



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

Master Thesis in Engineering

High-Order Weighted Essentially Non-
Oscillatory (WENO) scheme for
Incompressible Navier-Stokes equations

University of Ulsan

School of Naval Architecture and Ocean Engineering

Nguyen Thi Quynh

High-Order Weighted Essentially Non-Oscillatory (WENO) scheme for Incompressible Navier-Stokes equations

Supervisor: Prof. Hyung Teak Ahn

Master thesis in Naval Architecture and Ocean Engineering

June 2022

School of Naval Architecture and Ocean Engineering
University of Ulsan, Republic of Korea


Nguyen Thi Quynh

Approval Sheet

This thesis is submitted in partial fulfilment of the requirements for the degree of Mater

June 2022

유정수 교수님, committee chairman 

최윤락 교수님, committee member 

안형택 교수님, committee member 

University of Ulsan

Preface

This thesis was developed during my work as a graduate student at the Advanced Computing Engineering Laboratory at the University of Ulsan, South Korea.

Many thanks to my doctoral supervisor Prof. Dr Hyung Teak Ahn for the exceptional working conditions in his research group, especially for the scientific freedom. I gained much more than I expected under his supervision. Not only in the research but also the life, he supported and advised me to help come over when I got the pressure while I am doing research or life. Furthermore, I thank all my laboratory members at Advanced Computing Engineering Laboratory for the good working atmosphere and all the fruitful scientific discussions. They made me feel like this is my second family. A special thank you to Deokhun Kim and Eunteak Lee.

I would like to thank my family for their continued encouragement, patience, love, and support through this journey.

Abstract

A new high-order projection method for simulating incompressible turbulent flows is presented based on the weighted essentially non-oscillatory scheme (WENO) schemes. In such a sense, this paper can be considered an extension of Zhang and Jackson's work (Zhang & L. Jackson, 2009) in terms of solution accuracy and computational efficiency. Unlike the previous work, the present method employed the Adams-Bashforth scheme for the nonlinear convection and the Crank-Nicolson scheme for the viscous term. For the spatial discretization, the WENO scheme is employed for the convection and standard central differences are used for the viscous. By the combination of successively higher orders of WENO and CD schemes, the desired order of accuracy was achieved without appreciable extra CPU time or memory overhead. More specifically the combination of WENO3/CD2, WENO5/CD4, and WENO7/CD6 achieved the third, fifth, and seventh order of spatial accuracies respectively. A verification study was presented both in space and time by using the 2D Taylor Green vortex problem. More challenging turbulent flows are simulated by solving the 3D Taylor-Green vortex problem for successively raised Reynolds numbers of $Re = 1,600, 16,000, \text{ and } 160,000$. The results, including the evolution of total kinetic energy, enstrophy, energy spectra, and local and global vortex field, support that the proposed method can be utilized for simulating the higher Reynolds number turbulent flows.

Contents

Preface.....	i
Abstract.....	ii
1 Introduction	1
2 Numerical method	4
2.1 Time discretization.....	4
2.2 Projection method	5
2.2.1 Helmholtz-Hodge decomposition.....	6
2.2.2 The non-incremental pressure correction scheme.....	6
2.2.3 Incremental pressure correction scheme.....	7
2.2.4 Incremental pressure correction in rotational form.....	8
2.3 Courant-Friedrichs-Lewy Condition.....	8
2.4 Weighted Essentially Non-Oscillatory scheme (WENO).....	9
2.4.1 Third order of WENO scheme.....	10
2.4.2 Fifth order of the WENO scheme.....	11
2.4.3 Seventh order of the WENO scheme.....	13
2.5 Central difference scheme (CD).....	15
3 Parallel computing.....	17
3.1 Overview of PETSc.....	17
3.2 Parallel Implementation	17
3.3 A solution method for discretized linear system.....	20
4 Simulation results	21
4.1 Two dimensional.....	21
4.1.1 Mesh convergence study with fixed CFL	22
4.1.2 Spatial (by fixing Δt) and temporal accuracy (by fixing Δx) in an independent manner	29
4.1.3 Execution time and memory requirement.....	31
4.2 Three dimensional.....	32
4.2.1 Moderate Reynolds number case, $Re = 1,600$	33
4.2.2 The higher Reynolds number cases, $Re = 16,000$ and $160,000$	46
5 Conclusions	49
References.....	50

1 Introduction

Ever since the birth of the numerical methods for the Navier-Stokes equations for incompressible flows in the 1960s (Harlow & Welch, 1965; Chorin, A numerical method for solving incompressible viscous flow problems, 1967; Chorin, Numerical solution of the Navier-Stokes equations, 1968) the algorithm has been continuously developed for higher accuracy and efficiency. Roughly speaking the solution method for the incompressible flows could be categorized into either a pressure-based method which solves the pressure-Poisson equation (Chorin, Numerical solution of the Navier-Stokes equations, 1968) for enforcing the divergence-free condition or the artificial compressibility method which allows the compressibility momentarily until it reaches the steady-state in pseudo-time where the divergence-free velocity field is recovered (Chorin, A numerical method for solving incompressible viscous flow problems, 1967). The representative pressure-based methods are the projection methods (Chorin, Numerical solution of the Navier-Stokes equations, 1968; Kim & Moin, 1985; Bell, Colell, & MGlaz, 1989; Brown, Cortez, & Minion, 2001). By utilizing the efficient MAC grid (Harlow & Welch, 1965) system on the Cartesian grid and efficient and scalable solution algorithms for Poisson-type equations, the method has been the primary choice of the incompressible flow community. On the other hand, the method of artificial compressibility has been developed by adopting the high-order algorithms originally developed for the compressible flow communities (Kwak, Chang, Shanks, & Chakravarthy, 1986; Kwak & Kiris, Computation of Viscous Incompressible Flows, 1989; Rogers & Kwak, Upwind differencing scheme for the time-accurate incompressible Navier-Stokes equations, 1990; Rogers, Kwak, & Kiris, Steady and unsteady solutions of the incompressible Navier-Stokes equations, 1991; Kiris, Kwak, Rogers, & Chang, 1997; Kallinderis & Ahn, 2005) far, both the methods attract the flow simulation practitioners based on their advantages, depending on the target application, grid system, and efficiency in their own linear and non-linear algebraic solvers.

The focus of our interest is on the first type of method, namely the projection method. More specifically a higher-order accurate and efficient projection method on the Cartesian MAC grid system. Since the introduction of the first projection method by Chorin (Chorin, Numerical solution of the Navier-Stokes equations, 1968) the second-order accuracy in time for the projection method has been successfully established by several researchers (Guermonda, Minev, & Shen, 2006; Kim & Moin, 1985). One of the first attempts was made by Kim and Moin (Kim & Moin, 1985), and their methods are widely applied to incompressible turbulent flows as well (Kim, Moin, & Moser, 1987). Later the method has been extended by adopting the upwinding idea based on the Godunov-type method. Such a method has been applied and proven to be stable for the convection-dominated high-speed incompressible flows. More recently, Brown et al. (Brown, Cortez, & Minion, 2001) presented a fully second-order projection method not only for the velocity but also for the pressure, which would be considered such as the standard second-order projection method.

Even though the consistent effort for achieving the second-order projection methods in time, the algorithms for higher-order ($> 2^{\text{nd}}$ order) accuracy in space was relatively scarce, hence a few pioneering attempts (Henshaw & Schwendeman, 2006; Zhang & L. Jackson, 2009) can be easily

accounted for. Henshaw achieved fourth-order accuracy in space using central differences on overlapping grids. In his approaches, the laminar flows with moderate Reynolds numbers are successfully simulated. For simulating high-Reynolds number turbulent flows on the meshes not as fine enough as a DNS (direct numerical simulation) level, pure central difference without any numerical or sub-grid scale eddy viscosity would likely incur numerical instabilities. Perhaps the natural choice for resolving this issue would be introducing the upwind effect like Bell et al. (Bell, Colell, & MGlaz, 1989) in the late 1980s. However, the formal introduction of the high-order upwind scheme to the projection method was first made by Zhang and Jackson (Zhang & L. Jackson, 2009) based on the weighted essentially non-oscillatory (WENO) scheme (Jiang & Shu, 1996).

WENO scheme is a general high-order scheme originally developed for the hyperbolic type of conservation laws in the framework of finite volume (Liu, Osher, & Chan, 1994), the finite difference (Jiang & Shu, 1996; Cockburn, Shu, Johnson, & Tadmor, 1997; Shu, Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, 1998; Shu, High Order Weighted Essentially Nonoscillatory Schemes for Convection Dominated Problems, 2009), and even discontinuous Galerkin method at least for limiters (Qiu & Shu, 2005; Luo, D.Baum, & Löhner, 2007; Zhu, Zhong, Shu, & Qiu, 2013). The method was originally evolved from its predecessor ENO scheme (Shu & Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, 1988; Shu & Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, II, 1989) which adaptively selected numerical stencils among several candidate stencils nearby. Instead of selecting only one of the candidate neighbouring stencils, the WENO scheme combines all of them with optimal weights for accurate yet non-oscillatory solution interpolation and reconstruction for finite difference and finite volume framework, respectively. Through this WENO process, one can achieve $2r - 1$ order of accuracy, where r is the number of solution points within each stencil, i.e., a combination of four adjacent 4-points stencils can achieve 7th order accuracy in space.

Even for the great success of the WENO scheme for the compressible flow communities (Jiang & Wu, A High-Order WENO Finite Difference Scheme for the Equations of Ideal Magnetohydrodynamics, 1999) and artificial compressibility-based incompressible flow simulations (Chen, Yang, & Yang, 1999), the application of projection-type incompressible flow simulation has been awaited until Zhang and Jackson (Zhang & L. Jackson, 2009). They applied the fifth-order WENO scheme both in advective and conservative form for the discretization of the nonlinear convective term and the sixth-order compact finite difference scheme for the linear viscous term (K. Lele, 1992). For time advancement, they employed an alternating 4/6 stages of low-dissipation and low-dispersion Runge-Kutta (LDDRK) scheme (Hu, Hussaini, & Manthey, 1996) where each RK stage involves a separate Poisson problem and an additional linear system incurred by the compact scheme, i.e., at each time step four or six Poisson solutions in addition to the velocity derivatives from the compact finite differences.

This paper is about an extension of Zhang and Jackson's work in terms of both the accuracy perspective and efficiency point of view. Unlike their previous work, the baseline projection method is presented as Pm2 in Brown et al. (Brown, Cortez, & Minion, 2001) which is second-

order accurate in time for both velocity and pressure. With this method, the convective term is discretized by the explicit Adams-Bashforth scheme and viscous terms are with the implicit Crank-Nicolson scheme. In short, this scheme is commonly referred to as the ABCN (Kim & Moin, 1985; Perot, 1993; Johnston & Liu, 2004) method for the incompressible Navier-Stokes equations. The beauty of this method is that it requires just a single solution of intermediate velocity field and a Poisson solution at each time step. We then apply a series of high-order spatial discretization schemes, namely the third, fifth, and seventh order WENO schemes (WENO3, WENO5, WENO7) for the nonlinear convection terms, and the second, fourth, and sixth-order standard central differences (CD2, CD4, CD6) for the viscous term. Among the various combination of WENO schemes and central differences, an optimal combination, which is WENO(k)/CD($k - 1$) is presented. The optimal combination of WENO (k) /CD ($k - 1$) is supported by the error convergence based on $L2$ and infinity norms by simulating the 2D Taylor-Green vortex (TGV2d) problem. The performance of the proposed methods for turbulent flow simulation is demonstrated by simulating triply periodic homogenous decaying turbulent flows of the 3D Taylor-Green vortex (TGV3d) problem with successively higher Reynolds numbers, i.e., $Re = 1,600, 16,000, \text{ and } 160,000$. The result compared the evolution of turbulent kinetic energy history, dissipation rate, local vorticity fields, global vortex structure, and kinetic energy spectra at various time moments.

2 Numerical method

2.1 Time discretization

Incompressible Newtonian viscous flows are governed by the Navier-Stokes equations

$$\begin{cases} \rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \mu \nabla^2 \mathbf{u} + f \\ \nabla \cdot \mathbf{u} = 0 \end{cases} \quad (2.1)$$

where $\mathbf{u} = (u, v, w)$, p , ρ and t are Cartesian velocity components, pressure, fluid density, and time, and then μ is dynamic viscosity, and f is body forces respectively. In three-dimensional space and time, there are four equations (3 momenta, one continuity) and four unknowns (3 velocity components and 1 pressure). The system of equations may be solved sequentially or all at the same time. Either way, the equations need to be discretized in both space and time. Here we consider methods for discretization in time and space.

For simplifying or reducing the parameters of the equation we use the non-dimensional equation (2.1). Following an appropriate selection of scales for the non-dimensional process, leads to the identification of small terms in the equation. Neglecting the smaller terms against the bigger ones allows for the simplification of the situation. For the case of flow without heat transfer, the non-dimensional Navier–Stokes equation depends only on the Reynolds Number and hence all physical realizations of the related experiment will have the same value of non-dimensional variables for the same Reynolds Number (Salvi, 2002). We used scale parameters L, U , which are the reference length and velocity, to construct the momentum equations non-dimensional. The scale parameters for non-dimensional variables are as follows

$$\mathbf{u}^* = \frac{\mathbf{u}}{U}, t^* = \frac{t}{L/U}, p^* = \frac{p - p_\infty}{\rho U^2}, x^* = \frac{x}{L}, y^* = \frac{y}{L}, z^* = \frac{z}{L}$$

Substituting these non-dimensional variables into equation (2.1) to obtain

$$\begin{cases} \left(\frac{U}{L/U} \right) \frac{\partial \mathbf{u}^*}{\partial t^*} + \left(\frac{U^2}{L} \right) (\mathbf{u}^* \cdot \nabla) \mathbf{u}^* = - \left(\frac{\rho U^2}{\rho L} \right) \nabla p^* + \left(\frac{\nu U}{L^2} \right) \nabla^2 \mathbf{u}^* + f \\ \left(\frac{U}{L} \right) \nabla \cdot \mathbf{u}^* = 0 \end{cases} \quad (2.2)$$

Dropping $*$ as a sign of dimensionless variables and dividing both sides for U^2/L , we get

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla p + \frac{1}{Re} \nabla^2 \mathbf{u} + f \\ \nabla \cdot \mathbf{u} = 0 \end{cases} \quad (2.3)$$

The Reynolds number (Re) is defined as $Re = UL/\nu$, respectively.

We used an explicit scheme with the r^{th} order Adams-Bashforth scheme for the convection term and an implicit scheme with the r^{th} order Adams-Moulton scheme for the diffusion term.

Adams-Bashforth scheme:

$$(\mathbf{u}^* \cdot \nabla) \mathbf{u}^* = \sum_{i=0}^r \gamma_i (\mathbf{u}^{n+1-i} \cdot \nabla) \mathbf{u}^{n+1-i} \quad (2.4)$$

where γ_i is the appropriate weight. Some situations are as follows

$r = 1$	$\gamma_1 = 1$	Explicit Euler
$r = 2$	$\gamma_1 = \frac{3}{2}, \gamma_2 = -\frac{1}{2}$	2 nd order Adams-Bashforth
$r = 3$	$\gamma_1 = \frac{23}{12}, \gamma_2 = -\frac{16}{12}, \gamma_3 = \frac{5}{12}$	3 rd order Adams-Bashforth

Adams-Moulton scheme:

$$\nabla^2 \mathbf{u}^* = \alpha_i \nabla^2 \mathbf{u}^{n+1} + \sum_{i=0}^{r-1} \beta_i \nabla^2 \mathbf{u}^{n+1-i} \quad (2.5)$$

where α_i and β_i are the appropriate weights. Some situations are as follows

$r = 1$	$\alpha_1 = 1, \beta_1 = 0$	Implicit Euler
$r = 2$	$\alpha_1 = \frac{1}{2}, \beta_1 = \frac{1}{2}$	Crank-Nicolson
$r = 3$	$\alpha_1 = \frac{5}{12}, \beta_1 = \frac{8}{12}, \beta_2 = -\frac{1}{12}$	3 rd order Adams-Moulton

We use the Crank-Nicolson (CN) for diffusion term and the second-order Adams-Bashforth (AB) for convection term to get the second order of accuracy. This has the advantage of achieving overall $O(\Delta t^2)$ accuracy while still allowing for a reasonable size of Δt . Now, the equation (2.1) can be written as

$$\begin{cases} \frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} + \frac{3}{2} (\mathbf{u} \cdot \nabla \mathbf{u})^n + \frac{1}{2} (\mathbf{u} \cdot \nabla \mathbf{u})^{n-1} = -\nabla p^{n+1} + \frac{1}{Re} \nabla^2 \left(\frac{\mathbf{u}^{n+1} + \mathbf{u}^n}{2} \right) + f \\ \nabla \cdot \mathbf{u}^{n+1} = 0 \end{cases} \quad (2.6)$$

The superscripts $n + 1$, n , and $n - 1$ denote the next, current, and previous time steps, respectively.

2.2 Projection method

The projection method (or fractional step method, time splitting method) is employed for the pressure-velocity decoupling, which is non-iterative and based on a mathematical foundation. The evolution of the velocity only consists of two sub-steps at each time level, i.e., the prediction and the projection. We will present this method in detail in the following content along with the error analysis (Shang-Gui, Abdellatif, Julien, & Yannick, 2017).

2.2.1 Helmholtz-Hodge decomposition

The projection method is based on the mathematical foundation of Helmholtz-Hodge decomposition, which indicates that any smooth vector \mathbf{u} can be decomposed into a divergence-free component \mathbf{u}_d and a curl-free component \mathbf{u}_c

$$\mathbf{u} = \mathbf{u}_d + \mathbf{u}_c \quad (2.7)$$

Where $\nabla \cdot \mathbf{u}_d = 0$, $\nabla \times \mathbf{u}_c = 0$ and the curl-free component can be further expressed as a gradient of a potential field $\mathbf{u}_c = \nabla \phi$. The divergence-free component \mathbf{u}_d can be obtained by taking the divergence operator (2.1), we get

$$\nabla^2 \phi = \nabla \cdot \mathbf{u} \quad (2.8)$$

$$\mathbf{u}_d = \mathbf{u} - \nabla \phi \quad (2.9)$$

In projection methods, \mathbf{u} is usually obtained in the prediction step and then projected into the divergence-free field in the projection step (Shang-Gui, Abdellatif, Julien, & Yannick, 2017).

2.2.2 The non-incremental pressure correction scheme

The first projection method is proposed by Chorin (Chorin, Numerical solution of the Navier-Stokes equations, 1968) and Témam (Témam, 1969). We consider the following two steps for solving equations (2.4) in a computational domain bounded by the wall Γ

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} + \frac{3}{2}(\mathbf{u} \cdot \nabla \mathbf{u})^n + \frac{1}{2}(\mathbf{u} \cdot \nabla \mathbf{u})^{n-1} = \frac{1}{Re} \nabla^2 \left(\frac{\mathbf{u}^{n+1} + \mathbf{u}^n}{2} \right) + f \quad (2.10)$$

$$\begin{cases} \frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} = -\nabla p^{n+1} \\ \nabla \cdot \mathbf{u}^{n+1} = 0, \quad \mathbf{u}^{n+1} \cdot \mathbf{n}|_{\Gamma} = 0 \end{cases} \quad (2.11)$$

Here represents a step where it is solved for a tentative velocity vector \mathbf{u}^* with no slip on the boundary. The pressure takes no part in the first step. The second step corrects velocity using the pressure gradient. The second step is implemented by taking the divergence of the correction and using the divergence-free condition. This leads to the following equation to be solved for p^{n+1}

$$\nabla^2 p^{n+1} = \frac{1}{\Delta t} \nabla \cdot \mathbf{u}^*, \quad \nabla p^{n+1} \cdot \mathbf{n}|_{\Gamma} = 0 \quad (2.12)$$

Followed by velocity being updated as

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \nabla p^{n+1} \quad (2.13)$$

Note that velocity update is performed without boundary conditions and as such it is only ensured that $\mathbf{u}^{n+1} \cdot \mathbf{n} = 0$ and in principle one may obtain non-zero velocities on Γ . For this reason, boundary conditions are often enforced also after the velocity update, which is a slight

inconsistency with the formulation in equations (2.8) and (2.9). This is a direct consequence of the splitting, and for the current scheme, the splitting error is of order $\mathcal{O}(\Delta t)$. In other words, our originally second-order temporal discretization error has been degraded to first order because of the splitting.

2.2.3 The incremental pressure correction scheme

We may improve the splitting error by including the pressure gradient in the tentative velocity step. By observing that the pressure gradient is not used in equation (2.8), we add and known old value of the pressure gradient in the prediction step to obtain a formally second-order accurate scheme. This formally second-order scheme is called the standard incremental pressure correction scheme in Guermond et al. (Guermonda, Minev, & Shen, 2006). The formally second-order method can be expressed as

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} + \frac{3}{2}(\mathbf{u} \cdot \nabla \mathbf{u})^n + \frac{1}{2}(\mathbf{u} \cdot \nabla \mathbf{u})^{n-1} = -\nabla p^n + \frac{1}{Re} \nabla^2 \left(\frac{\mathbf{u}^{n+1} + \mathbf{u}^n}{2} \right) + f \quad (2.14)$$

$$\begin{cases} \frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} = -\nabla \phi^{n+1} \\ \nabla \cdot \mathbf{u}^{n+1} = 0, \quad \mathbf{u}^{n+1} \cdot \mathbf{n}|_{\Gamma} = 0 \end{cases} \quad (2.15)$$

Where ϕ^{n+1} is a pressure correction. By applying the divergence operator to equation (2.14) along with the divergence-free condition

$$\nabla^2 \phi^{n+1} = \frac{1}{\Delta t} \nabla \cdot \mathbf{u}^*, \quad \nabla \phi^{n+1} \cdot \mathbf{n}|_{\Gamma} = 0 \quad (2.16)$$

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \nabla \phi^{n+1} \quad (2.17)$$

where the homogeneous Neumann boundary condition is enforced on ϕ . The final pressure is then updated by

$$p^{n+1} = p^k + \phi^{n+1} \quad (2.18)$$

We sum up the equation (2.12) and (2.13) and then compare them to the original system equation (2.4). Since the pressure correction is the approximation of $\phi^{n+1} = p^{n+1} - p^n = \Delta t \frac{\partial p}{\partial t}$, the splitting error is found to be of second-order (Perot, 1993; Armfield & Street, 2002).

$$\frac{1}{2Re} \nabla^2 (\mathbf{u}^* - \mathbf{u}^{n+1}) = \frac{\Delta t}{2} \nabla^2 (\nabla \phi^{n+1}) = \frac{\Delta t^2}{2} \nabla^2 (\nabla p) \quad (2.19)$$

However, the tentative velocity step is closer to the optimal, and as such the tentative velocity \mathbf{u}^* is closer to the final \mathbf{u}^{n+1} and thus the convective and diffusive terms are closer to the optimal (where \mathbf{u}^{n+1} is used instead of \mathbf{u}^*). It may be shown that the splitting error is now second-order accurate in time, which is the same as the temporal discretization.

