

Domaine SxS

BE : Méthodes numériques multidomaines

François Rogier (Francois.Rogier@onera.fr)

Introduction

L'objectif de ce BE est d'étudier la méthode de Schwarz pour la résolution du Laplacien en 2D. Ce BE comporte un travail théorique suivi d'un travail d'implémentation et d'analyse des résultats numériques. L'évaluation se fera à la fin du BE sur l'avancée de l'implémentation et également sur le travail d'analyse à rendre sur un document écrit.

Partie théorique

L'équation de Poisson que l'on souhaite résoudre s'écrit :

$$\begin{cases} -\Delta u(x, y) &= \sin(\pi x) * \sin(\pi y) & (x, y) \in]0, 1[\times]0, 1[\\ u(0, y) &= u(1, y) = 0 & \forall y \in]0, 1[\\ u(x, 0) &= u(x, 1) = 0 & \forall x \in]0, 1[\end{cases} \quad (0.1)$$

1. Vérifier que la solution u de (0.1) est de la forme $v(x) * \sin(\pi y)$ où v satisfait à l'équation différentielle du second ordre :

$$\begin{cases} -\frac{d^2}{dx^2} v(x) + \pi^2 v &= \sin(\pi x) & x \in]0, 1[\\ v(0) &= v(1) = 0 \end{cases} \quad (0.2)$$

2. On s'intéresse à résoudre (0.2) par une méthode de décomposition de domaines avec recouvrement, avec deux sous-domaines, et dont la largeur de recouvrement est égale à 2δ . L'algorithme de Schwarz s'écrit alors :

$$\begin{cases} -\frac{d^2}{dx^2} v_1^{n+1}(x) + \pi^2 v_1^{n+1} &= \sin(\pi x) & x \in]0, 1/2 + \delta[& v_1^{n+1}(1/2 + \delta) = v_2^n(1/2 + \delta), & v_1^{n+1}(0) = 0 \\ -\frac{d^2}{dx^2} v_2^{n+1}(x) + \pi^2 v_2^{n+1} &= \sin(\pi x) & x \in]1/2 - \delta, 1[& v_2^{n+1}(1/2 - \delta) = v_1^n(1/2 - \delta), & v_2^{n+1}(1) = 0 \end{cases}$$

Montrer que l'algorithme de Schwarz converge géométriquement avec une raison q que l'on déterminera. Que se passe-t-il quand $\delta = 0$?

Partie Numérique I

On considère une discrétisation par différences finies de l'équation de Poisson sur un maillage cartésien : $x_i = (i - 1)h_x, y_j = (j - 1)h_y, i = 1, \dots, ntx + 1, j = 1, \dots, nty + 1$ avec $h_x = \frac{1}{ntx}$ et $h_y = \frac{1}{nty}$. Après discrétisation par différences finies, on trouve :

$$\begin{cases} \left(\frac{2}{h_x^2} + \frac{2}{h_y^2} \right) u_{i,j} - \frac{1}{h_x^2} (u_{i+1,j} + u_{i-1,j}) - \frac{1}{h_y^2} (u_{i,j+1} + u_{i,j-1}) = f(x_i, y_j), \\ \quad 2 \leq i \leq ntx, 2 \leq j \leq nty \\ u_{i,1} = u_{i,nty+1} = u_{1,j} = u_{ntx+1,j} = 0, \\ \quad i = 1, \dots, ntx + 1, j = 1, \dots, nty + 1 \end{cases}$$

On remarquera que les points (i, j) pour lesquels $i = 1, i = ntx + 1, j = 1$ ou $j = nty + 1$ permettent de prendre en compte les conditions aux limites : ce sont des points de calcul **fictifs**.

1. Mettre (par écrit) le système sous la forme de la résolution d'un système linéaire : $AU = F$ et expliciter les termes a_{ij} de la matrice. Que peut on dire de la structure de A ?
2. Calculer la solution U en s'aidant du programme principal "Main.m" et des routines "PoissonInit.m" et "PoissonRHS.m" qui permettent de calculer la matrice de rigidité et le second membre. On prendra pour les calculs $ntx = 20, nty = 20$. Visualiser la solution (on utilisera pour cela la fonction matlab "contour"). On notera dans la suite par U_e cette solution.

