

Méthodes de décomposition de domaine

Ecole thématique CNRS "Visualisation scientifique"

Aussois 9 - 13 mai 2005

Xavier Vasseur *

Résumé

En calcul scientifique, les méthodes de décomposition de domaine sont désormais couramment utilisées lors de la résolution de grands systèmes linéaires ou non-linéaires provenant de la discrétisation d'équations aux dérivées partielles. L'objectif de ce document est de présenter synthétiquement deux algorithmes courants et de détailler les aspects numériques mis en jeu.

Mots-clefs : Balancing Neumann Neumann, calcul parallèle, équation aux dérivées partielles, FETI, méthode de décomposition de domaine, méthodes de résolution directes et itératives, préconditionnement, résolution de grands systèmes linéaires.

1 Introduction

1.1 Contexte général

De nombreuses applications en sciences de l'ingénieur nécessitent la résolution de systèmes non-linéaires de grande taille. La recherche d'une méthode de résolution à la fois **efficace**, **robuste** et **adaptée** aux architectures modernes d'ordinateurs constitue donc une quête fondamentale. L'efficacité en temps de calcul est liée directement au nombre d'opérations nécessaires - ou complexité - permettant l'obtention de la solution du problème donné. Cette complexité est donnée généralement sous la forme n^α avec n la taille du système linéaire considéré [2, 3, 10]. La robustesse est équivalente à être capable de résoudre une large gamme de problèmes tout en obtenant des performances similaires.

Les simulations numériques tridimensionnelles et instationnaires induisent un large volume de données rendant les méthodes de résolution directes pénalisantes au niveau de l'emplacement mémoire nécessaire. Il est également à noter que les systèmes linéaires proviennent d'une discrétisation d'un problème continu. Résoudre le système linéaire à une précision plus élevée que l'erreur de discrétisation s'avère donc inutile. Ces deux principales raisons conduisent à l'emploi de méthodes de résolution itératives de type gradient conjugué ou méthode de Krylov [3, 9]. Les performances des méthodes itératives dépendent en grande partie des propriétés spectrales de la matrice et notamment de la répartition des valeurs propres. En cas de mauvais conditionnement de la matrice, les méthodes itératives classiques peinent à converger, stagnent ou éventuellement divergent. Il est alors nécessaire d'employer un préconditionnement i.e. un opérateur qui transforme le système linéaire initial en un autre possédant la même solution mais offrant un meilleur conditionnement et donc plus propice à l'utilisation de méthodes itératives [4]. Si possible, le préconditionnement doit être optimal i.e. ses performances doivent être indépendantes de la taille du problème. Enfin, dans le cas d'applications parallèles, ses performances ne doivent pas se dégrader lorsque le nombre de processeurs croît.

Une classe de préconditionnement, bien adaptée au calcul parallèle, est issue de l'idée de décomposition de domaine. Ces méthodes sont basées généralement sur un découplage de l'équation aux dérivées partielles,

*Seminar for Applied Mathematics, ETH Zürich, Rämistrasse 101, CH-8092 Zürich, Suisse. E-mail: vasseur@math.ethz.ch.

ou de son approximation, en des sous-problèmes couplés définis sur des domaines de taille réduite formant une partition du domaine global. Cette décomposition peut se faire au niveau continu, au niveau discret ou lors de la résolution du système linéaire global. L'idée sous-jacente consiste à résoudre des sous-problèmes sur des domaines variés, tout en imposant des conditions de continuité entre les domaines adjacents, jusqu'à ce que les solutions locales convergent vers la véritable solution globale. Les principaux avantages sont le parallélisme par nature et la possibilité d'un traitement local de domaines complexes ou irréguliers.

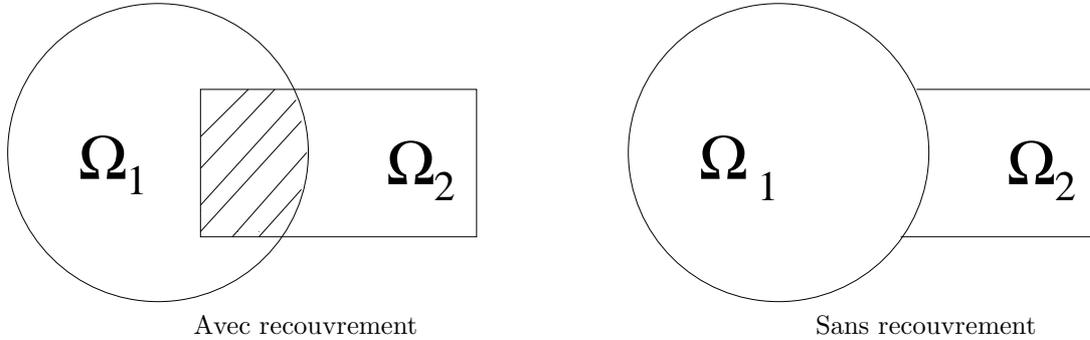


FIG. 1: Exemple de méthodes de décomposition de domaine avec recouvrement (gauche) et sans recouvrement (droite). La zone commune est hachurée.

Synthétiquement les méthodes de décomposition de domaine peuvent se scinder en deux grandes familles :

Les méthodes avec recouvrement ou méthodes de Schwarz. Une illustration est donnée figure 1 (gauche) pour le cas d'une décomposition en deux sous-domaines. Le domaine global est divisé en régions avec recouvrement (overlapping) et des problèmes locaux de type Dirichlet sont résolus sur chaque sous-domaine. Le couplage entre les solutions des différents sous-domaines est assuré par la région commune dite de recouvrement. Ces méthodes ont été originellement proposées par Schwarz en 1870 pour démontrer l'existence de solutions à des problèmes elliptiques sur des domaines complexes.

Les méthodes sans recouvrement ou méthodes du complément de Schur. Une illustration est donnée figure 1 (droite) dans le cas d'une décomposition en deux sous-domaines. Ces méthodes proviennent de la communauté mécanique des structures (elles sont aussi appelées méthodes de sous-structuration) et sont liées à des travaux initiaux sur les méthodes directes de résolution de systèmes linéaires [27]. La partition du domaine étant donnée, il est alors possible de scinder les inconnues en deux sous-ensembles : un formé par les inconnues situées sur l'interface et l'autre par les inconnues situées à l'intérieur des sous-domaines. Après formation du complément de Schur par rapport aux variables intérieures, un système portant sur les variables d'interface est résolu en premier lieu puis les inconnues intérieures sont déduites.

Les deux familles peuvent se généraliser au cas d'un nombre de sous-domaines important. Toutefois la vitesse de convergence de ces algorithmes se détériorera avec le nombre de sous-domaines. En effet, le seul mécanisme de partage de l'information est local (par l'interface ou par le recouvrement). Pourtant le domaine de dépendance pour des problèmes elliptiques est global : un mécanisme de transfert global de l'information est donc nécessaire pour rendre l'algorithme "scalable". L'idée revient alors à résoudre un problème approché sur une grille grossière. Par conséquent, les méthodes de décomposition de domaine peuvent être vues comme une procédure à deux échelles, une échelle de grille fine h sur laquelle la solution est cherchée et une échelle de grille grossière H qui procure le mécanisme de couplage global. Le but revient à trouver l'interaction appropriée entre ces deux mécanismes, afin d'obtenir une vitesse de convergence optimale i.e. indépendante de h et H .

Ces méthodes sont particulièrement intéressantes si des méthodes nouvelles ou plus efficaces sont nécessaires, tout en préservant la plupart des structures de données et concepts déjà présents dans des projets logiciels étendus.

1.2 Contexte au sein de l'école "Visualisation scientifique"

Il est reconnu que la complexité de la visualisation scientifique est due à la fois à un volume de données important, à la grande disparité d'échelles mises en jeu et à la complexité géométrique du domaine. De plus l'acte de visualisation est maintenant envisagé au sein même d'une simulation numérique et non plus à convergence. Une méthode de résolution capable de traiter un large volume de données et des domaines complexes est donc nécessaire. Comme vu précédemment, la méthode de décomposition de domaine répond à ces exigences. Elle permet même de coupler différents modèles sur différents sous-domaines [28] permettant ainsi des simulations multi-physiques par exemple.

Notre objectif ici consiste à présenter deux méthodes de décomposition de domaine dites méthodes de Balancing Neumann-Neumann [21] et FETI [16]. Ces méthodes sans recouvrement sont couramment utilisées pour résoudre des problèmes de grande taille dans une grande variété d'applications [6, 16]. Elles conduisent généralement pour des problèmes elliptiques à des nombres de conditionnement de l'opérateur préconditionné P de la forme :

$$\kappa(P) \leq C \left(1 + \log \left(\frac{H}{h} \right) \right)^2$$

où les distances H et h sont relatives aux problèmes grossier et fin respectivement, C est une constante indépendante de H et de h . Ces méthodes répondent en grande partie aux exigences fixées en introduction :

- elles sont *quasi-optimales* : le conditionnement ne dépend que polylogarithmiquement du rapport $\frac{H}{h}$.
- elles sont *robustes* car il a été démontré que la constante C est indépendante des éventuels sauts de coefficient présents dans l'équation aux dérivées partielles résolue.
- elles sont bien *adaptées* au calcul parallèle.

Qui plus est, ces méthodes de résolution permettent d'accroître l'efficacité des phases de résolution au sein d'un code déjà existant, tout en préservant la plupart des structures de données et concepts déjà présents. Cette caractéristique majeure explique l'abondance des applications envisagées¹ jusqu'à présent par exemple en électromagnétisme, acoustique, mécanique des structures et justifie donc en grande partie le choix d'une présentation succincte de ces méthodes.

