Case 3.5: Fourier Pseudo-Spectral Method

Brandon Cloutier, Benson K. Muite, Matteo Parsani

May 28, 2013

Numerical Methods Overview

- Spectral Methods: Dealing with periodic boundary conditions in all 3 spatial direction, so use Fourier basis functions.
- Parallel code uses library 2DECOMP&FFT is used for domain decomposition and parallel FFTs (FFTW).

Numerical Methods Overview

- Spectral Methods: Dealing with periodic boundary conditions in all 3 spatial direction, so use Fourier basis functions.
- Parallel code uses library 2DECOMP&FFT is used for domain decomposition and parallel FFTs (FFTW).
- Two different time stepping schemes are used:
  - Low order: Implicit midpoint rule (IMR) with a fixed time step solved by fixed point iteration.
  - High order: 4th-order explicit Runge-Kutta method developed by Carpenter-Kennedy with the Crank-Nicolson method for the viscous terms with fixed step.

Numerical Methods Overview

- Spectral Methods: Dealing with periodic boundary conditions in all 3 spatial direction, so use Fourier basis functions.
- Parallel code uses library 2DECOMP&FFT is used for domain decomposition and parallel FFTs (FFTW).
- Two different time stepping schemes are used:
  - Low order: Implicit midpoint rule (IMR) with a fixed time step solved by fixed point iteration.
  - High order: 4th-order explicit Runge-Kutta method developed by Carpenter-Kennedy with the Crank-Nicolson method for the viscous terms with fixed step.

Note

- Both time stepping schemes are done in Fourier space with the exception of the nonlinear terms that computed into real space.
- We avoid using a de-aliasing scheme by full resolving the simulation using a larger number of sample points.
- Verification: Code was tested against a exact solution (Shapiro, A. (1993)) for numerical accuracy comparison.

Note

- Both time stepping schemes are done in Fourier space with the exception of the nonlinear terms that computed into real space.
- We avoid using a de-aliasing scheme by full resolving the simulation using a larger number of sample points.
- Verification: Code was tested against a exact solution (Shapiro, A. (1993)) for numerical accuracy comparison.

Implicit Midpoint Rule

$$\frac{\mathbf{u}^{n+1,j+1}-\mathbf{u}^{n}}{\delta t}+\frac{\mathbf{u}^{n+1,j}+\mathbf{u}^{n}}{2}\cdot\nabla\left(\frac{\mathbf{u}^{n+1,j}+\mathbf{u}^{n}}{2}\right)$$
$$=\frac{\nabla\left[\Delta^{-1}\left(\nabla\cdot\left[(\mathbf{u}^{n+1,j}+\mathbf{u}^{n})\cdot\nabla(\mathbf{u}^{n+1,j}+\mathbf{u}^{n})\right]\right)\right]}{4}$$
$$+\Delta\frac{\mathbf{u}^{n+1,j+1}+\mathbf{u}^{n}}{2\mathrm{Re}},$$

Where u^n is the numerical solution at timestep t^n, and δt is the timestep size, and superscript j indicates the fixed point iterate.

Carpenter-Kennedy

Navier-Stokes divided into a linear,

$$l(u) = \frac{1}{\mathrm{Re}} \Delta u \tag{1}$$

and nonlinear part

$$\mathbf{g}(\mathbf{u}) = -\mathbf{u} \cdot \nabla \mathbf{u} + \nabla \left[\Delta^{-1} \left(\nabla \cdot [\mathbf{u} \cdot \nabla \mathbf{u}] \right) \right]. \tag{2}$$

Time stepping scheme is then,

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{g}(\mathbf{u}) + \mathbf{l}(\mathbf{u}) \tag{3}$$

Carpenter-Kennedy Algorithm

procedure Runge Kutta($\mathbf{u}$)
    $\mathbf{h} = \mathbf{0}$
    $\mathbf{u} = \mathbf{u}^n$
    for $k = 1 \to 5$ **do**
        $\mathbf{h} \leftarrow \mathbf{g}(\mathbf{u}) + \beta_k \mathbf{h}$
        $\mu = 0.5\delta t(\alpha_{k+1} - \alpha_k)$
        $\mathbf{v} - \mu \mathbf{l}(\mathbf{v}) = \mathbf{u} + \gamma_k \delta t \mathbf{h} + \mu \mathbf{l}(\mathbf{u})$
        $\mathbf{u} \leftarrow \mathbf{v}$
    end for
    $\mathbf{u}^{n+1} = \mathbf{u}$
end procedure

β, α, and γ are vectors of length 5 containing constants. Two levels of storage h and u.

