Numerical Solutions to PDEs - 2020

Problem Set #8

Due in class

Describe the setup and each step in your solutions with words and clearly label your final answers. Use Matlab for plotting and programming and include your code as an appendix to your problem set.

A classic test case for Navier-Stokes solvers is the lid-driven cavity flow problem. This problem is famous for its corner vortices and has been studied in the lab and numerically by dozens of researchers, for both laminar and turbulent flows in 2D and 3D. Physically this problem can be thought of as idealized flow in a wind-driven lake. Numerically, the choice of Dirichlet boundary conditions on all boundaries is very convenient.

Consider the lid-driven cavity problem in two dimensions:

$$\begin{aligned}
\frac{\partial u}{\partial t} + \frac{\partial uu}{\partial x} + \frac{\partial vu}{\partial y} &= -\frac{1}{\rho}\frac{\partial P}{\partial x} + \nu(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}) \\
\frac{\partial v}{\partial t} + \frac{\partial uv}{\partial x} + \frac{\partial vv}{\partial y} &= -\frac{1}{\rho}\frac{\partial P}{\partial y} + \nu(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}) \\
\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} &= 0
\end{aligned}$$

with domain size $x = 0$ to 1 and $y = 0$ to 1. Velocities are zero on all four boundaries except for $u = 1$ at $y = 1$.

You now have the numerical tools to write your own solver for the Navier-Stokes equations for this problem, but it would be a bit time consuming. So you decide to download a code from the web (many thanks to Benjamin Seibold at MIT! See navier.m and other files on the course website), but you still have to understand what it does and test it carefully to see if it is adequate for your purposes.

1. The code uses a uniform, rectangular grid, with staggered arrangement. The velocities are defined normal to the cell faces and the pressure is defined at the cell center.

   (a) Sketch the grid and show where and how boundary conditions are applied.

   (b) Explain why you do not need boundary conditions for the pressure.

2. The algorithm is a fractional step method, with 2nd-order central differences in space, outlined as follows:

- Add nonlinear (advective) terms. This is done with a first order explicit scheme. For numerical reasons, one adds $\partial u^2/\partial x + \partial uv/\partial y$ instead of $u\partial u/\partial x + v\partial u/\partial y$ (even though these are equal analytically due to incompressibility), and analogously for the v equation.
- Add diffusion terms. This is done with a first-order implicit method, to avoid the diffusive time step limit that would be present with an explicit scheme. Hence, a linear system for u and one for v must be solved at each time step. [The code uses an efficient LU-factorization method to solve the systems of equations which works well for grid sizes up to about 120×120.]
- Add the pressure term. The code uses a projection method for pressure, i.e. the first two steps above neglect the pressure (yielding a velocity field that violates incompressibility), so the velocity field is corrected by subtracting ∇P, where P is obtained by solving a Poisson equation.

Express this algorithm in semi-discrete form (i.e. write out the steps and equations used, as we did in class for a different fractional step method).

3. For Reynolds numbers of 0.1, 100, and 1000, run the simulation long enough to reach steady state. [Experiment with resolution and time stepping so that you have something fairly well resolved but that runs in a reasonable amount of time on your computer.]

   (a) Plot the velocity vectors and streamlines (as provided in navier.m) for each case. List the grid size, time step, and final time used for each plot.

   (b) Briefly comment on the differences in the three solutions, particularly with regard to symmetry and flow near the lower two corners.

4. Extract the converged velocity profiles along the 2 symmetry planes of the domain and compare to the benchmark data provided (see Benchmark.pdf on the course website) for Re = 1000. These data come from very high resolution previously published simulation results. [Note: because of the grid staggering, you will have to do some averaging to get the velocities at $x = 0.5$ and $y = 0.5$.] Give some reasons why your agreement is not perfect.

5. Check whether the code is actually second-order in space by doing a grid convergence study, using grids with factor of 2 refinement starting from 8×8 and ending with 128×128. Use Re = 0.1 for this part to ensure adequate resolution.

   (a) Derive a Richardson extrapolation formula to estimate the error using results from the two finest grids to estimate the solution on a 256×256 grid. See Sec. 3.9 in Ferziger and Peric.

(b) Plot the absolute value of the error in the u and v velocities at the center point of the domain versus Δx (on log-log axes) and comment on what you find. Include lines for 1st and 2nd order error on the same figure. [Using “grid on” will help too.]