Project 3 Solution: Flow Over a Backward-Facing Step

Erin Levesque

AE 523: Computational Fluid Dynamics

Fall 2022

1 Introduction

In this project, we were tasked with simulating flow behind a backward-facing step pf height H. The flow in this problem is described by two-dimensional incompressible Navier-Stokes equations, and a projection-based, primitive-variable algorithm was developed to solve the equations. We will solve this system under the following conditions: Dirichlet boundary conditions on the top and bottom walls of the domain and Neumann boundary conditions at the inflow and outflow. From the solution results, we observe how varying Reynolds numbers affect flow behavior in the domain, and how varying grid size affects computational cost and accuracy. The solutions were validated by comparing to experimental and numerical results from provided published resources. This report presents the problem formulation, solver methodology, and a discussion of the results.

2 Problem Set-up

The domain of interest is shown in Figure (1).

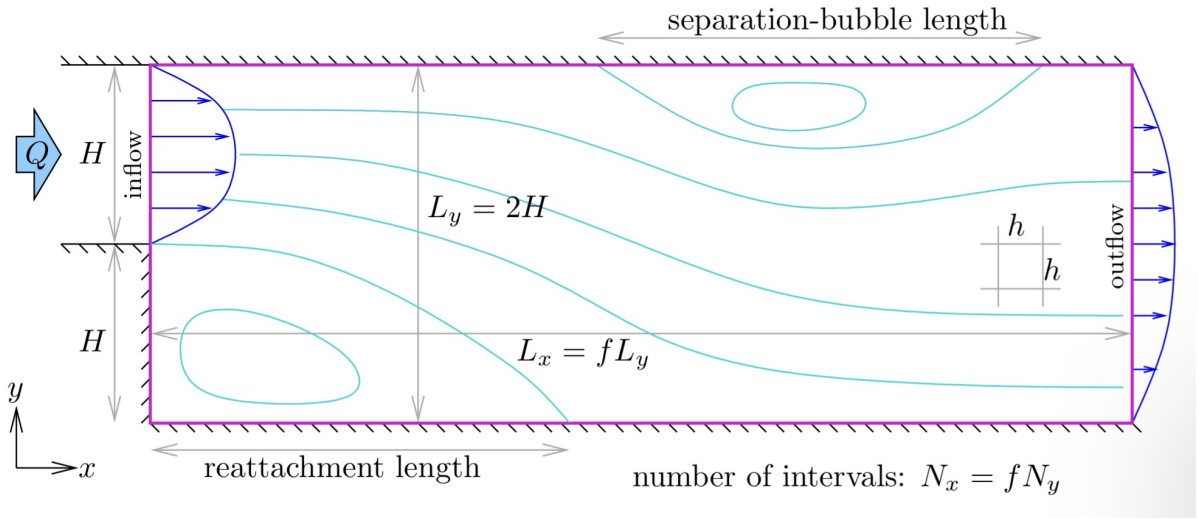


Figure 1: Backward-facing step domain setup

2.1 Governing Equations

This system is governed by the Navier-Stokes equations for incompressible flow. We use the given parameter that the flow has a globally constant density ρ, decoupling the energy equation from the conservation law system, and a constant viscosity ν. The 2D continuity (1) and momentum (2) equations are as follows:

$$\partial_t \rho + \nabla \cdot (\rho \vec{v}) = 0 \Rightarrow \nabla \cdot \vec{v} \tag{1}$$

$$\partial_t \vec{v} + (\vec{v} \cdot \nabla)\vec{v} + \frac{1}{\rho}\nabla p = \nu \nabla^2 \vec{v} \tag{2}$$

The nonlinear term in the above momentum expression is the convective term and the linear is the diffusive term. With the assumed $rho = 1$, we can re-write the conservation form of momentum as follows:

$$u_t + (F + p)_x + H_y^x = 0) \tag{3}$$

$$v_t + (G + p)_y + H_x^y = 0 \tag{4}$$

where F, G, H_x, H_y are defined as:

$$F = u^2 - \nu u_x \tag{5}$$

$$G = v^2 - \nu v_y \tag{6}$$

$$H_x = uv - \nu u_y \tag{7}$$

$$H_y = uv - \nu v_x \tag{8}$$

These momentum fluxes are used in the fractional update of the velocity, as discussed in more depth in Section 3. F and G are stored in the cell centers, while H_x and H_y are defined on grid nodes.
Taking the divergence of Equation 2 results in the elliptic pressure Poisson equation (PPE), which is used to get a value for pressure to correct and advance the velocity field at each time step. The 2D PPE expression is defined in (9).

$$\nabla^2 p = -\rho \nabla \cdot (\vec{v} \cdot \nabla \vec{v}) = 2\rho(u_x v_y - u_y v_x) \tag{9}$$

At each time step and at each point within the domain, the flow must conserve momentum and remain divergence-free.

