Université Pierre et Marie Curie – Paris 6 | Projet 2
Informatique scientifique en C++ | Mars 2005

Projet 2 - deuxième partie
Résolution de l'équation **stationnaire** *de la chaleur en 2D : problème du four*

Le four : problème physique et modélisation

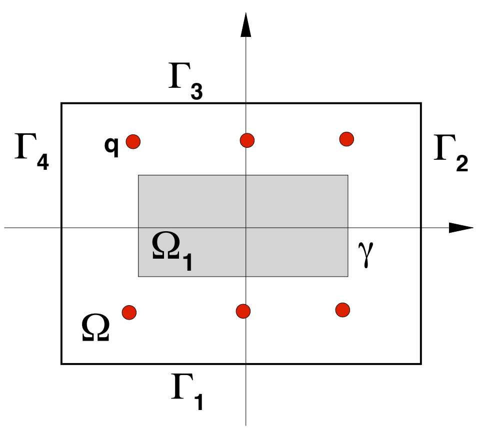
Il s'agit de déterminer la température dans un four destiné à la cuisson d'une pièce de résine thermo–formée (un pare–chocs de voiture par exemple). Les éléments chauffants sont des résistances électriques. À partir de la valeur connue de chacune des résistances, on cherche à calculer le champ de température à l'intérieur du four, et en particulier la température de l'objet à cuire. Sachant que la température idéale de cuisson conditionne la solidité du produit fini et que des essais expérimentaux sont longs et onéreux, cette simulation numérique permet de vérifier à moindre coût la validité des réglages du four.

Considérons les domaines

$$\Omega = [-L/2, L/2] \times [-H/2, H/2] \qquad \Omega_1 = [-l/2, l/2] \times [-h/2, h/2],$$

représentant, respectivement, le four et la pièce (nécessairement $\Omega_1 \subset \Omega$). Les frontières correspondantes seront notées par $\Gamma = \partial\Omega$ et $\gamma = \partial\Omega_1$. La température d'équilibre dans le four est la solution de l'équation :

$$-\nabla(k\nabla u) = f, \quad \Longleftrightarrow \quad -\left[\frac{\partial}{\partial x}\left(k(x,y)\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(k(x,y)\frac{\partial u}{\partial y}\right)\right] = f(x,y), \quad \text{dans} \quad \Omega \tag{1}$$



avec les conditions aux limites sur $\Gamma = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \Gamma_4$

$$\begin{cases} \text{sur } \Gamma_1 & u = u_s & \text{température sud imposée} \\ \text{sur } \Gamma_2 & \partial u/\partial x = 0 & \text{paroi isolée} \\ \text{sur } \Gamma_3 & u = u_n & \text{température nord imposée} \\ \text{sur } \Gamma_4 & \partial u/\partial x = 0 & \text{paroi isolée} \end{cases} \tag{2}$$

la diffusivité thermique

$$k(x,y) = \begin{cases} k_1 & \text{sur } \Omega \backslash \Omega_1 \\ k_2 & \text{sur } \Omega_1 \end{cases} \tag{3}$$

et le terme source $f(x,y)$ qui sera défini par la suite.

Problème direct

Q4 : Ecrire la formulation variationnelle du problème (1) en tenant compte de la discontinuité de la fonction $k(x,y)$. En sachant que le flux de chaleur à travers une frontière s'écrit sous la forme $k(x,y)\partial u/\partial n$, expliquer ce qui se passe sur la frontière γ (donner également une interprétation physique).

Q2 : On considère le **problème sans sources**, décrit par l'équation (1) avec $f(x,y) = f_0 = 0$ et les conditions aux limites (2) avec $u_n \neq 0, u_s \neq 0$. On écrit de manière formelle ce problème

$$Pb_0 := \{\text{éq. (1)} \oplus \text{cond. limites (2)}, \text{ avec } f(x,y) = f_0 = 0, u_n, u_s\} \tag{4}$$

a) Ecrire un script `FreeFem++` pour calculer la solution U_0 de ce problème.
b) En utilisant les classes `sfem`, écrire un programme `C++` pour résoudre le même problème. Comparer avec la solution trouvée au point a) (tracer les isolignes).
Données numériques : $L = H = 2, l = 1, h = 0.4, k_1 = 1, k_2 = 10, u_s = 100, u_n = 50$.

