• Muddiest Points from Class 04/19

- *"MMS seems to be a nothing more than a hypothesis but seems to work out but I didn't catch quite everything about it."*
  - MMS is more than a hypothesis. It's a way to generate arbitrary exact analytical solutions to PDEs that allow for source terms (basically all PDEs).
  - Note: MMS does not give exact solutions to the original PDEs without a source term!
  - Exact solutions are needed to calculate errors exactly (and not just estimate them via GCI).

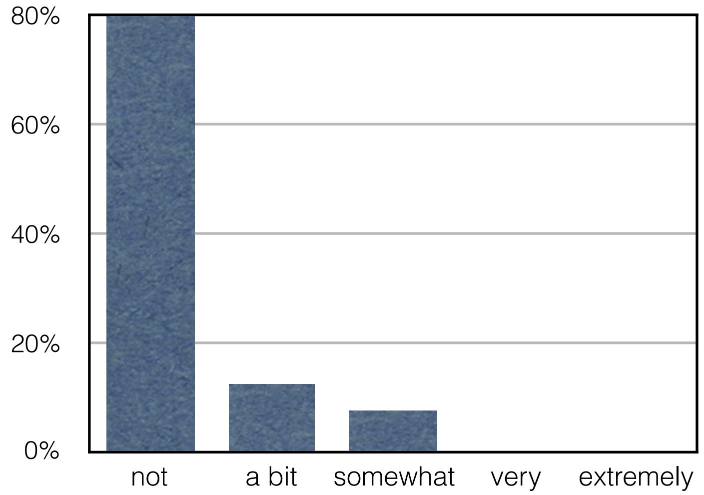
- *"I still confused about GCI analysis for steady state solution. How we exactly compare non-steady-state error with spatial error? should we do GCI analysis after a certain time period and then compare these two errors, then continue on the calculation, so on and on?"*
  - non-steady state errors are treated just the same as temporal errors or convergence errors.
  - You have to make sure that they are smaller than the spatial errors for spatial GCI.
  - non-steady state errors can be estimated by calculating dR/dt

- *"About the final project: Do you think you could explain how the right hand side would work when using the v-cycle for the Navier Stokes equation? In Homework 5 we were given a function f(x,y). How would this work for the Navier Stokes equation?"*
  - In HW5 we used an analytical function to calculate the numerical values of f_{ij} at every interior mesh point
  - For the fractional step method, simply use the given discrete equation to calculate the numerical values of f_{ij}

- *"Does it cause any problems at the end if we code our solution by making all the variables dimensionless ? (For any method)"*
  - No, it's actually preferred, since non-dimensional values are order 1, and thus have the full double precision accuracy of order 1 numbers
  - Yes, if you forget to dimensionalize your output results or make an error non-dimensionalizing all code variables (time!)

- Muddiest Points from Class 04/19

  - *“Can we still get bonus points if we only do some of the bonus problems (the ones we have time for) or will we need to do all the problems (is Homework 11 factored in like a normal homework)?”*
  - *“For the bonus homework, are we allowed to only do the problems for the outcomes we need, or should the entire assignment be completed?”*
    - Yes
  - *“how muddiest points will be considered in final scores? I mean average or total points?”*
    - Submission of each muddiest point survey results in 1 (or 2) homework bonus points added to the Homework section total score
  - *“Mostly unrelated: how can I work on CFD in industry? Are there companies that specialize in CFD, or is this mainly a field for academia? I really enjoy this class and I'd like to do this kind of work in the future!”*
    - Industry uses CFD more and more
    - Large companies with research devisions often have their own legacy codes
    - Even if companies use commercial codes (Ansys etc.) knowing (hopefully) how each numerical method performs is helpful to select appropriate solver choices
    - Solution verification (GCI) is crucial



Some (Random) Tips/Thoughts for the Final Project

- The cost is in the Poisson equation solution, so V-cycle rules
- To speed up V-cycle
  - do more iterations on coarser meshes than on finer meshes (it's cheaper there anyways)
  - go to the coarsest possible mesh in one of the directions, i.e. X by 1
- The cost is in the Poisson equation solution, so the fewer time steps the better: Implicit methods (Crank-Nicholson) for the parabolic terms rule

- So what about AEE471 students (all the above is MAE561)? or
  How can one get steady state solutions quickly (applies to MAE561 as well)?
  1) find the steady state solution on a coarse mesh
  2) interpolate this steady state solution (u,v,phi,Y) to a finer mesh
  3) use interpolated solution as initial solution for the finer mesh run and run to steady state
  4) repeat 2) - 4) for ever finer meshes as necessary
  - this directly gives data for the required GCI analysis at steady state!
  - be careful with the solution interpolation due to the staggered/centered mesh layout
  - to show the time evolution of the flow, start from rest with a reasonable fine mesh
  - Is this necessary for the final project? No, but it can reduce overall runtimes for the GCI a lot!

Advanced Considerations

Ideally, we would like our schemes to conserve mass, momentum, and energy

- for mass: ensure velocity is divergence free ⇒ converge Poisson system
- for momentum: use conservative form
- but what about energy, here kinetic energy?
  - another excursion into Linear Algebra:

    $$A\vec{x} = \vec{b}$$

    Q: are there any A, where $||\vec{x}|| = ||\vec{b}||$? A: Yes! For example if A is skew-symmetric.

    Q: What's the meaning of $||\vec{x}||$?

    $$||\vec{x}|| = \sum_i x_i^2 \qquad \text{so if } \vec{x} = \vec{v} \quad \Rightarrow \quad ||\vec{v}|| = 2E_{kin}$$

  - we can write our finite difference methods as $A\vec{v}^n = \vec{v}^{n+1}$

⇒ if A is skew-symmetric, the scheme will conserve kinetic energy!

Advanced Considerations

Skew-symmetric discretization on collocated grids

- let $\frac{\delta f_{i,j}}{\delta x} = \frac{f_{i+1,j} - f_{i-1,j}}{2\Delta x}$ $\qquad \frac{\delta f_{i,j}}{\delta y} = \frac{f_{i,j+1} - f_{i,j-1}}{2\Delta y}$

- skipping viscous terms:

$$\frac{\partial u}{\partial t} + \frac{1}{2}\frac{\delta(uu)}{\delta x} + \frac{1}{2}\frac{\delta(uv)}{\delta y} + \frac{1}{2}u\frac{\delta u}{\delta x} + \frac{1}{2}v\frac{\delta u}{\delta y} = -\frac{\delta \varphi}{\delta x}$$

$$\frac{\partial v}{\partial t} + \frac{1}{2}\frac{\delta(uv)}{\delta x} + \frac{1}{2}\frac{\delta(vv)}{\delta y} + \frac{1}{2}u\frac{\delta v}{\delta x} + \frac{1}{2}v\frac{\delta v}{\delta y} = -\frac{\delta \varphi}{\delta y}$$

Advanced Considerations

So what's the problem with collocated grids in the fractional step method?

or

Why are staggered meshes preferable?

