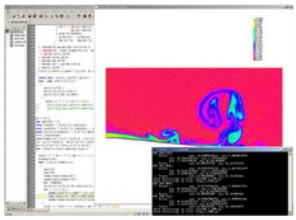
**Numerical solutions to PDEs** PhL 7.2.14

I did some web searching on this subject a while ago, but have no notes in my PDE area. A quick scan did not turn up anything so here I will just start over. There is a hint that Maple might now have a numeric capability with pdsolve, but I don't have that.

Here is the first offering I notice:

# FreeFem++ v 3.30-1 (July 01 2014 19:23:36.)

Introduction



**FreeFem++** is a partial differential equation solver. It has its own language. freefem scripts can solve multiphysics non linear systems in 2D and 3D.

Problems involving PDE (2d, 3d) from several branches of physics such as fluid-structure interactions require interpolations of data on several meshes and their manipulation within one program. FreeFem++ includes a fast 2^d-tree-based interpolation algorithm and a language for the manipulation of data on multiple meshes (as a follow up of bamg (now a part of FreeFem++ ).

FreeFem++ is written in C++ and the FreeFem++ language is a C++ idiom. It runs on Macs, Windows, Unix machines. FreeFem++ replaces the older freefem and freefem+.

If you use Freefem++ please cite the following reference in your work (books, articles, reports, etc.): Hecht, F. New development in FreeFem++. J. Numer. Math. 20 (2012), no. 3-4, 251–265. 65Y15

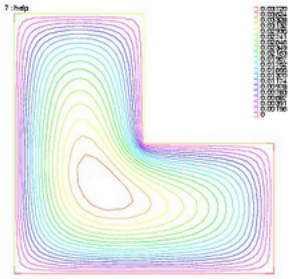
It looks very promising, but I would have to take a lot of time just learning how to make it do something. Here is an example they give

Examples 2d

- A small movie (340Kb) : Cool air (green) comes from the lower left and mix with hot air (magenta), the right boundary is free. This is Navier-Stok Boussinesq integrated with P1-bubble P1 mixte finite element.
- A very small example 2d of how to solve the Poisson equation on a L shape :

```
border aaa(t=0,1){x=t;y=0;};
border bbb(t=0,0.5){x=1;y=t;};
border ccc(t=0,0.5){x=1-t;y=0.5;};
border ddd(t=0.5,1){x=0.5;y=t;};
border eee(t=0.5,1){x=1-t;y=1;};
border fff(t=0,1){x=0;y=1-t;};
mesh Th = buildmesh (aaa(6) + bbb(4) + ccc(4) +ddd(4) + eee(4) + fff(6));
fespace Vh(Th,P1);   //  to change P1 in P2 to make P2 finite element.
Vh u=0,v;
func f= 1;
func g= 0;
int i=0;
real error=0.1, coef= 0.1^(1./5.);
problem Probem1(u,v,solver=CG,eps=-1.0e-6) =
    int2d(Th)(  dx(u)*dx(v) + dy(u)*dy(v))
  + int2d(Th) ( v*f )
  + on(aaa,bbb,ccc,ddd,eee,fff,u=g)  ;

for (i=0;i< 10;i++)
{
  real d = clock();
  Probem1; //  solve the problem
  plot(u,Th,wait=1);
  Th=adaptmesh(Th,u,inquire=1,err=error);
  error = error * coef;
} ;
```



Solution on adapted mes

So I will just put this hold for a while.