© *Jean-Baptiste APOUNG KAMGA <jean-baptiste.apoung@math.u-psud.fr>*

Problèmes elliptiques paraboliques, mise en oeuvre avec **FreeFem++**

Thème - 1 *Problème modèle et formulation variationnelle*

Soit $\Omega \subset \mathbb{R}^2$ un ouvert borné Lipschitzien et f, g_D deux fonctions de $\mathbb{R}^2 \to \mathbb{R}$. On considère le problème :

Chercher une fonction $u : \mathbb{R}^2 \to \mathbb{R}$ telle que

$$\left\{ \begin{array}{rcll} -\Delta u = & f & \text{dans} & \Omega, \\ u = & g_D & \text{sur} & \partial\Omega, \end{array} \right. \tag{1}$$

où $\Delta v = \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}$ et $\nabla v = (\frac{\partial v}{\partial x}, \frac{\partial v}{\partial y})$.

Exercice-1 **:** En supposant $f \in L^2(\Omega)$, $g_D \in H^{\frac{1}{2}}(\partial\Omega)$, montrer que la formulation variationnelle du problème (5) est donnée par

$$\left\{ \begin{array}{l} \text{Chercher } u \in u_D + V \text{ tel que} \\ a(u,v) = l(v) \quad \forall v \in V. \end{array} \right. \tag{2}$$

où

$$a(u,v) = \int_\Omega \nabla u \cdot \nabla v \; dx, \quad l(v) = \int_\Omega f v \; dx, \tag{3}$$

u_D est un relèvement continue de g_D dans $H^1(\Omega)$ et V est un sous-espace vectoriel de $H^1(\Omega)$ que l'on précisera.

Exercice-2 **:** Montrer l'existence et l'unicité de la solution du problème (2).

Thème - 2 *Discrétisation par éléments finis P1-Lagrange*

On se propose d'approcher le problème variationnel (2) par une méthode d'éléments finis. On opte pour l'élément fini Lagrange de degré 1. Pour cela, on introduit un maillage triangulaire $\tau_h = \{T_k\}_{1 \le k \le Nma}$ de Ω (c'est-à-dire une partition de Ω formée de triangles telle que le sommet d'un triangle ne soit pas interne à l'arête d'un autre triangle).

Pour tout triangle T, on désigne par h_T son diamètre (i.e sa plus longue arête), et par ρ_T le diamètre de son cercle inscrit T. L'indice h dans τ_h vaut $h = \max_{T \in \tau_h} h_T$.

On définit aussi les espaces

$$\begin{array}{l} H_h = \{v_h \in C^0(\overline{\Omega}) \quad \text{tel que} \quad v_h|_{T_k} \in P_1(T_k), \forall T_k \in \tau_h\}, \\ V_h = H_h \cap V. \end{array}$$

Où $P_1(T_k) = \{p$ définie sur T_k telle que $\exists(a_0, a_1, a_2) \in \mathbb{R}^3 : p(x, y) = a_0 + a_1 x + a_2 y\}$.

Le problème discret devient alors

$$\begin{cases} \text{Chercher } u_h \in u_{hD} + V_h \text{ tel que} \\ a(u_h, v_h) = l(v_h) \quad \forall v_h \in V_h, \end{cases} \tag{4}$$

où u_{hD} est un interpolé de u_D dans H_h.

Exercice-1 : Montrer que V_h est un sous espace vectoriel de V de dimension finie.

Exercice-2 : Montrer que le problème (4) admet une unique solution.

Exercice-3 : Montrer que si le maillage est régulier ((i.e $\exists \beta > 0$ tel que $\frac{h_T}{\rho_T} < \beta \quad \forall T \in \tau_h$), alors il existe une constante $C > 0$ indépendante de h telle que

$$\|u - u_h\|_{H^1(\Omega)} \leq Ch|u|_{H^1(\Omega)}.$$

On admettra la propriété suivante d'approximation par éléments finis Pk-Lagrange :

$\exists C > 0$ telle que $\forall v \in H^{l+1}(\Omega)$ avec $1 \leq l \leq k, \|v - \pi_h^k v\|_{L^2(\Omega)} + h|v - \pi_h^k v|_{H^1(\Omega)} \leq Ch^{l+1}|u|_{H^{l+1}(\Omega)}$.

où π_h^k désigne l'opérateur d'interpolation dans l'espace éléments finis Pk-Lagrange.

