A Novel, Near-Optimal Spectral Method for Simulating Fluids in a Cylinder

David Darrow

November 13, 2017

Abstract

Simulations of fluid flow offer theoretical insight into fluid dynamics and critical applications in industry, with implications ranging from blood flow to hurricanes. However, open problems in fluid dynamics require more accurate simulations and lower computational resource costs than current algorithms provide. Accordingly, we develop in this paper a novel, computationally efficient spectral method for computing solutions of the incompressible Navier–Stokes equations, which model incompressible fluid flow, on the cylinder. The method described addresses three major limitations of current methods. First, while current methods either underresolve the cylinder's boundary or overresolve its center (effectively overemphasizing less physically interesting non-boundary regions), this new method more evenly resolves all parts of the cylinder. Secondly, current simulation times scale proportionally as $N^{7/3}$ or higher (where N is the number of discretization points), while the new method requires at most $\mathcal{O}(N \log N)$ operations per time step. For large N, this means that calculations that required weeks can now be run in minutes. Lastly, current practical methods offer only low order (algebraic) accuracy. The new method has *spectral* accuracy, which often represents an improvement of the accuracy of the results by 5–10 orders of magnitude or more.

1 The Problem of Fluid Simulation

Simulations of fluid flow offer both theoretical insights into fluid dynamics and critical applications in industry. An open problem in fluid dynamical theory, for instance, is understanding how a boundary affects the energy cascade—the transfer of energy from large scales to small scales—of the flow; this cascade can be understood with better flow simulations. This particular application alone has wide implications, including studying blood flow in branch points and large arteries, understanding hurricane motion, and designing vacuum cleaners and golf balls [1].

Classically, an incompressible Newtonian fluid (e.g. water, blood, diesel fuel) is modeled by the *incompressible Navier–Stokes* equations [2] and some set of boundary conditions (discussed in the appendix). For a nonempty, closed, bounded domain $\Omega \subset \mathbb{R}^3$, these equations are as follows:

$$\nabla \cdot \vec{v} = 0, \qquad \left(\frac{\partial}{\partial t} + \vec{v} \cdot \nabla\right)\vec{v} = \frac{1}{Re}\nabla^2\vec{v} - \nabla p,$$

where $\vec{v}$ is the velocity field of the fluid, which measures the velocity of a particle of fluid at a point in Ω at time t, and p is a given time-dependent scalar field representing the pressure of the fluid at a point in Ω at time t. The parameter Re is called the *Reynolds number* and is defined as $Re = \frac{UL}{\nu}$, where U is the mean velocity of the fluid in the reference frame of the containing domain, L is the linear dimension of this containing domain, and ν is the kinematic viscosity of the fluid.

The Reynolds number determines the computational difficulty of simulating the flow. The higher the Reynolds number, the more chaotic the flow; past a critical threshold, the flow becomes *turbulent*, forming hard-to-model instabilities around its domain. This is one of the fundamental problems in computational fluid dynamics, and one which we have progressed significantly further to solving than have existing methods. We do this by reducing a solution of the Navier–Stokes equations to a solution of two nonlinear scalar equations that each resemble the *forced heat equation* (with a velocity-dependent source term), which governs temperature diffusion in the presence of some heat generating process.

When $Re \approx 0$, fluid flow is mostly *diffusive*, meaning that the ∇^2 term in the Navier–Stokes equations is the dominant expression. At this point, the Navier–Stokes equations closely resemble the heat equation. Solutions to both the heat equation and the Navier–Stokes equations must be *approximated numerically*, as not all initial and boundary conditions lend themselves to analytical solutions. With this in mind, the goal becomes maximizing the accuracy and minimizing the computation time of the approximate solver. Our forced heat equation solver involves three steps:

1. A method is derived to approximate continuous functions on the cylinder using a high degree series expansion. By using this expansion, we see *spectral* accuracy rather than the algebraic

accuracy of current methods. This represents an improvement in accuracy of several orders of magnitude—often 5–10 or more—which is essential for turbulence resolution [3].

2. Using backwards differentiation formulas to discretize in time, we obtain a Helmholtz equation at each time step and discretize it in the spatial domain using the method in step 1.

3. The alternating direction implicit (ADI) method is employed to accurately solve the discretized Helmholtz equation. This allows the equation to be solved in $\mathcal{O}(N \log N)$ time (where N is the chosen discretization size), which is significantly faster than current methods that employ large matrices costing $\mathcal{O}(N^{7/3})$ operations ($N^{1/3}$ matrices of size $N^{2/3} \times N^{2/3}$).

Once the heat equations are accurately solved in near-optimal time, as described above, we must relate them back to the Navier–Stokes equations.

Fluid flows with $Re \gg 1$ are dominated by *advection*. Instead of spreading information in an isotropic way, like the particles of a diffusive flow, particles of an advective flow tend to transport and exaggerate local values of the velocity field. At face value, this appears to have little to do with the heat equation, which is purely diffusive. However, the Navier–Stokes equations in the advective regime *can* be seen to connect intimately with the heat equation. Starting from our solution of the heat equation, the Navier–Stokes algorithm is developed in two more steps:

1. The Poloidal–Toroidal (PT) decomposition is introduced to split relevant vector fields into two scalar components each. The new variables allow for parallelization and avoid projection methods to maintain the incompressibility of the fluid. The PT decomposition can be found by solving two Poisson equations, which does not increase the computational complexity of $\mathcal{O}(N \log N)$ operations per time step.

2. The Navier–Stokes equations are decomposed into independent equations on poloidal and toroidal fields, which can be solved similarly to a set of two scalar heat equations (with non-linear forcing).

Our domain of interest for this paper is the closed cylinder in $\mathbb{R}^3$, though the method can be generalized to other polar or spherical geometries. The cylinder has many applications throughout the sciences, including studying blood flow, storing fuel in rockets, and understanding properties of hurricanes. Accordingly, the cylindrical coordinates (r, z, θ) are used throughout.

2 Spectral Approximation of Functions in the Cylinder

In order to approximate solutions to the Navier–Stokes equations in the cylinder, we first discretize the system by only enforcing the equations at N selected *discretization points*. Instead of working

with all functions, the discretized system is restricted to a certain N-dimensional space of high order polynomials on the cylinder. This approach, a *spectral method*, provides one unique and critical advantage over current methods: so-called *spectral accuracy*.

While standard fluid simulation methods achieve algebraic accuracy—errors of $\mathcal{O}(N^{-k})$ for some k [4]—spectral accuracy achieves errors of $\mathcal{O}(\rho^{-N})$ for some $\rho > 1$. Even when N is as small as 1000, this often represents an improvement over algebraically accurate methods of several orders of magnitude. This accuracy is also uniform throughout the cylinder; some current methods, such as *finite element methods*, underresolve the boundary of the region and overresolves the less interesting central portion of the cylinder, potentially missing important effects of "wall-bounded turbulence" [5].

Let $C \subset \mathbb{R}^3$ be the closed cylinder written in cylindrical coordinates (r, z, θ), where $-1 \leq r \leq 1$, $-\pi \leq \theta \leq \pi$, and $-1 \leq z \leq 1$. Let $m, n, p \in \mathbb{N}$ be the number of discretization points in the radial, vertical, and angular directions, respectively. The total number of discretization points in C is then $N = mnp$.

