



Méthodes fortement parallèles pour la simulation numérique en mécanique non linéaire des structures

Camille Negrello

► To cite this version:

Camille Negrello. Méthodes fortement parallèles pour la simulation numérique en mécanique non linéaire des structures. Mécanique des solides [physics.class-ph]. Université Paris Saclay (COmUE), 2017. Français. NNT : 2017SACLN049 . tel-01709834

HAL Id: tel-01709834

<https://theses.hal.science/tel-01709834>

Submitted on 15 Feb 2018

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.

Méthodes fortement parallèles pour la simulation numérique en mécanique non linéaire des structures

Thèse de doctorat de l'Université Paris-Saclay
préparée à l'École Normale Supérieure Paris-Saclay

École doctorale n°579
Sciences mécaniques, énergétiques, matériaux et géosciences - SMEMAG
Spécialité de doctorat : Solides, structures et matériaux

Thèse présentée et soutenue à l'École Normale Supérieure Paris-Saclay,
le 14 Novembre 2017, par

Mlle Camille Negrello

Composition du Jury :

M. David Dureisseix	
Professeur INSA Lyon	(Examineur)
M. Pierre Gosselet	
Chargé de recherche CNRS ENS Paris-Saclay	(Co-encadrant)
M. Frédéric Magoulès	
Professeur École Centrale Supélec	(Président)
M. Jean-Charles Passieux	
Maître de conférences (HDR) INSA Toulouse	(Rapporteur)
M. Christian Rey	
Professeur Safran Tech	(Directeur)
M. Daniel J. Rixen	
Professeur Technische Universität München	(Examineur)
M. François-Xavier Roux	
Professeur UPMC	(Rapporteur)

À l'issue de ces trois années de thèse, et en prologue de ce manuscrit, une page de remerciements était indispensable, pour rappeler leur importance à toutes les personnes sans qui je ne serais pas allée au bout de mes travaux.

Je tiens à remercier tout d'abord mon directeur de thèse, Christian Rey, ainsi que mon co-encadrant, Pierre Gosselet. Merci à eux de m'avoir accompagnée tout au long de ces trois années, de m'avoir guidée dans les passages flous de ma thèse tout en me laissant une autonomie qui m'a permis de creuser à fond chacun des sujets que j'ai abordés. Merci à Christian pour son enthousiasme et son dynamisme, son humour également, qui ont toujours rendu nos réunions agréables. Merci à Pierre pour sa présence et sa disponibilité, son aide toujours accordée avec plaisir, sur un spectre très large de problèmes, des détails techniques aux idées très générales, voire même celui de l'hébergement antillais... Merci à tous les deux.

Merci aux doctorants et post-doctorants du laboratoire, avec qui j'ai passé trois années dans la bonne humeur, et avec qui j'ai pu tisser des amitiés tout au long de ma thèse : Rana, Paul, Thomas, Snachou, Olivia, Werty, Maxime, François, Clémangelain, les deux Marie, Richou, Théau, Silvère, B. Voillot, Python, le Douanier, Charles, le Corse, Maxence, Bhimal... Et tous les autres. Merci entre autres pour votre motivation à organiser des barbecues, réserves, pauses café...

Merci également à tous ceux qui m'ont aidée pour mes présoutenances : Stéphane Roux, Manu, Yoann, Martin, PAG, Paul, Robin, Thomas, Snachou, Clément, Werty.

Merci aux permanents avec qui j'ai passé de bons moments, en particulier PAB, Martin, Manu, Yoann. Merci au directeur du labo, Fred, qui a su gérer l'équilibre entre sérieux et bonne humeur au sein du laboratoire, durant ces trois années notamment.

Merci à Boubou, Rémi, Canis et Frisou pour leur enthousiasme quotidien et leur aide. Merci à Philippe Sanchez et Pierre Lucotte en ce qui concerne les questions informatiques.

Je tiens à remercier Lydia, gestionnaire du secteur Structures, qui malgré sa charge de travail s'est toujours rendue disponible pour la moindre de mes questions, avec le sourire et beaucoup d'efficacité. Je remercie aussi Danielle, gestionnaire du DGM, pour sa bonne humeur et sa gentillesse, et Christine, qui a été là pour gérer les problèmes administratifs avec l'école doctorale.

Je n'oublie pas non plus l'équipe pédagogique du DGC, avec qui j'ai pu travailler pendant mes missions d'enseignement, en particulier Pascale, secrétaire du secteur GC, et Clément et Xavier, qui m'ont aidée notamment pour les derniers cas tests numériques de ma thèse, plutôt dans le domaine de la mécanique des sols.

Je souhaite enfin remercier plus que tout ma famille, qui a toujours été là pour moi, depuis bien avant le début de ma thèse, et jusqu'au pot final. Merci à mes parents qui m'ont soutenue, merci à mes sœurs, qui ont été à l'écoute. Merci aussi à mes grandparents, mes oncles et tantes, et tous ceux que j'oublie...

Table des matières

Introduction	1
1 Etat de l'art	7
1 Méthodes de Décomposition de Domaine, conditions d'interface	7
1.1 Méthodes contractantes : méthodes de Schwarz	9
1.2 Méthodes non contractantes : méthodes du complément de Schur	16
2 Méthodes de résolution des systèmes non linéaires	37
2.1 Méthodes de point fixe	38
2.2 Méthodes de Newton	42
2.3 Méthodes de Newton inexact	45
3 Résolution parallèle des systèmes non linéaires	47
3.1 Méthodes NKS	48
3.2 Méthode ASPIN	48
3.3 Méthode LaTIn	49
3.4 Relocalisation non linéaire	50
2 Sous-structuration et condensation non linéaires	52
1 Problème de référence, notations	52
2 Méthode classique	54
3 Construction d'un problème non-linéaire condensé à l'interface	57
3.1 Trois formulations distinctes	57
3.2 À propos de l'existence des analogues non linéaires aux compléments de Schur	63
4 Stratégie de résolution	64
4.1 Algorithme de Newton	64
4.2 Calcul de l'opérateur tangent	65
4.3 Étapes de résolution	67
4.4 Algorithmes	72
4.5 Analyse de l'erreur	72
5 Résultats	81

5.1	Plasticité parfaite	81
5.2	Écrouissage isotrope	90
5.3	Discussion	95
6	Conclusion	97
3	Conditions de Robin : impédance d'interface	99
1	Un problème bien connu des mécaniciens	100
1.1	Motivation	100
1.2	Une revue non exhaustive des techniques d'évaluation de l'impédance d'interface	103
2	Une nouvelle heuristique pour le calcul de l'impédance d'interface	107
2.1	Un modèle basé sur l'assemblage de ressorts en série	107
2.2	Une interprétation multi-échelle non linéaire	108
2.3	Approximation à deux échelles de la souplesse	111
3	Résultats	115
3.1	Deux cas-tests	115
3.2	Analyse élastique	115
3.3	Analyse plastique	118
3.4	Couplage avec la méthode SRKS	122
3.5	Enrichissement de l'approximation de rang faible	123
4	Conclusion	125
4	Préconditionneur non linéaire	126
1	Introduction	126
2	Une version non linéaire des preconditionneurs FETI/BDD	128
2.1	Cas linéaire : interprétation du système preconditionné FETI (resp. BDD) comme un problème de recherche de point fixe	129
2.2	Préconditionneur non linéaire	133
3	Résultats	140
3.1	Diffusion d'eau dans une colonne de sol	140
3.2	Thermique non linéaire	145
4	Conclusion	149
	Conclusion et perspectives	151
	A	155
	Bibliographie	159

Introduction

L'essor de la théorie des équations différentielles aux dérivées partielles, à partir du XVII^{ème} siècle, a permis le développement de la modélisation et prédiction du comportement des solides : trajectoires, déplacements et déformations sous sollicitations statiques ou dynamiques. Avec l'apparition des premiers ordinateurs et le développement des méthodes numériques, l'intérêt pour les modèles discrets a contribué à la popularisation des formulations faibles ou intégrales, et de leurs solveurs – intérêt également porté par l'impossibilité, dans la majorité des cas, de dégager par le calcul analytique toutes les informations utiles à l'ingénieur.

À la croisée des enjeux de productivité industrielle et d'optimisation des outils algorithmiques de résolution d'équations différentielles, la simulation numérique tend à devenir incontournable pour les concepteurs de structures complexes, par la prédiction peu coûteuse de leur comportement en état de service, et une optimisation fine des géométries, matériaux, rendue possible par la simplicité d'utilisation des logiciels de simulation numérique des structures existant actuellement. Les domaines de validité des solveurs utilisés par ces derniers sont toujours en expansion, de même que leur robustesse ou fiabilité, grâce aux travaux de recherche menés ces dernières décennies, permettant de simuler dans des temps tendant à se raccourcir des structures de taille toujours plus grande et de géométrie plus singulière, ou des phénomènes plus complexes.

Les systèmes discrets résultant de la projection de la formulation intégrale d'une équation différentielle sur une base d'un sous-espace de recherche des solutions de dimension finie peuvent être soit linéaires, soit non linéaires – en lien avec le caractère linéaire ou non linéaire du modèle de comportement matériau sous-jacent, la validité ou non de l'hypothèse des petites perturbations... Si la théorie linéaire peut être considérée comme suffisamment avancée à l'aube du XXI^{ème} siècle pour légitimer l'implémentation de certains solveurs numériques fiables et rapides dans de nombreux logiciels (mécanique des solides, des sols, des fluides...), les difficultés posées encore aujourd'hui par la résolution d'équations non linéaires nécessitent un travail plus approfondi des solveurs dédiés, et tout particulièrement de leur robustesse et performance.

Ce manuscrit sera en partie sous-tendu par cet objectif de robustification des sol-

veurs non linéaires, grâce à une amélioration de leurs performances (étendue du domaine de validité en termes d'intensité de non linéarité, de variété de géométries acceptables, de rapidité et d'assurance de convergence dans des cas complexes).

Si le domaine de validité des solveurs numériques tend à englober de plus en plus de phénomènes non linéaires, s'appuyant sur une expérience riche de nombreuses recherches et résultats concernant les comportements linéaires, l'un des autres défis majeurs de ce début de siècle consiste à repousser les limites de la taille des problèmes résolubles numériquement, enjeu fortement corrélé à l'avènement des super-calculateurs et des architectures parallèles, rendant possible l'augmentation de la précision des calculs sur un domaine donné (raffinement du maillage), ou la taille des structures physiques considérées.

Dans le contexte de la résolution numérique de systèmes de grande taille, l'un des outils en vogue depuis quelques décennies – i.e. le calcul *parallèle* – exploite le caractère distribué de la plupart des architectures numériques actuelles, divisées en de nombreuses unités comprenant différentes échelles (voir la figure 1). De façon générique, ces architectures distribuées, *super-calculateurs* ou *clusters*, se divisent en *nœuds*, entre lesquels des cartes ou un pilote de réseau local assurent la possibilité de transmettre des informations. Les unités composant chacun de ces nœuds sont les *sockets* (ou *processeurs*), un socket pouvant également comporter plusieurs (actuellement 72 pour un Xeon Phi) unités élémentaires : les *cœurs* (le terme « processeur » est aussi utilisé pour désigner un *cœur*, cette indifférentiation provenant des anciennes versions de processeurs, alors composés d'un seul cœur). Les accélérateurs GPU, des unités de calcul comprenant un grand nombre de cœurs ou unités de traitement – initialement dédiés au rendu graphique, – apparaissent également de plus en plus dans la course à la parallélisation.

Chaque cœur d'un socket peut être vu comme une unité de calcul élémentaire, à laquelle des jeux d'instructions, ou tâches, peuvent être envoyés. Ainsi, étant donné le grand nombre de processeurs disponibles dans les super-calculateurs, une manière simple d'augmenter la rapidité de traitement d'une liste d'instructions, si chacune d'entre elles nécessite un nombre d'opérations « raisonnable », est d'en effectuer chaque unité sur un cœur donné, simultanément (i.e. en *parallèle*). Toutefois, ceci n'est possible que si les instructions sont indépendantes, c'est-à-dire qu'aucune d'entre elles ne nécessite en entrée des informations obtenues en sortie d'une autre instruction. Dans le cas contraire, un bon niveau de parallélisme et un gain de temps conséquent peuvent souvent tout de même être obtenus en découpant la liste de tâches en sous-listes distribuées, et en regroupant dans une phase finale les données en sortie de chaque processeur.

Une hiérarchisation de la mémoire permet également d'améliorer les performances des architectures parallèles. Les accès à la mémoire persistante (disque dur, clé USB...) sont à éviter au maximum car transitant par un réseau lent : la mémoire vive (RAM), dont l'accès est facilité par un réseau plus performant, est capable de stocker une quantité limitée d'informations de façon temporaire, afin d'augmenter la rapidité de

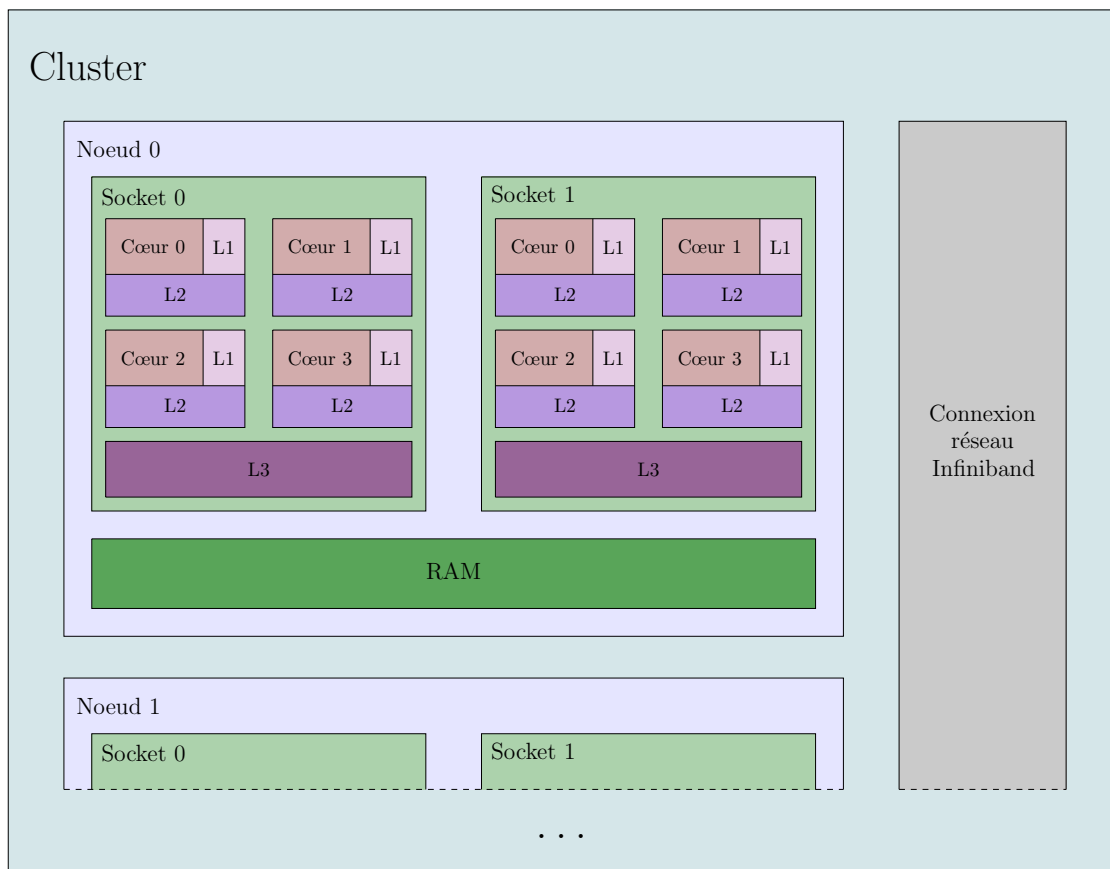


FIGURE 1 – Architecture parallèle

traitement des instructions. Trois niveaux de mémoire supplémentaires, ultra-rapides mais de capacité moindre, sont également disponibles dans les architectures mémoire distribuées actuelles : les caches L1, L2 et L3. Le cache de niveau L3 est associé à un socket, les informations qu'il contient en mémoire sont accessibles par tous les cœurs du socket ; les caches de niveau L1 et L2 sont spécifiques à un cœur donné. Les informations directement nécessaires au traitement des tâches par les cœurs sont stockées dans l'une ou l'autre de ces structures, afin d'optimiser les accès mémoire. La figure 1 présente les emplacements des différents niveaux de mémoire ; la RAM est de manière générale reliée à la mémoire persistante de l'ordinateur ou du cluster, assurant la possibilité d'y copier temporairement les données nécessaires pour lancer un programme.

Deux principaux modèles de programmation permettent actuellement à l'utilisateur de paralléliser les parties indépendantes d'un code. Le premier, dit modèle *multitâche* ou *multithreading model*, dont l'interface la plus courante est OpenMP, définit un niveau de parallélisme à l'échelle du nœud, où grâce à la rapidité du réseau les

structures de stockage de type mémoire vive (à l'échelle des sockets) et L3 (à l'échelle des cœurs) peuvent être considérées comme « partageables ». OpenMP permet d'ouvrir des *régions parallèles* dans un programme, et d'indiquer à l'ordinateur combien de *threads* (entités d'exécution avec une mémoire locale) allouer pour exécuter les différentes instructions de cette région : on parle alors de *multithreading*. Les différentes variables intervenant dans un programme peuvent alors avoir un statut *partagé* (accessible et modifiable par tous les cœurs, ces derniers en possèdent alors tous la même instance) ou *privé* (chaque cœur possède une instance de la variable, les modifications apportées par un cœur n'affectent pas la valeur de l'instance de la variable d'un autre cœur), défini par l'utilisateur, suivant l'usage prévu pour chacune d'entre elles. Ce modèle de programmation parallèle ne requiert toutefois pas de l'utilisateur de déterminer à l'avance quelle instruction sera réalisée sur un cœur donné, ni de contrôler finement les échanges d'informations entre cœurs : il peut aisément être utilisé pour paralléliser des boucles et des opérations de type réduction.

Le second modèle de programmation quant à lui, dont l'interface standard est Message Passing Interface (MPI), définit un niveau de parallélisme à l'échelle du cluster. Chaque cœur y est vu comme une entité de calcul indépendante, et la notion de mémoire *partagée* disparaît : un cœur donné connaît la valeur de ses variables, rendues automatiquement *privées* ; pour avoir accès à la valeur des variables d'autres cœurs, qu'ils soient localisés dans le même nœud (ou socket) ou non, les fonctions de la bibliothèque MPI, permettant de gérer les *communications* entre cœurs, doivent être appelées. Ces dernières peuvent être répertoriées en deux types : les unes concernent les communications entre deux cœurs ciblés (méthodes `Send`, `Recv`, `Sendrecv`), les autres font intervenir un échange d'informations entre tous les cœurs d'un même communicateur ou sous-communicateur (méthodes de type `all-to-all` ou `all-reduce`). Ce manuscrit s'intéressera particulièrement au deuxième modèle de programmation parallèle (MPI), plus adapté aux solveurs dits de *Décomposition de Domaine*, consistant à appliquer le principe de parallélisation d'instructions aux algorithmes de résolution numérique d'équations différentielles – pour lesquels un contrôle fin des échanges d'informations entre cœurs doit être assuré ; la théorie développée au cours de ce document s'appuiera fortement sur ces solveurs, ils seront notamment détaillés dans le premier chapitre – bibliographique.

Une deuxième caractéristique notable différencie le modèle de programmation parallèle MPI d'OpenMP : grâce à la possibilité de communication inter-nœuds, un niveau de parallélisme plus massif peut être créé, avec la gestion de plusieurs centaines ou milliers de cœurs. Bien que les résultats qui seront présentés dans les prochains chapitres ne concernent que des cas-tests académiques de quelques (dizaines de) milliers d'inconnues, l'objectif, à terme, est de pouvoir traiter des problèmes non linéaires à plusieurs millions d'inconnues. Une possibilité est alors d'utiliser le modèle MPI pour effectuer la parallélisation globale de l'algorithme de résolution, et OpenMP pour paralléliser les sous-jeux d'instructions résultants. Ces stratégies de parallélisation des algorithmes de résolution d'équations différentielles discrétisées sont actuellement au cœur des préoccupations des acteurs du *High Performance Computing* (HPC). En

particulier, la vitesse de convergence des solveurs numériques est corrélée à l'optimisation de leur parallélisation : l'un des courants de ces dernières décennies vise la réduction des opérations de communication au sein du processus de résolution, on parle alors de *communication avoiding algorithms*. Ce manuscrit s'inscrit dans la lignée de ce mouvement, basé sur la simple observation que le temps que les cœurs passent à communiquer est du temps « perdu », le cas idéal correspondant à un nombre de tâches égal au nombre de cœurs disponibles, toutes totalement indépendantes : le temps d'exécution est alors égal au temps nécessaire à la réalisation des opérations composant la tâche la plus longue.

Quatre chapitres permettront d'étoffer les deux axes définis ci-dessus :

- la robustesse des solveurs non linéaires
- la réduction des communications au sein du processus de résolution

Ces thématiques sont étroitement liées, l'amélioration des performances consécutive à la réduction des échanges entre cœurs participant de la robustesse du solveur.

Dans un premier temps, un chapitre bibliographique détaillera une partie des fondements théoriques, que l'on réutilisera largement dans les trois autres chapitres, nécessaires à la compréhension du vaste domaine de la résolution parallèle d'équations différentielles non linéaires. Deux thématiques en particulier guideront la structuration de ce chapitre :

- Les méthodes de *Décomposition de Domaine*, ou comment sous-structurer un problème mécanique défini sur un domaine en sous-problèmes indépendants, correspondant physiquement à une décomposition du domaine étudié, puis définir un processus de résolution permettant d'assurer la résolution de l'équation globale en n'effectuant que des opérations définies localement (sur les sous-problèmes) ou aux interfaces de ces sous-problèmes.
- Les algorithmes de résolution des systèmes non linéaires, ou comment définir une suite de problèmes tangents permettant de converger vers la solution non linéaire en utilisant les outils performants des solveurs linéaires.

Un second chapitre posera le cadre théorique de solveurs non linéaires parallèles, dits *méthodes de sous-structuration et condensation non linéaires*, choisis pour leur optimisation « naturelle » des communications au cours de leur déroulement. Si de premiers travaux relativement récents ont permis de poser les bases de définition de ces nouveaux solveurs, ce chapitre permettra de définir plus précisément les objets mis en jeu, le cadre de validité du processus de résolution, ainsi que de peaufiner les différentes approches existantes. En particulier, étant donnée l'imbrication de plusieurs algorithmes « internes » réalisée au sein de ces solveurs globaux, une étude des

critères d'arrêt permettra d'éviter de potentielles sur-résolutions occasionnées par une précision inutilement élevée sur des solutions intermédiaires. De premiers résultats permettront d'étudier les performances des différentes approches présentées, ainsi que de valider leur efficacité sur des cas académiques : les gains de la méthode de sous-structuration et condensation non linéaires, comparée à une méthode NKS plus classique, en termes de nombres d'itérations du solveur tangent de Krylov (représentant la quantité de communications inter-processeurs au cours du calcul) sont de l'ordre de 25 à 30%.

Les deux derniers chapitres seront consacrés à l'amélioration des performances de ces solveurs, autour de deux axes distincts :

- Le premier se place dans le cadre de l'approche mixte des méthodes de sous-structuration et condensation non linéaires, pour laquelle la condensation du problème aux interfaces des sous-domaines passe par la définition d'une inconnue mixte, combinaison linéaire de déplacements et d'efforts (dans le cadre de problèmes mécaniques). L'optimisation du solveur vise alors à optimiser le paramètre intervenant dans cette combinaison linéaire, homogène à une rigidité d'interface, afin de maximiser la vitesse de convergence de l'algorithme : des gains d'environ 20% – en termes de nombres d'itérations du solveur de Krylov – sont atteints.
- Le deuxième axe d'optimisation tend à modifier le processus même de résolution et de parallélisation de l'algorithme afin de maximiser les opérations réalisables localement (sans communication), il utilise une nouvelle version de préconditionneur, que l'on rendra non linéaire. Les gains, en termes de nombres d'itérations du solveur de Krylov, sont de l'ordre de 30 à 50%.

Etat de l'art

Ce premier chapitre, exclusivement bibliographique, fera état des connaissances et avancées actuelles en rapport avec les axes majeurs sur lesquels s'appuie cette thèse. On s'attachera donc à répertorier la diversité des techniques existantes principalement dans deux domaines :

- Les méthodes de Décomposition de Domaine (méthodes DD)
- Les méthodes de résolution des équations non linéaires

Dans un premier temps, une investigation des différents types de méthodes DD existantes sera menée, tant concernant les méthodes de point fixe – qui furent les premières, chronologiquement, à apparaître – que celles produisant des opérateurs non contractants – qui seront utilisées dans les chapitres suivants.

Dans un deuxième temps, les principales méthodes de résolution d'équations non linéaires seront répertoriées, et une étude sommaire de leurs performances, en termes de vitesse de convergence et de coût de calcul, permettra d'évaluer leurs avantages et inconvénients.

Enfin, dans l'optique d'introduire les méthodes qui seront détaillées dans les chapitres suivants, la dernière section de cette étude bibliographique sera consacrée à l'union des deux thématiques précédentes, ou en d'autres termes à la résolution parallèle des systèmes non linéaires.

1 Méthodes de Décomposition de Domaine, conditions d'interface

La résolution numérique d'un système linéaire ne nécessite que très rarement le calcul de l'inverse de son opérateur. Cette opération est en effet classiquement effectuée au moyen d'un solveur, qu'il soit direct (pivot de Gauss, factorisations LU ou de

Cholesky...) ou itératif (Gradient Conjugué et ses variantes...) – tous deux bien moins coûteux qu'une inversion directe [Lascaux and Théodor, 1987]. Depuis une soixantaine d'années, ces deux types de solveurs se disputent la suprématie dans le domaine des problèmes linéaires.

La théorie des méthodes directes est relativement bien achevée [Duff et al., 1986, Davis, 2006], elles présentent l'avantage d'être robustes et d'aboutir en un nombre fini d'opérations, connu a priori. Cependant, la capacité de stockage nécessaire augmente rapidement avec la taille du problème, et ce malgré une algorithmique multi-niveaux permettant une certaine parallélisation de la résolution [Koester et al., 1994]. Les récents travaux visant à développer des solveurs multifrontaux massivement parallèles – MUMPS, PARDISO [Amestoy et al., 1998, Schenk et al., 2001, Paszyński et al., 2010], – en lien avec l'augmentation du nombre de cœurs – et donc de la quantité de mémoire vive disponible – dans les clusters, ont permis de traiter des problèmes de taille de plus en plus importante. Cependant les constructeurs *hardware* se heurtent actuellement à une nouvelle barrière mémoire qui sera difficile à dépasser, et qui limite encore les performances des méthodes directes.

Les méthodes itératives, quant à elles, sont en principe plus extensibles (conservation des performances de l'algorithme pour des problèmes de taille croissante, lorsqu'on augmente le nombre de processeurs). Elles ne nécessitent pas le stockage de la matrice entière : on a juste besoin de connaître son action sur un vecteur, et permettent d'exploiter le caractère creux des matrices issues des formulations éléments finis, ainsi que de contrôler la précision de la solution. Cependant, leur convergence n'est pas assurée en un nombre d'itérations déterminé : celui-ci dépend du préconditionneur employé, du point de départ, de la structure de la matrice, du critère d'arrêt, etc. Bien que de récents travaux aient permis d'améliorer la robustesse de ces solveurs, ils ne concurrencent pas encore les méthodes directes dans les codes de calcul industriels. Toutefois, étant donné les limites actuelles des architectures mémoires, leur implémentation devrait se développer rapidement dans le cadre du traitement de problèmes à grand nombre d'inconnues. Un soin tout particulier doit être apporté à l'étude du conditionnement de l'opérateur du système linéaire à résoudre, qui influe de façon importante sur la rapidité de convergence : une technique de préconditionnement est bien souvent nécessaire, son calcul peut alourdir le processus de résolution et doit être traité de façon judicieuse.

Une troisième classe de solveurs tente de tirer parti des avantages des méthodes directes et itératives : les méthodes de Décomposition de Domaine (méthodes DD). Cette section a pour but de faire un historique et état des méthodes DD existantes.

De façon globale, on peut scinder ces méthodes en deux familles :

- les méthodes contractantes, ou méthodes de Schwarz
- les méthodes non contractantes, ou méthodes du complément de Schur

Chacune de ces familles fera l'objet d'un travail de bibliographie approfondi.

1.1 Méthodes contractantes : méthodes de Schwarz

Historiquement, ces méthodes furent les premières à être proposées, dans une optique de démonstration mathématique. En effet, H.A. Schwarz prouva dans les années 1870 l'existence de solutions au problème de Poisson sur certains domaines de forme complexe, où l'on ne pouvait calculer de solution analytique [Schwarz, 1869] : seuls des domaines de formes simples (disque, rectangle...) permettaient le calcul des solutions de cette EDP. La démonstration fut permise grâce au découpage du domaine en deux sous-domaines de formes simples sur lesquels on pouvait calculer une solution explicite : voir la figure 1.1 pour un exemple. En introduisant une suite de solutions de problèmes de Dirichlet résolus successivement dans chaque sous-domaine, et à l'aide du principe de maximum pour les fonctions harmoniques, Schwarz montra la convergence du processus vers la solution du problème sur le domaine global.

À chaque itération de l'algorithme, on résout donc alternativement un problème avec conditions de Dirichlet sur chaque sous-domaine. Le couplage entre les deux solutions est assuré par une région commune, dite de recouvrement : à chaque résolution, on utilise la solution précédente pour fixer les conditions d'interface.

La méthode est de type point fixe, sa convergence est assurée grâce à la contractance de l'opérateur du système à résoudre.

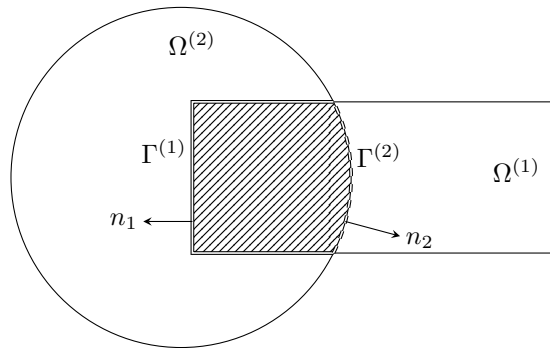


FIGURE 1.1 – Exemple de découpage en deux sous-domaines avec recouvrement

1.1.1 Algorithmes continus

Algorithme alterné La première version apparue de l'algorithme est dite *alternée* : en effet, les résolutions sur chaque sous-domaine sont successives, chaque sous-équilibre ayant besoin de la solution du système précédent pour poser ses conditions aux limites. L'algorithme 1 présente le processus de résolution pour l'équation de Poisson avec des conditions de Dirichlet homogènes – problème (1.1), et le découpage de la

figure 1.1 :

$$\begin{aligned}
 &\text{Soit } f \in \mathcal{L}^2(\Omega) \\
 &\text{Trouver } u \in \mathcal{H}^1(\Omega) \text{ tel que :} \\
 &\Delta u + f = 0 \quad \text{dans } \Omega \\
 &u = 0 \quad \text{sur } \partial\Omega
 \end{aligned} \tag{1.1}$$

Algorithme parallèle Dans l'optique d'améliorer les performances de l'algorithme, environ cent ans plus tard, P.L. Lions propose une extension très importante des travaux de Schwarz : il démontre sa convergence pour un nombre quelconque de sous-domaines avec recouvrement, ainsi qu'en donne une interprétation par formulation variationnelle, ce qui permet de démontrer la convergence pour tout type de problème elliptique.

Un certain niveau du parallélisme peut être atteint avec la version alternée de l'algorithme de Schwarz : en effet, les résolutions locales concernant des sous-domaines n'ayant aucune région de recouvrement en commun peuvent être effectuées en même temps sur des processeurs indépendants : c'est la technique du *coloriage* [Kahou et al., 2007]. Cependant, le nombre d'opérations pouvant être réalisé indépendamment reste limité, et le parallélisme n'est pas exploité de façon optimisée pour les architectures actuelles des clusters.

P.L. Lions introduit alors une variante plus parallèle où toutes les résolutions locales sont effectuées simultanément, donc plus adaptée au calcul sur ordinateur – bien que toujours au niveau continu [Lions, 1988, Lions and Lions, 1989], voir l'algorithme 2.

Algorithm 1 Alternated Schwarz

Require: Subdomains $\Omega^{(1)}, \Omega^{(2)}$, boundaries $\Gamma^{(1)}, \Gamma^{(2)}$

Initialization : u_0 on $\Gamma^{(1)}$, $n = 0$, status = 'NotConverged', threshold $\varepsilon > 0$

while status == 'NotConverged' **do**

1. solve : $\Delta u_{n+\frac{1}{2}} + f = 0$ in $\Omega^{(1)}$

$$u_{n+\frac{1}{2}} = 0 \text{ on } \partial\Omega^{(1)} \setminus \Gamma^{(1)}$$

$$u_{n+\frac{1}{2}} = u_n \text{ on } \Gamma^{(1)}$$

2. solve $\Delta u_{n+1} + f = 0$ in $\Omega^{(2)}$

$$u_{n+1} = 0 \text{ on } \partial\Omega^{(2)} \setminus \Gamma^{(2)}$$

$$u_{n+1} = u_{n+\frac{1}{2}} \text{ on } \Gamma^{(2)}$$

if $\|u_{n+1}|_{\Gamma^{(1)}} - u_n|_{\Gamma^{(1)}}\| \leq \varepsilon$ **and** $\|u_{n+1}|_{\Gamma^{(2)}} - u_n|_{\Gamma^{(2)}}\| \leq \varepsilon$ **then**

status = 'Converged'

Algorithm 2 Parallel Schwarz**Require:** Subdomains $\Omega^{(1)}, \Omega^{(2)}$, boundaries $\Gamma^{(1)}, \Gamma^{(2)}$ **initialization** : $u_0^{(1)}$ on $\Gamma^{(1)}$, $u_0^{(2)}$ on $\Gamma^{(2)}$, $n = 0$, status = 'NotConverged', threshold $\varepsilon > 0$ **while** status == 'NotConverged' **do** solve : $\Delta u_{n+1}^{(1)} + f = 0$ in $\Omega^{(1)}$ $u_{n+1}^{(1)} = 0$ on $\partial\Omega^{(1)} \setminus \Gamma^{(1)}$ $u_{n+1}^{(1)} = u_n^{(2)}$ on $\Gamma^{(1)}$ and : $\Delta u_{n+1}^{(2)} + f = 0$ in $\Omega^{(2)}$ $u_{n+1}^{(2)} = 0$ on $\partial\Omega^{(2)} \setminus \Gamma^{(2)}$ $u_{n+1}^{(2)} = u_n^{(1)}$ on $\Gamma^{(2)}$ **if** $\|u_{n+1}^{(1)}|_{\Gamma^{(1)}} - u_n^{(1)}|_{\Gamma^{(1)}}\| \leq \varepsilon$ **and** $\|u_{n+1}^{(2)}|_{\Gamma^{(2)}} - u_n^{(2)}|_{\Gamma^{(2)}}\| \leq \varepsilon$ **then**

status = 'Converged'

1.1.2 Algorithmes discrets

Enfin, les méthodes de Schwarz sont reformulées au niveau discret : on distingue alors les méthodes dites *multiplicative* [Chan and Mathew, 1994, Smith et al., 2004] et *additive* [Dryja and Widlund, 1990].

L'opérateur discrétisé de l'équation différentielle à résoudre est noté A . On introduit également des opérateurs rectangulaires de restriction R_k , permettant d'obtenir à partir d'une quantité définie sur les nœuds de la structure globale Ω une quantité définie uniquement sur les nœuds d'un sous-domaine $\Omega^{(k)}$. La transposée de cet opérateur, notée R_k^T , prolonge les quantités définies sur les nœuds de $\Omega^{(k)}$ dans le domaine entier Ω par des zéros. Pour la restriction d'un opérateur matriciel A à un sous-domaine $\Omega^{(k)}$ on a : $A_k = R_k A R_k^T$.

Méthode multiplicative Lorsqu'on travaille avec un découpage en deux sous-domaines, le principe de la méthode *multiplicative* est de créer les suites $(u_{n+\frac{1}{2}})_n$ et $(u_{n+1})_n$ définies par :

$$u_{n+\frac{1}{2}} = u_n + R_1^T A_1^{-1} R_1 (f - A u_n)$$

$$u_{n+1} = u_{n+\frac{1}{2}} + R_2^T A_2^{-1} R_2 (f - A u_{n+\frac{1}{2}})$$

Le calcul d'un élément $u_{n+\frac{1}{2}}$ consiste ainsi à effectuer une résolution locale avec l'opérateur A_1 , soit une résolution sur le sous-domaine $\Omega^{(1)}$, tandis que le calcul d'un élément u_{n+1} revient à effectuer une résolution sur le sous-domaine $\Omega^{(2)}$. Pour un nombre

quelconque de sous-domaines K , on déterminera à chaque itération n les éléments des suites :

$$u_{n+\frac{k}{K}} = u_{n+\frac{k-1}{K}} + R_k^T A_k^{-1} R_k \left(f - A u_{n+\frac{k-1}{K}} \right), \quad \forall k \in \{1, \dots, K\}$$

On peut alors déterminer l'itéré global u_{n+1} en fonction de l'itéré global n :

$$u_{n+1} = \prod_{k=1}^K (I - R_k^T A_k^{-1} R_k A) u_n + C f \quad (1.2)$$

où $C = \sum_{k=1}^K (-1)^k \sigma_k(\tilde{Y}_1, \dots, \tilde{Y}_K)$, avec $\tilde{Y}_k = R_k^T A_k^{-1} R_k$ et σ_k le k -ième polynôme symétrique élémentaire des K variables $(\tilde{Y}_k)_{1 \leq k \leq K}$ – i.e. la somme de tous les produits de k d'entre ces variables, ou encore, en notant $\mathcal{P}_k(\{1, \dots, K\})$ l'ensemble des parties à k éléments de l'ensemble $\{1, \dots, K\}$:

$$\sigma_k(\tilde{Y}_1, \dots, \tilde{Y}_K) = \sum_{I \in \mathcal{P}_k(\{1, \dots, K\})} \prod_{i \in I} \tilde{Y}_i$$

La méthode de Schwarz multiplicative est une méthode par nature séquentielle, de même que la méthode de Schwarz alternée.

Remarque 1 : Lorsqu'on utilise un recouvrement nul, la méthode de Schwarz multiplicative est équivalente à une méthode de Gauss Seidel par blocs, ou encore à la discrétisation de la méthode de Schwarz alternée (voir algorithme 1) avec recouvrement minimal – on rappelle que les méthodes classiques continues de Schwarz ne convergent pas lorsque le recouvrement est nul.

Méthode additive De même qu'au niveau continu, une version parallèle de la méthode de Schwarz multiplicative existe et est appelée méthode *additive* ; elle consiste à créer la suite d'itérés :

$$u_{n+1} = \left(I - \sum_{k=1}^K R_k^T A_k^{-1} R_k A \right) u_n + \sum_{k=1}^K R_k^T A_k^{-1} R_k f \quad (1.3)$$

Les résolutions locales par sous-domaine peuvent alors être réalisées simultanément, puisqu'elles sont rendues indépendantes de la solution des autres résolutions locales.

Lorsqu'on considère un recouvrement nul entre les sous-domaines, la méthode de Schwarz additive est équivalente à la discrétisation de la méthode parallèle présentée plus haut (voir algorithme 2) pour un recouvrement de taille minimale, ou encore à une méthode Jacobi par blocs.

Cependant, l'existence de recouvrements non nuls entre les sous-domaines implique la non-convergence de l'algorithme additif [Gander et al., 2008] : cette méthode n'est en général pas utilisée telle quelle mais en tant que préconditionneur d'un solveur de Krylov.

En effet, on reconnaît dans l'expression de la suite (1.3) une formulation de type :

$$u_{n+1} = g(u_n)$$

et on cherche un point fixe de $g : g(u_n) = u_n$.

$$g(u_n) - u_n = 0 \quad \Leftrightarrow \quad \left(\sum_{k=1}^K R_k^T A_k^{-1} R_k \right) A u_n = \left(\sum_{k=1}^K R_k^T A_k^{-1} R_k \right) f$$

On définit alors le préconditionneur de Schwarz additif (AS) :

$$M_{AS} = \sum_{k=1}^K R_k^T A_k^{-1} R_k$$

1.1.3 Restricted Additive Schwarz

Récemment, une variante plus efficace de la méthode de Schwarz additif a été conçue – de façon accidentelle, grâce à une erreur de programmation [Cai and Sarkis, 1999]. La modification apportée à l'algorithme initial consiste en une suppression d'une partie des communications entre processeurs, ce qui permet d'augmenter la rapidité de calcul dans le cas d'une implémentation parallèle. L'algorithme résultant est appelé Restrictive Additive Schwarz (RAS).

Afin d'explicitier le principe de RAS, on va définir deux types de décomposition de domaine (voir la figure 1.2) : la première consiste en une décomposition du domaine Ω en sous-domaines $\Omega^{(k)0}$ non recouvrants, et tels que $\Omega = \cup \Omega^{(k)0}$. À partir de cette décomposition, on crée une partition $\{\Omega^{(k)}\}$ avec recouvrements de taille $\delta = 1$, i.e. en incluant dans chaque sous-domaine $\Omega^{(k)0}$ tous les nœuds au voisinage immédiat de son bord. Par récurrence, on crée une partition avec recouvrements de taille δ , dont on peut se servir pour appliquer l'algorithme classique de Schwarz additif.

Deux types d'opérateurs sont associés à ces décompositions, dits de *restriction*, permettant d'obtenir des quantités locales à partir de quantités globales. Les premiers seront notés R_k^δ et correspondent à une restriction du domaine Ω au sous-domaine $\Omega^{(k)}$ – ce sont les opérateurs R_k de la section 1.1.2 (chap. 1) dans le cas d'un recouvrement de taille δ . Le second type correspond à une restriction du domaine Ω à un sous-domaine $\Omega^{(k)0}$ – sans recouvrement ; il sera noté R_k^0 , et on notera R_k^{0T} l'opérateur de prolongement du sous-domaine $\Omega^{(k)0}$ dans le domaine Ω . L'opérateur discrétisé de l'équation différentielle à résoudre est noté A , sa restriction à un sous-domaine $\Omega^{(k)}$ est notée $A_k = R_k A R_k^T$.

On peut écrire le préconditionneur de Schwarz additif grâce aux opérateurs définis ci-dessus :

$$M_{AS}^{-1} = \sum_{k=1}^K R_k^{\delta T} A_k^{-1} R_k^\delta$$

Le préconditionneur RAS s'écrit, quant à lui :

$$M_{RAS}^{-1} = \sum_{k=1}^K R_k^{0T} A_k^{-1} R_k^\delta \quad (1.4)$$

L'expression (1.4) permet de diminuer les communications entre processeurs, car le calcul de $\sum_{k=1}^K R_k^{0T} X_k$ pour une famille de vecteurs locaux X_k donnée, i.e. la somme des prolongements canoniques des X_k dans Ω , ne nécessite que la communication des données d'interface entre les processeurs voisins, et non des données portant sur les recouvrements. De même que la méthode de Schwarz additive, cette formulation est utilisée pour préconditionner des solveurs basés sur la théorie des espaces de Krylov, elle est aujourd'hui implémentée par défaut dans différents codes de calcul parallèles, comme PETSc.

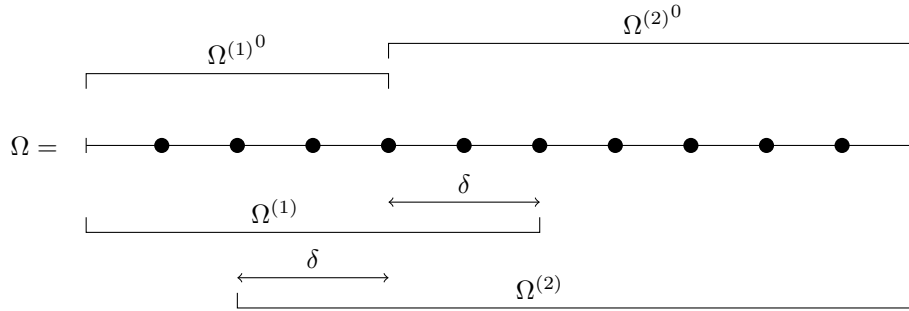


FIGURE 1.2 – Exemple de partition avec recouvrement de taille $\delta = 2$, pour deux sous-domaines en 1D

1.1.4 Conditions de Robin, ou méthode de Schwarz optimisée

P.L. Lions démontre en 1988 que la convergence de la méthode de Schwarz dépend de la taille du recouvrement δ : plus le recouvrement est important, meilleure est la convergence théorique de l'algorithme. Cependant, dans le cadre des méthodes parallèles (méthode additive notamment), plus le recouvrement est important, plus les communications entre processeurs augmentent, ce qui alourdit et ralentit le processus de résolution.

Une alternative est alors proposée dans les années 1990 (toujours par P.L. Lions), plus sophistiquée, et connue sous le nom de méthode de Schwarz optimisée [Lions, 1990]. Le principe de son efficacité réside dans l'utilisation de conditions d'interface plus complètes que celles employées jusqu'alors : à la place de conditions de Dirichlet – en déplacement – ou de Neumann – en effort, on va imposer une combinaison linéaire des deux : une optimisation de ces conditions d'interface est alors possible, permettant d'accélérer la rapidité de convergence de la méthode [Gander, 2006, Nataf, 2007, Gander and Dubois, 2015].

Étant donnée une constante positive $\eta > 0$, on cherche à résoudre le problème modèle suivant (équation de Helmholtz) :

$$(\eta - \Delta) u = f \text{ dans } \Omega = \mathbb{R}^2$$

On définit une taille de recouvrement $\delta > 0$, $\Omega^{(1)} =]-\infty, \delta]$ et $\Omega^{(2)} = [0, +\infty[$. Soit une constante $\alpha > 0$, l'algorithme de Schwarz optimisé consiste à résoudre pour chaque sous-structure $\Omega^{(k)}$, $k \in \{1, 2\}$:

$$\begin{aligned} (\eta - \Delta) u_{n+1}^{(k)} &= f(x, y) \text{ dans } \Omega^{(k)} \\ u_{n+1}^{(k)} &\text{ est bornée à l'infini} \\ \left(\frac{\partial}{\partial n_k} + \alpha \right) u_{n+1}^{(k)} &= \left(\frac{\partial}{\partial n_k} + \alpha \right) u_n^{(\bar{k})} \text{ sur } \Gamma^{(\bar{k})} \end{aligned} \quad (1.5)$$

L'exposant $(\bar{k})$ définit les grandeurs associées au complémentaire de k dans $\{1, 2\}$. Les vecteurs unitaires n_k sont les normales sortantes aux interfaces des sous-domaines (voir la figure 1.1 pour un exemple).

Grâce à une transformation de Fourier partielle dans la direction y , une analyse de la convergence permet d'évaluer l'efficacité de l'algorithme. On notera p la variable de Fourier, et on introduit la fonction :

$$\lambda(p) = \sqrt{\eta + p^2}$$

Le facteur de convergence s'écrit alors :

$$\rho(p, \delta, \alpha) = \left| \frac{\lambda(p) - \alpha}{\lambda(p) + \alpha} \right| \exp(-\lambda(p)\delta)$$

On peut faire les remarques suivantes quant à l'expression de ρ :

- On a $\rho(p, \delta, \alpha) < 1$, donc la méthode est convergente
- On peut prendre $\delta = 0$: $\rho(p, 0, \alpha) < 1$, ce qui signifie que la méthode est également convergente pour un découpage sans recouvrement – la démonstration en fut donnée par P.L. Lions dans un cadre plus général que la résolution du problème 1.5 [Lions and Lions, 1989].
- Quand il y a recouvrement $\delta > 0$, le facteur de convergence est uniformément borné pour $p \in \mathbb{R}$: $\rho(p, \delta > 0, \alpha) < \exp(-\eta\delta) < 1$. De plus, $\rho \rightarrow 0$ quand $p \rightarrow \infty$, la convergence de l'algorithme de Schwarz est donc géométrique (ou q-linéaire).
- La propriété précédente n'est plus vraie pour un recouvrement nul $\delta = 0$: $\rho(p, 0, \alpha) \rightarrow 1$ quand $p \rightarrow \infty$. Cependant, la diminution des communications apportée par l'absence de recouvrement peut largement compenser le taux de convergence moindre.

- Il existe un paramètre α optimal : soit $p_0 \in \mathbb{R}$, en prenant $\alpha_0 = \lambda(p_0)$ on a $\rho(p_0, \delta, \alpha_0) = 0$, quel que soit le recouvrement δ . La convergence est alors établie en deux itérations au plus (pour deux sous-domaines).
- Le facteur de convergence de l'algorithme classique de Schwarz additif est $\rho = \exp(-\lambda(p)\delta)$: on peut vérifier qu'un recouvrement nul $\delta = 0$ conduit à la non convergence de l'algorithme : $\rho(p, \delta = 0) = 1$. La version modifiée apporte une amélioration de la convergence quantifiée par le facteur $|\frac{\lambda(p) - \alpha}{\lambda(p) + \alpha}| < 1$.

1.2 Méthodes non contractantes : méthodes du complément de Schur

On a vu dans la section précédente que les premières Méthodes de Décomposition de Domaine, ou méthodes de Schwarz, produisaient des systèmes linéaires avec opérateurs contractants, et pouvaient être vues comme des techniques de point fixe. De plus, la plupart de ces méthodes impliquaient la présence d'un recouvrement entre les différents sous-domaines. Augmenter la taille du recouvrement permettait alors d'augmenter la vitesse de résolution.

Toutefois, un large recouvrement implique également une grande quantité d'information à échanger entre les sous-domaines, et donc de communications entre les processeurs dans le cas d'une programmation parallèle. Ainsi, le gain de temps CPU n'est-il pas uniquement dépendant du taux de convergence de l'algorithme, mais également de la diminution des communications au cours de la résolution, et la découverte ainsi que le développement de techniques de méthodes DD sans recouvrement fut-il naturel.

Durant la deuxième moitié du 20^è siècle, la découverte et le développement des solveurs de Krylov a révolutionné la résolution des problèmes linéaires en parallèle. Ils permirent, d'une part, d'accélérer les méthodes de Schwarz, mais surtout, participèrent de l'essor d'un nouveau type de méthodes DD, ne possédant pas la propriété de contractance et n'étant donc pas résolus par une méthode de point fixe. La grande fiabilité et extensibilité acquises au fur et à mesure de leur optimisation devrait encourager les développeurs de codes de calculs industriels à implémenter ces nouveaux solveurs dans un futur proche [Farhat, 1994, Le Tallec, 1994, Farhat et al., 2001, Gosselet and Rey, 2006].

On s'intéressera uniquement, dans cette section, à une formulation discrétisée et générale de tout type de problème de mécanique linéaire :

$$Ku = f$$

où K est un opérateur linéaire, u un vecteur d'inconnues nodales et f un second membre discrétisé. Les méthodes de Schur consistent à sous-structurer ce problème global, afin de définir deux nouveaux types de problèmes :

- un problème d'interface de taille réduite, résultant de la condensation du problème global
- des sous-problèmes locaux, de la taille des sous-domaines

L'idée de base se résume à résoudre le problème d'interface, comportant peu d'inconnues, en s'appuyant sur les résolutions locales, que l'on peut paralléliser.

1.2.1 Condensation de Schur

La condensation de Schur permet, comme mentionné au paragraphe précédent, de ramener un problème linéaire de taille n , défini pour un ensemble d'inconnues – par exemple, les degrés de liberté d'un domaine discrétisé – à un problème de taille $n_2 < n$, défini pour un sous-ensemble d'inconnues de taille n_2 . Elle est basée sur une décomposition par blocs de l'opérateur linéaire, par exemple pour une matrice K :

$$K = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix}$$

Les opérateurs K_{11} et K_{22} sont de taille $n_1 \times n_1$ et $n_2 \times n_2$ respectivement.

Si le bloc K_{11} est inversible, on peut réécrire tout problème de type $Ku = f$ sous la forme :

$$\begin{aligned} Ku = f &\Leftrightarrow \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \\ &\Leftrightarrow \begin{cases} u_1 = K_{11}^{-1} (f_1 - K_{12}u_2) \\ (K_{22} - K_{21}K_{11}^{-1}K_{12})u_2 = f_2 - K_{21}K_{11}^{-1}f_1 \end{cases} \end{aligned}$$

On note $K_{/K_{11}}$ et on appelle complément de Schur de K par rapport à K_{11} l'opérateur condensé :

$$K_{/K_{11}} = K_{22} - K_{21}K_{11}^{-1}K_{12}$$

On introduit également le second membre condensé :

$$f_{/K_{11}} = f_2 - K_{21}K_{11}^{-1}f_1 \quad (1.6)$$

On peut ainsi ramener le problème :

$$Ku = f : \text{de taille } n \quad (1.7)$$

au problème :

$$K_{/K_{11}}u_2 = f_{/K_{11}} : \text{de taille } n_2 \quad (1.8)$$

On constate que dans le cas où $n_2 \ll n$, la résolution du système (1.8) pourra être bien plus rapide que la résolution du système initial (1.7). Le problème réduit (1.8) est appelé problème condensé.

1.2.2 Lien avec la mécanique linéaire

Dans le cadre de la mécanique linéaire, les problèmes de type $Ku = f$ correspondent à la discrétisation d'équations aux dérivées partielles définies dans un domaine Ω .

Afin de posséder une solution unique, le problème de mécanique doit être muni de conditions aux limites de type Dirichlet sur une partie non vide $\partial_u \Omega$ du bord du domaine, ainsi que de conditions aux limites de type Neumann sur le complémentaire $\partial_F \Omega$ de cette partie [Hadamard, 1902].

Il est alors aisé de trouver une décomposition par blocs de l'opérateur K qui contienne un bloc diagonal toujours inversible : en effet, si on décompose les degrés de liberté du problème global en deux types :

- degrés de liberté u_i correspondant aux nœuds internes du maillage (définis dans $\Omega \setminus \partial\Omega$)
- degrés de liberté u_b correspondant aux nœuds de bord du maillage (définis sur $\partial\Omega$)

on constate que la restriction de l'opérateur K aux degrés de liberté internes du maillage – notée ici K_{ii} – correspond au remplacement des conditions aux limites par des conditions de Dirichlet sur tout le bord du domaine. Cet opérateur est donc inversible (existence et unicité d'une solution au problème associé).

Ainsi, le problème mécanique défini sur un domaine Ω :

$$Ku = b : \text{de taille } n$$

peut-il toujours se ramener au problème réduit :

$$K_{/K_{ii}} u_b = f_{/K_{ii}} : \text{de taille } n_b$$

où :

- n_b est le nombre de degrés de liberté appartenant au bord du domaine
- $K_{/K_{ii}}$ est le complément de Schur de l'opérateur K par rapport au bloc K_{ii} :

$$K_{/K_{ii}} = K_{bb} - K_{bi} K_{ii}^{-1} K_{ib}.$$
 On parle aussi d'opérateur condensé sur les degrés de liberté de bord
- $f_{/K_{ii}}$ est le second membre condensé introduit en (1.6) : $f_{/K_{ii}} = f_b - K_{bi} K_{ii}^{-1} f_i$

1.2.3 Lien avec les méthodes DD du complément de Schur

Le principe des algorithmes de Décomposition de Domaine de Schur revient à découper un problème aux dérivées partielles défini sur un maillage global en sous-problèmes, tels que les sous-domaines associés soient non recouvrants – voir la figure 1.3 pour un exemple. On s'affranchit alors du surplus de communications engendrées par le recouvrement existant pour la plupart des méthodes présentées à la section 1.1 (chap. 1). Le découpage du domaine en N_s sous-domaines non recouvrants définit deux objets :

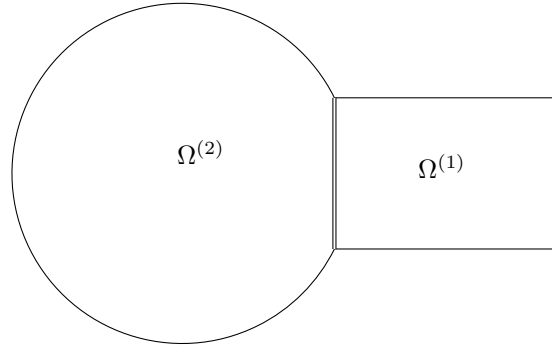


FIGURE 1.3 – Exemple : découpage en deux sous-domaines sans recouvrement

- les sous-structures $\Omega^{(s)}$, $\forall s \in \{1, \dots, N_s\}$
- les interfaces $\partial\Omega^{(s)} \setminus \partial\Omega$, sur lesquelles sont définis les nœuds de bord

Le problème global restreint à une sous-structure est appelé sous-problème, ou équilibre local. Les équilibres locaux sont rendus indépendants les uns des autres grâce à l'introduction d'une condition en efforts $\lambda_b^{(s)}$ – inconnue a priori – aux nœuds de bord du domaine, représentant les interactions avec les sous-domaines voisins de la sous-structure considérée. On notera dans la suite $\lambda^{(s)}$ son prolongement par des zéros dans $\Omega^{(s)}$.

La vérification des équations au niveau global – sur le domaine entier Ω – est alors assurée par l'introduction de conditions sur les grandeurs d'interface entre les sous-domaines : on veut avoir égalité – continuité – des déplacements d'interface pour des sous-domaines adjacents, et équilibre des réactions nodales de bord. On construit ainsi un *problème d'interface* portant sur les inconnues de bord de chaque sous-domaine. Par exemple, pour un découpage en deux sous-domaines $\Omega^{(1)}$ et $\Omega^{(2)}$, on aura :

Équilibres locaux :

$$\begin{aligned} K^{(1)}u^{(1)} &= \lambda^{(1)} + f^{(1)} & K^{(2)}u^{(2)} &= \lambda^{(2)} + f^{(2)} \\ &+ \text{C.L. sur } \partial\Omega^{(1)} & &+ \text{C.L. sur } \partial\Omega^{(2)} \end{aligned} \quad (1.9)$$

$$\text{avec } \lambda^{(s)} \text{ tel que } \lambda_i^{(s)} = 0_i, \forall s \in \{1, 2\} \quad (1.10)$$

Conditions d'interface :

$$\begin{aligned} u_b^{(1)} - u_b^{(2)} &= 0 \\ \lambda_b^{(1)} + \lambda_b^{(2)} &= 0 \end{aligned} \quad (1.11)$$

Remarque 2 : Les conditions aux limites imposées au niveau local sur $\partial\Omega^{(s)}$ sont composées :

- des conditions portant sur les inconnues d'interface $u_b^{(s)}$ ou $\lambda_b^{(s)}$: elles sont définies sur $\partial\Omega^{(s)} \setminus \partial\Omega$
- de la restriction à $\partial\Omega^{(s)}$ des conditions aux limites imposées sur le domaine global (conditions de Dirichlet sur $\partial_u\Omega \cap \partial\Omega^{(s)}$ et de Neumann sur $\partial_F\Omega \cap \partial\Omega^{(s)}$)

En définissant au niveau local les compléments de Schur des opérateurs $K^{(s)}$, les équilibres locaux permettent de relier les grandeurs de bord $u_b^{(s)}$ et $\lambda_b^{(s)}$:

$$K_{/K_{ii}}^{(s)} u_b^{(s)} = \lambda_b^{(s)} + f_{/K_{ii}}^{(s)} \quad \forall s \in \{1, 2\} \quad (1.12)$$

Les conditions d'interface (1.11) peuvent alors être réécrites en fonction d'un seul type d'inconnue. Par exemple, si on choisit de travailler sur l'inconnue d'interface u_b , le système (1.11) peut être reformulé en :

$$\begin{cases} u_b^{(1)} - u_b^{(2)} = 0 \\ K_{/K_{ii}}^{(1)} u_b^{(1)} + K_{/K_{ii}}^{(2)} u_b^{(2)} = f_{/K_{ii}}^{(1)} + f_{/K_{ii}}^{(2)} \end{cases} \Rightarrow \begin{cases} u_b \equiv u_b^{(1)} = u_b^{(2)} \\ \left(K_{/K_{ii}}^{(1)} + K_{/K_{ii}}^{(2)} \right) u_b = f_{/K_{ii}}^{(1)} + f_{/K_{ii}}^{(2)} \end{cases} \quad (1.13)$$

On introduit ainsi une unique inconnue d'interface u_b , et on se ramène à un problème de taille $n_b \equiv n_b^{(1)} = n_b^{(2)}$. Cette formulation est appelée formulation *primale*, le problème (1.13) *problème d'interface* ou *problème condensé* (à l'interface). Il est résolu par des solveurs itératifs de type Krylov.

