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

Devoir surveillé : 07 avril 2016 (Durée 1h15)

IMPORTANT- Consignes

- Les notes de cours et les scripts de TP sont autorisés. La consultation des pages Internet, en particulier de votre message électronique, est interdite. Le non respect de cette consigne entraînera l'annulation de votre note.
- Commencez par créer un répertoire **M325_DS1_###** où **###** est votre NOM. Travailler dans ce répertoire. Il devra contenir tous les fichiers dont vous aurez eu besoin.
- Lorsque la réponse à une question nécessite des commentaires rédigés, vous devez mettre ces commentaires dans vos fichiers en utilisant les commentaires du langage C.
- A la fin de l'examen, vous devez envoyer votre répertoire **M325_DS1_###** zippé, par mail à l' adresse suivante : **jean-baptiste.apoung@math.u-psud.fr**

Thème - 1 *Problème*

On considère l'équation aux dérivées partielles suivante : $\Omega \subset \mathbb{R}^2$.

$$\begin{cases} -\Delta u(x,y) = f(x,y), & \forall (x,y) \in \Omega =]a,b[\times]c,d[. \\ u(x,y) = g(x,y), & \forall (x,y) \in \partial\Omega, \end{cases} \tag{1}$$

où, f et g sont des fonctions données continues.

Nous nous intéressons à sa discrétisation par différences finies sur une grille de $\bar{\Omega}$ de pas h dans chaque direction et de sommets $(x_i, y_j), 0 \leq i \leq n_x + 1, 0 \leq j \leq n_y + 1$ avec $x_0 = a, x_{n_x+1} = b, y_0 = c, y_{n_y+1} = d$.

Un schéma aux différences finies

En séance de TDM (voir TP1) nous avons construit le schéma à 5 points suivant :

$$\begin{cases} \dfrac{-u_{i-1,j} - u_{i+1,j} - u_{i,j+1} - u_{i,j-1} + 4u_{i,j}}{h^2} = f(x_i, y_j), & 1 \leq i \leq n_x \quad 1 \leq j \leq n_y \\ u_{i,j} = g(x_i, y_j) & i, j \text{ tels que } (x_i, y_j) \in \partial\Omega \end{cases} \tag{2}$$

où $u_{i,j}$ est une valeur approchée de $u(x_i, y_j)$ pour tout $0 \leq i \leq n_x + 1, 0 \leq j \leq n_y + 1$.
Le *stencil* associé à ce schéma est le stencil G de la figure FIGURE1.

On souhaite à présent proposer un schémas dont le *stencil* est un peu plus large tout en demeurant compact.

Q-1 : (**Stencils plausibles :**) On propose les *stencils* ci-dessous voir Figure1 ; dire en vous basant sur le cours, lesquels sont susceptibles de produire un schéma numérique pour la discrétisation de (1).(*Les segments de droites sont les interactions engagées dans la production du schéma.*)

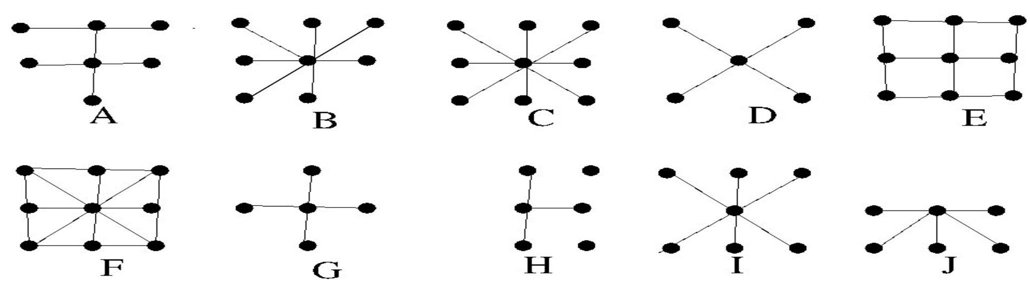

FIGURE 1 – Quelques *stencils*

Q-2 : (**Stencil retenu : schéma à 9 points**)
Dans la suite nous optons pour le schéma à 9 points dont le stencil de production est C de la FIGURE1 et défini par :

$$\left\{\begin{array}{rcl} \multicolumn{3}{l}{\dfrac{-u_{i-1,j-1}-u_{i-1,j+1}-u_{i+1,j-1}-u_{i+1,j+1}-2u_{i-1,j}-2u_{i+1,j}-2u_{i,j+1}-2u_{i,j-1}+12u_{i,j}}{4h^2}} \\ & = & f(x_i,y_j), \quad \forall\, i,j \quad \text{tels que} \quad 1\le i\le n_x, \quad 1\le j\le n_y \\ u_{i,j} & = & g(x_i,y_j), \quad \forall\, i,j \quad \text{tels que} \quad (x_i,y_j)\in\partial\Omega \end{array}\right. \tag{3}$$

Analyse

Q-2-1 : Sachant que le schéma (3) est une combinaison linéaire des schémas de *stencils* respectifs D et G de la FIGURE1, quel est l'odre minimum de consistance de ce schéma ?