Definition 1. The *doubled* $m \times n \times p$ *Chebyshev–Chebyshev–Fourier grid*, denoted by $\text{CCF}_{(m,n,p)}$, is the set (in cylindrical coordinates (r, z, θ)) of points $\mathbf{x}_{(j,k,\ell)}$, where

$$\mathbf{x}_{(j,k,\ell)} = \left(\cos\left(\frac{(m-j-1)\pi}{m-1}\right), \cos\left(\frac{(n-j-1)\pi}{n-1}\right), \frac{(2\ell-p)\pi}{p}\right),$$

$$j, k, \ell \in \mathbb{Z}, \quad 0 \leq j < m, \quad 0 \leq k < n, \quad 0 \leq \ell < p.$$

This "doubled" grid, shown in Figure 1, can be contrasted with the non-doubled Chebyshev–Chebyshev–Fourier grid in the cylinder, which chooses the radial Chebyshev points between 0 and 1. This clusters discretization points near the azimuth (centerline) of the cylinder: a full third of the points are within the center area $1/16^{th}$ the cross sectional size of the region, slowing down computation significantly without providing an increase in approximation accuracy. The doubled grid, however, chooses the radial Chebyshev points between -1 and 1. This grid is an improvement over the standard one; unlike standard approaches, it distributes points almost evenly with respect to cross sectional area.

Now, we must choose some N-dimensional basis to span the set of functions restricted to our discretization grid. As it turns out, the set of "$\text{CCF}_{(m,n,p)}$ polynomials", defined below, does the trick; the set satisfies necessary existence and uniqueness theorems, allows easy representation of certain differential operators, and interpolates with high accuracy [6].

Definition 2. A *CCF*$_{(m,n,p)}$ *polynomial* is a function $q_{(j,k,\ell)} : C \to \mathbb{R}^3$ of the following form:

$$q_{(j,k,\ell)}(r, z, \theta) = T_j(r)T_k(z)e^{i(\ell - \lfloor p/2 \rfloor)\theta},$$

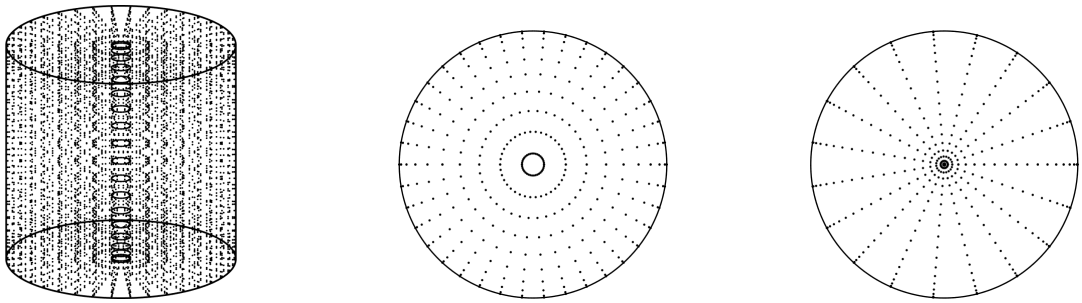

Figure 1: Illustration of the $\text{CCF}_{(m,n,p)}$ discretization grid (along with a cross-section), for $m = n = p = 20$, contrasted with the corresponding undoubled grid (the current paradigm). Unlike the "undoubled" grid on the right, $\text{CCF}_{(m,n,p)}$ does not cluster the majority of the points near the centerline of the cylinder.

where T_K is the degree K Chebyshev polynomial of the first kind and j, k, ℓ have the same bounds as in Definition 1. The complex vector space spanned by these $\text{CCF}_{(m,n,p)}$ polynomials is denoted $P_{(m,n,p)}$.

Also of use are these two sorts of polynomials: $U_K = C_K^{(1)}$ indicates the degree K Chebyshev polynomial of the second kind, and $C_K^{(2)}$ indicates the degree K ultraspherical polynomial such that the set $\{C_i^{(2)}\}_{i=0}^{\infty}$ is orthogonal with respect to the weight $(1-x^2)^{3/2}$ on $[-1, 1]$.

Any continuous function $f : C \to \mathbb{R}$ has a unique interpolant $f^* \in P_{(m,n,p)}$, i.e. $f = f^*$ on $\text{CCF}_{(m,n,p)}$. This interpolant is of the form

$$f^*(r, z, \theta) = \sum_{j=0}^{m-1} \sum_{k=0}^{n-1} \sum_{\ell=0}^{p-1} f^{(j,k,\ell)} T_j(r) T_k(z) e^{i(\ell - \lfloor p/2 \rfloor)\theta}.$$

In order to find the coefficients $f^{(j,k,\ell)}$, start with the set of values of f on the grid: $f_{(j,k,\ell)} = f(\mathbf{x}_{(j,k,\ell)})$. Perform the discrete cosine transform of type I (DCT) first on each m-tuple $(f_{(j,k,\ell)})_{j=0}^{m-1}$, so that

$$(f_{(k,\ell)}^{(j)})_{j=0}^{m-1} = \text{DCT}(f_{(j,k,\ell)})_{j=0}^{m-1}, \text{ for all } k, \ell.$$

Then, perform another DCT, but this time on the n-tuple $(f_{(k,\ell)}^{(j)})_{k=0}^{n-1}$, so that

$$(f_{(\ell)}^{(j,k)})_{k=0}^{n-1} = \text{DCT}(f_{(k,\ell)}^{(j)})_{k=0}^{n-1}, \text{ for all } j, \ell.$$

Finally, perform a discrete Fourier transform (DFT) on the p-tuple $(f_{(\ell)}^{(j,k)})_{\ell=0}^{p-1}$, so that

$$(f^{(j,k,\ell)})_{\ell=0}^{p-1} = \text{DFT}(f_{(\ell)}^{(j,k)})_{\ell=0}^{p-1}, \text{ for all } j, k.$$

Each value $f^{(j,k,\ell)}$ is then the coefficient on the basis function $q_{(j,k,\ell)}$. This series of transforms can be performed in $\mathcal{O}(N \log N)$ operations using the Fast Fourier Transform, where $N = mnp$. Accordingly, this provides an effective and convenient method by which to change from function

values to polynomial coefficients, which in turn allow us to use more computationally efficient and well-conditioned approximations to the Navier–Stokes and heat equations.

3 The Heat Equation

Our method for solving the Navier–Stokes equations, as is described later, relies on the *diffusive* core of fluid flow, wherein the fluid particles move in a nondirected fashion towards a net equilibrium. The nonlinear *advective* term is folded into the forcing term of a linear equation, offering significant computational time improvements without significantly affecting the accuracy of the computed solution. This linear equation takes the form of the *heat equation*, which models heat transfer and acts as a paradigm for all diffusive processes. As such, solving the heat equation is the first step to solving the Navier–Stokes equations, and the accuracy and computational cost of solving the former equation determines the accuracy and cost of solving the latter equations.

The heat equation in C is as follows:

$$\left(\frac{\partial}{\partial t} - \alpha\nabla^2\right) T = g(r, z, \theta, t),$$

where $T(r, z, \theta, t)$ denotes the temperature at a given point in space and time in the cylinder, $g(r, z, \theta, t)$ is the rate of heat generation at a point (r, z, θ) at time t, and $\alpha > 0$ is the "thermal diffusivity" of the domain, or the rate at which heat transfers to colder areas.

