué

---

- 1: Calculer  $r_0 = b - Sx_0$ , poser  $w_0 = r_0$
  - 2: **pour**  $j = 0, \dots, m$  **faire**
  - 3:    $\alpha_j = (r_j, r_j) / (Sw_j, w_j)$
  - 4:    $x_{j+1} = x_j + \alpha_j w_j$
  - 5:    $r_{j+1} = r_j - \alpha_j Sw_j$
  - 6:    $\beta_j = (r_{j+1}, r_{j+1}) / (r_j, r_j)$
  - 7:    $w_{j+1} = r_{j+1} + \beta_j w_j$
  - 8: **fin pour**
- 

L'algorithme repose sur la construction de différentes bases de  $\mathcal{K}_m(S, r_0)$  :  $R_m$  (base des résidus) est orthogonale,  $W_m$  (base des directions de descente) est  $S$ -orthogonale. L'étape 6 – 7 de l'algorithme I.3 correspond ainsi à la  $S$ -orthogonalisation de  $w_{j+1}$  vis-à-vis de  $w_j$  ce qui théoriquement implique l'orthogonalité de  $w_{j+1}$  vis-à-vis de toutes les directions de descente précédentes. Cependant numériquement cette conjugaison a tendance à être perdue au fur et à mesure des itérations, il convient alors d'utiliser une orthogonalisation complète des directions de descente ce qui conduit à l'algorithme I.4.

---

#### Algorithme I.4 Gradient conjugué réorthogonalisé

---

- 1: Calculer  $r_0 = b - Sx_0$ , poser  $w_0 = r_0$
  - 2: **pour**  $j = 0, \dots, m$  **faire**
  - 3:    $\alpha_j = (r_j, r_j) / (Sw_j, w_j)$
  - 4:    $x_{j+1} = x_j + \alpha_j w_j$
  - 5:    $r_{j+1} = r_j - \alpha_j Sw_j$
  - 6:   Pour  $0 \leq i \leq j$ ,  $\beta_j^i = -(r_{j+1}, Sw_i) / (w_i, Sw_i)$
  - 7:    $w_{j+1} = r_{j+1} + \sum_{i=1}^j \beta_j^i w_i$
  - 8: **fin pour**
- 

Pour les problèmes de structure, la réorthogonalisation est obligatoire. Différentes mises en oeuvre sont d'ailleurs possibles (entre autres Gram-Schmidt, Gram-Schmidt modifié, Gram-Schmidt itératif [58, 41]) permettant d'obtenir des précisions plus ou moins élevées. Notre expérience nous conduit à préférer l'algorithme de Gram-Schmidt modifié (celui utilisé dans GMRes, algorithme I.2) au Gram-Schmidt classique (celui utilisé dans le gradient conjugué, algorithme I.4). Notons qu'une fois réorthogonalisé, le gradient conjugué devient presque aussi "coûteux" que GMRes, au calcul de la solution finale près puisque le gradient conjugué permet de disposer à toute itération de l'approximation en cours de la solution.

De manière à relier le gradient conjugué à la procédure d'Arnoldi, notons le résultat suivant qui permet de reconstruire la matrice de Hessenberg à partir des coefficients du





fur et à mesure que le processus de résolution avance. L'analyse que nous donnons ici repose sur l'hypothèse que le second membre se décompose sur une base propre de l'opérateur, et pour simplifier nous supposons que l'opérateur est symétrique (donc diagonalisable).

Soit  $N_j$  une base orthonormale de l'espace de Krylov  $\mathcal{K}_j(S, r_0)$ , on appelle matrice de Rayleigh associée à la base  $N_j$  la matrice  $B_j = N_j^T S N_j$ . On appelle valeurs de Ritz les valeurs propres  $(\theta_j^i)_{0 \leq i < j}$  de la matrice de Rayleigh (on note  $\Theta_j = \text{diag}(\theta_j^i)_{0 \leq i < j}$ ) ; soit  $Q_j^B$  une base propre orthonormale de  $B_j$ , on appelle vecteurs de Ritz les vecteurs  $Y_j = N_j Q_j^B$ . Les valeurs et vecteurs de Ritz sont naturellement indépendants de la base  $N_j$ . Une propriété importante est qu'ils peuvent être calculés à partir de la matrice Hessenberg qui est la matrice de Rayleigh associée à la base d'Arnoldi  $V_j$ .

On rappelle que le projecteur orthogonal sur l'espace de Krylov  $P_{\mathcal{K}_j}$  peut être construit à partir d'une base orthonormale  $N_j$  comme suit :

$$P_{\mathcal{K}_j} = N_j N_j^T \quad (2.12)$$

On appelle restriction de  $S$  sur l'espace de Krylov l'opérateur  $S_j$  suivant :

$$S_j = P_{\mathcal{K}_j} S P_{\mathcal{K}_j} = N_j N_j^T S N_j N_j^T = N_j B_j N_j^T \quad (2.13)$$

On voit très simplement que les valeurs et vecteurs de Ritz sont les éléments propres de la matrice  $S_j$ .

$$S_j Y_j = S_j N_j Q_j^B = N_j B_j N_j^T N_j Q_j^B = N_j B_j Q_j^B = N_j Q_j^B \Theta_j = Y_j \Theta_j \quad (2.14)$$

Comme il semble assez naturel, on peut montrer que les valeurs de Ritz convergent vers les valeurs propres de l'opérateur  $S$ . La résolution itérative se traduisant par une série de multiplication par l'opérateur  $S$ , en faisant l'analogie avec la méthode des puissances itérées pour le calcul des valeurs propres, on comprend que l'on capte en premier lieu la partie haute du spectre de l'opérateur.

Il a été montré dans [103], un théorème fondamental qui lie la convergence des valeurs de Ritz à celle du gradient conjugué :

$$x - x_j = \pi_j(S)(x - x_0) \quad \text{avec} \quad \pi_j(\xi) = \prod_{i=0}^{j-1} \frac{\theta_j^i - \xi}{\theta_j^i} \quad (2.15)$$

De fait, on observe que lorsque les valeurs de Ritz donnent une approximation avec une précision suffisante de certaines valeurs propres, l'algorithme se comporte lors des itérations suivantes comme si les vecteurs propres correspondant avaient été éliminés de la base propre active participant au processus de résolution. Si ces vecteurs sont associés à des valeurs propres extrêmes de la distribution spectrale (typiquement valeur propre maximale), le conditionnement effectif  $\kappa_{eff}(j)$  qui est défini à l'itération  $j$  par le rapport entre la plus grande et la plus petite valeur propre associées à des vecteurs propres encore actifs et représente le paramètre qui régit le taux de convergence au cours du processus de résolution, est plus petit que le conditionnement actif. L'évolution du conditionnement effectif qui décroît donc au cours des itérations justifie la superconvergence des processus itératifs (accélération rapide du taux de convergence au bout d'un certain nombre d'itérations). Une compilation des principaux résultats et démonstrations relatifs aux valeurs de Ritz peut être trouvée dans [77].

## 2.5 Méthodes Krylov-augmenté, mise en oeuvre par projecteur

Les méthodes Krylov-augmenté [11, 88] sont largement employées pour adjoindre des contraintes lors de la résolution des systèmes. Le principe est de définir un sous-espace  $\mathcal{C}$  de  $\mathbb{R}^n$  de dimension  $n_c$  représenté par une matrice rectangulaire  $C$  ( $\text{Im}(C) = \mathcal{C}$ ), on suppose pour plus de simplicité que  $C$  est une base de  $\mathcal{C}$  ( $\text{rg}(C) = n_c$ ).

Le principe est de définir un espace de Krylov augmenté  $\tilde{\mathcal{K}}_m(S, r_0, C) = \mathcal{K}_m(S, r_0) + \text{Im}(C)$ , et d'utiliser le principe de recherche suivant :

$$\begin{cases} x_m \in x_0 + \tilde{\mathcal{K}}_m(S, r_0, C) \\ r_m \perp \tilde{\mathcal{K}}_m(S, r_0, C) \end{cases} \quad (2.16)$$

On choisit généralement de mettre en oeuvre les méthodes Krylov-augmenté par des algorithmes projetés. Le principe est de séparer l'espace de recherche en deux sous-espaces supplémentaires  $\text{Im}(C)$  et  $\text{Ker}(C^T S)^1$ . La partie dans  $\text{Im}(C)$  est détectée à l'initialisation, celle dans  $\text{ker}(C^T S)$  est cherchée itérativement, un projecteur  $P$  sur  $\text{ker}(C^T S)$  assure que la recherche se fait dans le bon espace :

$$x = x_0 + Px^* \quad (2.17)$$

$$C^T r_0 = C^T (b - Sx_0) = 0 \quad (2.18)$$

$$C^T SP = 0 \quad (2.19)$$

Ce qui conduit à :

$$x_0 = C (C^T SC)^{-1} C^T b \quad (2.20)$$

$$P = I - C (C^T SC)^{-1} C^T S \quad (2.21)$$

le système devient alors :

$$SPx^* = b - Sx_0 \quad (2.22)$$

$$\text{ou } P^T SPx^* = P^T (b - Sx_0) \quad (2.23)$$

la dernière forme permet de conserver l'éventuelle symétrie du système. Le système en  $x^*$  est bien posé, l'algorithme employé converge vers une solution qu'il convient de naturellement projeter avant de l'ajouter à la contribution initiale.

Bien que l'on puisse prouver que le problème ainsi projeté possède un meilleur conditionnement que le problème original [18], l'efficacité de la méthode repose essentiellement sur un choix adapté de la matrice  $C$ , choix qui reste le plus souvent un problème ouvert. Dans le cadre des décompositions de domaine, nous verrons qu'il peut être guidé par plusieurs considérations. Nous montrerons également que dans le cadre de la résolution d'une succession de systèmes linéaires des stratégies efficaces ont été développées, notamment au cours de cette thèse.

---

<sup>1</sup>on suppose que  $S$  est telle que la supplémentarité est vérifiée, une condition suffisante est  $S$  symétrique définie positive.



---

## Résolution itérative du problème mécanique condensé à l'interface

Le chapitre précédent laisse entendre l'importance du conditionnement dans la résolution d'un système linéaire par un solveur itératif. Il évoque également l'opportunité d'adjoindre des problèmes grossiers pour accélérer la convergence du processus.

Ce chapitre met en place les préconditionneurs et problèmes grossiers qui apparaissent assez naturellement pour les méthodes de décomposition de domaine, leur conférant une grande efficacité.

Ensuite, certains aspects pratiques relatifs aux méthodes de décomposition de domaine sont énumérés, concernant notamment la mesure de la convergence et des performances, ainsi que certains critères pour évaluer la pertinence de la décomposition d'une structure donnée.

Pour finir une notation synthétique des méthodes de décomposition de domaine est donnée dans la dernière section.

### 3.1 Préconditionneurs

La popularité de la résolution itérative des méthodes de décomposition de domaine tient dans l'existence de preconditionneurs efficaces conduisant à de bons résultats de convergence et des performances de calcul intéressantes.

#### 3.1.1 Préconditionneurs optimaux

Pour être efficace d'un point de vue calcul, un preconditionneur doit autant que possible consister en un assemblage de contributions locales. La stratégie retenue consiste alors d'approcher l'inverse de l'opérateur d'interface (qui est une somme de contributions locales) par la somme pondérée des inverses des contributions locales. Ce qui conduit aux preconditionneurs suivants :

$$\tilde{S}_p^{-1} = \sum_s W^{(s)} A^{(s)} S_d^{(s)} A^{(s)T} W^{(s)T} \quad (3.1)$$

$$\tilde{S}_d^{-1} = \sum_s \underline{W}^{(s)} \underline{A}^{(s)} S_p^{(s)} \underline{A}^{(s)T} \underline{W}^{(s)T} \quad (3.2)$$

Le preconditionneur de l'approche primale consiste à résoudre des problèmes de Neumann de part et d'autre de l'interface d'où son nom de preconditionneur Neumann-Neumann. Celui de l'approche duale, pour des raisons identiques, est le preconditionneur de Dirichlet.

Les matrices diagonales de pondération (*scaling*)  $W^{(s)}$  et  $\underline{W}^{(s)}$  jouent un rôle fondamental dans les performances du preconditionneur, elles permettent notamment de respecter des considérations mécaniques [82]. Dans le cas d'une structure homogène, elles sont généralement égales à l'inverse de la multiplicité du degré de liberté considéré. Dans le cas d'une structure hétérogène, elles permettent de capter les différences de rigidité de part et d'autre de l'interface. La rigidité est généralement évaluée en utilisant le coefficient diagonal de la matrice de rigidité (on considère que le degré de liberté  $j$  possède la même numérotation sur toutes les sous-structures qui le contiennent et sur l'interface) :

$$W_j^{(s)} = \frac{K_{jj}^{(s)}}{\sum_{r \in \iota(j)} K_{jj}^{(r)}} \quad (3.3)$$

$$\underline{W}_j^{(s)} = \frac{K_{jj}^{(t)}}{\sum_{r \in \iota(j)} K_{jj}^{(r)}} \text{ avec } j \in \Upsilon^{(s,t)} \quad (3.4)$$

Notons que cette pondération se révèle insuffisante sur certains problèmes multiphysiques et qu'une amélioration est proposée au chapitre 5.

Notons enfin la relation de complémentarité suivante entre les pondérations :

$$A^{(s)T} W^{(s)} A^{(s)} + \underline{A}^{(s)T} \underline{W}^{(s)} \underline{A}^{(s)} = I_{\Upsilon^{(s)}} \quad (3.5)$$

Pour ces preconditionneurs (avec les bons problèmes grossiers évoqués dans la section suivante), les conditionnements des approches primale et duale sont comparables [62, 53] :

$$\kappa = O\left(1 + \log \frac{H}{h}\right)^2 \quad (3.6)$$

où  $h$  représente la taille caractéristique des éléments-finis et  $H$  celle des sous-structures (notamment  $\frac{H}{h}$  représente, en 1D, le nombre d'éléments par sous-structure).

### 3.1.2 Préconditionneurs allégés

Les preconditionneurs optimaux correspondent à des calculs lourds (stockage et factorisation d'une matrice de Dirichlet ou de Neumann, descente-remontée à chaque itération). L'idée des preconditionneurs allégés est de dégrader les performances numériques au profit de la rapidité des calculs de manière à obtenir de meilleures performances en terme de temps CPU.

Des preconditionneurs allégés ont essentiellement été développés pour l'approche duale [39, 36]. Ils sont basés sur des approximations de l'opérateur de Schur primal local :

$$S_p^{(s)} \approx K_{bb}^{(s)} \Rightarrow (\tilde{S}_d^{-1})_L = \sum_s \underline{W}^{(s)} \underline{A}^{(s)} K_{bb}^{(s)} \underline{A}^{(s)T} \underline{W}^{(s)T} \quad (3.7)$$

$$S_p^{(s)} \approx \text{diag}(K_{bb}^{(s)}) \Rightarrow (\tilde{S}_d^{-1})_{SL} = \sum_s \underline{W}^{(s)} \underline{A}^{(s)} \text{diag}(K_{bb}^{(s)}) \underline{A}^{(s)T} \underline{W}^{(s)T} \quad (3.8)$$

le premier de ces preconditionneurs porte le nom de *lumped*, mécaniquement il consiste à négliger l'influence des points intérieurs de la sous-structure dans le comportement de l'interface ("peau" liée à un massif rigide), le second *superlumped* consiste à considérer chaque degré de liberté de l'interface indépendant (ressort lié à un bâti rigide).

## 3.2 Problèmes grossiers

On donne le nom de problème grossier à tout problème de faible dimension posé sur l'ensemble de l'interface. Plusieurs de ces problèmes interviennent dans le cadre des décompositions de domaine.

### 3.2.1 Admissibilité dans FETI

Si on reprend le système de FETI (1.40)

$$\begin{pmatrix} \underline{S}_d & \underline{G} \\ \underline{G}^T & 0 \end{pmatrix} \begin{pmatrix} \underline{\lambda}_b \\ \alpha \end{pmatrix} = \begin{pmatrix} -\underline{b}_d \\ -e \end{pmatrix} \quad (3.9)$$

la seconde ligne du système  $\underline{G}^T \underline{\lambda}_b = -e$  est un problème grossier dit "condition d'admissibilité du multiplicateur de Lagrange". Ce type de contrainte est à rapprocher des algorithmes Krylov-augmenté (mais ici la contrainte porte sur l'inconnue et non sur le résidu). La résolution est d'ailleurs conduite par un algorithme projeté :

$$\underline{\lambda}_b = \underline{\lambda}_0 + \underline{P} \underline{\lambda}^* \quad (3.10)$$

$$\underline{G}^T \underline{\lambda}_0 = -e \quad (3.11)$$

$$\underline{G}^T \underline{P} = 0 \quad (3.12)$$

ce qui conduit à :

$$\underline{\lambda}_0 = -\underline{Q} \underline{G} (\underline{G}^T \underline{Q} \underline{G})^{-1} e \quad (3.13)$$

$$\underline{P} = \underline{I} - \underline{Q} \underline{G} (\underline{G}^T \underline{Q} \underline{G})^{-1} \underline{G}^T \quad (3.14)$$

$$\underline{P}^T \underline{S}_d \underline{P} \underline{\lambda}^* = -\underline{P}^T (\underline{b}_d - \underline{S}_d \underline{\lambda}_0) \quad (3.15)$$

Le choix de la matrice  $\underline{Q}$  est une liberté de l'utilisateur. Dès [39] il a été conseillé de faire apparaître dans  $\underline{Q}$  les différences de rigidité entre les sous-structures, ce qui conduit aux choix classiques suivants :

- structure homogène :  $\underline{Q} = \underline{I}$ ;
- structure fortement hétérogène :  $\underline{Q} = \tilde{\underline{S}}_d^{-1}$  ;
- structure légèrement hétérogène :  $\underline{Q} = (\tilde{\underline{S}}_d^{-1})_L$  ou  $\underline{Q} = (\tilde{\underline{S}}_d^{-1})_{SL}$ .

Notons que le choix du projecteur *superlumped* se révèle souvent excellent. Enfin dans [83] il a été prouvé que les approches de type FETI-A avec opérateur d'assemblage orthonormal sont équivalentes à un choix de matrice  $\underline{Q}$  basé sur l'inverse de la multiplicité des degrés de liberté  $\underline{Q} = \underline{\text{diag}}_i(\frac{1}{m_i})$ .

### 3.2.2 Préconditionnement optimal dans l'approche primale

Si on reprend l'expression du préconditionneur Neumann-Neumann de l'approche primale, on voit qu'elle fait apparaître  $S_d^{(s)}$  la matrice de Schur dual local qui correspond à la résolution d'un problème de Neumann éventuellement mal posé si la sous-structure ne possède pas suffisamment de condition de Dirichlet. De manière à s'assurer que le problème est bien posé, il suffit d'imposer que la matrice  $K^{(s)+}$  ne multiplie que des vecteurs appartenant à  $\text{Im}(K^{(s)})$ . Le préconditionneur multiplie le vecteur résidu  $r$ , on a donc :

$$\tilde{S}_p^{-1} r = \sum_s W^{(s)} A^{(s)} t^{(s)} K^{(s)+} t^{(s)T} A^{(s)T} W^{(s)T} r$$

la condition d'optimalité s'écrit :

$$\begin{aligned} \forall s, t^{(s)T} A^{(s)T} W^{(s)T} r &\in \text{Im}(K^{(s)}) \\ \text{donc } \forall s, t^{(s)T} A^{(s)T} W^{(s)T} r &\perp \ker(K^{(s)}) \\ \text{soit } R^{(s)} &\text{ base de } \ker(K^{(s)}) \\ \forall s, R^{(s)T} t^{(s)T} A^{(s)T} W^{(s)T} r &= 0 \\ \forall s, \left( W^{(s)} A^{(s)} t^{(s)} R^{(s)} \right)^T r &= 0 \\ \text{soit } H &= \left( \dots, W^{(s)} A^{(s)} t^{(s)} R^{(s)}, \dots \right) \\ H^T r &= 0 \end{aligned} \tag{3.16}$$

Cette condition rentre directement dans le cadre des algorithmes Krylov-augmenté présentés précédemment, la mise en oeuvre se fait à l'aide d'un projecteur. L'adjonction du projecteur rend l'approche primale très efficace (elle permet notamment d'obtenir le conditionnement présenté (3.6)).

### 3.2.3 Algorithmes de second niveau

On appelle "premier niveau" la prise en compte des déplacements de solide rigide des sous-structures, problème qui apparaît naturellement lors de l'écriture de l'opérateur ou du préconditionneur. Le second niveau correspond à l'adjonction de contraintes optionnelles (matrice  $C$  des approches Krylov-augmenté) [34, 28]. Le choix de la matrice  $C$  n'obéit à aucune obligation, cependant l'expérience a conduit aux résultats suivants :

- Pour les problèmes 2D (plaque, coque, élasticité bidimensionnelle), un choix efficace consiste à forcer la nullité du résidu sur les points multiples.
- Pour les problèmes de dynamique, les modes rigides (qui n'apparaissent plus dans l'opérateur ou le préconditionneur) peuvent être utilisés.
- Pour les problèmes d'ondes [16, 33, 32], on injecte couramment une base d'ondes planes.
- D'une manière générale l'utilisation des vecteurs propres actifs du problème préconditionné conduit à des gains significatifs. Cependant leur calcul est coûteux d'où les idées suivantes :
  - approcher les modes propres par des contributions interface par interface (idée sous-jacente dans l'approche LaTIn [55]) ;
  - dans le cadre de la résolution d'une séquence de systèmes linéaires réutiliser l'information pour construire une approximation des vecteurs propres (voir partie IV).

Pour conclure cette partie sur la résolution itérative des approches primale et duale, nous donnons (alg. I.5 et I.6) les algorithmes de résolution de ces deux approches avec contraintes optionnelles (les stratégies avec plusieurs projecteurs sont mieux expliquées lors de la présentation de l'approche hybride (chapitre 7), les aspects mise en oeuvre sont traités dans la partie V).

---

**Algorithme I.5** Approche primale 2 niveaux

---

- 1:  $H = (\dots, W^{(s)}A^{(s)}t^{(s)}R^{(s)}, \dots)$ ,  $P = I - [H, C] ([H, C]^T S_p [H, C])^{-1} [H, C]^T S_p$
  - 2:  $\dot{x}_0$  arbitraire
  - 3:  $x_0 = [H, C] ([H, C]^T S_p [H, C])^{-1} [H, C]^T b_p + P\dot{x}_0$
  - 4:  $r_0 = P^T (b - S_p x_0) = (b - S_p \dot{x}_0)$
  - 5:  $z_0 = \tilde{S}_p^{-1} r_0$ ,  $w_0 = z_0$
  - 6: **pour**  $j = 0, \dots, m$  **faire**
  - 7:    $\alpha_j = (z_j, r_j) / (P^T S_p P w_j, w_j)$
  - 8:    $x_{j+1} = x_j + \alpha_j w_j$
  - 9:    $r_{j+1} = r_j - \alpha_j P^T S_p P w_j$
  - 10:    $z_{j+1} = \tilde{S}_p^{-1} r_{j+1}$
  - 11:   **Pour**  $0 \leq i \leq j$ ,  $\beta_j^i = -(z_{j+1}, P^T S_p P w_i) / (w_i, P^T S_p P w_i)$
  - 12:    $w_{j+1} = z_{j+1} + \sum_{i=1}^j \beta_j^i w_i$
  - 13: **fin pour**
- 

### 3.3 Facteurs influençant les performances des méthodes de décomposition de domaine

Dans cette section, nous essayons de résumer différents aspects pratiques relatifs aux méthodes de décomposition de domaine.

#### 3.3.1 Mesure de la convergence

Un premier problème concerne l'évaluation de la convergence des méthodes de décomposition de domaine. On dispose en effet de plusieurs mesures permettant d'évaluer l'écart à la solution. Le critère mécaniquement le plus acceptable est le résidu global, il



---

**Algorithme I.6** Approche duale 2 niveaux

---

- 1:  $\underline{G} = (\dots, \underline{A}^{(s)} t^{(s)} R^{(s)}, \dots)$ ,  $\underline{P} = I - \underline{Q}\underline{G}(\underline{G}^T \underline{Q}\underline{G})^{-1} \underline{G}^T$
  - 2:  $\underline{C}^* = \underline{P}\underline{C}$ ,  $\underline{P}^* = \underline{I} - \underline{C}^* (\underline{C}^{*T} \underline{S}_d \underline{C}^*)^{-1} \underline{C}^{*T} \underline{S}_d$
  - 3:  $\underline{\ddot{x}}_0$  arbitraire
  - 4:  $\underline{\dot{x}}_0 = \underline{Q}\underline{G}(\underline{G}^T \underline{Q}\underline{G})^{-1} e + \underline{P}\underline{\ddot{x}}_0$
  - 5:  $\underline{\dot{r}}_0 = \underline{P}^T (b_d - \underline{S}_d \underline{\dot{x}}_0)$
  - 6:  $\underline{x}_0 = \underline{C}^* (\underline{G}^T \underline{Q}\underline{G})^{-1} \underline{G}^T \underline{\dot{r}}_0$
  - 7:  $\underline{r}_0 = (\underline{P}^*)^T (\underline{\dot{r}}_0 - \underline{S}_d \underline{x}_0)$
  - 8:  $\underline{z}_0 = \underline{\tilde{S}}_d^{-1} \underline{r}_0$ ,  $\underline{w}_0 = \underline{z}_0$
  - 9: **pour**  $j = 0, \dots, m$  **faire**
  - 10:  $\alpha_j = (z_j, r_j) / (\underline{P}^{*T} \underline{P}^T \underline{S}_d \underline{P} \underline{P}^* w_j, w_j)$
  - 11:  $\underline{x}_{j+1} = \underline{x}_j + \alpha_j w_j$
  - 12:  $\underline{r}_{j+1} = \underline{r}_j - \alpha_j \underline{P}^{*T} \underline{P}^T \underline{S}_d \underline{P} \underline{P}^* w_j$
  - 13:  $\underline{z}_{j+1} = \underline{\tilde{S}}_d^{-1} \underline{r}_{j+1}$
  - 14: Pour  $0 \leq i \leq j$ ,  $\beta_j^i = -(z_{j+1}, \underline{P}^{*T} \underline{P}^T \underline{S}_d \underline{P} \underline{P}^* w_i) / (w_i, \underline{P}^{*T} \underline{P}^T \underline{S}_d \underline{P} \underline{P}^* w_i)$
  - 15:  $\underline{w}_{j+1} = \underline{z}_{j+1} + \sum_{i=1}^j \beta_j^i w_i$
  - 16: **fin pour**
- 

évalue la convergence du système linéaire complet :

$$\varepsilon_g = \frac{\|Ku - f\|}{\|f\|} \quad (3.17)$$

Il faut notamment remarquer que le calcul de la norme d'une grandeur définie sur l'ensemble de la structure requiert une attention particulière (notamment aux points multiples). Le problème est alors naturellement de relier ce critère à des mesures directement accessibles pendant la résolution du problème d'interface. Etant donné que les calculs locaux sont réalisés à l'aide de solveurs directs, on suppose que les inversions locales sont exactes. On a alors le résultat suivant :

$$\|Ku - f\| = \|S_p u_b - b_p\| \quad (3.18)$$

ce qui relie directement la norme du résidu primal à celle du résidu global et résout le problème de l'évaluation de la mesure de la convergence pour l'approche primale.

En ce qui concerne l'approche duale, on a le résultat suivant qui lie la norme du résidu projeté au saut de déplacement entre les sous-structures :

$$\underline{Pr} = \sum_s \underline{A}^{(s)} u^{(s)} \quad (3.19)$$

$$\sum_s \underline{A}^{(s)} u^{(s)} = \underline{\Delta}(u) \quad (3.20)$$

$$(\underline{\Delta}(u))|_{\Gamma^{(i,j)}} = u|_{\Gamma^{(i,j)}}^{(i)} - u|_{\Gamma^{(i,j)}}^{(j)} \quad (3.21)$$

Cependant cette interprétation mécanique du résidu dual ne permet pas d'évaluer le critère global de convergence. Le principal problème est de reconstruire un champ de déplacement unique à l'interface à partir de champs de déplacement locaux ("assez bien" convergés mais ne correspondant pas parfaitement). L'approche classique consiste à réaliser une moyenne

pondérée des déplacements locaux :

$$u_b = \sum_s A^{(s)} \underline{A}^{(s)T} \underline{W}^{(s)} \underline{\Delta}(u) \quad (3.22)$$

ce qui permet à partir de la relation (3.18) d'obtenir le résultat suivant :

$$\|Ku - f\| = \sum_s A^{(s)} \mathcal{S}_p^{(s)} \underline{A}^{(s)T} \underline{W}^{(s)} \underline{\Delta}(u) \quad (3.23)$$

où l'on voit que si l'utilisateur utilise le préconditionneur de Dirichlet pour l'approche duale, le calcul du résidu global peut être effectué à moindre coût (simplement un assemblage primal au cours du préconditionnement).

Pour finir il convient d'observer les normes disponibles pour le résidu du solveur itératif : pour un gradient conjugué, la norme  $(r, z) = \|r\|_{\tilde{\mathcal{G}}^{-1}}$  homogène à une énergie à l'interface est nécessairement calculée, cependant on dispose de tous les vecteurs pour évaluer une autre norme ; pour GMRes, on ne dispose naturellement que de la norme du résidu préconditionné  $\|z\|_2$  qui n'a pas d'interprétation physique simple, toute autre norme requiert le calcul explicite de l'approximation en cours et donc un fort coût CPU.

### 3.3.2 Autres mesures de performance

Outre la convergence, des critères de performance sont reliés à des aspects calculatoires, essentiellement sur des machines parallèles.

- temps de calcul : temps "humain" et temps CPU ;
- temps d'échange : le travail est d'autant mieux parallélisé qu'il nécessite peu d'échange ;
- mémoire utilisée ;
- extensibilité (*scalability*) : pour un problème donné, les performances doivent s'améliorer au moins linéairement en fonction du nombre de sous-structures.

Il est maintenant possible de rassembler quelques résultats sur les décompositions de domaines, plus en relation avec le décomposeur que le solveur.

### 3.3.3 Sur la charge de calcul

Pour réaliser une bonne parallélisation du calcul il convient d'équilibrer la charge de calcul entre les processeurs ce qui se traduit généralement par des sous-structures possédant approximativement le même nombre de degrés de liberté. Ensuite, étant donné que l'on résout un problème posé sur l'interface, le gain est d'autant plus important que l'interface est petite devant la taille des sous-structures.

Ces deux critères conduisent à une première génération de décomposeurs de domaine, cependant ces seules considérations topologiques ne permettent pas de définir une bonne décomposition. Sachant que le problème condensé est résolu par un solveur itératif, il faut autant que possible que la décomposition confère de bonnes propriétés spectrales au problème.

### 3.3.4 *aspect ratio*

On peut montrer très simplement que les sous-structures élancées conduisent à des opérateurs locaux mal conditionnés ce qui se traduit par une mauvaise contribution à

l'opérateur condensé. En effet sur une structure élancé, les modes longitudinaux (traction-compression) sont beaucoup plus rigides que les modes transversaux (flexion) et donc le rapport entre les valeurs propres associées est grand. Un bon décomposeur s'attache donc à conserver aux sous-structures des allures massives (bon *aspect ratio*). Notons cependant qu'une sous-structure massive possède une grande largeur de bande ce qui ralentit les opérations locales (forte connectivité entre les noeuds).

### 3.3.5 Modes rigides

L'existence de sous-structures flottantes entraîne naturellement l'utilisation d'un problème grossier et donc la transmission d'information sur l'ensemble de la structure; tant que le nombre de déplacements de solide rigide ne devient pas excessif (plusieurs centaines), la présence de sous-structures flottantes est donc considérée comme positive. Cependant la qualité du calcul de ces modes rigides est alors un facteur déterminant (surtout pour l'approche duale puisque le problème grossier est essentiel à la méthode alors qu'il n'apporte que de l'optimalité à l'approche primale). Notamment la précision du calcul pour les sous-structures avec un mauvais *aspect ratio* est souvent déficiente. Plusieurs algorithmes sont possibles [29, 36], ceux basés sur des considérations purement géométriques se révèlent très efficaces pour les sous-structures flottantes et à mécanisme (mode rigide interne). Pour les structures où apparaissent des modes à énergie nulle (typiquement flambage, comportement exotique de certains matériaux) une très bonne détection reste un problème ouvert.

### 3.3.6 Hétérogénéités

En présence d'hétérogénéités, la question se pose si l'interface entre les sous-structures doit autant que possible éviter les zones hétérogènes, si elle doit coïncider avec celle entre les matériaux, ou enfin si elle doit la traverser (voir figure 3.1). Grâce à l'emploi d'un *scaling* prenant en compte ces hétérogénéités, on observe que seul le dernier cas peut être critique.

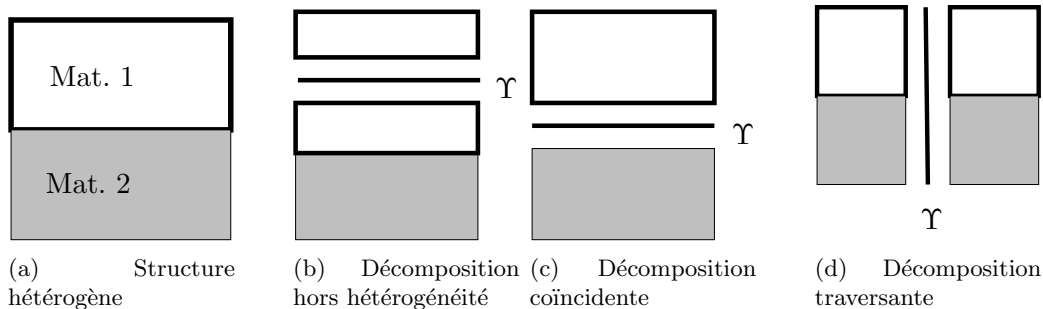


Figure 3.1 — Types d'hétérogénéité

### 3.3.7 Distance et problèmes grossiers

Etant donnée la nature des opérateurs d'interface (assemblage de contributions locales), l'information transite d'un sous-domaine à son voisin. Un critère important est la vitesse à laquelle l'influence d'un effort appliqué en un point d'une sous-structure peut se propager dans toute la structure. Une façon de mesurer la capacité d'une décomposition à propager l'information est d'introduire la notion de distance (au sens de la théorie des graphes). La distance entre deux sous-domaines est le nombre minimal d'interfaces à franchir pour passer

de l'un à l'autre ; la distance d'une décomposition est la plus grande distance entre tout couple de sous-domaines. A priori une décomposition est d'autant meilleure est sa distance est faible.

Cependant, l'utilisation d'un problème grossier bien conçu diminue singulièrement la dépendance vis-à-vis de la distance puisque une information est transmise entre toutes les sous-structures et non uniquement d'un voisin à un autre.

### 3.4 Notation par blocs des méthodes de décomposition de domaine

L'objectif de cette section est de familiariser le lecteur avec un système de notation élégant pour les décompositions de domaine. L'idée est de remplacer l'écriture des assemblages par l'emploi de matrices-blocs.

L'écriture résultante est allégée et rend plus accessibles certaines propriétés, elle sera notamment utile dans le chapitre 4, cependant les aspects relatifs à la répartition des données entre les sous-domaines deviennent moins explicites ce qui explique que nous avons choisi de ne la présenter qu'après l'écriture classique en <sup>(s)</sup>.

