FINITE ELEMENTS AND FAST SOLVERS COMPUTATIONAL EXERCISES *

§1. The Stokes Equations. Running the driver `stokes_testproblem` automatically sets up the data files `specific_flow` and `specific_bc` associated with Examples 5.1–5.4 in Chapter 5 of [ESW]. The function `it_solve` uses the built-in MATLAB routine `minres` and offers the possibility of a diagonal, block diagonal or a multigrid V-cycle preconditioning the $(1,1)$ block, together with a diagonal matrix preconditioning the $(2,2)$ block.

1.1. This exercise explores Example 5.1 in [ESW]. This problem represents steady horizontal flow in a channel driven by a pressure difference between the two ends, or *Poiseuille flow*. Here a solution is computed numerically on $\Omega_\square$ using the velocity $\vec{u} = (1 - y^2,\ 0)$ to define a Dirichlet condition on the inflow boundary $x = -1$. Take the default Neumann outflow condition and run `stokes_testproblem` to compute the blocks `Ast`, `Bst` and `C` for the Stokes matrix `K = [Ast,Bst';Bst,-(1/4)*C]` using stabilized $\boldsymbol{Q}_1$–$\boldsymbol{P}_0$ approximation on a 4×4 grid. Then, compute the eigenvalues of `K` using `eig(full(K))`, and check that the number of negative and zero eigenvalues agree with the dimension of the pressure space. Compare the number of unit and positive eigenvalues with the number of boundary velocity nodes and the number of interior velocity nodes, respectively. Finally, repeat the experiment with a Dirichlet outflow condition.

1.2. This is an important exercise. It explores the issue of local mass conservation using the step flow problem in Example 5.2 in [ESW]. The function `flowvolume` postprocesses a flow solution and computes the volume of fluid crossing a specified vertical grid line. Take the default 16×48 grid and use `flowvolume` to generate horizontal velocity profiles at $x = -1$, $x = 0$ and $x = 5$ for the default stabilized $\boldsymbol{Q}_1$–$\boldsymbol{P}_0$ and $\boldsymbol{Q}_1$–$\boldsymbol{Q}_1$ methods. Compare the results with those obtained using the higher order $\boldsymbol{Q}_2$–$\boldsymbol{P}_{-1}$ and $\boldsymbol{Q}_2$–$\boldsymbol{Q}_1$ methods.

1.3. This exercise explores Example 5.3 in [ESW]. This is a classical test problem used in fluid dynamics, known as *driven-cavity flow*. It is a model of the flow in a square cavity (the domain is $\Omega_\square$) with the lid moving from left to right. Consider solving the default regularized driven cavity flow problem in Example 5.3 using $\boldsymbol{Q}_1$–$\boldsymbol{P}_0$ approximation without stabilisation using a uniform 32×32 grid. The effect on the computed pressure solution will be self-evident, but the effect on iterative convergence can be observed by then running `it_solve` using the default AMG preconditioner choice. Contrast the iteration counts with those obtained using the same preconditioner in the case of stabilized $\boldsymbol{Q}_1$–$\boldsymbol{P}_0$ approximation.

1.4. Consider solving the regularized driven cavity flow problem in Example 5.3 using stabilized $\boldsymbol{Q}_1$–$\boldsymbol{Q}_1$ approximation. Select a stretched 32×32 grid and run `it_solve` with algebraic multigrid preconditioning and Jacobi smoothing. Compare the convergence profile with that obtained for a uniform 32×32 grid and the default stretched 64×64 grid. Then, repeat the exercise using Gauss-Seidel smoothing instead of Jacobi.

§2. The Navier-Stokes Equations. Running `navier_testproblem` automatically sets up the data files `specific_flow` and `specific_bc` associated with Examples 7.1–7.4 in Chapter 7 of [ESW]. The function `it_solve` offers the possibility of using preconditioned GMRES, BICGSTAB or IDR as a solver. Both ideal and inexact–AMG pressure convection-diffusion and least squares commutator preconditioners are provided.