2.2 Boundary Conditions

This domain has two horizontal walls on the top and bottom, and a vertical wall on the left, below the inflow. At all walls, the no-slip and zero flow-through conditions apply. Therefore, the tangential flow velocity is required to be the same as the wall velocity, and the normal flow velocity is required to be zero:

$$\text{at solid walls: } \vec{v} = 0 \Rightarrow \vec{v_t} = 0 \Rightarrow \nabla p = \nu \nabla^2 \Rightarrow \tag{10}$$

The inflow and outflow velocity profiles are parabolic. The flow at these boundaries is fully-developed, indicating the gradient of u and v velocities in the x-direction are zero. The flow rate Q, defined as the integral of the horizontal wall across a vertical channel, is used to determine the inflow and outflow velocities. First, the parabolic functions for the inflow and outflow velocity profiles were determined. Then, the y-values to be input into the function were defined as the vertical midpoint of each cell edge along the respective boundary. Finally, a discrete Q was calculated by taking the sum of the parabolic functions for each cell interval multiplied by the cell size h. The correct boundary velocities were then determined by multiplying the calculated $u(y)$, determined from the parabolic function, by the scaling factor of $s = \frac{Q}{Q_{discrete}}$.

2.3 Discretization

The projection method used in this solver implements a staggered storage grid for the primitive variables u, v, p. In a staggered storage grid, velocity and pressure are staggered by half of a grid point in order to prevent pressure oscillations that could lead to non-physical results. U-velocities are located on vertical edges, v-velocities are located on horizontal edges, and pressure values are located on cell centers. The grid is comprised of uniform $h \times h$ cells, where the cell size h is defined as the height of the domain (L_y)) divided by the number of vertical intervals (N_y)).

For all linear terms in the system, the use of a staggered grid is consistent with first-derivative and second-derivative central differences. However, the nonlinear convection terms are required for discretizations done at edges. Therefore, we implement a "ghost-cell" technique in which the domain is padded with extra cells around it that maintain the symmetry of the stencil. For the methods used in this solver, the ghost points are where boundary conditions are enforced.

3 Solution Method

The solver developed makes use of the projection pressure correction method to advance the flow. This approach, also known as the fractional steps method, takes the velocity field at each half time step and projects it onto the space of the divergence of free velocities. To march forward in time, an explicit Forward Euler scheme was implemented. Because the scheme is explicit, we consider this program to be conditionally stable.

The projection method implemented in this solver is described by the following procedure:

1. Initialize primitive variables u, v, p to zero.
2. Determine the inflow and outflow velocities. Set ghost points to enforce boundary conditions.
3. Calculate the momentum fluxes: F, G, H_x, H_y.
4. Update the velocity at the fractional time step, $\vec{v}^{n+\frac{1}{2}}$.
5. Solve the Pressure Poisson Equation
6. Correct the velocity field using the calculated pressure.
7. Calculate the L1 residual error norm and check convergence.
8. Once the convergence criterion is met, post-process the results.

3.1 Flux Calculation

In order to perform the fractional-step update for velocity, we must calculate the fluxes F, G, H_x, H_y from the conservation of momentum equation, neglecting the pressure terms, as outlined in Equations 5,6,7, and 8. The solver makes use of a high-order upwinding scheme called Quadratic Upstream Interpolation for Convection Kinematics (QUICK). This is necessary to improve solution accuracy by mitigate potential nonphysical oscillations that arise from simple central differencing of flux terms. To implement this scheme, we first recognize the inviscid terms of the fluxes are defined by a transport velocity (q) and a transported quantity (ϕ)). For example, the inviscid F flux is $q\phi = uu$ which represents the transport of u in the x-direction. The average transport velocity is utilized; for the F flux, $q = q_{i,j} = (u_{i-\frac{1}{2},j} + u_{i+\frac{1}{2},j})/2$. Next, we determine phi with the following 3-point upstream quadratic interpolation:

$$\text{if } q_{i,j} > 0,\ \phi_{i,j} = \frac{3u_{i+\frac{1}{2},j} + 6u_{i-\frac{1}{2},j} - u_{i-\frac{3}{2},j}}{8} \tag{11}$$

$$\text{if } q_{i,j} \leq 0,\ \phi_{i,j} = \frac{3u_{i-\frac{1}{2},j} + 6u_{i+\frac{1}{2},j} - u_{i+\frac{3}{2},j}}{8} \tag{12}$$

