Méthodes itératives pour la résolution de systèmes linéaires

Grégory Vial

9 mars 2009

On s'intéresse ici à la résolution du système linéaire $Ax = b$, où la matrice A, de taille $n \times n$, est donnée par

$$A = \begin{bmatrix} 4 & 1 & & & \\ 1 & 4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 4 & 1 \\ & & & 1 & 4 \end{bmatrix}.$$

Cette matrice intervient dans la détermination d'une spline interpolatrice (voir [3]).

L'objet de ces quelques pages est la mise en œuvre des méthodes de Jacobi et de relaxation, qu'on rappelle ici : D désigne la matrice diagonale $D = 4I_n$ et $-E$ la matrice strictement triangulaire inférieure composée de la partie inférieure de A ; on note $J = \frac{1}{\omega}D - E$; $\omega \in (0, 1)$. Pour $x_0 \in \mathbb{R}^n$ donné, on définit la suite (x^n) par récurrence comme suit :

Méthode de Jacobi : $\quad x^{n+1} = (I_n - D^{-1}A)x^n + D^{-1}b$;

Méthode de relaxation : $\quad x^{n+1} = (I_n - J^{-1}A)x^n + J^{-1}b$.

Ces deux méthodes convergent pour le système considéré, dont la matrice est à diagonale strictement dominante (voir [1] ou [2]).

Les fonctions matlab suivantes définissent ces méthodes.

```
function xx=jacobi(A,b,x0,eps,maxit)
err=1;
x=x0;xt=x;xx=[];iter=0;
D=diag(diag(A));
while err>eps & iter<maxit
  xt=x-D\A*x+D\b;
  err=norm(x-xt);
  x=xt;
  xx=[xx,x];
  iter=iter+1;
end
```

```
function xx=relax(A,b,x0,w,eps,maxit)
err=1;
x=x0;xt=x;xx=[];iter=0;
J=1/w*diag(diag(A))+tril(A,-1);
while err>eps & iter<maxit
  xt=x-J\A*x+J\b;
  err=norm(x-xt);
  x=xt;
  xx=[xx,x];
  iter=iter+1;
end
```

Noter l'utilisation de la commande \ pour résoudre le système linéaire à chaque étape (on n'inverse pas explicitement les matrices même si, dans le cas diagonal, ce ne serait pas plus coûteux).

Le script suivant permet la construction des graphes des figures 1 et 2.

```
n=5;eps=1e-12;maxiter=100;
A=4*eye(n)+diag(ones(1,n-1),1)
         +diag(ones(1,n-1),-1);
xex=ones(n,1);
b=A*xex;
x0=zeros(n,1);
x=jacobi(A,b,x0,eps,maxiter);
[p,q]=size(x);
err=[];
for i=1:q
  err=[err,norm(xex-x(:,i))];
end
figure(1)
plot(1:q,log(err))
a=polyfit(1:q,log(err),1);
exp(a(1))
D=diag(diag(A));
max(abs(eig(eye(n)-D\A)))
grid on
title('Convergence de la
        méthode de Jacobi')
```

```
xlabel('Itérés')
ylabel('log(erreur)')
figure(2)
h=.01;
omega=h:h:1;
ray=[];raye=[];
for ome=omega
  x=relax(A,b,x0,ome,eps,maxiter);
  [p,q]=size(x);
  err=[];
  for i=1:q
    err=[err,norm(xex-x(:,i))];
  end
  a=polyfit(1:q,log(err),1);
  ray=[ray,a(1)];
end
plot(omega,exp(ray))
title('Taux de convergence en
           fonction de omega')
xlabel('omega')
ylabel('Taux de convergence')
```

La figure 1 représente l'évolution du logarithme de l'erreur au cours des itérations de la méthode de Jacobi. La théorie fournit l'estimation suivante :

$$\|x^k - x\| \leq c\left[\rho(D^{-1}A)\right]^k.$$

Il est donc satisfaisant d'obtenir une droite de pente $\log\left[\rho(D^{-1}A)\right]$. Il est même possible d'utiliser la méthode de Jacobi pour estimer le rayon de convergence de la matrice $D^{-1}A$: l'exponentielle de la pente moyenne de la droite de la figure 1 est 0.43300948301476 et le rayon spectral $\rho(D^{-1}A)$ vaut 0.43301270189222.