1.3 Plan

Le problème modèle sera introduit en section 2. Ensuite deux algorithmes de méthodes de décomposition de domaine sans recouvrement seront présentés en sections 3.2 et 3.3. Les aspects pratiques impliqués par ces algorithmes seront détaillés section 4 puis des résultats numériques seront brièvement exposés en section 5. Enfin nous présenterons un bref survol des logiciels disponibles dans le domaine public en section 6.

2 Problème modèle

Nous considérons le problème modèle suivant défini sur un domaine polygonal fermé $\Omega \subset \mathbb{R}^2$:

$$\begin{aligned} -\varepsilon_x \frac{\partial}{\partial x} \left(\rho_x \frac{\partial u}{\partial x} \right) - \varepsilon_y \frac{\partial}{\partial y} \left(\rho_y \frac{\partial u}{\partial y} \right) + c u &= f, & \text{dans } \Omega, \\ u &= u_D, & \text{sur } \partial\Omega \end{aligned} \quad (1)$$

où (ρ_x, ρ_y) sont des coefficients de diffusion réels et positifs, $(\varepsilon_x, \varepsilon_y)$ des constantes réelles et positives et c un coefficient de réaction non-négatif. Ω est choisi comme le carré $(0, 1)^2$, bien que l'analyse théorique soit valable pour des configurations polygonales générales. Par simplicité des conditions de Dirichlet sont considérées, mais des conditions de type Neumann ou mixtes peuvent être également utilisées. Les constantes $(\varepsilon_x, \varepsilon_y)$ peuvent être arbitrairement faibles pour certains problèmes et nous obtenons alors des problèmes singulièrement perturbés. Les coefficients (ρ_x, ρ_y) peuvent être discontinus prenant des valeurs extrêmement différentes selon les sous-domaines. Enfin la fonction scalaire c peut provenir par exemple d'une discrétisation de type différences finies en temps.

¹ Consulter par exemple <http://caswww.colorado.edu/~charbel/Publications.html>

3 Méthodes du complément de Schur

Après discrétisation du problème modèle (1), un système linéaire de la forme

$$A u = b$$

est obtenu avec A symétrique, définie positive. Le nombre de conditionnement de la matrice A peut dans certaines applications être très important. De ce fait, un préconditionnement efficace et robuste est absolument nécessaire. Deux préconditionnements basés à partir de méthodes de décomposition de domaine sont présentés par la suite en sections 3.2 et 3.3.

3.1 Partition du domaine

Les méthodes de sous-structuration reposent sur une partition sans recouvrement du domaine Ω , $\mathcal{T}^{DD} = \{\Omega_i\}$, en sous-domaines ou sous-structures. Soit N le nombre de sous-domaines, H_i le diamètre de Ω_i et $H = \max(H_i)$ le maximum de ces diamètres. Un sous-domaine est dénommé flottant si l'intersection de $\partial\Omega_i$ avec $\partial\Omega$ est vide. Nous définissons également les frontières $\Gamma_i = \partial\Omega_i \setminus \partial\Omega$ et l'interface Γ comme leur union.

3.2 Méthode de Balancing Neumann-Neumann

3.2.1 Dérivation

Après assemblage, la matrice globale A est réordonnée suivant la permutation donnée par la décomposition du domaine. Les variables intérieures aux sous-domaines (sous-ensemble I) sont ordonnées en premier suivies des variables situées sur l'interface Γ (sous-ensemble Γ). De façon similaire, la matrice locale correspondant au sous-domaine i peut être décomposée sous la forme :

$$A^{(i)} = \begin{pmatrix} A_{II}^{(i)} & A_{I\Gamma}^{(i)} \\ A_{\Gamma I}^{(i)} & A_{\Gamma\Gamma}^{(i)} \end{pmatrix}.$$

En premier lieu, les inconnues intérieures sont éliminées par élimination de Gauss par bloc. Le complément de Schur $S = S_{NN}$ par rapport aux variables intérieures est ainsi formé. Le système résultant pour les variables situées sur l'interface Γ s'écrit donc :

$$S_{NN} u_{\Gamma} = g_{\Gamma}. \quad (2)$$

Soit le complément de Schur associé au sous-domaine Ω_i

$$S_i = A_{\Gamma\Gamma}^{(i)} - A_{\Gamma I}^{(i)} A_{II}^{(i)-1} A_{I\Gamma}^{(i)} \quad (3)$$

et le second membre local

$$g_{\Gamma_i} = b_{\Gamma_i} - A_{\Gamma I}^{(i)} A_{II}^{(i)-1} b_I^{(i)},$$

le complément de Schur global peut alors s'écrire

$$S = S_{NN} = \sum_{i=1}^N R_i^T S_i R_i \quad (4)$$

et le second membre correspondant g_{Γ} comme

$$g_{\Gamma} = \sum_{i=1}^N R_i^T g_{\Gamma_i}, \quad (5)$$

où la matrice de restriction R_i est une matrice composée de zéros et de uns qui extrait les variables de l'interface locale Γ_i à partir d'un vecteur de variables définies sur l'interface Γ .

Le préconditionnement issu de la méthode Balancing Neumann-Neumann \hat{S}^{-1} [21] conduit à un opérateur préconditionné P_{NN} de la forme

$$P_{NN} = \hat{S}^{-1} S_{NN} = P_0 + (I - P_0) \left(\sum_{i=1}^N P_i \right) (I - P_0). \quad (6)$$

Ici P_0 est associé à un problème global grossier, tandis que chaque opérateur P_i est associé à un sous-domaine. Plus précisément, les opérateurs locaux P_i sont définis comme :

$$P_i = R_i^T D_i S_i^\dagger D_i R_i S_{NN}, \quad (7)$$

où les matrices D_i sont diagonales et S_i^\dagger dénote soit l'inverse de S_i , si S_i est non-singulière (cas des problèmes de réaction-diffusion ou des sous-domaines qui touchent la frontière), ou un pseudoinverse de S_i , si S_i est singulière (cas des sous-domaines flottants et des problèmes purement diffusifs). Ici nous employons la formule du pseudo-inverse de Moore-Penrose [17, 24]. Pour définir les matrices D_i , nous avons besoin d'introduire une fonction de décompte pondérée δ_i , qui est associée à Ω_i et est linéaire par morceaux sur Γ_i ; cf. [11, 12, 21, 25, 32]. Elle est définie pour $\gamma \in [1/2, \infty)$ et est déterminée par la somme des contributions de Ω_i et de ses voisins les plus proches

$$\delta_i(x_l) = \sum_{j \in \mathcal{N}_{x_l}} \left(a_{ll}^{(j)} / a_{ll}^{(i)} \right)^\gamma, \quad x_l \in \Gamma_{i,h}, \quad (8)$$

où $a_{ll}^{(i)}$ dénote le l -ième élément de la diagonale de la matrice locale $A^{(i)}$ et \mathcal{N}_{x_l} , $x_l \in \Gamma_h$, est l'ensemble d'indices j des régions telles que $x_l \in \Gamma_{j,h}$. Dans le cas de problèmes purement diffusifs ($c = 0$), nous choisissons ρ_i et ρ_j à la place de $a_{ll}^{(i)}$ et $a_{ll}^{(j)}$. Enfin $\gamma = 1$ est choisi pour les applications numériques. Les pseudoinverses δ_i^\dagger sont définis, pour $x \in \Gamma_{i,h}$, par

$$\delta_i^\dagger(x) = \delta_i^{-1}(x), \quad x \in \Gamma_{i,h}. \quad (9)$$

Nous remarquons que ces fonctions procurent une partition de l'unité :

$$\sum_{i=1}^N R_i^T \delta_i^\dagger(x) \equiv 1. \quad (10)$$

Les matrices D_i sont enfin les matrices diagonales avec comme éléments $\delta_i^\dagger(x)$ correspondant aux noeuds de $\Gamma_{i,h}$. L'espace grossier est défini comme :

$$V_0 = \text{span}\{R_i^T \delta_i^\dagger\},$$

défini sur l'ensemble des sous-domaines flottants. Soit R_0^T la prolongation de l'espace grossier vers l'espace global. Par analogie avec (7), l'opérateur P_0 est défini comme :

$$P_0 = R_0^T S_0^{-1} R_0 S_{NN}, \quad (11)$$

où $S_0 = R_0 S_{NN} R_0^T$ dénote la restriction de S_{NN} à cet espace grossier. Si nous considérons uniquement des solveurs exacts pour les problèmes locaux et globaux, alors P_0 est une projection orthogonale; cf. [34].

