Project 1 Solution: Serpentine Pipe

AE 523: Computational Fluid Dynamics
Erin Levesque

Fall 2022

1 Introduction

In this project we were tasked to solve for two-dimensional potential flow in a pipe. We will map the domain of the pipe to a reference space in order to utilize finite difference method. We will solve this system under the following conditions: Dirichlet boundary conditions on the top and bottom walls of the pipe and Neumann boundary conditions at the inflow and outflow.

2 Problem Set-up

The pipe is a curved computational domain represented in the provided XY grid that we will map to a rectangular reference space. The coordinates (x, y) are in the global space, the space that contains the irregular pipe domain, on which we want to approximate finite differences. The given points in the XY files start on the lower-left corner and go downstream along the pipe first.The reference space is in the coordinate plane of ξ and η, where Ny is the number of points in the ξ direction and Nx is the number of points in the η direction.

Flow can be assumed to be perpendicular to the pipe at both the inflow and outflow. This is enforced a Neumann conditions at these boundary that states: $\nabla\psi \cdot \overrightarrow{n} = 0$, where $\overrightarrow{n}$ is the outward-facing normal vector. Because no flow crosses through the walls of the pipe and the area flow rate through the pipe is $Q = 1\frac{m^2}{s}$, the Dirichlet boundary conditions on the top and bottom walls will be enforced by setting $\psi = Q, 0$ respectively.

3 Solution Methods

3.1 Mapping Laplace Equation PDE Derivation and Implementation

We are first tasked with writing the Laplace equation from the stream function, Ψ, in terms of the reference frame coordinates ξ and η. The equation in global coordinates is as follows:

$$\nabla^2\psi = \frac{\partial^2\psi}{\partial x^2} + \frac{\partial^2\psi}{\partial y^2} = \psi_{xx} + \psi_{yy} \tag{1}$$

The mapped equation is found from the following derivation:

$$\psi_x = \psi_\xi \xi_x + \psi_\eta \eta_x \tag{2}$$

$$\psi_{xx} = \frac{\partial\psi_x}{\partial x} = \psi_{\xi\xi}\xi_x^2 + \psi_{\eta\eta}\eta_x^2 + 2\psi_{\xi\eta}\xi_x\eta_x + \psi_\xi\xi_{xx} + \psi_\eta\eta_{xx} \tag{3}$$

$$\psi_y = \psi_\xi \xi_y + \psi_\eta \eta_y \tag{4}$$

$$\psi_{yy} = \frac{\partial \psi_y}{\partial y} = \psi_{\xi\xi}\xi_y^2 + \psi_{\eta\eta}\eta_y^2 + 2\psi_{\xi\eta}\xi_y\eta_y + \psi_\xi\xi_{yy} + \psi_\eta\eta_{yy} \tag{5}$$

$$\nabla^2\psi = (\psi_{\xi\xi}\xi_x^2 + \psi_{\eta\eta}\eta_x^2 + 2\psi_{\xi\eta}\xi_x\eta_x + \psi_\xi\xi_{xx} + \psi_\eta\eta_{xx}) + (\psi_{\xi\xi}\xi_y^2 + \psi_{\eta\eta}\eta_y^2 + 2\psi_{\xi\eta}\xi_y\eta_y + \psi_\xi\xi_{yy} + \psi_\eta\eta_{yy}) \tag{6}$$

$$\boxed{= \psi_{\xi\xi}(\xi_x^2 + \xi_y^2) + \psi_{\eta\eta}(\eta_x^2 + \eta_y^2) + 2\psi_{\xi\eta}(\xi_x\eta_x + \xi_y\eta_y) + \psi_\xi(\xi_{xx} + \xi_{yy}) + \psi_\eta(\eta_{xx} + \eta_{yy})} \tag{7}$$

The ψ_ξ and ψ_η terms can be found using central difference approximations for first derivatives:

$$\psi_\xi = \frac{\partial \psi}{\partial \xi} = \frac{\psi_{i+1,j} - \psi_{i-1,j}}{2\Delta\xi} \tag{8}$$

$$\psi_\eta = \frac{\partial \psi}{\partial \eta} = \frac{\psi_{i,j+1} - \psi_{i,j-1}}{2\Delta\eta} \tag{9}$$