- the reason has to do with step 2&3: div and grad operators must be consistent
- Why?
  - let's try different discrete operators: ∇_h and ∇'_h $\nabla_h\cdot$ and $\nabla'_h\cdot$
  - to get to step 2:

$$\frac{\vec{v}^{n+1} - \vec{v}^*}{\Delta t} = -\nabla_h \varphi^{n+1} \qquad \Big|\, \nabla_h \cdot$$

but different on either side

$$\frac{\nabla_h \cdot \vec{v}^{n+1} - \nabla_h \cdot \vec{v}^*}{\Delta t} = -\nabla'_h \cdot \nabla_h \varphi^{n+1}$$

  - use step 3 with different discrete operator: $\vec{v}^{n+1} = \vec{v}^* - \Delta t \nabla'_h \varphi^{n+1}$

$$\frac{\nabla_h \cdot \left(\vec{v}^* - \Delta t \nabla'_h \varphi^{n+1}\right) - \nabla_h \cdot \vec{v}^*}{\Delta t} = -\nabla'_h \cdot \nabla_h \varphi^{n+1}$$

$$\nabla_h \cdot \nabla'_h \varphi^{n+1} = \nabla'_h \cdot \nabla_h \varphi^{n+1}$$

$\Rightarrow$ true, only if $\nabla_h = \nabla'_h$ and $\nabla_h\cdot = \nabla'_h\cdot$

  - as we saw before: this is easy to achieve on staggered meshes

Advanced Considerations

$$\nabla_h \cdot \nabla'_h \varphi^{n+1} = \nabla'_h \cdot \nabla_h \varphi^{n+1}$$ ⇒ true, only if $\nabla_h = \nabla'_h$ and $\nabla_h \cdot = \nabla'_h \cdot$

- as we saw before: this is easy to achieve on **staggered** meshes
  - step 2: $\mathrm{div}\left(\mathrm{grad}\ \varphi^{n+1}\right) = \frac{1}{\Delta t}\mathrm{div}\left(\vec{v}^*\right)$

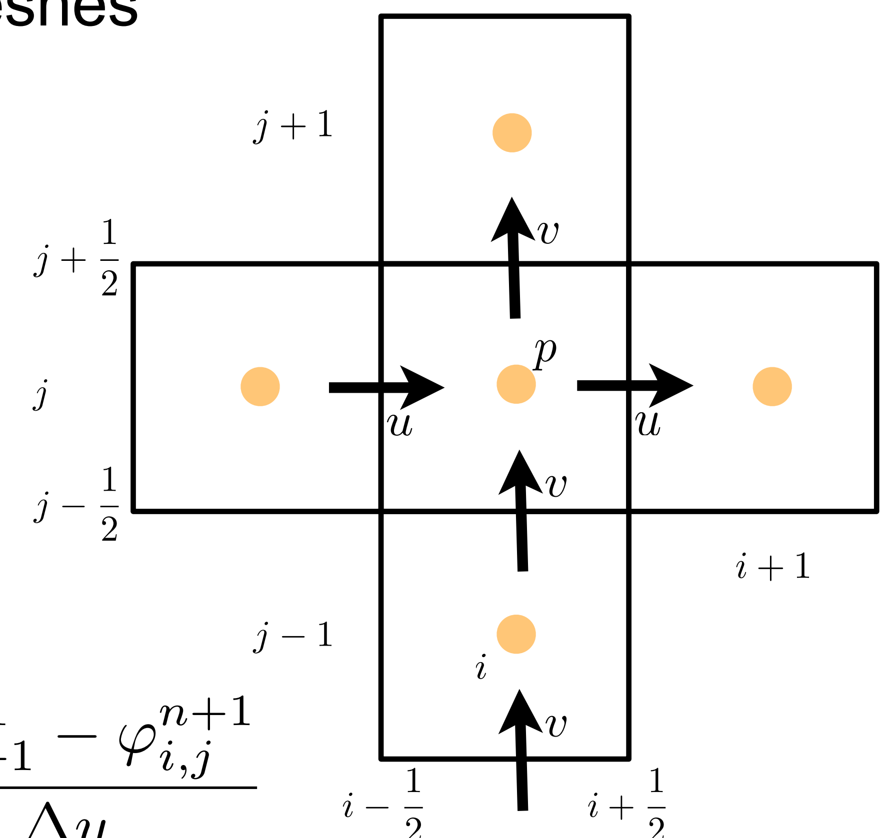
    ⇒ need div at (i,j):

    $$\mathrm{div}(\vec{v}^*) = \frac{u^*_{i+\frac{1}{2},j} - u^*_{i-\frac{1}{2},j}}{\Delta x} + \frac{v^*_{i,j+\frac{1}{2}} - v^*_{i,j-\frac{1}{2}}}{\Delta y}$$

  - step 3: $\vec{v}^{n+1} = \vec{v}^* - \Delta t\ \mathrm{grad}\ \varphi^{n+1}$

    ⇒ need grad_x at (i+1/2,j) and grad_y at (i,j+1/2):

    $$\mathrm{grad}_x(\vec{\varphi}^{n+1}) = \frac{\varphi^{n+1}_{i+1,j} - \varphi^{n+1}_{i,j}}{\Delta x} \qquad \mathrm{grad}_y(\vec{\varphi}^{n+1}) = \frac{\varphi^{n+1}_{i,j+1} - \varphi^{n+1}_{i,j}}{\Delta y}$$



  - thus div(grad) at (i,j) in step 2 is

$$\mathrm{div}(\mathrm{grad}(\vec{\varphi}^{n+1})) = \frac{\mathrm{grad}_x(\varphi)^{n+1}_{i+\frac{1}{2},j} - \mathrm{grad}_x(\varphi)^{n+1}_{i-\frac{1}{2},j}}{\Delta x} + \frac{\mathrm{grad}_y(\varphi)^{n+1}_{i,j+\frac{1}{2}} - \mathrm{grad}_y(\varphi)^{n+1}_{i,j-\frac{1}{2}}}{\Delta y}$$