Remarque 3 : L'opérateur du problème d'interface (1.13) est non contractant : on ne peut pas lui appliquer la théorie du point fixe – cf. théorie de Schwarz.

Pour travailler sur l'inconnue d'interface $\lambda_b^{(s)}$, on remanie les équilibres locaux (1.9) :

$$u^{(s)} = K^{(s)\dagger} (\lambda^{(s)} + f^{(s)}) + R^{(s)} \alpha^{(s)} \quad (1.14)$$

$$\Rightarrow u_b^{(s)} = (K^{(s)})_{bi}^{\dagger} f_i^{(s)} + (K^{(s)})_{bb}^{\dagger} (\lambda_b^{(s)} + f_b^{(s)}) + R_b^{(s)} \alpha^{(s)} \quad (1.15)$$

L'utilisation des pseudo-inverses des matrices de rigidité locales $K^{(s)\dagger}$ est obligatoire puisqu'on n'a pas de certitude quand à leur inversibilité. La contribution de la solution dans l'espace engendré par le noyau de l'opérateur $\text{Ker}(K^{(s)}) = \text{span}(R^{(s)})$ – ou encore

espace engendré par les modes rigides $R^{(s)}$ de la sous-structure – est introduite par l'ajout du terme :

$$R^{(s)}\alpha^{(s)}$$

L'étude de la prise en compte des modes rigides sera abordée en détail à la section 1.2.7 (chap. 1). Dans un premier temps, par souci de clarté, on fera l'hypothèse que les sous-structures n'ont aucun mode rigide :

$$\Rightarrow u_b^{(s)} = \left(K^{(s)\dagger} \right)_{bi} f_i^{(s)} + \left(K^{(s)\dagger} \right)_{bb} \left(\lambda_b^{(s)} + f_b^{(s)} \right)$$

La propriété suivante est valable pour toute matrice décomposable en blocs vérifiant les propriétés adéquates d'inversibilité :

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}^{-1} = \begin{pmatrix} (A - BD^{-1}C)^{-1} & -A^{-1}B(D - CA^{-1}B)^{-1} \\ -(D - CA^{-1}B)^{-1}CA^{-1} & (D - CA^{-1}B)^{-1} \end{pmatrix}$$

On a donc les propriétés suivantes du complément de Schur :

$$\begin{aligned} \left(K^{(s)\dagger} \right)_{bb} &= K_{/K_{ii}}^{(s)\dagger} \\ \left(K^{(s)\dagger} \right)_{bi} &= -K_{/K_{ii}}^{(s)\dagger} K_{bi} K_{ii}^{-1} \end{aligned}$$

Et les équilibres locaux peuvent être reformulés :

$$u_b^{(s)} = K_{/K_{ii}}^{(s)\dagger} \left(\lambda_b^{(s)} + f_{/K_{ii}}^{(s)} \right)$$

Le système d'équations d'admissibilité aux interfaces (1.11) devient :

$$\begin{cases} K_{/K_{ii}}^{(1)\dagger} \lambda_b^{(1)} - K_{/K_{ii}}^{(2)\dagger} \lambda_b^{(2)} = - \left(K_{/K_{ii}}^{(1)\dagger} f_{/K_{ii}}^{(1)} - K_{/K_{ii}}^{(2)\dagger} K_{/K_{ii}}^{(2)} \right) \\ \lambda_b^{(1)} + \lambda_b^{(2)} = 0 \end{cases} \quad (1.16)$$

ou encore :

$$\Rightarrow \begin{cases} \lambda_b \equiv \lambda_b^{(1)} = -\lambda_b^{(2)} \\ \left(K_{/K_{ii}}^{(1)\dagger} + K_{/K_{ii}}^{(2)\dagger} \right) \lambda_b = - \left(K_{/K_{ii}}^{(1)\dagger} f_{/K_{ii}}^{(1)} - K_{/K_{ii}}^{(2)\dagger} K_{/K_{ii}}^{(2)} \right) \end{cases}$$

Cette formulation est, quant à elle, appelée formulation *duale*.

Remarque 4 : *En mécanique des structures, le complément de Schur par rapport aux nœuds internes, aussi appelé complément de Schur primal, ainsi que la condensation des efforts extérieurs $f_{/K_{ii}}^{(s)}$ sont fréquemment notés :*

$$S^{(s)} \equiv K_{/K_{ii}}^{(s)}, \quad b_p^{(s)} \equiv f_{/K_{ii}}^{(s)}$$

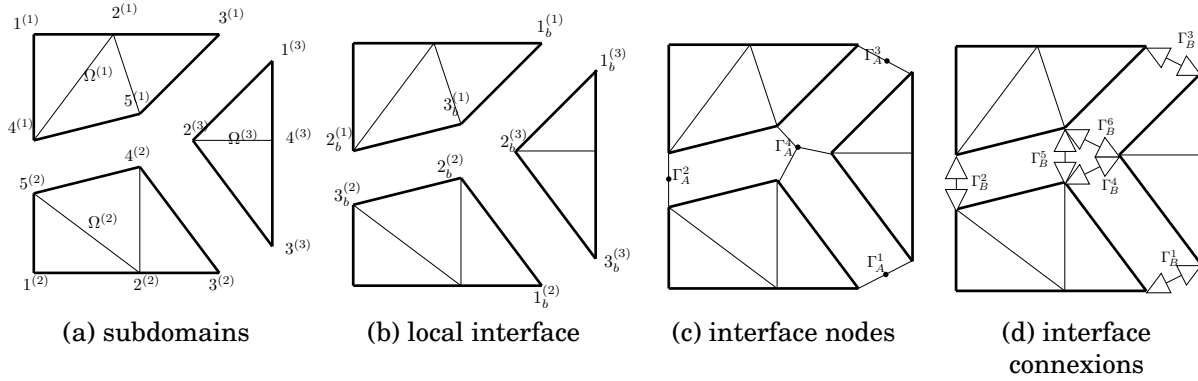
On utilisera cette notation dans la suite de ce manuscrit, ainsi que :

$$F^{(s)} \equiv K_{/K_{ii}}^{(s) \dagger}, \quad b_d^{(s)} \equiv -K_{/K_{ii}}^{(s) \dagger} b_p^{(s)}$$

pour la pseudo-inverse, ou complément de Schur dual, ainsi que la condensation duale des efforts extérieurs.

1.2.4 Notations générales

On va ici introduire les notations et objets utiles aux méthodes DD sans recouvrement dans le cas d'un nombre quelconque de sous-domaines n_{SD} .



$$\begin{aligned}
 t^{(1)} &= \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} & t^{(2)} &= \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} & t^{(3)} &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \\
 A^{(1)} &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} & A^{(2)} &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} & A^{(3)} &= \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \\
 B^{(1)} &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} & B^{(2)} &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix} & B^{(3)} &= \begin{pmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix}
 \end{aligned}$$

FIGURE 1.4 – Numérotations locale et d'interface, opérateurs de trace et d'assemblage

Numérotations Tout d'abord, on définit trois types de numérotations pour les nœuds du maillage (voir figure 1.4) :

- une numérotation locale des degrés de liberté sur chaque sous-maillage $\Omega_h^{(s)}$
- une numérotation locale des degrés de liberté de bord pour chaque sous-maillage $\Omega_h^{(s)}$
 - les interfaces locales seront notées $\Gamma^{(s)}$
 - les grandeurs de bord locales seront indicées $_b : x_b^{(s)} \equiv x_{|\Gamma^{(s)}}^{(s)}$
 - on note $n_b^{(s)}$ le nombre de degrés de liberté de bord d'une sous-structure $\Omega^{(s)}$
- une numérotation globale des interfaces
 - l'interface globale primale sera notée Γ_A , elle comprend la totalité des degrés de liberté d'interface ; on note n_A le nombre de degrés de liberté qui la composent.
 - l'interface globale duale sera notée Γ_B , elle peut être définie différemment suivant la description des multiplicités : elle comprend la totalité des degrés de liberté d'interface, potentiellement additionnée de degrés de liberté "fictifs" associés aux points multiples ; on note n_B le nombre de degrés de liberté qui la composent.

On introduit également la notation $\mathbb{R}_b^\Diamond$ pour décrire l'espace généré par les degrés de liberté des interfaces locales.

Trace, assemblage On définit également des opérateurs de trace $t^{(s)}$, d'assemblage primal $A^{(s)}$ et dual $B^{(s)}$. Les opérateurs de trace sont définis tels que pour une grandeur locale $u^{(s)}$ et la grandeur de bord associée $u_b^{(s)} \equiv u_{|\Gamma^{(s)}}^{(s)}$:

$$u_b^{(s)} = t^{(s)} u^{(s)}$$

Les opérateurs d'assemblage primal $A^{(s)}$ sont les opérateurs canoniques de prolongement de $\mathbb{R}_b^\Diamond$ dans $\mathbb{R}^{n_A}$: pour une sous-structure $\Omega^{(s)}$, $A^{(s)}$ est une matrice booléenne de taille $n_A \times n_b^{(s)}$ et de rang plein – un nœud de l'interface locale $\Gamma^{(s)}$ correspond à exactement un nœud de l'interface globale Γ_A .

On aura par exemple, dans le cas de trois sous-domaines sans points multiples :

Toute matrice $B^{(s)}$ satisfaisant à la relation $\text{Im} \left(\begin{bmatrix} \dots \\ B^{(s)T} \\ \dots \end{bmatrix} \right) = \text{Ker} ([\dots, A^{(s)}, \dots])$ peut être utilisée comme opérateur d'assemblage dual – la figure 1.4 représente le choix classiquement fait pour les matrices $B^{(s)}$.

On pourra avoir par exemple, dans le cas des trois sous-domaines sans points multiples évoqués ci-dessus :

$$\begin{array}{ccc}
\begin{array}{|c|c|c|} \hline \Omega^{(1)} & \Omega^{(2)} & \Omega^{(3)} \\ \hline \end{array} & \Rightarrow & \sum_s A^{(s)} v^{(s)} = \begin{bmatrix} \begin{bmatrix} v^{(1)} \end{bmatrix} \\ + \\ \begin{bmatrix} v^{(2)} \end{bmatrix} \\ + \\ \begin{bmatrix} v^{(3)} \end{bmatrix} \end{bmatrix}
\end{array}$$

$$\begin{array}{ccc}
\begin{array}{|c|c|c|} \hline \Omega^{(1)} & \Omega^{(2)} & \Omega^{(3)} \\ \hline \end{array} & \Rightarrow & \sum_s B^{(s)} v^{(s)} = \begin{bmatrix} \begin{bmatrix} v^{(1)} \end{bmatrix} \\ - \\ \begin{bmatrix} v^{(2)} \end{bmatrix} \\ + \\ \begin{bmatrix} v^{(3)} \end{bmatrix} \end{bmatrix}
\end{array}$$

Les grandeurs assemblées seront notées x_A ou x_B suivant l'interface globale sur laquelle elles seront définies :

$$\begin{aligned}
x_A &= \sum_s A^{(s)} x^{(s)} \in \mathbb{R}^{n_A} \\
x_B &= \sum_s B^{(s)} x^{(s)} \in \mathbb{R}^{n_B}
\end{aligned}$$

Reformulation du problème Les équilibres locaux (1.9) peuvent être reformulés par :

$$K^{(s)} u^{(s)} = t^{(s)T} B^{(s)T} \lambda_B + f^{(s)} \quad \forall s \in \{1, \dots, n_{SD}\}$$

Le problème condensé sur l'interface primale (1.13) peut être réécrit :

$$\sum_{s \in \{1, \dots, n_{SD}\}} A^{(s)} S^{(s)} A^{(s)T} u_A = \sum_{s \in \{1, \dots, n_{SD}\}} A^{(s)} b_p^{(s)}$$

De même, on peut reformuler le problème d'interface dual :

$$\sum_{s \in \{1, \dots, n_{SD}\}} B^{(s)} F^{(s)} B^{(s)T} \lambda_B = \sum_{s \in \{1, \dots, n_{SD}\}} B^{(s)} b_d^{(s)}$$

Notation diamant On utilisera dans la suite du manuscrit une notation commode pour les méthodes DD : la notation diamant. Elle est définie comme la concaténation des vecteurs locaux ou matrices locales, suivant la direction verticale, horizontale ou

diagonale du diamant :

$$x^\diamond = \begin{pmatrix} x^{(1)} \\ \vdots \\ x^{(n_{SD})} \end{pmatrix} \quad \text{et} \quad x^\diamond = (x^{(1)} \dots x^{(n_{SD})}),$$

$$M^\diamond = \begin{pmatrix} M^{(1)} & \dots & 0 \\ 0 & \ddots & \vdots \\ 0 & \dots & M^{(n_{SD})} \end{pmatrix}$$

Les équilibres locaux deviennent :

$$K^\diamond u^\diamond + t^{\diamond T} B^{\diamond T} \lambda_B = f^\diamond \quad (1.17)$$

Le problème d'interface primal devient :

$$A^\diamond S^\diamond A^{\diamond T} u_A = A^\diamond b_p^\diamond$$

et le problème d'interface dual :

$$B^\diamond F^\diamond B^{\diamond T} \lambda_B = B^\diamond b_d^\diamond$$

Remarque 5 : Deux propriétés de séparation de l'espace généré par les degrés de liberté des interfaces locales découlent des propriétés des opérateurs d'assemblage :

$$\mathbb{R}_b^\diamond = \text{Ker}(A^\diamond)^\perp \oplus \text{Ker}(B^\diamond) \quad (1.18)$$

$$\mathbb{R}_b^\diamond = \text{Im}(A^{\diamond T})^\perp \oplus \text{Im}(B^{\diamond T}) \quad (1.19)$$

En effet, on a la relation classique :

$$\text{Im}(A^{\diamond T})^\perp = \text{Ker}(A^\diamond) \Rightarrow \mathbb{R}_b^\diamond = \text{Im}(A^{\diamond T})^\perp \oplus \text{Ker}(A^\diamond)$$

Par définition de $B^\diamond$: $\text{Im}(B^{\diamond T}) = \text{Ker}(A^\diamond)$, on obtient (1.19).

Par la même propriété, on a $A^\diamond B^{\diamond T} = 0 = B^\diamond A^{\diamond T}$, d'où $\text{Im}(A^{\diamond T}) \subset \text{Ker}(B^\diamond)$.

Montrons $\text{Ker}(B^\diamond) = \text{Im}(A^{\diamond T})$: on suppose par l'absurde qu'il existe $v_b^\diamond \neq 0 \in \text{Ker}(B^\diamond)$, tel que $v_b^\diamond \notin \text{Im}(A^{\diamond T})$. D'après (1.19) :

$$\exists! (v_A, v_B) \in \mathbb{R}^{n_A} \times \mathbb{R}^{n_B}, \quad v_b^\diamond = A^{\diamond T} v_A + B^{\diamond T} v_B \text{ et } B^{\diamond T} v_B \neq 0$$

On a alors :

$$\begin{aligned} B^{\diamond} v_b^{\diamond} &= B^{\diamond} B^{\diamond^T} v_B = 0 \\ \Rightarrow v_B^T B^{\diamond} B^{\diamond^T} v_B &= 0 = \|B^{\diamond^T} v_B\|_2 \\ \Rightarrow B^{\diamond^T} v_B &= 0 \end{aligned}$$

D'où (1.18).

1.2.5 Résolution

On a vu que les méthodes de Schur permettaient d'obtenir un problème de taille réduite, condensé sur l'interface des sous-domaines. L'opérateur linéaire de ce problème est, dans le cas de la formulation primale :

$$S_A \equiv A^{\diamond} S^{\diamond} A^{\diamond^T} \quad (1.20)$$

et dans le cas d'une formulation duale :

$$F_B \equiv B^{\diamond} F^{\diamond} B^{\diamond^T} \quad (1.21)$$

Ce problème pourrait facilement être résolu par une méthode directe, cependant la difficulté réside dans le fait qu'on ne veut jamais assembler les opérateurs globaux (1.20) et (1.21). En effet, l'intérêt de ce type de méthode DD est de pouvoir paralléliser la résolution, en résolvant les équilibres locaux sur des processeurs différents simultanément, et de n'échanger que de faibles quantités d'informations entre les processeurs, grâce notamment à l'absence de recouvrement : ceci implique d'éviter tout assemblage de matrices sur la structure globale, et de n'effectuer que des produits matrice-vecteurs locaux.

On va alors utiliser un solveur itératif de type Krylov, facilement parallélisable. Par exemple, l'algorithme de Gradient Conjugué [Joly, 1986] ne requiert que des produits scalaires sur les vecteurs d'interface, et permet de maximiser les opérations locales : voir l'algorithme 3.

Remarque 6 : Dans l'algorithme 3, l'étape de réorthogonalisation est dite totale, i.e. on réorthogonalise la direction de recherche par rapport à toutes les autres direction de recherche des itérations précédentes. On peut également envisager de réorthogonaliser uniquement par rapport à l'itération précédente, ou encore par rapport à un certain nombre d'itérations, ou certaines itérations choisies (réorthogonalisations partielle ou sélective).

L'opération étoilée (*) de l'algorithme 3 correspond à l'unique produit matrice-vecteur de l'algorithme :

$$p_k = M r_k \quad (1.22)$$

Dans le cadre des méthodes DD de Schur, M correspond aux opérateurs (1.20) ou (1.21), et r_k à un résidu défini sur l'interface : r_A ou r_B suivant l'approche primale ou duale. On voit donc que l'opération (1.22) peut être effectuée essentiellement par des produits locaux matrice-vecteur, ne nécessitant qu'un seul assemblage de vecteurs d'interface. Par exemple en primal :

$$S_A r_A = \underbrace{A^{\diamond} S^{\diamond} A^{\diamond T} r_A}_{\text{vecteurs de bord locaux } r_b^{\diamond}} = \sum_{s \in \{1, \dots, n_{SD}\}} \underbrace{A^{(s)} S^{(s)} r_b^{(s)}}_{\text{produits matrice-vecteur locaux}}$$

Les opérations munies de doubles astérisques (**) correspondent aux produits scalaires de vecteurs d'interface. De même que pour le produit matrice-vecteur, on peut transformer cette opération a priori globale en opérations locales parallèles, plus un assemblage.

Remarque 7 : Pour plus de détails sur les méthodes de Krylov, le lecteur pourra se référer au livre [Saad, 2003].

Algorithm 3 Conjugate Gradient

Solve linear system $Mx = b$:

initialization : iterate x_0 , threshold ε , $k = 0$

compute residual $r_0 = b - Mx_0$

set search direction $w_0 = r_0$

while $\|r_k\| > \varepsilon$ **do**

compute $p_k = Mr_k$ (*)

compute $g_k = p_k^T w_k$ (**)

compute parameter $\alpha = \frac{r_k^T r_k}{g_k}$ (**)

actualize residual $r_{k+1} = r_k - \alpha p_k$

and iterate $x_{k+1} = x_k + \alpha w_k$

set new search direction $w_{k+1} = r_{k+1}$

complete reorthogonalization :

for $i \in \{1, \dots, k-1\}$ **do**

compute $\beta_i = \frac{w_{k+1}^T p_i}{g_i}$

actualize new search direction $w_{k+1} -= \beta_i * w_i$ (**)

$k += 1$

1.2.6 Préconditionnement

Si les méthodes DD de Schur permettent de réduire la taille du problème, rien n'indique que le nouveau problème condensé à l'interface aura un bon conditionnement, donc que le solveur de Krylov convergera vers la solution en un nombre raisonnable d'itérations.

On va donc avoir recours, pour améliorer la vitesse de résolution du solveur, à une technique de preconditionnement.

De façon classique, preconditionner un système $Mx = b$ consiste à le multiplier à gauche ou à droite par une matrice $\tilde{M}^{-1}$ telle que :

$$\text{Cond}(\tilde{M}^{-1}M) \ll \text{Cond}(M)$$

On résout alors le système – preconditionnement à gauche :

$$\tilde{M}^{-1}Mx = \tilde{M}^{-1}b$$

Les méthodes de Schur présentent, dans ce contexte, un avantage intéressant : on peut obtenir à moindre coût un bon preconditionneur du problème d'interface. En effet, les compléments de Schur duaux locaux sont les pseudo-inverses des compléments de Schur primaux :

$$F^{(s)} = S^{(s)\dagger}$$

On peut donc approximer l'inverse de l'opérateur de Schur primal assemblé en fonction des opérateurs de Schur locaux :

$$\begin{aligned} S_A^{-1} &= \left(A^{\diamond} S^{\diamond} A^{\diamond T} \right)^{-1} \simeq \left(A^{\diamond T} \right)^{\dagger} \left(S^{\diamond} \right)^{\dagger} \left(A^{\diamond} \right)^{\dagger} \\ &\simeq \left(A^{\diamond T} \right)^{\dagger} \left(S^{\dagger} \right)^{\diamond} \left(A^{\diamond} \right)^{\dagger} = \left(A^{\diamond T} \right)^{\dagger} F^{\diamond} \left(A^{\diamond} \right)^{\dagger} \end{aligned}$$

On a la propriété équivalente en formulation duale :

$$F_B^{-1} = \left(B^{\diamond T} F^{\diamond} B^{\diamond} \right)^{-1} \simeq \left(B^{\diamond T} \right)^{\dagger} S^{\diamond} \left(B^{\diamond} \right)^{\dagger}$$

On définit des opérateurs d'assemblage dits *pondérés* par :

$$\begin{aligned} \tilde{A}^{\diamond} &= \left(A^{\diamond T} \right)^{\dagger} \\ \tilde{B}^{\diamond} &= \left(B^{\diamond T} \right)^{\dagger} \end{aligned}$$

On peut par exemple, pour évaluer $\tilde{A}^{\diamond}$ et $\tilde{B}^{\diamond}$, utiliser une expression de la forme :

$$\begin{aligned} \tilde{A}^{\diamond} &= \left(A^{\diamond} \Delta^{\diamond} A^{\diamond T} \right)^{\dagger} A^{\diamond} \Delta^{\diamond} \\ \tilde{B}^{\diamond} &= \left(B^{\diamond} \Delta^{\diamond -1} B^{\diamond T} \right)^{\dagger} B^{\diamond} \Delta^{\diamond -1} \end{aligned} \tag{1.23}$$

ce qui permet de vérifier les propriétés fondamentales (voir [Klawonn and Widlund, 2001b]) :

$$A^\diamond \tilde{A}^{\diamond T} = I \quad (1.24)$$

$$B^\diamond \tilde{B}^{\diamond T} B^\diamond = B^\diamond \text{ et } \tilde{B}^\diamond B^{\diamond T} \tilde{B}^\diamond = \tilde{B}^\diamond \quad (1.25)$$

et on définit les préconditionneurs primal et dual par :

$$\tilde{S}_A^{-1} = \tilde{A}^\diamond F^\diamond \tilde{A}^{\diamond T} \quad (1.26)$$

$$\tilde{F}_B^{-1} = \tilde{B}^\diamond S^\diamond \tilde{B}^{\diamond T} \quad (1.27)$$

Les opérateurs d'assemblage vérifient alors la propriété supplémentaire :

$$\tilde{A}^{\diamond T} A^\diamond + \tilde{B}^{\diamond T} B^\diamond = I \quad (1.28)$$

Remarque 8 : Les opérateurs $\Delta^{(s)}$ introduits dans l'expression (1.23) permettent de tenir compte d'éventuelles hétérogénéités entre les sous-structures :

- *structures homogènes : on prend par exemple $\Delta^{(s)} = I^{(s)}$ pour chaque sous-structure $\Omega^{(s)}$*
- *structures hétérogènes : on peut prendre $\Delta^{(s)} = \text{diag}(K_{bb}^{(s)})$ par exemple pour chaque sous-structure $\Omega^{(s)}$*

En utilisant les opérateurs (1.26) et (1.27), on peut réaliser l'étape de préconditionnement du système en effectuant uniquement des résolutions locales – munies de conditions de Neumann dans le cas de l'approche primale, et de Dirichlet dans celui de l'approche duale. Les méthodes bien connues BDD et FETI se basent sur ce principe de préconditionnement : Le système d'interface préconditionné devient :

$$\text{approche primale : } \tilde{S}_A^{-1} S_A u_A = \tilde{S}_A^{-1} A^\diamond b_p^\diamond \quad (1.29)$$

$$\text{approche duale : } \tilde{F}_B^{-1} F_B \lambda_B = \tilde{F}_B^{-1} B^\diamond b_d^\diamond \quad (1.30)$$

Les algorithmes (4) et (5) détaillent le processus de résolution de ces solveurs. On y introduira les notations :

$$\begin{aligned} \lambda_D^{(s)} = \text{Solve}_D(u_b^{(s)}, f^{(s)}) & : \text{équilibres locaux} \\ & \quad - \text{conditions de Dirichlet } u_b^{(s)} \\ & \quad - \text{efforts locaux } f^{(s)} \\ u_N^{(s)} = \text{Solve}_N(\lambda_b^{(s)}, f^{(s)}) & : \text{équilibres locaux} \\ & \quad - \text{conditions de Neumann } \lambda_b^{(s)} \\ & \quad - \text{efforts locaux } f^{(s)} \end{aligned}$$

Depuis les travaux de Mandel au début des années 1990 [Mandel, 1993], de nombreuses variantes de ces algorithmes sont apparues, permettant de les rendre plus extensibles – par optimisation des préconditionneurs et introduction de problèmes grossiers notamment [Gosselet and Rey, 2006, Farhat et al., 2001, Rixen and Farhat, 1999].

La possibilité de traiter des seconds membres multiples, grâce à des techniques de déflation ou de solveurs-blocs [O’Leary, 1980, Saad, 1987, Morgan, 2005, Soodhalter, 2013], ou encore la réutilisation d’informations entre résolutions successives pour augmenter les espaces de Krylov [Risler and Rey, 2000, Gosselet et al., 2013] ou définir de meilleurs préconditionneurs [Rey, 1996], participèrent également à l’essor des méthodes DD de Schur.

1.2.7 Modes rigides, projecteurs

Formulation duale Les méthodes DD de Schur avec formulation duale doivent toujours être associées à une stratégie de traitement des modes rigides [Farhat and Roux, 1991, Farhat et al., 1993, Gosselet and Rey, 2006]. En effet, les équilibres locaux avec conditions de bord de Neumann n’ont pas de solution unique, si la sous-structure n’a pas de nœuds communs avec le bord de la structure globale soumis à des conditions de Dirichlet $\partial\Omega_u$. La solution locale est dans ce cas définie à un mode rigide près, et l’opérateur de Schur dual n’est pas inversible. On note $R^{(s)}$ le noyau de la matrice de rigidité $K^{(s)}$ de la sous-structure $\Omega^{(s)}$, on veut :

$$\begin{cases} u_b^{(s)} = F^{(s)} \left(b_p^{(s)} + B^{(s)T} \lambda_B \right) + R_b^{(s)} \alpha^{(s)} \\ R_b^{(s)T} \left(b_p^{(s)} + B^{(s)T} \lambda_B \right) = 0 \end{cases} \quad (1.31)$$

ou encore :

$$\begin{bmatrix} F_B & G_B \\ G_B^T & 0 \end{bmatrix} = \begin{bmatrix} \lambda_B \\ \alpha^\diamond \end{bmatrix} = \begin{bmatrix} B^\diamond b_p^\diamond \\ -e^\diamond \end{bmatrix}$$

où les modes rigides d’interface assemblés sont notés $G_B = B^\diamond R_b^\diamond$ et $e^\diamond = R_b^{\diamond T} b_p^\diamond$ tient lieu de second membre pour la condition d’admissibilité sur λ_B : cette condition traduit le fait qu’on n’excite pas les modes rigides de la structure. En pratique, elle est mise en œuvre par un projecteur P_B sur $\text{Ker}(R_b^{\diamond T} B^{\diamond T})$ et par une initialisation permettant de capter la partie de la solution liée aux efforts $b_p^\diamond$. On cherche alors $\lambda_B = \lambda_{B_0} + P_B d\lambda_B$.

Remarque 9 : De manière générale, les stratégies de projection sont fréquemment utilisées au sein des algorithmes de Krylov pour prendre en compte une contrainte d’admissibilité linéaire ou affine sur la solution. Les stratégies dites d’augmentation [Saad, 1997, Chapman and Saad, 1997], mises en œuvre par projection ou réorthogonalisation, permettent également la vérifica-

tion de contraintes d'admissibilité ou d'optimalité sur l'inconnue ; un couplage des stratégies de projection et d'augmentation peut être envisagé dans le cas de contraintes imposées d'admissibilité et d'optimalité [Gosselet and Rey, 2006].

Cette stratégie, dite de projection, permet de définir, lors des résolutions locales, la contribution du déplacement solution $u_b^{(s)}$ dans un espace orthogonal à l'espace des modes rigides de la sous-structure. La contribution de la solution assemblée $B^\diamond u_b^\diamond$ dans l'espace des modes rigides assemblés $\text{span}(G_B) \equiv \text{span}(B^\diamond R_b^\diamond)$ est ensuite déterminée grâce à une condition de minimisation du saut de déplacement aux interfaces :

$$\begin{aligned}\alpha_B &= \text{argmin} \|B^\diamond u_b^\diamond\|_Q = \text{argmin} \|B^\diamond F^\diamond (b_p^\diamond + B^{\diamond T} \lambda_B) + \alpha_B G_B\|_Q \\ &\equiv \text{argmin} \|r_B + \alpha_B G_B\|_Q\end{aligned}$$

où $\|\cdot\|_Q$ est une norme matricielle associée à un opérateur Q , tel que $G_B^T Q G_B$ soit inversible, permettant de fournir de l'information sur les autres sous-domaines. Dans l'idéal, on prend $Q = \tilde{B}^\diamond S^\diamond \tilde{B}^{\diamond T}$, ce qui permet de minimiser le saut de déplacement entre les sous-structures en fonction des écarts de rigidité entre les sous-structures. On peut également utiliser des expressions du type $Q = \tilde{B}^\diamond \text{diag}(K_{bb})^\diamond \tilde{B}^{\diamond T}$ pour minimiser les coûts de calcul.

On trouve :

$$\alpha_B = (G_B^T Q G_B)^{-1} G_B^T r_B$$

Et en posant :

$$\begin{cases} P_B = I - Q G_B (G_B^T Q G_B)^{-1} G_B^T \\ \lambda_{B_0} = -Q G_B (G_B^T Q G_B)^{-1} R_b^{\diamond T} b_p^\diamond \end{cases} \quad (1.32)$$

On a :

$$\begin{cases} G_B^T P_B = 0 \\ P_B^T r_B = r_B + \alpha_B G_B \end{cases}$$

On peut alors reformuler le problème dual :

$$P_B^T F_B P_B \lambda_B = P_B^T (B^\diamond b_d^\diamond - F_B \lambda_{B_0})$$

Formulation primale Dans le cas de la formulation primale, le problème grossier intervient au moment du préconditionnement par l'opérateur de Schur dual [Mandel, 1993, Le Tallec, 1994, Gosselet and Rey, 2006]. On doit alors s'assurer que le vecteur que l'on donne en entrée de l'opérateur de préconditionnement $\tilde{A}^\diamond F^\diamond \tilde{A}^{\diamond T}$ est orthogonal à l'espace des modes rigides pondérés $\tilde{G}_A^\diamond \equiv \tilde{A} R_b^\diamond$. Ceci se traduit par la condition d'optimalité :

$$\tilde{G}_A^T S_A u_A \equiv \tilde{G}_A^T r_A = 0$$

On introduit pour cela un projecteur P_A^T sur $\text{Ker}(\tilde{G}_A^T S_A)$:

$$P_A^T = I - S_A \tilde{G}_A (\tilde{G}_A^T S_A \tilde{G}_A)^{-1} \tilde{G}_A^T$$

Et on reformule le problème primal :

$$P_A^T S_A u_A = P_A^T A^\diamond b_p^\diamond$$

La perte de symétrie du nouvel opérateur primal n'est qu'apparente, étant donnée l'égalité :

$$P_A^T S_A = P_A^T S_A P_A$$

La conservation des propriétés de symétrie du problème permet d'appliquer les solveurs de Krylov dédiés.

Lien avec le préconditionneur et augmentation par GenEO Le système préconditionné primal (1.29) peut être réécrit en prenant en compte les modes rigides :

$$\tilde{S}_A^{-1} P_A^T S_A u_A = \tilde{S}_A^{-1} P_A^T A^\diamond b_p^\diamond$$

On peut alors choisir d'exprimer le préconditionneur (1.26) – qui correspondait au cas idéal sans modes rigides – comme une somme directe entre un terme appartenant à l'espace $\text{Ker}(\tilde{G}_A^T S_A)$ et un terme orthogonal :

$$\tilde{S}_A^{-1} \equiv P_A M_1^{-1} P_A^T + (I - P_A) M_2^{-1} (I - P_A)^T$$

En prenant $M_1^{-1} = \tilde{A}^\diamond F^\diamond \tilde{A}^{\diamond T}$ et $M_2 = S_A$, on obtient le préconditionneur BDD projeté :

$$\tilde{S}_A^{-1} = P_A \tilde{A}^\diamond F^\diamond \tilde{A}^{\diamond T} P_A^T + \tilde{G}_A (\tilde{G}_A^T S_A \tilde{G}_A)^{-1} \tilde{G}_A^T$$

et on trouve la condition d'initialisation :

$$u_0 = \tilde{G}_A (\tilde{G}_A^T S_A \tilde{G}_A)^{-1} \tilde{G}_A^T A^\diamond b_p^\diamond$$

Dans le cas de structures hétérogènes partitionnées en un grand nombre de sous-domaines, le conditionnement des opérateurs, même préconditionnés, n'est plus garanti. Les techniques GenEO consistent à augmenter le problème grossier – i.e. $\tilde{G}_A$ – afin d'obtenir une borne supérieure de ce conditionnement :

$$\kappa(\tilde{S}_A^{-1} S_A) \leq C(\mathcal{N}^\diamond, \gamma^\diamond)$$

où $\mathcal{N}^{(s)}$ représente, pour un sous-domaine donné, le nombre de ses voisins, et les $\gamma^{(s)}$ sont des seuils permettant de définir la taille des espaces GenEO pour chaque sous-domaine [Spillane and Rixen, 2013].

Pour la formulation duale, le préconditionneur peut être exprimé plus simplement :

$$\tilde{F}_B^{-1} = P_B \tilde{B}^\diamond S^\diamond \tilde{B}^{\diamond T} P_B^T$$

De même que pour la formulation primale, on peut ajouter à cette expression un terme tenant compte des espaces grossiers GenEO et permettant de garantir le conditionnement de l'opérateur FETI préconditionné projeté.

1.2.8 Algorithmes

Algorithm 4 Balancing Domain Decomposition**initialization :**

interface main unknown

$$u_A = \tilde{G}_A (\tilde{G}_A^T S_A \tilde{G}_A)^{-1} \tilde{G}_A^T A \diamond b_p^\diamond$$

threshold ε

local Dirichlet equilibriums

$$\lambda_D^\diamond = \text{Solve}_D(A \diamond^T u_A, f^\diamond)$$

assemble residual

$$r_A = A \diamond \lambda_D^\diamond$$

precondition :

split residual (scaled local nodal reactions)

$$\tilde{\lambda}_b^\diamond = \tilde{A} \diamond^T r_A$$

local Neumann equilibriums

$$u_N^\diamond = \text{Solve}_N(\tilde{\lambda}_b^\diamond, 0^\diamond)$$

preconditioned residual

$$z_A = \tilde{A} \diamond u_N^\diamond$$

search direction

$$w_A = P_A z_A$$

while $\sqrt{r_A^T z_A} > \varepsilon$ **do**

local Dirichlet equilibriums

$$\lambda_D^\diamond = \text{Solve}_D(A \diamond^T w_A, 0)$$

assemble

$$p_A = A \diamond \lambda_D^\diamond$$

scalar product (orthogonalization parameter)

$$\alpha = (r_A^T z_A) / (p_A^T w_A)$$

actualize iterate

$$u_A + = \alpha w_A$$

residual

$$r_A - = \alpha p_A$$

precondition :

split residual (scaled local nodal reactions)

$$\tilde{\lambda}_b^\diamond = \tilde{A} \diamond^T r_A$$

local Neumann equilibriums

$$u_N^\diamond = \text{Solve}_N(\tilde{\lambda}_b^\diamond, 0)$$

preconditioned residual

$$z_A = \tilde{A} u_N^\diamond$$

store old search direction

$$w_A^{old} = w_A$$

new search direction

$$w_A = P_A z_A$$

reorthogonalization :

search direction

$$w_A - = (p_A^T w_A) / (p_A^T w_A^{old}) w_A^{old}$$

Algorithm 5 Finite Element Tearing and Interconnecting

initialization :

interface main unknown	$\lambda_B = -QG_B (G_B^T QG_B)^{-1} e^\diamond$
threshold	
local Neumann equilibriums	$u_N^\diamond = \text{Solve}_N(B^{\diamond T} \lambda_B, f^\diamond)$
assemble residual	$r_B = P_B^T B^\diamond u_N^\diamond$

precondition :

split residual (scaled local displacement)	$\tilde{u}_b^\diamond = \tilde{B}^{\diamond T} r_B$
local Dirichlet equilibriums	$\lambda_D^\diamond = \text{Solve}_D(\tilde{u}_b^\diamond, 0^\diamond)$
preconditioned and projected residual	$z_B = P_B \tilde{B}^\diamond \lambda_D^\diamond$
search direction	$w_B = z_B$

while $\sqrt{r_B^T z_B} > \varepsilon$ **do**

local Neumann equilibriums	$u_N^\diamond = \text{Solve}_N(B^{\diamond T} w_B, 0)$
assemble	$p_B = P_B^T B^\diamond u_N^\diamond$
scalar product (orthogonalization parameter)	$\alpha = (r_B^T z_B) / (p_B^T w_B)$
actualize iterate	$\lambda_B \leftarrow \lambda_B + \alpha w_B$
residual	$r_B \leftarrow r_B - \alpha p_B$

precondition :

split residual (scaled local displacements)	$\tilde{u}_b^\diamond = \tilde{B}^{\diamond T} r_B$
local Dirichlet equilibriums	$\lambda_D^\diamond = \text{Solve}_D(\tilde{u}_b^\diamond, 0)$
preconditioned and projected residual	$z_B = P_B \tilde{B}^\diamond \lambda_D^\diamond$
store old search direction	$w_B^{old} = w_B$
new search direction	$w_B = z_B$

reorthogonalization :

search direction	$w_B \leftarrow (p_B^T z_B) / (p_B^T w_B^{old}) w_B^{old}$
------------------	--

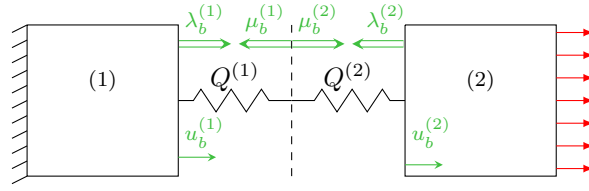


FIGURE 1.5 – Conditions de Robin : modélisation ressorts

1.2.9 Conditions d'interface mixtes

On a montré que l'on pouvait, dans le cadre des méthodes DD de Schur, indépendamment choisir de travailler sur une inconnue d'interface en déplacement ou en effort – approche primale ou duale : la vitesse de convergence théorique des méthodes BDD et FETI est identique [Farhat et al., 1994, Klawonn and Widlund, 2001a, Brenner, 2007, Mandel et al., 2005]. Cependant un troisième choix d'inconnue d'interface existe, qui peut potentiellement posséder des propriétés de convergence plus avantageuses : une combinaison linéaire de déplacements et d'efforts. Ces conditions sont appelées conditions mixtes, ou de Robin.

On a vu précédemment, dans le cadre des méthodes DD de Schwarz, que des conditions de transmission mixtes entre les sous-domaines permettaient la recherche d'un optimum qui accélérât grandement la convergence, et qui permettait également de recourir à un recouvrement nul, donc une diminution des communications.

Dans le cadre des méthodes DD de Schur, on introduit l'inconnue d'interface $\mu_b^\diamond$, telle que :

$$\mu_b^\diamond = \lambda_b^\diamond + Q_b^\diamond u_b^\diamond$$

Le paramètre $Q_b^{(s)}$ est l'analogue du paramètre α de la section 1.1.4 (chap. 1). Il est ici homogène à une rigidité d'interface : on l'appelle généralement *impédance d'interface*. En effet, l'imposition de la condition $\mu_b^{(s)}$ sur le bord d'un sous-domaine $\Omega^{(s)}$ revient à relier ce sous-domaine au reste de la structure par des ressorts de rigidité $Q_b^{(s)}$ et à imposer une condition de traction à l'extrémité de ce ressort – voir la figure 1.5.

L'intérêt principal de l'utilisation de conditions mixtes provient de la possibilité d'optimisation offerte par ce paramètre : un choix judicieux de $Q_b^{(s)}$ pour chaque sous-domaine permet d'accélérer la résolution. En effet, les équilibres locaux (1.17) peuvent être reformulés :

$$\begin{cases} K^\diamond u^\diamond = t^{\diamond T} (\mu_b^\diamond - Q_b^\diamond u_b^\diamond) + f^\diamond \\ + \text{C.L.} \end{cases} \Rightarrow \begin{cases} (K^\diamond + t^{\diamond T} Q_b^\diamond t^\diamond) u^\diamond = t^{\diamond T} \mu_b^\diamond + f^\diamond \\ + \text{C.L.} \end{cases}$$

Pour des décompositions structurelles dont le graphe des connectivités ne contient pas de cycles, on peut montrer [Magoulès et al., 2004, Gander and Kwok, 2011] que définir $Q_b^{(s)}$ comme le complément de Schur $S^{(\bar{s})}$ du complémentaire de la sous-structure $\Omega^{(s)}$

dans Ω , condensé sur l'interface $\Gamma^{(\bar{s})} = \Gamma^{(s)}$, permet d'assurer la convergence de la résolution en un nombre maximum d'itérations égal au nombre de sous-domaines (autant que nécessaires à la propagation du second membre dans toutes les sous-structures). En effet, cette valeur du paramètre $Q_b^\diamond$ permet de transformer l'équilibre local sur $\Omega^{(s)}$ en équilibre global de la structure (basé sur la rigidité totale).

Remarque 10 : Lorsque des points multiples sont présents au sein de la sous-structuration, la vitesse de convergence n'est plus bornée par le nombre de sous-domaines [Nier, 1998, Gander and Kwok, 2011]. De façon plus générique, la valeur optimale du paramètre $Q_b^\diamond$ n'est pas un opérateur matriciel mais affine, discrétisation de l'opérateur sous-jacent de Steklov-Poincaré (Dirichlet-to-Neumann) :

$$Q_b^{(s)} \left(u_b^{(s)} \right) = S^{(\bar{s})} u_b^{(s)} - b^{(\bar{s})}$$

où $b^{(\bar{s})}$ condense les efforts $f^{(\bar{s})}$ du complémentaire $\Omega^{(\bar{s})}$ sur l'interface $\Gamma^{(\bar{s})}$: $b^{(\bar{s})} = f_b^{(\bar{s})} - K_{bi}^{(\bar{s})} K_{ii}^{(\bar{s})^{-1}} f_i^{(\bar{s})}$ – les indices x_i (resp. x_b) faisant ici référence aux nœuds internes au (resp. de bord du) complémentaire $\Omega^{(\bar{s})}$. La reformulation du problème avec cette impédance permet, pour une initialisation nulle de l'inconnue d'interface $\mu_b^{(s)}$, une convergence en au plus une itération d'un solveur DD comme FETI-2LM [Roux, 2009], quel que soit le découpage choisi en sous-domaines.

Le calcul du complément de Schur du complémentaire d'une sous-structure est bien entendu proscrit dans le cadre des méthodes de résolution parallèles, car il nécessite la donnée et la manipulation de grandeurs matricielles globales sur la structure entière, cependant des techniques d'approximations sont possibles, permettant d'accélérer la résolution à moindre coût.

On peut définir, par analogie avec les approches primale et duale, un opérateur de condensation local $H^\diamond$, que l'on pourrait qualifier de complément de Schur mixte, reliant les grandeurs de bord $\mu_b^\diamond$ et $u_b^\diamond$:

$$u_b^\diamond = H^\diamond \mu_b^\diamond + b_m^\diamond$$

$$H^\diamond = t^\diamond \left(K^\diamond + t^{\diamond T} Q_b^\diamond t^\diamond \right)^{-1} t^{\diamond T}$$

Le vecteur $b_m^{(s)}$ est le second membre condensé :

$$b_m^\diamond = t^\diamond \left(K^\diamond + t^{\diamond T} Q_b^\diamond t^\diamond \right)^{-1} f^\diamond$$

Et le problème d'interface devient :

$$\left(A^{\diamond T} \left(A^\diamond Q_b^\diamond A^{\diamond T} \right)^{-1} A^\diamond - H^\diamond \right) \mu_b^\diamond = b_m^\diamond \quad (1.33)$$

Par exemple pour deux sous-domaines sans points multiples on a :

$$\begin{cases} (Q^{(1)} + Q^{(2)})^{-1} (\mu_b^{(1)} + \mu_b^{(2)}) = H^{(1)} \mu_b^{(1)} + b_m^{(1)} \\ (Q^{(1)} + Q^{(2)})^{-1} (\mu_b^{(1)} + \mu_b^{(2)}) = H^{(2)} \mu_b^{(2)} + b_m^{(2)} \end{cases}$$

Remarque 11 : Afin d'assurer l'existence d'une solution au problème condensé (1.33), on impose les conditions de symétrie, définition et positivité à la matrice $Q_b^\diamond$. On obtient alors – entre autres propriétés – l'inversibilité de la matrice assemblée $(A^\diamond Q_b^\diamond A^{\diamond T})$:

$$\begin{aligned} \forall x_A \in \mathbb{R}^{n_A}, \quad A^\diamond Q_b^\diamond A^{\diamond T} x_A = 0 &\Leftrightarrow x_A^T A^\diamond Q_b^\diamond A^{\diamond T} x_A = 0 \Leftrightarrow \|A^{\diamond T} x_A\|_{Q_b^\diamond} = 0 \\ &\Rightarrow A^{\diamond T} x_A = 0 \quad (Q_b^\diamond \text{ SPD}) \Rightarrow x_A = 0 \quad (A^\diamond \text{ de rang plein}) \end{aligned}$$

2 Méthodes de résolution des systèmes non linéaires

Le deuxième thème autour duquel s'organisera ce manuscrit concerne la résolution des systèmes non linéaires. L'importance de leur traitement dans la simulation numérique devient d'autant plus grande que les phénomènes abordés dans le cadre du *virtual testing* sont de plus en plus complexes. Que ce soient en termes de non linéarités matériau – plasticité mécanique, diffusion ionique, thermique non linéaire, ... – ou géométriques – grandes déformations, grands déplacements, ... – les applications sont variées, et les industriels tendent de plus en plus à prendre en compte ces phénomènes dans leurs études.

Cependant, le traitement des non linéarités est encore trop souvent synonyme de temps de calcul allongé, voire inacceptable, notamment dans le cas de systèmes à grand nombre de degrés de liberté. Le recours à des techniques de parallélisation est employé pour améliorer la rapidité des solveurs, mais potentiellement mal exploité [Negrello et al., 2016].

On s'attachera d'abord dans cette section, dans un cadre très général, à détailler les méthodes de résolution de tout type de problème non linéaire. Puis on se concentrera sur les méthodes de Newton, qui seront largement réutilisées dans la suite de ce manuscrit. Enfin, on verra comment les méthodes DD sont intégrées à la résolution non linéaire, et comment les méthodes exploitées dans la suite de ce manuscrit permettraient d'apporter une amélioration conséquente des performances de convergence non linéaire parallèle.

2.1 Méthodes de point fixe

Depuis le 18^e siècle, les algorithmes de résolution des systèmes non linéaires évoluent. Des méthodes de point fixe au performant algorithme de Newton ainsi que ses variantes, nous allons ici donner un aperçu de l'état de l'art dans le domaine de la résolution d'équations différentielles non linéaires.

2.1.1 Définitions

On se donne ici une fonction $g \in \mathcal{C}(\mathbb{R}^n, \mathbb{R}^m)$ continue de $\mathbb{R}^n$ dans $\mathbb{R}^m$. On cherche à trouver x solution de :

$$\begin{cases} x \in \mathbb{R}^n \\ g(x) = 0 \end{cases} \quad (1.34)$$

Point fixe de contraction On définit la fonction $f \in \mathcal{C}(\mathbb{R}^n, \mathbb{R}^n)$ telle que $\forall x \in \mathbb{R}^n, f(x) = x - g(x)$. On peut remarquer que $g(x) = 0$ si et seulement si $f(x) = x$. On a donc équivalence des systèmes :

$$g(x) = 0 \quad \Leftrightarrow \quad f(x) = x \quad (1.35)$$

Ceci revient à dire que trouver un zéro de g est équivalent à trouver un point fixe de f , si celui-ci existe – on rappelle à ce sujet le théorème de point fixe :

Théorème 2.1.1. *Soit E un espace métrique complet, d la distance sur E , et $f : E \rightarrow E$ strictement contractante, c'est-à-dire que :*

$$\exists \kappa \in]0, 1[, \forall (x, y) \in E^2, \quad d(f(x), f(y)) \leq \kappa d(x, y)$$

Alors il existe un unique point fixe $x^ \in E$ qui vérifie $f(x^*) = x^*$.*

De plus la suite $(x_k)_{k \geq 0}$ définie par :

$$\begin{cases} x_0 \in E \\ x_{k+1} = f(x_k) \quad \forall k > 0 \end{cases}$$

converge vers x^ :*

$$x_k \xrightarrow[k \rightarrow \infty]{} x^*$$

En prenant $E = \mathbb{R}^n$ et pour d la distance associée à la norme euclidienne, notée $\| \cdot \|$, la méthode du *point fixe*, ou méthode des itérations successives, consiste à calculer les itérés de la suite $(x_k)_{k \geq 0}$ du Théorème 2.1.1 pour trouver le zéro d'une fonction $g : \mathbb{R}^n \rightarrow \mathbb{R}^n$.

Le lecteur pourra se référer à la figure 1.6 pour un exemple de recherche de point fixe avec différentes conditions initiales. La fonction $f : x \mapsto x^2$ possède deux points

fixes : $f(0) = 0$ et $f(1) = 1$. Elle n'est strictement contractante que sur l'intervalle $]-\frac{1}{2}, \frac{1}{2}[$, on n'a donc de certitude sur la convergence de l'algorithme de point fixe que si l'itéré initial x_0 appartient à cet intervalle :

$$\begin{aligned} \forall (x, y) \in I_c \times I_c, \quad \|x^2 - y^2\| < \|x - y\| &\Leftrightarrow \|(x + y)(x - y)\| < \|x - y\| \xRightarrow{x \neq y} \|x + y\| < 1 \\ &\Rightarrow I_c = \left] -\frac{1}{2}, \frac{1}{2} \right[\end{aligned}$$

Remarque 12 : Soient $E = \mathbb{R}^n$ espace métrique complet pour la norme $\| \cdot \|$ et une fonction $f : E \rightarrow E$, dérivable sur E_1 sous-espace complet pour $\| \cdot \|$ de E . On suppose de plus que :

$$\forall x \in E_1, \quad \|f'(x)\| \leq 1 \quad (1.36)$$

où f' est la dérivée de f si $n = 1$, et sa matrice jacobienne si $n > 1$. Par corollaire du théorème des accroissements finis, f est strictement contractante sur E_1 .

2.1.2 Relaxation

Si la fonction $f = x - g(x)$ n'est pas contractante, on peut considérer une fonction paramétrée $f_\omega = x - \omega g(x)$. De même que précédemment, chercher les zéros de g dans $\mathbb{R}^n$ revient à chercher les points fixes de f_ω dans $(\mathbb{R}^n, \| \cdot \|)$, s'ils existent. On a alors la possibilité d'ajuster le paramètre ω afin d'obtenir une fonction f_ω satisfaisant aux conditions d'existence de point fixe dans l'espace considéré, c'est-à-dire que f_ω soit contractante dans $(\mathbb{R}^n, \| \cdot \|)$.

On peut aisément montrer que si g satisfait aux conditions :

- g est lipschitzienne de rapport $M > 0$
- $\exists \alpha$ tel que $\forall (x, y) \in \mathbb{R}^n \times \mathbb{R}^n, \quad (g(x) - g(y)) \cdot (x - y) \geq \alpha \|x - y\|^2$

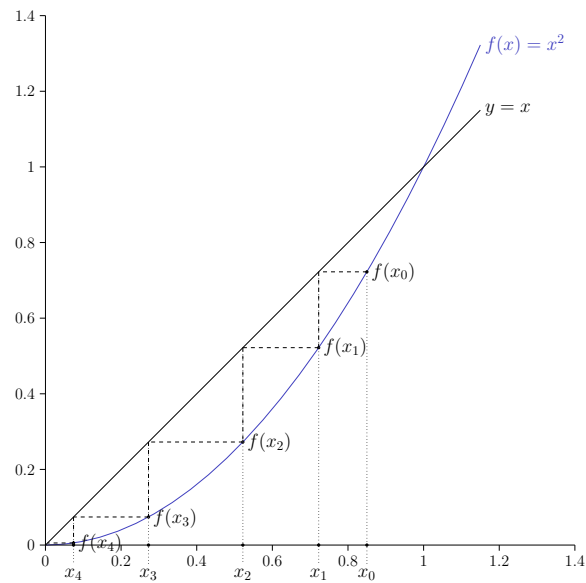
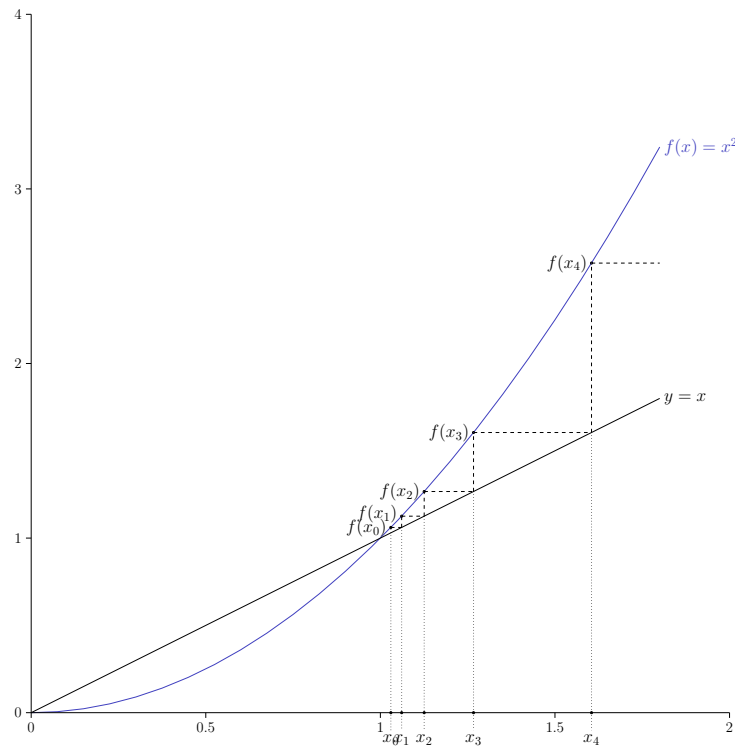
alors prendre $\omega \in]0, \frac{2\alpha}{M}[$ permet d'obtenir la propriété de contractance de f_ω .

On obtient alors l'algorithme de point fixe avec *relaxation* (le paramètre ω est dit de *relaxation*) en construisant la suite :

$$\begin{cases} x_0 \in \mathbb{R}^n \\ x_{k+1} = f_\omega(x_k) = x_k - \omega g(x_k) \end{cases}$$

Ce qui peut se réécrire sous la forme finale, dite algorithme de relaxation sur f :

$$\begin{cases} \tilde{x}_{k+1} = f(x_k) = x_k - g(x_k) \\ x_{k+1} = \omega \tilde{x}_{k+1} + (1 - \omega) \tilde{x}_k \end{cases}$$

(a) $x_0 = 0.85$ (b) $x_0 = 1.03$ FIGURE 1.6 – Itérations du point fixe pour la fonction $f : x \mapsto x^2$

2.1.3 Vitesse de convergence d'une suite

Soient une suite $(x_k)_{k \in \mathbb{N}} \in \mathbb{R}^n$ et $\bar{x} \in \mathbb{R}^n$. On suppose que la suite converge vers $\bar{x}$ en $k \rightarrow \infty$, et qu'elle est *non stationnaire*, c'est-à-dire que $x_k \neq \bar{x}, \forall k \in \mathbb{N}$. On définit la vitesse de convergence β de la suite $(x_k)_{k \in \mathbb{N}}$:

$$\beta = \lim_{k \rightarrow \infty} \frac{\|x_{k+1} - \bar{x}\|}{\|x_k - \bar{x}\|} \in [0, 1]$$

On dit que :

- la vitesse est *sous-linéaire* si $\beta = 1$
- la vitesse est *linéaire* si $\beta \in]0, 1[$ (au moins linéaire si $\beta \in [0, 1[$)
- la vitesse est *super-linéaire* si $\beta = 0$. Dans ce cas on dit aussi que :
 - la convergence est *quadratique* si :

$$\lim_{k \rightarrow \infty} \frac{\|x_{k+1} - \bar{x}\|}{\|x_k - \bar{x}\|^2} = \gamma > 0$$

- au moins quadratique si $\exists \gamma \in \mathbb{R}^+$ et $n_0 \in \mathbb{N}$ tels que :

$$\forall k \geq n_0, \|x_{k+1} - \bar{x}\| \leq \gamma \|x_k - \bar{x}\|^2$$

De façon plus générale, la convergence est *d'ordre p* si :

$$\lim_{k \rightarrow \infty} \frac{\|x_{k+1} - \bar{x}\|}{\|x_k - \bar{x}\|^p} = \gamma > 0$$

et au moins *d'ordre p* si :

$$\forall k \geq n_0, \|x_{k+1} - \bar{x}\| \leq \gamma \|x_k - \bar{x}\|^p$$

Soit une fonction $f \in \mathcal{C}^1(\mathbb{R}^n, \mathbb{R}^n)$, on suppose qu'il existe $x^* \in \mathbb{R}^n$ tel que $f(x^*) = x^*$. Étudions la vitesse de convergence de la suite définie par :

$$\begin{cases} x_0 \in \mathbb{R}^n \\ \forall k \in \mathbb{N}^*, x_{k+1} = f(x_k) \end{cases} \quad (1.37)$$

- Si on suppose que $f'(x^*) \neq 0$ et $\|f'(x)\| < 1$ sur $\mathbb{R}^n, \|\cdot\|$, alors il existe un voisinage V_{x^*} de x^* tel que si $x_0 \in V_{x^*}$ on a $x_k \rightarrow x^*$ quand $k \rightarrow \infty$. De plus, si la suite est non stationnaire, alors la convergence est linéaire.
- Si on suppose que $f'(x^*) = 0$ et $f \in \mathcal{C}^2(\mathbb{R}^n, \mathbb{R}^n)$, alors il existe un voisinage V_{x^*} de x^* tel que si $x_0 \in V_{x^*}$ on a $x_k \rightarrow x^*$ quand $k \rightarrow \infty$. De plus, si la suite est non stationnaire, la convergence est au moins quadratique (se démontre aisément par développement limité autour de x^*)

Pour $g \in \mathcal{C}^1(\mathbb{R}^n, \mathbb{R}^n)$, en prenant $f(x) = x - g(x)$ ou $f_\omega(x) = x - \omega g(x)$, on a $f'(x) = I - g'(x)$ et $f'_\omega = I - \omega g'(x)$. La suite des itérés de l'algorithme de point fixe avec ou sans relaxation, si elle converge – i.e. si $\|f'\| < 1$ ou $\|f'_\omega\| < 1$, converge donc au moins linéairement.

2.2 Méthodes de Newton

2.2.1 Principe

On peut formuler de manière générale la méthode itérative de résolution du problème non linéaire (1.34) : on va chercher à construire la suite $(x_k)_{k \geq 0}$ du problème (1.37), avec :

$$\begin{cases} f(x) = x - h(x)g(x) \\ h \in \mathcal{C}^2(\mathbb{R}^n, \mathbb{R}^n \times \mathbb{R}^n), \quad h(x) \neq 0 \quad \forall x \in \mathbb{R}^n \end{cases}$$

La méthode de point fixe classique consiste, avec ces notations, à prendre $h(x) = I$. La méthode de point fixe avec relaxation consiste à prendre $h(x) = \omega I$, avec $\omega \in]0, \frac{2\alpha}{M}[$, sous les conditions précisées à la section précédente.