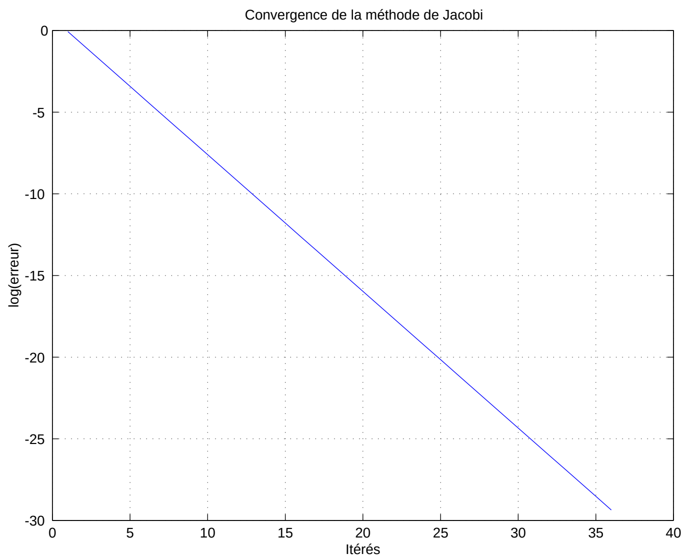

Figure 1 – Convergence de la méthode de Jacobi.

La seconde partie du script est dédiée à la mise en évidence d'un paramètre optimal. On a tracé sur la figure 2 le taux de convergence de la méthode de relaxation en fonction de ω.

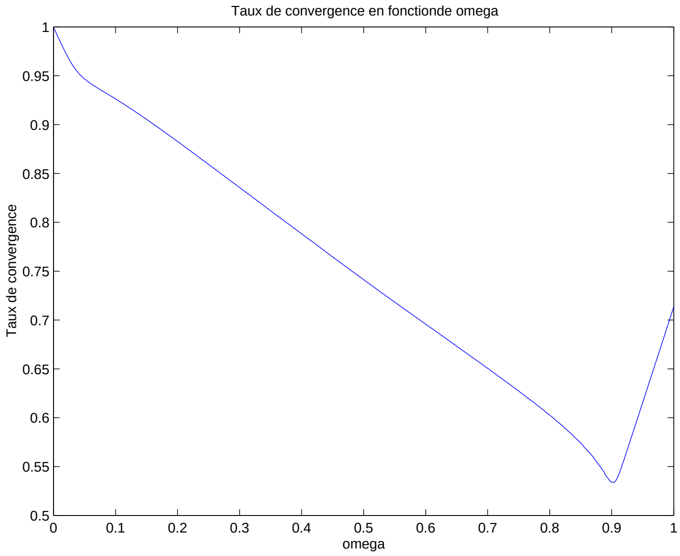

FIGURE 2 – Taux de convergence en fonction du paramètre de relaxation ω.

On retrouve la courbe bien connue, étudiée dans [1].

Références

[1] P. G. CIARLET. *Introduction à l'analyse numérique matricielle et à l'optimisation.* Collection Mathématiques Appliquées pour la Maîtrise. [Collection of Applied Mathematics for the Master's Degree]. Masson, Paris 1982.

[2] M. SCHATZMAN. *Analyse numérique.* InterEditions, Paris 1991. Cours et exercices pour la licence.

[3] J. STOER, R. BULIRSCH. *Introduction to numerical analysis*, volume 12 of *Texts in Applied Mathematics*. Springer-Verlag, New York, third edition 2002. Translated from the German by R. Bartels, W. Gautschi and C. Witzgall.