Partie Numérique II

On considère une méthode de décomposition de domaines "avec recouvrement". Pour cela, on découpe le domaine de calcul en $px \times py$ sous-domaines et on suppose pour simplifier l'étude que $ntx = nxb \times px$ et $nty = nyb \times py$, où nxb et nyb sont deux entiers. On repère un sous-domaine par un couple $(I, J), I = 1, \dots, px, J = 1, \dots, py$ et on considère uniquement le recouvrement entre sous-domaines ayant un côté en commun. Ainsi le sous-domaine (I, J) interagit avec les sous domaines $(I + 1, J), (I - 1, J), (I, J + 1)$ et $(I, J - 1)$ (voir figure 1). Le recouvrement est compté en nombre de mailles (identique en x ou en y) que l'on note i_δ . Ainsi si on examine (par exemple) l'interaction de (I, J) avec $(I + 1, J)$, on a :

- si $i_\delta = 0$, les domaines sont jointifs, la colonne qui se situe à l'extrémité du domaine (I, J) correspond avec la première colonne du domaine $(I + 1, J)$. Tous les points de recouvrement sont des points fictifs.
- $i_\delta = 1$, il y a 3 colonnes en commun par ligne dont une colonne de points fictifs.
- i_δ quelconque, il y a $2i_\delta + 1$ colonnes en commun par ligne dont une colonne de points fictifs.

1. Calculer le nombre de points $nx(I, J)$ et $ny(I, J)$ dans chaque sous-domaine en fonction de i_δ (on fera attention au traitement des sous-domaines ayant une frontière) calculer également les coordonnées du point bas gauche de chaque domaine que l'on notera $(x_{min}(I, J), y_{min}(I, J))$.

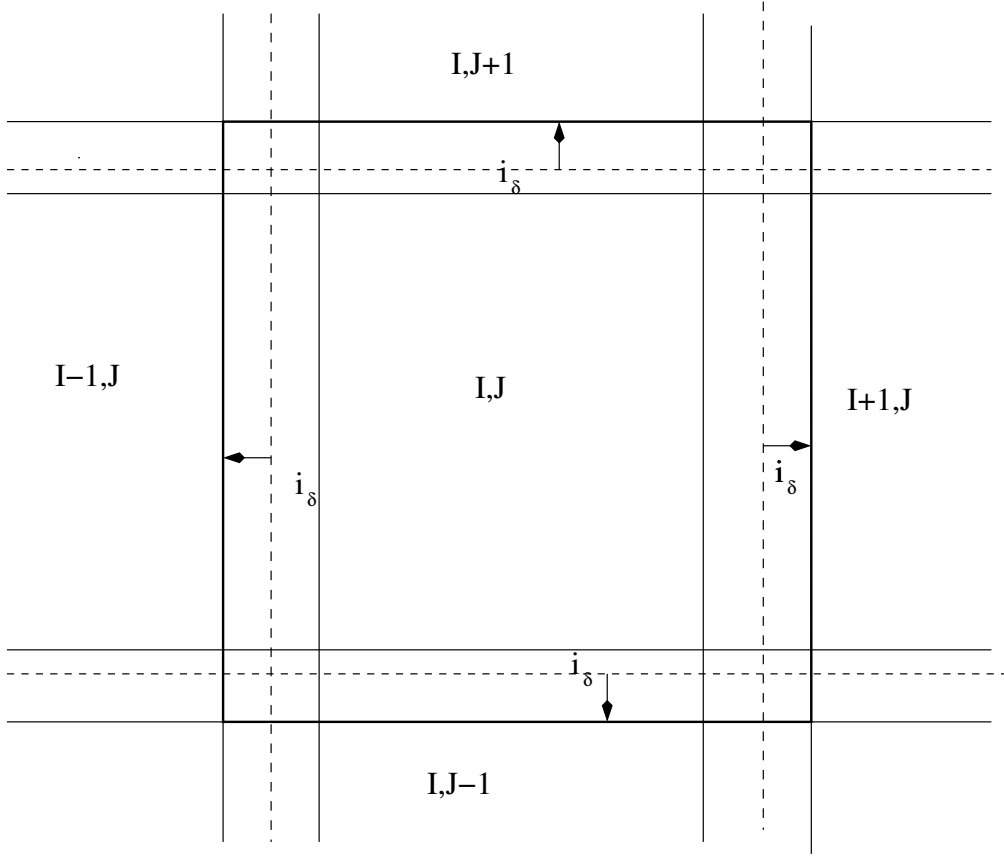


FIGURE 1 – découpage en sous-domaines et recouvrement

On utilise la structure "cell" qui permet de définir des tableaux de tableaux. Dans chacun des sous-domaines on définit la matrice de rigidité locale $A(I, J)$, le second membre local $B(I, J)$ et la solution dans le sous-domaine (I, J) à l'étape n de l'algorithme de Schwarz $U(I, J)^n$.

