Contents

NMTFD2

Isa Mammadli
Onkar Marathe

December 19, 2022

1 Introduction

1.1 Problem statement

The goal of the Task 2 is to solve the boundary layer equation over flat plate with Finite Difference methods.

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

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

- Here Reynolds number - $Re = u_\infty L/\nu.$,
  - (ν - kinematic viscosity, L - plate length, u_∞ - fresstream velocity)
- u and v are dimensionless velocities in x and y directions respectively.
- Corresponding dimensional velocities are $\bar{u}$ and $\bar{v}$.
  - (Hence, the relationship between - $u = \bar{u}/u_\infty$ and $v = \bar{v}/v_\infty$)
- Similar normalization were done on lengths, e.g. $x = \bar{x}/L$

1.2 Boundary conditions

Boundary conditions for the problem are stated as: Velocities at boundary layer equal to zero due to no-slip condition.

$$y = 0 \quad u = v = 0 \quad \textit{no slip condition} \tag{3}$$

$$y \to \infty \quad u \to 1\ (\bar{u} \to u_\infty) \quad free\ outer\ flow \tag{4}$$

In addition, one must notice the remaining conditions that arise from how the problem established and are necessary to solve the above equations. Thus, for $u = u(x, y)$, we can write $u(0, y) = 1$ except $u(0, 0) = 0$.

2 Solutions

2.1 Discretize the equation (1).

In our code implementation, we used the discretization suggested in the task sheet. In terms of v, the backward difference scheme was used basing on the available boundary conditions.
Note: Indices were chosen to be i for y direction and j for x direction.

$$\left(\frac{\partial u}{\partial x}\right)_{i,j} = \frac{1}{2}\left(\frac{u_{i,j+1} - u_{i,j}}{\Delta x} + \frac{u_{i-1,j+1} - u_{i-1,j}}{\Delta x}\right) \tag{5}$$

$$\left(\frac{\partial v}{\partial y}\right)_{i,j} = \left(\frac{v_{i,j+1} - v_{i-1,j+1}}{\Delta y}\right) \tag{6}$$

Substituting equations (5) and (6) in equation (1) and solving for $v_{i,j+1}$:

$$v_{i,j+1} = v_{i-1,j+1} - 0.5\frac{\Delta y}{\Delta x}\left(\frac{u_{i,j+1} - u_{i,j}}{\Delta x} + \frac{u_{i-1,j+1} - u_{i-1,j}}{\Delta x}\right) \tag{7}$$

Equation (7) will be used to compute v at each node. In order to do this we will need values of $u_{i,j+1}$. This will be handled in the following section with the help of equation 2.

2.2 Discretize the equation (2)

At this point x derivate can be approximated with forward difference scheme, while in y derivative central differencing will be employed for better accuracy.

$$\left(\frac{\partial u}{\partial x}\right)_{i,j} = \left(\frac{u_{i,j+1} - u_{i,j}}{\Delta x}\right) \tag{8}$$

$$\left(\frac{\partial u}{\partial y}\right)_{i,j} = \left(\frac{u_{i+1,j} - u_{i-1,j}}{2\Delta y}\right) \tag{9}$$

$$\left(\frac{\partial^2 u}{\partial y^2}\right)_{i,j} = \left(\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta y^2}\right) \tag{10}$$

Substituting above equations to equation (2) and solving for $u_{i,j+1}$.

$$\begin{aligned} u_{i,j+1} = u_{i,j} + \frac{1}{Re}\left(\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{u_{i,j}}\right)\frac{\Delta x}{\Delta y^2} \\ - v_{i,j}\left(\frac{u_{i+1,j} - u_{i-1,j}}{2u_{i,j}}\right)\frac{\Delta x}{\Delta y} \end{aligned} \tag{11}$$

Equation (11) is explicit and the first order accurate in x direction due to the use of forward difference scheme and the second order accurate in y direction due to central difference scheme. The numerical scheme is conditionally stable with the condition being $\Delta x \leq \frac{1}{2}\frac{u_{i,j}(\Delta y)^2}{\nu}$, (Reference 1) but we will come back to stability with grid sensitivity plots in the next sections.

2.3 MATLAB program

Given dimensions: $0 \le x \le 1$ and $0 \le y \le 2\delta$, where $\delta = \frac{5}{\sqrt{Re}}$ at boundary layer thickness $x = 1$. The main part of the code lies in iteration in x and y directions to calculate the velocity values (u and v) in the whole domain. As mentioned earlier, the eq. (11) will be used to compute $u_{i,j+1}$ first, and then the result will be inputted to eq. (7) to obtain $v_{i,j+1}$. One key point about the eq. (11) is that, the $u_{i,j+1}$ requires the value of $u_{i+1,j}$. Therefore, the inner loop must compute the values in the y-direction first, so the outer loop can then use these values to march in x-direction.

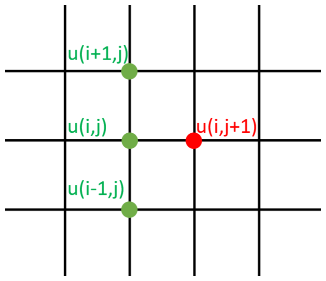

Figure 1: Computation of unknown $u(i, j + 1)$ (red) from known neighboring values (green).