The $\psi_{\xi\xi}$ and $\psi_{\eta\eta}$ terms can be found using central difference approximations for second derivatives:

$$\psi_{\xi\xi} = \frac{\partial^2 \psi}{\partial \xi^2} = \frac{\psi_{i+1,j} - 2\psi_{i,j} + \psi_{i-1,j}}{2\Delta\xi} \tag{10}$$

$$\psi_{\eta\eta} = \frac{\partial^2 \psi}{\partial \eta^2} = \frac{\psi_{i,j+1} - 2\psi_{i,j} + \psi_{i,j-1}}{2\Delta\eta} \tag{11}$$

The $\psi_{\xi\eta}$ term can be found using the method of undetermined coefficients for a mixed derivative and solving for the a, b, c and d:

$$\psi_{\xi\eta} = a\psi_{i+1,j+1} + b\psi_{i-1,j11} + c\psi_{i+1,j-1} + d\psi_{i-1,j+1} \tag{12}$$

$$a = b = \frac{1}{4\Delta\xi\Delta\eta}$$

$$c = d = -\frac{1}{4\Delta\xi\Delta\eta}$$

To find the terms, ξ_x, ξ_y, η_x, η_y, we first need to use central or one-sided (on boundaries) differences of the provided X and Y matrices in order to calculate X_ξ, X_η, Y_ξ, Y_η.

For the interior nodes of the grid, the following central difference formulas can be used:

$$x_\xi|_{ij} \approx \frac{X_{i+1,j} - X_{i-1,j}}{2d\xi}$$

$$y_\xi|_{ij} \approx \frac{Y_{i+1,j} - Y_{i-1,j}}{2d\xi}$$

$$x_\eta|_{ij} \approx \frac{X_{i,j+1} - X_{i,j-1}}{2d\eta}$$
$$y_\eta|_{ij} \approx \frac{Y_{i,j+1} - Y_{i,j-1}}{2d\eta}$$

To approximate these derivatives on the boundary nodes, a one-sided formula is needed. It is required that we use a second-order accurate formula for our finite-differences. Therefore, first-order forward and backward differences can not be used on the boundaries. A one-sided right second-order formula would be used on the first row, for example, because we can not look to i indices before i = 1.

$$i = 1$$
$$x_\xi|_{ij} = \frac{dx}{d\xi} \approx \frac{-3X_{i,j} + 4X_{i+1,j} - X_{i+2,j}}{2d\xi}$$
$$y_\xi|_{ij} = \frac{dy}{d\xi} \approx \frac{-3Y_{i,j} + 4Y_{i+1,j} - Y_{i+2,j}}{2d\xi}$$

This one-sided approximation for the first derivative is derived from the method of undetermined coefficients for the first derivative at node 0, where it is assumed at we know two points on the left and no points on the right. The same derivation can be done with the assumption that two points on the right are known and none on the left. However, due to symmetry, the one-sided formulation from the opposing side can be intuitively found by multiplying the previous expression by -1, as we know that it is reflected across the y-axis.

Similarly, a one-sided left second-order formula would be used on the last column because we cannot look to j indices after j = n (n = max number of columns).

$$j = n$$
$$x_\eta|_{ij} = \frac{dx}{d\eta} \approx \frac{3X_{i,j} - 4X_{i,j-1} + X_{i,j-2}}{2d\eta}$$
$$y_\eta|_{ij} = \frac{dy}{d\eta} \approx \frac{3Y_{i,j} - 4Y_{i,j-1} + Y_{i,j-2}}{2d\eta}$$

Once these derivatives are found, we can use them to construct our Jacobian matrix and use it to compute the inverse derivatives:

$$\begin{bmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{bmatrix} = \frac{1}{x_\xi y_\eta - y_\xi x_\eta} \begin{bmatrix} y_\eta & -x_\eta \\ -y_\xi & x_\xi \end{bmatrix}$$

Using the chain rule, we can compute the following second derivatives at the interior and boundary nodes: ξ_{xx}, ξ_{yy}, η_{xx}, η_{yy}. The central difference expressions for ξ_{xx} and ξ_{yy} at interiors are shown below, and the other second derivatives are found in a similar manner.

$$\xi_{xx}|_{ij} = \xi_x \frac{\partial \xi_x}{\partial x} + \eta_x \frac{\partial \eta_x}{\partial x} = \xi_x|_{ij} \frac{\xi_x|_{i+1,j} - \xi_x|_{i-1,j}}{2d\xi} + \eta_x|_{ij} \frac{\eta_x|_{i,j+1} - \eta_x|_{i,j-1}}{2d\eta}$$

$$\xi_{yy}|_{ij} = \xi_y \frac{\partial \xi_y}{\partial y} + \eta_y \frac{\partial \eta_y}{\partial y} = \xi_y|_{ij} \frac{\xi_y|_{i+1,j} - \xi_y|_{i-1,j}}{2d\xi} + \eta_y|_{ij} \frac{\eta_y|_{i,j+1} - \eta_y|_{i,j-1}}{2d\eta}$$