Thème - 3 *Mise en oeuvre avec* ***FreeFem++***

La résolution effective du problème (4) peut se faire en dimension deux et trois au moyen des outils **freeFEM** (`http://www.freefem.org/`), en particulier la version **FreeFEM++** (`http://www.freefem.org/ff++/index.htm`), dont la vocation est de rendre l'écriture du code aussi proche que possible de l'écriture variationnelle du problème.

Nous décrivons ci-dessous les étapes à suivre pour résoudre le problème (4) avec l'outil **FreeFEM++**.

1 Construction du maillage

La construction du maillage se fait en un certain nombre d'étapes :

1. **Définition de la frontière** : celle-ci est faite de réunion de courbes fournies sous forme paramétrique.
   (a) **Définition paramétrique de chaque portion de la frontière :** Une courbe paramétrique étant définie par exemple comme : $[t_i, t_f] \ni t \mapsto \begin{cases} x = f(t) \\ y = g(t) \end{cases}$
   On peut par exemple définir la frontière du carré par :

Listing 1 – Définition paramétrique du carré fichier : Laplace.edp

```
//CONSTRUCTION DU MAILLAGE
//definition des frontieres du carre
border abottom(t=0,1){x = t; y = 0; label = 1;};//cote inferieur
border aright(t=0,1) {x = 1; y = t; label = 2;};//cote droit
border atop(t=1,0)   {x = t; y = 1; label = 3;};//cote superieur
border aleft(t=1,0)  {x = 0; y = t; label = 4;};//cote gauche
```

Il faut noter qu'on a donné une référence (par exemple **label = 1**) a chaque courbe afin d'identifier cette portion de la frontière lors de l'introduction des conditions aux limites.

(b) **Discrétisation des portions et leur réunion :** On décide du nombre de points qu'on va mettre sur chaque courbe et on réuni ces portions de courbes pour définir la frontière du domaine. Ainsi, on peut par exemple dessiner la frontière du domaine ci-dessus par le bout de code

Listing 2 – Dessin de la frontière du carré fichier : Laplace.edp

```
//maillage et dessin de la frontiere du carre
int n=2;
plot(abottom(10*n) + aright(10*n) + atop(10*n) + aleft(10*n), wait↩
    =1);
```

2. **Maillage de la surface délimitée** : On maille la surface décrite par la frontière. Ceci suppose évidemment que la frontière décrite précédemment doit être une courbe fermé orientée de sorte que la surface délimitée se trouve à gauche lorsque l'on parcourt la courbe. Le code suivant construit le maillage le dessine et le stocke dans un fichier.

Listing 3 – Génération du maillage fichier : Laplace.edp

```
//generation du maillage de surface
mesh Th = buildmesh( abottom(10*n) + aright(10*n) + atop(10*n) + ↩
    aleft(10*n) );
//dessin du maillage
plot(Th,wait=1);
//sauvegarde du maillage
savemesh(Th,"carre.msh");//on pourra lire la maillage par: Th =↩
    readmesh("carre.msh");
```

2 Définition de l'espace éléments finis

La construction de l'espace élément finis à savoir V_h se fait de la manière suivante

Listing 4 – Espace élément fini fichier : Laplace.edp

```
//CONSTRUCTION DE L'ESPACE ELEMENT FINI
fespace Vh(Th,P1);  //On utilise le maille du carre et l'element fini P1 ↩
    Lagrange
```

Ici on fait usage de l'espace éléments fini P1-Lagrange. On aurait tout aussi pu prendre l'élément fini P2-Lagrange. On aurait dans ce cas remplacé **`P1`** par **`P2`**.

3 Définition du problème variationnel

La définition du problème variationnel suit fidèlement celle du cadre continu. Il convient de noter la manière dont les conditions aux limites sont introduites. Elles sont en effet introduites par pénalisation, le coefficient de pénalisation (*très grosse valeur sur la diagonale*) étant donné par la variable et mot clé (**tgv**). La méthode de résolution du système linéaire est aussi fournie à travers le mot clé **solver**. Dans le présent exemple c'est la méthode de Krylov connue sous l'appellation GMRES pour Generalized Minimal ReSidual. Plusieurs méthodes sont disponibles : CG (gradient conjugué), UMFPACK (méthode directe pour matrice creuse, avec factorisation multi-frontale) et bien d'autres.

