Graduate CFD-I: Project Report #4

Dina Soltani Tehrani
dina.soltani@ae.sharif.edu

Sharif University of Technology — February 7, 2022

Contents

Introduction

Laplace equation is an elliptic partial differential equation that governs several simple physical problems such as irrotational incompressible fluid flow named "Potential Flow" and steady state heat transfer in solids. The mathematical formula of a two-dimensional Laplace equation is as follows,

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \tag{1}$$

The formulation from theory for the velocity potential lines in Cylindrical and Cartesian coordinates are as follows, respectively,

$$\phi(\mathrm{r}, \theta) = 5.\left(\frac{a^2}{r} + r\right)\cos(\theta) \tag{2}$$

$$\phi(x, y) = \frac{5.x^2 + 5.y^2 + 0.002}{\sqrt{x^2+y^2}\sqrt{\frac{y^2}{x^2}+1}} \tag{3}$$

The formulation from theory for the velocity potential lines in Cylindrical and Cartesian coordinates are as follows, respectively,

$$\phi(\mathrm{r}, \theta) = 5.\left(r - \frac{a^2}{r}\right)\sin(\theta) \tag{4}$$

$$\phi(x, y) = \frac{5.xy\left(x^2+y^2-0.0004\right)\sqrt{\frac{y^2}{x^2}+1}}{\left(x^2+y^2\right)^{3/2}} \tag{5}$$

The finite difference representation of Laplace equation can be expressed in two methods, the five point stencil and the nine point stencil. The five point stencil method is the most common, and its formula is obtained based on Finite Difference Approximation formulations. The solution of the finite difference form of the Laplace equation is similar to the solution of linear algebraic equations. There exist two methods in order to solve any linear group of algebraic equation. The first method is concerned with direct solution of the system of equation. These methods include Crammer's rule, Gauss elimination and Thomas algorithm of tri-diagonal matrices. Although these methods are very simple, they require huge computational time. Iterative methods, on the other hand, are more efficient than direct methods in most cases since they require far less computational time.

1 Problem Description

The problem is concerned with two-dimensional steady incompressible irrational flow in the region described in 1. The parameter φ here represents the stream function of the potential flow. A uniform grid is constructed with constant spacing $\Delta x = \Delta y = 0.0025m$. The problem domain as well as the computational grid are shown in Figure 2.

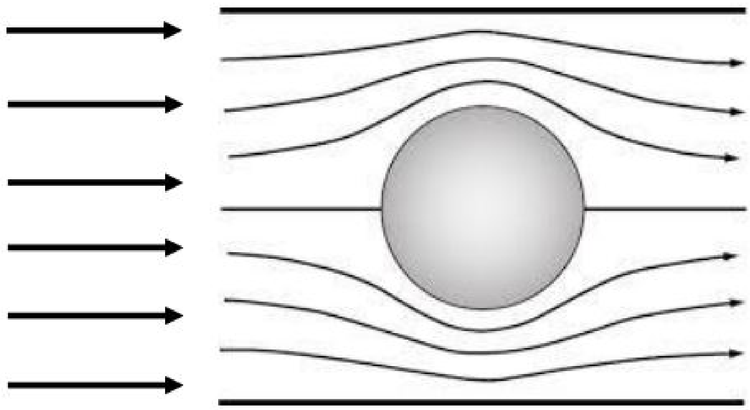


Figure 1: Flow Region From Theoretical Pint of View

1.1 Boundary Conditions

The boundary conditions are defined based on the theoretical solution. Using Mathematica, we have determined the conditions on the boundaries as functions of x in y-constant boundaries and functions of y in x-constant boundaries. The boundary conditions for flow streamlines are as follows, with 1, 2, 3, and 4 representing the left, lower, right, and upper boundaries, respectively.

$$\psi_1(y) = -\frac{0.1y^3\sqrt{2500.y^2+1}}{(y^2+0.0004)^{3/2}} \tag{6}$$

$$\psi_2(x) = 0.0 \tag{7}$$

$$\psi_3(y) = -\frac{0.2y\left(y^2+0.0012\right)\sqrt{625.y^2+1}}{(y^2+0.0016)^{3/2}} \tag{8}$$

$$\psi_4(x) = \frac{0.1x}{\sqrt{\frac{0.0004}{x^2}+1}\sqrt{x^2+0.0004}} \tag{9}$$

The boundary conditions for velocity potential lines are as follows, with 1, 2, 3, and 4 representing the left, lower, right, and upper boundaries, respectively.

