

© J.-B. A. K. <jean-baptiste.apoung@math.u-psud.fr>

Examen seconde session : 24 juin 2015
(Durée 3h00)

CONSIGNES :

- Les notes de cours et les scripts de TP sont autorisés. La consultation des pages Internet, des téléphones portables, de votre messagerie électronique, est interdite. Le non respect de cette consigne entraînera l’annulation de votre note.
- Commencez par créer un répertoire **M325_examen** Vous travaillerez dans ce répertoire. Il devra contenir tous les fichiers dont vous aurez besoin.
- Lorsqu’une question numérique nécessite des commentaires rédigés, mettez les dans vos fichiers sous forme de commentaires en langage C. **Les réponses aux questions théoriques devront être fournies sur les copies d’examen. Ces copies devront être anonymes et cachetées.**
- A la fin de l’examen, vous devez taper en ligne de commande dans votre répertoire la commande **copieexam**, qui sauvegardera votre répertoire de travail.

En cas de doute ou difficulté dans la réalisation de cette procédure, n’hésitez pas à demander de l’aide à l’enseignant en salle. **Vous devez aussi remettre les copies d’examen à l’enseignant en salle et vous assurez d’avoir émarginé la feuille de présence.**

Note 1 (Attention).

Dans l’énoncé qui suit, on a adapté une convention **d’indilage Fortran**, consistant à numéroter les éléments des tableaux, vecteurs et matrices, à partir de 1 (comme en MATLAB). Il faudra par conséquent lors de la mise en oeuvre en langage C se rappeler que dans ce langage, les tableaux, vecteurs et matrices sont indicés à partir de 0 : on parle **d’indilage C**.

Note 2.

Dans ce problème, on s’intéresse à la résolution dans \mathbb{R}^n , ainsi qu’à une application, d’un système de la forme

$$\mathbf{Ax} = \mathbf{f} \quad \text{où} \quad \mathbf{A} = \begin{pmatrix} a_1 & b_1 & 0 & \cdots & 0 & c_1 \\ c_2 & a_2 & b_2 & 0 & & 0 \\ 0 & c_3 & a_3 & b_3 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & & \ddots & c_{n-1} & a_{n-1} & b_{n-1} \\ b_n & 0 & \cdots & 0 & c_n & a_n \end{pmatrix} \quad (1)$$

On posera $\mathbf{a} = [a_1, \dots, a_n]^T$, $\mathbf{b} = [b_1, \dots, b_n]^T$, $\mathbf{c} = [c_1, \dots, c_n]^T$. [$(\cdot)^T$ est l’opérateur transposé].

Thème - 1 Inversion rapide de matrices tridiagonales : c'est-à-dire (1) avec $c_1 = 0, b_n = 0$

On considère ici le cas du système (1) où $c_1 = 0, b_n = 0$. La matrice A est alors dite tridiagonale.

Les questions Q-1 et Q-2 sont indépendantes.

Q-1 : Cas particulier où le calcul explicite de A^{-1} est possible

On suppose : $a_i = 2, \forall i = 1, \dots, n$, et $c_i = -1, \forall i = 2, \dots, n$, et $b_i = -1, \forall i = 1, \dots, n-1$

On introduit alors les n vecteurs $(d^{(k)})_{k=1}^n$ de \mathbb{R}^n

$$d_j^{(k)} = j \quad \text{pour } 1 \leq j \leq k \quad \text{et } d_j^{(k)} = 0 \quad \text{pour } k < j \leq n.$$

Q-1-1 : Calculer les produits scalaires $(Ad^{(k)}, d^{(k')})$ pour $1 \leq k, k' \leq n$.

Q-1-2 : En déduire que les $(d^{(k)})_{k=1}^n$ forment une base de \mathbb{R}^n .

Q-1-3 : On écrit la solution du système $Ax = f$ dans la base des $(d^{(k)})_{k=1}^n$: $x = \sum_{k=1}^n \alpha_k d^{(k)}$.