Q-2-2 : Peut-on aussi dire un mot sur la stabilité de de ce schéma ?

Mise en oeuvre

Dans cette partie, on prend $f(x,y)=-4, g(x,y)=x^2+y^2, \Omega=]0,1[\times]0,1[$, de sorte que la solution exacte soit $u(x,y)=x^2+y^2$. On se servira des développements faits au $TP1$ en particulier des supports fournis à savoir les fichiers : **`Laplacien2D.h, Laplacien2D.c, test_Laplacien2D.c`**

Q-3 : **Construction et validation :**
Q-3-1 : Construire le système linéaire $Au=F$ associé au problème (3). *On créera une nouvelle fonction*
`void Laplacien2D_construit_systeme_9pts(Laplacien2D* pb)`

Q-3-2 : Résoudre le système linéaire obtenu.

Q-3-3 : Enregistrer la solution obtenue dans un fichier "solution9Pts.txt" dans un format adapté pour *gnuplot*. Afficher (dans le terminal) l'erreur en norme L^∞ commise.

Q-4 : **Comparaison :**
Q-4-1 : Dans un même graphique afficher les solutions obtenue avec le schéma à 5 points (voir (2)) et le schéma à 9 points (voir (3)). Commentez les observations.

Q-4-2 : Afficher en norme L^∞ les erreurs commises pour ces deux schémas. Commentez les résultats.

Vers un problème non stationnaire

On considère l'équation de la chaleur suivante

$$\left\{\begin{array}{rcl} \frac{\partial u(t,x,y)}{\partial t}-\Delta u(t,x,y) & = & f(t,x,y), \quad \forall(t,x,y)\in]0,T[\times\Omega. \\ u(t,x,y) & = & g(t,x,y), \quad \forall(t,x,y)\in[0,T]\times\partial\Omega, \\ u(0,x,y) & = & u_0(x,y), \quad \forall(x,y)\in\bar{\Omega}, \end{array}\right. \tag{4}$$

où f,g,u_0 sont des fonctions suffisamment régulière données, avec une relation de compatibilité entre u_0 et g.

En utilisant le schéma à 9 points en espace sur un maillage $(x_i=a+ih, y_j=c+jh), i=0,\ldots,n_x+1, j=0,\ldots,n_y+1$ et un schéma Euler explicite en temps sur un maillage de pas δt (i.e. $t_k=k\delta t, k=0,\ldots M$) de $[0,T]$ dans sa discrétisation on obtient le schéma totalement discret suivant :

$$\begin{cases} \dfrac{u_{i,j}^{k+1}-u_{i,j}^{k}}{\delta t}+\dfrac{-u_{i-1,j-1}^{k}-u_{i-1,j+1}^{k}-u_{i+1,j-1}^{k}-u_{i+1,j+1}^{k}-2u_{i-1,j}^{k}-2u_{i+1,j}^{k}-2u_{i,j+1}^{k}-2u_{i,j-1}^{k}+12u_{i,j}^{k}}{4h^2} \\ \qquad = f(t_k,x_i,y_j), \quad \forall\, k,i,j \text{ tels que } \quad 0\le k\le M, \quad 1\le i\le n_x, \quad 1\le j\le n_y \\ u_{i,j}^{k} = g(t_k,x_i,y_j), \quad \forall\, k,i,j \text{ tels que } \quad 0\le k\le M, \quad (x_i,y_j)\in\partial\Omega \\ u_{i,j}^{0} = u_0(x_i,y_j), \quad \forall\, i,j \text{ tels que } \quad 0\le i\le n_x+1, \quad 0\le j\le n_y+1 \end{cases} \tag{5}$$

où $u_{i,j}^{k}$ est une valeur approchée de $u(t_k,x_i,y_j)$ pour $0\le k\le M, \quad 0\le i\le n_x+1, \quad 0\le j\le n_y+1$.

Q-5 : Si le schéma à 9 points (3) est d'ordre $q\in\mathbb{N}^*$, sans faire de calcul, quel est l'ordre de consistance en temps et en espace du schéma (5) ?

Q-6 : Toujours sans faire de calcul dire pourquoi le schéma (5) est conditionnellement stable en norme L^∞ et donnez la une relation entre le pas du temps δt et le pas d'espace h assurant la stabilité L^∞ de ce schéma.