2. Vérifier que l'algorithme de Schwarz s'écrit :

$$\begin{cases} A(I, J)U(I, J)^{n+1} = B(I, J) + C^1(I, J)U(I+1, J)^n + C^2(I, J)U(I-1, J)^n + \\ C^3(I, J)U(I, J+1)^n + C^4(I, J)U(I, J-1)^n \end{cases}$$

Les matrices $C^l(I, J)$, $l = 1, 4$ étant creuses (par convention, on définira $U(0, J)^n = U(px+1, J)^n = U(I, 0) = U(I, py+1) = 0$). En déduire que l'on peut prendre en compte les échanges entre sous-domaines à chaque itération en modifiant le second membre $B(I, J)$.

3. Construire les matrices locales $A(I, J)$ et les seconds membres locaux $B(I, J)$ dans chacun des sous domaine (I, J) à l'aide des modules "PoissonInit" et "Poisson-RHS".

4. On suppose $U^0(I, J) = 0 \forall (I, J)$. Noter que la solution est multivaluée dans le domaine de recouvrement. On utilise la fonction "Recons" pour reconstruire une solution U^n de dimension $ntx \times nty$ dans tout le domaine de calcul à partir des contributions $U(I, J)^n$ dans chacun des sous-domaines. Visualiser la solution U^1 .
5. Ecrire une boucle mettant en oeuvre l'algorithme de Schwarz. On utilisera pour cela le module "Exchange" qui permet de mettre à jour le second membre $B(I, J)$ en fonction des nouvelles valeurs données par la solution courante dans les sous-domaines voisins.
6. Calculer $\|U_e - U^n\|$, (on note $\|V\| = \sqrt{h_x h_y \sum_{i,j} (V_{i,j})^2}$).
On se fixe comme critère de convergence :

$$\|U_e - U^n\| \leq \epsilon \|U_e\|$$

(avec $\epsilon = 1d - 6$). Vérifier la convergence de l'algorithme. Que peut-on dire de la vitesse de convergence ? Comparer avec la vitesse de convergence théorique.

Partie Numérique III

1. Etudier la convergence de l'algorithme en fonction du recouvrement i_δ . Pour cela on calculera le nombre d'itérations nécessaire à atteindre le critère d'arrêt $\epsilon = 1d - 6$ que l'on représentera sur un graphique en fonction de i_δ .
2. Réaliser une étude en fonction du découpage en px, py pour le même taux de recouvrement $i_\delta = 2$. On prendra $ntx = nty = 64$ et on fera varier p_x et p_y . Montrer que la convergence se dégrade lorsque l'on multiplie le nombre de sous-domaines.

Partie Numérique IV

On s'intéresse maintenant à l'étude de conditions aux limites de transfert de type Robin. Ces conditions s'écrivent sur la frontière :

$$\frac{\partial u}{\partial n} + \alpha u = g$$

où n est la normale extérieure au domaine et $\alpha > 0$. Ainsi le schéma de décomposition de domaines (entre deux sous-domaines) s'écrit :

$$\left\{ \begin{array}{lll} -\Delta u_1^{n+1}(x, y) & = & f(x, y), (x, y) \in \Omega_1 \\ -\Delta u_2^{n+1}(x, y) & = & f(x, y), (x, y) \in \Omega_2 \\ (\frac{\partial u_1^{n+1}}{\partial x} + \alpha u_1^{n+1})(1/2 + \delta, y) & = & (\frac{\partial u_2^n}{\partial x} + \alpha u_2^n)(1/2 + \delta, y) \forall y \in]0, 1[\\ (\frac{\partial u_2^{n+1}}{\partial x} - \alpha u_2^{n+1})(1/2 - \delta, y) & = & (\frac{\partial u_1^n}{\partial x} - \alpha u_1^n)(1/2 - \delta, y) \forall y \in]0, 1[\\ u_1^{n+1}(0, y) = u_2^{n+1}(1, y) & = & 0 \\ u_1^{n+1}(x, 0) = u_1^{n+1}(x, 1) & = & u_2^{n+1}(x, 0) = u_2^{n+1}(x, 1) = 0 \end{array} \right.$$

1. Vérifier que la solution obtenue à convergence est continue et à dérivée normale continue à travers les interfaces. En déduire que c'est bien la solution du problème considéré. On utilisera la formule de Green sur la zone de recouvrement.
2. La discrétisation des conditions de Robin est faite à l'aide de différences finies d'ordre 1. La condition aux limites de Robin devient ainsi :

$$\left\{ \begin{array}{l} u_1^{n+1}(i_\delta, j) + (1 + \alpha)h u_1^{n+1}(i_\delta, j) = u_2^n(i_\delta, j) + (1 + \alpha h)u_2^n(i_\delta, j) \end{array} \right.$$

3. on prend $\alpha = \frac{1}{h}$, étudier la convergence de la méthode et la comparer avec les conditions de Dirichlet.
4. Montrer par des calculs successifs qu'il existe une valeur de α qui optimise la convergence.