3.2.2 Algorithme

A partir des relations (2) et (6), le système préconditionné peut s'écrire sous la forme suivante :

$$P_{NN} u = \hat{S}^{-1} g_\Gamma. \quad (12)$$

Comme P_0 est une projection, nous avons

$$P_0(I - P_0) = 0.$$

Ainsi la décomposition de la solution exacte u de (12)

$$u = P_0 u + w, \quad P_0 u = R_0^T S_0^{-1} R_0 g_\Gamma, \quad (13)$$

avec $w \in \text{Im}(I - P_0)$, conduit à la nouvelle formulation de (12) :

$$(I - P_0) \left(\sum_{i=1}^N P_i \right) (I - P_0) w = \hat{S}^{-1} g_\Gamma - P_0 u, \quad w \in \text{Im}(I - P_0). \quad (14)$$

Il est possible de vérifier que $S_{NN} P_0 = P_0^T S_{NN}$, et que la matrice présentée en (14) peut également s'écrire :

$$(I - P_0) \left(\sum_{i=1}^N P_i \right) (I - P_0) = \left[(I - P_0) \left(\sum_{i=1}^N R_i^T D_i S_i^\dagger D_i R_i \right) (I - P_0^T) \right] S_{NN},$$

ce qui donne l'expression finale pour le préconditionnement \hat{S}^{-1} . Par conséquent, la méthode de Balancing Neumann-Neumann se réduit à une méthode de gradient conjugué dans l'espace $\text{Im}(I - P_0)$ appliquée au système suivant :

$$\left[(I - P_0) \left(\sum_{i=1}^N R_i^T D_i S_i^\dagger D_i R_i \right) (I - P_0^T) \right] S_{NN} w = \hat{S}^{-1} g_\Gamma - P_0 u \quad (15)$$

si un itéré initial $u_0 = P_0 u + \tilde{w}$, avec $\tilde{w} \in \text{Im}(I - P_0)$, est choisi. La méthode de gradient conjugué projetée est présentée tableau 1. Dans ce tableau $\langle \cdot, \cdot \rangle$ dénote le produit scalaire euclidien. Grâce à (13) et au choix de u_0 , la première étape de projection, correspondant à l'application de $I - P_0^T$, peut être omise en pratique.

1. Initialisation	$u_0 = R_0^T S_0^{-1} R_0 g_\Gamma + \tilde{w}, \quad \tilde{w} \in \text{Im}(I - P_0)$ $q_0 = g_\Gamma - S_{NN} u_0$
2. Itération $j = 1, 2, \dots$ jusqu'à convergence	
Projection :	$w_{j-1} = (I - P_0^T) q_{j-1}$
Préconditionnement :	$z_{j-1} = \sum_{i=1}^N R_i^T D_i S_i^\dagger D_i R_i w_{j-1}$
Projection :	$y_{j-1} = (I - P_0) z_{j-1}$
	$\beta_j = \langle y_{j-1}, w_{j-1} \rangle / \langle y_{j-2}, w_{j-2} \rangle \quad [\beta_1 = 0]$
	$p_j = y_{j-1} + \beta_j p_{j-1} \quad [p_1 = y_0]$
	$\alpha_j = \langle y_{j-1}, w_{j-1} \rangle / \langle p_j, S_{NN} p_j \rangle$
	$u_j = u_{j-1} + \alpha_j p_j$
	$q_j = q_{j-1} - \alpha_j S_{NN} p_j$

TAB. 1: Méthode de Balancing Neumann-Neumann.

Les matrices S_{NN} et S_i^\dagger ne doivent pas être explicitement formées en pratique. L'action de S_{NN} sur un vecteur nécessite la solution d'un problème dit de Dirichlet sur chaque sous-structure (application de l'inverse de $A_{II}^{(i)}$), tandis que l'action de S_i^\dagger peut être obtenue en appliquant la matrice pseudo-inverse de $A^{(i)}$ à un vecteur convenable, correspondant à la solution d'un problème de Neumann, voir [34, Chap 4.]. Ainsi une itération de l'algorithme présenté tableau 1 induit une application de P_0 , la solution de problèmes de Neumann locaux sur chaque sous-domaine et la solution de problèmes de Dirichlet (S_{NN}). Comme l'application de P_0 induit aussi l'application de S_{NN} et la résolution d'un problème grossier, le coût global par itération revient à un problème de Neumann et deux problèmes de Dirichlet sur chaque sous-structure et un problème grossier.

3.2.3 Nombre de conditionnement

Une borne supérieure pour l'opérateur préconditionné P_{NN} restreint au sous-espace $Im(I - P_0)$, dans lequel sont confinés les itérés, est donnée par [21] :

$$\kappa(P_{NN}) \leq C \left(1 + \log \left(\frac{H}{h} \right) \right)^2, \quad (16)$$

où la constante C est indépendante des coefficients ρ , de h et des H_i . Notons que $\kappa(P_{NN})$ ne dépend pas du nombre de sous-structures et dépend seulement polylogarithmiquement du rapport $\frac{H}{h}$.

3.3 Méthode FETI

Les méthodes FETI (Finite Element Tearing and Interconnecting) ont été introduites par [16]. Depuis, un travail considérable a été effectué et de nombreuses variantes et améliorations ont été proposées. Nous référons le lecteur à [15] pour une introduction détaillée et à [19, 22] pour leur analyse.

3.3.1 Dérivation

Pour débiter, nous présentons uniquement la méthode FETI dans le cas de problèmes purement diffusifs ($c = 0$ dans le problème (1)). Au lieu de résoudre le système portant sur le complément de Schur (2), la méthode FETI utilise un espace de fonctions discontinues à travers l'interface Γ . La continuité de la solution est assurée par l'utilisation de multiplicateurs de Lagrange. Cette idée conduit à une formulation de type point selle :

$$\left. \begin{array}{l} S_F u_F + B^T \lambda = g_F \\ B u_F = 0 \end{array} \right\}, \quad (17)$$

avec

$$u_F = \begin{bmatrix} u^{(1)} \\ u^{(2)} \\ \vdots \\ u^{(N)} \end{bmatrix}, \quad S_F = \begin{bmatrix} S_1 & O & \cdots & O \\ O & S_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & O \\ O & \cdots & O & S_N \end{bmatrix}, \quad g_F = \begin{bmatrix} g^{(1)} \\ g^{(2)} \\ \vdots \\ g^{(N)} \end{bmatrix}$$

où chaque bloc diagonal de S_F est une matrice de complément de Schur de la forme (3) et B une matrice composée de $(-1, 0, 1)$ qui assure la continuité de la solution aux interfaces entre les sous-domaines. Nous considérons uniquement le cas de multiplicateurs non-redondants : le rang de la matrice B est maximal. En supplément, nous dénotons par R la matrice construite à partir des noyaux des éléments de S_F , i.e., les S_i correspondant aux domaines flottants :

$$R = \begin{bmatrix} r_1 & O & \cdots & O \\ O & r_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & O \\ O & \cdots & O & r_{N_f} \end{bmatrix}$$

où N_f dénote le nombre de sous-domaines flottants. Nous notons également $G = BR$. La prochaine étape revient à éliminer les variables primales u_F de (17) et à dériver une équation pour le multiplicateur de Lagrange λ . Cela conduit à

$$u_F = S_F^\dagger (g_F - B^T \lambda) + R\alpha, \quad (g_F - B^T \lambda) \perp \text{Kernel}(S_F),$$

avec S_F^\dagger un pseudo-inverse de S_F et

$$\left. \begin{array}{l} F\lambda - G\alpha = d \\ G^T \lambda = e \end{array} \right\}, \quad (18)$$

avec $F = BS_F^\dagger B^T$, $d = BS_F^\dagger g_F$, et $e = R^T g_F$. Après avoir introduit un opérateur de projection orthogonale P sur le complément orthogonal de $Im(G)$ et un préconditionnement M^{-1} (tous deux définis plus bas), la méthode FETI se réduit à une méthode de gradient conjugué appliquée dans l'espace des multiplicateurs de Lagrange au système suivant :

$$PM^{-1}P^T F\lambda = PM^{-1}P^T d, \quad (19)$$

avec une approximation initiale λ_0 qui vérifie la deuxième équation de (18). Il est alors possible de choisir

$$\lambda_0 = QG(G^T QG)^{-1}R^T g_F + \tilde{w}, \quad \tilde{w} \in Im(P), \quad (20)$$

où Q est une matrice symétrique à choisir. Ici P est un opérateur de projection orthogonale défini comme $P = I - QG(G^T QG)^{-1}G^T$. De nombreux choix ont été proposés pour le préconditionnement M^{-1} et la matrice Q . Le choix

$$M^{-1} = (BD^{-1}B^T)^{-1}BD^{-1}S_F D^{-1}B^T(BD^{-1}B^T)^{-1}, \quad Q = M^{-1} \quad (21)$$

assure un nombre de conditionnement indépendant de la variation des coefficients ; voir [19]. Ici D est une matrice diagonale par bloc : chaque bloc D_i est associé à un sous-domaine Ω_i et correspond à la matrice locale de mise à l'échelle introduite en section 3.2.1.

3.3.2 Algorithme

La méthode FETI est une méthode de gradient conjugué projetée dans l'espace des multiplicateurs de Lagrange $Im(P)$ appliquée au système (19) avec comme approximation initiale (20). Cet algorithme est donné tableau 2. Dû au choix de la première approximation, la première étape de projection peut être omise.

1. Initialisation	$\lambda_0 = QG(G^T QG)^{-1}R^T g_F + \tilde{w}, \quad \tilde{w} \in Im(P)$ $q_0 = d - F \lambda_0$
2. Itération $j = 1, 2, \dots$ jusqu'à convergence	<div style="margin-left: 20px;"> Projection : $w_{j-1} = P^T q_{j-1}$ Préconditionnement : $z_{j-1} = M^{-1} w_{j-1}$ Projection : $y_{j-1} = P z_{j-1}$ $\beta_j = \langle y_{j-1}, w_{j-1} \rangle / \langle y_{j-2}, w_{j-2} \rangle \quad [\beta_1 = 0]$ $p_j = y_{j-1} + \beta_j p_{j-1} \quad [p_1 = y_0]$ $\alpha_j = \langle y_{j-1}, w_{j-1} \rangle / \langle p_j, F p_j \rangle$ $\lambda_j = \lambda_{j-1} + \alpha_j p_j$ $q_j = q_{j-1} - \alpha_j F p_j$ </div>

TAB. 2: Méthode FETI.