La méthode de Newton, ou encore de Newton-Raphson – dont l'idée ne reviendrait ni à Newton ou à Raphson, mais au mathématicien Thomas Simpson au 18^{ème} siècle [Kollerstrom, 1992] – se base sur l'obtention de la propriété $f'(x^*) = 0$. En effet, d'après l'étude de la vitesse de convergence de la suite (1.37), cette hypothèse permet d'obtenir une convergence asymptotique quadratique, donc très intéressante pour augmenter les vitesses de résolution.

Une façon d'obtenir cette propriété de façon certaine, pour une fonction $g \in \mathcal{C}^2(\mathbb{R}^n, \mathbb{R}^n)$ telle que sa jacobienne g' soit inversible, est d'utiliser la fonction :

$$h = (g')^{-1}$$

On a alors :

$$\begin{cases} f(x) = x - (g'(x))^{-1} g(x) \\ f'(x) = I + g''(x) (g'(x))^{-2} g(x) - (g'(x))^{-1} g'(x) \end{cases}$$

Donc en x^* point fixe de f , i.e. zéro de g :

$$f'(x^*) = I - (g'(x^*))^{-1} g'(x^*) = 0$$

Et la vitesse de convergence est *quadratique*, pour une initialisation x_0 dans un certain voisinage V_{x^*} de x^* .

2.2.2 Algorithme

En réalité, on n'a pas besoin de l'inversibilité de g' sur le domaine entier pour définir la suite $(x_k)_{k \geq 0}$, mais seulement en x^* :

$$\begin{cases} x_0 \in \mathbb{R}^n \\ \forall k \in \mathbb{N}^*, \quad g'(x_k) (x_{k+1} - x_k) = -g(x_k) \end{cases} \quad (1.38)$$

Et on définit f telle que :

$$f g' = x g' - g$$

Résoudre le problème tangent $g'(x_k)(x_{k+1} - x_k) = -g(x_k)$ revient donc à définir $x_{k+1} = f(x_k)$.

On a également, si $g'(x^*) \neq 0$, la propriété d'équivalence : $g(x^*) = 0 \Leftrightarrow f(x^*) = x^*$.

De plus :

$$(fg')' = f'g' + fg'' = g' + xg'' - g' = xg'' \Rightarrow f'g' = (x - f)g''$$

Donc en $x = x^*$, la propriété $f(x^*) = x^*$ implique que $f'(x^*)g'(x^*) = 0$. Puisque g' est inversible en x^* , on a nécessairement $g'(x^*) \neq 0$, donc $f'(x^*) = 0$. f est $\mathcal{C}^2$ puisque g est $\mathcal{C}^2$, d'où la vitesse quadratique de convergence.

La grandeur incrémentale $dx_k \equiv x_{k+1} - x_k$ calculée à chaque itération de l'algorithme de Newton (voir algorithme (6)) est appelée *direction de descente*.

Algorithm 6 Newton-Raphson

Require: $g \in \mathcal{C}^2(\mathbb{R}^n, \mathbb{R}^n)$, $\| \cdot \|$ norm on $\mathbb{R}^n$, $\varepsilon < 0$

hypothesis : $\exists x^* \in \mathbb{R}^n$ such that $g(x^*) = 0$, $g'(x^*)$ invertible

initialization : $x_0 \in \mathbb{R}^n$, $r_0 = g(x_0)$, $k = 0$

while $\|r_k\| > \varepsilon$ **do**

 solve $g'(x_k)dx_k = -g(x_k)$

 actualize $x_{k+1} = x_k + dx_k$

 compute $r_{k+1} = g(x_{k+1})$

2.2.3 Variantes

De nombreuses variantes de l'algorithme de Newton-Raphson existent. En effet, si la vitesse de convergence quadratique permet de converger en un nombre réduit d'itérations, la méthode est en contrepartie assez coûteuse : elle nécessite, à chaque itération, le calcul de la jacobienne de g , ainsi que la résolution d'un système linéaire, potentiellement de grande taille.

Une des techniques possibles consiste à ne calculer la jacobienne que toutes les m itérations, avec m un entier (typiquement $m = 3, 4, \dots$). C'est la méthode du *Faux quasi-Newton*. On a ainsi moins de calculs de jacobienne à effectuer, et dans le cas d'une méthode de factorisation pour la résolution du système linéaire (LU, Cholesky...) on peut réutiliser la factorisation pour résoudre m systèmes. Cependant, on perd la propriété de convergence quadratique : cette méthode est en pratique peu utilisée.

Une variante usuelle consiste à supposer une décomposition possible de la fonction g :

$$g(x) = Ax + F_1(x) + F_2(x), \quad A \in \mathcal{M}_n(\mathbb{R}), \quad F_1, F_2 \in \mathcal{C}^1(\mathbb{R}^n, \mathbb{R}^n)$$

Le système (1.38) devient alors :

$$\begin{cases} x_0 \in \mathbb{R}^n \\ (A + F'_1(x_k) + F'_2(x_k))(x_{k+1} - x_k) = -Ax_k - F_1(x_k) - F_2(x_k) \end{cases}$$

La méthode de *Newton incomplet* consiste à ne pas tenir compte de la jacobienne de F_2 :

$$\begin{cases} x_0 \in \mathbb{R}^n \\ (A + F_1'(x_k))(x_{k+1} - x_k) = -Ax_k - F_1(x_k) - F_2(x_k) \end{cases}$$

On dit alors qu'on applique un "Newton sur F_1 " et un "point fixe sur F_2 ".

Bien que cette méthode entraîne également la perte de la vitesse quadratique de convergence, ses avantages sont multiples :

- on ne calcule pas F_2' , on peut utiliser cette méthode dans le cas où F_2 n'est pas dérivable
- on peut choisir F_1 et F_2 de façon à ce que la structure de $(A + F_1')$ soit plus intéressante que celle de g' : par exemple, si A est creuse (c'est le cas notamment si A provient de la discrétisation d'un Laplacien) on va choisir F_1 et F_2 de telle sorte que $A + F_1$ ait également une structure creuse.
- si la non linéarité de g peut être divisée en un couplage fort et un plus faible, on pourra essayer de répartir le couplage non linéaire plus important dans la fonction F_1 et le reste dans F_2 .

Ces avantages font du *Newton incomplet* une méthode fréquemment utilisée.

Une dernière variante, dite *méthode de quasi-Newton*, permettant encore une fois d'éviter le calcul de la jacobienne g' , consiste à trouver, à chaque itération et à moindre coût, une matrice approchée $B_k \simeq g'(x_k)$. En dimension $n = 1$ on peut simplement utiliser un quotient différentiel :

$$B_k = \frac{g(x_k) - g(x_{k-1})}{x_k - x_{k-1}}$$

et la méthode est dite *de la sécante*.

En dimension $n > 1$, on cherche B_k satisfaisant à la condition :

$$B_k(x_k - x_{k-1}) = g(x_k) - g(x_{k-1}) \quad (1.39)$$

Cette condition n'est pas suffisante pour définir B_k , différentes techniques existent pour cela. La *méthode BFGS* notamment – Broyden-Fletcher-Goldfarb-Shanno [Liu and Nocedal, 1989], consiste à chercher B_k sous la forme :

$$B_k = B_{k-1} + U_k + V_k$$

où U_k et V_k sont des matrices symétriques de rang 1.

On trouve :

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

avec $y_k = g(x_k) - g(x_{k-1})$.

2.2.4 Optimisation du pas de Newton

Un autre inconvénient de la méthode de Newton est que sa convergence n'est assurée que si l'initialisation est « assez proche » de la solution. Afin d'éviter ce problème, une étape d'optimisation peut être ajoutée lors du calcul de la direction de descente dx_k de l'algorithme (6). En effet, si l'initialisation est trop éloignée du zéro de g , il peut être judicieux de ne pas utiliser cette direction telle quelle en tant qu'incrément, mais de chercher le meilleur incrément possible le long de cette direction [Gould and Leyffer, 2003]. On considère alors un incrément optimal $dx_{k_{opt}} = \alpha_{opt} dx_k$, où α_{opt} est déterminé tel que :

$$\alpha_{opt} = \underset{\alpha \in \mathbb{R}}{\operatorname{argmin}} g(x_k + \alpha dx_k)$$

On parle alors de méthode de type *linesearch*. D'autres types de méthodes, dites de *trust regions*, permettent de modifier la direction de descente a posteriori, en approximant f par un modèle en général quadratique autour de chaque itéré [Conn et al., 2000]. Ces méthodes sont meilleures que celle de type *linesearch* en termes de renforcement de la probabilité de convergence, mais sont plus complexes à implémenter.

2.3 Méthodes de Newton inexact

La convergence quadratique de l'algorithme de Newton précédemment détaillé est intéressante, cependant plusieurs écueils peuvent dégrader les temps de calcul lors de son utilisation.

En effet, les étapes coûteuses du solveur de Newton sont :

- le calcul de la matrice jacobienne
- la résolution du problème tangent

On a vu comment on pouvait approximer, si nécessaire, le calcul de la jacobienne à chaque itération. Reste la résolution du problème tangent, dont le coût peut être très pénalisant dans le cas de systèmes à grand nombre de degrés de liberté.

L'utilisation d'un solveur itératif et parallélisable, peut permettre de diminuer dans ce cas les temps de calcul. La résolution tangente est alors effectuée jusqu'à une précision donnée : on entre dans le cadre des méthodes dites de *Newton inexact* (voir algorithme (7)) [Dembo et al., 1982].

2.3.1 Critère d'arrêt tangent

La question de la convergence de la méthode se pose alors : quid de la vitesse quadratique du solveur de Newton dans le cas d'une direction de descente dx_k inexacte ? Deux types de problèmes peuvent apparaître :

- une dégradation de la convergence si le critère d'arrêt du solveur tangent est trop élevé

Algorithm 7 Inexact Newton

Require: $g \in \mathcal{C}^2(\mathbb{R}^n, \mathbb{R}^n)$, $\| \cdot \|$ norm on $\mathbb{R}^n$, $\varepsilon_G > 0$, $\varepsilon_T > 0$

hypothesis : $\exists x^* \in \mathbb{R}^n$ such that $g(x^*) = 0$, $g'(x^*)$ invertible

initialization : $x_0 \in \mathbb{R}^n$, $r_{G_0} = g(x_0)$, $k = 0$

while $\|r_{G_k}\| > \varepsilon_G$ **do**

solve iteratively $g'(x_k)dx_k + g(x_k) = 0$, until $r_{T_k} = g(x_k) + g'(x_k)dx_k$ s.t. $\frac{\|r_{T_k}\|}{\|r_{T_0}\|} \leq \varepsilon_T$

actualize $x_{k+1} = x_k + dx_k$

→ une sur-résolution tangente dans le cas d'un critère d'arrêt trop faible

Le premier problème peut facilement être traité par la mise en place d'un seuil maximum sur le critère d'arrêt relatif du solveur tangent.

Le deuxième nécessite une étude plus poussée : en effet, on peut aisément imaginer que la précision dont on a besoin sur l'incrément linéaire de l'algorithme de Newton va varier au cours des itérations. Par exemple, en début de résolution, on n'a pas besoin d'une solution tangente très précise, car les itérés sont loin de la solution, les directions de recherche sont donc "fausses".

En revanche, lorsqu'on s'approche de la solution, on a besoin d'augmenter la précision de la solution du problème tangent.

Une manière d'adapter le critère d'arrêt tangent à la qualité des itérés est d'utiliser une fonction dépendant de la norme du résidu :

$$\varepsilon_{T_k} = f(\|g(x_k)\|)$$

Les démonstrations détaillées permettant d'établir les propriétés de convergence en fonction du critère d'arrêt relatif ε_T choisi pour le problème tangent – auquel on fait fréquemment référence en tant que *forcing term* – sont disponibles dans [Eisenstat and Walker, 1994, Eisenstat and Walker, 1996].

On retiendra principalement les propriétés suivantes :

- si $\forall k \in \mathbb{N}$, $0 \leq \varepsilon_{T_k} \leq \varepsilon_{T_{max}} < 1$, alors la convergence de la suite $(x_k)_{k \geq 0}$ définie dans l'algorithme (7) est au moins linéaire pour la norme $\|x\|_* \equiv \|g'(x^*)x\|$, avec une vitesse asymptotique $v_a \leq \varepsilon_{T_{max}}$
- si $\lim_{k \rightarrow \infty} \varepsilon_{T_k} = 0$, la vitesse de convergence est superlinéaire
- si $\varepsilon_{T_k} = O(\|g(x_k)\|)$, la convergence est quadratique

et les expressions pour ε_{T_k} :

$$\begin{aligned} \circ \varepsilon_{T_k} &= \text{constante} < 1 \\ \circ \varepsilon_{T_k} &= \frac{1}{2^{k+1}} \\ \circ \begin{cases} \varepsilon_{T_k} = \gamma \left(\frac{\|g(x_k)\|}{\|g(x_{k-1})\|} \right)^\alpha \\ \gamma \in [0, 1], \quad \alpha \in]1, 2], \quad \varepsilon_{T_0} \in [0, 1[\end{cases} \end{aligned} \quad (1.40)$$

$$\begin{aligned} \circ \begin{cases} \varepsilon_{T_k} = \frac{\|g(x_k)\| - \|g(x_{k-1}) + g'(x_k)dx_{k-1}\|}{\|g(x_{k-1})\|} \\ \varepsilon_{T_0} \in [0, 1[\end{cases} \end{aligned} \quad (1.41)$$

On peut montrer que le choix (1.40) conduit à une convergence superlinéaire et deux-pas quadratique, c'est-à-dire que :

$$\lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x^*\|}{\|x_{k-1} - x^*\|^2} \in [0, 1[$$

Le choix (1.41) conduit à une convergence d'ordre α pour $\gamma < 1$. Prendre $\alpha = 2$ permet donc théoriquement de conserver la vitesse de convergence quadratique de la méthode de Newton classique.

L'adaptation du critère d'arrêt du solveur tangent permet donc de conserver des propriétés théoriques de convergence convenables, tout en minimisant les itérations internes, et donc les communications entre les processeurs – en évitant la sur-résolution, – un soin particulier doit donc être apporté à son optimisation.

3 Résolution parallèle des systèmes non linéaires

La première section de ce chapitre faisait état des méthodes de résolution parallèle existant pour résoudre de grands systèmes linéaires.

La deuxième section retrace les grandes lignes des méthodes de résolution des équations non linéaires, en particulier des méthodes de Newton.

L'objectif de ce manuscrit est de développer une nouvelle technique portant sur la résolution parallèle des systèmes non linéaires : un bref état de l'art des méthodes existantes à ce sujet sera donc effectué avant d'introduire cette technique. Dans les années 1990, Lori Badea démontra la convergence de l'algorithme de Schwarz alterné, pour des problèmes non linéaires monotones et plus de deux sous-domaines [Badea, 1991]. Ceci ouvrit la voie à des méthodes de résolution non linéaires parallèles consistant à combiner un algorithme de Newton avec des méthodes DD.

3.1 Méthodes NKS

De façon générale, les méthodes N-K-S sont dédiées à la résolution parallèle de problèmes non linéaires à grand nombre d'inconnues. Les initiales N-K font référence aux méthodes de Newton-Krylov, soit les méthodes de Newton inexactes avec pour solveurs tangents des solveurs de Krylov (GMRES, MINRES, CG...). L'initiale S peut correspondre soit aux méthodes Schwarz, soit à celles de Schur, et concerne le type de préconditionneur utilisé pour le problème tangent. On pourra par exemple combiner le solveur de Newton inexact avec une méthode BDD ou de Schwarz additif avec préconditionneur RAS pour le problème tangent [Rey, 1994, De Roeck et al., 1992, Farhat et al., 2000b, Barbotteu et al., 2001] (voir la figure 1.7).

La performance des méthodes NKS a été démontrée principalement pour des problèmes à faible non linéarité diffuse.

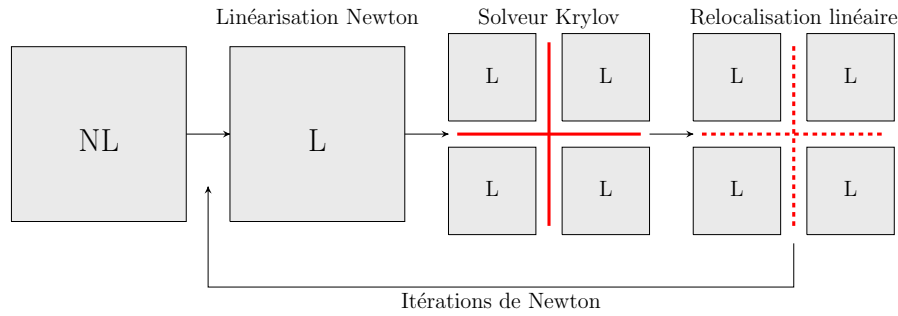


FIGURE 1.7 – Principe de la méthode NKS

3.2 Méthode ASPIN

Cai et Keyes publièrent en 2002 une proposition de préconditionnement non linéaire du système $g(x) = 0$. Les méthodes PIN (Preconditionned Inexact Newton) consistent à remplacer ce problème par un nouveau système non linéaire $h(g(x)) = 0$, ayant la même solution, mais tel que $h \simeq g^{-1}$ et $h(x)$ soit aisément calculable. On veut de plus que l'algorithme de Newton inexact appliqué au nouveau problème produise des opérateurs tangents dont la multiplication par des vecteurs d'interface soit simple et parallélisable.

La méthode proposée dans [Cai and Keyes, 2002, Cai et al., 2002], dite méthode *ASPIN*, consiste à utiliser l'algorithme de Schwarz additif non linéaire en guise de préconditionneur non linéaire : on définit des fonctions non linéaires locales $T^{(s)}$ solution des équilibres locaux, c'est-à-dire telles que $\forall v^{(s)} \in \mathbb{R}^{n^{(s)}} :$

$$g|_{\Omega^{(s)}}(v^{(s)} - T^{(s)}(v^{(s)})) = 0$$

L'opérateur de préconditionnement non linéaire peut alors s'écrire, avec les notations introduites à la section 1.2.4 (chap. 1) :

$$h(x) = \sum_s A^{(s)} T^{(s)}$$

Le calcul de l'opérateur tangent donne exactement l'opérateur de l'algorithme de Scharz additif, ce qui permet d'utiliser un algorithme de Newton inexact avec le solveur tangent sus-mentionné.

Remarque 13 : L'utilisation du préconditionneur RAS (algorithme présenté à la section 1.1.3 (chap. 1) a également fait l'objet d'une étude dans le cadre des résolutions de systèmes non linéaires par des algorithmes de Newton Exact (voir méthode RASPEN [Dolean et al., 2016]).

3.3 Méthode LaTIn

Ce chapitre bibliographique ne saurait être complet sans une dernière référence, faite à la méthode LaTIn (Large Time Increment) [Ladevèze, 1985, Ladevèze and Dureisseix, 1999], alternative aux méthodes NKS pour traiter des problèmes non linéaires d'évolution à grand nombre de degrés de liberté, à rapprocher d'une méthode de Schwarz optimisé non linéaire et d'une méthode de directions alternées [Godlewski, 1980].

Cette méthode est basée sur la séparation des équations en deux types :

- Les équations linéaires, i.e. les équations de liaison, d'équilibre aux interfaces... Elles définissent l'espace vectoriel conventionnellement noté A_d (voir la figure 1.8).
- Les équations locales, éventuellement non linéaires, i.e. les lois d'évolution des matériaux, lois de contact, etc... Elles définissent la variété conventionnellement notée Γ (voir la figure 1.8).

La méthode LaTIn utilise une résolution itérative en deux étapes : à chaque itération, on va résoudre successivement les équations locales et globales, au moyen de directions de recherche E^+ et E^- qui permettent de relier les deux espaces de solutions. Si les fonctions dépendent du temps, la solution est cherchée à chaque fois sur tout le domaine espace $\times$ temps.

La non linéarité est traitée par une approche en contraintes, donc aux points de Gauss. Le parallélisme de la méthode est très bon en mono-échelle, cependant afin d'améliorer son extensibilité, l'ajout d'un problème macroscopique est souvent nécessaire, qu'il concerne le domaine spatial [Cresta, 2008] ou temporel [Ladeveze et al., 2010a, Passieux et al., 2010].

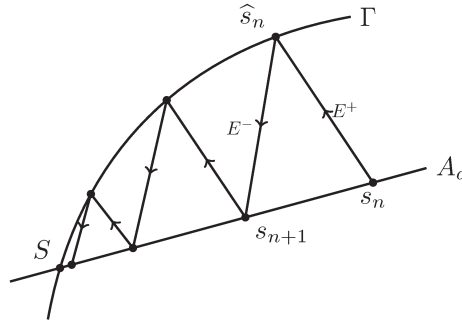


FIGURE 1.8 – Itérations de la méthode LaTIn

Principalement adaptée au calcul non linéaire dans le cadre de non linéarité localisée, elle a été appliquée aux problèmes d'élasto-plasticité [Boisse et al., 1989], de viscoplasticité [Cognard, 1989], ainsi qu'à des problèmes de dynamique [Royer, 1990, Ladevèze, 1996]. Certaines évolutions de la méthode LaTIn ont également été développées pour traiter des problèmes de flambage [Boucard and Ladeveze, 1997] et de grandes transformations [Bussy et al., 1990], et enfin plus récemment dans le cadre de problèmes de contact avec stratégie non intrusive [Oumaziz et al., 2016].

L'un des atouts principaux de cette méthode provient de son couplage aisé avec des méthodes multi-paramétriques comme les méthodes d'approximation radiale [Cognard and Ladevèze, 1993], dont récemment la PGD [Ladeveze et al., 2010b] – l'initialisation pouvant être réalisée par la solution d'un calcul précédent, et les fonctions d'espace pouvant être réutilisées [Boucard and Ladevèze, 1999, Boucard, 2001].

3.4 Relocalisation non linéaire

La suite de ce manuscrit découle des méthodes NKS : on verra qu'une optimisation de la parallélisation de ces méthodes peut être effectuée, et que les communications entre processeurs peuvent être minimisées au cours de la résolution. Par exemple, dans le cas d'une non linéarité forte et localisée, beaucoup d'itérations de Newton sont nécessaires avec un solveur "classique" NKS, tandis que la plus grande partie de la structure évolue de façon linéaire. Dans cette situation, il est intéressant de différencier les comportements des sous-structures (linéaires ou non linéaires), en opérant des résolutions locales non linéaires. Ceci permet d'améliorer la qualité de la solution en sortie des équilibres locaux, et donc de construire un meilleur problème d'interface, réduisant ainsi le nombre d'itérations de Newton nécessaires. On parle alors de *relocalisation non linéaire* : voir la figure 1.9.

Que ce soit dans le cadre des méthodes BDD [Bordeu et al., 2009, Cresta et al., 2007], FETI [Pebrel et al., 2008], FETI-DP [Klawonn et al., 2014] ou BDDC [Hinojosa et al., 2014], la possibilité de définir des analogues non linéaires des compléments de Schur classiques a commencé à être étudiée au cours des dernières années, bien

que leur implémentation à l'échelle industrielle soit encore loin d'être courante. Afin de pallier ce problème, des méthodes non-intrusives ont été proposées. Celles-ci réinterprètent les solveurs de manière à les rendre programmables sur la base des codes fermés utilisés par les industriels. La méthode de couplage local/global [Gendre et al., 2009], permet d'insérer au sein du modèle, relativement grossier et en particulier linéaire, d'une structure, des patches non-linéaires raffinés, potentiellement résolus par des logiciels adaptés. Cette méthode a également été appliquée dans le cas où la structure est entièrement recouverte par les patches conduisant à la mise en place d'un solveur DD par relocalisation non linéaire non intrusif [Duval et al., 2016]. Le modèle grossier linéaire s'interprète classiquement comme un problème grossier (sic) de décomposition de domaine ou la grille grossière (re-sic) d'une méthode multigrille localisée. Dans [Blanchard et al., 2017], la méthode est interprétée comme une technique de Schwarz optimisée alternée (sans recouvrement malgré les apparences) dont une partie des paramètres est particulièrement bien ajustée. Cependant, à moins de fortement déraffiner le problème grossier, le caractère alterné de la méthode peut s'avérer limitant en vue d'applications HPC.

Les prochains chapitres de ce manuscrit permettront tout d'abord de donner un cadre formel aux nouveaux solveurs DD (de Schur) non linéaires avec relocalisation. On verra par la suite comment les optimiser à moindre coût, puis les méthodes décrites seront testées sur des cas simples.

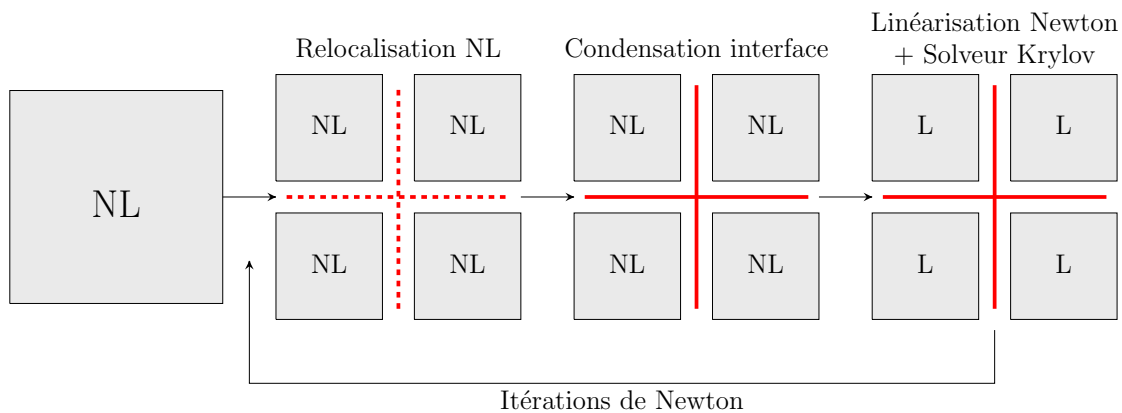


FIGURE 1.9 – Principe de la méthode NKS avec relocalisation non linéaire

Sous-structuration et condensation non linéaires

On se place ici dans le cadre de la résolution parallèle de problèmes non linéaires à grand nombre d'inconnues. L'une des catégories de méthodes de résolution dédiées, les algorithmes NKS, consistent à combiner un algorithme de Newton appliqué au problème global, puis de résoudre le problème tangent par une approche DD. Le problème d'interface ainsi créé est linéaire, de même que les résolutions locales impliquées dans le processus parallèle, quel que soit le comportement du matériau à l'intérieur des sous-domaines. Si l'efficacité de ce type de technique a été démontrée dans le cadre de non linéarité diffuse et d'intensité peu élevée, elle peut être améliorée par la construction d'un problème d'interface non linéaire, à partir d'une sous-structuration effectuée sur le problème global non linéaire, et non plus sur le problème tangent. Les résolutions locales deviennent non linéaires, et, en particulier dans le cas de non linéarité localisée, on permet à l'algorithme de différencier le comportement de chaque sous-domaine. Il itère ainsi plus sur les sous-domaines présentant une non linéarité, afin d'apporter une meilleure information dans la construction du résidu d'interface : on parle souvent de *relocalisation non linéaire*, bien qu'une dénomination plus précise pourrait être *sous-structuration et condensation non linéaires*. On utilisera cette dénomination dans la suite de ce manuscrit.

Ce chapitre s'attachera à présenter de façon détaillée la méthode de sous-structuration et condensation non linéaires, en précisant pour chaque type d'approche – conditions d'interface primales, duales ou mixtes – le processus de résolution. Il a fait l'objet d'un article publié en 2016 dans le journal international IJNME [Negrello et al., 2016].

1 Problème de référence, notations

On considère ici une équation aux dérivées partielles non linéaire – modélisant un problème de mécanique des structures quasi-statique – sur un domaine $\Omega \subset \mathbb{R}^n$ ($n=2$ ou 3), muni de conditions de Dirichlet u_g sur une partie non vide de son bord

$\partial_u \Omega \neq \emptyset$, de conditions de Neumann F sur le complémentaire $\partial_F \Omega = \partial \Omega \setminus \partial_u \Omega$ et d'efforts volumiques g (voir figure 2.1). Tous les efforts imposés sont supposés indépendants de u_c (charges mortes).

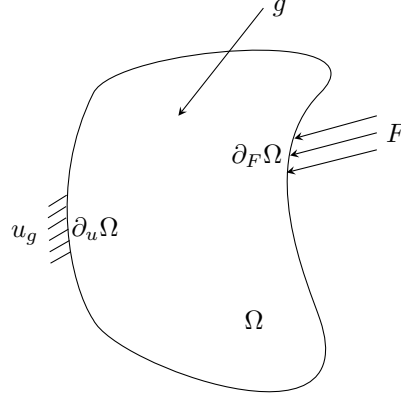


FIGURE 2.1 – Problème de référence

Une formulation continue du problème à résoudre est donnée par :

Au temps $t_c \in [0, T]$, trouver $u_c \in \mathcal{U}$ tel que $\forall v_c \in \mathcal{U}_0$:

$$\int_{\Omega} \sigma_c : \varepsilon_c(v_c) d\Omega = \int_{\Omega} v_c \cdot f_c d\Omega + \int_{\partial_F \Omega} v_c \cdot F_c dS$$

$$\sigma_c = \sigma_c(\varepsilon_c(u_c(\tau_c)), \tau_c \in [0, t_c])$$

où $\mathcal{U}$ est l'espace des champs cinématiquement admissibles, et $\mathcal{U}_0$ l'espace vectoriel associé :

$$\mathcal{U} = \left\{ u_c \in [H^1(\Omega)]^d, u_c = u_g \text{ sur } \partial_u \Omega \right\}$$

avec d la dimension du problème ($d = 1, 2$ ou 3).

Remarque 1 : On se place ici dans le cadre de la mécanique des structures, cependant on pourrait traiter des problèmes de thermique non linéaire par la même méthode ; on en verra notamment un exemple au chapitre suivant.

La notation $\sigma_c(\varepsilon_c(u_c(\tau_c)), \tau_c \in [0, t_c])$ exprime la dépendance de la contrainte en un point à son histoire totale (sur $[0, t_c]$) en ce point, ce qui signifie que son comportement local est non linéaire. L'histoire du matériau est en général représentée par des variables internes : déformation anélastique, écrouissage, endommagement.

Le problème est discrétisé en espace par la méthode des éléments finis ; on note V_h le sous-espace de $\mathcal{U}$ associé au maillage Ω_h – de dimension finie. On appelle N la matrice des fonctions de forme et u le vecteur inconnue contenant les déplacements

nodaux : on peut alors exprimer u_h , la projection de u_c sur V_h , en fonction des données et inconnues discrètes du problème : $u_h = Nu$. On obtient également les vecteurs d'efforts nodaux imposés f et F .

Une discrétisation temporelle vient achever le passage du problème continu au problème algébrique : au pas de temps t_n , le problème discret de référence à résoudre devient :

$$\text{Trouver } u(t_n) \text{ tel que } f_{int}(u) + f_{ext} = 0 \quad (2.1)$$

avec :

$$\begin{aligned} f_{ext} &= \int_{\Omega} N^T g d\Omega + \int_{\partial_F \Omega} N^T F dS \\ v_h^T f_{int} &= - \int_{\Omega} \sigma_h : \varepsilon_h(Nv) d\Omega \\ \sigma_h &= \sigma_h(\varepsilon_h(Nu(t_j))), \quad j \leq n \end{aligned}$$

La discrétisation de l'opérateur σ_c est ainsi notée σ_h , elle dépend de l'histoire discrète (sur les $(t_j)_{j \leq n}$). Les intégrales sont classiquement calculées numériquement par quadrature de Gauss, de sorte que la contrainte et les variables internes sont définies aux points de Gauss. Par souci de clarté, la variable temporelle (pas de temps t_n) ne sera plus mentionnée dans la suite de ce manuscrit. De plus, les conditions aux limites de Dirichlet u_g sont prises en compte de façon implicite dans f_{int} et f_{ext} .

On peut remarquer que dans le cas de l'élasticité linéaire, pour un domaine de matrice de rigidité K sous l'hypothèse des petites perturbations, on a :

$$f_{int}(u) = -Ku$$

Les notations de la section 1.2.4 (chap. 1) seront adoptées par la suite.

On se place également, dans toute la suite de ce manuscrit, dans l'optique de résolution de problèmes de grande taille, comportant en moyenne 10^6 à 10^9 inconnues. Le nombre d'inconnues du problème global sera noté n_G .

Remarque 2 : Bien que, pour des raisons techniques, la taille des exemples qui seront abordés à la section 5 (chap. 2) soit plus réduite que celle visée dans le cadre des résolutions parallèles, on verra qu'un choix de paramètres d'étude pertinent semble permettre a priori une prédiction correcte du comportement à une plus grande échelle.

2 Méthode classique

On pourrait, pour résoudre le problème (2.1), se servir d'une méthode NKS (voir section 3.1, chapitre 1), i.e. appliquer un algorithme de Newton pour linéariser le problème, puis utiliser un solveur DD pour résoudre le problème tangent en résultant, préconditionné par une méthode de Schur ou de Schwarz.

De manière générale, on peut résumer cette méthode, à laquelle on se référera dans la suite de ce manuscrit en termes de méthode classique, par l'algorithme 8.

Algorithm 8 Classic method

Require: $\varepsilon_G > 0, \varepsilon_T > 0$

initialization : unknown $u_0 \in \mathbb{R}^{n_G}$, global residual $r_{G_0} = f_{int}(u_0) + f_{ext}$ and tangent operator $f'_{int_0} = f'_{int}(u_0)$, counter $k = 0$

while $\|r_{G_k}\| > \varepsilon_G$ **do**

 initialize increment $du_{k_0} = 0, i = 0$

 compute $r_{T_{k_0}} = f'_{int_k} du_{k_0} + f_{int}(u_k) + f_{ext}$

while $\|r_{T_{k_i}}\| > \varepsilon_T$ **do**

 preconditioned DD solver iterations on tangent problem :

$$f'_{int_k} du_{k_i} + f_{int}(u_k) + f_{ext} = 0$$

 with counter i

 actualize iterate $u_{k+1} = u_k + du_k$

$k = k + 1$

 compute global residual $r_{G_k} = f_{int}(u_k) + f_{ext}$ and tangent operator f'_{int_k}

Remarque 3 : On note $f'_{int_k} = \frac{df_{int}}{du}(u_k)$ l'opérateur tangent (matriciel) au point u_k . Par exemple, pour un problème de plasticité en mécanique des structures, on a :

$$f'_{int_k} = -K_{t_k}$$

avec K_{t_k} la matrice de rigidité tangente de la structure en un état décrit par le déplacement u_k .

Le solveur DD utilisé à chaque itération globale k peut par exemple être l'un de ceux présentés à la section 1.2.8 (chap. 1), i.e. BDD/FETI. La sous-structuration est bien appliquée au problème tangent linéaire :

$$f'^{\diamond}_{int_k} du^{\diamond}_k + f^{\diamond}_{int}(u^{\diamond}_k) + f^{\diamond}_{ext} + t^{\diamond T} d\lambda^{\diamond}_{b_k} = 0^{\diamond} \quad (2.2)$$

où l'on a introduit l'inconnue d'interface locale $d\lambda^{(s)}_{b_k}$, soit l'incrément de réaction nodale imposé sur une sous-structure $\Omega^{(s)}$ par ses voisins, et permettant de rendre les équations locales indépendantes.

En notant $S^{(s)}_{t_k}$ la condensation primale de Schur de l'opposé de l'opérateur local

tangent $f'_{int_k(s)}$ sur les nœuds de bord d'une sous-structure $\Omega^{(s)}$:

$$S_{t_k}^{(s)} = - \left[\left(f'_{int_k(s)} \right)_{bb} - \left(f'_{int_k(s)} \right)_{bi} \left(f'_{int_k(s)} \right)_{ii}^{-1} \left(f'_{int_k(s)} \right)_{ib} \right]$$

et $b_{p_k}^{(s)}$ le second membre primal local condensé :

$$b_{p_k}^{(s)} = \left(f_{int}^{(s)}(u_k^{(s)}) + f_{ext}^{(s)} \right)_b - \left(f'_{int_k(s)} \right)_{bi} \left(f'_{int_k(s)} \right)_{ii}^{-1} \left(f_{int}^{(s)}(u_k^{(s)}) + f_{ext}^{(s)} \right)_i$$

on peut exprimer l'inconnue d'interface $d\lambda_{b_k}^\diamond$ en fonction de l'incrément de déplacement d'interface local $du_{b_k}^\diamond$:

$$d\lambda_{b_k}^\diamond = S_{t_k}^\diamond du_{b_k}^\diamond - b_{p_k}^\diamond \quad (2.3)$$

Remarque 4 : Par exemple, pour un problème de plasticité en mécanique des structures, on a sur une sous-structure $\Omega^{(s)}$:

$$S_{t_k}^{(s)} = K_{t_k}^{(s)} / K_{ii}$$

Autrement dit, $S_{t_k}^{(s)}$ condense l'opérateur de rigidité tangent à l'itération k – noté ici $K_{t_k}^{(s)}$ – sur les degrés de liberté de bord de la sous-structure.

En notant du_{A_k} l'inconnue globale d'interface primale – satisfaisant la continuité des incréments de déplacement d'interface – telle que $du_{b_k}^\diamond = A^\diamond du_{A_k}$, – la condition d'équilibre des incréments de réaction nodale aux interfaces permet de construire le problème tangent d'interface :

$$\begin{aligned} A^\diamond d\lambda_{b_k}^\diamond &= 0^\diamond \\ \Rightarrow A^\diamond S_{t_k}^\diamond A^{\diamond T} du_{A_k} &= A^\diamond b_{p_k}^\diamond \end{aligned} \quad (2.4)$$

Avec une approche primale, le problème d'interface résolu par le solveur DD serait donc le problème (2.4), grâce aux équilibres locaux (2.2). Pour obtenir la solution globale à convergence du solveur DD, il suffit d'effectuer des résolutions locales par sous-domaine pour déterminer les termes $du_{i_k}^{(s)}$.

De façon similaire, on peut définir la méthode classique avec approche duale, en prenant pour inconnue d'interface $d\lambda_{B_k}$ – satisfaisant la condition d'équilibre des incréments de réaction nodale d'interface : $d\lambda_{B_k}^\diamond = B^{\diamond T} d\lambda_{B_k}$.

La méthode qui sera développée ici, dite de sous-structuration et condensation non linéaires, et connue sous le nom de *relocalisation non linéaire* – abordée à la section 3.4

(chap. 1) – consiste à sous-structurer le problème non linéaire global, afin d'obtenir des équilibres locaux non linéaires, permettant de différencier les sous-domaines ayant un comportement non linéaire des autres, et de construire par les équilibres locaux non linéaires un problème d'interface plus proche de la réalité.

3 Construction d'un problème non-linéaire condensé à l'interface

Similairement au processus de sous-structuration linéaire, on reformule le problème (2.1) en introduisant la réaction nodale d'interface locale $\lambda_b^{(s)}$:

$$\begin{aligned} &\text{Trouver } (u^\diamond, \lambda_b^\diamond) \text{ tel que :} \\ &f_{int}^\diamond(u^\diamond) + f_{ext}^\diamond + t^{\diamond T} \lambda_b^\diamond = 0^\diamond \\ &A^\diamond \lambda_b^\diamond = 0 \\ &B^\diamond u_b^\diamond = 0 \end{aligned} \tag{2.5}$$

La formulation sous-structurée (2.5) est strictement équivalente au problème global (2.1). La stratégie de résolution varie alors en fonction de l'approche retenue – primale, duale ou mixte. Ces trois formulations seront détaillées dans les paragraphes suivants.

3.1 Trois formulations distinctes

3.1.1 Formulation primale

L'approche primale consiste à reformuler le système (2.5) en fonction d'une unique inconnue d'interface u_A – satisfaisant la condition de continuité des déplacements :

$$\begin{aligned} &\text{Trouver } u_A \in \mathbb{R}^{n_A} \text{ tel que } A^\diamond \lambda_b^\diamond = 0 \\ &\text{avec : } \lambda_b^\diamond \equiv -[f_{int}^\diamond(u^\diamond) + f_{ext}^\diamond]_b \\ &\text{et } u^\diamond \text{ solution de : } \begin{cases} [f_{int}^\diamond(u^\diamond) + f_{ext}^\diamond]_i = 0 \\ t^\diamond u^\diamond = A^{\diamond T} u_A \end{cases} \end{aligned} \tag{2.6}$$

Le jeu de deux équations (2.6) définit des problèmes mécaniques indépendants sur chaque sous-domaine, avec des conditions aux limites en déplacement (ou de Dirichlet), qui assurent automatiquement la continuité du déplacement d'interface u_A : $u_b^\diamond = t^\diamond u^\diamond = A^{\diamond T} u_A \Rightarrow B^\diamond u_b^\diamond = 0$. Le déplacement u_A doit alors être déterminé de sorte que les réactions nodales associées $\lambda_b^\diamond$ soient équilibrées aux interfaces.

En faisant l'hypothèse que le jeu d'équations (2.6) possède une unique solution pour tout déplacement imposé $A^{\diamond T} u_A$, on peut définir un opérateur noté $S_{nl}^{(s)}$ tel que :

$$\forall s \in \{1, \dots, N_s\}, \quad \lambda_b^{(s)} = S_{nl}^{(s)}(u_b^{(s)} = A^{(s)T} u_A; f_{ext}^{(s)})$$

On peut interpréter cet opérateur comme une version non linéaire du complément de Schur primal : il permet de déterminer la réaction nodale locale associée à un déplacement d'interface donné. On parle alors d'opérateur Dirichlet-vers-Neumann.

Remarque 5 : Dans le cas d'un problème de mécanique linéaire, pour un sous-domaine de matrice de rigidité $K^{(s)}$, cet opérateur peut aisément être explicité :

$$S_l^{(s)}(u_b^{(s)}; f_{ext}^{(s)}) = S^{(s)} u_b^{(s)} - b_p^{(s)}$$

$$\text{avec : } S^{(s)} = K_{bb}^{(s)} - K_{bi}^{(s)} K_{ii}^{(s)-1} K_{ib}^{(s)} \quad (2.7)$$

$$b_p^{(s)} = f_{ext_b}^{(s)} - K_{bi}^{(s)} K_{ii}^{(s)-1} f_{ext_i}^{(s)} \quad (2.8)$$

Pour l'approche primale, on obtient alors le système non linéaire condensé à l'interface (2.9) :

Trouver $u_A \in \mathbb{R}^{n_A}$ tel que :

$$A^{\diamond} S_{nl}^{\diamond} (A^{\diamond T} u_A; f_{ext}^{\diamond}) = 0$$

(2.9)

3.1.2 Formulation duale

L'approche duale consiste à reformuler le système (2.5) en fonction d'une unique inconnue d'interface λ_B – satisfaisant la condition d'équilibre des réactions nodales aux interfaces :

$$\begin{aligned} &\text{Trouver } \lambda_B \in \mathbb{R}^{n_B} \text{ tel que } B^{\diamond T} t^{\diamond} u^{\diamond} = 0 \\ &\text{où } u^{\diamond} \text{ est solution de } f_{int}^{\diamond}(u^{\diamond}) + f_{ext}^{\diamond} + t^{\diamond T} B^{\diamond T} \lambda_B = 0 \end{aligned} \quad (2.10)$$

L'équation (2.10) définit des problèmes mécaniques indépendants sur chaque sous-domaine, avec des conditions aux limites en effort (ou de Neumann), qui doivent être cherchées de telle sorte que les déplacements soient continus à l'interface. Les réactions nodales d'interface locales ont été implicitement définies par $\lambda_b^{(s)} = B^{(s)T} \lambda_B$, ce qui garantit automatiquement l'équilibre aux interfaces : $A^{\diamond} \lambda_b^{\diamond} = 0$.

Afin que le problème (2.10) soit bien posé, la condition nécessaire d'auto-équilibre de chaque sous-structure doit être vérifiée. Sous l'hypothèse des petites perturbations, on note $R^{(s)}$ la base des modes rigides cinématiquement admissibles d'une sous-structure $\Omega^{(s)}$ – si ces modes rigides existent, ils peuvent être représentés exactement dans l'espace des fonctions de forme de la méthode éléments finis, – on a alors :

$R^{(s)T} f_{int}^{(s)} = 0$, i.e. les modes rigides ne font pas travailler les efforts internes de la sous-structure. Ceci implique la condition d'admissibilité :

$$R^{\diamond T} \left(f_{ext}^{\diamond} + t^{\diamond T} B^{\diamond T} \lambda_B \right) = 0 \quad (2.11)$$

En supposant que pour toute condition aux limites en traction $B^{\diamond T} \lambda_B$, le problème (2.10) possède une solution unique à un mode rigide près $R^{\diamond} \alpha^{\diamond}$, où $\alpha^{\diamond}$ est l'amplitude (inconnue a priori) des modes rigides, on peut définir un opérateur $F_{nl}^{(s)}$ tel que :

$$\forall s \in \{1, \dots, N_s\}, \quad u_b^{(s)} = F_{nl}^{(s)} \left(B^{(s)T} \lambda_B; f_{ext}^{(s)} \right) + t^{(s)} R^{(s)} \alpha^{(s)}$$

Cet opérateur peut être interprété comme une version non linéaire du complément de Schur dual – employé dans le cadre des méthodes FETI, voir section 1.2 (chap. 1) : il permet de déterminer le déplacement associé à une réaction nodale d'interface locale donnée.

Remarque 6 : Dans le cas d'un problème de mécanique linéaire, pour un sous-domaine de matrice de rigidité $K^{(s)}$, cet opérateur peut aisément être explicité :

$$F_l^{(s)} \left(\lambda_b^{(s)}; f_{ext}^{(s)} \right) = F^{(s)} \lambda_b^{(s)} - b_d^{(s)} \quad (2.12)$$

$$avec : F^{(s)} = t^{(s)} K^{(s)} \dot{t}^{(s)T} \quad (2.13)$$

$$b_d^{(s)} = -t^{(s)} K^{(s)} \dot{f}_{ext}^{(s)} \quad (2.13)$$

On rappelle ici les relations classiques :

$$R^{(s)} = \text{Ker} \left(K_t^{(s)} \right)$$

$$R_b^{(s)} = t^{(s)} R^{(s)} = \text{Ker} \left(S_t^{(s)} \right)$$

Le problème non linéaire condensé dual s'écrit alors :

Trouver $\lambda_B \in \mathbb{R}^{n_B}$, $\alpha^{\diamond}$ tels que :

$$\begin{cases} B^{\diamond} \left(F_{nl}^{\diamond} \left(B^{\diamond T} \lambda_B; f_{ext}^{\diamond} \right) + t^{\diamond} R^{\diamond} \alpha^{\diamond} \right) = 0 \\ R^{\diamond T} \left(f_{ext}^{\diamond} + t^{\diamond T} B^{\diamond T} \lambda_B \right) = 0 \end{cases} \quad (2.14)$$

3.1.3 Formulation mixte

Pour la formulation mixte, on introduit une nouvelle variable d'interface $\mu_b^{(s)}$, définie sur chaque sous-domaine :

$$\mu_b^{\diamond} = \lambda_b^{\diamond} + Q_b^{\diamond} u_b^{\diamond}$$

La matrice $Q_b^\diamond$ est un paramètre de la méthode. Elle peut être, de même que dans le cas linéaire (voir section 1.2.9, chapitre 1), interprétée comme une rigidité – ou impédance – ajoutée à l'interface locale d'un sous-domaine : on parle également de conditions de Robin. Afin d'assurer l'existence d'une solution au problème global condensé, on impose les conditions de symétrie, définition et positivité à ce paramètre (voir section 1.2.9, chapitre 1, on assure ainsi notamment l'inversibilité de la matrice $A^\diamond Q_b^\diamond A^{\diamond T}$).

Les propriétés de la matrice $Q_b^\diamond$ impliquent un partitionnement de l'espace des degrés de liberté de bord locaux, noté $\mathbb{R}_b^\diamond$:

$$\mathbb{R}_b^\diamond = \text{Im} \left(Q_b^\diamond A^{\diamond T} \right) \oplus \text{Im} \left(B^{\diamond T} \right) \quad (2.15)$$

$$= \text{Ker} \left(A^\diamond Q_b^\diamond \right) \oplus \text{Ker} \left(B^\diamond \right) \quad (2.16)$$

Remarque 7 : La propriété (2.15) de séparation sur les images peut être montrée à partir de la séparation très similaire (1.19) introduite à la section 1.2.4 (chap. 1), ainsi que de l'inversibilité de la matrice $(A^\diamond Q_b^\diamond A^{\diamond T})$. En effet, on a :

$$\text{Ker} \left(A^\diamond Q_b^\diamond A^{\diamond T} \right) = \{0\} \Rightarrow \text{Ker} \left(A^\diamond \right) \cap \text{Im} \left(Q_b^\diamond A^{\diamond T} \right) = \{0\}$$

Par définition de $B^\diamond$, on a $\text{Im} \left(B^{\diamond T} \right) = \text{Ker} \left(A^\diamond \right)$. Les espaces $\text{Im} \left(B^{\diamond T} \right)$ et $\text{Im} \left(Q_b^\diamond A^{\diamond T} \right)$ sont donc en somme directe. De plus, $Q_b^\diamond$ est SPD donc l'égalité $\text{Rang} \left(Q_b^\diamond A^{\diamond T} \right) = \text{Rang} \left(A^{\diamond T} \right)$ est vérifiée, et par la propriété (1.19) on a :

$$\text{Rang} \left(A^{\diamond T} \right) + \text{Rang} \left(B^{\diamond T} \right) = \text{Rang} \left(\mathbb{R}_b^\diamond \right)$$

d'où le résultat.

De même, on a $\text{Ker} \left(A^\diamond Q_b^\diamond \right) \cap \text{Im} \left(A^{\diamond T} \right) = \{0\}$. Par définition de $B^\diamond$, on a $A^\diamond B^{\diamond T} = 0 = B^\diamond A^{\diamond T}$ donc $\text{Im} \left(A^{\diamond T} \right) = \text{Ker} \left(B^\diamond \right)$: les espaces $\text{Ker} \left(A^\diamond Q_b^\diamond \right)$ et $\text{Ker} \left(B^\diamond \right)$ sont en somme directe. En imposant $Q_b^\diamond$ SPD, on assure l'égalité dimensionnelle $\text{Dim} \left(\text{Ker} \left(A^\diamond Q_b^\diamond \right) \right) = \text{Dim} \left(\text{Ker} \left(A^\diamond \right) \right)$. La propriété de séparation sur les noyaux (1.18) permet de conclure quant à la propriété similaire (2.16).

La propriété (2.16) permet de reformuler en une unique équation les conditions d'interface en déplacement et en effort (2.17) :

$$\begin{cases} A^\diamond \lambda_b^\diamond = 0 \\ B^\diamond u_b^\diamond = 0 \end{cases} \quad (2.17)$$

$$\Leftrightarrow A^{\diamond T} \left(A^\diamond Q_b^\diamond A^{\diamond T} \right)^{-1} A^\diamond \mu_b^\diamond - u_b^\diamond = 0^\diamond \quad (2.18)$$

En effet, en multipliant (2.18) à gauche par $B^\diamond$ ou par $A^\diamond Q_b^\diamond$, on retrouve les conditions d'interface classiques :

$$\begin{cases} B^\diamond \left[A^{\diamond T} \left(A^\diamond Q_b^\diamond A^{\diamond T} \right)^{-1} A^\diamond \mu_b^\diamond - u_b^\diamond \right] = B^\diamond u_b^\diamond = 0 \\ A^\diamond Q_b^\diamond \left[A^{\diamond T} \left(A^\diamond Q_b^\diamond A^{\diamond T} \right)^{-1} A^\diamond \mu_b^\diamond - u_b^\diamond \right] = A^\diamond \mu_b^\diamond - A^\diamond Q_b^\diamond u_b = A^\diamond \lambda_b^\diamond = 0 \end{cases} \quad (2.19)$$

Donc (2.18) $\Rightarrow$ (2.17). D'autre part, la propriété (2.16) donne la décomposition suivante :

$$\forall v_b^\diamond \in \mathbb{R}_b^\diamond, \exists! (u_b^\diamond, w_b^\diamond) \in \text{Ker}(B^\diamond) \times \text{Ker}(A^\diamond Q_b^\diamond), \text{ tel que : } v_b^\diamond = u_b^\diamond + w_b^\diamond \quad (2.20)$$

Donc :

$$\begin{aligned} & \forall v_b^\diamond \in \mathbb{R}_b^\diamond, \\ & \begin{cases} B^\diamond v_b^\diamond = 0 \Leftrightarrow w_b^\diamond = 0^\diamond \\ A^\diamond Q_b^\diamond v_b^\diamond = 0 \Leftrightarrow u_b^\diamond = 0^\diamond \end{cases} \Leftrightarrow v_b^\diamond = 0^\diamond \end{aligned} \quad (2.21)$$

Et on trouve (2.17) $\Leftrightarrow$ (2.19) $\Rightarrow$ (2.18).

La formulation mixte du problème (2.5) peut alors s'écrire :

$$\begin{aligned} & \text{Trouver } \mu_b^\diamond \in \mathbb{R}_b^\diamond \text{ tel que : } A^{\diamond T} \left(A^\diamond Q_b^\diamond A^{\diamond T} \right)^{-1} A^\diamond \mu_b^\diamond - u_b^\diamond = 0 \\ & \text{où } u^\diamond \text{ est solution de : } f_{int}^\diamond(u^\diamond) - t^{\diamond T} Q_b^\diamond t^\diamond u^\diamond + t^{\diamond T} \mu_b^\diamond + f_{ext}^\diamond = 0 \end{aligned} \quad (2.22)$$

La dernière équation de (2.22) définit la solution de problèmes non linéaires indépendants sur chaque sous-domaine, sous conditions aux limites de Robin. En faisant l'hypothèse de l'existence d'une unique solution pour tout $\mu_b^\diamond$ donné (l'ajout de la matrice $Q_b^\diamond$ évite l'apparition de modes rigides lors de la sous-structuration), on peut définir un opérateur d'interface non linéaire H_{nl} :

$$\forall s \in \{1, \dots, N_s\}, \quad u_b^{(s)} = H_{nl}^{(s)} \left(\mu_b^{(s)}; f_{ext}^{(s)}, Q_b^{(s)} \right)$$

Remarque 8 : Dans le cas d'un problème de mécanique linéaire, pour un sous-domaine de matrice de rigidité $K^{(s)}$, cet opérateur peut aisément être explicité :

$$\begin{aligned} & H_l^{(s)} \left(\mu_b^{(s)}; f_{ext}^{(s)}, Q_b^{(s)} \right) = H^{(s)} \mu_b^{(s)} + b_m^{(s)} \\ & \text{avec : } H^{(s)} = t^{(s)} \left(K^{(s)} + t^{(s)T} Q_b^{(s)} t^{(s)} \right)^{-1} t^{(s)T} \\ & b_m^{(s)} = t^{(s)} \left(K^{(s)} + t^{(s)T} Q_b^{(s)} t^{(s)} \right)^{-1} f_{ext}^{(s)} \end{aligned} \quad (2.23)$$

On retrouve les opérateurs de cette formulation dans l'approche FETI-2LM

[Roux, 2009].

Le problème mixte non linéaire condensé peut alors être défini par :

$$\boxed{\begin{array}{l} \text{Trouver } \mu_b^\diamond \in \mathbb{R}_b^\diamond \text{ tel que :} \\ A^{\diamond T} \left(A^\diamond Q_b^\diamond A^{\diamond T} \right)^{-1} A^\diamond \mu_b^\diamond - H_{nl}^\diamond(\mu_b^\diamond; f_{ext}^\diamond, Q_b^\diamond) = 0^\diamond \end{array}} \quad (2.24)$$

L'opérateur $(A^\diamond Q_b^\diamond A^{\diamond T})$ possède exactement la structure d'une rigidité condensée et assemblée. À l'instar de l'impédance d'interface linéaire (voir section 1.2.9, chapitre 1), la factorisation de ce type de matrice est numériquement très coûteuse, dans le cas général. Cependant, en choisissant une structure spécifique pour les matrices locales $Q_b^{(s)}$, on peut obtenir un opérateur assemblé $(A^\diamond Q_b^\diamond A^{\diamond T})$ bloc-diagonal, et ainsi rendre son traitement plus abordable dans le cadre de résolutions parallèles.

Formulation équivalente Une formulation équivalente est possible, où l'inconnue d'interface $\mu_b^\diamond$ est remplacée par un couple (F_B, v_A) , où F_B est une réaction nodale d'interface auto-équilibrée, et v_A un déplacement continu aux interfaces :

$$\mu_b^\diamond = B^{\diamond T} F_B + Q_b^\diamond A^{\diamond T} v_A \quad (2.25)$$

La bijection entre $\mu_b^\diamond$ et (F_B, v_A) est assurée par la propriété (2.15) de partition de l'espace $\mathbb{R}_b^\diamond$. En effet, pour tout vecteur d'interface $x_b^\diamond \in \mathbb{R}_b^\diamond$, on obtient la décomposition suivante :

$$\begin{aligned} \forall x_b^\diamond \in \mathbb{R}_b^\diamond, \exists! (x_A, x_B) \in \mathbb{R}^{n_A} \times \mathbb{R}^{n_B}, \text{ tel que } x_b^\diamond &= Q_b^\diamond A^{\diamond T} x_A + B^{\diamond T} x_B \\ \text{on a alors : } \begin{cases} \left(A^\diamond Q_b^\diamond A^{\diamond T} \right)^{-1} A^\diamond x_b^\diamond = x_A \\ \left(B^\diamond Q_b^{\diamond -1} B^{\diamond T} \right)^\dagger B^\diamond Q_b^{\diamond -1} x_b^\diamond = x_B \end{cases} \end{aligned} \quad (2.26)$$

L'utilisation d'une pseudo-inverse pour la matrice $(B^\diamond Q_b^{\diamond -1} B^{\diamond T})^\dagger$ est obligatoire : en effet, la description des connectivités entre les sous-domaines peut être sujette à des redondances en présence de points multiples dans le maillage. Cependant, étant appliquée à un vecteur $B^\diamond Q_b^{\diamond -1} x_b^\diamond \in \text{Im}(B^\diamond Q_b^{\diamond -1}) = \text{Im}(B^\diamond Q_b^{\diamond -1} B^{\diamond T})$ (car $Q_b^{\diamond -1}$ SPD), le vecteur x_B ne dépend pas du choix de la pseudo-inverse : toute pseudo-inverse $M^\dagger$ d'une matrice M doit satisfaire :

$$\forall x \in \text{Im}(M), \quad x = M M^\dagger x \quad (2.27)$$

$$\forall x \in \text{Im}(M^\dagger), \quad x = M^\dagger M x \quad (2.28)$$

Remarque 9 : En supposant par l'absurde que pour tout élément $x_b^\diamond \in \mathbb{R}_b^\diamond$,
 $\exists x_{B1} \neq x_{B2} \in \mathbb{R}^{n_B}$ tels que :

$$\begin{aligned} x_{B1} &= \left(B^\diamond Q_b^{\diamond^{-1}} B^{\diamond^T} \right)^\dagger B^\diamond Q_b^{\diamond^{-1}} x_b^\diamond \\ x_{B2} &= \left(B^\diamond Q_b^{\diamond^{-1}} B^{\diamond^T} \right)^\sharp B^\diamond Q_b^{\diamond^{-1}} x_b^\diamond \end{aligned}$$

où † et $^\sharp$ sont deux types de pseudo-inverses donnés, on a alors par (2.27) :

$$\begin{aligned} B^\diamond Q_b^{\diamond^{-1}} B^{\diamond^T} x_{B1} &= B^\diamond Q_b^{\diamond^{-1}} x_b^\diamond \\ \Rightarrow x_{B2} &= \left(B^\diamond Q_b^{\diamond^{-1}} B^{\diamond^T} \right)^\sharp B^\diamond Q_b^{\diamond^{-1}} B^{\diamond^T} x_{B1} \end{aligned}$$

Donc d'après (2.28) on a $x_{B1} = x_{B2}$.

