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

Examen : 19 mai 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 émargé la feuille de présence.**

Thème - 1 *Gram-Schmidt et matrice symétrique définie positive*

Dans cet exercice, A désigne une matrice de $\mathcal{M}_{n,n}(\mathbb{R})$, symétrique définie positive. On sait dans ce cas que l'application $(x, y) \longmapsto (Ax, y)$ définit un produit scalaire sur $\mathbb{R}^n$. On notera $(\cdot,\cdot)_A$ ce produit scalaire et $\|\cdot\|_A$ la norme associée. On désignera toujours par V^* la transposée au sens du produit scalaire euclidien du vecteur ou matrice V.

$\boxed{\textbf{Q-}1\quad :}$ Soit $\{v_0, \dots v_{n-1}\}$ une famille de n vecteurs orthogonaux dans le produit scalaire $(\cdot,\cdot)_A$. On dit aussi que ces vecteurs sont A-conjugués. On les suppose en plus normés toujours dans ce produit scalaire. On pose $M = [v_0 \dots v_{n-1}]$ la matrice dont la colonne i est v_i.

Soit x un vecteur quelconque de $\mathbb{R}^n$. Posons α ses composantes dans la base (v_i). On écrit alors

$$x = \sum_{i=0}^{n-1} \alpha_i v_i \equiv M\alpha.$$

Q-1-1 **:** Montrer que $\alpha_i = (A\,x, v_i) \quad \forall\, 0 \le i \le n-1$. En déduire que $\alpha = M^*Ax$, puis que $x = MM^*Ax$.

Q-1-2 **:** Conclure que MM^* est l'inverse de A.

Conclusion : Si l'on dispose de n vecteurs A-conjugués, le calcul de l'inverse de A est immédiat.

$\boxed{\textbf{Q-}2\quad :}$ On s'engage à chercher n vecteurs A-conjugués, en orthonormalisant les vecteurs colonnes de A par le procédé d'orthonormalisation de Gram-Schmidt relativement au produit scalaire $(\cdot,\cdot)_A$.

Dans ce qui suit, A_i désignera la i-ème colonne de la matrice A, $i = 0, \dots, n-1$.

Q-2-1 **:** On pose $v_0 = A_0/\sqrt{(A_0, A_0)_A}$. Vérifier que $\|v_0\|_A = 1$.

Q-2-2 : On suppose construits $v_0, \dots v_{k-1}$, et on désire construire v_k. On écrit $v_k = A_k + \sum_{l=0}^{k-1} \gamma_l^k v_l$. Montrer que $\gamma_l^k = -(A_k, Av_l) \quad \forall\, 0 \leq l \leq k-1$. On achève la construction de v_k en le normalisant.

Q-2-3 : Écrire une fonction C de prototype

```
gsl_matrix* appliqueGramSchmidt(const gsl_matrix* A)
```

qui retourne la matrice (M) dont les colonnes sont des vecteurs orthonormés pour $(\cdot,\cdot)_A$ et sont obtenues selon le procédé décrit précédemment.

Q-3 : Expliquer comment résoudre un système linéaire $AX = B$ où $B \in \mathcal{M}_{n,m}(\mathbb{R})$.

Q-3-1 : Écrire une fonction C de prototype

```
void inverseGramSchmidt(const gsl_matrix* A, const gsl_matrix* B, gsl_matrix* X)
```

qui résout un système linéaire avec m second membres stockés comme vecteurs colonnes dans la matrice B. Le résultat est stocké dans la matrice à m colonnes X. *Cette fonction fera appel à la fonction* ***appliqueGramSchmidt*** *précédente.*

Thème - 2 *Comparaison de deux algorithmes de calcul*

Soit $A \in \mathcal{M}_{n,n}(\mathbb{R})$ et $B \in \mathcal{M}_{n,n}(\mathbb{R})$ deux matrices symétriques définies positives telles que $\varrho(A^{-1}B) < 1$. Soit $F \in \mathbb{R}^n$ un vecteur donné et $m \in \mathbb{N}^*$. On se propose de calculer le vecteur U^m le $m+1$-ième terme de la suite $AU^{k+1} = BU^k + F, \quad U^0 = U_0$.
Pour cela on considère les deux algorithmes suivants :

Algorithme 1 (avec résolution répétée)

```
Initialisation :
U^0 = U_0

Iterations :
 Pour k = 0 à m-1
  Calculer: V = B U^k + F
  Resoudre: A U^{k+1} = V
fin Pour
```

Algorithme 2 (avec inversion de la matrice)

```
Initialisation :
 C = A^{-1} B
 E = A^{-1} F
 U^0 = U_0

Iterations :
Pour k = 0 à m-1
  Calculer: U^{k+1} = C U^k + E
fin Pour
```

Q-4 : **On considère l'algorithme 1**

Q-4-1 : Quels formats de stockage sont mieux adaptés pour les matrices A et B ?