Le système fondamental des décompositions de domaine est :

$$\begin{cases} K^{(s)}u^{(s)} = f^{(s)} + \lambda^{(s)} \\ \sum_s A^{(s)}t^{(s)}\lambda^{(s)} = 0 \\ \sum_s \underline{A}^{(s)}t^{(s)}u^{(s)} = 0 \end{cases} \quad (3.24)$$

On introduit les notations suivantes :

$$\begin{aligned} u^\diamond &= \begin{pmatrix} u^{(1)} \\ \vdots \\ u^{(N)} \end{pmatrix} & f^\diamond &= \begin{pmatrix} f^{(1)} \\ \vdots \\ f^{(N)} \end{pmatrix} & \lambda^\diamond &= \begin{pmatrix} \lambda^{(1)} \\ \vdots \\ \lambda^{(N)} \end{pmatrix} \\ K^\diamond &= \begin{pmatrix} K^{(1)} & & \\ & \ddots & \\ & & K^{(N)} \end{pmatrix} & t^\diamond &= \begin{pmatrix} t^{(1)} & & \\ & \ddots & \\ & & t^{(N)} \end{pmatrix} \\ A^\diamond &= (A^{(1)} \quad \dots \quad A^{(N)}) & \underline{A}^\diamond &= (\underline{A}^{(1)} \quad \dots \quad \underline{A}^{(N)}) \end{aligned}$$

le système fondamental s'écrit alors :

$$\begin{cases} K^\diamond u^\diamond = f^\diamond + \lambda^\diamond \\ A^\diamond t^\diamond \lambda^\diamond = 0 \\ \underline{A}^\diamond t^\diamond u^\diamond = \underline{0} \end{cases} \quad (3.25)$$

et la propriété d'orthogonalité des assemblages :

$$\underline{A}^\diamond A^{\diamond T} = 0 \quad (3.26)$$

#### 3.4.1 Approche primale

On condense le système sur l'interface :

$$S_p^\diamond = \begin{pmatrix} S_p^{(1)} & & \\ & \ddots & \\ & & S_p^{(N)} \end{pmatrix} \quad b_p^\diamond = \begin{pmatrix} b_p^{(1)} \\ \vdots \\ b_p^{(N)} \end{pmatrix}$$

le système s'écrit :

$$\begin{cases} S_p^\diamond u_b^\diamond = b_p^\diamond + \lambda_b^\diamond \\ A^\diamond \lambda_b^\diamond = 0 \\ \underline{A}^\diamond u_b^\diamond = \underline{0} \end{cases} \quad (3.27)$$

On introduit alors le déplacement inconnu continu à l'interface  $\mathbf{u}_b$  et on pose  $u_b^\diamond = A^{\diamond T} \mathbf{u}_b$  ce qui permet de vérifier exactement la continuité du déplacement à l'interface. On obtient l'écriture bloc du système primal :

$$\left( A^\diamond S_p^\diamond A^{\diamond T} \right) \mathbf{u}_b = A^\diamond b_p^\diamond \quad (3.28)$$

### 3.4.2 Approche duale

Si on reprend le système (3.25), en introduisant l'inconnue  $\lambda_b$  et en posant  $t^\diamond \lambda^\diamond = \underline{A}^{\diamond T} \lambda_b$  de manière à assurer l'équilibre des interefforts, on a :

$$\begin{cases} K^\diamond u^\diamond = f^\diamond + t^{\diamond T} \underline{A}^{\diamond T} \lambda_b \\ \underline{A}^\diamond t^\diamond u^\diamond = \underline{0} \end{cases} \quad (3.29)$$

L'élimination de  $u^\diamond$  conduit à l'introduction des grandeurs suivantes :

$$K^{\diamond+} = \begin{pmatrix} K^{(1)+} & & \\ & \ddots & \\ & & K^{(N)+} \end{pmatrix} \quad R^\diamond = \begin{pmatrix} R^{(1)} & & \\ & \ddots & \\ & & R^{(N)} \end{pmatrix} \quad \alpha^\diamond = \begin{pmatrix} \alpha^{(1)} \\ \vdots \\ \alpha^{(N)} \end{pmatrix}$$

Ce qui finalement constitue le système dual :

$$\begin{cases} \underline{A}^\diamond t^\diamond K^{\diamond+} t^{\diamond T} \underline{A}^{\diamond T} \lambda_b + \underline{A}^\diamond t^\diamond R^\diamond \alpha^\diamond = -\underline{A}^\diamond t^\diamond K^{\diamond+} f^\diamond \\ R^{\diamond T} t^{\diamond T} \underline{A}^{\diamond T} \lambda_b = -R^{\diamond T} f^\diamond \end{cases} \quad (3.30)$$

Remarquons que l'on peut obtenir le même système à partir de la forme condensée (3.27) en éliminant le déplacement à l'interface (en notant par ailleurs que si  $R^{(s)}$  représente le noyau de  $K^{(s)}$  alors  $t^{(s)} R^{(s)}$  est le noyau de  $S_p^{(s)}$ ) :

$$\begin{cases} \underline{A}^\diamond S_p^{\diamond+} \underline{A}^{\diamond T} \lambda_b + \underline{A}^\diamond t^\diamond R^\diamond \alpha^\diamond = -\underline{A}^\diamond S_p^{\diamond+} b_p^\diamond \\ R^{\diamond T} t^{\diamond T} \underline{A}^{\diamond T} \lambda_b = -R^{\diamond T} t^{\diamond T} b_p^\diamond \end{cases} \quad (3.31)$$

### 3.4.3 Préconditionneurs

Dans [53], une écriture très synthétique du preconditionneur avec *scaling* a été proposée pour l'approche duale. Elle a été ensuite étendue à l'approche primale.

$$\tilde{S}_p^{\diamond-1} = \left( A^\diamond M^\diamond A^{\diamond T} \right)^{-1} A^\diamond M^\diamond S_p^{\diamond+} M^\diamond A^{\diamond T} \left( A^\diamond M^\diamond A^{\diamond T} \right)^{-1} \quad (3.32)$$

$$\tilde{S}_d^{\diamond-1} = \left( \underline{A}^\diamond M^{\diamond-1} \underline{A}^{\diamond T} \right)^+ \underline{A}^\diamond M^{\diamond-1} S_p^{\diamond+} M^{\diamond-1} \underline{A}^{\diamond T} \left( \underline{A}^\diamond M^{\diamond-1} \underline{A}^{\diamond T} \right)^+ \quad (3.33)$$

Le paramètre symétrique défini positif  $M^\diamond$  permet de représenter les différents *scalings* :  $M^\diamond = I$  pour le *scaling* topologique,  $M^\diamond = \text{diag}(K_{bb}^\diamond)$  pour le *scaling* rigidité. La relation de complémentarité des *scalings* s'écrit [51] :

$$M^\diamond A^{\diamond T} \left( A^\diamond M^\diamond A^{\diamond T} \right)^{-1} A^\diamond + \underline{A}^{\diamond T} \left( \underline{A}^\diamond M^{\diamond-1} \underline{A}^{\diamond T} \right)^{-1} \underline{A}^\diamond M^{\diamond-1} = I^\diamond \quad (3.34)$$

Ce résultat se démontre en constatant que l'égalité est trivialement vérifiée quand on multiplie l'expression à gauche par  $A^{\diamond T}$  et  $\underline{A}^{\diamond T} M^{\diamond -1}$ , et en remarquant, sous les hypothèses formulées sur  $M^{\diamond}$ , que  $\ker(A^{\diamond T})$  et  $\ker(\underline{A}^{\diamond T} M^{\diamond -1})$  sont supplémentaires.

On voit donc l'écriture bloc un peu spéciale des *scalings* (la pseudo-inverse est due à l'existence des redondances) :

$$\begin{aligned} \sum_s W^{(s)} A^{(s)} &\sim (A^{\diamond} M^{\diamond} A^{\diamond T})^{-1} A^{\diamond} M^{\diamond} \\ \sum_s \underline{W}^{(s)} \underline{A}^{(s)} &\sim (\underline{A}^{\diamond} M^{\diamond -1} \underline{A}^{\diamond T})^{+} \underline{A}^{\diamond} M^{\diamond -1} \end{aligned} \quad (3.35)$$

De fait le *scaling* ne peut apparaître indépendamment de l'assemblage et on définit ainsi des opérateurs d'assemblage pondérés.



## Deuxième partie

# Etude et amélioration des méthodes classiques de décomposition de domaine





Les méthodes classiques primale et duale de décomposition de domaine ont été décrites dans la partie précédente en faisant apparaître autant que possible leurs nombreuses connexions. Les résultats théoriques et les performances annoncées dans la littérature laissent supposer plusieurs propriétés :

- les performances des approches primale et duale sont sensées être pratiquement semblables,
- les performances des approches primale et duale ne sont pas sensées être significativement détériorées par la présence d'hétérogénéités.

Cette partie est dédiée à l'étude de classes de problèmes où ces propriétés ne sont pas vérifiées, et à la définition d'amélioration des méthodes classiques leur permettant de palier ces difficultés.

Le premier chapitre de cette partie est né de la volonté de confronter l'efficacité des méthodes de décomposition de domaine classiques sur des problèmes assez fortement hétérogènes mais où le comportement des matériaux reste "simple" (élasticité linéaire). Des différences significatives ont alors été observées en défaveur de l'approche duale, ce chapitre les explique et montre comment les résoudre.

Le second chapitre introduit des comportements plus complexes au sein des matériaux : des phénomènes incompressibles sont envisagés via l'étude des élastomères. L'incompressibilité est traitée par introduction d'un multiplicateur de Lagrange qui peut être interprété comme une pression hydrostatique. Un tel traitement conduit donc à l'étude d'un problème multichamps (déplacement, pression), et une question importante est alors comment traiter la pression. Ce chapitre se focalise sur les cas où le problème conduit à une pression discontinue aux interfaces, nous montrons alors comment une modification des préconditionneurs existants peut s'imposer pour rétablir les bonnes propriétés numériques des méthodes classiques.



---

# Comparaison des approches primale et duale de décomposition de domaine

Cette étude est issue d'une collaboration avec le professeur Daniel Rixen de l'université de DELFT (PAYS-BAS). Nous comparons les performances des approches primale (BDD) et duale (FETI) de décomposition de domaine sur plusieurs problèmes pathologiques. Comme il a été vu dans la première partie ces deux méthodes sont théoriquement très proches, elles sont par ailleurs numériquement équivalentes sur un grand nombre d'applications. Cependant, sur certains exemples simples un moins bon comportement de l'approche duale est observé. Après avoir lié ces mauvaises performances à une déficience de l'initialisation de l'approche duale et interprétée cette initialisation comme une répartition des efforts à l'interface entre les sous-structures, nous proposons une nouvelle initialisation de l'approche duale permettant de rétablir l'équivalence entre les approches primales et duales sur un plus grand nombre de problèmes.

Ces travaux ont fait l'objet d'une publication [51] et d'une communication [50].

Considérons le problème figure 4.1 d'un cube fortement hétérogène ( $E_1/E_2 = 10^5$ ) encastré sur une face et soumis à une pression homogène sur la face opposée. Le cube est décomposé  $3 \times 3 \times 3$  sous-structures cubiques, on utilise des éléments-finis hexaédriques  $Q_2$  (27 noeuds). Le problème possède 21000 degrés de liberté dont 6000 appartiennent à l'interface. La figure 4.2 montre l'évolution du résidu global pour les approches primale et duale équipées avec ce que la littérature considère comme leur meilleur préconditionneur et problème grossier (BDD = approche primale avec préconditionneur Neumann-Neumann et projecteur modes rigides, FETI = approche duale avec préconditionneur et projecteur de Dirichlet, voir chapitre 3). On observe que bien que les convergences asymptotiques soient les mêmes, la convergence de l'approche duale est moins monotone et la norme du résidu initial est beaucoup plus grande.

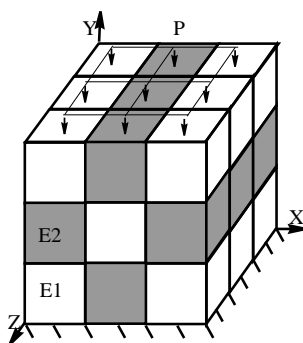


Figure 4.1 — Cube hétérogène décomposé

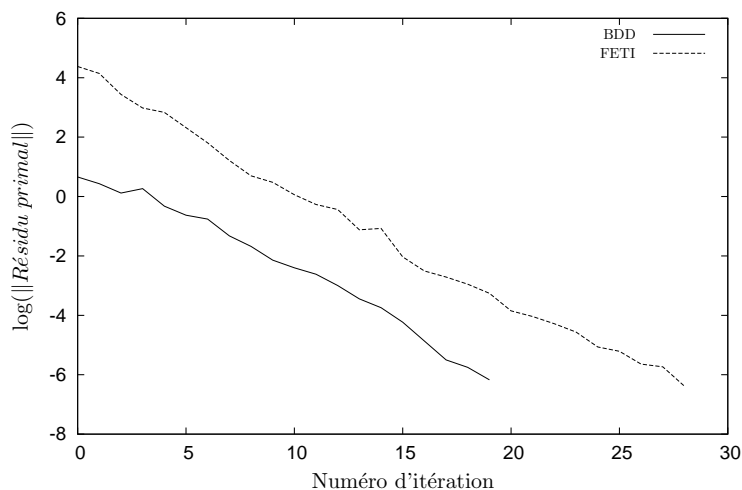


Figure 4.2 — Convergence des approches primale et duale (pb. fig. 4.1)

Ces résultats montrent que certains détails comme le choix de l'estimation initiale conduisent à des différences significatives entre les approches primale et duale.

## 4.1 Interprétation de l'initialisation dans l'approche duale

Reprenons le système de l'approche duale (en notation bloc) (1.34,1.25b).

$$\begin{cases} K^\diamond u^\diamond = f^\diamond + t^{\diamond T} \underline{A}^{\diamond T} \underline{\lambda}_b \\ \underline{A}^\diamond t^\diamond u^\diamond = \underline{0} \end{cases} \quad (4.1)$$

Supposons que l'on initialise ce problème avec un intereffort  $\underline{\lambda}_{b0}$ , on a

$$K^\diamond u^\diamond = f^\diamond + t^{\diamond T} \underline{A}^{\diamond T} \underline{\lambda}_b \quad (4.2)$$

$$\text{soit } \underline{\lambda}_b = \tilde{\underline{\lambda}}_b + \underline{\lambda}_{b0}$$

$$\begin{aligned} K^\diamond u^\diamond &= f^\diamond + t^{\diamond T} \underline{A}^{\diamond T} \tilde{\underline{\lambda}}_b + t^{\diamond T} \underline{A}^{\diamond T} \underline{\lambda}_{b0} \\ &= \tilde{f}^\diamond + t^{\diamond T} \underline{A}^{\diamond T} \tilde{\underline{\lambda}}_b \end{aligned} \quad (4.3)$$

$$\text{avec } \tilde{f}^\diamond = f^\diamond + t^{\diamond T} \underline{A}^{\diamond T} \underline{\lambda}_{b0} \quad (4.4)$$

On voit donc que l'initialisation  $\underline{\lambda}_{b0}$  s'interprète comme une modification  $t^{\diamond T} \underline{A}^{\diamond T} \underline{\lambda}_{b0}$  de l'intereffort entre les sous-structures : les problèmes locaux sont définis à un champ d'effort d'interface réparti près. L'essentiel est que les efforts locaux sur l'interface s'assemblent en un même champ (le seul champ qui ait un réel sens mécanique).

$$A^\diamond t^\diamond \tilde{f}^\diamond = A^\diamond t^\diamond f^\diamond = f_b \quad \text{effort d'interface "global"} \quad (4.5)$$

$$\text{car } A^\diamond t^\diamond t^{\diamond T} \underline{A}^{\diamond T} \underline{\lambda}_{b0} = A^\diamond \underline{A}^{\diamond T} \underline{\lambda}_{b0} = 0 \quad (4.6)$$

Une initialisation non nulle s'interprète donc comme une répartition de l'effort d'interface global  $f_b$ , les deux sections suivantes donnent deux stratégies pour cette répartition.

## 4.2 Répartition des efforts dans l'approche duale

### 4.2.1 Répartition classique des efforts

Bien qu'elle ne soit que rarement interprétée comme une initialisation, une répartition des efforts est couramment réalisée, basée sur les différences de rigidité entre les sous-structures voisines (idée très analogue au *scaling* rigidité), le but est de faire transiter la plus grande partie des efforts sur la sous-structure la plus rigide, ce qui correspond au flux naturel d'effort dans la structure.

L'effort global sur l'interface  $f_b$  est réparti sur la base du *scaling* rigidité ( $M = \text{diag}(K_{bb}^\diamond)$ ), ce qui permet d'obtenir un effort d'interface local modifié  $\tilde{f}_b^\diamond$ .

$$\tilde{f}_b^\diamond = M^\diamond A^\diamond \left( A^\diamond M^\diamond A^{\diamond T} \right)^{-1} f_b \quad (4.7)$$

L'effort complet réparti  $\tilde{f}^\diamond$  est constitué de  $f^\diamond$  à l'intérieur de la sous-structure ( $(I - t^{\diamond T} t^\diamond) f^\diamond$ ) et de l'effort réparti sur sa frontière ( $t^{\diamond T} \tilde{f}_b^\diamond$ ).

$$\tilde{f}^\diamond = (I - t^{\diamond T} t^\diamond) f^\diamond + t^{\diamond T} \tilde{f}_b^\diamond \quad (4.8)$$

Etant donnée la relation de complémentarité entre les *scalings* (3.34), on a

$$\tilde{f}^\diamond = f^\diamond - t^{\diamond T} \underline{A}^{\diamond T} \left( \underline{A}^\diamond M^{\diamond -1} \underline{A}^{\diamond T} \right)^+ \underline{A}^\diamond M^{\diamond -1} t^\diamond f^\diamond \quad (4.9)$$

Comme nous le montrerons sur les validations, cette répartition ne modifie pratiquement pas les performances de l'approche duale de décomposition de domaine. Nous avons donc proposé une nouvelle répartition des efforts sur l'interface basée sur la formulation de l'approche duale à partir des systèmes condensés.

### 4.2.2 Répartition des efforts condensés

Lorsque l'on réécrit le système dual après condensation :

$$\begin{cases} S_p^\diamond u_b^\diamond = b_p^\diamond + \underline{A}^{\diamond T} \underline{\lambda}_b \\ \underline{A}^\diamond u_b^\diamond = 0 \end{cases} \quad (4.10)$$

on voit que l'on peut répartir les efforts condensés à l'interface  $b_p^\diamond$  du moment que l'effort condensé "global" reste identique. L'effort condensé assemblé s'écrit :

$$b_p = A^\diamond b_p^\diamond \quad (4.11)$$

On le répartit selon les rigidités entre les sous-structures :

$$\tilde{b}_p^\diamond = M^\diamond A^\diamond \left( A^\diamond M^\diamond A^{\diamond T} \right)^{-1} b_p \quad (4.12)$$

Ce qui donne à l'aide de la complémentarité des *scalings* :

$$\tilde{b}_p^\diamond = b_p^\diamond - \underline{A}^{\diamond T} \left( \underline{A}^\diamond M^{\diamond-1} \underline{A}^{\diamond T} \right)^+ \underline{A}^\diamond M^{\diamond-1} b_p^\diamond \quad (4.13)$$

Pour mieux comparer avec la répartition précédente et bien voir la différence entre ces deux stratégies, on donne l'effort non-condensé associé, à comparer avec (4.9) :

$$\tilde{f}^\diamond = f^\diamond - t^{\diamond T} \underline{A}^{\diamond T} \left( \underline{A}^\diamond M^{\diamond-1} \underline{A}^{\diamond T} \right)^+ \underline{A}^\diamond M^{\diamond-1} b_p^\diamond \quad (4.14)$$

## 4.3 Amélioration de l'initialisation de l'approche duale

Dans cette section nous montrons que l'initialisation équivalente à la répartition des efforts condensés peut être retrouvée à partir des considérations qui conduisent traditionnellement à l'initialisation de l'approche primale.

L'initialisation de l'approche primale est réalisée en supposant le champ de déplacement de l'interface nul sur le problème condensé. En formulant à partir de (4.10) la même hypothèse, on déduit que l'équilibre des sous-domaines s'écrit :

$$\underline{A}^{\diamond T} \underline{\lambda}_{b_0} + b_p^\diamond \simeq 0 \quad (4.15)$$

Ce qui n'est certainement pas la solution du problème (sauf si un déplacement nul à l'interface est solution du problème). On décompose alors les efforts d'interface en une contribution équilibrée (globalement nulle) et son reste  $\varrho$  :

$$\begin{cases} b_p^\diamond = \underline{A}^{\diamond T} \gamma + \varrho \\ \gamma = \left( \underline{A}^\diamond D^\diamond \underline{A}^{\diamond T} \right)^+ \underline{A}^\diamond D^\diamond b_p^\diamond \end{cases} \quad (4.16)$$

$D^\diamond$  est une matrice symétrique définie positive, le reste  $\varrho$  est alors orthogonal à  $\text{Im}(D^\diamond \underline{A}^{\diamond T})$ . Si on choisit d'initialiser le calcul par :

$$\lambda_{b00} = -\gamma \quad (4.17)$$

alors le résidu initial  $\underline{A}^{\diamond T} \lambda_{b00} + b_p^\diamond = -\varrho$  est minimal au sens de la norme associée à  $D^\diamond$ . Si  $D^\diamond = \text{diag}(K_{bb}^\diamond)^{-1}$  on retrouve la répartition des efforts condensés selon la rigidité des sous-structures ;  $\text{diag}(K_{bb}^\diamond)$  étant une approximation de  $S_p^\diamond$  on voit que la norme considérée s'interprète comme une énergie.

L'initialisation par répartition des efforts condensés n'est *a priori* pas compatible avec les déplacements de solide rigide, il faut la rendre compatible en calculant (3.13) :

$$\lambda_{b0} = P \lambda_{b00} - \underline{QG} (\underline{G}^T \underline{QG})^{-1} e \quad (4.18)$$

Remarquons que si le choix  $D^\diamond = S_p^{\diamond+}$  n'était pas rédhibitoire en terme de coût de calcul l'utilisation de la nouvelle initialisation associée au projecteur de Dirichlet permettrait d'obtenir la convergence dès l'initialisation.

Pour le choix recommandé  $D^\diamond = \text{diag}(K_{bb}^\diamond)^{-1}$  on voit que la nouvelle stratégie ne requiert que le calcul des efforts condensés ce qui correspond à la résolution d'un problème de Dirichlet c'est à dire une étape de préconditionnement ou encore une demi-étape de FETI, soit un faible coût de calcul. Naturellement dans le cas où un préconditionneur allégé est utilisé, l'opérateur de Dirichlet n'est pas disponible et il convient alors de réaliser une répartition classique des efforts.

## 4.4 Validation

Pour valider les performances de la nouvelle initialisation des multiplicateurs de Lagrange, nous considérons différents problèmes résolus par les approches primale (BDD) et duale (FETI). Dans tous les cas, la convergence est contrôlée par le résidu global ( $\varepsilon = 10^{-6}$ ). Tous les résultats sont obtenus avec les préconditionneurs optimaux équipés avec les *scalings* rigidité. L'approche duale est testée avec ses différents projecteurs, le projecteur Dirichlet est noté  $P(\tilde{S}_d^{-1})$ , le projecteur *superlumped*  $P(\tilde{S}_d^{-1}_{SL})$  et le projecteur identité  $P(I)$ .

### 4.4.1 Cube hétérogène (échiquier)

Reprenons tout d'abord les différentes stratégies sur le problème décrit en introduction (fig. 4.1). La figure 4.3 présente la convergence au cours des itérations du gradient conjugué pour les approches primale et duale classiques. L'approche primale donne clairement de meilleurs résultats. De part la forte hétérogénéité le long des interfaces, le projecteur  $P(I)$  n'est manifestement pas adapté. On observe que, bien que le projecteur Dirichlet conduise à un meilleur taux de convergence, le projecteur *superlumped* permet d'obtenir un résidu initial significativement inférieur et ainsi atteint la convergence plus vite (résidu initial 1000 fois plus petit mais allure moins régulière de la convergence).

La figure 4.4 présente l'historique de convergence pour l'approche primale et pour la méthode FETI équipée du projecteur Dirichlet et de la nouvelle initialisation. Non seulement la nouvelle initialisation fournit un meilleur résidu initial mais aussi conduit à un taux de convergence équivalent à celui de la répartition classique. On obtient alors une courbe de



convergence très similaire à celle de l'approche primale (avec une allure malgré tout un peu moins régulière).

Le tableau 4.1 résume les performances des différentes stratégies possibles en comparant le nombre d'itérations à convergence et la norme du résidu initial. On remarque d'abord que la répartition classique n'est pratiquement d'aucun intérêt par rapport à l'absence de répartition. La nouvelle initialisation apporte une sérieuse amélioration à l'approche duale avec projecteur Dirichlet (35% d'itérations en moins), mais n'affecte que très peu la convergence quand les autres projecteurs sont utilisés.

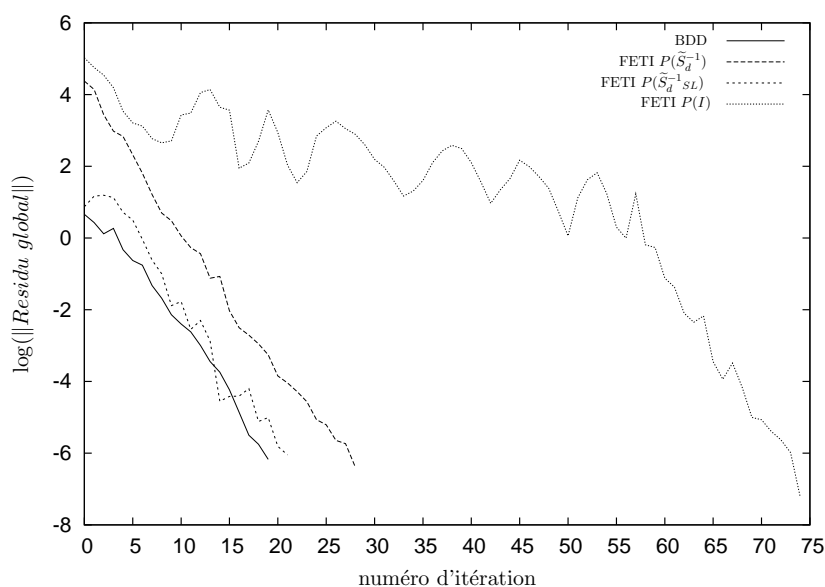


Figure 4.3 — Cube hétérogène (échiquier, fig. 4.1) : convergence pour les répartitions classiques

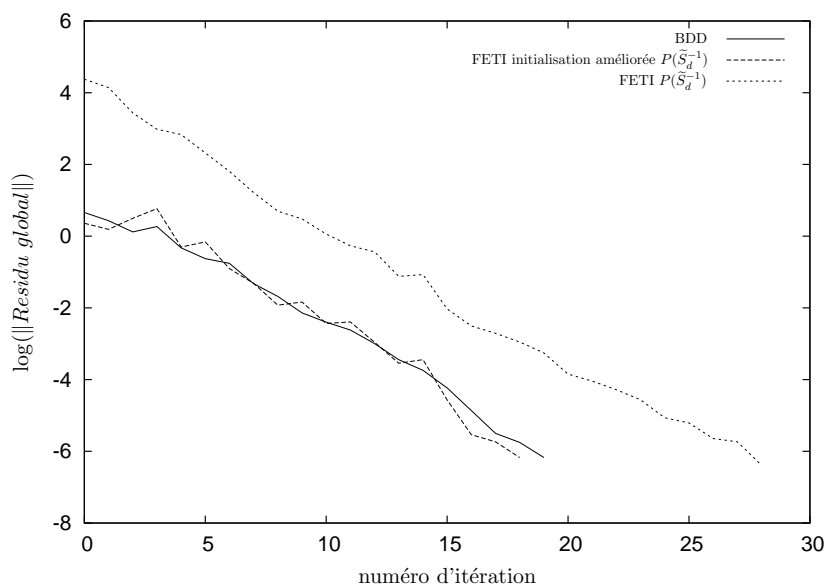


Figure 4.4 — Cube hétérogène (échiquier, fig. 4.1) : convergence avec la nouvelle initialisation

Approche		Nombre d'itérations	Résidu initial (Log)
BDD		19	0.659
FETI Dirichlet $P(\tilde{S}_d^{-1})$	Pas de répartition	28	4.428
	Répartition classique	28	4.377
	Nvlle. initialisation	18	0.359
FETI Dirichlet $P(\tilde{S}_d^{-1}{}_{SL})$	Pas de répartition	21	0.873
	Répartition classique	21	0.872
	Nvlle initialisation	20	0.868
FETI Dirichlet $P(I)$	Pas de répartition	74	5.029
	Répartition classique	74	5.026
	Nvlle initialisation	73	5.016

Tableau 4.1 — Performances des répartitions pour le cube fig. 4.1

#### 4.4.2 Cubes hétérogènes (lamifiés) droit et incliné

Cette validation porte sur deux nouvelles configurations du cube permettant d'évaluer les influences de la géométrie et de la distribution des hétérogénéités. Les structures décrites figures 4.5 et 4.6 sont similaires à celle de la figure 4.1 avec une distribution matérielle différente ; par ailleurs pour le problème figure 4.6, le cube a été incliné de  $60^\circ$ . le tableau 4.2 donne le nombre d'itérations avec l'approche primale et les différentes versions de l'approche duale.

Pour le cube droit figure 4.5, l'utilisation de la nouvelle initialisation n'apporte qu'une légère amélioration. De fait, l'approche duale avec projecteur Dirichlet sans répartition est déjà pratiquement équivalente à l'approche primale ; la nouvelle initialisation ne peut alors apporter de gain significatif.

Pour le cube incliné figure 4.6, en raison de la distorsion, l'approche primale se révèle plus efficace que l'approche duale classique. On voit alors que l'utilisation de la nouvelle initialisation permet de retrouver avec l'approche duale des performances très proches de l'approche primale.

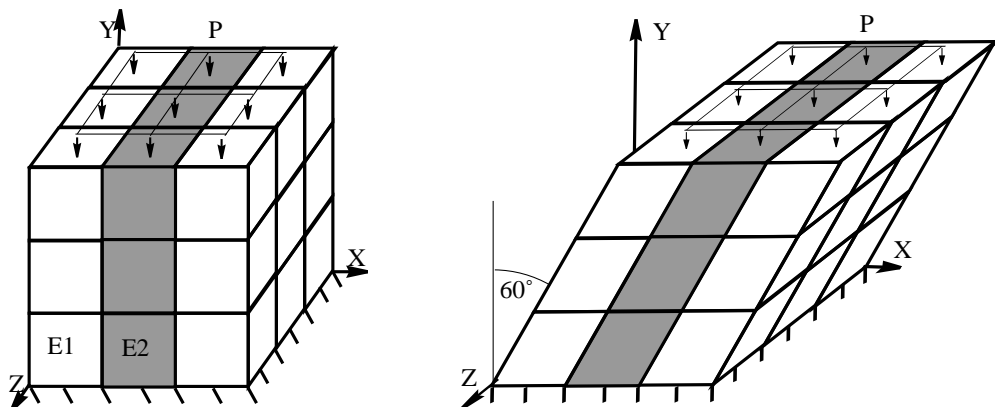


Figure 4.5 — Cube avec feuillets hétérogènes Figure 4.6 — Cube incliné avec feuillets hétérogènes

Approche		Cube, fig. 4.5 Nb. it.	Cube incliné, fig. 4.6 Nb.it.
BDD		19	73
FETI Dirichlet $P(\tilde{S}_d^{-1})$	Pas de répart.	20	85
	Répart. classique	21	85
	Nvlle init.	19	73
FETI Dirichlet $P(\tilde{S}_d^{-1}{}_{SL})$	Pas de répart.	22	88
	Répart. classique	22	88
	Nvlle init.	22	85
FETI Dirichlet $P(I)$	Pas de répart.	92	153
	Répart. classique	92	154
	Nvlle init.	90	159

Tableau 4.2 — Performances des répartitions sur les problèmes fig. 4.5 et 4.6

#### 4.4.3 Flexion non-linéaire d'une poutre composite

Pour le dernier exemple, nous analysons un problème non-linéaire résolu par une séquence de problèmes linéaires (ce type de problème est plus longuement étudié dans la partie IV). La structure représentée figure 4.7 est une poutre de section carré et de rapport d'élanement 9. Elle est composée de bandes longitudinales de métal et d'élastomère. Le chargement est une pression imposée sur un coté de la poutre. De part la présence de l'élastomère, la structure subit de grandes déformations. Le modèle retenu pour le métal est décrit par un potentiel élastique de Saint-Venant–Kirchoff (module d'Young  $E = 20000$  MPa, coefficient de Poisson  $\nu = 0.3$ ). Pour l'élastomère, il s'agit d'un modèle néo-Hookien quasi-incompressible (module de cisaillement  $G = 2.0$  MPa, module de compressibilité  $K = 2000$  MPa). La structure est décomposée en 27 sous-domaines homogènes parallélépipédiques et chaque sous-domaine est maillé en  $2 \times 2 \times 18$  éléments cubiques. Pour prendre en compte la quasi-incompressibilité de l'élastomère, une formulation mixte en déplacement/pression est retenue. L'élément utilisé est un hexaèdre  $Q_2 - P_1$  avec 27 noeuds de déplacement et 4 noeuds internes de pression. Le chapitre 5 donne des détails importants sur l'utilisation de tels éléments.

Le problème complet possède 55330 degrés de liberté dont 16400 appartiennent à l'interface. Le comportement de l'élastomère rend le problème fortement non-linéaire ; les itérations de Newton-Raphson sont réalisées avec mise à jour de la matrice tangente à chaque itération. Le chargement en pression est de 5 bars. Le critère d'arrêt est à  $10^{-3}$  sur le critère global pour la résolution des systèmes linéaires. Pour l'algorithme Newton-Raphson, la tolérance sur le résidu non-linéaire relatif est fixée à  $10^{-4}$  ce qui nécessite la résolution de 5 systèmes linéaires.

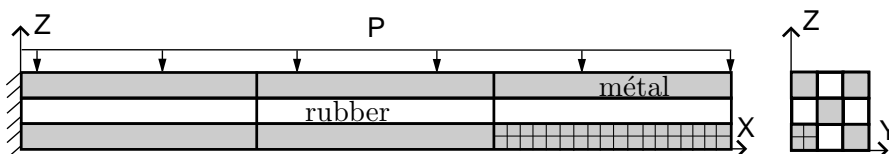


Figure 4.7 — Poutre composite à  $3 \times 3 \times 3$  sous-structures

La figure 4.8 indique le nombre d'itérations requis pour résoudre chaque système linéaire par l'approche FETI classique, par l'approche FETI munie de la nouvelle initialisation et

par l'approche primale (BDD). Les projecteurs Dirichlet et *superlumped* sont testés. L'augmentation significative du nombre d'itérations à convergence entre le premier système et les suivants peut s'expliquer par la perte de positivité des opérateurs tangents, due principalement à l'incompressibilité (et non au phénomène de perte de modes rigides décrit dans la partie IV). De fait le gradient conjugué (avec réorthogonalisation totale) reste applicable sur des problèmes non-positifs mais la convergence est alors significativement ralentie [68].

En ce qui concerne le nombre total d'itérations, FETI classique nécessite 10% d'itérations en plus que BDD alors que FETI avec nouvelle initialisation converge en légèrement moins d'itérations que BDD. On observe que l'initialisation améliore les performances de FETI pour tous les projecteurs étudiés, et dans ce cas précis le *superlumped* conduit aux meilleures performances.

Nos expériences numériques indiquent que pour une grande classe de problèmes tels que celui présenté ici, la nouvelle initialisation conduit à de légers mais non négligeables gains en nombre d'itérations et en temps de calcul. Un autre effet bénéfique de la nouvelle initialisation est de démarrer les calculs avec un résidu initial diminué de plusieurs degrés de magnitude ; la stagnation du résidu au cours des itérations de FETI qui arrive fréquemment pour des problèmes très fortement non-linéaires (chargement plus important), est suffisamment retardée pour qu'en pratique le redémarrage du calcul ne soit pas nécessaire.

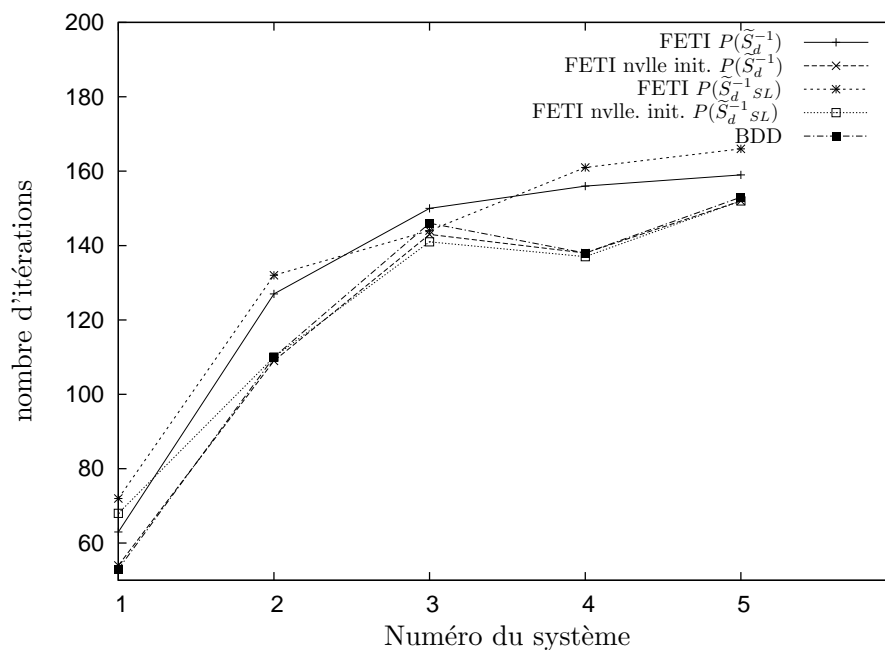


Figure 4.8 — Performances pour la flexion de la poutre composite

## 4.5 Conclusions

Pour certains problèmes de structure particuliers comme ceux présentant de fortes hétérogénéités et distorsions géométriques, la méthode duale peut conduire à une mauvaise convergence comparée à l'approche primale pourtant conceptuellement similaire. Dans ces cas, les mauvaises performances de FETI peuvent être ramenées au résidu initial trop élevé. Or le résidu initial est intimement lié à façon dont les efforts sont répartis entre les sous-domaines sur l'interface.

Nous avons proposé une nouvelle stratégie pour répartir les efforts entre les sous-domaines qui consiste à distribuer selon le *scaling* rigidité les efforts condensés à l'interface. Cela conduit à construire une initialisation plus efficace pour les interefforts, basée sur la condensation des efforts soit seulement le coût d'une demi-itération de FETI.

Les exemples numériques proposés indiquent que pour les problèmes où l'approche primale surclasse l'approche duale, la nouvelle initialisation donne une meilleure estimation des interefforts et conduit à un résidu initial équivalent à celui de l'approche primale. FETI converge alors de manière très similaire à BDD.

La nouvelle initialisation ne détériore jamais la convergence de FETI, et apporte des améliorations sensibles dans certains cas pathologiques. Nous recommandons donc de l'employer par défaut dans le solveur FETI.