Thème - 2 *Quelques scripts importants*

Il vous a été remis en TP les fichiers **Laplacien2D.h, Laplacien2D.c, test_Laplacien2D.c**. vous devez vous en servir pour cette épreuve. Seuls les contenus de **Laplacien2D.h** et **test_Laplacien2D.c** sont fournis ici.

Listing 1 – fichier test_Laplacien2D.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>
#include <mcheck.h>
#include <assert.h>
#include <string.h>

#include <gsl/gsl_linalg.h>
#include <gsl/gsl_rng.h>
#include <gsl/gsl_blas.h>
#include "Laplacien2D.h"

//compilation  gcc -I. Laplacien2D.c test_Laplacien2D.c -o test_Laplacien2D -lgsl -lgslcblas -lm

double  f(double x, double y){return -4.;}
double  g(double x, double y){return x*x + y*y;}

void TestLaplacien2D(){
 Laplacien2D pb;
 pb.a = 0; pb.b = 1;
 pb.c = 0; pb.d = 1;
 pb.f = f; pb.g = g;

int n0 = 4;
for(int i = 1; i < 4; i++){
n0 *= 2;
 pb.n = n0; pb.m = n0;
 Laplacien2D_construit_systeme(&pb);
 Laplacien2D_resout_systeme(&pb);
 Laplacien2D_affiche_solution_gnuplot(&pb,"solution.txt");
 double h = pb.x[1] -  pb.x[0];
 printf(" Erreur commise %f  %f\n", Laplacien2D_calcule_erreur(&pb,g) ,  h );
 Laplacien2D_libere(&pb);
 }
}

//
int main (int argc, char **argv)
{
  mtrace ();
  TestLaplacien2D ();
  muntrace ();
```

```
  return 0;
}
```

Listing 2 – fichier Laplacien2D.h

```
 /*
   M325 : Calcul Scientifique II
   L3 MINT  Univ. Paris Sud ORSAY
   Copyright (C) 2016 APOUNG KAMGA Jean-Baptiste
   This is part of S.A.F.E.M.I.R.A TOOLS
*/
#ifndef LAPLACIEN2D_H
#define LAPLACIEN2D_H
#include <stdio.h>
#include <stdlib.h>
#include <gsl/gsl_linalg.h>
/*===================================================================
//
//    Auteur: J.-B. APOUNG KAMGA
//    Date:  28 / 03 /2015
//    Time:  17:14
//===================================================================*/
/**
   discretisation de
   -u''(x,y) = f          sur ]a,b[ x ]c,d[
    u(a,y) = u(b,y) = g  sur [c,d]
    u(x,c) = u(x,d) = g  sur [a,b]
    sur une subdivision de [a,b] x [c,d] generant
    n points internes suivant l'axe des x
    m points internes suivant l'axe des y
    La matrice sera une matrice carree formee
    de n * m lignes et n* m colonnes.
*/
typedef double (*PFunc)(double x, double y);

typedef struct Laplacien2D{
// description du èproblme
double a, b; // x est dans [a,b]
double c, d; // y  est dans [c,d]
PFunc f; // terme source
PFunc g; // condition Dirichlet

// description de la grille
int n;    // nombre de points internes suivant  l'axe x
int m;    // nombre de points internes suivant  l'axe y

// TOUT CE QUI SUIT EST GENERE POUR VOUS
// VOUS NE DEVEZ FOURNIR QUE LES DONNEES CI-DESSUS
// VOUS NE DEVEZ PAS EFFACER LES MEMOIRES ALLOUEES
// UTILISER POUR CELA  LA FONCTION FOURNIE
// Laplacien2D_libere(void* pb);

double* x; // maillage de [a,b] uniforme de pas h = (b-a)/(n+1)
double* y; // maillage de [c,d] uniforme de pas h = (d-c)/(m+1)
int ** C2I; // table ne numerotation lexichographique des sommets
            // interne du maillage
            // la numerotation globale des sommets est dans un ordre
            // lexicographique: de bas en haut et de gauche a droite.
// èproblme discret
gsl_matrix* A;   //matrice obtenue en stockage pleine
gsl_vector* F; //second membre obtenu
gsl_vector* U;//solution du systeme lineaire
}Laplacien2D;

void Laplacien2D_bien_decrit(Laplacien2D* pb);
//
void Laplacien2D_construit_systeme(Laplacien2D* pb);
//
void Laplacien2D_resout_systeme(Laplacien2D* pb);
//
void Laplacien2D_affiche_solution_gnuplot(Laplacien2D* pb, const char* nomFichier);
//
double Laplacien2D_calcule_erreur(Laplacien2D* pb, PFunc solutionExacte);
//
double Laplacien2D_calcule_erreur(Laplacien2D* pb, PFunc solutionExacte);
//
void Laplacien2D_libere(void* pb);
#endif
```