3.2 À propos de l'existence des analogues non linéaires aux compléments de Schur

L'hypothèse d'existence des compléments de Schur non linéaires primal, dual et mixte est implicitement requise dans les sections précédentes. Dans le contexte de problèmes mécaniques bien posés, l'existence de ces opérateurs est très probable – au moins localement : le lecteur pourra se référer à [Ciarlet, 2013] pour une liste des contextes non linéaires où les solutions des problèmes définis plus haut existent.

Cependant, la formulation, le matériau, ainsi que la forme des sous-structures limitent fortement les domaines d'existence et d'unicité des solutions aux problèmes de Dirichlet, Neumann ou de Robin. Typiquement, en grands déplacements, l'usage de chargements en compression importants devrait être proscrit, étant donné le risque d'instabilités (flambement par exemple); à l'inverse dans le cas de l'étude d'un comportement endommageable, les chargements en traction sont à limiter. La figure 2.2 donne un aperçu de ces deux risques potentiels.

L'existence globale des opérateurs S_{nl} , F_{nl} et H_{nl} peut être prouvée dans le cas d'opérateurs maximaux monotones coercifs et continus – voir [Erdos, 1978, Showalter, 2013] par exemple pour une analyse au niveau de la formulation variationnelle, et [Feistauer and Ženíšek, 1986] pour une analyse de l'approximation éléments finis. Mécaniquement parlant, ce contexte se retrouve dans le cas de comportements plastiques à écrouissage positif, et certaines lois de contact – toujours sous contrainte de petites déformations [Ladevèze, 1985, Ladevèze, 2012]. Les problèmes de Dirichlet, Neumann ou de Robin sont alors bien posés : la solution existe, est unique et varie continûment avec le chargement. Le théorème de continuité de la trace permet de dé-

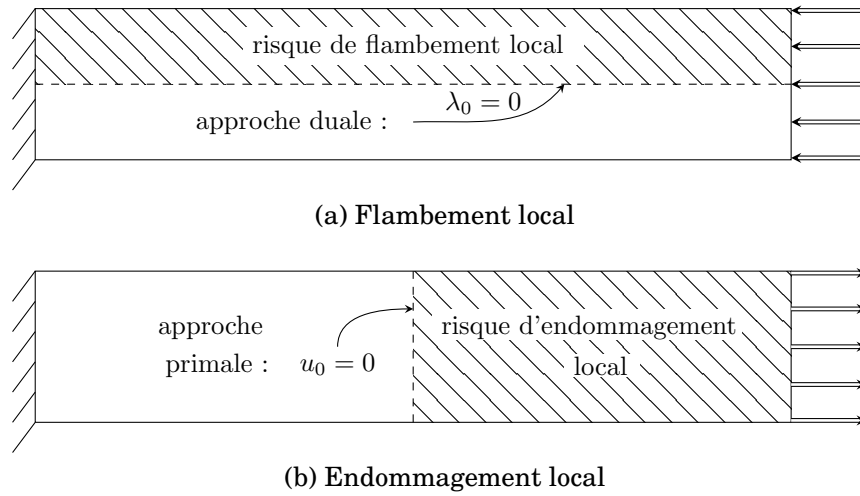


FIGURE 2.2 – Exemples de risques de non existence de solutions en mécanique non linéaire

finir la condensation de Schur des opérateurs impliqués dans ces contextes ; de plus, les compléments de Schur héritent des propriétés du problème global (typiquement, la monotonie, la coercivité et la continuité : voir [Hauer, 2015] et la bibliographie associée), ce qui assure aux problèmes globaux condensés aux interfaces (2.9), (2.14) et (2.24) d’être bien posés.

4 Stratégie de résolution

4.1 Algorithme de Newton

On fait à présent l’hypothèse d’existence locale des analogues non linéaires aux compléments de Schur par sous-domaine. Les problèmes non linéaires d’interface (2.9), (2.14) et (2.24) ont une structure que l’on peut qualifier d’« additive » : on peut les écrire sous la forme d’un assemblage de contributions locales. De façon générique, ces trois problèmes d’interface sont de la forme :

$$L(x) \equiv \sum_s L^{(s)}(x^{(s)}) = 0 \quad (2.29)$$

où $L^{(s)}$ est le complément de Schur local non linéaire défini plus haut, associé à une des trois approches primale, duale ou mixte. En primal par exemple, on a :

$$\begin{aligned} L(x) &= A^\diamond S_{nl}^\diamond \left(u_b^\diamond = A^\diamond{}^T u_A; f_{ext}^\diamond \right) \\ L^{(s)}(x^{(s)}) &= A^{(s)} S_{nl}^{(s)} \left(u_b^{(s)} = A^{(s)T} u_A; f_{ext}^{(s)} \right) \end{aligned}$$

La stratégie adoptée ici pour la résolution du problème générique (2.29) passe par un algorithme de Newton-Raphson (voir section 2.2, chapitre 1). À l'itération k , le système à résoudre devient :

$$\begin{cases} \frac{\partial L}{\partial x}(x_k) dx_k + L(x_k) = 0 \\ x_{k+1} = x_k + dx_k \end{cases} \quad (2.30)$$

Le second membre $-L(x_k)$ du problème tangent de (2.30) correspond à l'évaluation du résidu non linéaire d'interface en l'itéré x_k , obtenu par assemblage des solutions des équilibres locaux non linéaires indépendants, réalisés par sous-structure. L'évaluation du membre de gauche nécessite la construction de l'opérateur tangent du complément de Schur non linéaire, ce qui n'est pas une opération aisée a priori.

4.2 Calcul de l'opérateur tangent

L'un des principaux intérêts de la procédure choisie réside dans la possibilité de calculer simplement l'opérateur tangent condensé à chaque itération : en effet, à l'itération k , il peut s'écrire :

$$\begin{aligned} \frac{\partial L}{\partial x}(x_k) &= \left(\frac{\partial (\sum_s L^{(s)})}{\partial x} \right) \Big|_{x_k} = \sum_s \left(\frac{\partial L^{(s)}}{\partial x} \right) \Big|_{x_k^{(s)}} \\ \text{avec : } \frac{\partial L^{(s)}}{\partial x} &= \frac{\partial L^{(s)}}{\partial x^{(s)}} \frac{\partial x^{(s)}}{\partial x} \end{aligned} \quad (2.31)$$

En primal par exemple, en notant $S_{nl_A}(u_A) = A \diamond S_{nl}^\diamond (A^{\diamond T} u_A)$:

$$\begin{aligned} \frac{\partial L}{\partial x}(x_k) &= \frac{\partial S_{nl_A}}{\partial u_A}(u_{A_k}) = \sum_s A^{(s)} \left(\frac{\partial S_{nl}^{(s)}}{\partial u_b^{(s)}} \right) \Big|_{u_{b_k}^{(s)}} \frac{\partial u_b^{(s)}}{\partial u_A} \\ &= \sum_s A^{(s)} \left(\frac{\partial S_{nl}^{(s)}}{\partial u_b^{(s)}} \right) \Big|_{u_{b_k}^{(s)}} \frac{\partial (A^{(s)T} u_A)}{\partial u_A} \\ &= \sum_s A^{(s)} \left(\frac{\partial S_{nl}^{(s)}}{\partial u_b^{(s)}} \right) \Big|_{u_{b_k}^{(s)}} A^{(s)T} = \sum_s \left(\frac{\partial L^{(s)}}{\partial x^{(s)}} \right) \Big|_{x_k^{(s)}} \end{aligned}$$

On a donc exprimé l'opérateur global condensé tangent en fonction de la dérivée des opérateurs condensés locaux. Ces derniers ne sont autres, pour un sous-domaine donné, que la trace sur le bord de la solution de l'équilibre local (i.e. réaction nodale d'interface pour des conditions de Dirichlet en primal, ou déplacement d'interface pour

des conditions de Neumann). Par exemple en primal :

$$\begin{aligned} L^{(s)}(x_k^{(s)}) &= A^{(s)} S_{nl}^{(s)}(u_{b_k}^{(s)} = A^{(s)T} u_{A_k}; f_{ext}^{(s)}) = A^{(s)} \lambda_{b_k}^{(s)} \\ &= -A^{(s)} t^{(s)} \left[f_{int}^{(s)}(u_k^{(s)}) + f_{ext}^{(s)} \right] \end{aligned} \quad (2.32)$$

En différentiant l'équilibre local autour de l'itéré $u_k^{(s)}$, on fait apparaître la matrice de rigidité tangente $K_{t_k}^{(s)}$ de la sous-structure, que l'on sait calculer numériquement :

$$\frac{\partial f_{int}^{(s)}}{\partial u^{(s)}} du_k^{(s)} + t^{(s)T} d\lambda_{b_k}^{(s)} + o\left(du_k^{(s)}\right) = 0 = -K_{t_k}^{(s)} du_k^{(s)} + t^{(s)T} d\lambda_{b_k}^{(s)} + o\left(du_k^{(s)}\right)$$

On peut alors exprimer l'incrément de réaction nodale locale en fonction de la trace de l'opérateur de rigidité tangent :

$$\begin{aligned} d\lambda_{b_k}^{(s)} &= t^{(s)} \left(K_{t_k}^{(s)} du_k^{(s)} + o\left(du_k^{(s)}\right) \right) = \left(K_{t_k}^{(s)} \right)_{bi} du_{i_k}^{(s)} + \left(K_{t_k}^{(s)} \right)_{bb} du_{b_k}^{(s)} + o\left(du_{b_k}^{(s)}\right) \\ \text{avec : } \left(K_{t_k}^{(s)} \right)_{ii} du_{i_k}^{(s)} + \left(K_{t_k}^{(s)} \right)_{ib} du_{b_k}^{(s)} + o\left(du_{i_k}^{(s)}\right) &= 0 \\ \Rightarrow du_{i_k}^{(s)} &= -\left(K_{t_k}^{(s)} \right)_{ii}^{-1} \left(K_{t_k}^{(s)} \right)_{ib} du_{b_k}^{(s)} + o\left(\left(K_{t_k}^{(s)} \right)_{ii}^{-1} du_{i_k}^{(s)}\right) \end{aligned}$$

En supposant que le problème local avec conditions de Dirichlet est bien posé (i.e. $\left(K_{t_k}^{(s)} \right)_{ii}$ inversible : $\left\| \left(K_{t_k}^{(s)} \right)_{ii}^{-1} \right\| < \infty$, ce qui permet aux termes négligeables de le rester), on peut exprimer $d\lambda_{b_k}^{(s)}$ en fonction du complément de Schur tangent de la sous-structure $\Omega^{(s)}$, que l'on notera $S_{t_k}^{(s)}$:

$$\begin{aligned} d\lambda_{b_k}^{(s)} &= \left(\left(K_{t_k}^{(s)} \right)_{bb} - \left(K_{t_k}^{(s)} \right)_{bi} \left(K_{t_k}^{(s)} \right)_{ii}^{-1} \left(K_{t_k}^{(s)} \right)_{ib} \right) du_{b_k}^{(s)} \dots \\ &\quad \dots + o\left(du_{b_k}^{(s)} + \left(K_{t_k}^{(s)} \right)_{bi} \left(K_{t_k}^{(s)} \right)_{ii}^{-1} du_{i_k}^{(s)} \right) \\ &= S_{t_k}^{(s)} du_{b_k}^{(s)} + o\left(\underbrace{\left[I - \left(K_{t_k}^{(s)} \right)_{bi} \left(K_{t_k}^{(s)} \right)_{ii}^{-2} \left(K_{t_k}^{(s)} \right)_{ib} \right]}_{< \infty} du_{b_k}^{(s)} \right) + o\left(du_{i_k}^{(s)2} \right) \end{aligned} \quad (2.33)$$

Et d'après la relation (2.32), on peut exprimer l'incrément de réaction nodale d'interface en fonction de la variation du complément de Schur non linéaire primal :

$$\begin{aligned} A^{(s)} d\lambda_{b_k}^{(s)} &= A^{(s)} \left(S_{nl}^{(s)}(u_{b_k}^{(s)} + du_{b_k}^{(s)}; f_{ext}^{(s)}) - S_{nl}(u_{b_k}^{(s)}; f_{ext}^{(s)}) \right) \\ &= A^{(s)} \left(\frac{\partial S_{nl}^{(s)}}{\partial u_b^{(s)}} \right) \Big|_{u_{b_k}^{(s)}} du_{b_k}^{(s)} + o\left(A^{(s)} du_{b_k}^{(s)} \right) = A^{(s)} \left(\frac{\partial S_{nl}^{(s)}}{\partial u_b^{(s)}} \right) \Big|_{u_{b_k}^{(s)}} A^{(s)T} du_{A_k} + o\left(du_{b_k}^{(s)} \right) \end{aligned}$$

Donc au premier ordre, l'opérateur tangent condensé du problème d'interface linéarisé n'est autre que l'assemblage des condensations locales des matrices de rigidité tangentes :

$$\begin{aligned} \frac{\partial L}{\partial x}(x_k) dx_k &= \sum_s A^{(s)} \left(\frac{\partial S_{nl}^{(s)}}{\partial u_b^{(s)}} \right) \Big|_{u_{b_k}^{(s)}} A^{(s)T} du_{A_k} = \sum_s A^{(s)} d\lambda_{b_k}^{(s)} \\ &= \sum_s A^{(s)} S_{t_k}^{(s)} A^{(s)T} du_{A_k} \end{aligned}$$

Une démonstration similaire permet de montrer ce résultat pour les deux autres approches.

Les étapes principales du processus de résolution sont détaillées dans la suite pour les trois approches sus-introduites.

4.3 Étapes de résolution

4.3.1 Formulation primale

Dans le cas primal, le problème tangent de l'algorithme de Newton (2.30) devient, à chaque itération et avec les notations de la section précédente :

$$(A^{\diamond} S_{t_k}^{\diamond} A^{\diamond T}) du_{A_k} = -A^{\diamond} S_{nl}^{\diamond} (A^{\diamond T} u_{A_k}; f_{ext}^{\diamond}) \quad (2.34)$$

où l'opérateur $S_{t_k}^{(s)} = \frac{\partial S_{nl}^{(s)}}{\partial u_b^{(s)}} (u_{b_k}^{(s)} = A^{(s)T} u_{A_k}; f_{ext})$ est le complément de Schur tangent primal défini en (2.7).

Le second membre correspond à l'évaluation de problèmes de Dirichlet non linéaires par sous-domaine, avec pour conditions aux limites le déplacement d'interface imposé u_{A_k} :

$$\text{Trouver } u_k^{\diamond} \text{ tel que : } \begin{cases} (f_{int}^{\diamond}(u_k^{\diamond}) + f_{ext}^{\diamond})_i = 0 \\ t^{\diamond} u_k^{\diamond} = A^{\diamond T} u_{A_k} \end{cases}, \quad \text{puis : } \lambda_{b_k}^{\diamond} \equiv - (f_{int}^{\diamond}(u_k^{\diamond}) + f_{ext}^{\diamond})_b$$

Les déplacements $u_k^{\diamond}$ ainsi que les réactions nodales d'interface $\lambda_{b_k}^{\diamond}$ sont obtenus à partir des solutions de ces systèmes locaux – résolus en parallèle. Le second membre du problème tangent, i.e. le résidu non linéaire d'interface à l'itération k du solveur global de Newton, consiste en la non-vérification de l'équilibre des réactions nodales d'interface : $A^{\diamond} \lambda_{b_k}^{\diamond} = A^{\diamond} S_{nl}^{\diamond} (A^{\diamond T} u_{A_k}; f_{ext}^{\diamond})$.

Le membre de gauche du problème tangent (2.34) est obtenu par assemblage des compléments de Schur primaux tangents, ce qui correspond à une formulation de méthode DD primale classique.

Ainsi, après une initialisation arbitraire de l'inconnue en déplacement u_{A_0} , le processus de résolution consiste-t-il à répéter les étapes :

- (i) résoudre des problèmes de Dirichlet non linéaires locaux indépendants sur chaque sous-domaine (en parallèle)
- (ii) déterminer le défaut d'équilibre des réactions nodales à l'interface
- (iii) résoudre le problème tangent global condensé à l'interface par application d'un solveur BDD et actualiser le déplacement d'interface

4.3.2 Formulation duale

Dans le cas de l'approche duale, le problème tangent de l'algorithme de Newton (2.30) devient :

$$\begin{cases} \left(B^{\diamond} F_{t_k}^{\diamond} B^{\diamond T} \right) d\lambda_{B_k} + B^{\diamond} R_b^{\diamond} d\alpha_k^{\diamond} = -B^{\diamond} \left(F_{nl}^{\diamond} \left(B^{\diamond T} \lambda_{B_k}; f_{ext}^{\diamond} \right) + R_b^{\diamond} \alpha_k^{\diamond} \right) \\ R_b^{\diamond T} B^{\diamond T} d\lambda_{B_k} = 0 \end{cases} \quad (2.35)$$

où l'opérateur $F_{t_k}^{(s)} = \frac{\partial F_{nl}^{(s)}}{\partial \lambda_b^{(s)}} \left(\lambda_{b_k}^{(s)} = B^{(s)T} \lambda_{B_k}; f_{ext}^{(s)} \right)$ est le complément de Schur dual local tangent en l'itéré λ_{B_k} , défini en (2.12).

De même que pour les méthodes de FETI classiques (voir section 1.2.7, chapitre 1), l'introduction d'un projecteur P_B sur l'espace $\text{Ker} \left(R_b^{\diamond T} B^{\diamond T} \right)$ permet de rechercher la solution sous forme projetée $P_B d\lambda_{B_k}$, afin de déterminer sa contribution dans un espace orthogonal à celui généré par les modes rigides – la contribution liée aux modes rigides étant captée à l'initialisation du solveur global :

$$P_B^T \left(B^{\diamond} F_{t_k}^{\diamond} B^{\diamond T} \right) P_B d\lambda_{B_k} = -P_B^T B^{\diamond} F_{nl}^{\diamond} \left(B^{\diamond T} \lambda_{B_k}; f_{ext}^{\diamond} \right)$$

Le second membre du système (2.35) est obtenu par assemblage des solutions des équilibres locaux non linéaires par sous-domaine avec conditions aux limites de Neumann – réaction nodale d'interface imposée λ_{B_k} :

$$f_{int}^{\diamond} (u_k^{\diamond}) + f_{ext}^{\diamond} + t^{\diamond T} B^{\diamond T} \lambda_{B_k} = 0 \quad (2.36)$$

La résolution – menée en parallèle – de ce système détermine les déplacements $u_k^{\diamond}$, dont la discontinuité d'interface projetée $P_B^T B^{\diamond T} u_k^{\diamond} = P_B^T B^{\diamond} F_{nl}^{\diamond} \left(B^{\diamond T} \lambda_{B_k}; f_{ext}^{\diamond} \right)$ n'est autre que le second membre du problème tangent.

L'assemblage des compléments de Schur duaux locaux tangents en l'itéré $\lambda_{b_k}^{\diamond}$, calculés au moment de la construction du second membre, forme l'opérateur tangent du problème dual. On obtient ainsi la formulation duale classique d'une méthode DD.

Afin que les problèmes locaux de Neumann (2.36) soient bien posés, la condition d'admissibilité par rapport aux modes rigides (2.11) doit être vérifiée. Les incréments

$P_B d\lambda_{B_k}$ vérifient la condition d'admissibilité à zéro du problème (2.35), aussi ne reste-t-il qu'à ajouter une initialisation pertinente :

$$R^{\diamond T} \left(f_{ext}^{\diamond} + t^{\diamond T} B^{\diamond T} \lambda_{B_0} \right) = 0$$

En effet, la linéarité de la condition d'admissibilité implique que toute combinaison linéaire $\lambda_B = \lambda_{B_0} + \sum_k P_B d\lambda_{B_k}$, où λ_{B_0} est défini ci-dessus, vérifie (2.11). Cette condition sur λ_{B_0} correspond à l'initialisation du solveur FETI (voir algorithme 5) : on pose

$$\lambda_{B_0} = -QG_B \left(G_B^T Q G_B \right)^{-1} G_B^T f_{ext}$$

où G_B et Q sont définis à la section 1.2.7 (chap. 1).

Ainsi, déterminer la solution du problème dual (2.14) passe par la résolution d'un problème grossier de type FETI pour initialiser l'inconnue λ_B , puis l'itération des étapes suivantes :

- (i) résoudre des problèmes de Neumann non linéaires locaux indépendants sur chaque sous-domaine (en parallèle)
- (ii) assembler le saut de déplacement aux interfaces
- (iii) résoudre le problème tangent d'interface global par application d'un solveur FETI et actualiser la réaction nodale d'interface

Remarque 10 : L'initialisation peut être améliorée par une résolution complète du problème de FETI initial.

4.3.3 Formulation mixte

L'approche mixte pour la résolution du problème (2.24) produit le problème tangent suivant :

$$\left(A^{\diamond T} \left(A^{\diamond} Q_b^{\diamond} A^{\diamond T} \right)^{-1} A^{\diamond} - H_{t_k}^{\diamond} \right) d\mu_{b_k}^{\diamond} = H_{nl}^{\diamond} \left(\mu_{b_k}^{\diamond}; f_{ext}^{\diamond}, Q_b^{\diamond} \right) - A^{\diamond T} \left(A^{\diamond} Q_b^{\diamond} A^{\diamond T} \right)^{-1} A^{\diamond} \mu_{b_k}^{\diamond} \quad (2.37)$$

où $H_{t_k}^{(s)} = \frac{\partial H_{nl}}{\partial \mu_b^{(s)}} \left(\mu_{b_k}^{(s)}; f_{ext}^{\diamond}, Q_b^{\diamond} \right)$ est le complément de Schur mixte tangent en l'itéré $\mu_{b_k}^{(s)}$, défini par (2.23).

Le premier terme du second membre du problème (2.37) correspond à l'évaluation de problèmes locaux non linéaires par sous-domaine, avec conditions aux limites de Robin matérialisées par l'impédance d'interface $Q_b^{\diamond}$ ainsi que la réaction imposée aux interfaces $\mu_{b_k}^{\diamond}$:

$$f_{int}^{\diamond} (u_k^{\diamond}) - t^{\diamond T} Q_b^{\diamond} t^{\diamond} u_k^{\diamond} + t^{\diamond T} \mu_{b_k}^{\diamond} + f_{ext}^{\diamond} = 0$$

La résolution – effectuée en parallèle – de ces systèmes permet de déterminer le déplacement $u_k^\diamond$, dont on peut déduire le résidu d'interface mixte, équivalent au second membre du problème d'interface tangent, $b_{m_k}^\diamond = u_{b_k}^\diamond - A^{\diamond T} \left(A^\diamond Q_b^\diamond A^{\diamond T} \right)^{-1} A^\diamond \mu_{b_k}^\diamond$.

La condensation mixte de la matrice de rigidité tangente, calculée en l'itéré $u_k^\diamond$ lors de l'évaluation du second membre, permet par assemblage de construire l'opérateur tangent de (2.37). Le membre de gauche de ce système correspond donc à une formulation mixte classique de solveur DD.

Après une initialisation arbitraire, la résolution du problème non linéaire mixte (2.24) passe par l'itération des étapes suivantes :

- (i) résoudre sur chaque sous-domaine (en parallèle) des problèmes non linéaires locaux indépendants avec conditions aux limites de Robin
- (ii) calculer et assembler le résidu d'interface mixte
- (iii) résoudre le problème tangent d'interface global par application d'un solveur FETI-2LM et actualiser l'inconnue mixte d'interface

4.3.4 Intersion des solveurs linéaires

Pour chacune des trois formulations présentées ci-dessus, la procédure de résolution alterne des résolutions non linéaires locales sur chaque sous-domaine, avec différents types de conditions aux limites, et des résolutions tangentes aux interfaces pour lesquelles l'emploi de solveurs DD est adapté. Comme on le verra par la suite, le choix des conditions de transmission aux interfaces impacte fortement les résolutions non linéaires par sous-domaine. Il n'en va pas de même pour le problème tangent, puisque l'opérateur linéarisé est toujours associé à la matrice de rigidité tangente des sous-domaines : on peut donc a priori choisir n'importe quelle des trois formulations pour le problème tangent.

En effet, à partir des définitions (2.7), (2.12) et (2.23), on retrouve les relations classiques entre opérateurs tangents :

$$\begin{aligned} F_t^\diamond &= S_t^{\diamond \dagger} & H_t^\diamond &= (S_t^\diamond + Q_b^\diamond)^{-1} \\ b_d^\diamond &= F_t^\diamond b_p^\diamond & b_m^\diamond &= H_t^\diamond b_p^\diamond \end{aligned}$$

Ces relations impliquent que malgré la spécificité de chaque formulation quant au résidu obtenu par assemblage des solutions non linéaires locales, n'importe lequel des solveurs linéaires peut être employé pour la phase tangente du processus de résolution. Typiquement, la formulation mixte présente des aspects intéressants pour les résolutions non linéaires, cependant elle n'est pas très utilisée dans le cadre des systèmes linéaires, pour lesquels les formulations primale (BDD) ou duale (FETI) lui sont préférées, car plus standard et surtout munies de préconditionneurs naturels et efficaces. Après les calculs locaux $H_{nl}^\diamond(\mu_{b_k}^\diamond; f_{ext}^\diamond, Q_b^\diamond)$, le résidu mixte $b_{m_k}^\diamond = H_{nl}^\diamond(\mu_{b_k}^\diamond; f_{ext}^\diamond, Q_b^\diamond) -$

$A^{\diamond T} (A^{\diamond} Q_b^{\diamond} A^{\diamond T})^{-1} A^{\diamond} \mu_{b_k}^{\diamond}$ peut être assemblé et le second membre primal équivalent $b_{p_k}^{\diamond} = (S_{t_k}^{\diamond} + Q_b^{\diamond}) b_{m_k}^{\diamond}$ en découle directement. Le défaut d'équilibre résultant $A^{\diamond} b_{p_k}^{\diamond}$ peut alors être utilisé en entrée d'un solveur tangent BDD.

Cette propriété peut être observée plus directement en introduisant la formulation équivalente (2.25) :

$$\exists! (dF_B, dv_A) \in \mathbb{R}^{n_B} \times \mathbb{R}^{n_A}, \quad d\mu_b^{\diamond} = B^{\diamond T} dF_B + Q_b^{\diamond} A^{\diamond T} dv_A$$

dans le système tangent mixte (2.37). En effet, après multiplication à gauche par $(S_{t_k}^{\diamond} + Q_b^{\diamond})$, le problème tangent devient :

$$(S_{t_k}^{\diamond} + Q_b^{\diamond}) A^{\diamond T} dv_{A_k} - (Q_b^{\diamond} A^{\diamond T} dv_{A_k} + B^{\diamond T} dF_{B_k}) = (S_{t_k}^{\diamond} + Q_b^{\diamond}) b_{m_k}^{\diamond} = b_{p_k}^{\diamond} \quad (2.38)$$

$$\Rightarrow S_{t_k}^{\diamond} A^{\diamond T} dv_{A_k} - B^{\diamond T} dF_{B_k} = b_{p_k}^{\diamond} \quad (2.39)$$

On obtient donc le problème de type BDD :

$$(A^{\diamond} S_{t_k}^{\diamond} A^{\diamond T}) dv_{A_k} = A^{\diamond} b_{p_k}^{\diamond}$$

Les quantités suivantes peuvent ensuite être retrouvées à partir des données et solutions du problème primal :

$$\begin{aligned} (2.38) &\Leftrightarrow (S_{t_k}^{\diamond} + Q_b^{\diamond}) dv_{A_k} - d\mu_{b_k}^{\diamond} = b_{p_k}^{\diamond} \Rightarrow d\mu_{b_k}^{\diamond} = (S_{t_k}^{\diamond} + Q_b^{\diamond}) A^{\diamond T} dv_{A_k} - b_{p_k}^{\diamond} \\ du_k^{\diamond} &= (K_{t_k}^{\diamond} + t^{\diamond T} Q_b^{\diamond} t^{\diamond})^{-1} (f_{ext}^{\diamond} + t^{\diamond T} d\mu_{b_k}^{\diamond}) \\ du_{b_k}^{\diamond} &= t^{\diamond} du_k^{\diamond} \\ d\lambda_{b_k}^{\diamond} &= S_{t_k}^{\diamond} du_{b_k}^{\diamond} - b_{p_k}^{\diamond} = d\mu_{b_k}^{\diamond} - Q_b^{\diamond} du_{b_k}^{\diamond} \end{aligned} \quad (2.40)$$

Remarque 11 : L'inconnue dF_B ne joue aucun rôle réel dans l'algorithme, mais elle peut également être post-traitée :

$$dF_{B_k} = B^{\diamond T \dagger} (d\mu_{b_k}^{\diamond} - Q_b^{\diamond} A^{\diamond T} dv_{A_k})$$

où $B^{\diamond T \dagger}$ est un opérateur d'assemblage pondéré (voir section 1.2.6, chapitre 1). La propriété (2.26) de séparation de l'espace $\mathbb{R}_b^{\diamond}$ implique l'indépendance de dF_B par rapport au choix de la pondération, puisqu'on a toujours $B^{\diamond T \dagger} B^{\diamond T} = I$.

Remarque 12 : *Il va de soi que si une formulation duale avait été préférée pour la résolution du problème tangent, dF_B aurait été l'inconnue principale d'un système de type FETI obtenu par multiplication à gauche de (2.37) par $B^\diamond F_{t_k}^\diamond$.*

4.4 Algorithmes

Les algorithmes 9, 10 et 11 récapitulent les principales étapes de la méthode de sous-structuration et condensation non linéaires pour les trois approches primale, duale et mixte. Par souci de clarté, seul un incrément de chargement sera ici considéré.

Remarque 13 : *En plus du critère d'arrêt du solveur de Newton global ε_{NG} , deux autres critères apparaissent : ε_K et $\varepsilon_{NL}^\diamond$. Le premier est associé au problème tangent, et peut être piloté de la même façon que dans une procédure classique de Newton inexact (voir section 2.3, chapitre 1). Le second, défini localement, détermine la précision des solutions des algorithmes de Newton locaux. La gestion de ces critères sera abordée plus en détail dans la section suivante.*

Remarque 14 : *Le paramètre $Q_b^\diamond$ doit également faire l'objet d'une étude, afin d'en déterminer une valeur permettant de tirer le meilleur parti possible, à moindre coût, de son introduction dans les équilibres locaux. Le choix de son expression sera discuté en détail au chapitre suivant.*

4.5 Analyse de l'erreur

L'analyse des différents critères d'arrêt introduits plus haut, ainsi que des liens les rattachant entre eux et aux différentes grandeurs accessibles au cours de la résolution, peut potentiellement permettre un gain en termes d'itérations, si leurs valeurs sont judicieusement choisies. On s'attachera donc ici à tenter d'exprimer les différents résidus de chaque solveur, si possible en fonction du résidu global que l'on notera r :

$$f_{int}(u) + f_{ext} = r$$

l'objectif du processus de résolution pouvant alors être formulé ainsi : $\|r\| < \varepsilon_{NG} \|f_{ext}\|$.

On montrera tout d'abord comment les critères des algorithmes de Newton locaux $\varepsilon_{NL}^\diamond$ et du solveur tangent d'interface ε_K sont reliés pour les différentes approches. Puis, le processus de sous-structuration et condensation non linéaires sera interprété

Algorithm 9 Primal approach with BDD tangent solver**define :**

$$r_{nl}^{p\Diamond}(u^\Diamond) \text{ with } r_{nl_i}^{p\Diamond}(u^\Diamond) = [f_{int}^\Diamond(u^\Diamond) + f_{ext}^\Diamond]_i \text{ and } t^\Diamond r_{nl}^{p\Diamond}(u^\Diamond) = 0_b^\Diamond$$

initialization :

$$u_{00}^\Diamond = 0^\Diamond, K_{t_{00}}^\Diamond = f'_{int}(u_{00}^\Diamond)$$

local linear initialization step :

$$\text{solve } u_0^\Diamond = -K_{t_{00}}^{\Diamond^{-1}} r_{nl}^{p\Diamond}(u_{00}^\Diamond)$$

$$\text{compute } \lambda_{b_0}^\Diamond = -t^\Diamond [f_{int}^\Diamond(u_0^\Diamond) + f_{ext}^\Diamond]$$

$$\text{assemble } b_{A_0}^p = A^\Diamond \lambda_{b_0}^\Diamond$$

set $k = 0$ **while** $\|b_{A_k}^p\| > \varepsilon_{NG}$ **do****tangent problem :**set $m = 0$ and $du_{A_{k,m}} = 0$ **while** $\|b_{A_k}^p - (A^\Diamond S_{t_{k,m}}^\Diamond A^{\Diamond T}) du_{A_{k,m}}\| > \varepsilon_K$ **do**BDD iterations with index m

$$\text{compute } du_k^\Diamond \text{ with } du_{k_b}^\Diamond = A^{\Diamond T} du_{A_{k,m}} \text{ and } du_{k_i}^\Diamond = -K_{t_{k,m_{ii}}}^{\Diamond^{-1}} K_{t_{k,m_{ib}}}^{\Diamond^{-1}} du_{k_b}^\Diamond$$

$$\text{actualize } u_{k+1}^\Diamond = u_k^\Diamond + du_k^\Diamond$$

increment $k = k + 1$ **local nonlinear step :**set $j = 0$ and $u_{k,j}^\Diamond = u_k^\Diamond$ **while** $\|r_{nl}^{p\Diamond}(u_{k,j}^\Diamond)\| > \varepsilon_{NL}$ **do**

$$u_{k,j+1}^\Diamond = u_{k,j}^\Diamond - K_{t_{k,j}}^{\Diamond^{-1}} r_{nl}^{p\Diamond}(u_{k,j}^\Diamond)$$

increment $j = j + 1$

$$\text{compute } \lambda_{b_k}^\Diamond = -t^\Diamond [f_{int}^\Diamond(u_{k,j}^\Diamond) + f_{ext}^\Diamond]$$

$$\text{assemble } b_{A_k}^p = A^\Diamond \lambda_{b_k}^\Diamond$$

Algorithm 10 Dual approach with FETI tangent solver

define :

$$r_{nl}^{d\Diamond}(u^\Diamond, \lambda_b^\Diamond) = f_{int}^\Diamond(u^\Diamond) + f_{ext}^\Diamond + t^{\Diamond T} \lambda_b^\Diamond$$

initialization :

$$u_{00}^\Diamond = 0^\Diamond, K_{t_{00}}^\Diamond = f_{int}^{\prime\Diamond}(u_{00}^\Diamond), P_B = I - QG_B (G_B^T QG_B)^{-1} G_B^T$$

admissibility condition :

$$\text{set } \lambda_{B_0} = -QG_B (G_B^T QG_B)^{-1} G_B^T f_{ext}^\Diamond$$

local linear initialization step :

$$u_0^\Diamond = -K_{t_{00}}^{\Diamond^{-1}} r_{nl}^{d\Diamond}(u_{00}^\Diamond, \lambda_{B_0}^\Diamond)$$

$$\text{assemble } b_{B_0}^d = -B^\Diamond t^\Diamond u_0^\Diamond$$

set $k = 0$

while $\|b_{B_k}^d\| > \varepsilon_{NG}$ **do**

tangent problem :

 set $m = 0$ and $d\lambda_{B_{k,m}} = 0$

while $\|P_B^T b_{B_k}^d - P_B^T (B^\Diamond F_{t_{k,m}}^\Diamond B^{\Diamond T}) P_B d\lambda_{B_{k,m}}\| > \varepsilon_K$ **do**

 FETI iterations with index m

 compute $u_{k+1}^\Diamond = u_k^\Diamond - K_{t_{k,m}}^{\Diamond^{-1}} t^{\Diamond T} B^{\Diamond T} P_B d\lambda_{B_{k,m}}$

 actualize $\lambda_{B_{k+1}} = \lambda_{B_k} + P_B d\lambda_{B_{k,m}}$

 increment $k = k + 1$

local nonlinear step :

 set $j = 0$ and $u_{k,j}^\Diamond = u_k^\Diamond$

while $\|r_{nl}^{d\Diamond}(u_{k,j}^\Diamond, \lambda_{b_k}^\Diamond)\| > \varepsilon_{NL}$ **do**

$$u_{k,j+1}^\Diamond = u_{k,j}^\Diamond - K_{t_{k,j}}^{\Diamond^{-1}} r_{nl}^{d\Diamond}(u_{k,j}^\Diamond, \lambda_{b_k}^\Diamond)$$

 increment $j = j + 1$

$$\text{assemble } b_{B_k}^d = B^\Diamond t^\Diamond u_{k,j}^\Diamond$$

Algorithm 11 Mixed approach with BDD tangent solver**define :**

$$r_{nl}^{m\Diamond}(u^\Diamond, \mu_b^\Diamond) = f_{int}^\Diamond(u^\Diamond) + f_{ext}^\Diamond - t^{\Diamond^T} Q_b^\Diamond t^\Diamond u^\Diamond + t^{\Diamond^T} \mu_b^\Diamond$$

initialization :

$$\mu_{b_0} = 0, u_{00}^\Diamond = 0^\Diamond, K_{t_{00}}^\Diamond = f'_{int}^\Diamond(u_{00}^\Diamond)$$

local linear initialization step :

$$u_0^\Diamond = -K_{t_{00}}^{\Diamond^{-1}} r_{nl}^{m\Diamond}(u_{00}^\Diamond, \mu_{b_0}^\Diamond)$$

set $k = 0$ **while** $\|r_{nl}^{m\Diamond}(u_k^\Diamond, \mu_{b_k}^\Diamond)\| + \|B^\Diamond t^\Diamond u^\Diamond\|_B > \varepsilon_{NG}$ **do****linear right hand side :**

$$b_k^{m\Diamond} = A^{\Diamond^T} (A^\Diamond Q_b^\Diamond A^{\Diamond^T})^{-1} A^\Diamond \mu_{b_k}^\Diamond - t^\Diamond u_{k,j}^\Diamond$$

$$b_k^{p\Diamond} = (S_{t_{k,j}}^\Diamond + Q_b^\Diamond) b_k^{m\Diamond}$$

tangent problem :set $m = 0$ and $dv_{A_k} = 0$ **while** $\|b_k^{p\Diamond} - (A^\Diamond S_{t_{k,m}}^\Diamond A^{\Diamond^T}) dv_{A_{k,m}}\| > \varepsilon_K$ **do**BDD iterations with index m set $\mu_{b_{k+1}}^\Diamond = \mu_{b_k}^\Diamond + d\mu_{b_{k,m}}^\Diamond$, $u_{k+1}^\Diamond = u_k^\Diamond + du_{k,m}^\Diamond$ and $\lambda_{b_{k+1}}^\Diamond = \lambda_{b_k}^\Diamond + d\lambda_{b_{k,m}}^\Diamond$ using (2.40)increment $k = k + 1$ **local nonlinear step :**set $j = 0$ and $u_{k,j}^\Diamond = u_k^\Diamond$ **while** $\|r_{nl}^{m\Diamond}(u_{k,j}^\Diamond, \mu_{b,k}^\Diamond)\| > \varepsilon_{NL}$ **do**

$$u_{k,j+1}^\Diamond = u_{k,j}^\Diamond - (K_{t_{k,j}}^\Diamond + t^{\Diamond^T} Q_b^\Diamond t^\Diamond)^{-1} r_{nl}^{m\Diamond}(u_{k,j}^\Diamond, \mu_{b,k}^\Diamond)$$

increment $j = j + 1$

comme un solveur de Newton inexact (voir section 2.3, chapitre 1), et on envisagera l'adaptation de ces critères à chaque itération globale, en fonction de l'erreur au pas courant $\|r\|$. Les critères d'arrêt $\varepsilon_{NL}^\diamond$ et ε_K seront interprétés comme des *forcing terms* [Eisenstat and Walker, 1996] en ce qu'ils contrôlent la précision de la solution. Ils n'ont a priori besoin d'être proches de l'objectif global ε_{NG} que lorsque la convergence globale est quasiment atteinte, tandis qu'au début du processus de résolution les risques de sur-résolution devraient plutôt inciter à les relaxer.

4.5.1 Approche primale

Solveurs de Newton locaux Les itérations locales effectuées à chaque itération du solveur global sont, dans le cas de l'approche primale, dédiées à la résolution de problèmes de Dirichlet. En faisant l'hypothèse que ces conditions sont prises en comptes exactement, la convergence est contrôlée par le résidu des nœuds internes (voir algorithme 9) :

$$r_i^\diamond = f_{int_i}^\diamond(u^\diamond) + f_{ext_i}^\diamond \quad \text{avec} \quad \|r_i^\diamond\| < \varepsilon_{NL}^\diamond \|f_{ext_i}^\diamond\|$$

La donnée $u^\diamond$ déterminée grâce à ces équilibres locaux permet de calculer les réactions nodales d'interface, suivant la relation :

$$\lambda_b^\diamond \equiv - \left[f_{int_b}^\diamond(u^\diamond) + f_{ext_b}^\diamond \right]$$

Lien avec l'erreur globale Le cas primal simplifie l'analyse de l'erreur, puisque l'espace de recherche des déplacements est identique à celui du cas non sous-structuré – en effet, les déplacements sont choisis continus aux interfaces. On a alors directement :

$$\begin{aligned} \|f_{int}(u) + f_{ext}\| &\leq \|f_{int_i}^\diamond(u^\diamond) + f_{ext_i}^\diamond\| + \|A^\diamond (f_{int_b}^\diamond(u^\diamond) + f_{ext_b}^\diamond)\| \\ &\leq \varepsilon_{NL}^\diamond \|f_{ext_i}^\diamond\| + \|A^\diamond \lambda_b^\diamond\| \end{aligned}$$

Ainsi, la convergence globale est atteinte lorsque à la fois les résidus relatifs des Newton locaux ainsi que le défaut d'équilibre de l'interface sont faibles.

Solveur d'interface tangent Du point de vue du complément de Schur, le calcul des réactions nodales d'interface est perturbé par l'erreur acceptée sur les déplacements internes. En notant $S_{nl}^\diamond(A^{\diamond T} u_{A_k}; f_{ext}^\diamond)$ la réaction exacte (i.e. correspondant à $r_i^\diamond = 0^\diamond$), on a :

$$\lambda_b^\diamond = S_{nl}^\diamond(A^{\diamond T} u_{A_k}; f_{ext}^\diamond) + r_b^\diamond \quad (2.41)$$

Dans le cas linéaire, le résidu interne $r_i^\diamond$ se propagerait sur les degrés de liberté de bord en $r_b^\diamond = -K_{bi}K_{ii}^{-1}r_i^\diamond$:

$$S^\diamond u_b^\diamond = b_p^\diamond + \lambda_b^\diamond + K_{bi}K_{ii}^{-1}r_i^\diamond$$

Le calcul de $\lambda_b^\diamond$ une fois $u_b^\diamond$ déterminée n'apporte pas d'erreur additionnelle.

On peut alors faire l'hypothèse d'un ordre de grandeur similaire entre les erreurs locales interne et de bord : $\|r_i^\diamond\| \sim \|r_b^\diamond\|$, et l'erreur en découplant sur la solution du problème d'interface tangent peut être évaluée. L'incrément solution du_{A_k} satisfait à la relation :

$$\left(A^\diamond S_{t_k}^\diamond A^{\diamond T}\right) du_{A_k} = -A^\diamond S_{nl}^\diamond \left(A^{\diamond T} u_{A_k}; f_{ext}^\diamond\right) - A^\diamond r_b^\diamond + r_{A_k}^L$$

Le contrôle du résidu linéaire $r_{A_k}^L$ par le critère tangent : $\|r_{A_k}^L\| < \varepsilon_K \|r_{A_0}^L\|$, permet de contraindre l'erreur du solveur de Krylov sur l'incrément solution du_{A_k} du problème linéarisé. Les opérateurs d'assemblage primaux vérifiant $\|A^\diamond\| = O(1)$, l'influence de l'erreur des solveurs de Newton locaux semble être du même ordre de grandeur que celle du solveur tangent d'interface sur la solution du_{A_k} . On pourra donc considérer une approche consistant à fixer : $\varepsilon_{NL}^{(s)} = \varepsilon_K$.

4.5.2 Approche duale

Solveurs de Newton locaux Les itérations de Newton locales effectuées à chaque itération du solveur global sont, dans le cas de l'approche duale, dédiées à la résolution de problèmes de Neumann. La convergence est contrôlée par le résidu local (voir algorithme 10) :

$$f_{int}^\diamond(u^\diamond) + f_{ext}^\diamond + t^{\diamond T} B^{\diamond T} \lambda_B = r^\diamond \text{ avec } \|r^\diamond\| < \varepsilon_{NL}^\diamond \|f_{ext}^\diamond\|$$

Le déplacement d'interface $u_b^\diamond$ est obtenu en sortie de ces équilibres locaux, cependant il n'existe pas de lien direct entre l'erreur associée, que l'on notera $\delta_b^\diamond$, et le résidu local $r^\diamond$. La trace de la dernière correction apportée au déplacement $u_k^\diamond$ par les solveurs de Newton locaux donne toutefois a priori une bonne estimation de $\delta_b^\diamond$: on pourrait envisager un pilotage non pas en résidu d'effort mais en incrément de déplacement, avec un critère du type $\|du_b^\diamond\| < \varepsilon_\delta^\diamond \|du_{b_0}^\diamond\|$, où $du_{b_0}^\diamond$ est la trace du premier incrément de déplacement de l'algorithme de Newton considéré, et $\varepsilon_\delta^\diamond$ un seuil de précision minimale à respecter. Cette procédure impliquerait toutefois la réalisation d'au moins deux itérations par sous-domaine, à chaque étape locale : il semblerait alors nécessaire de jouer avec un critère absolu pour éviter ce problème. Une autre stratégie envisageable pourrait imposer la vérification des deux critères.

Lien avec l'erreur globale Les déplacements obtenus avec l'approche duale ne sont pas continus, aussi, l'espace dans lequel sont cherchées les solutions (*broken space*) est plus grand que l'espace original. Dans le cas linéaire, une procédure d'ajout d'un terme de saut pondéré au déplacement d'interface a été développée dans [Parret-Fréaud et al., 2010] :

$$\hat{u}_b^\diamond = u_b^\diamond - \tilde{A}^{\diamond T} \llbracket u_b \rrbracket, \text{ avec } \llbracket u_b \rrbracket = B^\diamond u_b^\diamond$$

Cette technique permet de reconstruire à moindre coût des déplacements continus à partir de la solution discontinue. Cependant, cette stratégie n'est pas réaliste dans

le cadre non linéaire, où la détermination des champs internes à partir des champs d'interface nécessiterait un calcul trop lourd.

On peut également choisir de définir une erreur étendue au *broken space*, que l'on notera ici e_{bs} :

$$e_{bs}^2 = \|f_{int}^\diamond(u^\diamond) + f_{ext}^\diamond + t^{\diamond T} B^{\diamond T} \lambda_B\|^2 + \|B^\diamond t^\diamond u^\diamond\|_B^2 \quad (2.42)$$

où $\|\cdot\|_B$ est une norme adaptée aux discontinuités d'interface. On retrouve ici une expression cumulant le critère local avec celui du solveur d'interface tangent.

Solveur d'interface tangent Le problème tangent, résolu de façon inexacte – par un solveur itératif – peut s'écrire :

$$P_B^T \left(B^\diamond F_{t_k}^\diamond B^{\diamond T} \right) P_B d\lambda_{B_k} = -P_B^T \left(B^\diamond F_{nl}^\diamond \left(B^{\diamond T} \lambda_{B_k}; f_{ext}^\diamond \right) + B^\diamond \delta_b^\diamond \right) + r_{B_k}^L$$

Le contrôle du résidu linéaire $r_{B_k}^L$ par le critère tangent : $\|r_{B_k}^L\| < \varepsilon_K \|r_{B_0}^L\|$, permet de contraindre l'erreur du solveur de Krylov sur l'incrément solution $d\lambda_{B_k}$ du problème linéarisé. Les opérateurs d'assemblage duaux vérifiant $\|B^\diamond\| = O(1)$, l'influence de l'erreur des solveurs de Newton locaux semble être du même ordre de grandeur que celle du solveur tangent d'interface sur la solution $d\lambda_{B_k}$. Dans le cas d'un pilotage en incrément de déplacement, on pourra donc considérer une approche consistant à fixer : $\varepsilon_\delta^{(s)} = \varepsilon_K$. Concernant le critère en effort, on peut remarquer que la résolution (supposée exacte) des problèmes tangents locaux à chaque itération de Newton sur un sous-domaine donné permet d'exprimer l'incrément de déplacement de bord en fonction du résidu local en effort :

$$\underbrace{-K_t^{(s)} du_k^{(s)} + f_{int}^{(s)}(u_k^{(s)}) + f_{ext}^{(s)} + t^{(s)T} B^{(s)T} \lambda_B}_{r_k} = 0 \quad \Rightarrow \quad du_{b_k}^{(s)} = F_{t_k}^{(s)} r_{b_k}^{(s)}$$

On a alors :

$$\|du_{b_k}^{(s)}\| < \varepsilon_\delta^{(s)} \|du_{b_0}^{(s)}\| \Leftrightarrow \|F_{t_k}^{(s)} r_{b_k}^{(s)}\| < \varepsilon_\delta^{(s)} \|F_{t_0}^{(s)} r_{b_0}^{(s)}\|$$

On note $\|\cdot\|$ la norme matricielle subordonnée à $\|\cdot\|$; on a :

$$\begin{aligned} \|F_{t_0}^{(s)} r_{b_0}^{(s)}\| &\leq \|F_{t_0}^{(s)}\| \|r_{b_0}^{(s)}\| \\ \text{et } \|r_{b_k}^{(s)}\| &= \|F_{t_k}^{(s)\dagger} F_{t_k}^{(s)} r_{b_k}^{(s)}\| \leq \|F_{t_k}^{(s)\dagger}\| \|F_{t_k}^{(s)} r_{b_k}^{(s)}\| \\ \text{donc } \|du_{b_k}^{(s)}\| &< \varepsilon_\delta^{(s)} \|du_{b_0}^{(s)}\| \Rightarrow \|r_{b_k}^{(s)}\| < \varepsilon_\delta^{(s)} \|F_{t_0}^{(s)}\| \|F_{t_k}^{(s)\dagger}\| \|r_{b_0}^{(s)}\| \end{aligned}$$

A priori on a $\|F_{t_0}^{(s)}\| \|F_{t_k}^{(s)\dagger}\| \simeq \text{Cond}(F_{t_k}^{(s)})$, donc sous l'hypothèse d'un problème bien conditionné on peut également imaginer une stratégie de pilotage local en effort, avec $\varepsilon_{NL}^{(s)} = \varepsilon_K$.

4.5.3 Approche mixte

Solveur de Newton locaux Les itérations de Newton locales effectuées à chaque itération du solveur global sont, dans le cas de l'approche mixte, dédiées à la résolution de problèmes avec conditions aux limites de Robin. La convergence est contrôlée par le résidu local (voir algorithme 11) :

$$f_{int}^\diamond(u^\diamond) - t^{\diamond T} Q_b^\diamond t^\diamond u^\diamond + t^{\diamond T} \mu_b^\diamond + f_{ext}^\diamond = r^\diamond \text{ avec } \|r^\diamond\| < \varepsilon_{NL}^\diamond \|f_{ext}^\diamond\| \quad (2.43)$$

De même que dans le cas de l'approche duale, en sortie de ces équilibres locaux on obtient le déplacement d'interface $u_b^\diamond$ dont l'erreur $\delta_b^\diamond$ peut être estimée par la norme de la trace du dernier incrément calculé par les solveurs de Newton locaux.

Lien avec l'erreur globale L'espace de recherche des solutions est, ici encore, un *broken space*, nécessitant l'emploi d'une norme étendue pour évaluer l'erreur globale : non seulement les déplacements ne sont pas continus aux interfaces, mais les réactions nodales ne sont pas non plus équilibrées aux interfaces. Avec la définition (2.41) de la réaction nodale $\lambda_b^\diamond$ et la formulation équivalente de $\mu_b^\diamond$ introduite dans (2.25), on peut exprimer la trace des résidus locaux à partir de (2.43) :

$$\begin{aligned} r_b^\diamond &= t^\diamond r^\diamond = \mu_b^\diamond - Q_b^\diamond t^\diamond u^\diamond - \lambda_b^\diamond \\ &= Q_b^\diamond \left(A^{\diamond T} v_A - t^\diamond u^\diamond \right) + \left(B^{\diamond T} F_B - \lambda_B^\diamond \right) \end{aligned}$$

Cette égalité permet de relier les deux erreurs d'interface (sur $u_b^\diamond$ et sur $\lambda_b^\diamond$), on peut donc en conclure que le critère dual associé à l'erreur e_{bs} définie en (2.42) est suffisant pour contrôler la convergence globale : si les deux quantités $\|r_b^\diamond\|$ et $\|B^{\diamond T} t^\diamond u^\diamond\|_B$ sont assez faibles, alors la quantité $\|A^{\diamond T} \lambda_b^\diamond\|$ le sera également.

Solveur d'interface tangent Le problème d'interface tangent peut être exprimé en sa forme primale :

$$\left(A^{\diamond T} S_{t_k}^\diamond A^{\diamond T} \right) dv_{A_k} = A^{\diamond T} \left(Q_b^\diamond + S_{t_k}^\diamond \right) \left(u_{b_k}^\diamond + \delta_b^\diamond - A^{\diamond T} v_{A_k} \right) + r_{A_k}^L$$

Le contrôle du résidu linéaire $r_{A_k}^L$ par le critère tangent : $\|r_{A_k}^L\| < \varepsilon_K \|r_{A_0}^L\|$ permet de contraindre l'erreur du solveur de Krylov sur l'incrément solution dv_{A_k} du problème linéarisé. De par la définition de $\delta_b^\diamond = - \left(S_{t_k}^\diamond + Q_b^\diamond \right)^{-1} r_b^\diamond$, on peut s'attendre à la relation : $\|A^{\diamond T} \left(S_{t_k}^\diamond + Q_b^\diamond \right) \delta_b^\diamond\| \sim \|A^{\diamond T} t^\diamond r^\diamond\|$, ce qui incite à considérer une approche consistant à fixer un critère de convergence du solveur de Krylov du même ordre de grandeur que ceux des solveurs de Newton locaux.

4.5.4 Mise au point des critères

Les sections précédentes ont permis d'établir les connections existant entre les différents critères d'arrêt des solveurs de Newton locaux ainsi que le critère du solveur global de Krylov. L'algorithme peut alors être analysé dans le contexte des méthodes de Newton inexact. En reprenant les notations génériques de la section 4 (chap. 2), le problème d'interface tangent est résolu itérativement par :

$$\begin{aligned} \frac{\partial L}{\partial x}(x_k) dx_k &= -L(x_k) + r_k, \text{ avec } \|r_k\| < \varepsilon_K \\ x_{k+1} &= x_k + dx_k \end{aligned}$$

Le paramètre ε_K est appelé *forcing term*, et sa valeur peut être actualisée à chaque itération de Newton global, afin d'optimiser la vitesse de convergence. Des propriétés plus précises de convergence sont données dans [Eisenstat and Walker, 1994, Eisenstat and Walker, 1996] et résumées à la section 2.3 (chap. 1). On rappelle ici seulement les expressions classiquement retenues pour le réglage des *forcing terms* :

$$\begin{aligned} \varepsilon_{K_k} &= \text{constante} < 1 \\ \varepsilon_{K_k} &= \frac{1}{2^{k+1}} \end{aligned} \tag{2.44}$$

$$\begin{cases} \varepsilon_{K_k} = \gamma \left(\frac{\|L(x_k)\|}{\|L(x_{k-1})\|} \right)^\alpha \\ \gamma \in [0, 1], \quad \alpha \in]1, 2], \quad \varepsilon_{T_0} \in [0, 1[\end{cases} \tag{2.45}$$

$$\begin{cases} \varepsilon_{K_k} = \frac{\| \|L(x_k)\| - \|L(x_{k-1})\| + L'(x_k)dx_{k-1} \|}{\|L(x_{k-1})\|} \\ \varepsilon_{K_0} \in [0, 1[\end{cases} \tag{2.46}$$

L'expression (2.44) est très simple à implémenter, et permet une décroissance du critère d'arrêt avec les itérations, ce qui correspond a priori à une décroissance avec le résidu d'interface global non linéaire $\|L(x_k)\|$. Cependant, la relation avec le résidu global n'étant pas explicitée, la convergence de la série des $(x_k)_k$ ne peut théoriquement pas être meilleure que superlinéaire. Les expressions (2.45) et (2.46) quant à elles, malgré une technicité d'implémentation plus complexe, permettent d'atteindre des taux de convergence plus élevés, moyennant un bon ajustement des paramètres α , γ et de l'initialisation ε_{K_0} , ainsi que de seuils inférieurs et/ou supérieurs garantissant la convergence. Par exemple, fixer la valeur du paramètre α à $\alpha = 2$ dans l'expression (2.45) permet de retrouver une vitesse de convergence quadratique du solveur de Newton inexact. Cependant, cette valeur entraîne également un nombre d'itérations du solveur tangent plus important (critère ε_{K_k} plus faible); un compromis est finalement à rechercher entre amélioration du taux de convergence du solveur de Newton

global, et diminution de la sur-résolution pouvant apparaître au niveau du solveur de Krylov.

5 Résultats

Un test des différentes configurations de la méthode de sous-structuration et condensation non linéaires sera réalisé dans cette section, sur un problème académique mettant en œuvre un comportement non linéaire réaliste. Les méthodes proposées dans ce chapitre sont bien entendu sensibles à un grand nombre de paramètres – entre autres, le type de non linéarité (pouvant occasionner de potentielles instabilités), le découpage en sous-domaines, la répartition des non linéarités au sein du domaine (à l'intérieur d'un sous-domaine : non linéarité localisée, ou répartie sur plusieurs sous-domaines : non linéarité diffuse), les différentes variantes existant pour chaque solveur, les incréments de chargement choisis, ainsi que les critères de convergence définis.

L'étude qui va suivre est loin de se prétendre exhaustive, cependant on cherchera à en dégager les tendances majeures des différentes méthodes, et à prouver leurs atouts en termes de coûts de calcul. La fin de cette section sera ouverte sur une discussion autour des différents champs potentiels d'application de la méthode, ainsi que des performances attendues selon les cas.

5.1 Plasticité parfaite

On considère dans cette section une plaque multiperforée rectangulaire chargée en traction par déplacement imposé (voir figure 2.3 pour une représentation après prise en compte des symétries du problème). Les caractéristiques géométriques et matériau de la plaque sont données dans le tableau 2.1. Le matériau choisi est un acier au comportement élasto-plastique parfait, le critère retenu pour la plasticité est celui de Von Mises ; on se place également sous l'hypothèse de contraintes planes. Le maillage est constitué d'éléments triangulaires à trois nœuds, de longueur caractéristique $l_c = 0.01L$. Le domaine est sous-structuré en huit sous-domaines, délimités par des cercles concentriques, et numérotés en partant du centre de la plaque.

Paramètres matériau		Paramètres géométriques	
Module d'Young	$E_1 = 210.10^6$	Longueur	$L = 3$
Coefficient de Poisson	$\nu_1 = 0.3$	Hauteur	$H = 1$
Limite élastique	$\sigma_0 = 420.10^3$	Rayon central	$r = 0.15$

TABLE 2.1 – Paramètres matériau et géométriques

Un trou principal se situe au centre de la plaque (voir figure 2.3). De petites perforations sont ajoutées autour de ce trou central afin de déclencher de fortes concentra-

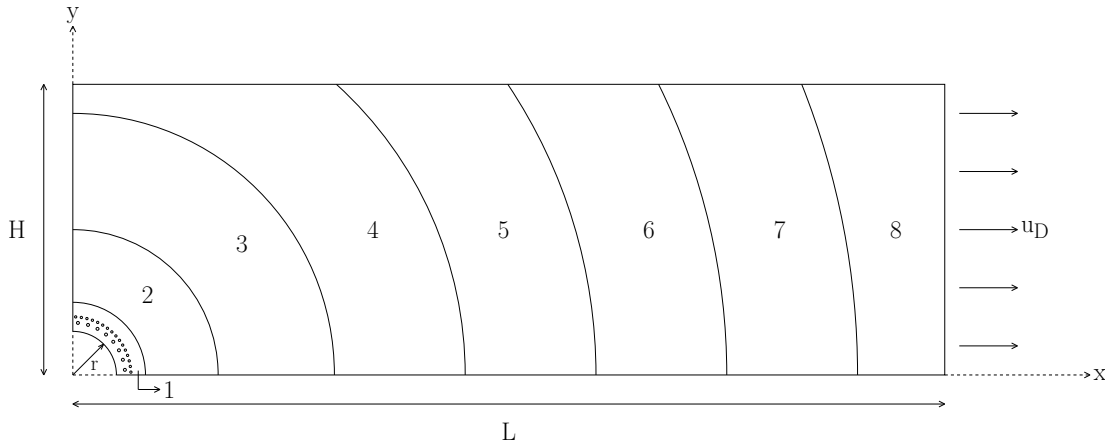


FIGURE 2.3 – Cas test : plaque multiperforée

tions de plasticité dans le sous-domaine 1. Le chargement de traction est imposé de façon incrémentale, par déplacement imposé u_D sur le bord défini par $x = L$:

$$u_D = [1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 5.75 \quad 6.5] u_e$$

$$u_e = 8.10^{-4}$$

Le premier incrément ($u_D = u_e$), correspond à un chargement dans le domaine élastique de la plaque. Les incréments compris entre $u_D = 2u_e$ et $u_D = 5u_e$ activent la plasticité dans le sous-domaine central 1, pour atteindre un maximum de plasticité cumulée $p = 0.051$. La plasticité diffuse ensuite dans les sous-domaines 2 et 3 jusqu'au maximum de plasticité cumulée $p = p_{max} = 0.15$.

