Présentation de FreeFem++

Pierre-Henri Tournier

tournier@ljll.math.upmc.fr

Laboratoire Jacques-Louis Lions
Sorbonne Université - Campus Pierre et Marie Curie - Paris VI
INRIA Paris - équipe ALPINES

Pour qui, pour quoi faire ?

FreeFem++ est un logiciel (freeware) écrit en C++ développé au laboratoire Jacques-Louis Lions de Sorbonne Université [F. Hecht, O. Pironneau], porté sous Windows, Unix (Linux) et MacOS. Il permet de résoudre des équations aux dérivées partielles par la méthode des éléments finis, en 2d et 3d.
http://www.freefem.org

Pour quoi faire

1. R&D
2. Recherche académique
3. Cours d'éléments finis, EDP, formulations variationnelles
4. Prototypage d'algorithmes
5. Expérimentations numériques
6. Calcul haute performance, calcul parallèle

Pour qui

chercheurs, ingénieurs, professeurs, étudiants, ...

Historique

1987 MacFem/PCFem première version en Pascal ! (O. Pironneau), pas libre

1992 FreeFem : réécriture en C++ (P1,P0, un seul maillage) O. Pironneau, D. Bernardi, F. Hecht (adaptation de maillage, bamg) , C. Prudhomme

1996 FreeFem+ : réécriture en C++ (P1,P0 plusieurs maillages) O. Pironneau, D. Bernardi, F. Hecht (algèbre de fonctions)

1998 FreeFem++ réécriture du noyau EF et nouveau langage ; F. Hecht, O. Pironneau, K.Ohtsuka

1999 FreeFem 3d (S. Del Pino), première version 3d avec la méthode de domaine fictif

2008 FreeFem++ v3 nouveau noyau EF multidimensionnel : 1d, 2d, 3d

2014 FreeFem++ v3.34 version parallèle

2017 FreeFem++ v3.56

Déc. 2018 FreeFem++ v4.0 !

Caractéristiques principales

- Grand nombre d'éléments finis disponibles: P1, P2, P3 continus, P0, P1 discontinus, RT0, RT1, BDM1, éléments finis d'arêtes, éléments vectoriels, mini-element, ...
- Interpolation automatique entre deux maillages (possibilité de construire la matrice d'interpolation). Une fonction éléments finis peut être vue comme une fonction de (x, y, z) ou bien comme un tableau (de degrés de liberté).
- Définition du problème (réel ou complexe) à partir de la formulation variationnelle, accès aux matrices et vecteurs sous-jacents.
- un grand nombre de solveurs : LU, Cholesky, Crout, CG, GMRES, UMFPack, SuperLU, MUMPS, HIPS, PASTIX, PETSc... ARPACK et SLEPc pour les problèmes aux valeurs propres
- Pour le graphique : OpenGL/GLUT/VTK
- Syntaxe proche du C++
- Version Javascript, utilisable directement dans un navigateur, online ou offline

Caractéristiques principales

- Description analytique des bords du domaine
- Génération de maillages automatique, basée sur l'algorithme de Delaunay-Voronoï, en 2d et 3d(tetgen)
- Sauvegarder et charger facilement le maillage et la solution
- Adaptation de maillage basée sur un calcul de métrique isotropique ou anisotropique. Possibilité de construire la métrique à partir de la hessienne de la solution (en 2d et 3d)
- Liens avec d'autres logiciels : paraview, gmsh, vtk, medit, gnuplot
- Ajout de plugins sous forme de bibliothèques dynamiques
- Interface MPI complète
- Optimisation non-linéaire : CG, Ipopt, NLOpt
- Un grand nombre d'exemples : Navier-Stokes 3d, élasticité 3d, fluide-structure, problèmes aux valeurs propres, décomposition de domaine (Schwarz), ...