Les matrices F et M^{-1} ne doivent pas être explicitement formées en pratique. L'action de M^{-1} sur un vecteur nécessite la solution d'un problème dit de Dirichlet sur chaque sous-structure (application de S_F et donc des S_i). La matrice $BD^{-1}B^T$ est bloc diagonale : chaque bloc correspond à une variable sur Γ et sa dimension correspond au nombre de contraintes imposées en ce noeud (17) : elle peut être aisément inversée. L'action de F sur un vecteur peut être obtenue en résolvant des problèmes de Neumann sur chaque sous-domaine (application des pseudoinverses S_i^\dagger). Au final, une application de P est nécessaire à chaque étape et nécessite la solution d'un problème grossier (application de $(G^T QG)^{-1}$) et une application de M^{-1} . Le coût global par itération revient à un problème de Neumann et deux problèmes de Dirichlet sur chaque sous-domaine et un problème grossier.

3.3.3 Nombre de conditionnement

Nous dénotons par $P_F = PM^{-1}P^T F$ l'opérateur préconditionné dans l'équation (19). Une borne supérieure pour l'opérateur préconditionné P_F restreint au sous-espace $Im(I-P_0)$, dans lequel sont confinés les itérés, est donnée par :

$$\kappa(P_F) \leq C \left(1 + \log \left(\frac{H}{h} \right) \right)^2, \quad (22)$$

où la constante C est indépendante des coefficients ρ , de h , H et des H_i . Notons que $\kappa(P_F)$ ne dépend pas du nombre de sous-structures et dépend seulement polylogarithmiquement du rapport $\frac{H}{h}$.

3.3.4 Problèmes de réaction-diffusion

Dans le cas de problèmes de réaction-diffusion ($c \neq 0$ dans le problème (1)), les compléments de Schur locaux (3) et S_F sont toujours non singuliers. Une modification [13] est donc employée pour simplifier l'algorithme FETI. Le système pour le multiplicateur de Lagrange est maintenant remplacé par $F\lambda = d$ avec $F = BS_F^{-1}B^T$ et $d = BS_F^{-1}g_F$. Selon [35], un nouvel opérateur de projection P peut être ainsi défini :

$$P = I - QG(G^T QFQG)^{-1}G^T QF.$$

Avec ces notations, le système préconditionné peut être écrit comme dans (19) avec maintenant un itéré initial de la forme :

$$\lambda_0 = QG(G^T QFQG)^{-1}G^T Qd + \tilde{w}, \quad \text{avec } \tilde{w} \in Im(P).$$

Un préconditionnement M^{-1} et une matrice de mise à l'échelle Q identiques à (21) sont employés. Des nombres de conditionnement indépendants des sauts de coefficients sont ainsi obtenus.

3.3.5 Extensions et variantes

Une nouvelle classe de méthodes FETI a été récemment introduite : les méthodes FETI duales primales (FETI-DP), cf [14]. En deux dimensions, les variables associées aux sommets des sous-domaines sont éliminées avec les variables intérieures. Le problème avec contraintes (17) induit maintenant uniquement des variables présentes à l'intérieur des éléments d'interface (segment (en deux dimensions) ou segment et face (en trois dimensions)) et le complément de Schur est désormais inversible. L'équation pour le multiplicateur de Lagrange devient maintenant : $F\lambda = d$, à la place de (18). Nous renvoyons le lecteur à [20, 23] pour l'analyse de certaines méthodes FETI duales primales. Un de leurs principaux avantages est l'utilisation d'un code de calcul identique pour les équations de diffusion pure ou de réaction-diffusion. En plus elles ne nécessitent pas l'introduction de l'opérateur de projection P et de la matrice de mise à l'échelle Q .

4 Aspects pratiques

L'objectif de cette section revient à donner synthétiquement quelques conseils (peut-être!) utiles lors de la mise en pratique des algorithmes de décomposition de domaine.

4.1 Structure de données

Nous avons vu qu'une méthode de décomposition de domaine repose essentiellement sur deux mécanismes : un local (sous-domaine) et un global (problème grossier). La structure de données à adopter doit donc refléter cette caractéristique. Ainsi en langage orienté objet, il est envisageable de créer une classe **Sous-domaine** qui gère l'ensemble des opérations locales (création et application des matrices diagonales D_i , application du complément de Schur local à un vecteur, solution d'un problème de Dirichlet, solution d'un problème de Neumann,...) ainsi qu'une classe **Domaine** qui assurera les opérations globales (construction de la matrice grossière, résolution du système grossier, définition du préconditionnement à base de décomposition de domaine, ...). La classe **Sous-domaine** sera bien sûr accessible au sein de la classe mère **Domaine**.

4.2 Partition du domaine global

Une question pratique revient à décider quelle échelle grossière H choisir. Si la grille fine est obtenue comme raffinement d'une grille grossière, alors la mesure H est d'ores et déjà définie.

Dans le cas contraire, lors de calcul sur machine parallèle, le nombre de processeurs ou de noeuds disponibles détermine parfois le nombre de sous-domaines à employer. Le partitionnement du domaine reviendra donc alors à répartir au mieux les variables et ainsi équilibrer la charge par processeur tout en minimisant les communications entre processeur. Un partitionnement de domaine ou partitionneur permet d'effectuer ce travail. Il est généralement basé sur la théorie des graphes et différentes variantes de découpage de graphes sont couramment proposées. Pour plus d'informations sous forme de liens, nous renvoyons le lecteur à la section 7 partie partitionnements de maillage.

4.3 Résolution des problèmes locaux

Chaque algorithme présenté nécessite de résoudre deux types de problèmes locaux : un problème dit de Dirichlet et un problème dit de Neumann.

4.3.1 Problème de Dirichlet

Le problème dit de Dirichlet revient à résoudre le système linéaire suivant sur chaque sous-domaine :

$$A_{II}^{(i)}u = b_I^{(i)}. \quad (23)$$

Dans le cas de l'utilisation de solveurs exacts sur chaque sous-domaine, il faut donc résoudre ce système avec une grande précision. Une méthode itérative munie d'un préconditionnement adéquat peut donc être utilisée jusqu'à convergence. Toutefois il apparaît judicieux - si l'emplacement mémoire est suffisamment disponible - de factoriser la matrice $A_{II}^{(i)}$ une fois pour toute et d'utiliser cette factorisation à chaque résolution de l'équation (23). L'emploi d'une méthode directe permet au détriment d'un espace mémoire accru d'obtenir une méthode de résolution performante. Pour plus d'informations, nous renvoyons le lecteur à la section 7 partie méthodes de résolution directes.

4.3.2 Problème de Neumann défini sur un domaine non flottant

Tout problème de Neumann défini sur un domaine non flottant équivaut à réaliser le produit matrice vecteur $S_i^{-1}v$, puisque la matrice S_i est inversible sur ce type de domaine. La construction explicite du complément de Schur local ou de son inverse est à éviter puisque cette matrice est généralement dense. Une technique communément adoptée revient à utiliser une factorisation LU par blocs pour la matrice locale $A^{(i)}$:

$$A^{(i)} = \begin{pmatrix} I_{II} & O_{I\Gamma} \\ A_{\Gamma I}^{(i)}A_{II}^{(i)-1} & I_{\Gamma\Gamma} \end{pmatrix} \begin{pmatrix} A_{II}^{(i)} & O_{I\Gamma} \\ O_{\Gamma I} & S_i \end{pmatrix} \begin{pmatrix} I_{II} & A_{II}^{(i)-1}A_{I\Gamma}^{(i)} \\ A_{\Gamma I}^{(i)} & I_{\Gamma\Gamma} \end{pmatrix}.$$

L'inverse de cette matrice est donc déduit et s'écrit :

$$A^{(i)-1} = \begin{pmatrix} I_{II} & -A_{II}^{(i)-1}A_{I\Gamma}^{(i)} \\ O_{\Gamma I} & I_{\Gamma\Gamma} \end{pmatrix} \begin{pmatrix} A_{II}^{(i)-1} & O_{I\Gamma} \\ O_{\Gamma I} & S_i^{-1} \end{pmatrix} \begin{pmatrix} I_{II} & O_{I\Gamma} \\ -A_{\Gamma I}^{(i)}A_{II}^{(i)-1} & I_{\Gamma\Gamma} \end{pmatrix}$$

ou schématiquement :

$$A^{(i)-1} = \begin{pmatrix} \cdots & \cdots \\ \cdots & S_i^{-1} \end{pmatrix}.$$

L'action de l'inverse du complément de Schur local sur un vecteur peut donc finalement s'écrire :

$$S_i^{-1}v = \begin{pmatrix} O & I \end{pmatrix} A^{(i)-1} \begin{pmatrix} O \\ I \end{pmatrix} v.$$

Ainsi il est uniquement nécessaire de résoudre un système linéaire du type :

$$A^{(i)}u = b^{(i)} \quad (24)$$

pour déduire le produit matrice vecteur recherché.

4.3.3 Problème de Neumann défini sur un domaine flottant

Dans ce cas, il est nécessaire de résoudre le système singulier :

$$S_i x = b \quad (25)$$

où S_i est symétrique et la multiplicité de la valeur propre 0 vaut 1 pour le problème envisagé (1). Il est possible de démontrer que pour les procédures de Balancing Neumann-Neumann ou FETI, ce système est consistant i.e. $b \in \text{Ker}(S_i)^\perp = \text{Im}(S_i)$. Une approche classique revient à trouver la solution pseudoinverse par :

$$x = S_i^\dagger b. \quad (26)$$

La matrice pseudoinverse S_i^\dagger peut être obtenue grâce à une décomposition en valeurs singulières (SVD) puis en utilisant la formule du pseudoinverse de Moore-Penrose [17, 24].

