2D Potential Flow in an S Shaped Container

Project Report
from
Sameer Shridhar Desai (2272588)

ISTM
Karlsruher Institut für Technologie
Summer 2020

Contents

1 Introduction

1.1 Potential flow

Potential flow is that type of flow where it is assumed that the velocity is the gradient of the potential. Since the gradient of a derivative is zero the flow is irrotational and the convective term in Navier Stokes equation vanishes and therefore it can be represented with the streamlines. One of the advantages of the potential flow is that one can plot the potential lines and understand the behavior of the flow, since the potential lines are always orthogonal to the streamlines.

We solve the continuity equation to obtain the potential, where in the velocity is substituted with potential which in turn arrives to the Laplace equation,

$$\nabla^2 \phi = 0 \tag{1.1}$$

We solve a boundary value problem and the discretisation method used to solve the above problem is the finite difference method where the values are approximated using the central, forward and backward difference approximations.

If the point lies inside the flow domain and not on the wall interface , central difference approximation and for the wall interface where the non penetrating boundary conditions apply forward and backward difference approximation as shown below are used

$$(\phi_W - 2\phi_P + \phi_E)/(\Delta x^2) + (\phi_W - 2\phi_P + \phi_E)/(\Delta y^2) = 0 \tag{1.2}$$

$$(\phi_{out} - \phi_{in})/(\Delta(x,y)) = 0 \tag{1.3}$$

$$(\phi_{in} - \phi_{out})/(\Delta(x,y)) = 0 \tag{1.4}$$

The boundary conditions for inlet and outlet are Dirichlet type and the wall boundary conditions are given by Neumann type.

1.2 Storage of 2D mesh in the computer memory

The numbering of nodes of a 2D grid in python starts from 0. The numbering is done column wise i.e. if a grid of 10 x 10 is defined, then the first column is numbered from 0 to 9 from down to up, the next column is numbered from 10 to 19 from down to up and so on.

The computer memory stores all the cells in one column and hence each number is stored in each row i.e in the above case the computer memory will have 100 rows with one column.

To access the east or west neighbor of the current point, one can simply add or subtract length of one column respectively, for instance the east and west neighbor of 13 are 23 and 3 respectively and similarly to get the north and south neighbor one can add or subtract 1. For instance, the north and south neighbor of 13 are 14 and 12

Similarly in order to get the coordinate position of the current point, one can take the modulus of the number with the total number of points in the y direction to get y position and divide the number with the total number of points in the y direction to get x position. For instance,