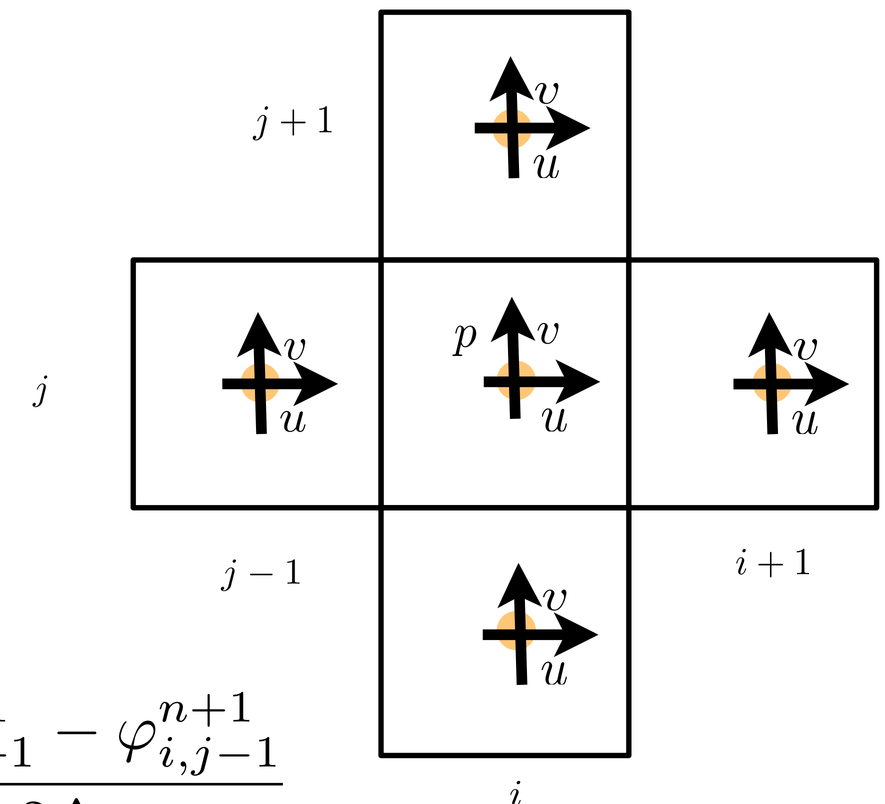
$$\mathrm{div}(\mathrm{grad}(\vec{\varphi}^{n+1})) = \frac{\frac{\varphi^{n+1}_{i+1,j}-\varphi^{n+1}_{i,j}}{\Delta x} - \frac{\varphi^{n+1}_{i,j}-\varphi^{n+1}_{i-1,j}}{\Delta x}}{\Delta x} + \frac{\frac{\varphi^{n+1}_{i,j+1}-\varphi^{n+1}_{i,j}}{\Delta y} - \frac{\varphi^{n+1}_{i,j}-\varphi^{n+1}_{i,j-1}}{\Delta y}}{\Delta y}$$

$$\mathrm{div}(\mathrm{grad}(\vec{\varphi}^{n+1})) = \frac{\delta^2_x \varphi^{n+1}_{i,j}}{\Delta x^2} + \frac{\delta^2_y \varphi^{n+1}_{i,j}}{\Delta y^2}$$

Advanced Considerations

$$\nabla_h \cdot \nabla'_h \varphi^{n+1} = \nabla'_h \cdot \nabla_h \varphi^{n+1}$$ ⇒ true, only if $\nabla_h = \nabla'_h$ and $\nabla_h \cdot = \nabla'_h \cdot$

- let’s look at collocated meshes



  - step 2: $\mathrm{div}\left(\mathrm{grad}\ \varphi^{n+1}\right) = \dfrac{1}{\Delta t}\mathrm{div}\left(\vec{v}^{*}\right)$

    ⇒ need div at (i,j):

$$\mathrm{div}(\vec{v}^{*}) = \frac{u^*_{i+1,j} - u^*_{i-1,j}}{2\Delta x} + \frac{v^*_{i,j+1} - v^*_{i,j-1}}{2\Delta y}$$

  - step 3: $\vec{v}^{n+1} = \vec{v}^{*} - \Delta t\ \mathrm{grad}\ \varphi^{n+1}$

    ⇒ need grad_x and grad_y at (i,j)

$$\mathrm{grad}_x(\vec{\varphi}^{n+1}) = \frac{\varphi^{n+1}_{i+1,j} - \varphi^{n+1}_{i-1,j}}{2\Delta x} \qquad \mathrm{grad}_y(\vec{\varphi}^{n+1}) = \frac{\varphi^{n+1}_{i,j+1} - \varphi^{n+1}_{i,j-1}}{2\Delta y}$$

  - thus div(grad) at (i,j) in step 2 is

$$\mathrm{div}(\mathrm{grad}(\vec{\varphi}^{n+1})) = \frac{\mathrm{grad}_x(\varphi)^{n+1}_{i+1,j} - \mathrm{grad}_x(\varphi)^{n+1}_{i-1,j}}{2\Delta x} + \frac{\mathrm{grad}_y(\varphi)^{n+1}_{i,j+1} - \mathrm{grad}_y(\varphi)^{n+1}_{i,j-1}}{2\Delta y}$$

$$\mathrm{div}(\mathrm{grad}(\vec{\varphi}^{n+1})) = \frac{\frac{\varphi^{n+1}_{i+2,j} - \varphi^{n+1}_{i,j}}{2\Delta x} - \frac{\varphi^{n+1}_{i,j} - \varphi^{n+1}_{i-2,j}}{2\Delta x}}{2\Delta x} + \frac{\frac{\varphi^{n+1}_{i,j+2} - \varphi^{n+1}_{i,j}}{2\Delta y} - \frac{\varphi^{n+1}_{i,j} - \varphi^{n+1}_{i,j-2}}{2\Delta y}}{2\Delta y}$$

$$\mathrm{div}(\mathrm{grad}(\vec{\varphi}^{n+1})) = \frac{\varphi^{n+1}_{i+2,j} - 2\varphi^{n+1}_{i,j} + 2\varphi^{n+1}_{i-2,j}}{4\Delta x^2} + \frac{\varphi^{n+1}_{i,j+2} - 2\varphi^{n+1}_{i,j} + 2\varphi^{n+1}_{i,j+2}}{4\Delta y^2}$$

