

Consistance des grilles grossières pour les méthodes déflation et balancing

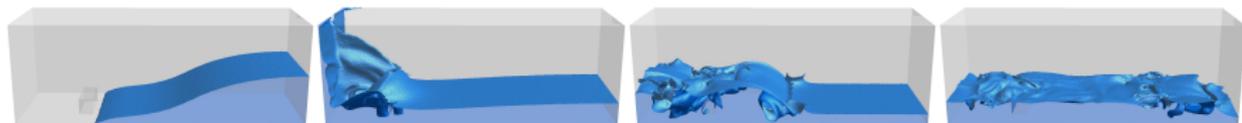
Hubert ALCIN

26 octobre 2010

Plan de la présentation

- 1** MOTIVATION
- 2** MÉTHODE DE SCHWARZ
 - Méthode de Schwarz additive
 - Scalabilité
- 3** PRÉCONDITIONNEUR GROSSIER
 - Méthode de déflation.
 - Méthode balancing
 - Consistance Grille Grossière
- 4** EXPÉRIENCES NUMÉRIQUES EN DEUX NIVEAUX
 - Comparaison Schwarz, Balancing
 - Comparaison fonctions lisses et caractéristiques
 - Application sur un maillage isotrope.
 - Résolution du problème de poisson pour la projection
- 5** MÉTHODE À 3 NIVEAUX
 - Motivation et approche
 - Comparaison du deux niveaux et du trois niveaux
- 6** CONCLUSIONS ET PERSPECTIVES

1. MOTIVATIONS : Equations de Navier-Stokes



Déferlement d'une colonne d'eau sur un obstacle cubique

1. MOTIVATIONS : Equations de Navier-Stokes

Equations de Navier-Stokes incompressible :

$$\begin{cases} \frac{\partial U}{\partial t} + h(U, \rho) + \frac{1}{\rho} \nabla p = 0 \\ \nabla \cdot U = 0 \end{cases}$$

- ρ la densité discontinu, ν la viscosité, p la pression et U la vitesse.
- avec $h(U, \rho) = U \cdot \nabla U - \nu(\rho) \Delta U - g$
- $\rho = \begin{cases} \rho_1 & \text{si } x \in \text{phase 1} \\ \rho_2 & \text{si } x \in \text{phase 2} \end{cases}$

Schéma temporel : méthode de projection

Calcul à l'instant t^{n+1} .

- calcul de la variable intermédiaire \bar{U}^{n+1} :

$$\bar{U}^{n+1} = U^n - \Delta h(U^n, \rho^n)$$

- Calcul de p^{n+1} :

$$\nabla \cdot \frac{1}{\rho^{n+1}} \nabla p^{n+1} = \frac{\nabla \cdot \bar{U}^{n+1}}{\Delta t}$$

- Calcul de U^{n+1} :

$$U^{n+1} = \bar{U}^{n+1} - \frac{\Delta t}{\rho^{n+1}} \nabla p^{n+1}$$

⇒ Système linéaire à résoudre pour la projection

Problème de Neumann à coefficients discontinus

$$-\nabla^* \frac{1}{\rho} \nabla p = RHS \text{ sur } \Omega$$

$$\frac{\partial p}{\partial n} = 0 \text{ sur } \partial\Omega$$

$$p(0) = 0.$$

2. METHODE DE SCHWARZ

Problème modèle :

$$A = \begin{cases} -\Delta + 1 \text{ sur } \Omega \\ \frac{\partial}{\partial n} \text{ sur } \partial\Omega \end{cases}$$

Résolution de $Au = f$

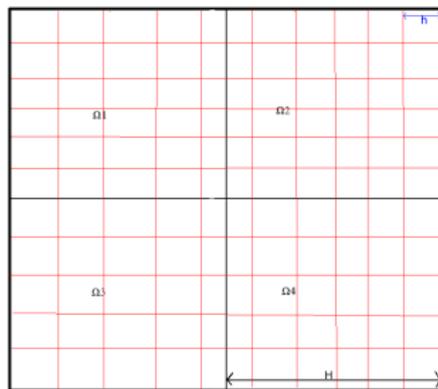
Problème discret (éléments finis 1D)

$$A_h u_h = f_h ; A_h = h \begin{pmatrix} \frac{1}{h^2} + \frac{1}{2} & \frac{-1}{h^2} & \dots & & 0 \\ \frac{-1}{h^2} & \frac{2}{h^2} + 1 & \frac{-1}{h^2} & 0 & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ 0 & & & \frac{-1}{h^2} & \frac{2}{h^2} + 1 & \frac{-1}{h^2} \\ & & \dots & 0 & \frac{-1}{h^2} & \frac{1}{h^2} + \frac{1}{2} \end{pmatrix}$$

- u_h et f_h sont des vecteurs de dimension n .
- A_h une matrice définie positive.
- Le conditionnement de la matrice A_h est de l'ordre de h^{-2} où h est la taille de maillage.

Décomposition du maillage

- h représente la taille du maillage fin.
- $\Omega = \cup_{i=1}^N \Omega_i$. **Partition** : un nœud appartient à un seul sous-domaine.
- H représente la taille d' un sous-domaine.



Condition d'interface classique : La méthode de Schwarz d'origine

Le Schwarz multiplicatif (H. Schwarz, 1870)

$$\left\{ \begin{array}{l} (-\Delta + 1)(u_1^{n+1}) = f \text{ dans } \Omega_1 \\ \frac{\partial}{\partial n} u_1^{n+1} = 0 \text{ sur } \partial\Omega_1 \cap \partial\Omega \\ u_1^{n+1} = u_2^n \text{ sur } \partial\Omega_1 \cap \overline{\Omega_2} \end{array} \right. \quad \left\{ \begin{array}{l} (-\Delta + 1)(u_2^{n+1}) = f \text{ dans } \Omega_2 \\ \frac{\partial}{\partial n} u_2^{n+1} = 0 \text{ sur } \partial\Omega_2 \cap \partial\Omega \\ u_2^{n+1} = u_1^{n+1} \text{ sur } \partial\Omega_2 \cap \overline{\Omega_1} \end{array} \right.$$

Le Schwarz additive

$$\left\{ \begin{array}{l} (-\Delta + 1)(u_1^{n+1}) = f \text{ dans } \Omega_1 \\ \frac{\partial}{\partial n} u_1^{n+1} = 0 \text{ sur } \partial\Omega_1 \cap \partial\Omega \\ u_1^{n+1} = u_2^n \text{ sur } \partial\Omega_1 \cap \overline{\Omega_2} \end{array} \right. \quad \left\{ \begin{array}{l} (-\Delta + 1)(u_2^{n+1}) = f \text{ dans } \Omega_2 \\ \frac{\partial}{\partial n} u_2^{n+1} = 0 \text{ sur } \partial\Omega_2 \cap \partial\Omega \\ u_2^{n+1} = u_1^n \text{ sur } \partial\Omega_2 \cap \overline{\Omega_1} \end{array} \right.$$

La solution globale :

$$u^{n+1} = 1_{\Omega_1} u_1^{n+1} + 1_{\Omega_2} u_2^{n+1}$$

La méthode de Schwarz additive corrective

La forme corrective du problème continu.