Avec l'initialisation proposée, FETI et BDD permettent d'obtenir les mêmes performances en terme de convergence et de coût de calcul pour des problèmes complexes où le projecteur Dirichlet est requis. Sur des problèmes où les préconditionneurs et projecteurs allégés de FETI peuvent être employés sans trop détériorer les performances, FETI est souvent plus efficace en terme de coût de calcul total.

---

# Prise en compte de la (quasi)-incompressibilité

L'incompressibilité (ou la quasi-incompressibilité) est un phénomène que l'on rencontre dans l'étude de nombreux phénomènes physiques, comme l'étude des écoulements de Stokes ou celle des élastomères. Nous nous intéressons plus particulièrement à ces derniers et de manière à illustrer ce chapitre, nous avons choisi d'appliquer nos méthodes à une butée flexible SNECMA-moteurs-de-fusées dont nous présentons la géométrie et les caractéristiques mécaniques (notamment la présence d'instabilités) dans l'annexe B.

Avant de rentrer dans les détails de la formulation employée pour décrire le comportement de l'élastomère et de manière à comprendre le domaine d'application de ce chapitre contentons-nous de dire que la prise en compte de la quasi-incompressibilité de l'élastomère est réalisée par l'emploi d'une formulation mixte déplacement/pression. Dès lors d'un point de vue décomposition de domaine, se pose la question de l'intégration de la pression sur le problème condensé à l'interface. Notre opinion est que l'introduction de la pression comme inconnue d'interface requiert une attention particulière qui conduit au développement d'une nouvelle méthode de décomposition de domaine appelée approche hybride qui fait l'objet de la partie III de cette thèse.

Dans ce chapitre nous nous concentrons sur les problèmes où la pression n'est pas une inconnue d'interface, ce qui revient à formuler l'hypothèse que la pression n'est pas continue à l'interface entre les sous-domaines. Cette hypothèse de pression discontinue correspond à plusieurs cas :

- l'interface sépare une pièce en caoutchouc d'une pièce en acier ;
- l'interface sépare deux pièces différentes de caoutchouc ;
- la formulation éléments-finis est discontinue d'un élément à l'autre ce qui implique naturellement la discontinuité sur toute interface ;
- l'interface est située au sein d'une pièce de caoutchouc, pour laquelle on tolère une éventuelle erreur sur la continuité de la pression (comme nous le verrons sur l'exemple de la butée flexible, cette approximation conduit à des résultats mécaniques largement satisfaisants).

Dans ce chapitre nous montrons comment une amélioration des préconditionneurs existants pour les approches classiques permet d'obtenir une stratégie de calcul pertinente pour de tels problèmes.

Ce travail a fait l'objet d'une publication [49].

## 5.1 Étude de l'élastomère

Pour une étude détaillée des matériaux hyperélastiques compressibles et incompressibles ainsi que des méthodes numériques associées, le lecteur peut consulter [67, 74, 98]. L'élastomère considéré est un matériaux hyperélastique quasi-incompressible. De manière à décrire les grandes déformations, la cinématique est représentée par le tenseur gradient  $\overline{\overline{F}}$ . Soit  $\overline{\overline{C}}$  le tenseur de Cauchy-Green droit (tenseur des dilatations)  $\overline{\overline{C}} = \overline{\overline{F}}^T \overline{\overline{F}}$  et  $\overline{\overline{E}}$  le tenseur de Green-Lagrange. Pour un matériau hyperélastique isotrope, le potentiel élastique  $W$  vérifie :

$$W \left( x, I_1(\overline{\overline{C}}), I_2(\overline{\overline{C}}), I_3(\overline{\overline{C}}) \right) \text{ avec } \begin{cases} I_1(\overline{\overline{C}}) = \text{trace}(\overline{\overline{C}}) \\ I_2(\overline{\overline{C}}) = \frac{1}{2} \left( \text{trace}(\overline{\overline{C}})^2 - \text{trace}(\overline{\overline{C}}^2) \right) \\ I_3(\overline{\overline{C}}) = \det(\overline{\overline{C}}) \end{cases} \quad (5.1)$$

Le premier tenseur des contraintes de Piola-Kirchhoff  $\overline{\overline{T}}$  est alors donné par :

$$\overline{\overline{T}} = 2\overline{\overline{F}} \frac{\partial W}{\partial \overline{\overline{C}}} \quad (5.2)$$

On trouve de nombreux potentiels hyperélastiques dans la littérature, citons notamment :

- Saint-Venant-Kirchhoff [67, 105]  $W = \frac{\lambda}{2} \text{trace}(E)^2 + \mu \text{trace}(E^2)$  ;
  - Mooney-Rivlin [80]  $W = \frac{C_{10}}{2} (I_1 - 3) + \frac{C_{01}}{2} (I_2 - 3)$  ;
  - Ciarlet-Geymonat [13]  $W = C_1(I_1 - 3) + C_2(I_2 - 3) + a(I_3 - 1) - (C_1 + 2C_2 + a) \ln(I_3)$  ;
- où  $\lambda$  et  $\mu$  sont les coefficients de Lamé, avec  $C_{10}$ ,  $C_{01}$ ,  $C_1$ ,  $C_2$  et  $a$ , ils caractérisent le matériau. Pour des potentiels plus récents, le lecteur pourra se référer à [56, 10].

Pour prendre en compte l'incompressibilité ( $J = \det(\overline{\overline{F}}) = \sqrt{I_3(\overline{\overline{C}})} = 1$ ) ou la quasi-incompressibilité de l'élastomère, plusieurs stratégies sont possibles :

- pénalisation : la variation de volume  $J$  apparaît pondérée par un fort coefficient de compressibilité  $K$  dans le potentiel élastique ;
- invariants modifiés (invariants de Penn) : on utilise une loi hyperélastique classique en remplaçant  $I_1$  et  $I_2$  par des invariants perturbés :  $\tilde{I}_1 = J^{-1/3} I_1$ ,  $\tilde{I}_2 = J^{-1/3} I_2$  ;
- formulation lagrangienne : la contrainte  $J = 1$  est dualisée par l'introduction d'un multiplicateur de Lagrange  $p$  qui s'interprète comme une pression hydrostatique.

On retient ici une formulation quasi-incompressible basée sur une approche lagrangienne avec pénalisation, le lagrangien dont on recherche le point selle s'écrit :

$$\mathcal{L}(u, p) = \int_{\Omega} W d\Omega + \int_{\Omega} p \left( h(J) - \frac{1}{2K} p \right) d\Omega - \int_{\Omega} g u d\Omega - \int_{\partial_f \Omega} f u dS \quad (5.3)$$

Pour la fonction  $h(J)$  de nombreux modèles sont possibles, parmi eux :

- modèle linéaire :  $h(J) = J - 1$
- modèle logarithmique :  $h(J) = \ln(J)$

La formulation variationnelle du problème s'écrit :

$$\left\{ \begin{array}{l} \text{Trouver } (u, p) \in \mathcal{U} \times \mathcal{P} / \forall (v, q) \in \mathcal{U} \times \mathcal{P} \\ \int_{\Omega} \frac{\partial W}{\partial \overline{\overline{F}}} (I + \overline{\overline{\nabla}}(\overline{u})) : \overline{\overline{\nabla}}(\overline{v} - \overline{u}) d\Omega + \int_{\Omega} p h'(J) \frac{\partial J}{\partial \overline{\overline{F}}} : \overline{\overline{\nabla}}(\overline{v} - \overline{u}) d\Omega = \\ \int_{\Omega} \overline{g} \cdot (\overline{v} - \overline{u}) d\Omega + \int_{\partial_f \Omega} \overline{f} \cdot (\overline{v} - \overline{u}) dS \\ \int_{\Omega} (h(J) - \frac{1}{K} p) q d\Omega = 0 \end{array} \right. \quad (5.4)$$

où  $\mathcal{U}$  et  $\mathcal{P}$  sont les espaces de déplacement et de pression admissibles, typiquement :

$$\begin{aligned}\mathcal{U} &= \{v \in H^1(\Omega), v = u_0 \text{ sur } \partial_u \Omega\} \\ \mathcal{P} &= L^2(\Omega)\end{aligned}\quad (5.5)$$

La recherche de solution approchée du problème variationnel est menée de manière classique par une approche éléments-finis, il convient de définir des sous-espaces de dimension finie  $\mathcal{U}_h$  et  $\mathcal{P}_h$  de  $\mathcal{U}$  et  $\mathcal{P}$ . Notons que la construction d'un espace éléments-finis mixte est soumise à la condition de compatibilité de Ladyzenskaya-Babuska-Brezzi (LBB [8]) restreignant théoriquement le choix des espaces (dans [95] diverses méthodes de construction d'éléments LBB compatibles sont proposées).

Quoi qu'il en soit citons quelques stratégies classiques d'approximation :

- hexaèdre  $Q_2 - P_1$  : 27 noeuds en déplacement, 4 noeuds en pression (compatible LBB)
- hexaèdre  $Q_2 - Q_1$  : 20 noeuds en déplacement, 8 noeuds en pression
- hexaèdre  $Q_{k+2} - Q_k$  ou  $Q_{k+1} - P_k$  [71] (compatible LBB)
- tétraèdre  $P_1^+ - P_1$  : 4 noeuds + 1 bulle en déplacement, 4 noeuds en pression (compatible LBB)

La linéarisation d'un tel problème variationnel est abordée dans la partie IV, nous supposons donc ici que nous avons à résoudre à chaque itération un système linéaire du type :

$$\begin{pmatrix} K_{uu} & K_{up} \\ K_{pu} & K_{pp} \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f_u \\ f_p \end{pmatrix}\quad (5.6)$$

Dans la partie IV, le lecteur peut vérifier l'inversibilité de  $K_{pp}$ . Cette propriété entraîne la possibilité de réaliser une condensation de Schur pour éliminer la pression du processus de résolution :

$$p = K_{pp}^{-1} (f_p - K_{pi}u)\quad (5.7)$$

$$\hat{K}u = \hat{f} \text{ avec } \begin{cases} \hat{K} = K_{uu} - K_{up}K_{pp}^{-1}K_{pu} \\ \hat{f} = f_u - K_{up}K_{pp}^{-1}f_p \end{cases}\quad (5.8)$$

Dans le cas où il n'y a pas de noeud de pression sur la frontière entre les éléments (cas de l'élément  $Q_2 - P_1$ ), cette condensation peut être réalisée au niveau élémentaire à un coût très faible car la base nodale de l'espace de pression peut être choisie orthonormale et alors  $K_{pp} = -\frac{1}{K}I$ .

## 5.2 Décomposition de domaine avec pression discontinue

On se place ici dans le cas où la pression est discontinue à l'interface, il suffit donc de distinguer les degrés de liberté en déplacement internes (indice  $i$ ) des degrés de liberté en déplacement d'interface (indice  $b$ ), la pression (indice  $p$ ) est considérée comme une variable interne à la sous-structure [49].

Supposons dans un premier temps que la pression a été éliminée au niveau de la sous-structure comme décrit dans (5.7). La condensation du problème sur l'interface entre les sous-domaines conduit au calcul des grandeurs condensées suivantes :

$$\hat{S}_p^{(s)} = \hat{K}_{bb}^{(s)} - \hat{K}_{bi}^{(s)} \hat{K}_{ii}^{(s)-1} \hat{K}_{ib}^{(s)}\quad (5.9)$$

$$\hat{b}_p^{(s)} = \hat{f}_b^{(s)} - \hat{K}_{bi}^{(s)} \hat{K}_{ii}^{(s)-1} \hat{f}_i^{(s)}\quad (5.10)$$



Avec, d'après (5.7)

$$\begin{pmatrix} \hat{K}_{ii}^{(s)} & \hat{K}_{ib}^{(s)} \\ \hat{K}_{bi}^{(s)} & \hat{K}_{bb}^{(s)} \end{pmatrix} = \begin{pmatrix} K_{ii}^{(s)} - K_{ip}^{(s)} K_{pp}^{(s)-1} K_{pi}^{(s)} & K_{ib}^{(s)} - K_{ip}^{(s)} K_{pp}^{(s)-1} K_{pb}^{(s)} \\ K_{bi}^{(s)} - K_{bp}^{(s)} K_{pp}^{(s)-1} K_{pi}^{(s)} & K_{bb}^{(s)} - K_{bp}^{(s)} K_{pp}^{(s)-1} K_{pb}^{(s)} \end{pmatrix} \quad (5.11)$$

$$\begin{pmatrix} \hat{f}_i^{(s)} \\ \hat{f}_b^{(s)} \end{pmatrix} = \begin{pmatrix} f_i^{(s)} - K_{ip}^{(s)} K_{pp}^{(s)-1} f_p^{(s)} \\ f_b^{(s)} - K_{bp}^{(s)} K_{pp}^{(s)-1} f_p^{(s)} \end{pmatrix} \quad (5.12)$$

Notons que quelle que soit l'approche de décomposition de domaine choisie (primale ou duale), le *scaling* rigidité est construit à partir de la diagonale de la matrice  $\begin{pmatrix} K_{bb}^{(s)} - K_{bp}^{(s)} K_{pp}^{(s)-1} K_{pb}^{(s)} \end{pmatrix}$ .

Supposons maintenant que la pression n'a pas été éliminée au niveau de la sous-structure, on a alors :

$$K^{(s)} = \begin{pmatrix} \begin{pmatrix} K_{ii}^{(s)} & K_{ip}^{(s)} \\ K_{pi}^{(s)} & K_{pp}^{(s)} \end{pmatrix} & \begin{pmatrix} K_{ib}^{(s)} \\ K_{pb}^{(s)} \end{pmatrix} \\ \begin{pmatrix} K_{bi}^{(s)} & K_{bp}^{(s)} \end{pmatrix} & K_{bb}^{(s)} \end{pmatrix} \quad f^{(s)} = \begin{pmatrix} f_i^{(s)} \\ f_p^{(s)} \\ f_b^{(s)} \end{pmatrix} \quad (5.13)$$

La condensation sur l'interface entre les sous-structures conduit à :

$$S_p^{(s)} = K_{bb}^{(s)} - \begin{pmatrix} K_{bi}^{(s)} & K_{bp}^{(s)} \end{pmatrix} \begin{pmatrix} K_{ii}^{(s)} & K_{ip}^{(s)} \\ K_{pi}^{(s)} & K_{pp}^{(s)} \end{pmatrix}^{-1} \begin{pmatrix} K_{ib}^{(s)} \\ K_{pb}^{(s)} \end{pmatrix} \quad (5.14)$$

$$b_p^{(s)} = f_b^{(s)} - \begin{pmatrix} K_{bi}^{(s)} & K_{bp}^{(s)} \end{pmatrix} \begin{pmatrix} K_{ii}^{(s)} & K_{ip}^{(s)} \\ K_{pi}^{(s)} & K_{pp}^{(s)} \end{pmatrix}^{-1} \begin{pmatrix} f_i^{(s)} \\ f_p^{(s)} \end{pmatrix} \quad (5.15)$$

Notons qu'ici le *scaling* rigidité est construit à partir de la diagonale de la matrice  $\begin{pmatrix} K_{bb}^{(s)} \end{pmatrix}$ .

Que l'on ait au préalable condensé la pression ou non, on n'a conservé que le déplacement comme inconnue d'interface, on a donc naturellement l'égalité des formulations :

$$\hat{S}_p^{(s)} = S_p^{(s)} \quad (5.16)$$

$$\hat{b}_p^{(s)} = b_p^{(s)} \quad (5.17)$$

Il est donc illogique que les *scalings* ne soient pas construits à partir de la diagonale de la même matrice. De fait, on observe que la condensation des noeuds de pression conduit à une surestimation de la rigidité.

Nous proposons alors deux *scalings* différents, chacun construit à partir d'une mesure pertinente de la rigidité de l'interface, qui fonctionnent parfaitement pour les matériaux compressibles ou non. Le premier est basé sur la rigidité avant condensation  $K_{bb}^{(s)}$ . Comme l'obtention de cette rigidité n'est pas nécessairement évidente dans le cas d'une condensation élémentaire, nous proposons un deuxième *scaling* plus simple mais encore plus efficace basé sur le module de cisaillement  $\mu$  des différents matériaux, on a (en supposant que le noeuds d'interface  $j$  est numéroté de la même manière sur toutes les sous-structures et qu'il connecte la sous-structure  $s$  à la sous-structure  $q$ ) :

$$W_j^{(s)} = \frac{\mu_j^{(s)}}{\sum_{r \in \iota(j)} \mu_j^{(r)}} \quad \text{et} \quad \underline{W}_j^{(s)} = \frac{\mu_j^{(q)}}{\sum_{r \in \iota(j)} \mu_j^{(r)}} \quad (5.18)$$

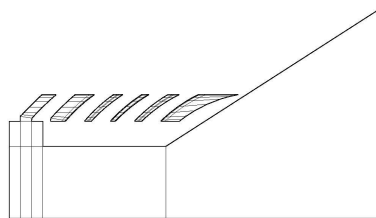
### 5.3 Validation et conclusions

Nous validons les nouveaux *scalings* sur la butée flexible avec élément  $Q_2 - P_1$  (pression condensée au niveau élémentaire). La décomposition est réalisée en 6 sous-domaines, ce qui fait apparaître 5 interfaces acier-caoutchouc, voir figure 5.1. Les résultats donnés dans la table 5.1 correspondent à l'emploi de l'approche primale de décomposition de domaine, ils sont sensiblement les mêmes avec l'approche duale.

Type de <i>scaling</i>	Nombre d'itérations	
	Premier système	Second système
Topologique	290	> 1000
Rigidité classique	120	726
Rigidité avant condensation	48	44
Module de cisaillement	43	39
Structure homogène	93	116

**Tableau 5.1** — Performances des différents *scalings* sur la butée flexible décomposée en 6 calottes (élément  $Q_2 - P_1$ )

On constate l'efficacité des nouveaux *scalings* puisqu'ils permettent même d'obtenir de meilleures performances que lors du calcul de la structure homogène (tout en acier). L'effet de la perturbation  $(-K_{bp}^{(s)} K_{pp}^{(s)-1} K_{pb}^{(s)})$  introduit par la condensation est visible sur le *scaling* rigidité classique, d'ailleurs cette perturbation est accrue lors du second système à un point tel que la convergence ne peut pas être atteinte en un nombre raisonnable d'itérations.



**Figure 5.1** — Décomposition de la butée flexible en 6 sous-domaines

En conclusion nous voyons que les nouveaux *scalings* proposés permettent d'obtenir de bonnes performances sur des structures fortement hétérogènes à composants incompressibles, on retrouve les propriétés d'extensibilité des méthodes de décomposition de domaine primale et duale. Cependant tous les problèmes ne sont pas réglés avec l'hétérogénéité (sans même parler d'incompressibilité) : quand l'interface traverse l'hétérogénéité (et ne la sépare pas) on se retrouve avec des sous-domaines localement hétérogènes et des interfaces "homogènes" (même matériaux en vis-à-vis), ce type d'hétérogénéité ne peut pas être pris en compte par un *scaling* et conduit à de mauvaises performances des méthodes de décomposition de domaine.

Pour finir, citons [44] qui de manière indépendante a proposé une stratégie similaire pour l'étude des écoulements de Stokes.



Troisième partie

Approche hybride de  
décomposition de domaine



---

Le chapitre précédent présente des solutions efficaces pour l'utilisation des méthodes de décomposition de domaine classiques dans le cas de problèmes multichamps (déplacement/pression) où la pression est discontinue à l'interface. Cependant pour de nombreux problèmes où elle représente une caractéristique fondamentale du phénomène étudié, la pression doit nécessairement être continue aux interfaces. On doit donc la considérer comme une inconnue d'interface, le problème condensé est alors posé sur une inconnue (déplacement/pression).

Un problème de ce type se pose pour la simulation des milieux poreux déformables où la pression est une grandeur thermodynamique fondamentale pour caractériser l'état du fluide au sein du squelette. Si l'on envisage la résolution par approche primale, l'inconnue à l'interface est alors un déplacement et une pression. De la même manière si on utilise l'approche duale, l'inconnue à l'interface est un effort et un flux de masse fluide. Dans ces deux cas l'inconnue est hétérogène au sens de ses dimensions physiques (déplacement/pression ou pression/flux) ce qui est connu pour conduire à un problème mal conditionné.

Nous proposons dans cette partie d'avoir un traitement différencié des degrés de liberté de déplacement et de pression : si on traite dualement le déplacement et primalement la pression on obtient une inconnue homogène (pression/pression) à l'interface et donc un problème *a priori* mieux conditionné.

Ce traitement différencié est réalisé au sein de l'approche que nous avons nommée "approche hybride" qui dépasse le cadre multiphysique pour former une généralisation des approches classiques de décomposition de domaine puisque chaque degré de liberté se voit définir un type de traitement (éventuellement indépendamment de sa nature physique). Il est alors possible d'obtenir les approches classiques en choisissant le même traitement pour tous les degrés de liberté (au choix tous primaux ou tous duaux).

Cette partie présente dans un premier chapitre l'approche hybride dans son cadre général en proposant notamment un préconditionneur et en évoquant les problèmes grossiers qui apparaissent naturellement au cours de la formulation. Le second chapitre propose diverses stratégies pour gérer l'ensemble de ces problèmes grossiers. Finalement le troisième chapitre propose une première validation sur un cas académique ainsi que sur un cas issu de l'étude des milieux poreux.



---

# 6 Présentation de l'approche hybride

L'approche hybride est une méthode de décomposition de domaine sans recouvrement développée dans le cadre de cette thèse dont les objectifs sont doubles :

- réunir au sein d'une même approche les méthodes primale et duale, en conservant le principe de leur préconditionnement et leurs problèmes grossiers ;
- offrir la possibilité de développer des approches plus respectueuses de la physique des problèmes multichamps.

Le principe de l'approche hybride réside dans la possibilité d'affecter à chaque degré de liberté un traitement spécifique primal ou dual. On obtient alors un problème condensé à l'interface hybride en variables primales et duales. Naturellement dans le cas où tous les degrés de liberté sont traités de manière primale on retrouve l'approche primale classique, et dans le cas où tous les degrés de liberté sont traités de manière duale on retrouve l'approche duale classique.

Dans ce chapitre l'approche hybride est présentée indépendamment de l'origine mécanique du problème, il s'agit de la présentation purement algébrique de la méthode à partir d'un système linéaire.

Ces travaux ont conduit à une publication [48] et deux communications [45, 47].



## 6.1 Formulation hybride

Considérons un domaine  $\Omega$  décomposé en  $N$  sous-domaines sans recouvrement  $\Omega^{(s)}$  et supposons que nous résolvons un problème linéarisé d'équilibre statique. L'équilibre du sous-domaine discrétisé s'écrit :

$$\forall s, 1 \leq s \leq N, K^{(s)}x^{(s)} = f^{(s)} + \lambda^{(s)} \quad (6.1)$$

où  $K^{(s)}$ ,  $x^{(s)}$  et  $f^{(s)}$  sont la matrice (symétrique) de rigidité, les inconnues (typiquement déplacement/pression) et les efforts appliqués.  $\lambda^{(s)}$  représente les réactions (efforts, flux) à l'interface avec les sous-domaines voisins ( $\lambda^{(s)}$  est nul à l'intérieur du sous-domaine). De manière à simplifier l'exposé et permettre des interprétations mécaniques plus simples nous parleront de déplacements pour  $x$  et efforts pour  $\lambda$ .

Nous supposons la discrétisation conforme, alors l'équilibre et la continuité à l'interface s'écrivent :

$$\sum_{s=1}^N A^{(s)}t^{(s)}\lambda^{(s)} = 0 \quad (6.2)$$

$$\sum_{s=1}^N \underline{A}^{(s)}t^{(s)}x^{(s)} = 0 \quad (6.3)$$

où  $t^{(s)}$  est l'opérateur booléen de trace qui extrait les degrés de liberté d'interface à partir du sous-domaine complet,  $A^{(s)}$  est l'opérateur booléen d'assemblage qui projette les degrés de liberté d'interface locaux sur l'interface géométrique complète,  $\underline{A}^{(s)}$  est l'opérateur booléen signé d'assemblage qui projette les degrés de liberté d'interface locaux sur la description par connectivité de l'interface, voir figure 6.1. Nous rappelons que la notation soulignée correspond aux grandeurs basées sur la description par connectivité de l'interface.

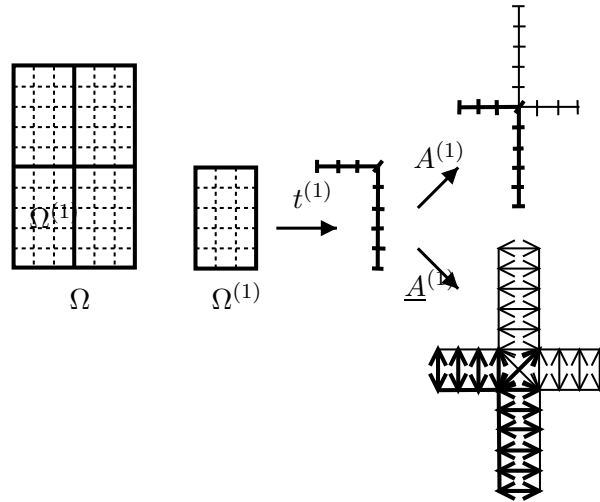


Figure 6.1 — Opérateurs de trace et d'assemblage

Les équations (6.1,6.2,6.3) forment un système strictement équivalent au problème global  $Kx = f$ . Rappelons les approches classiques pour résoudre ce système :

**Approche primale.** L'équation (6.3) est vérifiée *a priori* puisqu'un déplacement d'interface commun  $x_b$  est introduit et les déplacements d'interface locaux sont définis par  $x_b^{(s)} =$

$t^{(s)}x^{(s)} = A^{(s)T} \mathbf{x}_b$ . L'élimination de  $\lambda^{(s)}$  des équations (6.2,6.1) conduit à un problème d'interface d'inconnue  $\mathbf{x}_b$ .

**Approche duale.** L'équation (6.2) est vérifiée *a priori* puisqu'un intereffort commun  $\underline{\lambda}_b$  est défini et les réactions locales s'en déduisent  $\lambda_b^{(s)} = t^{(s)}\lambda^{(s)} = \underline{A}^{(s)T} \underline{\lambda}_b$ . L'élimination de  $x^{(s)}$  des équations (6.3,6.1) conduit à un problème d'interface d'inconnue  $\underline{\lambda}_b$ .

L'approche hybride consiste à mélanger les approches primale et duale en partitionnant les degrés de liberté de l'interface (indice  $b$ ) en deux sous-ensembles : les degrés de liberté primaux (indice  $p$ ) et les degrés de liberté duaux (indice  $d$ ). Nous définissons alors un champ d'inconnu d'interface  $\mathbf{x}_p$  sur les degrés de liberté primaux, et un champ de réaction  $\lambda_d$  sur les degrés de liberté duaux, communs à tous les sous-domaines. Nous allons condenser le problème global pour obtenir un système linéaire d'interface d'inconnues  $(\mathbf{x}_p, \underline{\lambda}_d)$ . En notant avec l'indice  $i$  les degrés de liberté internes aux sous-domaines, le système global devient :

$$\begin{pmatrix} \begin{pmatrix} K_{ii}^{(1)} & K_{id}^{(1)} \\ & K_{dd}^{(1)} \end{pmatrix} & & 0 & \begin{pmatrix} K_{ip}^{(1)} A_p^{(1)T} \\ K_{dp}^{(1)} A_p^{(1)T} \end{pmatrix} \\ & \ddots & & \vdots \\ & & \begin{pmatrix} K_{ii}^{(N)} & K_{id}^{(N)} \\ & K_{dd}^{(N)} \end{pmatrix} & \begin{pmatrix} K_{ip}^{(N)} A_p^{(N)T} \\ K_{dp}^{(N)} A_p^{(N)T} \end{pmatrix} \\ & & & \sum A_p^{(s)} K_{pp}^{(s)} A_p^{(s)T} \end{pmatrix} \begin{pmatrix} \begin{pmatrix} x_i^{(1)} \\ x_d^{(1)} \end{pmatrix} \\ \vdots \\ \begin{pmatrix} x_i^{(N)} \\ x_d^{(N)} \end{pmatrix} \\ \mathbf{x}_p \end{pmatrix} = \begin{pmatrix} \begin{pmatrix} f_i^{(1)} \\ f_d^{(1)} \end{pmatrix} \\ \vdots \\ \begin{pmatrix} f_i^{(N)} \\ f_d^{(N)} \end{pmatrix} \\ \sum A_p^{(s)} f_p^{(s)} \end{pmatrix} + \begin{pmatrix} \begin{pmatrix} 0 \\ \underline{A}_d^{(1)T} \end{pmatrix} \\ \vdots \\ \begin{pmatrix} 0 \\ \underline{A}_d^{(N)T} \end{pmatrix} \\ 0 \end{pmatrix} \underline{\lambda}_d \quad (6.4a)$$

$$\sum \underline{A}_d^{(s)} x_d^{(s)} = 0 \quad (6.4b)$$

où  $A_p^{(s)}$  est l'opérateur d'assemblage restreint aux degrés de liberté d'interface primaux, et  $\underline{A}_d^{(s)}$  l'opérateur signé d'assemblage restreint aux degrés de liberté d'interface duaux. La première équation correspond à l'équilibre des sous-domaines et de l'interface, la seconde à la continuité des inconnues primales sur l'interface.

Introduisons les notations suivantes (naturellement, il est possible de définir les opérateurs complémentaires en inversant les indices  $p$  et  $d$ ) :

$$\text{Restriction : si } v^{(s)} = \begin{pmatrix} v_i^{(s)} \\ v_p^{(s)} \\ v_d^{(s)} \end{pmatrix} \text{ alors } \bar{v}_p^{(s)} = \begin{pmatrix} v_i^{(s)} \\ v_d^{(s)} \end{pmatrix}, \quad \text{Trace : } t_p^{(s)} \begin{pmatrix} v_i^{(s)} \\ v_p^{(s)} \end{pmatrix} = v_p^{(s)}$$

$$\text{Matrices-blocs : } K_p^{(s)} = \begin{pmatrix} K_{ii}^{(s)} & K_{id}^{(s)} \\ K_{di}^{(s)} & K_{dd}^{(s)} \end{pmatrix}, \quad \bar{K}_p = \begin{pmatrix} K_{ip} \\ K_{dp} \end{pmatrix}$$

Le système (6.4) s'écrit alors :

$$\begin{pmatrix} K_p^{(1)} & 0 & \bar{K}_p^{(1)} A_p^{(1)T} \\ & \ddots & \vdots \\ & & K_p^{(N)} & \bar{K}_p^{(N)} A_p^{(N)T} \\ & & & \sum A_p^{(s)} K_{pp}^{(s)} A_p^{(s)T} \end{pmatrix} \begin{pmatrix} \bar{x}_p^{(1)} \\ \vdots \\ \bar{x}_p^{(N)} \\ \mathbf{x}_p \end{pmatrix} = \begin{pmatrix} \bar{f}_p^{(1)} \\ \vdots \\ \bar{f}_p^{(N)} \\ \sum A_p^{(s)} \bar{f}_p^{(s)} \end{pmatrix} + \begin{pmatrix} t_d^{(1)T} \underline{A}_d^{(1)T} \\ \vdots \\ t_d^{(N)T} \underline{A}_d^{(N)T} \\ 0 \end{pmatrix} \underline{\lambda}_d \quad (6.5a)$$

$$\sum \underline{A}_d^{(s)} x_d^{(s)} = \sum \underline{A}_d^{(s)} t_d^{(s)} \bar{x}_p^{(s)} = 0 \quad (6.5b)$$

Pour chaque sous-domaine l'équilibre local s'écrit ((s)<sup>eme</sup> ligne du système (6.5a)) :

$$K_p^{(s)} \bar{x}_p^{(s)} + \bar{K}_p^{(s)} A_p^{(s)T} \mathbf{x}_p = \bar{f}_p^{(s)} + t_d^{(s)T} \underline{A}_d^{(s)T} \underline{\lambda}_d \quad (6.6)$$

Suivant les conditions de Dirichlet imposées sur le sous-domaine, la matrice  $K_p^{(s)}$  peut ne pas être inversible, on utilise alors un pseudo-inverse  $K_p^{(s)+}$  et on introduit  $R_p^{(s)}$  une base de  $\ker(K_p^{(s)})$ , l'amplitude des modes à énergie nulle est notée  $\alpha^{(s)}$ . Le champ  $\bar{x}_p^{(s)}$  devient alors :

$$\bar{x}_p^{(s)} = K_p^{(s)+} \left( -\bar{K}_p^{(s)} A_p^{(s)T} \mathbf{x}_p + \bar{f}_p^{(s)} + t_d^{(s)T} \underline{A}_d^{(s)T} \underline{\lambda}_d \right) + R_p^{(s)} \alpha^{(s)} \quad (6.7)$$

L'utilisation d'une pseudo-inverse est associée à une condition d'admissibilité : mécaniquement parlant, les modes à énergie nulle ne doivent pas être excités par les efforts généralisés, mathématiquement parlant  $K_p^{(s)+}$  doit être appliqué à des vecteurs appartenant à  $\text{Im}(K_p^{(s)})$  ce qui signifie puisque  $K_p^{(s)}$  est symétrique :

$$R_p^{(s)T} \left( -\bar{K}_p^{(s)} A_p^{(s)T} \mathbf{x}_p + \bar{f}_p^{(s)} + t_d^{(s)T} \underline{A}_d^{(s)T} \underline{\lambda}_d \right) = 0 \quad (6.8)$$

L'élimination de  $\bar{x}_p^{(s)}$  dans la  $(N+1)$ <sup>eme</sup> ligne du système (6.5a) conduit à :

$$\begin{aligned} \sum A_p^{(s)} \left( K_{pp}^{(s)} - \bar{K}_p^{(s)T} K_p^{(s)+} \bar{K}_p^{(s)} \right) A_p^{(s)T} \mathbf{x}_p + \sum A_p^{(s)} \left( \bar{K}_p^{(s)T} K_p^{(s)+} t_d^{(s)T} \right) \underline{A}_d^{(s)T} \underline{\lambda}_d \\ + \sum A_p^{(s)} \bar{K}_p^{(s)T} R_p^{(s)} \alpha^{(s)} = \sum A_p^{(s)} \left( \bar{f}_p^{(s)} - \bar{K}_p^{(s)T} K_p^{(s)+} \bar{f}_p^{(s)} \right) \end{aligned} \quad (6.9)$$

L'élimination de  $\bar{x}_p^{(s)}$  dans l'équation (6.5b) donne :

$$\begin{aligned} - \sum \underline{A}_d^{(s)} \left( t_d^{(s)} K_p^{(s)+} \bar{K}_p^{(s)} \right) A_p^{(s)T} \mathbf{x}_p + \sum \underline{A}_d^{(s)} \left( t_d^{(s)} K_p^{(s)+} t_d^{(s)T} \right) \underline{A}_d^{(s)T} \underline{\lambda}_d \\ + \sum \underline{A}_d^{(s)} t_d^{(s)} R_p^{(s)} \alpha^{(s)} = - \sum \underline{A}_d^{(s)} t_d^{(s)} K_p^{(s)+} \bar{f}_p^{(s)} \end{aligned} \quad (6.10)$$

En introduisant alors :

$$S_{pd}^{(s)} = \begin{pmatrix} \left( K_{pp}^{(s)} - \bar{K}_p^{(s)T} K_p^{(s)+} \bar{K}_p^{(s)} \right) & \left( t_d^{(s)} K_p^{(s)+} \bar{K}_p^{(s)} \right)^T \\ - \left( t_d^{(s)} K_p^{(s)+} \bar{K}_p^{(s)} \right) & \left( t_d^{(s)} K_p^{(s)+} t_d^{(s)T} \right) \end{pmatrix} \quad (6.11)$$

$$S_{pd} = \sum \begin{pmatrix} A_p^{(s)} & 0 \\ 0 & \underline{A}_d^{(s)} \end{pmatrix} S_{pd}^{(s)} \begin{pmatrix} A_p^{(s)} & 0 \\ 0 & \underline{A}_d^{(s)} \end{pmatrix}^T \quad (6.12)$$

$$\begin{aligned}
G_p &= \left( \dots, A_p^{(s)} \bar{K}_p^{(s)T} R_p^{(s)}, \dots \right), \quad \underline{G}_d = \left( \dots, \underline{A}_d^{(s)} t_d^{(s)} R_p^{(s)}, \dots \right) \\
\alpha^T &= \left( \dots, \alpha^{(s)T}, \dots \right), \quad e^T = \left( \dots, \bar{f}_p^{(s)T} R_p^{(s)}, \dots \right) \\
b_p &= \sum A_p^{(s)} \left( f_p^{(s)} - \bar{K}_p^{(s)T} K_p^{(s)+} \bar{f}_p^{(s)} \right), \quad b_d = \sum \underline{A}_d^{(s)} t_d^{(s)} K_p^{(s)+} \bar{f}_p^{(s)}
\end{aligned}$$

le problème hybride d'interface (6.9,6.10,6.8) s'écrit :

$$\begin{pmatrix} S_{pd} & \begin{pmatrix} G_p \\ \underline{G}_d \end{pmatrix} \\ (-G_p^T & \underline{G}_d^T) & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x}_p \\ \underline{\lambda}_d \\ \alpha \end{pmatrix} = \begin{pmatrix} b_p \\ -b_d \\ -e \end{pmatrix} \quad (6.13)$$

Ce problème d'interface correspond à la résolution sous contrainte d'un système linéaire. La contrainte est liée à l'éventuelle non-inversibilité des matrices  $K_p^{(s)}$  et donc au choix des degrés de liberté primaux. Par ailleurs, ce système peut représenter, entre autre, l'approche primale classique (si on affecte tous les degrés de liberté de l'interface dans l'ensemble primal) ou l'approche duale classique (si tous les degrés de liberté de l'interface sont dans l'ensemble dual). L'opérateur  $S_{pd}$  est un complément de Schur "primal/dual", il est formé d'un assemblage de contributions locales  $S_{pd}^{(s)}$ . Les compléments de Schur locaux  $S_{pd}^{(s)}$  possèdent une interprétation mécanique qui sera exploitée dans la prochaine section. Quand on observe précisément  $S_{pd}^{(s)}$ , on peut reconnaître dans le premier terme diagonal un complément de Schur primal classique, dans le second terme diagonal un complément de Schur dual classique, et des termes extradiagonaux antisymétriques de couplage.

