Faculté : MI
Département : Maths
Année universitaire : 2022/2023

جامعة محمد بوضياف - المسيلة
Université Mohamed Boudiaf - M'sila

1^{re} année Master(AMN)
Semestre : 1
Matière : Ana. Matri. Opti. 1

TP3 : Méthode du gradient conjugué

Problème de Poisson

L'objectif de ce TP est d'utiliser la méthode du gradient conjugué pour calculer la solution approchée du problème de Poisson suivant :

$$\begin{cases} -u''(x) = f(x), \ 0 < x < 1, \\ u(0) = \alpha, \ u(1) = \beta. \end{cases} \tag{1}$$

On se donne une subdivision de $[0,1]$ en N intervalles de longueur $h = \frac{1}{N+1}$, et on considère les points $u_i = ih$, $i = 0, 1, \ldots, N+1$. On cherche alors une solution approchée sous la forme du vecteur $u_N = (u_i)$, $i = 1, \ldots, N$, où les u_i vérifient le schéma aux différences finies :

$$\begin{cases} \frac{-u_{i-1}+2u_i-u_{i+1}}{h^2} = f(x_i), \ i = 1, \ldots, N, \\ u_0 = \alpha, \ u_{N+1} = \beta. \end{cases} \tag{2}$$

1. Montrer que le problème (2) peut s'écrire sous la forme :

$$A_N u_N = F_N, \tag{3}$$

où A_N et F_N sont donnés par :

$$A_N = \frac{1}{h^2} \begin{pmatrix} 2 & -1 & 0 & \ldots 0 & \\ -1 & 2 & -1 & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & -1 & 2 & -1 \\ 0 & \ldots & 0 & -1 & 2 \end{pmatrix} \text{ et } F_N = \begin{pmatrix} f(x_1) + \alpha \\ \vdots \\ f(x_N) + \beta \end{pmatrix}.$$

2. Montrer que : U_N est solution de (3) si et seulement si U_N minimise sur $\mathbb{R}^N$ la fonction J_N définie par :

$$J_N(u) = \frac{1}{2} \langle A_N u, u \rangle - \langle F_N, u \rangle .$$

3. Dans toute la suite du TP, on prendra $\alpha = \beta = 0$ et $f(x) = 16\pi^2 \sin(4\pi x)$. Montrer que la solution exacte du problème (1) donnée par :

$$u(x) = \sin(4\pi x). \tag{4}$$

4. Calculer ∇J_N en fonction de A_N et F_N.

Méthode du gradient conjugué

On rappelle l'algorithme du gradient conjugué pour la fonction quadratique J_N, un point de départ u^0 et un test d'arrêt ε préalablement définis, est donné par :

Algorithme du gradient conjugué :

Initialiser $g_0 = \nabla J_N(u^0)$ et $d_0 = -\nabla J_N(u^0)$ et le compteur $k = 1$.

Tant que le résidu $r^k = \left\|\nabla J_N\left(u^k\right)\right\|_\infty$ est plus grand que ε et que le compteur n'est pas trop grand :

- Calculer $\rho_k = -\frac{g_k^T d_k}{d_k^T A_N d_k}$,
- Calculer $u^{k+1} = u^k + \rho_k d_k$,
- calculer $g_{k+1} = \nabla J_N\left(u^{k+1}\right)$,
- Calculer $\beta_k = \frac{g_{k+1}^T A_N d_k}{d_k^T A_N d_k}$,
- calculer la nouvelle descente $d_{k+1} = -g_{k+1} + \beta_k d_k$,
- Calculer le résidu $r^{k+1} = \|g_{k+1}\|_\infty$,
- incrémenter le compteur.

Créer un script **`scriptTP3_conjugue.m`** pour répondre aux questions suivantes :

1. Vérifier numériquement, pour quelques valeurs de N, que la matrice A_N est bien définie positive. **Fonction Matlab utile :** `eig`.
2. On se place dans le cas $N = 2$.
   (a) Calculer J_2 et ∇J_2. Tracer sur une même figure les courbes de niveaux de J_2 ainsi que le champ de vecteurs ∇J_2 sur le pavé $[-10, 10]^2$.
   Fonctions Matlab utiles : `contour`, `quiver`.
   (b) Calculer les itérations $u^k = \left(u_1^k, u_2^k\right)$ données par l'algorithme du gradient conjugué, et tracer sur la même figure que précédemment la ligne qui relie les u^k. On prendra $u^0 = (8, 4)$ et $\varepsilon = 10^{-12}$.
   Fonctions Matlab utiles : `hold on`, `plot`, `hold off`
3. On se place de nouveau dans le cas général.
   (a) Écrire une fonction **`gradient_conjugue.m`** qui prend en argument A_N, F_N, un point de départ u^0 et un test d'arrêt ε, et qui renvoie le vecteur u_N qui minimise J_N ainsi que le nombre d'itérations effectuées et le temps de calcul.
   (b) Tester cette fonction pour $\varepsilon = 10^{-12}$ et pour $N = 2, 5, 20, 50$. Afficher à l'aide de la fonction **`fprintf`** le nombre d'itérations ainsi que le temps de calcul pour chaque N. Tracer sur une même figure les solutions approchées u_N, ainsi que la solution exacte de (1).