Avec $\delta u_i^{n+1} = u_i^{n+1} - u^n$

$$\left\{ \begin{array}{l} (-\Delta + 1)\delta u_1^{n+1} = \Delta u^n + f \text{ dans } \Omega_1 \\ \frac{\partial}{\partial n}\delta u_1^{n+1} = 0 \text{ sur } \partial\Omega_1 \cap \overline{\Omega_2} \\ \delta u_1^{n+1} = 0 \text{ sur } \partial\Omega_1 \cap \partial\Omega \\ \\ (-\Delta + 1)\delta u_2^{n+1} = \Delta u^n + f \text{ dans } \Omega_2 \\ \frac{\partial}{\partial n}\delta u_2^{n+1} = 0 \text{ sur } \partial\Omega_2 \cap \overline{\Omega_1} \\ \delta u_2^{n+1} = 0 \text{ sur } \partial\Omega_2 \cap \partial\Omega \end{array} \right.$$

$$u^{n+1} = u^n + \delta u_1^n + \delta u_2^n$$

Avantage : Implémentation plus facile.

Méthode de Schwarz additive (matricielle)

Une méthode de résolution :

- On pose $\bar{\Omega}_j =$ les nœuds voisins d'un nœud de Ω_j .
- $A_h|_{\Omega_j} = a_{kl}, k, l \in \Omega_j$ correspond au problème avec condition de Dirichlet sur $\bar{\Omega}_j - \Omega_j$
- R_j projette canoniquement un vecteur défini sur Ω vers un vecteur sur Ω_j
- On pose : $M^{-1} = \sum_{i=1}^n B^i$, $B^i = R_i^T (A_h|_{\Omega_i})^{-1} R_i$
- Soit V^n la solution discrète du problème discrétisé.
- La forme discrétisée du problème correctif s'exprime ainsi :

$$V^{n+1} = V^n + M^{-1}(f - A_h V^n).$$

- Scalabilité forte ou “speadability” : Le temps de calcul est divisé par deux avec deux fois plus de processeurs.
- Scalabilité faible : Le temps de calcul est le même si pour un problème deux fois plus grand, on utilise deux fois plus de processeurs.
- Une méthode de décomposition de domaine (DDM) est scalable si son taux de convergence ne se dégrade pas quand le nombre de sous-domaines croît.

Scalabilité de Schwarz additif

- $k(M^{-1}A_h) \leq \frac{N}{C_0}$
N représente le nombre de sous-domaines.
cf. par ex. [Y. Saad, Iterative methods for sparse linear systems].
- Pour un problème deux fois plus grand le conditionnement augmente.
- Le nombre d'itérations augmente.
- La méthode de Schwarz précédente n'est pas scalable.

3. PRECONDITIONNEUR GROSSIER

Ajout d'un opérateur appelé : Solveur grille grossière.

- $M^{-1} = A_0^{-1} + \sum_{i=1}^n A_{|\Omega_i}^{-1}$.

[S. Brenner : Two-level additive schwarz preconditioners for plate elements, Numerische Mathematik, 72 :4, 1994].

- $k(M^{-1}A_h) \leq C(1 + (\frac{H_0}{\delta})^4)$

H_0 représente la taille du maillage grossier
 δ représente le degré de recouvrement.

- Convergence quasi-indépendante du nombre de sous domaines.

Construction de l'opérateur grossier

On cherche une méthode plus algébrique, sans maillage grossier.

Formulation :

- V l'espace fin.
- $V_0 \subset V$ l'espace grossier .
- $V_0 = [\Phi_1 \cdots \Phi_N]$.
- Z un opérateur de prolongement de V_0 dans V .
- Z^T un opérateur de restriction de V dans V_0 .
- $Z^T A_h Z u_H = Z^T f_h$ le système grossier.

Méthode de déflation

La méthode de déflation a été introduite par Nicolaides : [R. A. Nicolaides, Deflation of conjugate gradients with applications to boundary value problem , SIAM J.Numer.Anal., 24 (1987),pp.355-365]

Utilisé par Saad et al. ; [Y. Saad, M. Yeung, J. Erhel, and F. Guyomarc'H, A deflated version of the conjugate gradient algorithm SIAM J.Sci Comput. Vol. 21, No. 5, 2000 pp.1909-1926]

et par Aubry et al. : [R. Aubry , F. Mut , R. L ohner , J. R. Cebral, Deflated preconditioned conjugate gradient solvers for the Pressure-Poisson equation, Journal of Computational Physics, v.227 n.24, p.10196-10208, December, 2008]

Méthode de déflation

Résolution du système linéaire $A_h u = f_h$.

Soit la projection

$$P_D = I_n - A_h Z (Z^T A_h Z)^{-1} Z^T \text{ avec } A_h \in R^{n \times n} \text{ et } Z \in R^{n \times r}$$

La méthode de déflation consiste à :

- La résolution d'un système grossier : $Z^T A_h Z u_H = Z^T f_h$
- La résolution d'un système projeté : $P_D A_h \check{u} = P_D f_h$.

Calcul de u :

$$u = (I_n - P_D^T) u + P_D^T u = Z (Z^T A_h Z)^{-1} Z^T f_h + P_D^T \check{u}.$$

Balancing Domain Decomposition

Introduction :

- J. Mandel : Balancing domain decomposition, Comm. Numer. Methods Engrg., 9 (1993), pp. 233–241.

- P. Le Tallec, J. Mandel, M. Vidrascu- Balancing Domain Decomposition for Plates. Eighth International Symposium on Domain Decomposition Methods for Partial Differential Equations, Penn State, October 1993, Contemporary Mathematics, 180, AMS, Providence, 1994, 515-524.

Reformulation :

- C. Vuik , R. Nabben A comparison of deflation and the balancing preconditionner, SIAM J.SCI. Comput. Vol.27, No. 5, (2006) pp 1742-1759

■ Avec $P = (I - P_D^T)$

■ $T = M^{-1}$ un préconditionneur (ex. : ILU)

■ $P_B = P_D^T M^{-1} P_D + Z E^{-1} Z^T$.

où $E = Z^T A_h Z$

■ $Z E^{-1} Z^T$ correspond à l'espace grossier V_0

Une analyse : Z définie par des vecteurs propres

■ Définition de Z

- Soit $spectre(A_h) = \{\lambda_1 \dots \lambda_p\}$ l'ensemble des valeurs propres de A ordonnées de manière croissante.
- Soit v_i le vecteur propre de norme 1 associé à la valeur propre λ_i
- Supposons que

$$Z = \{v_1, \dots, v_r\}$$

Alors :

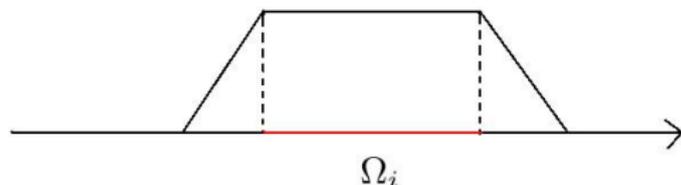
- $Spectre(P_D A_h) = \{0, \dots, 0, \lambda_{r+1} \dots \lambda_p\}$
- $Spectre(P_B A_h) = \{1, \dots, 1, \lambda_{r+1} \dots \lambda_p\}$

Définition de la base caractéristique

- Quand la déflation est combinée avec une méthode de Schwarz le choix canonique de la base est le suivant :

$$\Phi_i(x_j) = 1 \text{ si } x_j \in \Omega_i$$

- Dans le cas des éléments finis P^1 continus, chaque fonction de base Φ_i est interpolée linéairement sur les éléments. On continue de l'appeler "base de fonctions caractéristique".