Once the momentum fluxes are determined, we use second-order accurate central differences to get the following derivatives:

$$F_x|_{i+\frac{1}{2},j} = \frac{1}{h}(F_{i+1,j} - F_{i,j}) \tag{13}$$

$$H^x_y|_{i+\frac{1}{2},j} = \frac{1}{h}(H^x_{i+\frac{1}{2},j+\frac{1}{2}} - H^x_{i+\frac{1}{2},j-\frac{1}{2}}) \tag{14}$$

$$G_y|_{i,j+\frac{1}{2}} = \frac{1}{h}(G_{i,j+1} - G_{i,j}) \tag{15}$$

$$H^y_x|_{i,j+\frac{1}{2}} = \frac{1}{h}(H^y_{i+\frac{1}{2},j+\frac{1}{2}} - H^y_{i-\frac{1}{2},j+\frac{1}{2}}) \tag{16}$$

These first derivative values are required for updating the fractional-step velocity:

$$u^{n+\frac{1}{2}}_{i+\frac{1}{2},j} = u^n_{i+\frac{1}{2},j} - \Delta t(F_x|_{i+\frac{1}{2},j} + H^x_y|_{i+\frac{1}{2},j}) \tag{17}$$

$$v^{n+\frac{1}{2}}_{i,j+\frac{1}{2}} = v^n_{i.j+\frac{1}{2}} - \Delta t(G_y|_{i,j+\frac{1}{2}} + H^y_x|_{i,j+\frac{1}{2}}) \tag{18}$$

3.2 Pressure Poisson Equation

The fractional time step velocities were updating using momentum that excludes the pressure contribution. To correct these velocities, we now solve Pressure Poisson Equation, which is defined as:

$$\nabla^2 p^{n+1} = \frac{\nabla \cdot \vec{v}^{n+\frac{1}{2}}}{\Delta t} \tag{19}$$

This pressure-corrector value P in the domain is determined using a direct sparse matrix method. Following the form shown below in equation (20), A and b correspond to the left-hand and right-hand sides, respectively, of the corresponding 5-point stencil of (19). Before solving, a point in the A matrix must be set to zero such that the solution is non-singular.

$$A \setminus b = p \tag{20}$$

Boundary conditions must be enforced at each node that falls on the defined boundaries of the domain. The given no-slip boundary condition of the top and bottom walls set the b vector to zero at the corresponding indices.

After the pressure calculation is complete, the velocity field can be corrected as follows:

$$u^{n+1}_{i+\frac{1}{2},j} = u^{n+\frac{1}{2}}_{i+\frac{1}{2},j} - \frac{\Delta t}{h}(p_{i+1,j} - p_{i,j}) \tag{21}$$

$$v^{n+1}_{i,j+\frac{1}{2}} = v^{n+\frac{1}{2}}_{i,j+\frac{1}{2}} - \frac{\Delta t}{h}(p_{i,j+1} - p_{i,j}) \tag{22}$$

This pressure-correction step is implemented to make the velocity field at $n+1$ divergence free.

3.3 Residuals

Finally, the L1 residual norm is computed and logged before advancing to the next iteration. The total flux of the system is:

$$\vec{F} = \begin{bmatrix} F + p \\ H^y \end{bmatrix} \hat{x} + \begin{bmatrix} H^x \\ G + p \end{bmatrix} \hat{y} \tag{23}$$