Advanced Considerations

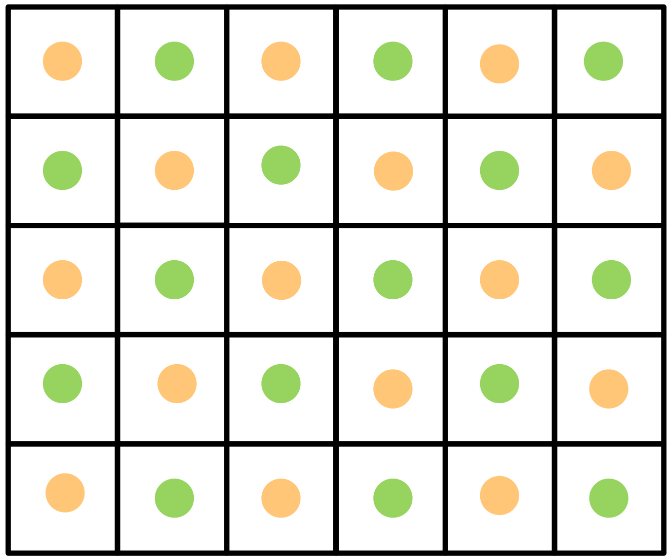
$$\nabla_h \cdot \nabla_h' \varphi^{n+1} = \nabla_h' \cdot \nabla_h \varphi^{n+1}$$ ⇒ true, only if $\nabla_h = \nabla_h'$ and $\nabla_h \cdot = \nabla_h' \cdot$

- let's look at collocated meshes
  - step 2: $\operatorname{div}\left(\operatorname{grad}\, \varphi^{n+1}\right) = \dfrac{1}{\Delta t}\operatorname{div}\left(\vec{v}^*\right)$

    ⇒ need div at (i,j):

$$\operatorname{div}(\vec{v}^*) = \frac{u^*_{i+1,j} - u^*_{i-1,j}}{2\Delta x} + \frac{v^*_{i,j+1} - v^*_{i,j-1}}{2\Delta y}$$

checker boarding!



  - step 3: $\vec{v}^{n+1} = \vec{v}^* - \Delta t \operatorname{grad}\, \varphi^{n+1}$

    ⇒ need grad_x and grad_y at (i,j)

$$\operatorname{grad}_x(\vec{\varphi}^{n+1}) = \frac{\varphi^{n+1}_{i+1,j} - \varphi^{n+1}_{i-1,j}}{2\Delta x} \qquad \operatorname{grad}_y(\vec{\varphi}^{n+1}) = \frac{\varphi^{n+1}_{i,j+1} - \varphi^{n+1}_{i,j-1}}{2\Delta y}$$

  - thus div(grad) at (i,j) in step 2 is

$$\operatorname{div}(\operatorname{grad}(\vec{\varphi}^{n+1})) = \frac{\operatorname{grad}_x(\varphi)^{n+1}_{i+1,j} - \operatorname{grad}_x(\varphi)^{n+1}_{i-1,j}}{2\Delta x} + \frac{\operatorname{grad}_y(\varphi)^{n+1}_{i,j+1} - \operatorname{grad}_y(\varphi)^{n+1}_{i,j-1}}{2\Delta y}$$

$$\operatorname{div}(\operatorname{grad}(\vec{\varphi}^{n+1})) = \frac{\frac{\varphi^{n+1}_{i+2,j} - \varphi^{n+1}_{i,j}}{2\Delta x} - \frac{\varphi^{n+1}_{i,j} - \varphi^{n+1}_{i-2,j}}{2\Delta x}}{2\Delta x} + \frac{\frac{\varphi^{n+1}_{i,j+2} - \varphi^{n+1}_{i,j}}{2\Delta y} - \frac{\varphi^{n+1}_{i,j} - \varphi^{n+1}_{i,j-2}}{2\Delta y}}{2\Delta y}$$

$$\operatorname{div}(\operatorname{grad}(\vec{\varphi}^{n+1})) = \frac{\varphi^{n+1}_{i+2,j} - 2\varphi^{n+1}_{i,j} + 2\varphi^{n+1}_{i-2,j}}{4\Delta x^2} + \frac{\varphi^{n+1}_{i,j+2} - 2\varphi^{n+1}_{i,j} + 2\varphi^{n+1}_{i,j+2}}{4\Delta y^2}$$

Advanced Considerations

$$\nabla_h \cdot \nabla'_h \varphi^{n+1} = \nabla'_h \cdot \nabla_h \varphi^{n+1}$$ $\Rightarrow$ true, only if $\nabla_h = \nabla'_h$ and $\nabla_h \cdot = \nabla'_h \cdot$

- explanation using linear algebra:
  - step 2: $\mathrm{div}\left(\mathrm{grad}\ \varphi^{n+1}\right) = \frac{1}{\Delta t}\mathrm{div}\left(\vec{v}^*\right)$

    $A\vec{\varphi}^{n+1} = \vec{b}$

    - what's the nullspace of A?

      ➡ for staggered meshes:

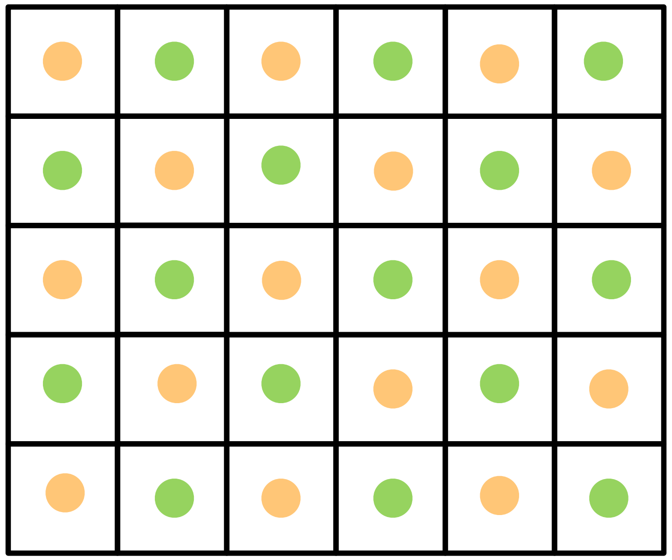
      $$N(A) = \alpha\,(1, 1, 1, \ldots, 1)$$

      ➡ for collocated meshes:

      basis of N(A) in 2D: $\{\widehat{\varphi}^0_{i,j} = 1;\ \widehat{\varphi}^1_{i,j} = -1^i;\ \widehat{\varphi}^2_{i,j} = -1^j;\ \widehat{\varphi}^2_{i,j} = -1^{i+j}\}$

      $\Rightarrow$ if φ^{n+1} is a solution, so is $\varphi^{n+1} + \sum_{l=0}^{2} a_l \widehat{\varphi}^l_{i,j}$

checker boarding!