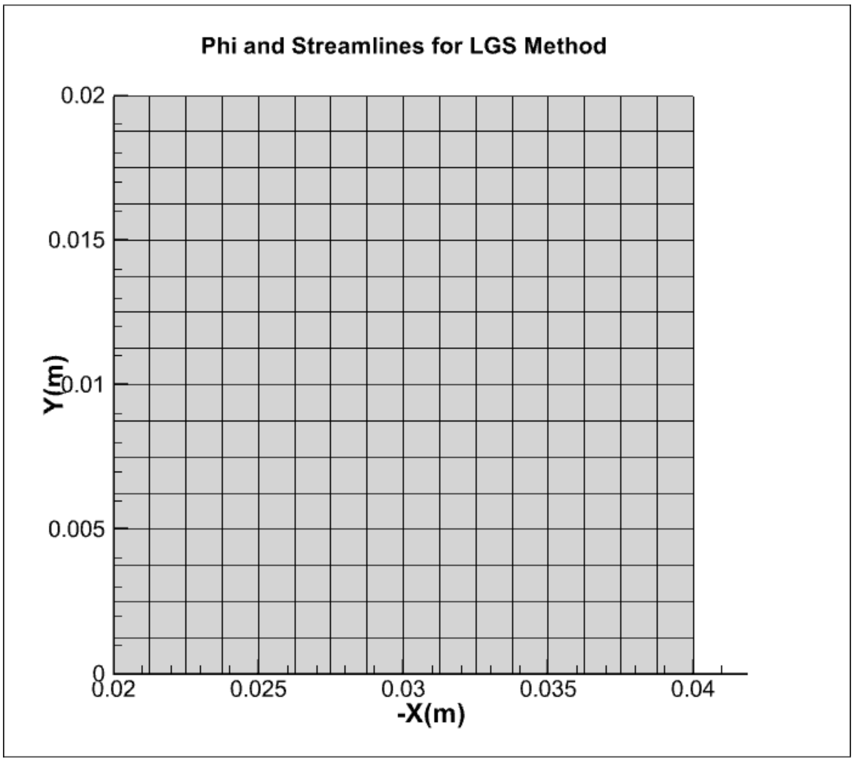


Figure 2: Computational Grid

$$\phi_1(y) = \frac{5.y^2 + 0.004}{\sqrt{y^2 + 0.0004}\sqrt{2500.y^2 + 1}} \tag{10}$$

$$\phi_2(x) = \frac{5.x^2 + 0.002}{\sqrt{x^2}} \tag{11}$$

$$\phi_3(y) = \frac{5.y^2 + 0.01}{\sqrt{y^2 + 0.0016}\sqrt{625.y^2 + 1}} \tag{12}$$

$$\phi_4(x) = \frac{5.x^2 + 0.004}{\sqrt{\frac{0.0004}{x^2} + 1}\sqrt{x^2 + 0.0004}} \tag{13}$$

Note that we needed to define three Dirichlet boundary conditions for the left, right and upper boundaries, and a Neumann boundary condition to maintain the symmetry of the flow field on the symmetry axis (lower boundary.)

2 Iterative Methods

Iterative methods are used to solve linear algebraic equation by assuming an initial value of the computed parameters and then applying the same algorithm for certain number of calculations in order to converge to the final value of the computed parameters. There exist two types of iterative methods, explicit and implicit.

2.1 Jacobi Method (J)

Jacobi method is an explicit iterative method in which at any iteration cycle every point in the domain is computed from the values of its neighboring points at the old iteration cycle. The finite difference formula employed to compute each point is,

$$\varphi_{i,j}^{k} = \frac{\varphi_{i+1,j}^{k} + \varphi_{i-1,j}^{k} + \beta^2(\varphi_{i,j+1}^{k} + \varphi_{i,j-1}^{k})}{2(1+\beta^2)} \tag{14}$$

where $\beta = \frac{\Delta x}{\Delta y}$. Since each element of the right hand side of this equation is at the same iteration cycle (i.e. old iteration), the computations in this method can be parallelized. Therefore, Jacobi method is said to be vectorizable.