Non consistance des fonctions caractéristiques

Problème grossier :

$$U^H(x) = \sum_i U_i \Phi_i(x)$$

$$\int \nabla U^H \nabla \Phi_i = \int f \Phi_i \quad \forall i$$

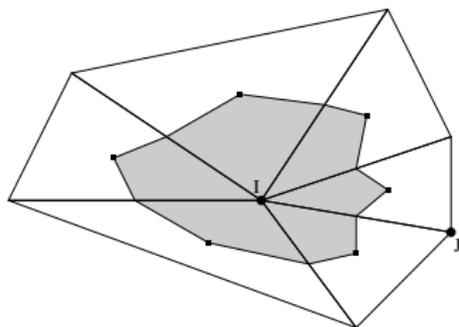
N. Marco, B. Koobus, and Dervieux. An Additive Multilevel Preconditioning Method.
Journal of Scientific Computing, 12, no 3, 233-251, 1997 :

U^H ne converge pas vers la solution continue U

Construction hiérarchique d'une grille grossière

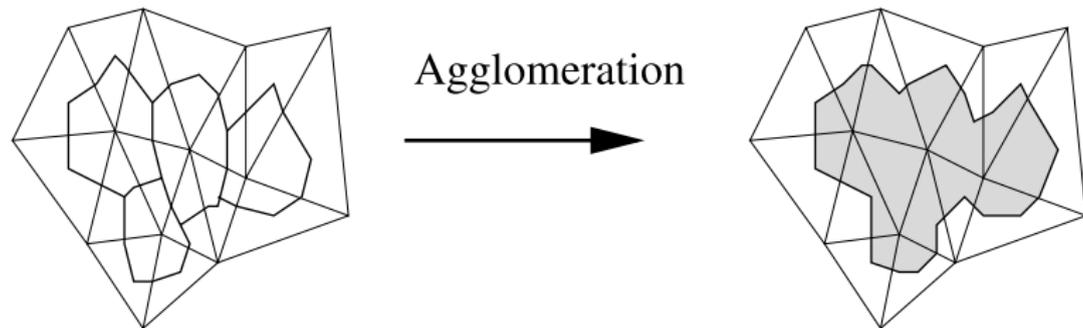
Construction de la grille grossière à partir d'un maillage en triangles :

- Partition en cellules duales fines construites en joignant les médianes des triangles.
- Passage à un niveau plus grossier par agglomération de cellules.



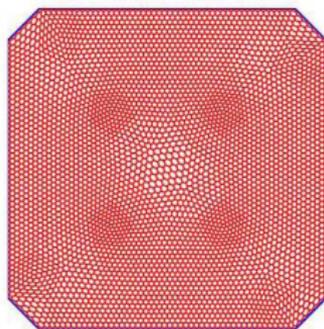
Construction de la grille grossière, suite

- Des cellules voisines sont agglomérées en une cellule grossière.
- En ré-itérant ce procédé on obtient une partition en cellules de plus en plus grossières.



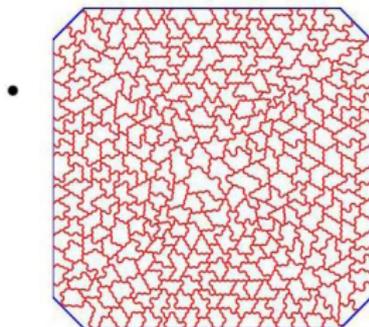
Construction des grilles grossières

Première grille grossière $\mathcal{G}_1 = \bigcup C_{1k}$



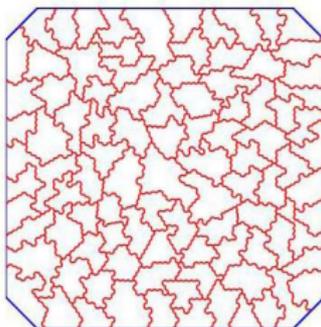
Construction des grilles grossières

Deuxième grille grossière $\mathcal{G}_2 = \bigcup_k \mathcal{C}_{2k}$



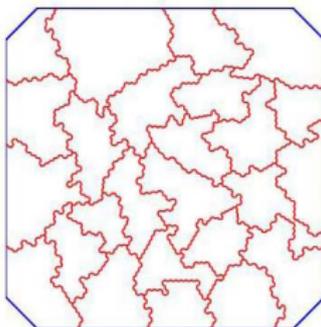
Construction des grilles grossières

Troisième grille grossière $\mathcal{G}_3 = \bigcup_k \mathcal{C}_{3k}$



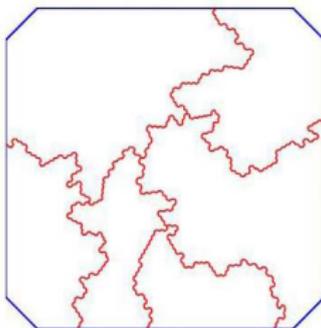
Construction des grilles grossières

Quatrième grille grossière $\mathcal{G}_4 = \bigcup_k \mathcal{C}_{4k}$



Construction des grilles grossières

Cinquième grille grossière $\mathcal{G}_5 = \bigcup_k \mathcal{C}_{5k}$



Fonctions caractéristiques

- Les macro-cellules les plus grossières sont prises comme sous-domaines. On obtient le niveau j :

$$\mathcal{G}_j = \cup_{k=1}^{N_j} C_{jk} \text{ et } k \in [1 \dots N_j]$$

- Les fonctions caractéristiques des cellules grossières C_{jk} sont prises comme fonctions de bases dans Balancing :

$$\Phi_k(x_i) = \begin{cases} 1 & \text{si } x_i \in C_{jk} \\ 0 & \text{sinon} \end{cases}$$

Opérateur de transfert

Opérateur de restriction entre la grille grossière fine \mathcal{G}_i et moins fine \mathcal{G}_j .

