

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

Laboratorio del 04/05/2015

Scarica dal sito del corso ed installa `ifiss3.2.tar.gz`

(sito originale: IFISS v.3.2, <http://www.maths.manchester.ac.uk/~djs/ifiss/>, D. Silvester, H. Elman e A. Ramage.)

Aggiungi la cartella con tutte le sue sottocartelle nel path di Matlab

oppure, Esegui i seguenti comandi da matlab:

- In Matlab vai nella cartella dove è stato installato IFISS

- Da prompt di Matlab, digita `pwd` (e invio)

- Apri il file `gohome.m` ed inserisci l'indirizzo della cartella apparso con `pwd`, al posto di quello corrente

- Da prompt di Matlab, digita `setpath` (e invio)

Esercizio 1.

È dato il problema $-\varepsilon \nabla^2 u + \beta \cdot \nabla u = 0$ in $[-1, 1]^2$, con $\beta = (0, 1)$. È nota la soluzione analitica. Le condizioni al bordo sono di Dirichlet, $u(x, -1) = x$, $u(x, 1) = 0$, $u(-1, y) \approx -1$ e $u(1, y) \approx 1$.

Questo corrisponde al problema n.1 di convection-diffusion di IFISS ("constant vertical wind").

1. Lancia `cd_testproblem` e scegli il problema n.1. Completa il run con tutti i valori di default. Osserva il grafico della soluzione (in alto a destra della figura) e commenta. Segna il *maximum element Peclet number* e la dimensione della matrice: `n=size(Asupg,1)`.
2. Rilancia `cd_testproblem`(problema n.1) per i seguenti valori dei parametri in tabella, annotando il *maximum element Peclet number*, l'errore massimo globale (*estimated global error (in energy)*) e la dimensione della matrice. Commenta i risultati.

#	grid parameter	viscosity	SUPG parameter	Pe_h	a-post error	dim(Asupg)
default	3	1/200	default			
1	3	1/50	0			
2	3	1/200	0			
3	5	1/200	0			
4	3	1/400	default			
5	5	1/400	default			

Esercizio 2.

È dato il problema $-\varepsilon \nabla^2 u + \beta \cdot \nabla u = 0$ in $[-1, 1]^2$ con $\beta = (2y(1 - x^2), -2x(1 - y^2))$ (vale $b \cdot n = 0$ su tutto il bordo). Le condizioni al bordo sono di Dirichlet.

Questo corrisponde al problema n.4 di convection-diffusion di IFISS ("Recirculating wind").

Fai girare `cd_testproblem` con la scelta 4, con **parametro di griglia = 8**, e tutti gli altri di default. Al termine, il run ha creato i dati per la risoluzione del sistema lineare `Asupg x = fsupg`:

- Risolvi il sistema `Asupg x = fsupg` mediante GMRES, includendo la chiamata a `gmres` in un `tic-toc`:

```
>> [X, FLAG, RELRES, ITER, RESVEC] = gmres(Asupg, fsupg, [], 1e-6, 100);
```

(fai `help gmres` per richiamare il significato delle quantità in uscita)

Visualizza la storia della convergenza mediante il plot del vettore `RESVEC` (norma del residuo ad ogni iterazione)

- Risolvi il sistema con un preconditionatore `[L,U]=luinc(Asupg,tol)` (o `ilu`), per due diverse scelte della tolleranza: `tol = 10-1, 10-2`:

```
>> [X, FLAG, RELRES, ITER, RESVEC] = gmres(Asupg, fsupg, [], 1e-6, 100, L, U);
```

Visualizza la storia del residuo sullo stesso grafico del punto precedente.

- Usa AMG come preconditionatore con (i comandi sono come per PCG):

```
>> global amg_grid smoother_params amg_smoother
```

```
>> amg_grid = amg_grids_setup(Asupg);
```

```
>> smoother_params = amg_smoother_params(amg_grid, 'PDJ');
```

```
>> amg_smoother = amg_smoother_setup(amg_grid, smoother_params);
```

```
>> [X, FLAG, RELRES, ITER, RESVEC] = gmres(Asupg, fsupg, [], 1e-6, 100, @(x) amg_v_cycle(x, amg_grid, amg_smoother))
```

Riporta la storia del residuo nel grafico precedente.