Numerical Simulation of 1D Coutte Flow

Gurpal Singh

February 7, 2017

Abstract

Coutte flow describes the fluid motion between two plates that are separated by a small distance. The flow occurs between one stationary plate and one moving plate. In this simulation the velocity profile of the fluid flow is calculated using a finite difference method and an exact solution.

1 Introduction

The velocity profiles of the Coutte flow will be first calculated using a finite difference method which gives a reasonably well approximation. To confirm the accuracy of this method, the velocity profile will also be computed using the exact method which stems down from the governing Navier-Stokes equation.

2 Calculation of Velocity Profiles

This section discusses the specifics of the finite difference method and the exact solution used to calculate the velocity profile. Before jumping into the methods used, it is useful to describe the set up of the problem. The bottom plate was stationary, while the top plate was moving at a velocity that is determined by the user defined Reynolds Number. The distance between the plates is one meter and there are 21 intervals in the y direction. The pressure gradient and Reynolds number are defined by the user. The fluid being analyzed was assumed to be water. The velocity of the top plate was assumed to be 1 meter per second.

2.1 Finite Difference Method

The Finite Difference approach uses the points below and above the point in question to determine the velocity profile for a time step n+1. The n+1 values are stored temporarily and then once the program has marched through the n time step, the n+1 values become the n time step values. This way, an iterative scheme is applied to calculate the velocity profile until steady state is reached. It is important to note that the boundary conditions can not be touched as they are the last points on the spectrum in the y-direction so including points below

