An introduction to scientific computing using free software FreeFem++

Ionut Danaila[1] and Frédéric Hecht[2]

[1]Université de Rouen, France

[2]Université Pierre et Marie Curie, Paris

Lesson 2: March 8, 2016.

Outline of this Lesson

Finite element representation

- **Functional spaces**

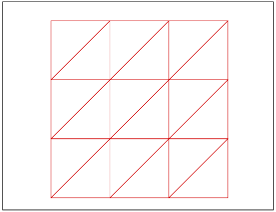
$$
\begin{aligned}
H^1(\Omega) &= \{v \in L^2(\Omega) \;:\; \frac{\partial v}{\partial x}, \frac{\partial v}{\partial y} \in L^2(\Omega)\}\\
V(\Omega) &= \{v \in H^1(\Omega) \;:\; v|_{\Gamma^D} = 0\}.
\end{aligned}
$$

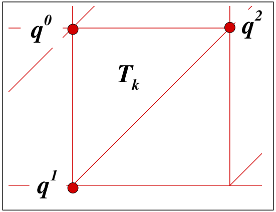
- **Approximation spaces**

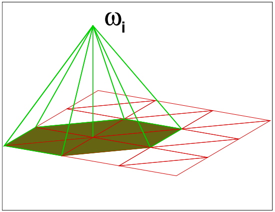
$\mathcal{T}_h$ ::= triangulation,
$\Omega_h = \cup_{k=1}^{n_t} T_k$, ($n_t$ is the number of triangles).
$H_h = \{v \in C^0(\Omega_h) \;:\; \forall T_k \in \mathcal{T}_h, v|_{T_k} \in P^1(T_k)\}$,
$V_h = \{v \in H_h \;:\; v|_{\Gamma_h^D} = 0\}$.

- **Basis functions**

$w^i \in H_h, \quad w^i(q^j) = \delta_{ij}$ (1 if $i = j,$ 0 otherwise).
$\nabla w^i|_{T_k} = const$,
$dim(H_h) = n_v$ (n_v is the number of vertices),
$f_h \in H_h$ ::= array of n_v values.





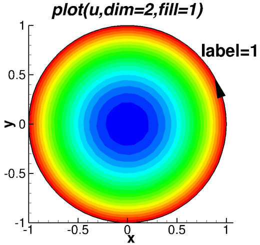


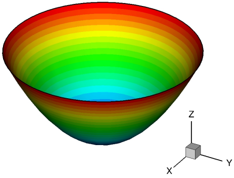
Definition of finite element spaces, integrals

FEspace is a type (as real or int) in FreeFem++

fespace_v01.edp

```
/* P1 representation */
// Parameters
int nbseg=100;
real R=1, xc=0, yc=0;
// border
border circle(t=0,2*pi){label=1;x=xc+R*cos(t);
                                y=yc+R*sin(t);}
// FE mesh
mesh Th = buildmesh(circle(nbseg));plot(Th);
// FE space
fespace Vh(Th, P1);
Vh u=x*x+y*y;  plot(u, dim=3,fill=1,wait=1);
// 2d integral
cout.precision(12);
real int2dU = int2d(Th)(u);
cout <<"int2dU="<<int2dU<<endl;
cout <<" exact="<<pi*R^4/2<<endl;
real int1dU = int1d(Th,1)(u);
cout <<"int1dU="<<int1dU<<endl;
cout <<" exact="<<2*pi*R^3<<endl;
```





plot(u,dim=3,fill=1)

Functions: x, y, z are reserved names for space variables

(part of) fespace_v02.edp

```
func fu=x*x+y*y;
cout.precision(12);
real int2dU = int2d(Th)(fu);
cout <<"int2dU="<<int2dU<<endl;
cout <<" exact="<<pi*R^4/2<<endl;
real int1dU = int1d(Th,1)(fu);
cout <<"int1dU="<<int1dU<<endl;
cout <<" exact="<<2*pi*R^3<<endl;
// other definition of functions
func string infoMesh(mesh & Th)
{
  string message="Mesh size:: nv="+ Th.nv+" nt="+Th.nt;
      return message;
}
cout << infoMesh(Th)<<endl;
func  real Meshsize(mesh & Th)
{
  fespace Ph(Th,P0);
  Ph h=hTriangle;
      return h[].max;
}
cout << Meshsize(Th)<<endl;
```