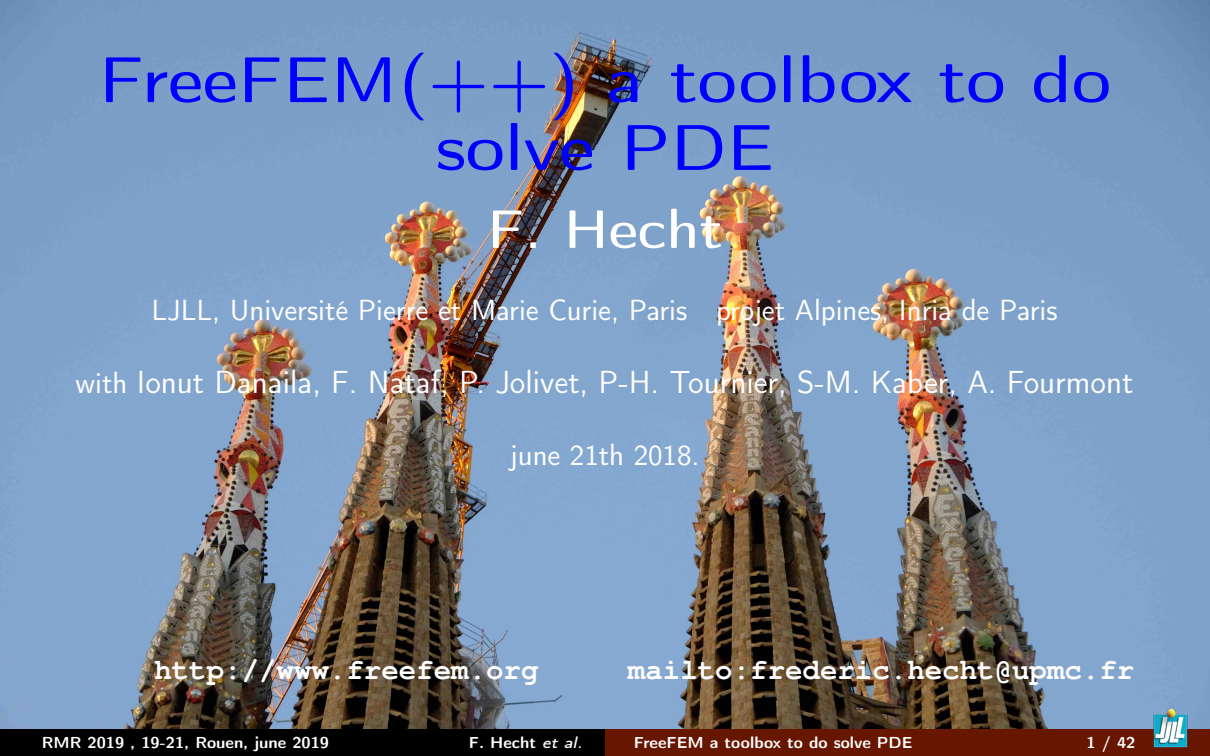
FreeFEM(++) a toolbox to do solve PDE

F. Hecht

LJLL, Université Pierre et Marie Curie, Paris projet Alpines, Inria de Paris

with Ionut Danaila, F. Nataf, P. Jolivet, P-H. Tournier, S-M. Kaber, A. Fourmont

june 21th 2018.



http://www.freefem.org mailto:frederic.hecht@upmc.fr

Outline

Outline

FreeFem++ is a software to solve numerically partial differential equations (PDE) in $\mathbb{R}^2$) , $\mathbb{R}^3$) or on surface with finite elements methods. We used a user language to set and control the problem. The FreeFem++ language allows for a quick specification of linear PDE's, with the variational formulation of a linear steady state problem and the user can write they own script to solve no linear problem and time depend problem. You can solve coupled problem or problem with moving domain or eigenvalue problem, do mesh adaptation , compute error indicator, etc ...

By the way, FreeFem++ is build to play with abstract linear, bilinear form on Finite Element Space and interpolation operator.

FreeFem++ is a freeware and this run on Mac, Unix and Window architecture, in parallel with MPI.
To try of cell phone https://www.ljll.math.upmc.fr/lehyaric/ffjs/

The 11^{th} FreeFem++ days, of Dec, 2019, Sorbonne Université, Jussieu, Paris, France
Info: FreeFem++ solve a problem with $22\,10^9$ unknowns in 200 s on 12,000 proc.

