Couette Flow Analysis Using Computational Methods
Fluid Mechanics Spring 2022

Alex Tschinkel

April 21, 2022

Abstract

The analysis of turbulent couette flow between two plates, beginning with an unsteady turbulent case, and continuing until a steady state flow was reached, was examined computationally. This was done using Python 3 in a Jupyter notebook. It was found that the behavior of the flow computationally computed was very similar to the exact solution to the differential equation given by Navier-Stokes. It was similarly found that the as a finer mesh was used, and as smaller time steps were taken, the results became increasingly precise.

1 Introduction

Couette flow is considered to be the flow between two parallel plates. When examining this flow for a viscous, Newtonian fluid, we can use the Navier-Stokes equation, which gives us, under the fully developed assumption, the following equation:

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial y^2} \tag{1}$$

which we can apply to a stream velocity $u(y,t)$ and has to satisfy the boundary conditions specified by the no slip condition, namely that u must be equal to the velocities of both of the plates at the points where the plates interface with the liquid, or:

$$u(0,t) = 0, u(h,t) = U \tag{2}$$

Now, removing the dimensions from these equations, such that $y = y/h$, $u = u/U$, and $t = \nu t/h^2$ allows us to compute this differential equation.

2 Methods

This calculation was performed using Python in a Jupyter Notebook. This was done by initializing a mesh, setting N points equally spaced in the height gap,

ie. between 0 and 1, and generating an initial, turbulent velocity profile for these points:

$$u(y,0) = U\frac{y}{h} + U\sin\frac{\pi y}{h} \tag{3}$$

Using the finite difference approximation, the second order central difference was used for the spacial derivative, and the first order backward difference was used for the time derivative when solving the PDE, yielding:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{(\Delta y)^2} \tag{4}$$

with n being the n^{th} step forward in time. This system of equations is then expressed as a tridiagonal matrix A of coefficients, multiplied by a vector x containing all of the u_j^{n+1} terms and equal to another vector b composed of u_j^n terms. Solving this equation for x the velocities in the mesh at the time Δt after b involves inverting the matrix A, and multiplying, such that $x = A^{-1}b$.

This matrix inversion was done using a Thomas algorithm that was found online (and is linked in the code). The exact solution to the PDE:

$$u(y,t) = y + exp(-\pi^2 t)\sin(\pi y) \tag{5}$$

was then simultaneously computed for the same time values, and used for error analysis. The steady state solution, a simple line, was also computed for each of the time steps, and was used to terminate the calculation once the flow was within a small margin of error of the steady state.

3 Results

It was found that the turbulent flow consistently approached the steady state case in under 1.4 seconds, and was below our error threshold of 10^{-6} at approximately 1.38 seconds for each of the configurations. It was also found that the temporal resolution had a larger impact on accuracy than the spatial resolution, although for all cases, increased resolution led to decreased error. As seen in the first graph below, as the number of points used in the mesh increased, the magnitude of the error decreased. This, however was not a linear decrease, as a point was reached where a substantial bump in resolution was needed to achieve a minimal increase in precision.

A similar trend can also be seen for temporal resolutions in the second figure below. Since the temporal resolutions used for this project were orders of magnitude apart, a far more exaggerated effect can be seen. The largest resolution is, in fact, great enough that the graph appears linear in some locations.

The approach of the calculated solution of the settling of the turbulent state as it approached the steady state solution of a straight line was similarly dependant on the resolutions used. The error decreased much more quickly for better resolutions. As seen in the Figure 3, the rate of approach for varying spatial resolutions is very closely exponential, which matches our exact solution. The increased resolution, $N = 81$ also is visibly the most precise.

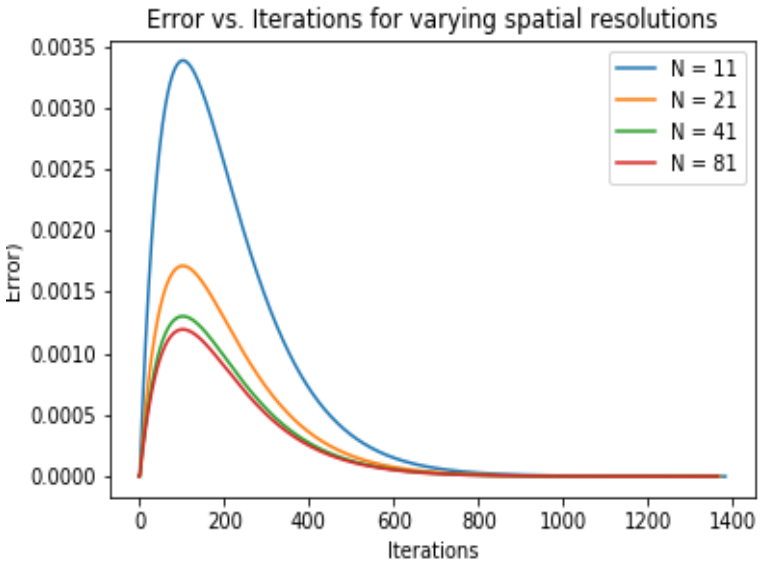

Figure 1: Error as a Function of Spatial Resolution

4 Conclusion

Thus, it was found that the exact solution to the Navier Stokes equation for Couette flow very closely matched the computational solution. As expected, the precision of the results was always improved by improving either the spatial or temporal resolutions. As time progressed, and more time steps were taken, the error decreased, although it generally spiked shortly after the problem was initialized.

