Faculté des Sciences et Techniques — Université de Limoges

Analyse Numérique 1 – Travaux Pratiques — L3 – Semestre 5 – 2019/2020

Feuille de TP n°1

Méthodes indirectes

Le travail en binôme est possible. On utilise de préference Scilab, sinon n'importe quel logiciel/language capable de résoudre les exercices de la feuille. Le compte rendu est à rendre[1] deux semaines après le TP, de préference en utilisant Community, sinon à envoyer par mail à simone.naldi@unilim.fr

Si l'ordinateur crash pendant un calcul, c'est peut être à cause de la taille des entrées ; si c'est le cas, diminuez la taille. N'hesitez pas à discuter de tels problèmes (et des solutions que vous avez apportées) dans le compte-rendu.

Généralités

Soit $A = (a_{ij}) \in \mathbb{R}^{n\times n}$ une matrice inversible. Nous souhaitons resoudre un système linéaire $Ax = b$, avec $b \in \mathbb{R}^n$ donné. On introduit une décomposition régulière de A, c'est-à-dire une couple de matrices (M, N) avec M inversible tel que $A = M - N$. La méthode itérative basée sur la décomposition (M, N) est définie par :

x^0 est donné dans $\mathbb{R}^n$,
x^{k+1} est solution du système $Mx^{k+1} = Nx^k + b$, pour tout $k \geq 0$.

Une condition nécessaire et suffisante de convergence de cette méthode (pour toute donnée initiale x^0) est que le rayon spectral de la matrice d'itération $B = M^{-1}N$ vérifie $\rho(B) < 1$. Dans ce cas, la suite (x^k) converge vers x, solution du système $Ax = b$.

Préliminaires

1) Pour ϵ et $n \geq 3$ fixés, soit A_ϵ la matrice symétrique, de taille $n \times n$, et pentadiagonale de motif $[\epsilon^2, \epsilon, 1, \epsilon, \epsilon^2]$. Dans un script `main.sce`, poser $\epsilon = 0.2$, $n = 10$, puis définir cette matrice :

```
Aeps = eye(n,n) + diag(eps*ones(n-1,1),1) + diag(eps^2*ones(n-2,1),2)
    + diag(eps*ones(n-1,1),-1) + diag(eps^2*ones(n-2,1),-2);
```

2) Définir les vecteurs $\hat{x} = (1, 1, \ldots, 1) \in \mathbb{R}^n$ et $b_\epsilon = A_\epsilon \hat{x}$ dans `main.sce` :

```
xhat = ones(n,1);
beps = Aeps * xhat;
```

3) Étude de la matrice A_ϵ : donner un intervalle de valeurs de ϵ pour lesquelles A_ϵ est à diagonale strictement dominante, et calculer numériquement le conditionnement de A_ϵ

```
conditionnement = cond(Aeps);
```

Méthode de Jacobi

En supposant que les termes diagonaux de A sont non nuls, on peut fixer $M = D$ et $N = D - A$, avec D la matrice diagonale contenant la diagonale de A. On obtient ainsi la méthode de Jacobi. Composante par composante, la relation de récurrence s'écrit :

$$x_i^{k+1} = \frac{1}{a_{ii}} \left(b_i - \sum_{j\neq i} a_{ij} x_j^k \right), 1 \leq i \leq n, 0 \leq k.$$

4) *Question théorique à faire à la maison.* Montrer que si la matrice A est à diagonale strictement dominante alors la méthode de Jacobi est convergente.
5) Code et premier test.

[1]Conseil : rendez un compte-rendu original, avec des commentaires qui aideront l'enseignant à comprendre que vous avez bien compris. Bref il ne s'agit pas seulement de donner une description passive de la feuille de TP.