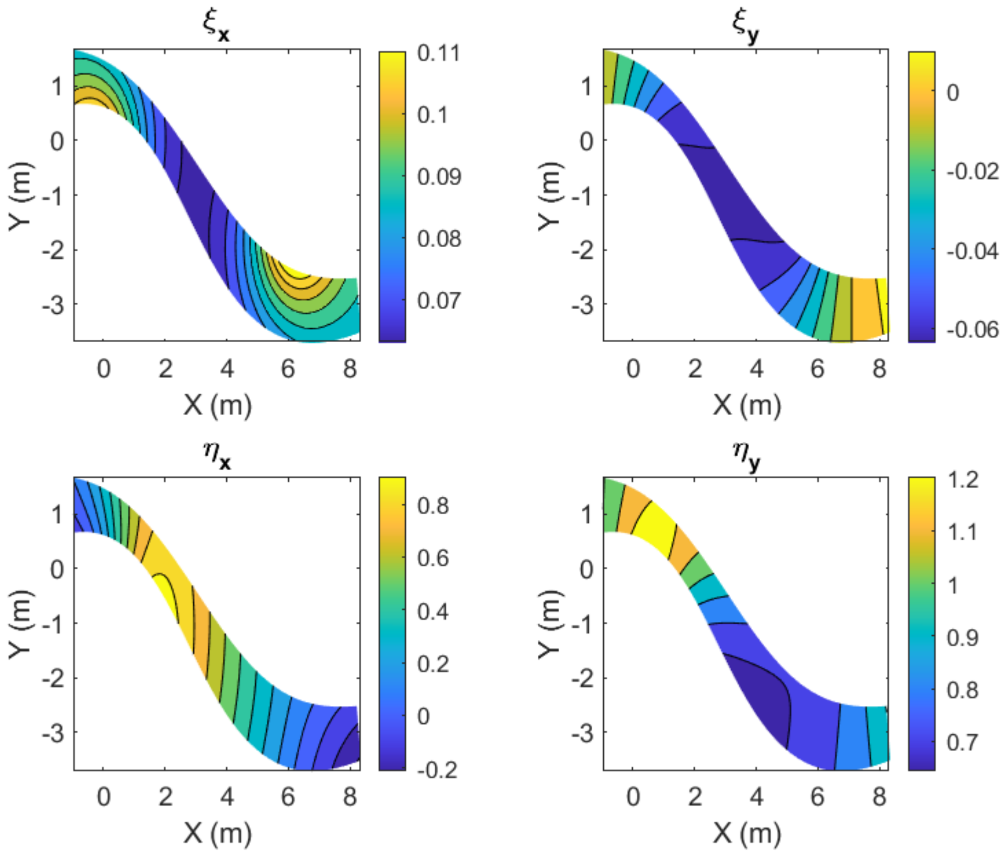

Figure 1: First Derivative Contours over Physical Domain XY

At the boundary nodes, these second derivatives are found using one-sided finite difference approximations. An example of these node expressions is shown below using η_{yy} at the right edge boundary:

$$\eta_{yy}|_{ij} = \xi_y \frac{\partial \eta_y}{\partial \xi} + \eta_y \frac{\partial \eta_y}{\partial \eta} = \xi_y \frac{\eta_y|_{i,j+1} + \eta_y|_{i,j-1}}{2d\xi} + \eta_y \frac{3\eta_y|_{i,j} - 4\eta_y|_{i,j-1} + \eta_y|_{i,j-2}}{2d\eta}$$

Once these first and second derivatives are found, they can be plugged back into Equation (7), $\nabla^2\psi$, along with the ψ central differences, resulting in the expression below.

