Implementation of a Pressure Poisson Equation method for Plane Couette Flow

Jonathan Halcrow

Georgia Institute of Technology[*]

This paper discusses my work through December 2004 to implement Johnston and Liu's Pressure Poisson Equation (PPE) method[1] for Plane Couette flow. Also, I discuss some results and plans for implementing a periodic orbit search. The advantage of this method is that it decouples the computation of velocity and pressure at each time step by treating the pressure term explicitly in time. The project is my first step in designing a program to search for periodic orbits in Plane Couette flow, using variational methods.

I. INTRODUCTION

Johnston and Liu decouple the calculation of pressure from the velocity of each step by transforming the standard incompressible Navier-Stokes formulation:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} = -\nabla p + \frac{1}{Re}\nabla^2\mathbf{u} + \mathbf{f} \tag{1a}$$

$$\nabla\cdot\mathbf{u} = 0 \tag{1b}$$

$$\mathbf{u}|_\Gamma = 0 \tag{1c}$$

into a new formulation with the incompressibility condition replaced with a Poisson equation for the pressure with Neumann boundary conditions.

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} = -\nabla p + \frac{1}{Re}\nabla^2\mathbf{u} + \mathbf{f} \tag{2a}$$

$$\nabla^2 p = -\nabla\cdot(\mathbf{u}\cdot\nabla\mathbf{u}) + \nabla\cdot\mathbf{f} \tag{2b}$$

$$\frac{\partial p}{\partial \mathbf{n}}\Big|_\Gamma = [-\nu\mathbf{n}\cdot(\nabla\times\nabla\times\mathbf{u}) + \mathbf{n}\cdot\mathbf{f}]|_\Gamma \tag{2c}$$