Remarquons que cette formulation peut être facilement symétrisée :

$$\begin{pmatrix} \hat{S}_{pd} & \begin{pmatrix} G_p \\ \underline{G}_d \end{pmatrix} \\ (G_p^T & \underline{G}_d^T) & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x}_p \\ \underline{\lambda}_d \\ \alpha \end{pmatrix} = \begin{pmatrix} b_p \\ -b_d \\ e \end{pmatrix} \quad (6.14)$$

avec

$$\hat{S}_{pd} = \sum \begin{pmatrix} A_p^{(s)} & 0 \\ 0 & \underline{A}_d^{(s)} \end{pmatrix} \hat{S}_{pd}^{(s)} \begin{pmatrix} A_p^{(s)} & 0 \\ 0 & \underline{A}_d^{(s)} \end{pmatrix}^T \quad (6.15)$$

$$\hat{S}_{pd}^{(s)} = \begin{pmatrix} \left( K_{pp}^{(s)} - \bar{K}_p^{(s)T} K_p^{(s)+} \bar{K}_p^{(s)} \right) & - \left( t_d^{(s)T} K_p^{(s)+} \bar{K}_p^{(s)T} \right)^T \\ - \left( t_d^{(s)T} K_p^{(s)+} \bar{K}_p^{(s)T} \right) & - \left( t_d^{(s)} K_p^{(s)+} t_d^{(s)T} \right) \end{pmatrix} = S_{pd}^{(s)} \begin{pmatrix} 1_p & 0 \\ 0 & -1_d \end{pmatrix} \quad (6.16)$$

Du point de vue résolution, la structure additive de  $S_{pd}$  nous conduit à choisir un solveur itératif, ce qui est un choix classique pour exploiter au mieux la structure parallèle des calculateurs modernes. Le système est défini positif non-symétrique (ou dans sa version 6.14, symétrique non-positif), nous proposons donc d'utiliser le solveur GMRes pour le résoudre (algorithme I.2). Par rapport aux approches primale et duale classiques qui grâce à leur opérateur symétrique défini positif, peuvent employer un gradient conjugué, l'obligation d'utiliser GMRes n'est pas si pénalisante puisque la nécessité de réorthogonaliser le gradient conjugué rend ce dernier presque aussi coûteux que GMRes.

Puisqu'un solveur itératif est retenu, une question importante est le choix d'un préconditionneur efficace, ce qui est le sujet de la prochaine section.

## 6.2 Préconditionnement du problème hybride

En s'inspirant du preconditionneur Neumann-Neumann de l'approche primale et du preconditionneur Dirichlet de l'approche duale, nous proposons de définir le preconditionneur de l'approche hybride comme une somme pondérée des (pseudo) inverses des compléments de Schur locaux :

$$\tilde{S}_{pd}^{-1} = \sum \begin{pmatrix} W_p^{(s)} A_p^{(s)} & 0 \\ 0 & W_d^{(s)} A_d^{(s)} \end{pmatrix} S_{pd}^{(s)+} \begin{pmatrix} W_p^{(s)} A_p^{(s)} & 0 \\ 0 & W_d^{(s)} A_d^{(s)} \end{pmatrix}^T \quad (6.17)$$

où  $W_p^{(s)}$  et  $W_d^{(s)}$  sont des matrices diagonales de pondération (*scaling*) détaillées dans les chapitres précédents.

L'inverse des compléments de Schur locaux peut être déduite d'une analyse mécanique de l'équilibre d'une sous-structure. Concentrons-nous sur un sous-domaine, l'équilibre de ce dernier sous un chargement équilibré (n'excitant pas les modes à énergie nulle) sur sa frontière s'écrit :

$$\begin{pmatrix} K_{ii}^{(s)} & K_{id}^{(s)} & K_{ip}^{(s)} \\ & K_{dd}^{(s)} & K_{dp}^{(s)} \\ & & K_{pp}^{(s)} \end{pmatrix} \begin{pmatrix} x_i^{(s)} \\ x_d^{(s)} \\ x_p^{(s)} \end{pmatrix} = \begin{pmatrix} 0 \\ \lambda_d^{(s)} \\ \lambda_p^{(s)} \end{pmatrix} \quad (6.18)$$

L'élimination de  $x_i^{(s)}$  conduit à :

$$S_{pd}^{(s)} \begin{pmatrix} x_p^{(s)} \\ \lambda_d^{(s)} \end{pmatrix} = \begin{pmatrix} \lambda_p^{(s)} \\ x_d^{(s)} \end{pmatrix} \quad (6.19)$$

Donc le complément de Schur local  $S_{pd}^{(s)}$  associe les efforts sur les degrés de liberté primaux et les déplacements sur les degrés de liberté duaux à une réaction imposée sur les degrés duaux et un déplacement imposé sur les degrés primaux. En définissant  $S_{dp}^{(s)}$  comme l'opérateur d'interface qui associe les réactions sur les degrés duaux et les déplacements sur les degrés primaux à une réaction imposée sur les degrés primaux et une condition de Dirichlet sur les degrés duaux, on obtient :

$$S_{dp}^{(s)} = \begin{pmatrix} \begin{pmatrix} t_p^{(s)} K_d^{(s)+} t_p^{(s)T} \\ \end{pmatrix} & - \begin{pmatrix} t_p^{(s)} K_d^{(s)+} \bar{K}_d^{(s)} \\ \end{pmatrix} \\ \begin{pmatrix} t_p^{(s)} K_d^{(s)+} \bar{K}_d^{(s)} \end{pmatrix}^T & \begin{pmatrix} K_{dd}^{(s)} - \bar{K}_d^{(s)T} K_d^{(s)+} \bar{K}_d^{(s)} \end{pmatrix} \end{pmatrix} \quad (6.20)$$

$$S_{dp}^{(s)} \begin{pmatrix} \lambda_p^{(s)} \\ x_d^{(s)} \end{pmatrix} = \begin{pmatrix} x_p^{(s)} \\ \lambda_d^{(s)} \end{pmatrix} \text{ et naturellement } S_{dp}^{(s)} = S_{pd}^{(s)+} \quad (6.21)$$

A partir de l'équation (6.16) un preconditionneur symétrique pour le système symétrisé peut être obtenu.

## 6.3 Condition d'optimalité du preconditionneur

Bien sûr, suivant le choix des degrés de liberté primaux,  $K_d^{(s)}$  peut (comme dans l'approche primale) ne pas être inversible. La condition d'optimalité du preconditionneur s'écrit alors

que tout vecteur  $v_d^{(s)}$  devant être multiplié par  $K_d^{(s)+}$  doit appartenir à  $\text{Im}(K_d^{(s)})$ . Soit  $R_d^{(s)}$  une base de  $\ker(K_d^{(s)})$  alors la condition d'optimalité s'écrit  $R_d^{(s)T} v_d^{(s)} = 0$ .

D'après (6.17), la condition d'optimalité devient

$$\begin{pmatrix} H_p \\ \underline{H}_d \end{pmatrix}^T \begin{pmatrix} \lambda_p \\ \underline{u}_d \end{pmatrix} = 0 \quad (6.22)$$

avec  $H_p = (\dots W_p^{(s)} A_p^{(s)} t_p^{(s)} R_d^{(s)} \dots)$  et  $\underline{H}_d = (\dots \underline{W}_d^{(s)} \underline{A}_d^{(s)} \bar{K}_d^{(s)T} R_d^{(s)} \dots)$ .

La mise en place de l'ensemble des conditions d'admissibilité, d'optimalité (et d'éventuelles contraintes optionnelles) conduit à des algorithmes assez sophistiqués quoique d'un principe relativement simple. Ils sont détaillés dans le chapitre suivant.

## 6.4 Conclusion

Telle qu'elle a été présentée dans ce chapitre, l'approche hybride constitue bien une généralisation des approches duale et primale classiques. L'ensemble des concepts relatifs au préconditionnement (éventuellement approché) et aux problèmes grossiers (obligatoires ou optionnels) sont respectés. Du point de vue mise en oeuvre, cette méthode ne présente aucune difficulté supplémentaire par rapport à la mise en oeuvre d'une approche classique à deux niveaux.

Les nouvelles opportunités offertes par l'approche hybride pour la résolution de problèmes mécaniques sont évoqués dans le chapitre 8. Le chapitre 7 détaille quant-à lui les différentes possibilités pour la mise en place des différents problèmes grossiers.

Le lecteur aura peut-être ressenti à la lecture de ce chapitre une forte connexion entre FETI-DP (voir annexe A) et l'approche hybride. En effet le système de base est le même, seulement dans FETI-DP les degrés primaux sont recondensés pour être éliminés du processus de résolution, ce qui a notamment pour conséquence de limiter le nombre de coins (degré de liberté primaux). De fait les philosophies de FETI-DP et de l'approche hybride sont différentes : FETI-DP cherche à supprimer le problème des sous-structures flottantes tout en restant attaché à l'approche duale ; l'approche hybride ne cherche pas à privilégier l'aspect dual ou l'aspect primal, il s'agit de résoudre itérativement un problème primal/dual sur l'interface complète. Nous verrons d'un point de vue programmation (partie V) qu'il est prévu d'étendre le concept de noeud recondensé à l'approche hybride toujours sans privilégier les aspects primaux ou duaux.

Nous n'avons pas cherché dans le cadre de ces travaux à prouver l'extensibilité théorique de la méthode hybride de décomposition de domaine. Une telle démonstration nécessiterait le développement de nouveaux outils par rapport aux méthodes classiques. En effet les démonstrations pour les approches primale, duale ou FETI-DP reposent essentiellement sur la définition d'un produit scalaire à l'interface construit sur le complément de Schur (qui est dans les cas évoqués symétrique (semi)-défini positif), or le complément de Schur hybride est non-symétrique non-positif ce qui rend inapplicable tel quel le principe des autres démonstrations.



CHAPITRE

# 7

---

## Gestion des problèmes grossiers dans l'approche hybride

Ce chapitre détaille les différentes stratégies possibles pour vérifier les conditions d'admissibilité, d'optimalité et les éventuelles contraintes optionnelles, qui interviennent dans l'approche hybride de décomposition de domaine.

Ces stratégies reposent sur le principe de séparation de l'espace de recherche évoqué dans le chapitre 2. Les différentes possibilités sont liées aux choix possibles de combiner ou emboîter les contraintes et conduisent à des stratégies à un, deux ou trois projecteurs.



Par simplicité, nous choisissons de travailler avec la formulation symétrisée (6.14), et de manière à condenser les écritures, nous adoptons les notations suivantes :

$$\begin{aligned} G &= (G_p^T \quad \underline{G}_d^T)^T & H &= (H_p^T \quad \underline{H}_d^T)^T \\ x &= (u_p^T \quad \underline{\lambda}_d^T)^T & b &= (b_p^T \quad -\underline{b}_d^T)^T \end{aligned}$$

En omettant pour le reste de ce chapitre la notation  $\hat{\phantom{x}}$  le système symétrisé devient :

$$\begin{pmatrix} S_{pd} & G \\ G^T & 0 \end{pmatrix} \begin{pmatrix} x \\ \alpha \end{pmatrix} = \begin{pmatrix} b \\ e \end{pmatrix} \quad (7.1)$$

Puisque le préconditionneur multiplie le résidu  $r = (b - S_{pd}x)$ , la condition d'optimalité associée au préconditionnement s'écrit :

$$H^T r = 0 \quad (7.2)$$

Des contraintes optionnelles peuvent être ajoutées [36, 27, 85, 46]. Si  $C$  est la matrice des contraintes, la condition s'écrit :

$$C^T r = 0 \quad (7.3)$$

Il y a donc trois conditions à vérifier : une associée à l'inconnue  $x$  (matrice  $G$ ) et deux associées au résidu (matrices  $H$  et  $C$ ). La première est obligatoire (si elle n'est pas vérifiée, la solution correcte ne peut pas être trouvée), les deux autres sont optionnelles et sont sensées rendre l'algorithme extensible et plus performant.

Le principe du processus de résolution est de séparer l'espace de recherche en sous-espaces supplémentaires afin que le processus itératif soit limité dans le sous-espace où toutes les conditions d'orthogonalité sont vérifiées. Cette stratégie correspond aux algorithmes de Krylov-augmenté (voir chapitre 2), et peut être mise en oeuvre soit par une orthogonalisation spécifique soit par des méthodes de projection. Nous choisissons de la présenter sous la forme de projecteurs.

Soit  $\mathcal{F} = \ker(F^T)$  et  $\mathcal{L} = \text{Im}(L)$  deux sous-espaces vectoriels supplémentaires. La projection sur  $\mathcal{F}$  parallèlement à  $\mathcal{L}$  se construit de la manière suivante :

$$P_F^L = I - L(F^T L)^{-1} F^T \quad (7.4)$$

On a les relations :  $P_F^L L = 0$  et  $P_F^L F = 0$ . Si  $L = F$  la projection est orthogonale. Une manière très classique de définir des projections est d'utiliser une matrice  $Q$  pour laquelle la matrice  $(F^T Q F)$  est inversible, et d'utiliser :

$$P_F^{QF} = I - QF(F^T Q F)^{-1} F^T \quad (7.5)$$

Les équations suivantes sont couramment associées à l'usage de projections, nous donnons leurs solutions classiques :

$$F^T x = e \quad \Leftrightarrow \quad x = L(F^T L)^{-1} e \quad (7.6)$$

$$L\alpha = r \quad \Leftrightarrow \quad \alpha = (F^T L)^{-1} F^T r \quad (7.7)$$

## 7.1 Approche à un projecteur

Dans cette approche toutes les conditions sont vérifiées ensembles. Si on introduit  $Y = [G, S_{pd}H, S_{pd}C]$ , si  $Y$  n'est pas de rang égal à son nombre de colonnes on suppose qu'un algorithme d'élimination a supprimé les vecteurs redondants, le système s'écrit :

$$\begin{pmatrix} S_{pd} & Y \\ Y^T & 0 \end{pmatrix} \begin{pmatrix} x \\ \alpha_Y \end{pmatrix} = \begin{pmatrix} b \\ e_Y \end{pmatrix} \quad \text{avec} \quad e_Y = \begin{pmatrix} e \\ H^T b \\ C^T b \end{pmatrix}, \quad \alpha_Y = \begin{pmatrix} \alpha \\ \alpha_H \\ \alpha_C \end{pmatrix} \quad (7.8)$$

On peut choisir de séparer l'inconnue :

$$x = P_Y^{QY} x^* + x_0 \quad (7.9)$$

alors le système devient :

$$\begin{cases} Y^T x_0 = e_Y & \Leftrightarrow & x_0 = QY (Y^T QY)^{-1} e_Y \\ S_{pd} P_Y^{QY} x^* + Y \alpha_Y = (b - S_{pd} x_0) & \Rightarrow & P_Y^{QY T} S_{pd} P_Y^{QY} x^* = P_Y^{QY T} (b - S_{pd} x_0) \end{cases} \quad (7.10)$$

D'abord l'initialisation  $x_0$  est déterminée de manière à vérifier les différentes contraintes ensuite l'opérateur  $S_{pd}^* = P_Y^{QY T} S_{pd} P_Y^{QY}$  est défini et le membre de droite modifié pour déterminer  $x^*$ , part de la solution appartenant à  $\ker(Y^T)$ . Finalement  $\alpha_Y$  peut être calculé :

$$Y \alpha_Y = (b - S_{pd}(x_0 + P_Y^{QY} x^*)) \quad \Leftrightarrow \quad \alpha_Y = (Y^T QY)^{-1} Y^T Q (b - S_{pd} x) \quad (7.11)$$

Connaissant  $x$  et  $\alpha$ , les sous-domaines possèdent toutes les informations nécessaires à la reconstruction des solutions locales.

A la connaissance de l'auteur, cette approche n'a jamais été mise en oeuvre, tout d'abord car le problème grossier résultant  $(Y^T QY)$  est calculatoirement coûteux, ensuite car l'interprétation et le choix de la matrice  $Q$  sont malaisés.

## 7.2 Approche à deux projecteurs

L'idée de cette méthode est de séparer la contrainte sur l'inconnue  $x$  des contraintes sur le résidu  $r$ . Tout d'abord, la même idée que dans l'exemple précédent est utilisée pour gérer la contrainte  $G$  :

$$\begin{pmatrix} S_{pd} & G \\ G^T & 0 \end{pmatrix} \begin{pmatrix} x \\ \alpha \end{pmatrix} = \begin{pmatrix} b \\ e \end{pmatrix} \quad (7.12)$$

$$\begin{cases} x = P_G^{QG} x^* + x_0 \\ G^T x_0 = e & \Leftrightarrow & x_0 = QG (G^T QG)^{-1} e \\ P_G^{QG T} S_{pd} P_G^{QG} x^* = P_G^{QG T} (b - S_{pd} x_0) = r_0^* \\ \alpha = (G^T QG)^{-1} G^T Q (b - S_{pd} x) \end{cases} \quad (7.13)$$

Dans ce cas, le choix de la matrice  $Q$  est classique, des discussions sur le sujet peuvent être trouvées dans la littérature relative à FETI [51, 82, 6] ou au chapitre 3 de cette thèse. Rappelons que l'on choisit typiquement  $Q = I$  pour les problèmes homogènes et un préconditionneur disponible pour les problèmes hétérogènes.

Soit  $S_{pd}^* = P_G^{QG^T} S_{pd} P_G^{QG}$ ,  $S_{pd}^*$  est toujours une matrice symétrique. Le résidu  $r^*$  est défini comme  $r^* = r_0^* - S_{pd}^* x^*$ , la contrainte est  $[H, C]^T r^* = 0$ . Soit alors  $Y = [H, C]$ , si le rang de  $Y$  n'est pas égal à son nombre de colonnes alors on suppose qu'un algorithme d'élimination a supprimé les vecteurs redondants, le système \* s'écrit alors :

$$\begin{pmatrix} S_{pd}^* & S_{pd}^* Y \\ Y^T S_{pd}^* & 0 \end{pmatrix} \begin{pmatrix} x^* \\ \alpha_Y \end{pmatrix} = \begin{pmatrix} r_0^* \\ Y^T r_0^* \end{pmatrix} \quad (7.14)$$

Nous introduisons une seconde séparation de l'espace de recherche

$$\begin{cases} x^* = P_{S_{pd}^* Y}^{Q^* S_{pd}^* Y} x^{**} + x_0^* \\ Y^T S_{pd}^* x_0^* = Y^T r_0^* \iff x_0^* = Q^* S_{pd}^* Y \left( Y^T S_{pd}^* Q^* S_{pd}^* Y \right)^{-1} Y^T r_0^* \\ P_{S_{pd}^* Y}^{Q^* S_{pd}^* Y^T} S_{pd}^* P_{S_{pd}^* Y}^{Q^* S_{pd}^* Y} x^{**} = P_{S_{pd}^* Y}^{Q^* S_{pd}^* Y^T} \left( r_0^* - S_{pd}^* x_0^* \right) \end{cases} \quad (7.15)$$

A la connaissance de l'auteur, la matrice  $Q^*$  n'a jamais été introduite explicitement. En fait, elle est choisie de manière à ce que  $Q^* S_{pd}^* = P_G^{QG}$ , ce qui permet de simplifier les expressions<sup>1</sup> :

$$\begin{cases} x_0^* = (P_G^{QG} Y) \left( (P_G^{QG} Y)^T S_{pd} (P_G^{QG} Y) \right)^{-1} (P_G^{QG} Y)^T r_0^* \\ P_{S_{pd}^* Y}^{Q^* S_{pd}^* Y} = I - (P_G^{QG} Y) \left( (P_G^{QG} Y)^T S_{pd} (P_G^{QG} Y) \right)^{-1} (P_G^{QG} Y)^T S_{pd} = P_{S_{pd}^* P_G^{QG} Y}^{P_G^{QG} Y} \end{cases} \quad (7.16)$$

Ce qui s'interprète comme rendre la contrainte  $Y$  admissible par rapport à la contrainte  $G$  et consiste à appliquer  $P_G^{QG}$  à la matrice  $Y$  (et si nécessaire, rendre la matrice de rang plein). Alors  $P_{S_{pd}^* P_G^{QG} Y}^{P_G^{QG} Y}$  est lui même un projecteur.

### 7.3 Approche à trois projecteurs

Sur la même base que l'approche précédente, cette stratégie consiste à séparer les deux contraintes  $H$  et  $C$ . En supposant que des matrices "Q" similaires sont utilisées pour ces deux contraintes, cette approche conduit exactement au même processus numérique que la précédente. Cependant elle donne une formulation un peu plus élégante puisqu'elle permet de définir un préconditionneur projeté autonome. Nous commençons par définir une contrainte  $H^*$  admissible par rapport à la contrainte  $G$  :

$$H^* = P_G^{QG} H \quad (7.17)$$

on introduit alors un préconditionneur modifié :

$$\tilde{S}_{pd}^{*-1} = P_{S_{pd}^* H^*}^{H^*} \tilde{S}_{pd}^{-1} P_{S_{pd}^* H^*}^{H^* T} \quad (7.18)$$

Autrement dit, on utilise la stratégie précédente avec  $Y = H$  :

$$\begin{pmatrix} S_{pd}^* & S_{pd}^* H \\ H^T S_{pd}^* & 0 \end{pmatrix} \begin{pmatrix} x^* \\ \alpha_H \end{pmatrix} = \begin{pmatrix} r_0^* \\ H^T r_0^* \end{pmatrix} \quad (7.19)$$

<sup>1</sup>Il convient de noter que  $P_G^{QG^T} P_G^{QG^T} = P_G^{QG^T}$ ,  $r_0^* = P_G^{QG^T} r_0^*$

$$\begin{cases} x_0^* = H^* (H^{*T} S_{pd} H^*)^{-1} H^{*T} r_0^* \\ P_{S_{pd} H^*}^{H^* T} S_{pd}^* P_{S_{pd} H^*}^{H^*} x^{**} = P_{S_{pd} H^*}^{H^* T} (r_0^* - S_{pd}^* x_0^*) = r_0^{**} \end{cases} \quad (7.20)$$

Soit  $S_{pd}^{**} = P_{S_{pd} H^*}^{H^* T} S_{pd}^* P_{S_{pd} H^*}^{H^*}$ , le résidu est  $r_0^{**} = r_0^* - S_{pd}^* x_0^*$ , la contrainte  $C^T r_0^{**} = 0$ . Le système s'écrit alors :

$$\begin{pmatrix} S_{pd}^{**} & S_{pd}^{**} C \\ C^T S_{pd}^{**} & 0 \end{pmatrix} \begin{pmatrix} x^{**} \\ \alpha_C \end{pmatrix} = \begin{pmatrix} r_0^{**} \\ C^T r_0^{**} \end{pmatrix} \quad (7.21)$$

Ce qui conduit à la même séparation d'espace que précédemment :

$$\begin{cases} x^{**} = P_{S_{pd}^{**} C}^{Q^{**} S_{pd}^{**} C} x^{***} + x_0^{**} \\ C^T S_{pd}^{**} x_0^{**} = C^T r_0^{**} \Leftarrow x_0^{**} = Q^{**} S_{pd}^{**} C (C^T S_{pd}^{**} Q^{**} S_{pd}^{**} C) C^T r_0^{**} \\ P_{S_{pd}^{**} C}^{Q^{**} S_{pd}^{**} C^T} S_{pd}^{**} P_{S_{pd}^{**} C}^{Q^{**} S_{pd}^{**} C} x^{***} = P_{S_{pd}^{**} C}^{Q^{**} S_{pd}^{**} C^T} (r_0^{**} - S_{pd}^{**} x_0^{**}) \end{cases} \quad (7.22)$$

Ici  $Q^{**}$  est choisie de manière à ce que  $Q^{**} S_{pd}^{**} = P_{S_{pd} H^*}^{H^*}$  ainsi si on note  $C^* = P_{S_{pd} H^*}^{H^*} C$ , le système devient :

$$\begin{cases} x_0^{**} = C^* (C^{*T} S_{pd}^* C^*)^{-1} (C^{*T} r_0^{**}) \\ P_{S_{pd}^{**} C}^{Q^{**} S_{pd}^{**} C} = P_{S_{pd}^* C^*}^{C^*} \end{cases} \quad (7.23)$$

ce qui s'interprète ici encore comme rendre la contrainte  $C$  admissible par rapport aux contraintes précédentes  $G$  et  $H$ .

## 7.4 Conclusion

Bien que la mise en place des problèmes grossiers pour l'approche hybride conduise à des écritures un peu lourdes, le principe n'en reste pas moins simple, récursif et similaire à celui des approches classiques à deux niveaux.

Bien que toutes les approches décrites permettent de satisfaire les contraintes de continuité grossière, nous préconisons l'emploi des approches à deux ou trois projecteurs qui nous semblent plus élégantes et cohérentes. Le principe est alors de rendre compatible les contraintes avec celles de plus bas niveau, ce qui est réalisé par une simple projection.

Pour finir, remarquons que la mise en place des problèmes grossiers sur la formulation non-symétrisée (qui respecte mieux la nature mécanique des opérateurs) ne conduit pas à de difficultés supplémentaires mais nécessite légèrement plus d'attention pour la prise en compte de la contrainte d'admissibilité et pour la mise en oeuvre des projecteurs.



CHAPITRE

# 8

---

## Une première validation de l'approche hybride

Pour valider l'approche hybride nous proposons dans un premier temps de nous intéresser à un problème académique issu de l'élasticité linéaire afin de constater la force algorithmique de l'approche, ensuite nous verrons qu'elle prend tout son sens dans le cadre des problèmes multichamps notamment pour l'étude des milieux poreux.

## 8.1 Cisaillement d'un carré élastique

On étudie le cisaillement d'un carré élastique en déformations planes (voir figure 8.1) discrétisé en 400 éléments-finis  $Q_2$  (8 noeuds). On décompose cette structure en 10 sous-

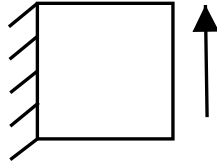


Figure 8.1 — Chargement en cisaillement

domaines, et de manière à voir la portée de l'approche hybride, on définit différents ensembles de degrés de liberté (voir figure 8.2) : les points multiples (PM, en clair sur la figure) et les "coins" (C, "sommets" des sous-structures qui ne sont pas des points multiples, en foncé sur la figure). La convergence est évaluée d'après le critère de GMRes fixé à  $10^{-8}$ .

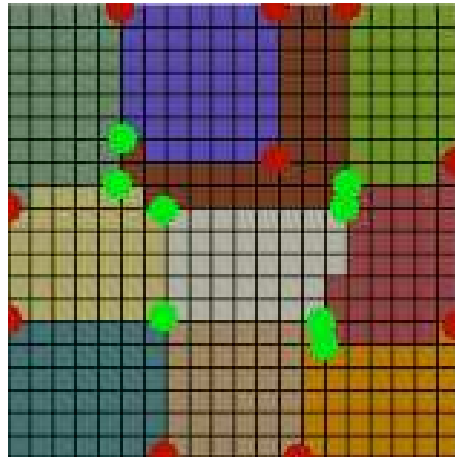


Figure 8.2 — Définition des ensembles de degrés de liberté

Le tableau 8.1 rassemble les résultats de l'approche hybride pour différentes configurations : les deux premières colonnes décrivent la répartition des degrés de liberté d'interface entre les ensembles primaux et duaux, les deux suivantes donnent le nombre de déplacements de solides rigides apparaissant respectivement dans l'opérateur et le préconditionneur (si on choisit de ne pas utiliser le projecteur du préconditionneur, ce nombre est mis à  $\mathbf{0}$ ), la dernière colonne indique le nombre d'itérations pour converger.

La première ligne (tous duaux) correspond à l'approche FETI classique, la seconde ligne (tous primaux) à l'approche primale classique et la troisième à l'approche primale sans projecteur modes rigides. On observe que naturellement les approches primale et duale classiques convergent de manière très proche, l'approche primale non-optimale est significativement moins bonne (27 itérations en plus pour 21 modes rigides en moins). Les 5 lignes suivantes font varier les points multiples et les coins entre les ensembles primaux et duaux conduisant à des espaces rigides de plus ou moins grande dimension et des convergences plus ou moins bonnes. Notons tout de même l'approche où les points multiples sont duaux, sans problème grossier la convergence est bien meilleure que pour l'approche primale sans problème grossier ; on peut donc raisonnablement espérer trouver des configurations où l'approche hybride puisse se comporter de manière efficace d'un point de vue temps de calcul.

ddl primaux	ddl duaux	mnr $S_{pd}$	mnr $\tilde{S}_{pd}^{-1}$	nb itérations
$\emptyset$	tous	21	0	20
tous	$\emptyset$	0	21	18
tous	$\emptyset$	0	<b>0</b>	45
PM	rest	3	0	53
reste	PM	0	3	38
reste	PM	0	<b>0</b>	36
PM+C	reste	0	0	56
reste	PM+C	0	0	50
U1	U2	7	7	31
U1	U2	7	<b>0</b>	28
U2	U1	7	7	35
U2	U1	7	<b>0</b>	32

**Tableau 8.1** — Convergence de l'approche hybride pour diverses configurations

Les quatre dernières lignes correspondent à des séparations au niveau du degré de liberté et non au niveau du noeud : selon leur direction, les déplacements sont répartis entre les espaces primaux et duaux, on voit encore dans ces cas là que la convergence peut être bonne avec un espace grossier réduit.

Cet exemple académique démontre les possibilités de l'approche hybride en terme de souplesse d'utilisation. Au niveau des performances numériques, elles sont toujours en retrait par rapport aux approches classiques, cependant, l'usage de l'approche hybride s'accompagne d'une réduction du nombre de modes rigides, ce qui peut conduire à des temps de calculs meilleurs que les approches classiques. Un problème essentiel sur un problème académique est comment choisir la répartition entre degrés de liberté primaux et duaux, pour l'instant aucune stratégie générale et efficace n'a été développée.

Cette difficulté est levée pour les problèmes multiphysiques où nous montrons que l'approche hybride permet de résoudre les problème avec un sens physique plus aigu et des performances meilleures que les approches classiques.

## 8.2 Compactage d'un sol

### 8.2.1 Étude des milieux poreux

On considère l'évolution isotherme dans l'intervalle de temps  $[0, t_f]$  d'une structure  $\Omega$  formée d'un milieu poreux saturé, en petites perturbations autour de sa position de référence [14]. Le milieu est perturbé par un chargement imposé  $\bar{u}_0$  sur la partie  $\partial_u\Omega$  de la frontière, d'un effort imposé  $\bar{f}$  sur la partie complémentaire  $\partial_f\Omega$ , d'une pression imposée  $p_0$  sur une autre partie  $\partial_p\Omega$  de la frontière et d'un flux de masse fluide  $M_0$  sur la partie complémentaire  $\partial_M\Omega$  (voir figure 8.3). On note  $m$  l'apport de masse fluide et  $\bar{M}$  le flux de masse fluide.

Les équations qui gouvernent l'évolution du milieu poreux ainsi chargé sont les suivantes :

- équilibre du squelette et conservation de la masse de fluide :

$$\overline{\text{div}}(\bar{\sigma}) = \bar{0} \quad (8.1)$$

$$\dot{m} = -\text{div}(\bar{M}) \quad (8.2)$$



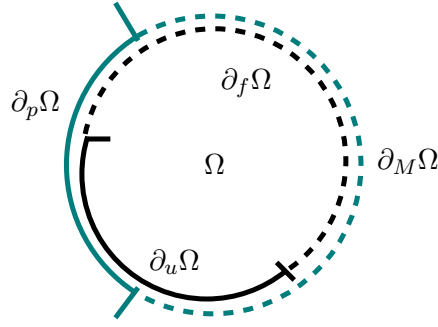


Figure 8.3 — Conditions aux limites sur un massif poreux

- comportement élastique linéaire du squelette, loi d'état du fluide :

$$\overline{\sigma} = \mathbf{a} : \overline{\varepsilon}(\overline{u}) - p\overline{\overline{B}} \quad (8.3)$$

$$\frac{m}{\rho} = Np + \overline{\varepsilon}(\overline{u}) : \overline{\overline{B}} \quad (8.4)$$

où  $\mathbf{a}$  est le tenseur de rigidité drainée du squelette,  $\overline{\overline{B}}$  le tenseur de Biot,  $N$  l'inverse du module de Biot et  $\rho$  la masse volumique du fluide ;

- transport hydrique (loi de Darcy) :

$$\frac{\overline{M}}{\rho} = -\overline{\overline{K}}.\overline{\text{grad}}(p) \quad (8.5)$$

où  $\overline{\overline{K}}$  est le tenseur de perméabilité du fluide ;

- conditions aux limites :

$$\overline{\sigma}.\overline{n} = \overline{f} \quad \text{sur} \quad \partial_f\Omega \quad (8.6)$$

$$\overline{u} = \overline{u}_0 \quad \text{sur} \quad \partial_u\Omega \quad (8.7)$$

$$p = p_0 \quad \text{sur} \quad \partial_p\Omega \quad (8.8)$$

$$\overline{M}.\overline{n} = M_0 \quad \text{sur} \quad \partial_M\Omega \quad (8.9)$$

Pour réaliser une approche en déplacement/pression on introduit les espaces admissibles  $\mathcal{U}(\Omega)$  et  $\mathcal{P}(\Omega)$

$$\mathcal{U}(\Omega) = \{ \overline{u} \in H^1(\Omega)^n, \overline{u} = \overline{u}_0 \text{ sur } \partial_u\Omega \}$$

$$\mathcal{P}(\Omega) = \{ p \in H^1(\Omega), p = p_0 \text{ sur } \partial_p\Omega \}$$

L'affaiblissement des relations d'équilibre et de conservation (8.1) conduit à la formulation variationnelle suivante :

trouver  $(\overline{u}, p) \in \mathcal{U}(\Omega) \times \mathcal{P}(\Omega) / \forall (\overline{v}, q) \in \mathcal{U}(\Omega) \times \mathcal{P}(\Omega), \forall t \in [0, t_f]$

$$\int_{\Omega} (\mathbf{a} : \overline{\varepsilon}(\overline{u})) : \overline{\varepsilon}(\overline{v} - \overline{u}) d\Omega - \int_{\Omega} p\overline{\overline{B}} : \overline{\varepsilon}(\overline{v} - \overline{u}) d\Omega = \int_{\partial_f\Omega} \overline{f} . (\overline{v} - \overline{u}) dS$$

$$\int_{\Omega} \rho \left( N\dot{p} + \overline{\overline{B}} : \overline{\varepsilon}(\dot{\overline{u}}) \right) (q - p) + \int_{\Omega} \rho \left( \overline{\overline{K}}.\overline{\text{grad}}(p) \right) . \overline{\text{grad}}(q - p) + \int_{\partial_M\Omega} M_0(q - p) dS = 0 \quad (8.10)$$

La mise en oeuvre éléments-finis est conduite classiquement par la définition de sous espaces d'approximation de dimension finie (voir le chapitre 5 pour plus de détails sur le

choix d'éléments-finis mixtes). On obtient alors le système suivant :

$$\begin{pmatrix} 0 & 0 \\ -B^T & -C \end{pmatrix} \begin{pmatrix} \dot{u} \\ \dot{p} \end{pmatrix} + \begin{pmatrix} K_1 & -B \\ 0 & -K_2 \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f_u \\ f_p \end{pmatrix} \quad (8.11)$$

où  $K_1$  et  $K_2$  sont les matrices de rigidité et de perméabilité,  $B$  la matrice de couplage poromécanique,  $C$  la matrice de compressibilité du fluide,  $f_u$  le vecteur des efforts généralisés et  $f_p$  le vecteur des flux généralisés.

La résolution de ce système est obtenue en utilisant une discrétisation temporelle selon un schéma d'Euler implicite, ce qui conduit à la résolution à chaque instant  $t_{n+1}$  du système :

$$\begin{pmatrix} K_1 & -B \\ -B^T & -(C + \Delta t K_2) \end{pmatrix} \begin{pmatrix} u^{n+1} \\ p^{n+1} \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ -B^T & -C \end{pmatrix} \begin{pmatrix} u^n \\ p^n \end{pmatrix} + \begin{pmatrix} f_u^{n+1} \\ \Delta t f_p^{n+1} \end{pmatrix} \quad (8.12)$$

Classiquement, le système (8.12) est résolu soit par une approche directe sur le système couplé ("approche monolithique"), ce qui conduit avec des notations triviales à l'inversion du système suivant :

$$Kx^{n+1} = f^{n+1} \quad (8.13)$$

soit par une approche fondée sur des découplages partiels entre les phénomènes physiques, cela conduit à utiliser des solveurs de type Jacobi, Gauss-Siedel ou relaxation, ou des méthodes comme décrites dans [64, 20].