Performance

- $\delta t = 0.005$ for 512^3 and $\delta t = 0.01$ for 256^3 grid points.
- For IMR scheme, fixed point iteration procedure was stopped once the difference between two successive iterates was less than 10^{-10} in l^∞ norm of velocity fields.

Method	Grid Size	Cores	Time Steps	Time(s)	$\frac{\text{Core Hours}}{\text{Time Step}}$
IMR	256^3	512	1000	4060	0.578
IMR	512^3	1024	500	9899	5.68
CK	512^3	4096	2000	7040	4.0

Table : Performance of Fourier pseudospectral code on Shaheen. IMR is an abbreviation for implicit midpoint rule and CK is an abbreviation for Carpenter–Kennedy.

Kinetic Energy Evolution

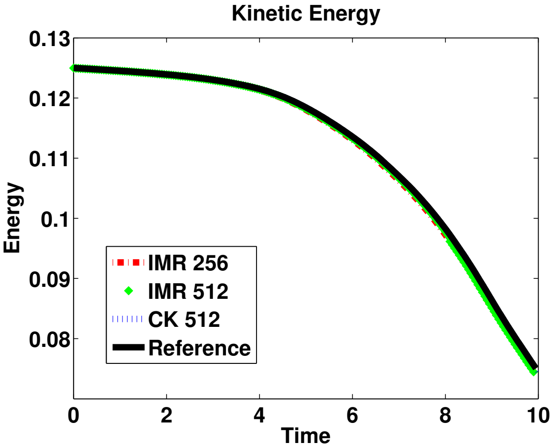

Figure : KE of solutions are so close they are almost indistinguishable

Kinetic Energy Dissipation Rate

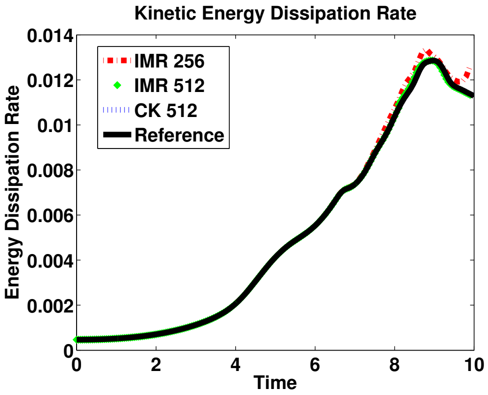

Figure : Plot during the initial stage, where flow is essentially inviscid and laminar. Fully developed turbulent flow is observed around $t_{max} \approx 8$.

Kinetic Energy Dissipation Rate

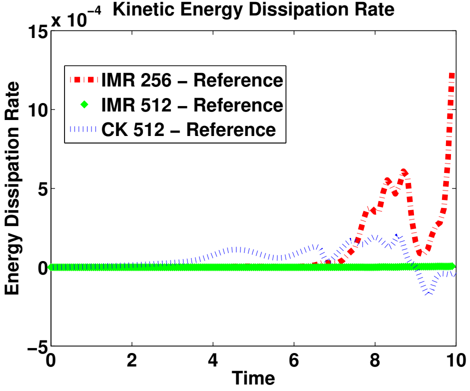

Figure : Difference in kinetic energy dissipation rates between the current discretizations and the reference solution.

Vorticity

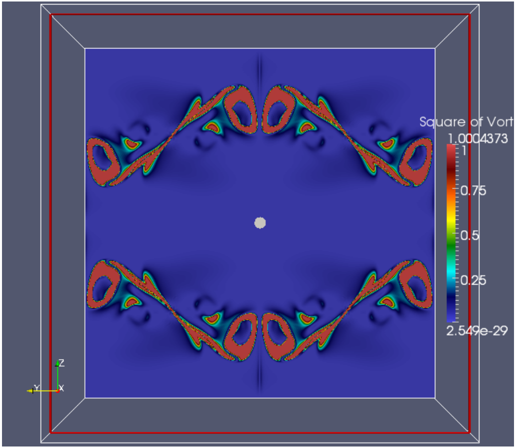

Figure : Square of the vorticity in the plane centered at $(\pi, 0, 0)$ with normal vector $(1, 0, 0)$.

Conclusion

- At almost the same computational cost, both 2nd-order accurate IMR and 4th-order Carpenter-Kennedy time stepping method, capture same amount of detail of the flow for 512^3.
- Simulations with 256^3 grid points resulted in poor spatial convergences.