$13 mod 10 = 3 = y$
$13/10 = 1 = x$

2 The Program

2.1 Code design for the geometry

The geometry of the S shaped container has the following inner configurations

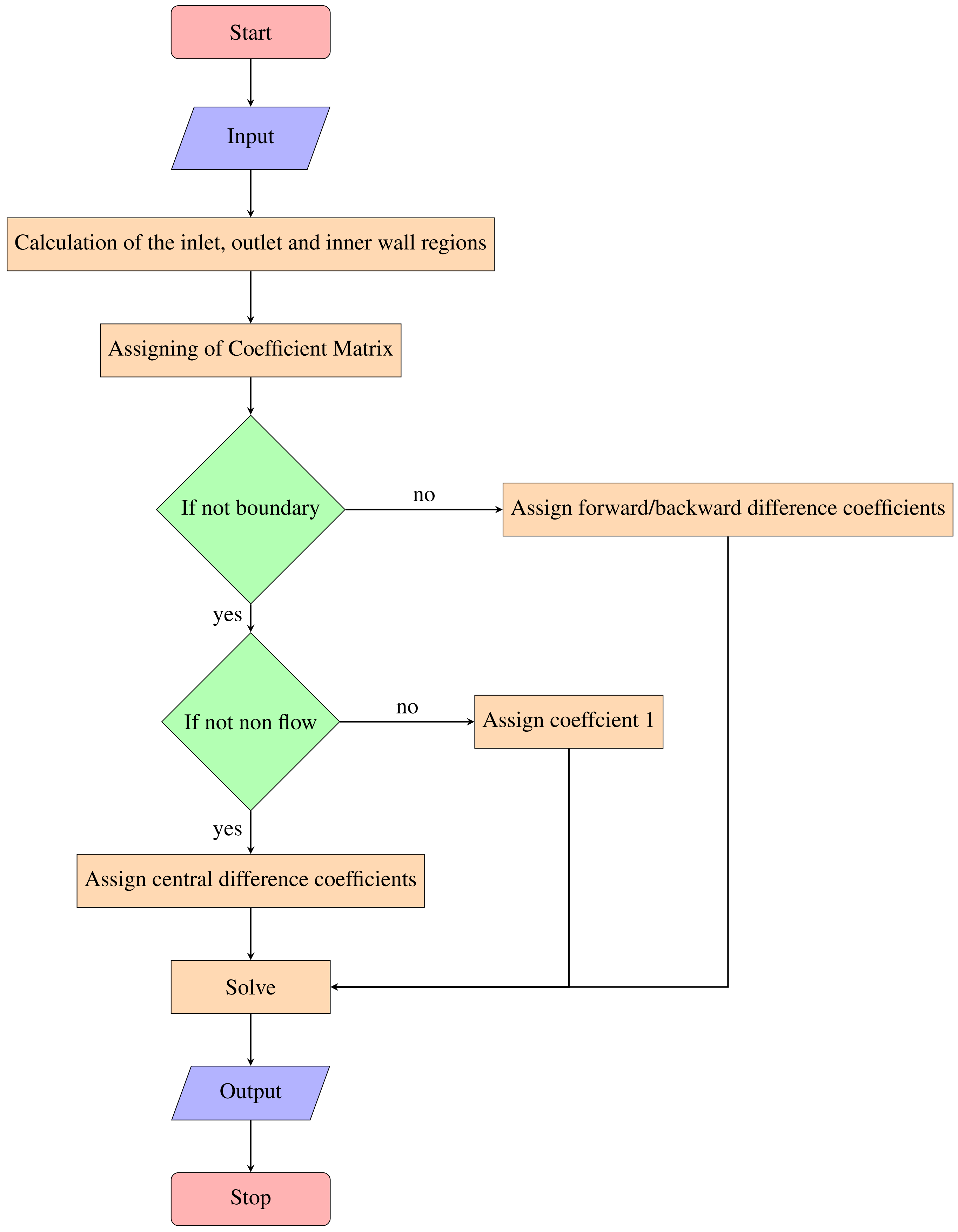
- Inlet
- Rectangular wall projecting inside from the west boundary near the inlet
- Rectangular wall projecting inside from the east boundary near the outlet
- Outlet

The program is designed in such a way that the user has to define only the length, breadth and corresponding number of grid points of the total outer area. Rest of the configurations are defined implicitly proportional to the length and breadth. The dimensions of the inner geometries are as given below

Table 2.1: Geometric Configurations of the S shaped container

Geometry	**parts of total length (ly)**	**parts of total breadth (lx)**
inlet	0.1	0
lower block walls	0.2 to 0.3	0 to 0.5
upper block walls	0.6 to 0.8	0.5 to 1
outlet	0.9 to 1	1

2.2 Program flow


Start
Input
Calculation of the inlet, outlet and inner wall regions
Assigning of Coefficient Matrix
If not boundary
no
Assign forward/backward difference coefficients
yes
If not non flow
no
Assign coeffcient 1
yes
Assign central difference coefficients
Solve
Output
Stop