Outline

History

1987 MacFem/PCFem the old ones (O. Pironneau in Pascal) no free.

1992 FreeFem rewrite in C++ (P1,P0 one mesh) O. Pironneau, D. Bernardi, F. Hecht (mesh adaptation , bamg) , C. Prudhomme .

1996 FreeFem+ rewrite in C++ (P1,P0 more mesh) O. Pironneau, D. Bernardi, F. Hecht (algebra of function).

1998 FreeFem++ rewrite with an other finite element kernel and an new language ; F. Hecht, O. Pironneau, K.Ohtsuka.

1999 FreeFem 3d (S. Del Pino) , a fist 3d version base on fictitious domaine method.

2008 FreeFem++ v3 use a new finite element kernel multidimensionnels: 1d,2d,3d...

2014 FreeFem++ v3.34 parallel version

2017 FreeFem++ v3.57 parallel version

2019 FreeFEM v4.1 New matrix type, Surface element, New Parallel tools ... `https://doc.freefem.org/introduction/index.html`

For who, for what!

For what

1. R&D
2. Academic Research ,
3. Teaching of FEM, PDE, Weak form and variational form
4. Algorithmes prototyping
5. Numerical experimentation
6. Scientific computing and Parallel computing

For who: the researcher, engineer, professor, student...

The mailing list mailto:Freefempp@ljll.math.upmc.fr with 546 members with a flux of 1-10 messages per day.

More than 3000 true Users (more than 1000 download / month)

Outline

- Wide range of finite elements: continuous P1,P2 elements, discontinuous P0, P1, RT0,RT1,BDM1, elements ,Edge element, vectorial element, mini-element, ...
- Automatic interpolation of data from a mesh to an other one (with matrix construction if need), so a finite element function is view as a function of (x, y, z) or as an array.
- Definition of the problem (complex or real value) with the variational form with access to the vectors and the matrix.
- Discontinuous Galerkin formulation (only in 2d to day).
- LU, Cholesky, Crout, CG, GMRES, UMFPACK, SuperLU, MUMPS , Dissection, PETSc. ... sparse linear solver; eigenvalue and eigenvector computation with ARPACK or SLEPc.
- Online graphics with OpenGL/GLUT/VTK, C++ like syntax.
- Javascript version works straight out of an HTML page, both online or offline (here).

- Analytic description of boundaries, with specification by the user of the intersection of boundaries in 2d.
- Automatic mesh generator, based on the Delaunay-Voronoï algorithm. (2d,3d (tetgen))
- load and save Mesh, solution
- Mesh adaptation based on metric, possibly anisotropic (only in 2d), with optional automatic computation of the metric from the Hessian of a solution. (2d,Surface,3d).
- Link with other soft: parview, gmsh , vtk, medit, gnuplot
- Dynamic linking to add plugin.
- Full MPI interface
- Nonlinear Optimisation tools: CG, Ipopt, NLOpt, stochastic
- Wide range of examples: Navier-Stokes 3d, elasticity 3d, fluid structure, eigenvalue problem, Schwarz' domain decomposition algorithm, residual error indicator ...

Outline

- to get the Version 4.2-1 do:

```
git clone -b develop   https://github.com/FreeFem/FreeFem-sources ff++
```

- Mesh intersection of get conservative formulation in 2d (too hard, numerical instability $\Longrightarrow$ no)
- rewrite of sparse matrix kernel (Done) ($\Longrightarrow$ new GMRES, new CG, ...)
- Finite Element on curve and Surface (In progress) Thank to A. Fourmont (Inria SED)
- cmake (In progress), Thank to C. Doucet, (Inria SED)
- BEM method (near futur)
- DG in 3d (Future)
- rewrite of the Finite element kernel of isoparametric FE and to mixte surface FE and 3d FE (in future) in a problem.(no template).
- debugger (client-server graphics services) (Good idea but technical)
- more general problem (possible with new matrix)

Outline

2. Sequential Academic Examples
   - Weak form
   - Anisotropic Mesh adaptation
   - Bose Einstein Condensate
   - Search all local min
   - Best Fit
   - Incompressible Navier-Stokes
   - Linear elasticty equation
   - Hyper elasticity equation
   - Surface finite element

Laplace equation, weak form

Let a domain Ω with a partition of $\partial\Omega$ in Γ_2, Γ_e.
Find u a solution in such that:

$$-\Delta u = 1 \text{ in } \Omega, \quad u = 2 \text{ on } \Gamma_2, \quad \frac{\partial u}{\partial \vec{n}} = 0 \text{ on } \Gamma_e \tag{1}$$