4.4 Emploi de solveurs inexacts

Particulièrement lors d'applications tridimensionnelles, où le nombre d'inconnues peut être élevé, l'emploi de solveurs exacts pour les problèmes locaux peut s'avérer pénalisant en temps de calcul. Une technique communément employée - amenuisant les temps de calcul au détriment d'une augmentation du nombre global d'itérations - revient à employer des solveurs inexacts.

4.4.1 Problèmes non singuliers

Problème de Dirichlet Ainsi il suffit de résoudre le problème de Dirichlet (23) par une méthode itérative. Une méthode de gradient conjugué ou toute méthode de Krylov préconditionnée peut donc convenir. L'idée principale est de résoudre uniquement *approximativement* ce système à une tolérance donnée tol et/ou en prescrivant un nombre maximal d'itérations $Itmax$

$$\|r_{It}\|_2 / \|r_0\|_2 \leq tol \quad \text{avec} \quad It \leq Itmax \quad (27)$$

où $r = b_I^{(i)} - A_{II}^{(i)} u$ désigne le résidu du système linéaire (23). La tolérance tol peut par exemple être comprise entre 10^{-1} et 10^{-3} . Comme préconditionnement, une méthode LU incomplète [2], méthode multigrille [29, 38] ou factorisation inverse approchée [5, 18] peuvent être considérées.

Problème de Neumann défini sur un sous-domaine non flottant Une méthode itérative de type gradient conjugué ou méthode de Krylov doit donc être employée pour résoudre le système linéaire suivant :

$$\begin{pmatrix} A_{II}^{(i)} & A_{I\Gamma}^{(i)} \\ A_{\Gamma I}^{(i)} & A_{\Gamma\Gamma}^{(i)} \end{pmatrix} \begin{pmatrix} u_I \\ u_\Gamma \end{pmatrix} = \begin{pmatrix} b_I^{(i)} \\ b_\Gamma^{(i)} \end{pmatrix}.$$

La question centrale revient à déterminer quel préconditionnement adopter. Si des méthodes de résolution performantes sont disponibles pour les blocs diagonaux $A_{II}^{(i)}$ et $A_{\Gamma\Gamma}^{(i)}$, une technique de préconditionnement par blocs de type Gauss-Seidel peut être alors utilisée. Elle présente l'avantage de nécessiter la résolution d'un problème de Dirichlet et donc d'utiliser du code déjà existant.

4.4.2 Problème de Neumann défini sur un domaine flottant

Dans le cas où la matrice singulière S_i est de grande taille, ou si le nombre de sous-domaines est important, le calcul des matrices pseudoinverses peut représenter un effort en temps de calcul important. Une méthode itérative est donc préférable; il a été montré [7] que si la méthode GMRES [30] appliquée au système linéaire (25) avec comme itéré initial $x_o \in \text{Im}(S_i)$ converge, alors elle converge vers la pseudosolution recherchée (26).

4.5 Résolution du problème grossier

Décomposition de domaine est aussi équivalente à décomposition de données : chaque sous-domaine est associé à un processeur du calculateur parallèle. Reste à déterminer comment résoudre le problème grossier. Deux possibilités sont souvent retenues :

- rassembler les données sur un unique processeur et diffuser le résultat
- répartir les données sur chaque processeur et résoudre le système sur chaque processeur en parallèle.

4.6 Avec ou sans recouvrement ?

Deux méthodes sans recouvrement ont été présentées et étudiées. La question de choisir entre une méthode avec ou sans recouvrement s'avère délicate. Les méthodes avec recouvrement sont plus aisées à décrire, à implémenter et à optimiser. Toutefois il faut reconnaître qu'un travail superflu est opéré sur les régions communes. Des travaux anciens ont démontré que les méthodes avec recouvrement sont équivalentes à une méthode sans recouvrement munie d'un préconditionnement spécifique à l'interface ; cf [8] pour une discussion. Le choix est avant tout guidé par l'application et de la disponibilité de solveurs performants sur des domaines précis.

5 Résultats numériques

L'objectif de cette section revient à présenter quelques résultats sur deux exemples simples tirés du problème modèle (1). La méthode d'approximation utilisée est la méthode aux éléments finis de type *hp* [33] i.e. l'approximation dans chaque élément fini (quadrilatère ici) est assurée par des polynômes de degré k dans chaque variable. Le principal intérêt est d'obtenir une convergence exponentielle de l'approximation même en présence de singularités. Ces méthodes sont donc utilisées en sciences de l'ingénieur (mécanique des fluides, électromagnétisme, interaction fluide-structure, mécanique du contact), lorsque la précision numérique de la solution représente un facteur déterminant.

5.1 Problème de Laplace

Afin de posséder des résultats de référence, nous considérons en premier lieu l'opérateur de Laplace avec des conditions de Dirichlet inhomogènes $((\varepsilon_x, \varepsilon_y) = (1, 1), (\rho_x, \rho_y) = (1, 1), c = 0$ dans le problème modèle (1)) :

$$\begin{aligned} -\Delta u &= f, & \text{dans } \Omega, \\ u &= u_D, & \text{sur } \partial\Omega. \end{aligned} \quad (28)$$

Le terme source f est dérivé de telle sorte que u_D soit la solution exacte de ce problème. Le maillage uniforme \mathcal{T} composé de $N = N_x \times N_y$ éléments représente aussi la partition du domaine choisie. Le nombre total de sous-structures est donc $N = N_x \times N_y$. Pour ce genre d'approximations, il a été prouvé [36] que le nombre de conditionnement de l'opérateur préconditionné P vérifiait :

$$\kappa(P) \leq C (1 + \log(k))^2$$

où k représente le degré polynômial de l'approximation et C une constante indépendante de k et du nombre de sous-domaines.

Les résultats pour les méthodes de préconditionnement Balancing Neumann-Neumann et FETI sont présentés tableau 3 et tableau 4, respectivement. Dans la partie supérieure des tableaux, le nombre de sous-domaines est fixe ($N_x \times N_y = 3 \times 3$), tandis que le degré polynômial varie de 2 à 12. Dans la partie inférieure, le degré polynômial est fixé à 4, tandis que le nombre de sous-domaines augmente de 2×2 à 12×12 . Les quatre premières colonnes rassemblent le nombre d'itérations nécessaires pour satisfaire le critère d'arrêt (29)

$$\|r_{It}\|_2 / \|r_0\|_2 \leq 10^{-14} \quad (29)$$

où r représente le résidu du système linéaire global, les valeurs propres minimales et maximales et le nombre de conditionnement pour les opérateurs non-préconditionnés S_{NN} et F . Les colonnes suivantes rassemblent les mêmes données, quand les méthodes Balancing Neumann-Neumann ou FETI sont utilisées comme préconditionnement du gradient conjugué.

Nombre de sous-domaines $N_x \times N_y = 3 \times 3$								
	Sans préconditionnement				NN			
k	It	λ_{max}	λ_{min}	$\kappa(S_{NN})$	It	λ_{max}	λ_{min}	$\kappa(P_{NN})$
2	9	5.3161	0.6667	7.9741	6	1.076	1	1.076
3	19	5.6508	0.3964	14.2544	9	1.4364	1	1.4364
4	26	5.7291	0.28	20.4629	10	1.7542	1	1.7542
5	31	5.7737	0.2157	26.7612	11	2.1137	1	2.1137
6	34	5.8029	0.1752	33.1169	12	2.4471	1	2.4471
7	38	5.8264	0.1474	39.5316	13	2.7688	1	2.7688
8	42	5.8465	0.1271	45.995	13	3.07	1	3.07
9	45	5.8644	0.1117	52.5011	13	3.3575	1	3.3575
10	48	5.8807	0.0996	59.0453	14	3.629	1	3.629
11	52	5.896	0.0898	65.624	14	3.8884	1	3.8884
12	55	5.9103	0.0818	72.2349	14	4.1352	1	4.1352

Degré polynômial $k = 4$								
	Sans préconditionnement				NN			
$N_x \times N_y$	It	λ_{max}	λ_{min}	$\kappa(S_{NN})$	It	λ_{max}	λ_{min}	$\kappa(P_{NN})$
2×2	10	5.5968	0.544	10.2891	3	1.5034	1	1.5034
3×3	26	5.7291	0.28	20.4629	10	1.7542	1	1.7542
4×4	36	5.7773	0.1655	34.9023	14	1.8179	1	1.8179
5×5	45	5.8	0.1084	53.5172	16	1.8528	1	1.8528
6×6	57	5.8124	0.0762	76.287	16	1.8725	1	1.8725
7×7	67	5.8199	0.0564	103.2052	16	1.8854	1	1.8854
8×8	76	5.8248	0.0434	134.269	16	1.8939	1	1.8939
9×9	85	5.8281	0.0344	169.477	16	1.8998	1	1.8998
10×10	94	5.8305	0.0279	208.8288	16	1.9041	1	1.9041
11×11	103	5.8323	0.0231	252.3238	16	1.9073	1	1.9073
12×12	111	5.8336	0.0194	299.9618	16	1.9098	1	1.9098

TAB. 3: **Problème de Laplace.** Gradient Conjugué pour le complément de Schur sans préconditionnement et avec le préconditionnement de Balancing Neumann-Neumann : nombre d'itérations, valeurs propres minimale et maximale et conditionnement en fonction du degré polynômial et du nombre de sous-domaines, respectivement.