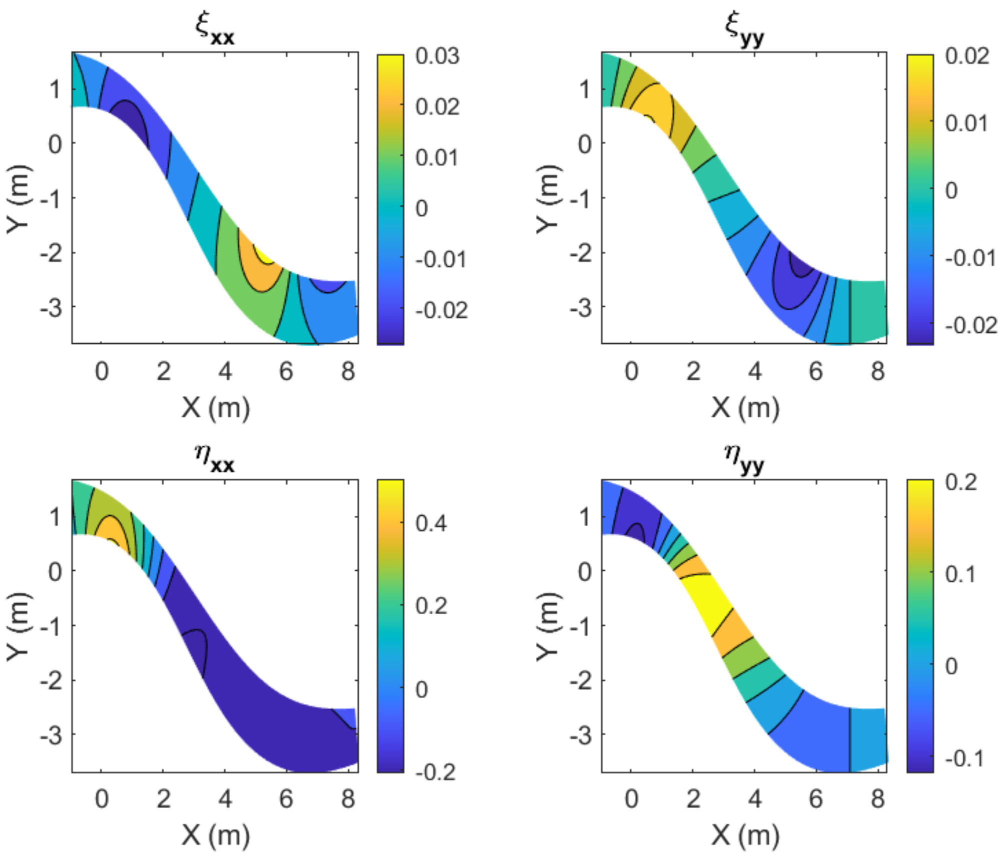

Figure 2: Second Derivative Contours over Physical Domain XY

$$
\begin{aligned}
\nabla^2\psi = (\psi_{i,j})(&\frac{-2(\xi_x^2+\xi_y^2)}{\Delta\xi^2} - \frac{2(\eta_x^2+\eta_y^2)}{\Delta\eta^2}) + \\
&(\psi_{i+1,j})(\frac{\xi_{xx}+\xi_{yy}}{2\Delta\xi} + \frac{\xi_x^2+\xi_y^2}{\Delta\xi^2}) + \\
&(\psi_{i-1,j})(\frac{-(\xi_{xx}+\xi_{yy})}{2\Delta\xi} + \frac{\xi_x^2+\xi_y^2}{\Delta\xi^2}) + \\
&(\psi_{i,j+1})(\frac{\eta_{xx}+\eta_{yy}}{2\Delta\eta} + \frac{\eta_x^2+\eta_y^2}{\Delta\eta^2}) + \\
&(\psi_{i,j-1})(\frac{-(\eta_{xx}+\eta_{yy})}{2\Delta\eta} + \frac{\eta_x^2+\eta_y^2}{\Delta\eta^2}) + \\
&(\psi_{i+1,j+1}+\psi_{i-1,j-1}-\psi_{i+1,j-1}-\psi_{i-1,j+1})(\frac{\xi_x\eta_x+\xi_y\eta_y}{2\Delta\xi\Delta\eta})
\end{aligned}
\tag{13}
$$

3.2 Finite-Difference Method Solution

A sparse linear system of equations is used to solve for the values of ψ in the domain. The system will take the form shown below in in equation (14), where $\psi$is an unrolled state vector of unknowns, A is the right-hand side (LHS) of the corresponding stencil formula, and $b$is the unrolled vector of the left-hand side (RHS) of the corresponding stencil formula.