Cette séquence de chargement a été utilisée tant pour l'approche classique (algorithme de Newton pour le problème global et solveur DD pour le problème tangent) que pour la méthode de sous-structuration et condensation non linéaires, avec approche primale ou mixte. Cependant, la difficulté posée par la résolution de problèmes de Neumann locaux avec un comportement matériau élasto-plastique parfait a imposé un raffinement de la discrétisation du chargement pour l'approche duale ; les incréments suivants ont dû être ajoutés à la séquence initiale :

$$u_D = [4.5, \quad 5.25, \quad 5.5, \quad 6, \quad 6.125, \quad 6.25, \quad 6.375] u_e$$

Un contrôle de la solution a permis d'assurer que cette séquence de chargement raffinée donne des résultats de qualité comparables à ceux obtenus avec la séquence plus grossière.

La suite de cette section vise à comparer l'approche classique (voir algorithme 8) avec la méthode de sous-structuration et condensation non linéaires, pour les différentes approches primale, duale et mixte. Au sein de l'approche mixte, deux types de

paramètres $Q_b^\diamond$ (ou impédance d'interface, voir section 1.2.9, chapitre 1) seront considérés. Pour une sous-structure $\Omega^{(j)}$ donnée :

- l'impédance $Q_b^{(j)} = K_{t_{bb}}^{\text{neigh}(j)} \equiv A^{(j)T} \sum_{s \neq j} A^{(s)} K_{t_{bb}}^{(s)} A^{(s)T} A^{(j)}$, soit la rigidité tangente de bord des voisins
- l'impédance $Q_b^{(j)} = S_t^{(\bar{j})}$, soit le complément de Schur tangent du complémentaire $\Omega^{(\bar{j})}$ de la sous-structure $\Omega^{(j)}$ dans Ω , choix optimal en linéaire (voir section 1.2.9, chapitre 1)

Le premier choix est peu coûteux à évaluer et courant dans le contexte mécanique parallèle. Le calcul du second choix n'est pas envisageable en pratique dans ce contexte, cependant il a été implémenté ici pour la comparaison, en tant que valeur de référence. On discutera en détail du paramètre $Q_b^\diamond$ dans le chapitre suivant ; on y introduira notamment une heuristique permettant de construire à moindre coût une impédance d'interface pertinente dans le cadre des résolutions mécaniques non linéaires parallèles.

Afin d'évaluer la performance de chaque solveur, plusieurs grandeurs sont mesurées au cours de la résolution :

- le nombre d'itérations de Newton globales, cumulées sur les incréments de chargement : on attend des approches de sous-structuration et condensation non linéaires une convergence plus rapide que la méthode classique, au sens du solveur global
- le nombre maximum de calculs de l'opérateur tangent, parmi les sous-domaines : pour l'approche classique, cette quantité est égale au nombre d'itérations de Newton globales, tandis que pour les approches sous-structurées et condensées non linéairement, il faut y ajouter le nombre de boucles de Newton locales
- le nombre d'itérations de Krylov (Gradient Conjugué) cumulées sur les itérations de Newton globales ainsi que sur les incréments de chargement : de cette quantité principalement dépendent les communications entre les sous-domaines

Les approches de sous-structuration et condensation non linéaires doivent permettre en théorie de diminuer le nombre d'itérations de Newton globales, et donc de Krylov, au prix de boucles de Newton locales supplémentaires. L'une des opérations les plus coûteuses des solveurs parallèles réside en la communication d'informations entre processeurs. Plus le nombre de sous-domaines (et donc le nombre de processeurs affectés à la résolution) augmente, plus les opérations de communication de type *all to all* (échange d'information entre tous les processeurs) deviennent coûteuses, alors que le nombre de voisins est pratiquement constant. Ainsi, les produits scalaires de vecteurs d'interface et les produits matrice-vecteur d'interface deviennent de plus en plus prédominants en termes de temps CPU, au fur et à mesure de l'augmentation de la taille du problème d'interface.

Au sein du processus de sous-structuration et condensation non linéaires, ces opérations sont quasiment l'apanage du solveur de Krylov : à chacune de ses itérations k , deux produits matrice-vecteur d'interface (dans le cas d'un solveur préconditionné : BDD ou FETI) ainsi que deux produits scalaires de vecteurs d'interface (ou jusqu'à k produits scalaires dans le cas d'une réorthogonalisation totale) sont nécessaires (voir section 1.2.5, chapitre 1). La construction du projecteur du problème grossier également, requis par les solveurs BDD et FETI, nécessite autant de produits matrice-vecteur d'interface, ainsi que de produits scalaires d'interface, qu'il y a de modes rigides dans le domaine global sous-structuré (en ce qui concerne ce dernier point, les techniques de *multiple right-hand side* permettent de réduire considérablement les coûts de calcul). Seul le calcul de la norme du résidu global non linéaire d'interface, à chaque itération de Newton globale, est une opération nécessitant des échanges d'information entre les processeurs non liée au solveur de Krylov.

L'outil de travail utilisé ici pour réaliser l'évaluation des performances de la méthode de sous-structuration et condensation non linéaires est un code Octave utilisant la bibliothèque d'envoi de messages MPI. Étant donné la petite taille de problèmes pouvant être gérée par ce code – intrinsèque au nombre maximum d'inconnues autorisé à être stocké en mémoire par Octave, – ainsi que la non optimalité de la syntaxe parallèle du code (MPI), une étude en termes de temps CPU paraît inappropriée pour juger de la performance de l'une ou l'autre des approches présentées ici. La comparaison des nombres d'itérations de Newton globales et de Krylov semble a priori être un choix plus raisonnable, et augurer des performances sur des cas test de plus grande taille, où le grand nombre de sous-domaines impliqués permettrait de gagner un temps CPU important par itération de Krylov.

Le coût de la méthode de sous-structuration et condensation non linéaires réside en l'ajout de résolutions non linéaires locales, matérialisées par un algorithme de Newton par processeur. Le nombre d'itérations locales varie alors d'un sous-domaine à l'autre, dépendamment de la répartition ainsi que de l'intensité de la non linéarité, ce qui peut rendre la résolution sujette à un mauvais *load balancing*. Ce problème n'est pas à négliger et devra faire l'objet de prochaines études – on peut citer à ce propos l'émergence de techniques parallèles dites *asynchrones*, permettant de faire travailler les processeurs sans "temps mort", soit sans réactualiser les informations d'un processeur donné à chaque itération si ses voisins n'ont pas fini de travailler [Magoulès and Venet, 2016]. Bien que la théorie ne reste actuellement valable que dans le contexte linéaire, son extension au non linéaire serait, dans notre cas, pertinente pour pallier la répartition inégale du travail entre les différents processeurs.

Un autre axe de recherche à privilégier concernant le problème du *load balancing* consisterait à opérer par étapes, avec un redécoupage possible entre chacune d'entre elles : une première résolution assez grossière permettrait de cibler les sous-domaines où la non linéarité risque d'être plus lourde à traiter, et qui devraient être redécoupés en plus petites structures, puis on affinerait la précision de la solution au fur et à mesure des résolutions, tout en adaptant la sous-structuration. Cette solution semble présenter l'avantage d'être plus simple à mettre en œuvre qu'une technique

asynchrone non linéaire.

Une dernière possibilité consisterait à rendre les processeurs en "attente" de leurs voisins disponibles pour d'autres tâches non reliées à la résolution en cours, ou tout simplement de les mettre en veille pour économiser de l'énergie, dans la mouvance du *green computing*. Quoi qu'il en soit, une réduction significative en termes d'itérations de Newton globales, et donc de Krylov (i.e. de communications), semble déjà être une proposition intéressante.

5.1.1 Méthode classique

Les performances de la méthode classique sont données dans le tableau 2.2, pour un solveur tangent BDD. Les critères d'arrêt du solveur de Newton global et du solveur BDD sont fixés à $\varepsilon_{NG} = 10^{-5}$ et $\varepsilon_K = 10^{-6}$ respectivement.

Remarque 15 : Concernant la méthode classique, le solveur BDD a ici été choisi pour la résolution des problèmes tangents, cependant un solveur FETI aurait donné un nombre similaire d'itérations de Krylov.

Incrément de chargement	u_e	$2u_e$	$5u_e$	$5.75u_e$	$6.5u_e$
Étendue de la non linéarité	Élastique	1SD plastifie		Plusieurs SDs plastifient	
# itérations globales	1	2	13	17	23
# itérations de Krylov cumulées	19	38	255	336	460

TABLE 2.2 – Performances de la méthode classique avec solveur tangent BDD et critères d'arrêt fixes

5.1.2 Méthode de sous-structuration et condensation non linéaires avec critères d'arrêt fixes

La méthode de sous-structuration et condensation non linéaires est ici comparée à la méthode classique, à critères d'arrêts fixes. Afin de donner plus de poids à cette étude, ces derniers sont fixés à des valeurs typiques dans la littérature :

$$\begin{aligned}\varepsilon_{NG} &= 10^{-5} \\ \varepsilon_{NL} &= \varepsilon_K = 10^{-6}\end{aligned}$$

	Incrément de chargement	u_e	$2u_e$	$5u_e$	$5.75u_e$	$6.5u_e$
	Étendue de la non linéarité	Élastique	SD 1		SD 1-2	SD 1-3
Newton Global	Primal/Classique	1	1	0.77	0.82	0.83
	Dual/Classique	1	1	1.46	2.12	3.35
	Mixte/Classique, $Q_b^{(j)} = K_{t_{bb}}^{\text{neigh}(j)}$	1	1	0.77	0.82	0.78
	Mixte/Classique, $Q_b^{(j)} = S_t^{(\bar{j})}$	1	1	0.62	0.65	0.70
Krylov	Primal/Classique	1	1	0.76	0.82	0.82
	Dual/Classique	0.95	0.95	1.36	1.96	3.08
	Mixte/Classique, $Q_b^{(j)} = K_{t_{bb}}^{\text{neigh}(j)}$	1	1	0.76	0.82	0.78
	Mixte/Classique, $Q_b^{(j)} = S_t^{(\bar{j})}$	1	1	0.61	0.64	0.69
Local Newton	Primal/Classique	1	2	2.62	2.88	3.04
	Dual/Classique	1	2	5.69	9.24	16.74
	Mixte/Classique, $Q_b^{(j)} = K_{t_{bb}}^{\text{neigh}(j)}$	1	2	2.69	2.94	3
	Mixte/Classique, $Q_b^{(j)} = S_t^{(\bar{j})}$	1	2	1.85	2.12	2.43

TABLE 2.3 – Itérations cumulées : ratios entre méthode sous-structurée et condensée non linéairement, et méthode classique, avec critères d'arrêt fixes

Les résultats, en termes de nombres d'itérations normalisés par ceux de l'approche classique, pour les approches primale, duale et mixte, sont répertoriés dans le tableau 2.3.

Le tableau 2.3 indique clairement une décroissance du nombre d'itérations globales cumulées lors de l'utilisation de la méthode sous-structurée et condensée non linéairement, avec les approches primale ou mixte, comparée à la méthode classique : à la fin de la résolution, on obtient environ 20% de moins pour les méthodes primale et mixte- $K_{t_{bb}}^{\text{neigh}(j)}$, et jusqu'à 30% pour la méthode mixte- $S_t^{(\bar{j})}$. Les nombres d'itérations de la méthode duale, quant à eux, augmentent en raison des incréments de chargement supplémentaires insérés pour assurer la convergence.

Le gain en itérations de Krylov cumulées est très similaire à celui des itérations globales : en effet, chaque résolution tangente requiert un nombre d'itérations de Krylov quasiment constant au cours des itérations de Newton globales. On pressent ainsi l'utilité d'une adaptation du critère d'arrêt tangent en fonction du résidu global non linéaire, afin d'éviter la sur-résolution.

Remarque 16 : Avant l'insertion d'incrément de chargement supplémentaires, le solveur tangent FETI de la méthode duale converge en une itération de moins que le solveur BDD de la méthode primale – par itération globale – ceci expliquant le ratio $0.95 \simeq 18/19$ dans le tableau 2.3.

Les versions primale et mixte de la méthode de sous-structuration et condensation non linéaires semblent donc présenter des performances intéressantes sur cet exemple, et d'autant plus que la non linéarité reste localisée : en effet, les nombres d'itérations des solveurs de Newton global ainsi que de Krylov décroissent encore plus fortement lorsque seul le sous-domaine central (numéroté 1 sur la figure 2.3) plastifie : à l'incrément de chargement $u_D = 5u_e$, les ratios de nombres d'itérations de Krylov cumulées, pour ces deux méthodes, varient de 0.61 à 0.76 (contre de 0.69 à 0.82 à la fin de la résolution), soit une décroissance d'autant en communications.

Quant au coût de cette diminution des communications, il reste limité, puisque le nombre d'itérations locales (ou de calculs des opérateurs tangents locaux) requises n'est pas plus de 3 fois supérieur à celui de la méthode classique (et même 2 fois pour la méthode mixte avec $Q_b^{(j)} = S_t^{(\bar{j})}$). Étant donnée l'indépendance des résolutions locales, qui ne requièrent donc aucune communications entre les processeurs, le gain en temps CPU devrait être notable pour des problèmes à grand nombre d'inconnues.

L'approche duale, sur cet exemple, n'a pas donné de résultats significatifs, puisque des incréments supplémentaires ont dû être ajoutés pour assurer sa convergence. Elle ne sera pas considérée dans le paragraphe suivant, qui présentera une méthode d'adaptation des critères sur l'exemple de plaque trouée sus-introduit, cependant un deuxième type de comportement non linéaire – plasticité avec écrouissage – sera étudié dans cette section, qui permettra de mettre en évidence la dépendance des performances de l'approche choisie au problème étudié.

5.1.3 Adaptation des critères d'arrêt

Ce paragraphe a pour but d'étudier l'influence d'une adaptation des critères d'arrêt locaux ε_{NL} et tangents ε_K en fonction de l'avancement de la résolution, i.e. de la valeur du résidu global non linéaire au cours des itérations globales. Deux des expressions de la section 4.5.4 (chap. 2) ont été retenues, ainsi qu'un jeu de paramètres choisis pour leurs performances (voir tableau 2.4). Les critères d'arrêt locaux ont été choisis égaux aux critères tangents ($\varepsilon_{NL} = \varepsilon_K$), en accord avec les conclusions de la section 4.5 (chap. 2). La même séquence de chargement que celle du paragraphe précédent est appliquée à la plaque multiperforée.

Les performances relatives des différentes méthodes avec adaptation des critères par les Choix 1 et 2 sont répertoriées dans les tableaux 2.5 et 2.6 respectivement. Suivant la grandeur observée, la normalisation est effectuée par rapport aux mêmes grandeurs mais avec critères fixes (nombres d'itérations de Newton globales), ou par rapport aux mêmes grandeurs avec la méthode classique et critères également adap-

Choix 1	Éq. (2.45)	$\gamma = 0.7, \alpha = 1.5, \varepsilon_{K_0} = 10^{-6}$
Choix 2	Éq. (2.46)	$\varepsilon_{K_0} = 10^{-4}$

TABLE 2.4 – Jeu de paramètres choisis pour les *forcing terms*

tés (nombres d'itérations de Krylov et de Newton locales). L'exposant ^(a) fait référence dans les deux tableaux à l'activation de l'adaptation des critères.

Remarque 17 : Seules les méthodes de sous-structuration et condensation non linéaires avec approche mixte ont été étudiées avec le Choix 2 : en effet, ce critère d'arrêt semble être trop "laxiste" pour permettre la convergence de son homologue primal, ou encore de la méthode classique.

On rappelle ici que l'objectif de l'adaptation des critères d'arrêt est d'éviter les itérations internes inutiles, sans modifier la convergence globale (i.e. les itérations de Newton globales). Cette absence de perturbation correspond aux ratios de valeur 1 dans la ligne "Newton global" des tableaux 2.5 et 2.6, vérifiée quasiment à tous les incréments de chargement.

Concernant les nombres d'itérations de Krylov, les chiffres du tableau 2.5 font état d'une diminution assez importante liée à l'adaptation des critères d'arrêt : elle est de 30% pour la méthode classique, et varie entre 20 et 25% pour les méthodes sous-structurées et condensées non linéairement. On constate alors l'apparition d'une légère baisse de performance des méthodes primale et mixte avec le Choix 1, lorsqu'on compare ces résultats à ceux du tableau 2.3 précédemment présentés (sans adaptation des critères), cependant on obtient encore un gain de 20% avec l'approche mixte- $S_t^{(j)}$.

Le tableau 2.6 quant à lui montre une hausse des performances pour l'approche mixte et le Choix 2, comparée à l'approche mixte avec critères fixes : à la fin de la résolution, on passe de 0.69 ($S_t^{(j)}$) et 0.78 ($K_{t_{bb}}^{\text{neigh}(j)}$) à 0.57 et 0.66 respectivement, tandis que la réduction des itérations de Krylov liée à l'adaptation des critères est de plus de 40% pour ces deux approches. L'adaptation des critères ne semble ainsi pas en contradiction avec la relocalisation non linéaire, et devrait être utilisée afin de réduire les échanges d'informations entre les processeurs au cours de la résolution.

L'adaptation des critères permet également de réduire les nombres d'itérations de Newton locales – correspondant au coût de la sous-structuration et condensation non linéaires : si la diminution est plutôt légère pour le Choix 1 (tableau 2.5), elle devient plus significative pour l'approche mixte avec le Choix 2 (voir tableau 2.6).

	Incrément de chargement		u_e	$2u_e$	$5u_e$	$5.75u_e$	$6.5u_e$
	Étendue de la non linéarité		Élastique	SD 1		SD 1-2	SD 1-3
Newton global	Classique ^(a) / Classique		1	1	1	1.06	1
	Primal ^(a) / Primal		1	1	1	1	1
	Mixte ^(a) / Mixte, $Q_b^{(j)} = K_{t_{bb}}^{\text{neigh}(j)}$		1	1	1	1	1.06
	Mixte ^(a) / Mixte, $Q_b^{(j)} = S_t^{(\bar{j})}$		1	1	1	1.09	1.06
Krylov	IN/fix.	Classique ^(a) / Classique	1	1	0.77	0.77	0.69
		Primal ^(a) / Primal	1	1	0.87	0.81	0.75
		Mixte ^(a) / Mixte, $Q_b^{(j)} = K_{t_{bb}}^{\text{neigh}(j)}$	1	1	0.87	0.81	0.81
		Mixte ^(a) / Mixte, $Q_b^{(j)} = S_t^{(\bar{j})}$	1	1	0.92	0.89	0.79
	NL/class.	Primal ^(a) / Classique ^(a)	1	1	0.86	0.85	0.89
		Mixte ^(a) / Classique ^(a) , $Q_b^{(j)} = K_{t_{bb}}^{\text{neigh}(j)}$	1	1	0.86	0.85	0.91
		Mixte ^(a) / Classique ^(a) , $Q_b^{(j)} = S_t^{(\bar{j})}$	1	1	0.73	0.73	0.79
Newton locaux	Classique ^(a) / Classique		1	1	1	1.06	1
	Primal ^(a) / Classique ^(a)		1	2	2.54	2.56	2.78
	Mixte ^(a) / Classique ^(a) , $Q_b^{(j)} = K_{t_{bb}}^{\text{neigh}(j)}$		1	2	2.31	2.33	2.57
	Mixte ^(a) / Classique ^(a) , $Q_b^{(j)} = S_t^{(\bar{j})}$		1	2	1.85	2.06	2.30

TABLE 2.5 – Performances de l'adaptation de critères : Choix 1

	Incrément de chargement		u_e	$2u_e$	$5u_e$	$5.75u_e$	$6.5u_e$
	Étendue de la non linéarité		Élastique	SD 1		SD 1-2	SD 1-3
Newton global	Mixte ^(a) /Mixte, $Q_b^{(j)} = K_{t_{bb}}^{\text{neigh}(j)}$		1	1	1	1	1.06
	Mixte ^(a) /Mixte, $Q_b^{(j)} = S_t^{(\bar{j})}$		1	1	1	1	1
Krylov	IN/fix.	Mixte ^(a) /Mixte, $Q_b^{(j)} = K_{t_{bb}}^{\text{neigh}(j)}$	0.79	0.79	0.71	0.63	0.59
		Mixte ^(a) /Mixte, $Q_b^{(j)} = S_t^{(\bar{j})}$	0.79	0.79	0.74	0.68	0.57
	NL/class.	Mixte ^(a) /Classique ^(a) , $Q_b^{(j)} = K_{t_{bb}}^{\text{neigh}(j)}$	1	0.79	0.71	0.67	0.66
		Mixte ^(a) /Classique ^(a) , $Q_b^{(j)} = S_t^{(\bar{j})}$	1	0.79	0.58	0.56	0.57
Newton locaux	Mixte ^(a) /Classique ^(a) , $Q_b^{(j)} = K_{t_{bb}}^{\text{neigh}(j)}$		1	2	2.23	2.28	2.43
	Mixte ^(a) /Classique ^(a) , $Q_b^{(j)} = S_t^{(\bar{j})}$		1	2	1.85	1.83	2.04

TABLE 2.6 – Performances de l’adaptation de critères : Choix 2

5.2 Écrouissage isotrope

On a vu à la section 5.1.2 (chap. 2) que la méthode de sous-structuration et condensation non linéaires avec approche duale ne convergeait pas pour un certain découpage en incréments de chargement donné, tandis que les approches primale et mixte convergeaient sans problème : des pas de chargement supplémentaires ont dû être insérés afin d’assurer la convergence. En effet, étant donné le comportement élasto-plastique parfait choisi pour le matériau de la plaque multiperforée, une discrétisation trop grossière du chargement produit des équilibres locaux à conditions aux limites de Neumann impossibles à résoudre par un algorithme de Newton : l’initialisation semble trop éloignée de la solution finale pour obtenir la convergence.

Une étude du même problème, avec ajout d’un écrouissage isotrope au comportement élasto-plastique du matériau de la plaque multiperforée, a donc été menée, afin de mettre en évidence l’influence du type de conditions d’interface choisi sur la qualité du processus de sous-structuration et condensation non linéaires, en fonction du problème considéré. Le but de cette section n’est plus de comparer les performances de la méthode classique avec celles de la méthode de sous-structuration et condensation non linéaires, mais uniquement, dans le cadre de cette dernière, de montrer la différence existant entre les approches primale et duale (on ne s’attachera pas non plus à adapter les critères d’arrêt pour optimiser la résolution).

5.2.1 Écrouissage isotrope linéaire

On envisage dans un premier temps un écrouissage de type isotrope linéaire, soit :

$$R(p) = \sigma_0 + h \cdot p \quad (2.47)$$

où p représente la déformation plastique cumulée à un chargement donné, et la fonction $R(p)$ définit la limite élastique du matériau pour une plasticité cumulée donnée. Le paramètre h est appelé coefficient d'écrouissage linéaire.

Remarque 18 : Pour un matériau au comportement élasto-plastique parfait, on a simplement $R(p) = \sigma_0$.

Le calcul de l'incrément de plasticité cumulée, à chaque itération de Newton locale, est effectué grâce à un algorithme de retour radial [Bonnet et al., 2014] sur chaque élément du maillage. L'algorithme se décompose en deux grandes étapes, à chaque itération $n + 1$, connaissant l'état n :

1. à partir de l'incrément de déformation $\Delta \varepsilon_n$ et de l'état de contrainte déviatorique s_n , la loi de comportement permet de calculer une prédiction élastique en contrainte s_{n+1}^{elas} , et le critère de Von Mises associé $\sigma_{n+1}^{\text{elas,eq}}$
2. à partir de la nouvelle prédiction élastique $\sigma_{n+1}^{\text{elas,eq}}$ et de la déformation plastique cumulée à l'itération précédente p_n , la fonction critère :

$$f_{n+1}^{\text{elas}} \equiv f(\sigma_{n+1}^{\text{elas,eq}}, p_n) = \sigma_{n+1}^{\text{elas,eq}} - R(p_n)$$

peut être évaluée. Suivant le signe de sa valeur en $(\sigma_{n+1}^{\text{elas,eq}}, p_n)$, l'évolution du matériau reste élastique, ou au contraire est sujette à la plastification :

- si $f_{n+1}^{\text{elas}} \leq 0$: l'évolution reste élastique, la variable $\sigma_{n+1} = \sigma_{n+1}^{\text{elas}}$ peut être calculée en fonction de s_{n+1}^{elas} et de la déformation $\varepsilon_n, \Delta \varepsilon_n$; l'incrément de plasticité cumulé est alors nul : $\Delta p_n = 0$
- si $f_{n+1}^{\text{elas}} > 0$: l'évolution devient élasto-plastique, et les trois étapes suivantes permettent de déterminer le nouvel état de contrainte :
 - (i) la condition de cohérence (i.e., géométriquement, la contrainte déviatorique finale s_{n+1} doit se trouver sur la surface limite définie par le critère de Von Mises dans l'espace des contraintes), soit encore :

$$\sigma_{n+1}^{\text{elas,eq}} - 3\mu\Delta p_n - R(p_n + \Delta p_n) = 0 \quad (2.48)$$

avec μ coefficient de Lamé du matériau, permet d'obtenir la variable Δp_n

- (ii) l'incrément de déformation plastique $\Delta \varepsilon_n^p$ peut être déduit de l'incrément de plasticité cumulée Δp_n , ainsi que de l'état de contrainte déviatorique élastique s_{n+1}^{elas}
- (iii) l'état de contrainte σ_{n+1} peut être calculé à partir des autres variables

Pour l'écoulement isotrope linéaire donné en (2.47), la condition de cohérence (2.48) devient :

$$\sigma_{n+1}^{\text{elas,eq}} - 3\mu\Delta p_n - (\sigma_0 + h(p_n + \Delta p_n)) = 0$$

$$\Rightarrow \Delta p_n = \frac{\sigma_{n+1}^{\text{elas,eq}} - \sigma_0 - hp_n}{3\mu + h}$$

Les résultats du tableau 2.7 correspondent à un coefficient d'écrouissage $h = 10^7$. Les critères d'arrêt tangent et locaux sont fixes : $\varepsilon_K = \varepsilon_{NL} = 10^{-6}$; le critère d'arrêt global est de nouveau fixé à $\varepsilon_{NG} = 10^{-5}$. Le chargement reste le même que pour le précédent cas-test. La plasticité cumulée varie de $p = 0$ à $p = 0.0438$ lorsque seul le sous-domaine central (numéroté 1) pastifie, puis elle augmente jusqu'à $p_{\max} = 0.1067$.

	Incrément de chargement	u_e	$2u_e$	$5u_e$	$5.75u_e$	$6.5u_e$
	Étendue de la non linéarité	Élastique	SD 1		SD 1-2	SD 1-3
Newton Global	Primal/Classique	1	1	0.77	0.82	0.82
	Dual/Classique	1	1	1	1.06	1.05
	Mixte/Classique, $Q_b^{(j)} = K_{t_{bb}}^{\text{neigh}(j)}$	1	1	0.77	0.82	0.82
	Mixte/Classique, $Q_b^{(j)} = S_t^{(\bar{j})} \text{ opti}$	1	1	0.62	0.71	0.73
Krylov	Primal/Classique	1	1	0.76	0.82	0.82
	Dual/Classique	0.95	0.95	0.93	0.98	0.97
	Mixte/Classique, $Q_b^{(j)} = K_{t_{bb}}^{\text{neigh}(j)}$	1	1	0.76	0.82	0.81
	Mixte/Classique, $Q_b^{(j)} = S_t^{(\bar{j})} \text{ opti}$	1	1	0.61	0.70	0.72
Local Newton	Primal/Classique	1	2	2.62	2.82	2.86
	Dual/Classique	1	2	3.54	3.88	4.05
	Mixte/Classique, $Q_b^{(j)} = K_{t_{bb}}^{\text{neigh}(j)}$	1	2	2.62	2.82	3
	Mixte/Classique, $Q_b^{(j)} = S_t^{(\bar{j})} \text{ opti}$	1	2	1.85	2.24	2.41

TABLE 2.7 – Itérations cumulées : ratios entre méthode sous-structurée et condensée non linéairement, et méthode classique, avec critères d'arrêt fixes, cas de l'écrouissage linéaire

Les résultats du tableau 2.7 montrent clairement une amélioration des performances de la méthode de sous-structuration et condensation non linéaires avec approche duale, bien qu'elles restent en-deçà des performances de la méthode classique : si les ratios d'itérations de Newton globales ainsi que de Krylov sont – avec l'introduction de l'écroutissage linéaire – équivalents pour ces deux méthodes, il n'en reste pas moins que des itérations locales additionnelles rendent la méthode duale moins efficace pour ce problème. Les méthodes primale et mixte restent, quant à elles, performantes comparées à la méthode classique, surtout lorsque la plasticité reste localisée dans un seul sous-domaine (gains de 25 à 40% environ).

5.2.2 Écroutissage isotrope exponentiel

Un deuxième type d'écroutissage isotrope, dit exponentiel, est considéré ici, soit :

$$R(p) = \sigma_0 + R_\infty (1 - \exp(-b.p))$$

La fonction $R(p)$ définissant la limite élastique du matériau en fonction de la plasticité cumulée est à présent paramétrée par les réels R_∞ et b : la figure 2.4 présente différents comportements d'écroutissage suivant la valeur de ces paramètres.

Afin de mieux comprendre l'influence de ces deux paramètres, on peut exprimer la contrainte en fonction de la plasticité cumulée. En effet, une fois la limite élastique atteinte, la valeur de la contrainte dans le matériau doit appartenir à la surface limite définie par le critère de plasticité de Von Mises ; en notant ε^e la déformation maximum élastique ($\varepsilon^e = \sigma_0/E = 0.002$) et ε^p la déformation plastique atteinte au sein du matériau, on a donc au cours du cycle de charge :

$$\begin{aligned} \forall \varepsilon > \varepsilon^e, \quad \sigma(\varepsilon) &= R(p) \\ \text{et } p &= \int |d\varepsilon^p| = \varepsilon^p = \varepsilon - \varepsilon^e \end{aligned}$$

La pente à partir du point où le matériau atteint sa limite élastique (ε^e, σ_0) peut donc être exprimée en fonction des paramètres de l'écroutissage :

$$\frac{d\sigma}{d\varepsilon}(\varepsilon^e) = \frac{dR}{dp}(0) \frac{dp}{d\varepsilon} = bR_\infty$$

D'autre part la valeur limite de la contrainte devient :

$$\lim_{\varepsilon \rightarrow \infty} \sigma = \sigma_0 + R_\infty$$

Pour obtenir une contrainte limite valant une fois et demi la limite élastique initiale par exemple, on peut donc prendre :

$$R_\infty = \frac{\sigma_0}{2} = 210.10^3$$

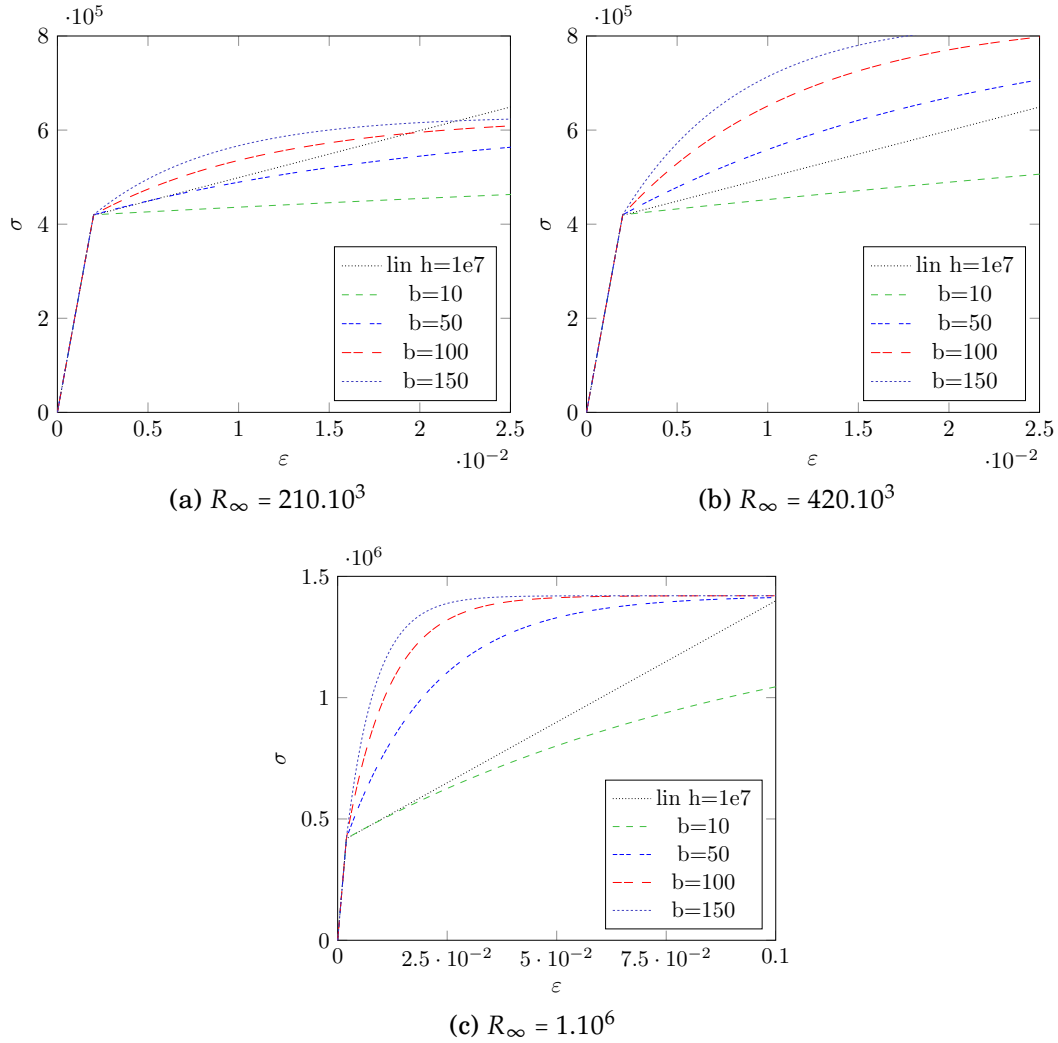


FIGURE 2.4 – Comportement du matériau pour différents paramètres d'écroissage

Et pour obtenir une pente en ε^e valant un dixième de la pente initiale du régime élastique, on peut prendre :

$$b \cdot R_{\infty} = \frac{E}{10} = 21.10^6 \Rightarrow b = 100$$

On considèrera dans la suite un écroissage avec les paramètres $R_{\infty} = 210.10^3$ et $b = 100$. La forme exponentielle de l'écroissage modifie la condition de cohérence de l'algorithme de retour radial :

$$\sigma_{n+1}^{\text{elas,eq}} - 3\mu\Delta p_n - (\sigma_0 + R_{\infty} (1 - \exp(-b \cdot (p_n + \Delta p_n)))) = 0$$

Pour déterminer l'incrément de plasticité cumulée à chaque itération de la méthode de retour radial, on doit donc résoudre une équation non linéaire en Δp_n : l'implémen-

tation d'un algorithme de Newton est alors nécessaire, appelé pour chaque point de Gauss au cours de la résolution.

Afin de ne pas surcharger ce chapitre, seuls les résultats de la comparaison de l'approche primale et duale sur le cas-test précédent sont donnés dans le tableau 2.8. Le chargement a été légèrement augmenté sur la fin de la résolution, pour obtenir des valeurs de plasticité cumulée cohérentes avec les tests précédents :

$$u_D = [u_e, \quad 2u_e, \quad 3u_e \quad 4u_e \quad 5u_e \quad 6u_e \quad 7u_e]$$

La plasticité cumulée varie alors de $p = 0$ à $p = 0.0385$ lorsque seul le sous-domaine central 1 plastifie, puis augmente jusqu'à $p_{max} = 0.1304$: la plasticité s'étend alors sur quatre sous-domaines.

	Incrément de chargement	u_e	$2u_e$	$5u_e$	$6u_e$	$7u_e$
	Étendue de la non linéarité	Élastique	SD 1		SD 1-2	SD 1-4
Newton global	Primal	1	2	10	14	19
	Dual	1	2	10	14	19
Krylov	Primal	18	38	195	275	376
	Dual	18	36	200	276	352
Newton locaux	Primal	1	4	30	44	63
	Dual	1	4	37	55	74

TABLE 2.8 – Itérations cumulées : comparaison entre la méthode primale et la méthode duale, cas de l'écroûissage exponentiel

On constate sur cet exemple que les méthodes primale et duale sont équivalentes, la méthode duale nécessitant légèrement moins d'itérations de Krylov, mais au prix de quelques itérations locales supplémentaires.

5.3 Discussion

Les différents résultats obtenus font état de la capacité de la méthode de sous-structuration et condensation non linéaires à réduire les nombres d'itérations de Newton, et donc Krylov, du solveur global. Les communications entre processeurs diminuent ainsi au cours de la résolution – comparées à celles d'une approche classique de type Newton-Krylov-Schur, – au prix de quelques itérations locales supplémentaires. Des critères d'arrêt tangents et locaux adaptés en fonction de l'avancement de la résolution ont également permis de réduire les communications, sans toutefois modifier la performance globale des différentes méthodes.

Le cas-test étudié a été choisi – entre autres raisons – pour démontrer la haute dépendance de la performance de ces méthodes au type de conditions d'interface choisi (primal, dual ou mixte). On notera tout particulièrement les difficultés de convergence occasionnées par l'utilisation de conditions de Neumann pour un comportement de matériau de type élasto-plastique parfait : en effet, dans ce cas la convergence se fait "par sur-estimation initiale", c'est-à-dire que pour chaque incrément de chargement, l'estimation de la plasticité cumulée donnée par la première itération de Newton globale est supérieure à la valeur finale, le reste des itérations faisant décroître cette valeur jusqu'à convergence. Ainsi, un incrément de chargement important peut plus aisément conduire à une divergence des algorithmes de Newton locaux, tandis que dans le cas des approches primale et mixte, la convergence se fait dans l'autre sens, donc "par sous-estimation initiale". Les cas-test avec écrouissage isotrope montrent d'autre part que l'approche duale peut devenir aussi efficace voire presque meilleure que la méthode primale, notamment dans le cas d'un fort écrouissage.

De façon plus générale, il semble aisé de construire des cas-test où l'une des variantes se comporte médiocrement comparée aux autres, ou même aboutit à une divergence du solveur. La figure 2.2 de la section 3.2 (chap. 2) de ce chapitre présente deux de ces situations : la première met en défaut l'approche duale dans le cas d'une résolution en grands déplacements, car la décomposition est effectuée de telle sorte que les sous-structures sont assez élancées, et deviennent sujettes au flambement. La deuxième met en défaut l'approche primale dans le cas d'une étude d'endommagement : en effet, la décomposition amplifie la déformation initiale et peut conduire à la ruine dès la première itération malgré un problème bien posé.

Les conclusions suivantes semblent finalement se dégager de ce chapitre :

- La littérature concernant les méthodes de Schwarz non linéaires [Badea, 1991, Ladevèze et al., 2007] incite à penser que la méthode devrait converger sans condition dans le cas d'opérateurs maximaux coercifs monotones continus (typiquement pour des matériaux à écrouissage positif en petites déformations), pour lesquels l'existence des opérateurs non linéaires de Schur, ainsi que l'unicité et la stabilité des solutions aux problèmes locaux, peuvent être démontrées [Showalter, 2013] – de par le théorème de continuité de la trace, les opérateurs condensés héritent alors des propriétés du problème global. Cependant, on constate dans la pratique que la non convergence de la méthode est possible, par divergence des équilibres non linéaires locaux si les incréments de chargement sont trop grands (apparition d'instabilités locales, artefacts de la résolution, ou emprunt d'un chemin non adéquat).
- La forme des sous-domaines joue un rôle important dans le développement potentiel des effets non linéaires.
- La condition initiale est critique pour le bon déroulement de la résolution, notamment, une initialisation nulle peut être un très mauvais point de départ pouvant engendrer des opérations locales complexes, artefacts de la résolution. Effectuer,

par exemple, une prédiction élastique préalablement aux équilibres non linéaires semble être une option plus raisonnable.

- Des conditions de Robin bien choisies devraient permettre de pousser dans la bonne direction (i.e. celle de la solution monolithique). Dans les exemples ci-dessus, le complément de Schur tangent du complémentaire de chaque sous-structure a été explicitement calculé, ce qui constitue a priori une valeur pertinente pour l'impédance d'interface, mais dont l'intérêt reste purement académique, au vu du coût numérique de la manipulation et du stockage de matrices globales. L'approximation de cet opérateur est une question récurrente de la communauté d'utilisateurs des conditions de Robin [Japhet et al., 2001, Gendreau et al., 2011], et sera discutée plus en détail dans le chapitre suivant. L'approximation étudiée dans cet exemple – i.e. la rigidité de bord des voisins – a donné de bons résultats dans le cas-test étudié, mais reste une information assez locale sur le complémentaire de la sous-structure, et pourrait ne pas suffire dans le cas d'une structure moins conforme.

6 Conclusion

Des versions non linéaires des opérateurs BDD, FETI et FETI-2LM ont été présentées dans ce chapitre, consistant à introduire des résolutions non linéaires en parallèle sur chaque sous-domaine. Le point de départ de ces nouvelles formulations réside en une *condensation non linéaire* du problème global, engendrant une évaluation parallèle du résidu par des problèmes non linéaires locaux aux conditions aux limites caractéristiques de l'approche choisie, ainsi que, après linéarisation, la résolution d'un système DD linéaire classique.

La méthode modifie le système non linéaire à résoudre et introduit des solveurs internes inexacts, dont les critères d'arrêt doivent être judicieusement adaptés afin d'éviter tout problème de sur-résolution ou, à l'inverse, de dégradation du taux de convergence global. La proposition retenue ici a été d'utiliser la théorie des méthodes de Newton inexacts.

Un cas-test simple a permis de montrer le potentiel de ces méthodes, avec une réduction intéressante en termes de nombres d'itérations de Krylov (correspondant aux communications au sein du réseau de processeurs), au prix d'itérations de Newton locales supplémentaires.

Cependant, il est clair que ces méthodes souffrent de plusieurs inconvénients. Tout d'abord, leur performance dépend du type de conditions d'interfaces choisi (primal, dual ou mixte), ce qui constitue une différence fondamentale par rapport au cas linéaire (où les approches primale et duale sont équivalentes). Deuxièmement, la forme des sous-domaines joue un rôle encore plus important que dans le cas linéaire, puisqu'elle peut déclencher de faux effets non linéaires au sein des sous-domaines. Enfin, dans le cas de non linéarité localisée, la méthode permet de concentrer les opérations uniquement sur les sous-domaines au comportement non linéaire, cependant

les sous-domaine dont le comportement reste linéaire deviennent alors inactifs, cas typique d'un mauvais *load balancing*. La question de la répartition du travail entre processeurs devra se poser dans de futurs travaux, lorsqu'une implémentation plus puissante sera possible. Typiquement, on peut espérer qu'une stratégie visant à englober le problème non linéaire dans une boucle d'adaptation du maillage et/ou de la sous-structuration serait efficace lorsqu'une précision garantie est requise par l'utilisateur, ou encore que le développement des méthodes *asynchrones* pourra s'étendre aux solveurs non linéaires.

Avant de pouvoir assurer pleinement la fiabilité de cette méthode, certains axes restent donc encore à développer. Cependant, on verra dans les chapitres suivants que des conditions de Robin bien choisies, ou encore une évolution du préconditionnement, peuvent devenir des ingrédients de sa stabilité et de son efficacité.

Conditions de Robin : impédance d'interface

Les méthodes de sous-structuration et condensation non linéaires ont été introduites au chapitre précédent, pour trois types distincts de conditions d'interface : primal, dual et mixte. Les conditions d'interface de type primal et dual sont équivalentes dans le cas linéaire sous-structuré – pour les algorithmes BDD/FETI, – cependant on a vu que dans le cas non linéaire des différences majeures pouvaient intervenir entre leurs performances respectives, dépendamment de la forme des sous-domaines, du chargement et du comportement du matériau, ainsi que de l'initialisation des solveurs. Les conditions mixtes quant à elles, permettent de donner un fort sens mécanique aux méthodes DD, en affectant une « loi de comportement » (fictive) à l'interface entre les sous-domaines – de manière générale dans le contexte des solveurs DD, voir sections 1.1.4 et 1.2.9 (chap. 1), et plus précisément dans celui de la méthode de sous-structuration et condensation non linéaire, voir section 3.1.3 (chap. 2). Elles font en effet intervenir un paramètre noté dans ce manuscrit $Q_b^\diamond$, homogène dans le cadre de la mécanique des structures à une rigidité d'interface, et jouant un rôle prépondérant dans l'efficacité du solveur en permettant d'introduire une information sur la structure globale dans les équilibres locaux.

Dans le contexte linéaire, la recherche de la valeur optimale de ce paramètre a fait l'objet de plusieurs études (voir les sections 1.1.4 et 1.2.9 du chapitre 1). Concernant les méthodes de Schur en particulier, et le cadre de l'élasticité linéaire, un choix judicieux consiste à lui assigner la rigidité de son complémentaire dans le domaine total, condensée sur son interface. Cette valeur permet de transformer les équilibres locaux en équilibres globaux, et ainsi de converger sous certaines hypothèses (voir section 1.2.9, chap. 1) en seulement autant d'itérations que nécessaires à la propagation du second membre au sein de la structure (l'ajout d'un terme correspondant à la condensation du chargement extérieur du complémentaire sur l'interface de la sous-structure permettrait de rendre la convergence immédiate). Toutefois, cette augmentation de la vitesse de convergence a un coût : en effet, le calcul du paramètre $Q_b^\diamond$

linéaire optimal n'est pas abordable en pratique dans le contexte sous-structuré, car il requiert la manipulation et le stockage de matrices globales. On en utilise en général des approximations, dont la plus classique consiste en l'assemblage des rigidités des sous-domaines voisins, tracées sur leur bord (voir section 1.2 de ce chapitre pour un examen plus détaillé de l'état de l'art dans ce domaine).

Dans le contexte non linéaire, une telle impédance n'est bien entendu plus optimale, si tant est qu'une valeur optimale existe encore. De plus, même en partant du principe qu'une approximation linéaire reste cohérente dans le cadre non linéaire, pour obtenir la meilleure d'entre elles le calcul devrait être effectué à partir des grandeurs de l'état convergé de la structure, inconnu a priori. Malgré la réalité de ces écueils, on a vu au chapitre précédent que la rapidité de convergence de la méthode de sous-structuration et condensation non linéaires variait en fonction de l'expression affectée au paramètre $Q_b^\diamond$ (voir 5, chapitre 2), suggérant l'utilité d'accorder un soin particulier au choix de ce dernier. Ce chapitre a pour vocation de détailler une nouvelle procédure de calcul de l'impédance d'interface dans le contexte sous-structuré non linéaire, à la fois peu coûteuse et plus performante – sur plusieurs cas-tests – que le choix classiquement fait en mécanique des structures, mais également que la valeur optimale en linéaire.

La procédure de calcul introduite dans ce chapitre, ainsi que les résultats obtenus, ont fait l'objet d'un article soumis en 2017 [Negrello et al., 2017].

1 Un problème bien connu des mécaniciens

1.1 Motivation

1.1.1 Élasticité linéaire

Le paramètre $Q_b^\diamond$, ou *impédance d'interface*, est introduit dans l'inconnue mixte des conditions de transmission de Robin $\mu_b^\diamond$, en tant que coefficient – homogène à une rigidité attribuée à l'interface – de la combinaison linéaire des déplacements et efforts :

$$\mu_b^\diamond = \lambda_b^\diamond + Q_b^\diamond u_b^\diamond$$

Il intervient tout au long de la résolution sous-structurée, qu'elle soit linéaire ou non, et sa pertinence en tant que représentant mécanique des interfaces locales se doit d'être soigneusement évaluée. En particulier, dans le cas de l'élasticité linéaire, en considérant le contexte plus général d'un opérateur de rigidité d'interface $Q_{b,l}^\diamond(u_b^\diamond)$, les équilibres locaux deviennent :

$$\left(S_t^{(j)} + Q_{b,l}^{(j)}\right)\left(u_b^{(j)}\right) = b_p^{(j)} + \mu_b^{(j)}$$

Ainsi, une valeur optimale peut être définie pour $Q_{b,l}^{(j)}$ en considérant la discrétisation affine de l'opérateur de Steklov-poincaré *Dirichlet-to-Neumann* du complémentaire de

chaque sous-structure $\Omega^{(j)}$:

$$Q_{b,l,opt}^{(j)}(u_b^{(j)}) \equiv S_t^{(\bar{j})} u_b^{(j)} - b_p^{(\bar{j})} = \left(K_{bb}^{(\bar{j})} - K_{bi}^{(\bar{j})} K_{ii}^{(\bar{j})^{-1}} K_{ib}^{(\bar{j})} \right) u_b^{(j)} - f_b^{(\bar{j})} + K_{bi}^{(\bar{j})} K_{ii}^{(\bar{j})^{-1}} f_i^{(\bar{j})}$$

où l'exposant $(\bar{j})$ fait référence à des quantités définies sur le complémentaire $\Omega^{(\bar{j})} \equiv \Omega \setminus \Omega^{(j)}$ de la sous-structure $\Omega^{(j)}$ dans Ω . Chaque équilibre local devient ainsi une résolution globale (basée sur la rigidité ainsi que le second membre totaux de la structure) :

$$\left(S_t^{(j)} + S_t^{(\bar{j})} \right) u_b^{(j)} = b_p^{(j)} + b_p^{(\bar{j})} + \mu_b^{(j)}$$

Pour une initialisation nulle $\mu_{b,0}^{(j)} = 0$, la solution d'un algorithme FETI-2LM muni de l'impédance $Q_{b,l,opt}^\diamond$ est obtenue à la première itération.

Remarque 1 : On recense peu de solveurs DD linéaires avec approche mixte basés sur les méthodes du complément de Schur, le plus connu étant FETI-2LM (ou 2-fields) [Roux, 2009, Gosselet and Rey, 2006], ne disposant pas d'un problème grossier naturel. Concernant les méthodes de Schwarz, l'utilisation des conditions de Robin a donné naissance aux méthodes de Schwarz optimisées : on peut citer les solveurs ORAS, SORAS entre autres [St-Cyr et al., 2007, Haferssas et al., 2015], possiblement couplés à des techniques d'augmentation GenEO [Dolean et al., 2015].

Dans le cas d'un opérateur linéaire (matriciel) $Q_{b,l}^\diamond(u_b^\diamond) = Q_t^\diamond u_b^\diamond$, il est possible de montrer [Magoulès et al., 2004, Gander and Kwok, 2011] que pour une décomposition de la structure suivant un seul axe (ou dont le graphe des connectivités ne contient pas de cycles), la définition $Q_t^{(j)} \equiv S_t^{(\bar{j})}$ est optimale, dans le sens où la convergence est atteinte en un nombre d'itérations égal au plus au nombre de sous-domaines (les itérations sont nécessaires seulement à la propagation du second membre au sein de la structure). Si un problème grossier adéquat est intégré au solveur, la convergence peut même a priori devenir quasi-instantanée. Pour une décomposition arbitraire, l'optimalité de cette valeur peut être perdue, en raison notamment de la propagation plus complexe du second membre en présence de coins (ou points multiples) [Gander and Kwok, 2011, Nier, 1998] – cependant la pertinence du complément de Schur $S_t^{(\bar{j})}$ en tant qu'impédance d'interface du sous-domaine $\Omega^{(j)}$ ne semble pas devoir être remise en question.

Partant de ces considérations, on étudie pour tout sous-domaine $\Omega^{(j)}$ l'impédance :

$$Q_b^{(j)} = K_{bb}^{(\bar{j})} - K_{bi}^{(\bar{j})} K_{ii}^{(\bar{j})^{-1}} K_{ib}^{(\bar{j})} \quad (3.1)$$

L'expression (3.1) comprend deux termes : le premier, $K_{bb}^{(\bar{j})}$, permet de prendre en compte les interactions très locales avec la structure. Ce terme est creux (bloc-diagonal), et son remplissage est identique à celui de $K_{bb}^{(j)}$. Le second terme, $K_{bi}^{(\bar{j})} K_{ii}^{(\bar{j})^{-1}} K_{ib}^{(\bar{j})}$,

permet de rendre compte des interactions avec la structure lointaine : il dépend de la structure totale (géométrie et matériau) en couplant fortement tous ses degrés de liberté, ce terme est donc une matrice pleine ; cette propriété peut être vue comme une conséquence de la non localité de l'opérateur pseudo-différentiel de Steklov-Poincaré sous-jacent, dont le complément de Schur est la discrétisation. Le signe moins entre les deux termes est également important : la partie locale de l'expression (3.1) est très rigide, et les interactions globales permettent de l'assouplir.

De par la contribution globale du terme $K_{bi}^{(\bar{j})} K_{ii}^{(\bar{j})^{-1}} K_{ib}^{(\bar{j})}$, il va sans dire que le calcul de l'impédance optimale linéaire (3.1) n'est pas envisageable dans un contexte distribué. Différentes stratégies ont été développées et mises en œuvre pour en obtenir des approximations moins coûteuses – une revue (non exhaustive) en sera présentée à la section 1.2 de ce chapitre.

1.1.2 Cas non linéaire

La question d'une impédance d'interface à la fois performante et aisément calculable en parallèle se complexifie encore dans le contexte non linéaire. En effet, si certaines techniques – que l'on verra à la section suivante – permettent d'approximer le complément de Schur du complémentaire à moindre coût, quelle pertinence ce dernier conserve-t-il en tant qu'approximation linéaire d'une impédance d'interface non linéaire ? Un tel opérateur, même calculé exactement, ne peut bien entendu plus prétendre à l'optimalité, toutefois il semble en première approche difficile de trouver une meilleure expression analytique. Considérant ainsi que dans le cadre non linéaire la valeur optimale linéaire reste pertinente, cette dernière devient cependant également dépendante de l'état de la structure auquel on la calcule, puisqu'obtenue à partir de grandeurs tangentes. Celle de ses valeurs possédant alors le plus fort sens mécanique proviendrait d'un calcul effectué à partir des grandeurs de l'état convergé final de la résolution, inconnu a priori : seule une approximation à partir de l'état initial de la structure est possible. Une technique visant à actualiser l'impédance d'interface à chaque boucle de Newton globale serait plus complexe à mettre en œuvre : en effet, ce paramètre de la méthode deviendrait alors dépendant de l'inconnue : $Q_b^\diamond(u^\diamond)$ ou encore $Q_b^\diamond(\mu_b^\diamond)$, et la résolution du problème tangent, obtenu par dérivation du résidu d'interface – faisant intervenir $Q_b^\diamond$ – par rapport à la variable $\mu_b^\diamond$, deviendrait plus complexe.

On peut toutefois, en gardant ces remarques à l'esprit, faire l'hypothèse d'une relative pertinence de l'expression (3.1) dans le contexte de la méthode de sous-structuration et condensation non linéaires, et en chercher alors une heuristique aisément calculable, que l'on appliquerait à l'état initial de la résolution.

1.2 Une revue non exhaustive des techniques d'évaluation de l'impédance d'interface

La question d'une impédance d'interface pertinente et efficace se retrouve au cœur des méthodes de décomposition de domaine mixtes, telles que les méthodes de Schwarz optimisées [Gander, 2006], de Schur ou encore de la méthode LaTIn [Ladevèze and Dureisseix, 1999]. L'une des difficultés posées par cette problématique réside dans la nécessité de rendre compte aussi bien des effets *courte distance* (comme les hétérogénéités locales) que des effets globaux sur la structure totale (comme l'anisotropie engendrée par l'élancement d'une poutre par exemple [Saavedra et al., 2012]) – effets que l'on peut qualifier de *longue distance*. Différentes stratégies ont été développées dans les trois contextes sus-introduits, tant concernant les problèmes linéaires que non linéaires, dont voici un aperçu.

1.2.1 Analyse au niveau variationnel

En se plaçant au niveau de l'analyse variationnelle, un simple scalaire (ou tenseur dans le cas anisotrope) par interface peut déjà rendre compte de certains phénomènes : dans le cadre de la méthode LaTIn par exemple, où les directions de recherche sont assimilables au paramètre de Robin dans notre cas, une interprétation en termes de grandeurs globales peut être choisie, telle que le ratio E/L_Γ , où E est le module d'Young du matériau et L_Γ la grandeur caractéristique de l'interface – un facteur de pondération peut également être introduit afin d'accélérer la résolution, correspondant par exemple à la rigidité globale, soit le ratio entre force appliquée et déflexion maximum [Saavedra et al., 2017]. La structure matricielle de la discrétisation de cet opérateur – assimilable à une matrice de masse – n'étant pas toujours adaptée, on peut avoir recours à des techniques de *lumping*, éventuellement diagonal (*super-lumping*) pour retrouver une commode sparsité.

Toujours dans le cadre de la méthode LaTIn, une stratégie extrême consiste à éliminer du raisonnement les efforts d'interface – quantités appartenant à l'espace de Sobolev fractionnaire $H^{-1/2}(\partial\Omega)$ – en introduisant un "représentant" dans l'espace $H^{1/2}(\partial\Omega)$ du travail associé [Desmeure et al., 2011]. En effet, dans le cadre sous-structuré, la méthode LaTIn permet de découpler les équations locales (comportement à l'intérieur des domaines) des équations d'interface, permettant ainsi d'associer des comportements complexes à ces dernières (lois de contact par exemple). Cependant, cette possibilité doit être associée à une discrétisation adaptée : on pourra par exemple utiliser des éléments P1 pour les déplacements internes aux sous-domaines, et des éléments P0 pour les quantités d'interface (permettant de représenter la non-continuité des efforts d'interface et sauts de déplacements d'interface). Le calcul des efforts à l'interface est alors effectué aux points de Gauss, et un raccord de type mortar [Bernardi, 1994] est nécessaire pour faire le lien entre sous-structures et interfaces, que les maillages soient compatibles ou non. Le théorème de Riesz permet d'exhiber un représentant unique dans $H^{1/2}(\partial\Omega)$ de la forme linéaire continue associée au travail

des efforts d'interface, et ainsi de s'affranchir des problèmes de discrétisation et d'interpolation. La reformulation de la méthode LaTIn sous-structurée passe alors par le produit scalaire dans $H^{1/2}(\partial\Omega)$, et on peut l'interpréter notamment comme une modification de la direction de recherche, avec – au niveau discrétisé – ajout d'un terme plein : une adaptation de cette stratégie semble peu raisonnable dans notre contexte, où l'on cherche à conserver une structure aisément exploitable pour l'impédance d'interface.

1.2.2 Analyse par transformée de Fourier

Dans le cadre des méthodes de Schwarz optimisées, où l'utilisation de conditions de Robin permet de se défaire de la nécessité du recouvrement inter-sous-domaines (voir section 1.1.4, chapitre 1), une analyse par transformée de Fourier permet d'exprimer les paramètres de Robin optimaux dans le domaine fréquentiel – correspondant dans le domaine spatial à des opérateurs non locaux, analogues aux opérateurs de Steklov-Poincaré *Dirichlet-to-Neumann*. Des approximations polynômiales dans le domaine fréquentiel (et donc locales dans le domaine spatial, puisqu'opérateurs différentiels classiques) peuvent en être dérivées ; ces dernières sont en général d'ordre 0 (conditions de Robin classiques) ou 2 (conditions de Ventcell) [Gander, 2006, Gander and Dubois, 2015]. Les conditions de Ventcell permettent notamment d'introduire de l'information sur la dérivée tangentielle à l'ordre 2, privilégiées pour l'heure par les mécaniciens des fluides plus que ceux des solides.