➡ strategies to deal with this:

✓ Rhie-Chow interpolation (1983): adds dissipation $\Rightarrow$ destroys kinetic energy conservation

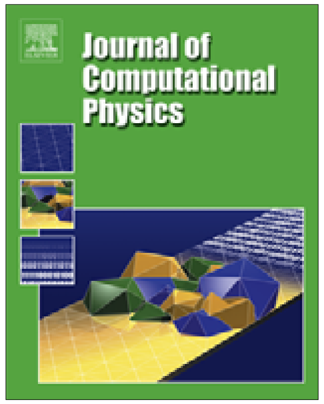
✓ Shashank et al., JCP (2010): find nullspace vector that minimizes local non-smoothness $\Rightarrow$ local least squares projection

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

ELSEVIER



Short Note

A co-located incompressible Navier–Stokes solver with exact mass, momentum and kinetic energy conservation in the inviscid limit

Shashank *, Johan Larsson, Gianluca Iaccarino

Flow Physics and Computational Engineering, Department of Mechanical Engineering, Stanford University, CA, USA

A R T I C L E   I N F O

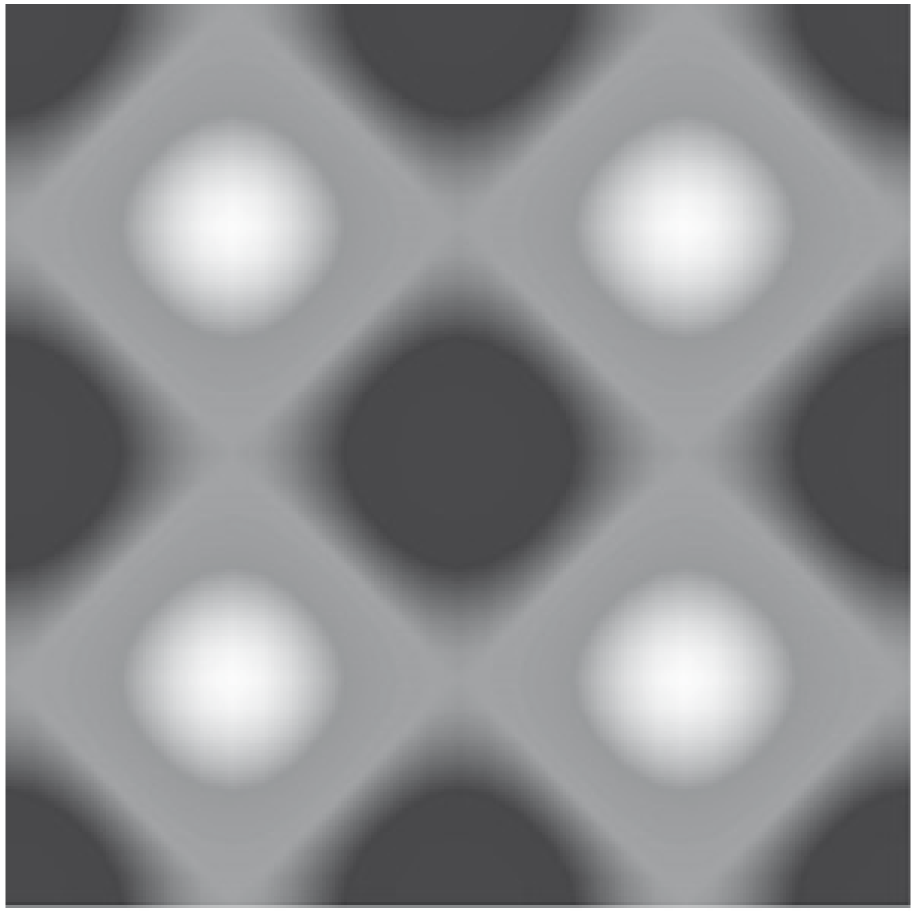
Article history:
Received 5 August 2009
Received in revised form 21 January 2010
Accepted 9 March 2010
Available online 17 March 2010

- Taylor Vortex
  - Test case with analytical solution to the 2D inviscid Navier-Stokes equations in periodic domains

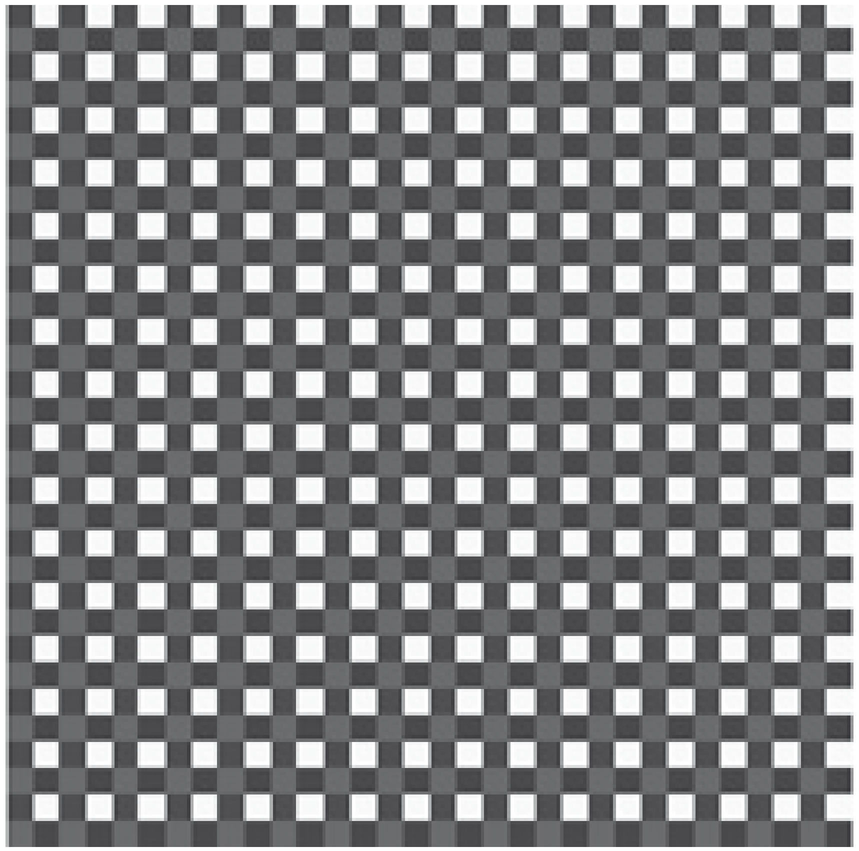
$$u = -\cos(\pi x)\sin(\pi y),$$
$$v = \sin(\pi x)\cos(\pi y),$$
$$p = -\frac{\cos(2\pi x) + \cos(2\pi y)}{4}$$