2.2.4 Incremental pressure correction in rotational form

In (Timmermans, Minev, & Vosse, 1996; Guermonda, Minev, & Shen, 2006) describes a rotational incremental pressure correction. The basic idea behind this method is to incorporate the splitting error into the pressure, resulting in a sum of sub-steps that is consistent with the discretized momentum equation (2.6). By considering the identity $\nabla^2 \mathbf{u} = \nabla(\nabla \cdot \mathbf{u}) - \nabla \times \nabla \times \mathbf{u}$, the error term can be rewritten as

$$\frac{1}{2Re} \nabla^2 (\mathbf{u}^* - \mathbf{u}^{n+1}) = \frac{1}{2Re} \nabla(\nabla \cdot \mathbf{u}^*) \quad (2.20)$$

where $\nabla \times \nabla \times \mathbf{u}^* = \nabla \times \nabla \times \mathbf{u}^{n+1}$ is used, which can be verified by the Helmholtz-Hodge decomposition. Now the error term in this form is absorbed into the pressure

$$p^{n+1} = p^n + \phi^{n+1} - \frac{1}{2Re} \nabla \cdot \mathbf{u}^* \quad (2.21)$$

In this study, we have used the non-dimensional form of Navier-Stokes equations as equation (2.3). Applied the ABCN scheme and projection method for the equation above, we obtained the incremental pressure correction scheme in rotational form may be formulated as

$$\begin{cases} \frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} + \frac{3}{2} (\mathbf{u} \cdot \nabla \mathbf{u})^n - \frac{1}{2} (\mathbf{u} \cdot \nabla \mathbf{u})^{n-1} = -\nabla p^n + \frac{1}{2Re} [\nabla^2 \mathbf{u}^* + \nabla^2 \mathbf{u}^n] + f \\ \nabla^2 \phi^{n+1} = \frac{1}{\Delta t} \nabla \cdot \mathbf{u}^* \\ \mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \nabla \phi^{n+1}, \\ p^{n+1} = p^n + \phi^{n+1} - \frac{1}{2Re} \nabla \cdot \mathbf{u}^* \end{cases} \quad (2.22)$$

Algorithm 1 Rotational incremental pressure-correction projection method

Initialize $\mathbf{u}^{-1}, \mathbf{u}^0, p^0$ and $t = 0$, iteration

For $n = 1$: iteration

Solving the Helmholtz equations to get \mathbf{u}^*

$\mathbf{u}^{n-1} \leftarrow \mathbf{u}^n$

Solving the Poisson equation to get ϕ^{n+1}

Update \mathbf{u}^{n+1} and p^{n+1}

$p^n \leftarrow p^{n+1}$

end

2.3 Courant-Friedrichs-Lewy Condition

The Courant-Friedrichs-Lewy (CFL) condition is a necessary condition for the stability of any explicit one-level numerical scheme (Courant & Lewy, 1928). It establishes that the domain of dependence on the solution is contained in the numerical domain of dependence on the numerical method. In this work, for the low Reynolds, the CFL as

$$\Delta t \leq \frac{Re}{2} \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right)^{-1} \quad (2.23)$$

For the high Reynolds, the CFL as

$$\Delta t \leq \min \left\{ \frac{\Delta x}{\max(u)}, \frac{\Delta y}{\max(v)} \right\} \quad (2.24)$$

where $Re = \frac{UL}{\nu}$, L, U are the reference length and velocity, to construct the momentum equations non-dimensional. The u and v are velocities along x -direction and y -direction.

2.4 Weighted Essentially Non-Oscillatory scheme (WENO)

The convective term expressed in the advective form is discretized by using a series of high-order WENO schemes. The 3rd, 5th, and 7th order WENO schemes are illustrated for finite difference discretization of the convective terms. Finite difference WENO scheme is concerned with the accurate and stable solution interpolation at the cell-midpoint based on the combination of the reconstructions on the selected sub-stencils. Depending on the sign of the convective velocity, the appropriate upwind biased sub-stencil construction is illustrated for a series of high-order WENO schemes.

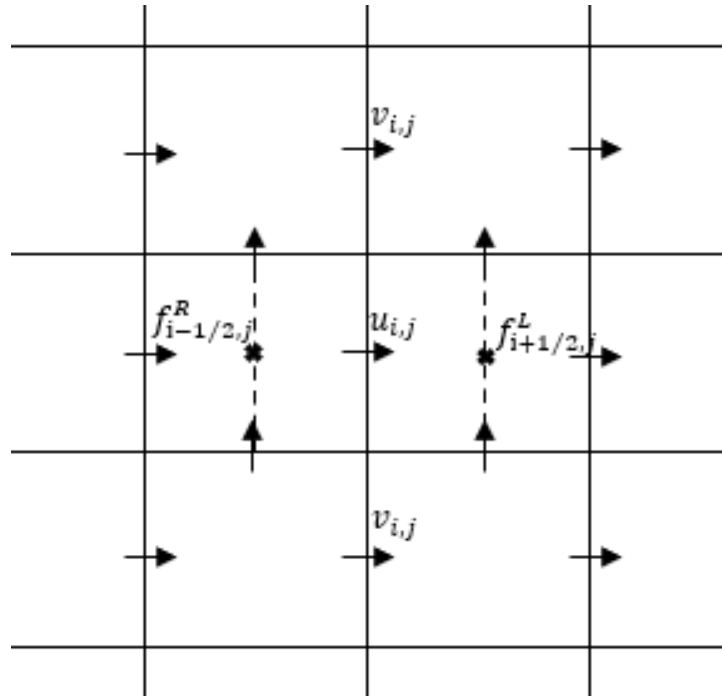


Fig. 1. The staggered mesh in two dimensions

Based on the 3rd, 5th, and 7th WENO schemes, the gradients of the convection term can be expressed as

$$\frac{\partial f}{\partial \eta} = \frac{\hat{f}_{i+1/2} - \hat{f}_{i-1/2}}{h} \quad (2.25)$$

where h is the distance between node $i + 1/2$ and node $i - 1/2$. The fluxes reconstructed \hat{f} can be written in the following form [11, 12, 13, 14].

$$f_{i+\frac{1}{2}}^L = \sum_{k=0}^N w_k f_{i+1/2}^k, \quad f_{i-\frac{1}{2}}^R = \sum_{k=0}^N w_k f_{i-1/2}^k$$

Where $f_{i+\frac{1}{2}}^L$, $f_{i-\frac{1}{2}}^R$, and w_k are polynomial reconstructed of f on the k^{th} set of stencils, the nonlinear weight of each sub-stencils. Nonlinear weights are defined as

$$w_k = \frac{\alpha_k}{\sum_{k=0}^N \alpha_k}, \quad \alpha_k = \frac{d_k}{(IS_k^{L/R} + \epsilon)^2}$$

Where $IS_k^{L/R}$, d_k and N donate smoothness indicators, the optimal linear weighting coefficients, and polynomials of degree. It is typical to set $\epsilon = 10^{-6}$ for eliminating zero denominators.

As previously stated, the signs of velocities in convective terms can naturally determine upwinding (Zhang & L. Jackson, 2009; A.Shetty, C.Fisher, Chunekar, & Frankel, 2010; Mishra, Par'es-Pulido, & Pressel, 2021). We use $u_{i,j}$ to calculate $f_{i+\frac{1}{2}}^L$ and $f_{i-\frac{1}{2}}^R$, following the upwind with a sign of $u_{i,j}$ as

$$\hat{f}_{i+1/2} = \begin{cases} f_{i+\frac{1}{2}}^L & \text{if } u_{i,j} \geq 0 \\ f_{i+\frac{1}{2}}^R & \text{otherwise} \end{cases}, \quad \hat{f}_{i-1/2} = \begin{cases} f_{i-\frac{1}{2}}^L & \text{if } u_{i,j} \geq 0 \\ f_{i-\frac{1}{2}}^R & \text{otherwise} \end{cases}$$

A polynomial is reconstructed from the solution's cell averages in each stencil, and then a weighted combination of all these polynomials is created. The smoothness of each polynomial in each stencil is considered when calculating the nonlinear weights.

In this section, we are going to present the WENO reconstruction in third, fifth, and seventh order for the following interpolation points $i + 1/2$ and $i - 1/2$.

2.4.1 Third-order of WENO scheme

The 3rd WENO scheme (WENO3) employs two sub-stencils, and each has two-point. To approximate the value of the function f at the points $i + 1/2$ and $i - 1/2$ by using polynomials of degree $N = 1$ with the final numerical flux resulting from a convex combination of two local solutions on the sub-stencils.

The stencil to get the corresponding values: $S_0 = \{i - 1, i\}$, $S_1 = \{i, i + 1\}$

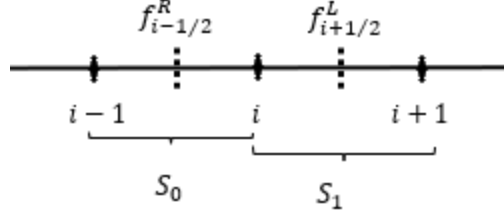


Fig. 2. Interpolation stencil for the WENO3 scheme, S_0, S_1 are the 2 stencils

The corresponding interpolated values for the point at $i + 1/2$ and $i - 1/2$ are

$$f_{i+1/2}^0 = -\frac{1}{2}f_{i-1} + \frac{3}{2}f_i, \quad f_{i-1/2}^0 = -\frac{1}{2}f_{i+1} + \frac{3}{2}f_i$$

$$f_{i+1/2}^1 = \frac{1}{2}f_i + \frac{1}{2}f_{i+1}, \quad f_{i-1/2}^1 = \frac{1}{2}f_i + \frac{1}{2}f_{i-1}$$

Thus, the reconstruction can be written as

$$f_{i+1/2}^L = w_0 f_{i+1/2}^0 + w_1 f_{i+1/2}^1$$

$$f_{i-1/2}^R = w_0 f_{i-1/2}^0 + w_1 f_{i-1/2}^1$$

The smoothness indicators

$$IS_0^L = (f_i - f_{i-1})^2, \quad IS_0^R = (f_{i+1} - f_i)^2$$

$$IS_1^L = (f_{i+1} - f_i)^2, \quad IS_1^R = (f_i - f_{i-1})^2$$

Where f is a local solution on each sub-stencil. The optimal linear weighting coefficients are $d_0 = 1/3$ and $d_1 = 2/3$.

2.4.2 Fifth-order of the WENO scheme

The 5th order WENO scheme (WENO5) employs three sub-stencils, and each has three-point. To approximate the value of the function f at the points $i + 1/2$ and $i - 1/2$ by using polynomials of degree $N = 2$ with the final numerical flux resulting from a convex combination of two local solutions on the sub-stencils.

The stencil to get the corresponding values: $S_0 = \{i - 2, i - 1, i\}, S_1 = \{i - 1, i, i + 1\}, S_2 = \{i, i + 1, i + 2\}$.

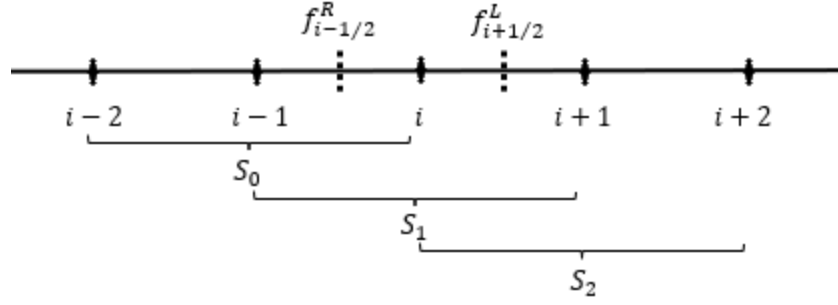


Fig. 3. Interpolation stencil for the WENO5 scheme, S_0, S_1, S_2 are the 3 stencils

The corresponding interpolated values for the point at $i + 1/2$ and $i - 1/2$ are

$$\begin{aligned}
 f_{i+1/2}^0 &= \frac{2}{6}f_{i-2} - \frac{7}{6}f_{i-1} + \frac{11}{6}f_i, & f_{i-1/2}^0 &= \frac{2}{6}f_{i+2} - \frac{7}{6}f_{i+1} + \frac{11}{6}f_i \\
 f_{i+1/2}^1 &= -\frac{1}{6}f_{i-1} + \frac{5}{6}f_i + \frac{2}{6}f_{i+1}, & f_{i-1/2}^1 &= -\frac{1}{6}f_{i+1} + \frac{5}{6}f_i + \frac{2}{6}f_{i-1} \\
 f_{i+1/2}^2 &= \frac{2}{3}f_i + \frac{5}{6}f_{i+1} - \frac{1}{6}f_{i+2}, & f_{i-1/2}^2 &= \frac{2}{6}f_i + \frac{5}{6}f_{i-1} - \frac{1}{6}f_{i-2}
 \end{aligned}$$

Thus, the reconstruction can be written as

$$\begin{aligned}
 f_{i+1/2}^L &= w_0 f_{i+1/2}^0 + w_1 f_{i+1/2}^1 + w_2 f_{i+1/2}^2 \\
 f_{i-1/2}^R &= w_0 f_{i-1/2}^0 + w_1 f_{i-1/2}^1 + w_2 f_{i-1/2}^2
 \end{aligned}$$

The smoothness indicators are defined as, for the point at $i + 1/2$

$$\begin{aligned}
 IS_0^L &= \frac{13}{12}(f_{i-2} - 2f_{i-1} + f_i)^2 + \frac{1}{4}(f_{i-2} - 4f_{i-1} + 3f_i)^2 \\
 IS_1^L &= \frac{13}{12}(f_{i-1} - 2f_i + f_{i+1})^2 + \frac{1}{4}(f_{i-1} - f_{i+1})^2 \\
 IS_2^L &= \frac{13}{12}(f_i - 2f_{i+1} + f_{i+2})^2 + \frac{1}{4}(3f_i - 4f_{i+1} + f_{i+2})^2
 \end{aligned}$$

For the point at $i - 1/2$ are

$$\begin{aligned}
 IS_0^R &= \frac{13}{12}(f_{i+2} - 2f_{i+1} + f_i)^2 + \frac{1}{4}(f_{i+2} - 4f_{i+1} + 3f_i)^2 \\
 IS_1^R &= \frac{13}{12}(f_{i+1} - 2f_i + f_{i-1})^2 + \frac{1}{4}(f_{i+1} - f_{i-1})^2 \\
 IS_2^R &= \frac{13}{12}(f_i - 2f_{i-1} + f_{i-2})^2 + \frac{1}{4}(3f_i - 4f_{i-1} + f_{i-2})^2
 \end{aligned}$$

The optimal linear weighting coefficients are $d_0 = \frac{1}{10}, d_1 = \frac{6}{10}, d_2 = \frac{3}{10}$

2.4.3 Seventh-order of the WENO scheme

The 7th order WENO scheme (WENO7) employs four sub-stencils, and each has four-point. To approximate the value of the function f at the points $i + 1/2$ and $i - 1/2$ by using polynomials of degree $N = 2$ with the final numerical flux resulting from a convex combination of two local solutions on the sub-stencils.

The stencil to get the corresponding values: $S_0 = \{i - 3, i - 2, i - 1, i\}$, $S_1 = \{i - 2, i - 1, i, i + 1\}$, $S_2 = \{i - 1, i, i + 1, i + 2\}$, $S_3 = \{i, i + 1, i + 2, i + 3\}$.

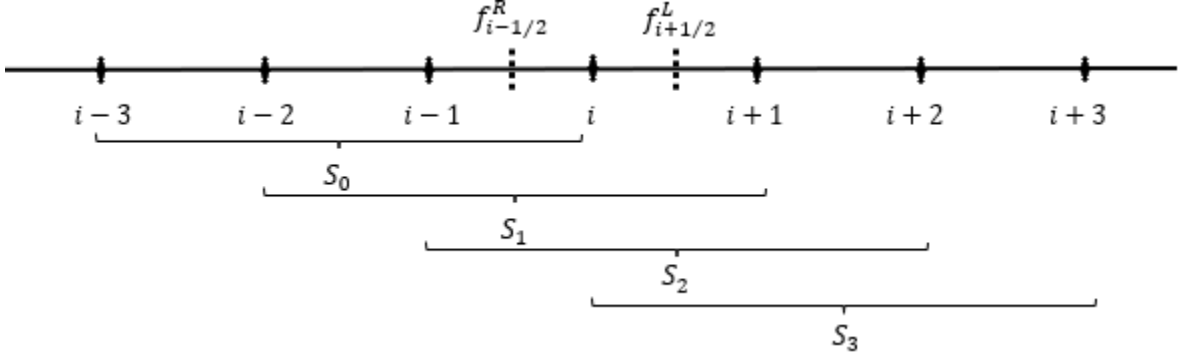


Fig. 4. Interpolation stencil for the WENO7 scheme, S_0, S_1, S_2, S_3 are the 4 stencils

The corresponding interpolated values for the point at $i + 1/2$ are

$$f_{i+1/2}^0 = -\frac{3}{12}f_{i-3} + \frac{13}{12}f_{i-2} - \frac{23}{12}f_{i-1} + \frac{25}{12}f_i$$

$$f_{i+1/2}^1 = \frac{1}{12}f_{i-2} - \frac{5}{12}f_{i-1} + \frac{13}{12}f_i + \frac{3}{12}f_{i+1}$$

$$f_{i+1/2}^2 = -\frac{1}{12}f_{i-1} + \frac{7}{12}f_i + \frac{7}{12}f_{i+1} - \frac{1}{12}f_{i+2}$$

$$f_{i+1/2}^3 = \frac{1}{4}f_i + \frac{13}{12}f_{i+1} - \frac{5}{12}f_{i+2} + \frac{1}{12}f_{i+3}$$

The corresponding interpolated values for the point at $i - 1/2$ are

$$f_{i-1/2}^0 = -\frac{3}{12}f_{i+3} + \frac{13}{12}f_{i+2} - \frac{23}{12}f_{i+1} + \frac{25}{12}f_i$$

$$f_{i-1/2}^1 = \frac{1}{12}f_{i+2} - \frac{5}{12}f_{i+1} + \frac{13}{12}f_i + \frac{3}{12}f_{i-1}$$

$$f_{i-1/2}^2 = -\frac{1}{12}f_{i+1} + \frac{7}{12}f_i + \frac{7}{12}f_{i-1} - \frac{1}{12}f_{i-2}$$

$$f_{i-1/2}^3 = \frac{1}{4}f_i + \frac{13}{12}f_{i-1} - \frac{5}{12}f_{i-2} + \frac{1}{12}f_{i-3}$$

Thus, the reconstruction can be written as

$$f_{i+\frac{1}{2}}^L = w_0 f_{i+1/2}^0 + w_1 f_{i+1/2}^1 + w_2 f_{i+1/2}^2 + w_3 f_{i+1/2}^3$$

$$f_{i-\frac{1}{2}}^R = w_0 f_{i-1/2}^0 + w_1 f_{i-1/2}^1 + w_2 f_{i-1/2}^2 + w_3 f_{i-1/2}^3$$

In which the smoothness indicators for the four-candidate-stencil of $f_{i+\frac{1}{2}}^L$ is defined as

$$\begin{aligned} IS_0^L &= f_{i-3}(547f_{i-3} - 3882f_{i-2} + 4642f_{i-1} - 185f_i) \\ &\quad + f_{i-2}(7043f_{i-2} - 17246f_{i-1} + 7042f_i) \\ &\quad + f_{i-1}(11003f_{i-1} - 9402f_i) + f_i(2107f_i) \end{aligned}$$

$$\begin{aligned} IS_1^L &= f_{i-2}(267f_{i-2} - 1642f_{i-1} + 1602f_i - 494f_{i+1}) \\ &\quad + f_{i-1}(2843f_{i-1} - 5966f_i + 1922f_{i+1}) \\ &\quad + f_i(3443f_i - 2522f_{i+1}) + f_{i+1}(547f_{i+1}) \end{aligned}$$

$$\begin{aligned} IS_2^L &= f_{i-1}(547f_{i-1} - 2522f_i + 1922f_{i+1} - 494f_{i+2}) \\ &\quad + f_i(3443f_i - 5966f_{i+1} + 1602f_{i+2}) \\ &\quad + f_{i+1}(2843f_{i+1} - 1642f_{i+2}) + f_{i+2}(267f_{i+2}) \end{aligned}$$

$$\begin{aligned} IS_3^L &= f_i(2107f_i - 9402f_{i+1} + 7042f_{i+2} - 1854f_{i+3}) \\ &\quad + f_{i+1}(11003f_{i+1} - 17246f_{i+2} + 4642f_{i+3}) \\ &\quad + f_{i+2}(7043f_{i+2} - 3882f_{i+3}) + f_{i+3}(547f_{i+3}) \end{aligned}$$

The smoothness indicators for the $f_{i-\frac{1}{2}}^R$

$$\begin{aligned} IS_0^R &= f_{i+3}(547f_{i+3} - 3882f_{i+2} + 4642f_{i+1} - 185f_i) \\ &\quad + f_{i+2}(7043f_{i+2} - 17246f_{i+1} + 7042f_i) \\ &\quad + f_{i+1}(11003f_{i+1} - 9402f_i) + f_i(2107f_i) \end{aligned}$$

$$\begin{aligned} IS_1^R &= f_{i+2}(267f_{i-2} - 1642f_{i+1} + 1602f_i - 494f_{i-1}) \\ &\quad + f_{i+1}(2843f_{i+1} - 5966f_i + 1922f_{i-1}) \\ &\quad + f_i(3443f_i - 2522f_{i-1}) + f_{i-1}(547f_{i-1}) \end{aligned}$$

$$\begin{aligned} IS_2^R &= f_{i+1}(547f_{i+1} - 2522f_i + 1922f_{i-1} - 494f_{i-2}) \\ &\quad + f_i(3443f_i - 5966f_{i-1} + 1602f_{i-2}) \\ &\quad + f_{i-1}(2843f_{i-1} - 1642f_{i-2}) + f_{i-2}(267f_{i-2}) \end{aligned}$$

$$\begin{aligned} IS_3^R &= f_i(2107f_i - 9402f_{i-1} + 7042f_{i-2} - 1854f_{i-3}) \\ &\quad + f_{i-1}(11003f_{i-1} - 17246f_{i-2} + 4642f_{i-3}) \\ &\quad + f_{i-2}(7043f_{i-2} - 3882f_{i-3}) + f_{i-3}(547f_{i-3}) \end{aligned}$$

The optimal linear weights are given by $d_0 = \frac{1}{35}$, $d_1 = \frac{12}{35}$, $d_2 = \frac{18}{35}$, $d_3 = \frac{4}{35}$

2.5 Central difference scheme (CD)

The diffusive terms are temporally discretized by the implicit Crank-Nicolson scheme. Hence, a linear system must be solved for the intermediate velocity fields. The standard central difference scheme is selected for the spatial discretization of the viscous term mainly because of its simplicity and computational efficiency. Depending on the choice of the WENO scheme for the nonlinear convection term, the order of the central difference (CD) scheme is also increased accordingly, i.e., second, fourth, and sixth orders. To be presented in the next section, this combination of WENO and CD is indeed quite efficient yet preserves the expected accuracy.