Nous proposons de résoudre le système couplé (8.13) par une approche de décomposition de domaine. Dans cette partie nous n'étudions pas la forme très particulière de cette séquence de systèmes linéaires (matrice invariante), cet aspect est évoqué dans la partie IV. Nous nous focalisons donc sur la résolution d'un de ces systèmes linéaires, et par soucis de lisibilité nous omettons le numéro de l'itération.

### 8.2.2 Simulation

Considérons un sol à la pression atmosphérique, un effort de compactage (effort croissant linéairement jusqu'à un seuil) est appliqué sur une surface carrée de  $2m \times 2m$ , un massif de  $10m \times 8m \times 2m$  centré sur la zone de compactage est isolé, il est supposé suffisamment grand pour que sur ces bords latéraux et inférieur les flux, les déplacements normaux et les efforts tranchants soient nuls. L'utilisation des symétries conduit à la schématisation représentée (en 2D) sur la figure 8.4).

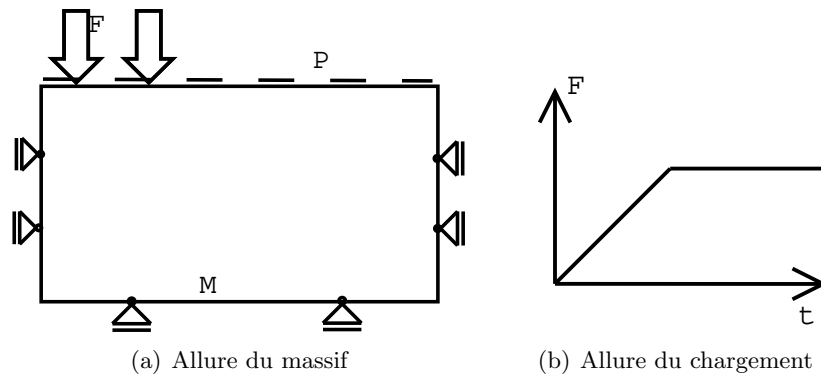


Figure 8.4 — Essai de compactage de sol

Le schéma d'intégration temporelle conduit à la résolution de 20 systèmes linéaires, l'élément retenu est l'hexaèdre  $Q_2 - Q_1$  (20 noeuds en déplacement, 8 noeuds en pression), le problème possède 10000 degrés de liberté et le domaine est décomposé en 3 sous-structures. Le critère de convergence est celui de GMRes fixé à  $10^{-10}$ . La stratégie hybride est la suivante :

- déplacement traité comme un variable duale,
- pression traitée comme une variable primale,

ainsi l'inconnue d'interface duale/primale est homogène à une pression. L'approche primale/duale (inconnue hétérogène déplacement/flux) a également été testée. Le tableau 8.2 résume les performances moyennes par système des différentes approches non préconditionnées ou préconditionnées (meilleurs préconditionneurs et projecteurs).

On y observe que quand elles sont préconditionnées, toutes les approches sont sensiblement équivalentes, ce qui laisse à supposer que le préconditionneur proposé pour l'approche hybride possède certaines propriétés d'optimalité. Quand le système n'est pas préconditionné on observe alors que le respect de la physique par l'approche hybride conduit bien à des performances meilleures : l'approche duale/primale est nettement devant les approches classiques et encore meilleure par rapport à l'approche primale/duale.

Approche	non-préconditionné	préconditionné
primale	286	26
duale	265	28
duale/primale	214	27
primale/duale	420	28

**Tableau 8.2** — Performances de l'approche hybride sur le problème de compactage

La figure 8.5 présente pour le premier système l'évolution du résidu au cours des itérations de GMRes pour le problème non-préconditionné. On y observe que l'allure de la convergence de l'approche duale/primale possède des propriétés intéressantes : au début du processus elle est très similaire à l'approche duale qui décroît rapidement, elle emprunte ensuite un palier avant de retrouver une évolution décroissante rapide similaire à celle de l'approche primale (qui est obtenue beaucoup plus tardivement pour cette dernière). L'évolution du résidu étant intimement lié au spectre des opérateurs, l'approche duale/primale emprunte donc de bonnes propriétés aux spectres des deux approches classiques ce qui permet d'obtenir une courbe de convergence remarquable ; à l'opposé l'approche primale/duale récupère de moins bonnes propriétés ce qui conduit à une convergence lente et oscillante.

### 8.3 Conclusions sur l'approche hybride

L'approche hybride de décomposition de domaine est une généralisation des approches primale et duale de décomposition de domaine. Elle permet donc d'incorporer au sein d'un même algorithme (et donc un même programme, voir partie V) ces approches classiques tout en offrant de nouvelles possibilités pour formuler le problème d'interface.

Sur des problèmes académiques il est par exemple possible de définir des stratégies numériquement moins performantes mais calculatoirement moins coûteuses que les approches classiques. Par ailleurs, on peut légitimement penser que la baisse des performances numériques est due à la diminution importante de la taille de l'espace grossier issu des déplacements de solide rigide ; une stratégie à développer peut être alors de réinjecter une partie de l'information perdue sous forme de contraintes optionnelles pour espérer retrouver

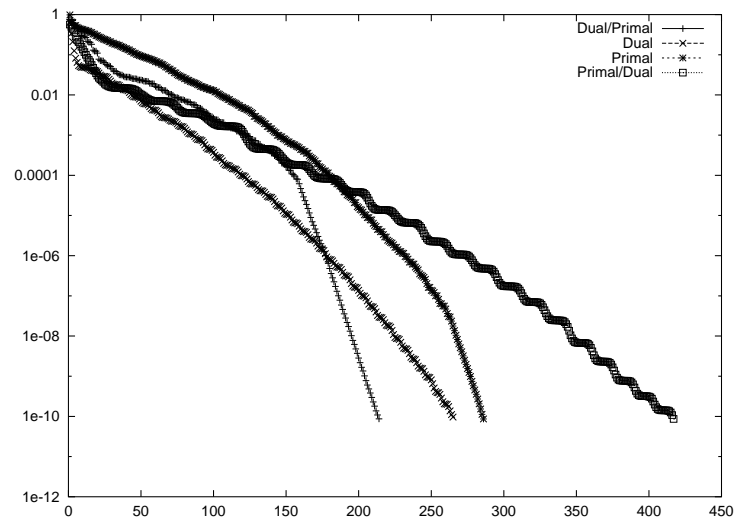


Figure 8.5 — Evolution du résidu GMRes

l'optimalité numérique des approches classiques.

Sur des problèmes multichamps l'approche hybride permet d'envisager un traitement plus respectueux de la physique des phénomènes ce qui entraîne de meilleures performances numériques. Une idée intéressante à creuser à ce propos peut être le développement de préconditionneurs "allégés" de l'approche hybride basés par exemple sur des découplages entre les phénomènes physiques ou les échelles du problème ; le respect de la physique des phénomènes conjugué à l'utilisation de préconditionneurs efficaces et peu coûteux doit permettre d'obtenir de très bonnes performances de calcul.

Pour l'instant la plus grande faiblesse de l'approche hybride est l'absence de sens mécanique clair du résidu GMRes, notamment suivant la répartition des degrés de liberté duaux et primaux le résidu GMRes ne correspond pas à la convergence des mêmes grandeurs physiques. Une réflexion poussée doit être menée pour évaluer un critère mécanique commun de convergence (si possible à un moindre coût).



Quatrième partie

**Accélération Krylov pour la  
multirésolution**



Dans cette partie, nous quittons les aspects "formulations" qui ont conduit à des problèmes condensés sur l'interface, avec autant que possible les meilleurs initialisation et spectre, pour nous focaliser sur la résolution des systèmes obtenus. Plus précisément, nous nous intéressons au cas où une séquence de systèmes linéaires doit être résolue. Cette séquence peut être issue de l'étude d'une discrétisation temporelle d'un phénomène dynamique (voir l'étude des milieux poreux chapitre 8), la linéarisation d'un problème nonlinéaire (voir la suite de ce chapitre), la recherche d'optimum (par exemple par plan d'expériences), l'étude d'un phénomène aléatoire (par exemple par une méthode de Monte-Carlo).

Dans le cas où la matrice est invariante, les méthodes dites à seconds membres multiples (*multiple right-hand side*), qui sont généralement des méthodes Krylov-augmenté par les espaces de Krylov précédents ont fait leur preuve. Nous nous intéressons ici au cas où la matrice n'est pas invariante, ce qui nous a conduit à définir deux algorithmes d'accélération GIRKS et SRKS. GIRKS a été développé par C. Rey et F. Risler, nous proposons ici de prouver son efficacité dans un cadre industriel sur l'approche primale. SRKS est un algorithme développé par l'auteur et C. Rey.





CHAPITRE

# 9

---

## Cadre de l'étude / algorithme de référence

Dans ce chapitre nous présentons tout d'abord le cadre d'obtention de la séquence de systèmes linéaires : la linéarisation de Newton-Raphson de problèmes complexes nécessitant la mise à jour de la matrice tangente à chaque itération. Dans un second temps nous voyons le principe des algorithmes à seconds membres multiples, et constatons leur inadéquation à des matrices non-invariantes.

## 9.1 Linéarisation de Newton Raphson

Supposons que le problème résultant de la méthode des éléments-finis soit non-linéaire et notons le :

$$\mathcal{F}(x) = 0 \quad (9.1)$$

Le principe des algorithmes de Newton est de construire une séquence de systèmes linéaires dont les solutions convergent vers la solution du problème non-linéaire. Il existe de nombreuses versions des algorithmes de Newton, la plus répandue est celle de Newton-Raphson qui consiste à substituer itérativement l'équation  $\mathcal{F}(x) = 0$  par son développement au premier ordre autour de l'estimation précédente notée  $x^k$  :

$$\mathcal{F}(x^k) + \frac{d\mathcal{F}(x^k)}{dx}(x^{k+1} - x^k) = 0 \quad (9.2)$$

Un schéma classique représentant cette méthode en 1D est rappelé figure 9.1. Cette méthode est connue pour converger rapidement quand elle est bien initialisée. Une extension classique est l'algorithme incrémental qui consiste à définir des "pas de chargement" et à trouver la solution exacte de ces étapes intermédiaires en utilisant la solution de la précédente comme initialisation efficace.

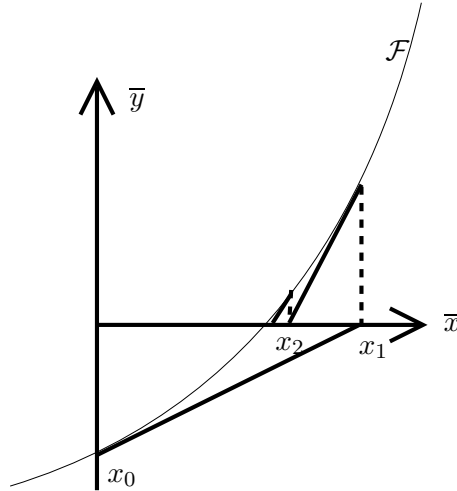


Figure 9.1 — Algorithme Newton-Raphson

Prenons par exemple le cas de l'élastomère évoqué dans le chapitre 5. On a vu que la formulation variationnelle associée est :

$$\left\{ \begin{array}{l} \text{Trouver } (u, p) \in \mathcal{U} \times \mathcal{P} / \forall (v, q) \in \mathcal{U} \times \mathcal{P} \\ \int_{\Omega} \frac{\partial W}{\partial \bar{F}} (I + \bar{\nabla}(\bar{u})) : \bar{\nabla}(\bar{v} - \bar{u}) d\Omega + \int_{\Omega} p h'(J) \frac{\partial J}{\partial \bar{F}} : \bar{\nabla}(\bar{v} - \bar{u}) d\Omega = \\ \int_{\Omega} \bar{g} \cdot (\bar{v} - \bar{u}) d\Omega + \int_{\partial_f \Omega} \bar{f} \cdot (\bar{v} - \bar{u}) dS \\ \int_{\Omega} (h(J) - \frac{1}{K} p) q d\Omega = 0 \end{array} \right. \quad (9.3)$$

Soit  $(\Phi_i)$  et  $(\Psi_\alpha)$  les bases des champs de déplacement et de pression discrétisés, le système résultant de la linéarisation de Newton-Raphson s'écrit :

$$\begin{pmatrix} K_{uu} & K_{up} \\ K_{up}^T & K_{pp} \end{pmatrix} \begin{pmatrix} v^k \\ q^k \end{pmatrix} = \begin{pmatrix} f_u \\ f_p \end{pmatrix} \text{ avec } \begin{cases} v^k = u^{k+1} - u^k \\ q^k = p^{k+1} - p^k \end{cases} \quad (9.4)$$

$$\begin{aligned}
\left(K_{uu}(u^k, p^k)\right)_{ij} &= \int_{\Omega} \left( \frac{\partial^2 \mathcal{W}}{\partial F^2} : \nabla \Phi_i \right) : \nabla \Phi_j d\Omega \\
&\quad + \int_{\Omega} p^k h''(J) \left( \frac{\partial J}{\partial F} : \nabla \Phi_i \right) \left( \frac{\partial J}{\partial F} : \nabla \Phi_j \right) d\Omega \\
&\quad + \int_{\Omega} p^k h'(J) \left( \frac{\partial^2 J}{\partial F^2} : \nabla \Phi_i \right) : \nabla \Phi_j d\Omega \\
\left(K_{up}(u^k, p^k)\right)_{ib} &= \int_{\Omega} h'(J) \left( \frac{\partial J}{\partial F} : \nabla \Phi_i \right) \Psi_b d\Omega \\
\left(K_{pp}(u^k, p^k)\right)_{ab} &= -\frac{1}{K} \int_{\Omega} \Psi_{\alpha} \Psi_{\beta} d\Omega \\
\left(f_u(u^k, p^k)\right)_i &= \int_{\Omega} f \Phi_i d\Omega + \int_{\partial_g \Omega} g \Phi_i dS - \int_{\Omega} \frac{\partial \mathcal{W}}{\partial F} : \nabla \Phi_i d\Omega \\
&\quad - \int_{\Omega} p^k h'(J) \frac{\partial J}{\partial F} : \nabla \Phi_i d\Omega \\
\left(f_p(u^k, p^k)\right)_a &= - \int_{\Omega} \Psi_{\alpha} \left( h(J) - \frac{1}{K} p^k \right) d\Omega
\end{aligned} \tag{9.5}$$

De plus amples informations peuvent être obtenues sur les algorithmes de Newton, notamment appliqués aux élastomères, dans [98].

Dans le cadre des décompositions de domaine, en raison de conditions de Dirichlet insuffisantes ou de mécanismes internes, la matrice de rigidité de certaines sous-structures peut ne pas être inversible; le calcul du noyau est alors un point important. A la connaissance de l'auteur, il n'existe pas de résultat général sur la composition du noyau dans le cadre des matrices tangentes issues de l'algorithme Newton Raphson. Ce qui est sûr pour les sous-structures sans mécanisme est que pour le premier système (élasticité linéarisée) le noyau est composé par les déplacements de solide rigide (et pression nulle), et que pour les systèmes suivants les translations admissibles appartiennent toujours au noyau. L'expérience ne nous a jamais amené à observer d'autres modes à énergie nulle. D'un point de vue pratique, il peut suffire de calculer les modes rigides à la première itération et supprimer les rotations pour les itérations suivantes.

Dans le cadre de notre étude, nous résolvons les systèmes linéaires par décomposition de domaine, une hypothèse fondamentale est que le maillage et la décomposition ne sont pas modifiés d'un système à un autre.

## 9.2 Algorithme Krylov-augmenté en multirésolution

On note la séquence de systèmes linéarisés condensés à résoudre :

$$S^k x^k = b^k \tag{9.6}$$

ces systèmes sont résolus par un solveur itératif de Krylov (voir chapitre 2) qui conduit à la construction de sous-espaces de Krylov  $\mathcal{K}(S^k, z_0^k)$ . On a vu que les solveurs utilisés reposent sur la construction d'une ou plusieurs bases du sous-espace de Krylov vérifiant certaines propriétés. Ainsi pour GMRes, on construit la base orthonormée des  $(v_i^k)$ , pour le gradient conjugué, on construit les bases des  $(z_i^k)$  (résidus préconditionnés) et  $(w_i^k)$  (directions de descente). On adopte pour la suite la notation majuscule :

$$\begin{aligned}
M_i^k &= \left( \dots, m_j^k, \dots \right)_{0 \leq j \leq i-1} \\
M^k &= M_m^k \text{ avec } m \text{ itération où l'algorithme converge}
\end{aligned}$$

On a les propriétés suivantes :

$$V^{kT} V^k = I \quad (9.7)$$

$$V^{kT} \widetilde{S}^{k-1} S^k V^k = H^k \quad (9.8)$$

$$W^{kT} S^k W^k = \Gamma^k \text{ diagonale} \quad (9.9)$$

$$Z^{kT} R^k = \text{diagonale} \quad (9.10)$$

On a vu que l'intégralité de l'information générée au cours d'une résolution par un solveur de Krylov est contenue dans le sous-espace de Krylov (et dans l'éventuelle initialisation). On rappelle ainsi que si on définit l'espace Krylov-augmenté  $\widetilde{\mathcal{K}}_m(\widetilde{S}^{k-1} S^k, z_0^k, C^k) = \mathcal{K}_m(\widetilde{S}^{k-1} S^k, z_0^k) + \text{Im}(C^k)$ , on utilise alors le principe de recherche suivant :

$$\begin{cases} x_m^k \in x_0^k + \widetilde{\mathcal{K}}_m(\widetilde{S}^{k-1} S^k, z_0^k, C^k) \\ r_m^k \perp \widetilde{\mathcal{K}}_m(\widetilde{S}^{k-1} S^k, z_0^k, C^k) \end{cases} \quad (9.11)$$

L'idée des méthodes d'accélération Krylov pour la multirésolution est de réutiliser l'information générée au cours des résolutions précédentes et de l'injecter dans la résolution du système courant pour en accélérer la convergence. L'idée de base est d'utiliser un algorithme Krylov-augmenté par les espaces précédents :

$$\text{Im}(C^k) = \bigoplus_{1 \leq j < k} \mathcal{K}(\widetilde{S}^{j-1} S^j, z_0^j) \quad (9.12)$$

On rappelle que la mise en place de l'algorithme Krylov-augmenté passe par l'inversion de la matrice du problème grossier  $(C^{kT} S^k C^k)$ . Dans le cadre d'une séquence de systèmes à matrices invariantes ( $S^k = S$  et  $\widetilde{S}^{j-1} = \widetilde{S}^{-1}$ ), un choix astucieux des bases représentant les espaces précédents permet d'obtenir des calculs très simples (9.8, 9.9) ce qui conduit aux méthodes classiques à seconds membres multiples [87, 24].

Dans le cadre de matrices non-invariantes, les propriétés de conjugaison vis-à-vis de l'opérateur sont perdues d'un système à l'autre, ce qui rend la réutilisation de l'ensemble des sous-espaces de Krylov beaucoup trop coûteuse en ressources de calcul. Nous proposons donc deux algorithmes originaux d'accélération. Le premier, appelé GIRKS développé par F. Risler et C. Rey [73, 77, 74, 79] peut s'interpréter comme une réutilisation complète des espaces précédents à l'aide d'un solveur approché. Le second, appelé SRKS développé par l'auteur et C. Rey [46, 75] repose sur une idée évoquée dans [76, 78] et consiste à utiliser un algorithme Krylov-augmenté exact en limitant la taille des sous-espaces réutilisés.

CHAPITRE

# 10

---

## Méthode GIRKS

La méthode GIRKS ayant été développée par F. Risler et C. Rey, le travail relatif à cette méthode a consisté à l'incorporer au sein des méthodes primale et duale de décomposition de domaine, et à pratiquer des évaluations de performance sur des problèmes industriels complexes. Dans un premier temps nous rappelons rapidement la méthode et présentons l'algorithme dans le cadre de décompositions de domaine avec sous-structures flottantes puis présentons les performances sur le calcul d'une butée flexible SNECMA-moteurs-de-fusées.

Ces travaux ont fait l'objet d'une publication [49] et de deux communications [75, 1].

## 10.1 Rappel de la méthode

Supposons que  $k > 1$  systèmes linéaires ont été résolus et cherchons à résoudre le  $(k + 1)^{\text{eme}}$ . La méthode GIRKS (*Generalized Iterative Reuse of Krylov Subspaces*) est la combinaison de l'initialisation IRKS (*Iterative Reuse of Krylov Subspaces*) [79] et du préconditionnement GKC (*Generalized Krylov Correction*) [72, 84, 73].

Le principe fondamental est d'approcher les problèmes grossiers issus de l'augmentation par les espaces précédents ( $j < k$ ) :

$$\text{Gradient conjugué} \quad W^{jT} S^{k+1} W^j \approx W^{jT} S^j W^j = \Gamma^j \text{ diagonale} \quad (10.1)$$

$$\text{GMRes} \quad V^{jT} S^{k+1} V^j \approx V^{jT} S^j V^j = H^j \text{ Hessenberg supérieure} \quad (10.2)$$

dans tous les cas la matrice approchante est facile à inverser (pour GMRes, la matrice de Hessenberg est en fait stockée sous-forme de rotations de Givens et d'une matrice triangulaire supérieure, ce qui rend l'inversion également très simple). Désormais, de manière à fixer les idées nous nous intéresserons essentiellement au gradient conjugué, quoique l'application à GMRes ne demande pas plus d'effort que remplacer  $W^j$  par  $V^j$  et  $\Gamma^j$  par  $H^j$  dans les expressions.

Une autre approximation consiste à résoudre des équations en projection sur les espaces de Krylov. Supposons que l'on ait à résoudre :

$$S^j x^j = b^j \quad (10.3)$$

la solution au problème projeté est celle appartenant à l'espace de Krylov rendant l'erreur orthogonale à ce même espace :

$$\begin{aligned} \tilde{x}^j &\in \text{Im}(W^j) \Rightarrow \tilde{x}^j = W^j \alpha \\ S^j \tilde{x}^j - b^j &\perp \text{Im}(W^j) \end{aligned} \quad (10.4)$$

soit :

$$\tilde{x}^j = W^j (W^{jT} S^j W^j)^{-1} W^{jT} b^j \quad (10.5)$$

L'initialisation IRKS est basée sur une approche itérative qui permet d'évaluer à un faible coût une initialisation pertinente vis-à-vis des espaces précédents. Il s'agit de résoudre le système en projection sur les espaces de Krylov précédents en utilisant pour préconditionneur l'inverse des opérateurs antérieurs projetés sur leurs sous-espaces ; ces itérations conduisent rapidement à une stagnation quand le résidu ne peut plus être minimisé sur les espaces précédents. L'algorithme est alors redémarré et augmenté par l'espace généré durant l'initialisation (matrice de projection  $P_\Lambda$  dans l'algorithme IV.1), l'approximation obtenue sert d'initialisation.

L'algorithme GKC consiste à corriger le préconditionneur de manière à vérifier approximativement une condition d'optimalité. Un préconditionneur idéal  $\widehat{S}^{k+1}{}^{-1}$  devrait vérifier pour toute itération  $j$  :

$$S^{k+1} \hat{z}_j^{k+1} = S^{k+1} \widehat{S}^{k+1}{}^{-1} r_j^{k+1} = r_j^{k+1} \quad (10.6)$$

Ce préconditionneur idéal peut s'interpréter comme une correction  $\hat{r}_j^{k+1}$  d'un préconditionneur classique  $\widetilde{S}^{k+1}{}^{-1}$

$$S^{k+1} \hat{z}_j^{k+1} = S^{k+1} \left( \widetilde{S}^{k+1}{}^{-1} r_j^{k+1} + \hat{r}_j^{k+1} \right) = r_j^{k+1} \quad (10.7)$$

Naturellement une telle correction  $\hat{r}_j^{k+1}$  n'est pas réaliste, on peut cependant, à l'aide des espaces de Krylov en proposer une approximation  $\tilde{r}_j^{k+1}$  cohérente vis-à-vis des résolutions précédentes. On remplace l'équation (10.7) par l'équation suivante (on rappelle  $\widetilde{S}^{k+1} r_j^{k+1} = z_j^{k+1}$ ) :

$$\begin{aligned} \forall q \leq k, \quad s_q^{k+1-1} S^q \left( z_j^{k+1} + \tilde{r}_j^q \right) &= r_j^{k+1} \\ \Rightarrow S^q \tilde{r}_j^q &= s_q^{k+1} r_j^{k+1} - S^q z_j^{k+1} \end{aligned} \quad (10.8)$$

$s_q^{k+1}$  est un scalaire introduit pour des raisons d'efficacité numérique (typiquement  $s_q^{k+1} = 1$ ). Le système (10.8) est résolu en projection dans  $\mathcal{K}(\widetilde{S}^{q-1} S^q, z_0^q)$ .

$$\tilde{r}_j^q = W^q \Gamma^{q-1} W^{qT} \left( s_q^{k+1} r_j^{k+1} - S^q z_j^{k+1} \right) \quad (10.9)$$

L'usage de GKC peut donc s'interpréter comme un préconditionnement non-symétrique, la réorthogonalisation totale est alors nécessaire pour pouvoir appliquer un algorithme de type gradient conjugué. Le lecteur pourra trouver de plus amples détails sur GIRKS dans [74].

L'algorithme IV.1 présente GIRKS dans le cadre d'un gradient conjugué projeté. En effet cette configuration recoupe tous les systèmes obtenus par décomposition de domaine présentés.  $x_{0R}$  correspond à une initialisation compatible avec les contraintes de mode rigide (et éventuellement avec les contraintes optionnelles),  $P$  est le projecteur issu de la formulation de décomposition de domaine. Par soucis de lisibilité l'exposant  $(k+1)$  est omis.

---

**Algorithme IV.1** GIRKS pour gradient conjugué projeté

---

**APPROCHE IRKS**

vecteur initial  $x_{00}$

initialisation  $\dot{x}_0 = x_{0R} + P x_{00}$

calculer  $\dot{r}_0 = b - S \dot{x}_0$

**pour**  $j = 0, \dots, p$  / stagnation **faire**

$$\dot{z}_j = P \left[ \sum_{q=1}^k W^q \Gamma^{q-1} W^{qT} \right] \dot{r}_j$$

$$\dot{w}_j = \dot{z}_j + \sum_{i=0}^{j-1} \beta_j^i \dot{w}_i \text{ avec } \dot{w}_0 = \dot{z}_0 \text{ et } \beta_j^i = -(\dot{z}_j, S \dot{w}_i) / (\dot{w}_i, S \dot{w}_i)$$

$$\dot{x}_{j+1} = \dot{x}_j + \dot{\alpha}_j \dot{w}_j \text{ avec } \dot{\alpha}_j = (\dot{r}_j, \dot{z}_j) / (\dot{w}_j, S \dot{w}_j)$$

$$\dot{r}_{j+1} = \dot{r}_j - \dot{\alpha}_j S \dot{w}_j$$

**fin pour**

On pose  $\dot{W} = (\dot{w}_0, \dots, \dot{w}_p)$ ,  $\Lambda = \dot{W}^T S \dot{W}$  matrice diagonale,  $P_\Lambda = I - \dot{W} \Lambda^{-1} \dot{W}^T S$

**CORRECTION GKC**

initialisation  $x_0 = \dot{x}_{p+1}$  et  $r_0 = r_{p+1}$

**pour**  $j = 0, \dots$ , convergence **faire**

$$z_j^0 = \widetilde{S}^{-1} r_j$$

**pour**  $q = 1, \dots, k$  **faire**

$$\tilde{r}_j^q = W^q \Gamma^{q-1} W^{qT} \left( s_q^{k+1} r_j^{k+1} - S^q z_j^{q-1} \right)$$

$$z_j^q = z_j^{q-1} + \tilde{r}_j^q$$

**fin pour**

$$z_j = P_\Lambda P z_j^k$$

$$w_j = z_j + \sum_{i=0}^{j-1} \beta_j^i w_i \text{ avec } w_0 = z_0 \text{ et } \beta_j^i = -(z_j, S w_i) / (w_i, S w_i)$$

$$x_{j+1} = x_j + \alpha_j w_j \text{ avec } \alpha_j = (r_j, z_j) / (w_j, S w_j)$$

$$r_{j+1} = r_j - \alpha_j S w_j$$

**fin pour**

---



## 10.2 Application au calcul de butée flexible par approche primale

Nous présentons ici les performances de l'algorithme GIRKS sur une butée flexible développée par la SNECMA équipant les booster de la fusée Ariane 5 [49, 1]. Cette structure, ses principales caractéristiques mécaniques et les différentes décompositions utilisées sont détaillées dans l'annexe B.

Les calculs ont tous été réalisés sur le SGI ORIGIN 2000 du pôle de calcul Paris Sud. Du fait de sa plus grande stabilité numérique, notamment vis-à-vis de la précision du calcul des modes rigides, l'approche primale de décomposition de domaine a été privilégiée. Nous comparons donc l'approche primale classique, l'approche primale avec GIRKS et enfin une approche séquentielle directe (sans décomposition de domaine). Comme base de comparaison, chaque système linéaire de l'approche séquentielle directe est résolu en 8667s.

### 10.2.1 Problème sans flambage

Cette histoire de chargement nécessite le calcul de 48 systèmes linéaires avec matrices non-invariantes. L'approche séquentielle est accomplie en 115h30min. Les performances parallèles sont données table 10.1.

Problème		temps CPU moy. (s) / sys		nb. it.	Gain
Décomposition	Méthode	Factorisation	Total	moy./sys	Seq./Par.
17a-1r (17 proc.)	Primal	14.5	140.7	164	<b>61.6</b>
17a-1r (17 proc.)	GIRKS	14.5	78	50	<b>111.1</b>
6a-3r (18 proc.)	Primal	22.4	412.8	398	<b>21</b>
6a-3r (18 proc.)	GIRKS	22.4	256.7	183	<b>33.7</b>
3a-6r (18 proc.)	Primal	144	939.2	362	<b>9.2</b>
3a-6r (18 proc.)	GIRKS	144	400.9	120	<b>21.6</b>

Tableau 10.1 — Performances numériques calcul Parallèle/Séquentiel (sans flambage)

Comme on peut le constater, les performances numériques dépendent fortement du choix de la décomposition bien que le nombre de sous-domaines soit pratiquement constant. Deux facteurs expliquent ce phénomène, la nature des hétérogénéités sur l'interface et l'*aspect ratio* des sous-structures (qui implique des calculs locaux plus ou moins longs, voir le temps de factorisation).

GIRKS conduit à des accélérations très significatives, le temps de calcul pour l'approche primale est réduit entre 38% et 58%. GIRKS permet de résoudre le problème non-linéaire jusqu'à 111 fois plus vite que l'approche séquentielle directe avec seulement 17 processeurs.

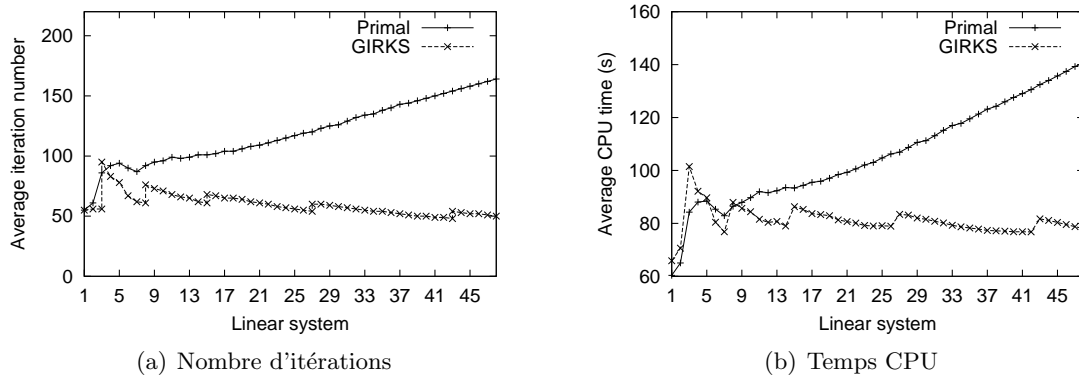


Figure 10.1 — Performances de GIRKS au cours du processus non-linéaire

La figure 10.1 montre l'évolution du nombre moyen d'itérations du gradient conjugué et le temps CPU moyen associé pour résoudre chaque système linéaire. Grâce au très faible coût de GIRKS, les graphes d'itérations et de CPU sont très similaires. Comme on peut l'observer, l'action de GIRKS s'amplifie au fur et à mesure des systèmes résolus, cela correspond à l'accroissement de l'information stockée dans les sous-espaces de Krylov. Au cours du processus non-linéaire, il peut arriver que l'information stockée dans les sous-espaces de Krylov devienne non-significative et conduise à une perturbation entraînant la stagnation. La résolution est alors redémarrée avec remise à zéro de la pile d'espaces de Krylov. La procédure de redémarrage peut être observée figure 10.1 quand deux points sont associés à un même système linéaire.

### 10.2.2 Problème avec flambage

Cette histoire de chargement conduit à la résolution de 37 systèmes linéaires. L'approche séquentielle directe est menée à bien en 89h. Les performances parallèles sont données dans le tableau 10.2. Pour le problème de flambage GIRKS n'est pas aussi efficace que précédemment mais son impact demeure positif. En effet le flambage implique la présence de bifurcations et des changements importants du spectre des matrices tangentes, donc les espaces de Krylov obtenus précédemment ne sont pas en accord avec l'opérateur en cours et ne permettent pas une forte accélération de celui-ci. Les meilleurs résultats donnent une résolution 67 fois plus rapide que l'approche séquentielle avec seulement 17 processeurs.

Problème		temps CPU moy. (s) / sys		nb. it.	Gain
Décomposition	Méthode	Factorisation	Total	moy./sys	Seq./Par.
17a-1r (17 proc.)	Primal	14.9	139	162	<b>62.3</b>
17a-1r (17 proc.)	GIRKS	14.9	130	102	<b>66.7</b>

Tableau 10.2 — Performances numériques calcul Parallèle/Séquentiel (flambage)

## 10.3 Conclusion

Dans ce chapitre nous avons présenté la méthode GIRKS qui peut s'interpréter comme une résolution approximative d'un problème grossier issu de l'augmentation de l'espace de

Krylov du problème en cours par l'ensemble des espaces de Krylov précédents. Les approximations effectuées conduisent à des calculs extrêmement rapides et fortement parallélisables. La méthode a été présentée dans le cadre du gradient conjugué, cependant les bases de sont extension à des solveurs plus généraux tels GMRes ont été évoquées.

La validation de la méthode a été conduite sur le calcul d'une structure composite acier/élastomère combinant une géométrie problématique, de grandes hétérogénéités, de grandes déformations, un comportement incompressible et la présence d'instabilités. Sur le chemin non-flambé les performances de GIRKS sont très intéressantes et conduisent à 50% de gain sur le temps de calcul de l'approche primale classique et une résolution 110 fois plus rapide que l'approche séquentielle. Sur le chemin flambé, la plus grande variation du spectre des opérateurs tangents conduit à des performances moins spectaculaires de GIRKS pour une résolution au mieux 66 fois plus rapide que l'approche séquentielle.

La contre-partie de la résolution très rapide du problème grossier approché réside dans le fait que la "perturbation" induite par GIRKS peut devenir néfaste et entraîner la stagnation d'un système linéaire. La résolution est alors redémarrée avec mise à zéro de la pile des espaces de Krylov. Les gains de GIRKS sont largement suffisants pour couvrir ce léger inconvénient, cependant l'étude *a priori* de la pertinence des espaces de Krylov stockés vis-à-vis de la résolution en cours reste un problème ouvert. Par ailleurs une stratégie de réinitialisation de la pile des espaces de Krylov a pu être développée sur la base de considérations de complexité de calcul [77].

L'approche SRKS développée dans le prochain chapitre ne donne pas une réponse à l'éventuelle perte d'intérêt des espaces précédents vis-à-vis de la résolution en cours, elle permet cependant d'éviter les stagnations par l'usage d'un solveur exact.

Le développement de cette nouvelle méthode est essentiellement issu de la volonté d'éviter les stagnations observables avec GIRKS. Pour cela, l'utilisation d'un solveur exact du problème grossier issu de l'augmentation s'impose clairement. Le problème est alors qu'il devient rapidement trop coûteux en ressources informatiques de réutiliser l'intégralité des sous-espaces de Krylov précédents. Un algorithme de sélection doit être mise en place.

Nous avons vu dans le chapitre 2 que la convergence du processus de résolution est liée à la convergence des valeurs de Ritz. L'idée de l'approche SRKS (*Selective Reuse of Krylov Subspaces*) est donc d'améliorer l'allure du spectre de l'opérateur préconditionné de manière à accélérer la convergence du spectre de Ritz. Pour cela il faudrait idéalement injecter des vecteurs propres dans l'espace augmenté pour éliminer la recherche de la valeur propre associée.

Un calcul de vecteurs propres étant aussi coûteux qu'une résolution nous formulons les hypothèses que les vecteurs propres d'un système sont proches de ceux des systèmes précédents, et que ces vecteurs propres peuvent être correctement approchés par les vecteurs de Ritz les mieux convergés. De manière naturelle notre méthode cible plus précisément les grandes valeurs propres et les valeurs propres négatives ce qui a pour conséquence de filtrer le spectre et de conserver des valeurs petites et regroupées, donnant un conditionnement effectif bas et tendant à enclencher rapidement la superconvergence.