(a) Implémenter la méthode de Jacobi dans un fichier `jacobi.sce`. La fonction prend en arguments: la matrice `A`, le second membre `b`, le vecteur initial `x0`, une borne supérieure `nmax` sur le nombre d'itérations, ainsi qu'une tolérance `tol`: on décide que l'algorithme a convergé lorsque le résidu relatif est inférieur à cette tolérance. En sortie, la fonction renvoie l'estimation de la solution calculée `x`, ainsi que le nombre d'itérations `iter`.

```
function [x,iter] = jacobi(A,b,x0,nmax,tol)
// appel :  [x,iter]=jacobi(A,b,x0,nmax,tol)
// en entree : A matrice nxn; b,x0 vecteurs nx1; nmax entier 1x1; tol>0 1x1
// en sortie : x (nx1), solution de Ax=b, calculee en iter (1x1) iterations
//             par la methode de Jacobi
// x0 : etat initial
// nmax : borne superieure sur le nombre d'iterations
// tol : l'algorithme a converge quand norme(residu)/norme(b)<=tol
n = size(A,1);
iter = 0;
r = b-A*x0;normb=norm(b);err=norm(r)/normb;x=x0;
while err > tol & iter < nmax
   iter = iter+1;
   xnew = zeros(n,1);
   for i = 1:n
      xnew(i) = (b(i)-A(i,[1:i-1 i+1:n])*x([1:i-1 i+1:n]))/A(i,i);
   end
   x = xnew; r = b-A*x; err = norm(r)/normb;
end
endfunction
```

Expliquer chaque ligne du code.

(b) La méthode de Jacobi est-elle convergente pour $\epsilon = 0.2$ et $n = 10$? Résoudre le système $A_\epsilon x = b_\epsilon$ par cette méthode. Fixer pour état initial $x_0 = 0$, et fixer une tolérance de 10^{-8} :

```
nmax = 1000;
tol = 10^(-8);
x0 = zeros(n,1);
[x,iter] = jacobi(Aeps,beps,x0,nmax,tol);
```

En combien d'itérations la méthode converge-t-elle ?

6) On fixe maintenant $n = 100$ (si l'ordi crash, diminuer) et $\epsilon = 0.3$. Calculer la nouvelle matrice A_ϵ, poser $\hat{x} = (1, 1, \ldots, 1) \in \mathbb{R}^n$ et $b_\epsilon = A_\epsilon \hat{x}$.

(a) Calculer le conditionnement de la matrice A_ϵ et le rayon spectral $\rho(B)$ de la matrice d'itération B associée à la matrice A_ϵ.

```
conditionnement = cond(Aeps);
M = diag(diag(Aeps));
N = M-Aeps;
B = inv(M)*N;
rho = max(abs(spec(B)));
```

(b) Tracer $\rho(B)$ en fonction de $\epsilon \in [0, 1]$, pour n fixé (par exemple $n = 20$). Conclure graphiquement quant à la convergence de la méthode de Jacobi. Tester la méthode pour $\epsilon = 0.5$. Commenter le résultat.

7) Erreur résiduelle relative au cours des itérations :
(a) Compléter la fonction `jacobi` pour qu'elle renvoie en troisième sortie un vecteur `err` contenant les erreurs résiduelles relatives. (Copier au préalable le fichier `jacobi.sce` en `jacobi0.sce` pour conserver la première version du travail).
(b) Appliquer la méthode de Jacobi, avec une tolérance de 10^{-12}. Tracer en échelle logarithmique les erreurs résiduelles relatives `err` (`plot2d("nl",err)`).

Méthode de Gauss-Seidel

En supposant que les termes diagonaux de A sont non nuls, on peut fixer $M = T$ et $N = T - A$, avec T matrice triangulaire inférieure contenant la partie triangulaire inférieure de A. On obtient ainsi la méthode de Gauss-Seidel. Composante par composante, la relation de récurrence s'écrit :

$$x_i^{k+1} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{k+1} - \sum_{j=i+1}^{n} a_{ij} x_j^{k} \right), 1 \leq i \leq n, 0 \leq k.$$

On peut démontrer que si la matrice A est à diagonale strictement dominante, cette méthode est convergente.
8) Écrire une fonction `gaussSeidel` qui résout un système par la méthode de Gauss-Seidel. Tester cette fonction sur un des exemples précédents.
9) Pour $n = 100$, tracer le rayon spectral $\rho(B)$ de la matrice d'itération (associée à A_ϵ) en fonction de $\epsilon \in [0, 1]$. Conclure graphiquement quant à la convergence de la méthode de Gauss-Seidel. Comparer avec le rayon spectral et les conclusions de la méthode de Jacobi.