One potential method for improving the precision of the computational method, other than increasing spatial or temporal resolution, would be to use a different method to computationally evaluate derivatives. This would change the properties of the matrix A. Another potential topic to explore with this method, since we have found it to be fairly accurate, would be to explore other initial conditions, particularly ones were we do not know the exact solution to the PDE.

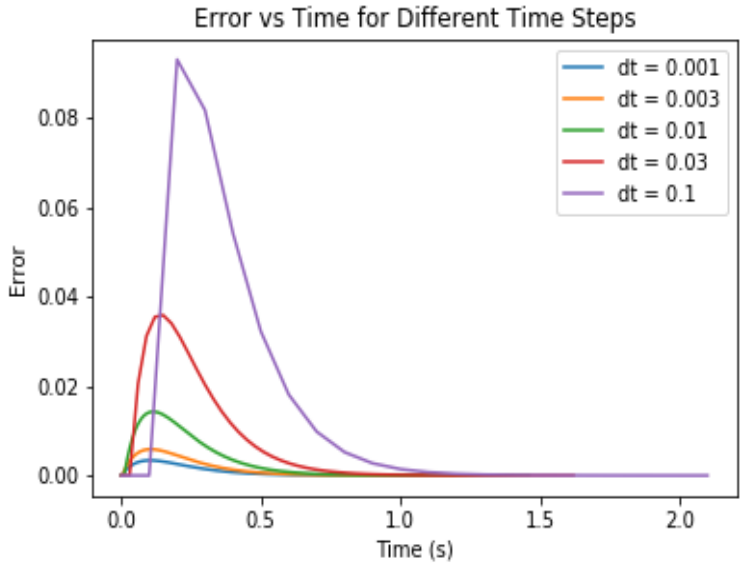

Figure 2: Error as a Function of Temporal Resolution

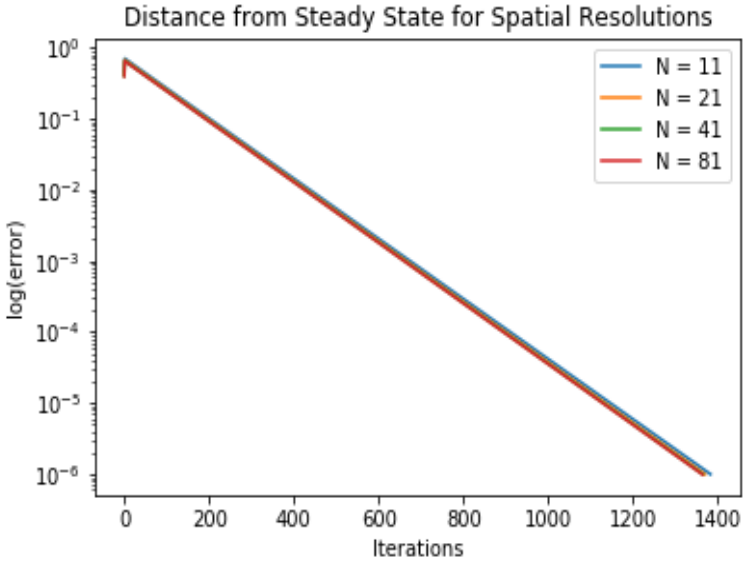

Figure 3: Distance From Steady State for Different Spatial Resolutions

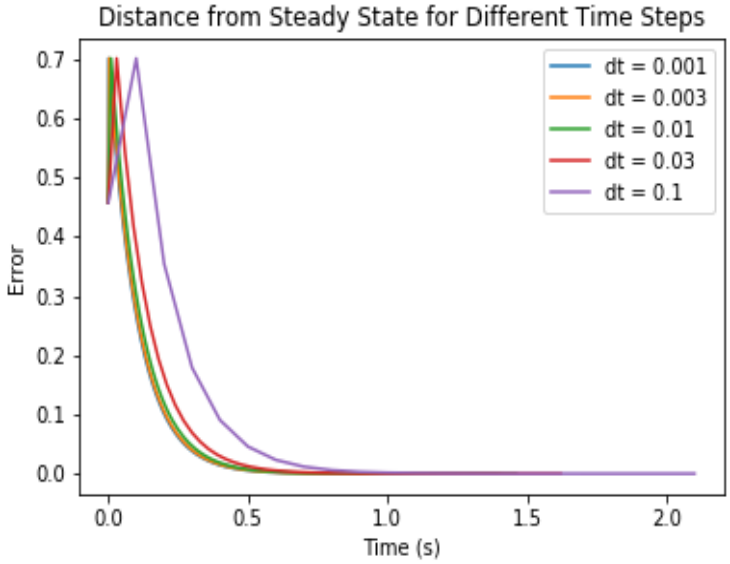

Figure 4: Distance from steady state for Different Temporal Resolutions