With the staggered storage grid technique, we can most conveniently define two flux residuals for the cell control volume: horizontal and vertical,

$$R_{i+\frac{1}{2},j} = h(F_{i+1,j} + p_{i+1,j} - F_{i,j} - p_{i,j}) + h(H^x_{i+\frac{1}{2},j+\frac{1}{2}} - H^x_{i+\frac{1}{2},j-\frac{1}{2}}) \tag{24}$$

$$R_{i,j+\frac{1}{2}} = h(G_{i,j+1} + p_{i,j+1} - G_{i,j} - p_{i,j}) + h(H^y_{i+\frac{1}{2},j+\frac{1}{2}} - H^y_{i-\frac{1}{2},j+\frac{1}{2}}) \tag{25}$$

The total L1 norm residual is then the sum of the L1 norm of the horizontal residual and of the vertical residual. Figure (2), displays the convergence of this quantity as the solver progresses through time for the case of Ny = 16 Re = 200. The FVM solver deems the steady-state solution accurate when the (L1) norm is less than the specified tolerance of 10^{-5}. The iterations required to reach steady-state ranged from approximately 6000 to 50k depending on what grid size and Reynolds number are input.

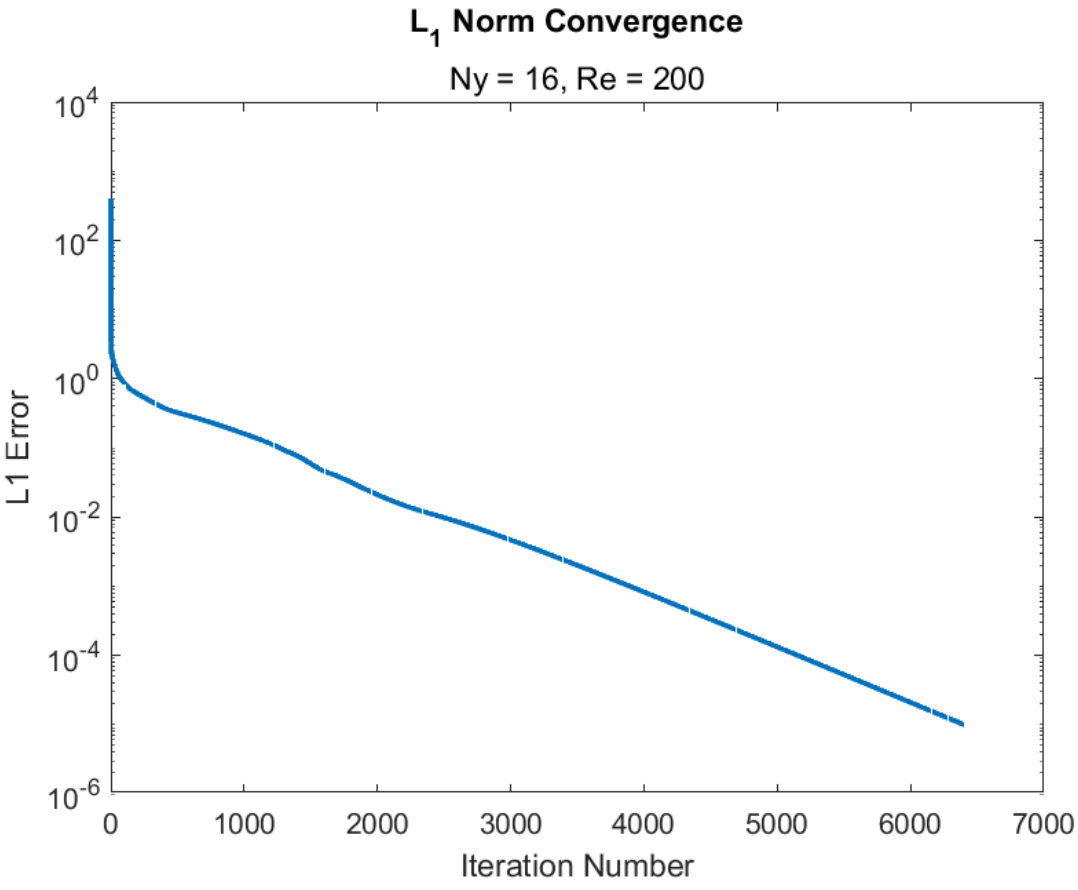


Figure 2: Convergence plot of L1 Residual Norm, Ny = 16, Re = 200

4 Post-Processing

After the solution has reached a converged state, quantities can be extracted to better visualize the flow data through the domain.