  - constant mass, momentum, and kinetic energy over time

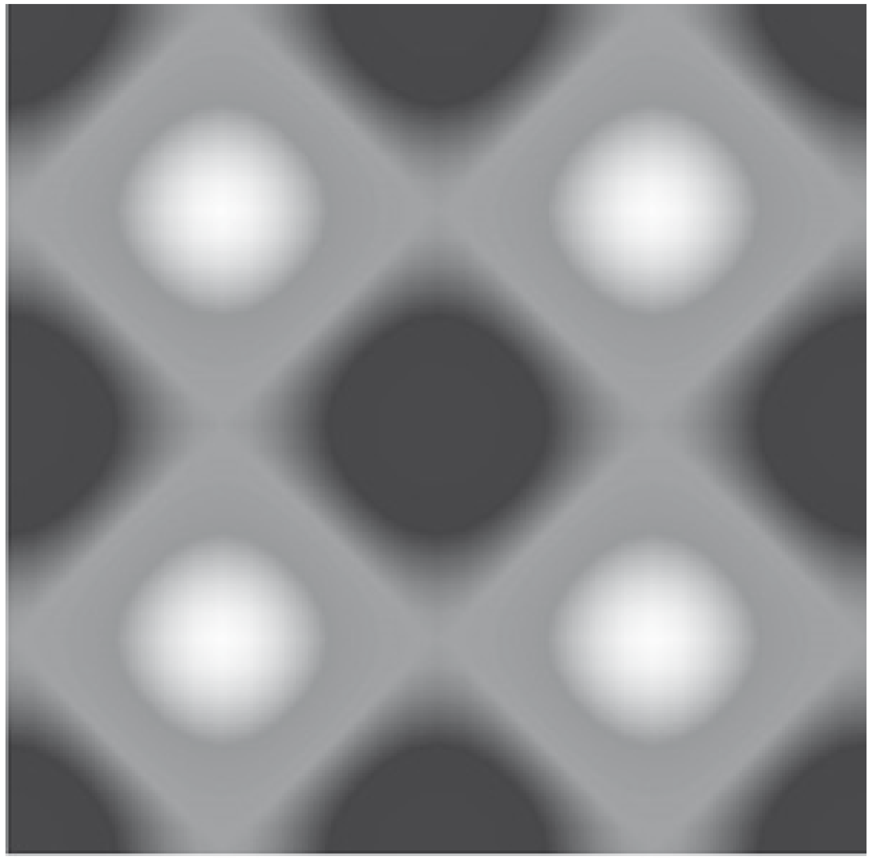
pressure contours



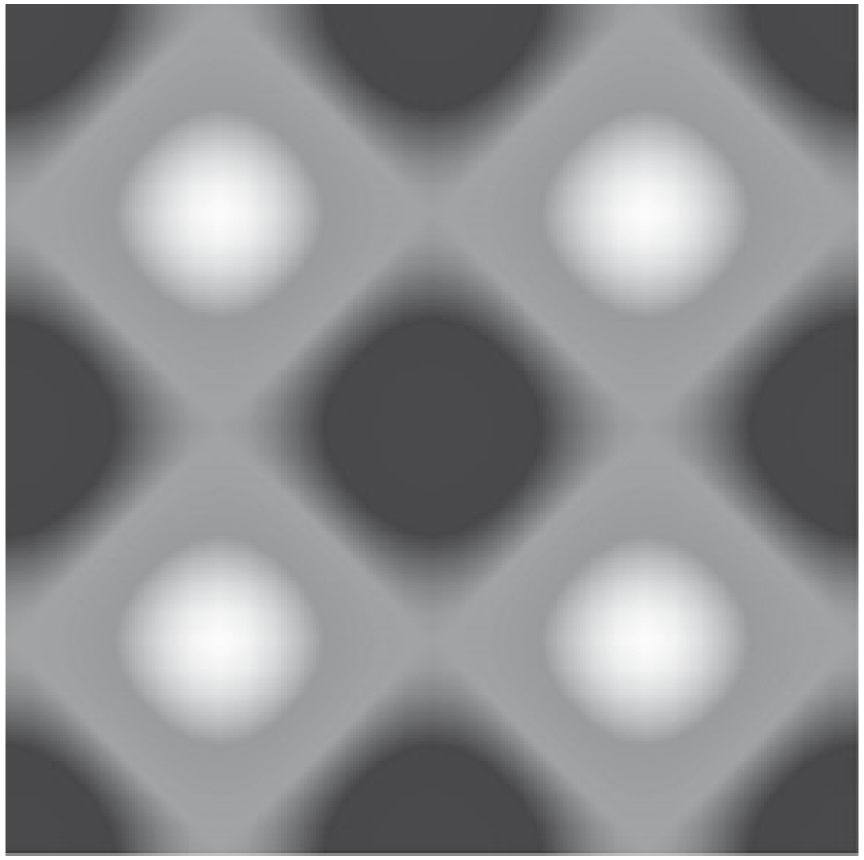
pressure contours



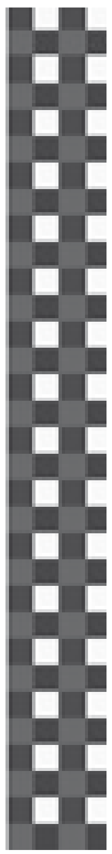
standard

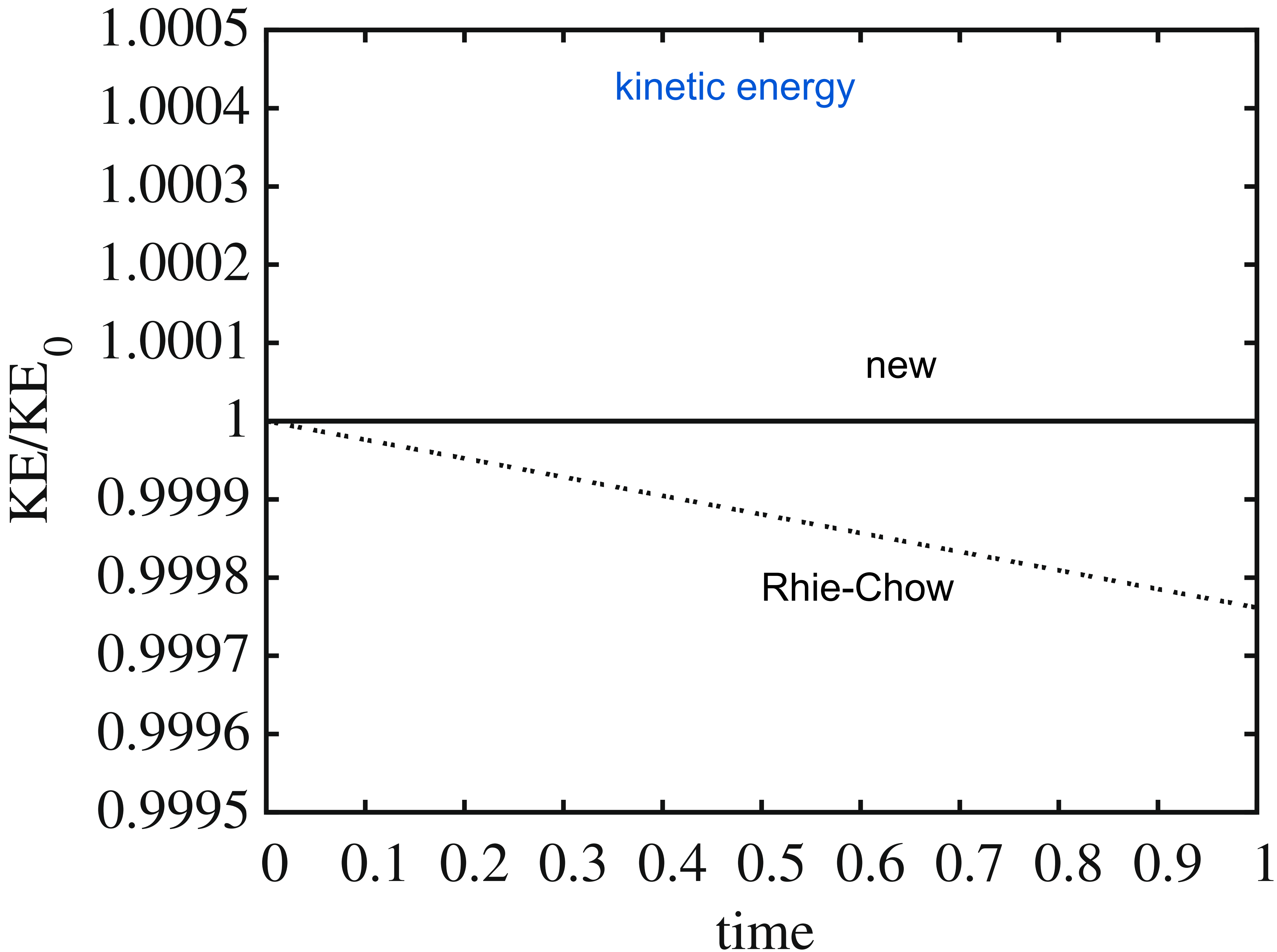


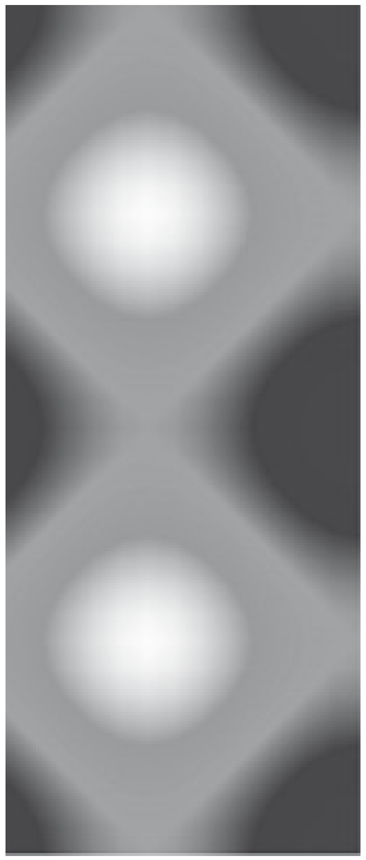
