

T.P.5 : Résolution et Post-traitement

1 – Résolution

On a construit la matrice A et le second membre b du système linéaire $Au = b$ dont la solution contient le vecteur des valeurs nodales de la solution u du problème discrétisé. Ce vecteur est obtenu par résolution, à l'aide de la méthode de Cholesky, du système dont la matrice est définie selon la Structure de données **Profil** que l'on construira à partir de la Structure de données **Morse Ordonnée** :

- ① **NbLign**
- ② **SecMembre** (de taille **NbLign**)
- ③ **AdPrCoefLi** (de taille **NbLign**)
- ④ **Matrice** (de taille **NbLign+NbCoef**)
- ⑤ **NumCol** (de taille **NbCoef**)

où $\text{NbCoef} = \text{AdPrCoefLi}(\text{NbLign}) - 1$ désigne le nombre de coefficients de la partie triangulaire inférieure stricte de la matrice A .

Structure Profil d'une matrice symétrique

La structure de données représentant ce stockage est définie à l'aide d'un tableau d'entiers **Profil** (les 'pointeurs') et un tableau de réels **MatProf** contenant les coefficients du PROFIL INFÉRIEUR d'une matrice A de taille $n \times n$.

Le profil inférieur d'une ligne est défini par l'ensemble des coefficients compris (au sens des numéros de colonne croissants) entre le premier coefficient non logiquement nul de la ligne et son coefficient sous-diagonal ; le PROFIL INFÉRIEUR de A (dit aussi enveloppe inférieure) résulte de la concaténation par numéros de ligne croissants, des profils inférieurs de ses lignes.

Le tableau **MatProf** contenant les coefficients de A est divisé en deux parties : la première contient les coefficients diagonaux de A rangés par numéro de ligne croissant ; la seconde contient les coefficients strictement sous-diagonaux rangés selon la définition du PROFIL INFÉRIEUR ci-dessus.

Le tableau **Profil** (de longueur utile $n = \text{NbLign}$) contenant les "pointeurs" de la structure permet d'accéder directement au premier coefficient d'une ligne :

Profil($i - 1$), $i = 2, \dots, n$ contient l'adresse (ou rang, compté à partir de 1) du premier coefficient de la ligne i dans la seconde partie du tableau **MatProf** et

Profil(n)-1 représente le nombre de coefficients de l'enveloppe inférieure, *i.e.* le dernier élément de **Profil** contient l'adresse (fictive) du premier coefficient de la ligne (fictive) $n+1$.

- Écrire la fonction **dSMOaPR** qui permet de passer de la structure SMO (**AdPrCoefLi**, **Matrice**, **NumCol**) à la structure profil (**Profil**, **MatProf**). Cela consiste donc simplement à initialiser à 0, puis à reporter les valeurs aux endroits repérés par les numéros de colonnes.
- Une fois cela fait, tout est prêt pour la décomposition de Cholesky de la matrice, puis la résolution du système (descente-remontée), opérations réalisées respectivement à l'aide des fonctions **ltlpr_**, **rsprl_** et **rspru_**. Il s'agit de trois sous-programmes Fortran fournis.

2 – Post-traitement : Affichage de la solution

On suppose connue la solution exacte du problème traité. On note **U(I)**, $I=1, \text{NbLign}$ la solution obtenue par éléments finis, et **UEX(I)**, $I=1, \text{NbLign}$ le vecteur des valeurs nodales de la solution exacte, qui est calculée à l'aide de la fonction **SOLEX**.

- Écrire la procédure `void CalSol(int NbLign, float **coord, float *UEX)`

qui calcule la valeur de la solution exacte **UEX(I)** au nœud I , à partir des coordonnées des nœuds (tableau **coord**), et à l'aide de la fonction `float SOLEX(float *COORDPT)`

qui fournit la solution exacte au point de coordonnées COORPT (voir ci-après).

- On fournit la procédure Fortran :

```
subroutine AFFSOL(NbLign, COOR, U, UEX, IMPFCH)
```

qui, si IMPFCH <= 0, affiche à la demande, de manière interactive, pour 1 ou plusieurs nœuds I consécutifs :

ses coordonnées et les valeurs $U(I), UEX(I), 2 \frac{|U(I)-UEX(I)|}{|U(I)+UEX(I)|}$

(si ce dernier rapport est défini).

Si IMPFCH > 0, ce sous-programme calcule et affiche, dans le fichier `fort.n` avec `n = IMPFCH`, l'erreur quadratique relative (approximation du quotient de $\|u - u_h\|_{L^2}$ par $\|u\|_{L^2}$), ainsi que l'erreur maximum relative (approximation du quotient de $\|u - u_h\|_{L^\infty}$ par $\|u\|_{L^\infty}$).

La dimension caractéristique du maillage h est aussi affichée dans le but de faciliter l'obtention d'une courbe d'erreur.