2.2 Gauss-Seidel Method (GS)

At a current iterative cycle $k+1$ in the calculation of the point i, j using GS method, the neighboring points $i1$ and $j1$ that proceed the current point i, j are not considered at the old iteration cycle k. Their updated values at the current iteration $k+1$ are used instead of their values at the old iteration k.The finite difference formula that is used in GS iterations is expressed as,

$$\varphi_{i,j}^{k+1} = \frac{\varphi_{i+1,j}^{k} + \varphi_{i-1,j}^{k+1} + \beta^2(\varphi_{i,j+1}^{k} + \varphi_{i,j-1}^{k+1})}{2(1+\beta^2)} \tag{15}$$

GS iterative method is considered more efficient that Jacobi iterative method in terms of the number of iterations. However, it might not be more efficient in terms of computational time since it is not vectorizable.

2.3 Successive Over Relaxation Method (SOR)

In this method, a relaxation factor w is used in order to accelerate the iterative procedure. SOR method applied to Gauss-Seidel method is,

$$\varphi_{i,j}^{k+1} = (1-\omega)\varphi_{i,j}^{k} + \omega\,\frac{\varphi_{i+1,j}^{k} + \varphi_{i-1,j}^{k+1} + \beta^2(\varphi_{i,j+1}^{k} + \varphi_{i,j-1}^{k+1})}{2(1+\beta^2)} \tag{16}$$

If $1 < w < 2$ over relaxation is employed and the solution is accelerated. If $w < 1$ under-relaxation is employed which makes the solution more stable but slower. If $w > 2$ the solution might be unstable. The effect of w on the solution stability and speed is discussed in details in the results and discussion section.

2.4 Point Versus Line Methods

Each of the mentioned methods can be implemented in a Point mode or Line mode. the difference is in the combination of old and new values. That said, in Point methods, we can implement the algorithm explicitly. However, in a Line method, we need to implement the algorithm in two steps. Step 1, solving the difference equation implicitly in one direction, vertical or horizontal, and Step2, sweeping the domain explicitly from left to right.

2.5 Objectives

C++ algorithms have been developed to solve the two-dimensional Laplace equation using five iterative methods, Point Jacobi, Point Gauss-Seidel, Line Gauss-Seidel, Point Successive Over Relaxation, Line Successive Over Relaxation. A flowchart showing the basic solver algorithms is described in figure 3. The norm infinity error is defined as,

$$Error = max\,|\varphi_{i,j}^{k+1} - \varphi_{i,j}^{k}| \tag{17}$$

The stopping criteria is when the error reaches $1E7$.
The whole algorithms are written in C++ language and are appended to this document.
The codes are also available on GitHub git, and will be emailed with this report as well.