$$A\psi = b \tag{14}$$

Using the coefficients of the ψ points in (13), we can fill interior points of the A matrix. Each row of the A matrix corresponds to a different node (i,j). The other points in the row correspond with the neighbor nodes of (i,j). The indices within the row correspond with where each of the neighbor nodes are located in the global grid space. The 9-point stencil used in this system, the point indices are listed below.

```
%n = number of columns in grid
iR = i+1; %to the right of (i,j)
iL = i-1; %to the left
iU = i+n; %up
iD = i-n; %down
iUL = i+n-1; %up and left
iUR = i+n+1; %up and right
iDL = i-n-1; %down and left
iDR = i-n+1; %down and right
```

Boundary conditions must be enforced at each node that falls on the defined boundaries of the domain. The given Dirichlet boundary conditions fully specify ψ on the top and bottom walls of the pipe. We must use second-order one-sided differences to derive the Neumann Boundary condition expressions, as shown below:

$$\nabla\psi \cdot \overrightarrow{n} = 0 \tag{15}$$

$$\psi_x n_x + \psi_y n_y = 0 \tag{16}$$

We can substitute in ψ_x and ψ_y mapped in terms of ξ and η to get:

$$= (n_x\xi_x + n_y\xi_y)(\frac{\psi_{i+1,j} - \psi_{i-1,j}}{2\Delta\xi}) + (n_x\eta_x + n_y\eta_y)(\frac{3\psi_{i,j} - 4\psi_{i,j-1} + \psi_{i,j-2}}{2\Delta\eta}) \tag{17}$$

OR

$$= (n_x\xi_x + n_y\xi_y)(\frac{\psi_{i+1,j} - \psi_{i-1,j}}{2\Delta\xi}) + (n_x\eta_x + n_y\eta_y)(\frac{-3\psi_{i,j} + 4\psi_{i,j-1} - \psi_{i,j-2}}{2\Delta\eta}) \tag{18}$$

where the finite difference equation used depends on if you are at the inflow or outflow boundary. As seen in equations (17) and (18), we need to a second-order accurate approach to calculate the normal vector at each node where the Neumann boundary condition is enforced. To do so, we can use the gradient of our reference space variables transformed to be with respect to reference variables. We can understand why this is true by considering a function constant on a given surface. If we take, for example, η to be the constant function along a boundary, then the derivative in any direction tangent to that surface must be 0. So, $\nabla\eta$ has 0 dot product with any tangent vector to the surface, making it normal a normal vector. To use in our Neumann boundary condition expression, we normalize that vector by diving by its magnitude.

Once we have a filled A matrix and b vector, we can solve for ψ using MATLAB's backslash operator. This gives us ψ as a vector which we then reshape into matrix.