- On définit l'opérateur de transfert P dans le cas où $\mathcal{G}_i \subset \mathcal{G}_j$:

$$P_i^j : \mathcal{G}_i \rightarrow \mathcal{G}_j$$

$$P_i^j(u)(C_{k'i}) = u(C_{kj}) \text{ avec } C_{k'i} \subset C_{kj}$$

Opérateur de lissage pour une Grille grille grossière \mathcal{G}_k

Lissage indépendant de l'équation aux dérivées partielles (degré < 2) :

N. Marco, B. Koobus and A. Dervieux, An additive multilevel preconditioning method and its application to unstructured meshes, Journal of Scientific Computing, 1997

- Soit L_k l'opérateur de lissage :

$$(L_k u)_i = \sum_{j \in \mathcal{N}(i) \cup \{i\}} \text{Mes}(j) u_j / \left\{ \sum_{j \in \mathcal{N}(i) \cup \{i\}} \text{Mes}(j) \right\}$$

où $\mathcal{N}(i)$ représente l'ensemble des cellules voisines de la cellule i

- Le lissage est appliqué à chaque transfert de niveau en niveau.

Construction des fonctions lisses à partir des fonctions caractéristiques

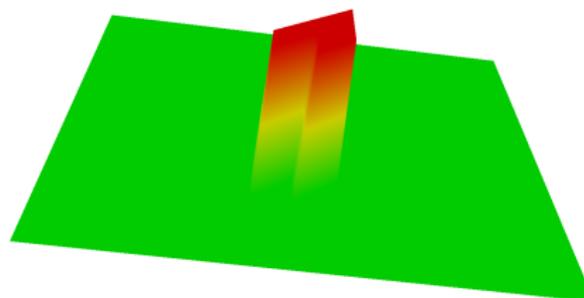
- Soit p grilles grossières : $\mathcal{G}_1 \cdots \mathcal{G}_p$
- Soit $\tilde{\Phi}_k$ une fonction caractéristique définie sur le niveau p .
On peut écrire sa représentation sur le niveau fin comme suit :

$$\Phi_k = (P_1^2 \cdots P_{p-1}^p) \tilde{\Phi}_k$$

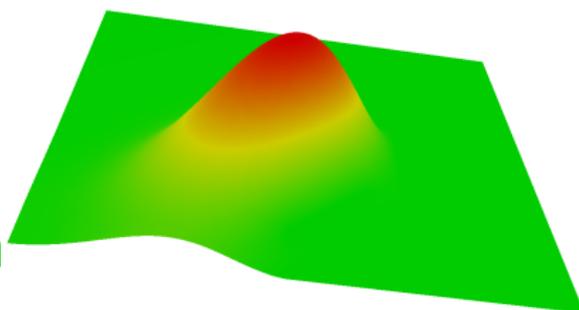
- Nous définissons la fonction de base lisse Ψ_k comme suit :

$$\Psi_k = (L_1 P_1^2 L_2 \cdots P_{p-2}^{p-1} L_{p-1} P_{p-1}^p) \tilde{\Phi}_k$$

Définition de la base grossière



Fonction de base caractéristique



Fonction de base lissée

Consistance de la base grossière

Solution grossière en base caractéristique :

$$U_c^H(x) = \sum_i U_{c,i} \Phi_i(x)$$

$$\int \nabla U_c^H \nabla \Phi_i = \int f \Phi_i \quad \forall i$$

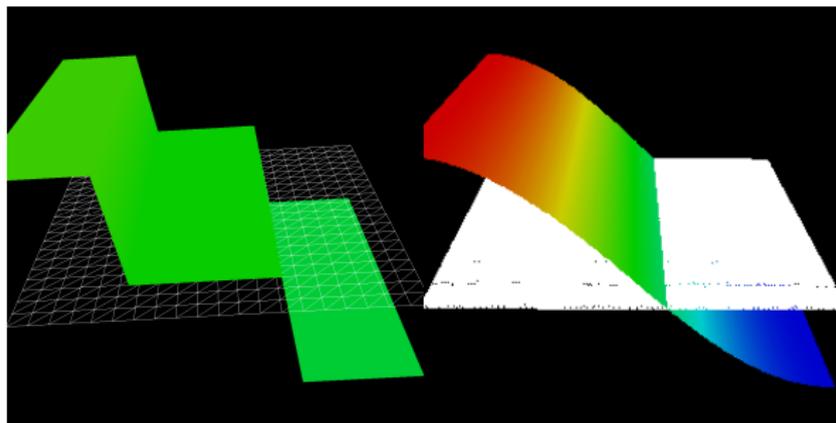
Solution grossière en base lisse :

$$U_l^H(x) = \sum_i U_{l,i} \Psi_i(x)$$

$$\int \nabla U_l^H \nabla \Psi_i = \int f \Psi_i \quad \forall i$$

Base discontinue vs lisse

Vérification sur le recalcul d'une sinusoïde (amplitude 2.)



- À gauche : solution grossière avec base caractéristique (amplitude ≈ 0.06)
- À droite : solution grossière avec base lisse (amplitude ≈ 1.8).

Problème

- On considère la résolution du problème de Neumann 2D :
- $-\Delta u + u = (\pi^2 + 1)\cos(\pi x)$ dans Ω ; $\partial u / \partial n = 0$ sur $\partial\Omega$
- $-\Delta u + u = \text{Dirac}$ dans Ω ; $\partial u / \partial n = 0$ sur $\partial\Omega$

Mise en œuvre pratique de Schwarz

Résolution complète du système local :

- coûteuse en méthode directe
- coûteuse en méthode itérative

On remplace A_i^{-1} par une inversion approximative :

$$A_i^{-1} \text{ exact} \longrightarrow (ILU(A_i))^{-1}$$

Conséquences possibles :

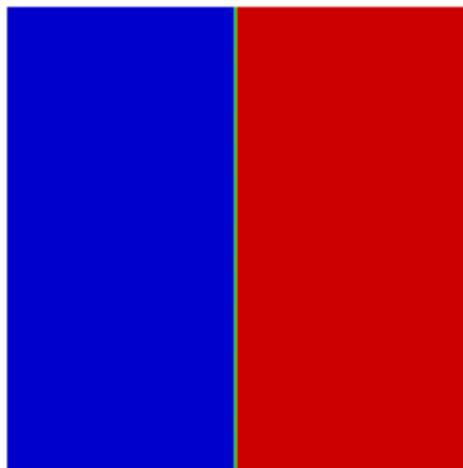
- perte de convergence trop forte vis-à-vis du Schwarz,
- impact sur la scalabilité.