For the completeness of the discretization scheme, the central difference formulas of successive order of accuracy are presented briefly. By the symmetric Taylor series expansions concerning the point of interest (i), one can obtain central difference approximations of the second derivative

$$f_{i+3} = f_i + \frac{\partial f}{\partial \eta} 3h + \frac{1}{2!} \frac{\partial^2 f}{\partial \eta^2} 9h^2 + \frac{1}{3!} \frac{\partial^3 f}{\partial \eta^3} 27h^3 + \frac{1}{4!} \frac{\partial^4 f}{\partial \eta^4} 81h^4 + \dots$$

$$f_{i+2} = f_i + \frac{\partial f}{\partial \eta} 2h + \frac{1}{2!} \frac{\partial^2 f}{\partial \eta^2} 4h^2 + \frac{1}{3!} \frac{\partial^3 f}{\partial \eta^3} 8h^3 + \frac{1}{4!} \frac{\partial^4 f}{\partial \eta^4} 16h^4 + \dots$$

$$f_{i+1} = f_i + \frac{\partial f}{\partial \eta} h + \frac{1}{2!} \frac{\partial^2 f}{\partial \eta^2} h^2 + \frac{1}{3!} \frac{\partial^3 f}{\partial \eta^3} h^3 + \frac{1}{4!} \frac{\partial^4 f}{\partial \eta^4} h^4 + \dots$$

$$f_{i-1} = f_i - \frac{\partial f}{\partial \eta} h + \frac{1}{2!} \frac{\partial^2 f}{\partial \eta^2} h^2 - \frac{1}{3!} \frac{\partial^3 f}{\partial \eta^3} h^3 + \frac{1}{4!} \frac{\partial^4 f}{\partial \eta^4} h^4 + \dots$$

$$f_{i-2} = f_i - \frac{\partial f}{\partial \eta} 2h + \frac{1}{2!} \frac{\partial^2 f}{\partial \eta^2} 4h^2 - \frac{1}{3!} \frac{\partial^3 f}{\partial \eta^3} 8h^3 + \frac{1}{4!} \frac{\partial^4 f}{\partial \eta^4} 16h^4 + \dots$$

$$f_{i-3} = f_i - \frac{\partial f}{\partial \eta} 3h + \frac{1}{2!} \frac{\partial^2 f}{\partial \eta^2} 9h^2 - \frac{1}{3!} \frac{\partial^3 f}{\partial \eta^3} 27h^3 + \frac{1}{4!} \frac{\partial^4 f}{\partial \eta^4} 81h^4 + \dots$$

For instance, adding the two expansions f_{i+1} and f_{i-1} , gives

$$f_{i+1} + f_{i-1} = 2f_i + \frac{\partial^2 f}{\partial \eta^2} h^2 + \frac{1}{16} \frac{\partial^4 f}{\partial \eta^4} h^4 + \dots \quad (2.26)$$

So that,

$$\frac{f_{i+1} - 2f_i + f_{i-1}}{h^2} - \frac{\partial^2 f}{\partial \eta^2} = \frac{1}{16} \frac{\partial^4 f}{\partial \eta^4} h^4 + \dots$$

Hence, $\frac{f_{i+1}-2f_i+f_{i-1}}{h^2} = \frac{\partial^2 f}{\partial \eta^2}$ is a second-order centred differential approximation of the second derivative $\frac{\partial^2 f}{\partial \eta^2}$, so second-order centred differential (CD2) as

$$\frac{\partial^2 f}{\partial \eta^2} = \frac{f_{i+1} - 2f_i + f_{i-1}}{h^2}$$

Similarly, adding the two expansions f_{i+2}, f_{i-2} , gives

$$f_{i+2} + f_{i-2} = 2f_i + \frac{\partial^2 f}{\partial \eta^2} 4h^2 + \frac{1}{16} \frac{\partial^4 f}{\partial \eta^4} 16h^4 + \dots \quad (2.27)$$

Taking equation (2.26) multiple by 16, we get

$$16f_{i+1} + 16f_{i-1} = 32f_i + \frac{\partial^2 f}{\partial \eta^2} 16h^2 + \frac{1}{16} \frac{\partial^4 f}{\partial \eta^4} 16h^4 + \dots \quad (2.28)$$

Subtracting equations (2.28) and (2.27), we obtained the fourth-order centred differential approximation (CD4) for the second derivative

$$\frac{\partial^2 f}{\partial \eta^2} = \frac{-f_{i+2} + 16f_{i+1} - 30f_i + 16f_{i-1} - f_{i-2}}{12h^2}$$

By adding the two expansions f_{i+3} and f_{i-3} , and time with 2 gives

$$2f_{i+3} + 2f_{i-3} = 4f_i + \frac{\partial^2 f}{\partial \eta^2} 18h^2 + \frac{1}{24} \frac{\partial^4 f}{\partial \eta^4} 162h^4 + \dots \quad (2.29)$$

In the equations (2.27) and (2.26) donated multiple by 27 and 270, we obtained

$$27f_{i+2} + 27f_{i-2} = 54f_i + \frac{\partial^2 f}{\partial \eta^2} 108h^2 + \frac{1}{16} \frac{\partial^4 f}{\partial \eta^4} 432h^4 + \dots \quad (2.30)$$

$$270f_{i+1} + 270f_{i-1} = 540f_i + \frac{\partial^2 f}{\partial \eta^2} 270h^2 + \frac{1}{16} \frac{\partial^4 f}{\partial \eta^4} 270h^4 + \dots \quad (2.31)$$

Subtracting equation (2.31) and (2.30), so

And then adding equation (2.29), we obtained the sixth order of centred differential approximation (CD6) for the second derivative

$$\frac{\partial^2 f}{\partial \eta^2} = \frac{2f_{i+3} - 27f_{i+2} + 270f_{i+1} - 490f_i + 270f_{i-1} - 27f_{i-2} + 2f_{i-3}}{180h^2}$$

3 Parallel computing

3.1 Overview of PETSc

This thesis uses PETSc for the parallel implementation of simulation. PETSc (Portable, Extensible Tool kit for Scientific Computation) is a software library for parallel computing developed at Argonne National Laboratory. Implementing efficient algorithms for modern scientific applications requires the use of a swath of concepts in computer science and numerical analysis, and it would be impractical and ineffective for one person to implement everything. PETSc provides data structures and routines for creating efficient, scalable solution methods for problems that can be modelled by partial differential equations (Balay, et al., 2015). Since very many problems in science can be modelled by PDEs, PETSc has been used frequently in the academic community (Balay, et al., 2015). Some of the basic data structures in PETSc include matrices, vectors, and index sets. Higher-level objects like unstructured mesh utilities, iterative linear solvers, and preconditioners take the basic data structures as inputs. Parallelism of data structures and numerical routines is accomplished with the Message Passing Interface (MPI), which PETSc uses behind the scenes. MPI is a message passing library for parallel computing based on the consensus and input of industry leaders in high-performance computing. Programming a scientific application from scratch using MPI is a difficult endeavour, and PETSc aims to hide most of that effort from the scientist. For an accomplished PETSc programmer, few if any calls to MPI are required to design a parallel program.

3.2 Parallel Implementation

Principally, PETSc is used to solve the linear systems resulting from the Navier-Stokes equations. The linear system coefficients into PETSc matrices and vectors, which can then be used with PETSc Krylov Subspace (KSP) routines. There are several challenges to this approach that are closely related:

- (1) Storing the data associated with each nodal point in the grid
- (2) Communicating these data between processors when necessary
- (3) Relating the 3D mesh of nodal points with the resulting matrix of coefficients in parallel and extensible. PETSc provides a set of data management routines for distributed array (*DMDA*) that addresses all three challenges.

PETSc *DMDA* object must be provided with the number of nodal points in each direction, the stencil type and width, and the boundary condition in each direction. Given this information, PETSc determines an efficient way of partitioning the mesh among processors. The *DMDA* object is initialized using *DMDACreate1()* for one-dimensional, *DMDACreate2()* for two-dimensional, and *DMDACreate3()* for three-dimensional. *DMDA* creation as

```
DMDACreate2d(MPI_Comm comm, DMDABoundaryType Xperiod,  
              DMDABoundaryType Yperiod, DMDAStencilType st,
```


int Nx, int Ny, int n, int m, int dof, int s, int lx, int ly, DM da)

with *comm* is PETSc communicator (PETSC_COMM_WORLD and PETSC_COMM_SELF). *Xperiod* and *Yperiod* are the types of ghost nodes such as DM_BOUNDARY_PERIODIC, DM_BOUNDARY_NODE, and DM_BOUNDARY_GHOSTED.

Stencil type is *st* as DMDA_STENCIL_STAR or DMDA_STENCIL_BOX, and where *Nx*, *Ny*, *n*, *m*, *dof* and *s* donated global dimension in each direction, number of processors in each direction, number of degrees of freedom per node, and stencil width.

Example:

```
DMDACreate2d(PETSC_COMM_WORLD, DM_BOUNDARY_PERIODIC,
             DM_BOUNDARY_PERIODIC, DMDA_STENCIL_STAR, nx, ny,
             PETSC_DECIDE, PETSC_DECIDE, 1, stencil_CD, NULL, NULL, &da);
```

```
DMSetFromOptions(da);
```

```
DMSetUp(da);
```

```
DMCreateMatrix(da, &A);
```

```
MatSetFromOptions(A);
```

```
DMCreateGlobalVector(da, &b);
```

```
VecDuplicate(b, &x);
```

The details of the resulting decomposition will depend on the number of nodes and the number of processors used. For a four processors case, *DMDA* might decompose the domain as follows

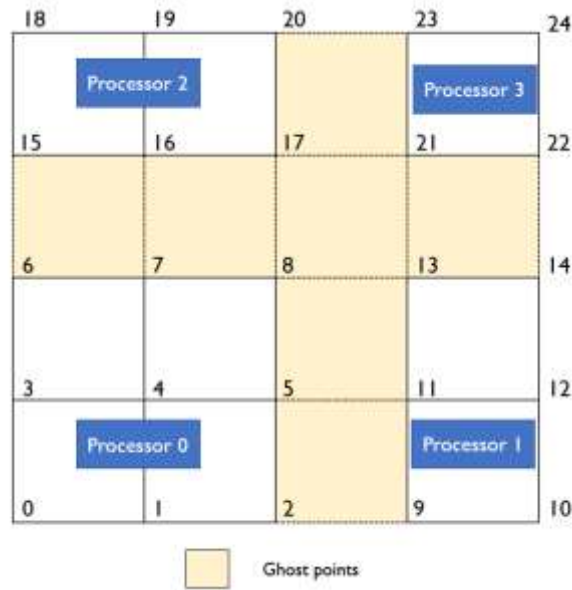


Fig. 5. Local position on each process for four processors

One of the more useful features of *DMDA* is the ability to associate matrices and vectors with domain decomposition. A matrix *A*, solution vector *x* and right-hand side vector *b* can be associated with the *da* object created in Example 1.

Normally, the user must manually decompose and parallelize the computational domain and then parallelize matrices and vectors in a manner that is conformal with the domain parallelization. In example 1, this process is handled automatically. Furthermore, when constructing the matrix, it is usually necessary to carefully map the domain indices to matrix rows and columns. PETSc offers the *MatStencil* data structure as a way to automatically compute these index transformations. For example, at node (i, j) in the interior of the domain, a matrix row corresponding to the discrete 2D Laplacian could be entered as follows

```

PetscInt mx, my, xm, ym, xs, ys;
PetscScalar val[5], lhs[5];
MatStencil row, col[5];
for (j = ys; j < (ys + ym); j++){
    for (i = xs; i < (xs + xm); i++){
        row.i = i; row.j = j;
        val[0] = lhs[0]; col[0].i = i; col[0].j = j - 1;
        val[1] = lhs[1]; col[1].i = i - 1; col[1].j = j;
        val[2] = lhs[2]; col[2].i = i; col[2].j = j;
        val[3] = lhs[3]; col[3].i = i + 1; col[3].j = j;
        val[4] = lhs[4]; col[4].i = i; col[4].j = j + 1;
        MatSetValuesStencil(U, 1, &row, 5, col, val, INSERT_VALUES);
    }
}
MatAssemblyBegin(U, MAT_FINAL_ASSEMBLY);
MatAssemblyEnd(U, MAT_FINAL_ASSEMBLY);

```

Because matrix *A* was associated with the *DMDA* object *da* at initialization, PETSc can map the node indices (i, j) to the corresponding matrix rows and columns. In addition, boundary conditions are more easily implemented using *DMDA*. For periodic boundary conditions, *DMDA* maps indices that are too large or small to the other edge of the domain.

For high-fidelity simulations, the numerical solution of these systems is intractable without the use of parallel computing. PETSc and MPI are used to parallelize the data structures and iteratively

solve the linear systems. However, special care is required for the all-Neumann boundary condition case for the Poisson equation, which results in a singular system. This occurs, for example, when velocities are prescribed at all boundaries of the domain. To remedy this, PETSc allows the removal of the null space of constant functions, which makes the problem nonsingular.

```
MatNullSpaceCreate(PETSC_COMM_WORLD, PETSC_TRUE, 0, 0, &nullspace);
```

```
MatNullSpaceRemove(nullspace, b);
```

```
MatNullSpaceDestroy(&nullspace);
```

3.3 A solution method for discretized linear system

The Semi-Implicit method was used to solve the Navier-Stokes equations. To solve the linear systems used PETSc library (Balay, et al., 2015), was known as a library to solve the linear equation and for parallel computing used MPI for communications between the CPU cores. In this study, we used PETSc Krylov solver (KSP) as iterative methods, which Conjugate Gradient (CG), Generalized Minimal Residual (GMRES), Bi-Conjugate Gradient (BICG), etc. As (Shang-Gui, Abdellatif, Julien, & Yannick, 2017) shows, CG has the best convergence speed then we decided to use the CG method to solve the linear system for both Helmholtz and Poisson equations (shown in Algorithm 1) in this study.

Algorithm 2 The Conjugate Gradient method

Initial $r = b - Ax, p = r, E1 = r^T * r$

For i = 1: length(b)

$A_p = A * p$

$\alpha = \frac{E1}{p^T * A_p}, x = x + \alpha p; r = r - \alpha A_p$

$E2 = \text{sqrt}(r^T * r)$

If $E2 \leq \text{tolerance}$

break

end

$p = r + \frac{E2 * E2}{E1} p, E1 = E2$

end

The tolerance was set to 1e-15 for relative and 1e-50 for absolute. Additionally, the preconditioners can be employed to get more efficiency while solving the linear system, such as Jacobi, Incomplete LU factorization (ILU), Multigrid (MG), Algebraic Multigrid (AMG), etc.

4 Simulation results

To achieve the high accuracy of the numerical method described above, the Taylor Green Vortex problem (Taylor & Green, 1937) was simulated in this study. To ensure that the methods described in the numerical methods section are accurate, first, simulate the Taylor Green Vortex problem in two dimensional and then compare the simulation results with the analytical results. Then, choose the high-precision methods listed in the previous section to simulate the Taylor Green Vortex problem in three-dimensional.

Boundary conditions applied for this case is periodic boundary condition and body force $f = 0$.

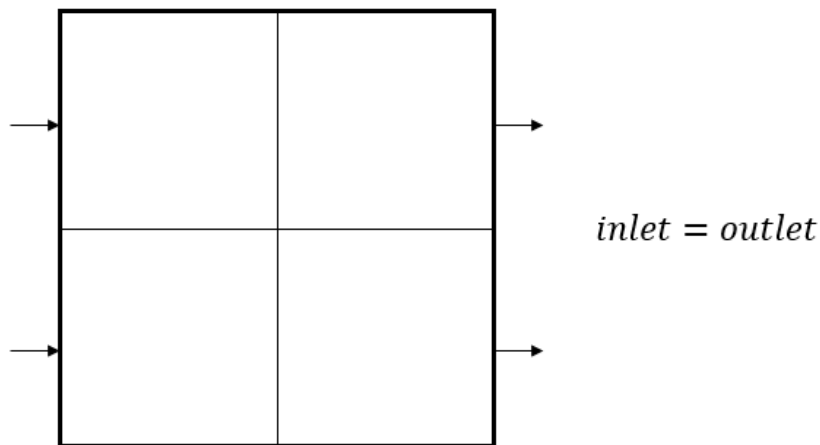


Fig. 6. Boundary condition

4.1 Two dimensional

The analytical solution of the two-dimensional Taylor-Green Vortex, unsteady incompressible Navier–Stokes equations is given as

$$\begin{aligned}u &= \sin(x) \cos(y) e^{-2t/Re} \\v &= -\cos(x) \sin(y) e^{-\frac{2t}{Re}} \\p &= \frac{1}{4} (\cos(2x) + \cos(2y)) e^{-4t/Re}\end{aligned}$$

We use u , v and p at $t = 0$ as an initial condition for our simulation. The domain is a square of length $L = 2\pi$ as $[0, L] \times [0, L]$, the time dimensionless interval is $0 \leq t \leq T = \pi$, and the $Re = 10000$. The periodic boundary condition is applied.

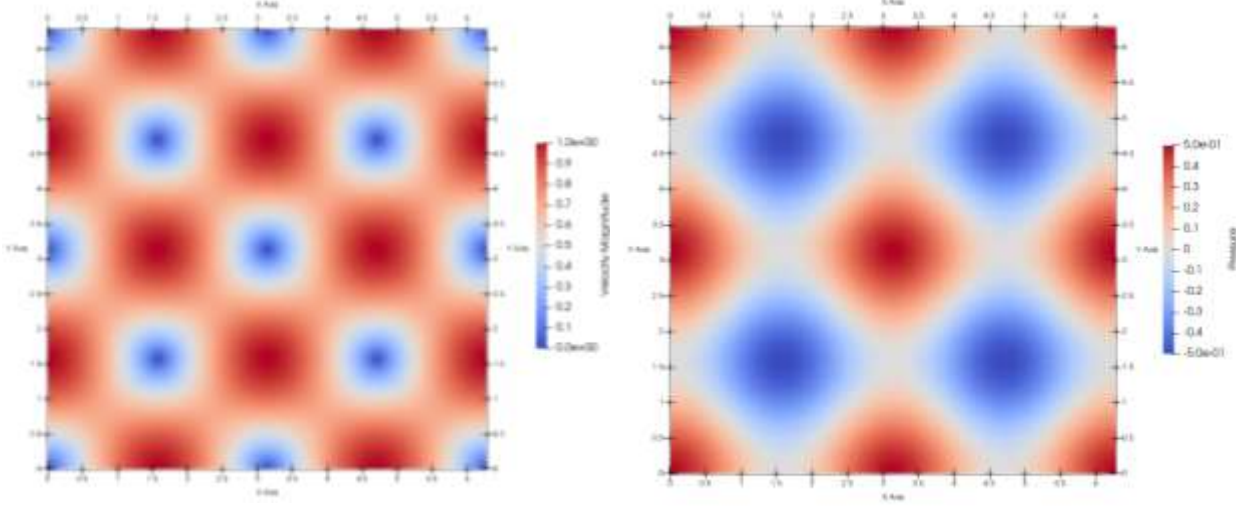


Fig. 7. Velocity magnitude and Pressure at $t=0$

To measure the order of accuracy, the L_2 and $L_{infinity}$ norm error is computed to assess the order of accuracy.

$$L_2 = \sqrt{\left| \sum (u_{analytical} - u)^2 \right| / N} \quad (4.1)$$

$$L_{infinity} = \max |u_{analytical} - u| \quad (4.2)$$

where $N = Nx \times Ny$ represents the total number of nodes (with Nx number of nodes along x direction and Ny number of nodes along y direction). Time step is a varied following number of nodes along the x -direction, $\Delta t = T/Nx$ with $Nx = 8, 12, 16, 24, 32, 48, 64, 96, 128, 192, 256$, $Nx = Ny$.

4.1.1 Mesh convergence study with fixed CFL

The order of accuracy is measured in all combinations of the discretization schemes, namely 3rd, 5th, and 7th order WENO schemes and 2nd, 4th, and 6th order CD schemes, which results in a total of 9 different discretization schemes. For each of the schemes, the order of accuracy is measured by a mesh refinement study. A series of successively refined meshes of $Nx = 8, 12, 16, \dots, 256$, are prepared and the TGV2d problem was solved until $T_{end} = \pi$. Due to the semi-implicit nature of the time-advancement scheme, the time step must be refined also with the condition of fixed CFL which is $1/4$.

The errors are measured for the velocity, u -component only as v is almost identical, and pressure in the sense of both the L_2 and maximum norms. By increasing the order of the WENO scheme, the results are displayed in Figs. 8-16. For WENO3, all combination of CD schemes shows expected 3rd order accurate velocity both in the L_2 and infinity norms. Hence the WENO3/CD2

would be sufficient for the expected accuracy. For the pressure, the order of accuracy stays in the second-order, which is expected to be so due to the second-order accurate Poisson solver.

For WENO5, as shown in Fig. 11-13, the trend differs from the previous one. For WENO5/CD2 case, the order of accuracy starts with fifth-order but as mesh refines the slope plateaus and loses the expected order of accuracy. This is due to the low order accuracy of the central difference part. As the scheme changes to WENO5/CD4, it shows the expected fifth-order accuracy until the last stage of mesh refinement. The WENO5/CD6 shows the same trends, which tells that WENO5/CD4 would suffice for the accuracy.

Finally, the most accurate WENO7 schemes are presented in Fig. 14-16. WENO7/CD2 combination shows the interesting transition. At the early stage of refinement, it starts with the full seventh order of accuracy, but as mesh refines the degrade to the second which is the order of accuracy from the central counterpart. A similar trend is also observable for the last stage of the WENO7/CD4 combination. The best combination for the most accuracy would be concluded as the WENO7/CD6 scheme and this is supported by the last plot in Fig. 16.

