Feuille de TP 1 – Mer 23 novembre 2016
Maillage – Mise en œuvre de la méthode des éléments finis P1 conformes

1 Génération d'un maillage par l'outil open source `Freefem++`

Exécuter le script `Freefem++` suivant et analyser son contenu.

```
// SimpleMesh.edp (freefem++)
// Freefem++ :  http://www.freefem.org/ff++/
// Author : F. De Vuyst -- Dec 2013
//
border bord1(t=0,4){x=t; y=0;     label=0;}
border bord2(t=0,4){x=4; y=t;     label=0;}
border bord3(t=2,0){x=2+t; y=6-t; label=0;}
border bord4(t=2,0){x=t; y=4+t;   label=0;}
border bord5(t=4,0){x=0; y=t;     label=0;}
border bord6(t=2*pi,0){x=2+0.5*cos(t); y=2+0.5*sin(t); label=1;}
//
plot(bord1(20)+bord2(20)+bord3(10)+bord4(10)+bord5(20)+bord6(40));
//
mesh Th = buildmesh(bord1(20)+bord2(20)+bord3(10)+bord4(10)+bord5(20)+bord6(40));
plot(Th);
//
savemesh(Th, "simple.amdba");
// Done :).
```

Ouvrir le fichier `simple.amdba` ainsi généré avec un éditeur de texte (type `gedit` ou `Notepad`) et étudier son contenu.

2 Eléments finis

Soit un domaine polygonal Ω du plan $\mathbb{R}^2$. On suppose que $\Gamma_D \cup \Gamma_N = \partial\Omega$ et que Γ_D est de mesure non nulle. On considère le problème suivant (u fonction de $\mathbb{R}^2$ dans $\mathbb{R}$) :

$$\left\{ \begin{array}{rcll} -\nabla \cdot (\mathcal{K}(.)\nabla u) & = & f(x,y) & \text{dans } \Omega \\ u & = & g_1 & \text{sur } \Gamma_D \\ \mathcal{K}\nabla u \cdot n & = & 0 & \text{sur } \Gamma_N \end{array} \right. \tag{1}$$

où f est une fonction de $L^2(\Omega)$ et g_1 une fonction suffisamment régulière.

Exercice 1) 1ere partie

Le but de cet exercice est d'**écrire un script** Matlab qui lit le maillage dans le fichier `DOM1.amdba`, trace le domaine et son maillage, assemble la matrice et le second membre, trace le profil des matrices, tient compte des conditions aux limites de type Dirichlet, calcule la solution et la trace (surface 3d et iso-valeurs). Pour cela on utilisera les programmes fournis dans le paquetage `EF.tgz` (ou `EF.zip` au format .zip) :

- `lect_mesh.m`
- `trace_bord.m`
- `assemb_A.m` (à compléter)
- `assemb_F.m` (à compléter)
- `tri_contour.m`

et les programmes `triplot`, `trisurf`, `spy` et \ de MATLAB.

On prendra pour le premier exemple $\mathcal{K} = 1$, $f = 0$, $\Gamma_N = \emptyset$ et
- $g_1(S) = 0.0$ si zone$(S) = 1$
- $g_1(S) = 2.0$ si zone$(S) = 2$
- $g_1(S) = -1.0$ si zone$(S) = 3$.

Lecture d'un maillage

Un maillage de Ω composé de nbt triangles et de nbs sommets est fourni par l'intermédiaire d'un fichier ayant l'extension `amdba` comme par exemple `DOM1.amdba`. La première ligne de ce fichier donne nbs, nbt. Les nbs lignes suivantes fournissent le numéro, les coordonnées et le numéro de zone des sommets. Les nbt lignes suivantes donnent pour chaque triangle le numéro, les numéros de sommets et le numéro de zone des triangles.