Q-4-2 : Peut-on avoir des soucis d'espace mémoire pour cet algorithme ?

Q-4-3 : Quelle est la complexité de cet algorithme lorsque les matrices A, B sont pleines ?

Q-5 : **On considère l'algorithme 2**

Q-5-1 : En s'inspirant de l'exercice 1, justifier la non nécessité de calculer explicitement l'inverse A^{-1} de la matrice A pour déterminer C et E.

Q-5-2 : Quelle est la complexité de cet algorithme lorsque les matrices A, B sont pleines ?

Q-5-3 : Sur un exemple simple de matrice de taille 3, montrer que $A^{-1}B$ peut-être pleine alors que A et B sont creuses.

Q-5-4 : Peut-on être confronté à un problème d'espace mémoire lié au stockage dans cet algorithme ?

Q-6 : Conseillez sur le choix de l'un de ces algorithmes, en vous appuyant sur le potentiel problème d'espace mémoire et sur le fait que pour m fixé, on peut se trouver dans les cas $m \gg n$ ou $n \gg m$ (avec $\gg$ *signifiant très grand).*

On considère l'équation aux dérivées partielles (1) modélisant le déplacement de la chaleur dans une barre homogène $\Omega \subset \mathbb{R}^2$.

$$\left\{\begin{array}{l} \frac{\partial u}{\partial t}(t,x,y) - \alpha\Delta u(t,x,y) = f(t,x,y), \quad \forall (t,x,y) \in]0,T[\times\Omega. \\ u(t,x,y) = 0, \quad \forall (t,x,y) \in [0,T]\times\Gamma, \\ u(0,x,y) = u_0(x,y), \quad \forall (x,y) \in \bar{\Omega}, \end{array}\right. \tag{1}$$

où Γ désigne la frontière de Ω, α est une constante désignant le coefficient de conductivité thermique du matériau isotrope, $f(t,x,y) \in C^0([0,T]\times\Omega)$ est la fonction source de chaleur et $u_0 \in C^0(\Omega)$ est la température à l'instant initial. On a posé $\Delta u(x,y) = \frac{\partial^2 u(x,y)}{\partial^2 x} + \frac{\partial^2 u(x,y)}{\partial^2 y}$. Dans tout ce qui suit, on prendra $\Omega =]0,1[\times]0,1[$.

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-7 **:**	**Semi-discrétisation spatiale.**

On commence par recouvrir Ω d'une grille de pas $h = \frac{1}{n+1}$ (resp. $h = \frac{1}{n+1}$) dans la direction x (resp. y). Ceci génère les noeuds

$$\mathcal{Y}_n = \left\{(x_i,y_j), x_i = i\times h, y_j = j\times h, (i,j)\in\{0,\dots,n+1\}^2\right\}.$$

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

$$\frac{\partial u}{\partial t}(t,x_i,y_j) - \alpha\Delta u(t,x_i,y_j) = f(t,x_i,y_j), \quad \forall (i,j)\in\{1,\dots,n\}^2, \quad \forall 0<t<1.$$

On remplace enfin les opérateurs différentiels par des différences divisées.

Q-7-1 **:** Montrer que

$$\Delta u(t,x_i,y_j) = \frac{-4u(t,x_i,y_j) + u(t,x_{i-1},y_j) + u(t,x_{i+1},y_j) + u(t,x_i,y_{j-1}) + u(t,x_i,y_{j+1})}{h^2} + \mathcal{O}(h^2).$$

Q-7-2 **:** On pose pour tout $t\in[0,T]$, $u_{i,j}(t)$ la valeur approchée de $u(x_i,y_j,t)$. On pose aussi $F_{i,j}(t) = f(t,x_i,y_j)$. On introduit une numérotation naturelle des noeuds internes consistant à les parcourir lignes par lignes, de bas en haut et de gauche à droite. On introduit alors le vecteur $U(t) = (u_{1,1}(t),\dots u_{n,1}(t),\dots u_{1,n-1}(t)\dots u_{n,n-1}(t), u_{1,n}(t)\dots u_{n,n}(t)^T$.
En suivant la même numérotation, on pose $U_0 = (u_0(x_1,y_1),\dots,u_0(x_n,y_1)\dots u_0(x_n,y_n)$ et $F(t) = (F_{1,1}(t),\dots,F_{n,1}(t),\dots,F_{n,n}(t))$.
Montrer que le problème semi-discret spatial approché s'écrit

$$\left\{\begin{array}{l} \frac{dU(t)}{dt} + AU(t) = F(t) \quad \forall 0<t<T, \\ U(0) = U_0, \end{array}\right. \tag{2}$$

où A est le produit par α de la matrice du Laplacien avec conditions aux limites de Dirichlet homogènes, obtenue par différences finies sur le carré unité recouvert d'une grille uniforme de pas $h = \frac{1}{n+1}$, dans chaque direction.