1.2.3 Analyse au niveau algébrique

Interactions courte distance Une analyse au niveau algébrique concerne plus directement les approximations matricielles des opérateurs de Steklov-Poincaré *Dirichlet-to-Neumann*, soit du complément de Schur du complémentaire de chaque sous-structure dans le cadre des solveurs DD avec conditions de Robin. Ce dernier peut être approximé simplement par l'assemblage des compléments de Schur locaux des voisins pour chaque sous-domaine [Cresta et al., 2007, Hinojosa et al., 2014], dans l'optique de réduire les communications – bien que la matrice reste alors dense et que les compléments de Schur locaux doivent être calculés explicitement : afin de remédier à ces écueils, la matrice de rigidité de chaque sous-domaine voisin peut n'être condensée que sur une bande d'éléments voisine de l'interface locale (dans le contexte linéaire voir [Magoulès et al., 2006]). Une procédure pour calculer de façon non-intrusive cette approximation "bande" du complément de Schur peut passer par l'ajout de bandes d'éléments externes au sous-domaine (ou *semelles*), à déplacement bloqué sur leur bord extérieur, dont on évaluerait l'effet à l'interface pour un chargement donné – ce qui revient à évaluer le complément de Schur de la semelle [Oumaziz et al., 2016], la rigidité assignée à ces bandes additionnelles d'éléments étant alors en général prise égale à une moyenne des rigidités locale et voisines. L'esprit global de cette technique rappelle celui des méthodes RAS (voir section 1.1.3, chapitre 1), les semelles pouvant

être interprétées comme des recouvrements – bien que dans le cas des méthodes RAS le bord extérieur des recouvrements soit affecté d’une condition à la limite provenant des sous-domaines voisins.

Un autre type d’approximation, dit *lumped*, peut également se révéler très efficace dans le cadre de la mécanique des structures, parfois utilisé en guise de préconditionneur FETI [Farhat, 1994], mais également d’impédance d’interface des conditions de Robin pour les solveurs DD – dans des contextes linéaires [Roux et al., 2002] comme non linéaires, notamment concernant la méthode de *relocalisation non linéaire* [Cresta et al., 2007, Negrello et al., 2016]. Étant donnée une sous-structure $\Omega^{(j)}$, on désigne par $\text{neigh}(j)$ l’ensemble de ses voisins ; les approximations *lumped* et *super-lumped* sont données par :

$$K_{t_{bb},l}^{\text{neigh}(j)} = A^{(j)T} \left(\sum_{s \in \text{neigh}(j)} A^{(s)} K_{t_{bb}}^{(s)} A^{(s)T} \right) A^{(j)} \quad (3.2)$$

$$K_{t_{bb},sl}^{\text{neigh}(j)} = A^{(j)T} \left(\sum_{s \in \text{neigh}(j)} A^{(s)} \text{diag} \left(K_{t_{bb}}^{(s)} \right) A^{(s)T} \right) A^{(j)} \quad (3.3)$$

Assemblages de termes (bloc)-diagonaux sur des ensembles réduits de sous-domaines, ces approximations sont très peu chères à construire et ne requièrent aucune opération additionnelle puisque les matrices de rigidité locales tangentes sont calculées et stockées sur chaque processeur au cours du processus naturel de résolution. Certaines études tendent à qualifier son efficacité en tant qu’impédance d’interface au sein du processus de sous-structuration et condensation non linéaires d’acceptable, comparée à celle d’autres approximations plus fines mais également plus chères, voire d’un coût de calcul inacceptable dans un contexte distribué (complément de Schur global du complémentaire : voir section 5, chapitre 2 et [Negrello et al., 2016], ou encore compléments de Schur locaux assemblés sur les sous-domaines voisins [Cresta et al., 2007]). Cependant, on peut imaginer que dans le cas de structures très hétérogènes ou élancées, cette efficacité pourrait être diminuée : en effet, l’approximation ainsi réalisée ne rend compte que des interactions d’un sous-domaine avec la structure *proche*, et dans ces cas précis l’information perdue sur le comportement global pourrait se révéler pénalisante quant à l’efficacité du solveur.

Interactions longue distance Les techniques visant à ne conserver de l’opérateur global *Dirichlet-to-Neumann* que sa restriction aux compléments de Schur locaux assemblés sur les sous-structures voisines, et plus encore leurs approximations bande, privilégient une information très localisée, au détriment des effets de la structure lointaine sur le sous-domaine considéré. La réponse de la structure avoisinante, ou encore les *interactions proches* sont correctement représentés, contrairement à leur pendant, ou *interactions lointaines*. Inspirés par les méthodes d’homogénéisation, d’autres types d’approximation on vu le jour, permettant de capter les effets globaux de la structure sur la réponse à une sollicitation locale. De façon très générale, dans le

contexte de la décomposition de domaine pour les problèmes linéaires, les interactions lointaines sont prises en compte grâce à des problèmes grossiers [Mandel, 1993, Farhat, 1994, Ladeveze and Nouy, 2003], ce qui leur permet notamment d'être conformes au principe de Saint-Venant. Étroitement liées à la notion de problème grossier, les méthodes de projection, – inspirées des techniques d'homogénéisation, – permettent d'obtenir des approximations de rang faible d'un opérateur [Ladevèze and Dureisseix, 1999, Feyel and Chaboche, 2000, Ibrahimbegović and Markovič, 2003, Guidault et al., 2008]. Génériquement, étant donnée une base orthonormale U d'un sous-espace de l'espace dans lequel on cherche la solution (par exemple, dans notre cas, un sous-espace engendré par des déplacements ou déformations d'interface choisis), l'approximation d'un opérateur $S^{(\bar{j})}$ dans ce sous-espace peut s'écrire :

$$S^{(\bar{j})} \simeq U \left(U^T S^{(\bar{j})} U \right) U^T$$

Dans le cadre d'une approximation du comportement macroscopique du complément de Schur $S^{(\bar{j})}$ du complémentaire $\Omega^{(\bar{j})}$ d'une sous-structure $\Omega^{(j)}$, le principe de Saint-Venant impose que U contienne au moins les modes rigides de chaque sous-domaine. Pour améliorer la qualité de l'approximation (ou la richesse de l'information sélectionnée par la projection), on peut leur ajouter des modes de déformation affine, ou des déplacements définis indépendamment par interface.

Qu'on se limite à la représentation des interactions *courte* ou *longue* distance, on ne pourra obtenir une information complète et précise sur la réponse structurale à une sollicitation, du point de vue d'une sous-structure. Sous l'hypothèse de petits déplacements, une mauvaise précision sur l'environnement proche d'une sous-structure semble a priori plus pénalisant que le filtre des effets structuraux plus globaux : les réactions prédominantes proviennent généralement dans cette situation des sous-domaines avoisinants ; c'est la raison pour laquelle les approximations *courte distance* du paragraphe précédent sont trouvées de façon plus récurrente au sein des solveurs mécaniques DD. Toutefois, afin d'améliorer la vitesse de convergence de ces algorithmes, la meilleure stratégie consisterait à combiner ces deux formulations. Ce couplage n'a pas, jusqu'à présent, fait l'objet d'investigations poussées. En particulier, il n'est pas si aisé d'assurer la positivité de l'impédance résultant d'un couplage de deux approximations calculées indépendamment. Une séparation d'échelle assez coûteuse a été développée dans [Gendre et al., 2011], dans le contexte non-intrusif de résolutions par schéma itératif global/local, où le complémentaire $\Omega^{(\bar{j})}$ et les grandeurs associées sont disponibles, à l'inverse de notre situation algorithmiquement pleinement distribuée.

Les sections suivantes de ce chapitre sont dédiées à la construction d'un nouveau paramètre $Q_b^\diamond$ des conditions d'interface de Robin dans la méthode de sous-structuration et condensation non linéaires avec approche mixte, combinant interactions *proches* et *lointaines* à moindre coût.

2 Une nouvelle heuristique pour le calcul de l'impédance d'interface

2.1 Un modèle basé sur l'assemblage de ressorts en série

La nouvelle heuristique introduite dans [Negrello et al., 2017] repose sur l'intuition que la construction d'une approximation à deux échelles de la souplesse du complémentaire $\Omega^{(j)}$ peut s'avérer plus évidente que celle de sa rigidité. En effet, conserver la structure additive de (3.1) devrait constituer un atout pour découpler aisément les deux échelles d'interactions (proche et lointaine) : un simple modèle de deux ressorts en série motive alors l'idée d'un raisonnement sur les souplesses (voir figure 3.1). L'un des ressorts y représente, pour une sous-structure $\Omega^{(j)}$ donnée, la rigidité de la structure avoisinante, tandis que l'autre modélise celle des sous-domaines plus lointains.

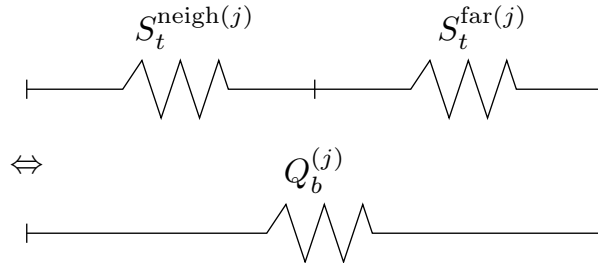


FIGURE 3.1 – Modélisation par deux ressorts en série

Grâce à leur propriété d'additivité, les souplesses sommées des deux ressorts donnent simplement la souplesse équivalente de ce modèle, soit :

$$Q_b^{(j)-1} = S_t^{\text{neigh}(j)-1} + S_t^{\text{far}(j)-1}$$

Des considérations pratiques nous amènent à proposer une structure creuse pour la matrice de rigidité de la structure proche $S_t^{\text{neigh}(j)}$ – afin de retrouver pour la composante *proche*, comme on le verra par la suite, la structure de l'expression (3.1), – ainsi qu'un rang faible pour la souplesse correspondant aux interactions lointaines $S_t^{\text{far}(j)-1}$ – ceci pour limiter les problèmes liés à l'inversion, soit remédier au profil plein de la composante *lointaine* de l'expression (3.1). Cette dernière condition est également motivée par la théorie des $\mathcal{H}$ -matrices (ou matrices hiérarchiques), permettant d'exhiber des approximations de rang faible d'inverses potentiellement de rang plein d'opérateurs issus de la discrétisation éléments finis de problèmes elliptiques [Amestoy et al., 2017, Bebendorf and Hackbusch, 2003]. Typiquement, on aura pour une sous-structure $\Omega^{(j)}$:

$$Q_b^{(j)-1} = K_{t_{bb}}^{\text{neigh}(j)-1} + A^{(j)T} V F V^T A^{(j)} \quad (3.4)$$

où l'on peut utiliser pour $K_{t_{bb}}^{\text{neigh}(j)}$ l'une des deux expressions (3.2), (3.3), F est une matrice carrée de petite taille $m \times m$ et V une base de rang faible de vecteurs d'interface, de taille $n_A \times m$. On note dans la suite $V^{(j)} = A^{(j)T} V$ la contribution locale de la base d'interface V sur le sous-domaine $\Omega^{(j)}$.

On rappelle ici la formule de Sherman-Morrisson, qui permet d'exprimer l'inverse d'une somme de matrices $M = (R + UST)$, avec R de taille $n \times n$ et S de taille $m \times m$ (et M, R, S inversibles) :

$$M^{-1} = (R + UST)^{-1} = R^{-1} - R^{-1}U(S^{-1} + TR^{-1}U)^{-1}TR^{-1}$$

L'impédance d'interface $Q_b^{(j)}$ peut alors être simplement exprimée en fonction des différentes grandeurs introduites dans ce chapitre :

$$Q_b^{(j)} = \underbrace{K_{t_{bb}}^{\text{neigh}(j)}}_{W_b^{(j)}} - \underbrace{K_{t_{bb}}^{\text{neigh}(j)} V^{(j)} \left(F^{-1} + V^{(j)T} K_{t_{bb}}^{\text{neigh}(j)} V^{(j)} \right)^{-1}}_{M^{(j)-1}} \underbrace{V^{(j)T} K_{t_{bb}}^{\text{neigh}(j)}}_{W_b^{(j)T}} \quad (3.5)$$

La structure de l'impédance ainsi définie n'est autre qu'une matrice creuse corrigée par un terme de rang faible, soit celle recherchée en s'inspirant de l'expression (3.1) et du modèle de la figure 3.1. Le calcul de $Q_b^{(j)}$ ne nécessite alors que l'inversion de deux matrices F et $M^{(j)}$, toutes deux de petite taille $m \times m$, ainsi que quelques produits locaux matrice-vecteur (opérations totalement réalisables en parallèle).

Pour résoudre les équilibres locaux avec conditions de Robin, on peut à nouveau appliquer la formule de Sherman-Morrisson :

$$\begin{aligned} \text{soient } \tilde{K}_t^{(j)} &\equiv K_t^{(j)} + t^{(j)T} K_{t_{bb}}^{\text{neigh}(j)} t^{(j)} \quad \text{et} \quad W^{(j)} \equiv t^{(j)T} W_b^{(j)} : \\ \left(K_t^{(j)} + t^{(j)T} Q_b^{(j)} t^{(j)} \right)^{-1} &= \left(\tilde{K}_t^{(j)} - W^{(j)} M^{(j)-1} W^{(j)T} \right)^{-1} \\ &= \tilde{K}_t^{(j)-1} + \tilde{K}_t^{(j)-1} W^{(j)} \left(M^{(j)} - W^{(j)T} \tilde{K}_t^{(j)-1} W^{(j)} \right)^{-1} W^{(j)T} \tilde{K}_t^{(j)-1} \end{aligned}$$

Pour un second membre donné $F^{(j)}$, la structure de $\tilde{K}_t^{(j)}$ étant la même que celle de $K_t^{(j)}$, le calcul de $\tilde{K}_t^{(j)-1} F^{(j)}$ n'est pas plus coûteux que celui de $K_t^{(j)\dagger} F^{(j)}$ – il est même simplifié : du fait de l'inversibilité de $\tilde{K}_t^{(j)}$, la nécessité de traiter les modes rigides disparaît des résolutions locales. L'ajout du terme de rang faible à l'expression de $Q_b^{(j)}$ n'implique que l'inversion d'une matrice de petite taille $m \times m$, ainsi que deux résolutions locales linéaires avec second membre multiple de taille m (soit $\tilde{K}_t^{(j)-1} W^{(j)}$).

2.2 Une interprétation multi-échelle non linéaire

À l'instar de [Oumaziz et al.,] pour la méthode LaTIn, une interprétation multi-échelle non linéaire peut être dérivée de la structure additive (3.5) choisie pour l'impé-

dance d'interface. En effet, en considérant en premier lieu le cas linéaire, les équilibres locaux condensés aux interfaces s'écrivent :

$$S_t^\diamond u_b^\diamond = b_p^\diamond + \lambda_b^\diamond \Leftrightarrow (S_t^\diamond + Q_b^\diamond) u_b^\diamond = b_p^\diamond + \mu_b^\diamond$$

Les quantités suivantes sont définies :

$$\begin{cases} \bar{\lambda}_b^\diamond = (I^\diamond - \tilde{A}^{\diamond T} A^\diamond) \lambda_b^\diamond \\ \bar{u}_b^\diamond = A^{\diamond T} \tilde{A}^\diamond u_b^\diamond \end{cases}$$

afin de pouvoir récrire les conditions d'interface comme :

$$\begin{cases} u_b^\diamond = \bar{u}_b^\diamond \\ \lambda_b^\diamond = \bar{\lambda}_b^\diamond \end{cases} \quad (3.6)$$

ou encore comme la solution de la minimisation d'une fonctionnelle :

$$\underset{\lambda_b^\diamond, u_b^\diamond \in \mathbb{R}_b^\diamond}{\operatorname{argmin}} \left[\frac{1}{2} (\lambda_b^\diamond - \bar{\lambda}_b^\diamond)^T Q_b^{\diamond -1} (\lambda_b^\diamond - \bar{\lambda}_b^\diamond) + (\lambda_b^\diamond - \bar{\lambda}_b^\diamond)^T (u_b^\diamond - \bar{u}_b^\diamond) \right]$$

L'inconnue d'interface mixte est alors cherchée comme : $\mu_b^\diamond = Q_b^\diamond \bar{u}_b^\diamond + \bar{\lambda}_b^\diamond$.

Les étapes d'une méthode DD mixte linéaire peuvent ainsi être résumées par :

- (i) Initialisation : $\mu_b^\diamond = 0$
- (ii) Résolutions parallèles : $(S_t^\diamond + Q_b^\diamond) u_b^\diamond = b_p^\diamond + \mu_b^\diamond$
- (iii) Post-processing parallèle : $\lambda_b^\diamond = S_t^\diamond u_b^\diamond - b_p^\diamond$
- (iv) Assemblage : $\bar{u}_b^\diamond, \bar{\lambda}_b^\diamond$
- (v) Actualisation de l'inconnue d'interface (en parallèle) : $\mu_b^\diamond = Q_b^\diamond \bar{u}_b^\diamond + \bar{\lambda}_b^\diamond$

Afin d'intégrer un caractère multi-échelle à la résolution, à la manière de la méthode LaTIn, une condition macroscopique peut être ajoutée sur l'équilibre des efforts d'interface, relativement à une base C_A définie sur l'interface primale globale :

$$C_A^T A^\diamond \lambda_b^\diamond = 0 \quad (3.7)$$

Les conditions d'interface (3.6) deviennent :

$$\underset{C_A^T A^\diamond \lambda_b^\diamond = 0}{\operatorname{argmin}} \left[\frac{1}{2} (\lambda_b^\diamond - \bar{\lambda}_b^\diamond)^T Q_b^{\diamond -1} (\lambda_b^\diamond - \bar{\lambda}_b^\diamond) + (\lambda_b^\diamond - \bar{\lambda}_b^\diamond)^T (u_b^\diamond - \bar{u}_b^\diamond) \right]$$

La définition d'un Lagrangien $\mathcal{L}(u_b^\diamond, \lambda_b^\diamond, \alpha_C)$ permet de prendre en compte par un multiplicateur de Lagrange α_C la condition macroscopique d'équilibre des efforts d'interface (3.7) :

$$\mathcal{L}(u_b^\diamond, \lambda_b^\diamond, \alpha_C) = \frac{1}{2} (\lambda_b^\diamond - \bar{\lambda}_b^\diamond)^T Q_b^{\diamond-1} (\lambda_b^\diamond - \bar{\lambda}_b^\diamond) + (\lambda_b^\diamond - \bar{\lambda}_b^\diamond)^T (u_b^\diamond - \bar{u}_b^\diamond) + \alpha_C^T C_A^T A^\diamond \lambda_b^\diamond$$

Enfin, la minimisation de $\mathcal{L}(u_b^\diamond, \lambda_b^\diamond, \alpha_C)$ par rapport aux efforts $\lambda_b^\diamond$ revient à résoudre la condition d'interface globale :

$$(\lambda_b^\diamond - \bar{\lambda}_b^\diamond) + Q_b^\diamond (u_b^\diamond - \bar{u}_b^\diamond) + Q_b^\diamond A^{\diamond T} C_A \alpha_C = 0 \quad (3.8)$$

En multipliant (3.8) à gauche par $C_A^T A^\diamond$ on obtient une expression du multiplicateur α_C en fonction de $u_b^\diamond$, $\bar{u}_b^\diamond$, C_A et $Q_b^\diamond$. Puis, en réinjectant cette expression dans (3.8), les réactions nodales $\lambda_b^\diamond$ peuvent être exprimées en fonction de $u_b^\diamond$, $\bar{u}_b^\diamond$, $\bar{\lambda}_b^\diamond$ et $Q_b^\diamond$. Les équilibres locaux deviennent :

$$\left[K^\diamond + t^{\diamond T} Q_b^\diamond (I^\diamond - P_{C_A}^\diamond) t^\diamond \right] u^\diamond = f_{ext}^\diamond + t^{\diamond T} \left[\bar{\lambda}_b^\diamond + Q_b^\diamond (I^\diamond - P_{C_A}^\diamond) \bar{u}_b^\diamond \right]$$

où $P_{C_A}^\diamond \equiv A^{\diamond T} C_A (C_A^T A^\diamond Q_b^\diamond A^{\diamond T} C_A)^{-1} C_A^T A^\diamond Q_b^\diamond$ est un projecteur appliqué au déplacement, dépendant simplement de l'impédance d'interface $Q_b^\diamond$ et de la base macroscopique choisie C_A . Ainsi, l'addition de la contrainte macroscopique (3.7) peut être entièrement réinterprétée comme une modification locale de l'impédance d'interface :

$$\bar{Q}_b^\diamond = Q_b^\diamond (I^\diamond - P_{C_A}^\diamond)$$

Dans le contexte non linéaire, par extension de cette vision du multi-échelle dans le cas mixte linéaire, en considérant une impédance d'interface classique du type $Q_b^{(j)} = K_{t_{bb}}^{\text{neigh}(j)} -$ précédemment définie en (3.2) ou (3.3), – la modification $\bar{Q}_b^\diamond \equiv K_{t_{bb}}^{\text{neigh}(j)} - W_b^{(j)} M^{(j)-1} W_b^{(j)T}$ avec $W_b^{(j)} = K_{t_{bb}}^{\text{neigh}(j)} \tilde{W}_b^{(j)}$ proposée en (3.5) peut être vue comme l'introduction d'un problème grossier dans la méthode de sous-structuration et condensation non linéaires avec approche mixte.

De plus, dans le contexte non linéaire, l'addition de contraintes macroscopiques non linéaires sur les efforts ou déplacements d'interface n'étant pas toujours ni possible ni aisée à réaliser, le concept de calcul multi-échelle est rarement défini ou utilisé tel quel. Dans [Klawonn et al., 2017], le solveur NL-FETIDP2 fait intervenir la résolution d'un problème grossier non linéaire portant sur les déplacements aux coins des sous-domaines, cependant ce type de contrainte macroscopique additionnelle ne semble pas si éloignée d'une simple initialisation appropriée du solveur global non linéaire, un premier calcul multi-échelle linéaire classique (comme le pré-processing linéaire BDD de l'algorithme proposé ici) permettant déjà de propager une information sur le second membre au sein de la structure.

2.3 Approximation à deux échelles de la souplesse

L'idée principale sur laquelle repose ce nouveau processus de calcul de l'impédance d'interface peut dès lors être résumée simplement : on cherche à construire une approximation de la valeur optimale linéaire (3.1) dont l'inverse – soit la souplesse – aurait la structure additive de (3.4). Pour une sous-structure $\Omega^{(j)}$ donnée, on va tout d'abord montrer que le complément de Schur tangent du complémentaire $S_t^{(\bar{j})} = K_{t_{bb}}^{(\bar{j})} - K_{t_{bi}}^{(\bar{j})} K_{t_{ii}}^{(\bar{j})^{-1}} K_{t_{ib}}^{(\bar{j})}$, que l'on cherche à approximer, peut s'exprimer en fonction de l'inverse de l'assemblage des opérateurs locaux sur $\Omega^{(\bar{j})}$. Cette grandeur pourra ensuite être évaluée grâce à la théorie des préconditionneurs BDD [Mandel, 1993].

Étant donnée une sous-structure $\Omega^{(j)}$, on note $S_A^{(\bar{j})} = \sum_{s \neq j} A^{(s)} S^{(s)} A^{(s)T}$ l'assemblage des compléments de Schur locaux sur le complémentaire $\Omega^{(\bar{j})}$. On indice $x_i^{(\bar{j})}$ les degrés de liberté internes à $\Omega^{(\bar{j})}$, et $x_b^{(\bar{j})}$ ses degrés de liberté de bord, soit appartenant à $\Gamma^{(\bar{j})} = \Gamma^{(j)}$. Avec des notations évidentes ainsi que celles de la section 1.2.1 (chap. 1), la formule du quotient nous permet d'exprimer le complément de Schur du complémentaire condensé sur son interface en fonction de $S_A^{(\bar{j})}$:

$$S_t^{(\bar{j})} = S_{A/S_{A_{i,\bar{i}}}}^{(\bar{j})}$$

La formule de l'inverse du complément de Schur permet ensuite d'exprimer l'inverse de $S_t^{(\bar{j})}$ en fonction de celui de $S_A^{(\bar{j})}$:

$$S_t^{(\bar{j})^{-1}} = \left(S_{A/S_{A_{i,\bar{i}}}}^{(\bar{j})} \right)^{-1} = \left(S_A^{(\bar{j})^{-1}} \right)_{\bar{b},\bar{b}} = A^{(j)T} S_A^{(\bar{j})^{-1}} A^{(j)}$$

Les préconditionneurs classiques BDD peuvent alors être utilisés pour approximer l'inverse de $S_A^{(\bar{j})}$, que l'on peut voir comme l'opérateur d'un problème BDD défini sur le domaine $\Omega^{(\bar{j})}$.

Remarque 2 : L'opérateur $S_A^{(\bar{j})}$ est inversible à la condition que le problème BDD associé soit bien posé sur $\Omega^{(\bar{j})}$. Cette propriété est vraie si le problème de départ (sur Ω tout entier) l'est également, que les conditions de Dirichlet ne sont pas concentrées sur $\partial\Omega \cap \partial\Omega^{(j)}$ et que le complémentaire $\Omega^{(\bar{j})}$ est connexe, ou que chaque partie connexe du complémentaire est munie d'une condition de Dirichlet sur son bord. Dans le cas contraire, un redécoupage suffit à contourner ce problème.

On introduit ici $\hat{G}_A^{(\bar{j})} = \left[\dots, \hat{A}^{(s)} R_b^{(s)}, \dots \right]_{s \neq j}$ la concaténation des modes rigides locaux ($R^{(s)}$) tracés sur le bord des sous-domaines appartenant à $\Omega^{(\bar{j})}$, et pondérés par

des opérateurs d'assemblage $\hat{A}_j^{(s)}$ prenant en compte l'absence de matière au sein du sous-domaine $\Omega^{(j)}$. Typiquement, étant donnée la définition (1.23) des opérateurs d'assemblage classiques $\tilde{A}^{(s)}$ introduits à la section 1.2.6 (chap. 1), on peut définir $\hat{A}_j^\diamond$ par :

$$\hat{A}_j^{(s)} = \begin{cases} \left(A^\diamond \Delta^\diamond A^\diamond{}^T - A^{(j)} \Delta^{(j)} A^{(j)T} \right)^{-1} A^{(s)} \Delta^{(s)} & \text{si } s \neq j \\ 0 & \text{si } s = j \end{cases}$$

avec $\Delta^{(s)} = I^{(s)}$ ou $\text{diag}(K_{t_{bb}}^{(s)})$

On considèrera dans la suite de ce manuscrit une pondération basée sur les rigidités : $\Delta^{(s)} = \text{diag}(K_{t_{bb}}^{(s)})$.

Soit P_A le projecteur $S_A^{(\bar{j})}$ -orthogonal sur $\text{Ker}(\hat{G}_A^{(\bar{j})T} S_A^{(\bar{j})})$:

$$P_A^{(\bar{j})} = I - \hat{G}_A^{(\bar{j})} \left(\hat{G}_A^{(\bar{j})T} S_A^{(\bar{j})} \hat{G}_A^{(\bar{j})} \right)^{-1} \hat{G}_A^{(\bar{j})T} S_A^{(\bar{j})}$$

On a, de manière générale :

$$S_A^{(\bar{j})-1} = P_A^{(\bar{j})} S_A^{(\bar{j})-1} P_A^{(\bar{j})T} + \left(I - P_A^{(\bar{j})} \right) S_A^{(\bar{j})-1} \left(I - P_A^{(\bar{j})} \right)^T \quad (3.9)$$

La théorie BDD justifie l'approximation de $S_A^{(\bar{j})-1}$ dans le premier terme de l'expression (3.9) par une somme pondérée d'inverses locaux¹. Après développement et simplification, l'opérateur résultant constitue une première expression possible pour la souplesse d'interface, notée $Q_{BDD}^{(j)-1}$:

$$Q_{BDD}^{(j)-1} \equiv A^{(j)T} \left(P_A^{(\bar{j})} \sum_{s \neq j} \hat{A}_j^{(s)} S_t^{(s)\dagger} \hat{A}_j^{(s)T} P_A^{(\bar{j})T} + \hat{G}_A^{(\bar{j})} \left(\hat{G}_A^{(\bar{j})T} S_A^{(\bar{j})} \hat{G}_A^{(\bar{j})} \right)^{-1} \hat{G}_A^{(\bar{j})T} \right) A^{(j)} \quad (3.10)$$

Interactions lointaines Le second terme de l'expression (3.10), que l'on notera :

$$\hat{F}_{A,2}^{(j)} \equiv A^{(j)T} \hat{G}_A^{(\bar{j})} \left(\hat{G}_A^{(\bar{j})T} S_A^{(\bar{j})} \hat{G}_A^{(\bar{j})} \right)^{-1} \hat{G}_A^{(\bar{j})T} A^{(j)}$$

est une matrice de rang faible $m^{(j)}$, où $m^{(j)}$ est le nombre de modes rigides des voisins de la sous-structure $\Omega^{(j)}$, et dont le calcul passe par l'inversion de la matrice $\hat{G}_A^{(\bar{j})T} S_A^{(\bar{j})} \hat{G}_A^{(\bar{j})}$, elle-même de rang $m^{(\bar{j})}$, où $m^{(\bar{j})}$ est le nombre de modes rigides de tous les sous-domaines du complémentaire de $\Omega^{(j)}$ dans la structure totale Ω . On pourrait

1. La théorie GenEO [Spillane and Rixen, 2013] assure également qu'au besoin, des modes supplémentaires peuvent être ajoutés à $\hat{G}_A^{(\bar{j})}$ afin d'améliorer la qualité de l'approximation.

conserver le terme $\hat{F}_{A,2}^{(j)}$ tel quel dans l'expression de l'impédance d'interface, cependant au cours de la résolution, et plus précisément de la construction du problème grossier BDD, une quantité très proche est déjà calculée et factorisée : la matrice $\tilde{G}_A^T S_A \tilde{G}_A$ – avec $\tilde{G}_A \equiv [\dots, \tilde{A}^{(s)} R_b^{(s)}, \dots]$, et $S_A = \sum_{s=1}^{N_s} A^{(s)} S^{(s)} A^{(s)T}$. Pour des structures de grande taille, comprenant un nombre de sous-domaines important, le rang $m^{(\bar{j})}$ de la matrice $\hat{G}_A^{(\bar{j})T} S_A^{(\bar{j})} \hat{G}_A^{(\bar{j})}$ peut augmenter drastiquement, et faire l'économie de son calcul ainsi que de sa factorisation en vue de son inversion peut alors représenter un gain d'opérations et de temps non négligeable. En prenant en considération le fait que l'ajout du terme local lié à $\Omega^{(j)}$ dans l'opérateur $\tilde{G}_A^T S_A \tilde{G}_A$ (comparé à l'expression $\hat{G}_A^{(\bar{j})T} S_A^{(\bar{j})} \hat{G}_A^{(\bar{j})}$) est en quelque sorte contrebalancé par la pondération classique $\tilde{A}^{(s)}$ sur le bord du sous-domaine $\Omega^{(j)}$, on fait l'hypothèse :

$$\hat{F}_{A,2}^{(j)} \simeq A^{(j)T} \hat{G}_A^{(\bar{j})} (\tilde{G}_A^T S_A \tilde{G}_A)^{-1} \hat{G}_A^{(\bar{j})T} A^{(j)} \equiv \tilde{F}_{A,2}^{(j)}$$

Interactions proches Le calcul du premier terme $\hat{F}_{A,1}^{(j)}$ de l'expression (3.10) :

$$\hat{F}_{A,1}^{(j)} \equiv A^{(j)T} P_A^{(\bar{j})} \left(\sum_{s \neq j} \hat{A}_j^{(s)} S_t^{(s)\dagger} \hat{A}_j^{(s)T} \right) P_A^{(\bar{j})T} A^{(j)}$$

nécessite autant d'opérations que l'étape de préconditionnement d'un système BDD de taille équivalente. En vue d'une efficacité numérique accrue, l'une des techniques d'approximation du complément de Schur présentées à la section 1.2 de ce chapitre peut être utilisée : la méthode du *lumping* – la moins coûteuse d'entre elles – sera ici choisie pour notre nouvelle heuristique. Bien que la qualité de l'approximation obtenue soit également la plus faible, les résultats qui seront présentés à la section 3 de ce chapitre nous semblent déjà assez prometteurs, au regard du coût de calcul très faible de cette nouvelle impédance d'interface, pour envisager dans le futur une implémentation à plus grande échelle. D'autre part, la section 3.5 de ce chapitre présentera une tentative d'enrichissement de cette approximation simple, dont les résultats peu probants nous ont conduits à conserver la technique de *lumping*, basique mais efficace. Afin de s'affranchir aisément des problèmes de densité liés à l'inversion, on utilisera en particulier la technique du *lumping* diagonal (ou *super-lumping*, voir expression (3.3)).

$$S_t^{(s)} \simeq \text{diag} \left(K_{t_{bb}}^{(s)} \right)$$

De plus, afin de conserver la structure creuse de l'assemblage des opérateurs *lumpés*, on se défait des projecteurs $P_A^{(\bar{j})}$, présents dans l'expression du préconditionneur BDD

uniquement pour traiter la non-inversibilité de $K_t^{(s)}$:

$$\begin{aligned}\hat{F}_{A,1}^{(j)} &\simeq A^{(j)T} \left(\sum_{s \in \text{neigh}(j)} \hat{A}_j^{(s)} \text{diag} \left(K_{t_{bb}}^{(s)} \right)^{-1} \hat{A}_j^{(s)T} \right) A^{(j)} \\ &= A^{(j)T} \left(\sum_{s \in \text{neigh}(j)} A^{(s)} \text{diag} \left(K_{t_{bb}}^{(s)} \right) A^{(s)T} \right)^{-1} A^{(j)T} = K_{t_{bb},sl}^{\text{neigh}(j)-1}\end{aligned}$$

Grâce à la diagonalité de l'opérateur *lumpé*, on retrouve ainsi finalement l'inverse de l'opérateur classique $K_{t_{bb},sl}^{\text{neigh}(j)}$ (3.3).

Opérateurs d'assemblage pondéré Enfin, on peut s'affranchir de l'étape de construction des opérateurs $\hat{A}_j^{(s)}$ en remarquant qu'une relation simple existe entre les opérateurs classiques $\tilde{A}^{(s)}$ [Klawonn and Widlund, 2001b] et modifiés :

$$\begin{aligned}\forall s \neq j, \quad \hat{A}_j^{(s)} &= \tilde{D}_j \tilde{A}^{(s)} \\ \text{avec } \tilde{D}_j &\equiv \left(A^{\diamond} \Delta^{\diamond} A^{\diamond T} \right) \left(A^{\diamond} \Delta^{\diamond} A^{\diamond T} - A^{(j)} \Delta^{(j)} A^{(j)T} \right)^{-1}\end{aligned}$$

On peut alors localiser l'effet de l'absence de matière au sein du sous-domaine $\Omega^{(j)}$ sur le scaling :

$$\forall s \neq j, \quad A^{(j)T} \hat{A}_j^{(s)} = A^{(j)T} \tilde{D}_j \tilde{A}^{(s)} = \underbrace{A^{(j)T} \tilde{D}_j A^{(j)}}_{\tilde{D}^{(j)}} A^{(j)T} \tilde{A}^{(s)}$$

La matrice locale $\tilde{D}^{(j)}$ peut être aisément construite à partir de $\tilde{A}^{(j)}$, à moindre coût :

$$\tilde{D}^{(j)} = A^{(j)T} \left(I - A^{(j)} \tilde{A}^{(j)T} \right)^{-1} A^{(j)}$$

On a alors :

$$\tilde{F}_{A,2}^{(j)} = \tilde{D}^{(j)} A^{(j)T} \tilde{G}_A^{(j)} \left(\tilde{G}_A^T S_A \tilde{G}_A \right)^{-1} \tilde{G}_A^T A^{(j)} \tilde{D}^{(j)}$$

Remarque 3 : Une définition plus explicite encore de l'opérateur $\tilde{D}^{(j)}$ peut être donnée, par exemple pour une pondération basé sur les matrices de rigidité locales (i.e. $\Delta^{(s)} = \text{diag} \left(K_{t_{bb}}^{(s)} \right)$), le coefficient diagonal de $\tilde{D}^{(j)}$ correspondant au degré de liberté x de la sous-structure peut être exprimé comme :

$$\tilde{D}_{xx}^{(j)} = \frac{\sum_s K_{t_{xx}}^{(s)}}{\sum_{s \neq j} K_{t_{xx}}^{(s)}} = \left(1 - \frac{K_{t_{xx}}^{(j)}}{\sum_s K_{t_{xx}}^{(s)}} \right)^{-1}$$

En effet, on a :

$$\tilde{A}_x^{(s)} = \frac{K_{t_{xx}}^{(s)}}{\sum_s K_{t_{xx}}^{(s)}} , \quad \hat{A}_{j,x}^{(s)} = \frac{K_{t_{xx}}^{(s)}}{\sum_{s \neq j} K_{t_{xx}}^{(s)}} (\forall s \neq j) , \quad \tilde{D}_{xx}^{(j)} = \left(1 - \tilde{A}_x^{(j)}\right)^{-1}$$

□

Expression finale On retrouve ainsi finalement la structure de (3.4), permettant l'application de Sherman-Morrison et une résolution à moindre coût (voir 2.1 de ce chapitre). Nous proposons donc la nouvelle souplesse d'interface à deux-échelles :

$$\left(Q_{b,2s}^{(j)}\right)^{-1} \equiv K_{t_{bb},sl}^{\text{neigh}(j)-1} + \tilde{D}^{(j)} A^{(j)T} \tilde{G}_A^{(\bar{j})} \left(\tilde{G}_A^T S_A \tilde{G}_A\right)^{-1} \tilde{G}_A^{(\bar{j})T} A^{(j)} \tilde{D}^{(j)} \quad (3.11)$$

3 Résultats

3.1 Deux cas-tests

Les performances de la nouvelle expression (3.11) sont ici évaluées, sur la base de deux cas-tests numériques. Le premier est une poutre bi-matériaux soumise à un chargement de flexion, représentée sur la figure 3.2. Le second est une poutre homogène multiperforée, également sous sollicitation de flexion, et représentée sur la figure 3.3. Les paramètres matériau et géométriques choisis pour ces deux cas-tests sont répertoriés dans le tableau 3.1 : pour le premier cas-tests, le matériau de faible module d'Young est choisi élastique, tandis que le comportement du matériau à module d'Young plus important – avec un ratio de presque 10^4 entre les deux – est élasto-plastique avec écrouissage isotrope linéaire. Concernant le deuxième cas-tests, l'unique matériau de la poutre homogène est choisi élasto-plastique avec écrouissage isotrope linéaire. Le chargement est, dans les deux cas, appliqué par déplacement imposé en $x = L$.

3.2 Analyse élastique

Une analyse préliminaire élastique permet de comparer une expression de l'impédance d'interface donnée à la valeur optimale de référence, simple à exprimer dans ce cadre : $Q_{b,lin,opti}^{(j)} = S_l^{(\bar{j})}$ (voir section 1.1.1 de ce chapitre). Bien que le coût de calcul de ce paramètre soit, en situation réelle, bien trop élevé dans le contexte de résolutions parallèles, sa construction sera effectuée ici, afin de pouvoir comparer les différents résultats. Les sollicitations imposées seront choisies assez faibles pour que chaque matériau reste dans son domaine élastique : pour les poutres bi-matériau ainsi que multiperforée, l'intensité du déplacement imposé de flexion sera fixée à $u_D = 1.5 * 10^{-3}$.

Les deux impédances suivantes seront dans cette section comparées à la valeur de référence :

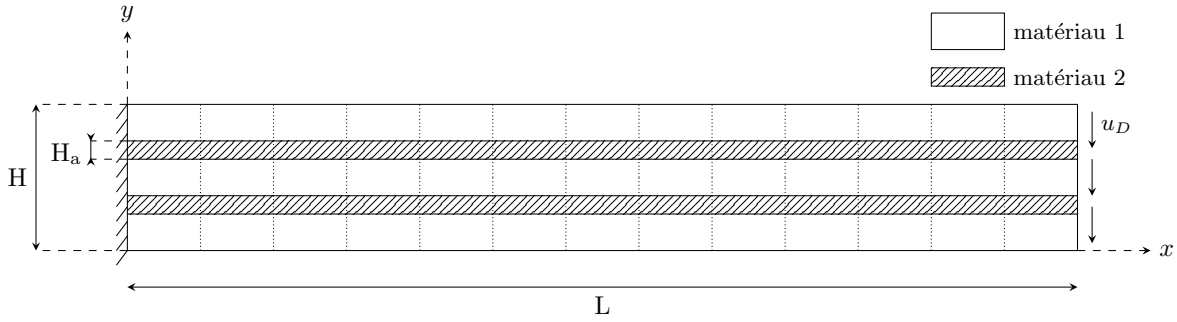


FIGURE 3.2 – Poutre bi-matériaux

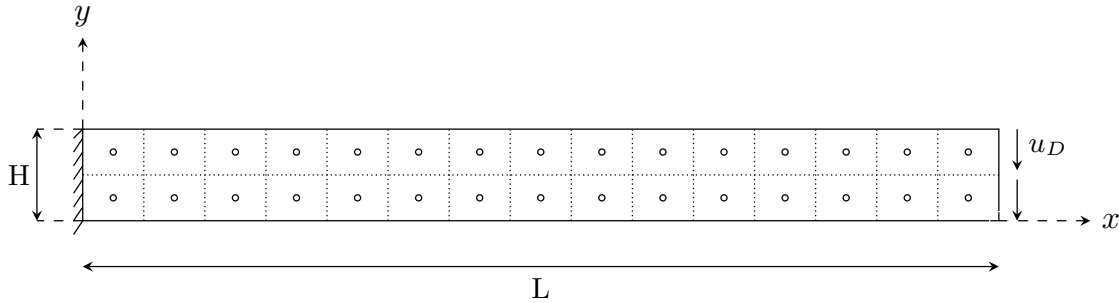


FIGURE 3.3 – Poutre multiperforée

- $K_{bb,l}^{\text{neigh}(j)}$: voir (3.2)
- $Q_{b,2s}^{(j)}$: voir (3.11)

On rappelle ici que la formulation alternative choisie pour l'approche mixte de la méthode de sous-structuration et condensation non linéaires (voir 4.3.4 et algorithme 11, chapitre 2) intervertit les solveurs linéaires FETI-2LM et BDD : la résolution d'un problème linéaire par le biais de cette méthode serait ainsi totalement équivalente à celle d'une approche primale : au cours de la première – et seule en linéaire – itération, la matrice d'impédance n'intervient pas ; les expressions (3.2), (3.11) et (3.1) ne peuvent ainsi être comparées par le biais de l'algorithme 11. Conséquemment à cette observation, un solveur FETI-2LM a été implémenté, afin de pouvoir résoudre le problème sous sa forme initiale (2.24) – dans le seul but d'effectuer une étude de performance des différentes impédances d'interface sur un problème linéaire. Ce solveur linéaire mixte permet de résoudre des problèmes en parallèle, avec conditions d'interface de Robin, cependant il ne dispose pas d'un problème grossier naturel : une stratégie d'augmentation serait nécessaire afin d'améliorer son extensibilité. Toutefois, dans le cas d'une étude de la qualité des conditions de transmission inter-sous-structures, la dépendance critique des performances au nombre de sous-domaines ne

Poutre bi-matériaux			Poutre multiperforée	
Paramètres matériau			Paramètres matériau	
Young	Matériau 1 $E_1 = 420e2$	Matériau 2 $E_2 = 210e6$	Young	$E = 210e6$
Coeff. Poisson	$\nu_1 = 0.3$	$\nu_2 = 0.3$	Coeff. Poisson	$\nu = 0.3$
Limite élastique		$\sigma_{0_2} = 420e3$	Limite élastique	$\sigma_0 = 420e3$
Coeff. écrouissage		$h_2 = 1e3$	Coeff. écrouissage	$h = 1e6$
Paramètres géométriques			Paramètres géométriques	
Longueur totale	$L = 13$		Longueur	$L = 10$
Hauteur totale	$H = 2$		Hauteur	$H = 1$
Hauteur d'armature	$H_a = 0.25$		Rayon	$r = 2/30$

TABLE 3.1 – Paramètres géométriques et matériau

fera a priori que renforcer les différences, sans les dénaturer.

#SD	FETI-2LM #iterations			Gain (%)
	S	K	2S	2S vs. K
3	2	1347	27	98
4	3	1265	38	97
7	6	3697	84	98
13	12	4393	342	92

(a) Poutre bi-matériaux

#SD	FETI-2LM #iterations			Gain (%)
	S	K	2S	2S vs. K
3	2	79	20	77
4	3	382	23	94
8	7	430	40	91
15	14	1265	113	91

(b) Poutre multiperforée

TABLE 3.2 – Comparaison des trois impédances d'interface : comportement linéaire

Le tableau 3.2 récapitule les résultats des trois impédances d'interface sur les deux cas-tests présentés à la section précédente 3.1. Pour des raisons de clarté, les notations simplifiées S , K et $2S$ sont utilisées pour les impédances $Q_{b,lin,opti}^{(j)}$, $K_{tbb,l}^{neigh(j)}$ et $Q_{b,2s}^{(j)}$ respectivement. La décomposition en sous-domaines est pour l'heure effectuée uniquement selon l'axe x (voir figures 3.2 et 3.3).

Conformément à la théorie, dans le cas de l'emploi de l'impédance optimale linéaire S , le processus de résolution ne nécessite qu'un nombre d'itérations égal au nombre de sous-domaines moins un pour atteindre la solution. Étant donnée la répartition des sous-domaines (sans points multiples) ainsi que l'absence de problème grossier, ce taux de convergence est le meilleur que l'on puisse obtenir (avec un opérateur $Q_b^{\diamond}$ matriciel, toutefois l'usage d'un opérateur affine permettrait de s'affranchir des itérations de propagation du second membre, voir 1.2.9, chapitre 1) : les conditions de transmission obtenues avec l'impédance d'interface S sont bien optimales.

Le choix classique K ne donne aucune information, dans les équilibres locaux, sur les interactions d'un sous-domaine avec la structure lointaine, et le nombre d'itérations augmente en conséquence drastiquement avec le nombre de sous-structures.

La nouvelle expression (3.11) à deux échelles, comparée au choix classique K , permet de réduire massivement le nombre d'itérations du solveur FETI-2LM : les gains varient de 77 à 98% sur l'ensemble de la résolution. La structure additive de $2S$ semble donc bien permettre l'introduction d'information globale dans la réponse de la structure aux sollicitations locales. Bien entendu, l'absence de problème grossier ou de stratégie d'augmentation du solveur renforce les bénéfices du terme représentant les interactions lointaines dans l'expression (3.11) : à la section suivante, cette dernière sera remplacée dans le contexte de résolutions multi-échelle.

Les performances de l'impédance $2S$ n'équivalent évidemment pas celles de l'impédance S , cependant l'augmentation constatée du nombre d'itérations de FETI-2LM n'est que d'environ dix fois leur valeur optimale, tandis qu'elle est de cent à un millier de fois la valeur optimale pour le choix K . On rappelle également que l'impédance d'interface S ne peut en pratique pas être calculée dans le contexte parallèle : l'expression $2S$, au contraire, est très peu chère et simple à évaluer.

3.3 Analyse plastique

L'analyse des performances de l'expression à deux échelles (3.11) est complétée par une étude en plasticité. Les deux cas-tests sont soumis à un chargement de flexion par déplacement imposé, appliqué de façon incrémentale. Le chargement de la poutre bi-matériaux se décompose en onze incréments :

$$u_D = [0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.375, 0.4, 0.425, 0.45] u_{max}$$

$$u_{max} = 7.1$$

La poutre multiperforée quant à elle, est soumise au chargement incrémental suivant :

$$u_D = [0.4, 0.6, 0.8, 1, 1.15, 1.3, 1.45, 1.5] u_{max}$$

$$u_{max} = 0.275$$

Remarque 4 : Afin de ne pas surcharger inutilement les tableaux de résultats, un incrément sur deux seulement sera représenté par la suite.

La poutre bi-matériaux est découpée en 13 sous-domaines, le long de l'axe x , tandis que la sous-structuration de la poutre multiperforée fait intervenir 30 sous-domaines avec points multiples (voir figures 3.2 et 3.3).

Les nombres d'itérations du solveur de Krylov, cumulées sur les boucles de Newton globales ainsi que les incréments de chargement, sont répertoriées pour les trois impédances – S , K et $2S$, – et les deux cas-tests dans les tableaux 3.3 et 3.4. Le choix d'un travail sur les itérations de Krylov est motivé par le lien fort existant entre elles et la quantité de communications du processus de résolution (voir à ce propos la section 5.1 du chapitre 2, où une explication plus détaillée de ce lien est donnée), que l'on cherche à minimiser grâce à une augmentation de la part des opérations locales.

	Krylov						Global Newton
load inc.	0.05	0.15	0.25	0.35	0.4	0.45	0.45
S	37	229	x				x
K	36	259	598	978	1357	1772	47
2S	37	266	580	891	1204	1514	39
C	74	296	633	970	1344	1795	48

	Gains (%)						
2S vs. K	-3	-3	3	9	11	15	17
2S vs. C	50	10	8	8	10	16	19

TABLE 3.3 – Poutre bi-matériaux : itérations de Krylov cumulées sur les incréments de chargement, itérations de Newton globales cumulées

Le calcul des opérateurs tangents locaux, à chaque itération globale, est également une opération coûteuse qu'il est intéressant de limiter. Les nombres d'itérations du solveur de Newton global, cumulées sur les incréments de chargement, sont également répertoriés dans les tableaux 3.3 et 3.4, pour les deux cas-tests et les trois impédances d'interface – toutefois, seule la valeur des itérations de Newton globales cumulées au dernier incrément de chargement sera donnée, dans le souci d'éviter la redondance d'information, puisque dans ce cas précis, la diminution des itérations de Krylov liée à une meilleure impédance est fortement corrélée (quasiment proportionnelle) à une diminution des itérations de Newton globales.

Une quatrième approche est ajoutée à l'étude, notée C dans les deux tableaux de résultats, et correspondant au processus de résolution "classique" en mécanique non linéaire des structures : un algorithme de Newton pour le problème global non linéaire, combiné à un solveur DD pour les problèmes tangents, dans le cas d'un grand nombre de degrés de liberté (voir section 2, chapitre 2). La différence principale entre

Krylov					Global Newton
load inc.	0.6	1	1.3	1.5	1.5
S	48	172	322	481	30
K	61	212	373	548	35
2S	48	170	300	438	28
C	73	222	385	561	36

Gains (%)					
2S vs. K	21	20	20	20	20
2S vs. S	0	1	7	9	7
2S vs. C	34	23	22	22	22

TABLE 3.4 – Poutre multiperforée : itérations de Krylov cumulées sur les incréments de chargement, itérations de Newton globales cumulées

cette méthode et le processus de sous-structuration et condensation non linéaires réside dans le caractère linéaire/non linéaire des algorithmes utilisés pour la résolution des équilibres locaux. La comparaison en résultant avec les approches S , K et $2S$ représente ainsi le gain total pouvant être obtenu grâce à la méthode de sous-structuration et condensation non linéaires avec approche mixte, dans le contexte plus général des solveurs non linéaires.

Une première observation peut être établie en comparant les résultats obtenus avec l'impédance S dans le contexte non linéaire et ceux du contexte linéaire de la section précédente : si en non linéaire on ne pensait pouvoir, au début de ce chapitre, définir analytiquement de meilleure impédance d'interface que le complément de Schur S , ce dernier perd bien le caractère optimal que lui conférait le contexte linéaire. Pour la poutre bi-matériaux par exemple, des divergences au niveau des algorithmes de Newton locaux apparaissent dès les premiers incréments de chargements et conduisent à la divergence globale de l'algorithme. Ces instabilités locales sont causées par des niveaux anormalement hauts de plasticité cumulée au sein des sous-domaines, artefacts de la résolution – potentiellement causés par une impédance d'interface trop souple, permettant au matériau de se déformer plus que nécessaire.

Deuxièmement, bien que notre première hypothèse consistant à utiliser pour exprimer l'impédance d'interface des approximations de S puisse alors être remise en question, l'expression additive $2S$ ainsi dérivée semble se comporter de façon très satisfaisante, et même être plus performante que S (en sus d'offrir l'avantage d'un coût de calcul très réduit, comparé à celui de S). Les gains, en termes d'itérations de Krylov cumulées, par comparaison avec l'impédance d'interface classique K , varient de 15 à 20% à la fin de la résolution : une réduction qui devrait permettre une décroissance non négligeable du temps CPU, surtout pour des problèmes de grande taille où les communications entre processeurs peuvent devenir extrêmement chronophages. La

comparaison avec l'impédance d'interface S ne peut être effectuée que sur le cas de la poutre multiperforée, étant donnée la non convergence de la méthode avec l'impédance S pour la poutre bi-matériaux : les gains, pour l'approche $2S$, atteignent 9% à la fin de la résolution – en termes d'itérations de Krylov cumulées. Cette valeur est assez faible, mais confirme l'idée que la nouvelle impédance $2S$ est finalement meilleure – à la fois plus stable et légèrement plus rapide – que l'optimum tangent, ce dernier étant de plus, comme précédemment mentionné, d'un coût de calcul trop élevé dans le contexte parallèle.

Remarque 5 : Le maillage de la poutre bi-matériaux comprend 25 789 degrés de liberté, et l'interface globale générée par sa sous-structuration en 13 sous-domaines en comprend 984. Le maillage de la poutre multiperforée, quant à lui, comprend 30 515 degrés de liberté, et sa sous-structuration en 30 sous-domaines génère une interface de 1641 degrés de liberté. Malgré la taille limitée de ces cas-tests, on peut espérer une certaine représentativité de calculs à plus grande échelle, notamment grâce au choix des nombres d'itérations de Krylov et de Newton en tant que critère de comparaison. Malheureusement, le code réalisé pour obtenir ces résultats est basé sur l'interpréteur Octave, dont la limite, en termes de nombre d'inconnues, ne nous a pas permis d'être plus pertinents du point de vue de la taille des problèmes testés. Une étude en termes de temps de calcul ne nous aurait pas non plus semblé pertinente dans ce contexte, pour deux raisons : la non optimisation de la partie MPI du code n'aurait pas été à l'avantage d'une diminution des communications, de même que les problèmes d'interface de petite taille, générant des communications all-to-all trop rapides.

La comparaison avec la méthode classique donne des résultats similaires pour les deux cas-tests : à la fin de la résolution, les gains varient de 16 à 22% pour les itérations cumulées de Krylov, et de 19 à 22% pour les itérations cumulées de Newton globales. Ce gain correspond à la performance globale de la méthode de sous-structuration et condensation non linéaires pouvant être obtenue avec l'approche mixte, comparée aux procédures de résolution classiques.

Remarque 6 : Les performances plutôt limitées des méthodes de sous-structuration et condensation non linéaires mixtes avec l'impédance d'interface K , comparées à la méthode classique de résolution, peuvent probablement être imputées à la difficulté de donner une bonne représentation des phénomènes lointains avec une impédance d'interface privilégiant les interactions proches ; elles devraient en effet prévaloir dans le cas d'hétérogénéités locales (poutre bi-matériaux) ou de structures élancées (poutre multiperforée).

3.4 Couplage avec la méthode SRKS

Un post-processing des solveurs de Krylov permet l'extraction des vecteurs et valeurs de Ritz à la fin de chaque résolution tangente, et une stratégie d'augmentation des sous-espaces de Krylov peut en être dérivée, en les stockant et les réutilisant pour construire une base d'augmentation à chaque itération globale non linéaire. La méthode ainsi définie, dénommée TRKS [Gosselet et al., 2013], réutilise tous les vecteurs de Ritz produits ; une sélection des "meilleures" valeurs (i.e. les valeurs de Ritz approchant le mieux les valeurs propres de l'opérateur du problème tangent) permet de ne conserver que l'information la plus pertinente (méthode SRKS [Gosselet et al., 2013]). Le couplage de la méthode SRKS à la méthode de sous-structuration et condensation non linéaires est ici étudié pour les deux cas-tests définis à la section 3.1. Les résultats sont répertoriés dans les tableaux 3.5 et 3.6.

Krylov							
	avec SRKS						sans SRKS
load inc.	0.05	0.15	0.25	0.35	0.4	0.45	0.45
S	37	97	x				x
K	36	113	218	339	452	578	1772
2S	37	98	197	304	398	492	1514
C	53	130	245	370	492	648	1795

Gains (%)							
2S vs. K	-3	13	10	10	12	15	15
2S vs. C	30	25	20	18	19	24	16

TABLE 3.5 – Poutre bi-matériaux, couplage avec SRKS : itérations de Krylov cumulées sur les incréments de chargement

Conformément à la théorie, la méthode SRKS entraîne une réduction globale des nombres d'itérations de Krylov, que l'on peut observer en comparant les colonnes "avec" et "sans" SRKS des deux tableaux de résultats. Pour la poutre bi-matériaux, les itérations de Krylov sont réduites en moyenne de 67% au dernier incrément de chargement ; cette réduction n'est que de l'ordre de 8% pour la poutre multiperforée (cette différence entre les deux cas-tests peut probablement s'expliquer en partie par le bien plus faible nombre d'itérations de Krylov requis pour le deuxième, ne permettant la génération que de peu de vecteurs de Ritz).

Les nombres d'itérations cumulées de Newton globales ne variant pas suivant que la méthode est ou non couplée avec SRKS – conformément à la théorie, – ils ne sont pas représentés à nouveau dans cette section.

Les tableaux 3.5 et 3.6 confirment les conclusions de la section précédente. En effet, bien que les itérations de Krylov des deux cas-tests diminuent à cause de la procédure SRKS, les gains générés par la nouvelle impédance d'interface 2S restent

Krylov					
	avec SRKS				sans SRKS
load inc.	0.6	1	1.3	1.5	1.5
S	48	164	304	445	481
K	61	201	351	506	548
2S	47	159	280	410	438
C	72	212	362	517	561

Gains (%)					
2S vs. K	20	20	20	20	20
2S vs. S	2	3	8	8	9
2S vs. C	35	25	23	21	22

TABLE 3.6 – Poutre multiperforée, couplage avec SRKS : itérations de Krylov cumulées sur les incréments de chargement

similaires lorsqu'on introduit ce couplage. Pour la poutre bi-matériaux, la réduction massive d'itérations de Krylov liée à la procédure SRKS ne dégrade pas les performances de 2S, voire même permet une amélioration des performances de l'expression N , comparée à la méthode classique (cette dernière souffre en effet sur ce cas-test d'une détérioration de ses performances liée au couplage avec SRKS, et les gains de l'expression N passent de 16 à 24%). Concernant la poutre multiperforée, au contraire de la méthode SRKS, l'impédance 2S est également performante.

3.5 Enrichissement de l'approximation de rang faible

La partie représentative des interactions *courte distance* dans l'expression (3.11) approxime – comme on l'a fait précédemment remarquer – de façon très grossière le terme $\hat{F}_{A,1}^{(j)}$ introduit à la section précédente. En particulier, la restriction des pseudo-inverses à des opérateurs diagonaux (ou même bloc-diagonaux si on avait retenu le *lumping* simple (3.2)) conduit probablement à une surestimation de la rigidité des voisins. Un procédé d'enrichissement de cette approximation a été développé suite à ces considérations, cependant les résultats ne s'en sont pas ou peu ressentis, ne justifiant pas le coût numérique plus élevé de l'impédance d'interface résultante ; aussi, seule une description succincte en sera donnée dans cette section.

La tentative a consisté en l'ajout d'un second terme de rang faible à l'approximation creuse du terme d'interactions proches :

$$\hat{F}_{A,1}^{(j)} \simeq K_{t_{bb}}^{\text{neigh}(j)-1} + \tilde{D}^{(j)} A^{(j)T} V_k \Theta_k V_k^T A^{(j)} \tilde{D}^{(j)} \equiv \tilde{F}_{A,\text{ritz}}^{(j)}$$

où la matrice diagonale Θ_k et la base orthonormale de vecteurs d'interface V_k sont des approximations des éléments propres de $S_A^{(\bar{j})-1}$. Ces éléments peuvent être obtenus

à moindre coût à partir d'un post-processing du solveur BDD, dans l'esprit des méthodes SRKS [Gosselet et al., 2013] – voir l'annexe A pour une démonstration plus détaillée.