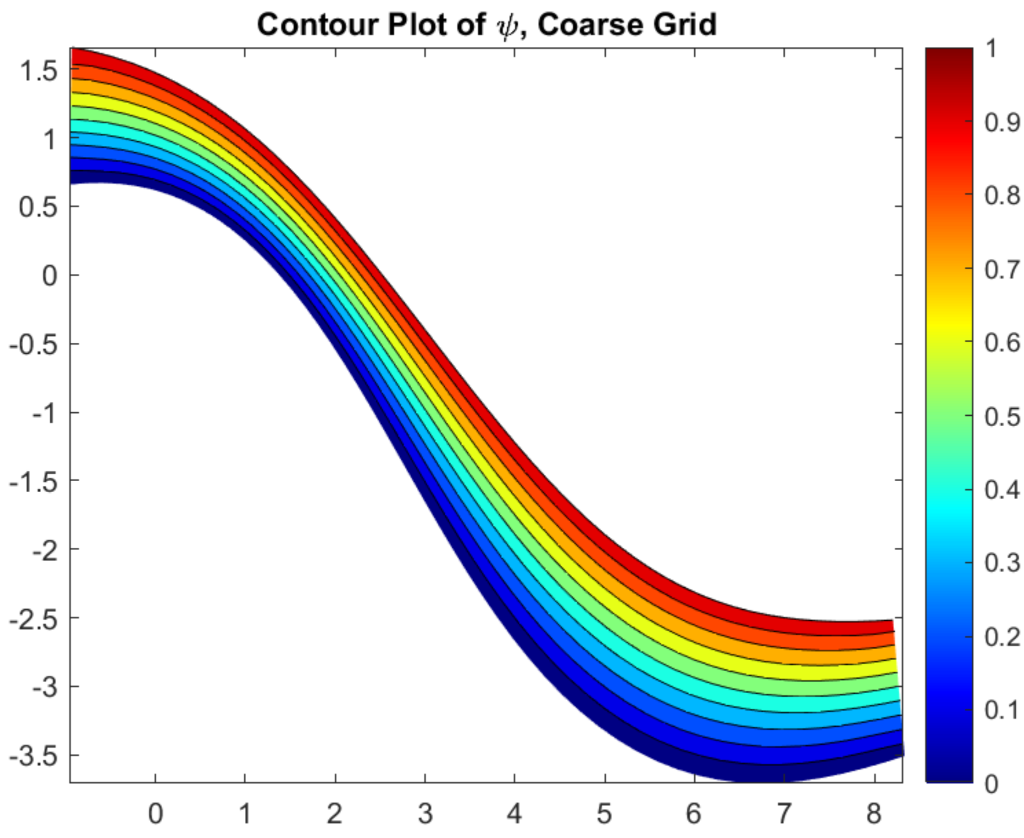

Figure 3: Contour Plot of ψ for Coarse Grid

The above contour plots for ψ show no clear differences between the coarse and fine grid.

3.3 Visualizing the Flowfield

3.3.1 Pressure Coefficient Derivation and Implementation

To better visualize the flowfield, we can generate a contour plot of the pressure coefficient for each point inside the pipe. We define pressure coefficient as:

$$c_p = 1 - \frac{|\overrightarrow{v}|^2}{V_0^2} \tag{19}$$

where $\overrightarrow{v}$ is the velocity of the fluid and $V_0 = 1$ is the reference average speed in a cross-section of the pipe. Because the fluid is moving through a curved pipe, we can assume the velocity will not generally be purely in the x or y direction. Therefore, the we must obtain both components of the velocity vector by differentiating ψ found in the previous section.

$$\overrightarrow{v} = v\hat{i} + u\hat{j} = -(\psi_\xi \xi_x + \psi_\eta \eta_x)\hat{i} + (\psi_\xi \xi_y + \psi_\eta \eta_y)\hat{j} \tag{20}$$