To solve the heat equation, we use the b^{th}*-order backwards differentiation formula* (BDF), given below for $b = 1, 2, 3, 4$:

BDF1:
$$f(t+h) - h\,\frac{d}{d\tau}f(\tau)|_{\tau=t+h} = f(t) + \mathcal{O}(h^2),$$

BDF2:
$$f(t+h) - \frac{2}{3}h\,\frac{d}{d\tau}f(\tau)|_{\tau=t+h} = \frac{4}{3}f(t) - \frac{1}{3}f(t-h) + \mathcal{O}(h^3),$$

BDF3:
$$f(t+h) - \frac{6}{11}h\,\frac{d}{d\tau}f(\tau)|_{\tau=t+h} = \frac{18}{11}f(t) - \frac{9}{11}f(t-h) + \frac{2}{11}f(t-2h) + \mathcal{O}(h^4),$$

BDF4:
$$f(t+h) - \frac{12}{25}h\,\frac{d}{d\tau}f(\tau)|_{\tau=t+h} = \frac{48}{25}f(t) - \frac{36}{25}f(t-h) + \frac{16}{25}f(t-2h) - \frac{3}{25}f(t-3h) + \mathcal{O}(h^5),$$

where h is the size of the time-step.

Define $\delta^{(b)} f(t)$ to be the RHS of the b^{th} order BDF (with the $\mathcal{O}(h^K)$ term omitted), and define $\kappa^{(b)}$ to be the (positive) coefficient of $\frac{d}{d\tau} f(\tau)$ in the same equation. For instance,

$$\kappa^{(4)} = \frac{12}{25} h$$

and

$$\delta^{(4)} f(t) = \frac{48}{25} f(t) - \frac{36}{25} f(t-h) + \frac{16}{25} f(t-2h) - \frac{3}{25} f(t-3h).$$

We can work this approximation into the heat equation by solving the latter for $\frac{\partial}{\partial t} T$:

$$\frac{\partial}{\partial \tau} T(r, z, \theta, \tau)|_{\tau=t+h} \approx \alpha \nabla^2 T(r, z, \theta, t+h) + g(r, z, \theta, t+h). \tag{1}$$

Once we know $\delta^{(b)} f(t)$, which we would if we were progressing through equidistant time steps, we can plug (1) into the b^{th} order BDF and get a *Helmholtz equation*:

$$\left(I - \kappa^{(b)} \alpha \nabla^2\right) T(r, z, \theta, t+h) \approx \delta^{(b)} T(r, z, \theta, t) + \kappa^{(b)} g(r, z, \theta, t+h),$$

where the RHS is already known and I is the identity operator. An initial problem is in the apparent unboundedness of the operator on the LHS at $r = 0$, as it includes the terms $\frac{1}{r}$ and $\frac{1}{r^2}$:

$$I - \kappa^{(b)} \alpha \nabla^2 = I - \kappa^{(b)} \alpha \left(\frac{1}{r} \frac{\partial}{\partial r} + \frac{\partial^2}{\partial r^2} + \frac{1}{r^2} \frac{\partial^2}{\partial \theta^2} + \frac{\partial^2}{\partial z^2} \right).$$

To resolve this, we multiply both sides of the Helmholtz equation by r^2 to obtain

$$r^2 \left(I - \kappa^{(b)} \alpha \nabla^2\right) T(r, z, \theta, t+h) \approx r^2 \delta^{(b)} T(r, z, \theta, t) + \kappa^{(b)} r^2 g(r, z, \theta, t+h). \tag{2}$$

The CCF$_{(m,n,p)}$ polynomial basis provides significant benefits for solving the above equation. If we work in "coefficient space", or the set of coefficients $(T^{(j,k,\ell)})$, then this $T(r, z, \theta, t+h)$ can be found in near-optimal time and with spectral accuracy.

We can "discretize" the left-hand operator in equation (2) in the space $P_{(m,n,p)}$. To start with, the $m \times n$ matrix $\mathbf{X}_\ell(t)$ is defined by $[\mathbf{X}_\ell(t)]_{j,k} = T^{(j,k,\ell)}(t)$, where $[\mathbf{X}_\ell(t)]_{j,k}$ is the element in the j^{th} row and k^{th} column of $\mathbf{X}_\ell(t)$. We similarly define the $m \times n$ matrix $\mathbf{g}_\ell(t)$ as the matrix of interpolation coefficients of $g(r, z, \theta, t)$.

From here, we discretize necessary operators as matrices to act on $\mathbf{X}_\ell(t)$.

- The matrix C_{01}, which converts a T_i coefficient vector to a U_i coefficient vector; and the matrix C_{12}, which converts a $U_i \equiv C_i^{(1)}$ coefficient vector to a $C_i^{(2)}$ coefficient vector:

$$C_{01} = \begin{bmatrix} 0 & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ & \frac{1}{2} & \ddots & \ddots & \\ & & \ddots & \ddots & \frac{1}{2} \\ & & & \frac{1}{2} & 0 \end{bmatrix}, \qquad C_{12} = \begin{bmatrix} 1 & 0 & -\frac{1}{3} & & \\ & \frac{1}{2} & 0 & \ddots & \\ & & \frac{1}{3} & \ddots & -\frac{1}{n} \\ & & & \ddots & 0 \\ & & & & \frac{1}{n} \end{bmatrix}.$$

  The matrix C_{02} is defined by $C_{02} = C_{12}C_{01}$. It converts a T_i coefficient vector to a $C_i^{(2)}$ coefficient vector.

- The b^{th}-order BDF operator $\delta^{(b)}$ is found by sampling its argument at $(t, t-h, ..., t-(b-1)h)$ and taking differences, as with the RHS of the b^{th} order BDF. For instance,

$$\delta^{(4)}\mathbf{X}_\ell(t) = \frac{48}{25}\mathbf{X}_\ell(t) - \frac{36}{25}\mathbf{X}_\ell(t-h) + \frac{16}{25}\mathbf{X}_\ell(t-2h) - \frac{3}{25}\mathbf{X}_\ell(t-3h).$$

- The multiplication operator R, which multiplies a $C_k^{(2)}(r)$ coefficient vector by r:

$$R = \begin{bmatrix} 0 & \frac{2}{3} & & & \\ \frac{1}{4} & 0 & \frac{5}{8} & & \\ & \frac{1}{3} & 0 & \ddots & \\ & & \ddots & \ddots & \frac{n+2}{2n+2} \\ & & & \frac{n-1}{2n} & 0 \end{bmatrix}.$$

  Note that the subdiagonal pattern starts with $\frac{2-1}{4} = \frac{1}{4}$, not with $\frac{1-1}{2} = 0$. For convenience, also define $R_2 = R^2 C_{02}$.

- The differentiation operator $D_1 = C_{01}\frac{\partial}{\partial r}$, where $\frac{\partial}{\partial r}$ differentiates a $T_k(r)$ coefficient vector by r; and the second differentiation operator $D_2 = C_{02}\frac{\partial^2}{\partial r^2}$, where $\frac{\partial^2}{\partial r^2}$ differentiates a $T_k(r)$ coefficient vector twice by r:

$$D_1 = \begin{bmatrix} 0 & 1 & & & \\ & & 2 & & \\ & & & \ddots & \\ & & & & n-1 \\ & & & & 0 \end{bmatrix}, \qquad D_2 = \begin{bmatrix} 0 & 0 & 4 & & \\ & & & \ddots & \\ & & & & 2n-2 \\ & & & & 0 \\ & & & & 0 \end{bmatrix}.$$

With these operators, equation (2) can be discretized (with the extrapolation $g(t+h) \approx g(t)$) as the following matrix equation:

$$\left(R_2 - \kappa^{(b)}\alpha(RC_{12}D_1 + R^2D_2 - \ell^2 C_{02})\right)\mathbf{X}_\ell(t+h)C_{02}^T - \kappa^{(b)}\alpha R_2 \mathbf{X}_\ell(t+h)D_2^T = R_2\left(\delta^{(b)}\mathbf{X}_\ell(t) + \kappa^{(b)}\mathbf{g}_\ell(t)\right)C_{02}^T. \quad (3)$$

The discretization process used above is discussed further in [7]. There are two notable properties of equation (3): each value of ℓ can be solved for independently (we can "decouple in θ"), and every matrix used in the construction of the equation is very sparse. We use the ADI method to solve equation (3) in $\mathcal{O}(N \log N)$ operations [8].

