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

LABORATORIO 1 DE ELEMENTOS FINITOS

En este laboratorio veremos algunos de los comandos básicos del programa `FreeFem++ 2.x`. En `http://www.freefem.org/ff++/` están las instrucciones para bajarlo (alrededor de 10 MB) e instalarlo en Linux, Windows y MacOS X.

`FreeFem++ 2.x` es un programa que interpreta y ejecuta *scripts* de texto, a los que típicamente se les pone la extensión `.edp`. En este laboratorio vamos a conocer la manera de trabajar con este programa y exploraremos algunos de sus comandos básicos a través de la resolución del problema de Poisson en el plano.

1. Problema de Poisson en el plano. Consideremos el problema de valores de contorno en dos dimensiones

$$\begin{aligned} -\Delta u &= f \quad \text{en} \quad \Omega \\ u &= 0 \quad \text{en} \quad \partial\Omega \end{aligned} \tag{1}$$

donde Ω es la circunferencia unitaria

$$\Omega := \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 < 1\}$$

y $f \in L^2(\Omega)$ es

$$f(x, y) = 1.$$

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

1. Copien el código que aparece en el recuadro de la página 3 a un archivo de texto llamado `lapp1.edp`.
2. Ejecuten el *script* a través del comando `FreeFem++ lapp1.edp`.

Si todo anduvo bien debieran aparecer, en forma sucesiva, tres gráficos. Uno con la malla generada, otro con la solución aproximada y otro con una proyección de la solución verdadera.

Las tareas siguientes implican alterar el *script* `lapp1.edp` en distintos detalles para ilustrar qué rol cumple y cómo se usa cada comando.

3. Guarden estos cambios en un archivo con otro nombre y ejecuten:
   - Después de la línea que comienza con `border` agreguen `border D(t=2*pi,0){ x=0.5*cos(t); y=0.5*sin(t);}`
   - En la línea que comienza con `mesh` reemplacen el argumento de la función `buildmesh` por `C(50) + D(50)`.
   - En la línea que comienza con `- int2d` "resten" `int1d(Th,D)(0.25*v)`.

4. De nuevo a partir de `lapp1.edp` guarden los siguientes cambios en un archivo con otro nombre y ejecuten:

- Justo antes de la línea que empieza con `mesh` agreguen la línea `for (int i=50;i<=400;i=i*2) {`.
- Reemplacen `buildmesh(C(50))` por `buildmesh(C(i))`.
- Pongan el signo de comentario (`//`) antes de cada comando `plot` (o borren los comandos `plot`).
- Después de la última línea cierren el bucle `for` agregando la línea `};`.

5. A partir del programa escrito en la parte anterior alteren las funciones `f` y `sol` y el valor en el contorno (en el comando `on`) para el caso en que la solución es la función $u(x, y) = \sqrt{x^2 + y^2}$. Guarden con otro nombre. Comparen la tasa de caída del error en $L^2(\Omega)$ de este caso y el anterior.

```
//Descripcion del contorno
border C(t=0,2*pi){ x=cos(t); y=sin(t);}

//Generacion de una malla con 50 puntos en el contorno
mesh Th = buildmesh(C(50));

//Grafico de la malla
plot(Th,wait=true);

//Descripcion de un espacio de elementos finitos
fespace Vh(Th,P1);

//Definicion de u y v como miembros de P1 continuo
Vh u,v,solh;

//Funcion de lado derecho
func f = 1.0;
func sol = (1.0 - x^2 - y^2)/4.0;

//Que hora es?
real cpu=clock();

//Asi no mas:
problem Poisson(u,v,solver=LU) =
        int2d(Th)(dx(u)*dx(v) + dy(u)*dy(v))
        - int2d(Th)(f*v) + on(C,u=0);

//Resolver
Poisson;

//Grafico
plot(u,wait=true,value=true);

//Proyecto la solucion verdadera al espacio discreto
//usado para graficarlo con plot
solh=sol;

plot(solh,wait=true,value=true);

//Tiempo ocupado
cout << "Tiempo de CPU: " << clock()-cpu << endl;
cout << "Error en L^2: " << sqrt(int2d(Th)((u-sol)^2)) <<endl;
```

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