Denote $V_g = \{v \in H^1(\Omega)/v_{|\Gamma_2} = g\}$.
The Basic variationnal formulation with is: find $u \in V_2(\Omega)$, such that

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

The finite element method is just: replace V_g with a finite element space, and the `FreeFem++` code:

Poisson equation in a fish with FreeFem++

The finite element method is just: replace V_g with a finite element space, and the FreeFem++ code:

```
mesh3 Th("fish-3d.msh");// read a mesh 3d
fespace Vh(Th,P1);    // define the P1 EF space

 Vh u,v;//set test and unknown function in Vh.
macro Grad(u) [dx(u),dy(u),dz(u)] //EOM Grad def
solve laplace(u,v,solver=CG) =
   int3d(Th)(  Grad(u)'*Grad(v)    )
   - int3d(Th) ( 1*v)
   + on(2,u=2);    // BC on Γ2
plot(u,fill=1,wait=1,value=0,wait=1);
```

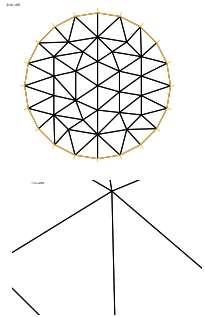
Run:fish.edp Run:fish3d.edp

2. Sequential Academic Examples
   - Weak form
   - Anisotropic Mesh adaptation
   - Bose Einstein Condensate
   - Search all local min
   - Best Fit
   - Incompressible Navier-Stokes
   - Linear elasticty equation
   - Hyper elasticity equation
   - Surface finite element

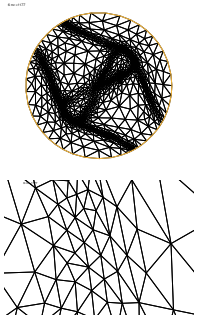
Example of adaptation process

Find optimal mesh in norm L^∞ to represent:

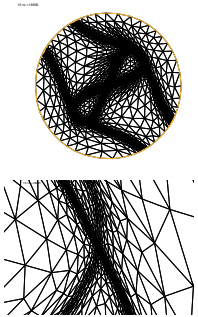
$$u = 10x^3 + y^3 + tanh(50\ (sin(5y) - 2x)\ );$$
$$v = 10y^3 + x^3 + tanh(500(sin(5x) - 2y)\ );$$



Run:Adapt-uv.edp



Run:CornerLap.edp



Run:LaplaceDiracP2.edp

Example of adaptation process in 3d with mmg3

Let a domain $\Omega =]0,1[^3 \backslash [\frac{1}{2},1[^3$ The border of $\partial\Omega$ is split in 2 part

- Γ_2, if $x = 1$, $y = 1$, or $z = 1$
- Γ_1, else.

Find u a solution in such that:

$$\begin{aligned} -\Delta u &= 1 && \text{in } \Omega, \\ \frac{\partial u}{\partial \vec{n}} &= 0 && \text{on } \Gamma_2, \\ u &= 0 && \text{on } \Gamma_1. \end{aligned}$$

Thank to mmg3 to do 3d mesh adaptation.

Run:Laplace-Adapt-3d.edp

Run:Laplace-Adapt-aniso-3d.edp

2. Sequential Academic Examples
   - Weak form
   - Anisotropic Mesh adaptation
   - Bose Einstein Condensate
   - Search all local min
   - Best Fit
   - Incompressible Navier-Stokes
   - Linear elasticty equation
   - Hyper elasticity equation
   - Surface finite element

With. I. Danaila (Univ. Rouen), G. Vergez (Phd Becasim), P-E Emeriau (Stage ENS/2016)

Just a direct use of `Ipopt` interface (2 day of works to start script see, + n month)
The problem is find a complex field u on domain $\mathcal{D}$ such that:

$$u = \underset{||u||=1}{\operatorname{argmin}} \int_{\mathcal{D}} \frac{1}{2}|\nabla u|^2 + V_{trap}|u|^2 + \frac{g}{2}|u|^4 - \Omega i\overline{u}\left(\begin{pmatrix} -y \\ x \end{pmatrix}.\nabla\right)u$$

to code that in FreeFem++
use

- Ipopt interface (`https://projects.coin-or.org/Ipopt`)
- Adaptation de maillage

The idea to mixte Ipopt and adapt mesh is to play with the stop criterion, and finally use freefem++ to analyse the result.