La fonction **[mesh]=lect_mesh(text)** reçoit en argument la variable `text` qui est une chaîne de caractères contenant le nom du fichier du maillage sans l'extension et retourne la variable `mesh` qui est une structure au sens Matlab présentant les champs suivants : `nbs, nbt, nbab, elm_som, som_coo, som_zon, abd_som`. Cette fonction lit le fichier du maillage, construit la structure mesh. En particulier, elld numérote les arêtes de bord (dont les zones des 2 sommets sont > 0) et remplit le champ `abd_som` qui repère pour chaque arête de bord le numéro des sommets. Les champs de la structure sont donc
- nbs est le nombre de sommets
- nbt est le nombre de triangles
- $nbab$ est le nombre d'arêtes sur le bord
- elm_som est un tableau tel que $elm\_som(ie, j) = is_j^{ie}$ avec $j = 1, 2, 3$ et $ie = 1, ..., nbt$
- som_coo est un tableau tel que $som\_coo(is, j) = x_j^{is}$ avec $j = 1, 2$ et $is = 1, ..., nbs$
- som_zon est un tableau tel que $som\_zon(is) = z^{is}$ avec $is = 1, ..., nbs$
- abd_som est un tableau tel que $abd\_som(nf, j) = is_j^{nf}$ avec $nf = 1, ..., nbab$

Les structures au sens Matlab sont des objets encapsulant des données (pouvant être de n'importe quel type) dans des "champs". Pour accéder aux champs, les structures utilisent le point $\cdot$. Pour construire une structure, il suffit d'assigner aux champs des valeurs : $mesh.nbt = 1424$.

Remarque : $\mathcal{K}$ est supposée constante par élément et donnée par un tableau tel que $\mathcal{K}(ie)$ avec $ie = 1, ..., nbt$ est la valeur moyenne de $\mathcal{K}$ sur l'élément ie.

Pour lire le fichier de maillage, tracer le bord du maillage ainsi que le maillage, dans la fenêtre de commande, tester les commandes suivantes :

```
mesh = lect_mesh('DOM1') % Lecture du fichier 'DOM1.amdba'
trace_bord(mesh)
triplot(mesh.elm_som, mesh.som_coo(:,1), mesh.som_coo(:,2)); % help triplot
```

Résolution du problème du Laplacien par la méthode des éléments finis P1

On rappelle que (1) est équivalent au problème variationnel suivant

$$\begin{cases} u \in \{w \in H^1(\Omega), w = g_1 \text{ sur } \Gamma_D\} \\ \displaystyle\int_\Omega \mathcal{K}\nabla u \cdot \nabla v \, dxdy = \int_\Omega fv \, dxdy \quad \forall v \in \{w \in H^1(\Omega), w = 0 \text{ sur } \Gamma_D\} \end{cases} \tag{2}$$

et que la méthode des éléments finis P1 consiste à chercher une approximation de la solution de ce problème dans l'espace défini par

$$V_h = \{v \in C^0(\Omega),\, v_{|K} \in P_1(K), \forall K \in \mathcal{T}_h\}$$

construit sur un maillage $\mathcal{T}_h$ de Ω. On note $(\varphi_i)_{1 \leqslant i \leqslant nbs}$ la base nodale de V_h : on rappelle que $\varphi_i \in V_h$ et que $\varphi_i(S_j)$=δ_{ij} pour S_j sommet du maillage.
On cherche alors $u_h \in V_h$ sous la forme

$$u_h = \sum_{S_i \notin \Gamma_D} u_i \varphi_i + \sum_{S_i \in \Gamma_D} g_1(S_i)\varphi_i$$

telle que

$$\int_\Omega \mathcal{K}\nabla u_h \nabla \varphi_j \, dxdy = \int_\Omega f\varphi_j \, dxdy \quad \forall j \quad \backslash \quad S_j \notin \Gamma_D\,. \tag{3}$$

On notera que l'on est conduit à la résolution d'un système linéaire où les inconnues sont les coordonnées u_i (valeur de u_h en S_i pour $S_i \notin \Gamma_D$).