$$\mathbf{u}|_\Gamma = 0 \tag{2d}$$

[*]Electronic address: `gte899j@prism.gatech.edu`

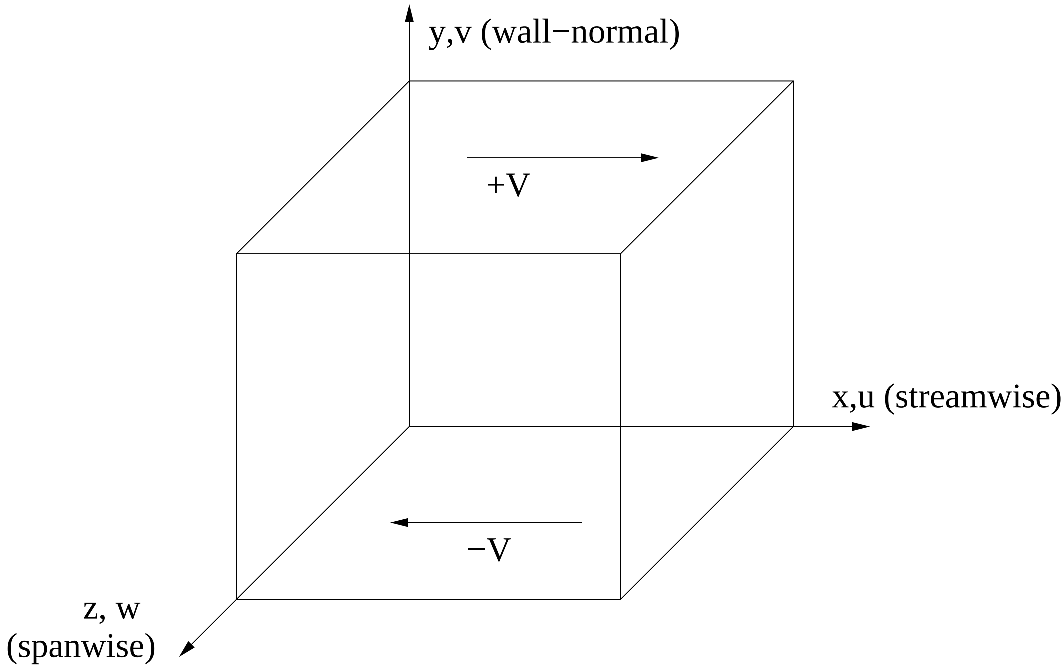

FIG. 1: Coordinate System

The Plane Couette system is made up of two infinite parallel plates moving in opposite directions at constant velocity with the space between filled by an incompressible fluid. To deal with the infinite extent of the system, I treat the system as if it is tiled by periodic boxes. Figure 1 indicates the geometry of the system, with u,v and w being the velocities corresponding to the x, y, and z directions. The boundary conditions in this system do not match those in the formulation which I am using, so the velocities will be computed as deviations from the laminar solution $\mathbf{U_L} = \frac{Vy}{L}\hat{x}$, where L is the separation between the moving plates and V being the speed of the plates. Since this is a solution to Navier-Stokes, the linear terms can be decomposed into the laminar solution and deviations. In fact, all of the linear terms in this expanded Navier-Stokes are zero. However, this requires that the nonlinear convective term, $(\mathbf{u} \cdot \nabla)\mathbf{u}$, needs to be treated including the laminar solution.

II. ALGORITHM

Integration is done by Chebyshev collocation in the y-direction and Fourier expansion in the x and z directions. More specifically, the velocities and pressure are evaluated on a grid of l chebyshev points in the y direction and an $m \times n$ square grid in the periodic x and z

directions. Derivatives in the y direction are taken by multiplying by a Chebyshev matrix and derivatives in the x and z are taken by multiplying the Fourier transformed data by a diagonal matrix with the wave numbers on the diagonal. Temporal discretization is done with Crank-Nicholson for the viscous term and Adams-Bashforth for the convective term and pressure terms. Using capital letters to represent their nondiscretized counterparts and $\mathcal{D}_{x,y,x}$ indicate the derivative matrices in the x, y, and z directions. Note that U, V, W, and P are actually three dimensional arrays. The derivative matrices are "transposed" as if they were acting on column vectors where the other two physical directions are held constant. In order to avoid the extra notation, this is to be assumed in products of differentiation matrices with the arrays. The Adams-Bashforth discretization is written as $A^{(n+\frac{1}{2})} = \frac{3}{2}A^n - \frac{1}{2}A^{n-1}$, which is different from $A^{n+\frac{1}{2}}$. Discretization of 2 for u gives

$$\begin{aligned}\left(\mathcal{I} - \frac{\nu\Delta t}{2}\nabla^2\right) U^{n+1} = U^n - \Delta t\Big[&\big((U+U_L).*(\mathcal{D}_x(U+U_L)) + V.*(\mathcal{D}_y(U+U_L)) \\ &+ W.*(\mathcal{D}_z(U+U_L)) + \mathcal{D}_xP\big)^{(n+\frac{1}{2})} - \frac{\nu}{2}\nabla^2 U^n - F_x^{n+\frac{1}{2}}\Big]\end{aligned} \tag{3a}$$

$$U^{n+1}|_\Gamma = 0 \tag{3b}$$

Note the use for MATLAB style notation for the products in the nonlinear term. These are intended to mean $(A.*B)_{i,j.k} = a_{i,j,k} * b_{i,j.k}$. Since the fourier transforms do not commute with this operation, the velocity field must be transformed back to physical space in order to do the computation in practice.

In order to apply the boundary condition, we set the value of U^{n+1} to match and then solve for the interior. Taking G to be the righthand side of equ 3a, and $\widetilde{A}$ to represent A in the whole space without the walls. This gives

$$\widetilde{U} - \frac{\nu\Delta t}{2}\left(\widetilde{\mathcal{D}}_x^2\widetilde{U} + \widetilde{\mathcal{D}}_y^2\widetilde{U} + \widetilde{\mathcal{D}}_z^2\widetilde{U}\right) = \widetilde{G} \tag{4}$$