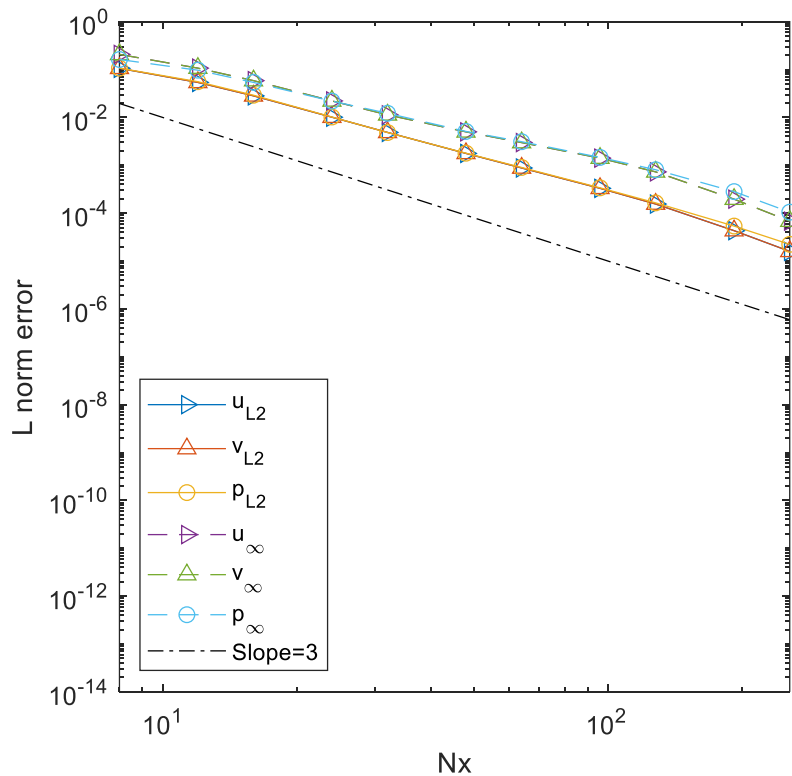


Fig. 8. The L_2 and $L_{infinity}$ norm error of WENO3 with CD2

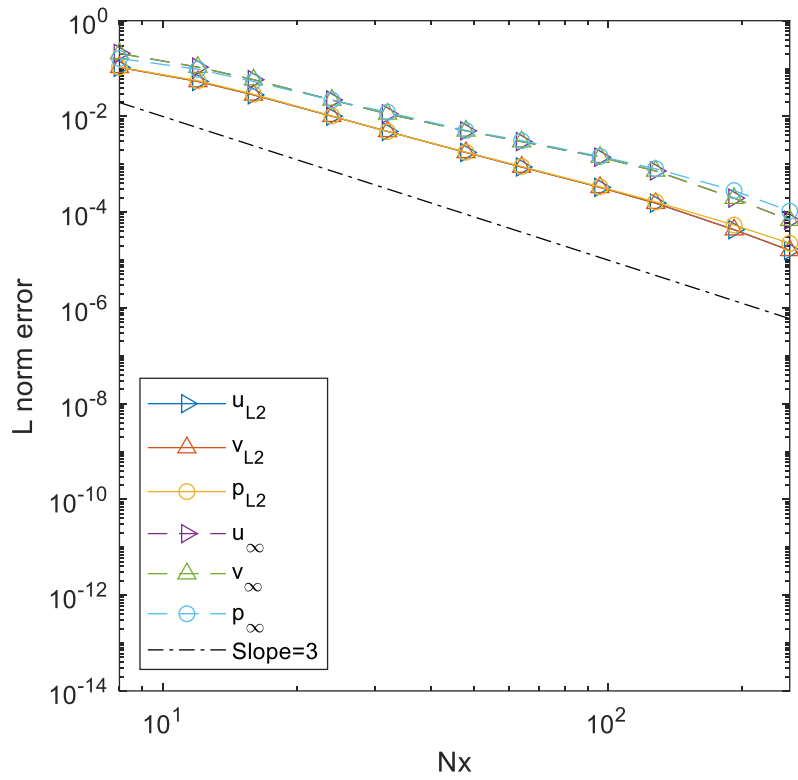


Fig. 9. The L_2 and $L_{infinity}$ norm error of WENO3 with CD4

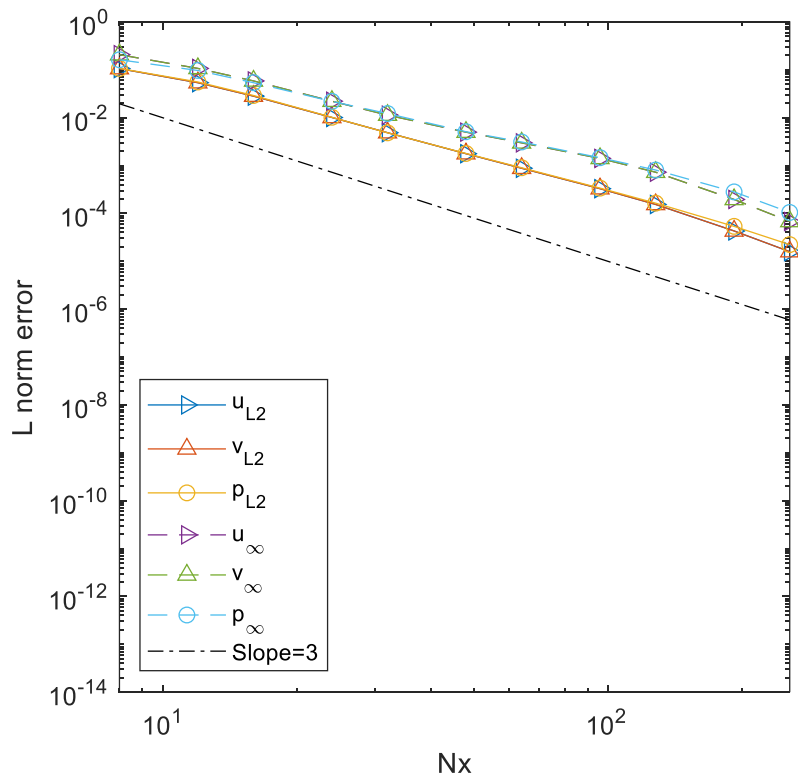


Fig. 10. The L_2 and $L_{infinity}$ norm error of WENO3 with CD6

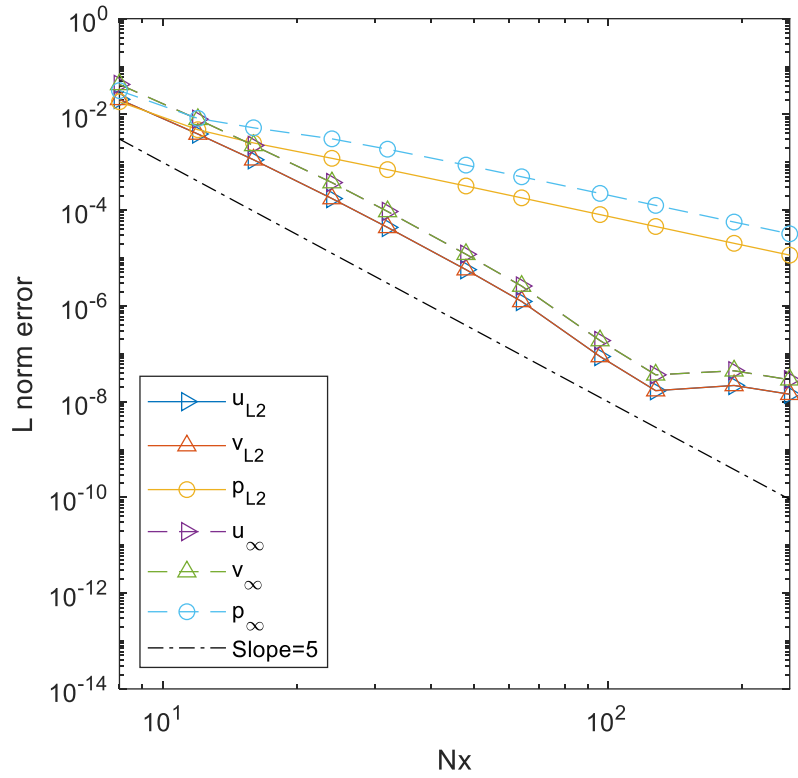


Fig. 11. The L_2 and L_{∞} norm error of WENO5 with CD2

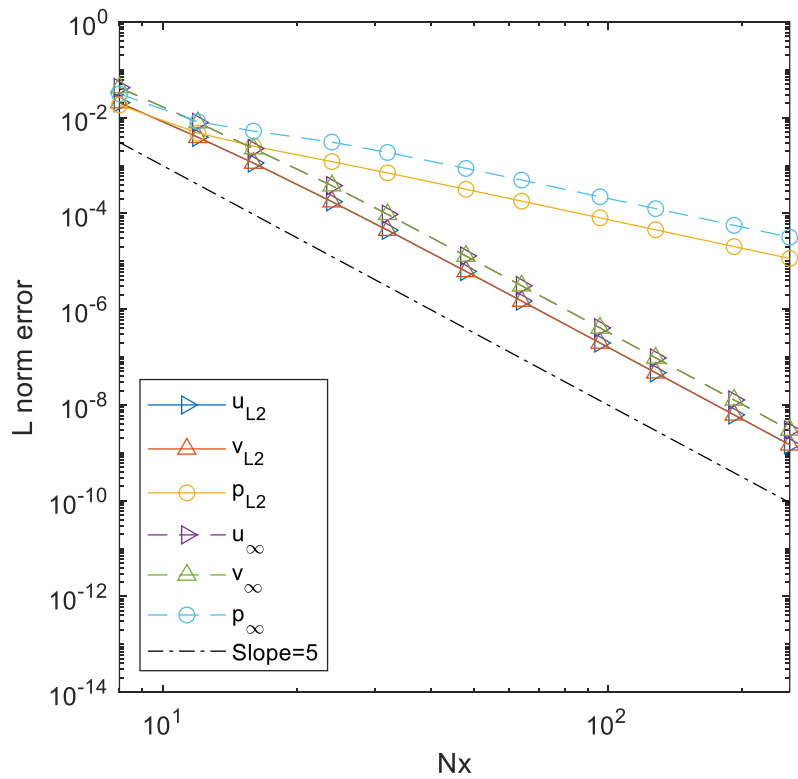


Fig. 12. The L_2 and L_{∞} norm error of WENO5 with CD4

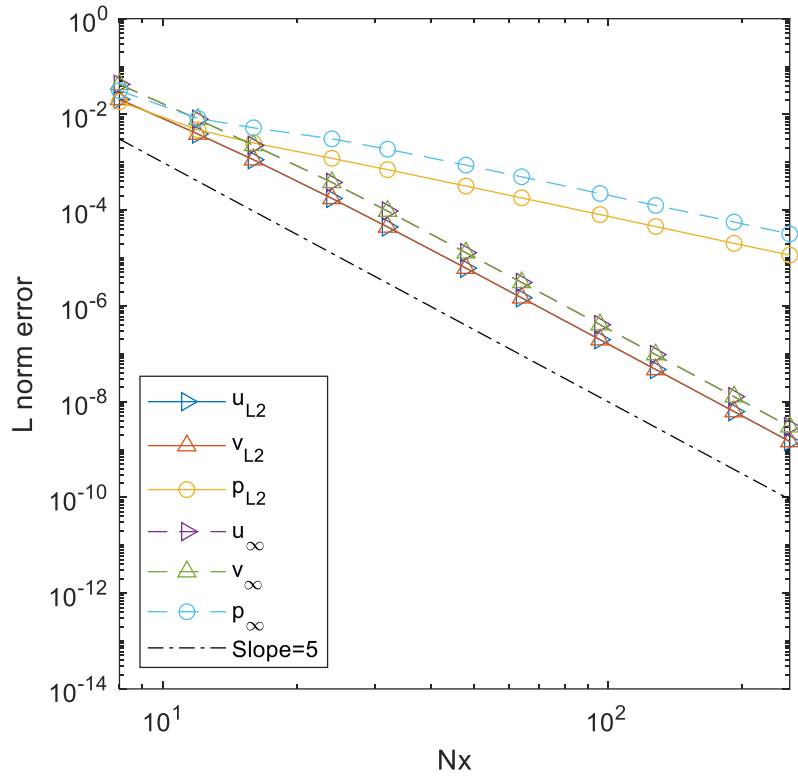


Fig. 13. The L_2 and L_{∞} norm error of WENO5 with CD6

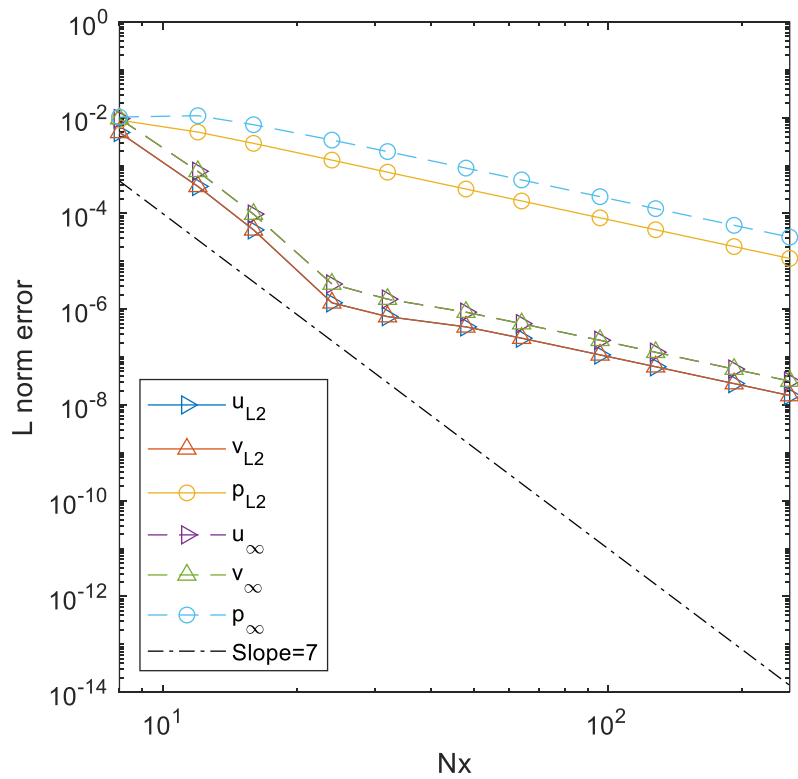


Fig. 14. The L_2 and L_{∞} norm error of WENO7 with CD2

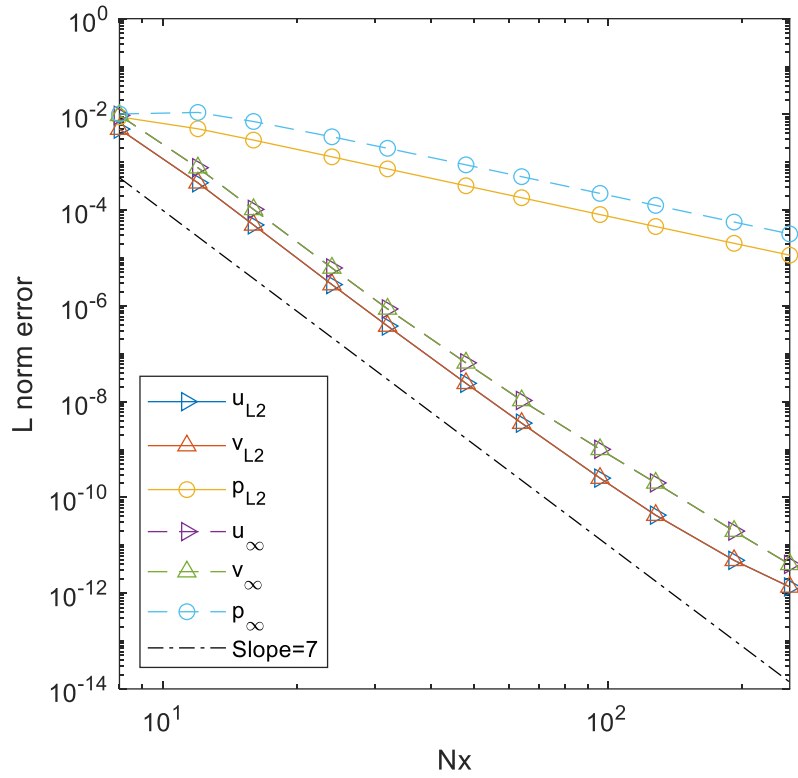


Fig. 15. The L_2 and L_∞ norm error of WENO7 with CD4

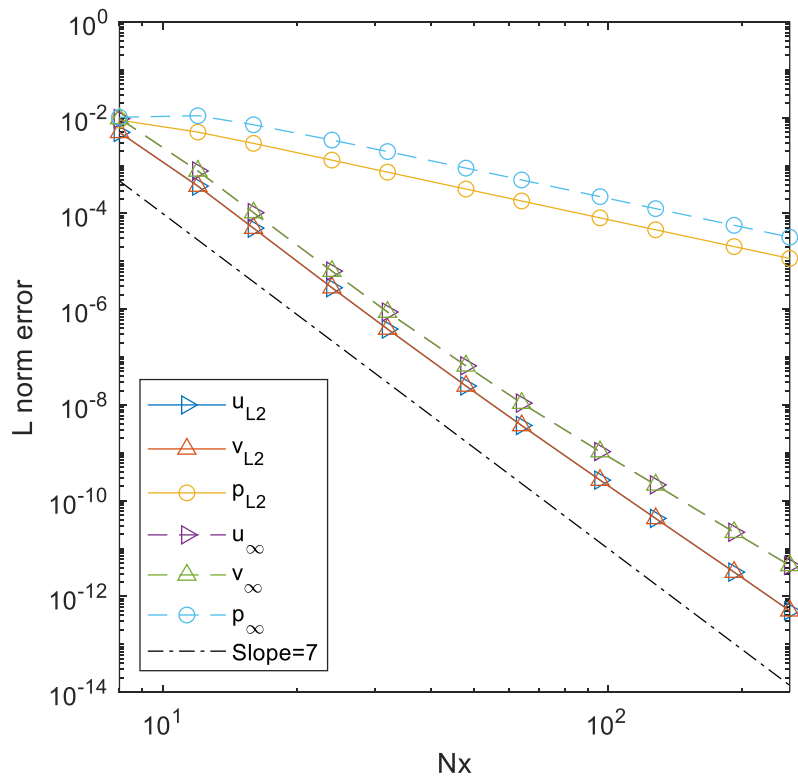


Fig. 16. The L_2 and L_∞ norm error of WENO7 with CD6

To verify the aforementioned results, the raw data of convergence rate is shown in Table 1-3, the designed order of accuracy is demonstrated for the third, fifth, and seventh sequence for the WENO3, WENO5, and WENO7 schemes when it was coupled with proper CD schemes, i.e., WENO(k)/CD($k - 1$). It should be noted that the order of accuracy applies for the velocities, not for the pressure which is second-order due to the Poisson discretization.

Table 1. The L_2 and $L_{infinity}$ of the u , p component, and the convergence rate of CD2/WENO3

Nx	u_{L2}	$Rate_{L2}$	u_{∞}	$Rate_{\infty}$	p_{L2}	$Rate_{L2}$	p_{∞}	$Rate_{\infty}$
8	1.06e-01	-	2.07e-01	-	1.07e-01	-	3.14e-02	-
16	2.82e-02	1.91	5.86e-02	1.82	2.91e-02	1.88	5.25e-03	1.59
32	4.86e-03	2.54	1.13e-02	2.37	4.83e-03	2.59	1.88e-03	2.14
64	8.78e-04	2.47	2.98e-03	1.92	9.01e-04	2.42	4.98e-04	1.97
128	1.55e-04	2.50	7.28e-04	2.03	1.64e-04	2.46	5.65e-05	1.95
256	1.59e-05	3.29	6.81e-05	3.42	2.23e-05	2.88	3.19e-05	2.94

Table 2. The L_2 and $L_{infinity}$ of the u , p component, and the convergence rate of CD4/WENO5

Nx	u_{L2}	$Rate_{L2}$	u_{∞}	$Rate_{\infty}$	p_{L2}	$Rate_{L2}$	p_{∞}	$Rate_{\infty}$
8	2.06e-02	-	4.24e-02	-	1.84e-02	-	3.14e-02	-
16	1.13e-03	4.19	2.26e-03	4.23	2.54e-03	2.86	5.25e-03	2.58
32	4.46e-05	4.66	9.58e-05	4.56	7.03e-04	1.85	1.88e-03	1.48
64	1.48e-06	4.91	3.09e-06	4.95	1.81e-04	1.96	4.98e-04	1.92
128	4.70e-08	4.98	9.61e-08	5.01	4.55e-05	1.99	5.65e-05	1.98
256	1.47e-09	5.00	3.00e-09	5.00	1.15e-05	1.98	3.19e-05	1.98

Table 3. The L_2 and $L_{infinity}$ of the u , p component, and the convergence rate of CD6/WENO7

Nx	u_{L2}	$Rate_{L2}$	u_{∞}	$Rate_{\infty}$	p_{L2}	$Rate_{L2}$	p_{∞}	$Rate_{\infty}$
8	4.93e-03	-	9.59e-03	-	8.90e-03	-	1.02e-02	-
16	4.92e-05	6.65	1.04e-04	6.53	2.91e-03	1.61	7.12e-03	0.51
32	3.85e-07	7.00	8.71e-07	6.90	7.28e-04	2.00	1.96e-03	1.86
64	3.72e-09	6.69	1.09e-08	6.32	1.82e-04	2.00	5.01e-04	1.97
128	4.26e-11	6.45	2.14e-10	5.67	4.55e-05	2.00	1.26e-04	1.99
256	5.02e-13	6.41	4.44e-12	5.59	1.15e-05	1.98	3.19e-05	1.98

To compare the simulation time of all schemes shown above, we calculated the computational time for the resolution of mesh is 256×256 , which is the highest mesh in this study. As mentioned in the section on numerical method, the conjugate gradient (CG) of Krylov solver with multi-grid (MG) as a pre-conditioner is used in this case. The MG method is found to be more efficient when used as a preconditioner in conjunction with Krylov solvers instead of a pure solver (Balay, et al., 2015). The code was run with 4 processors on the desktop, which has device information is "Intel® Core™ i7-6700 CPU @ 3.40Ghz 3.41GHz and installed RAM 32.0GB". The time elapsed wall clock time is given in the table below,

Table 4. Accuracy preserving capability of the various combination between WENO and CD schemes

Order of accuracy	CD2	CD4	CD6
WENO3	O	O	O
WENO5	×	O	O
WENO7	×	×	O

Table 4 summarizes the order of accuracy preservation capability throughout the mesh refinement study, where the O stands for the scheme preserving the design accuracy and × for the scheme which failed to preserve. The combination corresponds to the diagonal components in the table, namely $WENO(k) / CD(k - 1)$, would be recommended in terms of accuracy and efficiency. As shown in Figs. 8-16, any combination above the diagonal will not lose accuracy but also will not gain either. This is because the dominant error originated from the nonlinear convection term discretization where WENO directly concerns, at least for the present test cases. This implies that $WENO(k) / CD(k - 1)$ the combination would be the recommended choice and the upper diagonal combination would be just an over-kill.