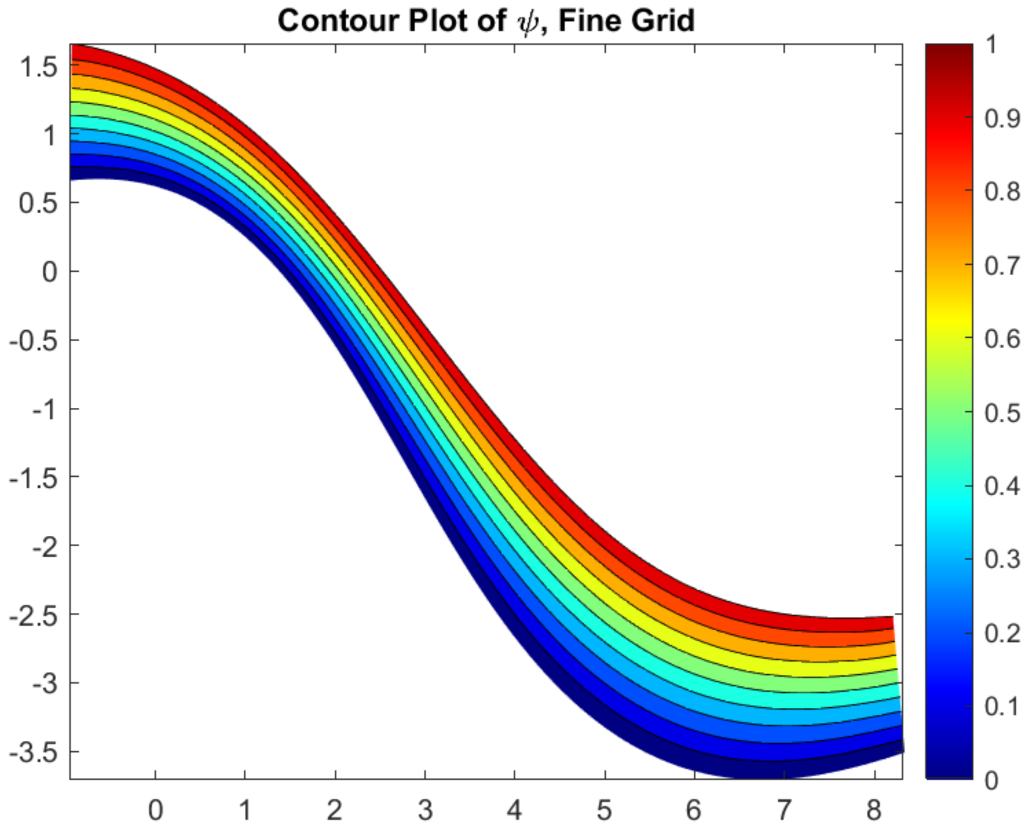

Figure 4: Contour Plot of ψ for Fine Grid

Values for ξ_x, ξ_y, η_x, and η_y at each node are already known from previous calculations, and we can approximate ψ_η and ψ_ξ using central difference for interior nodes and one-sided difference for boundaries. From here, we can calculate the velocity vector for each node in the domain and take the magnitude of that vector to complete our pressure coefficient calculations. Figures (5) and (6) show the contour plots for the pressure coefficient over the physical domain for both the coarse and fine grid.
From the contour plots, we can see that pressure is generally increasing in the direction of flow through the pipe. It can also be notes the concentration of high pressure flow at the bottom turn and the concentration of low pressure flow at the top curve near the wall.

3.3.2 Force Exerted in Pipe

The final quantity of interest is the force exerted on the top and bottom walls of the pipe by the fluid. This force vector can be found by taking the following integral in the global domain.

$$\overrightarrow{F} = \int_S c_p \overrightarrow{n}\, ds \tag{21}$$

To implement this force calculation, we first need to break the top and bottom walls into linear panel segments between the nodes. Then, loop through the segments and calculate the average

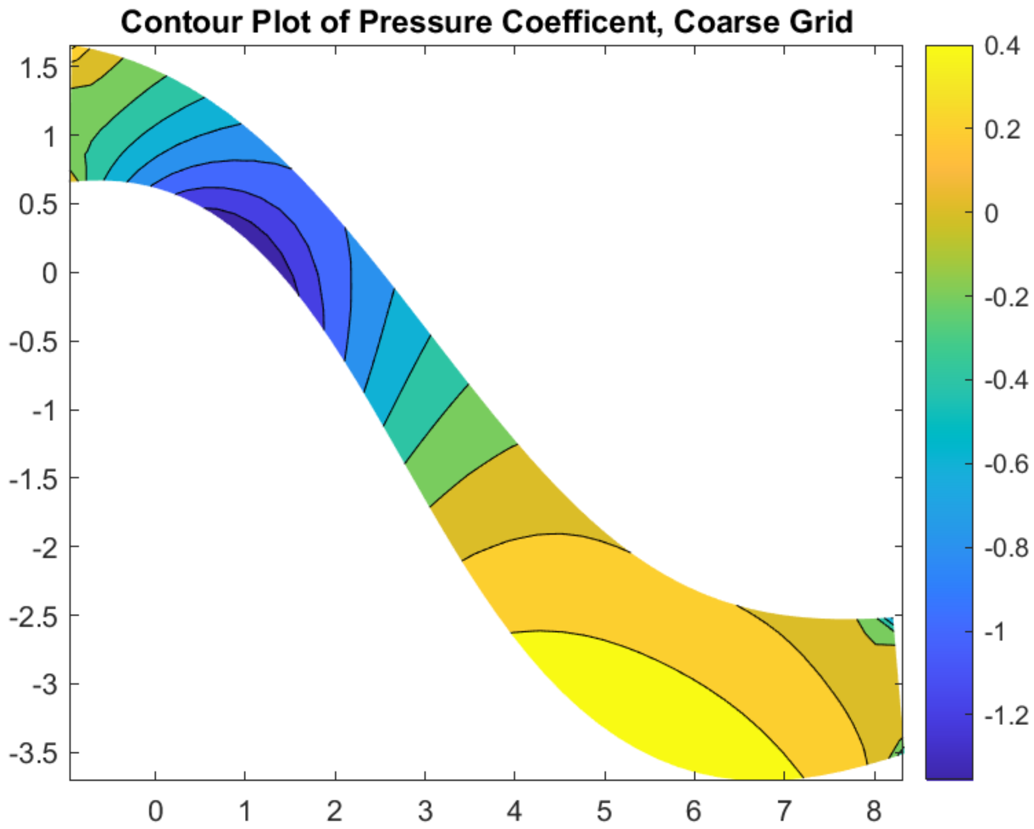

Figure 5: Contour Plot of Pressure Coefficient, Coarse Grid

pressure coefficient value on each.