Now, we diagonalize $\mathcal{D}_y^2 = \mathcal{P}^T\Lambda\mathcal{P}$, and define $\widehat{U} = (\mathcal{P}^{-1})\widetilde{U}(\mathcal{P}^T)^{-1}$ and $\widehat{G} = \mathcal{P}^{-1}\widetilde{G}(\mathcal{P}^T)^{-1}$. This gives

$$\widehat{U}_{i,j,k} = \frac{\widehat{G}_{i,j.k}}{1 - (\nu\Delta t/2)(\lambda_i + \gamma_j + \gamma_k)} \tag{5}$$

Where λ_i is the i-th eigenvalue of $\mathcal{D}_y^2$ and γ_j and γ_k are the Fourier wavenumbers. The equation for pressure is discretized similarly. However, the method for solving for P^{n+1} requires a slightly different method owing to the Neumann boundary conditions. First, the boundary conditions are calculated using the same discretization and differentiation matrices. Then $\mathcal{D}_y^2$, and the right hand side of the pressure Poisson equation must be adjusted according to the method laid out in [2]. After this is accomplished, the same diagonalization procedure as above is carried out. However, the new $\mathcal{D}_y^2$ has one eigenvalue which is zero (owing the fact that the Neumann boundary conditions leave the solution unique up to a constant). So, we choose to set the pressure corresponding to that point (the $\widehat{P}_{1,1,1}$ point) to be zero. After this procedure is completed, we now have a new pressure and velocity field at the next time step and the process may be repeated to find solutions at subsequent time steps.

This is implemented in FORTRAN 77 with the FFTW library used for the fourier transforms and LAPACK/BLAS for the linear algebra.

III. RESULTS

To check the accuracy of this method, I used a forced exact solution. The solution used is:

$$u(x,y,z) = 2y(1-y^2)\sin(x)\cos(z) \tag{6}$$

$$v(x,y,z) = (1-y^2)\cos(x)\cos(z) \tag{7}$$

$$w(x,y,z) = 2y(1-y^2)\cos(x)\sin(z) \tag{8}$$

$$p(x,y,z) = e^y\sin(x)\sin(z) \tag{9}$$

Then the body force is calculated to make this a solution to Navier-Stokes, with and without the walls moving. Running the simulation for a large number of time steps at various Reynolds numbers shows a reasonable degree of accuracy. To check the accuracy of an unforced solution, I calculated the L_∞ norm of the divergence at various Reynolds numbers. In order to do this well, I need a valid set of initial conditions. Using the forced solution as a starting point (which does not satisfy Navier-Stokes), the divergence does not remain zero. However, it seems to settle towards a divergence free solution in the long run.

To improve this, I am currently working on a code to take a divergence-free velocity field as an initial condition. Pressure at this step is calculated from the PPE. However, since I'm using an implicit method, two initial conditions are required. So, I bootstrap from the