2.4 Numerical results of velocities.

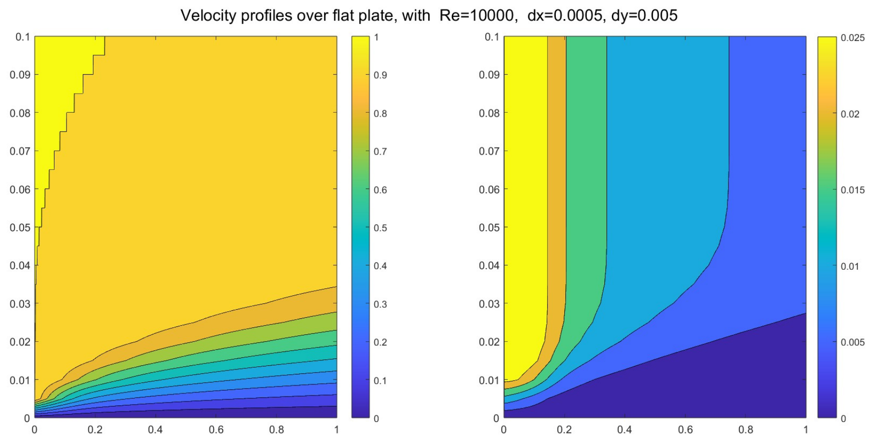

Figure 2: u (left) and v (right) profiles over whole domain.

Above given figure shows (left) how the boundary layer develops (increasing thickness) from left to right, causing decreased velocities at near bottom boundary. Initially to the left, free stream velocity still is unaffected by this development but this drag becomes increasingly effective, forming the final velocity profile. In the previous section it was mentioned that, the discretization scheme is conditionally stable and grid size in x direction should be less than certain limit which is determined by grid size in y direction, Reynolds and $u_{i,j}$. As a simple numerical stability analysis, different grid sizes in x direction were used for the iterations with fixed Δy and Re. Following that averages of velocities over whole domain were deduced and cross-plotted with gridsizes (Δx) to understand convergence.

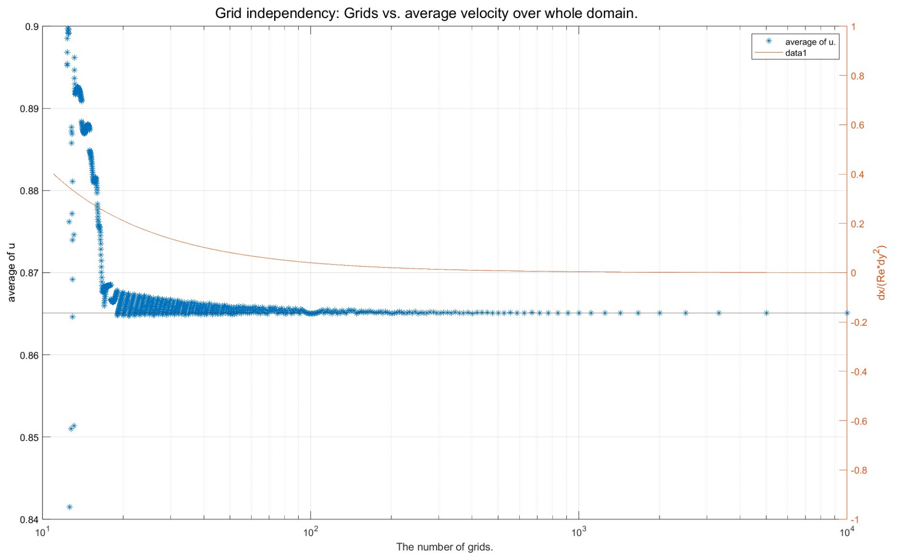

Figure 3: Grid independence study with mean of u.

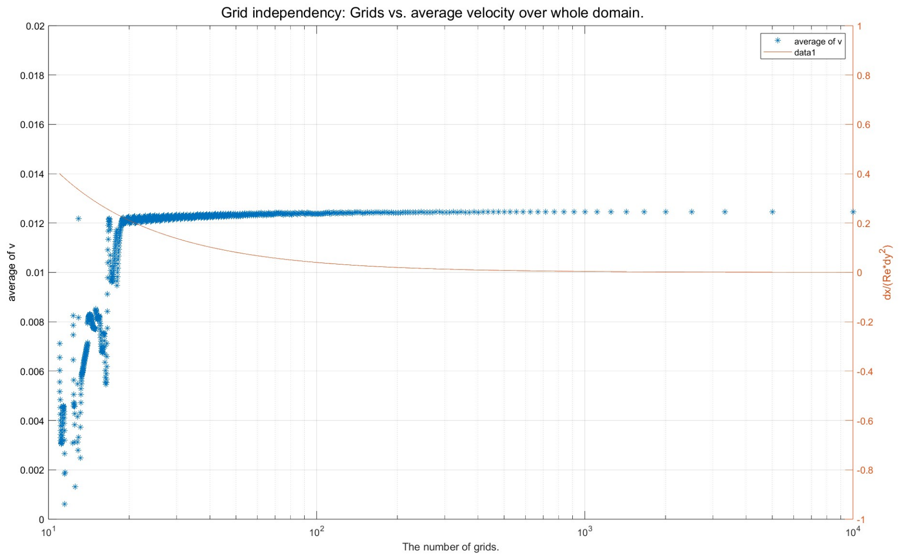

Figure 4: Grid independence study with mean of v.