Algorithm 1 (ADI Method)**.** *Let A, C be $m \times m$ complex matrices, B, D be $n \times n$ complex matrices, and X, E be $m \times n$ complex matrices. Set two complex-valued sequences $(u_k)_{k=0}^{K-1}$ and $(v_k)_{k=0}^{K-1}$, and set $X_0 = 0$.*

The Sylvester equation $AXB + CXD = E$ can be approximately solved for X as follows:

1. *Solve $(u_kC - A)X_{k+\frac{1}{2}}B = CX_k(u_kB + D) + E$ for $X_{k+\frac{1}{2}}$.*
2. *Solve $CX_{k+1}(v_kB + D) = (v_kC - A)X_{k+\frac{1}{2}}B - E$ for X_{k+1}.*
3. *Repeat steps 1 and 2 for K iterations; $X_K \approx X$.*

With an appropriate choice of $(u_k)_{k=0}^{K-1}$ and $(v_k)_{k=0}^{K-1}$, we can get sufficient accuracy with $K = \mathcal{O}(\log mn)$. This iteration then gives an accurate solution in $\mathcal{O}(mn \log mn)$ operations, assuming that the initial Sylvester equation is well-conditioned. All such equations in this work are well-conditioned unless otherwise stated.

Equation (3) is of the sort described in Algorithm 1; at each time step, we can solve for each $\mathbf{X}_\ell$ in $\mathcal{O}(mn \log mn)$ operations. There are p choices of ℓ, though, so this is scaled up to $\mathcal{O}(mnp \log mn)$ total operations at each time step. Since $N = mnp$, the final algorithm has a total of $\mathcal{O}(N \log N)$ operations. This compares to times of $\mathcal{O}(N^{7/3})$, or $\mathcal{O}(m^3n^3p)$, with current spectral methods. A solution example is illustrated in Figure 2.

4 The Poloidal–Toroidal Decomposition

The *poloidal–toroidal decomposition* (PT decomposition) is a critical component for quickly and accurately solving the incompressible Navier–Stokes equations for two reasons. First, it allows these more complex vector equations to "decouple" into two scalar heat-like equations. We can maintain our spectral accuracy and near-optimal computation time from the heat equation solver,

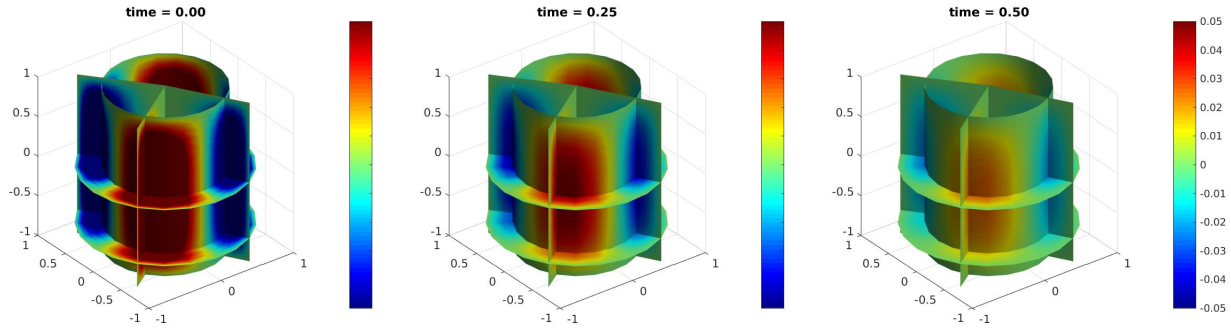

Figure 2: A simulation of the heat equation in the cylinder C. This is shown on various slices of the cylinder, where the temperature value is given by the color bars on the right of each figure. There is no advective motion, making the heat equation computationally faster to solve than the Navier–Stokes equations.

avoiding the large computational times faced by current solvers that work with the primitive form of the equations. Second, it ensures that the computed velocity field $\vec{v}$ is exactly incompressible, whereas current solvers maintain this central property by projecting the velocity field after each time step, introducing potentially large errors in the simulation [9].

Let $\hat{z}$ be the unit vector in the direction of the azimuth (the "z-direction") of C.

Definition 3. The (cylindrical) *poloidal–toroidal decomposition*, or "PT decomposition", of an incompressible vector field $\vec{V}$ is an expression of $\vec{V}$ in terms of the sum of a *toroidal* field and a *poloidal* field.

A *toroidal* vector field $\mathbf{T}$ is one of the form $\mathbf{T} = \nabla \times [\lambda \hat{z}]$, where λ is a scalar field.

A *poloidal* vector field $\mathbf{P}$ is one of the form $\mathbf{P} = \nabla \times \nabla \times [\gamma \hat{z}]$, where γ is a scalar field.

We prove the existence of the PT decomposition of a general incompressible vector field in the cylinder. A more general discussion of the PT decomposition is given in [10].

Theorem 1. *Let $\vec{V} : C \to \mathbb{R}^3$ be an arbitrary incompressible vector field on C, i.e., $\nabla \cdot \vec{V} = 0$. Then $\vec{V}$ can be decomposed as a sum of a toroidal and a poloidal field:*

$$\vec{V} = \nabla \times [\lambda \hat{z}] + \nabla \times \nabla \times [\gamma \hat{z}], \tag{4}$$

where λ and γ are real-valued functions on C.

Proof. Because $\vec{V}$ is incompressible and C is bounded, $\vec{V} = \nabla \times \vec{A}$ for some vector potential $\vec{A}$. It is then sufficient to show that

$$\vec{A} = \lambda \hat{z} + \nabla \times [\gamma \hat{z}] + \nabla \beta,$$

as $\nabla \beta$ is annihilated by the curl operator. Form an orthonormal basis $(\hat{x}, \hat{y}, \hat{z})$ of $\mathbb{R}^3$. Within this basis, it must be shown that there exist functions $\lambda = \lambda(x, y, z)$, $\gamma = \gamma(x, y, z)$, and $\beta = \beta(x, y, z)$

such that

$$\vec{A} = (\beta_x + \gamma_y)\hat{x} + (\beta_y - \gamma_x)\hat{y} + (\beta_z + \lambda)\hat{z}.$$

For any $\vec{A}$, the function $\lambda = \vec{A} \cdot \hat{z} - \beta_z$ is sufficient for our target equation, leaving the system

$$\beta_x + \gamma_y = \vec{A} \cdot \hat{x}, \qquad \beta_y - \gamma_x = \vec{A} \cdot \hat{y}.$$

Differentiate the former equation by x and the latter by y, retrieving two new equations from their sum and difference respectively:

$$\beta_{xx} + \beta_{yy} = A_x \cdot \hat{x} + A_y \cdot \hat{y}, \qquad \gamma_{xx} + \gamma_{yy} = A_x \cdot \hat{x} - A_y \cdot \hat{y}.$$

Along with the conditions that our target equation and its derivatives hold at the boundary, both Poisson equations have solutions. Thus, there exist λ and γ that satisfy equation (4). □

The scalar fields in this decomposition are not generally unique, though the vector fields are. To decompose the Navier–Stokes equations, the PT decomposition is required for two distinct vector fields: $\vec{\omega} = \nabla \times \vec{v}$ and $\nabla \times (\vec{v} \times \vec{\omega})$. The former field, $\vec{\omega} = \nabla \times \vec{v}$, is defined as the "vorticity" of the fluid, which gives the axis of rotation and rotational velocity of the fluid in the cylinder.