4.1.2 Spatial (by fixing Δt) and temporal accuracy (by fixing Δx) in an independent manner

In general, the error of a transient flow simulation comes from both the spatial (Δx) and temporal (Δt) discretization. To measure the spatial and temporal order of accuracies independently, the mesh refinement and timestep refinement study are presented by fixing the Δt and Δx respectively.

For the successful demonstration of the spatial or temporal accuracies independently, the error of the other counterpart must remain small enough so that, the Δx or Δt refinement exclusively affects the spatial and temporal accuracies respectively.

To guarantee this situation (for the spatial accuracy measurement the temporal error must be small enough, and vice versa), the following test case set-up was utilized. The TGV2d problem is designed for a longer period of simulation in time simply by lowering the Reynolds number which controls the decay rate of the vortex from $Re = 10,000$ to $Re = 100$ with the same end time of $T_{end} = \pi$. This change results in essentially an identical case of $Re = 10,000$ simulation until $T_{end} = 100\pi$, which can be easily verified by the analytical solution.

For measuring the spatial accuracy exclusively, the time step is a fixed value $\Delta t = 2.5 \times 10^{-4}$ and this guarantees the $CFL \leq 1$ even on the coarsest mesh. As shown in the left columns of Figs. 11-12, the slope of the plots shows that current algorithms have spatial order of accuracy of 3, 5 and close to 7 both in $L2$ and infinity norms. These are consistent with those of the previous section.

For the measurement of pure temporal accuracy, the situation is a bit trickier. The other counterpart, i.e., the spatial error must be far below the temporal error to measure. For this purpose, the finest mesh of $Nx = 256$ is prepared and the time step is successively refined from $Nt = 144$ to $Nt = 1024$.

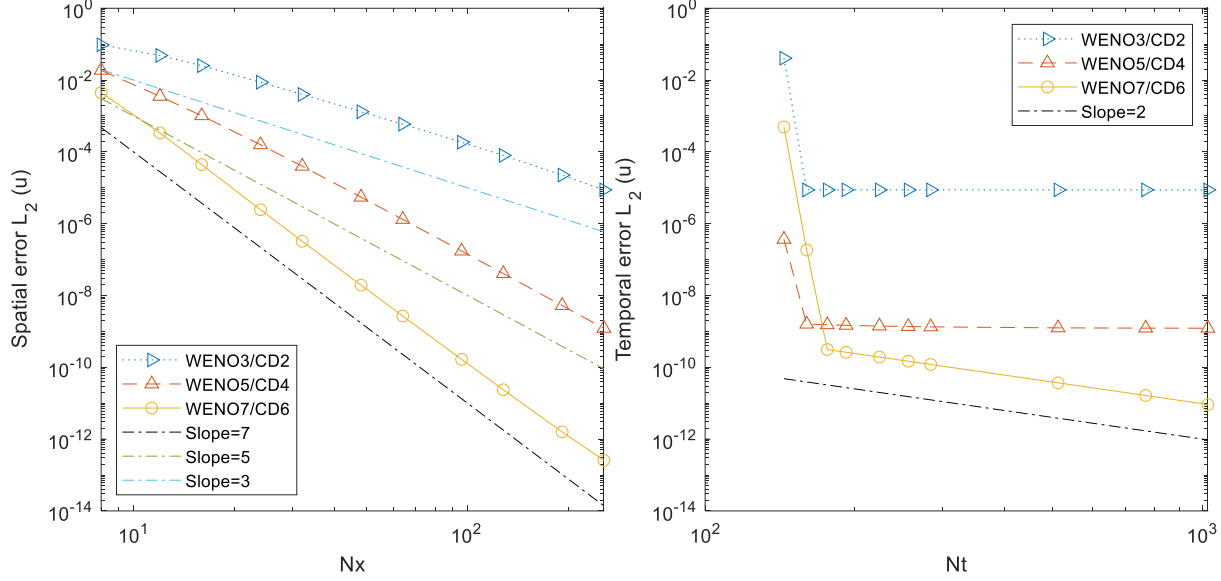


Fig. 17. The L_2 norm error of u component

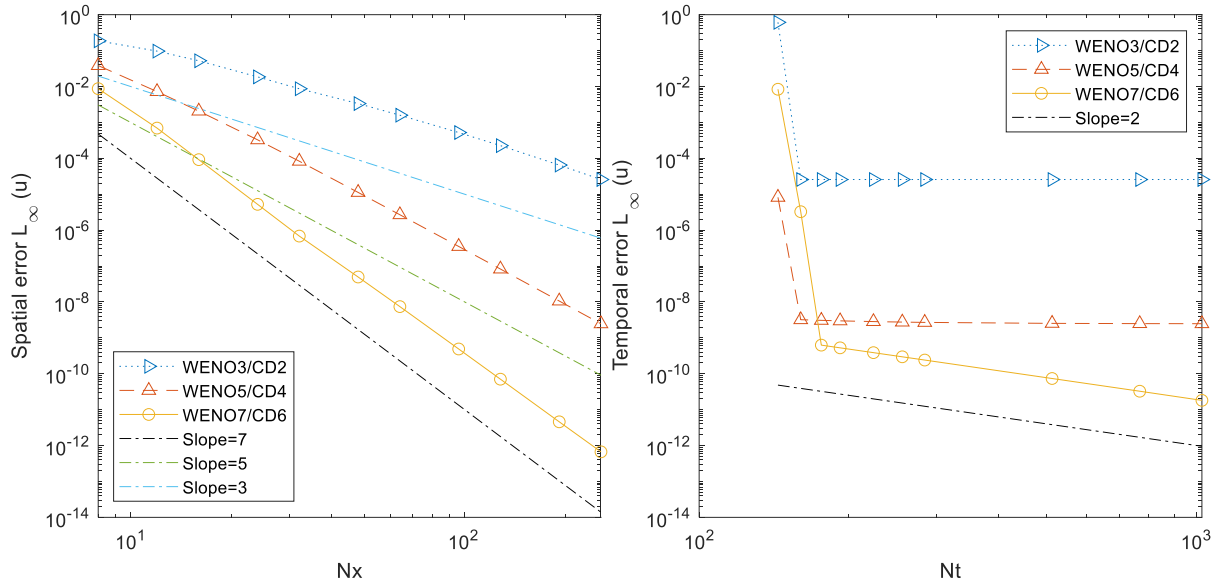


Fig. 18. The L_∞ norm error of u component

The right column of Figs. 17-18 is a bit interesting. To measure the temporal order accuracy exclusively, the error due to the spatial discretization must stay below the temporal one. All three recommended combinations of the schemes, i.e., WENO3/CD2, WENO5/CD4, and WENO7/CD6 are tested on the finest mesh of 256^2 grid. The timestep refinement is conducted reversely by starting with the finest time step of $\Delta t = \pi/1,024$ whose $CFL = 0.125$ back to the largest timestep of $\Delta t = \pi/144$ whose $CFL = 0.89$.

As shown in Figs. 17-18 and confirmed by the raw data presented in Table 5, the expected temporal order of accuracy is observed in the most accurate spatial discretization scheme, which is WENO7/CD6. The other schemes, i.e., WENO3/CD2 and WENO5/CD4, show essentially no

change in the measurement error. This can be interpreted as the temporal error turning out to be hidden below the spatial error that is the dominant error component which indeed must not change because Δx is fixed.

This tells that the second-order accuracy of the temporal discretization (the ABCN method) is adequate for the present WENO(k)/CD($k - 1$) spatial discretization schemes provided that the stability satisfying timesteps, i.e., $CFL < 1$. In other words, the expected spatial order of accuracy is supposed to be obtained if the time step satisfies the stability condition of $CFL < 1$.

Table 5. Timestep refinement result of WENO7/CD6. Error measured at $T_{end} = \pi$ for $Re = 100$

N_t	u_{L2}	$Rate_{L2}$	u_∞	$Rate_\infty$
176	3.13e-10	—	6.26e-10	—
192	2.63e-10	2.00	5.26e-10	2.00
224	1.93e-10	2.01	3.87e-10	1.99
256	1.48e-10	1.98	2.96e-10	2.01
284	1.20e-10	2.02	2.40e-10	2.02
512	3.69e-11	2.00	7.37e-11	2.00
768	1.64e-11	2.00	3.26e-11	2.01
1,024	9.21e-12	2.01	1.81e-11	2.05

4.1.3 Execution time and memory requirement

The computational efficiency is presented in two different aspects, namely CPU time and memory requirement for simulation. The total computational time of the simulation for the 256^2 grid with total time steps of $N_t = 512$ (corresponding to $CFL = 0.25$) was measured by the wall-clock time. The code was compiled with GCC ver. 5.4.0 and run on a standard commodity desktop, whose CPU is 3.40Ghz Intel® Core™ i7-6700 running on the total RAM of 32GB. The total execution time and memory footprint are presented in Table 6.

Table 6. Executed time and the total memory spent for $N_t = 512$ time stepping on 256^2 grid.

Interpolation	CD2		CD4		CD6	
	Time	MEM	Time	MEM	Time	MEM
WENO3	8m33.044s	0.07G	8m53.456s	0.10G	9m18.456s	0.12G
WENO5	8m4.370s	0.10G	8m17.224s	0.11G	8m59.520s	0.13G
WENO7	8m6.116s	0.11G	8m30.149s	0.12G	9m8.111s	0.19G

One very interesting point is that regardless of the spatial order of accuracy of the chosen scheme, the expected execution time is virtually the same, which means the highest order scheme (WENO7/CD6) among the various combinations delivers the most accurate result, without demanding an appreciable computation overhead. In terms of memory slight increase was observed mostly depending on the choice of the central differences, which can be expected due to the stencil width of the velocity Helmholtz solver. The mild increase in the total memory usage for the higher-order WENO scheme could be attributed to the increasing overlapped region in each

decomposed sub-domain utilized for the four-MPI parallel processing on the quad-core CPU. In a conclusion, one can obtain the highest order accuracy without extra cost.

4.2 Three dimensional

As a test case for the turbulent flows, the Taylor green vortex in the 3D problem is simulated for various Reynolds numbers, i.e. $Re = 1.6 \times 10^3$, 1.6×10^4 , and 1.6×10^5 . The computational domain $[0, L_x] \times [0, L_y] \times [0, L_z]$ with $L_x = L_y = L_z = 2\pi$ and the boundary is treated as periodic in all directions. The initial solution for the three-dimensional Taylor-Green vortex problem is given as

$$u = \sin(x) \cos(y) \cos(z)$$

$$v = -\cos(x) \sin(y) \cos(z)$$

$$w = 0, \quad p = 0$$

The time dimensionless simulation interval is $0 < t \leq T = 20$. The simulations are run on three different grids with 64^3 , 128^3 and 256^3 grid nodes which are run with $CFL = 0.25$. For all test cases, the initially isolated vortex starts to break and a transition to a turbulent flow takes place. The results are compared in terms of the evolution of total kinetic energy, enstrophy, local vorticity field, vortex structures visualization, and energy spectra at multiple time moments.

From the velocity fields, we generated the three-dimensional energy spectrum using the code from (Navah, Plata, & Couaillier, 2020). The three-dimensional Fourier transform of such as correlation produces a spectrum that depends only on one wavenumber k (Dietzsch, 2018). The expression of the three-dimensional energy spectrum by integrating over spherical shells

$$\begin{cases} E_u(k) = \iint |\Phi_u|^2 dS(k), \\ E_v(k) = \iint |\Phi_v|^2 dS(k), \\ E_w(k) = \iint |\Phi_w|^2 dS(k), \end{cases} \quad (4.3)$$

where the 3D wavenumber is defined as $k = (k_x^2 + k_y^2 + k_z^2)^{1/2}$ based on the one-dimensional ones of $k_x = \frac{\pi N_x}{L_x}$, $k_y = \frac{\pi N_y}{L_y}$, and $k_z = \frac{\pi N_z}{L_z}$. As the surface of a sphere is completely determined by its radius the surface integral can be certainly evaluated. Thus, the energy spectrum

$$E(|k|) = \frac{1}{2}(E_u + E_v + E_w) \quad (4.4)$$

where k is the wavenumber and Φ_u , Φ_v and Φ_w donated Fourier transform of velocity u , v , w .

Lambda2 (λ_2) (Jeong & Hussain, 1995) vortex criterion, is a vortex core line detection algorithm that can adequately identify vortices from a three-dimensional fluid velocity field.

The Lambda2 (λ_2) method consists of several steps. First, we define the velocity gradient tensor \mathbf{J}

$$\mathbf{J} \equiv \nabla u = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} \\ \frac{\partial w}{\partial x} & \frac{\partial w}{\partial y} & \frac{\partial w}{\partial z} \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & l \end{bmatrix}$$

The velocity gradient tensor is then decomposed into its symmetric and antisymmetric parts with the transpose of tensor \mathbf{J} as

$$\mathbf{J}^T = \begin{bmatrix} a & d & g \\ b & e & h \\ c & f & l \end{bmatrix}$$

$$\mathbf{S} = \frac{\mathbf{J} + \mathbf{J}^T}{2} = \frac{1}{2} \begin{bmatrix} 2a & b+d & c+g \\ b+d & 2e & f+h \\ c+g & f+h & 2l \end{bmatrix} \text{ and } \mathbf{\Omega} = \frac{\mathbf{J} - \mathbf{J}^T}{2} = \frac{1}{2} \begin{bmatrix} 0 & b-d & c-g \\ d-b & 0 & f-h \\ g-c & h-f & 0 \end{bmatrix}$$

the three eigenvalues of $\mathbf{S}^2 + \mathbf{\Omega}^2$ are calculated so that for each point in the velocity field there are three corresponding eigenvalues: λ_1, λ_2 and λ_3 . The eigenvalues are ordered in such a way that $\lambda_1 > \lambda_2 > \lambda_3$. A point in the velocity field is part of a vortex core only if at least two of its eigenvalues are negative i.e., if $\lambda_2 < 0$.

4.2.1 Moderate Reynolds number case, $Re = 1,600$

Several diagnostic quantities can be computed from the flow as it evolves in time, allowing the characteristics of the numerical scheme to be observed. The evolution of the kinetic energy of the flow is obtained by integrating the square of the velocity norm over the domain.

$$E_k = \frac{1}{2\Omega} \int_{\Omega} \|u^2 + v^2 + w^2\| d\Omega \quad (4.5)$$

Similarly, the enstrophy is calculated by integrating the square of the vorticity norm over the domain as

$$\zeta = \frac{1}{2\Omega} \int_{\Omega} \|\omega^2\| d\Omega \quad (4.6)$$

where Ω is the volume of the computational domain.

The dissipation rate $\epsilon(E_k)$ of the evolution of the kinetic energy by differencing time

$$\epsilon(E_k) = -\frac{dE_k}{dt} \quad (4.7)$$

The result from the Pseudo-Spectral method (Rees, Leonard, Pullin, & Koumoutsakos, 2011) is taken as a reference solution which we consider to be sufficiently accurate for this study.

Table 7. The total memory requirement of the TGV3d problem for various combinations of schemes and grids

Scheme/Grid	64^3	128^3	256^3
WENO3/CD2	0.28G	1.57G	14.01G
WENO5/CD4	0.31G	1.98G	16.31G
WENO7/CD6	0.35G	2.21G	17.34G

The total memory required for various schemes on a series of successively refined meshes is summarized in Table 7. As mesh refines by half, the total number of cells increases by a multiple of eight. This is directly observed in the memory requirement of the simulation, which tells the linear growth of the memory requirement of the current solution method. This growth of memory requirement does not apply to the solution order elevation. From the lowest order scheme of WENO3/CD2 to the most accurate scheme of WENO7/CD6, the memory usage increment is very mild. This is attributed to the wider stencil of the higher-order schemes, which in turn requires a wider overlapping layer and message passing from the distributed memory parallel processing. Overall, like in the 2D simulations, the computational overhead due to the order elevation is minute.

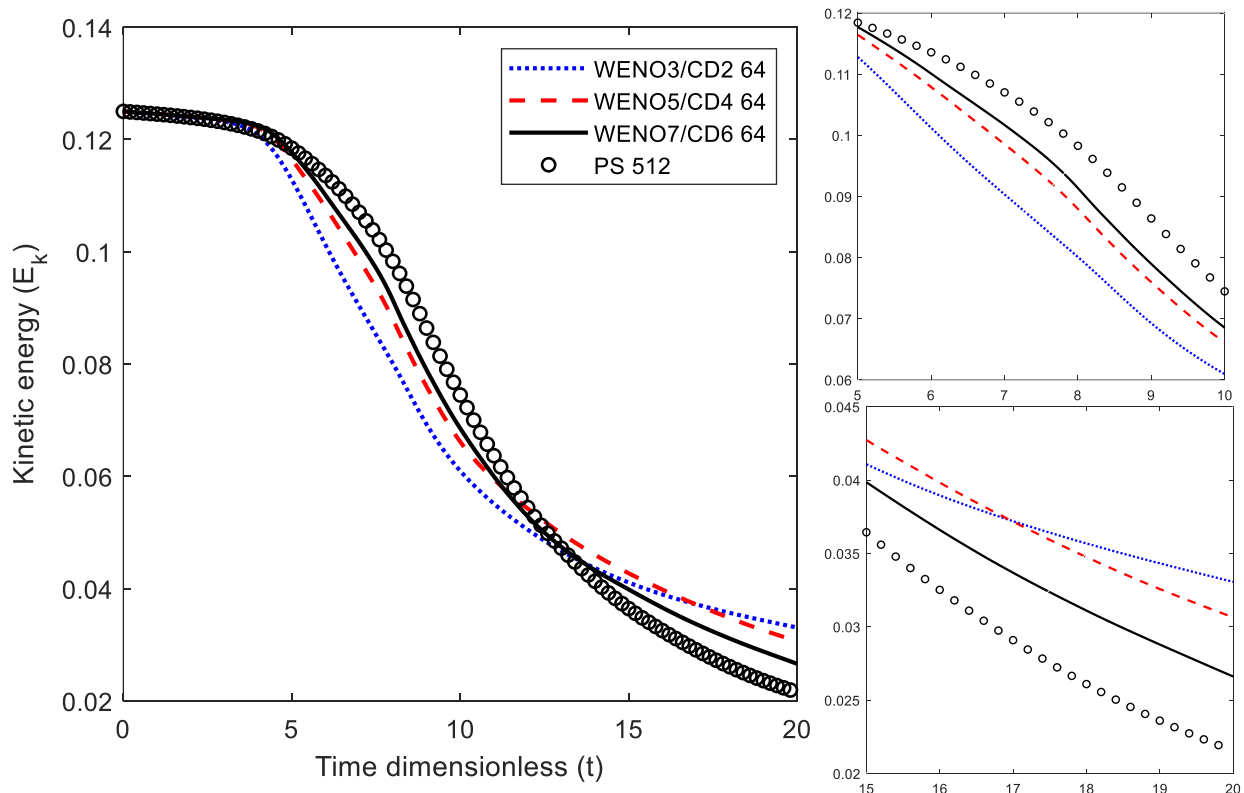


Fig. 19. The kinetic energy of mesh 64^3

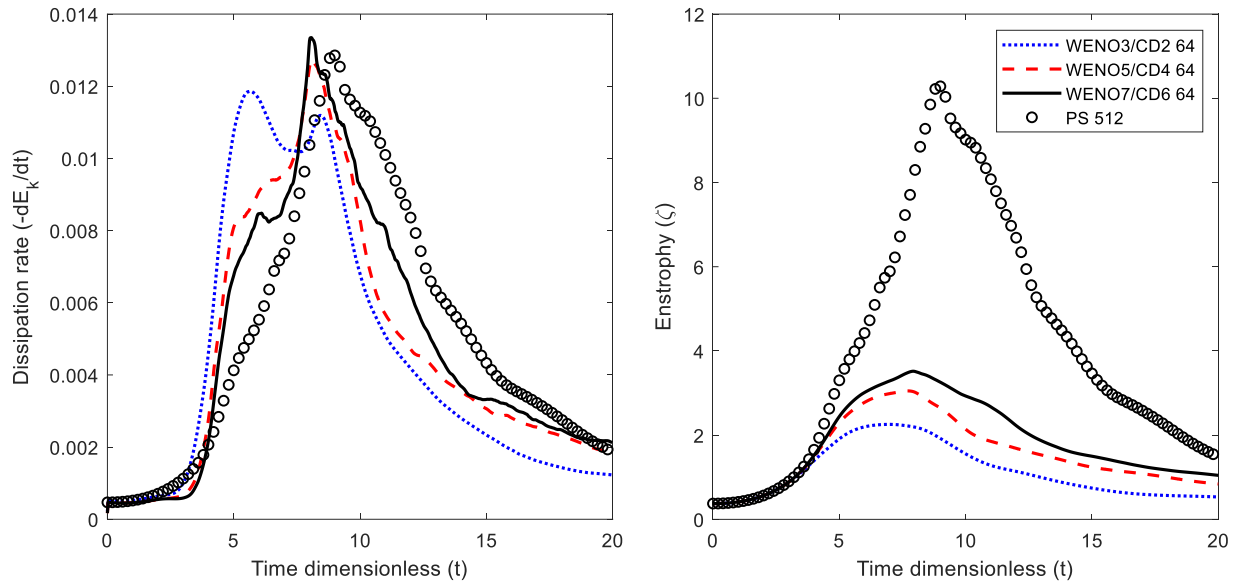


Fig. 20. Kinetic energy dissipation rate (left), the enstrophy (right) of mesh 64^3

