Analisi Numerica e Software Scientifico, a.a.2014-2015

Laboratorio del 23/04/2015

Prima di cominciare: scarica dal sito

<http://www.math.mtu.edu/~msgocken/fembook/index.html>

ed installa il file (Gzipped tar file) delle funzioni da utilizzare, corrispondenti alle funzioni descritte in *Understanding and Implementing the Finite Element Method*, M. S. Gockenbach.

Inserisci nel path di matlab la directory delle funzioni, ed in particolare le cartelle “version1” e “version2”.

1. È dato il problema ai valori al bordo

$$-\nabla \cdot (\kappa \nabla u) = f, \quad (x, y) \in \Omega, \quad u(x, y) = g(x, y) \quad \text{per } (x, y) \in \partial\Omega$$

dove $\kappa = 1$ ed Ω è un dominio poligonale. Il forzante è definito come $f(x, y) = 12xy$, in modo che la soluzione esatta sia $u(x, y) = xy(1 - x^2 - y^2)$, da cui si possono determinare i valori della funzione g definita sul bordo. Le info per una mesh rada di questo dominio sono scaricabili dal nostro sito (lettura in Matlab: `load mesh_basic.mat`).

- Crea e visualizza la mesh ed il supporto per uno dei nodi:

```
T=MakeMesh1(p',t(1:3,:),'Dirichlet');  
ShowMesh1(T)
```

Visualizza T da prompt e nota che T è una *struttura* (fai `help struct`)

- Crea la funzione `fun_k.m` che definisce il coefficiente $\kappa(x, y) = 1$, e poi crea la matrice di Stiffness:

```
K=Stiffness1(T,'fun_k');
```

- Crea il *vettore* `g_bc` che contiene i valori della funzione $g = u|_{\partial\Omega}$. Le coordinate dei nodi di bordo si possono ottenere con:

```
xx=T.Nodes(T.CNodePtrs,1);  
yy=T.Nodes(T.CNodePtrs,2);
```

- Crea la funzione che definisce $f(x, y)$, cioè `fun_f.m`, e determina il termine noto F :

```
F=Load1(T,'fun_f','fun_k',g_bc,[]);
```

- Risolvi $Ku = F$ e mostra il grafico della soluzione:

```
ShowPWLInFcn1(T,u,g_bc)
```

- Raffina due volte la mesh: (usa due volte: `T=Refine1(T);`) e visualizza la nuova mesh

- Determina la soluzione u sulla nuova mesh e fai il grafico (come per la prima griglia)

- Riparti dalla griglia originale, ed effettua 2 raffinamenti successivi. Per ognuno di questi:

Calcola e mostra il parametro di griglia h (funzione `gethT.m` nel nostro sito)

- Calcola l'errore in norma energia ed in norma L_2 , mediante le funzioni

```
EnergyNormErr1, L2NormErr1,
```

(fai `help` della funzione) fornendo una tabella come la seguente :

h	$\ \nabla(u - u_h)\ $	$\ u - u_h\ $
3.3514e-01	...	...
1.6757e-01	...	...
8.3784e-02	...	...

Commenta sull'ordine di convergenza nelle due norme.

2. Considera lo stesso problema precedente.

- Crea la struttura T con più nodi per approssimazione quadratica: `T2=GenLagrangeMesh2(T,2);`

- Crea la matrice di stiffness ed il termine noto usando polinomi a tratti quadratici (`Stiffness2, Load2`). Confronta dimensione e sparsità di K rispetto ai casi precedenti, per un h confrontabile.

- Confronta il grafico della soluzione esatta con quella approssimata u_h : funzione `ShowPWPolyFcn2`
- Calcola l'errore in norma energia ed in norma L_2 , mediante le funzioni `EnergyNormErr2`, `L2NormErr2`, e confrontalo con quelli con elementi finiti lineari.

3. Sotto opportune ipotesi, lo studio della filtrazione dell'acqua in un mezzo poroso che occupi una regione $\Omega \in \mathbb{R}^2$ può essere ricondotto alla risoluzione del problema $-\kappa(x, y)u_{xx} - \kappa(x, y)u_{yy} = 0$, dove κ è la conduttività idraulica del mezzo poroso ed u rappresenta la variazione del livello dell'acqua nel mezzo. Considera il caso in cui

$$\kappa(x, y) = \begin{cases} 1 & \text{se } x^2 + y^2 - 0.08 > 0 \\ 101 & \text{altrimenti} \end{cases}$$

e $\Omega \approx [-1, 1] \times [-0.8, 0.8]$ corrispondente alla griglia `mesh.Rectangle` nel nostro sito. Le condizioni al bordo sono date da $g(x, y) = 10$ per $y < 0$ sul bordo, e $g(x, y) = 0$ altrimenti.

- Risolvi il problema usando i comandi visti in precedenza.
- Confronta graficamente con la soluzione che si otterrebbe per $\kappa \equiv 1$ in Ω e commenta.