4.1 Streamlines

In order to visualize the streamlines on the staggered grid, we compute the stream function (Eq. 26) and plot the contour, as shown in Figure (3).

$$\frac{\partial \psi}{\partial y} = u, \quad \frac{\partial \psi}{\partial x} = -v \tag{26}$$

The stream function on an arbitrary path between two points A and B can be calculated discretely. This start, we pick an initial corner A and set $\psi = 0$ at that location. We use the u and v velocities at the vertical and horizontal edge midpoints, respectively, and calculate a discrete flow rate through an edge. Summing these values gives is $\psi(B)$. Because the converged solution should be divergence-free throughout the domain, the order that edges are iterated does not matter, as long as the normal direction remains consistent. The boundary conditions in the domain are implemented explicitly to ensure they are accounted for in the stream function.

4.2 Reattachment Length

To visualize the reattachment length of the flow over the step, we compute $\frac{\delta u}{\delta y}$ on the walls. To get the u values, we compute a central finite difference for a u and each x location. From this calculation, we identify the reattachment point to be where $\frac{\delta u}{\delta y} = 0$ on the bottom wall. For cases where a secondary recirculation separation bubble appears, the separation point is identified as where $\frac{\delta u}{\delta y} = 0$ on the top wall.

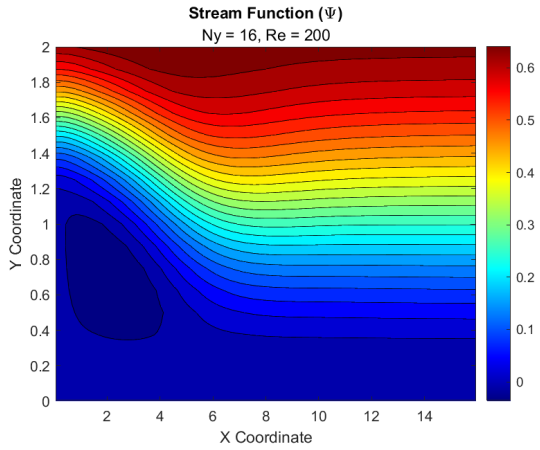
The experimental and theoretical results represented in Figure (5) indicate the for Re = 200,400,600, the reattachment length are expected to be approximately L = 4,6,10. The reattachment lengths calculated by this solver are L = 4.53, 7.15, and 8.41 for Ny = 16 and L = 5.08, 8.22, 10.42 for Ny = 32. These values strongly agree with the results from the provided papers. The provided papers also indicate that separation bubbles begin to appear at Re = 400. However, a separation bubble only appears in the solution of this project for the cases of Re = 600. For Ny = 32, the separation length is L = 5.201 and for Ny = 16, the separation length is L = 3.125.

5 Results and Discussion

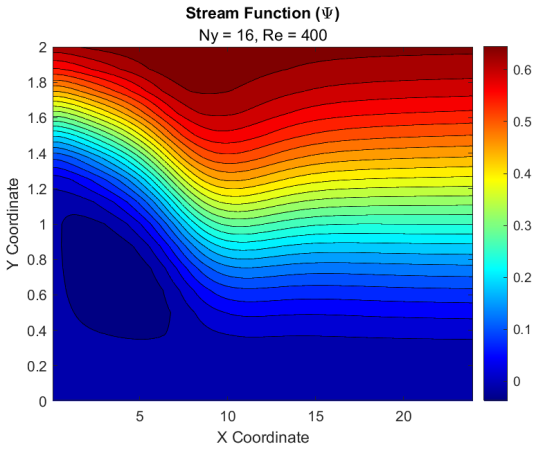
Simulations for flow over a backward-facing step were carried out for Reynolds numbers = 200,400,600 and grid sizes = 16,32. As mentioned previously, the domain has solid, stationary walls and parabolic inlet and outlet velocity profiles. The algorithm uses a global time step in Forward Euler to march the solution toward a steady state.

The streamlines shown in Figure (3) display the flow separation phenomena that occurs at a sudden change in geometry such as the backwards-facing step. A single circulation region appears attached to the step for all cases, and a secondary recirculation region arises for higher Reynolds numbers as the flow direction changes.

