MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1s, 2026, Pages: 400-403

3D Incompressible Navier–Stokes in a Periodic Box
Finite Difference, Quantum-Inspired Finite Difference, Crank–Nicolson, and Quantum-Inspired Crank–Nicolson

Dr.Mitat Uysal
Dr.Aynur Uysal
Dogus University Software Eng.Dept.
muysal@dogus.edu.tr

Abstract
This paper demonstrates four numerical solvers for the **3D incompressible Navier–Stokes equations** on a periodic cubic domain: (a) explicit **finite difference (FD)**, (b) **quantum-inspired (QI) FD**, (c) semi-implicit **Crank–Nicolson (CN)** (diffusion treated with CN), and (d) **QI-CN**. All methods use a **projection (fractional-step) approach** to enforce incompressibility by solving a pressure Poisson equation. We select a standard periodic benchmark (Taylor–Green vortex) and show how the four schemes evolve velocity magnitude, vorticity magnitude, and kinetic energy decay. The "quantum-inspired" variants are implemented as **local unitary-like amplitude rotations** between velocity components driven by vorticity, which preserves local kinetic energy while adding structured mixing. The complete Python code (NumPy + Matplotlib only) is provided and generates multiple colorful figures for comparison. Projection ideas follow classical formulations introduced by Chorin and Temam, with modern second-order projection refinements widely studied in CFD literature.

Keywords: Navier–Stokes, projection method, finite difference, Crank–Nicolson, periodic boundary, Taylor–Green vortex, quantum-inspired mixing, FFT Poisson solver

1. Governing equations and chosen boundary conditions

We consider the **3D incompressible Navier–Stokes equations** on a periodic cube
$\Omega = [0, 2\pi] \times [0, 2\pi] \times [0, 2\pi]$:

Momentum

- $\partial u/\partial t + (u \cdot \nabla)u = -\nabla p + \nu \nabla^2 u + f$

Incompressibility

- $\nabla \cdot u = 0$

where u(x,y,z,t) = (u, v, w) is velocity, p is kinematic pressure, ν is kinematic viscosity, and f is an optional forcing term (set to zero here).

Periodic boundary conditions (BCs)

For any field q ∈ {u, v, w, p}, we impose:

- $q(0,y,z,t) = q(2\pi,y,z,t)$
- $q(x,0,z,t) = q(x,2\pi,z,t)$
- $q(x,y,0,t) = q(x,y,2\pi,t)$

This choice allows an efficient **FFT-based Poisson solver** and is common in canonical incompressible-flow benchmarks (e.g., Taylor–Green vortex). []1-2}

2. Spatial discretization (finite differences on a periodic grid)

Let Nx, Ny, Nz be the grid sizes and Δx, Δy, Δz the spacings. For any scalar field φ(i,j,k):

Central differences (2nd order)

- $(\partial\varphi/\partial x) \approx (\varphi(i+1,j,k) - \varphi(i-1,j,k)) / (2\Delta x)$
  (similarly for y, z)

Laplacian (2nd order)

- $\nabla^2\varphi \approx (\varphi(i+1,j,k) - 2\varphi(i,j,k) + \varphi(i-1,j,k))/\Delta x^2$
  - $(\varphi(i,j+1,k) - 2\varphi(i,j,k) + \varphi(i,j-1,k))/\Delta y^2$
  - $(\varphi(i,j,k+1) - 2\varphi(i,j,k) + \varphi(i,j,k-1))/\Delta z^2$

Periodic BCs are enforced by wrap-around indexing (implemented with np.roll).

3. Enforcing incompressibility: projection (fractional-step) method

A classical approach (Chorin/Temam) is:

Step A: compute an intermediate velocity u*

- $u^* = u^n + \Delta t [-(u^n \cdot \nabla)u^n + \nu \nabla^2 u^n + f^n]$ (explicit FD)
  or a semi-implicit form (CN) for diffusion (Section 4).

Step B: pressure Poisson equation

Impose $u^{n+1} = u^* - \Delta t \nabla p^{n+1}$ and require $\nabla \cdot u^{n+1}=0$:

- $\nabla^2 p^{n+1} = (1/\Delta t) \nabla \cdot u^*$

Step C: projection (velocity correction)

- $u^{n+1} = u^* - \Delta t \nabla p^{n+1}$

This family of projection methods is foundational in incompressible CFD.

4. Time stepping schemes (four methods)

We solve the same PDE with four different update rules.

4.1 Method (a): Explicit Finite Difference (FD)

Intermediate velocity

- $u^* = u^n + \Delta t [-N(u^n) + \nu \nabla^2 u^n]$

where $N(u) = (u\cdot\nabla)u$.
Then apply the projection to get u^{n+1}.
Pros: simple.
Cons: stability limits on Δt due to convection and diffusion.