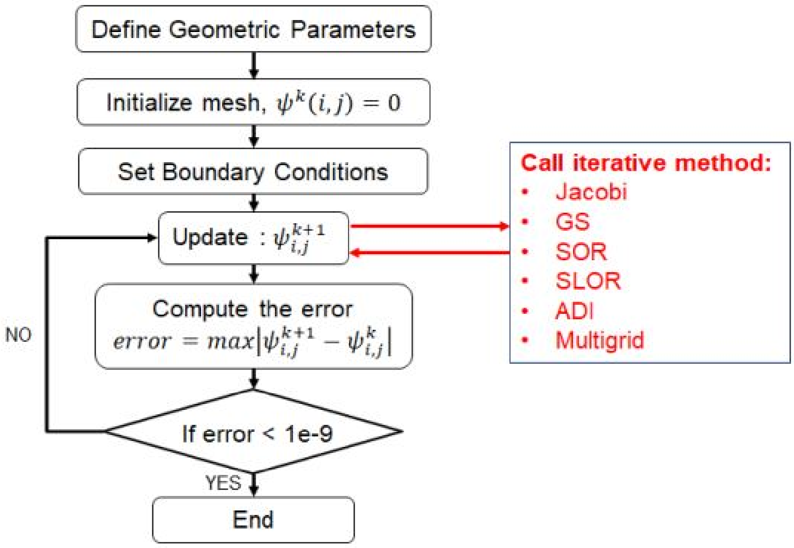


Figure 3: General Solution Algorithm

3 Results and Discussion

In order to fully express the potential flow, both the streamlines and the velocity potential lines were calculated using the six presented methods. Figures 4 represents ψ- and ϕ- constant lines. Moreover, Figures 5 provide ψ and ϕ contours, respectively. These results are based on the Line Successive Over Relaxation Method. The results of other methods will be attached to this report. One may observe that the flow streamlines and the velocity potential lines are perpendicular to each other, as expected.

4 Finding The Optimum Relaxation Factor

To find the optimum relaxation factor for the SOR method, we ran the algorithm for a wide range of relaxation factors, and comapred the convergence criteria in terms of "Run Time" and "Number of Iterations". Figure **??** represents the results.
The results reveal that a relaxation factor around $\omega = 1.2$ would provide the least number of iterations.

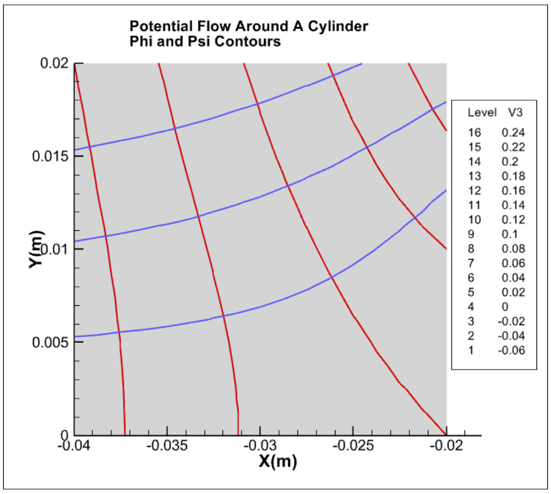
5 Comparison of The SIX Methods

Table **??** summerizes the Run Time results of the six implemented methods.

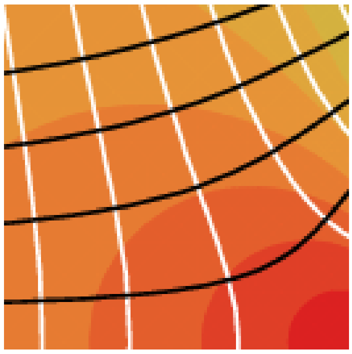
Method	Run Time
PJ	1.390 s
LJ	0.108 s
PGS	3.543 s
LGS	1.420 s
PSOR	3.771 s
LSOR	1.501

Table 1: Summary of Method Run Times

One may notice that Line Methods generally win the game when compared to the Point Methods. Moreover, LSOR seams to be the best method. Although Run Time may not serve as the best predictor of convergence speed, it represents the time consumed acceptably.

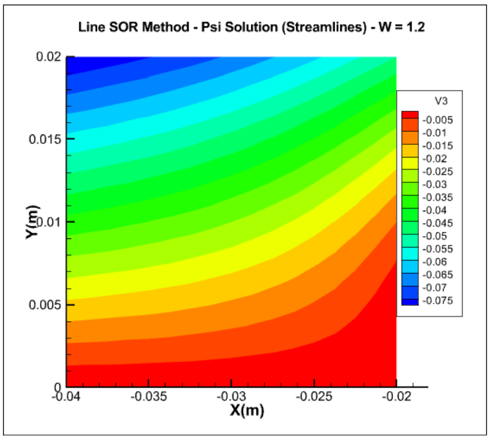


(a)

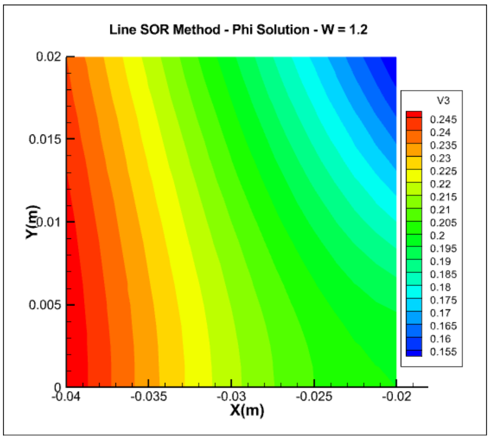


(b)