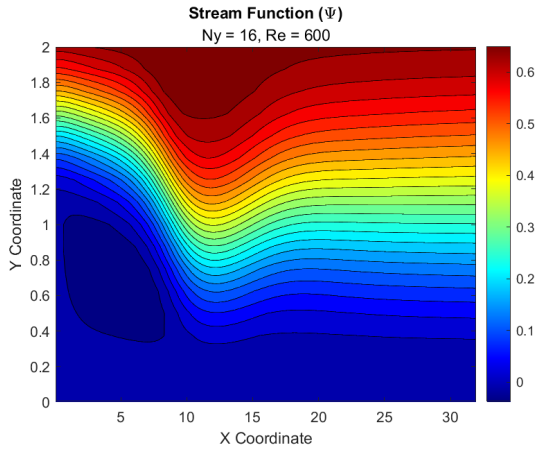
The converged, steady-state solution was compared to results from published paper written by Kim and Moin and by Armaly et. al. The flow simulation developed by Kim and Moin is second-order accurate



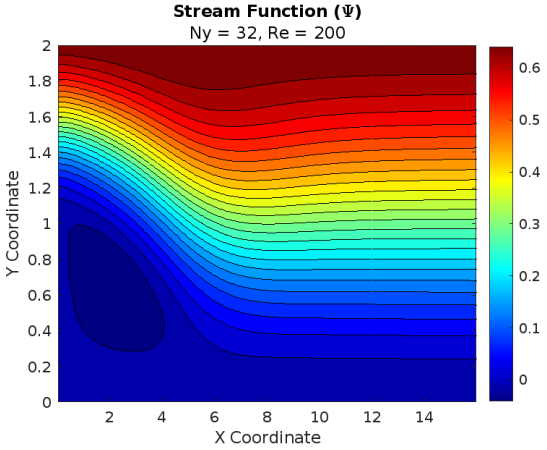
(a) Ny=16, Re=200



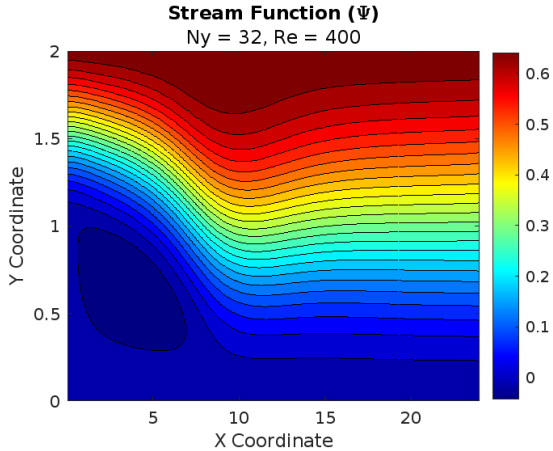
(b) Ny=16, Re=400



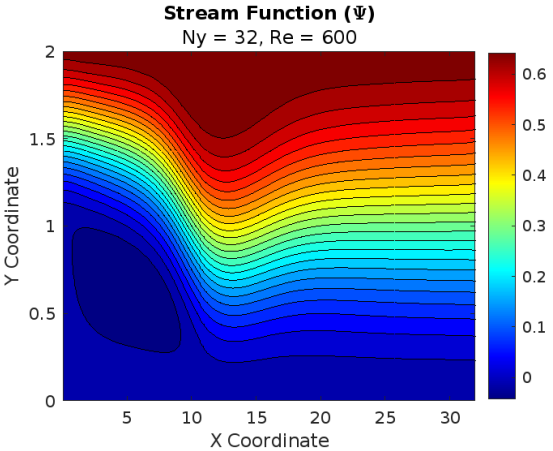
(c) Ny=16, Re=600



(d) Ny=32, Re=200

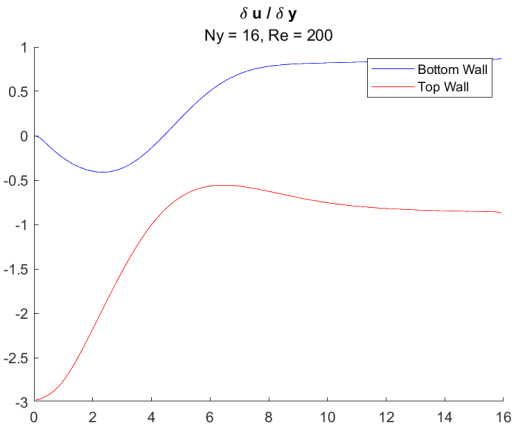


(e) Ny=32, Re=400

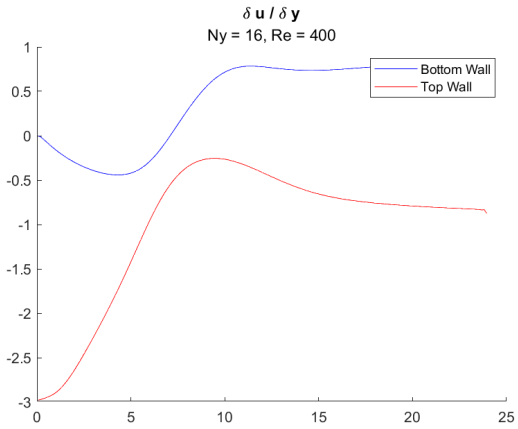


(f) Ny=32, Re=600

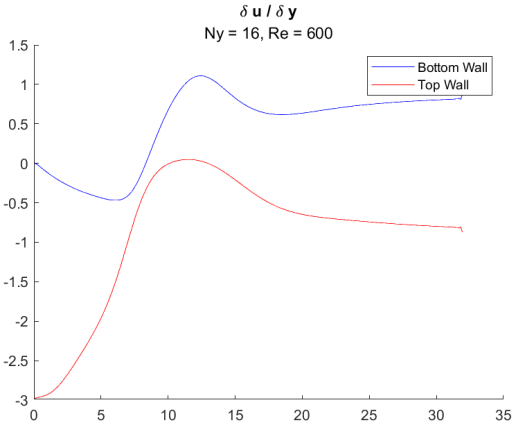
Figure 3: Streamline Plot for Each Converged Case