Ce terme de rang faible peut être concaténé avec le terme associé aux modes rigides de la nouvelle expression du paramètre $Q_b^\diamond$ (i.e. l'approximation de $\tilde{F}_{A,2}^{(j)}$ retenue)², et ne modifie ainsi en rien la commodité de la structure additive de (3.11). La figure 3.4 présente une comparaison des spectres (ordonnés par valeur propre croissante) des trois opérateurs suivants :

- $S_t^{(\bar{j})}$, $K_{t_{bb}}^{\text{neigh}(j)}$ et $Q_{b,2s}^{(j)}$, notés S , K et $2S$ respectivement
- $Q_{b,\text{ritz}}^{(j)} \equiv \tilde{F}_{A,\text{ritz}}^{(j)} + \tilde{F}_{A,2}^{(j)}$, noté R

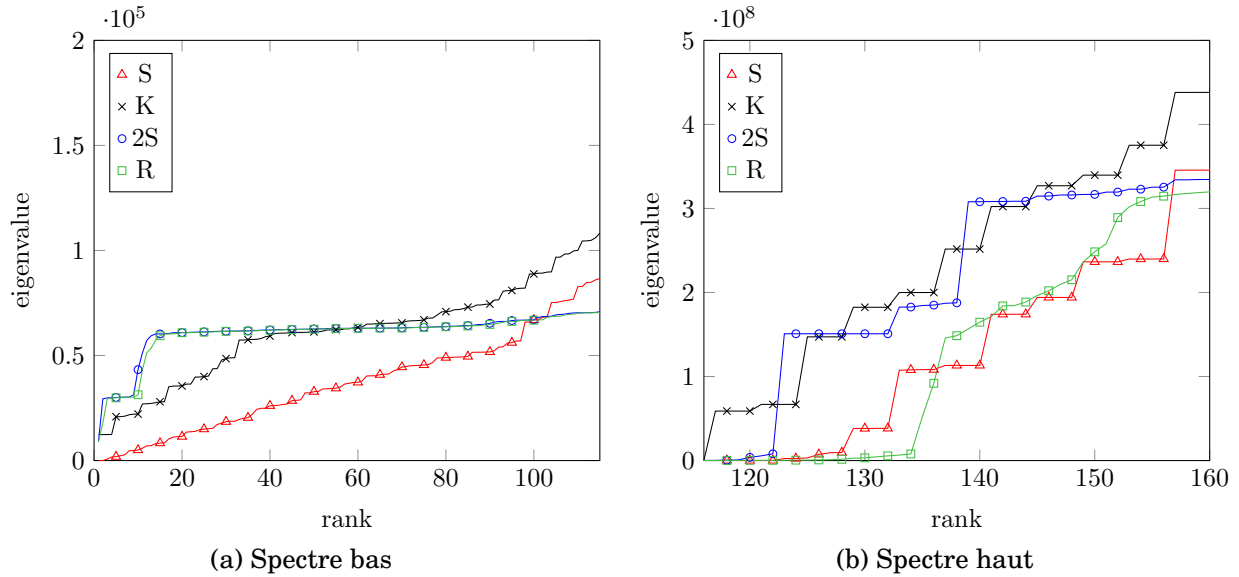


FIGURE 3.4 – Spectres pour une poutre bi-matériau

De manière générale, le spectre de l'impédance d'interface avec enrichissement du terme d'interactions proches est plus proche de celui de S que celui de $2S$. Cependant, on a vu à la section de résultats que l'impédance S n'était pas optimale, et de plus les faibles améliorations des performances de la méthode nous ont conduits à abandonner cette piste.

2. On retrouve ainsi l'esprit des techniques d'augmentation GenEO

4 Conclusion

Une nouvelle approximation de l'impédance d'interface a été développée, dans le contexte des méthodes de sous-structuration et condensation non linéaires avec approche mixte. L'expression de l'impédance d'interface introduite ici permet de coupler, grâce à sa structure additive, les interactions proches et lointaines d'une sous-structure donnée avec le reste du domaine.

La procédure de construction d'un tel paramètre consiste à évaluer, pour un sous-domaine donné, le complément de Schur tangent de son complémentaire dans la structure globale (i.e. la valeur optimale dans un contexte linéaire), initialement la meilleure valeur analytique que l'on pouvait exprimer pour évaluer l'impédance d'interface dans le contexte non linéaire. Cette évaluation met en jeu un terme de champ proche, assimilable à la rigidité des sous-domaines voisins, ainsi qu'un terme de champ lointain, composé de la projection du complément de Schur tangent dans l'espace généré par les mouvements de corps rigides d'interface, capturant ainsi les interactions globales avec la structure éloignée.

L'étude des nombres d'itérations d'un solveur FETI-2LM a permis, dans un cas linéaire, de comparer les performances de cette impédance avec une valeur classique en mécanique des structures - i.e. la rigidité de bord des voisins - ainsi qu'avec le complément de Schur du complémentaire - qui dans ce contexte linéaire est la valeur optimale, et a été calculée ici spécialement pour les besoins de l'étude, malgré son coût de construction inabordable en pratique dans le contexte parallèle. Bien que, conformément à la théorie, la nouvelle expression de l'impédance d'interface n'ait pas donné d'aussi bons résultats que la valeur optimale, elle a permis d'atteindre des gains très élevés, comparée au choix classique.

La performance de cette nouvelle impédance a également été étudiée dans le cadre de la méthode mixte de sous-structuration et condensation non linéaires, sur deux problèmes de plasticité. La conclusion principale de cette seconde étude semble être que non seulement l'impédance linéaire optimale ne l'est plus du tout dans le contexte non linéaire, mais de plus la nouvelle expression - dont le coût de calcul est extrêmement faible en parallèle - permet d'obtenir de meilleurs résultats. Ceci suggère que le niveau de précision atteint par l'impédance d'interface quant à la représentation de l'environnement d'une sous-structure s'est trouvé accru sur les cas-tests considérés.

Pour terminer, l'étude d'un couplage avec une procédure sélective de réutilisation des éléments de Ritz, obtenus par un post-processing peu coûteux du solveur de Krylov, vient confirmer la tendance globale du comportement de l'impédance d'interface à deux échelles, alors même que les nombres d'itérations de Krylov sont réduits. Ces considérations laissent à penser qu'une implémentation à plus grande échelle pourrait s'avérer intéressante.

Préconditionneur non linéaire

Les chapitres précédents ont permis de commencer à répondre à l'un des objectifs sous-tendant le présent manuscrit : l'amélioration de la robustesse de la méthode de sous-structuration et condensation non linéaires. Dans la même optique, le chapitre à venir se concentre sur l'étape de préconditionnement de ce processus de résolution, jusqu'à présent impliquée au niveau de la résolution des problèmes tangents (linéarisés) par une méthode DD. Nous en construirons ici une variante non linéaire qui permettra, grâce à la définition d'un « meilleur » problème d'interface non linéaire condensé, dont l'opérateur tangent sera intrinsèquement bien conditionné, de diminuer le nombre d'itérations globales nécessaires à convergence – et ainsi les communications entre processeurs au cours de la résolution (i.e. les itérations du solveur tangent de Krylov). Le coût de cette décroissance en termes de communications se comptera, comme précédemment, en opérations locales additionnelles, matérialisées par des boucles de Newton locales supplémentaires en parallèle ; le postulat initial à l'origine de la méthode de sous-structuration et condensation non linéaires elle-même, selon lequel une diminution des communications au sein d'un solveur parallèle justifie l'ajout d'un nombre raisonnable d'opérations locales – dans le cas d'un grand nombre de sous-domaines, – est ainsi à nouveau sollicité et mis en application.

Ce chapitre a fait l'objet de la rédaction d'un article (en préparation).

1 Introduction

La méthode de sous-structuration et condensation non linéaires est basée sur un principe simple : dans le cadre d'un solveur parallèle et d'un problème découpé en un grand nombre de sous-domaines, les communications entre processeurs de type *all-to-all* sont très coûteuses ; une stratégie visant à les réduire par une augmentation des opérations locales indépendantes est ainsi à privilégier. Le détail de cette méthode, donné notamment au chapitre 2, a permis de faire état, par comparaison avec les méthodes plus classiques dites NKS (voir section 3.1, chapitre 1), de la possibilité de construire un « meilleur » problème condensé à l'interface, grâce à des ré-

solutions locales rendues non linéaires, et ainsi de converger en un nombre réduit d'itérations globales, diminuant de la sorte le nombre d'itérations de Krylov nécessaires. L'augmentation « raisonnable » du nombre d'itérations locales est considérée, pour des problèmes de grande taille, comme moins problématique que celle des itérations tangent, qui concentrent les plus coûteuses des opérations de communication du solveur (réductions).

Une stratégie similaire peut être appliquée à l'étape du préconditionnement de la méthode de sous-structuration et condensation non linéaires. De manière générale, que l'on se place dans le cas d'une étude linéaire ou non linéaire, les solveurs linéaires/tangents considérés dans notre étude (BDD et FETI) nécessitent l'emploi d'un préconditionneur à des fins de régularisation, et d'amélioration de leur extensibilité. Les préconditionneurs les plus classiques font intervenir un assemblage pondéré de pseudo-inverses locales, soit de compléments de Schur locaux primaux (resp. duaux) pour l'approche FETI (resp. BDD) [Klawonn and Widlund, 2001b]. Ces derniers sont quasi-optimaux pour un problème linéaire (ou tangent) donné, dans le sens où le conditionnement κ de l'opérateur ainsi préconditionné dépend du ratio entre la taille caractéristique d'un élément h et la taille d'un sous-domaine H , selon une loi de la forme :

$$\kappa = f\left(\log(H/h)^2\right)$$

Muni de ce préconditionneur dit de *Dirichlet* (resp. *Neumann*), la stabilité numérique de l'algorithme FETI est assurée (bien qu'en pratique son extensibilité parallèle dépende également de la capacité à calculer à moindre coût le problème grossier [Bhardwaj et al., 1999]). Dans le cas de structures hétérogènes notamment, une pondération basée sur les matrices de rigidité locales permet de s'affranchir d'un mauvais conditionnement du problème global – des études plus spécifiques ont été nécessaires pour traiter des hétérogénéités plus singulières, des cas incompressibles, des interfaces tordues ou des sous-domaines élancés [Gosselet et al., 2002, Hauret, 2004, Spillane, 2014, Bovet et al., 2017]... Bien que ce type de préconditionneur soit aisément calculable durant le processus de résolution linéaire/tangent – en effet, seules des résolutions locales sont requises, ainsi qu'un assemblage, – certaines variantes moins chères mais toutefois efficaces peuvent être utilisées afin d'augmenter l'efficacité en temps de calcul du solveur, tels que les préconditionneurs *lumpés* [Farhat, 1994, Farhat et al., 2000a], ou de manière plus générale des préconditionneurs dits *d'interface* [Gander and Zhang, 2014].

Dans le contexte non linéaire, de même que les solveurs NKS, la méthode de sous-structuration et condensation non linéaires ne tire pas entièrement avantage de la décomposition de domaine pour traiter la nonlinéarité, en n'appliquant le préconditionnement qu'au niveau du problème d'interface tangent. En effet, une stratégie de préconditionnement du problème non linéaire d'interface lui-même est possible – et sera explicitée dans ce chapitre, – qui par une augmentation des opérations locales de préconditionnement – alors rendues non linéaires – permettra de diminuer le nombre de communications nécessaires au sein du processus de résolution. À partir d'une in-

interprétation du système linéaire FETI (ou BDD) préconditionné comme un problème de point fixe [Rixen and Farhat, 1999], dont le pendant non linéaire peut aisément être extrapolé, le problème non linéaire condensé à l'interface de la méthode de sous-structuration et condensation non linéaires est régularisé par un preconditionneur supposé quasi-optimal (dans les cas suffisamment réguliers et homogènes), permettant de prendre en compte le caractère linéaire ou non linéaire du comportement de chaque sous-domaine, et ainsi de mieux représenter l'état de la structure. L'opérateur tangent de ce nouveau problème d'interface est alors intrinsèquement bien conditionné : dans le cadre des méthodes DD de Schwarz, une idée similaire est à l'origine du preconditionneur ASPIN [Cai and Keyes, 2002], et plus récemment du solveur RASPEN [Dolean et al., 2016] (voir section 3.2, chapitre 1). Bien que la philosophie du processus soit ainsi profondément modifiée, la stratégie globale de résolution reste toutefois inchangée, exceptées quelques résolutions non linéaires locales additionnelles, ainsi qu'un assemblage supplémentaire. Très peu de complications viennent ainsi perturber la technicité d'implémentation : un algorithme de Newton est toujours appliqué au problème d'interface non linéaire (préconditionné), tandis que les systèmes tangents ont toujours la forme des problèmes FETI. Seule une décroissance des nombres d'itérations de Newton globales (et ainsi des nombres d'itérations de Krylov, coûteuses en termes de communications) est à attendre de ces modifications légères, grâce à une régularisation du problème au niveau non linéaire, et à une augmentation de la part des opérations locales dans le processus de résolution.

La première section de ce chapitre introduira une interprétation des systèmes linéaires classiques préconditionnés FETI et BDD en termes d'un problème de recherche de point fixe d'un opérateur que l'on définira. Puis, la version non linéaire de cette interprétation en sera dérivée, et les modifications apportées à la stratégie de résolution seront détaillées. Les performances de ce nouveau preconditionneur non linéaire seront évaluées sur deux cas-tests numériques : un problème de diffusion d'eau dans les sols, ainsi qu'un problème d'évolution thermique non linéaire.

2 Une version non linéaire des preconditionneurs FETI/BDD

Cette section a pour objectif de définir et développer une nouvelle technique de preconditionnement, appliquée au problème non linéaire condensé à l'interface, de sorte que son opérateur tangent soit intrinsèquement bien conditionné. Un opérateur de type point fixe est ici construit à partir du problème non linéaire d'interface, assimilable à une version non linéaire du preconditionneur FETI/BDD. Les problèmes tangents sont résolus classiquement par un solveur de Krylov.

2.1 Cas linéaire : interprétation du système préconditionné FETI (resp. BDD) comme un problème de recherche de point fixe

L'étude du cas linéaire est intéressante en ce qu'elle inspire fortement la nouvelle version non linéaire du préconditionnement. Cette section donne la vision « point fixe » des préconditionneurs FETI et BDD, déjà présente dans [Rixen and Farhat, 1999], et sous-tendant les techniques de construction de champs admissibles en parallèle [Parret-Fréaud et al., 2009, Parret-Fréaud et al., 2016].

2.1.1 Approche duale

On rappelle ici l'expression du problème FETI projeté et préconditionné par l'opérateur pondéré de Dirichlet (voir section 1.2.6, chapitre 1) :

$$\tilde{B}^\diamond S_t^\diamond \tilde{B}^{\diamond T} P_B^T B^\diamond F_l^\diamond \left(B^{\diamond T} \lambda_B; f_{ext}^\diamond \right) = 0 \quad (4.1)$$

où $F_l^\diamond$ est l'opérateur discret *Neumann-to-Dirichlet* affine introduit en (2.12) :

$$\begin{aligned} F_l^\diamond \left(B^{\diamond T} \lambda_B; f_{ext}^\diamond \right) &= F_t^\diamond B^{\diamond T} \lambda_B - b_d^\diamond \\ \text{avec } b_d^\diamond &= -t^\diamond K^{\diamond \dagger} f_{ext}^\diamond = -F^\diamond b_p^\diamond \end{aligned}$$

et les opérateurs d'assemblage pondéré duaux $\tilde{B}^\diamond$ sont classiquement définis par (1.23).

Remarque 1 : La stratégie de projection mise en place pour traiter du problème de la semi-définition des opérateurs $F_t^\diamond$ est valable, en mécanique des structures, sous l'hypothèse des petites perturbations. On fera dans la suite de ce chapitre, de manière plus générale, l'hypothèse suivante sur l'opérateur $f_{int}^\diamond$: pour tout sous-domaine $\Omega^{(s)}$ non muni de conditions de Dirichlet (i.e. pour tout sous-domaine « flottant »), il existe une base $R^{(s)}$ de modes rigides satisfaisant :

$$\forall v_b^{(s)} \in \mathbb{R}^\diamond, \begin{cases} f_{int}^{(s)} \left(v_b^{(s)} + R^{(s)} \beta^{(s)} \right) = f_{int}^{(s)} \left(v_b^{(s)} \right), \quad \forall \beta^{(s)} \\ R^{(s)T} f_{int}^{(s)} \left(v_b^{(s)} \right) = 0 \end{cases} \quad (4.2)$$

La base $R^{(s)}$ des modes rigides est directement liée au noyau de l'opérateur tangent :

$$R^{(s)} = \text{Ker} \left(K_t^{(s)} \right), \quad \text{avec } K_t^{(s)} = \frac{\partial f_{int}^{(s)}}{\partial u^{(s)}}$$

La condition (4.2) est équivalente à l'hypothèse des petites déformations en mécanique des structures.

On justifie en général l'expression (4.1) par la qualité de l'approximation de l'opérateur FETI $(B^\diamond F_t^\diamond B^\diamond)^T$ obtenue par l'assemblage pondéré des pseudo-inverses locales $(\tilde{B}^\diamond S_t^\diamond \tilde{B}^\diamond)^T$, cependant une autre interprétation peut être exhibée. On notera dans la suite $u_b^\diamond$ le vecteur des déplacements d'interface solution des équilibres locaux :

$$u_b^\diamond = F_l^\diamond \left(B^\diamond{}^T \lambda_B; f_{ext}^\diamond \right) + R_b^\diamond \alpha^\diamond$$

où $\alpha^{(s)}$ est un vecteur de taille $d^{(s)}$, avec $d^{(s)}$ le nombre de modes rigides d'un sous-domaine donné $\Omega^{(s)}$. On peut construire, à partir de ce déplacement a priori non continu aux interfaces, une quantité continue en lui soustrayant sa discontinuité pondérée :

$$\underbrace{\hat{u}_b^\diamond}_{\text{continu}} = u_b^\diamond - \underbrace{\tilde{B}^\diamond{}^T B^\diamond u_b^\diamond}_{\text{discontinuité d'interface}}$$

En effet, on a :

$$B^\diamond \hat{u}_b^\diamond = \left[B^\diamond - B^\diamond \tilde{B}^\diamond{}^T B^\diamond \right] u_b^\diamond$$

donc par (1.24) on trouve $B^\diamond \hat{u}_b^\diamond = 0$.

La réaction nodale solution d'un équilibre local à conditions aux limites de type Dirichlet ayant pour valeur $\hat{u}_b^\diamond$ sera notée $\hat{\lambda}_b^\diamond$:

$$\hat{\lambda}_b^\diamond = S_l^\diamond (\hat{u}_b^\diamond; f_{ext}^\diamond) = S_t^\diamond \hat{u}_b^\diamond - b_p^\diamond$$

où $S_l^\diamond$ est l'opérateur discret linéaire *Dirichlet-to-Neumann* défini en (2.7) et $b_p^\diamond$ le second membre primal condensé. On rappelle l'expression du second membre dual condensé : $b_d^\diamond = -t^\diamond (K^\diamond)^\dagger f_{ext}^\diamond = -F_t^\diamond b_p$ (voir section 1.2.3, chap. 1).

Remarque 2 : On a l'égalité :

$$P_B^T B^\diamond F_l^\diamond \left(B^\diamond{}^T \lambda_B; f_{ext}^\diamond \right) = B^\diamond F_l^\diamond \left(B^\diamond{}^T \lambda_B; f_{ext}^\diamond \right) + B^\diamond R_b^\diamond \alpha^\diamond \quad (4.3)$$

En effet, la contribution $\alpha^\diamond$ du déplacement dans l'espace généré par les modes rigides peut être déterminée par une considération physique de minimisation du saut de déplacement d'interface :

$$\alpha^\diamond = \underset{\mathbb{R}^{\sum_s d^{(s)}}}{\operatorname{argmin}} \| B^\diamond u_b^\diamond \|_Q$$

où Q est SPD (voir section 1.2.7, chap. 1). En notant $g_B = B^\diamond F_{nl}^\diamond \left(B^\diamond{}^T \lambda_B; f_{ext}^\diamond \right)$,

on a :

$$\begin{aligned}\|B^\diamond u_b^\diamond\|_Q &= (B^\diamond u_b^\diamond)^T Q B^\diamond u_b^\diamond \\ &= \left(F_l^\diamond \left(B^{\diamond T} \lambda_B; f_{ext}^\diamond \right) + R_b^\diamond \alpha^\diamond \right)^T B^{\diamond T} Q B^\diamond \left(F_l^\diamond \left(B^{\diamond T} \lambda_B; f_{ext}^\diamond \right) + R_b^\diamond \alpha^\diamond \right) \\ &= \left(g_B^T + \alpha^{\diamond T} G_B^T \right) Q \left(g_B + G_B \alpha^\diamond \right)\end{aligned}$$

D'où

$$\frac{\partial}{\partial \alpha^\diamond} (\|B^\diamond u_b^\diamond\|_Q) = 2G_B^T Q g_B + 2G_B^T Q G_B \alpha^\diamond \Rightarrow G_B \alpha^\diamond = - (G_B^T Q G_B)^{-1} G_B^T Q g_B$$

□

Par la propriété (4.3) on a finalement la relation :

$$\begin{aligned}\hat{u}_b^\diamond &= \left[I - \tilde{B}^{\diamond T} P_B^T B^\diamond \right] F_l^\diamond \left(B^{\diamond T} \lambda_B; f_{ext}^\diamond \right) + R_b^\diamond \alpha^\diamond \\ \Rightarrow S_t^\diamond \hat{u}_b^\diamond &= S_t^\diamond \left[I - \tilde{B}^{\diamond T} P_B^T B^\diamond \right] F_l^\diamond \left(B^{\diamond T} \lambda_B; f_{ext}^\diamond \right)\end{aligned}\tag{4.4}$$

De même que précédemment, la réaction nodale $\hat{\lambda}_b^\diamond$ n'étant pas a priori équilibrée d'un sous-domaine à l'autre, grâce aux propriétés (1.24) une quantité équilibrée peut être obtenue par soustraction du déséquilibre pondéré :

$$\hat{\hat{\lambda}}_b^\diamond = \hat{\lambda}_b^\diamond - \tilde{A}^{\diamond T} A^\diamond \hat{\lambda}_b^\diamond$$

À convergence, on a :

$$\begin{aligned}B^\diamond u_b^\diamond = 0 \Rightarrow \hat{\lambda}_b^\diamond &= S_l^\diamond (u_b^\diamond + R_b^\diamond \alpha^\diamond; f_{ext}^\diamond) = S_t^\diamond u_b^\diamond - b_p^\diamond = S_t^\diamond \left(F_t^\diamond B^{\diamond T} \lambda_B - \underbrace{b_d^\diamond + R_b^\diamond \alpha^\diamond}_{-F_t^\diamond b_p^\diamond} \right) - b_p^\diamond \\ &= B^{\diamond T} \lambda_B\end{aligned}$$

D'où on tire la relation à convergence :

$$\hat{\hat{\lambda}}_b^\diamond = \left[I - \tilde{A}^{\diamond T} A^\diamond \right] B^{\diamond T} \lambda_B = B^{\diamond T} \lambda_B$$

On note $H_{l,d}^\diamond$ l'opérateur linéaire défini par :

$$\begin{aligned}H_{l,d}^\diamond \left(B^{\diamond T} \lambda_B; f_{ext}^\diamond \right) &\equiv \hat{\hat{\lambda}}_b^\diamond \\ &= \left[I - \tilde{A}^{\diamond T} A^\diamond \right] S_l^\diamond \left(\left[I - \tilde{B}^{\diamond T} P_B^T B^\diamond \right] F_l^\diamond \left(B^{\diamond T} \lambda_B; f_{ext}^\diamond \right); f_{ext}^\diamond \right)\end{aligned}$$

Pour des raisons de clarté d'écriture, la dépendance de $H_{l,d}^\diamond$ aux efforts extérieurs $f_{ext}^\diamond$ sera rendue implicite dans la suite de ce paragraphe. Le problème d'interface dual linéaire devient :

$$H_{l,d}^\diamond \left(B^{\diamond T} \lambda_B \right) = B^{\diamond T} \lambda_B\tag{4.5}$$

On a ainsi défini un problème de recherche de point fixe à partir du problème d'interface dual. L'équivalence avec le problème préconditionné classique peut être aisément montrée. En effet, en considérant les propriétés satisfaites par les opérateurs d'assemblage pondéré (1.24) et (1.28), et en notant classiquement $\lambda_B = \lambda_{B_0} + P_B \underline{\lambda}_B$ où λ_{B_0} et P_B sont définis en (1.32), on a :

$$\begin{aligned} [H_{l,d}^\diamond - I] (B^{\diamond^T} \lambda_B) &= B^{\diamond^T} \left[\tilde{B}^\diamond \left(S_t^\diamond \left[I - \tilde{B}^{\diamond^T} P_B^T B^\diamond \right] \left(F_t^\diamond B^{\diamond^T} \lambda_B - b_d^\diamond \right) - b_p^\diamond \right) - \lambda_B \right] \\ &= -B^{\diamond^T} \tilde{B}^\diamond S_t^\diamond \tilde{B}^{\diamond^T} P_B^T B^\diamond \left[F_t^\diamond B^{\diamond^T} (\lambda_{B_0} + P_B \underline{\lambda}_B) - b_d^\diamond \right] \end{aligned} \quad (4.6)$$

Le problème ainsi construit n'est autre que la contribution locale sur chaque sous-domaine du problème d'interface global FETI préconditionné (4.1). Enfin, les relations (1.24) satisfaites par les opérateurs d'assemblage $\tilde{B}^\diamond$ et $B^{\diamond^T}$ permettent d'assurer :

$$\forall x_b^\diamond \in \mathbb{R}_b^\diamond, \quad B^{\diamond^T} \tilde{B}^\diamond x_b^\diamond = 0 \Rightarrow \tilde{B}^\diamond B^{\diamond^T} \tilde{B}^\diamond x_b^\diamond = 0 = \tilde{B}^\diamond x_b^\diamond$$

Soit encore :

$$\text{Ker} \left(B^{\diamond^T} \right) \cap \text{Range} \left(\tilde{B}^\diamond \right) = \{0\} \quad (4.7)$$

Cette propriété permet de s'affranchir du facteur $B^{\diamond^T}$ dans l'équation (4.6) afin de recouvrer la forme classique d'interface du problème FETI.

Remarque 3 : L'opérateur $H_{l,d}^\diamond$ défini ci-dessus n'est pas contractant. En effet, on peut montrer que la plus petite valeur propre $\sigma_{\min}$ de l'opérateur FETI préconditionné projeté vérifie $\sigma_{\min} > 1$ [Toselli and Widlund, 2005] : un algorithme de point fixe appliqué au système (4.5) ne pourrait ainsi pas converger, un algorithme de type Krylov doit être utilisé. Même dans le cas contraire, la rapidité de convergence des algorithmes de Krylov inciterait tout de même à utiliser la formulation (4.1); l'interprétation en termes de point fixe n'est donnée ici qu'à titre de prélude au preconditionneur non linéaire.

2.1.2 Approche primale

Une méthode similaire permet de créer un problème d'interface primal de type point fixe bien conditionné. On définit $\hat{u}_b^\diamond$ un déplacement d'interface continu par :

$$\hat{u}_b^\diamond = \left[I - \tilde{B}^{\diamond^T} B^\diamond \right] F_l^\diamond \left(\left[I - \tilde{A}^{\diamond^T} P_A^T A^\diamond \right] S_l^\diamond \left(A^{\diamond^T} u_A; f_{ext}^\diamond \right); f_{ext}^\diamond \right)$$

et on note $H_{l,p}^\diamond$ l'opérateur linéaire défini par :

$$\begin{aligned} H_{l,p}^\diamond \left(A^{\diamond^T} u_A; f_{ext}^\diamond \right) &\equiv \hat{u}_b^\diamond \\ &= A^{\diamond^T} \tilde{A}^\diamond F_l^\diamond \left(\left[I - \tilde{A}^{\diamond^T} P_A^T A^\diamond \right] S_l^\diamond \left(A^{\diamond^T} u_A; f_{ext}^\diamond \right); f_{ext}^\diamond \right) \end{aligned}$$

Comme précédemment, on négligera de rappeler la dépendance de $H_{l,p}^\diamond$ aux efforts extérieurs $f_{ext}^\diamond$ dans la suite.

Le problème classique d'interface primal $A^\diamond S_l^\diamond (A^{\diamond T} u_A; f_{ext}^\diamond) = 0$ est alors équivalent à la recherche de point fixe de l'opérateur $H_{l,p}^\diamond$:

$$H_{l,p}^\diamond (A^{\diamond T} u_A) = A^{\diamond T} u_A$$

De plus, on a équivalence entre le problème de recherche de point fixe et le problème BDD classiquement préconditionné :

$$\tilde{A}^\diamond F_t^\diamond \tilde{A}^{\diamond T} P_A^T A^\diamond S_l^\diamond (A^{\diamond T} u_A; f_{ext}^\diamond) = 0 \quad (4.8)$$

En effet, d'après les propriétés (1.24) et (1.28) vérifiées par les opérateurs d'assemblage pondéré, on a :

$$\begin{aligned} [H_{l,p}^\diamond - I] (A^{\diamond T} u_A) &= A^{\diamond T} \tilde{A}^\diamond \left(F_t^\diamond [I - \tilde{A}^{\diamond T} P_A^T A^\diamond] (S_t^\diamond A^{\diamond T} u_A - b_p^\diamond) - \underbrace{b_d^\diamond}_{-F_t^\diamond b_p^\diamond} \right) \\ &\stackrel{\Leftrightarrow}{\text{Ker}(A^{\diamond T})=\{0\}} \quad (4.8) \end{aligned}$$

2.2 Préconditionneur non linéaire

La stratégie mise en place dans le cas linéaire peut être généralisée au cas non linéaire : la construction d'un problème de recherche de point fixe non linéaire permet d'obtenir un système condensé à l'interface dont l'opérateur tangent est intrinsèquement bien conditionné. L'approche duale sera détaillée, on verra que l'approche primale quant à elle est difficilement exploitable dans un contexte où la présence de modes rigides imposerait une profonde modification de la stratégie.

2.2.1 Approche duale

Construction d'un nouveau problème non linéaire condensé à l'interface De même que précédemment, à partir des solutions $u_b^\diamond$ des équilibres locaux non linéaires, on peut construire un déplacement continu aux interfaces :

$$\hat{u}_b^\diamond \equiv [I - \tilde{B}^{\diamond T} B^\diamond] u_b^\diamond = [I - \tilde{B}^{\diamond T} P_B^T B^\diamond] F_{nl}^\diamond (B^{\diamond T} \lambda_B; f_{ext}^\diamond) + R_b^\diamond \alpha^\diamond \quad (4.9)$$

La réaction nodale $\hat{\lambda}_b^\diamond = S_{nl}^\diamond (\hat{u}_b^\diamond; f_{ext}^\diamond)$ solution d'équilibres locaux non linéaires à conditions de Dirichlet continues $\hat{u}_b^\diamond$ imposées peut également être équilibrée :

$$\hat{\lambda}_b^\diamond = [I - \tilde{A}^{\diamond T} A^\diamond] \hat{\lambda}_b^\diamond = [I - \tilde{A}^{\diamond T} A^\diamond] S_{nl}^\diamond (\hat{u}_b^\diamond; f_{ext}^\diamond)$$

La dépendance des opérateurs $F_{nl}^\diamond$ et $S_{nl}^\diamond$ au chargement extérieur $f_{ext}^\diamond$ sera à nouveau considérée de façon implicite dans la suite, afin de ne pas surcharger les notations. À convergence, on a :

$$B^\diamond u_b^\diamond = 0 \Rightarrow \hat{\lambda}_b^\diamond = \left[I - \tilde{A}^\diamond{}^T A^\diamond \right] S_{nl}^\diamond \left(F_{nl}^\diamond \left(B^{\diamond T} \lambda_B \right) + R_b^\diamond \alpha^\diamond \right)$$

L'hypothèse (4.2) faite sur l'opérateur $f_{int}^\diamond$ entraîne la propriété suivante :

$$\forall v_b^\diamond \in \mathbb{R}^{\Sigma_s n_b^{(s)}}, \forall \beta^\diamond \in \mathbb{R}^{\Sigma_s d^{(s)}}, \quad S_{nl}^\diamond (v_b^\diamond + R_b^\diamond \beta^\diamond) = S_{nl}^\diamond (v_b^\diamond) \quad (4.10)$$

On a alors à convergence :

$$\hat{\lambda}_b^\diamond = B^{\diamond T} \lambda_B$$

Ainsi, en définissant l'opérateur $H_{nl,d}^\diamond$ tel que :

$$\begin{aligned} H_{nl,d}^\diamond (B^{\diamond T} \lambda_B) &= \hat{\lambda}_b^\diamond \\ &= B^{\diamond T} \tilde{B}^\diamond S_{nl}^\diamond \left(\left[I - \tilde{B}^\diamond{}^T P_B^T B^\diamond \right] F_{nl}^\diamond (B^{\diamond T} \lambda_B) \right) \end{aligned}$$

et en prenant en compte les propriétés (1.24), (1.28), on a l'implication :

$$P_B^T B^\diamond F_{nl}^\diamond (B^{\diamond T} \lambda_B) = 0 \Rightarrow H_{nl,d}^\diamond (B^{\diamond T} \lambda_B) = B^{\diamond T} \lambda_B \quad (4.11)$$

Ce nouveau problème condensé à l'interface est dit préconditionné non linéairement ; on verra en effet dans la suite que le conditionnement de l'opérateur tangent résultant est proche de 1.

Remarque 4 : La condition d'admissibilité portant sur la réaction nodale d'interface λ_B n'est pas modifiée par l'introduction du preconditionneur non linéaire :

$$R^{\diamond T} f_{int}^\diamond = 0 \Leftrightarrow R^{\diamond T} \left(f_{ext}^\diamond + t^{\diamond T} B^{\diamond T} \underline{\lambda}_B \right) = 0$$

La stratégie de projection classiquement définie précédemment (voir sections 1.2.7, chap. 1 et 3.1.2, chap. 2) pour traiter les modes rigides peut donc être utilisée : on définit un projecteur constant P_B sur $\text{Ker} \left(R_b^{\diamond T} B^{\diamond T} \right)$ et λ_B comme :

$$\lambda_B \equiv \underbrace{\lambda_{B_0}}_{\in \text{Vect} \left(R_b^{\diamond T} B^{\diamond T} \right)} + P_B \lambda_B$$

Remarque 5 : La stratégie de préconditionnement non linéaire est moins évidente avec une approche en déplacement. En effet, la présence de modes rigides au sein des sous-structures implique alors la vérification par l'inconnue d'interface u_A d'une condition d'optimalité non linéaire :

$$\tilde{R}_A^T A^\diamond S_{nl}^\diamond (A^\diamond{}^T u_A) = 0 \quad (4.12)$$

Tandis que dans le cas d'un préconditionnement linéaire, l'espace de recherche des solutions défini par la condition d'optimalité est un espace vectoriel (voir section 1.2.7, chap. 1), l'espace défini par (4.12) est une variété. Une stratégie de minimisation sous contraintes non linéaires serait envisageable, cependant la complexité supplémentaire engendrée par un espace de recherche non vectoriel nous a incités à abandonner l'analyse de la méthode dans le cas primal. Dans [Klawonn et al., 2017], la résolution de problèmes grossiers non linéaires est impliquée dans la formulation du solveur NL-FETIDP2. Néanmoins, leur structure semble plus simple, en particulier une résolution inexacte est compatible avec la convergence de la méthode.

2.2.2 Opérateur tangent

Calcul de l'opérateur tangent Une preuve de la pertinence de l'introduction du nouveau problème non linéaire condensé à l'interface (4.11) peut être trouvée en la qualité du conditionnement de l'opérateur tangent obtenu. En effet, une stratégie de résolution basée sur un algorithme de Newton inexact, dont le résidu serait évalué par des résolutions locales non linéaires, et le problème tangent résolu par une méthode DD, est toujours envisageable dans le cas de l'approche duale. Le calcul de l'opérateur tangent est alors particulièrement aisé. En notant classiquement :

$$\lambda_B = \lambda_{B_0} + P_B \underline{\lambda}_B \quad \Rightarrow \quad d\lambda_B = P_B d\underline{\lambda}_B$$

avec le projecteur P_B et l'initialisation λ_{B_0} définis en (1.32), on a par dérivation composée :

$$H_{t,d}^\diamond \equiv \frac{\partial [H_{nl,d}^\diamond - I] (B^\diamond{}^T \lambda_B)}{\partial \underline{\lambda}_B} = B^\diamond{}^T \left(\tilde{B}^\diamond S_t^\diamond [I - \tilde{B}^\diamond{}^T P_B^T B^\diamond] F_t^\diamond B^\diamond{}^T - I \right) P_B$$

L'opérateur tangent $F_t^\diamond$ est calculé à partir de la condensation des matrices tangentes locales au point $\lambda_b^\diamond = B^\diamond{}^T \lambda_B$, tandis que $S_t^\diamond$ est calculé à partir des matrices tangentes locales au point $\hat{u}_b^\diamond = [I - \tilde{B}^\diamond{}^T P_B^T B^\diamond] F_{nl}^\diamond (B^\diamond{}^T \lambda_B)$. L'utilisation d'une stratégie de Newton modifié où $F_t^\diamond$ et $S_t^\diamond$ sont extraits des mêmes matrices locales tangentes permet de retrouver la propriété $S_t^\diamond = F_t^{\diamond\dagger}$, et la simplification suivante peut être effectuée :

$$H_{t,d}^\diamond = -B^\diamond{}^T \left(\tilde{B}^\diamond S_t^\diamond \tilde{B}^\diamond{}^T P_B^T B^\diamond F_t^\diamond B^\diamond{}^T P_B \right)$$

On reconnaît la contribution locale de l'opérateur classique linéaire FETI projeté et préconditionné (voir section 1.2.7, chap. 1), assurant un conditionnement quasi-optimal du problème tangent. La $k^{\text{ième}}$ itération d'un algorithme de Newton, $\lambda_{B_k} = \lambda_{B_0} + P_B \lambda_{B_k}$, appliqué au problème préconditionné non linéairement est ainsi constituée des étapes :

$$\begin{cases} H_{t,d}^{\diamond} d\lambda_B = - \left[H_{nl,d}^{\diamond} - I \right] \left(B^{\diamond^T} \lambda_{B_k} \right) \\ \lambda_{B_{k+1}} = \lambda_{B_k} + P_B d\lambda_B \end{cases} \quad (4.13)$$

Le facteur $B^{\diamond^T}$ des membres de gauche et de droite du problème tangent peut, de même que dans le cas linéaire de la section précédente, être simplifié grâce à la propriété (4.7) afin de retrouver une formulation définie sur l'interface duale globale pour le problème tangent :

$$\tilde{B}^{\diamond} S_t^{\diamond} \tilde{B}^{\diamond^T} P_B^T B^{\diamond} F_t^{\diamond} B^{\diamond^T} P_B d\lambda_B = \tilde{B}^{\diamond} S_{nl}^{\diamond} \left(\left[I - \tilde{B}^{\diamond^T} P_B^T B^{\diamond} \right] F_{nl}^{\diamond} \left(B^{\diamond^T} \lambda_{B_k} \right) \right) - \lambda_{B_k} = 0$$

On note $\tilde{F}_B^{-1} = \tilde{B}^{\diamond} S_t^{\diamond} \tilde{B}^{\diamond^T}$ et $F_B = B^{\diamond} F_t^{\diamond} B^{\diamond^T}$. On définit pour la suite de ce chapitre l'opérateur tangent sur l'interface globale :

$$\tilde{g}_{t_B} \equiv \tilde{F}_B^{-1} P_B^T F_B P_B$$

et le résidu d'interface global préconditionné non linéairement :

$$\tilde{g}_B(\lambda_B) \equiv \tilde{B}^{\diamond} S_{nl}^{\diamond} \left(\left[I - \tilde{B}^{\diamond^T} P_B^T B^{\diamond} \right] F_{nl}^{\diamond} \left(B^{\diamond^T} \lambda_B \right) \right) - \lambda_B$$

Implémentation : problème tangent Le schéma de résolution FETI préconditionné projeté linéaire se base de manière classique sur l'utilisation d'un algorithme de Gradient Conjugué avec préconditionneur (GCP), dont la forme générale est donnée à l'algorithme 12 pour la résolution du problème linéaire type :

$$\tilde{M}^{-1} Mx = \tilde{M}^{-1} b$$

L'initialisation de l'algorithme GCP est réalisée en deux étapes :

- (i) à partir d'une initialisation de l'inconnue x_0 , le calcul du résidu non préconditionné $r_0 = Mx_0 - b$ est effectué
- (ii) on calcule le résidu préconditionné $z_0 = \tilde{A}^{-1} r_0$

Les deux grandeurs r_k et z_k sont ensuite utilisées distinctement au cours de l'algorithme, qui suppose donc la connaissance de r_0 et z_0 à l'initialisation. Dans le cas du problème tangent au système préconditionné non linéairement (4.13), on résout une équation linéaire de la forme :

$$\tilde{M}^{-1} Mx = \tilde{b}_{nl}$$

où $\tilde{b}_{nl} = \tilde{g}_B(\lambda_B)$ représente le résidu d'interface préconditionné non linéairement. La stratégie de Gradient Conjugué préconditionné n'est donc pas applicable dans ce contexte, puisque la donnée du second membre non préconditionné b n'est pas accessible. Toutefois, une stratégie de résolution passant par un algorithme de Krylov (non préconditionné) dont l'opérateur serait défini comme l'opérateur tangent préconditionné $\tilde{M}^{-1}M = \tilde{g}_{t_B}$ est envisageable, pour une vitesse de convergence équivalente. Le problème ainsi défini ayant perdu la propriété de symétrie nécessaire à la bonne convergence du Gradient Conjugué, un algorithme adapté doit être utilisé (GMRES ou Orthodir [Saad, 2003] par exemple), sans perte de performance.

Algorithm 12 Preconditioned Conjugate Gradient

Solve preconditioned linear system $\tilde{M}^{-1}Mx = \tilde{M}^{-1}b$:

initialization : iterate x_0 , threshold ε , $k = 0$
 compute residual $r_0 = b - Mx_0$
 precondition residual $z_0 = \tilde{M}^{-1}r_0$
 set search direction $w_0 = z_0$

while $\sqrt{z_k^T r_k} > \varepsilon$ **do**
 compute $p_k = Mr_k$
 compute $g_k = p_k^T w_k$
 compute parameter $\alpha = \frac{z_k^T r_k}{g_k}$
 actualize residual $r_{k+1} = r_k - \alpha p_k$
 and iterate $x_{k+1} = x_k + \alpha w_k$
 compute preconditioned residual $z_{k+1} = \tilde{M}^{-1}r_{k+1}$
 set new search direction $w_{k+1} = z_{k+1}$
 complete reorthogonalization :
 for $i \in \{1, \dots, k-1\}$ **do**
 compute $\beta_i = \frac{w_{k+1}^T p_i}{g_i}$
 actualize new search direction $w_{k+1} -= \beta_i * w_i$

$k += 1$

2.2.3 Équivalence des problèmes d'interface non linéaires classique et non linéairement préconditionné

On considère dans cette section la question de l'équivalence entre les systèmes :

(i) non linéairement condensé :

$$P_B^T B^\diamond F_{nl}^\diamond (B^{\diamond T} \lambda_B) = 0 \quad (4.14)$$

(ii) non linéairement condensé et préconditionné :

$$B^{\diamond T} \tilde{B}^\diamond S_{nl}^\diamond \left(\left[I - \tilde{B}^{\diamond T} P_B^T B^\diamond \right] F_{nl}^\diamond (B^{\diamond T} \lambda_B) + R_b^\diamond \alpha^\diamond \right) = B^{\diamond T} \lambda_B \quad (4.15)$$

Le résidu non linéairement préconditionné du problème (4.15) n'est autre que la contribution locale de $\tilde{g}_B$. On notera également dans la suite :

$$g_B (B^{\diamond T} \lambda_B) = P_B^T B^\diamond F_{nl}^\diamond (B^{\diamond T} \lambda_B)$$

le résidu d'interface condensé non linéairement et projeté.

On cherche à prouver ici l'équivalence des systèmes :

$$B^{\diamond T} \tilde{g}_B (\lambda_B) = 0 \Leftrightarrow g_B (B^{\diamond T} \lambda_B) = 0$$

($\Leftarrow$) L'implication $g_B (B^{\diamond T} \lambda_B) = 0 \Rightarrow B^{\diamond T} \tilde{g}_B (\lambda_B) = 0$ est triviale (par construction de $\tilde{g}_B$).

($\Rightarrow$) L'implication $B^{\diamond T} \tilde{g}_B (\lambda_B) = 0 \Rightarrow g_B (B^{\diamond T} \lambda_B) = 0$ peut être prouvée au moins localement (dans un voisinage de la solution). En effet, en notant :

$$u_b^\diamond = F_{nl}^\diamond (B^{\diamond T} \lambda_B)$$

et en considérant la propriété (4.10) (vérifiée par hypothèse), le problème préconditionné non linéairement $B^{\diamond T} \tilde{g}_B (\lambda_B) = 0$ peut être reformulé :

$$B^{\diamond T} \tilde{B}^\diamond S_{nl}^\diamond (u_b^\diamond - \tilde{B}^{\diamond T} P_B^T B^\diamond u_b^\diamond) = S_{nl}^\diamond (u_b^\diamond)$$

D'après les relations (1.24), on a alors :

$$B^{\diamond T} \tilde{g}_B (\lambda_B) = 0 \Rightarrow \tilde{B}^\diamond \left(S_{nl}^\diamond (u_b^\diamond - \tilde{B}^{\diamond T} P_B^T B^\diamond u_b^\diamond) - S_{nl}^\diamond (u_b^\diamond) \right) = 0 \quad (4.16)$$

Afin de prouver le résultat localement, on considère un voisinage $\mathcal{V}_\lambda (\lambda_{B_{sol}})$ de la solution $\lambda_{B_{sol}}$ du problème $g_B (B^{\diamond T} \lambda_B) = 0$, à l'intérieur duquel la quantité $g_B (B^{\diamond T} \lambda_B)$ est arbitrairement petite, ou de façon équivalente, un voisinage $\mathcal{V}_u (u_{b_{sol}}^\diamond)$ de la solution $u_{b_{sol}}^\diamond$ du problème $P_B^T B^\diamond u_b^\diamond = 0$. À l'intérieur de ce voisinage, on a :

$$\forall u_{b_v}^\diamond \in \mathcal{V}_u (u_{b_{sol}}^\diamond), \quad \tilde{B}^\diamond S_t^\diamond \tilde{B}^{\diamond T} P_B^T B^\diamond u_{b_v}^\diamond + o \left(\tilde{B}^{\diamond T} P_B^T B^\diamond u_{b_v}^\diamond \right) = 0$$

où $S_t^\diamond$ est la jacobienne de $S_{nl}^\diamond(u_b^\diamond)$ calculée au point $u_{b_v}^\diamond \in \mathcal{V}_u(u_{b_{sol}}^\diamond) : S_t^\diamond \equiv \frac{\partial S_{nl}^\diamond}{\partial u_b^\diamond} \Big|_{u_b^\diamond = u_{b_v}^\diamond}$.

Le résultat final découle des propriétés de symétrie, définition et positivité satisfaites par la matrice $(\tilde{B}^\diamond S_t^\diamond \tilde{B}^{\diamond T})$, assurant l'implication :

$$\forall \lambda_{B_v} \in \mathcal{V}_\lambda(\lambda_{B_{sol}}), \quad B^{\diamond T} \tilde{g}_B(\lambda_{B_v}) = 0 \Rightarrow \underbrace{P_B^T B^\diamond u_{b_v}^\diamond}_{F_{nl}^\diamond(B^{\diamond T} \lambda_{B_v})} = 0 \Leftrightarrow g_B(B^{\diamond T} \lambda_{B_v}) = 0$$

2.2.4 Une stratégie de résolution basée sur un algorithme de Newton-Raphson

La stratégie définie pour la méthode de sous-structuration et condensation non linéaires est applicable à la résolution du problème d'interface préconditionné non linéairement :

$$\tilde{g}_B(\lambda_B) = 0 \quad (4.17)$$

avec quelques modifications mineures. De même que dans le cas linéaire, l'opérateur $H_{nl,d}^\diamond$ n'étant pas contractant, un solveur de type point fixe n'est pas envisageable ; tandis que rien ne s'oppose à l'utilisation d'un algorithme de Newton. Après l'initialisation classique :

$$\lambda_{B_0} = -QG_B(G_B^T QG_B)^{-1} G_B^T f_{ext}^\diamond$$

l'algorithme de Newton appliqué au problème d'interface global devient :

$$\begin{cases} \tilde{g}_{t_B} d\lambda_B = \tilde{g}_B \\ \lambda_B += P_B d\lambda_B \end{cases}$$

et les problèmes tangents ont la forme exacte de systèmes FETI.

La procédure de résolution, à laquelle on se référera par la suite en tant que FETI-precNL, fait ainsi intervenir trois étapes principales :

(i) Construire le résidu d'interface non linéaire :

- (a) Résoudre les équilibres locaux à conditions aux limites de Neumann $B^{\diamond T} \lambda_B$ et assembler le résidu d'interface dual
- (b) Construire un déplacement d'interface continu d'après (4.9), résoudre les équilibres locaux à conditions aux limites de Dirichet imposées $\hat{u}_b^\diamond$, assembler et pondérer le résidu pour obtenir $\tilde{g}_B$. Vérifier le critère de convergence.

(ii) Appliquer un algorithme de Newton global pour résoudre le problème (4.17), condensé à l'interface duale et préconditionné non linéairement.

(iii) Résoudre les problèmes tangents linéarisés associés par un solveur de Krylov adapté aux systèmes non symétriques.

3 Résultats

Deux cas-tests numériques sont présentés dans cette section, afin de juger de l'efficacité de la procédure de préconditionnement non linéaire introduite dans le chapitre en cours. Le premier met en jeu l'équation de Richards régissant les phénomènes de diffusion dans un milieu poreux au cours du temps ; le second traite de l'évolution thermique non linéaire d'un domaine en régime permanent.

Les résultats numériques ont été obtenus grâce au logiciel FEniCS [Logg et al., 2012], interfacé par un code Python.

3.1 Diffusion d'eau dans une colonne de sol

Le premier cas-test s'inspire fortement du problème standard dit de *Polmann* [Celia et al., 1990, Sochala, 2008] : on étudie un phénomène de diffusion d'eau dans une colonne de sol, dans deux directions (voir figure 4.1a). Les paramètres géométriques du problème sont donnés dans le tableau 4.1. Le champ de pression initial est défini comme spatialement homogène :

$$h(x, t < 0) = h_{D_0}$$

Au temps $t = 0$, une pression $h = h_{D_1}$ est imposée sur le bord défini par $y = L$, et $h = h_{D_2}$ sur le bord défini par $(x = 0, y \in [3L/8; L/2])$. On impose également une condition d'imperméabilité (flux normal nul) sur les parois verticales restantes. Les paramètres du chargement sont donnés dans le tableau 4.1.

Geometrical parameters	
L	1
W	0.2
Van Genuchten parameters	
θ_r	0.368
θ_s	0.102
K_s	9.22×10^{-5}
n	2
m	0.5
Load parameters	
h_{D_0}	-10
h_{D_1}	-1.5
h_{D_2}	-1.5

TABLE 4.1 – Paramètres géométriques, de Van Genuchten et de chargement : colonne de sol

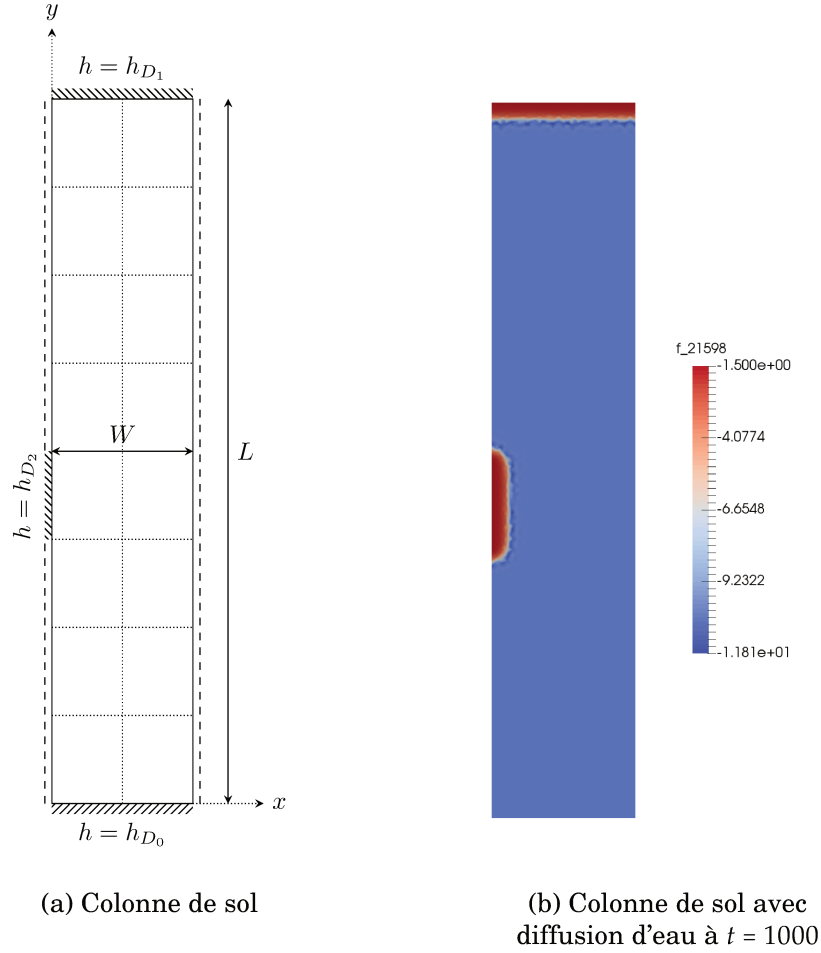


FIGURE 4.1 – Équation de Richards : colonne de sol avec diffusion d'eau dans deux directions

L'équation de Richards, modélisant les phénomènes de diffusion d'eau dans les milieux poreux, est considérée ici sous sa formulation classique :

Trouver un champ de pression scalaire $h(x, t)$ tel que :

$$\frac{\partial \theta(h)}{\partial t} - \nabla \cdot (K(h) \nabla (h - z))$$

où :

- $\theta(h)$ est la teneur en eau volumique :

$$\theta(h) = \theta_r + (\theta_s - \theta_r) S(h)$$

avec θ_r , θ_s les teneurs en eau résiduelle et à saturation, et S le degré de saturation.

- $K(h)$ est la conductivité hydraulique du sol :

$$K(h) = K_s K_r(h)$$

avec K_s, K_r les conductivités intrinsèque et relative du sol.

Le modèle de Van Genuchten est utilisé pour le degré de saturation et la conductivité relative :

$$\begin{cases} S(h) = (1 + \varepsilon|h|^n)^{-m} \\ K_r(h) = \left(1 - (\varepsilon|h|)^{n-1} (1 + (\varepsilon|h|)^n)^{-m}\right)^2 (S(h))^{1/2} \end{cases}$$

Les paramètres matériau sont donnés dans le tableau 4.1.

Remarque 6 : La sous-structuration du problème global défini par la discrétisation de la formulation faible de l'équation de Richards ne fait pas intervenir de modes rigides au sein des sous-structures. La section suivante présentera les résultats de l'approche préconditionnée non linéairement pour un cas de thermique non linéaire où chaque opérateur local tangent peut admettre un noyau de dimension $n_{ker} = 1$.

L'intégration temporelle est réalisée au niveau discret par un schéma implicite d'Euler. Le pas de temps est fixé à $\Delta t = 20$. La discrétisation spatiale fait intervenir 5244 nœuds (ou degrés de liberté sur ce problème scalaire), et la sous-structuration en 16 sous-domaines 333 nœuds d'interface (voir figure 4.1a).

La précision relative de chaque solveur est fixée à $\varepsilon_{rel} = 10^{-6}$, et la précision absolue à $\varepsilon_{abs} = 10^{-20}$.

Les trois méthodes suivantes sont comparées :

- méthode de sous-structuration et condensation non linéaires avec approche primale, à laquelle on se référera dans la suite en termes de méthode BDD-NL
- méthode de sous-structuration et condensation non linéaires avec approche duale, à laquelle on se référera dans la suite en termes de méthode FETI-NL
- méthode FETI-precNL

La comparaison est effectuée sur les nombres d'itérations des trois solveurs imbriqués au sein du processus de résolution de chaque méthode : les nombres d'itérations du solveur de Newton global (cumulées sur les pas de temps), les nombres d'itérations de Krylov (cumulées sur les itérations de Newton globales et sur les pas de temps), et les itérations de Newton locales (ces dernières étant effectuées en parallèle sur différents processeurs, à chaque boucle de Newton globale, seul le maximum – sur les sous-domaines – du nombre d'itérations locales est retenu ; cette valeur est ensuite cumulée sur les itérations de Newton globales et les pas de temps)

Les résultats sont donnés dans les tableaux 4.2, 4.3 et 4.4, en termes de nombres d'itérations de Newton globales, de Krylov et de Newton locales respectivement. La figure 4.1b présente l'état de la colonne de sol au temps $t = 1000$.

Sur ce problème en particulier, les performances des méthodes BDD-NL et FETI-NL sont très proches (voir les tableaux 4.2 et 4.3) : au dernier pas de temps, un total de 1535 itérations cumulées de Krylov est effectué pour les deux approches, tandis que les nombres d'itérations cumulées de Newton globales atteignent 311 et 325 pour les approches primale et duale respectivement. L'approche primale présente cependant un coût inférieur, en termes d'itérations cumulées de Newton locales, à l'approche duale : au dernier pas de temps, seules 2559 itérations cumulées locales sont nécessaires, contre 3173 pour l'approche duale (voir tableau 4.4).

Il apparaît également clairement dans les tableaux 4.2 et 4.3 que la méthode FETI-precNL est sur ce cas-test bien plus performante que les solveurs BDD-NL et FETI-NL. Un total de 179 itérations de Newton globales seulement est cumulé au dernier pas de temps avec la méthode FETI-precNL, qui totalise également 835 itérations de Krylov seulement au même instant. Les gains correspondants, en termes de nombres d'itérations cumulées de Newton globales, comparés à ceux du solveur BDD-NL (resp. FETI-NL), sont compris entre 42 et 57% (resp. 45 et 62%) tout au long de la résolution. En termes d'itérations cumulées de Krylov, les gains sont compris entre 46 et 59% (resp. 46 et 62%). Ces gains varient en fonction du temps : après avoir rapidement atteint un maximum, ils décroissent ensuite lentement au cours de la résolution. En effet, l'importance du gain dépend fortement de l'intensité de la non linéarité à traiter au cours du pas de temps. Cette dernière diminuant au cours de l'avancée du front d'eau dans la colonne, le gain décroît en conséquence car les pas de temps sont gardés constants.

Le coût du préconditionneur non linéaire, i.e. les itérations locales non linéaires parallèles additionnelles, est évalué dans le tableau 4.4 par les ratios des nombres d'itérations de Newton locales pour le solveur FETI-precNL, divisés par ceux des autres méthodes : un maximum de 1.41 est atteint à la fin de la résolution pour le ratio avec la méthode BDD-NL (1.14 pour la méthode FETI-NL). Ce léger surcoût devrait être largement compensé par la décroissance d'environ 50% en nombres d'itérations cumulées de Krylov (représentatives des communications entre processeurs au cours du processus de résolution).

Remarque 7 : La remarque du chapitre précédent (section 1) sur la taille des cas-tests présentés dans ce manuscrit est également à appliquer aux résultats de ce chapitre.