2.1. Consider solving the Poiseuille flow problem in Example 7.1 Verify that the *exact* solution is computed using $\boldsymbol{Q}_2$–$\boldsymbol{Q}_1$ approximation on the default 16×16 grid and the hybrid nonlinear iteration strategy that is built into IFISS. Rerun the same problem using a uniform 32×32 grid for a range of viscosity parameters ν. What happens in the limit $\nu \to 0$? Repeat this experiment using the stabilized $\boldsymbol{Q}_1$–$\boldsymbol{P}_0$ method instead. Can you explain the difference in the observed behavior?

2.2. This is a fundamental modeling exercise. It explores the natural outflow condition in Example 7.2 in [ESW]. The default problem has the outflow at $x = 5$. Consider solving the problem on a longer domain with the outflow at $x = 10$. Take the locally mass conserving $\boldsymbol{Q}_2$–$\boldsymbol{P}_{-1}$ approximation method, and the default uniform grid. Then, use `flowvolume` to compare the computed horizontal velocity profiles at $x = 5$ for increasing Reynold numbers (specifically; $\nu = 1/50$, $\nu = 1/125$ and $\nu = 1/200$) on the original and the extended domain.

2.3. Compare GMRES iteration counts and CPU timings for the ideal and AMG–iterated pressure convection-diffusion preconditioners by solving the driven cavity problem in Example 7.3 with $\nu = 1/100$ on a sequence of three uniform grids. Take $\boldsymbol{Q}_2$–$\boldsymbol{Q}_1$ and $\boldsymbol{Q}_1$–$\boldsymbol{P}_0$ mixed approximation and compare the results.

2.4. Compare the effectiveness of the default AMG–iterated pressure convection-diffusion, modified pressure convection-diffusion and least-squares commutator preconditioners for solving systems associated with the step flow problem in Example 7.2. Take the stable $\boldsymbol{Q}_2$–$\boldsymbol{Q}_1$ mixed approximation method using the level **5** grid with the outflow at $x = 10$ and set up an Oseen problem by running five Picard steps. Consider three cases: $\nu = 1/50$, $\nu = 1/125$ and $\nu = 1/200$.

§3. Unsteady Navier-Stokes Flow. Running `unsteady_navier_testproblem` sets up the data files `specific_flow` and `specific_bc` associated with the reference unsteady flow problems. The function `snapshot_solve` offers the possibility of using preconditioned GMRES as a solver for the linearized system that must be solved at a given (snapshot) time.

3.1. Assess the effectiveness of the pressure convection-diffusion and least-squares commutator preconditioners for solving snapshot systems that arise at times $t = 5$ and $t = 90$ when solving the unsteady cavity flow problem with $\nu = 1/1000$. Take a 64×64 stretched grid with $\boldsymbol{Q}_2$–$\boldsymbol{Q}_1$ approximation. Run the function `square_flowmovie` and verify that the solution converges to the *steady-state* solution that is generated by running `solve_navier`. Compare `snapshot_solve` GMRES iteration counts with those obtained when using `it_solve` to solve the system that arises at the final step of Newton iteration.

3.2. Consider the problem of unsteady flow over a square obstacle with the grid parameter set to **5** using $\boldsymbol{Q}_2$–$\boldsymbol{P}_{-1}$ approximation, with all other parameters set to default values. Verify that the solution converges to the *steady-state* solution by running `solve_obstacle_navier` and plotting the solution via:

```
>> obstacle_unsteadyflowplot(qmethod,mv,xns(1:2*nv),inf,By,Bx, ...
      G(1:nv,1:nv),xy,xyp,x,y,bound,bndxy,bnde,obs,0.1,198);
```

What happens when you repeat this experiment: this time running the driver function `unsteady_obstacle_navier` with $\nu = 1/400$?