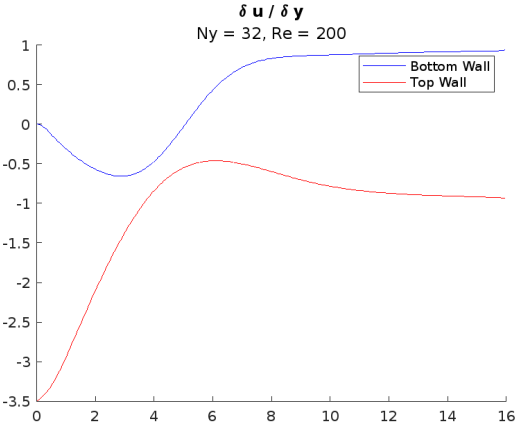
(a) Ny=16, Re=200



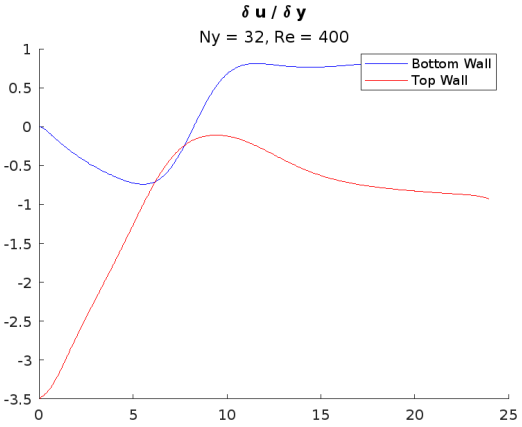
(b) Ny=16, Re=400



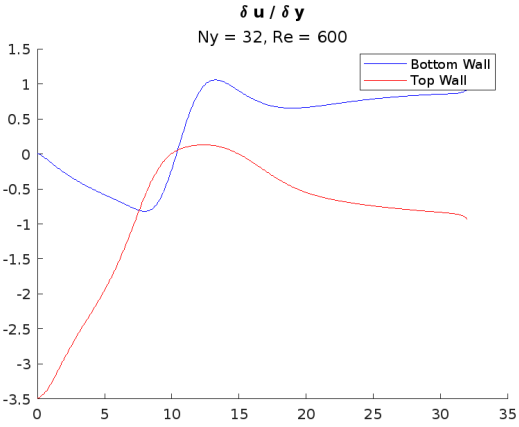
(c) Ny=16, Re=600



(d) Ny=32, Re=200



(e) Ny=32, Re=400



(f) Ny=32, Re=600

Figure 4: Plot of $\frac{\partial u}{\partial y}$ for Each Converged Case

in space and time. Kim and Moin's results strongly agree with the experimental and analytical data presented in Armaly et. al for Reynolds numbers less than 500, as shown in Figure (5). For Reynolds numbers greater than 500, the results deviate, likely due to the three-dimensionality of the experimental flow.