Dans un cas particulier très courant de gradient conjugué, l'algorithme se met en place sans calcul coûteux, conférant à la méthode une grande efficacité. SRKS est validée sur le calcul du flambage d'une poutre fortement hétérogène.

Ces travaux ont fait l'objet d'une publication [46] et d'une communication [75].

## 11.1 Principe de la méthode

L'objectif de SRKS est d'améliorer les propriétés spectrales des problèmes linéaires à résoudre en éliminant le maximum d'éléments propres du spectre actif du problème  $S^k x^k = b^k$ ,  $\widetilde{S}^{k-1}$ -préconditionné. Pour cela on décide d'approcher les vecteurs propres du système  $S^k x^k = b^k$ ,  $\widetilde{S}^{k-1}$ -préconditionné par les vecteurs de Ritz bien convergés des systèmes précédents.

Soit  $Y^{k-1}$  la matrice des vecteurs de Ritz du système  $(k-1)$  et  $\overline{Y}^{k-1}$  une sélection de ces vecteurs bien convergés, on pose alors  $C^k = (C^{k-1}, \overline{Y}^{k-1})$ .

Deux questions restent en suspens : comment calculer le spectre de Ritz et comment évaluer la convergence des vecteurs. Pour ces questions techniques, qui ne concernent qu'un seul système linéaire (ici  $k-1$ ), on omet l'exposant du système.

Le calcul du spectre de Ritz commence par l'utilisation d'une base orthonormale  $N_j$  du sous-espace de Krylov  $\mathcal{K}_j(\widetilde{S}^{-1}S, z_0)$  pour le calcul d'une matrice de Rayleigh  $B_j = N_j^T S N_j$ , la diagonalisation de cette matrice donne le spectre de Ritz  $(\theta_j^i)$  et la matrice de passage  $Q_j^B$ . Soit  $\Theta_j = \text{diag}(\theta_j^i)$ , on a  $B_j = Q_j^{B^T} \Theta_j Q_j^B$  ; les vecteurs de Ritz sont les  $Y_j = Q_j^B K_j$ .

Une idée pertinente est d'utiliser la matrice de Hessenberg  $H_j$  qui est la matrice de Rayleigh associée à la base d'Arnoldi  $V_j$ , car  $H_j$  et  $V_j$  sont toutes les deux accessibles depuis les algorithmes de résolution (directement sous GMRes, par un léger calcul pour le gradient conjugué).

La convergence des vecteurs de Ritz est évaluée en mesurant celle des valeurs de Ritz associées, en effet l'hypothèse est que les valeurs de Ritz ont convergé si elles stagnent d'une itération à l'autre :

$$y_j^i \text{ a convergé si } \left| \frac{\theta_j^i - \theta_{j-1}^{i-1}}{\theta_j^i} \right| \leq \varepsilon \quad (11.1)$$

Les expériences montrent qu'un choix objectif de la valeur du seuil de convergence  $\varepsilon$  peut être réalisé.

Comme les algorithmes de Krylov explorent en premier lieu le haut du spectre (en valeur absolue) et que ces valeurs sont souvent les plus discriminantes, l'analyse menée sur un système convergé donne nécessairement une bonne approximation de cette partie du spectre. Si on suppose que l'opérateur possède un spectre suffisamment proche du précédent, on élimine l'essentiel de l'information que le solveur aurait dû chercher au cours du système précédent.

## 11.2 Cas du gradient conjugué avec préconditionneur positif

Le cas du gradient conjugué avec préconditionneur positif offre la possibilité de développer un algorithme extrêmement efficace pour le calcul des éléments de Ritz. La base de cet algorithme est l'équivalence du gradient conjugué augmenté préconditionné par un opérateur symétrique défini positif avec un gradient conjugué augmenté non préconditionné. En effet si  $\widetilde{S}^{-1}$  est symétrique défini positif alors il peut être factorisé sous forme de Cholevsky  $\widetilde{S} = LL^T$ , et on a alors l'équivalence entre la résolution du système  $Sx = b$ ,  $\widetilde{S}$ -préconditionné et  $C$ -augmenté, avec le système  $\hat{S}\hat{x} = \hat{b}$  non préconditionné  $\hat{C}$ -augmenté. La table 11.1 donne l'ensemble des relations de passage.

A partir de là, on peut réaliser très simplement l'accélération sur le système symétrique

$$\boxed{\begin{aligned}\hat{S} &= L^{-1}SL^{-T} \\ \hat{x}_j &= L^T x_j \\ \hat{b} &= L^{-1}b \\ \hat{C} &= L^T C \\ \hat{r}_j &= L^{-1}r_j \\ \hat{w}_j &= L^T w_j \\ \hat{z}_j &= L^T z_j \\ \hat{\alpha}_j &= \alpha_j \\ \hat{\beta}_j^i &= \beta_j^i\end{aligned}}$$

Tableau 11.1 — Symétrisation du gradient conjugué préconditionné

équivalent. On a vu dans le chapitre 2 équation (2.8) que le calcul de la matrice de Hesseberg peut se faire très simplement sur un gradient conjugué non-préconditionné à partir des coefficients  $\hat{\alpha}_j$  et  $\hat{\beta}_j^{j-1}$ , on dispose donc à chaque itération  $j$ , sans calcul, de la matrice  $\hat{H}_j$  qui est tridiagonale symétrique et de la base orthonormale de l'espace de Krylov  $\tilde{\mathcal{K}}_j(\hat{S}, \hat{r}_0, \hat{C})$  issue de la procédure d'Arnoldi  $\hat{V}_j = \left( \dots, (-1)^i \frac{\hat{r}_i}{\|\hat{r}_i\|}, \dots \right)_{0 \leq i < j}$ ; on rappelle que  $\hat{H}_j = \hat{V}_j^T \hat{S} \hat{V}_j$ . La diagonalisation d'une telle matrice est classique (une routine LAPACK est même dédiée à ce calcul) on a alors  $\hat{H}_j = Q_j^H \Theta_j Q_j^{HT}$  où  $\Theta_j = \text{diag}(\theta_j^i)_{0 \leq i < j}$  est la matrice des valeurs de Ritz et  $Q_j^H$  la matrice passage à la base propre. Les vecteurs de Ritz sont les  $\hat{Y}_j = \hat{V}_j Q_j^H$ .

Maintenant que la technique de calcul est donnée sur le système symétrique, on peut voir que le transport sur le système original se fait très simplement.

Soit  $V_j = \left( \dots, (-1)^i \frac{\hat{z}_i}{(z_i, r_i)}, \dots \right)_{0 \leq i < j}$ ,  $V_i$  est une base  $\tilde{S}$ -orthonormale de  $\mathcal{K}_j(\tilde{S}^{-1}S, z_0, C)$  et on a :

$$\hat{H}_j = \hat{V}_j^T \hat{S} \hat{V}_j = V_j^T S V_j = \text{Tridiag}(\eta_{j-1}, \delta_j, \eta_j) \text{ avec } \eta_j = \frac{\sqrt{\beta_j^{j-1}}}{\alpha_j} \text{ et } \delta_j = \frac{1}{\alpha_j} \frac{\beta_j^{j-2}}{\alpha_{j-1}} \quad (11.2)$$

On définit alors les vecteurs de Ritz transportés  $Y_j$  :

$$Y_j = L^{-T} \hat{Y}_j = V_j Q_j^H \quad (11.3)$$

Ils vérifient les propriétés d'orthonormalité suivantes :

$$Y_j^T S Y_j = \Theta_j \quad (11.4)$$

$$Y_j^T \tilde{S} Y_j = I \quad (11.5)$$

On vérifie aisément que construire la matrice  $C$  par extraction depuis  $Y$  est équivalent à construire  $\hat{C}$  à partir des vecteurs de  $\hat{Y}$  associés.

## 11.3 Application au calcul du flambage d'une poutre composite par approche duale

### 11.3.1 Description du problème

La structure figure 11.1 est une poutre composite encastree-libre. Les matériaux sont modélisés par des potentiels de Saint-Venant–Kirchoff, les fibres sont 1000 fois plus rigides

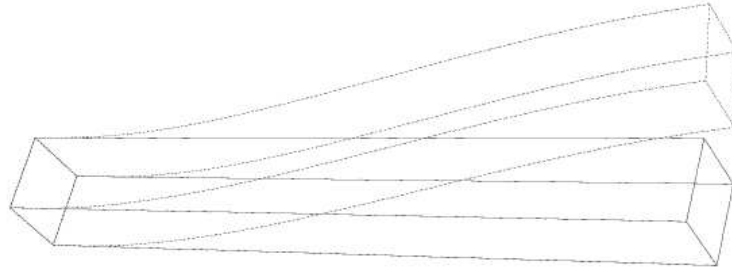


Figure 11.1 — Flambage d'une poutre composite

que la matrice. Une pression constante est imposée sur la partie libre, un léger chargement volumique permet de dissymétriser le problème et enclencher le flambage. La poutre est décomposée en 32 sous-structures parallélépipédiques. Le problème est linéarisé par un algorithme de Newton-Krylov, la résolution est conduite en 28 systèmes linéaires avec matrices non-invariantes. On utilise la méthode FETI-2 équipée du préconditionneur de Dirichlet et du projecteur *superlumped*.

### 11.3.2 Justification du choix de l'approche duale

Le choix de la méthode duale de décomposition de domaine de préférence à l'approche primale est guidé par un soucis d'efficacité de calcul. En effet rien n'empêche d'appliquer SRKS sur l'approche primale cependant l'approche duale possède plusieurs propriétés qui la rendent plus attractive.

Tout d'abord on rappelle que les problèmes grossiers optionnels ("matrice  $C^k$ ") doivent être rendus compatibles avec les contraintes de mode rigide ("matrices  $G^k$  ou  $H^k$ "), voir les algorithmes I.5 et I.6. Cela peut être réalisé en appliquant le projecteur mode rigide sur la matrice de contrainte  $C^{k*} = P^k C^k$ , on rappelle par ailleurs que pour l'approche primale  $H^{kT} S^k P^k = 0$  et pour l'approche duale  $G^{kT} P^k = 0$ . On a vu pour  $k > 1$ ,  $\text{Im}(G^1) \supset \text{Im}(G^2) = \text{Im}(G^k)$  (la même relation s'applique à  $H$ ). Par construction de l'espace de Krylov, on a  $Y^k \perp G^k$  et  $Y^k \perp S^k H^k$ . Pour l'approche duale on a donc  $\forall i < k$ ,  $P^k Y^i = Y^i$  propriété qui n'est pas vérifiée sur l'approche primale car l'orthogonalité est réalisée au sens de l'opérateur qui est modifié d'une itération à l'autre. Ainsi pour l'approche duale l'étape de mise en compatibilité des contraintes vis-à-vis des modes rigides est inutile, ce qui économise les calculs et permet de découpler les problèmes grossiers de mode rigide des problèmes grossiers optionnels issus des sous-espaces de Krylov, ce qui économise beaucoup de calcul notamment quand les espaces grossiers deviennent de taille importante.

Par ailleurs, on observe sans savoir vraiment le justifier que les modes de rotation de l'opérateur de Neumann observés au premier système deviennent des modes négatifs aux systèmes suivants. Cela entraîne que l'opérateur de Neumann n'est plus positif à partir du second système. Or l'algorithme pour le gradient conjugué est valide à condition que le préconditionneur soit positif, on ne peut donc pas utiliser tel quel cet algorithme sur l'approche primale (préconditionneur de Neumann), par contre il fonctionne parfaitement sur l'approche duale (préconditionneur de Dirichlet).

Pour appliquer SRKS sur l'approche primale, puisque l'équivalence avec un problème symétrique est rompue, il faut calculer une matrice de Rayleigh non-symétrique ( $N_j^T S^{-1} S N_j$ ). Dans ce cas, le choix du solveur GMRes qui donne directement accès à la matrice de Hessen-

berg est sans doute plus intéressant.

### 11.3.3 Performances

Le premier point concerne le choix du paramètre  $\varepsilon$  permettant de déterminer si une valeur de Ritz a convergé ou non. Les expériences (voir la figure 11.2) ont montré que le critère est soit très bas ( $< 10^{-14}$ ) soit très haut ( $> 10^{-8}$ ), ce qui laisse une grande plage de choix de  $\varepsilon$  laissant la sélection invariante, typiquement nous avons choisi  $\varepsilon = 10^{-13}$ .

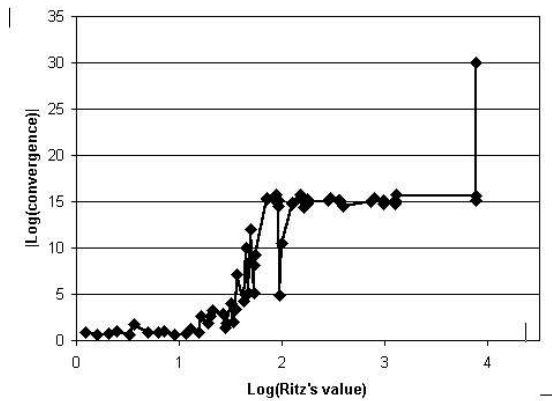


Figure 11.2 — Convergence des valeurs de Ritz

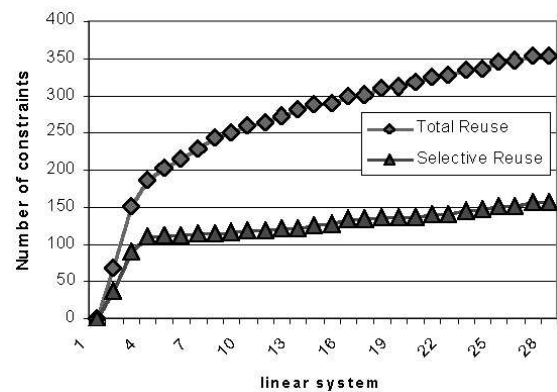


Figure 11.3 — Action de SRKS : nombre de contraintes

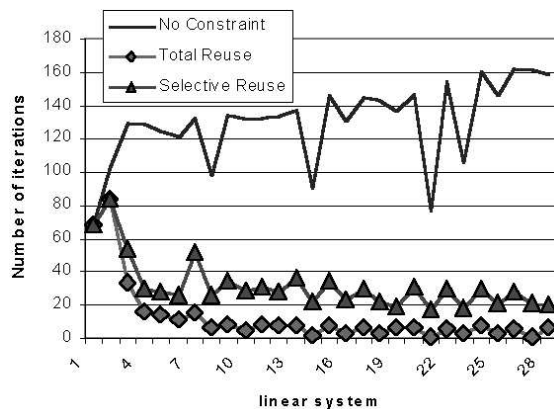


Figure 11.4 — Action de SRKS : nombre d'itérations

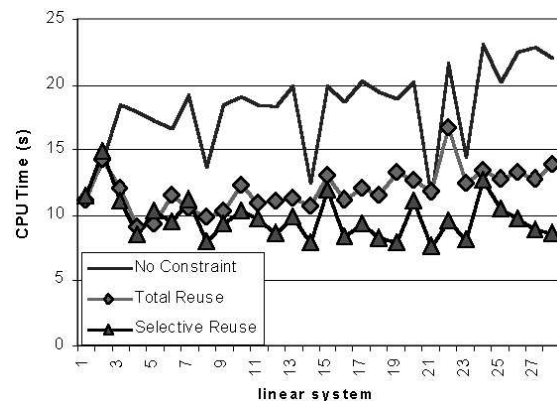


Figure 11.5 — Action de SRKS : temps CPU

Les figures 11.3, 11.4 et 11.5 résument l'action de la réutilisation des sous-espaces de Krylov durant la résolution des systèmes linéaires. Tout d'abord, la figure 11.3 montre l'efficacité de la sélection : le nombre de contraintes est rapidement divisé par un facteur 2. La figure 11.4 présente l'évolution du nombre d'itérations par système linéaire, la réutilisation totale correspond au meilleur résultat possible à partir des sous-espaces de Krylov, le nombre d'itérations est divisé par un facteur 10, ce qui prouve l'intérêt de l'information stockée au sein des sous-espaces de Krylov. La réutilisation sélective montre aussi sa force : avec deux fois moins de contraintes, les performances sont très proches de la réutilisation complète. La figure 11.5 montre les performances en terme de temps CPU : la réutilisation complète est déjà valable, la réutilisation sélective (puisque ses performances en terme d'itérations sont



presque équivalentes avec un nombre de contraintes beaucoup plus faible), conduit à un gain significatif avec une résolution 60% plus rapide que l'approche non accélérée.

### 11.3.4 Etude spectrale

Les figures 11.6 et 11.7 permettent de vérifier l'action spectrale de la réutilisation, elles montrent le spectre de Ritz pour 4 systèmes linéaires (les 1<sup>er</sup>, 5<sup>eme</sup>, 10<sup>eme</sup> et 28<sup>eme</sup>). La réutilisation sélective filtre les plus hautes valeurs propres et les valeurs négatives, tout en supprimant une partie des valeurs centrales, ce qui permet de donner de meilleures propriétés spectrales pour la résolution.

Les figures 11.8 et 11.9 montrent comment le processus de résolution est amélioré par la réutilisation sélective. Deux actions sont combinées : tout d'abord une meilleure initialisation est trouvée ensuite la super-convergence est obtenue dès le début de la résolution.

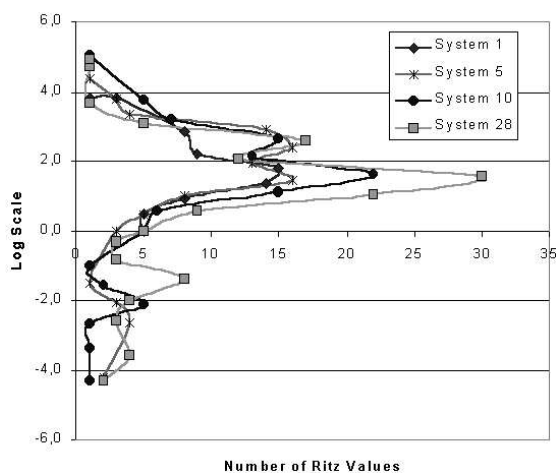


Figure 11.6 — Spectre de Ritz sans contrainte

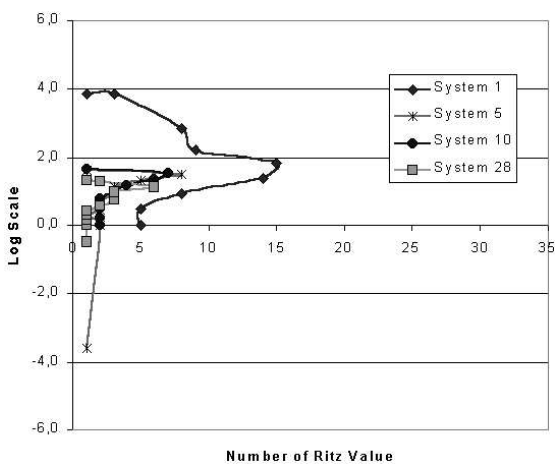


Figure 11.7 — Spectre de Ritz avec réutilisation sélective

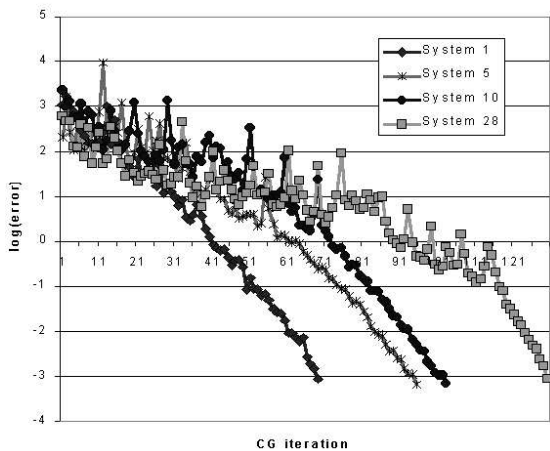


Figure 11.8 — Evolution du résidu sans contrainte

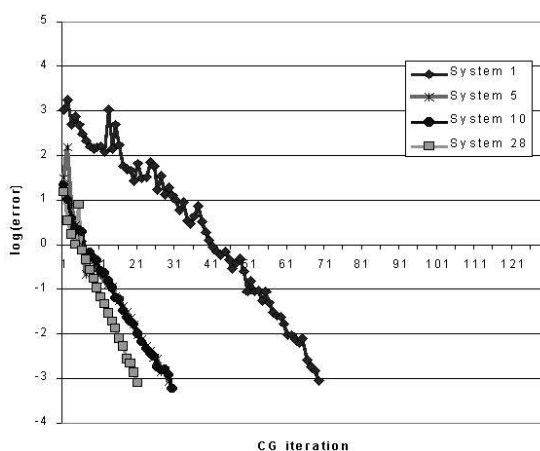


Figure 11.9 — Evolution du résidu avec réutilisation sélective

## 11.4 Conclusion

Le principe de SRKS est donc de réutiliser de manière exacte au sein d'un algorithme Krylov-augmenté l'information générée au cours des résolutions précédentes. Devant la charge de calcul trop importante qui résulterait d'une réutilisation de l'ensemble des espaces précédents, une stratégie de sélection de l'information est mise en place : aux bases naturelles des espaces de Krylov on substitue la base des vecteurs de Ritz et on ne conserve que les vecteurs associés à des valeurs bien convergées. Dans le cadre du gradient conjugué préconditionné par un opérateur symétrique défini positif, une mise en oeuvre particulièrement efficace a été exposée.

L'algorithme de sélection permet de diviser par 2 la taille des espaces stockés, tout en ne diminuant que de moins de 10 points le gain sur le nombre d'itérations par rapport à la réutilisation totale (qui représente un gain d'environ 90%). Au niveau temps CPU on arrive à une accélération de 60% par rapport à une approche duale classique sur un problème fortement hétérogène non-linéaire avec présence d'instabilités géométriques.

Par ailleurs on vérifie par l'étude des spectres de Ritz et l'allure des courbes de résidu que SRKS a bien une action sur le spectre des systèmes en éliminant les valeurs les plus grandes et les valeurs négatives ce qui permet d'enclencher rapidement la superconvergence.

Bien que sans doute moins performant que GIRKS en terme de temps de calcul, la grande stabilité que confère l'usage d'un solveur exact à SRKS, fait de cet algorithme un outil efficace pour l'accélération des séquences de systèmes linéaires. La principale évolution envisagée consiste à évaluer plus finement la pertinence des espaces de Krylov pour concentrer encore plus l'information. La mesure de cette pertinence passe sans doute par la mesure de la variation des spectres actifs d'un système à l'autre le fait de disposer des vecteurs de Ritz rendra cette analyse sans doute plus aisée que pour GIRKS.



## Cinquième partie

Mise en oeuvre orientée objet des  
méthodes et solveurs présentés



Dans cette partie, nous présentons l'état de la réflexion conduite en collaboration avec Christian Rey, Frédéric Feyel et Laurent Séries sur l'organisation de la partie parallèle d'un code éléments-finis basé sur des paradigmes modernes de programmation que l'on rassemble généralement sous la dénomination "langage orienté objet".

Les principales propriétés exploitées sont les notions d'héritage entre classes et d'abstraction de donnée. Grâce à elles, il est possible d'organiser les objets programmés en fonction des données auxquelles ils doivent avoir accès et des méthodes qu'ils doivent définir.

Cette organisation permet d'aboutir à un code très hiérarchisé où chaque objet possède un "cahier des charges" bien défini, ce qui conduit à une programmation efficace et élégante. Il en résulte une grande souplesse d'utilisation, et la possibilité de réutiliser facilement les objets déjà programmés et d'étendre le code assez simplement.

La partie pratique de cette étude a été menée à partir du code ZeBuLoN [66, 65] (NW Numerics, Mines de Paris, ONERA) et a conduit à la première mise en oeuvre de l'approche hybride. Ce code est programmé en C++ [96].

Cette partie repose essentiellement sur l'explication de diagrammes UML représentant les parties les plus significatives du code. Des explications sur la norme UML peuvent être trouvées sur le site <http://www.uml.org>.



Ce chapitre permet de situer les interventions requises pour l'intégration des solveurs et formulations étudiées au sein d'un code préexistant. Pour la réalisation de ce projet nous nous sommes fixé un certain nombre d'objectifs :

- mettre en oeuvre de l'approche hybride ;
- prévoir la possibilité de définir d'autres approches ;
- avoir la possibilité d'utiliser différents solveurs ;
- définir une hiérarchie des classes permettant une réutilisation aussi simple que possible des objets créés ;
- s'intégrer à ZeBuLoN en perturbant le moins possible le code existant.

Tout d'abord ce chapitre décrit comment les données relatives à la topologie de la décomposition de domaine s'organisent, il précise ensuite comment se définit la stratégie de résolution avec notamment la caractérisation de chacun des degrés de liberté de l'interface, enfin il présente comment s'articule la formulation et le solveur de manière à garantir la souplesse d'utilisation voulue.



## 12.1 Organisation des données au sein des sous-domaines

Nous choisissons une organisation où chaque sous-domaine est géré par un processeur (éventuellement virtuel) sans hiérarchie de type maître/esclave. Chaque processeur est le seul à posséder les données relatives au sous-domaine qu'il stocke, une description de l'interface lui permet de repérer ses voisins et quels degrés de liberté il partage avec ceux-ci.

Ces données sont générées lors de l'application du décomposeur de maillage qui à partir d'un maillage d'une structure complète définit les sous-domaines et leur interface. Chaque processeur est alors en possession des données présentées figure 12.1 :

- la classe `DOF` est instanciée pour chaque degré de liberté du sous-domaine et contient entre autres un lien vers les éléments de la sous-structure qui le partagent, son rang dans la numérotation du maillage du sous-domaine et des informations relatives à son éventuel état "fixé" par une condition de Dirichlet.
- la classe `GLOBAL_MATRIX` construite à partir des données des `DOF` contient la matrice de rigidité et le vecteur des efforts généralisés locaux. Cette classe possède des méthodes d'inversion directe du problème éléments-finis.
- la classe `B_DOF` caractérise chaque degré de liberté de l'interface (B pour *boundary*). Un `B_DOF` connaît notamment sa multiplicité, à quels sous-domaines il appartient, quelle est sa position dans l'interface pour ses différentes occurrences.
- la classe `BOUNDARY` permet de positionner ces différents `B_DOF`s sur l'interface, elle contient notamment la liste des voisins et la taille de l'interface associée et un lien vers les `B_DOF` de cette interface.

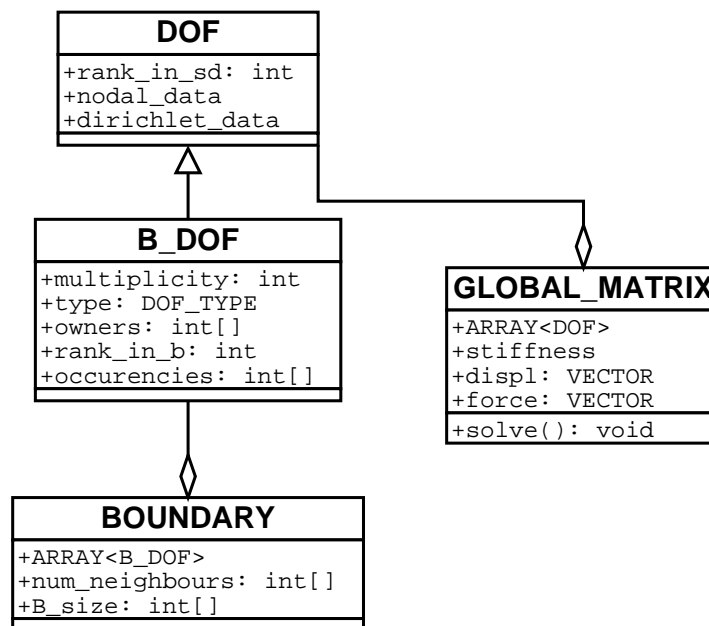


Figure 12.1 — Données issues de la décomposition

On suppose par ailleurs que le code possède un protocole d'échange de données entre les processeurs. On représente l'outil d'échange par une classe `EXCHANGER` possédant essentiellement deux méthodes : envoi (`send`) et réception (`recv`) de données (voir figure 12.2).

Pour les échanges, on doit nécessairement utiliser une description par connectivité de l'interface (les échanges se font toujours deux-à-deux). Classiquement on choisit donc de stocker au niveau local les données de l'interface-connectivité, ce qui fait notamment apparaître des

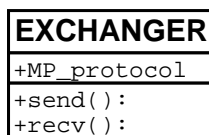


Figure 12.2 — Classe gérant les échanges

redondances au niveau local. Cela conduit à quelques précautions lors de la programmation puisque quand les méthodes ont été présentées, les redondances étaient portées uniquement par l'interface globale.

## 12.2 Définition de la formulation / mise en données

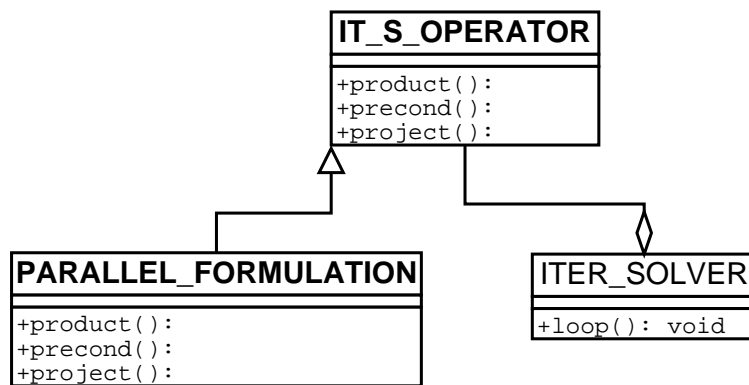
Le paramètre `DOF_TYPE` qui apparaît dans `B_DOF` est fondamental, il est formé par une énumération et permet d'associer un traitement spécifique à chacun des degrés de liberté de l'interface. Pour l'approche hybride de base on a :

```
ENUM DOF_TYPE = (PRIMAL, DUAL)
```

La mise en donnée est réalisée sous la forme d'un fichier de paramètres (fichier ".inp"). Les paramètres relatifs à nos développements sont les suivants :

- pour le solveur :
  - le nom du solveur (gradient-conjugué, GMRes, ...);
  - la précision de la résolution;
  - le nombre minimal/maximal d'itérations;
  - la taille maximale de la base stockée du sous-espace de Krylov.
- pour l'approche hybride :
  - la répartition des degrés de liberté de l'interface en degrés primaux et duaux;
  - le solveur local (Cholevsky, Crout, bisections emboîtées ...);
  - le type de préconditionneur (complet, *lumped*, ...);
  - le type de projecteur (complet, *lumped*, ...);
  - la prise en compte des modes rigides du préconditionneur.

Comme on peut le voir on sépare l'aspect solveur de l'aspect formulation de décomposition de domaine. Cela se traduit par les relations au niveau du code représentées figure 12.3. Dans notre esprit, un solveur itératif est essentiellement un algorithme qui enchaîne les opérations de produit matrice/vecteur, préconditionnement, projection, produit scalaire. La formulation "parallèle" sert donc de serveur au solveur itératif, elle permet de définir les méthodes dont il a besoin pour effectuer ses boucles. Le passage de ces méthodes est effectué via la classe `IT_S_OPERATOR`.



*Figure 12.3* — Relation solveur / formulation

CHAPITRE

# 13

---

## Organisation du code

Ce chapitre décrit à proprement parler la partie exploitation de la décomposition du domaine, que l'on a séparé en "formulation parallèle" et "solveur".

Etant client de la formulation, c'est le solveur qui impose les méthodes et les objets qu'il faut définir dans les formulations qui souhaiteront l'exploiter. Nous développons donc dans un premier temps le côté solveur du code puis nous montrons comment organiser les données issues de la décomposition pour surcharger les méthodes du solveur. L'articulation se fait au sein de l'objet `PARALLEL_FORMULATION` qui décrit donc comment se déroule l'approche de décomposition de domaine.

Des possibilités d'extension du code sont évoquées à la fin de ce chapitre.

## 13.1 Solveurs

Nous nous intéressons dans cette section aux solveurs itératifs indépendamment de leur utilisation par les méthodes de décomposition de domaine. Nous proposons l'organisation représentée figure 13.1.

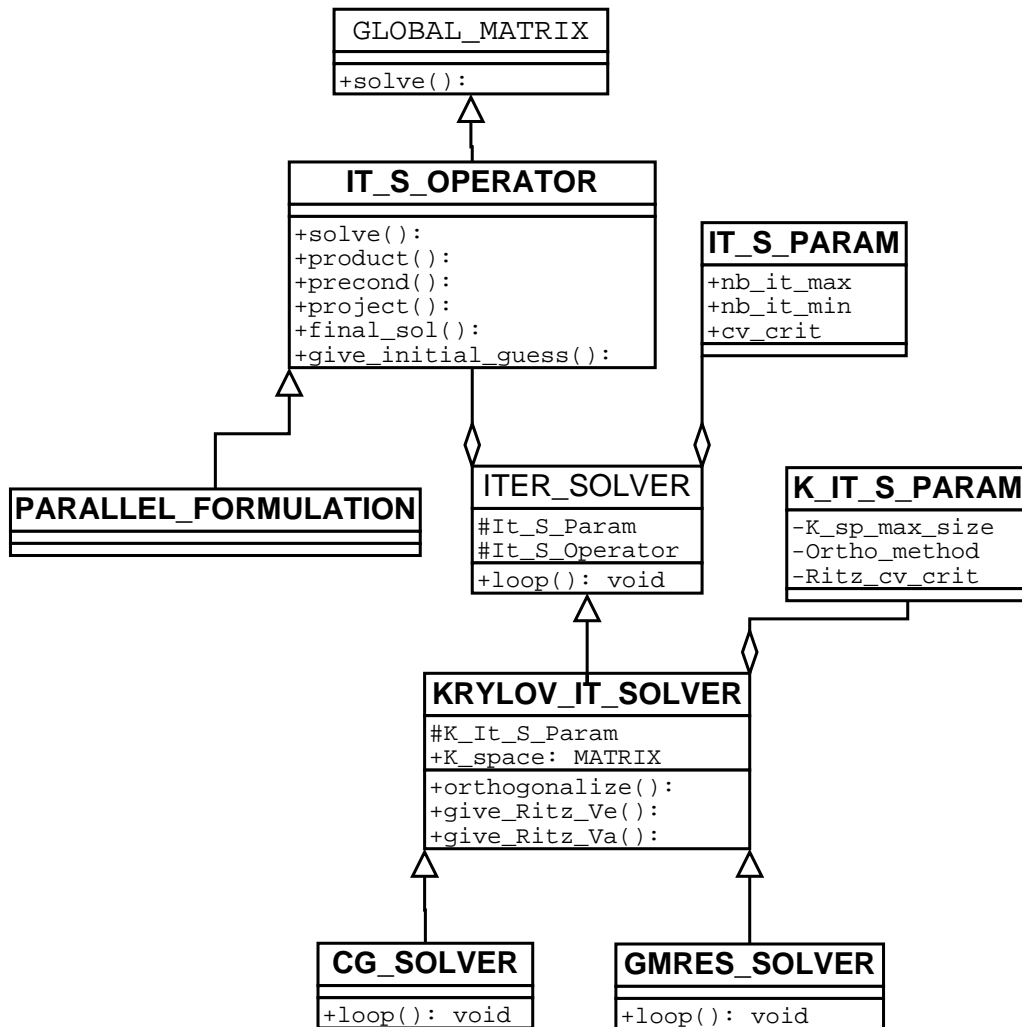


Figure 13.1 — Organisation des solveurs

Les solveurs s'organisent autour de la classe abstraite `ITER_SOLVER` qui contient une méthode `loop()`, cette méthode doit être définie explicitement dans les classes dérivées. De manière à hiérarchiser les solveurs et envisager la mise en oeuvre d'autres classes de solveurs (Jacobi, Gauss-Siedel, relaxation), on intercale la classe abstraite `KRYLOV_IT_SOLVER` qui regroupe les solveurs itératifs de Krylov et dont dérive naturellement le gradient conjugué et GMRes (c'est dans ces objets que la méthode `loop()` est mise en oeuvre). Les paramètres sont transmis via des classes `PARAMETER` dans lesquelles des données issues de la mise en donnée sont stockées. On observe des paramètres génériques comme la précision et le nombre maximal d'itérations et des paramètres spécifiques aux solveurs de Krylov notamment la taille de l'espace stocké pour la réorthogonalisation et la méthode choisie (Gramm-Schmidt classique, modifié ou itératif, Householder). On note également des méthodes et paramètres relatifs au calcul des valeurs et vecteurs de Ritz.

Un solveur itératif nécessite essentiellement de savoir réaliser le produit opérateur/vecteur, le produit préconditionneur/vecteur, éventuellement des projections, une initialisation et le calcul de la solution finale, ces méthodes sont passées via la classe `IT_S_OPERATOR`. Cette classe dérive de `GLOBAL_MATRIX` et redéfinit la méthode `solve()`; toutes les méthodes de `IT_S_OPERATOR` sont donc définies par défaut sur la matrice globale mais elles peuvent être redéfinies par une classe dérivée. C'est sur ce principe que se greffe la formulation parallèle via la classe `PARALLEL_FORMULATION`.

## 13.2 Gestion de l'interface

La gestion de l'interface permet de fournir les objets nécessaires à la classe `PARALLEL_FORMULATION` pour définir ses méthodes. On définit d'une part les vecteurs d'interface (ainsi que des matrices d'interface, *ie* des ensembles de vecteurs) et les opérateurs d'interface qui manipulent ces vecteurs.