2.5 Comparison with Blasius

Numerical solutions are finally compared with Blasius solutions for quality check. Note: MATLAB codes of Blasius method, were readily taken from Ref.3. Generally, it is known that Blasius method provides good approximation to laminar boundary layer problem at $Re < 5 \times 10^5$. Hence if we take this as a reference case, it can be inferred that, numerical solution provides fairly good results, with some variations. While u values in the first figure agrees with Blasius solution quite well, first order accurate scheme in estimations of v might have contributed to the overestimated values in numerical side. Especially, at x=0.5 (away from boundary) numerical values show developed velocity profiles, but at near boundary location of x=0.0005, first few data points do not exactly match the Blasius solution. In this case, since the chosen x=0.0005 exactly equals Δx - grid size, the discreet point falls away from well refined Blasius solution.

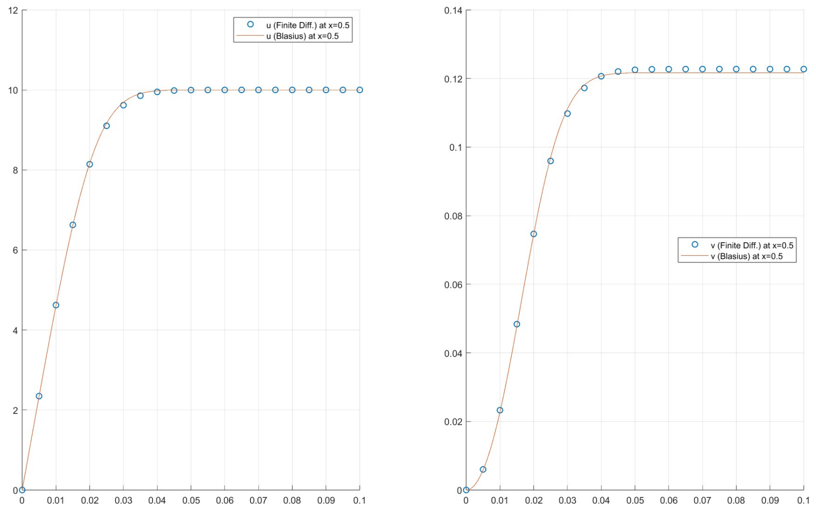

Figure 5: Comparison at x=0.5. Circles are Numerical solutions at available y values, while orange line is Blasius solution.

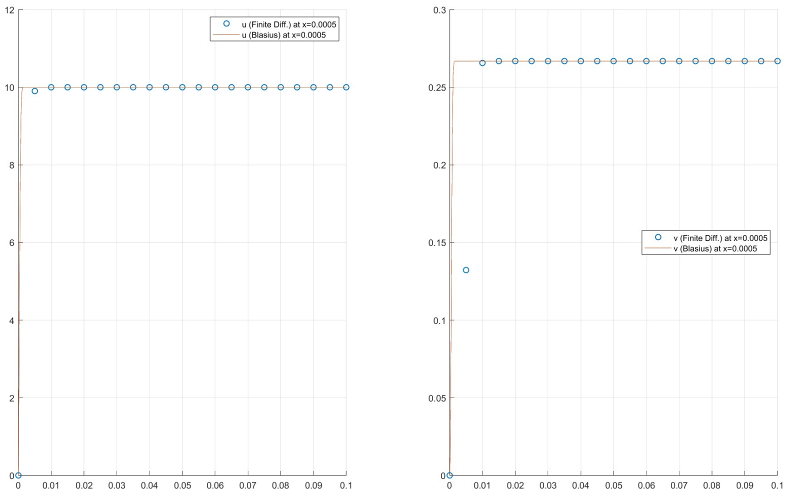

Figure 6: Comparison at x=0.0005.

From the preliminary promising results of the numerical solution, it can be recommended that numerical methods for boundary layer flow is worth pursuing, but computational cost can increase quite quickly due to the conditional stability of problem.

3 References

1. Boundary layer over flat plate, Bsc report, available at: $https://essay.utwente.nl/63314/1/BSc_{r}eport_{P}eter_{P}uttkammer.pdf$

2. J.H. Ferziger and M. Peric. Computational Methods for Fluid Dynamics. Springer Berlin Heidelberg, 2012.

3. Codes for Blasius method: Solving Blasius Equation Using RK-4 Numerical method, available at: https://de.mathworks.com/matlabcentral/fileexchange/102189-solving-blasius-equation-using-rk-4-numerical-method