Mech 510 Final Project

Due Date: 8 am, Thursday, December 19, 2019
Report and Code are both due by this time.
Late Penalty: 5% @8:01, plus 0.5% per hour after that.

The project for this course requires that you write a program to solve the two-dimensional incompressible laminar Navier-Stokes equations on regular cartesian grids. To demonstrate your successful program, you will be asked to solve the driven cavity problem (a rectangular cavity with horizontally-moving top), including an exploration of the vortex structure in the cavity.

This document describes the validation test cases your should run to test your code and results your should present for each case. Your report should include, at a minimum, the information requested under **Report:** in each subsection. Also, you should comment on anything else that you come across that strikes you as particularly interesting (or confusing, for that matter). **Section 4 of the assignment is open-ended; as such I expect some discussion and interpretation of results in your report for that section.**

1 General Validation

1.1 Correctness of residual

Consider a square geometry on $[0,1]\times[0,1]$ with no-slip boundary conditions and the following distribution of velocity and pressure:

$$\begin{pmatrix} P \\ u \\ v \end{pmatrix} = \begin{pmatrix} P_0\cos(\pi x)\cos(\pi y) \\ u_0\sin(\pi x)\sin(2\pi y) \\ v_0\sin(2\pi x)\sin(\pi y) \end{pmatrix}$$

Note that this data satisfies the boundary conditions. The exact flux integral divided by the cell size is given by

$$-\frac{\partial F}{\partial x}-\frac{\partial G}{\partial y}=\begin{pmatrix} -\frac{\pi}{\beta}\left(u_0C_xS_{2y}+v_0S_{2x}C_y\right) \\ \\ \{P_0\pi S_xC_y-u_0^2\pi S_{2x}S_{2y}^2 \\ -u_0v_0\pi S_xS_{2x}\left(C_yS_{2y}+2C_{2y}S_y\right) \\ -u_0\frac{5\pi^2S_xS_{2y}}{Re}\} \\ \\ \{P_0\pi C_xS_y-v_0^2\pi S_{2x}^2S_{2y} \\ -u_0v_0\pi S_yS_{2y}\left(C_xS_{2x}+2C_{2x}S_x\right) \\ -v_0\frac{5\pi^2S_{2x}S_y}{Re}\} \end{pmatrix}$$

where

$$\begin{aligned} C_x &\equiv \cos(\pi x) \\ S_x &\equiv \sin(\pi x) \\ C_y &\equiv \cos(\pi y) \\ S_y &\equiv \sin(\pi y) \\ C_{2x} &\equiv \cos(2\pi x) \\ S_{2x} &\equiv \sin(2\pi x) \\ C_{2y} &\equiv \cos(2\pi y) \\ S_{2y} &\equiv \sin(2\pi y) \end{aligned}$$

Compute the flux integral divided by cell size on a 20×20 and a 40×40 mesh using second-order accurate centered flux evaluations as discussed in class. Set u_0, v_0, andP_0 to 1, $\beta = 1$, and Re $= 10$ for the final calculation. For each case, compute the L_2 norm of the error in this quantity for each component of the flux. If the norm is approximately four times smaller for the finer mesh, then your flux calculation is likely to be correct.

Several things to note:

- In searching for errors in your fluxes, try having only one of u_0, v_0, andP_0 non-zero. Also, to distinguish errors in viscous terms from errors in inviscid terms, try setting Re $= 10^{-5}$ or Re $= 10^5$.
- About signs: the above results use the negative of the flux integral, because this is what appears in the final

discretization. You should be aware of this sign convention when writing and testing your code so that you can have the right signs in the right places. In particular, you will likely find it useful to print out (for each control volume in a small mesh) the exact flux integral and your value. If the magnitudes are the same, but the signs are opposite, then you simply need to adjust a sign in your code, not go on a major bug hunt.

Report: In your report, tabulate the L_2 norm of the error in the flux integral divided by cell size for all components and both meshes.

1.2 Correctness of implicit discretization — flux Jacobian check

The fully-implicit time discretization of the Navier-Stokes equations can be written as

$$\begin{aligned}
\delta U_{i,j} + \Delta t\, A_x \delta U_{i-1,j} + \Delta t\, B_x \delta U_{i,j} + \Delta t\, C_x \delta U_{i+1,j} & \\
+ \Delta t\, A_y \delta U_{i,j-1} + \Delta t\, B_y \delta U_{i,j} + \Delta t\, C_y \delta U_{i,j+1} &= \\
-\Delta t \frac{F^n_{i+\frac{1}{2},j} - F^n_{i-\frac{1}{2},j}}{\Delta x} & \\
-\Delta t \frac{G^n_{i,j+\frac{1}{2}} - G^n_{i,j-\frac{1}{2}}}{\Delta y} &
\end{aligned}$$

