

PPSC2020F - Projet I

J.PERDIGON

3 février 2021

1 Introduction

On considère le problème de Poisson à 2 dimensions avec des conditions aux bords de Dirichlet

$$\begin{aligned} -\Delta_{\mathbf{x}}u(\mathbf{x}) &= f(\mathbf{x}) , \quad \mathbf{x} \in \Omega = (0, L_x) \times (0, L_y) \\ u(\mathbf{x}) &= \alpha(\mathbf{x}) , \quad \mathbf{x} \in \partial\Omega \end{aligned} \quad (1)$$

Dans un premier temps, on peut réécrire le problème en effectuant les changements de variables

$$\mathbf{x} = (L_x \tilde{x}, L_y \tilde{y}) , \quad \tilde{x} = \frac{x}{L_x} , \quad \tilde{y} = \frac{y}{L_y} \quad (2)$$

L'opérateur de Laplace se réécrit ainsi

$$-\Delta_{\mathbf{x}}u(\tilde{\mathbf{x}}) = \frac{1}{L_x^2} \partial_{\tilde{x}}^2 u + \frac{1}{L_y^2} \partial_{\tilde{y}}^2 u \quad (3)$$

On choisit d'approximer le problème à l'aide de différences finies. L'espace est discrétisé en une grille uniforme de $(n+1)$ points dans chaque direction avec le pas de grille $h = 1/n$. L'opérateur Laplacien Δ est quant à lui approximé par :

$$\Delta_{\mathbf{x}}u(\tilde{\mathbf{x}}) \approx \frac{1}{L_x^2} \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} + \frac{1}{L_y^2} \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h^2} \quad (4)$$

Où $u_{i,j}$ représente une approximation discrete de la solution, sur la grille d'espace définie précédemment. $u_{i,j}$ peut être représenté sous la forme d'une matrice U , de taille $m \times m$ ($m = (n+1)$) et dont les indices i et j sont respectivement les indices de lignes et de colonnes. Le problème se réécrit sous forme matricielle

$$\frac{1}{L_x^2} TU + \frac{1}{L_y^2} UT = h^2 F \quad (5)$$

Où U et T sont définies comme :

$$U = \begin{pmatrix} u_{1,1} & \cdots & \cdots & \cdots & u_{1,n-1} \\ \vdots & & & & \vdots \\ \vdots & & u_{i,j} & & \vdots \\ \vdots & & & \ddots & \vdots \\ u_{n-1,1} & \cdots & \cdots & \cdots & u_{n-1,n-1} \end{pmatrix}, \quad T = \begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & -1 & 2 & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{pmatrix} \quad (6)$$

Les matrices T et F sont respectivement la matrice de Laplace unidimensionnelle et la matrice associée au membre de droite. A noter que l'implémentation des conditions aux bords a été absorbée dans la définition de la matrice F ,

$$F = \begin{pmatrix} f_{1,1} + \frac{\alpha_{0,1}}{L_x^2} + \frac{\alpha_{1,0}}{L_y^2} & f_{1,2} + \frac{\alpha_{0,2}}{L_x^2} & \cdots & f_{1,n-2} + \frac{\alpha_{0,n-2}}{L_x^2} & f_{1,n-1} + \frac{\alpha_{0,n-1}}{L_x^2} + \frac{\alpha_{1,n}}{L_y^2} \\ f_{2,1} + \frac{\alpha_{2,0}}{L_y^2} & & \vdots & & f_{2,n-1} + \frac{\alpha_{2,n}}{L_y^2} \\ \vdots & \cdots & f_{i,j} & \cdots & \vdots \\ f_{n-2,1} + \frac{\alpha_{n-2,0}}{L_y^2} & & \vdots & & f_{n-2,n-1} + \frac{\alpha_{n-2,n}}{L_y^2} \\ f_{n-1,1} + \frac{\alpha_{n-1,0}}{L_x^2} + \frac{\alpha_{n-1,1}}{L_y^2} & f_{n-1,2} + \frac{\alpha_{n-1,2}}{L_x^2} & \cdots & f_{n-1,n-2} + \frac{\alpha_{n-1,n-2}}{L_x^2} & f_{n-1,n-1} + \frac{\alpha_{n-1,n-1}}{L_x^2} + \frac{\alpha_{n-1,n}}{L_y^2} \end{pmatrix} \quad (7)$$

Pour résoudre le système (5), on peut utiliser une méthode itérative. Cependant, il existe une méthode directe de résolution, en remarquant que l'on connaît les valeurs propres ainsi que les vecteurs propres de l'opérateur de Laplace associé à la matrice T

$$T s_i = \lambda_i s_i, \quad s_i = [\sin(i\pi h), \sin(2i\pi h), \dots, \sin(mi\pi h)]^T, \quad \lambda_i = 4 \sin\left(\frac{i\pi h}{2}\right)^2, \quad i = 1, \dots, m \quad (8)$$

Les vecteurs propres $\{S_i\}$ forment les colonnes de la matrice S . L'équation (8) se réécrit ainsi

$$TS = DS, \quad S = [s_1^T, s_2^T, \dots, s_m^T], \quad D = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_m) \quad (9)$$

