Numerical Linear Algebra, PhD Course 2021

Computer Laboratory, 17/12/2021

Step 0. (to be used later).
Download IFISS from by D. Silvester, H. Elman and A. Ramage:

`https://personalpages.manchester.ac.uk/staff/david.silvester/ifiss/download.html`

Add its folder with all its subfolders to Matlab path, or, type the following matlab commands:
- In Matlab go to the IFISS folder
- From the Matlab prompt, type `pwd` (and Enter)
- Open the file `gohome.m` and modify the address with the address shown in the command window
- From the Matlab prompt, type `setpath` (and Enter)

Step 1. FE discretization using PDEToolbox di Matlab.
Open the Toolbox by typing: `pdetool` within Matlab.

1. **Es.1. Poisson equation.**
   (1) In "Option"-"Specification" choose `Generic Scalar` as problem. In "PDE"-"PDE Specifications" choose `Elliptic problem`. Define a polygonal domain (e.g., using the polygon button), and a grid refinement.
   Note: By clicking on the multiple triangle, the grid is refined and more triangles/nodes appear.
   (2) Export `Mesh`, `PDE` coeffs. and `Boundary` data
   (3) Assemble PDE: type `[K,F]=assempde(b,p,e,t,c,a,f);`
   This gives the data for solving the system $Ku = F$
   The problem size depends on the grid refinement. We are interested in analyzing the performance of iterative methods by refining the mesh.
   (4) For each dimension, use the data `K, F` to run CG with different choices of accelerations:
   – No preconditioning:
   `tic;[X,FLAG,RELRES,ITER,RESVEC] = pcg(K,F,1e-8,1000);timeCG=toc;`
   – Incomplete Cholesky preconditioning:
   `tic;[X,FLAG,RELRES,ITER,RESVEC] = pcg(K,F,1e-8,1000,R',R);timeCGP1=toc;`
   where `R` is obtained by using the incomplete Cholesky Matlab function `ichol`) (type `help ichol` to see several options).
   – AMG:
   `tic;[X,FLAG,RELRES,ITER,RESVEC] =`
   `pcg(K,F,1e-8,1000,@(x)amg_v_cycle(x,amg_grid,amg_smoother));timeCG P2=toc`
   where AMG is obtained as follows:

```
global amg_grid smoother_params amg_smoother
>> amg_grid = amg_grids_setup(K);
>> smoother_params = amg_smoother_params(amg_grid, 'PDJ');
>> amg_smoother = amg_smoother_setup(amg_grid, smoother_params);
```

   For each dimension complete the following table

pb size	CG no prec		CG Ichol		CG AMG	
	#its	time	#its	time	#its	time
...	...	...	...	...	...	...
...	...	...	...	...	...	...
...	...	...	...	...	...	...
...	...	...	...	...	...	...
...	...	...	...	...	...	...
...	...	...	...	...	...	...

   Comment on the results.

2. **Nonsymmetric system.**
(We abandon PDEToolbox). Run `cd_testproblem` (IFISS) for Problem 4, with **grid parameter = 8**, and all other default parameters. The run provides the system data `Asupg, fsupg`.
This corresponds to the following convection-diffusion problem $-\varepsilon\nabla^2 u + \boldsymbol{\beta}\cdot\nabla u = 0$ in $[-1,1]^2$ with $\boldsymbol{\beta} = (2y(1-x^2), -2x(1-y^2))$ ("Recirculating wind"), with $\boldsymbol{\beta}\cdot n = 0$ on the whole boundary. Dirichlet boundary conditions are considered.
As before, compare the use of GMRES in the following settings.

   - No preconditioning
```
>> tic;[X,FLAG,RELRES,ITER,RESVEC] = gmres(Asupg, fsupg, [], 1e-6, 100);
                                timeGMRES=toc;
```
   - LU-type preconditioning. Modify the call so as to include the preconditioning matrices L, U obtained as `[L,U]=ilu(Asupg,setup)` (type help to explore different options), for `setup.droptol` $= 10^{-1}, 10^{-2}$.
   - AMG preconditioning (the same as for PCG)

   Record CPU time, number of iterations for the various choices, including more options obtained by modifying the ILU call. Try also using **restarted** GMRES with/without preconditioning.

3. **Saddle point problem (advanced)**
We consider the linear system
$$\begin{bmatrix} A & B^T \\ B & -\beta C \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}$$
In IFISS, run `stokes_testproblem` for Problem 2, using different meshsizes and Q2-Q1 elements (choice 3). The full run provides the data `Ast, Bst, fst, gst, C, beta, Q` with beta=0.
   - Run the iterative solver MINRES for symmetric and indefinite problems, without preconditioning (check `help minres` to visualize the call)
   - Run MINRES with preconditioner $\mathcal{P} = \text{blkdiag}(\widetilde{A}, Q)$, where $\widetilde{A}$ is an approximation of `Ast` obtained with incomplete Cholesky (pass as input the factors of $\widetilde{A}$ and `Q`).
   - Run MINRES with preconditioner $\mathcal{P} = \text{blkdiag}(\widetilde{A}, Q)$, where $\widetilde{A}$ is a call to AMG (create a function as a handle for MINRES that applies both preconditioning blocks).