Calculer les coefficients α_k en fonction de f et des $(d^{(k)})_{k=1}^n$. Commenter.

Q-1-4 : On écrit cette fois x et f dans la base canonique $(e_k)_{k=1}^n$ de \mathbb{R}^n : $x = \sum_{k=1}^n x_k e_k, \quad f = \sum_{k=1}^n f_k e_k$

Exprimer x_k en fonction des f_k . En déduire l'expression explicite suivante de la matrice A^{-1} :

$$(A^{-1})_{k,j} = j \frac{n+1-k}{n+1} \quad \text{pour } j \leq k. \quad (2)$$

Q-1-5 : Est-il plus économique (en nombre d'opérations, i.e. multiplications et divisions) de résoudre le système linéaire $Ax = f$ ou de calculer x par la formule $x = A^{-1}f$? On suppose A^{-1} connu, c'est-à-dire que les coefficients $(A^{-1})_{k,j}$ sont déjà calculés.

Q-2 : Retour au cas où $a_i, b_i, c_i, i = 1 \dots, n$ sont quelconques (Décomposition LU)

On pose

$$L = \begin{pmatrix} d_1 & 0 & \dots & \dots & 0 \\ l_2 & d_2 & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & l_{n-1} & d_{n-1} & 0 \\ 0 & \dots & 0 & l_n & d_n \end{pmatrix} \quad U = \begin{pmatrix} 1 & r_1 & 0 & \dots & 0 \\ 0 & 1 & r_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & 1 & r_{n-1} \\ 0 & \dots & \dots & 0 & 1 \end{pmatrix}$$

Q-2-1 : Montrer que $LU = A$ si et seulement si

$$\begin{aligned} d_1 &= a_1, & d_1 r_1 &= b_1 \\ l_k &= c_k, & d_k &= -l_k r_{k-1} + a_k, & d_k r_k &= b_k \quad k = 2, \dots, n-1, \\ l_n &= c_n, & d_n &= -l_n r_{n-1} + a_n. \end{aligned}$$

Q-2-2 : En résolvant ces équations d'inconnues $l_k, d_k, r_k, k = 1, \dots, n$, déduire que l'on peut stocker A à l'aide de trois vecteurs c, a, b , de taille au plus n chacun. Déduire aussi que l'on peut encore stocker dans ces trois vecteurs la décomposition LU de A . Implementer alors en langage C un algorithme de décomposition LU de la matrice tridiagonale A dont les diagonales principales sont stockées dans c, a, b comme décrit précédemment. Fournir pour cette fin, une fonction de prototype

```
void decompLUTridiag(gsl_vector* c, gsl_vector* a, gsl_vector* b)
```

Q-2-3 : Proposer un algorithme de résolution du système $LUx = f$ et fournir une fonction C

```
void descenteRemonteeTridiag(const gsl_vector* c, const gsl_vector* a, const gsl_vector* b,
                             const gsl_vector* f, gsl_vector* x)
```

qui détermine la solution de x du système tridiagonal $Ax = f$ dont la factorisation LU de la matrice A est stockée dans a, b, c comme décrit à la question précédente.

Thème - 2 Application a une matrice tridiagonale circulante

On considère ici le cas du système (1) où $c_1 \neq 0, b_n \neq 0$. La matrice A est alors dite **tridiagonale circulante**.

Q-3 : On se donne deux réels α et γ . Montrer que l'on peut déterminer deux vecteurs de \mathbb{R}^n :

$$u = \left[\frac{1}{\alpha}, 0, \dots, 0, \frac{1}{\gamma}\right]^T \quad \text{et} \quad v = [v_1, 0, \dots, 0, v_n]^T$$

tels que

$$A = B + uv^T, \quad \text{avec} \quad B = \begin{pmatrix} \tilde{a}_1 & b_1 & 0 & \dots & 0 \\ c_2 & a_2 & b_2 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & c_{n-1} & a_{n-1} & b_{n-1} \\ 0 & \dots & 0 & c_n & \tilde{a}_n \end{pmatrix} \quad (3)$$

On exprimera $\tilde{a}_1, \tilde{a}_n, v_1, v_n$ en fonction de $a_1, a_n, c_1, b_n, \alpha, \gamma$.