Run:BEC.edp

2. Sequential Academic Examples
   - Weak form
   - Anisotropic Mesh adaptation
   - Bose Einstein Condensate
   - Search all local min
   - Best Fit
   - Incompressible Navier-Stokes
   - Linear elasticty equation
   - Hyper elasticity equation
   - Surface finite element

Search all local min

The function `findalllocalmin` find all the le local min and use a greedy algorithm to to the local attraction zone, by adding the triangle through the minimal vertices.

```
mesh Th=square(50,50,[x*2-1,y*2-1]);
load "isoline"
fespace Vh(Th,P1), Ph(Th,P0);
int k =2;
Vh u= sin(k*pi*x)*sin(k*pi*y);
plot(u, wait=1);
Ph r;
int[int] lm=findalllocalmin(Th,u[],r[]);
// lm array gives the number list of vertex which are local min
// r is function P0 defined the attraction zone of the local min
plot(r,u,fill=1,wait=1);
// to see where is the minimuns
Ph mx= Th(lm[real(r)]).x -x, my= Th(lm[real(r)]).y -y;
plot([mx,my],u,wait=1,fill=0);
```

Run:findalllocalmin.edp

Run:findalllocalminbec.edp

2. Sequential Academic Examples
   - Weak form
   - Anisotropic Mesh adaptation
   - Bose Einstein Condensate
   - Search all local min
   - Best Fit
   - Incompressible Navier-Stokes
   - Linear elasticty equation
   - Hyper elasticity equation
   - Surface finite element

(With. P-E Emeriau) Just use `ipopt` to find the arg min of $J(\alpha) = \int_\Omega (u - \phi_\alpha)^2)$ where $alpha$ is the set of parameters.
On the domain with no vortex, all just do the L2 projection on axisymmetric space with a laplace regularisation

1. Run:fit-axi.edp
2. Run:analyssolbec.edp

To solve $F(u) = 0$ the Newton's algorithm is

1. u^0 a initial guest
2. do
   - find w^n solution of $DF(u^n)w^n = F(u^n) + BC0$
   - $u^{n+1} = u^n - w^n$ or $DF(u^n)u^{n+1} = DF(u^n)u^n - F(u^n) + BC$
   - if($||w^n|| < \varepsilon$) break;

For Navier Stokes problem the algorithm is: $\forall v, q$,

$$F(u,p) = \int_\Omega (u.\nabla)u.v + u.v + \nu\nabla u : \nabla v - q\nabla.u - p\nabla.v + BC$$

$$\begin{aligned} DF(u,p)(w,w_p) &= \int_\Omega (w.\nabla)u.v + (u.\nabla)w.v \\ &\quad + \int_\Omega \nu\nabla w : \nabla v - q\nabla.w - p_w\nabla.v + BC0 \end{aligned}$$

Run:cavityNewton.edp
Run:Rayleigh-Benard-long.edp
Run:NSNewtonCyl-100-mpi.edp
Run:Rayleigh-Benard-time.edp

The generalise Stokes problem is find u, p solution of

$$Au + Bp = f, \quad {}^tBu = 0$$

with $A \equiv (\alpha Id + \nu\Delta)$ and $B \equiv \nabla$. remark,if A est symmetric positive the you can use a conjugate gradient to solve the the following problem

$${}^tBA^{-1}B\,p = {}^tBA^{-1}f$$

Now in a periodic domain, all differential operators commute and the Uzawa algorithm comes to solving the linear operator $-\nabla.((\alpha Id - \nu\Delta)^{-1}\nabla$, where Id is the identity operator. So the preconditioner suggested is $-\alpha\Delta^{-1} + \nu Id$.
the term $\frac{\partial u}{\partial t} + (u \cdot \nabla)u$ is the total derivative and discretization in time

$$\begin{array}{rl} \frac{1}{\tau}(u^{n+1} - u^n \circ X^n) - \nu\Delta u^{n+1} + \nabla p^{n+1} & = 0, \\ \nabla \cdot u^{n+1} & = 0 \end{array} \tag{3}$$

The term $X^n(x) \approx x - \tau u^n(x)$ will be computed with convect operator.
Run:NSUzawaCahouetChabart.edp Run:NSUzawaCahouetChabart-3d-aorte.edp