Quel que soit le choix du préconditionnement, le nombre d'itérations pour le gradient conjugué préconditionné reste borné indépendamment de $N_x \times N_y$; cf les parties inférieures des tableaux 3 et 4. Les nombres de conditionnement $\kappa(P_{NN})$ et $\kappa(P_F)$ sont représentés figure 2 en fonction du degré polynômial. Comme attendu, la croissance est quadratique en $\log(k)$.

5.2 Problème à interface

Le problème précédent représente le problème standard qui permet d'obtenir une première estimation des performances des algorithmes par comparaison avec des préconditionnements classiques. Ce problème était défini sur un maillage uniforme.

Toutefois de nombreuses applications en sciences de l'ingénieur font état d'une grande variété d'échelles dans la solution (combustion, météorologie, ...) : il s'avère donc essentiel de capturer correctement les phénomènes physiques et d'utiliser donc des méthodes d'approximation élevées généralement sur des maillages raffinés. Des maillages étirés aux forts rapports d'aspect sont donc obtenus. En conséquence,

<i>Nombre de sous-domaines $N_x \times N_y = 3 \times 3$</i>								
	Sans préconditionnement				FETI			
k	It	λ_{max}	λ_{min}	$\kappa(F)$	It	λ_{max}	λ_{min}	$\kappa(P_F)$
2	24	8.309	0.57385	14.4793	9	2.0516	1.0002	2.0512
3	36	11.089	0.62896	17.6308	11	2.7284	1.0001	2.7281
4	39	13.685	0.64515	21.2113	12	3.4415	1.0002	3.4409
5	42	16.263	0.65642	24.7761	12	4.0378	1.0003	4.0364
6	44	18.867	0.65997	28.5881	12	4.5902	1.0003	4.5888
7	48	21.51	0.66639	32.2776	14	5.0849	1.0001	5.0843
8	47	24.186	0.66492	36.3747	13	5.5443	1.0007	5.5404
9	52	26.892	0.67048	40.1091	14	5.9669	1.0006	5.9633
10	50	29.622	0.66676	44.427	14	6.3628	1.0011	6.3558
11	56	32.37	0.67049	48.2788	15	6.7327	1.0009	6.7267
12	53	35.134	0.66716	52.6613	15	7.0821	1.0016	7.0708

<i>Degré polynômial $k = 4$</i>								
	Sans préconditionnement				FETI			
$N_x \times N_y$	It	λ_{max}	λ_{min}	$\kappa(F)$	It	λ_{max}	λ_{min}	$\kappa(P_F)$
2×2	15	10.846	0.65997	16.4338	4	2.2515	1	2.2515
3×3	39	13.685	0.64515	21.2113	12	3.4415	1.0002	3.4409
4×4	49	14.076	0.63808	22.0603	16	3.0693	1.0002	3.0686
5×5	61	14.443	0.45086	32.0348	18	3.0485	1.0006	3.0467
6×6	69	14.652	0.34379	42.618	19	2.9854	1.0004	2.9844
7×7	75	14.777	0.29059	50.852	20	2.9877	1.0004	2.9864
8×8	81	14.858	0.26057	57.0227	19	2.9769	1.0004	2.9758
9×9	88	14.913	0.24195	61.6358	19	2.9768	1.0004	2.9757
10×10	92	14.952	0.2296	65.1196	20	2.9769	1.0003	2.9759
11×11	97	14.98	0.22098	67.7892	20	2.9771	1.0003	2.9761
12×12	101	15.001	0.21471	69.8672	20	2.9768	1.0003	2.9759

TAB. 4: **Problème de Laplace.** Méthode de Gradient Conjugué pour le système de la méthode FETI : même légende que le tableau 3.

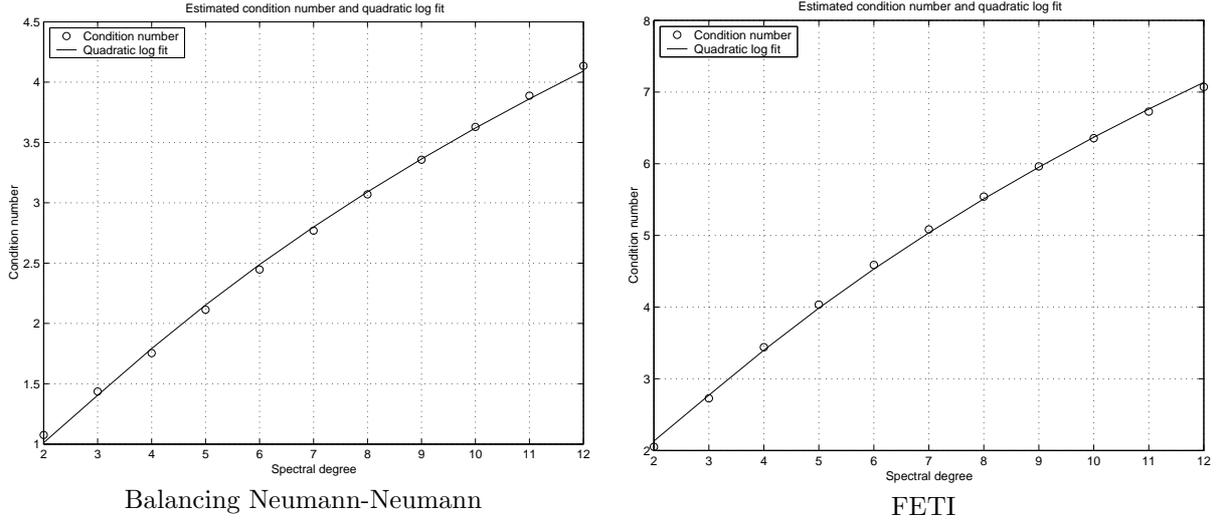


FIG. 2: **Problème de Laplace.** Nombres de conditionnement estimés (cercles) et approximations polynômiales logarithmiques au sens des moindres carrés (ligne continue) en fonction du degré spectral pour la méthode de Balancing Neumann-Neumann (gauche, résultats du tableau 3) et pour la méthode FETI (droite, résultats du tableau 4).

l'approximation par éléments finis (par exemple) conduira à des matrices mal conditionnées posant de sérieuses difficultés pour toute méthode itérative. Le but de cet exemple est de montrer les performances des deux algorithmes sur un tel problème.

Les raffinements nécessaires peuvent intervenir non seulement à proximité des frontières du domaine mais également à l'interface de régions possédant des propriétés différentes. Ainsi les problèmes d'interface en géophysique, en ingénierie pétrolière (cf. [39]) ou en électronique numérique (cf. [26]) peuvent nécessiter des maillages extrêmement raffinés à l'intérieur du domaine. De tels problèmes impliquant simultanément des sauts de coefficients et de larges rapports d'aspect sont importants en pratique. Le problème à interface est défini ainsi :

$$\begin{aligned} -\nabla \cdot (\rho \nabla u) &= 1, & \text{dans } \Omega, \\ u &= u_D, & \text{sur } \partial\Omega. \end{aligned} \quad (30)$$

Nous supposons que le domaine Ω est divisé en quatre carrés identiques et que le coefficient ρ possède une répartition en damier donnée par $\rho_1 = 10^4$ et $\rho_2 = 1$. Comme partition du domaine \mathcal{T}^{DD} , nous choisissons celle donnée par ρ . Nous avons donc $N_x \times N_y = 2 \times 2$ sous-domaines. Afin de capturer les effets proches de l'interface, nous avons employé des maillages géométriquement raffinés des deux cotés de l'interface $x = 1/2$ et $y = 1/2$ en accord avec la théorie des éléments finis hp . Par construction, le maillage raffiné comporte $(2 + 2k) \times (2 + 2k)$ quadrilatères.

Le degré polynômial k varie de 2 à 8. Les résultats sont présentés tableaux 5 et 6, respectivement. Le nombre d'itérations de gradient conjugué reste borné uniformément pour k croissant. La figure 3 montre $\kappa(P_{NN})$ et $\kappa(P_F)$ en fonction du degré polynômial k dans un graphe log-log : un comportement linéaire est obtenu. Numériquement $\log(\kappa(P_{NN}))$ et $\log(\kappa(P_F))$ croissent comme k^p avec $p = 0.8$ et $p = 0.74$, respectivement. Remarquons le faible nombre d'itérations dans ce cas. Notons que le nombre de conditionnement de l'opérateur non préconditionné $\kappa(S_{NN})$ croît exponentiellement avec le degré polynômial k ! Le préconditionnement à base de méthode de décomposition de domaine s'avère donc efficace.