$$c_p|_{ds} = \frac{c_p|_{i,j} - c_p|_{i,j+1}}{2} \tag{22}$$

The force of each panel is the product of the average pressure coefficient, the linear segment size, and the outward-facing normal vector. The normal vector is found by finding the tangent vector of two points on the boundary, rotating by 90 degrees, and normalizing to get a magnitude of 1. This method of determining normal vector is second-order accurate at the midpoint between the nodes. This calculation is shown below for the case of the top of the pipe when i = 1:

```
    i = 1;
    for j = 1:n-1
       tan_vec = [X(i,j+1),Y(i,j+1)]-[X(i,j),Y(i,j)];
       nv = [tan_vec(2), -tan_vec(1)]; %get normal vector
       nv = nv./(norm(nv)); %normalize
   end
```

After looping through each segment of the walls, we get the total x and y force components from the sum of the force calculations on each panel. The top and bottom wall forces from the coarse and fine grids as displayed below in Table 1.

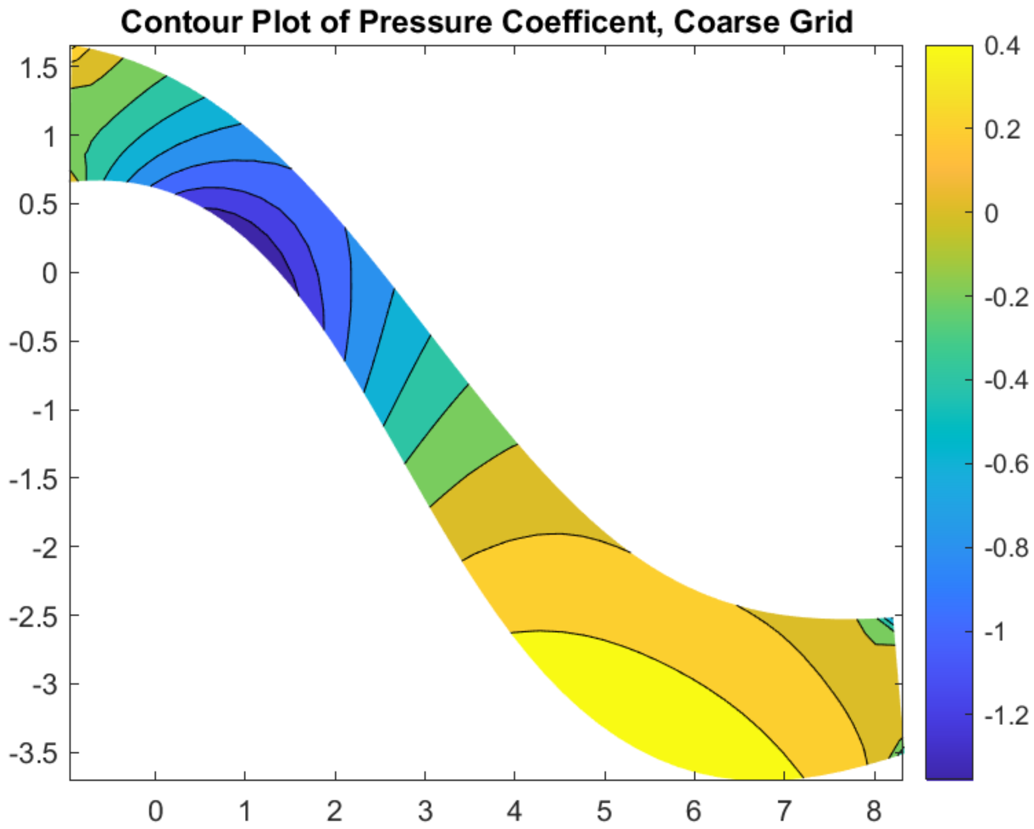

Figure 6: Contour Plot of Pressure Coefficient, Fine Grid

Table 1: Components of Force Vectors acting on Pipe Wall

	Top, Fx	**Top, Fy**	**Bottom, Fx**	**Bottom, Fy**
Coarse	-1.16985	-1.44668	1.18478	1.116062
Fine	-1.17428	-1.43954	1.19048	1.11593

4 Conclusion

This project tasked us with characterizing the flow through the serpentine pipe by mapping to a rectangular reference domain in the space of ξ and η. We then performed a pressure coefficient and force calculation to better visualize the flow through the pipe. By comparing the different contour plots for the derivatives, stream function, and pressure coefficient, we can compare the effects of grid refinements. I observed very little difference in the plots generated for the coarse and fine grids. This leads me to the assumption that numerical error in this method of solving is visually small. Therefore, in this project we demonstrated in the accuracy of finite difference approximation and determining a discrete solution by direct solving. 7