UNIVERSIDAD DE CONCEPCIÓN
Facultad de Ciencias Físicas y Matemáticas
Departamento de Ingeniería Matemática

LABORATORIO 3 DE ELEMENTOS FINITOS (521537)

4. Uso de mallas externas. Es frecuente que la malla que se usa para discretizar el dominio de un problema variacional provenga de programas externos. `FreeFem++` puede leer varios formatos de mallas, entre los cuales se encuentran las mallas con extensión `.nopo` de `MODULEF` y varios formatos basados en texto.

Nuestro primer objetivo será la importación de una malla generada en `MATLAB`, que deber ser transformada para poder ser ocupada por `FreeFem++`.

En el archivo `lab03a.tar` se encuentra el directorio `mallaMatlab`, el directorio `mallaTriangle` y el archivo `extmesh.edp`. Para extraer sus contenidos se escribe, en un directorio de trabajo,

```
tar -xvf lab03.tar
```

En esta parte les pido que lleven a cabo las siguientes tareas:

1. Extraigan los contenidos del archivo `lab03a.tar`.

2.a Si `MATLAB` estuviera disponible, corran en él el programa `pet2msh.m` del directorio `mallaMatlab`. Este programa transforma la malla contenida en `sudamerica.mat` en la misma malla, pero ahora contenida en `sudamerica.msh`, en el directorio de trabajo, y con formado `.msh`.

2.b Si `MATLAB` no está disponible, extraigan `sudamerica.msh` desde `lab03b.tar`.

pet2msh.edp

```
load sudamerica.mat;

p=p'; t=t'; e=e';
e=e(:,[1 2]);

p = [p zeros(size(p,1),1)];
p(unique(e),3)=1;

f = fopen('../sudamerica.msh','w');

fprintf(f,'%d %d %d\n',size(p,1),size(t,1),size(e,1));
for k=1:size(p,1)
        fprintf(f,'%4.5f %4.5f %d\n',p(k,:));
end
for k=1:size(t,1)
        fprintf(f,'%d %d %d %d\n',[t(k,1:3) 0]);
end
for k=1:size(e,1)
        fprintf(f,'%d %d %d\n',[e(k,:) 1]);
end

fclose(f);
```

3. De manera análoga al caso anterior, transformen la malla en formato `TRIANGLE`, guardada en tres archivos de extensiones `.node`, `.ele` y `.poly` a un formato legible por `FreeFem++`, usando cualquier lenguaje que quieran y tengan disponible, o bien, copiando y pegando en forma adecuada.

5. Uso de una malla importada y exportación de matrices. Las mallas importadas se usan con el comando `readmesh`. Si el formato usado no guarda los nombres de los contornos, éstos deben ser referenciados con números.

En esta parte, además de hacer uso de mallas importadas, exportaremos arreglos matriciales y vectoriales para usarlos en programas externos.

En esta parte les pido que realicen las siguientes tareas:

EXTMESH.EDP

```
mesh Th = readmesh("sudamerica.msh");

fespace Vh(Th,P1);

int n = Vh.ndof;

int [int] Ind(n);
Ind  = 0;

varf a(u,v) = int2d(Th) (dx(u)*dx(v) + dy(u)*dy(v));
varf rhs(unused,v) = int2d(Th) (-4.0*v);

matrix sparseA = a(Vh,Vh);
real [int] f(n);
f = rhs(0,Vh);

for(int i=0;i<n;i++)
        for(int j=0;j<3;j++)
                if(Th[i][j].label)
                        Ind(Th[i][j]) = 1;



{
   ofstream fid("sparseA.dat");
   fid << sparseA;
}
{  ofstream fid("rhs.dat");
    for(int i=0;i<n;i++)
            fid << f[i] << " ";
}
{
   ofstream fid("bc.dat");
   for(int i=0;i<n;i++)
           fid << Ind[i] << " ";
}
```

1. Corran el *script* `extmesh.edp`. Se exportarán los archivos `sparseA.dat`, `rhs.dat` y `bc.dat`, conteniendo la matriz de rigidez, el vector del lado derecho y un indicador de pertenencia de un nodo a la frontera, respectivamente, para el problema $-\Delta u = f$.

2. Extraigan los contenidos del archivo `lab03c.tar`, a saber, `leerYresolver.m` y `extsol.edp`.

3.a Si `MATLAB` estuviera disponible, corran en él el programa `leerYresolver.m`. Este programa lee la matriz de rigidez, el vector del lado derecho y el indicador de pertenencia de un nodo a la frontera y resuelve el sistema lineal previa realización de un *levantamiento* para satisfacer la condición de contorno Dirichlet que consiste en que u sea idénticamente 1 en la frontera del dominio. Se exportará el archivo `u.dat` con la solución en los nodos.

3.b Si `MATLAB` no está disponible, extraigan `u.dat` desde `lab03d.tar`.

4. Corran el *script* `extsol.edp`. Se importará la solución `u.dat` para visualizarla.

5. El problema consistente en hallar $(\alpha, u) \in H_0^1(\Omega) \times \mathbb{R}$ de manera tal que

$$\begin{aligned} -\Delta u &= \alpha u \quad && \text{en } \Omega \\ u &= 0 && \text{en } \partial\Omega \end{aligned}$$

se conoce como el *problema de los autovalores del laplaciano*. Se sabe que testeando primera ecuación de la manera usual y usando el método de Galerkin este problema se reduce a un *problema de autovalores generalizado* para matrices, de la forma

$$A\vec{u} = \hat{\alpha} B\vec{u}.$$

Este problema se puede resolver en `MATLAB` mediante el comando `[u,alpha] = eigs(A,B,n,'sm')`, que hallará los `n` autovalores más pequeños y sus respectivos autovalores.
Adapte los programas anteriores para hallar las primeras tres autofunciones del laplaciano en el mismo dominio que los programas anteriores.

EXTSOL.EDP

```
mesh Th = readmesh("sudamerica.msh");

fespace Vh(Th,P1);
Vh u;

int n = Vh.ndof;

{
   ifstream fid("u.dat");
   for(int i=0;i<n;i++)
        fid >> u[][i];
}

plot(u,wait=true);
```

leerYresolver.m

```
function leerYResolver

flag = 1;
aux = 0;
aux2=[0 0 0];
load bc.dat;
load rhs.dat;

f = fopen('sparseA.dat');

while flag==1
        a = fgetl(f);
        if a(1)~='#'
                flag=0;
        end
end

[m,n,aux,nnz] = strread(a,'%d %d %d %d');

matA = sparse(m,n);

for k=1:nnz
        aux2=fscanf(f,'%f %f %f',3);
        matA(aux2(1),aux2(2)) = aux2(3);
end

fclose(f);

bndrs = find(bc==1);
intrs = find(bc==0);

u = zeros(m,1);

%Uso condicion de Dirichlet identicamente 1 en la frontera y
%lado derecho cero.
u(bndrs) = 1.0;
u(intrs) = matA(intrs,intrs)\(rhs(intrs)' -matA(intrs,bndrs)*u(bndrs) );

save u.dat u -ascii;
```

Por Leonardo Figueroa Candia
Abril de 2007
lfiguero@ing-mat.udec.cl