High Performance Computing

Projet final : Parallélisation de l'algorithme de Jacobi

Rémi Garde

Mars 2018

Table des matières

1 Pre-Processing **3**

1.1 Définition du problème 3

1.2 Résolution 3

1.3 Convergence 3

2 Processing **4**

2.1 Parallélisation 4

3 Post-Processing **4**

1 Pre-Processing

1.1 Définition du problème

Soit $n \in \mathbb{N}$, $A = (a_{ij})_{(i,j)\in[\![1,n]\!]^2}$ une matrice carrée de taille n, $B = (b_i)_{i\in[\![1,n]\!]}$ un vecteur de taille n. Nous cherchons à résoudre le système d'équations linéaires

$$AX = B \tag{1}$$

pour $X = (x_i)_{i\in[\![1,n]\!]}$ vecteur de taille n.

1.2 Résolution

Pour résoudre ce problème, A est séparée en 2 matrices D et R, avec D matrice de la diagonale de A et R les éléments non diagonaux de A :

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{pmatrix} = \begin{pmatrix} a_{1,1} & 0 & \cdots & 0 \\ 0 & a_{2,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{n,n} \end{pmatrix} + \begin{pmatrix} 0 & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & 0 & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & 0 \end{pmatrix}$$

Ceci est intéressant car le calcul de D^{-1} est trivial :

$$D^{-1} \begin{pmatrix} \frac{1}{a_{1,1}} & 0 & \cdots & 0 \\ 0 & \frac{1}{a_{2,2}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{a_{n,n}} \end{pmatrix}$$

La formulation (1) devient donc équivalente à $(D + R)X = B$ ce qui donne :

$$X = D^{-1}(B - RX) \tag{2}$$

Nous pouvons ensuite poser la suite $X^{(k)}, n \in \mathbb{N}$ définie comme suit :

$$\begin{cases} X^{(0)} = 0 \\ X^{(k+1)} = D^{-1}(B - RX^{(k)}), \quad k \in \mathbb{N}^* \end{cases}$$

La formule de récurrence s'écrit pour chacun des éléments de X :

$$\forall k \in \mathbb{N}^*, \forall i \in [\![1, n]\!], x_i^{(k+1)} = \frac{1}{a_{ii}}(b_i - \sum_{i\neq j} a_{ij}x_j^{(k)}) \tag{3}$$

1.3 Convergence

Déterminons une condition tel que $\lim_{k\to\infty} X^{(k)} = X$). En posant $A' = -D^{-1}R$ et $B' = D^{-1}B$
(2) devient alors :

$$X = A'X + B'$$

et ainsi

$$X^{(k+1)} - X = A'X^{(k)} + B' - (A'X + B') = A'(X^{(k)} - X)$$

ce qui indique

$$\lim_{k\to\infty} \|X^{(k+1)} - X\| = 0 \Leftrightarrow \rho(A') < 1$$

On remarque que les coefficients de A' s'écrivent :

$$a'_{i,j} = \begin{cases} 0 & \text{si } i = j \\ \frac{-a_{ij}}{a_{ii}} & \text{si } i \neq j \end{cases}$$

Soit λ une valeur propre de A' et $Y = (y_i)$ un vecteur propre associé. $A'Y = \lambda Y$ donne donc pour tout i :

$$\begin{aligned} |\lambda y_i| &= |\sum_j a'_{ij} y_j| \\ |\lambda y_i| &= |\sum_{j\neq i} \frac{-a_{ij}}{a_{ii}} y_j| \\ |\lambda y_i| &= |\frac{1}{a_{ii}} y_i| \cdot |\sum_{j\neq i} -a_{ij}| \\ |\lambda y_i| &\leq |\frac{1}{a_{ii}} y_i| \sum_{j\neq i} |a_{ij}| \end{aligned}$$

Or, pour toute matrice $M = (m_{ij})$ à diagonale strictement dominante :

$$\forall i, |m_{ii}| > \sum_{j\neq i} |m_{ij}|$$

Ainsi, si A est diagonale strictement dominante, alors son rayon spectral est inférieur à 1, ce qui fait converger la méthode de Jacobi.

2 Processing

2.1 Parallélisation

3 Post-Processing