### 13.2.1 Vecteurs d'interface

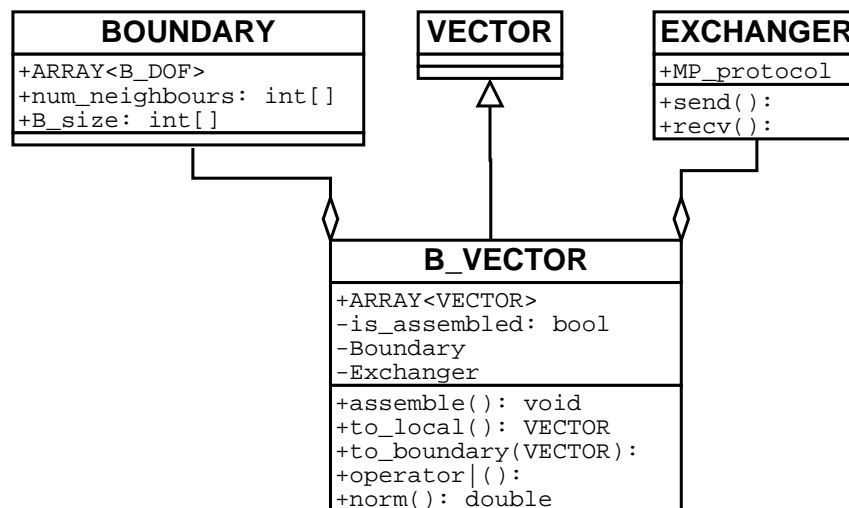


Figure 13.2 — Classe vecteur d'interface

La classe `B_VECTOR` représente les vecteurs d'interface, elle dérive de la classe de base `VECTOR` de manière à pouvoir être interprétée par le compilateur comme un vecteur et donc être manipulée par les solveurs (le mécanisme qui permet de remplacer des vecteurs par des vecteurs d'interface ressort des techniques de programmation et n'est pas présenté dans ce document).

Le `B_VECTOR` stocke les données sous la forme d'un vecteur, il permet également d'y accéder via un tableau de sous-vecteurs chacun relatif à l'interface avec un sous-domaine. Les principales méthodes sont la "trace" `to_boundary()` qui donne les valeurs d'un vecteur d'interface à partir d'un vecteur défini sur l'ensemble du sous-domaine, la "trace transposée" `to_local()` qui crée un vecteur local à partir d'un vecteur d'interface en relevant par zéro les degrés de liberté internes. Comme il a été précisé l'interface locale est stockée voisin par voisin donc avec redondances, il convient alors de porter une attention particulière à la nature des degrés de liberté primaux ou duaux (sur les degrés primaux multiples l'information est

redondante, la "trace transposée" ne doit donc pas cumuler l'information de chacune des occurrences).

Par ailleurs le `B_VECTOR` puisqu'il possède un `EXCHANGER` est capable de réaliser des opérations d'assemblage (échange éventuellement signé pour les degrés de liberté duaux), il redéfinit également le produit scalaire et la norme euclidienne qui nécessitent une attention particulière aux degrés de liberté primaux multiples (il faut éviter de compter plusieurs occurrences d'un point multiple).

### 13.2.2 Opérateurs d'interface

Les opérateurs d'interface manipulent les vecteurs d'interface. Ils définissent principalement les méthodes `B_VECTOR product(B_VECTOR)` et `B_VECTOR t_product(B_VECTOR)` qui associent un vecteur d'interface à un vecteur d'interface. Il est important de noter que ces méthodes ne sous-entendent pas la construction explicite des opérateurs mais uniquement la réalisation d'un produit ; cela explique qu'on ne surcharge pas l'opérateur `*` car il n'existe pas d'opérateur "produit transposé" et on ne peut pas surcharger la méthode `transpose()`.

On distingue parmi les opérateurs d'interface les opérateurs locaux qui ne nécessitent aucun échange entre les sous-domaines et des opérateurs globaux qui impliquent des communications (voir figure 13.3) ; les opérateurs globaux sont traités dans la section suivante.

En ce qui concerne les opérateurs locaux, on considère les compléments de Schur (`SCHUR`), les *scalings* (`SCALING`) et leur combinaison, les compléments de Schur pondérés (`SCALED_SCHUR`).

Un complément de Schur est caractérisé par son `d_type` (`PRIMAL` ou `DUAL`). En fonction de son `d_type` et de celui des degrés de liberté de l'interface, il les traite différemment (traitement primal quand le `d_type` du complément et le type du degré de liberté coïncident, dual sinon). Les différentes approximations du complément de Schur sont déclinées sous forme de classes dérivées (*full*, *lumped*, *superlumped*, *identity*). Dans le cas d'un complément de Schur exact (*full*), il est nécessaire de construire et stocker une `GLOBAL_MATRIX` correspondant à la matrice de rigidité du sous-domaine où les degrés de liberté traités primalement sont pris en compte comme des conditions de Dirichlet, un complément de Schur exact doit également être capable de fournir une solution sur tout le sous-domaine à partir des données sur son interface, c'est le but de la méthode `final_solution()`. Naturellement la `GLOBAL_MATRIX` n'est pas nécessairement inversible, on doit alors calculer et stocker son noyau (membre `kernel`).

Un *scaling* est caractérisé par son `d_type` (`PRIMAL` ou `DUAL`) suivant que l'on ait besoin de calculer  $W^{(s)}$  ou  $\underline{W}^{(s)}$ . De même que pour le complément de Schur, les diverses sortes de *scaling* (topologique, rigidité, module de cisaillement ...) apparaissent comme des classes dérivées.

La classe `SCALED_SCHUR` permet simplement de manipuler l'ensemble formé par un *scaling* et un complément de Schur. On rappelle que cette association est possible au niveau local car l'interface locale est stockée avec redondance (dans les algorithmes algébriques, le *scaling* est toujours associé à un assemblage car l'interface locale n'est pas redondante).

## 13.3 Problèmes grossiers

Les problèmes grossiers sont manipulés à l'aide d'un objet `C_MATRIX` dont l'objectif est d'être manipulé exactement comme les matrices de contraintes  $G$ ,  $H$  et  $C$ . L'information est

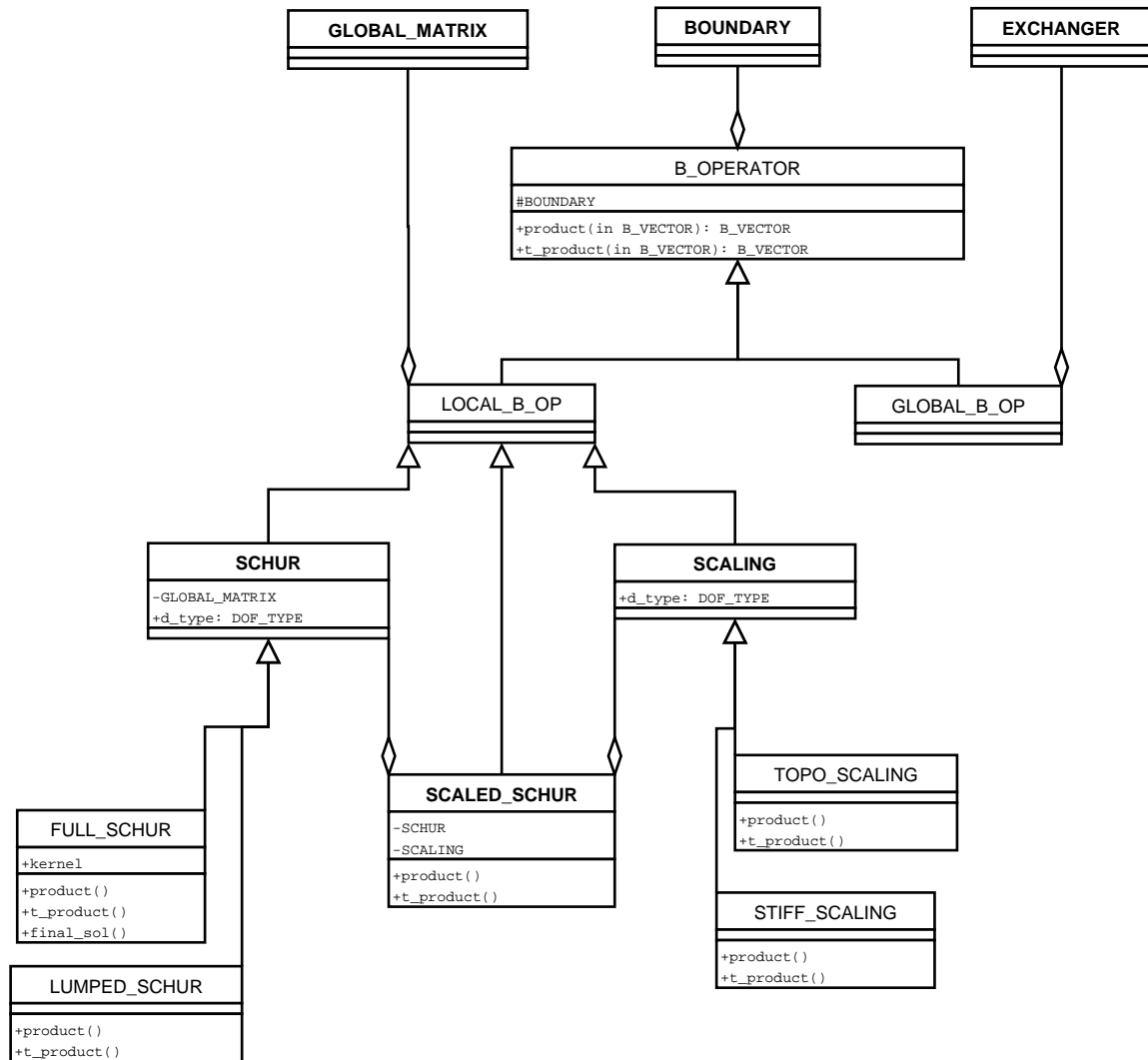


Figure 13.3 — Les opérateurs d'interface

stockée sous la forme d'un tableau de matrices à double entrée. En effet un sous-domaine stocke son information selon ses interfaces avec ses voisins, une première entrée est donc relative au voisin auquel elle se rapporte; l'information de type mode rigide (matrices  $G$  et  $H$ ) est issue de sous-domaines que l'on doit pouvoir identifier, la seconde entrée est donc relative au sous-domaine dont est issue la contrainte considérée, dans le cas de contrainte non-associée à un sous-domaine (par exemple des vecteurs de Ritz) on rajoute un sous-domaine virtuel. Les méthodes mises en oeuvre pour la manipulation de contraintes sont :

- le produit  $C\_MATRIX * VECTOR = B\_VECTOR$ ;
- le produit  $C\_MATRIX^T * B\_VECTOR = VECTOR$ ;
- le produit  $B\_OPERATOR * C\_MATRIX = C\_MATRIX$ ;
- le produit  $C\_MATRIX^T * C\_MATRIX = MATRIX$  de problème grossier (type  $C^TSC$ );
- la concaténation de deux  $C\_MATRIX$ ;
- la suppression des colonnes redondantes dans une  $C\_MATRIX$ .

Une  $C\_MATRIX$  est construite soit à partir du noyau calculé dans un  $FULL\_SCHUR$  soit à partir de données fournies par un mécanisme extérieur (par exemple les vecteurs de Ritz peuvent être transmis par le solveur).



Divers paramètres permettent d'optimiser le stockage et l'utilisation d'une `C_MATRIX` permettant de repérer si les données sont partagées entre les processeurs et si les points multiples répètent la même information ou non.

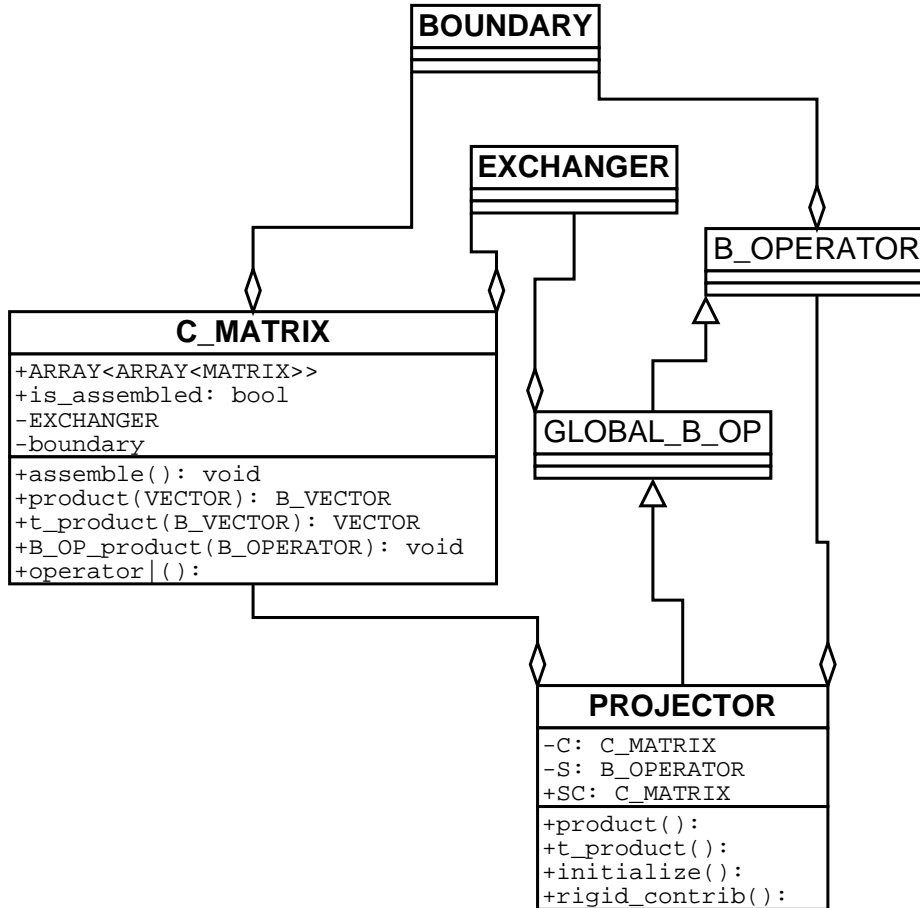


Figure 13.4 — Les problèmes grossiers

A partir d'une `C_MATRIX` et d'un opérateur d'interface `B_OPERATOR`, on peut construire un projecteur. Le principe est de construire en interne la `C_MATRIX` produit de la `C_MATRIX` donnée et de l'opérateur d'interface, il est alors possible de calculer la matrice du problème grossier et de l'inverser. La classe `PROJECTOR` dérive de `GLOBAL_B_OP`, elle définit les méthodes `B_VECTOR` `product(B_VECTOR)` et `B_VECTOR` `t_product(B_VECTOR)` qui agissent et renvoient nécessairement des `B_VECTOR` assemblés.

Un `PROJECTOR` contient également une méthode permettant de rendre une initialisation compatible avec sa contrainte associée. Il contient également une méthode pour calculer l'amplitude des déplacements de solide rigide à rajouter à la solution finale (coefficient  $\alpha$  dans l'approche duale).

## 13.4 L'objet `PARALLEL_FORMULATION`

Comme on l'a vu précédemment, la classe `PARALLEL_FORMULATION` permet de faire la liaison entre l'interface et le solveur choisi en définissant entre autres les méthodes `product()`, `precond()` et `project()`.

A partir de la GLOBAL\_MATRIX on construit un objet FULL\_SCHUR de d\_type PRIMAL correspondant à l'opérateur condensé du problème, et un SCALED\_SCHUR correspondant au préconditionneur, composé d'un complément de Schur de d\_type DUAL éventuellement approché et d'un *scaling* de d\_type PRIMAL.

On détaille ici une stratégie à 3 projecteurs, on construit donc autant de PROJECTOR que de problèmes grossiers (voir chapitre 7).

PARALLEL_FORMULATION
+BOUNDARY +EXCHANGER +op: FULL_SCHUR +preco: SCALED_SCHUR +P_op,P_prec,P_aug: PROJECTOR
+product(): +precond(): +project(): +t_project():

Figure 13.5 — La formulation parallèle

On donne par exemple la structure des trois grandes opérations :

```
PARALLEL_FORMULATION :: product(const VECTOR in,VECTOR out){
out = P_op.t_product((op.product(P_op.product(in))).assemble());}
PARALLEL_FORMULATION :: precond(const VECTOR in,VECTOR out){
out = P_prec.t_product((preco.product(P_prec.product(in))).assemble());}
PARALLEL_FORMULATION :: project(const VECTOR in,VECTOR out){
out = P_aug.product(in);}
PARALLEL_FORMULATION :: t_project(const VECTOR in,VECTOR out){
out = P_aug.t_product(in);}
```

Le calcul du résidu initial (entrée du solveur itératif), ainsi que la valeur captée dans les espaces grossiers de la solution sont réalisés de la manière suivante (on note *ini* une initialisation arbitraire qui sera transformée en initialisation compatible, *b* le second membre condensé, *e* la partie rigide du second membre et *r0* le résidu initial, ces vecteurs sont des membres de PARALLEL\_FORMULATION) :

```
PARALLEL_FORMULATION :: give_initial_guess(){
B_VECTOR vtemp ;
vtemp = P_aug.product(P_prec.product(P_op.product(ini))) ;
ini = vtemp + P_op.compatible_ini(e) ;
r0 = P_op.t_product(b-(op.product(inout)).assemble()) ;
ini += P_prec.compatible_ini2(r0) ;
vtemp = P_prec.t_product(r0) ; r0 = vtemp ;
ini += P_aug.compatible_ini2(r0) ;
vtemp = P_aug.t_product(r0) ; r0 = vtemp ;}
```

Le calcul de la solution finale définie sur l'ensemble du sous-domaine combine une contribution de déformation et une contribution rigide (on note *b\_sol* la solution trouvée sur l'interface, *ini* l'initialisation dans l'espace grossier, *b* le second membre condensé, *f\_sol* la solution sur le sous-domaine).

```
PARALLEL_FORMULATION :: final_solution(){
```

```
B_VECTOR vtemp, rtemp ;
vtemp = P_op.product(b_sol)+ini ; \\ solution complète sur l'interface
rtemp = b - (op.product(vtemp)).assemble() ; \\ résidu
f_sol = op.final_sol(vtemp) + op.kernel*P_op.rigid_contrib(rtemp) ; }
```

## 13.5 Possibilités d'extension

L'organisation de ce code permet donc très simplement de mettre en oeuvre l'approche hybride de décomposition de domaine, résolue par n'importe quel solveur itératif. De plus elle offre un cadre ouvert pour de nombreuses extensions.

La principale extension du code consiste à ajouter de nouveaux traitements des degrés de liberté en augmentant les possibilités de l'énumération `DOF_TYPE`. On peut notamment envisager :

- `PRIMAL_DIRECT` : les degrés de liberté se comportent comme les "coins" de FETI-DP (voir annexe A), ils sont traités primalement puis recondensés pour être éliminés du problème d'interface.
- `DUAL_DIRECT` : sur le même principe que précédemment, les degrés de liberté sont traités dualement puis recondensés pour être éliminés du problème d'interface.
- `DUAL_NON_REDONDANT` : des redondances sont éliminées de la table de connectivité des points multiples, cette stratégie peut admettre diverses versions (notamment assemblage orthonormal et non-redondant [69]).
- `MIXED` : l'inconnue sur les degrés de liberté est une combinaison linéaire du déplacement et de la réaction (condition de Robin). Cette stratégie permet d'introduire des comportements mécaniques complexes à l'interface tels du contact et du frottement.
- `KOSLOV` : dans le cadre d'algorithmes de complétion de données, on peut indiquer une partie de la frontière où les conditions aux limites connues sont incomplètes ou au contraire surnuméraires.

Ces diverses stratégies recourent l'essentiel des formulations de décomposition de domaine sans recouvrement, elles donnent même de nouvelles perspectives d'application des méthodes sur de nouveaux problèmes mécaniques.

L'architecture proposée pour les solveurs paraît convenir à l'ensemble des solveurs itératifs en les regroupant sans pour autant brider leurs capacités (notamment en ce qui concerne l'étude spectrale des opérateurs). Une extension possible consiste à envisager le développement d'une arborescence relative aux solveurs directs.

---

# Conclusion

NOTRE étude a porté sur la simulation numérique de problèmes mécaniques complexes mettant en jeu de grandes hétérogénéités matérielles, des géométries torturées, des formulations de milieux nécessitant l'introduction de plusieurs champ inconnus (essentiellement déplacement/pression), et de fortes non-linéarités (dues à des comportements non-linéaires, à la prise en compte de grandes déformations et la présence d'instabilités géométriques).

L'approche retenue est la combinaison des méthodes de Newton pour la linéarisation des problèmes et les approches Schur–Krylov pour la résolution des systèmes linéaires. Dans ce cadre nous avons été amené à étudier les approches classiques de décomposition de domaine sans recouvrement que sont l'approche primale de décomposition de domaine et l'approche duale ainsi que les solveurs de Krylov dont les plus courants sont le gradient conjugué et GMRes.

De la comparaison des approches, conceptuellement très proches, primale et duale de décomposition de domaine, dans le cadre de problèmes fortement hétérogènes, nous avons constaté, expliqué et pallié une faiblesse de l'approche duale en proposant une meilleure initialisation du problème d'interface basée sur une meilleure évaluation des flux d'efforts au sein des sous-structures. Dans le cadre de l'étude des élastomères dont la prise en compte de la quasi-incompressibilité conduit à l'introduction d'une inconnue de pression éventuellement condensée au niveau élémentaire, nous avons initié une étude des problèmes multichamps. Dans un premier temps nous nous sommes focalisés sur le cas où la seconde inconnue n'était pas continue aux interfaces entre les sous-structures et nous avons montré que les préconditionneurs classiques nécessitaient une amélioration pour prendre en compte correctement la rigidité des interfaces. Nos travaux ont donc permis d'une part de restaurer des performances équivalentes des approches primale et duale sur une plus grande classe de problèmes, et d'étendre leurs bonnes propriétés de convergence sur certains problèmes multichamps.

L'étape suivante de notre étude, était naturellement l'extension aux problèmes multichamps où la seconde inconnue devait être continue d'un sous-domaine à l'autre. Le cadre de cette étude nous a été donné par l'étude des milieux poreux (formulation en déplacement/pression). De manière à mieux respecter la nature physique des inconnues en jeu, nous avons proposé une approche originale de décomposition de domaine appelée approche hybride. Cette approche constitue une généralisation des approches primale et duale classiques (incluant leurs préconditionneurs et leurs problèmes grossiers), elle consiste à individualiser le traitement des degrés de liberté de l'interface. Les premières validations de cette méthode sont prometteuses : résultats efficaces en temps de calcul pour les problèmes d'élasticité linéaire, meilleures propriétés spectrales des opérateurs dans le cadre des problèmes multichamps.

La plupart des problèmes abordés induisent une multirésolution c'est à dire la résolution d'une séquence de systèmes linéaires issue dans les cas évoqués plus haut d'un processus de linéarisation ou d'une discrétisation temporelle, on pourrait également évoquer les problèmes d'optimisation (résolution de systèmes linéaires où les propriétés de la structure évoluent).

Nous avons appliqué l'approche GIRKS aux approches primale et duale de décomposition de domaine avec sous-structures flottantes et validé cette dernière sur la simulation d'une butée flexible SNECMA-moteurs-de-fusées. Les résultats obtenus sont impressionnant avec dans le meilleur des cas une accélération de 60% par rapport à la même méthode de décomposition avec un solveur classique et une résolution 110 fois plus rapide qu'une approche séquentielle directe (en utilisant seulement 17 processeurs). Par ailleurs nous avons développé et validé l'approche SRKS qui consiste à sélectionner la partie la plus significative de l'information générée au cours des résolutions précédentes pour accélérer à l'aide d'un algorithme Krylov-augmenté exact la résolution du système courant. Sur un problème de flambage de poutre composite on obtient une accélération de 60% par rapport à une approche de décomposition de domaine avec solveur classique.

Enfin, nous avons étudié la mise en oeuvre de l'ensemble des méthodes et solveurs proposés au sein d'un code de calcul orienté objet. La base retenue est le code ZeBuLoN, et nous avons montré comment hiérarchiser les différentes classes pour obtenir un code efficace et extensible à l'essentiel des méthodes de décomposition de domaine sans recouvrement et des solveurs itératifs.

Les perspectives de ces travaux sont nombreuses. Du point de vue formulation de décomposition de domaine, une vaste campagne d'essais de l'approche hybride sur des problèmes d'élasticité classique et sur des problèmes multichamps doit être mise en place afin de dégager des stratégies pertinentes d'utilisation. Parallèlement la réintroduction de l'information relative aux modes rigides perdus lors de l'hybridation à l'aide de problèmes grossiers auxiliaires doit être envisagée, on doit également s'intéresser à la définition de préconditionneurs allégés pour les problèmes multiphysiques basés sur des découplages entre les phénomènes physiques ou les échelles de la structure (échelles de l'élément, de la sous-structure, des problèmes de continuité grossière). La recherche d'un résultat théorique sur le taux de convergence de l'approche hybride est également un problème ouvert nécessitant le développement de nouveaux outils pour l'analyse des performances des décompositions de domaine.

Evoquée dans la partie relative à la mise en oeuvre objet du code, l'adjonction de nouveaux traitements aux noeuds de l'interface est également une perspective importante. Notamment les reconcondensations permettront d'englober les principales méthodes de calcul intensif par décomposition de domaine (FETI, FETI-DP, Neumann-Neumann). Le comportement mixte des degrés de liberté à l'interface, en cours de mise en oeuvre, permettra de s'intéresser à de nouvelles classes de problèmes mécaniques tels ceux de contact et de frottement aux interfaces.

En ce qui concerne les accélérations Krylov, on peut envisager plusieurs perspectives comme la combinaison des méthodes GIRKS et SRKS ou la mise en oeuvre sur de nouveaux solveurs (nous pensons avoir posé les bases pour une extension rapide des deux méthodes). Le principal projet reste néanmoins d'augmenter encore notre compréhension de la pertinence des espaces de Krylov précédents pour la résolution d'un nouveau système. Le développement d'une mesure *a priori* de cette pertinence (étroitement liée à la modification du spectre des opérateurs d'un système à l'autre) permettra d'éviter les stagnations dans GIRKS et de diminuer encore la taille des espaces employés par SRKS. Enfin une grande évolution pour ces techniques d'accélération consistera à les adapter aux problèmes de multirésolution où la structure des opérateurs est modifiée d'un système à l'autre notamment en raison d'un remaillage ou de la séparation d'éléments due à la propagation d'une fissure.

La combinaison des méthodes Newton-Krylov-Schur offre dès aujourd'hui un cadre efficace

pour le calcul de problèmes non-linéaires multichamps. Au cours de cette thèse nous avons amélioré et étendu les possibilités et les performances de ces méthodes, en essayant d'offrir une vision aussi unifiée que nous le pouvions des approches de décomposition de domaine sans recouvrement et des accélérations Krylov.

Devant l'incroyable évolution des calculateurs dont les performances, entraînées par les progrès techniques constants notamment en micro-gravure de composants, croissent exponentiellement et ne semblent pas être sur le point de stagner, certains prétendent que le développement de méthodes de calcul intensif est rapidement rendu inutile, rattrapé par les performances des supports. A cela nous donnons deux objections, la première est que pour assumer ces progrès techniques, les besoins en calcul croissent pratiquement aussi rapidement que les performances des machines, la seconde est que nos méthodes sont toujours liées à une préoccupation mécanique et qu'elles offrent de nouvelles perspectives pour aborder la modélisation de comportements complexes.



---

# Bibliographie

- [1] K. ACH, P. GOSSELET, C. REY, F. LÉNÉ et P. DASSET, *Approche séquentielles et parallèles pour l'étude du comportement mécanique d'une butée flexible*, dans *Actes du cinquième colloque national en calcul des structures*, vol. II, (pp. 683–690), 2001.
- [2] Y. ACHDOU et Y. A. KUZNETSOV, *Substructuring preconditioners for finite element methods on nonmatching grids*, *East-West J. Numer. Math.*, vol. 3, no. 1, pp. 1–28, 1995.
- [3] Y. ACHDOU, Y. MADAY et O. B. WIDLUND, *Méthode itérative de sous-structuration pour les éléments avec joints*, *C.R. Acad. Sci. Paris*, vol. I, no. 322, pp. 185–190, 1996.
- [4] J. BALL, *Convexity conditions and existence theorems in nonlinear elasticity*, *Arch. Rat. Mech. Anal.*, vol. 63, pp. 337–403, 1977.
- [5] J. BATOZ et G. DHATT, *Modélisation des structures par éléments-finis*, Hermès, 1990.
- [6] M. BHARDWAJ, D. DAY, C. FARHAT, M. LESOINNE, K. PIERSON et D. RIXEN, *Application of the FETI method to ASCII problems : Scalability results on a thousand-processor and discussion of highly heterogeneous problems*, *Int. J. Num. Meth. Eng.*, vol. 47, no. 1-3, pp. 513–536, 2000.
- [7] J.-P. BOUFFLET, P. BREITKOPF, C. DENIS, A. RASSINEUX et M. VAYSSADE, *Renumérotation des éléments-finis pour un solveur parallèle multifrontal*, dans *Actes du 5ème Colloque National en Calcul des Structures*, (pp. 699–708), 2001.
- [8] F. BREZZI et M. FORTIN, *Mixed and hybrid finite element methods*, Springer, 1991.
- [9] F. BREZZI et L. MARINI, *A three-field domain decomposition method*, dans *Proceedings of the sixth international conference on domain decomposition methods*, (pp. 27–34), 1993.
- [10] S. CANTOURNET, *Endommagement et fatigue des élastomères*, Thèse de doctorat, Université P. et M. Curie, 2002.
- [11] A. CHAPMAN et Y. SAAD, *Deflated and augmented Krylov subspace techniques*, *Numer. Linear Algebra Appl.*, 1997.
- [12] P. CIARLET, *The finite element method for elliptic problems*, North Holland, 1979.
- [13] P. CIARLET et G. GEYMONAT, *Sur les lois de comportement en élasticité nonlinéaire*, *CRAS*, vol. 295, no. série II, pp. 423–426, 1982.
- [14] O. COUSSY, *Mécanique des milieux poreux*, Technip, 1990.
- [15] R. DAUTRAY et J. LIONS, *Analyse mathématique et calcul numérique*, Masson, 1987.
- [16] A. DE LA BOURDONNAYE, C. FARHAT, A. MACEDO, F. MAGOULES et F.-X. ROUX, *Advances in Computational Mechanics with High Performance Computing*, chapitre A



- method of finite element tearing and interconnecting for the Helmholtz problem, (pp. 41–54), Civil-Comp Press, Edinburgh, United Kingdom, 1998.
- [17] G. DHATT et G. TOUZOT, *Une présentation de la méthode des éléments-finis*, Maloine, 1984.
- [18] Z. DOSTAL, *Conjugate gradient method with preconditioning by projector*, Int. J. Comput. Math., vol. 23, pp. 315–323, 1988.
- [19] D. DUREISSEIX et C. FARHAT, *A numerically scalable domain decomposition method for the solution of frictionless contact problems*, Internat. J. Num. Meth. Engin., vol. 50, no. 12, pp. 2643–2666, 2001.
- [20] D. DUREISSEIX, P. LADEVÈZE et B. SCHREFLER, *A LaTin computational strategy for multiphysics problems - application to poroelasticity*, Int. J. Numer. Meth. Engng., vol. 56, no. 10, pp. 1489–1510, 2003.
- [21] G. DUVAUT, *Mécanique des milieux continus*, Masson, 1990.
- [22] G. DUVAUT et J. LIONS, *Inequalities in mechanics and physics*, Springer Berlin, 1976.
- [23] J. ERHEL, K. BURRAGE et B. POHL, *Restarted GMRes preconditioned by deflation*, J. Comput. Appl. Math., vol. 69, pp. 303–318, 1996.
- [24] J. ERHEL et F. GUYOMARC'H, *An augmented conjugate gradient method for solving consecutive symmetric positive definite linear systems*, SIAM J. Matrix Anal. Appl., vol. 21, no. 4, pp. 1279–1299, 2000.
- [25] C. FARHAT, *A saddle-point principle domain decomposition method for the solution of solid mechanics problems*, dans *Domain Decomposition Methods for Partial Differential Equations* (édité par D. KEYES, T. CHAN, G. MEURANT, J. SCROGGS et R. VOIGT), (pp. 271–292), 1992.
- [26] C. FARHAT, P. CHEN et F. RISLER, *A simple and unified framework for accelerating the convergence of iterative substructuring methods with Lagrange multipliers : Application to the design of new FETI coarse problems*, Rapport technique CU-CAS-96-26, Center for Aerospace Structures, CU, November 1996.
- [27] C. FARHAT, P. CHEN et F. RISLER, *A unified framework for accelerating the convergence of iterative substructuring methods with Lagrange multipliers*, Int. J. Numer. Meth. Engng., vol. 42, pp. 257–288, 1998.
- [28] C. FARHAT, P.-S. CHEN et F.-X. ROUX, *The two-level FETI method - part II : Extension to shell problems. parallel implementation and performance results*, J. Comp Meth. Appl. Mech. Eng., vol. 155, pp. 153–180, 1998.
- [29] C. FARHAT et M. GÉRADIN, *On the computation of the null space and generalized inverses of large matrix, and the zero energy modes of a structure*, Rapport technique CU-CAS-96-15, Center for aerospace structures, may 1996.
- [30] C. FARHAT, M. LESOINNE, P. LETALLEC, K. PIERSON et D. RIXEN, *FETI-DP : a dual-primal unified FETI method - part i : a faster alternative to the two-level FETI method*, Int. J. Num. Meth. Eng., vol. 50, no. 7, pp. 1523–1544, 2001.
- [31] C. FARHAT, M. LESOINNE et K. PIERSON, *A scalable dual-primal domain decomposition method*, Numer. Linear Algebra Appl., vol. 7, no. 7-8, pp. 687–714, 2000.
- [32] C. FARHAT, A. MACEDO et M. LESOINNE, *A two-level domain decomposition method for the iterative solution of high frequency exterior Helmholtz problems*, Numerische Mathematik, vol. 85, pp. 283–308, 2000.

- [33] C. FARHAT, A. MACEDO et R. TEZAUER, *FETI-H : a scalable domaine decomposition method for high frequency exterior Helmholtz problems*, dans *Domain decomposition methods in science and engineering*, Domain decomposition press, 1999.
- [34] C. FARHAT et J. MANDEL, *The two-level FETI method for static and dynamic plate problems - part I : An optimal iterative solver for biharmonic systems*, *J. Comp Meth. Appl. Mech. Eng.*, vol. 155, pp. 129–152, 1998.
- [35] C. FARHAT, J. MANDEL et F. ROUX, *Optimal convergence properties of the FETI domain decomposition method*, *Comp. Meth. Appl. Mech. Eng.*, vol. 115, pp. 365–385, 1994.
- [36] C. FARHAT, K. PIERSON et M. LESOINNE, *The second generation FETI methods and their application to the parallel solution of large-scale linear and geometrically non-linear structural analysis problems*, *J. Comp Meth. Appl. Mech. Eng.*, vol. 184, no. 2-4, pp. 333–374, 2000.
- [37] C. FARHAT et F.-X. ROUX, *A method of finite tearing and interconnecting and its parallel solution algorithm*, *Int. J. Num. Meth. Eng.*, vol. 32, pp. 1205–1227, 1991.
- [38] C. FARHAT et F.-X. ROUX, *The dual Schur complement method with well-posed local Neumann problems*, *Contemporary Mathematics*, vol. 157, pp. 193–201, 1994.
- [39] C. FARHAT et F. X. ROUX, *Implicit parallel processing in structural mechanics*, *Computational Mechanics Advances*, vol. 2, no. 1, pp. 1–124, 1994. North-Holland.
- [40] J. FILL et D. FISHKIND, *The moore-penrose generalized inverse for sums of matrices*, *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 2, pp. 629–635, 1999.
- [41] V. FRAYSSÉ, L. GIRAUD et H. KHARRAZ-AROUSHI, *On the influence of the orthogonalization scheme on the parallel performance of GMRes*, Rapport technique, CERFACS, 1998.
- [42] P. GERMAIN, *Mécanique*, Ecole polytechnique, Ellipses, 1986.
- [43] R. GLOWINSKI et P. L. TALLEC, *Augmented lagrangian interpretation of the nonoverlapping Schwartz alternating method*, dans *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*, (pp. 224–231), 1990.
- [44] P. GOLDFELD, *Balancing Neumann-Neumann for (in)compressible linear elasticity and (generalized) Stokes – parallel implementation*, dans *Proceedings of the 14<sup>th</sup> international conference on domain decomposition method*, (pp. 209–216), 2002.
- [45] P. GOSSELET, V. CHIARUTTINI, C. REY et F. FEYEL, *Une approche hybride de décomposition de domaine pour les problèmes multiphysiques : application à la poroélasticité*, dans *Actes du sixième colloque national en calcul de structures*, vol. 2, (pp. 297–304), 2003.
- [46] P. GOSSELET et C. REY, *On a selective reuse of krylov subspaces in newton-krylov approaches for nonlinear elasticity*, dans *Proceedings of the 14<sup>th</sup> conference on domain decomposition methods*, (pp. 419–426), 2002.
- [47] P. GOSSELET et C. REY, *An hybrid domain decomposition*, dans *15th international conference on domain decomposition*, 2003.
- [48] P. GOSSELET et C. REY, *An hybrid domain decomposition*, en préparation, 2003.
- [49] P. GOSSELET, C. REY, P. DASSET et F. LÉNÉ, *A domain decomposition method for quasi incompressible formulations with discontinuous pressure field*, *Revue européenne des éléments finis*, vol. 11, pp. 363–377, 2002.