Corollary 1. *The following two equalities hold:*

$$\vec{\omega} \equiv \nabla \times \vec{v} = \nabla \times [\lambda_\omega \hat{z}] + \nabla \times \nabla \times [\gamma_\omega \hat{z}], \tag{5}$$

$$\nabla \times [\vec{v} \times \vec{\omega}] = \nabla \times [\lambda_a \hat{z}] + \nabla \times \nabla \times [\gamma_a \hat{z}]. \tag{6}$$

Proof. As $\vec{\omega} \equiv \nabla \times \vec{v}$ and $\nabla \times (\vec{v} \times \vec{\omega})$ are both the curls of their respective vector potentials, the divergence operator annihilates both. Thus, they are both incompressible functions on C, and Theorem 1 says that a PT decomposition exists. □

Lastly, we need some fast and accurate method of *computing* the PT decomposition of an arbitrary vector field. Again, let $\vec{V}$ be an arbitrary incompressible vector field, with its PT decomposition given by equation (4). Let

$$\Delta_h = \partial_x^2 + \partial_y^2 = \frac{1}{r}\partial_r + \partial_r^2 + \frac{1}{r^2}\partial_\theta^2,$$

in standard Cartesian and cylindrical coordinates respectively. This is the *horizontal Laplacian operator*, and takes the Laplacian as if its argument were only a function in two horizontal dimensions (at a given value of z).

Theorem 2. *With $\vec{V}$, γ, and λ defined as in Theorem 1, then*

$$\hat{z} \cdot \vec{V} = -\Delta_h \gamma, \qquad \hat{z} \cdot \nabla \times \vec{V} = -\Delta_h \lambda. \tag{7}$$

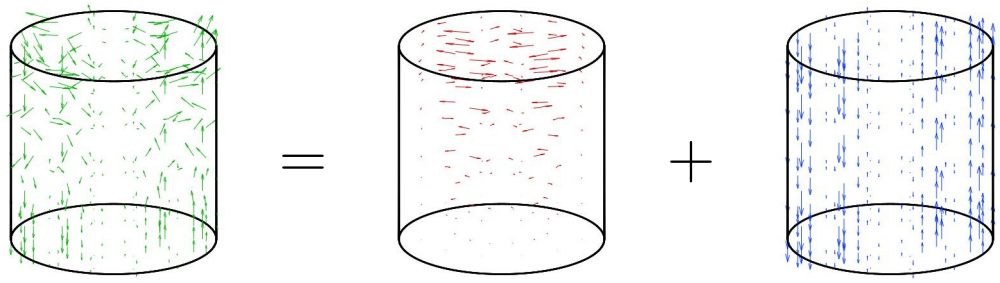

Figure 3: An example of a vorticity vector field being decomposed into its poloidal and toroidal components. The components are each given by a single scalar field. The poloidal and toroidal fields were computed numerically.

Proof. Form an orthonormal basis of $\mathbb{R}^3$, $(\hat{x}, \hat{y}, \hat{z})$. Using the identity $\hat{z} \cdot \nabla \times \vec{B} = \vec{B}_x \cdot \hat{y} - \vec{B}_y \cdot \hat{x}$, we retrieve the first of our claims:

$$\begin{aligned}
\hat{z} \cdot \vec{V} &= \hat{z} \cdot \nabla \times [\lambda \hat{z} + \nabla \times (\gamma \hat{z})] \\
&= (\hat{y} \cdot \partial_x - \hat{x} \cdot \partial_y)(\lambda \hat{z} + \nabla \times [\gamma \hat{z}]) \\
&= -\gamma_{xx} - \gamma_{yy} \\
&= -\Delta_h \gamma.
\end{aligned}$$

We perform the same calculation with $\hat{z} \cdot \nabla \times \vec{V}$. From direct calculation, $\vec{V} \cdot \hat{x} = \lambda_y + \gamma_{xz}$ and $\vec{V} \cdot \hat{y} = -\lambda_x + \gamma_{yz}$, so

$$\begin{aligned}
\hat{z} \cdot \nabla \times \vec{V} &= (\hat{y} \cdot \partial_x - \hat{x} \cdot \partial_y)\vec{V} \\
&= -\lambda_{xx} + \gamma_{yzx} - \lambda_{yy} - \gamma_{xzy} \\
&= -\lambda_{xx} - \lambda_{yy} \\
&= -\Delta_h \lambda.
\end{aligned}$$

□

Just like we did with the heat equation, define the $m \times n$ matrix $\mathbf{\Lambda}_\ell$ [resp. $\mathbf{\Gamma}_\ell$] by $(\mathbf{\Lambda}_\ell)_{j,k} = \lambda^{(j,k,\ell)}$ [resp. $(\mathbf{\Gamma}_\ell)_{j,k} = \gamma^{(j,k,\ell)}$], where $(\lambda^{(j,k,\ell)})$ [resp. $(\gamma^{(j,k,\ell)})$] are the interpolation coefficients for λ [resp. γ] in $\text{CCF}_{(n,m,p)}$. Define $\mathbf{V}_{r,\ell}$, $\mathbf{V}_{\theta,\ell}$, and $\mathbf{V}_{z,\ell}$ similarly for the radial, angular, and vertical components of $\vec{V}$ respectively. With the same operators as before, the system (7) can be written as matrix equations:

$$(RC_{12}D_1 + R^2 D_2 - \ell^2 C_{02})\mathbf{\Lambda}_\ell = -R_2 \mathbf{V}_{z,\ell},$$

$$(RC_{12}D_1 + R^2 D_2 - \ell^2 C_{02})\mathbf{\Gamma}_\ell = i\ell R C_{02}\mathbf{V}_{r,\ell} - (R^2 C_{12} D_1 + R C_{02})\mathbf{V}_{\theta,\ell}.$$

These can be solved using the ADI method in $\mathcal{O}(N \log N)$ operations. An example decomposition is shown in Figure 3.

5 Rewriting the Navier–Stokes Equations

We now have all of the tools at our disposal to rewrite the Navier–Stokes equations as a set of heat equations (with nonlinear forcing terms) in terms of the toroidal and poloidal components of $\vec{\omega}$.