Discrétisation et maillages

- Une discrétisation par éléments finis P_1 continus.
- Maillage uniforme anisotrope de 10010 sommets avec un étirement d'un facteur 10.
- Maillage cartésien isotrope de 441 sommets.
- Maillage cartésien isotrope de 10000 sommets.
- Maillage irrégulier isotrope de 11973 sommets

Schwarz à **deux** sous-domaines :

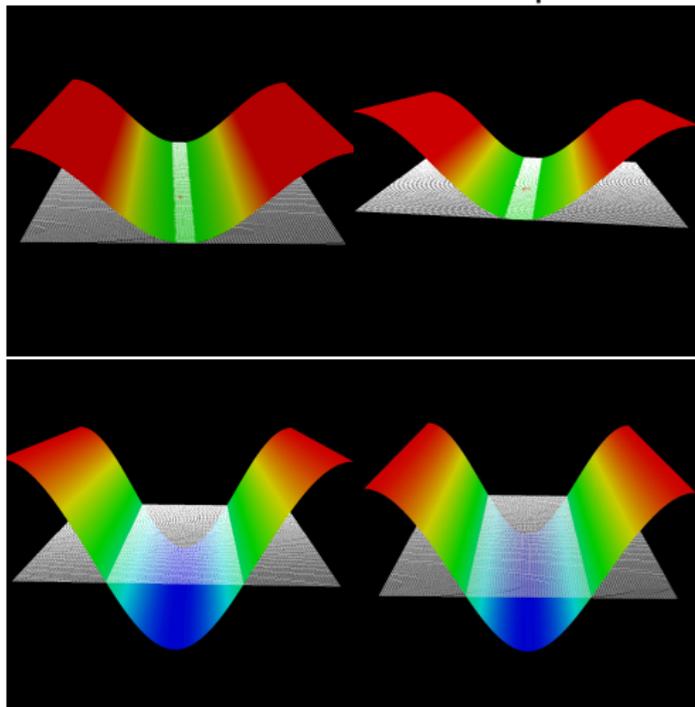
Décomposition du domaine



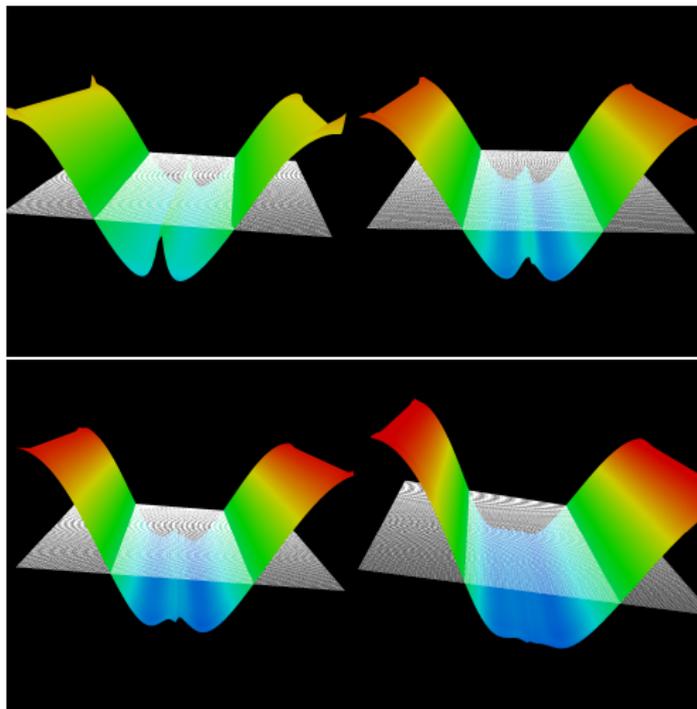
Solution : un cosinus.

Quatre premières itérations avec solveur local exact

La forme de la solution est trouvée dès la première itération



Quatre premières itérations avec un solveur ILU



Difference entre une inversion ILU et une inversion exacte du système local

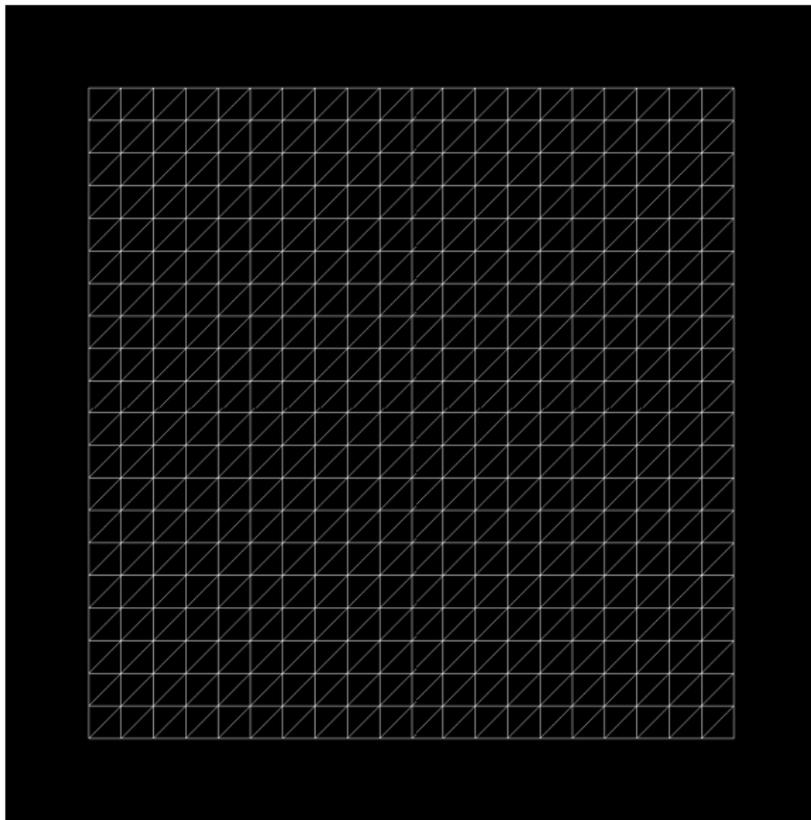
nbre de domaine	nbre sommets	solveur	nbre Iterations
2	400	ILU	55
2	400	EXACTE	28
2	10000	ILU	221
2	10000	EXACTE	74

La perte en vitesse de convergence est élevée et croît avec le nombre de sommets par domaine.

Préconditionneur et Algorithme.

- **Préconditionneur** : ILU, Balancing et Schwarz additive.
- **Algorithme** : BicgStab et Gradient conjugué.

