

## Méthodes itératives

L'idée des méthodes itératives est de construire une suite de vecteurs  $\mathbf{x}^{(k)}$  qui converge vers le vecteur  $\mathbf{x}$ , solution du système  $A\mathbf{x} = \mathbf{b}$ ,

$$\mathbf{x} = \lim_{k \rightarrow \infty} \mathbf{x}^{(k)}.$$

L'intérêt des méthodes itératives, comparées aux méthodes directes, est d'être simples à programmer et de nécessiter moins de place en mémoire. En revanche le temps de calcul est souvent plus long.

Une stratégie est de considérer la relation de récurrence linéaire

$$\mathbf{x}^{(k+1)} = B\mathbf{x}^{(k)} + \mathbf{g}, \quad (2)$$

où  $B$  est la *matrice d'itération* de la méthode itérative (dépendant de  $A$ ) et  $\mathbf{g}$  est un vecteur (dépendant de  $\mathbf{b}$ ), tels que

$$\mathbf{x} = B\mathbf{x} + \mathbf{g}.$$

Étant  $\mathbf{x} = A^{-1}\mathbf{b}$ , on obtient  $\mathbf{g} = (I - B)A^{-1}\mathbf{b}$ ; la méthode itérative (2) est donc complètement définie par la matrice  $B$ .

En définissant l'erreur au pas  $k$  comme

$$\mathbf{e}^{(k)} = \mathbf{x} - \mathbf{x}^{(k)},$$

on obtient la relation de récurrence

$$\mathbf{e}^{(k)} = B\mathbf{e}^{(k-1)}, \quad \text{et donc } \mathbf{e}^{(k)} = B^{(k)}\mathbf{e}^{(0)}, \quad k = 0, 1, \dots$$

On démontre que  $\lim_{k \rightarrow \infty} \mathbf{e}^{(k)} = 0$  pour tout  $\mathbf{e}^{(0)}$  (et donc pour tout  $\mathbf{x}^{(0)}$ ) si et seulement si  $\rho(B) < 1$ , où  $\rho(B)$  est le *rayon spectral* de la matrice  $B$ , défini comme

$$\rho(B) = \max |\lambda_i(B)|,$$

et  $\lambda_i(B)$  sont les valeurs propres de la matrice  $B$ .

Une technique générale pour construire des méthodes itératives est basée sur une décomposition (*splitting*) de la matrice  $A$  sous la forme  $A = P - N$ , où  $P$  et  $N$  sont des matrices à déterminer avec  $P$  non singulière. La matrice  $P$  est appelée *matrice de préconditionnement*. Plus précisément,  $\mathbf{x}^{(0)}$  étant donné, on peut calculer  $\mathbf{x}^{(k)}$ , pour  $k \geq 1$ , en résolvant le système

$$P\mathbf{x}^{(k+1)} = N\mathbf{x}^{(k)} + \mathbf{b}, \quad k \geq 0. \quad (3)$$

Clairement, la solution exacte  $\mathbf{x}$  satisfait  $P\mathbf{x} = N\mathbf{x} + \mathbf{b}$  et donc  $A\mathbf{x} = \mathbf{b}$ .

Le système (3) peut être écrit également sous la forme (2), avec  $B = P^{-1}N$ , et  $\mathbf{g} = P^{-1}\mathbf{b}$ .

Une relation de récurrence équivalente à (3) est la suivante :

$$P(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}) = \mathbf{r}^{(k)}, \quad k \geq 0, \quad (4)$$

où  $\mathbf{r}^{(k)} = \mathbf{b} - A\mathbf{x}^{(k)}$  est le *résidu* à l'itération  $k$ . On peut généraliser la relation précédente comme

$$P(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}) = \alpha_k \mathbf{r}^{(k)}, \quad k \geq 0, \quad (5)$$

où on a introduit un paramètre  $\alpha_k$  (qui peut être différent à chaque itération  $k$ ) afin d'accélérer la convergence. Cette méthode est dite de *Richardson*.

Les relations (3), (4) et (5) montrent qu'on doit résoudre un système linéaire de matrice  $P$  à chaque itération; donc  $P$  doit être telle que la résolution du système ait un coût raisonnable. Par exemple, on pourra choisir  $P$  diagonale ou triangulaire.

### La Méthode de Jacobi

On remarque que si les éléments diagonaux de  $A$  sont non nuls, le système linéaire  $A\mathbf{x} = \mathbf{b}$  est équivalent à :

$$x_i = \frac{1}{a_{ii}} \left( b_i - \sum_{j=1, j \neq i}^n a_{ij} x_j \right), \quad i = 1, \dots, n.$$

Pour une donnée initiale  $\mathbf{x}^{(0)}$  choisie, on calcule  $\mathbf{x}^{(k+1)}$  par

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left( b_i - \sum_{j=1, j \neq i}^n a_{ij} x_j^{(k)} \right), \quad i = 1, \dots, n.$$

Cela permet d'identifier le *splitting* suivant pour  $A$  :

$$\begin{aligned} A &= D - E - F \\ D &: \text{ diagonale de } A \\ E &: \text{ triangulaire inférieure avec des 0 sur la diagonale} \\ F &: \text{ triangulaire supérieure avec des 0 sur la diagonale} \end{aligned}$$

La matrice d'itération de la méthode de Jacobi est donnée par :

$$B_J = D^{-1}(E + F) = I - D^{-1}A.$$

L'algorithme de Jacobi nécessite le stockage des deux vecteurs  $x_j^{(k)}$  et  $x_j^{(k+1)}$ .