Q-8 **:**	**Semi-discrétisation temporelle.**

Pour achever la discrétisation, on remplace le problème (2) par un problème discret approché. Pour cela on introduit une discrétisation de $[0,T]$. On partitionne $[0,T]$ 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 de schémas sont possibles, on opte pour le schéma de Crank-Nicholson : on écrit que le problème (2) est satisfait en tous les points $t_{k+\frac{1}{2}} = \frac{1}{2}(t_k + t_{k+1})$, soit

$$\frac{dU(t_{k+\frac{1}{2}})}{dt} + AU(t_{k+\frac{1}{2}}) = F(t_{k+\frac{1}{2}}).$$

Q-8-1 **:** En utilisant les développements limités de $h(t+\frac{1}{2}\Delta t)$ et $h(t-\frac{1}{2}\Delta t)$, montrer que

$$h'(t) = \frac{h(t+\frac{1}{2}\Delta t) - h(t-\frac{1}{2}\Delta t)}{\Delta t} + \mathcal{O}(\Delta t^2) \quad \text{et donc} \quad \frac{dU(t_{k+\frac{1}{2}})}{dt} = \frac{U(t_{k+1}) - U(t_k)}{\Delta t} + \mathcal{O}(\Delta t^2).$$

Q-8-2 **:** On utilise aussi la formule $U(t_{k+\frac{1}{2}}) = \frac{1}{2}(U(t_{k+1}) + U(t_k)) + \mathcal{O}(\Delta t^2)$.
On suppose f constante en temps et on pose $F = F^k = F(t_{k+\frac{1}{2}})$.
Si l'on désigne par U^k la valeur approchée de $U(t_k)$, montrer que le problème discret s'écrit

$$\left\{\begin{array}{l} U^0 = U_0, \\ BU^{k+1} = CU^k + \Delta t\ F, \quad \forall\ 0\le k\le m-1. \end{array}\right. \tag{3}$$

Expliciter B, C et F. On exprimera B, C en fonction de A et de la matrice identité I de même taille que A.

Q-9 :	*Analyse du schéma*

Q-9-1 : Consistance. Montrer que le schéma est consistant. Quel est son ordre de consistance en temps et en espace ? (*En se référant à la démarche de construction du schéma, on pourra répondre à la question sans faire de calcul !*).

Q-9-2 : Stabilité. Montrer que que ce schéma numérique est stable en norme $l2$ (i.e en norme $\|\cdot\|_h$).
On rappelle que pour le problème du Laplacien en dimension 2 sur le carré unité discrétisé par différences finies avec un pas $h = \frac{1}{n+1}$ dans chaque direction, si l'on repère les noeuds par leur indices d'abscisse, k et d'ordonnées, l (on parle de numérotation coordonnées par opposition à la numérotation lexicographique décrite ci-dessus), alors les valeurs propres de la matrice du Laplacien seront données par :

$$\lambda_{k,l} = \frac{4}{h^2}\sin^2(\frac{k\pi h}{2}) + \frac{4}{h^2}\sin^2(\frac{l\pi h}{2}), \quad 1 \leq k, l \leq n$$

Q-9-3 : Déduire que ce schéma est convergent et préciser son ordre de convergence en temps et en espace.

Q-10 :	**Mise en oeuvre et validation**

Proposer un algorithme pour résoudre l'équation de la chaleur basée sur l'utilisation du procédé d'orthonormalisation de Gram-Schmidt décrit précédemment.

Q-10-1 : Écrire une fonction C de prototype

```c
void assembleChaleur(gls_matrix* B, gsl_matrix* C, gsl_vector* F, int n, int m, double alpha, double T)
```

où n et m sont respectivement les paramètres de discrétisation spatiale et temporelle, **`alpha`** la conductivité α, T le temps final, qui génère les matrices B, C et le vecteur F. Le terme source $f(t, x, y)$ est supposée fournie à travers une fonction **`double f(double t, double x, double y)`**, on pourra aussi considérer l'expression $f \equiv 1$ donnée dans le test ci-dessous.

Q-10-2 : Écrire une fonction C de prototype

```c
void
miseAjour(const gsl_matrix* B, const gsl_matrix* C, const gsl_vector* F, const gsl_vector* U, gsl_vector*  V)
```

qui produit à une itération k, le vecteur V valeur de U^{k+1} lorsque $U = U^k$.

Note : *Soyez libre de modifier le prototype de la fonction **`miseAjour`**, en particulier si vous avez opté pour un format plus économique de stockage de la matrice C.*

Q-10-3 : On prend $f = 1,\ \alpha = 1,\ T = 1,\ n = 10,\ m = 10,\ u_0 = 0.$
Ecrire dans un fichier **`solution_1.txt`** respectivement **`solution_m.txt`** la solution U^1 respectivement U^m.