Listing 5 – Problème variationnel fichier : Laplace.edp

```
//DEFINITION DU PROBLEME VARIATIONEL
Vh uh,vh;            // uh est l'inconnu et vh la fonction test
func f  = -4.;       //second membre
func gd = x*x +y*y; //condition de dirichlet

problem Laplace(uh,vh,solver=GMRES,tgv=1e5) =  //definition du èproblme
    int2d(Th)(dx(uh)*dx(vh) + dy(uh)*dy(vh))    //forme bilineaire
    -int2d(Th)(f*vh)                            //forme lineaire
    + on (1,2,3,4, uh = gd);                     //condition aux limite
```

. Il faut remarquer que la condition aux limites de Dirichlet est imposée sur toute la frontière, il y en a 4 dont les labels sont (1,2,3,4) voir mot clé **on** dans le script. Si la fonction `gd` n'était pas le même sur chaque frontière, on aurait pu remplacer la dernière ligne par

Listing 6 – Condition aux limites fichier : Laplace.edp

```
 + on(1, uh = gd) + on(2, uh = gd) + on(3, uh = gd) + on(4, uh = gd)
```

4 Résolution du problème

Pour résoudre le problème il suffit d'écrire son nom suivi de point-virgule.

Listing 7 – Résolution fichier : Laplace.edp

```
//RESOLUTION DU PROBLEME
Laplace;
```

Le système linéaire est ainsi résolu et la solution est accessible dans la variable **uh** qui est l'inconnu du problème, ici premier argument du constructeur du problème.

5 Post-Processing

Pour le post-traitement, on peut représenter graphiquement la solution. On peut aussi calculer l'erreur entre la solution exacte et la solution approchée dans diverses normes.

Listing 8 – Posttraitement fichier : Laplace.edp

```
//POST PROCESSING
//Dessin de la solution
plot(Th,uh,ps="Laplace.eps", fill=false, value=true,cmm="solution Laplace"↩
    );
//Calcul de l'erreur
Vh errh = uh - (x*x + y*y); //solution exacte
real errL2 = sqrt(int2d(Th)( errh*errh));          //erreur en norme L2
real errH1 = sqrt(int2d(Th)( dx(errh)* dx(errh)    //erreur en seminorm H1
                      + dy(errh) * dy(errh)
                      ));
cout<<"Erreur L2 "<< errL2<<"\n";
cout<<"Erreur H1 "<< errH1<<"\n";
```

Thème - 4 *Exercices*

Exercice-1 **:** **Conditions aux limites mixtes.** Modifier le problème précédent de sorte à résoudre le même problème avec cette fois une condition de Neumann sur une partie de la frontière.
On rappelle que pour calculer une intégrale sur une partie de la frontière du maillage **`Th`***, on écrit* **`int1d(Th,lab)(...)`**, où **`lab`** désigne le label (l'identifiant) de ladite frontière.

Exercice-2 **:** Comparer les solutions par éléments finis P1 et P2 Lagrange.

Exercice-3 **:** Résoudre le problème de Laplace sur un disque centré en $(0, 0)$ et de rayon 1.

Thème - 5 *Equation de la chaleur*

On considère l'équation de la chaleur donnée par (5) où $\partial\Omega = \Gamma^D \cup \Gamma^N$:

$$\left\{\begin{array}{rcll} \frac{\partial u}{\partial t} - \nabla \cdot (K \nabla u) = & f & \text{dans} &]0, T[\times\Omega, \\ (K\nabla u) \cdot n = & g_N & \text{sur} & [0, T] \times \Gamma^N, \\ u = & g_D & \text{sur} & [0, T] \times \Gamma^D, \\ u(0, \cdot) = & u_0 & \text{sur} & \Omega, \end{array}\right. \tag{5}$$

En vous inspirant de ce qui précède, expliquez comment la résolution de cette équation avec **`FreeFem++`** peut être conduite. Validez l'idée retenue sur un exemple.