Theorem 3. *Let $\lambda_\omega, \lambda_a, \gamma_\omega, \gamma_a$ be defined as in equations (5) and (6). The Navier–Stokes equations can be rewritten in the following way:*

$$\left(\frac{\partial}{\partial t} - \frac{1}{Re}\nabla^2\right)\lambda_\omega = \lambda_a, \qquad \left(\frac{\partial}{\partial t} - \frac{1}{Re}\nabla^2\right)\gamma_\omega = \gamma_a. \tag{8}$$

Proof. The primitive form of the incompressible equation is as follows:

$$\left(\frac{\partial}{\partial t} + \vec{v}\cdot\nabla\right)\vec{v} = \frac{1}{Re}\nabla^2\vec{v} - \nabla p,$$

with the additional constraint that $\vec{v}$ is incompressible. Take the curl of both sides of the equation, noting that $\nabla \times \nabla p = 0$ and that the curl commutes with univariate derivatives and the vector Laplacian, and substitute $\vec{\omega} = \nabla \times \vec{v}$ where applicable:

$$\frac{\partial}{\partial t}\vec{\omega} + \nabla \times [\vec{v}\cdot\nabla\vec{v}] = \frac{1}{Re}\nabla^2\vec{\omega}.$$

Using the identity $\frac{1}{2}\nabla(\vec{V}\cdot\vec{V}) = (\vec{V}\cdot\nabla)\vec{V} + \vec{V}\times\nabla\times\vec{V}$,

$$\frac{\partial}{\partial t}\vec{\omega} - \nabla \times [\vec{v}\times\vec{\omega}] = \frac{1}{Re}\nabla^2\vec{\omega}.$$

Now, substitute the PT decompositions in equations (5) and (6):

$$\frac{\partial}{\partial t}(\nabla\times[\lambda_\omega\hat{z}]+\nabla\times\nabla\times[\gamma_\omega\hat{z}]) - \frac{1}{Re}\nabla^2(\nabla\times[\lambda_\omega\hat{z}]+\nabla\times\nabla\times[\gamma_\omega\hat{z}]) = \nabla\times[\lambda_a\hat{z}]+\nabla\times\nabla\times[\gamma_a\hat{z}].$$

Commute the operators such that the curls are applied last:

$$\nabla\times[\left(\frac{\partial}{\partial t} - \frac{1}{Re}\nabla^2\right)\lambda_\omega\hat{z}] + \nabla\times\nabla\times[\left(\frac{\partial}{\partial t} - \frac{1}{Re}\nabla^2\right)\gamma_\omega\hat{z}] = \nabla\times[\lambda_a\hat{z}] + \nabla\times\nabla\times[\gamma_a\hat{z}].$$

The LHS and RHS of this equation represent two different PT decompositions of the same vector field. Because the poloidal and toroidal *vector fields* are unique, this equation can be decoupled:

$$\nabla\times[\left(\frac{\partial}{\partial t} - \frac{1}{Re}\nabla^2\right)\lambda_\omega\hat{z}] = \nabla\times[\lambda_a\hat{z}], \qquad \nabla\times\nabla\times[\left(\frac{\partial}{\partial t} - \frac{1}{Re}\nabla^2\right)\gamma_\omega\hat{z}] = \nabla\times\nabla\times[\gamma_a\hat{z}].$$

Nonphysical gauge symmetries are available for each scalar component due to the presence of curls. Thus, these equations can be satisfied and these gauge symmetries removed by removing the curls and enforcing that the poloidal and toroidal scalars are the same on each side. Then we

retrieve system (8). To show that a solution of this system uniquely determines $\vec{v}$, note that its PT decomposition *does* uniquely determine $\vec{\omega}$. From the Helmholtz Theorem [11], specifying a vector field's curl and divergence (along with boundary conditions, as detailed in the appendix) uniquely determines that vector field. As $\nabla \cdot \vec{v} = 0$ and $\vec{\omega} \equiv \nabla \times \vec{v}$ is also known, $\vec{v}$ is determined. □

Taking $\lambda_\omega, \lambda_a, \gamma_\omega, \gamma_a$ all to be functions of time, our first step in solving the system (8) is to write these equations as matrix equations.

The $m \times n$ matrix $\mathbf{\Lambda}_{\omega,\ell}(t)$ [resp. $\mathbf{\Gamma}_{\omega,\ell}(t)$] is defined by $(\mathbf{\Lambda}_{\omega,\ell}(t))_{j,k} = \lambda_\omega^{(j,k,\ell)}$ [resp. $(\mathbf{\Gamma}_{\omega,\ell}(t))_{j,k} = \gamma_\omega^{(j,k,\ell)}$], where $(\lambda_\omega^{(j,k,\ell)})$ [resp. $(\gamma_\omega^{(j,k,\ell)})$] are the interpolation coefficients for λ_ω [resp. γ_ω] in $\text{CCF}_{(n,m,p)}$. We define $\mathbf{\Lambda}_{a,\ell}(t)$ $\mathbf{\Gamma}_{a,\ell}(t)$ the same way, but for λ_a and γ_a.

Initial conditions $\mathbf{\Lambda}_{\omega,\ell}(0)$ and $\mathbf{\Gamma}_{\omega,\ell}(0)$ can be found by replacing $\vec{V}$ by $\vec{\omega}(r, z, \theta, 0)$ in (7) and solving the discretized Poisson equations. The same can be said of $\mathbf{\Lambda}_{a,\ell}(t)$ and $\mathbf{\Gamma}_{a,\ell}(t)$, once $\nabla \times [\vec{v}(t) \times \vec{\omega}(t)]$ is known. To find $\nabla \times [\vec{v}(t) \times \vec{\omega}(t)]$, the following theorem is needed:

Theorem 4. *Let $\vec{\Psi}$ be the vector potential for $\vec{v}$; that is, $\vec{v} = \nabla \times \vec{\Psi}$. If the PT decomposition of $\vec{\Psi}$ is $\vec{\Psi} = \nabla \times [\lambda_\psi \hat{z}] + \nabla \times \nabla \times [\gamma_\psi \hat{z}]$, then the following is a decomposition of $\vec{\omega} \equiv \nabla \times \vec{v}$:*

$$\vec{\omega} = \nabla \times [-\nabla^2 \lambda_\psi \hat{z}] + \nabla \times \nabla \times [-\nabla^2 \gamma_\psi \hat{z}].$$

Proof. By taking two curls of the PT decomposition of Ψ, it is shown that $\vec{\omega} = \nabla \times \nabla \times \nabla \times [\lambda_\psi \hat{z}] + \nabla \times \nabla \times \nabla \times \nabla \times [\gamma_\psi \hat{z}]$. From the definition of a vector Laplacian, $\nabla \times \nabla \times \vec{V} = \nabla[\nabla \cdot \vec{V}] - \nabla^2 \vec{V}$. Under a curl, the gradient term vanishes. Then

$$\nabla \times \nabla \times \nabla \times \vec{V} = \nabla \times [-\nabla^2 \vec{V}].$$

Applying this to both sides of the above equation retrieves our desired equality. □

Define $\mathbf{\Gamma}_{\psi,\ell}(t)$ and $\mathbf{\Gamma}_{\psi,\ell}(t)$ the same way we did $\mathbf{\Lambda}_{\omega,\ell}(t)$ and $\mathbf{\Gamma}_{\omega,\ell}(t)$, but for λ_ψ and γ_ψ. We can find λ_ψ and γ_ψ at any time t using the following discretized Poisson equations:

$$(RC_{12}D_1 + R^2 D_2 - \ell^2 C_{02})\mathbf{\Lambda}_{\psi,\ell}(t)C_{02}^T + R_2 \mathbf{\Lambda}_{\psi,\ell}(t) D_2^T = -R_2 \mathbf{\Lambda}_{\omega,\ell} C_{02}^T,$$

$$(RC_{12}D_1 + R^2 D_2 - \ell^2 C_{02})\mathbf{\Gamma}_{\psi,\ell}(t)C_{02}^T + R_2 \mathbf{\Gamma}_{\psi,\ell}(t) D_2^T = -R_2 \mathbf{\Gamma}_{\omega,\ell} C_{02}^T.$$