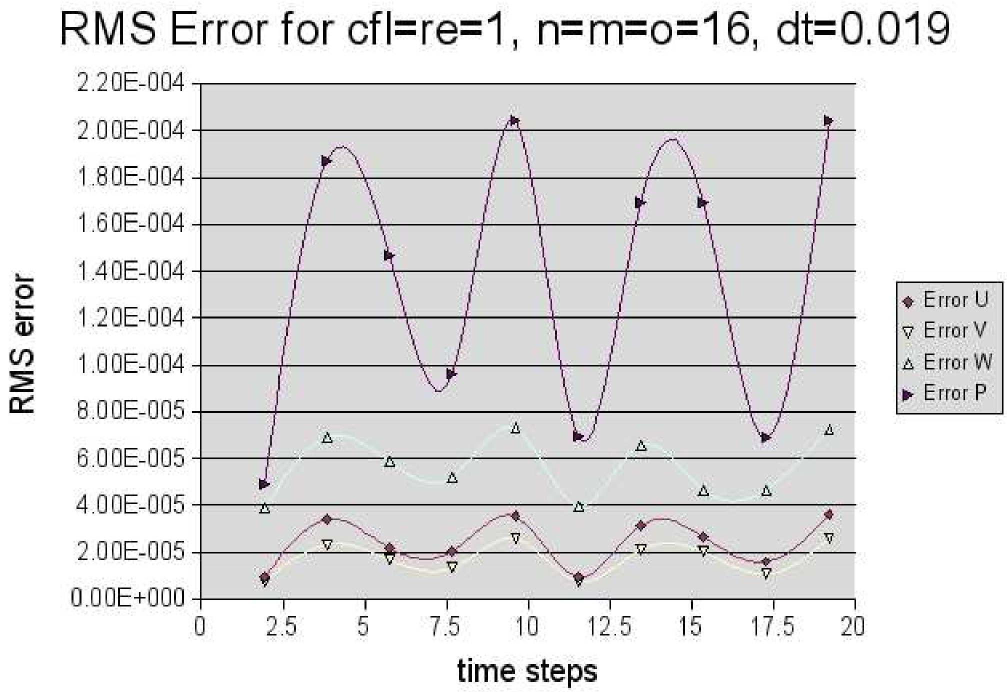

FIG. 2: RMS error vs. time for a 16x16x16 grid

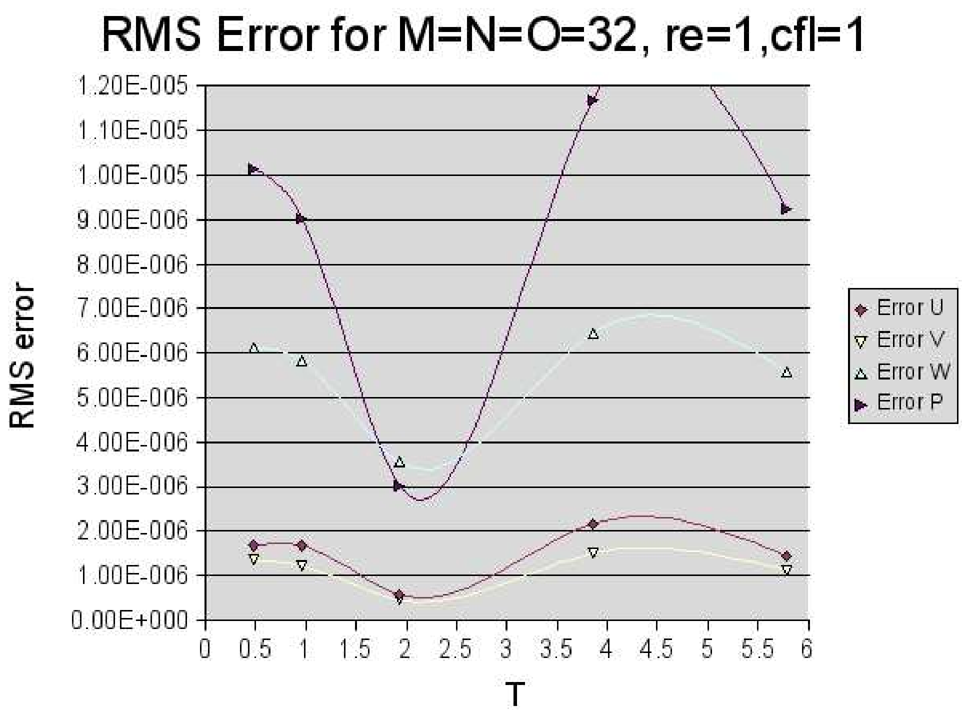

FIG. 3: RMS error vs. time for a 32x32x32 grid

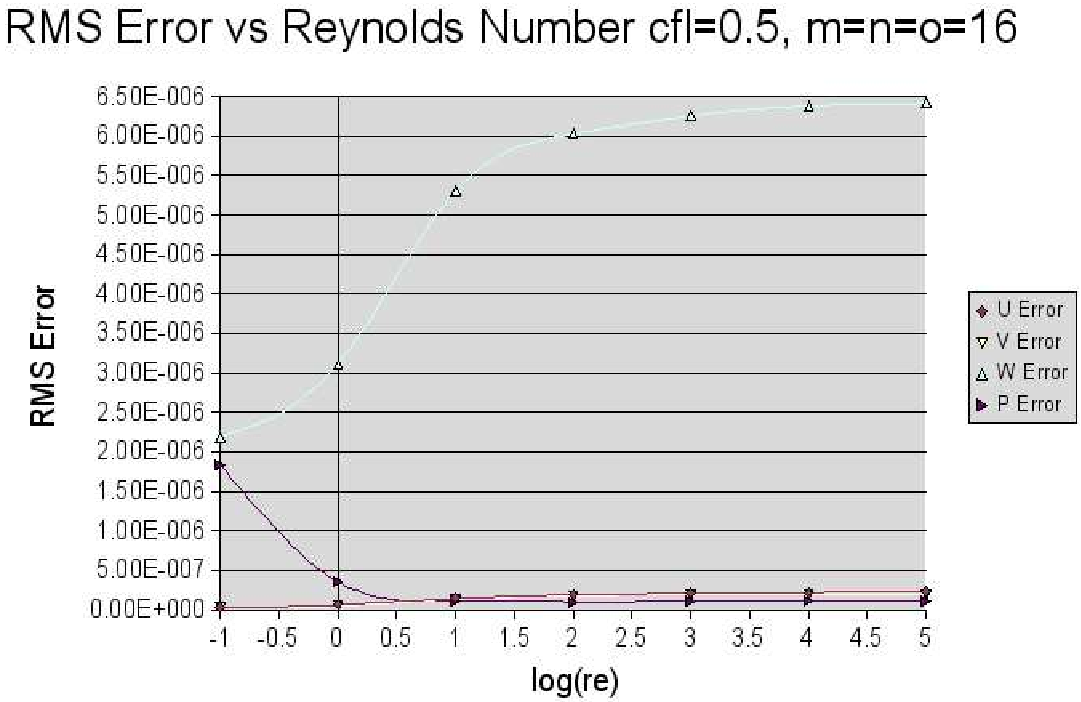

FIG. 4: Scaling of error with Reynolds number

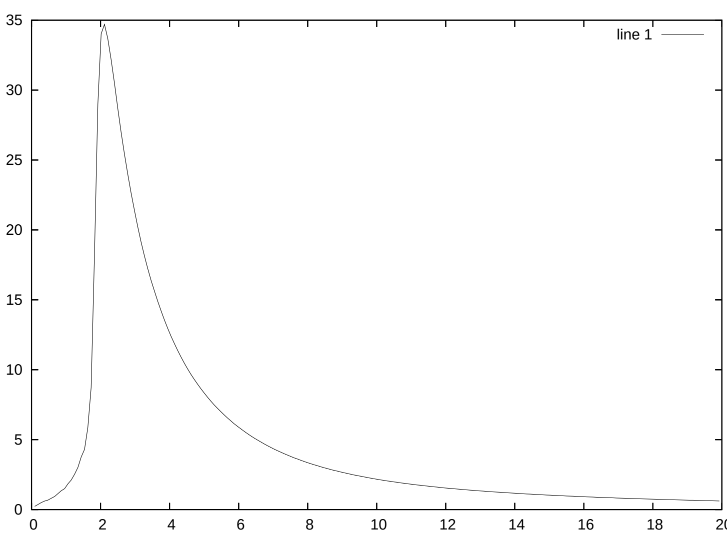

FIG. 5: $\|\nabla \cdot u\|_\infty$ vs. time

first condition using a modified Euler method, which is a second order explicit method, to calculate a second. As of December 12, 2004, that code is still a work in progress.

---

[1] Jian-Guo Liu Hans Johnston. Accurate, stable and efficient navier-stokes solvers based on explicit treatment of the pressure term. *Journal of Computational Physics*, xxx(xxx):xxx, 2004.

[2] Roger Peyret. *Spectral Methods for Incompressible Viscous Flow*. Springer, 2002.