- [50] P. GOSSELET, C. REY et D. RIXEN, *Etude comparative des méthodes de décomposition de domaine primale et duale : vers une meilleure initialisation de FETI*, dans *Actes du 16eme congrès français de mécanique*, 2003.
- [51] P. GOSSELET, C. REY et D. RIXEN, *On the initial estimate of onterface forces in FETI methods*, *Comp. meth. appl. mech. engrg.*, vol. 192, pp. 2749–2764, 2003.
- [52] A. KLAWONN et O. WIDLUND, *Dual and dual-primal FETI methods for elliptic problems with discontinuous coefficients*, dans *Proceedings of the 12th International Conference on Domain Decomposition Methods, Chiba, Japan, October 1999*. Submitted.
- [53] A. KLAWONN et O. WIDLUND, *FETI and Neumann-Neumann iterative substructuring methods : connections and new results*, *Comm. pure and appl. math.*, vol. LIV, pp. 0057–0090, 2001.
- [54] P. LADEVÈZE, *Nonlinear Computational Structural Mechanics - New Approaches and Non-Incremental Methods of Calculation*, Springer Verlag, 1999.
- [55] P. LADEVÈZE, O. LOISEAU et D. DUREISSEIX, *A micro-macro and parallel computational strategy for highly heterogeneous structures*, *Int. J. Num. Meth. Engnrg.*, vol. 52, pp. 121–138, 2001.
- [56] J. LAMBERT-DIANI et C. REY, *New phenomenological behavior laws for rubbers and thermoplastic elatomers*, *Eur. J. Mech. A/Solids*, vol. 18, pp. 1027–1043, 1999.
- [57] M. LESOINNE et K. PIERSON, *FETI-DP : An efficient, scalable, and unified Dual-Primal FETI method*, dans *Domain Decomposition Methods in Sciences and Engineering*, (pp. 421–428), 1999.
- [58] F. LINGEN, *Efficient Gram-Schmidt orthonormalisation on parallel computers*, *Com. Numer. Meth. Engng.*, vol. 16, pp. 57–66, 2000.
- [59] J. LIONS, *Contrôle optimal de systèmes gouvernés par des équations aux dérivées partielles*, Dunod, 1986.
- [60] J. MANDEL, *Balancing domain decomposition*, *Comm. Appl. Num. Meth. Engng.*, vol. 9, pp. 233–241, 1993.
- [61] J. MANDEL et M. BREZINA, *Balancing domain decomposition for problems with large jumps in coefficients*, *Math. Comp.*, vol. 65, no. 216, pp. 1387–1401, 1996.
- [62] J. MANDEL et R. TEZAU, *Convergence of a substructuring method with Lagrange multipliers*, *Numerische Mathematik*, vol. 73, pp. 473–487, 1996.
- [63] J. MANDEL et R. TEZAU, *On the convergence of a dual-primal substructuring method*, UCD/CCM Report 150, Center for Computational Mathematics, University of Colorado at Denver, April 2000. To appear in *Numer. Math.*
- [64] R. MATTEAZZI, B. SCHREFLER et B. VITALIANI, *Comparaison of partitioned solution procedures for transient coupled problems in sequential and parallel processing*, *Adv. Comput. Struct. Tech.*, (pp. 351–357), 1996.
- [65] Northwest Numerics, *Z-set developper manual*, 2001.
- [66] Northwest Numerics, *Z-set user manual*, 2001.
- [67] R. OGDEN, *Large deformation isotropic elasticity. on the correlation of theory and experiment for incompressible rubber-like solids*, dans *Proceed. Roy. Soc. London*, vol. A328, (pp. 567–583), 1972.
- [68] C. PAIGE, *Approximate solutions and eigenvalue bounds from Krylov subspaces*, *Num. Lin. Algebra Appl.*, vol. 2, no. 2, pp. 115–133, 1995.

- [69] K. PARK, M. JUSTINO et C. FELIPPA, *An algebraically partitioned FETI method for parallel structural analysis : algorithm description*, Int. J. Num. Meth. Eng., vol. 40, no. 15, pp. 2717–2737, 1997.
- [70] K. PARK, M. JUSTINO et C. FELIPPA, *An algebraically partitioned FETI method for parallel structural analysis : performance evaluation*, Int. J. Num. Meth. Eng., vol. 40, no. 15, pp. 2739–2758, 1997.
- [71] L. PAVARINO et O. WIDLUND, *Balancing Neumann-Neumann methods for incompressible Stokes equations*, Comm. Pure Appl. Math., vol. 55, no. 2, pp. 302–335, 2002.
- [72] C. REY, *Développement d’algorithmes parallèles de résolution en calcul non-linéaire de structures hétérogènes : application au cas d’une butée acier-élastomère*, Thèse de doctorat, Ecole Normale Supérieure de Cachan, 1994.
- [73] C. REY, *Une technique d’accélération de la résolution de problèmes d’élasticité non linéaire par décomposition de domaines*, C.R. Acad. Sci. Paris, vol. 322 Série IIb, pp. 601–606, 1996.
- [74] C. REY, *Approches itératives pour le calcul de structures non linéaires*, Habilitation à diriger des recherches, Université Pierre et Marie Curie, 2000.
- [75] C. REY et P. GOSSELET, *Solution to large nonlinear systems : acceleration strategies based on domain decomposition and reuse of krylov subspaces*, dans *Proceedings of the 6<sup>th</sup> ESAFORM conference on material forming*, 2003.
- [76] C. REY et F. RISLER, *A Rayleigh-Ritz preconditioner for the iterative solution to large scale nonlinear problems*, Numerical Algorithms, vol. 17, pp. 279–311, 1998.
- [77] F. RISLER, *Algorithmes parallèles de résolution de problèmes non linéaire de grande taille. Application à la simulation numérique de structures hyperélastiques*, Thèse de doctorat, Ecole Normale Supérieure de Cachan, 1999.
- [78] F. RISLER et C. REY, *On the reuse of Ritz vectors for the solution to nonlinear elasticity problems by domain decomposition methods*, dans *DD10 Proceedings, Contemporary Mathematics*, vol. 218, (pp. 334–340), 1998.
- [79] F. RISLER et C. REY, *Iterative accelerating algorithms with Krylov subspaces for the solution to large-scale nonlinear problems*, Numerical Algorithms, vol. 23, pp. 1–30, 2000.
- [80] R. RIVLIN et D. SAUNDERS, *Large elastic deformation of isotropic materials. experiments on the deformation of rubber*, Phil. Trans. Roy. Soc., vol. A243, pp. 251–288, 1951.
- [81] D. RIXEN, *Substructuring and dual methods in structural analysis*, Thèse de doctorat, University of Liège, Belgique, 1997.
- [82] D. RIXEN et C. FARHAT, *A simple and efficient extension of a class of substructure based preconditioners to heterogeneous structural mechanics problems*, Int. J. Num. Meth. Eng., vol. 44, no. 4, pp. 489–516, 1999.
- [83] D. RIXEN, C. FARHAT, R. TEZAUER et J. MANDEL, *Theoretical comparison of the FETI and algebraically partitioned FETI methods, and performance comparisons with a direct sparse solver*, Int. J. Num. Meth. Eng., vol. 46, no. 4, pp. 501–534, 1999.
- [84] F.-X. ROUX, *Parallel implementation of domain decomposition method for non-linear elasticity problems*, dans *Workshop on domain-based parallelism and problem decomposition methods in computational science and engineering*, Mineapolis, 1994.
- [85] F.-X. ROUX, *Parallel implementation of direct solution strategies for the coarse grid solvers in 2-level FETI method*, Rapport technique, ONERA, Paris, France, 1997.

- [86] F.-X. ROUX, *Parallel solver based on domain decomposition*, 2000.
- [87] Y. SAAD, *On the Lanczos method for solving symmetric linear systems with several right hand sides*, Math. Comp., vol. 48, pp. 651–662, 1987.
- [88] Y. SAAD, *Analysis of augmented Krylov subspace methods*, SIAM J. Matrix Anal. Appl., vol. 18, no. 2, pp. 435–449, April 1997.
- [89] Y. SAAD, *Iterative methods for sparse linear systems*, PWS Publishing Company, 3rd édition, 2000.
- [90] Y. SAAD et M. H. SCHULTZ, *GMRes : a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Comput., vol. 7, pp. 856–869, 1986.
- [91] J. SALENÇON, *Mécanique des milieux continus*, Ecole polytechnique, Ellipses, 1988.
- [92] L. SERIES, F. FEYEL et F.-X. ROUX, *Une méthode de décomposition de domaine avec deux multiplicateurs de Lagrange*, dans *Actes du 16<sup>ème</sup> congrès français de mécanique*, 2003.
- [93] L. SERIES, F. FEYEL et F.-X. ROUX, *Une méthode de décomposition de domaine avec deux multiplicateurs de Lagrange, application au calcul des structures, cas du contact*, dans *Actes du sixième colloque national en calcul des structures*, vol. III, (pp. 373–380), 2003.
- [94] D. STEFANICA et A. Klawonn, *The FETI method for mortar finite elements*, dans *Proceedings of 11th International Conference on Domain Decomposition Methods*, (pp. 121–129), 1999.
- [95] R. STENBERG et M. SURI, *Mixed hp finite element methods for problem in elasticity and stokes flow*, Numer. Math., vol. 72, pp. 367–389, 1996.
- [96] B. STROUSTRUP, *Le langage C++*, Pearson Education, 2003.
- [97] P. L. TALLEC, *Domain-decomposition methods in computational mechanics*, Computational Mechanics Advances, vol. 1, no. 2, pp. 121–220, 1994. North-Holland.
- [98] P. L. TALLEC, *Handbook of numerical analysis*, vol. III, chapitre Numerical methods for nonlinear three-dimensional elasticity, (pp. 465–622), Elsevier science, 1994.
- [99] P. L. TALLEC, J. MANDEL et M. VIDRASCU, *A Neumann-Neumann domain decomposition algorithm for solving plate and shell problems*, SIAM J. Num. Ana., vol. 35, no. 2, pp. 836–867, April 1998.
- [100] P. L. TALLEC, Y.-H. D. ROECK et M. VIDRASCU, *Domain-decomposition methods for large linearly elliptic three dimensional problems*, J. Comp. Appl. Math., vol. 34, pp. 93–117, 1991. Elsevier Science Publishers, Amsterdam.
- [101] P. L. TALLEC et M. VIDRASCU, *Méthodes de décomposition de domaines en calcul de structures*, dans *Actes du premier colloque national en calcul des structures*, vol. I, (pp. 33–49), 1993.
- [102] P. L. TALLEC et M. VIDRASCU, *Generalized Neumann-Neumann preconditioners for iterative substructuring*, dans *Proceedings of the ninth conference on Domain Decomposition*, June Bergen 1996. To appear.
- [103] A. VAN DER SLUIS et H. CAN DER VORST, *The rate of convergence of conjugate gradients*, Numer. Math., vol. 48, pp. 543–560, 1986.
- [104] A. VAN GENDEREN et N. VAN DER MEIJS, *A frontal computation scheme for the schur algorithm to efficiently solve large boundary-element problems*, dans *Proc. of the CompEuro 92 Conference*, (pp. 568–573), 1992.

- 
- [105] K. WASHIZU, *Variational methods in elasticity and plasticity*, Pergamon, 2nd édition, 1975.
- [106] O. ZIENKIEWICZ et R. TAYLOR, *The finite element method*, Mc Graw-Hill Book Company, 1989.



# Annexes





# A

## FETI-DP

FETI-DP (pour dual primal) a été proposé récemment et a connu un succès assez rapide qui le place aujourd'hui comme une alternative crédible aux deux méthodes classiques [57, 52, 31, 63, 30].

Le principe de FETI-DP est, dans le cadre de l'approche duale, de définir un petit nombre de degrés de liberté de l'interface (les "coins" ou *corners*) sur lesquels un traitement primal est appliqué, garantissant ainsi l'inversibilité des matrices locales de "Neumann" (puisque des conditions de Dirichlet sont imposées aux coins).

Séparons donc les degrés de liberté de l'interface (notés  $b$ ) en degrés duaux  $d$  et en coins  $c$ , et notons par ailleurs  $r$  l'ensemble des degrés de liberté internes et duaux (" $r = d \cup i$ ") :

$$u^{(s)} = \begin{pmatrix} u_i^{(s)} \\ u_d^{(s)} \\ u_c^{(s)} \end{pmatrix} \quad u_d^\diamond = \begin{pmatrix} u_d^{(1)} \\ \vdots \\ u_d^{(N)} \end{pmatrix} \quad u_c^\diamond = \begin{pmatrix} u_c^{(1)} \\ \vdots \\ u_c^{(N)} \end{pmatrix}$$

Soit  $\underline{A}_d^{(s)}$  l'opérateur d'assemblage de connectivité restreint aux degrés de libertés duaux, et  $A_c^{(s)}$  l'opérateur d'assemblage d'équilibre restreint aux coins, on pose :

$$\underline{A}_d^\diamond = \left( \underline{A}_d^{(1)} \quad \dots \quad \underline{A}_d^{(N)} \right) \quad A_c^\diamond = \left( A_c^{(1)} \quad \dots \quad A_c^{(N)} \right)$$

De même on définit la trace restreinte  $t_d^\diamond$  qui passe des degrés de liberté duaux et internes aux degrés de liberté duaux seuls ( $r \rightarrow d$ ). La structure des blocs de matrice de rigidité et du second membre est modifiée :

$$K_{rr}^\diamond = \begin{pmatrix} K_{rr}^{(1)} & & \\ & \ddots & \\ & & K_{rr}^{(N)} \end{pmatrix} \quad K_{cc}^\diamond = \begin{pmatrix} K_{cc}^{(1)} & & \\ & \ddots & \\ & & K_{cc}^{(N)} \end{pmatrix}$$

$$K_{rc}^\diamond = \begin{pmatrix} K_{rc}^{(1)} & & \\ & \ddots & \\ & & K_{rc}^{(s)} \end{pmatrix} \quad f_c^\diamond = \begin{pmatrix} f_c^{(1)} \\ \vdots \\ f_c^{(N)} \end{pmatrix} \quad f_r^\diamond = \begin{pmatrix} f_r^{(1)} \\ \vdots \\ f_r^{(N)} \end{pmatrix}$$

Le choix des coins est fait de manière à garantir que  $K_{rr}^\diamond$  soit inversible, on a alors :

$$\begin{cases} \begin{pmatrix} K_{rr}^\diamond & K_{rc}^\diamond \\ K_{rc}^{\diamond T} & K_{cc}^\diamond \end{pmatrix} \begin{pmatrix} u_r^\diamond \\ u_c^\diamond \end{pmatrix} = \begin{pmatrix} f_r^\diamond \\ f_c^\diamond \end{pmatrix} + \begin{pmatrix} \lambda_r^\diamond \\ \lambda_c^\diamond \end{pmatrix} \\ \underline{A}_d^\diamond t_d^\diamond u_r^\diamond = 0 \\ A_c^\diamond f_c^\diamond = 0 \end{cases} \quad (\text{A.1})$$

Deux inconnues sont introduites,  $\underline{\lambda}_d$  intereffort sur les degrés de liberté duaux et  $\mathbf{u}_c$  déplacement des coins. De manière à garantir la continuité du déplacement sur les coins et l'équilibre sur les degrés duaux, posons

$$\begin{cases} \mathbf{u}_c^\diamond = \mathbf{A}_c^{\diamond T} \mathbf{u}_c \\ \lambda_r^\diamond = t_d^{\diamond T} \underline{\mathbf{A}}_d^{\diamond T} \underline{\lambda}_d \end{cases} \quad (\text{A.2})$$

Une première condensation conduit à :

$$\begin{aligned} \begin{pmatrix} \underline{\mathbf{A}}_d^\diamond & 0 \\ 0 & \mathbf{A}_c^\diamond \end{pmatrix} \begin{pmatrix} t_d^\diamond K_{rr}^{\diamond -1} t_d^{\diamond T} & t_d^\diamond K_{rr}^{\diamond -1} K_{rc}^\diamond \\ K_{rc}^{\diamond T} K_{rr}^{\diamond -1} t_d^{\diamond T} & -(K_{cc}^\diamond - K_{rc}^{\diamond T} K_{rr}^{\diamond -1} K_{rc}^\diamond) \end{pmatrix} \begin{pmatrix} \underline{\mathbf{A}}_d^\diamond & 0 \\ 0 & \mathbf{A}_c^\diamond \end{pmatrix}^T \begin{pmatrix} \underline{\lambda}_d \\ \mathbf{u}_c \end{pmatrix} \\ = \begin{pmatrix} \underline{\mathbf{A}}_d^\diamond & 0 \\ 0 & \mathbf{A}_c^\diamond \end{pmatrix} \begin{pmatrix} K_{rr}^{\diamond -1} f_r^\diamond \\ -(f_c^\diamond - K_{rc}^{\diamond T} K_{rr}^{\diamond -1} f_r^\diamond) \end{pmatrix} \end{aligned} \quad (\text{A.3})$$

ce qui est noté

$$\begin{pmatrix} F_{dd} & F_{dc} \\ F_{dc}^T & -F_{cc} \end{pmatrix} \begin{pmatrix} \underline{\lambda}_d \\ \mathbf{u}_c \end{pmatrix} = \begin{pmatrix} \underline{b}_r \\ -b_c \end{pmatrix} \quad (\text{A.4})$$

Les coins sont alors condensés pour aboutir au système symétrique défini positif suivant :

$$\underline{F} \underline{\lambda}_d = \left[ F_{dd} + F_{dc} F_{cc}^{-1} F_{dc}^T \right] \underline{\lambda}_d = \underline{b}_d - F_{cc}^{-1} F_{dc}^T b_c \quad (\text{A.5})$$

Le calcul de  $F_{cc}$  doit être explicite (pour réaliser l'inversion) cette opération est le problème grossier de FETI-DP.

Le préconditionneur optimal de FETI-DP est basé sur le Dirichlet de FETI ( $M^\diamond = I$  pour le *scaling* topologique,  $M^\diamond = \text{diag } K_{bb}^{\diamond -1}$  pour le *scaling* rigidité) :

$$\underline{\tilde{F}}^{\diamond -1} = \left( \underline{\mathbf{A}}_d^\diamond M^\diamond \underline{\mathbf{A}}_d^{\diamond T} \right)^{-1} \underline{\mathbf{A}}_d^\diamond M^\diamond \underline{\mathbf{A}}_d^\diamond F_p^\diamond \underline{\mathbf{A}}_d^{\diamond T} M^\diamond \underline{\mathbf{A}}_d^{\diamond T} \left( \underline{\mathbf{A}}_d^\diamond M^\diamond \underline{\mathbf{A}}_d^{\diamond T} \right)^{-1} \quad (\text{A.6})$$

où  $F_p^\diamond$  est le complément de Schur primal sur les degrés de liberté  $d$  :

$$F_p^\diamond = K_{dd}^\diamond - K_{id}^{\diamond T} K_{ii}^{\diamond -1} K_{id}^\diamond \quad (\text{A.7})$$

Bien sur les versions *lumped* et *superlumped* du préconditionneur existent également basées sur les mêmes approximations que dans FETI.

Un problème critique dans FETI-DP est le choix des coins. L'objectif est de supprimer les modes rigides en laissant un problème grossier suffisamment petit, tout en essayant d'octroyer de bonnes propriétés spectrales au problème. En 2D, les coins sont pris aux points multiples et aux extrémités des interfaces. En 3D, diverses stratégies ont été proposées, par exemple choisir trois coins par face de manière à maximiser la superficie du triangle ainsi défini.

Naturellement l'adjonction de contraintes optionnelles est également possible via un algorithme Krylov-augmenté.

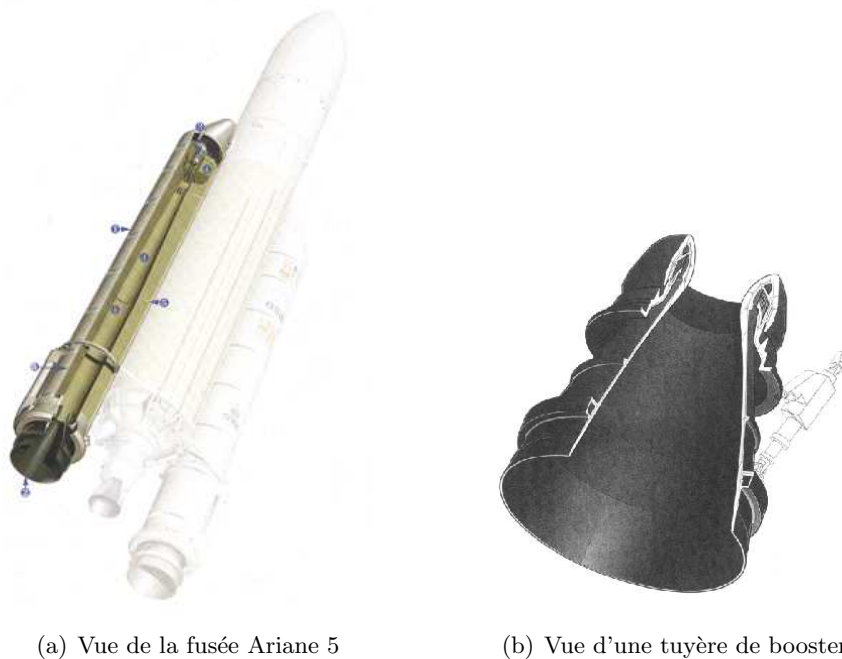
# B

## Butée flexible SNECMA-moteurs-de- fusées

Nous présentons dans cette annexe une butée flexible, conçue par la SNECMA-moteurs-de-fusées, du type de celles équipant les boosters de la fusée Ariane 5. Les objectifs de l'étude étaient tout d'abord de retrouver numériquement les caractéristiques mécaniques de cette structure mesurées lors d'essais expérimentaux, puis de définir des accélérations efficaces pour l'obtention de ces résultats.

Dans un premier temps, nous donnons une description de la structure, puis présentons ses principales caractéristiques mécaniques notamment relatives à l'éventuel flambage d'une armature, enfin diverses décompositions de cette structure sont exposées.

### B.1 Présentation de la butée flexible



*Figure B.1* — Position de la butée flexible

La butée flexible est un lamifié acier-élastomère équipant les boosters d'Ariane 5 permettant la liaison entre le corps du booster et la tuyère (voir figure B.1). On distingue mieux la partie lamifiée sur la figure de la butée après discrétisation éléments-finis, figure B.2 : la couleur claire représente le caoutchouc, le gris foncé l'acier. Les lames sont des calottes sphériques concentriques de quelques millimètres d'épaisseur.

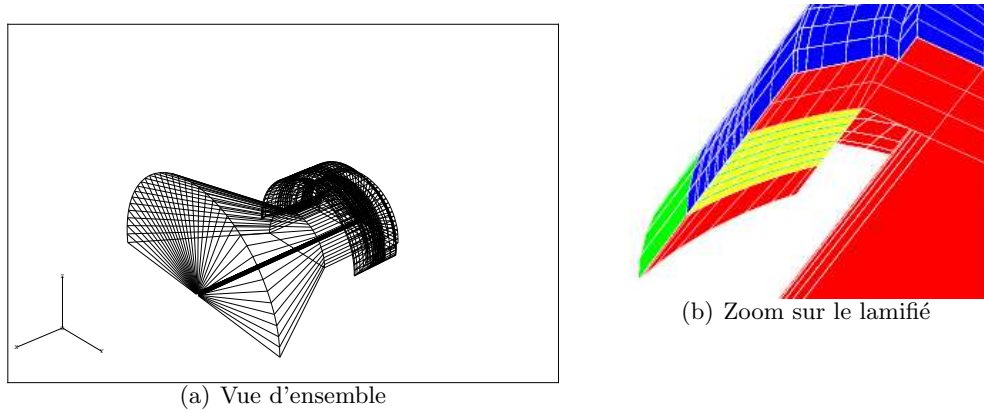


Figure B.2 — Discretisation de la butée

La structure est soumise à un chargement complexe. L'encastrement avec le corps du booster a lieu sur la partie cylindrique de la tuyère (collerette gris clair dans la figure B.2). De par la poussée des gaz permettant l'entraînement de la fusée, le sommet de la tuyère (partie noire dans la figure B.2) est soumis à une pression de 4 MPa. Enfin un vérin (visible sur la figure B.1) permet d'incliner la tuyère de quelques degrés et donc d'orienter la fusée, il se traduit au niveau du modèle éléments-finis par un déplacement imposé dans la direction radiale en un point à l'extrémité de la tuyère.

Du fait de ce chargement complexe, le problème est tridimensionnel, par symétrie on ne s'intéresse qu'à une demi butée flexible. Le maillage est obtenu par rotation de  $180^\circ$  d'une coupe axiale. Ce processus conduit à utiliser des éléments hexaédriques d'allure très déséquilibrée. Etant donné que l'on cherche à obtenir le flambage d'une armature, la discrétisation doit être suffisamment fine pour que la longueur caractéristique du maillage soit inférieure à la longueur d'onde du flambage, typiquement on décompose la rotation en 32 quartiers. Finalement, ce maillage conduit à un problème d'environ 76000 degrés de liberté.

Les modèles utilisés sont, pour l'acier un Saint-Venant-Kirchoff (module d'Young  $E = 2.10^5$  MPa, coefficient de Poisson  $\nu = 0.3$ ), et pour l'élastomère un potentiel élastique Mooney-Rivlin quasi-incompressible ( $C_{10} = 0.2$  MPa,  $C_{01} = 0$  MPa et  $K = 2000$  MPa), voir chapitre 5. Pour un tel problème (voir figure B.3), l'élément  $Q_2 - P_1$  (27 noeuds en déplacement, 4 en pression, noté QC) engendre du blocage à partir de 3 MPa et ne permet pas d'arriver à un chargement de 4 MPa. L'élément  $Q_2 - P_1$  (20 noeuds en déplacement, 4 en pression noté Q2) permet d'obtenir des chargements plus élevés mais ne reproduit pas le comportement mécanique observé lors des essais expérimentaux. L'élément  $Q_2 - Q_1$  (20 noeuds en déplacement, 8 en pression noté IC) permet lui de converger jusqu'au chargement final en s'accordant parfaitement aux mesures expérimentales.

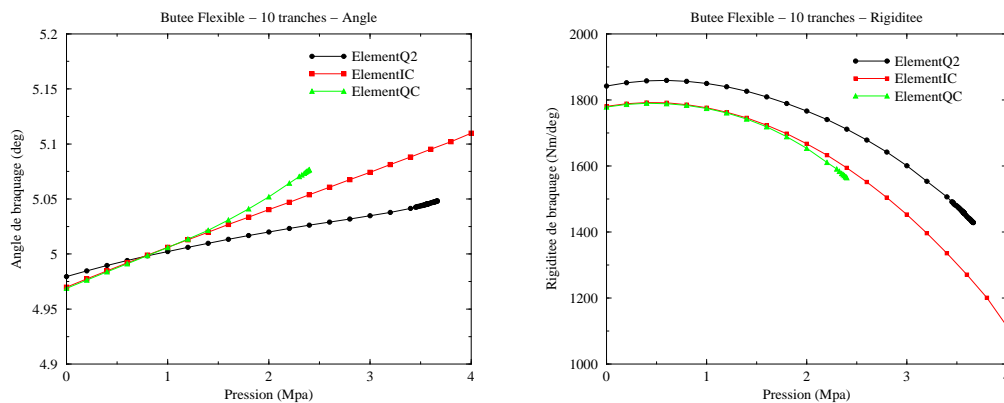


Figure B.3 — Rigidité et angle de braquage

## B.2 Propriétés mécaniques

L'intérêt d'une butée flexible lamifiée est de réaliser la fonction de liaison de manière beaucoup simple et efficace que les solutions antérieures basées sur des mécanismes complexes. Ainsi la butée possède une grande rigidité en compression et une bonne souplesse en flexion.

### B.2.1 Evaluation de la rigidité de braquage

La rigidité de braquage diminue lorsque la pression dans le propulseur augmente, elle peut même devenir négative (la butée devient motrice). Cette évolution, due au déplacement des pièces, est bien prévue par les calculs (voir figure B.5).

### B.2.2 Evaluation des phénomènes liés au braquage

Le braquage peut entraîner un doublement des contraintes dans les armatures, et sous une pression de 3 MPa, la déformation circonférentielle maximale passe de  $-0,33\%$  à  $-0,54\%$  quand l'inclinaison varie de 0 à 5 degrés.

### B.2.3 Evaluation du flambage des armatures

Le principal problème de la butée flexible est le risque de flambage des armatures d'acier, voir figure B.4.

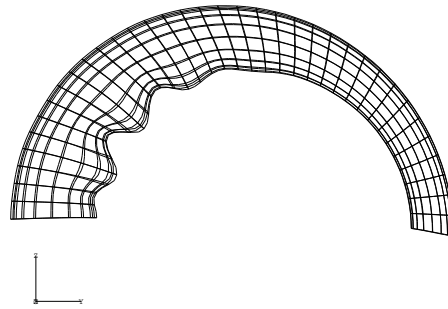


Figure B.4 — Flambage d'une armature en acier

Pour mener à bien la simulation de la butée flexible, il convient d'adapter le solveur de Newton-Raphson incrémental à l'objectif fixé : pour obtenir la configuration non flambée, on choisit un pas de chargement fin et une bonne précision des solveurs (10 incréments, 48 systèmes linéaires) ; pour obtenir la configuration flambée, on relâche la précision et augmente le pas de chargement ce qui permet d'initier numériquement le flambage (4 incréments, 37 systèmes linéaires). La figure B.5 permet d'évaluer l'évolution de deux grandeurs mécaniques suivant la configuration choisie : la rigidité de braquage et le déplacement radial d'un point de l'armature sujette au flambage ; le flambement est calculé à une pression de 3 MPa en bon accord avec un essai, on note une bifurcation très nette des grandeurs mécaniques lors du flambage.

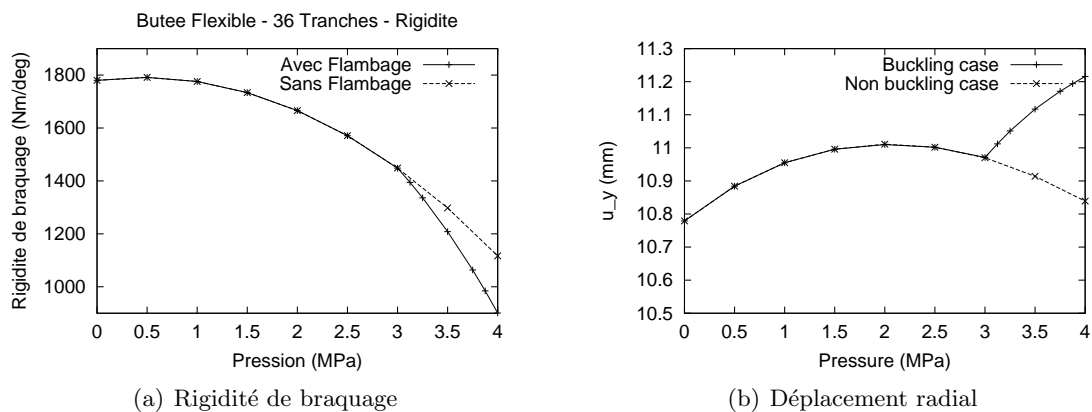


Figure B.5 — Evolution au cours de la compression

### B.3 Décompositions employées

La décomposition de la butée est réalisée par combinaison de deux types de découpe : les découpes "axiales" en calottes sphériques (l'interface sépare le caoutchouc de l'acier) et les découpes "en rotation" (l'interface traverse les hétérogénéités matérielles). Cela conduit à des décompositions axiales pures (voir la coupe axiale figure B.6), des décompositions purement dans la rotation (par exemple 16 quartiers), des décompositions mixtes (voir figure B.7 3 calottes et 6 quartiers soit 18 sous-domaines).

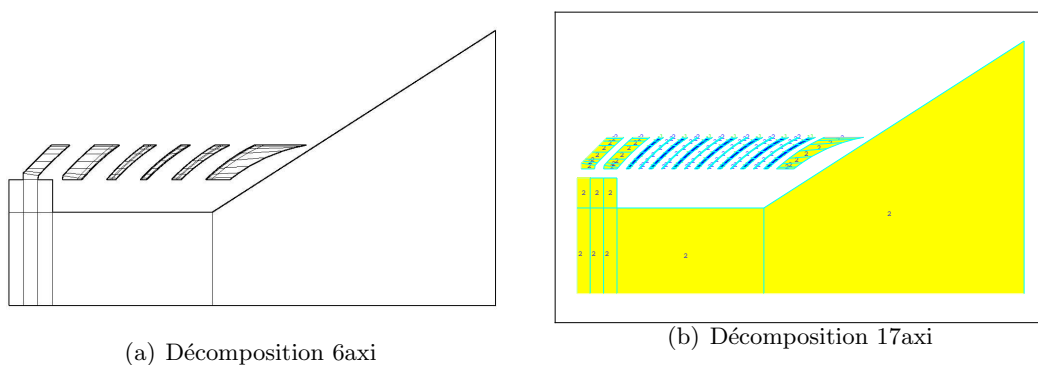


Figure B.6 — Décompositions en calottes sphériques

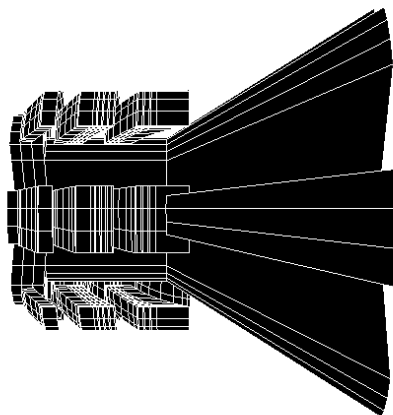
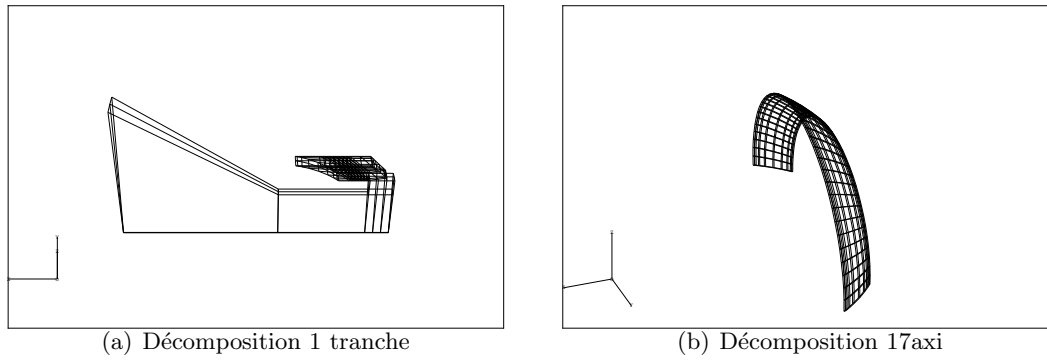


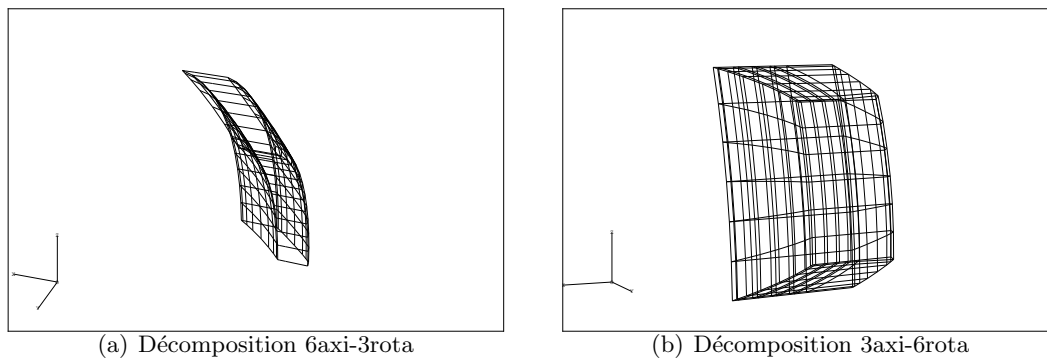
Figure B.7 — Décomposition mixte 3axi-6rota

Les conséquences du choix de la décomposition concernent le type d'hétérogénéité (déjà évoqué, conformité décomposition/interface matérielle), la continuité de la pression, et l'*aspect ratio* des sous-domaines. Les figures B.8 et B.9 donne les allures des sous-domaines pour différentes décompositions. Rappelons qu'un bon *aspect ratio* entraîne *a priori* un meilleur conditionnement mais des calculs locaux plus lourds (sous-structure massive), par ailleurs pour ce problème un bon aspect ratio entraîne nécessairement des interfaces traversant les hétérogénéités ce qui est connu pour conduire à des difficultés de résolution et par ailleurs dans notre cas le choix ayant été fait de ne considérer qu'une inconnue d'interface en déplacement, à des pressions discontinues.





*Figure B.8* — Allure des sous-domaines



*Figure B.9* — Allure des sous-domaines