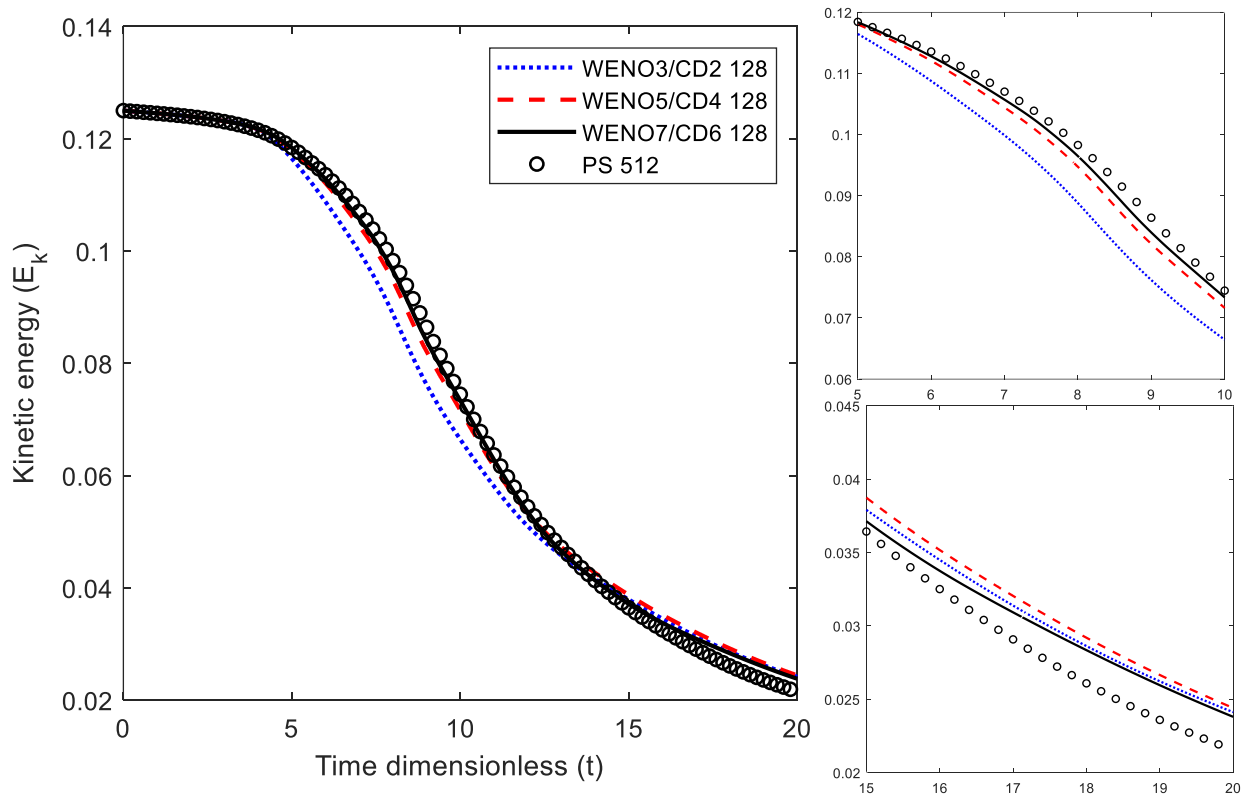


Fig. 21. The kinetic energy of mesh 128^3

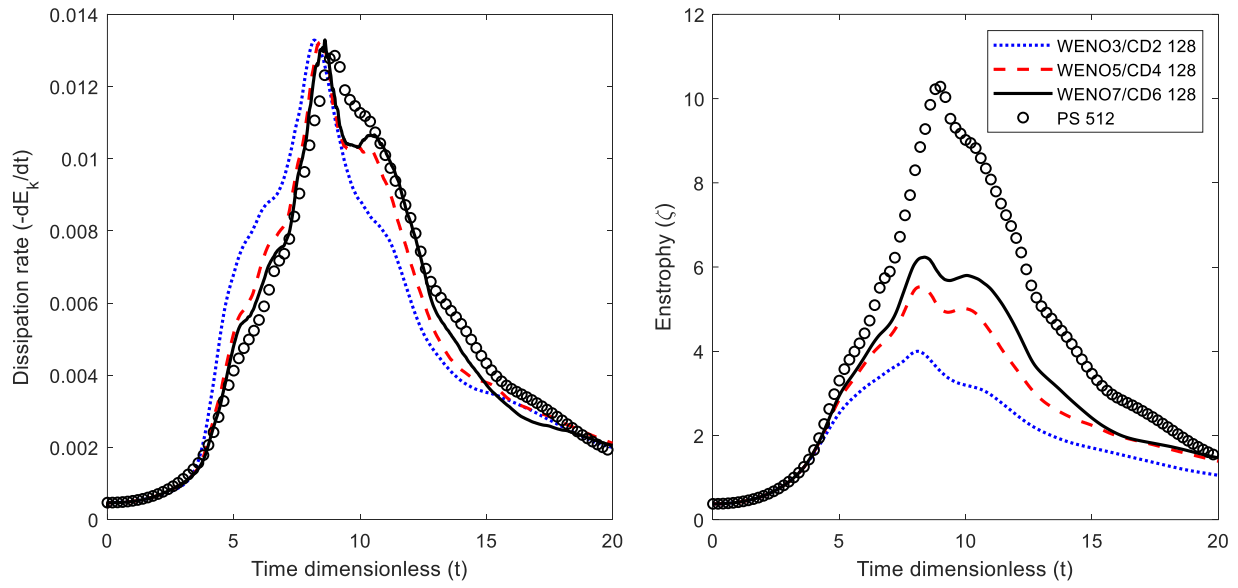


Fig. 22. Kinetic energy dissipation rate (left), the enstrophy (right) of mesh 128^3

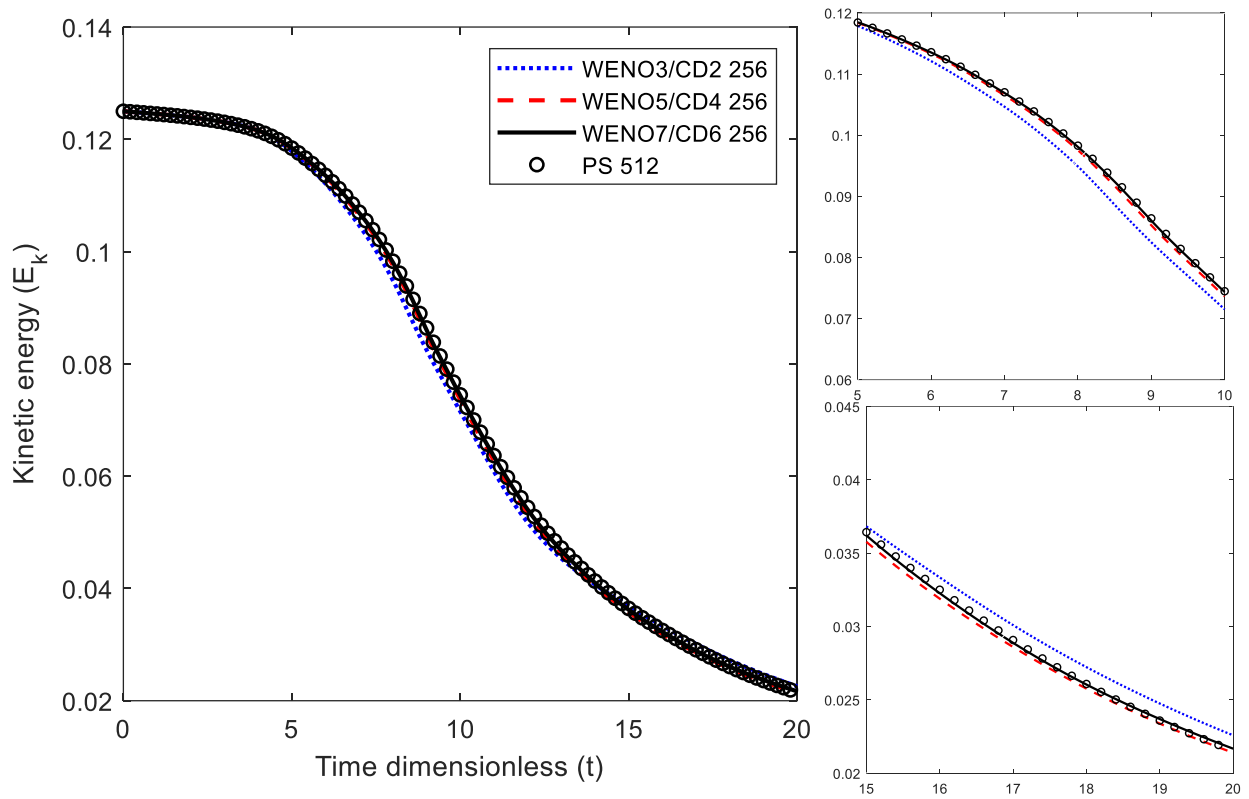


Fig. 23. The kinetic energy of mesh 256^3

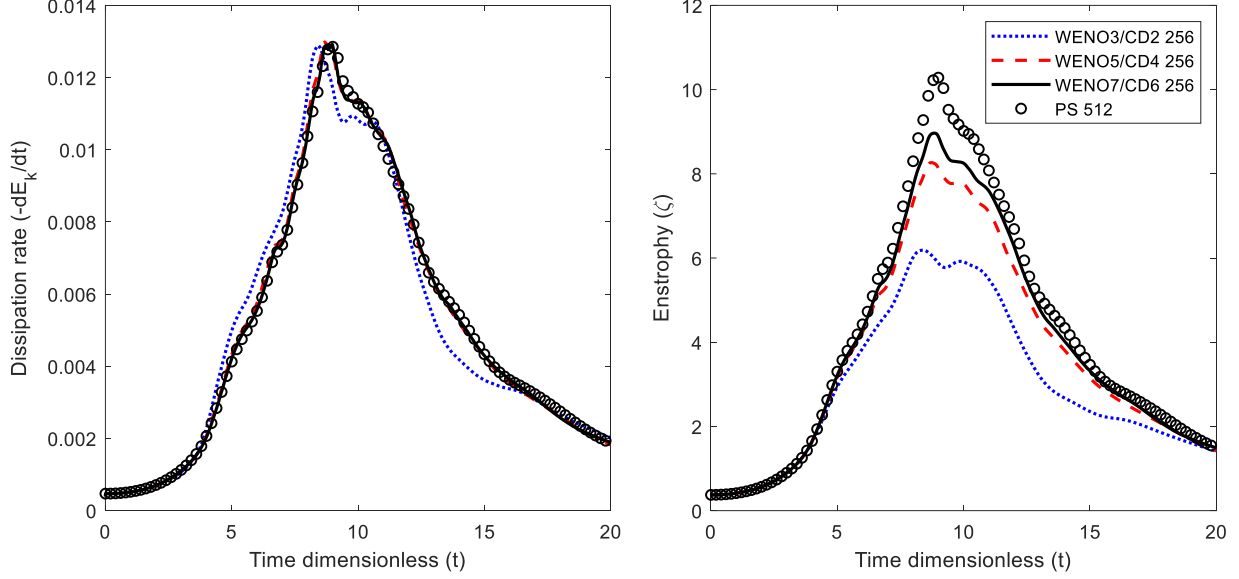


Fig. 24. Kinetic energy dissipation rate (left), the enstrophy (right) of mesh 256^3

The simulation results are presented in the order of successive mesh refinements. Figs. 19-24, shows the time history of the total kinetic energy decay and dissipation rates computed both by the direct computation of time derivative and also by the enstrophy. The results are compared for all three combinations of the schemes, i.e., WENO3/CD2, WENO5/CD4, and WENO7/CD6. The overall trends of the results can be summarized that the current simulation rapidly approaches the pseudo-spectral results (Rees, Leonard, Pullin, & Koumoutsakos, 2011) as the mesh refines and the order of the scheme improves.

The detail of the comparative study can be discussed as follows. Figs. 19, 21, and 23 show the history of total kinetic energy decay. It starts with relatively mild slopes and then it decays rapidly as the turbulence develops until the final stage of simulation. For all three levels of meshes, all three combinations of the scheme show fast convergence to the reference solution of pseudo-spectral code. The small difference between the results is better noticeable for the dissipation rates presented in Figs. 20, 22, and 24. Faster convergence of kinetic energy dissipation rate $-dE_k/dt$ to the enstrophy ζ history seems due to the post-processing techniques. The kinetic energy dissipation rate $-dE_k/dt$ were obtained by the finite-difference of it in time. The time histories of E_k displayed in Figs. 19, 21, and 23 were firstly interpolated by cubic splines and the slope of the splines is approximated by the finite differences. The enstrophy history is obtained in simpler ways. The vorticity fields are firstly approximated by the standard second-order central difference and then integrated over the domain for the enstrophy. In doing so, it should be admitted that there is a possibility that higher-order accuracy may be contaminated by the low-order vorticity computation method. Nevertheless, the present results show the solution quality of the raw velocity field certainly affects the accuracy of global enstrophy history.

The turbulent decay process has been mentioned in previous studies (DeBonis, 2013; Mimeau, Mortazavi, & Cottet, 2019). The laminar stage is maintained until $t = 3$, after which the dissipation rate begins to increase and peaks around $t = 9$. The developed turbulent flow occurs from $9 < t \leq 13$, after which the turbulent flow decays. In general, our methods, particularly the WENO7/CD6 and WENO5/CD4 methods, have produced results that are consistent with the above statement. However, our WENO3/CD2 method appears to have picked a bit early $t = 8.5$. We can see from the comparison of enstrophy that the WENO7/CD6 scheme produced the closest to the PS result.

The qualitative comparison can be made further by the local vorticity fields at a specific time moment. In Figs. 25-30, the performance of the present simulation is compared to the reference vorticity field of the pseudo-spectral solution (Rees, Leonard, Pullin, & Koumoutsakos, 2011). The vorticity in the WENO3/CD2 scheme is shaped like an oval over a large area, but the structure is not well defined. The structure becomes less oval, the curve of the shape becomes clearer and more defined, and the peak vorticity level increases as the scheme is used higher as WENO5/CD4 and WENO7/CD6. The WENO5/CD4 and WENO7/CD6 solutions differ only slightly, and the plots closely resemble the reference solution. Compared with the solution of the Pseudo-Spectral method, the solution of WENO7/CD6 detected the local extremum (indicated by a closed contour) around region $y = 0.8$, while the other schemes cannot do that.

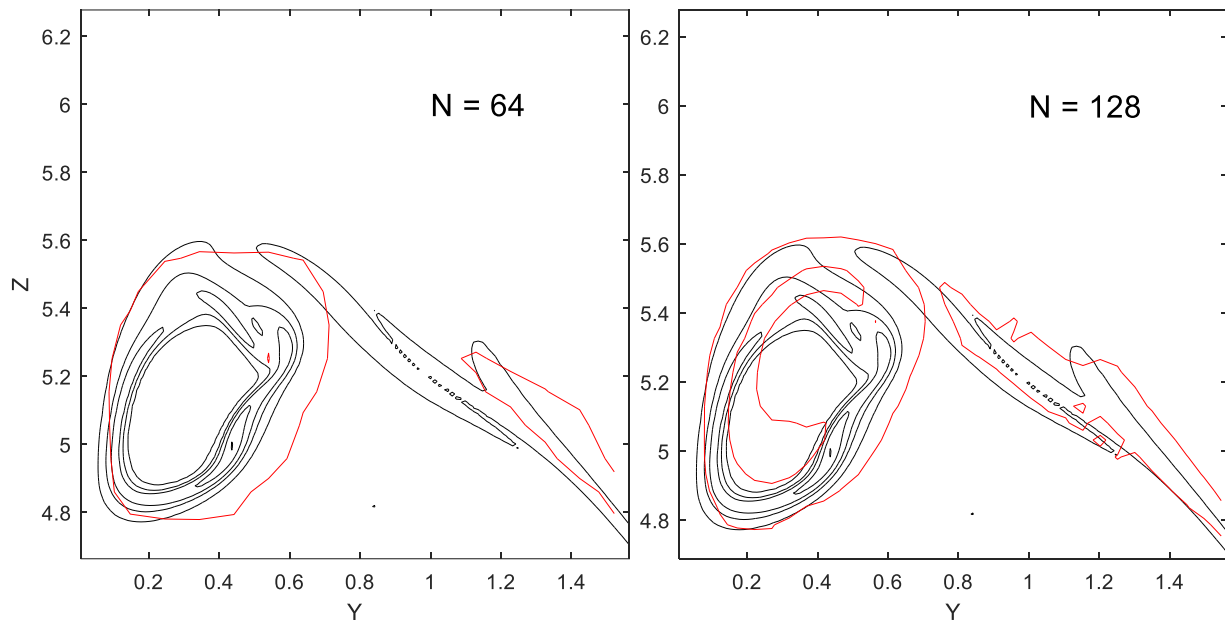


Fig. 25. Contours of the vorticity norm from grid resolution 64^3 (left), 128^3 (right) of WENO3/CD2 scheme at $t = 8$ on the plane $x = 0$, in the region $z = \frac{3\pi}{2}$ to 2π and $y = 0$ to $\frac{\pi}{2}$, compared with PS 512 (black solid).

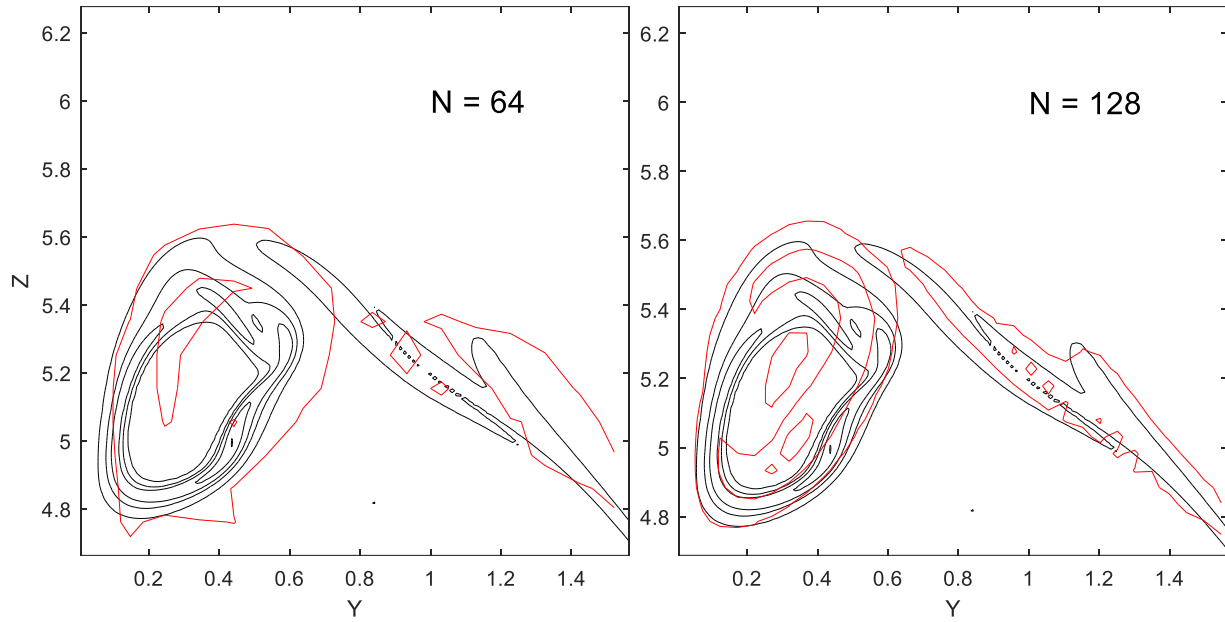


Fig. 26. Contours of the vorticity norm from grid resolution 64^3 (left), 128^3 (right) of WENO5/CD4 scheme at $t = 8$ on the plane $x = 0$, in the region $z = \frac{3\pi}{2}$ to 2π and $y = 0$ to $\frac{\pi}{2}$, compared with PS 512 (black solid).

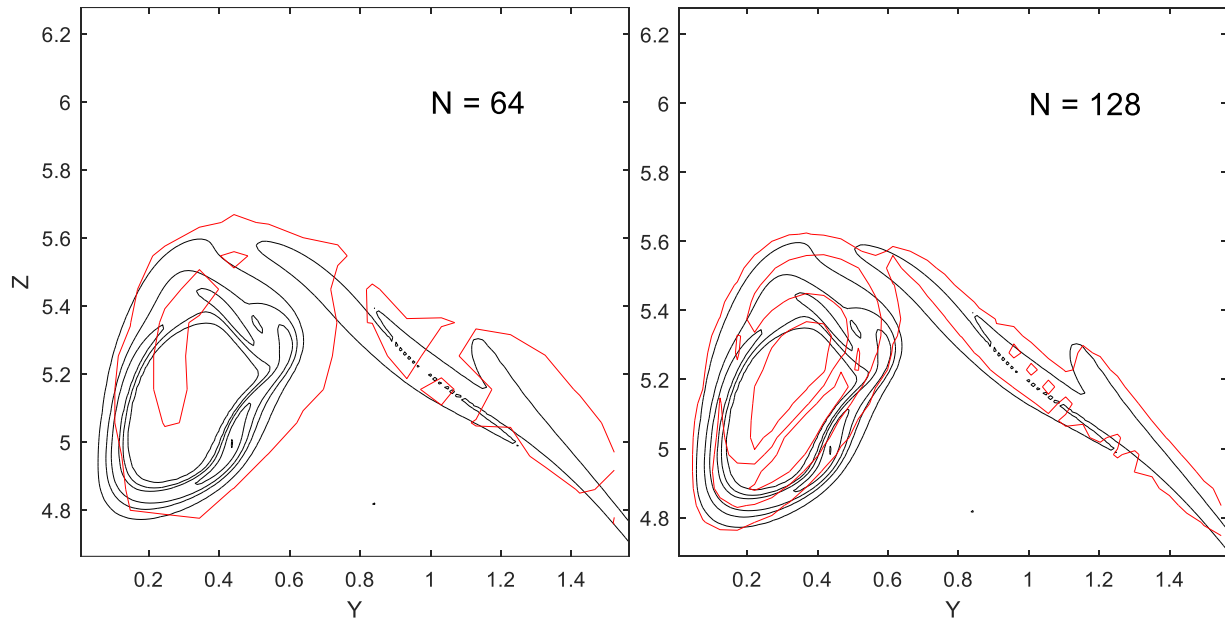


Fig. 27. Contours of the vorticity norm from grid resolution 64^3 (left), 128^3 (right) of WENO7/CD6 scheme at $t = 8$ on the plane $x = 0$, in the region $z = \frac{3\pi}{2}$ to 2π and $y = 0$ to $\frac{\pi}{2}$, compared with PS 512 (black solid).