Maillage uniforme isotrope : 441 nœuds

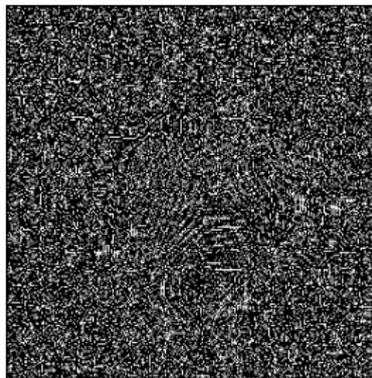


Résultats pour le maillage isotrope à 441 sommets

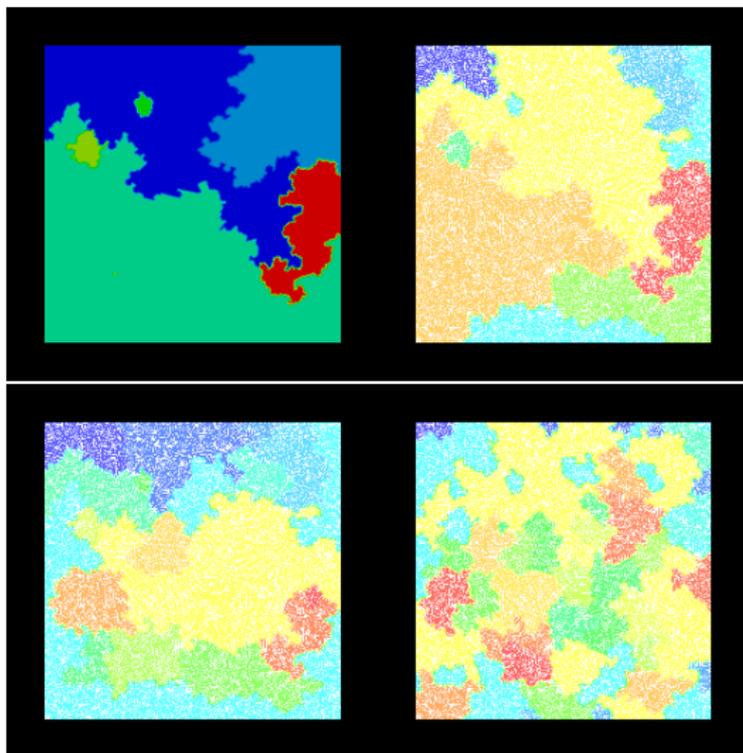
Méthode	Préconditionneur	# Sous domaines	Fcn de base	# Itérations (res. < 10^{-20})
CG	none	1	none	>6000
CG	ILU	1	none	93
CG+BDD	ILU	8	caract.	54
CG+BDD	ILU	8	lisse	46
CG+BDD	ILU	32	caract.	66
CG+BDD	ILU	32	lisse	46
CG+BDD	ILU	127	caract.	50
CG+BDD	ILU	127	lisse	45

Maillage isotrope

- Maillage isotrope.
- 11973 sommets.

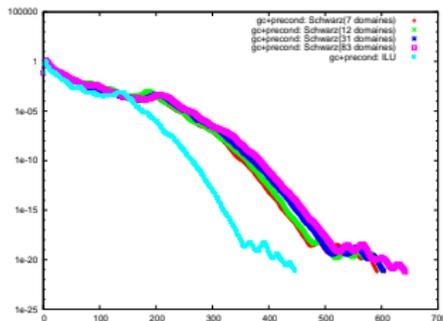


Domaines



Gradient conjugué + Schwarz

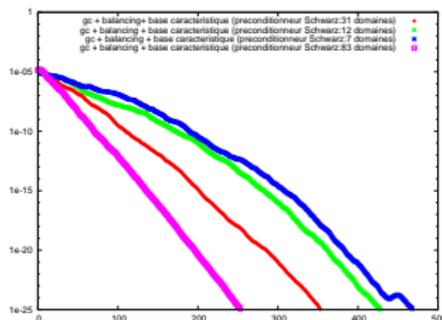
- Préconditionneur Schwarz additif sans grille grossière :
- bleu clair : préconditionnement ILU en mono-domaine.
- rouge : Schwarz additif avec 7 domaines.
- vert : Schwarz additif avec 12 domaines.
- bleu foncé : Schwarz additif avec 31 domaines.
- violet : Schwarz additive avec 83 domaines



- Convergence ralentie par rapport à ILU mono-domaine.
- Convergence plus lente avec plus de domaines.

Gradient conjugué+Schwarz+Balancing caract.

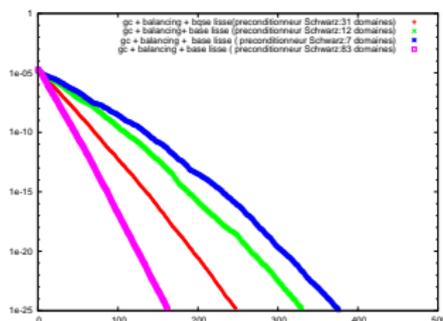
- Préconditionneur : Schwarz additif + balancing.
- Base caractéristique :
 - La convergence s'accélère avec le nombre de domaines,
 - meilleure que ILU mono-domaine à partir de 12 domaines.



- bleu : Schwarz additive avec 7 domaines.
- vert : Schwarz additive avec 12 domaines.
- rouge : Schwarz additive avec 31 domaines.
- violet : Schwarz additive avec 83 domaines

Balancing method

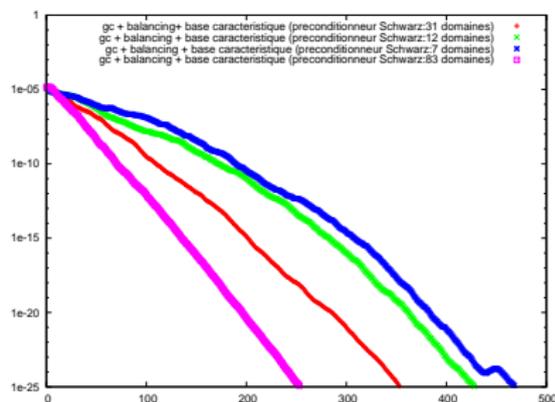
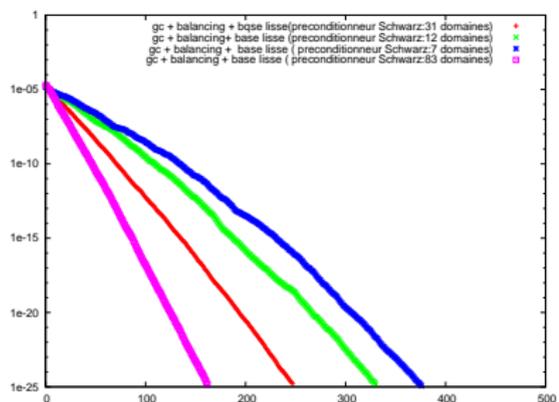
- Préconditionneur : Schwarz additive.
- Base lisse.
 - La convergence avec 7 domaines est meilleure qu'en ILU mono-domaine,
 - La convergence est meilleure qu'avec des fonctions caractéristiques.



- bleu foncé : Schwarz additive avec 7 domaines.
- vert : Schwarz additive avec 12 domaines.
- rouge : Schwarz additive avec 31 domaines.
- violet : Schwarz additive avec 83 domaines

Comparaison des bases lisses et caractéristiques

■ 100 itérations de différence.

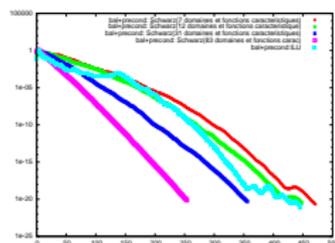
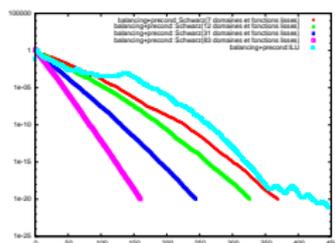


■ Figure de gauche : base lisse

■ Figure de droite : base caractéristique

Schwarz+Balancing, synthèse :

- Base caractéristique : à partir de 12 domaines Schwarz + balancing est aussi ou plus efficace que ILU mono-domaine.
- Base lisse : toujours plus efficace.