By taking appropriate curls, $\vec{v}$ and $\vec{\omega}$ can be recovered, and thus $\nabla \times [\vec{v} \times \vec{\omega}]$ can be calculated directly: $\vec{v} = \nabla \times [\gamma_\omega \hat{z}] + \nabla \times \nabla \times [\lambda_\psi \hat{z}]$ and $\vec{\omega} = \nabla \times [\lambda_\omega \hat{z}] + \nabla \times \nabla \times [\gamma_\omega \hat{z}]$. The first equality results from the fact that the curl of the toroidal field of $\vec{A}$ is the poloidal field of $\nabla \times \vec{A}$ for any incompressible $\vec{A}$. From here, we can write the Navier–Stokes equations as a system of discretized

heat equations. The system (8) can be written as follows and solved using the ADI method:

$$\left(R_2 - \frac{\kappa^{(b)}}{Re}(RC_{12}D_1 + R^2D_2 - \ell^2 C_{02})\right)\mathbf{\Lambda}_{\omega,\ell}(t+h)C_{02}^T - \frac{\kappa^{(b)}}{Re}R_2\mathbf{\Lambda}_{\omega,\ell}(t+h)D_2^T = R_2\left(\delta^{(b)}\mathbf{\Lambda}_{\omega,\ell}(t) + \kappa^{(b)}\mathbf{\Lambda}_{a,\ell}(t)\right)C_{02}^T,$$

$$\left(R_2 - \frac{\kappa^{(b)}}{Re}(RC_{12}D_1 + R^2D_2 - \ell^2 C_{02})\right)\mathbf{\Gamma}_{\omega,\ell}(t+h)C_{02}^T - \frac{\kappa^{(b)}}{Re}R_2\mathbf{\Gamma}_{\omega,\ell}(t+h)D_2^T = R_2\left(\delta^{(b)}\mathbf{\Gamma}_{\omega,\ell}(t) + \kappa^{(b)}\mathbf{\Gamma}_{a,\ell}(t)\right)C_{02}^T.$$

6 Numerical Experiments

Three methods for solving the forced heat equation, the primary component in our Navier–Stokes equation solution algorithm, were tested against an exact solution using various discretization sizes. Below are the maximum relative error values for these methods, as well as the computational times required to find these solutions. The first two methods were run for 200 time steps, with $N = 7^3, 11^3, 15^3$, and 19^3 in trials 1, 2, 3, and 4 respectively. Due to machine limitations, the finite difference method was only run for 6 time steps, with $N = 61^3, 81^3, 101^3$, and 121^3 respectively. The new spectral method achieves the highest accuracy in the least computational time of the tested methods. Numerical experiments for the final system of equations (8) are in progress.

Maximum Error

Method \ Trial #	1	2	3	4
New Spectral Method	$6.03 \cdot 10^{-5}$	$5.88 \cdot 10^{-5}$	$3.22 \cdot 10^{-5}$	$1.76 \cdot 10^{-5}$
Spectral Collocation	$6.40 \cdot 10^{-5}$	$5.85 \cdot 10^{-3}$	$3.51 \cdot 10^{-5}$	$5.90 \cdot 10^{-3}$
Finite Difference	$3.80 \cdot 10^{-1}$	$2.93 \cdot 10^{-1}$	$2.39 \cdot 10^{-1}$	$2.00 \cdot 10^{-1}$

Computational Time (sec)

Method \ Trial #	1	2	3	4
New Spectral Method	1.00	3.19	5.88	6.50
Spectral Collocation	1.06	4.42	11.9	17.9
Finite Difference	25.1	28.3	56.8	100

7 Future Work

After completion of the incompressible Navier-Stokes solver, the primary goal will be to generalize this work to compressible Navier-Stokes solutions. The general form of the equations is as follows:

$$\left(\frac{\partial}{\partial t} + \vec{v} \cdot \nabla\right) \vec{v} = \frac{1}{Re}\nabla^2 \vec{v} - \nabla p + \vec{F},$$

where the new term, $\vec{F}$, is the *external force* on the fluid. This force could be due to an electromagnetic field, which would come about if we were modeling electrically conductive fluids (e.g. solar wind). It could be gravity, which would be an important factor in modeling any gas near a planetary body. Whatever the force may be, it usually takes a very specific form: fundamental forces are all *irrotational*, meaning that the potential energy can be defined.

The best current methods cannot take advantage of this property, though; separating the irrotational component from a vector field leaves behind an *incompressible* component, so any method that decouples the two would require a fast and accurate incompressible Navier-Stokes solver. Because of this, most solutions spend a great deal of time computing both parts of the vector field. With the above work on the incompressible equations, though, we can completely sidestep this problem by solving for the irrotational component separately.

Let $\vec{v}$ be a vector field in the cylinder, as before, but not necessarily incompressible. $\vec{v}$ can be decomposed into two components as follows, using the Helmholtz Decomposition:

$$\vec{v} = -\nabla Y + \nabla \times \vec{\Psi}.$$

Take the divergence of both sides of the Navier-Stokes equation, letting $\vec{F} - \nabla p = -\nabla U$:

$$\nabla \cdot \left(\frac{\partial}{\partial t} + \vec{v} \cdot \nabla\right) \vec{v} = \nabla \cdot \frac{1}{Re}\nabla^2 \vec{v} - \nabla^2 U.$$

Let $\nabla \cdot (\vec{v} \cdot \nabla \vec{v}) = -\nabla^2 \beta$; this can be found at any time with a Poisson solve. Then

$$\nabla \cdot \frac{\partial}{\partial t}\vec{v} = \nabla \cdot \frac{1}{Re}\nabla^2 \vec{v} + \nabla^2(\beta - U),$$

and

$$\nabla^2 \left(\frac{\partial}{\partial t} - \frac{1}{Re}\nabla^2\right) Y = \nabla^2(\beta - U).$$

The Laplacian can be removed from both sides, as this does not remove physical information in the system. This gives the following equation:

$$\left(\frac{\partial}{\partial t} - \frac{1}{Re}\nabla^2\right) Y = \beta - U,$$

which is just a third heat equation solve. Coupling this with the incompressible solver from before, $\vec{v}$ can be determined completely in any system. The advantage of doing it in this decoupled fashion is that it allows us the same spectral accuracy from before, near optimal time (still $\mathcal{O}(N \log N)$), and the freedom to use a much smaller time step with the more complex irrotational component of $\vec{v}$ without sacrificing much-needed algorithm speed.

8 Conclusion

There are several published current methods which can be used to approximate solutions to the Navier–Stokes equations in a cylinder, but they share three major limitations: 1) They often either cluster points near the centerline of the cylinder or resolve the boundary only with linearly decaying error, 2) they have algebraically decaying errors over the whole cylinder, and 3) they have relatively high computational cost (often $\mathcal{O}(N^{7/3})$ or higher).