Q5 : On va placer maintenant une résistance chauffante q : elle va agir dans le sous-domaine Ω_r, défini comme le cercle de rayon R_r, centré en (x_r, y_r). La présence de la résistance sera modélisée en considérant dans l'équation (1) le terme source :

$$f(x,y) = f_r(x,y) = \begin{cases} q & \text{sur } \Omega_r \\ 0 & \text{ailleurs.} \end{cases} \tag{5}$$

On définit les problèmes suivants :

$$Pb := \{\text{éq. (1)} \oplus \text{cond. limites (2), avec } f(x,y) = f_r, u_n, u_s\} \tag{6}$$

$$Pb_r := \{\text{éq. (1)} \oplus \text{cond. limites (2), avec } f(x,y) = f_r, u_n = 0, u_s = 0\} \tag{7}$$

et on note par U, respectivement, U_r les solutions correspondantes.

a) Montrer que $U = U_0 + U_r$, où U_0 est la solution du problème sans sources (4).

b) Ecrire un programme `C++` utilisant les classes `sfem` pour résoudre les problèmes (6) et (7) et vérifier la relation prouvée au point **a)**.
Données numériques : $q = 25000, x_r = -l/2, y_r = (H+h)/4, R_r = (H-h)/20$.
<u>Question facultative :</u> Résoudre le même problème avec `FreeFem++` et comparer avec la solution obtenue précédemment.

c) Résoudre le problème (6) avec 6 résistances définies par $q = 25000, R_r = (H-h)/20$ et

$$(x_r, y_r) \in \left\{(-l/2, \frac{H+h}{4}), (0, \frac{H+h}{4}), (l/2, \frac{H+h}{4}), (-l/2, -\frac{H+h}{4}), (0, -\frac{H+h}{4}), (l/2, -\frac{H+h}{4})\right\}.$$

Q6 : Problème inverse

Le problème inverse consiste à déterminer les valeurs des résistances (**la position des résistances est fixe**) qui permettent de chauffer l'objet à la température idéale de cuisson. On utilise la propriété démontrée au point **a)** de la question **Q4** (la linéarité de la solution) pour écrire la solution du problème (6) avec nr résistances sous la forme :

$$U(x,y) = U_0(x,y) + \sum_{k=1}^{nr} q_k U_{rk}(x,y), \tag{8}$$

où U_0 est la solution du problème sans sources (4), q_k la valeur (intensité) de la résistance k et U_{kr} la solution obtenue quand seule la résistance k chauffe le four (i.e. solution du problème (7) pour $q = 1$ et f_r dépendant de la position de la résistance).

Pour obtenir la température idéale U_{opt} dans l'objet à cuire Ω_1 il faut donc déterminer les valeurs q_k des résistances. On les obtient par minimisation de la fonctionnelle

$$J(q) = \int_{\Omega_1} \left[U_{opt}(x,y) - U_0(x,y) - \sum_{k=1}^{nr} q_k\, U_{rk}(x,y)\right]^2 dxdy.$$

La fonctionnelle J est strictement convexe en q. Elle atteint son unique minimum pour la valeur de q qui annule son gradient. Pour $k = 1, 2, \ldots, nr$ la composante k du vecteur gradient est

$$\frac{\partial J}{\partial q_k} = 2 \int_{\Omega_1} \left[U_{opt}(x,y) - U_0(x,y) - \sum_{k'=1}^{nr} q_{k'}\, U_{k'}(x,y)\right] U_{rk}(x,y) dxdy.$$

Définissons alors la matrice $\tilde{A} \in \mathbb{R}^{nr \times nr}$ et le vecteur $\tilde{b} \in \mathbb{R}^{nr}$ par

$$\tilde{A}_{k,k'} = \int_{\Omega_1} U_{rk}(x,y)\, U_{rk'}(x,y) dxdy \quad \text{et} \quad \tilde{b}_k = \int_{\Omega_1} U_{rk}(x,y) \big(U_{opt}(x,y) - U_0(x,y)\big) dxdy.$$

La valeur optimale des résistances est obtenue comme solution du système linéaire : $\tilde{A}q = \tilde{b}$.

Question 6 : Trouver les valuers optimales des 6 résistances utilisées à la question **Q4** point **c)** afin d'avoir $U_{opt}(x, y) = 250$.

Partie 3 : Interperteur et Java

Le problème est de faire un interface au problème inverse de la question **Q6**. La position des résistances et la forme doit être des paramètres, pour cela les résistances seront modélise par un fonction analytique $\psi(x, y)$, c'est a dire que

$$f(x) = \sum_{k=1,nr} q_k \psi(x - c_k) \tag{9}$$

où les c_k sont les centre de résistances. où la formule définissant $\psi(x, y)$ soit modifiable. Pour les tests la fonction ψ est tel que

$$\psi(x, y) = R_r^2 exp(-(x^2 + y^2)/R_r^2). \tag{10}$$

Q7) Modifier l'interpréteur de formule pour qu'il puisse évaluer la fonction ψ qui sera défini dans le fichier de nom `"projects.txt"`, voir annexe.

Pour cela , il faut définir un nouveau terminal `GVAR` pour les 3 paramètres x, y et R_r dans l'analyseur dans la calculette, (voir `http://www.ann.jussieu.fr/~hecht/ftp/InfoSci/fonction/exp.cpp`).