In Section 1.1, you demonstrated the correctness of your implementation of the right-hand side of this equation. Now you must check the left-hand side. Recall that the LHS arises from the approximation that

$$\begin{aligned}
&\left(\frac{F_{i+\frac{1}{2},j} - F_{i-\frac{1}{2},j}}{\Delta x} + \frac{G_{i,j+\frac{1}{2}} - G_{i,j-\frac{1}{2}}}{\Delta y} \right)^{n+1} \\
&- \left(\frac{F_{i+\frac{1}{2},j} - F_{i-\frac{1}{2},j}}{\Delta x} + \frac{G_{i,j+\frac{1}{2}} - G_{i,j-\frac{1}{2}}}{\Delta y} \right)^{n} \approx \\
&\quad A_x \delta U_{i-1,j} + B_x \delta U_{i,j} + C_x \delta U_{i+1,j} \\
&\quad + A_y \delta U_{i,j-1} + B_y \delta U_{i,j} + C_y \delta U_{i,j+1}
\end{aligned} \tag{1}$$

Begin with $U^n_{i,j}$ set to the values prescribed in Section 1.1 for a 20×20 mesh; calculate the residual with this data. Now set

$$\delta U_{i,j} = \begin{cases} 10^{-6} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} & i = j = 10 \\ 0 & \text{otherwise} \end{cases}$$

Then set $U^{n+1}_{i,j} = U^n_{i,j} + \delta U_{i,j}$, and calculate the residual again. Now you can evaluate the LHS of Eqn. 1.

Now use your Jacobian calculations and δU to evaluate the RHS of Eqn. 1. Calculate the error for each term for each cell where the flux integral changes when you add $\delta U_{10,10}$; that is, for all cells where either the LHS or RHS of Eqn. 1 is non-zero (it should be the same cells). The error should in all cases be about 10^{-10} or smaller — perhaps substantially smaller.

Note that (with a little finesse and a large time step) you can use the approximately factored version of the LHS here; this has the advantage that it also tests your matrix assembly for the approximate factors.

For debugging purposes, feel free to set components of U and δU equal to zero as desired — this will narrow down the errant terms much more quickly. For example, if you set $\delta U = 10^{-6} \begin{pmatrix} 0 & 1 & 0 \end{pmatrix}^T$, you'll be able to test the second column of your Jacobians. If you also manipulate (for instance) the Reynolds number in your testing, you'll be able to tell if errors you see come from viscous or inviscid terms.

Report: Report the error in your approximation to Equation 1 for the case described in the previous section and a Reynolds number of 10.

1.3 Block Thomas Algorithm

An implementation of the block Thomas algorithm is available on Canvas. You will probably want to use this code rather than writing your own.

1.4 Approximate factorization scheme

Use your residual and Jacobian code, and the provided block tri-diagonal code to construct an approximately factored time advance scheme for the momentum equations. On Canvas, you will find comparison data for the intermediate values $\delta\tilde{U}$ and final value δU for the first time step for the case in Section 3.1 with $\Delta t = 0.05$. Note that the data specifically assumes that Neumann boundary conditions have been applied for pressure and no slip conditions have been applied for velocities. You'll want to confirm that your initial solution, including ghost cell data, matches mine before proceeding to other parts of the test.

2 Things to Watch Out For

There are several things that you should look out for or experiment with while debugging and optimizing your code for the cases in the following section.

1. **Pressure oscillations.** You will almost certainly find that your steady-state pressure distributions have some oscillations. These are present because of decoupling between pressure in alternate lines of the mesh. To fix this problem, you can add a term to the right-hand side of the pressure equation that looks something like: $A\Delta x\Delta y\nabla^2 P$. You want to choose the constant A so that you smooth out the oscillations, but you want A to be as small as possible, since you are after all introducing an error in the steady-state enforcement of continuity. The good news is that this error will be second-order.

2. **Choice of β.** While using $\beta = 1$ is a reasonable choice for these cases, it is probably not optimal. Feel free to experiment with this to see if you can find a value that works significantly better.

3. **Overrelaxation.** Overrelaxing your solution update does in fact improve convergence rate (although including overrelaxation is unlikely to stop your code from blowing up, if it's exhibiting that behavior already). Experiment with this also, if you like.

If you choose to implement or experiment with any of these items, include a description of what you did and the effect it had in your report.