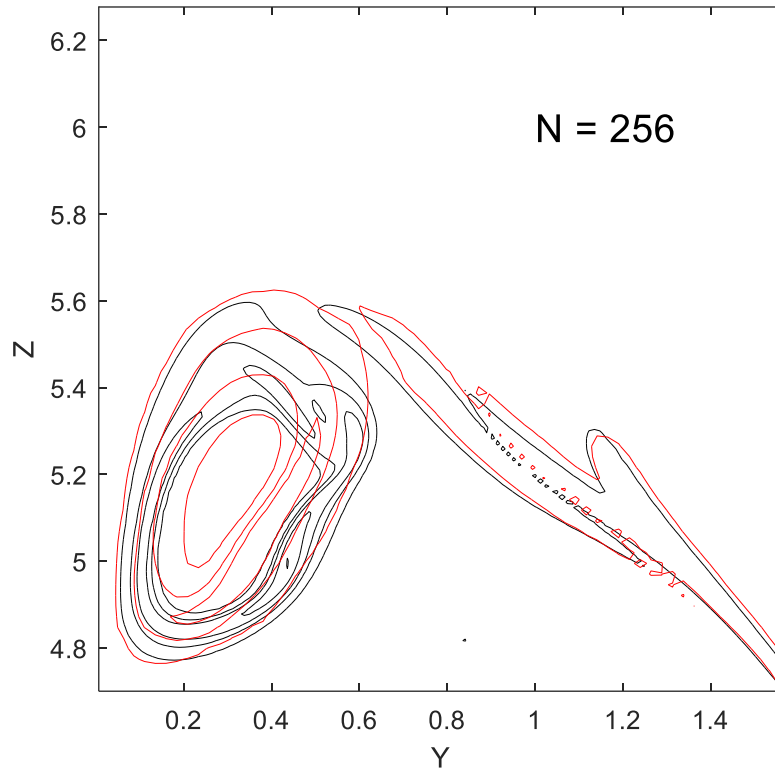


Fig. 28. Contours of the vorticity norm from grid resolution 256^3 of CD2/WENO3 scheme

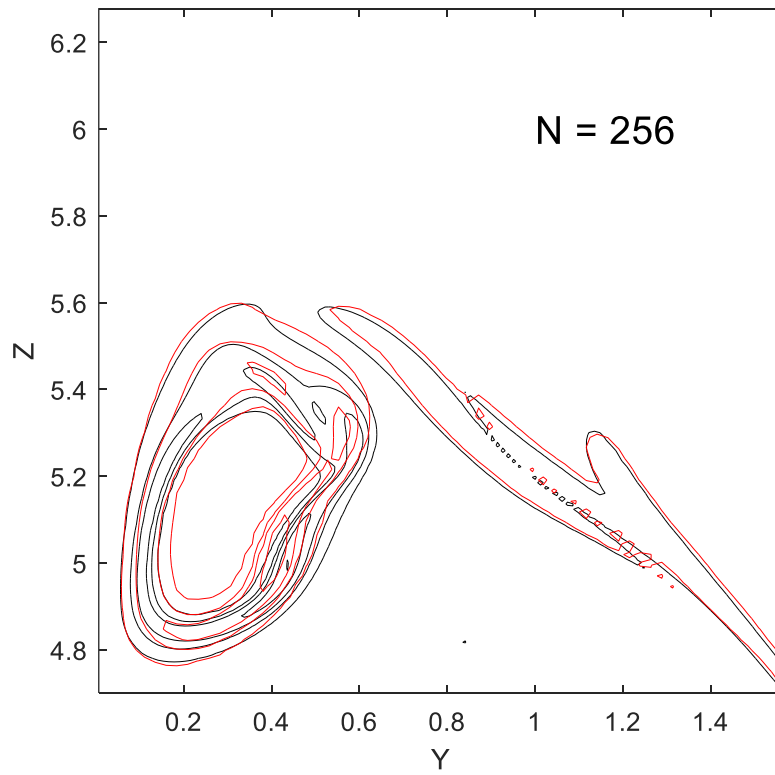


Fig. 29. Contours of the vorticity norm from grid resolution 256^3 CD4/WENO5 scheme

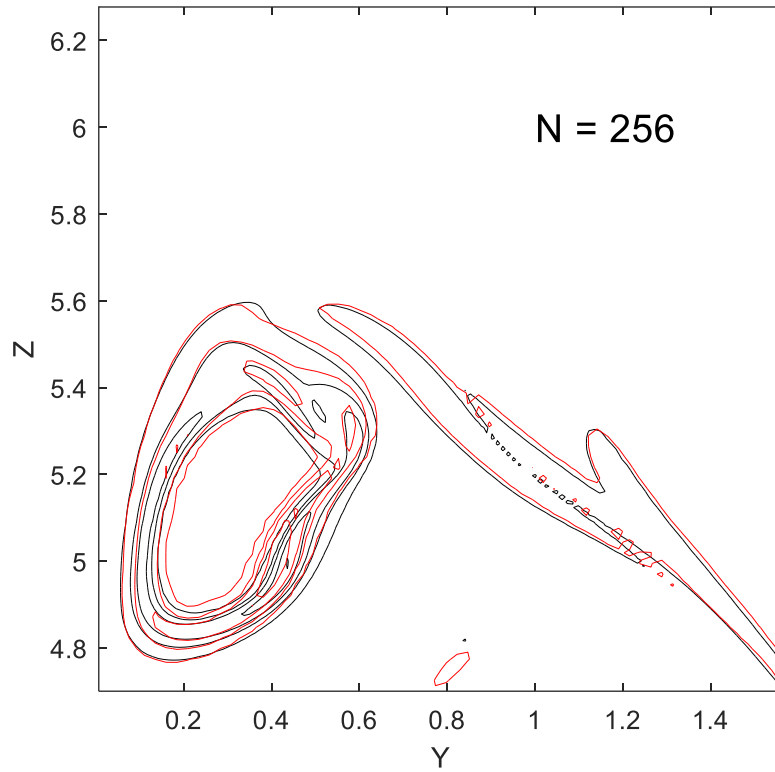


Fig. 30. Contours of the vorticity norm from grid resolution 256^3 CD6/WENO7 scheme

In Figs. 31-36, the energy spectra of the present methods are shown at the two different time moments, i.e., one relatively early stage of the turbulence onset $t = 5$ and the other slight after the peak of turbulence dissipation $t = 10$. As turbulence develops, the width of the inertial subrange whose slope is parallel to Kolmogorov's $-5/3$ expectation becomes extends further and this trend is reproduced by the present schemes as mesh refines and also solution order increases.

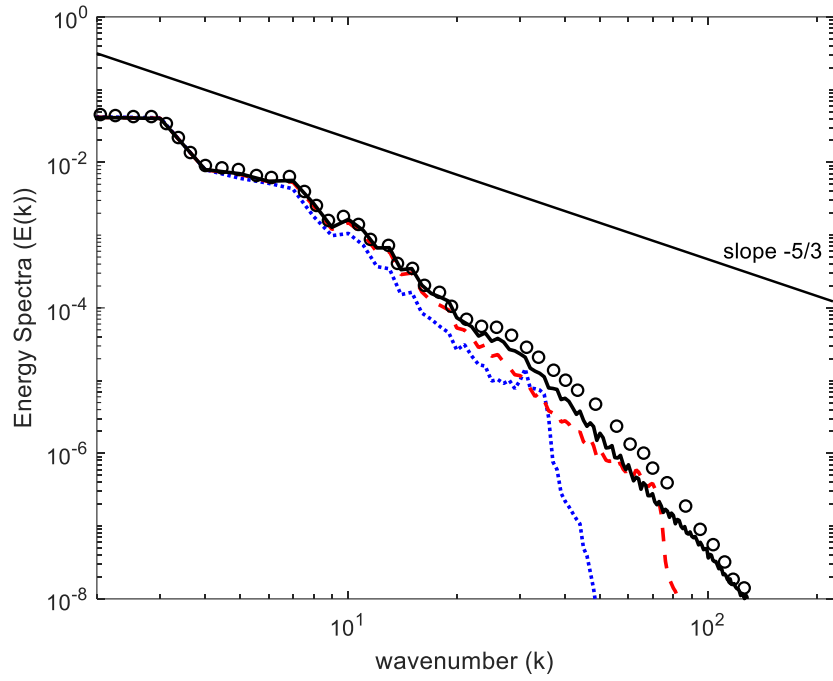


Fig. 31. The Kinetic energy spectra of WENO3/CD2 at $t = 5$

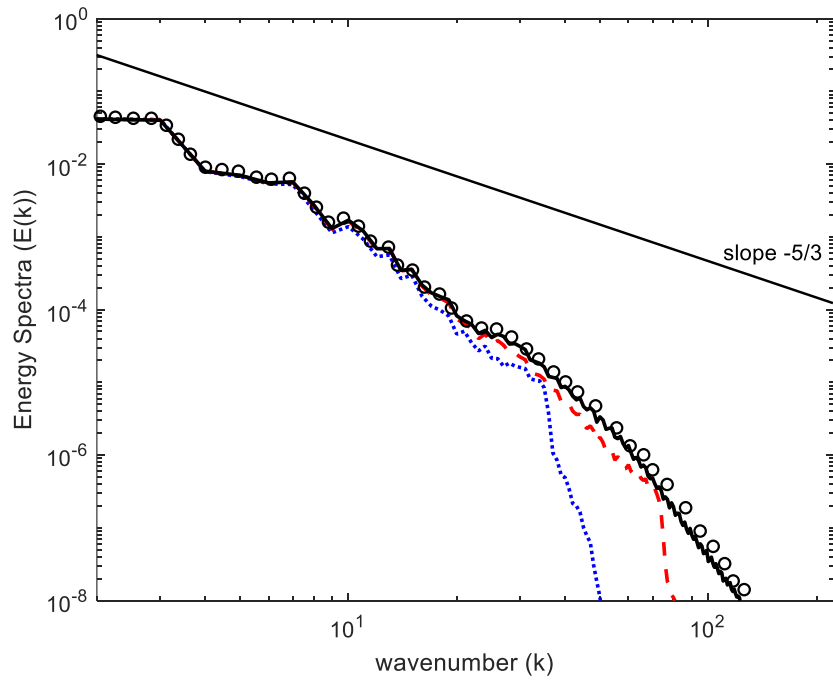


Fig. 32. The Kinetic energy spectra of WENO5/CD4 at $t = 5$

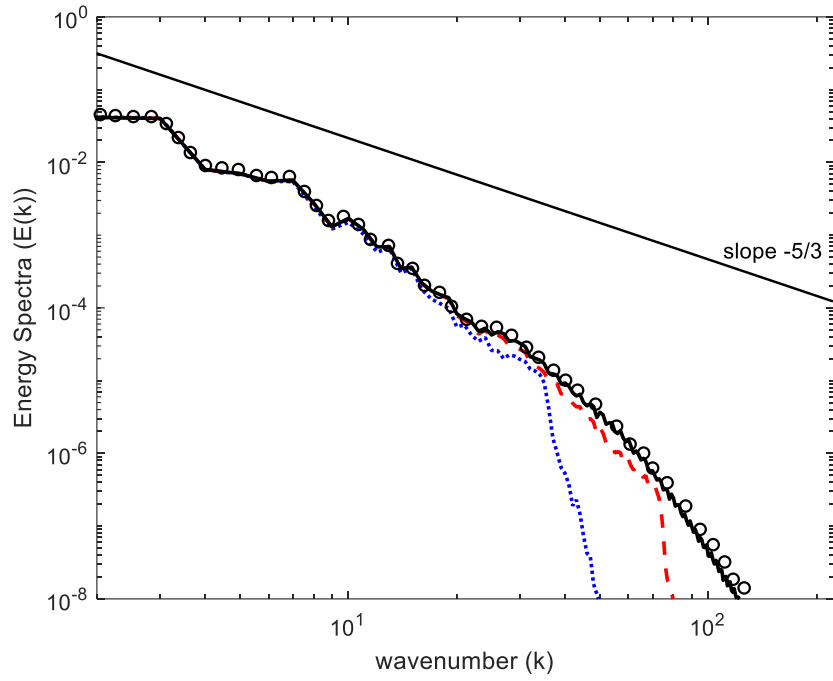


Fig. 33. The Kinetic energy spectra of WENO7/CD6 at $t = 5$

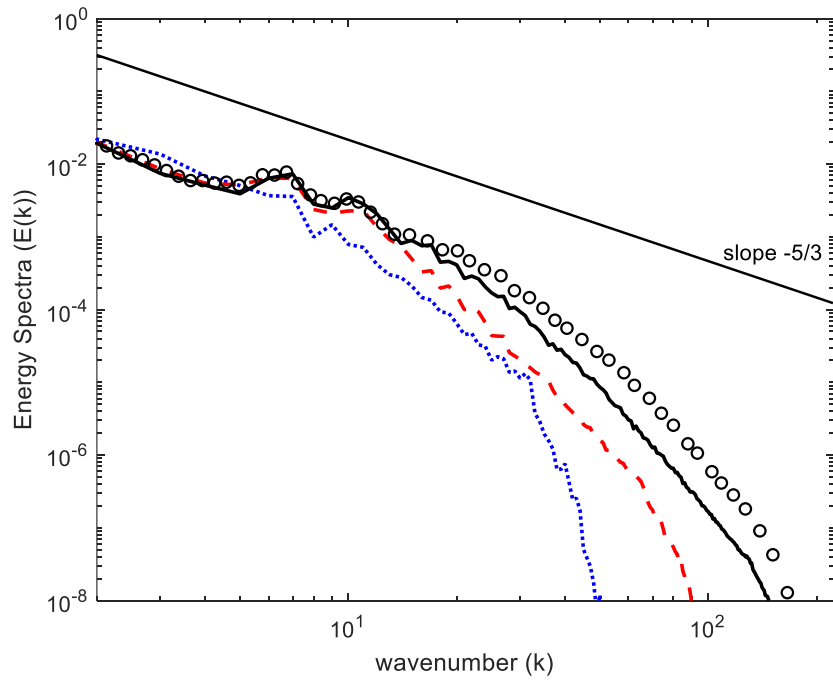


Fig. 34. The Kinetic energy spectra of WENO3/CD2 at $t = 10$

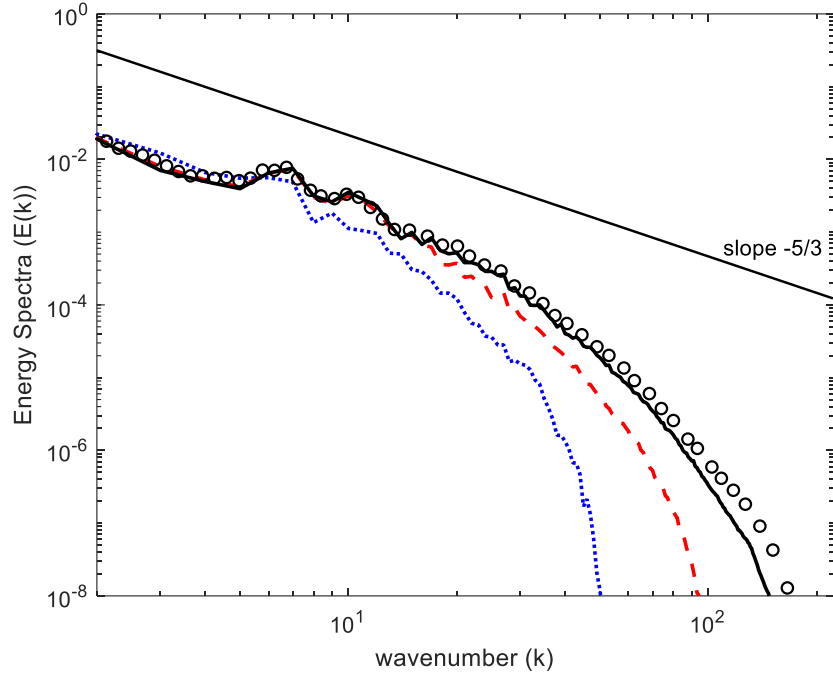


Fig. 35. The Kinetic energy spectra of WENO5/CD4 at $t = 10$

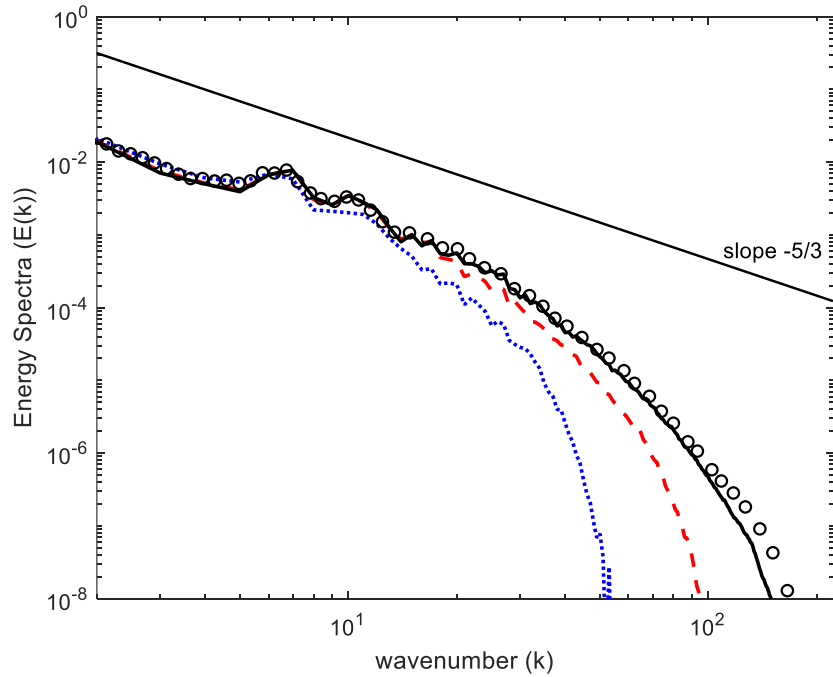


Fig. 36. The Kinetic energy spectra of WENO7/CD6 at $t = 10$

The vortex structure of the flow field is also visualized by using the λ_2 method using the flow field at $t = 9$, which corresponds to the peak of the dissipation rate. As noticeable from the graphics, the vortex structures become finer and stronger as mesh refines and the order of accuracy improves. It also needs to be mentioned that for the vorticity computation, which is the raw data for the λ_2 computation, merely the second-order central differences are utilized for all schemes. This implies

that such a vorticity field could be many finers depending on higher-order post-processing techniques, which is not the major subject of this paper.

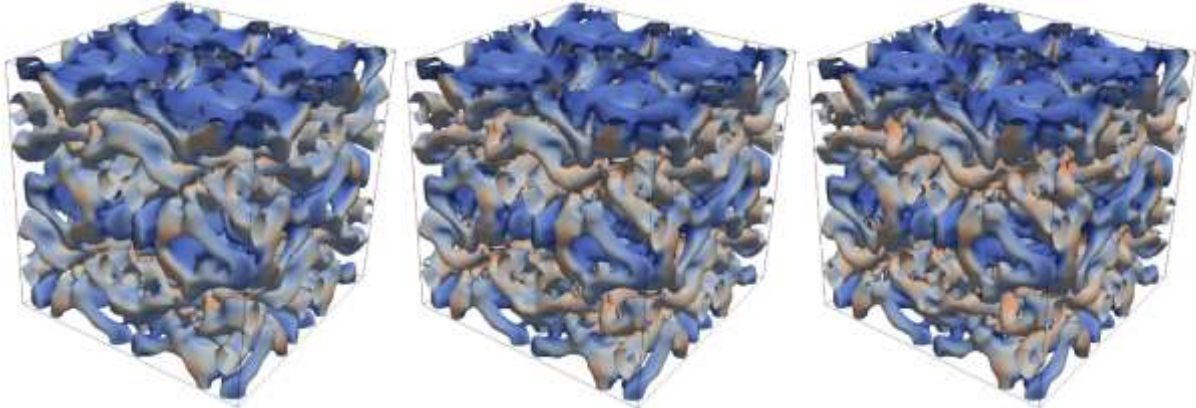


Fig. 37. The temporal evolution of the flow field of WENO3/CD2 (left), WENO5/CD4 (mid) and WENO7/CD6 (right) from grid resolution 64^3 at $t = 9$. The colored with the velocity magnitude.

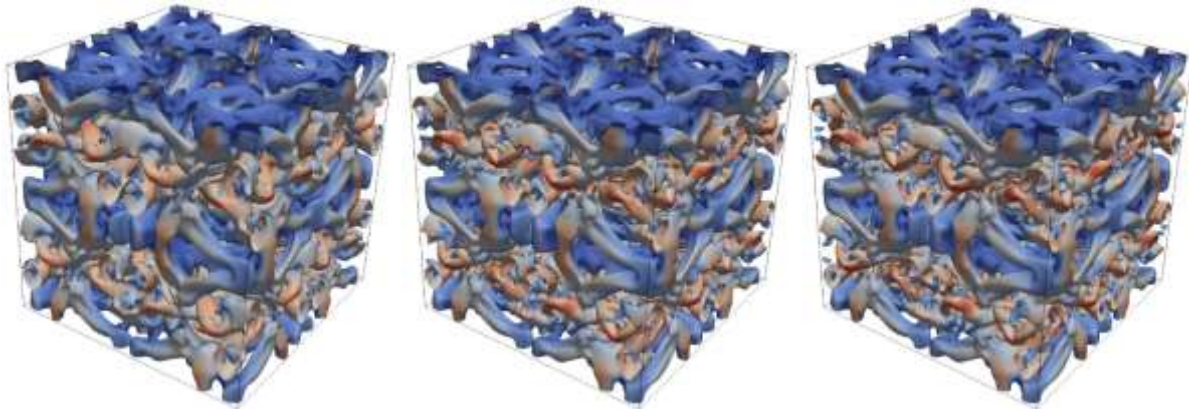


Fig. 38. The temporal evolution of the flow field of WENO3/CD2 (left), WENO5/CD4 (mid) and WENO7/CD6 (right) from grid resolution 64^3 at $t = 9$. The colored with the velocity magnitude.