2. Sequential Academic Examples
   - Weak form
   - Anisotropic Mesh adaptation
   - Bose Einstein Condensate
   - Search all local min
   - Best Fit
   - Incompressible Navier-Stokes
   - Linear elasticty equation
   - Hyper elasticity equation
   - Surface finite element

Linear Lame equation, weak form

Let a domain $\Omega \subset \mathbb{R}^d$ with a partition of $\partial\Omega$ in Γ_d, Γ_n.
Find the displacement $\boldsymbol{u}$ field such that:

$$-\nabla.\sigma(\boldsymbol{u}) = \boldsymbol{f} \text{ in } \Omega, \quad \mathbf{u = 0} \text{ on } \mathbf{\Gamma_d}, \quad \sigma(\mathbf{u}).\mathbf{n = 0} \text{ on } \mathbf{\Gamma_n} \tag{4}$$

Where $\varepsilon(\boldsymbol{u}) = \frac{1}{2}(\nabla\boldsymbol{u} + {}^t\nabla\boldsymbol{u})$ and $\sigma(\boldsymbol{u}) = \boldsymbol{A}\varepsilon(\boldsymbol{u})$ with $\boldsymbol{A}$ the linear positif operator on symmetric $d \times d$ matrix corresponding to the material propriety. Denote
$V_{\boldsymbol{g}} = \{\boldsymbol{v} \in H^1(\Omega)^d / \boldsymbol{v}_{|\Gamma_d} = \boldsymbol{g}\}$.
The Basic displacement variational formulation is: find $\boldsymbol{u} \in V_0(\Omega)$, such that:

$$\int_\Omega \varepsilon(\boldsymbol{v}) : \boldsymbol{A}\varepsilon(\boldsymbol{u}) = \int_\Omega \boldsymbol{v}.\boldsymbol{f} + \int_\Gamma ((\boldsymbol{A}\varepsilon(\boldsymbol{u}))\boldsymbol{n}).v, \quad \forall \boldsymbol{v} \in V_0(\Omega) \tag{5}$$

Run:beam-3d.edp Run:beam-EV-3d.edp Run:free-cyl-3d.edp Run:beam-3d-Adapt.edp

2. Sequential Academic Examples
   - Weak form
   - Anisotropic Mesh adaptation
   - Bose Einstein Condensate
   - Search all local min
   - Best Fit
   - Incompressible Navier-Stokes
   - Linear elasticty equation
   - Hyper elasticity equation
   - Surface finite element

Hyper elasticity equation

The Hyper elasticity problem is the minimization of the energy $W(I_1, I_2, I_3)$ where I_1, I_2, I_3 are the 3 invariants. For example The Ciarlet Geymonat energy model is

$$W = \int_\Omega \kappa_1(J_1 - 3) + \kappa_2(J_2 - 3) + \kappa(J - 1) - \kappa \ln(J)$$

where $J_1 = I_1 I_3^{-\frac{1}{3}}$, $J_2 = I_2 I_3^{-\frac{2}{3}}$, $J = I_3^{\frac{1}{2}}$,
let u the deplacement, when

- $F = I_d + \nabla u$
- $C = {}^tFF$
- $I_1 = \mathrm{tr}(C)$
- $I_2 = \frac{1}{2}(\mathrm{tr}(C)^2 - \mathrm{tr}(C^2))$
- $I_3 = \det(C)$

The problem is find

$$u = \underset{u}{\mathtt{argmin}}\, W(I_1, I_2, I_3)$$

Hyper elasticity equation

```
fespace Wh(Th,[P2,P2]);
// Newton's method
Wh [d1,d2]=[0,0];
Wh [w1,w2],[v1,v2];
for(int i=0;i<Nnewton;++i)
 {
    solve dWW([w1,w2],[v1,v2]) =
          int2d(Th)( ddW2d([d1,d2],[w1,w2],[v1,v2]) )
        - int2d(Th)( dW2d([d1,d2],[v1,v2]) -[v1,v2]'*[f1,f2] )
        + on(1,w1=0,w2=0);

     d1[] -= w1[];
     real err = w1[].linfty;
     if(err< epsNewton) break;
 }
```

Run:Hyper-Elasticity-2d.edp Run:ElasticLaw2d.idp Run:CiarletGemona.idp
Run:HyperElastFH.edp (Hyper Elastycity with contact / IPOPT)

