

# RÉSOLUTION DE SYSTÈMES LINÉAIRES

semaine du 2 avril 2012

## 1 Contexte et problématique

On s'intéresse à la résolution d'un SEL

$$\mathbf{Ax} = \mathbf{b}; \mathbf{A} \in \mathbb{R}^{n \times n}, \mathbf{x}, \mathbf{b} \in \mathbb{R}^n$$

du point de vue numérique. On notera

- $\xi \in \mathbb{R}^n$  la solution exacte ;
- $\mathbf{x}(k) \in \mathbb{R}^n$  la solution de la  $k$ -ième itération ;
- $\mathbf{e}(k) = \xi - \mathbf{x}(k)$  l'erreur lors de la  $k$ -ième itération, et
- $\mathbf{r}(k) = \mathbf{b} - \mathbf{Ax}(k)$  l'erreur résiduelle lors de la  $k$ -ième itération.

On étudiera la qualité de la solution sous quatre aspects :

1. Influence de la valeur du conditionnement de la matrice.
2. Influence de la dimension de la matrice.
3. Influence de la méthode de la résolution utilisée (directe ou itérative).
4. Influence de la relaxation.

De plus, on vérifiera des relations entre l'erreur et l'erreur résiduelle, vues en TD.

Nous donnons ci-après une présentation détaillée pour chacun de ces points, ainsi que du livrable.

## 2 Matrice à conditionnement donné

On définit le conditionnement d'une matrice  $\mathbf{A} \in \mathbb{R}^{n \times n}$  relativement à une norme matricielle  $|\cdot|_\alpha$  par

$$\kappa_\alpha(\mathbf{A}) = |\mathbf{A}|_\alpha |\mathbf{A}^{-1}|_\alpha$$

Dans la suite on prendra  $\alpha = 2$ .

Soit le SEL

$$\mathbf{Ax} = \mathbf{b}$$

Pour étudier l'influence du conditionnement à la résolution d'un SEL, on effectuera une étude en prenant  $\kappa = 10, 10^3$  et  $10^6$  respectivement. Pour ce faire, on engendrera des matrices carrées régulières avec la valeur du conditionnement fixée d'avance. La méthode pour créer de telles matrices est la suivante :

1. On crée la matrice diagonale

$$\Delta = \text{diag} \left( 1, \kappa^{-\frac{1}{n-1}}, \kappa^{-\frac{1}{n-2}}, \dots, \kappa^{-1} \right)$$

où  $n$  est la dimension de la matrice et  $\kappa$  le conditionnement défini.

On peut vérifier que  $\kappa_2(\Delta) = \kappa$ .

2. On construit deux matrices carrées orthogonales  $U$  et  $V$  de même dimension que la matrice  $\Delta \in \mathbb{R}^{n \times n}$ .
3. On calcule la matrice  $A$  selon la formule

$$A = U \cdot \Delta \cdot V^T$$

Nous avons

$$\kappa_2(A) = \|A\|_2 \|A^{-1}\|_2 = \|U\Delta V^T\|_2 \|V\Delta^{-1}U^T\|_2 = \|\Delta\|_2 \|\Delta^{-1}\|_2 = \kappa_2(\Delta) = \kappa$$

Les détails pratiques pour la création de la matrice  $A$  en Scilab sont donnés à l'annexe A.

### 3 Dimension de la matrice

La dimension de la matrice a une influence sur la qualité des résultats dans la mesure où le nombre d'opérations est différent pour des matrices de dimension différente.

On se propose d'étudier cette influence, en utilisant des matrices, construites comme précédemment, avec des dimensions  $n \times n$  différentes.

On prendra  $n = 10, 50, 100$  et  $150$ .

Pour chacune de ces matrices on effectuera les calculs avec les trois valeurs différentes de conditionnement.

### 4 Méthodes de résolution de système

Il est possible que la méthode utilisée pour la résolution d'un SEL influe sur les résultats de cette résolution. Pour étudier cette question, on utilisera deux méthodes :

- Une méthode de résolution directe, faisant partie de la bibliothèque de Scilab.

Il s'agit de la fonction `linsolve` qu'on pourra utiliser comme suit :

$$x = \text{linsolve}(A, b);$$

où  $A$  est la matrice carrée des coefficients du système,  $b$  est le vecteur du second membre et  $x$  est le vecteur solution. Pour plus de détails, se reporter à l'aide de Scilab.

- Une méthode itérative, à savoir la méthode Jacobi, dont le schéma, que vous devez programmer, est rappelé ici :

$$\begin{aligned} x(0) &\in \mathbb{R}^n \\ x(k+1) &= -D^{-1}(L+U)x(k) + D^{-1}b; k \geq 0 \end{aligned}$$

Les calculs doivent se faire pour les trois valeurs différentes de conditionnement et les quatre valeurs différentes de la dimension de la matrice.

Les détails pour la mise en œuvre de ces méthodes sont donnés à l'annexe B.

