Université de Rouen
M2 MFA
2015-2016

Outils informatiques pour la recherche

Initiation au logiciel FreeFem++
Résolution d'une EDP en 2D

Exercice 1 Première utilisation du logiciel *FreeFem++* en 2D

Q1. Écrire dans "éditeur" du logiciel le programme suivant

```
real L1=2, L2=4;
int nbseg= 20;
real aspratio = L2/L1;

    border Gamma1(t=0,L1) {label=1;x=t ;y=0;};
    border Gamma2(t=0,L2) {label=2;x=L1;y=t;};
    border Gamma3(t=L1,0) {label=3;x=t ;y=L2;};
    border Gamma4(t=L2,0) {label=4;x=0 ;y=t;};

//===============
// Maillage
//===============

    mesh Th=buildmesh(Gamma1(nbseg)+Gamma2(nbseg*aspratio)
                     +Gamma3(nbseg)+Gamma4(nbseg*aspratio));

    plot(Th, wait=1);

//===============
// Une fonction elements finis
//===============
    fespace Vh(Th,P1);    // espace P1
    Vh f = x*x+y*y;

    plot(f,fill=1,wait=1,cmm="Graph de f");

//===============
// Valeur au point P
//===============

    real xp=L1/2, yp=L2/2;
    cout << "valeur au point P(" << xp << "," << yp << ")=" << f(xp,yp) <<endl;
```

• Créer le dossier `Initiation` (commande `mkdir` dans un Terminal). Sauver, dans le dossier `Initiation`, le programme comme `Initiation_ex1_q1.edp`.

Q2. Rajouter un trou circulaire au domaine. Sauver le nouveau programme comme `Initiation_ex1_q2.edp`.

Q3. En utilisant la représentation paramétrique des courbes frontières, construire le maillage d'un *smiley*. Sauver le nouveau programme comme `Initiation_ex1_q3.edp`.

Exercice 2 Interpolation 2D avec FreeFem++ (programme `Initiation_ex2.edp`)

Nous allons explorer les possibilités offertes par FreeFem++ pour interpoler une fonction 2D, $f(x, y)$.

Q1. Construire le maillage (`Thg`) du carré $[-1, 1] \times [-1, 1]$ avec $(n = 20) \times (m = 30)$ points distribués sur les frontières (consulter la documentation pour l'instruction `square`).

Q2. Définir l'espace des éléments finis P^1 (linéaires) associé à `Thg` et représenter sur ce maillage grossier la fonction $f(x, y) = (x + y) * exp(-6. * (x * x + y * y))$. On désigne par `fg` l'approximation P^1 ainsi obtenue. Visualiser cette variable.

Q3. Construire un maillage fin (`Thf`) du même carré avec $(n = 70) \times (m = 70)$ points distribués sur les frontières. Trouver `ff`, l'interpolé de `fg`, sur le nouveau maillage. Visualiser `ff`. Commenter.

Q4. Réaliser la situation inverse : on dispose d'un maillage fin et on calcule l'interpolé de la fonction sur un maillage grossier. (programme `Initiation_ex2b.edp`)

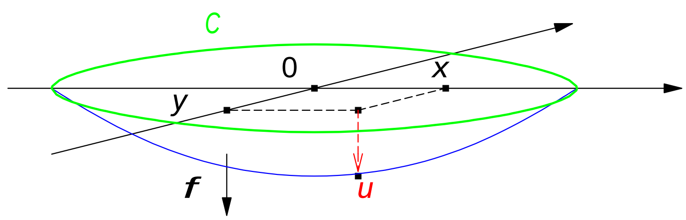
Exercice 3 Intégration 2D avec FreeFem++ (programme `Initiation_ex3.edp`)

Q1. Calculer avec FreeFem++ l'aire d'une ellipse de demi-axes a et b. Comparer avec la valeur exacte.

Q2. Calculer ensuite le périmètre de l'ellipse. Comparer avec une formule approchée.

Exercice 4 Déformation d'une membrane élastique

Considérons une membrane élastique fixée à un anneau circulaire horizontal définie par $\mathcal{C}$ = cercle de centre 0 et de rayon 1. Le déplacement vertical $u(x, y)$ de la membrane sous l'action d'une force de pression verticale $f(x, y)$ est le solution de l'EDP :



$$\begin{cases} -\Delta u &= f \qquad \text{pour} \quad (x, y) \in \mathcal{D} \\ u &= 0 \qquad \text{pour} \quad (x, y) \in \partial\mathcal{D} = \mathcal{C} \end{cases}$$

avec $\mathcal{D}$ le disque de frontière $\partial\mathcal{D} = \mathcal{C}$.