Quelques éléments de syntaxe

FreeFem++ emploie un langage dédié (DSL) de haut niveau pour les éléments finis, proche du C/C++
FreeFem++ est un compilateur et prend en entrée un fichier de script

On retrouve les opérateurs du C :
+ - * / ^ // a^b= a^b
== != < > <= >= & | // a|b ≡ a or b, a&b≡ a and b
= += -= /= *=

booléens : 0 <=> false , ≠ 0 <=> true = 1

```
// cast automatique pour les types numériques: bool, int, real, complex
func heavyside = real(x>0.);

for (int i=0;i<n;i++) { ... ;}
if ( <bool exp> ) { ... ;} else { ...;};
while ( <bool exp> ) { ... ;}
break continue key words
```

Quelques éléments de syntaxe pour les éléments finis

```
x,y,z                    //    coordonnées du point courant
label, region            //    label élément de frontière ou
volume
N.x, N.y, N.z     //    vecteur normal unitaire au bord
int i = 0;                                    //    un entier
real a=2.5;                                   //    un réel
bool b=(a<3.);                                //    un booléen
real[int]  array(10);          //    tableau de réels de 10
valeurs
mesh Th; mesh3 Th3;                     //    maillage 2d et 3d
fespace Vh(Th,P2);   //    définition d'un espace EF 2d
fespace Vh3(Th3,P1); //    définition d'un espace EF 3d
Vh u=x;                                 //    fonction EF u ∈ Vh
Vh3<complex> uc = x+ 1i *y; //    fonction EF complexe
u(.5,.6,.7);      //    valeur de la fonction u au point
(.5,.6,.7)
u[];//    tableau des valeurs des degrés de liberté de u
u[][5];                                 //    6e valeur du tableau
```

Quelques éléments de syntaxe : formulation variationnelle

```
fespace V3h(Th,[P2,P2,P1]);
V3h [u1,u2,p]=[x,y,z];     //   fonction EF vectorielle

 //    définition de macros (équivalent de #define en C
)
macro div(u,v) (dx(u)+dy(v))//   EOM
macro Grad(u) [dx(u),dy(u)]//    on signale la fin avec
//
varf a([u1,u2,p],[v1,v2,q])=
             int2d(Th)( Grad(u1)'*Grad(v1)
                        +Grad(u2)'*Grad(v2)
                        -div(u1,u2)*q -div(v1,v2)*p )
             +on(1,2,u1=g1,u2=g2);

matrix A=a(V3h,V3h,solver=UMFPACK);     //   la matrice
real[int] b=a(0,V3h);                   //    le second membre
u1[] =A^-1*b;       //    on résout le système linéaire
```

Equation de Poisson, formulation variationnelle

Soit un domaine Ω, avec $\partial\Omega = \Gamma_2 \cup \Gamma_e$.
Trouver u solution de

$$-\Delta u = 1 \text{ dans } \Omega, \quad u = 2 \text{ sur } \Gamma_2, \quad \frac{\partial u}{\partial \mathbf{n}} = 0 \text{ sur } \Gamma_e \qquad (1)$$

On note $V_g = \{v \in H^1(\Omega)/v_{|\Gamma_2} = g\}$.
Formulation variationnelle : trouver $u \in V_2(\Omega)$ t.q.

$$\int_\Omega \nabla u.\nabla v = \int_\Omega 1v + \int_\Gamma \frac{\partial u}{\partial n} v, \quad \forall v \in V_0(\Omega) \qquad (2)$$

Méthode des éléments finis $\Rightarrow$ remplacer V_g par un espace EF

Equation de Poisson, formulation variationnelle

Méthode des éléments finis $\Rightarrow$ remplacer V_g par un espace EF

Le code FreeFem++ :

```
mesh Th = square(10,10);
fespace Vh(Th,P1);

Vh u,v;
macro Grad(u) [dx(u),dy(u)]//    EOM

solve laplace(u,v,solver=CG) =
     int2d(Th)( Grad(u)'*Grad(v) )
    -int2d(Th)( 1*v )
    +on(2,u=2);

plot(u,fill=1,wait=1);
```