6 Logiciels issus du domaine public

Nous mentionnons ici quelques bibliothèques de calcul scientifique destinées au calcul parallèle pouvant être utilisées dans un projet à propos de méthodes de décomposition de domaine. L'avantage principal

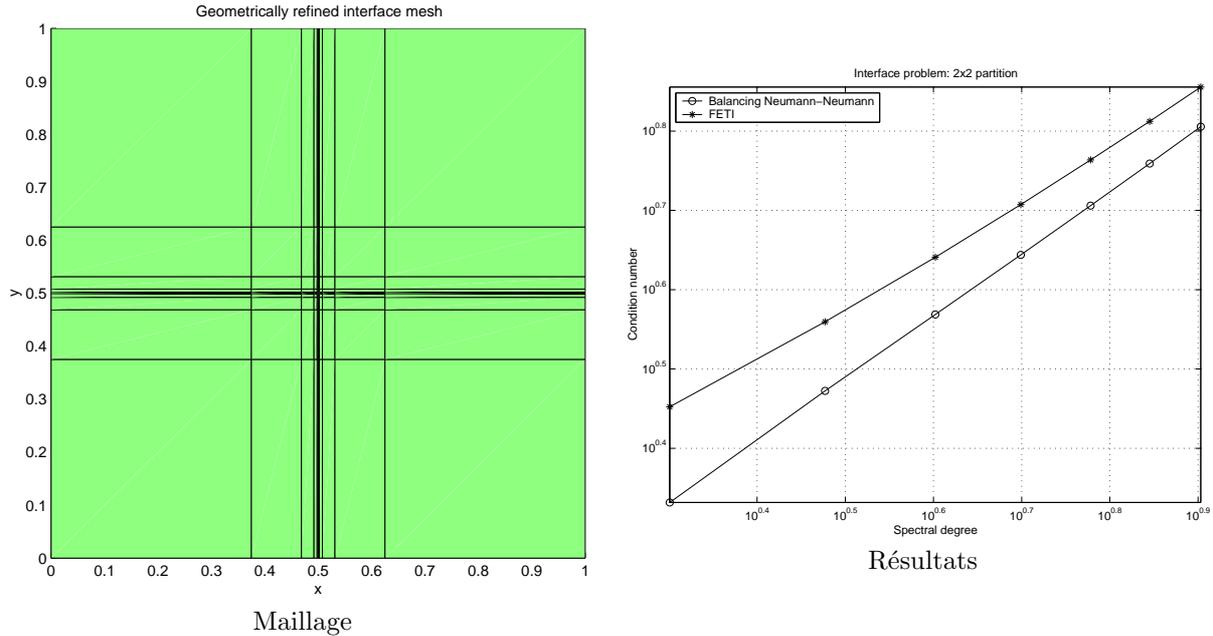


FIG. 3: **Problème à interface.** Maillage anisotrope avec raffinement intérieur dans le cas $n = k = 4$ (gauche). Nombres de conditionnement issus des méthodes de Balancing Neumann-Neumann (tableau 5) et FETI (tableau 6) en fonction du degré polynômial (graphe log-log) (droite).

Nombre de sous-domaines $N_x \times N_y = 2 \times 2$								
	Sans préconditionnement				NN			
k	It	λ_{max}	λ_{min}	$\kappa(S_{NN})$	It	λ_{max}	λ_{min}	$\kappa(P_{NN})$
2	19	26848	839.29	31.9891	4	2.1464	1	2.1464
3	34	27946	302.87	92.2692	6	2.9672	1	2.9672
4	47	28313	143.42	197.4121	6	3.7049	1	3.7049
5	60	28561	79.144	360.8691	6	4.4046	1	4.4046
6	73	28753	48.261	595.7833	6	5.0814	1	5.0814
7	85	28910	31.587	915.2486	7	5.7424	1	5.7424
8	98	29043	21.797	1332.4393	7	6.3915	1	6.3915

TAB. 5: **Problème à interface.** Gradient Conjugué pour le complément de Schur sans préconditionnement et avec le préconditionnement de Balancing Neumann-Neumann : nombre d'itérations, valeurs propres minimale et maximale et conditionnement en fonction du degré polynômial pour un nombre de sous-domaines fixé.

réside en leur flexibilité.

6.1 PETSC

PETSC² (Portable and Extensible Toolkit for Scientific Computing) est une bibliothèque écrite en C qui est destinée au calcul parallèle grâce à la bibliothèque de passage de messages MPI. Elle comporte entre autre des interfaces permettant d'utiliser des méthodes de décomposition de domaine comme préconditionnement de méthodes de Krylov. Cette bibliothèque constitue l'un des premiers exemples de réalisation logicielle prouvant le bien-fondé des méthodes de décomposition de domaine dans le cadre du

²<http://www-unix.mcs.anl.gov/petsc/petsc-as/>

Nombre de sous-domaines $N_x \times N_y = 2 \times 2$									
	Sans préconditionnement				FETI				
k	It	λ_{max}	λ_{min}	$\kappa(F)$	It	λ_{max}	λ_{min}	$\kappa(P_F)$	
2	24	18.903	0.00024261	7.7915E+04	4	2.8336	1	2.8335	
3	34	41.593	0.00033396	1.2455E+05	4	3.6267	1	3.6265	
4	40	79.826	0.00041491	1.9239E+05	4	4.3742	1.0001	4.3738	
5	47	138.04	0.00049044	2.8147E+05	4	5.0978	1.0002	5.0971	
6	50	220.55	0.00056243	3.9213E+05	4	5.8046	1.0002	5.8033	
7	49	331.61	0.00063183	5.2485E+05	3	6.4982	1.0013	6.4899	
8	48	475.52	0.00069917	6.8012E+05	3	7.1814	1.0015	7.171	

TAB. 6: **Problème à interface**. Gradient Conjugué pour le complément de Schur sans préconditionnement et avec le préconditionnement de FETI : nombre d'itérations, valeurs propres minimale et maximale et conditionnement en fonction du degré polynômial pour un nombre de sous-domaines fixé.

calcul intensif sur machines parallèles.

6.2 AZTEC

AZTEC³ est une bibliothèque d'algèbre linéaire destinée à la résolution de grands systèmes linéaires sur calculateurs parallèles. Cette bibliothèque détient pour fondement l'idée de décomposition de domaine ou de partitionnement de la matrice et est donc un outil idéal pour construire de nouvelles méthodes de décomposition de domaine ou pour tester des variantes algorithmiques. La structure de cette bibliothèque permet d'insérer aisément un produit matrice-vecteur ad hoc ou un nouveau préconditionnement. Les méthodes de Krylov y sont déjà présentes ainsi que des méthodes de décomposition de domaine de type Schwarz avec des préconditionnements à base de factorisations complètes ou incomplètes. Sa flexibilité est l'un des atouts majeurs de cette bibliothèque et peut être considérée comme supérieure à celle de PETSC. A noter qu'une extension de cette bibliothèque à des méthodes de décomposition de domaine sans recouvrement a été récemment proposée [31].

6.3 TRILINOS

Trilinos⁴ représente l'un des projets récents les plus ambitieux en matière de développement de logiciels en calcul scientifique. Il comporte une collection de bibliothèques chacune destinée à une application bien spécifique (optimisation, résolution de systèmes linéaires par méthodes directes, résolution de problèmes aux valeurs propres...). L'ensemble de ces bibliothèques est écrit en C++ et fait également appel à MPI pour les applications parallèles. Une bibliothèque nommée AZTECOO est destinée aux méthodes de décomposition de domaine. Elle correspond à une réécriture version orientée objet de la bibliothèque AZTEC. Une autre bibliothèque ML permet d'utiliser des préconditionnements à base de méthodes multigrille ou multi-échelles. L'avantage majeur de ce projet est de pouvoir combiner à volonté ces sous-bibliothèques et de disposer d'une unique interface entre les structures de données et les bibliothèques.

7 Quelques liens

Liens généraux

DDM : <http://www.ddm.org>

DD17 : <http://www.ricam.oeaw.ac.at/dd17/>

³<http://www.cs.sandia.gov/CRF/aztec1.html>

⁴<http://software.sandia.gov/Trilinos/>

Méthodes de résolution directes

AMESOS : <http://software.sandia.gov/trilinos/packages/amesos/index.html>
 MUMPS : <http://graal.ens-lyon.fr/~jylexcel/MUMPS/>
 PARDISO : <http://www.computational.unibas.ch/cs/scicomp/software/pardiso/>
 PASTIX : <http://www.labri.fr/Perso/~ramet/pastix/>
 SPOOLES : <http://www.netlib.org/linalg/spooles/spooles.2.2.html>
 SUPERLU : <http://crd.lbl.gov/~xiaoye/SuperLU/>
 WSMP : <http://www-users.cs.umn.edu/~agupta/wsmp.html>
<http://www.netlib.org/utk/people/JackDongarra/la-sw.html>

Méthodes de résolution itératives

AZTEC : <http://www-unix.mcs.anl.gov/petsc/petsc-as/>
 PETSC : <http://www.cs.sandia.gov/CRF/aztec1.html>
 TRILINOS : <http://software.sandia.gov/Trilinos/>
<http://www.netlib.org/utk/people/JackDongarra/la-sw.html>

Partitionnements de matrice ou de maillage

COLAMD : <http://www.cise.ufl.edu/research/sparse/colamd/>
 METIS : <http://www-users.cs.umn.edu/~karypis/metis/>
 MONDRIAAN : <http://www.math.uu.nl/people/bisseling/Mondriaan/mondriaan.html>
 PARTY : <http://wwwcs.uni-paderborn.de/fachbereich/AG/monien/RESEARCH/PART/party.html>
 PATOH : <http://bmi.osu.edu/~umit/software.htm>
 SCOTCH : <http://www.labri.fr/Perso/~pelegrin/scotch/>

8 Conclusion

Ce document a permis de présenter succinctement deux méthodes de décomposition de domaine faisant partie de la famille des méthodes sans recouvrement. Ces méthodes sont utilisées comme préconditionnement de méthodes de gradient conjugué dans le cas symétrique défini positif et permettent d'obtenir des méthodes numériques quasi-optimales. Elles présentent également l'avantage d'une insertion relativement aisée dans un code déjà existant, notamment dans le cadre d'applications sur machines parallèles.