Q-4 : Un choix quelconque de α et γ est-il possible ?

Pour un choix de α et β donné, afin de résoudre $Ax = f$, on écrit $(I + B^{-1}uv^T)x = B^{-1}f$ ou encore $(I + zv^T)x = B^{-1}f$ où $z = B^{-1}u$. Il ne reste plus qu'à chercher une expression de $(I + zv^T)^{-1}$.

Q-5 : Soit $M \in \mathcal{M}_{k,k}$ une matrice inversible, $Z, V \in \mathcal{M}_{n,k}$ des matrices données, montrer que

$$(I + ZV^T)(I - ZM^{-1}V^T) = I + Z \left(M - (I_k + V^T Z) \right) M^{-1}V^T$$

où I_k est la matrice identité de $\mathcal{M}_{k,k}$.

En déduire une expression de M pour que $(I + ZV^T)^{-1} = I - ZM^{-1}V^T$

Q-6 : En déduire que la solution de $Ax = f$ est

$$x = \left(I - B^{-1}u(1 + v^T B^{-1}u)^{-1}v^T \right) B^{-1}f$$

Q-7 : Proposer un déroulement de calculs pour déterminer x à l'aide de la formule précédente, sans calculer explicitement la matrice $(I - B^{-1}u(1 + v^T B^{-1}u)^{-1}v^T)B^{-1}$

Q-8 : Écrire une fonction C

```
void inverseCirculante (const gsl_vector* cB, const gsl_vector* aB, const gsl_vector* bB,
                       const gsl_vector* f,  gsl_vector* x)
```

qui calcule la solution x de $Ax = f$ où cB, aB, bB sont respectivement les diagonales inférieure, principale et supérieure de la factorisation LU de la matrice B issue de l'écriture de A sous la forme indiquée ci-dessus voir équation (3).

Thème - 3 Application à une équation aux dérivées partielles

On considère l'équation

$$\begin{cases} \frac{\partial u}{\partial t}(t, x) - \varepsilon \frac{\partial^2 u}{\partial x^2}(t, x) + \beta \frac{\partial u}{\partial x}(t, x) = 0 & \text{dans }]0, T[\times]0, 1[, \\ u(t, 0) = u(t, 1), & \text{dans } [0, T], \\ u(0, x) = u_0(x), & \text{dans } [0, 1] \end{cases} \quad (4)$$

avec ε et β des constantes positives.

On se propose de discrétiser cette équation par une méthode de différences finies en espace (semi-discrétisation spatiale) et en temps (semi-discrétisation temporelle).

Q-9 : **Semi-discrétisation spatiale.** On commence par recouvrir $[0, 1]$ d'une grille de pas $h = \frac{1}{n-1}$. Ceci génère les noeuds

$$\mathcal{X}_n = \{x_i, x_i = (i-1) \times h, i \in \{1, \dots, n\}\}.$$

On écrit ensuite que l'équation est vérifiée en chaque noeud interne, soit

$$\frac{\partial u}{\partial t}(t, x_i) - \varepsilon \frac{\partial^2 u}{\partial x^2}(t, x_i) + \beta \frac{\partial u}{\partial x}(t, x_i) = 0, \quad \forall i \in \{1, \dots, n\}, \quad \forall 0 < t < T.$$

On remplace ensuite les opérateurs différentiels par des différences divisées.