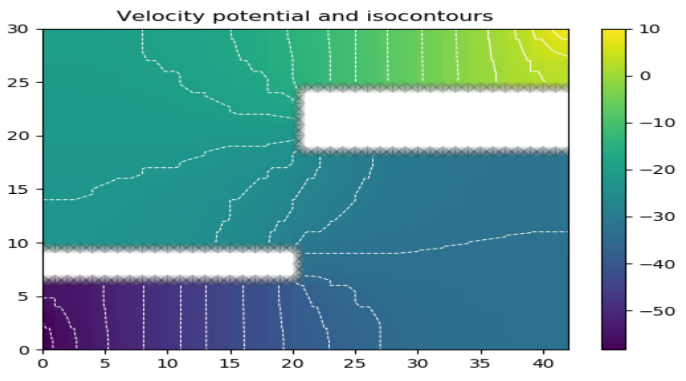
2.3 Structure of the code

The program consists of two modules -

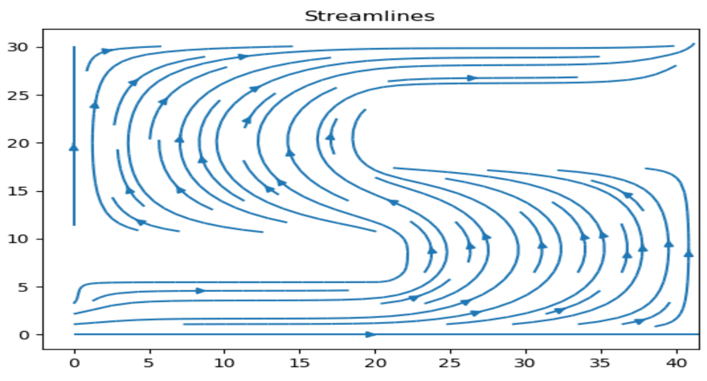
- **potential_flow.py** is the main module which does the following tasks -
  - takes the inputs length, breadth and the number of grid points in horizontal and vertical direction.
  - defines the inlet and outlet positions
  - defines the inner wall region
  - allocates grid points to the coefficient matrix
  - solves the linear equation $A\vec{f} = b$
  - plots the graphs
- **mesh_utils.py** is the supporting class which performs the grid operations and consists of following functions -
  - returns the matrix position or the coordinate position of the current point
  - returns the memory position given the matrix position
  - returns the memory position of the neighboring points i.e. North, South, East and West
  - returns the boundary and their corresponding neighboring point which is on the outgoing normal vector

3 Results and Discussion

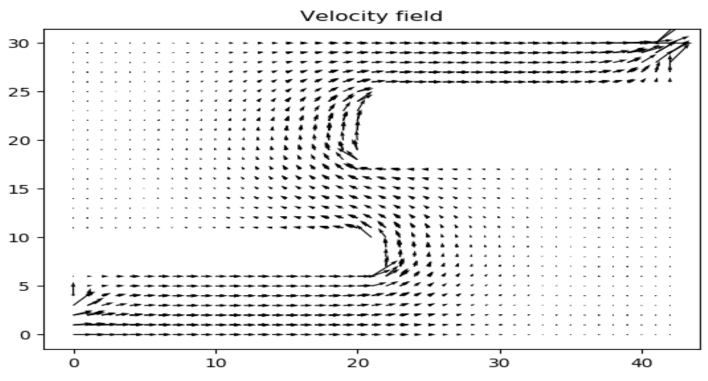
After the post processing the results look as shown below. We can see that the flow is along the S shape of the container. As discussed above, one can see that the isocontours are orthogonal to the streamlines. From the velocity field plot we can see that the velocity is higher where the areas are more constricted.



(a)



(b)



(c)

Figure 3.1: Results

4 Answers to the questions

- The grid points are necessary to solve the problem because only with the grid points we can convert the partial differential equation into a linear equation which will be solved to find the solution.
- The points inside the letter are treated as walls and provided with Dirichlet boundary conditions and hence the wall is having non penetrating boundary condition.
- One of the options to reduce the memory requirements is to carefully choose the number of grid points. However, its always a trade off between accuracy and the memory. Another method can be choosing the type of cells example, triangular, rectangular, etc.
- Since we are using central difference method the accuracy of this method is 2. The accuracy can be determined by the Truncation error in the Taylor series.