2. Sequential Academic Examples
   - Weak form
   - Anisotropic Mesh adaptation
   - Bose Einstein Condensate
   - Search all local min
   - Best Fit
   - Incompressible Navier-Stokes
   - Linear elasticty equation
   - Hyper elasticity equation
   - Surface finite element

Surface finite element on Tore

With Axel Fourmont (Ing. INRIA)

```
load "msh3"
real R = 3, r=1, h = 0.2; //
int nx = R*2*pi/h, ny = r*2*pi/h;
mesh Th2 = square(nx,ny,[x*pi*2,y*pi*2],region=1);
func torex= (R+r*cos(y))*cos(x);
func torey= (R+r*cos(y))*sin(x);
func torez= r*sin(y);
meshS Th=movemesh2S(Th2,transfo=[torex,torey,torez],orientation=1);
plot(Th);
fespace VhS(Th,P1);
VhS u,v;
macro grad3(u) [dx(u),dy(u),dz(u)]  // EOM
solve Lap(u,v) = int2d(Th)( u*v
       +grad3(u)'*grad3(v)) -int2d(Th)(x*v);
plot(u,wait=1,nbiso=20,fill=1);
```

Run:Lap-on-Tore.edp

Outline

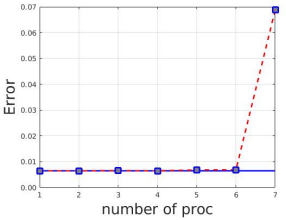
3. Parallel examples
   - Partial fraction decomposition and parallelization in time
   - Domain decomposition: FFDDM/HPDDM

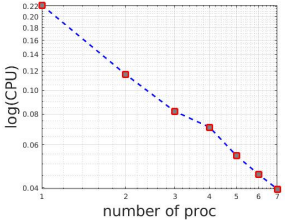
Partial fraction decomposition and parallelization in time

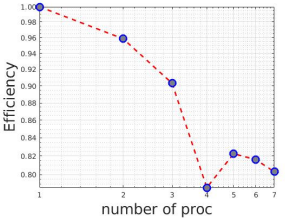
An idea of *S.M. Kaber* with my help. Starting the time discretization of this equation $\boldsymbol{u_t = \mathcal{L}u}$ by implicit Euler scheme with time step h_1 reads $\frac{\boldsymbol{u}^1-\boldsymbol{u}^0}{h_1} = \mathcal{L}\boldsymbol{u}^1$. With a Finite Difference scheme $(I - h_1X)u^1 = u^0$. with X a Finite Difference approximation of the spatial operator $\mathcal{L}$.
Iterating p steps of the implicit scheme, we obtain an equation for u^p
$(I - h_pX)\cdots(I - h_2X)(I - h_1X)u^p = u^0$. And the fractional decompositions of matrice, give

$$(I - h_pX)\cdots(I - h_2X)(I - h_1X) = \sum_{k=1}^{p} \alpha_i(I - h_iX))^{-1}.$$

Consequently, the vector u^p could be split into $u^p = \sum_{i=1}^{p} \alpha_i x_i$.







The $2D$ nonhomogeneous heat equation. Run:Para-rhs.edp
see https://hal.archives-ouvertes.fr/hal-01878765

3. Parallel examples
   - Partial fraction decomposition and parallelization in time
   - Domain decomposition: FFDDM/HPDDM

Domain decomposition: FFDDM/HPDDM

With Pierre-Tournier , P. Jolivet, Frédéric Nataf.

- Documentation FFDDM
- tutorial FFDDM
- Run:DDM-Schwarz-Lap-2dd.edp
- Run:diffusion-3d-minimal-ddm.edp

Some hpddm examples from the distribution

Diffusion Run:diffusion-2d.edp , Run:diffusion-3d.edp

Elastically Run:elasticity-2d.edp , Run:elasticity-3d.edp

Outline

Conclusion/Future

FreeFEM(++) v4 is

- very good tool to solve non standard PDE in 2D/3D and of surface
- to try new domain decomposition domain algorithm

The future we try to do:

- 3d anisotrope mesh adaptation (with new version mmg3d software)
- link with MFront: a code generation tool dedicated to material knowledge (CEA/EDF France)
- automate the parallel tool (see ffddm, in progress P-H Tournier, F. Nataf, P. Jolivet)
- Add integral method (in progress, A. Fourmont, X. Claeys) , and finite volume method (works of G. Sadaka).
- Build more graphic with VTK, paraview , ... (in progress)

Thank for you attention.