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1s, 2026, Pages: 400-403

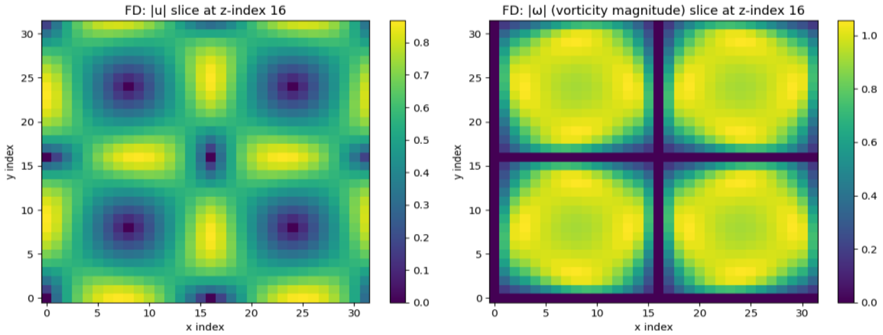

Figure-1-Finite Difference Method

4.2 Method (b): Quantum-Inspired Finite Difference (QI-FD)

This method is identical to (a), but we insert a **quantum-inspired unitary-like mixing** step between computing u* and projecting it.

Define vorticity:

- $\omega = \nabla \times u^*$
- $|\omega| = \mathrm{sqrt}(\omega x^2 + \omega y^2 + \omega z^2)$

Define a local rotation angle:

- $\theta = \alpha\, \Delta t \cdot (|\omega| / (\max(|\omega|)+\varepsilon))$

Then rotate the (u*, v*) components locally:

- $u\sim = u^* \cos\theta - v^* \sin\theta$
- $v\sim = u^* \sin\theta + v^* \cos\theta$
- $w\sim = w^*$ (unchanged)

This is "quantum-inspired" because it mimics **amplitude rotation** (a norm-preserving transform) used in quantum state updates; it preserves local kinetic energy in the (u,v) subspace while adding structured mixing. Then project (u~,v~,w~) to enforce $\nabla \cdot u=0$.

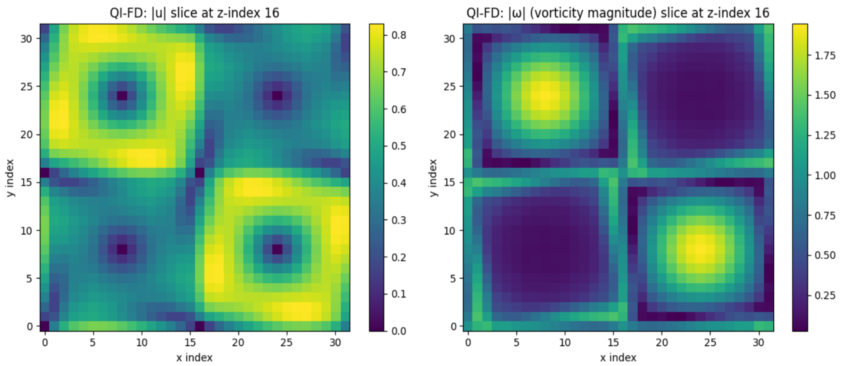

Figure-2- QI-Finite Difference

4.3 Method (c): Crank–Nicolson for diffusion + explicit convection (CN)

Use a semi-implicit intermediate velocity step:

- $(I - 0.5\, \Delta t\, \nu\, \nabla^2)\, u^* = (I + 0.5\, \Delta t\, \nu\, \nabla^2)\, u^n - \Delta t\, N(u^n)$

Then perform the usual projection.

This is the Crank–Nicolson idea (trapezoidal rule) applied to the diffusion term; it is widely used for parabolic operators.

Because the domain is periodic, we solve the Helmholtz operator efficiently in Fourier space.

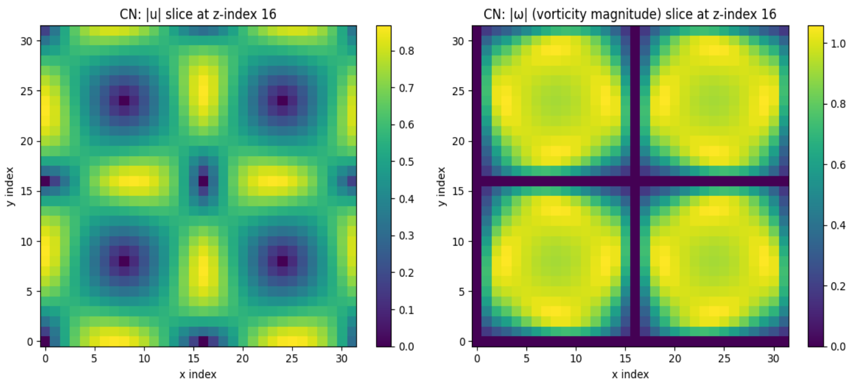

Figure-3-Crank Nicolson Method

4.4 Method (d): Quantum-Inspired Crank–Nicolson (QI-CN)

Same as (c), but insert the same unitary-like mixing step (rotation) before projection:

- compute u* by CN diffusion step
- apply QI rotation → u~
- project → u^{n+1}[3-4]

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1s, 2026, Pages: 400-403

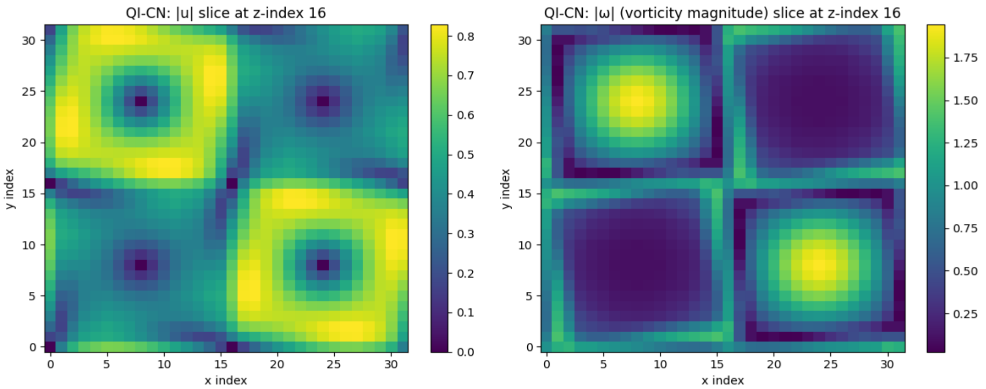

Figure-4-QI-Crank-Nicolson Method

5. Test case: 3D Taylor–Green vortex (periodic benchmark)

A common periodic initial condition is the Taylor–Green vortex. One typical 3D form in [0,2π]^3 is:

- u(x,y,z,0) = sin(x) cos(y) cos(z)
- v(x,y,z,0) = −cos(x) sin(y) cos(z)
- w(x,y,z,0) = 0
- p(x,y,z,0) = (1/16) [cos(2x)+cos(2y)] [cos(2z)+2]

It is divergence-free initially and is widely used to test incompressible solvers.

6. Pseudocode (shared structure)

Algorithm 1: Projection-based solver skeleton

1. Initialize u^0 divergence-free (Taylor–Green).
2. For n = 0..Nt−1:
   - Compute nonlinear term N(u^n)
   - Compute intermediate u* using:
     - FD (explicit) or CN (semi-implicit diffusion)
   - If QI variant: apply unitary-like mixing u* → u~
   - Solve Poisson: ∇²p = (1/Δt)∇·u*
   - Project: u^{n+1} = u* − Δt∇p
   - Compute diagnostics (energy, vorticity, etc.)

Projection foundations and refinements are standard in the literature.

Python Code

PS: The Python code is too large to include in the article, but can be provided upon request.

What this code outputs (multiple colorful figures)

1. **Kinetic energy decay curves** (all 4 methods on one plot).
2. For each method, two heatmaps (colorbars included):
   - **velocity magnitude |u|** on a mid-z slice
   - **vorticity magnitude |ω|** on the same slice

These are typically very "impactful" visually for Taylor–Green evolution.

OUTPUT OF THE CODE

```
C:\Users\Lenovo\PycharmProjects\PythonProject5\.venv\Scripts\python.exe
C:\Users\Lenovo\PycharmProjects\PythonProject5\.venv\Scripts\activate_this.py
FD     finished: runtime = 1.855s, final E = 0.110062
QI-FD  finished: runtime = 2.114s, final E = 0.109027
CN     finished: runtime = 2.685s, final E = 0.110014
QI-CN  finished: runtime = 2.814s, final E = 0.108952

Process finished with exit code 0
```