- Figure de gauche : base lisse
- Figure de droite : base caractéristique

5. BI CG STAB

x_0 la condition initial ; $r_0 = b - Ax_0$; $K = K_1 K_2$

\bar{r}_0 est un vecteur arbitraire ; $(\bar{r}_0, r_0) \neq 0$

$\rho_0 = \alpha = \omega_0 = 1$ et $v_0 = p_0 = 0$

for $i=1,2,3$

$\rho_i = (\bar{r}_0, r_{i-1})$; $\beta = (\rho_i / \rho_{i-1})(\alpha / \omega_{i-1})$

$p_i = r_{i-1} + \beta(p_{i-1} - \omega_{i-1}v_{i-1})$

résoudre $Ky = p_i$

$v_i = Ay$

$\alpha = \rho_i / (\bar{r}_0, v_i)$

$s = r_{i-1} - \alpha v_i$

résoud $Kz=s$

$t=Az$

$w_i = (K1^{-1}t, K1^{-1}s) / (K1^{-1}t, K1^{-1}t)$

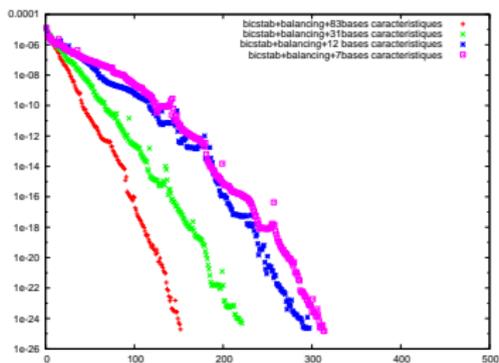
$x_i = x_{i-1} + \alpha y + \omega_i z$

Si x_i est assez précis alors quitter

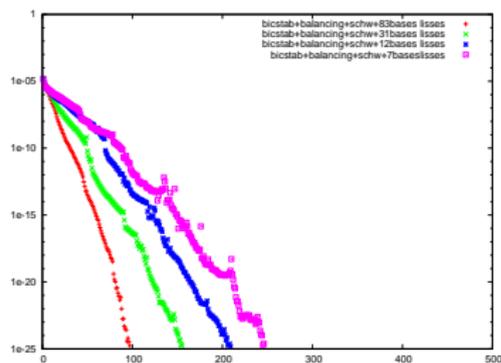
$r_i = s - \omega_i t$.

On gagne lorsque la vitesse de convergence du bicstab est deux fois meilleure que celle du gradient conjugué.

Bicgstab combiné à Balancing



Base caractéristique

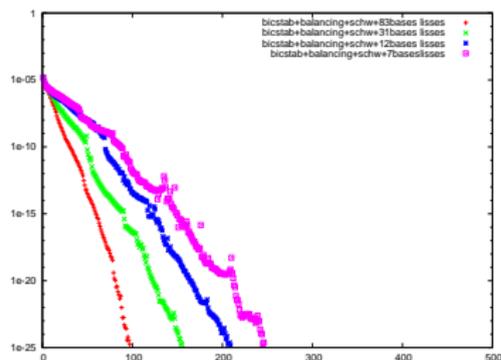
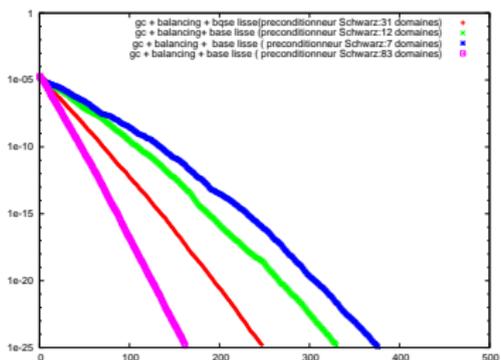


Base lisse

- violet : Schwarz additive avec 7 domaines.
- bleu foncé : Schwarz additive avec 12 domaines.
- vert : Schwarz additive avec 31 domaines.
- rouge : Schwarz additive avec 83 domaines.

Comparaison BicgStab+balancing et gradient Conjugué+balancing

- Le bicgstab ne converge pas deux fois plus vite que le gradient conjugué.



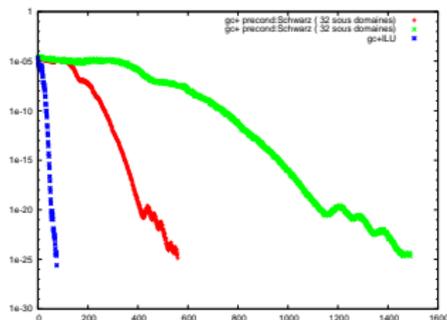
- Figure de gauche : gradient conjugué
- Figure de droite : bicgstab

6. APPLICATION A UN MAILLAGE ANISOTROPE

- Type de maillage rencontré dans les couches limites.
- 10010 sommets.
- Etirement d'un facteur 10.

Gradient conjugué avec un préconditionneur Schwarz

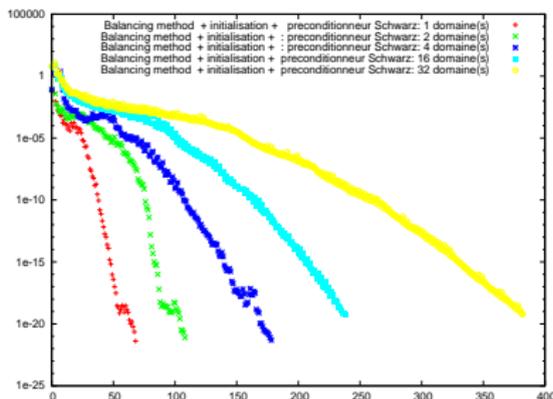
- Le mono-domaine préconditionné ILU est très rapide.
- L'apport de Balancing (lisse) double la convergence du Schwarz additif.



- vert : Schwarz additif avec 32 domaines.
- rouge foncé : Balancing Schwarz additif avec 32 domaines.
- bleu : Gradient conjugué + ILU

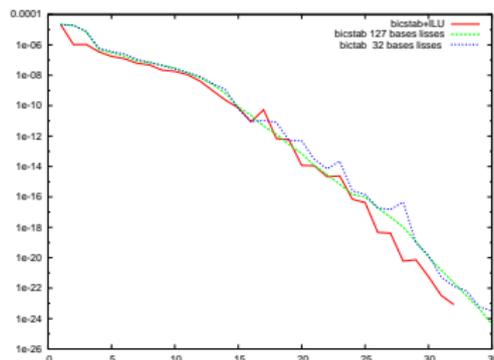
Maillage anisotrope : étiré

- Méthode Balancing + gradient conjugué.
- Découpage vertical du domaine carré.
- Convergence dégradée selon le nombre de sous domaines.



Une solution possible

- Changement du partitionnement : découpage horizontal.