La matrice S est symétrique ($S^T = S$) et orthogonale à un facteur près ($S^T S = S^2 = 1/2h I$). Ainsi, en décomposant la solution U dans la base formée par les vecteurs propres de T ($U = S\tilde{U}S$), l'équation (5) se réécrit

$$\frac{1}{L_x^2} TS\tilde{U}S + \frac{1}{L_y^2} S\tilde{U}ST = h^2 F \quad (10)$$

En multipliant à gauche et à droite de chaque terme par la matrice S

$$\frac{1}{L_x^2} STS\tilde{U}S^2 + \frac{1}{L_y^2} S^2\tilde{U}STS = h^2 \tilde{F} \quad (11)$$

Où $\tilde{F} = SFS$. En utilisant $TS = DS = SD$, on obtient

$$\frac{1}{L_x^2} S^2 D\tilde{U}S^2 + \frac{1}{L_y^2} S^2\tilde{U}S^2 D = h^2 \tilde{F} \quad (12)$$

Finalement, on obtient l'équation pour \tilde{U} en utilisant $S^2 = 1/2h I$,

$$\frac{1}{L_x^2} D\tilde{U} + \frac{1}{L_y^2} \tilde{U}D = 4h^4 \tilde{F} \leftrightarrow \tilde{U}_{i,j} = 4h^4 \frac{\tilde{F}_{i,j}}{\frac{\lambda_i}{L_x^2} + \frac{\lambda_j}{L_y^2}} = 4h^4 L_x^2 L_y^2 \frac{\tilde{F}_{i,j}}{L_y^2 \lambda_i + L_x^2 \lambda_j} \quad (13)$$

Ainsi, en se donnant une matrice F de taille $m \times m$, il faut effectuer 4 produits matriciels ($4m^3$ opérations) et un produit de matrices terme à terme (produit d'Hadamard en m^2 opérations) pour obtenir la solution U . La solution du problème de Poisson peut être calculée $\mathcal{O}(n^3)$. Ce nombre peut-être réduit, en remarquant que le produit matriciel de la matrice S avec une matrice A correspond en réalité à la DST des colonnes de cette même matrice A . La DST d'un vecteur \mathbf{v} de taille N peut être déduite de la FFT du vecteur \mathbf{v}_L de taille $2N + 2$ selon

$$DST(\mathbf{v})_k = -\frac{1}{2} \text{Imag}(FFT(\mathbf{v}_L)_{k+1}), \quad \mathbf{v}_L = [0, v_0, v_1, \dots, v_{N-1}, 0, -v_{N-1}, -v_{N-2}, \dots, -v_0]^T, \quad k = 0, 1, \dots, N-1 \quad (14)$$

De cette manière, la DST se calcule en $\mathcal{O}(N \ln(N))$ opérations. La solution du problème de Poisson est ainsi calculée en $n^2 \ln(n)$.

2 Implementation numérique

L'implémentation se fait au sein d'une classe "TwoDPoisson" qui prend en arguments n (puissance de 2), L_x et L_y . Le constructeur se charge ensuite de calculer la solution U (la variable privée u_-) en cinq étapes :

- $F_1 = S F$
- $\tilde{F} = F_1 S$
- $\tilde{U}_{i,j} = 4h^4 L_x^2 L_y^2 \frac{\tilde{F}_{i,j}}{L_y^2 \lambda_i + L_x^2 \lambda_j}$
- $U_1 = S \tilde{U}$
- $U = U_1 S$

Les produits matriciels $S F$ et $S \tilde{U}$ sont calculés à l'aide de la fonction de classe privée $dst_col()$ qui effectue respectivement la DST des colonnes de F et \tilde{U} . Les produits matriciels $F_1 S$ et $U_1 S$ sont quant à eux calculés à l'aide de la fonction $dst_row()$ qui effectue la DST des lignes de F_1 et U_1 .

2.1 Parallélisation avec MPI

On veut à présent paralléliser la classe “TwoDPoisson”. La tâche que l’on souhaite paralléliser est le calcul de la *DST* sur les colonnes/lignes de la matrice considérée. Soit P le nombre de processeurs alloué au calcul ; le nombre de colonnes/lignes assigné à chaque processeur sur lesquelles celui-ci calculera la *DST* est $M = m/P$. Le reste de colonnes/lignes $M_r = m \% P$ est pris en charge par le dernier processeur $P - 1$. Afin d’optimiser le nombre de communications entre les processeurs (au détriment de la mémoire), on opte pour la stratégie suivante ; chaque processeur $p = 0, \dots, P - 1$ possède une version de la matrice dont on veut calculer la *DST* des colonnes/lignes, et il ne calcule cependant que la *DST* des colonnes/lignes associées aux indices Mp à $M(p + 1)$ (le dernier processeur va de $M(P - 1)$ à $MP + M_r$). Chaque processeur calcule donc la *DST* d’une sous-partie de la matrice. Finalement la matrice est mise à jour dans son ensemble, grâce aux fonctions “communication_col()” et “communication_row()” où chaque processeur va communiquer aux autres processeurs la sous-partie considérée (via la fonction *MPI_Allgatherv()*). Une barrière *MPI_Barrier()* est ensuite ajoutée en sécurité afin de s’assurer que l’ensemble des versions soit mise à jour et identique, avant d’effectuer d’autre produit matriciels.

3 Cas test : $-\Delta_{\mathbf{x}}u(\mathbf{x}) = 1$, $\alpha(\mathbf{x}) = 0$