Figure 4: Comparison of the calculated ψ- and ϕ- constant lines (a) with the valid result(b)



(a)



(b)

Figure 5: ψ (a) and ϕ (b) contours

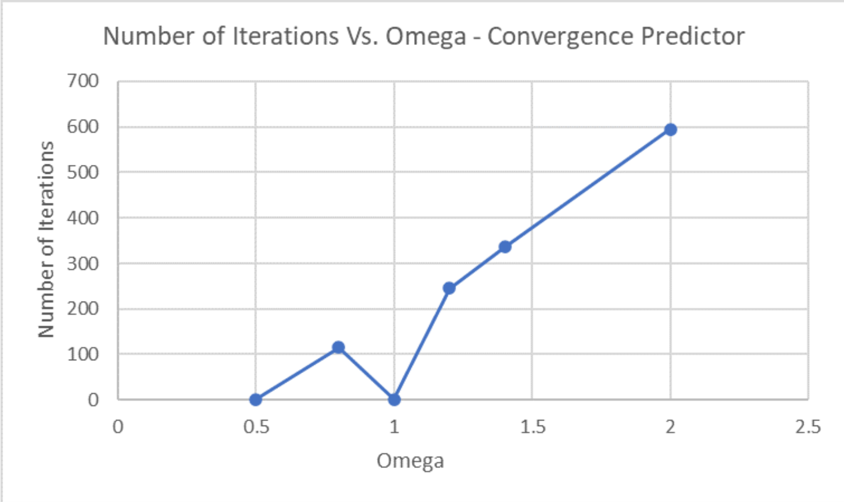


Figure 6: Number of Iterations Versus ω for the Successive Over Relaxation Method

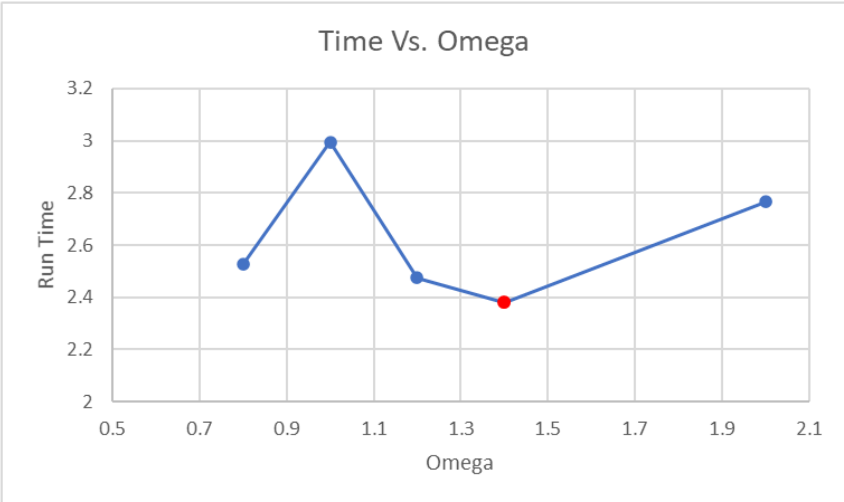


Figure 7: Run Time Versus ω for the Successive Over Relaxation Method

6 References

1. George Lindfield, John Penny, Numerical Methods Using MATLAB, Second Edition, Prentice Hall, 1999, ISBN: 0-13-012641-1, LC: QA297.P45.

2. Simon Fraser University, Computer Science Department, Numerical Methods Resource Website of John Burkardt, John Burkardt, 2020.

3. Advanced CFD Course Lecture Notes by Dr. Taeibi Rahni, Aerospace Engineering Department, Sharif University of Technology, 2021-2022.

4. Hoffmann, Alex C., and Louis E. Stein. "Computational fluid dynamics." Gas Cyclones and Swirl Tubes: Principles, Design and Operation (2008): 139-162.