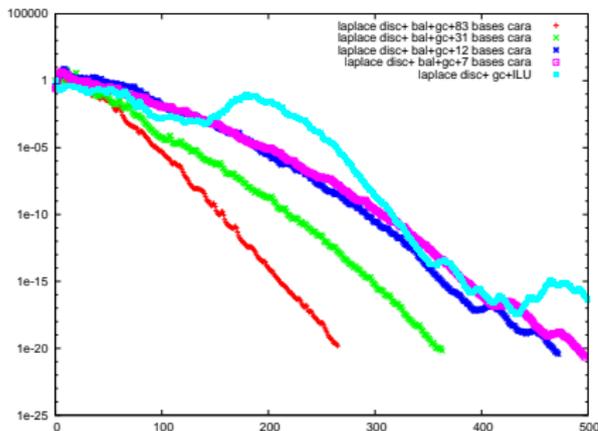
- Rouge : ILU mono-domaine,
- Vert et bleu : Schwarz+Balancing-Base-lisse (32 et 127 domaines)

Autres approches (combinables)

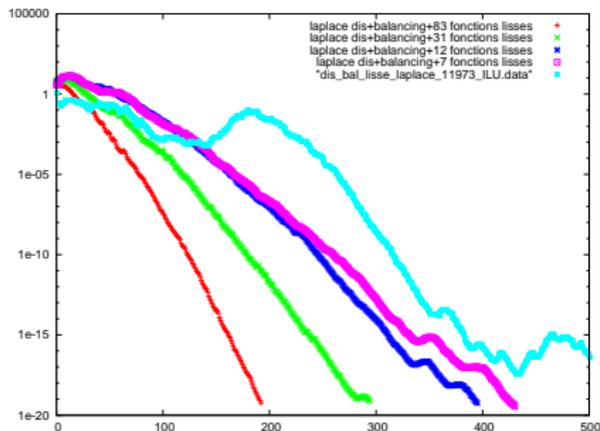
- Préconditionnement par linelet (Lohner).
- Améliorer l'efficacité du ILU : changement de numérotation.

Résolution du problème de Poisson pour la projection

Equation de Poisson à coefficients discontinus (1 :1000) avec conditions de Neumann + 1 point Dirichlet. Maillage irrégulier de 11973 nœuds.



Fcns Caractéristiques



Fcns Lisses

Scalabilité faible

nb points	10000	20000	47000	94000
nb domaines	12	28	66	142
nb points par domaines	833	714	712	661
cv fct lisses	400	391	444	491
cv caractéristiques	480	546	750	810

- Bonne scalabilité pour les fonctions lisses.
- Mauvaise scalabilité pour les fonctions caractéristiques.

Conclusion Schwarz+Balancing :

Les deux types de base améliorent efficacité et scalabilité. La consistance des grilles grossières apporte :

- Une bonne initialisation.
- L'accélération de la convergence.
- Construction de base consistante.
- Amélioration de la scalabilité.

Mais à grand nombre de nœuds par sous-domaine, l'option ILU se révèle progressivement insuffisante (491 >> 93).

8. METHODE A TROIS NIVEAUX

Motivation :

Diminuer la sensibilité de la convergence au nombre de nœuds par sous-domaine.

Approche :

Ajouter une grille grossière dans chaque sous-domaine, à terme on voudrait une fonction de base locale pour (par ex.) 1000 points.

Cf. :

P.T . Lin, M. Sala, J N. Shadi, R S. Tuminaro, Performance of Fully-Coupled Algebraic Multilevel Domain Decomposition Preconditioners for Incompressible Flow and Transport : **Méthode à trois niveaux avec la librairie ML en non-smoothed aggregation.**

Définitions des trois niveaux

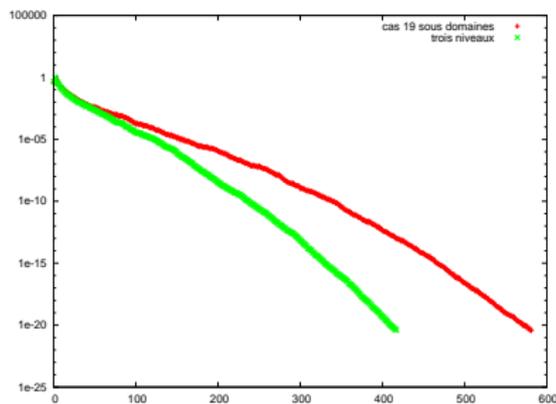
- Le niveau fin est défini comme précédemment
- Construction de l'espace grossier global V_{GG}
 - Définition des sous domaines Ω_i .
 - Création de l'espace moyen a l'aide des sous domaines
 - Les fonctions de bases sont les fonctions caractéristiques par sous domaines.
- Construction des sous-espaces grossiers locaux V_{G_i}
 - décomposition du sous domaines Ω_i en zone $(\Omega_{ij})_j \quad j = 1..n_i$
 - Construction des fonctions caractéristiques $(\Phi_{ij})_j \quad j = 1..n_i$ pour les $(\Omega_{ij})_j \quad j = 1..n_i$
 - Lissage des $(\Phi_{ij})_j \quad j = 1..n_i$
 - $V_{\Omega_i} = \text{vect}(\Phi_{ij})_j$

Options pour résoudre le système grossier local

- Option 1 (parallèle) : calcul local des $A_{|\Omega_i}^{-1}$
 - Résolution du problème grossier local et correction.
 - Résolution approximative ILU du problème corrigé.
- Option 2 (pour ce démonstrateur) :
Schwarz + Balancing avec l'union des deux bases.
On note que, les bases grossières étant nulles au voisinage de l'interface, les différentes bases (globale, locales) sont indépendantes.
[Algorithme testé ici.](#)

Comparaison deux et trois niveaux

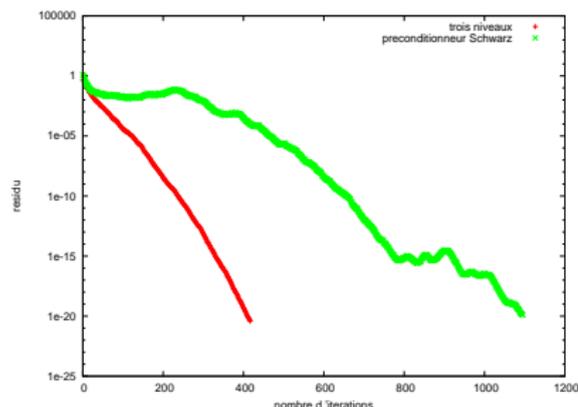
- Maillage à 90000 nœuds irrégulier.
- Balancing avec 19 domaines :
- rouge : deux niveaux dont un niveau grossier a 19 fonctions de base caractéristiques,
- vert : trois niveaux dont un niveau grossier a 19 fonctions de base caractéristiques et 341 fonctions de bases lisses réparties dans les 19 sous-domaines



La méthode à trois niveaux apporte une amélioration de la convergence

Preconditionneur Schwarz et Balancing 3-niveaux

La convergence est nettement améliorée par rapport au preconditionneur Schwarz additif



- vert : preconditionneur Schwarz
- rouge : balancing 3-niveaux.

9. CONCLUSIONS ET PERSPECTIVES

- Nous avons identifié une approche pour une double scalabilité :
- maîtrise de la convergence quand le nombre de processeurs croît.
 - maîtrise de la convergence quand le nombre de nœuds par processeur croît (à valider plus complètement).

Pour les applications 2D envisagées, nous obtenons assez couramment une convergence deux fois meilleure qu'un simple Schwarz, pour un petit nombre d'opérations supplémentaires (à évaluer avec plus de précision quand le maillage sera changé au cours du temps).

Travail en cours : optimisation des conditions d'interfaces (vers un "Robin").