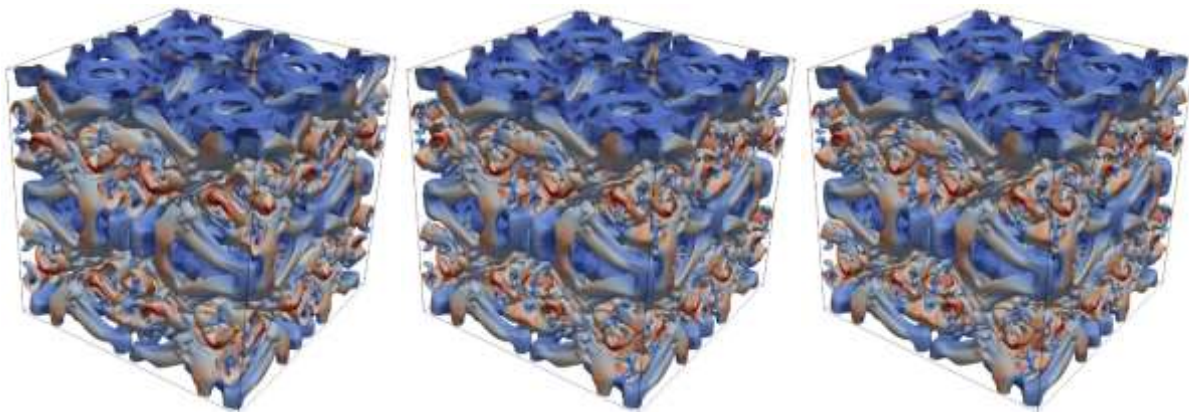


Fig. 39. The temporal evolution of the flow field of WENO3/CD2 (left), WENO5/CD4 (mid) and WENO7/CD6 (right) from grid resolution 64^3 at $t = 9$. The colored with the velocity magnitude.

4.2.2 The higher Reynolds number cases, $Re = 16,000$ and $160,000$

Perhaps one of the major advantages of the present WENO-based method is that the high Reynolds number turbulent flows could be simulated without any user-specified modification or parameter adjustment. The stable and accurate turbulent simulation capability is demonstrated for even higher Reynolds numbers, i.e. $Re = 1.6 \times 10^4$ and 1.6×10^5 . No modification has been made from the previous $Re = 1,600$ case, except the Reynolds numbers.

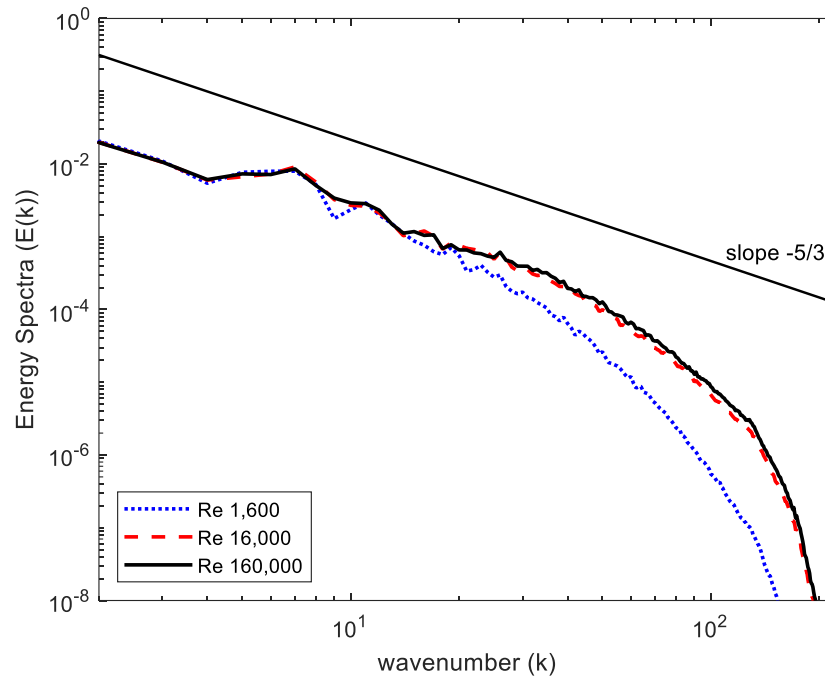


Fig. 40. The Kinetic spectra energy of high Re at $t = 9$ of WENO7/CD6 scheme with mesh resolution 256^3

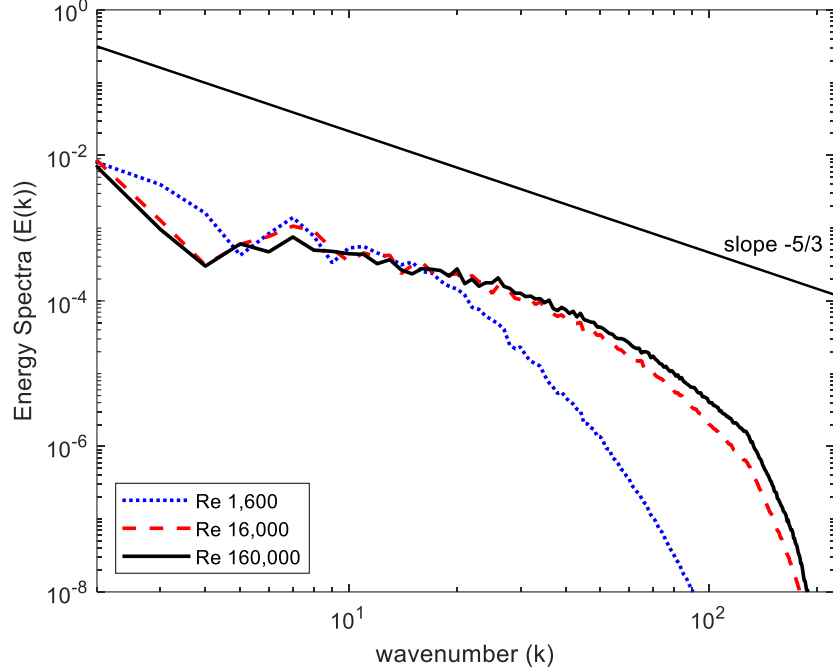


Fig. 41. The Kinetic spectra energy of high Re at $t = 20$ of WENO7/CD6 scheme with mesh resolution 256^3

The results are presented in Figs. 40-42. To the best of the authors' knowledge, there are no DNS results available for such a high Reynolds case, and no comparison was made with a reference solution. Nevertheless, a relative comparison between our results is presented which is directly attributed to the Reynolds number effect resolved by current methods. As shown in Fig. 22, the inertial subrange extends and this trends down to the highest wavenumber $k_{max} = 2\pi/\lambda_{min}$, where λ_{min} is the minimum size of the vortex 256^3 mesh can represent. This is far larger than Kolmogorov's microscale, and hence the current WENO-based simulation without any sub-grid scale (SGS) eddy viscosity would be considered an implicit large-eddy simulation (iLES). Even without any explicit SGS eddy viscosity model, the current simulation shows stable results. The scale resolving capability is also confirmed by the vortex structure displayed in Fig. 23. It is quite self-explanatory that the wider spectrum of vortex sizes as Reynolds number increased. The relatively smaller difference in vortex structures at the initial stage amplifies to a wider disparity in the vortex size as the flow evolves. The similarity of energy spectra between the $Re = 16,000$ and $Re = 160,000$ delineated in Figs. 40-41 is also reproduced by a similarity in vortex structure between the two higher Reynolds number cases as displayed in Fig. 42. This could be an indication that the present WENO-based projection method affects little dissipation on the energy cascade within the resolved viscous subrange, but mostly dampens out the smaller size eddies below that of the grid cell. In other words, the WENO-based scheme seems to be implicitly equipped with just the right amount of dissipation so that the simulation remains to be stable and perhaps accurate too.

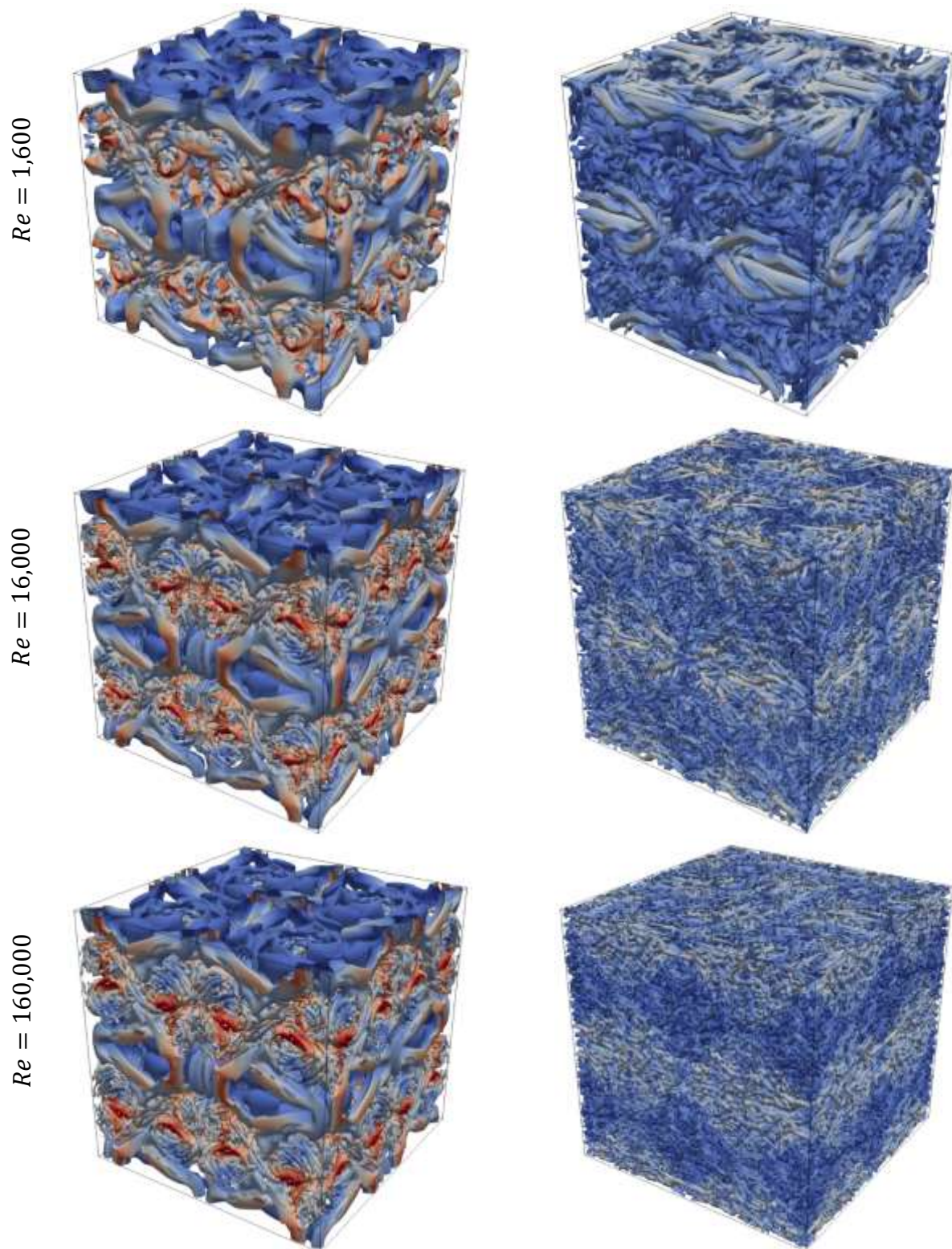


Fig. 42. Scale resolving capability of the present WENO-based scheme (WENO7/CD6) for increasing Reynolds numbers. The vortex structure is visualized using λ_2 method at $t = 9$ (left) and $t = 20$ (right), where the iso-surfaces of the $\lambda_2 = -10^{-5}$. The colored with the velocity magnitude.

5 Conclusions

In this presentation, we have shown the successful development and implementation of the high-order WENO-based semi-implicit projection method for incompressible turbulent flows. In general, WENO(k)/CD($k - 1$) the scheme is recommended for preserving the intended order of accuracy. The formal orders of accuracy were demonstrated by solving the TGV2d problems, and the third, fifth, and seventh order of accuracies are confirmed for WENO3/CD2, WENO5/CD4, and WENO7/CD6 schemes respectively. Based on the given meshes, the highest order combination, i.e., WENO7/CD6 is recommended due to its little computational overhead. Such methods are demonstrated for a series of higher Reynolds number turbulent flows and the results are compared to the pseudo-spectral reference data available. The present method is expected to be used as an efficient simulation tool for high Reynolds number turbulent flow in various cases.

References

- A.Shetty, D., C.Fisher, T., Chunekar, A. R., & Frankel, S. H. (2010). High-order incompressible large-eddy simulation of fully inhomogeneous turbulent flows. *Journal of Computational Physics*, 229(23), 8802-8822. doi:10.1016/j.jcp.2010.08.011
- Armfield, S., & Street, R. (2002). An analysis and comparison of the time accuracy of fractional-step methods for the Navier–Stokes equations on staggered grids. *International Journal for Numerical Methods In Fluids*, 38(3), 255-282. doi:10.1002/flid.217
- Balay, S., Abhyankar, S., Adams, M., J. B., Brune, P., Buschelman, K., . . . Sanan, P. (2015). *PETSc Users Manual*. U.S: Argonne National Laboratory.
- Bell, J., Colell, P., & MGlaz, H. (1989). A second-order projection method for the incompressible navier-stokes equations. *Journal of Computational Physics*, 85(2), 257-283. doi:10.1016/0021-9991(89)90151-4
- Brown, D., Cortez, R., & Minion, M. (2001). Accurate Projection Methods for the Incompressible Navier–Stokes Equations. *Journal of Computational Physics*, 168(2), 464-499. doi:10.1006/jcph.2001.6715
- Chen, Y.-N., Yang, S.-C., & Yang, J.-Y. (1999). Implicit weighted essentially non-oscillatory schemes for the incompressible Navier–Stokes equations. *International Journal for Numerical Methods in Fluids*, 31(3), 747-765. doi:10.1002/(SICI)1097-0363(19991030)31:4<747::AID-FLD901>3.0.CO;2-F
- Chorin, A. (1967). A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics*, 2(1), 12-26. doi:10.1016/0021-9991(67)90037-X
- Chorin, A. (1968). Numerical solution of the Navier-Stokes equations. *Mathematics of Computation*, 22, 745-762. doi:10.1090/S0025-5718-1968-0242392-2
- Cockburn, B., Shu, C.-W., Johnson, C., & Tadmor, E. (1997). Advanced Numerical Approximation of Nonlinear Hyperbolic Equations. In *Lecture Notes in Mathematics* (Vol. 1697, pp. 325-432).
- Courant, R. F., & Lewy, H. (1928). On the Partial Difference Equations of Mathematical Physics. *Math. Ann*, 100, 32.
- DeBonis, J. (2013). Solutions of the Taylor-Green Vortex Problem Using High-Resolution Explicit Finite Difference Methods. *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. Grapevine (Dallas/Ft. Worth Region), Texas: AIAA. doi:10.2514/6.2013-382
- Dietzsch, F. (2018). *Matlab code for the spectral analysis of homogenous isotropic turbulence*. Freiberg. Retrieved from <https://github.com/fdietzsc/hita>

- Guermonda, J., Mineev, P., & Shen, J. (2006). An overview of projection methods for incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 195(44-47), 6011-6045. doi:10.1016/j.cma.2005.10.010
- Harlow, F. H., & Welch, J. E. (1965). Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface. *Physics of Fluids*, 8(12), 2182-2189. doi:10.1063/1.1761178
- Henshaw, D. W., & Schwendeman, W. D. (2006). Moving overlapping grids with adaptive mesh refinement for high-speed reactive and non-reactive flow. *Journal of Computational Physics*, 216(2), 744-779. doi:10.1016/j.jcp.2006.01.005
- Hu, F., Hussaini, M., & Manthey, J. (1996). Low-Dissipation and Low-Dispersion Runge–Kutta Schemes for Computational Acoustics. *Journal of Computational Physics*, 124(1), 177-191. doi:10.1006/jcph.1996.0052
- Jeong, J., & Hussain, F. (1995). On the identification of a vortex. *Journal of Fluid Mechanics*, 285, 69–94. doi:10.1017/S0022112095000462
- Jiang, G.-S., & Shu, C.-W. (1996). Efficient Implementation of Weighted ENO Schemes. *Journal of Computational Physics*, 126(1), 202-228. doi:10.1006/jcph.1996.0130
- Jiang, G.-S., & Wu, C.-c. (1999). A High-Order WENO Finite Difference Scheme for the Equations of Ideal Magnetohydrodynamics. *Journal of Computational Physics*, 150(2), 561-594. doi:10.1006/jcph.1999.6207
- Johnston, H., & Liu, J.-G. (2004). Accurate, stable and efficient Navier–Stokes solvers based on explicit treatment of the pressure term. *Journal of Computational Physics*, 199(1), 221-259. doi:10.1016/j.jcp.2004.02.009
- K. Lele, S. (1992). Compact finite difference schemes with spectral-like resolution. *Journal of Computational Physics*, 103(1), 16-42. doi:10.1016/0021-9991(92)90324-R
- Kallinderis, Y., & Ahn, H. (2005). Incompressible Navier–Stokes method with general hybrid meshes. *Journal of Computational Physics*, 210(1), 75-108. doi:10.1016/j.jcp.2005.04.002
- Kim, J., & Moin, P. (1985). Application of a fractional-step method to incompressible Navier–Stokes equations. *Journal of Computational Physics*, 59(2), 308323. doi:10.1016/0021-9991(85)90148-2
- Kim, J., Moin, P., & Moser, R. (1987). Turbulence statistics in fully developed channel flow at low Reynolds number. *Journal of Fluid Mechanics* , 177, 133-166. doi:doi.org/10.1017/S0022112087000892
- Kiris, C. C., Kwak, D., Rogers, S., & Chang, I. (1997). Computational Approach for Probing the Flow Through Artificial Heart Devices. *Journal of Biomechanical Engineering*, 119(4), 452-460. doi:10.1115/1.2798293.

- Kwak, D., & Kiris, C. C. (1989). *Computation of Viscous Incompressible Flows*. Springer Dordrecht. doi:10.1007/978-94-007-0193-9
- Kwak, D., Chang, J. L., Shanks, S. P., & Chakravarthy, S. R. (1986). A three-dimensional incompressible Navier-Stokes flow solver using primitive variables. *AIAA Journal*, 24(3), 390-396. doi:10.2514/3.9279
- Liu, X.-D., Osher, S., & Chan, T. (1994). Weighted Essentially Non-oscillatory Schemes. *Journal of Computational Physics*, 115(1), 200-212. doi:10.1006/jcph.1994.1187
- Luo, H., D.Baum, J., & Löhner, R. (2007). A Hermite WENO-based limiter for discontinuous Galerkin method on unstructured grids. 225(1), 686-713. doi:10.1016/j.jcp.2006.12.017
- Mimeau, C., Mortazavi, I., & Cottet, G.-H. (2019). A Semi-Lagrangian Vortex Penalization Method. *Journal of Mathematical Study*, 52(3), 277-298. doi: 10.4208/jms.v52n3.19.04
- Mishra, S., Parés-Pulido, C., & Pressel, K. G. (2021). Arbitrarily High-Order (Weighted) Essentially Non-Oscillatory Finite Difference Schemes for Anelastic Flows on Staggered Meshes. *Communications in Computational Physics*, 29(5), 1299-1335. doi:10.4208/CICP.OA-2020-0046
- Navah, F., Plata, M. d., & Couaillier, V. (2020). A high-order multiscale approach to turbulence for compact nodal schemes. *Computer Methods in Applied Mechanics and Engineering*, 363, 112885. doi:10.1016/j.cma.2020.112885
- Perot, J. (1993). An Analysis of the Fractional Step Method. *Journal of Computational Physics*, 108(1), 51-58. doi:10.1006/jcph.1993.1162
- Qiu, J., & Shu, C.-W. (2005). Runge--Kutta Discontinuous Galerkin Method Using WENO Limiters. *SIAM Journal on Scientific Computing*, 26(3), 907-929. doi:10.1137/S1064827503425298
- Rees, W. M., Leonard, A., Pullin, D., & Koumoutsakos, P. (2011). A comparison of vortex and pseudo-spectral methods for the simulation of periodic vortical flows at high Reynolds numbers. *Journal of Computational Physics*, 230(8), 2794-2805. doi:10.1016/j.jcp.2010.11.031
- Rogers, S. E., & Kwak, D. (1990). Upwind differencing scheme for the time-accurate incompressible Navier-Stokes equations. *AIAA Journal*, 28(2), 253-262. doi:10.2514/3.10382
- Rogers, S. E., Kwak, D., & Kiris, C. (1991). Steady and unsteady solutions of the incompressible Navier-Stokes equations. *AIAA Journal*, 29(4), 603-610. doi:10.2514/3.10627
- Salvi, R. (2002). *The Navier-Stokes equations : theory and numerical methods*. New York: Marcel Dekker.
- Shang-Gui, C., Abdellatif, O., Julien, F., & Yannick, H. (2017). Moving Immersed Boundary Method. *Numerical Methods in Fluids*, 85(5), 288-323.

- Shu, C.-W. (1998). Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. In *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations: Lectures given at the 2nd Session of the Centro Internazionale Matematico Estivo (C.I.M.E.) held in Cetraro, Italy, June 23--28, 1997* (pp. 325-432). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/BFb0096355
- Shu, C.-W. (2009). High Order Weighted Essentially Nonoscillatory Schemes for Convection Dominated Problems. *SIAM Review*, 51(1), 82-162. doi:10.1137/070679065
- Shu, C.-W., & Osher, S. (1988). Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics*, 77(2), 439-471. doi:10.1016/0021-9991(88)90177-5
- Shu, C.-W., & Osher, S. (1989). Efficient implementation of essentially non-oscillatory shock-capturing schemes, II. *Journal of Computational Physics*, 83(1), 32-78. doi:10.1016/0021-9991(89)90222-2
- Taylor, I. G., & Green, E. A. (1937). Mechanism of the production of small eddies from large ones. *Proceedings of the Royal Society A*, 158(895), 499-521. doi:10.1098/rspa.1937.0036
- Témam, R. (1969). Sur l'approximation de la solution des équations de Navier-Stokes par la méthode des pas fractionnaires (II). *Archive for Rational Mechanics and Analysis*, 33, 377-385. doi:10.1007/BF00247696
- Timmermans, L., Mineev, P., & Vosse, F. V. (1996). An Approximate Projection Scheme for Incompressible Flow using Spectral Elements. *International Journal for Numerical Methods In Fluids*, 22(7), 673-688. doi:10.1002/(SICI)1097-0363(19960415)22:7<673::AID-FLD373>3.0.CO;2-O
- Zhang, J., & L. Jackson, T. (2009). A high-order incompressible flow solver with WENO. *Journal of Computational Physics*, 228(7), 2426-2442. doi:10.1016/j.jcp.2008.12.009
- Zhu, J., Zhong, X., Shu, C.-W., & Qiu, J. (2013). Runge–Kutta discontinuous Galerkin method using a new type of WENO limiters on unstructured meshes. *Journal of Computational Physics*, 248, 200-220. doi:10.1016/j.jcp.2013.04.012