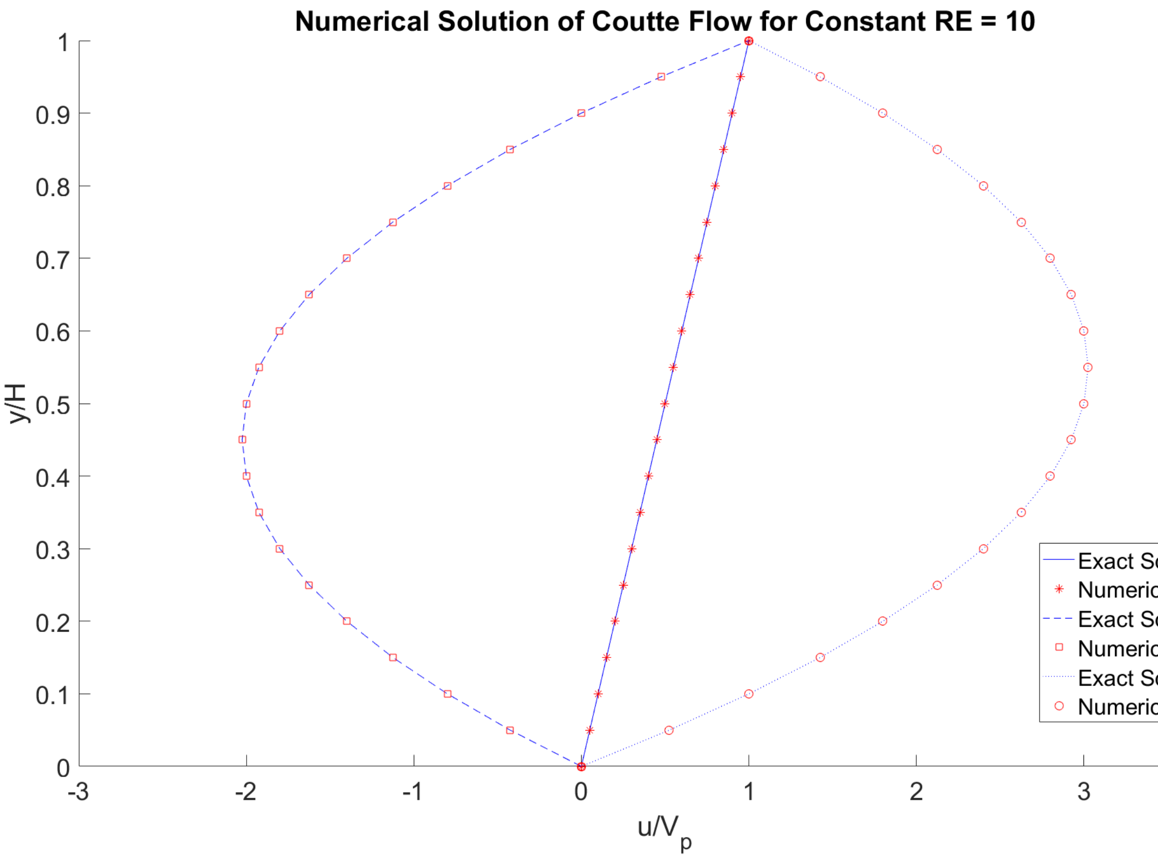

Figure 1: Velocity profile for constant Reynolds Number and changing Pressure Gradient

or above the boundary is unreasonable. Before jumping into the equations used, the characteristics of the fluid must be defined.

Once the user has defined the Reynolds Number and pressure gradient, the dynamic viscosity must be obtained.

$$\mu = \rho * \nu \tag{1}$$

Next, the Reynolds Number must be satisfied which helps determine ν.

$$Re = \frac{U_p * H}{\nu} \tag{2}$$

The maximum time step is defined as follows.

$$\Delta t \leq 0.5[\frac{(\Delta y)^2}{\nu}] \tag{3}$$

The time to reach steady state is calculated to help determine the number of timesteps needed.

$$T_{ss} = \frac{H^2}{\nu} \tag{4}$$

The finite difference formula that will be used to compute the velocity profile is as follows. The subscripts n and j represent time and position in the y-direction respectively.

$$u_j^{n+1} = \Delta t * [\frac{-1}{\rho} * \frac{dP}{dx} + \frac{u_{j+1}^n - 2 * u_j^n + u_{j-1}^n}{(\Delta y)^2}] \tag{5}$$

2.2 Exact Solution

The exact solution is determined using the same values for the characteristics of the fluid used in the finite difference method. For this method, there is no time march required. The solution is calculated via the following formula. The time to reach steady state is calculated to help determine the number of timesteps needed.

$$u(y) = V_{plate}(\frac{y}{H}) + \frac{1}{2\mu}(\frac{dP}{dx})(y^2 - Hy) \tag{6}$$

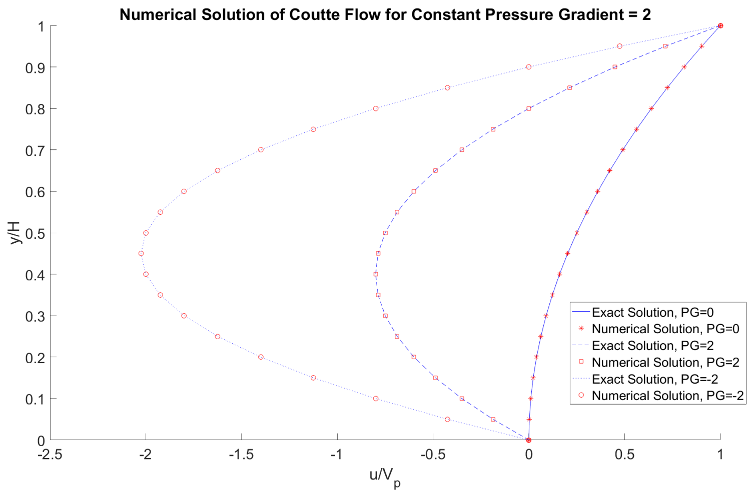

Figure 2: Velocity profile for constant Pressure Gradient and changing Reynolds Number

3 Results

The finite difference method and exact solution method were used to calculate velocity profiles for flows with different Reynolds Numbers and pressure gradients. First, the Reynold's number was kept constant and the pressure gradient was varied. Second, the pressure gradient was kept constant and the Reynolds Number was varied. The data was then plotted to show the effects of varying the Reynolds Number and the pressure gradient. The computations were done in code written with the C++ language[1].

4 Acknowledgments

References

[1] B. Stroustrup, *The C++ Programming Language*. Addison-Wesley, 2014.