Par souci de brièveté, la discussion était restreinte aux systèmes linéaires issus de la discrétisation d'équations aux dérivées partielles elliptiques. Une équation de réaction-diffusion était étudiée. Les résultats obtenus peuvent s'étendre au cas d'équations de convection-diffusion [1, 8]. Toutefois le cadre d'application des méthodes de décomposition de domaine est nettement plus vaste. Des applications dans le cadre d'équations aux dérivées partielles paraboliques et hyperboliques sont discutées par exemple dans [28]. Des applications à des systèmes d'équations sont également envisagées dans [8, 37].

Il nous est donc impossible de survoler l'ensemble des travaux consacrés aux méthodes de décomposition de domaine. Nous renvoyons le lecteur à une littérature fournie. Les aspects théoriques sont ainsi abordés en détail dans les ouvrages de référence [28, 34, 37]. [34] propose une approche orientée vers la pratique et les aspects numériques, tandis que [37] présente les développements les plus récents. Enfin les actes de conférences consacrées aux méthodes de décomposition de domaine disponibles sur le site <http://www.ddm.org> permettent de prendre connaissance des nombreuses applications envisagées en sciences de l'ingénieur.

9 Remerciements

Je tiens à remercier Anne Cadiou pour l'invitation à cette école CNRS. Je remercie également Andrea Toselli pour son encadrement efficace et ses conseils toujours avisés durant ces dernières années. Ce travail commun est en partie financé par le Fonds National Suisse (FNS) (projet 20-63397.00).

Références

- [1] Yves Achdou, Patrick Le Tallec, Frédéric Nataf, and Marina Vidrascu. A domain decomposition preconditioner for an advection-diffusion problem. *Comput. Methods Appl Mech. Engrg.*, 184 :145–170, 2000.
- [2] Owe Axelsson. *Iterative Solution Methods*. Cambridge University Press, 1994.
- [3] Richard Barrett, Michael Berry, Tony F. Chan, James Demmel, June Donato, Jack Dongarra, Victor Eijkhout, Roldan Pozo, Charles Romine, and Henk Van der Vorst. *Templates for the Solution of Linear Systems : Building Blocks for Iterative Methods, 2nd Edition*. SIAM, Philadelphia, PA, 1994.
- [4] Michele Benzi. Preconditioning techniques for large linear systems : a survey. *J. Comput. Phys.*, 182(2) :418–477, 2002.
- [5] Michele Benzi, Jane Cullum, and Miroslav Tuma. Robust approximate inverse preconditioning for the conjugate gradient method. *SIAM J. Sci. Comput.*, 22(4) :1318–1332, 2000.
- [6] Manoj Bhardway, Kendall Pierson, Garth Reese, Tim Walsh, David Day, Ken Alvin, James Peery, Charbel Farhat, and Michel Lesoinne. Salinas : a scalable software for high-performance structural and solid mechanics simulations. 2002. Paper submitted for the Gordon Bell Award, Supercomputing 2002, Baltimore 16-22 November 2002, URL : <http://www.sc-2002.org/paperpdfs/pap.pap216.pdf>.
- [7] Peter Brown and Homer Walker. GMRES on (nearly) singular systems. *SIAM J. Matrix Anal. Appl.*, 18(1) :37–51, 1997.
- [8] Tony Chan and Tarek Mathew. Domain decomposition algorithms. *Acta Numerica*, pages 61–143, 1994.
- [9] Henk Van der Vorst. *Iterative Krylov methods for large linear systems*. Cambridge University Press, 2003.
- [10] Jack Dongarra, Ian Duff, Danny Sorensen, and Henk Van der Vorst. *Numerical Linear Algebra for High-Performance Computers*. SIAM, Philadelphia, PA, 1998.
- [11] Maksymilian Dryja, Marcus V. Sarkis, and Olof B. Widlund. Multilevel Schwarz methods for elliptic problems with discontinuous coefficients in three dimensions. *Numer. Math.*, 72(3) :313–348, 1996.
- [12] Maksymilian Dryja and Olof B. Widlund. Schwarz methods of Neumann-Neumann type for three-dimensional elliptic finite element problems. *Comm. Pure Appl. Math.*, 48(2) :121–155, February 1995.
- [13] Charbel Farhat and Po-Shu Chen. A scalable Lagrange multiplier based domain decomposition method for time-dependent problems. *Int. J. Numer. Meth. Engrg.*, 38 :3831–3853, 1995.
- [14] Charbel Farhat, Michel Lesoinne, Patrick LeTallec, Kendall Pierson, and Daniel Rixen. FETI-DP : a dual-primal unified FETI method. I. A faster alternative to the two-level FETI method. *Internat. J. Numer. Methods Engrg.*, 50(7) :1523–1544, 2001.
- [15] Charbel Farhat and François-Xavier Roux. Implicit parallel processing in structural mechanics. In J. Tinsley Oden, editor, *Computational Mechanics Advances*, volume 2 (1), pages 1–124. North-Holland, 1994.
- [16] Charbel Farhat and Francois-Xavier Roux. A method of finite element tearing and interconnecting and its parallel solution algorithm. *Int. J. Numer. Meth. Engrg.*, 32 :1205–1227, 1991.
- [17] Gene Golub and Charles Van Loan. *Matrix Computations*. The John Hopkins University Press, 1996. Third edition.

- [18] Marcus Grote and Thomas Huckle. Parallel preconditioning with sparse approximate inverses. *SIAM J. Sci. Comput.*, 18(3) :838–853, 1997.
- [19] Axel Klawonn and Olof B. Widlund. FETI and Neumann-Neumann iterative substructuring methods : connections and new results. *Comm. Pure Appl. Math.*, 54(1) :57–90, 2001.
- [20] Axel Klawonn, Olof B. Widlund, and Maksymilian Dryja. Dual-Primal FETI methods for three-dimensional elliptic problems with heterogeneous coefficients. Technical Report 815, Department of Computer Science, Courant Institute, April 2001.
- [21] Jan Mandel and Marian Brezina. Balancing domain decomposition for problems with large jumps in coefficients. *Math. Comp.*, 65 :1387–1401, 1996.
- [22] Jan Mandel and Radek Tezaur. Convergence of a substructuring method with Lagrange multipliers. *Numer. Math.*, 73 :473–487, 1996.
- [23] Jan Mandel and Radek Tezaur. On the convergence of a dual-primal substructuring method. *Numer. Math.*, 88(3) :543–558, January 2001.
- [24] Carl D. Meyer. *Matrix Analysis and Applied Linear Algebra*. SIAM, 2000.
- [25] Luca F. Pavarino. Neumann-Neumann algorithms for spectral elements in three dimensions. *RAIRO Mathematical Modelling and Numerical Analysis*, 31 :471–493, 1997.
- [26] S.J. Polak, C. Den Heijer, W.H. Schilders, and P. Markowich. Semiconductor device modelling from the numerical point of view. *Int. J. for Num. Methods in Eng.*, 24 :763–838, 1987.
- [27] J.S. Przemieniecki. Matrix structural analysis of substructures. *Amer. Inst. Aero. Astro.*, 1 :138–147, 1963.
- [28] Alfio Quarteroni and Alberto Valli. *Numerical Approximation of Partial Differential Equations*. Springer-Verlag, Berlin, 1994.
- [29] John Ruge and Klaus Stüben. Algebraic multigrid. In S. Mc Cormick, editor, *Multigrid Methods*, pages 73–130. SIAM Philadelphia, 1987.
- [30] Youcef Saad and Martin Schultz. GMRES : a generalized minimal residual algorithm for solving nonsymmetric linear system. *SIAM J. Sci. Statist. Comput.*, 7(3) :856–869, 1986.
- [31] Marzio Sala. *Domain Decomposition Preconditioners : theoretical properties, application to the compressible Euler equations, parallel aspects*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, 2003.
- [32] Marcus V. Sarkis. *Schwarz Preconditioners for Elliptic Problems with Discontinuous Coefficients Using Conforming and Non-Conforming Elements*. PhD thesis, Courant Institute, New York University, September 1994. TR671, Department of Computer Science, New York University, URL : `file ://cs.nyu.edu/pub/tech-reports/tr671.ps.Z`.
- [33] Christoph Schwab. *p - and hp - Finite Element Methods*. Oxford Science Publications, 1998.
- [34] Barry F. Smith, Petter E. Bjørstad, and William D. Gropp. *Domain Decomposition : Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, 1996.
- [35] Andrea Toselli. FETI domain decomposition methods for scalar advection-diffusion problems. *Comp. Methods Appl. Mech. Eng.*, 190 :5759–5776, 2001.
- [36] Andrea Toselli and Xavier Vasseur. Domain decomposition methods of Neumann-Neumann type for hp -approximations on geometrically refined boundary layer meshes in two dimensions. Technical Report 02–15, Seminar für Angewandte Mathematik, ETH, Zürich, September 2002. Submitted to Numerische Mathematik.
- [37] Andrea Toselli and Olof Widlund. *Domain Decomposition Methods - Theory and Algorithms*. Springer Verlag, 2004.
- [38] Ulrich Trottenberg, Cornelis Oosterlee, and Anton Schüller. *Multigrid*. Academic Press, London, 2000. Guest contribution by Klaus Stüben : "An Introduction to Algebraic Multigrid".
- [39] C. Vuik, A. Segal, and J.A. Meijerink. An efficient preconditioned CG method for the solution of a class of layered problems with extreme contrasts in the coefficients. *J. Comput. Phys.*, 152 :385–403, 1999.