The method described resolves all of these in turn. By choosing a "doubled" Chebyshev-Chebyshev-Fourier discretization grid, it resolves all parts of the cylinder with high accuracy and without clustering discretization points. The algebraically decaying error associated with other methods is replaced with *spectral accuracy* by using high-degree polynomial interpolants to the velocity field over the whole cylinder. This accuracy, generally an improvement of several orders of magnitude over previous methods, is critical for understanding turbulent flow in bounded regions. Three steps were involved in cutting the computational cost to $\mathcal{O}(N \log N)$ operations ("near-optimal time") per time step. First, the heat equation was written as an iteration of Helmholtz equations, each of which was approximately solved in near-optimal time using the ADI method. Second, an algorithm was created to compute the Poloidal–Toroidal decomposition of a general incompressible vector field in near-optimal time. Lastly, the Navier–Stokes equations were transformed into two equations—one on the poloidal and one on the toroidal component of the vorticity field—and solved as heat equations by including a time lag on the remaining advective term. This near-optimal time offers a substantial improvement to most current methods' cost of $\mathcal{O}(N^{7/3})$ or more operations, generally dropping calculation time by several orders of magnitude.

Appendix: Boundary Conditions

There are three sets of boundary conditions that need to be employed by our method. First, conditions for computing the PT decomposition of a given vector field. Second, conditions for finding the components of the vector potential of $\vec{v}$, as in Theorem 4. Finally, we need conditions for solving the discretized Navier–Stokes equations (8).

For the PT decomposition $\vec{V} = \nabla \times [\lambda \hat{z}] + \nabla \times \nabla \times [\gamma \hat{z}]$ to be correct, it must satisfy one equality for each of the three vector components $(\hat{r}, \hat{\theta}, \hat{z})$ of $\vec{V}$. Since the $\hat{z}$ component is folded into the horizontal Poisson equation, we are left with $\partial_\theta \lambda + r\partial_r \partial_z \gamma = rV^r$ and $\partial_\theta \partial_z \gamma - r\partial_r \lambda = rV^\theta$, where $\vec{V} = V^r \hat{r} + V^\theta \hat{\theta} + V^z \hat{z}$. These equations only need to be enforced on the boundary of the cylinder, and can be discretized in the same manner as were the Helmholtz and Poisson equations.

In order to find the vector potential $\vec{\Psi}$, we use *no-slip* boundary conditions (though other conditions would work equally well)—that is, the boundary of the cylinder is non-moving and frictional. With that restriction, the vector potential is normal to the boundary of the cylinder (which can be seen from Stokes' Theorem). When $z = \pm 1$, the $\hat{r}$ and $\hat{\theta}$ components of Ψ must vanish: $\partial_\theta \lambda_\psi + r\partial_r \partial_z \gamma_\psi = \partial_\theta \partial_z \gamma_\psi - r\partial_r \lambda_\psi = 0$. When $r = 1$, the $\hat{z}$ and $\hat{\theta}$ components of Ψ must vanish: $(r\partial_r + r^2\partial_r^2 + \partial_\theta^2)\gamma_\psi = \partial_\theta \partial_z \gamma_\psi - r\partial_r \lambda_\psi = 0$. These equations can be discretized as before.

For solving the Navier–Stokes equations, we continue to use no-slip conditions. Due to work by Quartepelle[12], it is known that this no-slip condition on the velocity field is equivalent to the vorticity field being L^2 orthogonal to all incompressible vector fields $\vec{\eta}$ with $\nabla^2 \vec{\eta} = 0$. To form these $\vec{\eta}$, first let (η_i) be a basis of scalar cylindrical harmonics. For some i, let $\vec{\eta}_i = \nabla \times [\eta_i \hat{z}]$ and $\vec{\eta}_i^{(2)} = \nabla \times \nabla \times [\eta_j \hat{z}]$. As all incompressible vector fields have a PT decomposition and the Laplacian commutes with the curl, the set $\{\vec{\eta}_i\} \cup \{\vec{\eta}_i^{(2)}\}$ spans the set of $\vec{\eta}$. To achieve L^2 orthogonality between $\vec{\omega}$ and $\vec{\eta}$ in C, the following equation must be met:

$$\int_C (\vec{\eta} \cdot (\nabla \times [\lambda_\omega \hat{z}] + \nabla \times \nabla \times [\gamma_\omega \hat{z}]))\ dV = \int_C ([\nabla \times \vec{\eta}] \cdot \lambda_\omega \hat{z})\, dV + \int_{\partial C} (\vec{n} \cdot [\lambda_\omega \hat{z} \times \vec{\eta}])\, dS + \int_{\partial C} (\vec{n} \cdot [\gamma_\omega \hat{z} \times \nabla \times \vec{\eta} + (\nabla \times \vec{\eta}) \times \gamma_\omega \hat{z}])\, dS = 0.$$

Let $\vec{\eta} = \eta^r \hat{r} + \eta^\theta \hat{\theta} + \eta^z \hat{z}$. Then the above equation simplifies to

$$\int_{r=1} \eta^\theta \lambda_\omega\ d\theta dz + \int_C [\eta^\theta + r\partial_r \eta^\theta - \partial_\theta \eta^r] \lambda_\omega\ dr d\theta dz + \int_{r=1} ([\partial_z \eta^r - \partial_r \eta^z] \gamma_\omega + \eta^z \partial_r \gamma_\omega)\ d\theta dz - \int_{z=1} \left(\eta^\theta \partial_\theta \gamma_\omega + r\eta^r \partial_r \gamma_\omega\right)\ dr d\theta + \int_{z=-1} \left(\eta^\theta \partial_\theta \gamma_\omega + r\eta^r \partial_r \gamma_\omega\right)\ dr d\theta = 0.$$

This equation can be discretized as before, creating one boundary condition for each $\vec{\eta}$. These conditions can also be included without "coupling in θ" by first enforcing a simpler set of boundary conditions and then performing a low-rank update on the solution matrix.

Acknowledgments

I would like to thank my two mentors, Professors Grady Wright and Alex Townsend, for their helpful advice and many contributions during the course of this project. I would also like to thank the MIT PRIMES–USA program for the opportunity to conduct this research.

References

[1] F. Alam, H. Chowdhury, H. Moria, R. La Brooy, A. Subic, A comparative study of golf ball aerodynamics, 17th Australasian Fluid Mechanics Conference, 2010.

[2] G. Stoyan, A. J. Chorin, and J. E. Marsden, A Mathematical Introduction to Fluid Mechanics, Springer–Verlag, New York, 1979.

[3] Y. Zhiyin, Large-eddy simulation: Past, present and the future, Chinese Journal of Aeronautics, 28 (2015), pp. 11–24.

[4] R. J. LeVeque, Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems, 2007.

[5] T. Krauthammer, Accuracy of the finite element method near a curved boundary, Computers & Structures, 10 (1979), pp. 921-929.

[6] L. N. Trefethen, Approximation Theory and Approximation Practice, SIAM, Philadelphia, 2012.

[7] S. Olver and Mentor, A fast and well-conditioned spectral method, SIAM Review, 55 (2013), pp. 462–489.

[8] D. W. Peaceman and H. H. Rachford, Jr., The numerical solution of parabolic and elliptic differential equations, J. Soc. Ind. App. Math., 3 (1955), pp. 28–41.

[9] J. Shen, On error estimates of the projection methods for the Navier–Stokes equations: second-order schemes, Math. Comput., 65 (1996), pp. 1039–1065.

[10] P. Boronski and L. S. Tuckerman, Poloidal-toroidal decomposition in a finite cylinder. I. Influence matrices for the magnetohydrodynamic equations, J. Comput. Phys., 227 (2007), pp. 1523–1543.

[11] X. L. Zhou, On uniqueness theorem of a vector function, PIER, 65 (2006), pp. 93–102.

[12] L. Quartapelle and F. Valz-Gris, Projection conditions on the vorticity in viscous incompressible flows, Int. J. Numer. Meth. Fluids, 1 (1981), pp. 129–144.