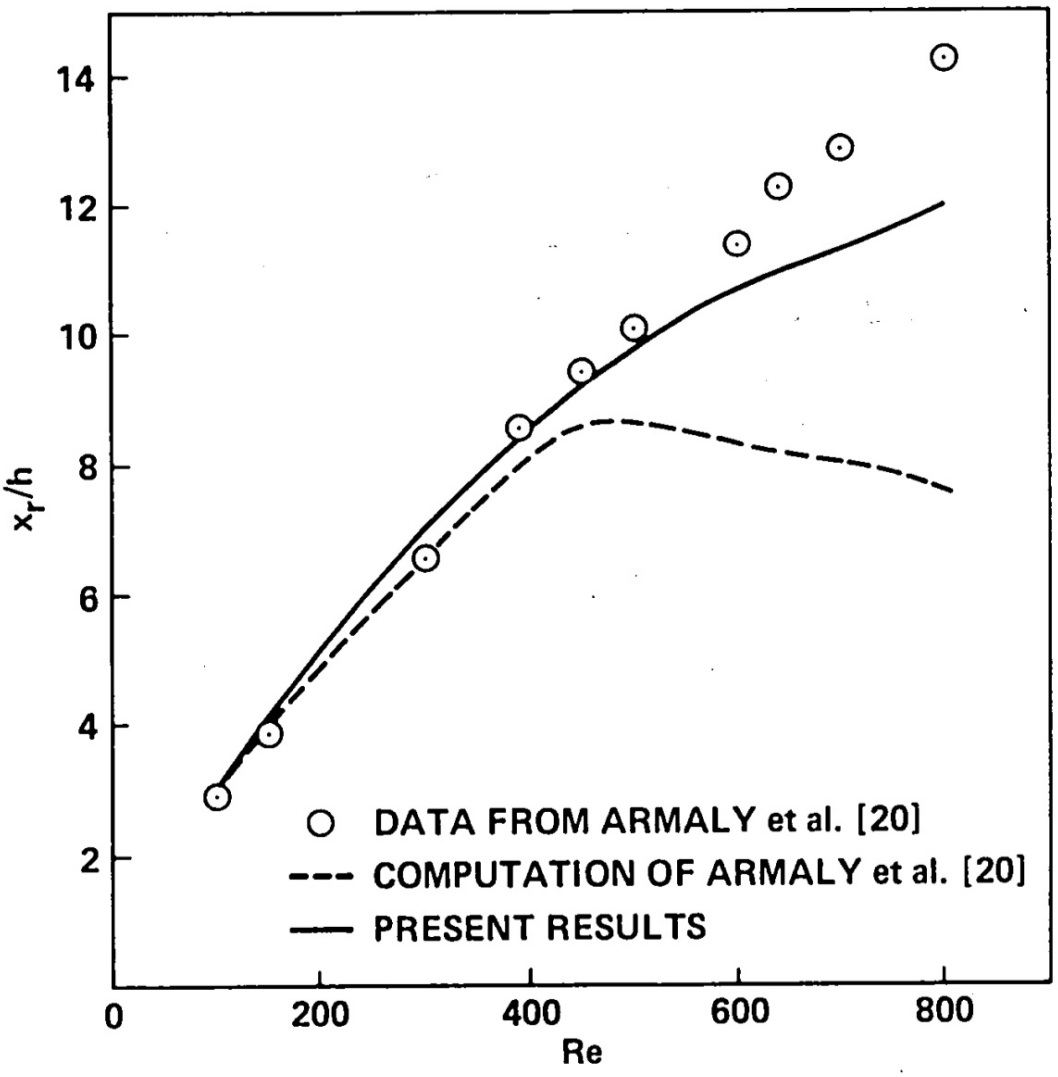


Figure 5: Comparison of Computational and Experimental Results, by J. Kim and P. Moin

The most apparent pitfall of the developed solver is the computational time. For The case of Re = 200 and Ny = 16, the solution converged around 6000 iterations. However, for the case of Re = 600 and Ny = 32, the solution converged around 50k iterations, and the computational time was on the order of hours rather than seconds. An improvement in convergence time was found by adjusting the stability factor, β.