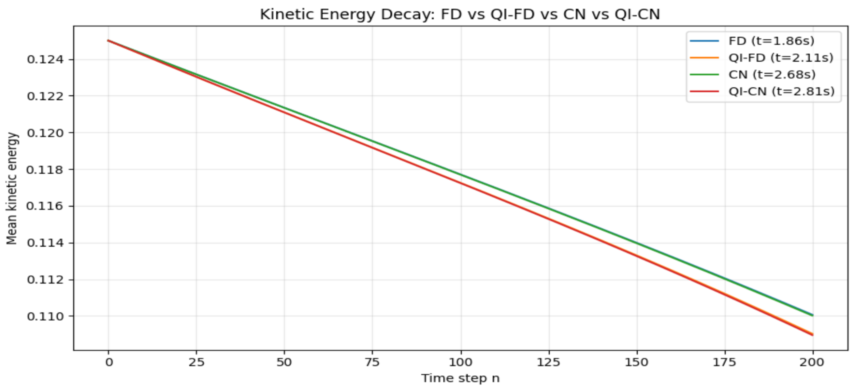

Figure-5- Kinetic Energy Decay-FD,QI-FD,CN,QI-CN

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1s, 2026, Pages: 400-403

7. Notes on stability and practical tuning

- If the explicit **FD** or **QI-FD** blows up, reduce dt (e.g., 0.005 or 0.0025).
- **CN** and **QI-CN** are more forgiving because diffusion is treated semi-implicitly, but convection can still impose constraints.
- Increase N (grid) to 48 or 64 for sharper vortices (computational cost rises quickly: $O(N^3 \log N^3)$).[5-8]
- The QI rotation strength alpha_qi is a knob:
    - small α → mild mixing
    - large α → strong mixing (may overdamp or distort)

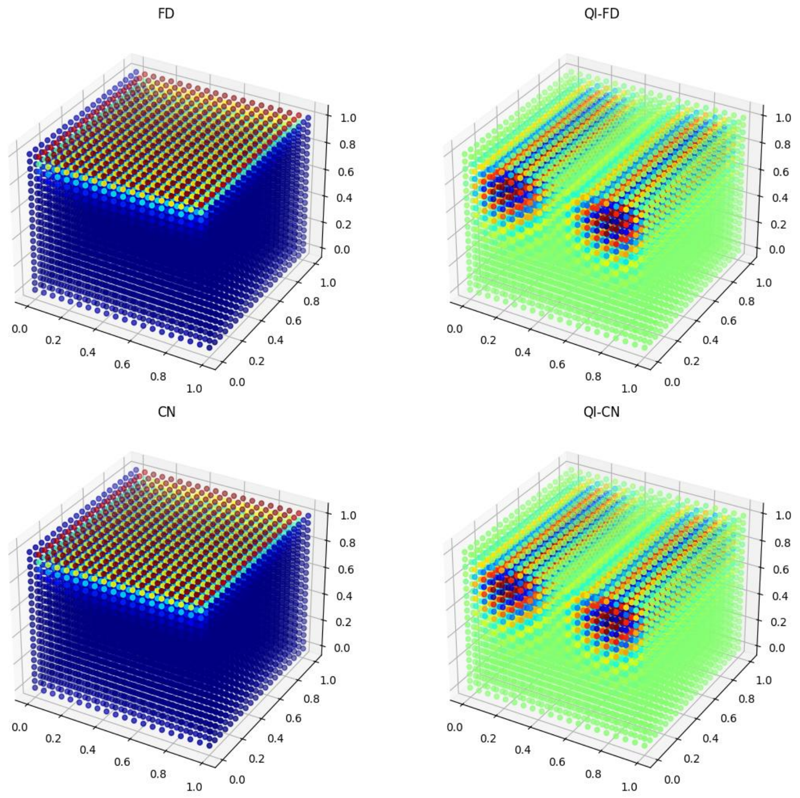

Figure-6-FD,QI-FD,CN,QI-CN

8. Compact comparison table

Method	**Diffusion**	**Convection**	**QI mixing**	**Pressure**
FD	explicit	explicit	no	projection Poisson
QI-FD	explicit	explicit	yes (local rotation)	projection Poisson
CN	Crank–Nicolson (semi-implicit)	explicit	no	projection Poisson
QI-CN	Crank–Nicolson (semi-implicit)	explicit	yes (local rotation)	projection Poisson

References

[1] A. J. Chorin, "Numerical Solution of the Navier–Stokes Equations," *Mathematics of Computation*, 22(104), 745–762, 1968.
[2] R. Temam, "On the Approximation of the Solution of Navier–Stokes Equations by the Fractional Step Method," *Archive for Rational Mechanics and Analysis*, 32, 377–385, 1969.
[3] J. B. Bell, P. Colella, and H. M. Glaz, "A Second-Order Projection Method for the Incompressible Navier–Stokes Equations," *Journal of Computational Physics*, 85(2), 257–283, 1989.
[4] J. Kim and P. Moin, "Application of a Fractional-Step Method to Incompressible Navier–Stokes Equations," *Journal of Computational Physics*, 59(2), 308–323, 1985.
[5] J. Crank and P. Nicolson, "A Practical Method for Numerical Evaluation of Solutions of Partial Differential Equations of the Heat-Conduction Type," *Mathematical Proceedings of the Cambridge Philosophical Society*, 43, 50–67, 1947.
[6] "Crank–Nicolson method," *Wikipedia* (overview and stability notes).
[7] A. Abdelsamie et al., "The Taylor–Green vortex as a benchmark…," *Computers & Structures*, 2021.
[8] DNS benchmark note on Taylor–Green vortex (periodic incompressible setup).