## 5 Évaluation du nombre d'itérations

On sait que si on veut diminuer l'erreur  $e(k)$  d'un facteur  $H$  par rapport à l'erreur  $e(0)$ , il faut faire  $m$  itérations, où la valeur de  $m$  est donnée par

$$m \geq \frac{H}{\log \rho(\mathbf{R}_J)}$$

où  $\rho(\mathbf{R}_J)$  le rayon spectral de la matrice d'itération de Jacobi

$$\mathbf{R}_J = -\mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})$$

Calculer pour les quatre matrices de taille différente ci-dessus et les trois valeurs de conditionnement, le nombre d'itérations pour avoir chaque fois une erreur de l'ordre de  $10^{-8}$ .

Les détails pour la mise en œuvre de cette méthode sont donnés à l'annexe C.

## 6 Influence de la relaxation

Pour une valeur de  $\kappa = 10^4$  et une matrice de dimension  $n \times n$ , avec  $n = 150$  on utilisera la méthode de Jacobi relaxée, avec valeurs  $\omega = 0.5, 0.8, 1.2, 1.5, 1.8$ . Le schéma de Jacobi relaxé est

$$\begin{aligned} \mathbf{x}(0) &\in \mathbb{R}^n \\ \mathbf{x}(k+1) &= [(1-\omega)\mathbf{I} - \omega\mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})] \mathbf{x}(k) + \omega\mathbf{D}^{-1}\mathbf{b}; k \geq 0, \omega \in ]0, 2[ \end{aligned}$$

Vérifier aussi si la valeur préconisée dans la littérature

$$\omega = \frac{2}{2 - \lambda_1 - \lambda_n}$$

où  $\lambda_1 \geq \dots \geq \lambda_n$  sont les valeurs propres, supposées réelles et avec  $\lambda_1 \neq -\lambda_n$ , de la matrice d'itération de Jacobi relaxé

$$\mathbf{R}_{J\omega} = (1-\omega)\mathbf{I} - \omega\mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})$$

est optimale.

Les détails pour la mise en œuvre de cette méthode sont donnés à l'annexe D.

## 7 Livrable

Un rapport écrit impérativement en  $\LaTeX$ , dans lequel on présente :

- les expériences effectuées avec leurs résultats ;
- les conclusions à tirer de l'étude effectuée en ce qui concerne le conditionnement de la matrice, sa dimension, la méthode utilisée

Le livrable doit contenir

- le rapport en format pdf et le source en  $\LaTeX$ , et
- les programmes Scilab associés à une notice d'utilisation de ces programmes.

Ce livrable sera stocké dans un fichier compacté (format zip ou rar). Pour l'envoi vous utiliserez à Cergy le site `sifoci`, rubrique Analyse numérique et vous cliquerez sur le bouton correspondant à votre groupe de TD.

Date de livraison : 12.04.2012

## 8 Recommandations

Sur la forme, il est vivement conseillé d'utiliser des tableaux et des graphiques, afin d'améliorer la lisibilité des résultats.

Le rapport doit être écrit selon les règles de l'art, dont un bref rappel est donné par le document "Écriture d'un rapport en Analyse Numérique" qui se trouve et sur le site [http //sifoci eisti fr](http://sifoci.eisti.fr), rubrique Analyse Numérique.

# APPENDICES

## A Construction des matrices avec Scilab

Pour faire en Scilab, les programmes nécessaires pour construire la matrice  $A$  il faut suivre la démarche suivante :

1. Construire la matrice diagonale

$$\Delta = \text{diag} \left( 1, \kappa^{-\frac{1}{n-1}}, \kappa^{-\frac{1}{n-2}}, \dots, \kappa^{-1} \right)$$

pour  $\kappa = 10, 1000, 10^8$  et  $10^{16}$  et  $n = 10, 50, 100$  et  $150$ .

2. Pour chacune des matrices  $\Delta$ , construites précédemment, calculer la matrice  $A = U\Delta V^T$ .

(a) Pour l'élaboration des matrices orthogonales  $U$  et  $V$ , on procède comme suit :

(b) On construit la matrice tridiagonale

$$\mathbf{H} = \begin{bmatrix} 2 & -1 & 1 & 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots \\ -1 & 2 & -1 & 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & -1 & 2 & -1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & -1 & 2 & -1 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & -1 & 2 & -1 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & 0 & -1 & 2 \end{bmatrix}$$

de dimension  $(n \times n)$ .

(c) On utilise la fonction de Scilab `svd` pour faire une décomposition en valeurs singulières de cette matrice, en utilisant l'appel

i.  $[U, S, V] = \text{svd}(\mathbf{H})$

Les matrices  $U$  et  $V$  sont orthogonales.

(d) On construit la matrice  $A = U\Delta V^T$ . On peut vérifier que le conditionnement de  $A$  a la valeur requise. On peut aussi vérifier que la méthode de Jacobi converge.

## B Calculs à effectuer avec Scilab

Nous donnons ici à la démarche à suivre pour obtenir la solution d'un SEL avec les deux méthodes : librairie de Scilab et Jacobi.