Une comparaison avec le processus de résolution classique – voir section 2, chap. 2 – est également présentée, pour un algorithme NKS avec solveur tangent BDD. Les résultats sont donnés dans le tableau 4.5 pour les trois algorithmes imbriqués (Newton global, Krylov et Newton locaux). Pour chaque pas de temps, les nombres d'itérations de Newton locales sont égaux, pour la méthode classique NKS, au nombre d'itérations

# d'itérations cumulées de Newton globales					
Temps	BDD-NL	FETI-NL	FETI-precNL	Gains FETI-precNL (%)	
				vs. BDD-NL	vs. FETI-NL
20	8	9	4	50	56
40	16	17	7	56	59
60	25	27	11	56	59
80	35	39	15	57	62
100	45	51	20	56	61
200	83	94	42	49	55
300	115	129	58	50	55
400	147	159	73	50	54
500	174	184	88	49	52
600	201	211	103	49	51
700	231	241	121	48	50
800	261	271	141	46	48
900	286	300	161	44	46
1000	311	325	179	42	45

TABLE 4.2 – Équation de Richards - itérations globales

# d'itérations cumulées de Krylov					
Temps	BDD-NL	FETI-NL	FETI-precNL	Gains FETI-precNL (%)	
				vs. BDD-NL	vs. FETI-NL
20	32	37	16	50	57
40	64	69	29	55	58
60	107	114	47	56	59
80	157	169	65	59	62
100	207	225	89	57	60
200	397	417	189	52	55
300	557	569	263	53	54
400	715	714	338	53	53
500	850	839	413	51	51
600	985	974	488	50	50
700	1135	1123	573	50	49
800	1285	1269	663	48	48
900	1410	1410	753	47	47
1000	1535	1535	835	46	46

TABLE 4.3 – Équation de Richards - itérations de Krylov

# d'itérations cumulées de Newton locales					
Temps	BDD-NL	FETI-NL	FETI-precNL	Ratios FETI-precNL	
				sur BDD-NL	sur FETI-NL
20	224	249	241	1,08	0,97
40	366	393	364	0,99	0,93
60	477	529	485	1,02	0,92
80	595	714	618	1,04	0,87
100	700	899	769	1,10	0,86
200	1085	1429	1354	1,25	0,95
300	1359	1783	1706	1,26	0,96
400	1576	2020	1981	1,26	0,98
500	1763	2211	2256	1,28	1,02
600	1939	2402	2505	1,29	1,04
700	2113	2604	2779	1,32	1,07
800	2283	2815	3079	1,35	1,09
900	2423	3011	3366	1,39	1,12
1000	2559	3173	3608	1,41	1,14

TABLE 4.4 – Équation de Richards - itérations locales

de Newton globales, plus une itération correspondant à l'initialisation du solveur global. Les gains sont compris, pour les itérations cumulées de Newton globales et du solveur de Krylov, entre 51 et 82% au cours de la résolution. Le ratio des itérations cumulées de Newton locales est proche de 10 au début de la résolution, puis décroît lentement jusqu'à atteindre 8.63 : un coût que l'on espère largement compensé par le gain en itérations de Krylov (et donc en communications entre processeurs).

3.2 Thermique non linéaire

Un second cas-test numérique traite de l'évolution thermique non linéaire d'un domaine Ω , décrite par l'équation aux dérivées partielles classique :

$$r_{vol} - \nabla \cdot (-K(T) \nabla T) = 0$$

avec T la température, r_{vol} une source volumique de chaleur, et $K(T)$ la conductivité thermique : la non linéarité provient de la dépendance de K à la température T . Afin de réutiliser le code déjà mis en place pour l'équation de Richards, tout en s'approchant de l'élasticité HPP, qui vérifie (4.10) et dont l'opérateur tangent a un noyau constant, on choisit l'expression suivante pour la conductivité :

$$K(T) = (1 + \nabla T \cdot \nabla T)^\alpha$$

# d'itérations cumulées de Newton globales			
Temps	NKS	FETI-precNL	Gains FETI-precNL (%)
20	21	4	81
100	72	20	72
200	117	42	64
400	189	73	61
700	279	121	57
1000	368	179	51

# d'itérations cumulées de Krylov			
Temps	NKS	FETI-precNL	Gains FETI-precNL (%)
20	84	16	81
100	321	89	72
200	546	189	65
400	906	338	63
700	1356	573	58
1000	1801	835	54

# d'itérations cumulées de Newton locales			
Temps	NKS	FETI-precNL	Ratio FETI-precNL sur NKS
20	22	241	10.95
100	77	769	9.99
200	127	1354	10.66
400	209	1981	9.48
700	314	2779	8.85
1000	418	3608	8.63

TABLE 4.5 – Équation de Richards - Comparaison avec la méthode NKS

au lieu des expressions plus classiques en thermique non linéaire (lois de type affine ou puissance) :

$$K(T) = 1 + \alpha T \quad \text{ou} \quad K(T) = (1 + T)^\alpha \quad (4.18)$$

Le domaine Ω est rectangulaire et soumis à des conditions aux limites de Dirichlet (voir figure 4.2). Les paramètres géométriques et de chargement sont donnés dans le tableau 4.6. Différents niveaux de non linéarité sont considérés, via une variation incrémentale du paramètre α , dont on fixera les bornes maximale et minimale à $\alpha_{min} = 0$ (comportement de thermique linéaire) et $\alpha_{max} = 0.8$ respectivement.

La discrétisation spatiale fait intervenir 5244 nœuds (ou degrés de liberté sur ce problème scalaire). La discrétisation en 16 sous-domaines implique 333 nœuds d'interface (voir figure 4.2).

La comparaison entre les solveurs BDD-NL, FETI-NL et FETI-precNL est effectuée à partir des nombres d'itérations des différents solveurs imbriqués au cours du

processus de résolution : itérations de Newton globales (pour chaque valeur du paramètre α), itérations de Krylov (cumulées sur les boucles de Newton globales) et itérations de Newton locales (cumulées sur les boucles de Newton globales). Les résultats sont donnés dans les tableaux 4.7 et 4.8. La figure 4.3 représente le domaine Ω pour $\alpha = \alpha_{max}$.

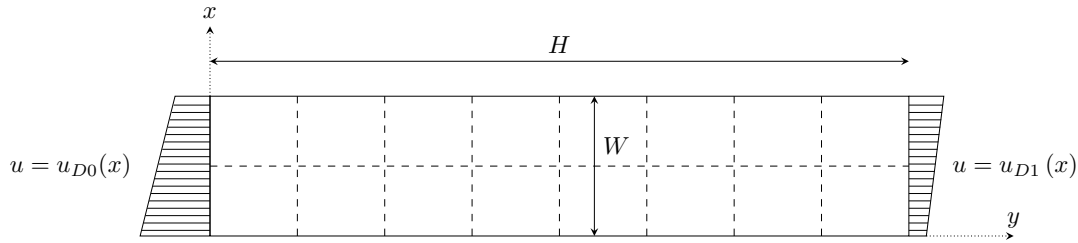


FIGURE 4.2 – Problème de thermique non linéaire

Paramètres géométriques	
H	1
W	0.2
Paramètres matériau	
α	$\in \{0, 0.2, 0.4, 0.6, 0.8\}$
Paramètres de chargement	
$u_{D1}(x)$	$10x + 2$
$u_{D2}(x)$	$-10x + 4$
r_{vol}	0

TABLE 4.6 – Paramètres géométriques, matériau et de chargement : évolution thermique non linéaire

Une première observation peut être faite en comparant les solveurs BDD-NL et FETI-NL uniquement : sur ce cas-test en particulier, la méthode primale est plus performante que l'approche duale. En effet, excepté le cas $\alpha = \alpha_{max}$, les nombres d'itérations de Krylov nécessaires à convergence sont en moyenne inférieurs de 20%. Pour $\alpha = 0.8$, les deux méthodes sont équivalentes.

En dépit de l'avantage de la méthode primale, qui semble prévaloir pour ce comportement précis, le solveur FETI-precNL, basé sur une approche duale, permet une convergence plus rapide que celle des solveurs BDD-NL et FETI-NL. Les gains, en termes de nombres d'itérations de Newton globales, sont compris entre 25 et 40%, dépendamment de l'intensité de la non linéarité (i.e. du paramètre α) au sein des sous-domaines. En termes de nombres d'itérations de Krylov, les gains varient de 13

FIGURE 4.3 – Nonlinear thermal problem : $\alpha = 0.8$

# d'itérations de Newton globales					
α	BDD-NL	FETI-NL	FETI-precNL	Gains FETI-precNL (%)	
				vs. BDD-NL	vs. FETI-NL
0	1	1	1	0	0
0.2	3	4	3	0	25
0.4	4	5	3	25	40
0.6	4	5	3	25	40
0.8	5	6	3	40	40
# d'itérations cumulées de Krylov					
α	BDD-NL	FETI-NL	FETI-precNL	Gains FETI-precNL (%)	
				vs. BDD-NL	vs. FETI-NL
0	8	8	8	0	0
0.2	24	32	25	-4	22
0.4	32	39	28	13	28
0.6	34	40	28	18	30
0.8	43	40	28	35	30

TABLE 4.7 – Évolution thermique non linéaire : itérations de Newton globales et du solveur de Krylov

à 35%, tandis que le coût du preconditionneur non linéaire n'excède pas un facteur 3 sur les nombres d'itérations locales.

# d'itérations de Newton locales					
α	BDD-NL	FETI-NL	FETI-precNL	Ratios FETI-precNL (%)	
				sur BDD-NL	sur FETI-NL
0	1	1	3	3.0	3.0
0.2	7	11	19	2.71	1.73
0.4	10	12	22	2.20	1.83
0.6	11	16	26	2.36	1.63
0.8	12	17	32	2.67	1.88

TABLE 4.8 – Évolution thermique non linéaire : itérations locales

4 Conclusion

Ce chapitre investigate une nouvelle technique de préconditionnement pour le solveur FETI, dans le contexte de la méthode de sous-structuration et condensation non linéaires avec approche duale (ou solveur FETI-NL). Une version non linéaire du préconditionneur classique de Dirichlet pondéré est construite sous la forme d'un système de recherche de point fixe condensé à l'interface, remplaçant le problème d'interface non linéaire classiquement condensé de la méthode FETI-NL. Un algorithme de Newton global est appliqué pour résoudre ce nouveau système, dit *non linéairement préconditionné*, et les systèmes tangents ont la forme exacte de problèmes FETI préconditionnés projetés, assurant ainsi un conditionnement de qualité et une régularité suffisante des problèmes linéaires. Le processus de résolution ainsi défini est dénommé FETI-precNL.

Les résultats font état d'une performance bien supérieure du solveur FETI-precNL, comparé aux solveurs BDD-NL et FETI-NL, sur deux cas-tests numériques. Le premier consiste en une étude de diffusion d'eau dans une colonne de sol (ne faisant pas intervenir de modes rigides au sein du processus de résolution), et le second suit l'évolution thermique non linéaire d'un domaine rectangulaire. Pour chacun de ces exemples, la méthode FETI-precNL a clairement permis d'augmenter la vitesse de convergence, en termes de nombres d'itérations de Newton globales et de Krylov : les gains sont d'environ 50% pour le premier cas-test, et de 30% pour le second.

Ces premiers résultats sont encourageants et permettent d'envisager à la fois une implémentation à plus grande échelle, et une extension de la méthode à des problèmes à inconnue vectorielle (plasticité par exemple), où un soin spécial devra être apporté à la possible présence de discontinuités importantes aux coins des sous-domaines lors des résolutions locales avec conditions de Dirichlet artificiellement rendues continues entre les sous-domaines. De plus, étant données les meilleures performances de la stratégie de préconditionnement non linéaire sur un problème n'impliquant pas de modes rigides, un couplage avec une méthode de décomposition de domaine *hybride* [Gosselet and Rey, 2006], où le choix du couple : déplacement tangentiel, effort normal en tant qu'inconnue d'interface permet d'éliminer les modes rigides, pourrait per-

mettre une performance accrue sur des problèmes mécaniques.

Conclusion et perspectives

Les quatre chapitres de ce manuscrit ont cherché à apporter une réponse à deux objectifs étroitement liés :

- la robustification des solveurs non linéaires parallèles
- la diminution de la quantité d'information échangée entre les cœurs impliqués dans le processus de résolution

Cette réponse – non exhaustive – a été construite en s'appuyant sur une formulation existante favorisant le caractère local des opérations non linéaires au cours du processus de résolution, et s'étendant au-delà de ce cadre à partir de deux idées majeures : la première s'axe autour de l'optimisation d'un paramètre introduit dans la formulation du solveur, permettant d'accélérer la vitesse de convergence, tandis que la deuxième opère une refonte du processus de résolution permettant d'améliorer ses performances théoriques, sans toutefois complexifier sa technicité d'implémentation.

Dans un premier temps, les fondements et le cadre théoriques des solveurs non linéaires parallèles, et plus précisément de la méthode de sous-structuration et condensation non linéaires, ont été développés dans le détail. Ce processus de résolution consiste à découper le problème non linéaire à résoudre (équation aux dérivées partielles dans un domaine muni de conditions aux limites) en sous-problèmes définis à partir d'une sous-structuration du domaine physique (afin de conserver un découplage de la majorité des degrés de liberté, et donc la structure creuse des opérateurs discrétisés par la méthode des éléments finis). Sur les degrés de liberté partagés par au moins deux sous-domaines – i.e. sur l'interface globale définie par la sous-structuration, – des conditions de continuité en déplacement et d'équilibre des efforts entre les sous-domaines permettent de définir un problème condensé, tandis que les quantités d'interface sont évaluées localement par les équilibres non linéaires de chaque sous-domaine. Un algorithme de Newton appliqué à ce problème non linéaire condensé à l'interface globale – portant sur l'assemblage des quantités d'interface locales – produit une suite de problèmes tangents, dont la résolution (linéaire) est effectuée par une méthode de Décomposition de Domaine classique, s'appuyant sur la même sous-structuration que celle du problème non linéaire.

Trois types de conditions de transmission aux interfaces ont permis de définir trois approches pour la méthode de sous-structuration et condensation non linéaires : la première, ou approche primale, choisit pour inconnue le déplacement, continu aux interfaces, la deuxième, ou approche duale, choisit pour inconnue la réaction nodale, équilibrée aux interfaces, la troisième enfin, ou approche mixte, définit l'inconnue d'interface comme une combinaison linéaire des déplacements et efforts, homogène à une réaction nodale. Dans le cas primal, le résidu d'interface non linéaire est constitué du déséquilibre des efforts, dans le cas dual du saut de déplacement ; dans le cas mixte il provient d'une combinaison des deux.

Les performances de la méthode ont été comparées à celles d'un processus de résolution plus classique, dit NKS, consistant à appliquer un algorithme de Newton au problème global non linéaire (non sous-structuré), et une méthode DD aux problèmes linéarisés à résoudre à chaque itération globale – les équilibres locaux sont alors linéaires, quelle que soit la nature des phénomènes à l'œuvre localement. La comparaison est effectuée principalement sur les nombres d'itérations du solveur tangent, ce dernier comptabilisant la quasi-totalité des opérations de communication du solveur. Le coût de la méthode est évalué par la comparaison des nombres d'opérations locales (ne nécessitant pas de communications entre les processeurs), plus importants a priori avec une sous-structuration et condensation non linéaires – les équilibres locaux sont alors non linéaires.

De manière générale, la comparaison entre ces deux solveurs montre que sur tous les cas-tests étudiés dans ce manuscrit, la sous-structuration et condensation du problème non linéaire permet de construire un « meilleur » problème d'interface, et ainsi de diminuer le nombre d'itérations de Newton globales à convergence du solveur : le nombre de problèmes tangents à résoudre diminue ainsi d'autant, résultant en une décroissance des nombres d'itérations du solveur DD (de Krylov) tangent, et donc des communications, au prix d'un nombre « raisonnable » d'itérations locales additionnelles. Cette amélioration des performances des processus de résolution d'équations non linéaires, grâce à une augmentation de la rapidité des solveurs, permet donc de traiter des problèmes de plus grande taille dans des laps de temps réduits, ou de complexifier les lois de comportement matériau, d'augmenter la précision sur la solution, de raffiner le maillage... tous ces éléments participant de la robustesse du solveur.

En ce qui concerne l'influence du type d'approche choisi, les conditions d'interface primales et mixtes ont semblé être les plus intéressantes en termes de réduction des communications, comparées aux conditions d'interface duales qui présentent, dans le contexte mécanique d'études de structures au comportement élasto-plastique, le désavantage d'être fortement tributaires des coefficients d'écrouissage du matériau.

Dans un deuxième temps, les conditions de transmission aux interfaces de type mixte ont été étudiées en détail, et notamment l'optimisation du paramètre introduit dans l'inconnue d'interface mixte, définie comme une combinaison linéaire des déplacements et efforts. Ce paramètre, défini localement, est homogène à une rigidité – ou impédance – d'interface, dans le cas linéaire élastique notamment sa valeur optimale est connue (bien que son coût de calcul soit inacceptable en pratique dans le

contexte distribué), et représente, pour une sous-structure donnée, la discrétisation de l'opérateur de Steklov-Poincaré *Dirichlet-to-Neumann* du complémentaire de cette sous-structure. À partir d'une étude préliminaire du cas linéaire, une nouvelle heuristique a été développée dans le cas non linéaire, permettant de construire à moindre coût une impédance d'interface prenant en compte à la fois la structure proche de chaque sous-domaine, mais également la partie plus lointaine du complémentaire, le tout regroupé dans une unique expression matricielle.

La méthode de sous-structuration et condensation non linéaires avec approche mixte et optimisation de l'impédance d'interface a été comparée à cette même méthode, pour d'autres expressions de l'impédance d'interface, ainsi qu'avec le processus plus classique de résolution NKS. Dans les deux cas, le principal constat fait état, à nouveau, d'une diminution des communications au prix de quelques itérations locales supplémentaires.

Enfin, la dernière partie de ce manuscrit s'est intéressée à l'étape de préconditionnement du solveur, intervenant jusqu'ici au niveau tangent, au sein des méthodes DD. Un nouveau procédé non linéaire de préconditionnement a été développé, à partir d'une vision des préconditionneurs classiques DD en tant que systèmes de recherche de point fixe d'opérateurs non contractants. Le problème non linéaire condensé à l'interface est dans ce contexte modifié, de sorte que les problèmes tangents produits par l'algorithme de Newton global qui lui est appliqué soient intrinsèquement bien conditionnés. Là encore, la construction d'un « meilleur » problème d'interface permet de diminuer le nombre d'itérations globales à convergence, et ainsi les communications.

Les performances des différentes méthodes développées au cours de cette thèse ont toutes été évaluées sur des cas-tests très « académiques ». Ainsi, la taille des problèmes étudiés est-elle bien en-deçà de celle des problèmes sur lesquels les solveurs parallèles sont réellement avantageux, comparés à des résolutions directes avec *multithreading*. Seules les comparaisons des nombres d'itérations ont donc été retenues en guise d'indicateurs de performance, bien qu'une étude en termes de temps CPU (ou encore en termes de temps humain : *wall-clock time*) soit le juge de paix final. Une implémentation à plus grande échelle sera donc nécessaire pour évaluer les performances réelles des nouveaux solveurs présentés ici, ainsi que pour découvrir les potentiels écueils indécélables sur des cas-tests de petite taille.

De plus, si une diminution des communications a bien été observée, différentes questions restent encore en suspens, notamment celle d'un potentiel mauvais équilibrage de charge (ou *load balancing*) inhérent à la résolution parallèle de problèmes d'intensité non linéaire variable d'un sous-domaine à l'autre. Plusieurs solutions peuvent être envisagées afin d'éviter que les sous-domaines ayant terminé leurs itérations locales non linéaires n'attendent les autres : une extension des méthodes *asynchrones* au cas non linéaire, ou bien une stratégie adaptative visant à procéder par étapes couplée à un processus d'estimation d'erreur, en affinant la précision de la solution à chaque pas, et en redécoupant automatiquement les sous-domaines travaillant plus que les autres.

D'autres types de non linéarité devraient également être testés, tels que des problèmes en grands déplacements (flambement...) pour l'approche mixte avec impédance d'interface optimisée, et des problèmes mécaniques (plasticité...) à inconnue vectorielle concernant le préconditionneur non linéaire. La robustesse du solveur avec préconditionneur non linéaire doit également être encore améliorée, notamment par un potentiel couplage avec une méthode DD aux conditions de transmission hybrides (où l'inconnue d'interface est un couple formé du déplacement tangentiel et de la réaction nodale normale) permettant de s'affranchir de la présence de modes rigides au sein des sous-structures.

Annexe A

A possibility for computing the term $\tilde{F}_{A,1} = \tilde{S}_A^{-1}$ is to resort to a Ritz analysis of the Krylov solver iterations. Indeed, Ritz analysis, combined with the Krylov subspaces theory, performs an approximation of eigenelements of the linear system operator by extracting Krylov coefficients along the iterations. The spectrum of the preconditioner can then be derived at low cost. Let (A.1) be a given linear preconditioned system :

$$M^{-1}Ax = M^{-1}b \quad (\text{A.1})$$

solved by a Krylov solver in m iterations. Krylov subspace of size j associated with left-preconditioned operator $M^{-1}A$ and initial preconditioned residual $z_0 = M^{-1}r_0$ is written :

$$\mathcal{K}_j(M^{-1}A, z_0) = \text{span} \left(z_0, \dots, (M^{-1}A)^{j-1} z_0 \right)$$

The Ritz vectors and values $(Y_m, \Theta_m) = (y_m^i, \theta_m^i)_{0 \leq i < m}$ associated to the solving of problem (A.1) are the solution to a generalized eigenproblem :

$$\begin{aligned} &\text{Find } (y_m^i, \theta_m^i) \in \mathcal{K}_m(M^{-1}A, M^{-1}v_0) \times \mathbb{R} \text{ such that :} \\ &\forall u \in \mathcal{K}_m(M^{-1}A, M^{-1}v_0), \quad u^T (Ay_m^i - \theta_m^i y_m^i) = 0 \end{aligned} \quad (\text{A.2})$$

where v_0 is the M^{-1} -normalized initial residual :

$$\begin{aligned} v_0 &= \frac{r_0}{\|r_0\|_{M^{-1}}} = \frac{r_0}{(r_0, z_0)^{1/2}} \\ \Rightarrow M^{-1}v_0 &= \frac{z_0}{(r_0, z_0)^{1/2}} \end{aligned}$$

Ritz elements are thus defined as the approximation of (A, M) eigenelements in the m -dimensional Krylov subspace $\mathcal{K}_m(M^{-1}A, M^{-1}v_0)$ and satisfy the following relations :

$$\begin{cases} Y_m^T A Y_m = \Theta_m \\ Y_m^T M Y_m = I_m \end{cases} \quad (\text{A.3})$$

Being given the definition of eigenproblem (A.2) – similar to that of conjugate gradient algorithm – Ritz vectors can be expressed in function of the residuals r_j and preconditionned residuals z_j of Krylov solver iterations :

$$Y_m = V_m Q_m \quad (\text{A.4})$$

$$\text{where } V_m = \left[\dots, (-1)^j \frac{z_j}{(r_j, z_j)^{1/2}}, \dots \right]_{0 \leq j < m}$$

$$\text{and } Q_m \text{ is such that } V_m^T A V_m = Q_m \Theta_m Q_m^T \quad (\text{A.5})$$

Y_m is a matrix of size $n^A \times m$ – with n^A the number of degrees of freedom belonging to the primal interface Γ_A , – V_m is orthogonal of size $n^A \times m$ and Q_m is orthonormal of size $m \times m$: $Q_m Q_m^T = I_m$.

Remarque 1 : $V_m^T A V_m$ is a tridiagonal symmetric matrix, whose coefficients can be expressed in function of Krylov coefficients. Its eigenelements $(q_m^j, \theta_m^j)_{0 \leq j < m}$ can thus be easily calculated, using a Lapack procedure for instance. However, computation of Q_m and Θ_m will not be required here.

In the context of nonlinear parallel computations with mixed approach, we are interested in the calculation of eigenelements of $\tilde{S}_A^{-1} = V_F D_F V_F^T$ – with D_F diagonal and V_F orthonormal, – and the preconditionned linear system solved at each global Newton iteration reads :

$$\left(P_A \tilde{A}^\diamond S_t^{\diamond\dagger} \tilde{A}^{\diamond T} P_A^T \right) A^\diamond S_t^\diamond A^{\diamond T} dv_A = P_A^T A^\diamond (Q_b^\diamond + S_t^\diamond) b_m^\diamond$$

The following connection with notations of problem (A.1) can be made :

$$\begin{cases} A = A^\diamond S_t^\diamond A^{\diamond T} \\ b = P_A^T A^\diamond (Q_b^\diamond + S_t^\diamond) b_m^\diamond \\ M^{-1} = P_A \tilde{A}^\diamond S_t^{\diamond\dagger} \tilde{A}^{\diamond T} P_A^T = P_A \tilde{S}_A^{-1} P_A^T \end{cases}$$

We define :

$$\tilde{Y}_m = \left[\dots, \frac{r_j}{(r_j, z_j)^{1/2}}, \dots \right] = M V_m \quad (\text{A.6})$$

From (A.4) and (A.6) the following relations hold :

$$\begin{aligned}
 Y_m &= V_m Q_m = M^{-1} \tilde{Y}_m Q_m \\
 \Rightarrow Y_m^T M Y_m &= Q_m^T \tilde{Y}_m^T M^{-T} M M^{-1} \tilde{Y}_m Q_m \\
 &= Q_m^T \tilde{Y}_m^T M^{-1} \tilde{Y}_m Q_m \\
 \Rightarrow \tilde{Y}_m^T M^{-1} \tilde{Y}_m &= Q_m Y_m^T M Y_m Q_m^T
 \end{aligned} \tag{A.7}$$

From (A.7), (A.3) and orthonormality property of square matrix Q_m , vectors $\tilde{Y}_m$ satisfy the following property :

$$\tilde{Y}_m^T M^{-1} \tilde{Y}_m = I_m \tag{A.8}$$

A QR-factorization $\tilde{Y}_m = \tilde{Q}_m \tilde{R}_m$ – with $\tilde{Q}_m$ orthonormal and $\tilde{R}_m$ triangular – is used to reformulate equation (A.8) :

$$\tilde{Q}_m^T M^{-1} \tilde{Q}_m = \tilde{R}_m^{-T} \tilde{R}_m^{-1}$$

Remarque 2 : Matrix $\tilde{Y}_m$ is of size $n^A \times m$ with m the number of Krylov iterations. Its QR-factorization can be performed using a modified Gram-Schmidt algorithm for example : $O(2n^A m^2)$ operations. If m is too high, a selection procedure of vectors y_m^i – and $\tilde{y}_m^i$ by extension – can be derived from SRKS algorithm [Gosselet et al., 2013] to reject vectors which are not enough good approximations of eigenvectors of A .

Matrix $\tilde{R}_m^{-T} \tilde{R}_m^{-1}$ is symmetric, real-valued of size $m \times m$: its diagonalization can be performed locally at low cost :

$$\begin{aligned}
 \tilde{R}_m^{-T} \tilde{R}_m^{-1} &= P_m D_m P_m^T \\
 \Rightarrow (P_m^T \tilde{Q}_m^T) M^{-1} (\tilde{Q}_m P_m) &= D_m \\
 \text{and } (\tilde{Q}_m P_m)^T \tilde{Q}_m P_m &= I_m
 \end{aligned}$$

Eventually, the wished eigenelements can be approximated :

$$\begin{cases} V_F \simeq \tilde{V}_m \equiv \tilde{Q}_m P_m \\ D_F \simeq D_m \end{cases}$$

Term $\tilde{F}_{A,1}$ can thus be approximated by :

$$\tilde{F}_{A,1} = \tilde{S}_A^{-1} = V_F D_F V_F^T \simeq \tilde{V}_m D_m \tilde{V}_m^T \tag{A.9}$$

In conclusion, we have :

$$Q_{b,G}^{(j)-1} = K_{bb}^{neigh(j)-1} + A^{(j)T} (\tilde{F}_{A,1} + \tilde{F}_{A,2}) A^{(j)} \quad (\text{A.10})$$

$$\simeq K_{bb}^{neigh(j)-1} + A^{(j)T} \left(\tilde{V}_m D_m \tilde{V}_m^T + G_A (G_A^T S_A G_A)^{-1} G_A^T \right) A^{(j)} \quad (\text{A.11})$$

And computation of term :

$$A^{(j)T} \tilde{F}_{A,1} A^{(j)} \simeq A^{(j)T} \tilde{V}_m D_m \tilde{V}_m^T A^{(j)}$$

requires only :

- storage of normalized Krylov residuals $\tilde{Y}_m = \left[\dots, \frac{r_j}{(r_j, z_j)^{1/2}}, \dots \right]_{0 \leq j < m}$
- QR-factorization of $\tilde{Y}_m = \tilde{Q}_m \tilde{R}_m$: size $n^A \times m$
- local diagonalization of $m \times m$ matrix $\tilde{R}_m^{-T} \tilde{R}_m^{-1} = P_m^T D_m P_m^T$
- local products $(A^{(j)T} \tilde{V}_m) D_m$ with $A^{(j)T} V_m$ of size $n^{(j)} \times m$ and D_m of size $m \times m$, $n^{(j)}$ being the number of interface nodes belonging to $\Gamma^{(j)}$

Bibliographie

- [Amestoy et al., 2017] Amestoy, P., Buttari, A., L'Excellent, J.-Y., and Mary, T. (2017). On the complexity of the block low-rank multifrontal factorization. *Methods and Algorithms for Scientific Computing*, 39 :A1710–A1740.
- [Amestoy et al., 1998] Amestoy, P. R., Duff, I. S., and L'Excellent, J.-Y. (1998). MUMPS multifrontal massively parallel solver version 2.0.
- [Badea, 1991] Badea, L. (1991). On the Schwarz alternating method with more than two subdomains for nonlinear monotone problems. *SIAM Journal on Numerical Analysis*, 28(1) :179–204.
- [Barboteu et al., 2001] Barboteu, M., Alart, P., and Vidrascu, M. (2001). A domain decomposition strategy for nonclassical frictional multi-contact problems. *Computer Methods in Applied Mechanics and Engineering*, 190(37) :4785–4803.
- [Bebendorf and Hackbusch, 2003] Bebendorf, M. and Hackbusch, W. (2003). Existence of H-matrix approximants to the inverse FE-matrix of elliptic operators with L8-coefficients. *Numerische Mathematik*, 95(1) :1–28.
- [Bernardi, 1994] Bernardi, C. (1994). A new nonconforming approach to domain decomposition : the mortar element method. *Nonlinear Partial Differential Equations and Their Applications*.
- [Bhardwaj et al., 1999] Bhardwaj, M., Day, D., Farhat, C., Lesoinne, M., Pierson, K., and Rixen, D. (1999). Application of the FETI method to ASCI problems : Scalability results on one thousand processors and discussion of highly heterogeneous problems. Technical report, Sandia National Labs., Albuquerque, NM (US) ; Sandia National Labs., Livermore, CA (US).
- [Blanchard et al., 2017] Blanchard, M., Gosselet, P., Allix, O., and Guguin, G. (2017). Non-invasive global-local coupling as a schwarz domain decomposition method : acceleration and generalization. *in preparation*.

- [Boisse et al., 1989] Boisse, P., Ladevèze, P., and Rougée, P. (1989). A large time increment method for elastoplastic problems. *European journal of mechanics. A. Solids*, 8(4) :257–275.
- [Bonnet et al., 2014] Bonnet, M., Frangi, A., and Rey, C. (2014). *The finite element method in solid mechanics*. McGraw Hill Education.
- [Bordeu et al., 2009] Bordeu, F., Boucard, P.-A., and Gosselet, P. (2009). Balancing domain decomposition with nonlinear relocalization : Parallel implementation for laminates. In *First international conference on parallel, distributed and grid computing for engineering*, pages CCP–90.
- [Boucard and Ladeveze, 1997] Boucard, P. and Ladeveze, P. (1997). Post-buckling analysis using a nonincremental method. In *15th IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics*, volume 3, pages 449–454.
- [Boucard, 2001] Boucard, P.-A. (2001). Application of the LaTIn method to the calculation of response surfaces. In *Proceeding of the First MIT Conference on Computational Fluid and Solid Mechanics, Cambridge, USA*, volume 1, pages 78–81.
- [Boucard and Ladevèze, 1999] Boucard, P.-A. and Ladevèze, P. (1999). Une application de la méthode LaTIn au calcul multirésolution de structures non linéaires. *Revue européenne des éléments finis*, 8(8) :903–920.
- [Bovet et al., 2017] Bovet, C., Parret-Fréaud, A., Spillane, N., and Gosselet, P. (2017). Adaptive multipreconditioned feti : scalability results and robustness assessment.
- [Brenner, 2007] Brenner, S. C. (2007). Lower bounds in domain decomposition. In *Domain Decomposition Methods in Science and Engineering XVI*, pages 27–39. Springer.
- [Bussy et al., 1990] Bussy, P., Rougée, P., and Vauchez, P. (1990). The large time increment method for numerical simulation of metal forming processes. *Proceedings of Numerical Methods in Engineering*, pages 102–109.
- [Cai and Keyes, 2002] Cai, X.-C. and Keyes, D. E. (2002). Nonlinearly preconditioned inexact Newton algorithms. *SIAM Journal on Scientific Computing*, 24(1) :183–200.
- [Cai et al., 2002] Cai, X.-C., Keyes, D. E., and Marcinkowski, L. (2002). Non-linear additive Schwarz preconditioners and application in computational fluid dynamics. *International journal for numerical methods in fluids*, 40(12) :1463–1470.
- [Cai and Sarkis, 1999] Cai, X.-C. and Sarkis, M. (1999). A restricted additive Schwarz preconditioner for general sparse linear systems. *Siam journal on scientific computing*, 21(2) :792–797.

- [Celia et al., 1990] Celia, M. A., Bouloutas, E. T., and Zarba, R. L. (1990). A general mass-conservative numerical solution for the unsaturated flow equation. *Water resources research*, 26(7) :1483–1496.
- [Chan and Mathew, 1994] Chan, T. F. and Mathew, T. P. (1994). Domain decomposition algorithms. *Acta numerica*, 3 :61–143.
- [Chapman and Saad, 1997] Chapman, A. and Saad, Y. (1997). Deflated and augmented krylov subspace techniques. *Numerical linear algebra with applications*, 4(1) :43–66.
- [Ciarlet, 2013] Ciarlet, P. G. (2013). *Linear and nonlinear functional analysis with applications*, volume 130. Siam.
- [Cognard, 1989] Cognard, J.-Y. (1989). *Une nouvelle approche des problèmes de plasticité et de viscoplasticité : la méthode à grand incrément de temps*. PhD thesis, Paris 6.
- [Cognard and Ladevèze, 1993] Cognard, J.-Y. and Ladevèze, P. (1993). A large time increment approach for cyclic viscoplasticity. *International Journal of plasticity*, 9(2) :141–157.
- [Conn et al., 2000] Conn, A. R., Gould, N. I., and Toint, P. L. (2000). *Trust region methods*, volume 1. Siam.
- [Cresta, 2008] Cresta, P. (2008). *Décomposition de domaine et stratégies de relocalisation non-linéaire pour la simulation de grandes structures raidies avec flambage local*. PhD thesis, École normale supérieure de Cachan-ENS Cachan.
- [Cresta et al., 2007] Cresta, P., Allix, O., Rey, C., and Guinard, S. (2007). Nonlinear localization strategies for domain decomposition methods : Application to post-buckling analyses. *Computer Methods in Applied Mechanics and Engineering*, 196(8) :1436–1446.
- [Davis, 2006] Davis, T. A. (2006). *Direct methods for sparse linear systems*, volume 2. Siam.
- [De Roeck et al., 1992] De Roeck, Y.-H., Le Tallec, P., and Vidrascu, M. (1992). A domain-decomposed solver for nonlinear elasticity. *Computer Methods in Applied Mechanics and Engineering*, 99(2-3) :187–207.
- [Dembo et al., 1982] Dembo, R. S., Eisenstat, S. C., and Steihaug, T. (1982). Inexact Newton methods. *SIAM Journal on Numerical analysis*, 19(2) :400–408.
- [Desmeure et al., 2011] Desmeure, G., Gosselet, P., Rey, C., and Cresta, P. (2011). Etude de différentes représentations des interefforts dans une stratégie de décomposition de domaines mixte. In *10e colloque national en calcul des structures*, pages Clé–USB.

- [Dolean et al., 2016] Dolean, V., Gander, M. J., Kwok, F., Masson, R., and Kheriji, W. (2016). Nonlinear preconditioning : How to use a nonlinear Schwarz method to precondition Newton’s method. *arXiv preprint arXiv :1605.04419*.
- [Dolean et al., 2015] Dolean, V., Jolivet, P., and Nataf, F. (2015). *An introduction to domain decomposition methods : algorithms, theory, and parallel implementation*. SIAM.
- [Dryja and Widlund, 1990] Dryja, M. and Widlund, O. B. (1990). *Some domain decomposition algorithms for elliptic problems*. Academic Press Professional, Inc.
- [Duff et al., 1986] Duff, I. S., Erisman, A. M., and Reid, J. K. (1986). *Direct methods for sparse matrices*. Clarendon press Oxford.
- [Duval et al., 2016] Duval, M., Passieux, J.-C., Salaün, M., and Guinard, S. (2016). Non-intrusive coupling : recent advances and scalable nonlinear domain decomposition. *Archives of Computational Methods in Engineering*, 23(1) :17–38.
- [Eisenstat and Walker, 1994] Eisenstat, S. C. and Walker, H. F. (1994). Globally convergent inexact Newton methods. *SIAM Journal on Optimization*, 4(2) :393–422.
- [Eisenstat and Walker, 1996] Eisenstat, S. C. and Walker, H. F. (1996). Choosing the forcing terms in an inexact Newton method. *SIAM Journal on Scientific Computing*, 17(1) :16–32.
- [Erdos, 1978] Erdos, J. (1978). *Hilbert Space Methods for Partial Differential Equations (Monographs and Studies in Mathematics Vol 1)*.
- [Farhat, 1994] Farhat, C. (1994). Implicit parallel processing in structural mechanics. *Compt. Mech. Adv.*, 2 :1–124.
- [Farhat et al., 1993] Farhat, C., Chen, P.-S., and Roux, F.-X. (1993). The dual Schur complement method with well-posed local Neumann problems : regularization with a perturbed Lagrangian formulation. *SIAM Journal on Scientific Computing*, 14(3) :752–759.
- [Farhat et al., 2001] Farhat, C., Lesoinne, M., LeTallec, P., Pierson, K., and Rixen, D. (2001). FETI-DP : a dual–primal unified FETI method—part I : A faster alternative to the two-level FETI method. *International journal for numerical methods in engineering*, 50(7) :1523–1544.
- [Farhat et al., 2000a] Farhat, C., Lesoinne, M., and Pierson, K. (2000a). A scalable dual-primal domain decomposition method. *Numerical linear algebra with applications*, 7(7-8) :687–714.

- [Farhat et al., 1994] Farhat, C., Mandel, J., and Roux, F. X. (1994). Optimal convergence properties of the FETI domain decomposition method. *Computer methods in applied mechanics and engineering*, 115(3) :365–385.
- [Farhat et al., 2000b] Farhat, C., Pierson, K., and Lesoinne, M. (2000b). The second generation FETI methods and their application to the parallel solution of large-scale linear and geometrically non-linear structural analysis problems. *Computer methods in applied mechanics and engineering*, 184(2) :333–374.
- [Farhat and Roux, 1991] Farhat, C. and Roux, F.-X. (1991). A method of finite element tearing and interconnecting and its parallel solution algorithm. *International Journal for Numerical Methods in Engineering*, 32(6) :1205–1227.
- [Feistauer and Ženíšek, 1986] Feistauer, M. and Ženíšek, A. (1986). Finite element solution of nonlinear elliptic problems. *Numerische Mathematik*, 50(4) :451–475.
- [Feyel and Chaboche, 2000] Feyel, F. and Chaboche, J.-L. (2000). Fe 2 multiscale approach for modelling the elastoviscoplastic behaviour of long fibre sic/ti composite materials. *Computer methods in applied mechanics and engineering*, 183(3) :309–330.
- [Gander, 2006] Gander, M. J. (2006). Optimized schwarz methods. *SIAM Journal on Numerical Analysis*, 44(2) :699–731.
- [Gander and Dubois, 2015] Gander, M. J. and Dubois, O. (2015). Optimized schwarz methods for a diffusion problem with discontinuous coefficient. *Numerical Algorithms*, 69(1) :109–144.
- [Gander et al., 2008] Gander, M. J. et al. (2008). Schwarz methods over the course of time. *Electron. Trans. Numer. Anal*, 31(5) :228–255.
- [Gander and Kwok, 2011] Gander, M. J. and Kwok, F. (2011). Optimal interface conditions for an arbitrary decomposition into subdomains. *Domain Decomposition Methods in Science and Engineering XIX*, 78 :101–108.
- [Gander and Zhang, 2014] Gander, M. J. and Zhang, H. (2014). Optimized interface preconditioners for the FETI method. In *Domain Decomposition Methods in Science and Engineering XXI*, pages 657–665. Springer.
- [Gendre et al., 2011] Gendre, L., Allix, O., and Gosselet, P. (2011). A two-scale approximation of the schur complement and its use for non-intrusive coupling. *International Journal for Numerical Methods in Engineering*, 87(9) :889–905.
- [Gendre et al., 2009] Gendre, L., Allix, O., Gosselet, P., and Comte, F. (2009). Non-intrusive and exact global/local techniques for structural problems with local plasticity. *Computational Mechanics*, 44(2) :233–245.

- [Godlewski, 1980] Godlewski, E. (1980). *Méthodes à pas multiples et de directions alternées pour la discrétisation d'équations d'évolution*. PhD thesis, Thèse de 3^e cycle, Université Pierre et Marie Curie, Chichester.
- [Gosselet and Rey, 2006] Gosselet, P. and Rey, C. (2006). Non-overlapping domain decomposition methods in structural mechanics. *Archives of computational methods in engineering*, 13(4) :515–572.
- [Gosselet et al., 2002] Gosselet, P., Rey, C., Léné, F., and Dasset, P. (2002). A domain decomposition method for quasi-incompressible formulations with discontinuous pressure field : Application to the mechanical study of a flexible bearing. *Revue Européenne des Eléments*, 11(2-4) :363–377.
- [Gosselet et al., 2013] Gosselet, P., Rey, C., and Pebrel, J. (2013). Total and selective reuse of Krylov subspaces for the resolution of sequences of nonlinear structural problems. *International Journal for Numerical Methods in Engineering*, 94(1) :60–83.
- [Gould and Leyffer, 2003] Gould, N. I. and Leyffer, S. (2003). An introduction to algorithms for nonlinear optimization. In *Frontiers in numerical analysis*, pages 109–197. Springer.
- [Guidault et al., 2008] Guidault, P.-A., Allix, O., Champaney, L., and Cornuault, C. (2008). A multiscale extended finite element method for crack propagation. *Computer Methods in Applied Mechanics and Engineering*, 197(5) :381–399.
- [Hadamard, 1902] Hadamard, J. (1902). Sur les problèmes aux dérivées partielles et leur signification physique. *Princeton university bulletin*, 13(49-52) :28.
- [Haferssas et al., 2015] Haferssas, R., Jolivet, P., and Nataf, F. (2015). An additive schwarz method type theory for lions' algorithm and optimized schwarz methods.
- [Hauer, 2015] Hauer, D. (2015). The p-Dirichlet-to-Neumann operator with applications to elliptic and parabolic problems. *Journal of Differential Equations*, 259(8) :3615–3655.
- [Hauret, 2004] Hauret, P. (2004). *Méthodes numériques pour la dynamique des structures non-linéaires incompressibles à deux échelles*. PhD thesis, Ecole Polytechnique X.
- [Hinojosa et al., 2014] Hinojosa, J., Allix, O., Guidault, P.-A., and Cresta, P. (2014). Domain decomposition methods with nonlinear localization for the buckling and post-buckling analyses of large structures. *Advances in Engineering Software*, 70 :13–24.

- [Ibrahimbegović and Marković, 2003] Ibrahimbegović, A. and Marković, D. (2003). Strong coupling methods in multi-phase and multi-scale modeling of inelastic behavior of heterogeneous structures. *Computer Methods in Applied Mechanics and Engineering*, 192(28) :3089–3107.
- [Japhet et al., 2001] Japhet, C., Nataf, F., and Rogier, F. (2001). The optimized order 2 method : application to convection–diffusion problems. *Future generation computer systems*, 18(1) :17–30.
- [Joly, 1986] Joly, P. (1986). Présentation de synthese des méthodes de gradient conjugué. *RAIRO-Modélisation mathématique et analyse numérique*, 20(4) :639–665.
- [Kahou et al., 2007] Kahou, G. A. A., Kamgnia, E., and Philippe, B. (2007). An explicit formulation of the multiplicative Schwarz preconditioner. *Applied numerical mathematics*, 57(11) :1197–1213.
- [Klawonn et al., 2014] Klawonn, A., Lanser, M., and Rheinbach, O. (2014). Nonlinear FETI-DP and BDDC Methods. *SIAM Journal on Scientific Computing*, 36(2) :A737–A765.
- [Klawonn et al., 2017] Klawonn, A., Lanser, M., Rheinbach, O., and Uran, M. (2017). New nonlinear feti-dp methods based on a partial nonlinear elimination of variables. In *Domain Decomposition Methods in Science and Engineering XXIII*, pages 207–215. Springer.
- [Klawonn and Widlund, 2001a] Klawonn, A. and Widlund, O. B. (2001a). Dual and dual-primal FETI methods for elliptic problems with discontinuous coefficients in three dimensions. *Domain Decomposition Methods in Sciences and Engineering (Chiba, 1999)*, DDM. org, Augsburg, Germany, pages 29–39.
- [Klawonn and Widlund, 2001b] Klawonn, A. and Widlund, O. B. (2001b). FETI and Neumann-Neumann iterative substructuring methods : connections and new results. *Communications on pure and applied Mathematics*, 54(1) :57–90.
- [Koester et al., 1994] Koester, D. A., Mahadevan, R., and Markus, K. W. (1994). Multi-User MEMS Processes (MUMPs) Introduction and Design Rules. *available from MCNC MEMS Technology Applications Center*, 3021 :39.
- [Kollerstrom, 1992] Kollerstrom, N. (1992). Thomas Simpson and ‘Newton’s method of approximation’ : an enduring myth. *The British journal for the history of science*, 25(03) :347–354.
- [Ladevèze, 1985] Ladevèze, P. (1985). Sur une famille d’algorithmes en mécanique des structures. *Comptes-rendus des séances de l’Académie des sciences. Série 2, Mécanique-physique, chimie, sciences de l’univers, sciences de la terre*, 300(2) :41–44.

- [Ladevèze, 1996] Ladevèze, P. (1996). *Mécanique non linéaire des structures : nouvelle approche et méthodes de calcul non incrémentales*. Hermès.
- [Ladevèze, 2012] Ladevèze, P. (2012). *Nonlinear computational structural mechanics : new approaches and non-incremental methods of calculation*. Springer Science & Business Media.
- [Ladevèze and Dureisseix, 1999] Ladevèze, P. and Dureisseix, D. (1999). Une nouvelle stratégie de calcul micro/macro en mécanique des structures. *Comptes Rendus de l'Académie des Sciences-Series IIB-Mechanics-Physics-Astronomy*, 327(12) :1237–1244.
- [Ladevèze et al., 2007] Ladevèze, P., Néron, D., and Gosselet, P. (2007). On a mixed and multiscale domain decomposition method. *Computer Methods in Applied Mechanics and Engineering*, 196(8) :1526–1540.
- [Ladeveze et al., 2010a] Ladeveze, P., Néron, D., and Passieux, J.-C. (2010a). On multiscale computational mechanics with time-space homogenization. *Multiscale methods—Bridging the scales in Science and Engineering*, pages 247–282.
- [Ladeveze and Nouy, 2003] Ladeveze, P. and Nouy, A. (2003). On a multiscale computational strategy with time and space homogenization for structural mechanics. *Computer Methods in Applied Mechanics and Engineering*, 192(28) :3061–3087.
- [Ladeveze et al., 2010b] Ladeveze, P., Passieux, J.-C., and Néron, D. (2010b). The latin multiscale computational method and the proper generalized decomposition. *Computer Methods in Applied Mechanics and Engineering*, 199(21) :1287–1296.
- [Lascaux and Théodor, 1987] Lascaux, P. and Théodor, R. (1987). *Analyse numérique matricielle appliquée à l'art de l'ingénieur*, volume 2. Masson Paris.
- [Le Tallec, 1994] Le Tallec, P. (1994). Domain decomposition methods in computational mechanics. *Computational mechanics advances*, 1(2) :121–220.
- [Lions and Lions, 1989] Lions, P. and Lions, P. (1989). On the Schwarz alternating method, II. *Domain Decomposition Methods for Partial Differential Equations, SIAM, Philadelphia (1988–1989)*, pages 1–42.
- [Lions, 1988] Lions, P.-L. (1988). On the Schwarz alternating method. I. In *First international symposium on domain decomposition methods for partial differential equations*, pages 1–42. Paris, France.
- [Lions, 1990] Lions, P.-L. (1990). On the Schwarz alternating method. III : a variant for nonoverlapping subdomains. In *Third international symposium on domain decomposition methods for partial differential equations*, volume 6, pages 202–223. SIAM Philadelphia, PA.

- [Liu and Nocedal, 1989] Liu, D. C. and Nocedal, J. (1989). On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1-3) :503–528.
- [Logg et al., 2012] Logg, A., Mardal, K.-A., and Wells, G. (2012). *Automated solution of differential equations by the finite element method : The FEniCS book*, volume 84. Springer Science & Business Media.
- [Magoulès et al., 2004] Magoulès, F., Roux, F.-X., and Salmon, S. (2004). Optimal discrete transmission conditions for a nonoverlapping domain decomposition method for the helmholtz equation. *SIAM Journal on Scientific Computing*, 25(5) :1497–1515.
- [Magoulès et al., 2006] Magoulès, F., Roux, F.-X., and Series, L. (2006). Algebraic approximation of dirichlet-to-neumann maps for the equations of linear elasticity. *Computer Methods in Applied Mechanics and Engineering*, 195(29) :3742–3759.
- [Magoulès and Venet, 2016] Magoulès, F. and Venet, C. (2016). Asynchronous iterative sub-structuring methods. *Mathematics and Computers in Simulation*.
- [Mandel, 1993] Mandel, J. (1993). Balancing domain decomposition. *Communications in numerical methods in engineering*, 9(3) :233–241.
- [Mandel et al., 2005] Mandel, J., Dohrmann, C. R., and Tezaur, R. (2005). An algebraic theory for primal and dual substructuring methods by constraints. *Applied numerical mathematics*, 54(2) :167–193.
- [Morgan, 2005] Morgan, R. B. (2005). Restarted block-GMRES with deflation of eigenvalues. *Applied Numerical Mathematics*, 54(2) :222–236.
- [Nataf, 2007] Nataf, F. (2007). Recent developments on optimized schwarz methods. In *Domain Decomposition Methods in Science and Engineering XVI*, pages 115–125. Springer.
- [Negrello et al., 2017] Negrello, C., Gosselet, P., and Rey, C. (2017). A new impedance accounting for short and long range effects in mixed substructured formulations of nonlinear problems.
- [Negrello et al., 2016] Negrello, C., Gosselet, P., Rey, C., and Pebrel, J. (2016). Substructured formulations of nonlinear structure problems–influence of the interface condition. *International Journal for Numerical Methods in Engineering*.
- [Nier, 1998] Nier, F. (1998). Remarques sur les algorithmes de décomposition de domaines. *Séminaire Équations*, 2054 :1–24.
- [O’Leary, 1980] O’Leary, D. P. (1980). The block conjugate gradient algorithm and related methods. *Linear algebra and its applications*, 29 :293–322.

- [Oumaziz et al., 2016] Oumaziz, P., Gosselet, P., and Boucard, P.-A. (2016). A non-intrusive implementation of a mixed domain decomposition method for frictional contact problems. *submitted to Computational Mechanics*.
- [Oumaziz et al.,] Oumaziz, P., Gosselet, P., Boucard, P.-A., and Abbas, M. A parallel non-invasive multi-scale strategy for a mixed domain decomposition method with frictional contact. *In preparation*.
- [Parret-Fréaud et al., 2009] Parret-Fréaud, A., Gosselet, P., and Rey, C. (2009). Error estimation for substructured problems. In *International conference on adaptive modeling and simulation, ADMOS 2009*, pages N–A.
- [Parret-Fréaud et al., 2010] Parret-Fréaud, A., Rey, C., Gosselet, P., and Feyel, F. (2010). Fast estimation of discretization error for FE problems solved by domain decomposition. *Computer Methods in Applied Mechanics and Engineering*, 199(49) :3315–3323.
- [Parret-Fréaud et al., 2016] Parret-Fréaud, A., Rey, V., Gosselet, P., and Rey, C. (2016). Improved recovery of admissible stress in domain decomposition methods—application to heterogeneous structures and new error bounds for feti-dp. *International Journal for Numerical Methods in Engineering*.
- [Passieux et al., 2010] Passieux, J.-C., Ladeveze, P., and Néron, D. (2010). A scalable time–space multiscale domain decomposition method : adaptive time scale separation. *Computational Mechanics*, 46(4) :621–633.
- [Paszyński et al., 2010] Paszyński, M., Pardo, D., and Paszyńska, A. (2010). Parallel multi-frontal solver for p-adaptive finite element modeling of multi-physics computational problems. *Journal of Computational Science*, 1(1) :48–54.
- [Pebrel et al., 2008] Pebrel, J., Rey, C., and Gosselet, P. (2008). A nonlinear dual-domain decomposition method : Application to structural problems with damage. *International Journal for Multiscale Computational Engineering*, 6(3).
- [Rey, 1994] Rey, C. (1994). *Développement d’algorithmes parallèles de résolution en calcul non-linéaire de structures hétérogènes : application au cas d’une butée acier élastomère*. PhD thesis.
- [Rey, 1996] Rey, C. (1996). An acceleration technique for the solution of non-linear elasticity problems by domain decomposition. *Comptes rendus de l’Académie des Sciences Série II : Fascicule B - Mécanique, Physique, Chimie, Astronomie*, 322(8) :601–606.
- [Risler and Rey, 2000] Risler, F. and Rey, C. (2000). Iterative accelerating algorithms with Krylov subspaces for the solution to large-scale nonlinear problems. *Numerical algorithms*, 23(1) :1–30.

- [Rixen and Farhat, 1999] Rixen, D. J. and Farhat, C. (1999). A simple and efficient extension of a class of substructure based preconditioners to heterogeneous structural mechanics problems. *International Journal for Numerical Methods in Engineering*, 44(4) :489–516.
- [Roux, 2009] Roux, F.-X. (2009). A FETI-2LM method for non-matching grids. In *Domain Decomposition Methods in Science and Engineering XVIII*, pages 121–128. Springer.
- [Roux et al., 2002] Roux, F.-X., Magoulès, F., Salmon, S., and Series, L. (2002). 29. optimization of interface operator based on algebraic approach. *Domain Decomposition Methods in Science and Engineering*, page 297.
- [Royer, 1990] Royer, C. (1990). *Une approche des problèmes de dynamique non-linéaires par la méthode à grand incrément de temps*. PhD thesis, Paris 6.
- [Saad, 1987] Saad, Y. (1987). On the Lanczos method for solving symmetric linear systems with several right-hand sides. *Mathematics of computation*, 48(178) :651–662.
- [Saad, 1997] Saad, Y. (1997). Analysis of augmented krylov subspace methods. *SIAM Journal on Matrix Analysis and Applications*, 18(2) :435–449.
- [Saad, 2003] Saad, Y. (2003). *Iterative methods for sparse linear systems*. Siam.
- [Saavedra et al., 2012] Saavedra, K., Allix, O., and Gosselet, P. (2012). On a multiscale strategy and its optimization for the simulation of combined delamination and buckling. *International Journal for Numerical Methods in Engineering*, 91(7) :772–798.
- [Saavedra et al., 2017] Saavedra, K., Allix, O., Gosselet, P., Hinojosa, J., and Viard, A. (2017). An enhanced nonlinear multi-scale strategy for the simulation of buckling and delamination on 3d composite plates. *Computer Methods in Applied Mechanics and Engineering*, 317 :952–969.
- [Schenk et al., 2001] Schenk, O., Gärtner, K., Fichtner, W., and Stricker, A. (2001). PARDISO : a high-performance serial and parallel sparse linear solver in semiconductor device simulation. *Future Generation Computer Systems*, 18(1) :69–78.
- [Schwarz, 1869] Schwarz, H. (1869). Ueber einige abbildungsaufgaben. *Journal für die reine und angewandte Mathematik*, 70 :105–120.
- [Showalter, 2013] Showalter, R. E. (2013). *Monotone operators in Banach space and nonlinear partial differential equations*, volume 49. American Mathematical Soc.
- [Smith et al., 2004] Smith, B., Bjorstad, P., and Gropp, W. (2004). *Domain decomposition : parallel multilevel methods for elliptic partial differential equations*. Cambridge university press.

- [Sochala, 2008] Sochala, P. (2008). *Méthodes numériques pour les écoulements souterrains et couplage avec le ruissellement*. PhD thesis, École nationale des ponts et chaussées (France).
- [Soodhalter, 2013] Soodhalter, K. (2013). Mathematics and implementation details of a block MINRES algorithm. *Review by Journal*.
- [Spillane, 2014] Spillane, N. (2014). *Méthodes de décomposition de domaine robustes pour les problèmes symétriques définis positifs*. PhD thesis, Paris 6.
- [Spillane and Rixen, 2013] Spillane, N. and Rixen, D. J. (2013). Automatic spectral coarse spaces for robust finite element tearing and interconnecting and balanced domain decomposition algorithms. *International Journal for Numerical Methods in Engineering*, 95(11) :953–990.
- [St-Cyr et al., 2007] St-Cyr, A., Gander, M. J., and Thomas, S. J. (2007). Optimized multiplicative, additive, and restricted additive schwarz preconditioning. *SIAM Journal on Scientific Computing*, 29(6) :2402–2425.
- [Toselli and Widlund, 2005] Toselli, A. and Widlund, O. B. (2005). *Domain decomposition methods : algorithms and theory*, volume 34. Springer.

Titre : Méthodes fortement parallèles pour la simulation numérique en mécanique non linéaire des structures

Mots-clés : mécanique non linéaire, décomposition de domaine, conditions d'interface mixtes, pré-conditionneur non linéaire

Résumé : Cette thèse vise à contribuer à l'adoption du virtual testing, pratique industrielle encore embryonnaire qui consistera à optimiser et certifier par la simulation numérique le dimensionnement de pièces industrielles critiques. Le virtual testing permettra des économies colossales dans la conception des pièces mécaniques et un plus grand respect de l'environnement grâce à des designs optimisés. Afin d'atteindre un tel objectif, de nouvelles méthodes de calcul doivent être mises en place, plus sûres, plus respectueuses des architectures matérielles, plus rapides, compatibles avec les contraintes temporelles de l'ingénierie. Nous nous intéressons à la résolution parallèle de

problèmes non linéaires de grande taille par des méthodes de décomposition de domaine. Notre objectif est d'atteindre une approximation de la solution exacte en minimisant les communications entre les sous-domaines. Pour cela nous souhaitons maximiser les calculs réalisés indépendamment par sous-domaine à l'aide d'approches de relocalisation non linéaire, contrôler les critères de convergence des solveurs imbriqués de manière à éviter la sur-résolution et les divergences, améliorer la construction de conditions d'interface mixtes, et non linéariser l'étape de préconditionnement du solveur. L'objectif à terme étant de traiter des problèmes de complexité industrielle, la robustesse des méthodes sera un souci constant.

Title : Highly parallel methods for numerical simulation in nonlinear structural mechanics

Keywords : nonlinear mechanics, domain decomposition, mixed interface conditions, nonlinear preconditioner

Abstract :

This thesis is aimed to contribute to the adoption of virtual testing, an industrial practice still embryonic which consists in optimizing and certifying by numerical simulations the dimensioning of critical industrial structures. The virtual testing will allow colossal savings in the design of mechanical parts and a greater respect for the environment, thanks to optimized designs. In order to achieve this goal, new calculation methods must be implemented, satisfying more requirements concerning safety, respect for hardware architectures, fastness, and compatibility with the time constraints of engineering. We are interested in the parallel resolution of

large nonlinear problems by domain decomposition methods. Our goal is to approximate the exact solution by minimizing communication between subdomains. In order to do this, we want to maximize the computations performed independently by subdomain, using nonlinear relocation approaches. We also try to control the convergence criteria of the nested solvers in order to avoid over-resolution and divergences, to improve the construction of conditions Of mixed interface, and non-linearizing the preconditioning step of the solver. The ultimate objective being to deal with problems of industrial complexity, the robustness of the methods we develop will be a constant concern.