Rhie-Chow



new method

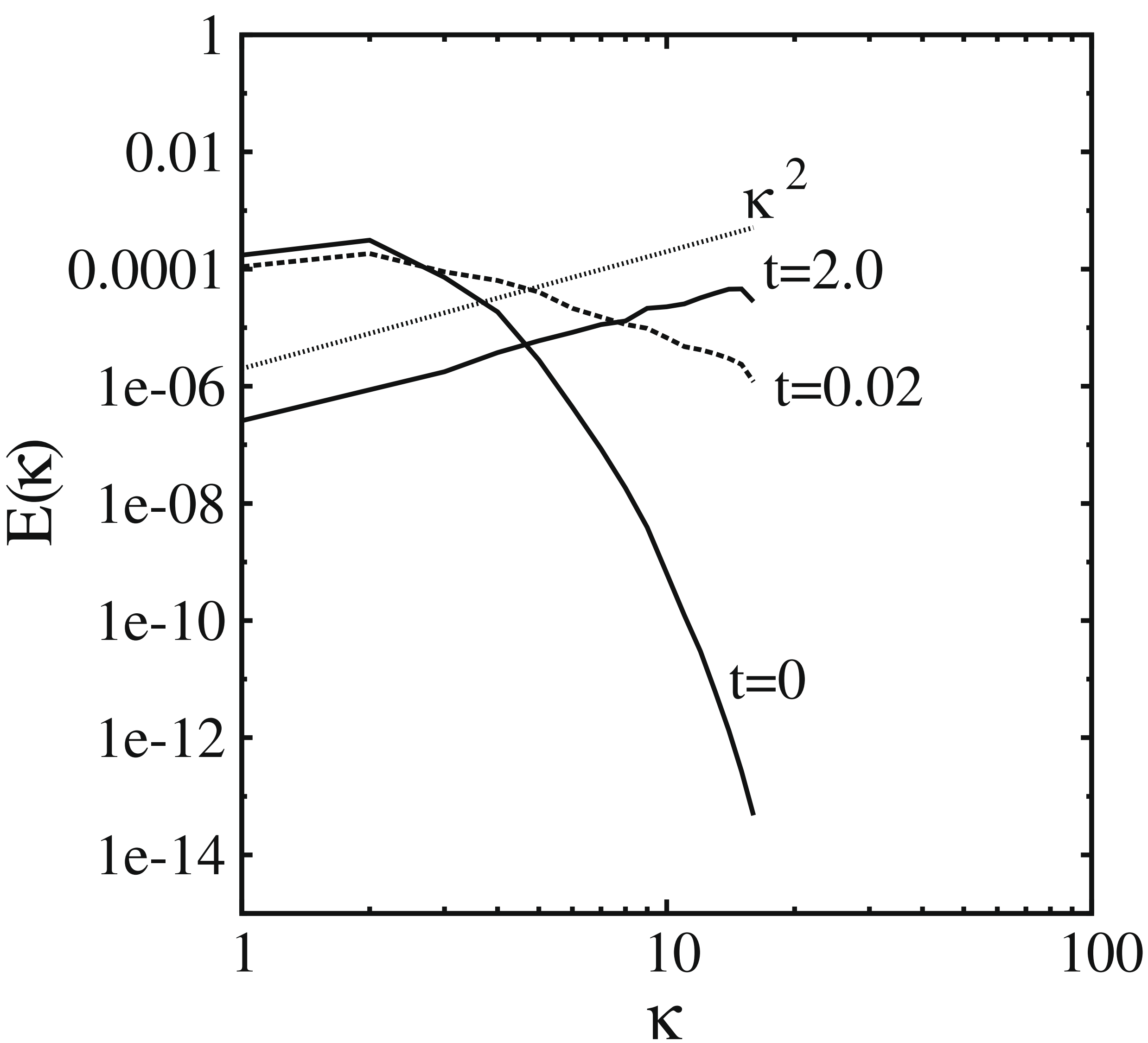



kinetic energy
new
Rhie-Chow
1.0005
1.0004
1.0003
1.0002
1.0001
1
0.9999
0.9998
0.9997
0.9996
0.9995
KE/KE_0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
time

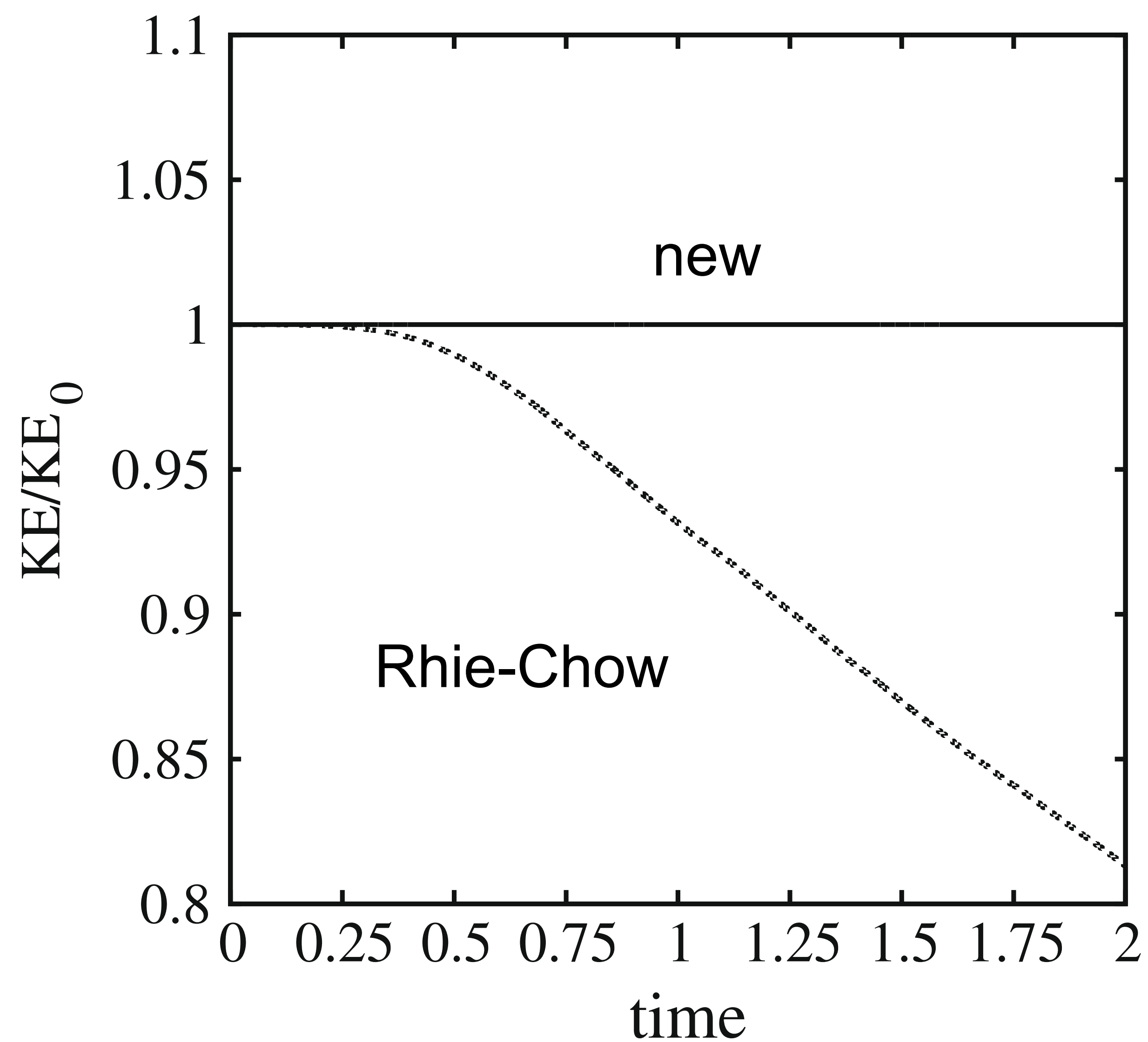


ethod

- Inviscid Turbulence in a 3D Periodic Box
  - Initial condition: Solenoidal random field with $E(\kappa) \propto \kappa^4 e^{-2\kappa^2}$



Temporal evolution of the energy spectrum



Temporal evolution of the total kinetic energy

- Please fill out the course evaluation forms online!
- Questions about the Final Project?