1. Effectuer, pour les mêmes matrices que précédemment, la résolution du système linéaire. La démarche à suivre est la suivante :

(a) Construire le second membre  $\mathbf{b} = [1 \ 0 \ \dots \ 0 \ 1]^T$  de dimension  $(n \times 1)$  ce qui donne comme solution exacte  $\xi = [1 \ 1 \ 1 \ \dots \ 1]^T$ .

(b) Utiliser la fonction de Scilab `linsolve` pour calculer la solution numérique  $\mathbf{x}$  du système linéaire  $\mathbf{Ax} = \mathbf{b}$  :

$$\mathbf{x} = \text{linsolve}(\mathbf{A}, \mathbf{b})$$

(c) Évaluer l'erreur de la solution obtenue, en utilisant la norme et le fait qu'on connaît la solution exacte  $\xi$ .

(d) Programmer la méthode itérative de Jacobi, dont nous donnons le schéma :

$$\begin{aligned} \mathbf{x}(k+1) &= -\mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})\mathbf{x}(k) + \mathbf{D}^{-1}\mathbf{b}; k = 1, 2, \dots; \text{ où } k = n^\circ \text{ d'itération} \\ \mathbf{x}(0) &= [0 \ 0 \ 0 \ \dots \ 0]^T \end{aligned}$$

avec

- $\mathbf{D}$  la diagonale de  $\mathbf{A}$  ;
- $\mathbf{L}$  la partie de  $\mathbf{A}$  strictement inférieure à la diagonale ;
- $\mathbf{U}$  la partie de  $\mathbf{A}$  strictement supérieure à la diagonale.

Nous avons ainsi  $\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{U}$ .

(e) Contrairement à la méthode directe, l'erreur résiduelle de la solution obtenue sera fixée par vous-même. Le critère d'arrêt doit être l'erreur résiduelle relative, à savoir la valeur

$$r(k) = \frac{\|\mathbf{Ax}(k) - \mathbf{b}\|_2}{\|\mathbf{b}\|_2}$$

avec  $k$  numéro d'itération.

Vous fixerez vous-mêmes le seuil du critère. Habituellement on prend comme seuil une valeur  $\leq 10^{-8}$ .

(f) Le calcul itératif doit s'arrêter

- soit lorsque la valeur voulue de l'erreur résiduelle est atteinte ;
- soit lorsque le nombre d'itérations, fixé d'avance, est dépassé. Il est évident que, dans ce cas, l'erreur résiduelle ne sera pas inférieure au seuil fixé.

(g) Vous devez fixer un nombre d'itérations assez grand, par exemple 500 000.

(h) À la fin des calculs, le programme doit afficher l'erreur résiduelle de la solution obtenue et comparer avec celle de la méthode directe. Au vu de ces résultats et si l'arrêt des itérations se fait sur le critère du nombre d'itérations, l'utilisateur doit avoir la possibilité de poursuivre les calculs, à partir du point d'arrêt, afin d'obtenir une meilleure erreur résiduelle (plus proche du seuil, voire inférieure au seuil).

N.B. Si, pendant les itérations, le critère commence à augmenter, il faut arrêter les calculs en indiquant qu'il y a eu divergence.

Pour les calculs et les différentes comparaisons, nous vous rappelons que Scilab fonctionne avec la norme IEEE-754 en double précision ce qui signifie qu'il a une mantisse de 52 bits et un exposant de 11 bits. Ainsi un nombre réel  $x$  en décimal sera compris par l'ordinateur comme le nombre binaire  $(-1)^S \times (1 + Y) \times 10^{E-B}$ , où  $S$  est le signe,  $Y$  est la mantisse,  $E$  est l'exposant et  $B$  le biais (ici égal à 1023). Nous vous rappelons aussi que nous avons  $1 + Y$  pour la mantisse à cause du bit caché.

## C Erreur et nombre d'itérations

On voudrait avoir  $|e(k)| \leq 10^{-8}$ . On sait que pour avoir  $\frac{|e(k)|}{|e(0)|} \leq 10^{-H}$  il faut  $m \geq \frac{H}{\log \rho(\mathbf{R}_J)}$  itérations, avec  $\rho(\mathbf{R}_J)$  le rayon spectral de la matrice  $\mathbf{R}_J$ . Du fait qu'on est en mesure d'évaluer  $|e(0)|$ , on peut calculer  $m$  pour avoir  $|e(k)| \leq 10^{-8}$ .

Rappelons que le rayon spectral d'une matrice  $\mathbf{X}$  est donnée par

$$\rho(\mathbf{X}) = \max \{ |\lambda_i| / \lambda_i \text{ valeur propre de } \mathbf{X} \}$$

Pour son calcul on utilise la fonction `spec` de Scilab. En effet la commande

```
lambda = spec X;
```

range dans `lambda` les valeurs propres de  $\mathbf{X}$ , selon l'ordre croissant algébrique.

## D Relaxation

Répéter la même démarche que ci-dessus en utilisant le coefficient de relaxation. Comparer avec les résultats sans relaxation.