# Navier-Stokes equations

Introduction to spectral methods for the CSC Lunchbytes Seminar Series.
Incompressible, hydrodynamic turbulence is described completely by the Navier-Stokes equations

$$\partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla P + \nu \nabla^2 \mathbf{u} \tag{1}$$

$$\nabla \cdot u = 0$$

where

- $\mathbf{u}$ is the velocity vector in two or three dimensions,
- $P$ is the pressure
- $\nu$ is the viscosity.

These equations contain

- advection $(\mathbf{u} \cdot \nabla)\mathbf{u}$ and
- dissipation $\nu \nabla^2 \mathbf{u}$.
- The ratio between these terms is the
- Reynolds number $Re = UL/\nu$, where

- $L$ is a large length scale.

- Turbulence arises when $R$ is large.

# Boundary conditions and pressure

Pressure equation:

$$\nabla \cdot \mathrm{N.S.} \rightarrow -\nabla^2 P = \partial_i u_j \partial_j u_i \tag{2}$$

To define a complete system you must also define the

- boundary conditions
- any inflow or outflow.

For realistic, no-slip boundary conditions

- the trickiest part of solving the incompressible Navier-Stokes equations is
- solving for the pressure
- given the boundary conditions.
- I will return to this at the end

# Spectral decomposition

Assume $u(x)$ is periodic in $x$ with period $2\pi$

$$u(x) = \int u(k)e^{ikx}dk \tag{3}$$

where

$$u(k) = \frac{1}{2\pi}\int u(x)e^{-ikx}dx \tag{4}$$

Then

$$\frac{\partial u(x)}{\partial x} = \int u(k)\frac{\partial}{\partial x}e^{-ikx}dk = \int iku(k)e^{ikx}dk \tag{5}$$

Now generalise to 3D:

$$\overrightarrow{u}(\overrightarrow{r}) = \int \overrightarrow{u}(\overrightarrow{k})e^{i\overrightarrow{k}\cdot\overrightarrow{r}}d^3k \tag{6}$$

where

$$\overrightarrow{u}(\overrightarrow{k}) = \frac{1}{(2\pi)^3}\int \overrightarrow{u}(\overrightarrow{r})e^{-i\overrightarrow{k}\cdot\overrightarrow{r}}d^3r \tag{7}$$

# Dissipation

$$\nabla^2 u(\mathbf{r}) = \int -|\mathbf{k}|^2 \mathbf{u}(\mathbf{k}) \mathbf{d^3 k} \tag{8}$$

because

$$\partial_x^2 + \partial_y^2 + \partial_z^2 \rightarrow (ik_x)^2 + (ik_y)^2 + (ik_z)^2 = -|\mathbf{k}|^2$$

As explained below,

- diffusion equation can be solved in time
- exactly and efficiently in Fourier space.

Assume we know the nonlinear term $F(x) = (u(r) \cdot \nabla)u(r)$ which in Fourier space becomes $F(k)$.

The forced Stokes equations in physical space is

$$\dot{\mathbf{u}}(\mathbf{r}) + \mathbf{F}(\mathbf{r}) = \frac{\partial}{\partial t}\mathbf{u} + \mathbf{F}(\mathbf{r}) = -\nabla P + \nu \nabla^2 \mathbf{u} \tag{9}$$

and the forced Stokes equation in Fourier space is

$$\dot{\mathbf{u}}(k) + \mathbf{F}(k) = -i\mathbf{k}P(k) - \nu|\mathbf{k}|^2 \mathbf{u}(k)$$

# Pressure elimination

Including incompressibility we get:

$$\dot{\mathbf{u}}(k) + \mathbf{F}(k) = -i\mathbf{k}P(k) - \nu|\mathbf{k}|^2\mathbf{u}(k) \tag{10}$$

$$\mathbf{k} \cdot \mathbf{u} = 0$$

- Poisson equation for pressure reduces to:

$$|\mathbf{k}|^2 P(k) = i\mathbf{k} \cdot \mathbf{F}(k) \tag{11}$$

- instead of having to invert in physical space $-\nabla^2 P = \partial_i u_j \partial_j u_i$
- ALL TERMS ARE LINEAR

$$\dot{u}_i(k) + \big(\delta_{ij} - \frac{k_i k_j}{k^2}\big) F_j(k) = -\nu|k|^2 u_i(k) \tag{12}$$

Pressure acts to project the non-linear term onto the basis of incompressible functions satisfying $\mathbf{k} \cdot \mathbf{u} = 0$