- Modifier le programme du TP précédent en ajoutant les actions suivantes :

- création de la structure **Profil** (`dSMOaPR`) ;
- calcul de la solution éléments finis (`ltlpr_`, `rsprl_`, `rspru_`) ;
- calcul du tableau des valeurs de la solution exacte `UEX` (`CalSol`) ;
- affichage de la solution (`affsol_`).

Tester ces nouvelles actions du programme avec l'exemple traité dans les T.P. précédents.

3 – Exploitation

On dispose à présent d'un petit code d'éléments finis réalisant les actions suivantes :

- maillage d'un domaine de type quadrilatère ;
- assemblage, au cours duquel interviennent la définition de l'opérateur elliptique (fonctions $a_{\alpha\beta}$ et a_{00}), du second membre (fonction f_Ω) et des conditions aux limites (fonctions f_N , b_N et u_D), les différentes fonctions associées au problème à résoudre ;
- construction du système linéaire : 'rangement' de la matrice assemblée et prise en compte des conditions de Dirichlet ;
- résolution du système linéaire symétrique ;
- affichage et contrôle des résultats.

Il s'agit maintenant de l'exploiter afin de mettre en évidence expérimentalement les résultats théoriques concernant l'estimation de l'erreur dans la méthode des éléments finis.

Vous rendrez un petit rapport joint au programme que vous aurez écrit décrivant vos expérimentations et les conclusions que vous en tirez.

Jeux de données à traiter

Le paramètre de la maille h prend successivement les valeurs $\frac{1}{2}, \frac{1}{4}, \frac{1}{8} \dots$, et le degré d'interpolation vaut 1.

1. **Domaine 1** : Le domaine de calcul est le carré $[0, 1] \times [0, 1]$.

- Résoudre $-\Delta u = f_\Omega$ en choisissant

1. $u(x, y) = 16xy(1-x)(1-y)$, duquel on déduira f_Ω et en imposant une condition de Dirichlet sur les 4 côtés.

2. $u(x, y) = \sin(\pi x) \sin(\pi y)$ en imposant une condition de Dirichlet sur les 4 côtés.

- Résoudre $-\Delta u + u = f_\Omega$ en choisissant

3. $u(x, y) = \cos(\pi x) \cos(\pi y)$, dont on déduira d'une part f_Ω , d'autre part f_N en imposant des conditions de Neumann sur les 4 côtés.

2. Domaine 2 : Le domaine de calcul est maintenant le carré $[0, \frac{1}{2}] \times [0, \frac{1}{2}]$.

Reprendre les exemples précédents ; pour l'exemple 3, imposer les conditions aux limites suivantes : Dirichlet sur les côtés $x = 0$ et $y = 0$, et Neumann sur les deux autres.

Dans tous les cas, on précisera les paramètres (fonctions, conditions de bord, type d'élément...) dont dépendent le calcul, et on expliquera les résultats obtenus, en s'aidant au besoin de graphiques illustrant les résultats de convergence en fonction des paramètres h et d'interpolation. (Préciser dans le rapport les valeurs numériques des erreurs relatives calculées pour les différentes valeurs de h)

Pour faciliter l'exploitation du programme, on pourra utiliser une variable externe `nucas`, lue dans le programme principal et permettant de différencier les différentes fonctions dans les sous-programmes fournissant les fonctions f_Ω , u_D , u_{exacte} , etc, dépendant de chaque cas traité. La fonction SOLEX par exemple pourra alors être écrite comme suit :

```
#include<stdlib.h>
#include<math.h>
#include "resol.h"
/*
    Evaluation de la solution exacte
    -----
*/
extern int nucas ;

float solex(float *coor) {
    const float PI=3.141592 ;
    float val ;

    switch (nucas) {
        case 1 :
            val=16.*coor[0]*coor[1]*(1-coor[0])*(1-coor[1]) ;
            break ;
        case 2 :
            val=sin(PI*coor[0])*sin(PI*coor[1]) ;
            break ;
        case 3 :
            val=cos(PI*coor[0])*cos(PI*coor[1]) ;
            break ;
        default :
            printf("*** SOLEX : exemple non prevu. Abandon.\n") ;
            exit(1) ;
            break ;
    }
    return(val) ;
}
```

La même technique peut être utilisée dans le cas de la fonction f_N pour transmettre la normale au bord du domaine au point d'évaluation (ou le numéro de référence du bord). Cette information peut aussi être transmise en argument.

Nota

Les différents fichiers fournis sont situés dans le répertoire `/usr/local/anam/csmef/tp5`. Afin de faciliter l'obtention d'une courbe de convergence, il est conseillé d'utiliser les maillages situés dans le répertoire `/usr/local/anam/csmef/tp5/maillage`. Leur description est faite dans le fichier README associé.

→ Ne traiter le domaine 2 que si vous disposez d'assez de temps.