Q-9-1 : Montrer que

$$\frac{\partial^2 u}{\partial x^2}(t, x_i) = \frac{u(t, x_{i-1}) - 2u(t, x_i) + u(t, x_{i+1}))}{h^2} + \mathcal{O}(h^2), \quad \frac{\partial u}{\partial x}(t, x_i) = \frac{u(t, x_i) - u(t, x_{i-1}))}{h} + \mathcal{O}(h)$$

Q-9-2 : On pose pour tout $t \in [0, T]$, $U_i(t)$ la valeur approchée de $u(t, x_i)$ et U_0 le vecteur dont la i -ème composante est $u_0(x_i)$. Montrer que le problème semi-discret spatial approché s'écrit

$$\begin{cases} \frac{dU(t)}{dt} + AU(t) = 0 & \forall 0 < t < 1, \\ U(0) = U_0, \end{cases} \quad (5)$$

où A est une matrice dont la structure est celle donnée en (1). On remarquera et justifiera que $u_{n+1}(t) = u_1(t)$ et $u_0(t) = u_n(t)$.

Q-10 : **Semi-discrétisation temporelle.** Pour achever la discrétisation, on remplace le problème (5) par un problème discret approché. Pour cela on introduit un maillage de $[0, T]$. On découpe $[0, T]$ en m intervalles de pas $\Delta t = \frac{T}{m}$, ce qui génère les points $\mathcal{T}_m = \{t_k = k \times \Delta t, k \in \{0, \dots, m\}\}$. Plusieurs choix sont possibles, on opte pour le schéma implicite : on écrit que le problème (5) est satisfait en tous les points $t_{k+1} = t_k + \Delta t$, soit

$$\frac{dU(t_{k+1})}{dt} + AU(t_{k+1}) = 0.$$

Q-10-1 : En utilisant les développements limités de $h(t + \Delta t)$ montrer que

$$h'(t) = \frac{h(t + \Delta t) - h(t)}{\Delta t} + \mathcal{O}(\Delta t) \quad \text{et donc} \quad \frac{dU(t_{k+1})}{dt} = \frac{U(t_{k+1}) - U(t_k)}{\Delta t} + \mathcal{O}(\Delta t).$$

Q-10-2 : Si on désigne par U^k la valeur approchée de $U(t_k)$, montrer que le problème discret s'écrit

$$\begin{cases} U^0 = U_0, \\ G U^{k+1} = U^k, \quad \forall 0 \leq k \leq m-1. \end{cases} \quad (6)$$

Avec G définie comme en (1) avec $\begin{cases} c_i = -\Delta t \frac{\varepsilon}{h^2} - \Delta t \frac{\beta}{h} \\ a_i = 1 + \Delta t \frac{2\varepsilon}{h^2} + \Delta t \frac{\beta}{h} \\ b_i = -\Delta t \frac{\varepsilon}{h^2} \end{cases} \quad \forall i = 1, \dots, n$

Q-11 : **Mise en oeuvre.**

Proposer un algorithme pour résoudre l'équation de convection-diffusion faisant usage des développements des exercices précédents.

Q-11-1 : Écrire une fonction C de prototype

```
void assembleConvDiff(gsl_vector* cG, gsl_vector* aG, gsl_vector* bG, gsl_vector* U, gsl_vector* X,
                    int n, int m, double epsi, double beta, double T)
```

où n et m sont respectivement les paramètres de discrétisation spatiale et temporelle et **epsi**, **beta** les coefficients de diffusion (ε) et de convection (β), T le temps final, et qui retourne le vecteur X des points du maillage, le vecteur U contenant les valeurs de la donnée initiale u_0 aux points X et la matrice G stocké à l'aide des trois vecteurs cG, aG, bG comme décrit précédemment.

Q-11-2 : Écrire une fonction C de prototype

```
void
miseAJour(const gsl_vector* cG, const gsl_vector* aG, const gsl_vector* bG, const gsl_vector* U, gsl_vector* V)
```

qui retourne à une itération k , le vecteur V valeur de U^{k+1} lorsque $U = U^k$. On essaiera de faire usage des développements des questions précédentes.

Q-11-3 : On prend $\varepsilon = 1e-5, \beta = 0.1, T = 10, n = 200, m = 10000, u_0(x) = e^{-1000(x-0.5)^2}$. Écrire dans un fichier **solution.1.txt** respectivement **solution.m.txt** la solution U^1 respectivement U^m .