TP 3 – Systèmes linéaires

Théorie. Si la résolution de systèmes linéaires de petite taille (jusqu'à une dizaine de lignes) peut être effectuée indifféremment avec n'importe quelle méthode, le cas des grands systèmes (qui peuvent rapidement compter des centaines de milliers, voire des millions d'équations) requiert un traitement particulier. Les méthodes se divisent en deux groupes : les *méthodes directes* (Gauss, LU, QR) qui conduisent à la solution exacte au bout d'un nombre fini d'itérations (aux erreurs d'arrondi près), et les *méthodes itératives* pour lesquelles on construit une suite approchante de vecteurs, convergeant vers la solution recherchée (il faut alors se donner un critère d'arrêt).

Méthodes itératives. On décompose $A = M - N$ avec M "facile" à inverser, la suite (x_k) d'éléments de $\mathbb{R}^n$ approchant la solutions s'obtient par la récurrence :

$$\begin{cases} x_0 \in \mathbb{R}^n & \text{donné,} \\ Mx_{k+1} = Nx_k + b. \end{cases}$$

Les propriétés de convergence de l'algorithme sont liées aux propriétés spectrales de la matrice d'itération $M^{-1}N$. Écrivons $A = D - E - F$ où D est la partie diagonale de A : $D = (A_{ij}\delta_{ij})_{1\leq i,j\leq n}$; $-E$ la partie triangulaire inférieure stricte de A : $E_{ij} = -A_{ij}$ si $i > j$, 0 sinon ; et $-F$ la partie triangulaire supérieure stricte de A : $F_{ij} = -A_{ij}$ si $i < j$, 0 sinon.

Méthode de Jacobi : $M = D$ et $N = E + F$.

Méthode de Jacobi relaxée : $M = \frac{1}{\omega}D$ et $N = \frac{1-\omega}{\omega}D + E + F$.

Méthode de Gauss-Seidel : $M = D - E$ et $N = F$.

Méthode de Gauss-Seidel relaxée : $M = \frac{1}{\omega}D - E$ et $N = \frac{1-\omega}{\omega}D + F$ avec $\omega \in \mathbb{R}_*^+$.

Remarque. D'un point de vue algorithmique, chacune de ces méthodes repose sur la résolution itérative de systèmes linéaires diagonaux ou triangulaires inférieurs et ne nécessite à ce titre à aucun moment le calcul de la matrice $M^{-1}N$. Pour se concentrer sur les aspects mathématiques du problèmes, on pourra cependant librement utiliser l'opérateur \ de Scilab ou la commande `numpy.linalg.solve` de Python.

But du TP. *Il s'agit de mettre en œuvre certaines méthodes itératives pour la résolution d'un système $Ax = b$ et d'estimer leur vitesse de convergence. Afin de manipuler des matrices simples couramment rencontrées, on pourra tester les méthodes sur les deux matrices suivantes.*

La matrice des fonctions splines :

$$A_{\text{spline}} = \begin{pmatrix} 4 & 1 & 0 & 0 \\ 1 & 4 & \ddots & 0 \\ 0 & \ddots & \ddots & 1 \\ 0 & 0 & 1 & 4 \end{pmatrix}$$

La matrice du Laplacien en dimension 1 :

$$A_{\text{lapl}} = \begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & \ddots & 0 \\ 0 & \ddots & \ddots & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix}$$

Mise en œuvre.

- ☐ Programmer les méthodes de **Jacobi** et de **Jacobi relaxée**.
- ◇ Tester la convergence de ces deux méthodes sur les exemples proposés, en dimension $n = 15$, en retenant les données pour lesquelles la solution est le vecteur $x = {}^t(1, 1, \ldots, 1)$.

- ◇ Pour la méthode de Jacobi, estimer la vitesse de convergence [1] et la relier numériquement au rayon spectral de la matrice d'itération [2].
- ◇ Pour la méthode de Jacobi relaxée, tracer le graphe donnant la vitesse de convergence en fonction du paramètre de relaxation ω. Obtenir, par la théorie, la vitesse de convergence exacte et la superposer à la courbe précédente.
- ◇ Pour parfaire la concordance entre les courbes précédentes, on pourra résoudre un grand nombre de systèmes $Ax = b$ dont la solution x est un vecteur unitaire choisi aléatoirement, et retenir finalement la pire vitesse de convergence.
- ◇ Tracer, pour chacune des deux matrices envisagées, la vitesse de convergence de la méthode de Jacobi en fonction de la taille n de la matrice. L'obtenir analytiquement.
  On pourra dans un deuxième temps tracer le nombre d'itérations nécessaires pour obtenir une précision donnée, en fonction de n et comparer le résultat à l'analyse.

☐ *Ouverture 1.* Reprendre la mise en œuvre précédente pour les méthodes de **Gauss-Seidel** et de **Gauss-Seidel relaxée**.

☐ *Ouverture 2.* Une classe de méthodes plus élaborées, dites semi-itératives, consiste à choisir une suite de paramètres réels (α_k) et de remplacer la récurrence $x_{k+1} = M^{-1}Nx_k + M^{-1}b$ par

$$x_{k+1} = \alpha_k M^{-1}Nx_k + (1-\alpha_k)x_k + M^{-1}b.$$

Lorsque $\alpha_k = 1$ pour tout k, on retrouve la méthode itérative de base et l'erreur est donnée par $e_k = (M^{-1}N)^k e_0$. Le choix de ces nouveaux paramètres est guidé par la possibilité de modifier le polynôme en $M^{-1}N$ qui intervient (ici $P_k(X) = X^k$) de sorte à minimiser la norme de l'erreur (avec plus généralement $e_k = P_k(M^{-1}N)e_0$). Une heuristique naturelle consiste alors à utiliser le polynôme de Chebyshev de degré donné, et dont la norme uniforme est minimale sur $[-1,1]$.
Supposons α et β deux réels vérifiant : $-1 < \alpha < \beta < 1$ choisis de sorte que le spectre de $M^{-1}N$, supposé réel, soit inclus dans le segment $[\alpha,\beta]$. En s'appuyant sur la relation de récurrence à trois termes vérifiée par les polynômes de Chebyshev, on définit [3] sur ce principe la méthode de **Jacobi-Chebyshev** :

$$\begin{aligned} y_0 &= x_0, \\ y_1 &= y_0 - \frac{2}{2-\alpha-\beta}D^{-1}(Ay_0 - b) \\ y_{k+1} &= \sigma_{k+1}\left(y_k - \frac{2}{2-\alpha-\beta}D^{-1}(Ay_k - b)\right) + (1-\sigma_{k+1})y_{k-1}, \quad k \geq 2, \end{aligned}$$

où la suite (σ_k) est définie par

$$\begin{aligned} \sigma_1 &= 2 \\ \sigma_{k+1} &= \frac{4}{4 - \left(\frac{\kappa-1}{\kappa+1}\right)^2 \sigma_k}, \quad k \geq 1, \quad \kappa = \frac{1-\alpha}{1-\beta}. \end{aligned}$$

◇ Tester cette méthode. Convaincu ?

1. quantifiée par la plus petite constante $\rho > 0$ telle que pour tout k, $\|x_k - x\| \leq C(x_0)\rho^k$.
2. On rappelle (et on pourra redémontrer) que le spectre de A_{lapl} est l'ensemble $\left\{4\sin^2\left(\frac{p\pi}{2(n+1)}\right), 1 \leq p \leq n\right\}$.
3. Source Hackbusch, *Iterative solution of large sparse systems of equations*, AMS 95 (2016) Springer.