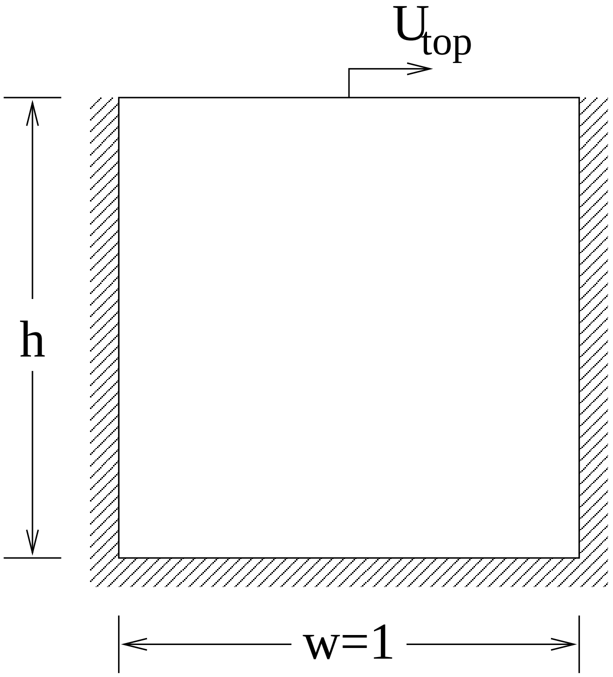

Figure 1: Problem geometry for flow in a box

3 Flow in a Box with a Moving Top

This problem concerns flow in an enclosed box with a moving top, as shown in Figure 1. The walls all have no-slip boundary conditions. The lid of the box also has a no-slip boundary condition, but in this case the boundary condition is for no slip at the constant velocity U_{top}. The fluid is isothermal, so buoyancy effects are not present. As the top of the box moves to the right, the fluid in the box circulates clockwise (for a square box). For all cases in this section, use a Reynolds number of 100 and $\beta = 1$. Your code should be able to handle a time step of at least $\Delta t = 0.05$ and possibly much higher; higher time steps will of course mean faster convergence, at least up to a point.

The basic physical results for the problem will be reported using the u-velocity along the line of vertical symmetry of the box.

3.1 Validation case: Stability

Begin with the velocity and pressure distributions of Section 1.1. Verify that your code converges toward zero velocity and uniform pressure. Use $h = 1$ and $U_{\text{top}} = 0$.

Report: Convergence history, plotting the log of the L_2 norms of change in u, v, and P versus iteration number for the first 200 time steps at $\Delta t = 0.05$ and $\beta = 1$, with no over-relaxation.

3.2 Basic solution

For all of the cases in this subsection, use $h = 1$. You may want to experiment with $\Delta t > 0.05$ or with over-relaxation for improved convergence. Be sure to indicate the value used so that I can assess your convergence histories in the proper context.

3.2.1 Solution for $U_{\text{top}} = 1$

Solve the problem for $U_{\text{top}} = 1$ for a 20×20 mesh.

Report: Convergence history until the largest L_2 norm of the change in solution is smaller than 10^{-6}. [1] Plot of u along the symmetry line ($x = 1/2$). Surface or contour plot of pressure.

[1] For reference, my code takes about 600 time steps and 0.3 seconds on a laptop for this case at $\Delta t = 0.05$ with no overrelaxation. Your code will probably not be quite this fast, but you should be concerned if you're vastly slower (more than 2000 time steps and/or more than about 20 seconds on modern computer hardware).

3.2.2 Sanity check: Symmetry

Solve the problem for $U_{\text{top}} = -1$ for a 20×20 mesh.

Report: Surface or contour plot of $u_{U_{\text{top}}=1}(x,y) + u_{U_{\text{top}}=-1}(1-x,y)$, which should be very small everywhere, though not quite zero, because of lack of complete iterative convergence.

3.2.3 Grid convergence

Solve the problem for $U_{\text{top}} = 1$ for a 10×10 mesh and as many meshes finer than 20×20 as you feel are needed to demonstrate grid convergence.

Report: On the same plot, show u on the symmetry line for each mesh as a function of y. How fine a mesh do you feel is needed for grid convergence for this quantity?[2] Back up your answer quantitatively.

4 Exploratory case for flow in a box: Effect of increasing *h*

As the height h of the box increases, the single vortex eventually becomes unstable, and a second vortex forms below it (and eventually a third, and so on). The exploratory part of this problem will focus on this formation of additional vortices with increasing h. Be careful to converge your solutions far enough; some of the flow features you're after are quite weak, and won't show up reliably if you only converge to, say, solution update norms of 10^{-7}. If you have streamlines that are not closed loops (or almost closed loops), then you've got a problem.

For this part of the project, use $\text{Re} = 300$, $U_{\text{top}} = 1$, and $w = 1$. For $h = 2.5$, compute the solution for flow in the box. Use meshes (of whatever size you think is appropriate) with $\Delta x = \Delta y$.

Report: Present evidence that you consider convincing about the number and strength of the vortices present for this case, including demonstrating that your numerical solution is grid converged. Give the center location of each vortex. Also, estimate the relative strength of the vortices (and describe how you have made this estimate). For both the vortex locations and the vortex strengths, provide a quantitative bound on the error in your results. (*Hint #1: you can do better than mesh size as an error bound on vortex location. Hint #2: Use the ASME JFE guidelines on reporting numerical error to demonstrate grid convergence.*)

[2] Note that a different resolution may be required for grid convergence of other quantities.