Assemblage de la matrice de rigidité

Compléter la fonction matlab **function [A]=assemb_A(mesh)** qui assemble la matrice (de type creuse, sparse en anglais) de rigidité A égale à

$$A(is, js) = \int_\Omega \mathcal{K}\nabla\varphi_{is}\nabla\varphi_{js} \, dxdy$$

On rappelle que $\displaystyle\int_\Omega \mathcal{K}\nabla\varphi_{is}\nabla\varphi_{js} \, dxdy = \sum_K \int_K \mathcal{K}\nabla\varphi_{is}\nabla\varphi_{js} \, dxdy$ et que $\displaystyle\int_K \mathcal{K}\nabla\varphi_{is}\nabla\varphi_{js} \, dxdy$ est nul si S_{is} ou S_{js} n'est pas sommet de K. On a donc

$$A(is, js) = \sum_{K \ni \{S_{is}, S_{js}\}} \int_K \mathcal{K}\nabla\varphi_{is}\nabla\varphi_{js} \, dxdy\,.$$

Pour le triangle K, on appelle **contributions élémentaires**

$$B_K(i,j) = \int_K \mathcal{K}\nabla\varphi_{is(i)}\nabla\varphi_{is(j)} \, dxdy$$

où $i = 1, 2, 3$ et $j = 1, 2, 3$ et $is(i)$ (resp. $is(j)$) représente le numéro du $i$ème (resp. $j$ème) sommet de K ; elles sont fonctions des coordonnées des sommets de K (voir ci-dessous).

Pour assembler la matrice, il suffit alors de parcourir les triangles K, de calculer les contributions élémentaires $B_K(i, j)$ et de les accumuler dans $A(is(i), is(j))$. Pour initialiser la matrice creuse, l'instruction `sparse(n,n)` crée *the ultimate sparse matrix*, la matrice nulle d'ordre n.

Calcul des gradients élémentaires

Il est donc nécessaire d'estimer pour $S_{is(i)}$, $i$ème sommet du triangle K, le gradient de la fonction de base associée :

$$\nabla\varphi_{is(i)_{|K}} = \begin{pmatrix} a_i \\ b_i \end{pmatrix}$$

où $\varphi_{is(i)}(x, y) = a_i x + b_i y + c_i$. En utilisant, $\varphi_{is(i)}(S_{is(j)}) = \delta_{ij}$, on obtient

$$a_i = \frac{\begin{vmatrix} \delta_{1i} & y_1 & 1 \\ \delta_{2i} & y_2 & 1 \\ \delta_{3i} & y_3 & 1 \end{vmatrix}}{\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}} \quad \text{et} \quad b_i = \frac{\begin{vmatrix} x_1 & \delta_{1i} & 1 \\ x_2 & \delta_{2i} & 1 \\ x_3 & \delta_{3i} & 1 \end{vmatrix}}{\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}}$$

ou encore, en observant que $\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} = 2|K|$, on a

$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} = \frac{1}{2|K|} \begin{pmatrix} y_2 - y_3 \\ y_3 - y_1 \\ y_1 - y_2 \end{pmatrix} \quad \text{et} \quad \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \frac{1}{2|K|} \begin{pmatrix} -x_2 + x_3 \\ -x_3 + x_1 \\ -x_1 + x_2 \end{pmatrix}$$

Assemblage du second membre

On approche l'intégrale $\displaystyle\int_\Omega f\,\varphi_{is}\,dxdy$ par la formule de quadrature suivante : $\displaystyle\sum_K |K|\, f(G_K)\,\varphi_{is}(G_K)$ où G_K est le centre de gravité de K.

Déterminer sur feuille la valeur de $\varphi_{is}(G_K)$ pour S_{is} est sommet de K. Compléter la fonction matlab **function [F]=assemb_F(f,mesh)** qui assemble le vecteur colonne F tel que

$$F(is) = \int_\Omega f\varphi_{is}\,dxdy.$$

On supposera que la fonction f est une fonction de 2 variables. On pourra par exemple la définir dans le script à l'aide de l'opérateur @.