Cet interpréteur doit est évaluer très souvent, donc il doit construit une pseudo fonction. C'est pseudo fonction peut être défini via une classe `Fonction0` (cf. `http://www.ann.jussieu.fr/~hecht/ftp/InfoSci/fonction/fonction0.hpp` qui est réécriture des classes de l'algèbre de fonction pour des fonction sans paramètre car ici les 3 paramètres x, y, R_r sont des variables globale et sont donc des paramètres cachés.

La fonction suivante doit fonctionner :

```
extern double *xxxx,*yyyy;
double  Fr(const Fonction0 & f,  R2 P,R2 C)
{
  R2 X=P-C;
  *xxxx = X.x;                              //    def de x dans l'interpreteur
  *yyyy = X.y;                              //    def de y dans l'interpreteur
  return f();
}
```

où C est le centre de la resistance, f la fonction correspondant à ψ, P le point courant (x, y), et où $xxxx, yyyy$ sont les deux pointeurs (variable global) sur les deux variables qui stockent x, y dans l'interpréteur de calculette.

Remarque la classe `istringstream` (cf. exemple `http://www.ann.jussieu.fr/~hecht/ftp/InfoSci/stl/stringstream.cpp`) permet de faire de la lecture (même type de classe que `istream`) dans une chaine.

Q8) Modifié le programme de la question 6, pour qu'il utilise comme second membre f défini part l'équation (9) ou un problème direct raisonnable du type de la question 5.

Q9) Ecrire une interface en java pour définir le maillage, la fonction ψ et les centres des résistances, et qui lance le programme `C++`, en modifiant à volonter `projet03.java`.

Remarque 1 tous les fichiers sont stocker dans l'archive `http://www.ann.jussieu.fr/~hecht/ftp/InfoSci/file-pj3.tgz`, pour voir la liste des fichiers de l'archive faire

```
tar ztvf files-pj3.tgz
```

et pour extraire tous les fichiers faire (attention au possibilité d'écrasement de fichiers)

```
  tar zxvf files-pj3.tgz
```

les fichiers sont :

```
files-pj3/
files-pj3/projet2_p3_2005.tex
files-pj3/domaine.pdf
files-pj3/projet2_p2_2005.pdf
files-pj3/java/
files-pj3/java/NACA.MSH
files-pj3/java/project.txt
files-pj3/java/projet03.java
files-pj3/java/lecture.cpp
files-pj3/projet2_p3_2005.pdf
files-pj3/domaine.eps
files-pj3/FFF.sty
files-pj3/cpp/
files-pj3/cpp/fonction0.cpp
files-pj3/cpp/fonction0.hpp
files-pj3/cpp/exp.cpp
files-pj3/cpp/CheckPtr.cpp
files-pj3/cpp
```

Remarque, téléchargement de java sous `cygwin`. Dans la page `http://java.sun.com/j2se/1.4.2/download.html` cliquez sur `download J2SE SDK` , puis ok, puis cliquez le item de la liste `Window Platform Window Offline .... 52.23 MO` à télécharger.

Puis installer le logiciel sous windows.

la command javac est : `/cygdrive/c/j2sdk1.4.2_08/bin/javac`

Annexe Voici le exemple de fichier `"projects.txt"` qui contient les données générer par l'interface java contient 5 lignes :

```
"exp( -(x*x+ y*y)*100.)/100."
0.3
0.4
200
0
```

ligne 1 définition de la fonction ψ entre `""`

ligne 2 et 3 donnent respectivement `cx` et `cy` telle que les 6 centres des resistances soit égales à :

$$(-cx, cy), \quad (0, cy), \quad (cx, cy), \quad (-cx, -cy), \quad (0, -cy), \quad (cx, -cy)$$

ligne 4 Valeur de la température cible.

ligne 5 Type de problème à résoudre directe ou optimisation

Exemple complet de lecture du fichier `project.txt` en `C++`

```
/ exemple de gestion du fichier project.txt
                                          //    -----------------------------------------
#include <string>
#include <iostream>
#include <fstream>
#include <sstream>
#include <cassert>
using namespace std;
int main()
```

```
{
  string psi;
  double cx,cy,temperaturecible;
  int cas;
  {
    ifstream file("project.txt");
    assert(file);
                                            //    lecture " .... " complique mais universel
                                                           //    recherche de la premiere "
    while( file.get()!='"' )
      {
        assert(file.good());
      }
    psi = "";
                                                           //    recherche de la derniere "
    while( file.peek()!='"' )
      {
        psi += file.get();                          //    ajoute du caractere a la chaine
      }
    file.get();                                                                  //    eat "
    assert(file.good());                                                  //    fichier ok?
                                        //    fin lecture " .... " complique mais universel
    file >> cx >> cy >> temperaturecible >> cas;
    assert(file.good());
  }
  cout << psi << endl;
  cout << cx << " " << cy << " " << temperaturecible<< " " << cas << endl;
                                                             //    un pseudo fichier string
  istringstream filepsi(psi);
                                                         //    impression du pseudo fichier
  while ( filepsi.peek()!= EOF)
    cout << (char) filepsi.get();
  cout << endl;

return 0;
}
```