Q1. Ecrire la formulation variationnelle correspondante.

Q2. Si la solution exacte du problème est $u_{ex}(x, y) = 1 - x^2 - y^2$, trouver l'expression correspondante de $f(x, y)$.

Q3. Définir le maillage du disque $\mathcal{D}$, avec n points distribués sur la frontière. La valeur de n sera introduite par l'utilisateur du programme (instructions `cin, cout`).

Q4. **(programme `Initiation_ex44.edp`)**
Résoudre le problème pour la fonction $f(x, y)$ trouvée à la question Q2. Tracer la solution u (iso-contours et en 3D).

Q5. Comparer u avec la solution exacte u_{ex}. Commenter.

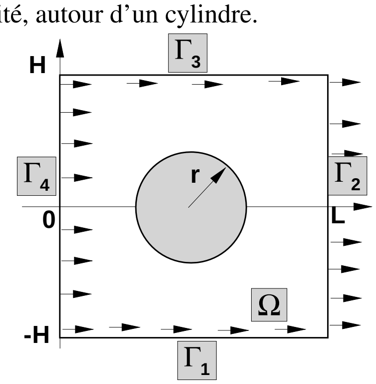
Q6. **(programme `Initiation_ex46.edp`)**
Modifier le script précédent afin de résoudre le problème

$$-\Delta u + u = f \quad \text{sur} \quad \Omega, \quad \text{et} \quad u|_\Gamma = g. \tag{1}$$

avec $\Omega = [0, L] \times [0, L]$, $\Gamma = \partial\Omega$. Choisir f et g afin d'avoir la même solution exacte u_{ex} que précédemment.

Exercice 5 Écoulement fluide autour d'un cylindre

Considérons l'écoulement plan, stationnaire d'un fluide parfait sans viscosité, autour d'un cylindre.



La vitesse d'une particule située en un point (x, y) est dérivée à partir de la fonction potentiel φ

$$\vec{v} = \nabla\varphi = \begin{pmatrix} \dfrac{\partial\varphi}{\partial x} \\ \dfrac{\partial\varphi}{\partial y} \end{pmatrix}, \qquad \text{avec } \varphi \text{ solution de :} \quad -\Delta\varphi = 0.$$

Le domaine de calcul sera le carré $\Omega = [0, L] \times [-H, H]$.
Le cylindre de rayon r est placé au milieu du domaine.

Nous supposons que les frontières du domaines ont suffisamment éloignées du cylindre pour retrouver un courant uniforme, de vitesse constante. **Les conditions aux limites seront donc** :

$$\text{sur} \quad \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \Gamma_4 : \quad \vec{v} = v_0\, \vec{e}_x, \quad \text{avec } \vec{e}_x \text{ le vecteur unitaire suivant } x.$$

$$\text{sur la frontière du cylindre} \quad \Gamma_5 : \quad \vec{v} \cdot \vec{n} = 0, \quad \text{la vitesse normale est nulle.}$$

Q1. Exprimer les conditions aux limites pour φ, en calculant $\partial\varphi/\partial n$ sur chaque frontière $\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4, \Gamma_5$.
Rappelons la définition de la dérivée suivant la normale : $\partial\varphi/\partial n = \nabla\varphi \cdot \vec{n}$**.**
Attention, la normale $\vec{n}$ **est dirigée toujours vers l'extérieur du domaine !**

Q2. Écrire la formulation variationnelle du problème $-\Delta\varphi = 0$, en tenant compte des conditions aux limites établies précédemment.

Q3. Résoudre le problème avec FreeFem++. Tracer la solution φ.

$$\boxed{\text{Données numériques : } L = 8, H = 4, r = 1, v_0 = 1.}$$

Q4. Calculer les deux composantes de la vitesse et tracer les vecteurs vitesse.
Indication : on peut utiliser la syntaxe FreeFem++ pour calculer les dérivées `u=dx(phi); v =dy(phi);`
Pour représenter les vecteurs, on utilise `plot([u,v]);`