# Fourier transformed non-linear term

$$F_i(k) = \int d^3 r e^{-ik\cdot r}\big(\int d^3 q e^{iq\cdot r} u_j(q)\big)\partial_j\big(\int d^3 p e^{ip\cdot r} u_i(p)\big) \quad (13)$$

$$\int d^3 r e^{-i(k-p-q)\cdot r} = \delta(-k+p+q)$$

$$\partial_j e^{ip\cdot r} = ipe^{ip\cdot r}$$

so

$$\mathbf{F}(k) = \sum_{k=p+q} i\big(\mathbf{p}\cdot\mathbf{u}(\mathbf{q})\big)\mathbf{u}(\mathbf{p}) \quad (14)$$

or

$$F_i(k) = \sum_{k=p+q} ip_j u_i(p) u_j(q)$$

Number of operations

- $n^3$ mesh, $\rightarrow$ roughly $n^3$ wavenumber operations,
- to get the nonlinear term,
- for each $\mathbf{k}$ you have to do another $n^3$ wavenumber sums
- Total operations: $n^3 \times n^3 = n^6$

TOO EXPENSIVE

# Full Fourier transformed Navier-Stokes equations.

$$\dot{u}_i(k) + \left(\delta_{ij} - \frac{k_i k_j}{k^2}\right) \sum_{k=p+q} i p_j u_i(p) u_j(q) = -\nu |k|^2 u_i(k) \quad (15)$$

- Linear parts cheap in Fourier space
- Nonlinear parts expensive in Fourier space

Finite-difference/physical space

- Linear parts expensive (All global $\nabla^2$ operations)
- Nonlinear parts cheap ($u \partial_x u \rightarrow$

$$u(x_i) \frac{u(x_{i+1}) - u(x_{i-1})}{2dx}$$

SOLUTION:

Combine approaches

STEPS

# Pseudospectral

- Calculate derivatives exactly in Fourier space
- Transform velocities and derivatives to physical space
- Calculate non-linear terms
- Transform non-linear terms back to Fourier space
- Solve forced Stokes equation exactly

No errors?
WRONG

Aliasing errors: products of Fourier transforms yield ghost terms.

## Construction of transforms

- $n$-grid points yields $n$ real numbers: $u(x_i)$, $i = 1, n$
- $n$-Fourier coefficients is $n$ complex numbers
- or $n$-Fourier coefficients is $2n$ real numbers
- pure Fourier transforms overspecify

Solution:

- real to half-complex transforms

- $$u(k) = \frac{1}{2\pi} \int e^{-ikx} dx = \left(\frac{1}{2\pi} \int e^{ikx} dx\right)^* = u^*(-k)$$

  and

  $$u(k = n/2 + 1) = u(-n/2 + 1)$$

  (transforms are periodic too)

- Therefore only specify the first $n/2$ complex Fourier coefficients.

- Actually, $u(k = 0)$ and $u(k = n/2 + 1)$ are real, but in the end get $n$ real numbers

- 2nd and 3rd directions:

- Use Fourier transforms, mapped between $-n/2$ and $n/2 + 1$

- This is because the first transform from real to half-complex already yields complex numbers.

# Aliasing

In the multiplication of the non-linear terms,

$$u(k = n/3 + 1)u(k = n/3 + 1) =$$
$$NL(k = 2n/3 + 1) = NL(k = -n/3 + 1) = NL^*(n/3 - 1)$$

Two high wavenumbers, which should create a still higher wavenumber, create a smaller wavenumber

## Solution

- Truncation
- Truncate all $|k| > n/3$
- This is called the 2/3rds number because you keep 2/3rds of the wavenumbers
- 1/3 for $-n/3 < k < 0$, 1/3 for $0 < k < n/3$
- It was not originally appreciated that this simple approach was the best.

Now we have a fast, efficient and accurate method. SPECTRAL ACCURACY. The standard against which any competitive method should be compared.

# Solution of Stokes

$$\dot{u}_i(k) + G_i(k) = -\nu|k|^2 u_i(k)$$

where $G_i(k) = (\delta_{ij} - \frac{k_i k_j}{k^2})F_j(k)$

- integrating factors: Know $u(k, t = T)$ and fix $F(k)$ at $t = T$. What is $u(k, t = T + \Delta t)$?

$$u(k, t = T + \Delta t) =$$

- implicit: $\dot{u}(k, t = T + \Delta t/2) \approx (u(k, t = T) - u(k, t = T + \Delta t))/dt$

- $u(k, t = T + \Delta t/2) \approx (u(k, t = T) - u(k, t = T + \Delta t))/2$

  $(u(k, t = T) - u(k, t = T + \Delta t))/dt + G(k, t = T) = -\nu|k|^2(u(k, t =$

  Now solve for $u(k, T + \Delta t)$. This scheme is Crank-Nicholson.

- Implicit can be second order on time and unconditionally stable.

# Semi-implict

- Non-linear terms are solved explicitly.
- Want higher-order time accuracy on non-linear terms
- Therefore add combinations of past or intermediate time-step results
- Methods: leapfrog, Adams-Bashforth, Runge-Kutta. I use 3rd-order Runge-Kutta
- Combining: Get $F(k)$ by an explicit algorithm
- Solve Stokes equation implicitly
- SEMI-IMPLICIT

What about non-periodic boundaries?

- Special cases: free-slip on velocity: sine and cosine transforms

- General cases: Must abandon pure spectral.

- Can maintain spectral accuracy and efficiency if Galerkin in one direction.

- Still must be spectral (or sine and cosine) in other directions