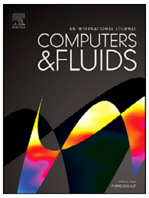
Computers and Fluids 189 (2019) 94–107

Contents lists available at ScienceDirect

Computers and Fluids

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



Benchmark solutions

Numerical symmetry-preserving techniques for low-dissipation shock-capturing schemes

Nico Fleischmann[*], Stefan Adami, Nikolaus A. Adams

Department of Mechanical Engineering, Chair of Aerodynamics and Fluid Mechanics, Technical University of Munich, Boltzmannstraße 15, Garching 85748, Germany

ARTICLE INFO

Article history:
Received 6 November 2018
Accepted 9 April 2019
Available online 24 May 2019

Keywords:
Symmetry breaking
Floating-point arithmetic
Low-dissipation schemes
High-resolution schemes
WENO
Shock-capturing

ABSTRACT

Modern applications of computational fluid dynamics involve complex interactions across scales such as shock interactions with turbulent structures and multiphase interfaces. Such phenomena, which occur at very small physical viscosity, require high-resolution and low-dissipation compressible flow solvers. Many recent publications have focused on the design of high-order accurate numerical schemes and provide e.g. weighted essentially non-oscillatory (WENO) stencils up to 17th order for this purpose. As shown in detail by different authors, such schemes tremendously decrease adverse effects of numerical dissipation. However, such schemes are prone to numerically induced symmetry breaking which renders validation for the targeted problem range problematic.

In this paper, we show that symmetry-breaking relates to vanishing numerical viscosity and is driven systematically by algorithmic floating-point effects which are no longer hidden by numerical dissipation. We propose a systematic procedure to deal with such errors by numerical and algorithmic formulations which respect floating-point arithmetic. We show that by these procedures inherent symmetries are preserved for a broad range of test cases with high-order shock-capturing schemes in particular in the high-resolution limit for both 2D and 3D.

1. Introduction

Many modern applications of computational fluid dynamics require high-order shock-capturing methods for the accurate prediction of compressible flow features involving shock waves and complex flow structures [1–4]. In order to achieve an efficient and accurate computation, high-order spatial reconstructions are employed together with essentially non-oscillatory (ENO) schemes to avoid spurious oscillations near discontinuities [5–7].

In their pioneering work, Harten et al. [8] introduced the essentially non-oscillatory (ENO) method. In this method the smoothest interpolation stencil out of a set of candidate stencils is selected to adapt the stencil to the actual solution. A further major development is the weighted-essentially non-oscillatory (WENO) method, first introduced by Liu et al. [9]. They proposed to construct a $(2r-1)$th-order accurate reconstruction from a weighted combination of several candidate stencils of order r. Effectively, the order $(2r-1)$ is recovered in smooth regions of the flow, whereas in the vicinity of discontinuities the ENO property is maintained. The work of Jiang and Shu [10] on the efficient implementation of WENO stencils allowed for a straightforward design of higher-order methods and marks a major breakthrough of the WENO methods family. Due to its robust behavior combined with the increased spatial order, their fifth-order variant WENO5-JS gained rapid popularity in numerical solutions of various kinds of applications not only in the field of fluid mechanics [11–14]. A thorough overview of recent applications is given in [15]. Different authors have further improved the method without changing the overall concept. Henrick et al. [16] dealt with the problem that classical WENO schemes are not able to recover the full order of accuracy near critical points by the introduction of a mapping procedure resulting in the WENO-M scheme. Furthermore, Borges et al. [17] suggested to introduce a high-order global smoothness indicator and increase the weights of non-smooth stencils. With the resulting WENO5-Z stencil the dissipation in the vicinity of discontinuities can be further decreased while maintaining the ENO property.

Despite the remarkable success in many applications, classical WENO schemes are still too dissipative to fully resolve the

[*] Corresponding author.
E-mail addresses: nico.fleischmann@tum.de (N. Fleischmann), stefan.adami@aer.mw.tum.de (S. Adami), nikolaus.adams@tum.de (N.A. Adams).

https://doi.org/10.1016/j.compfluid.2019.04.004

small-scale flow features of turbulence in direct numerical simulations. For this purpose, the order of the methods has been systematically increased starting with the work of Balsara and Shu [18], who followed the design rules of [10], and later by Gerolymos et al. [19], who extended and catalogued the reconstruction for stencils up to 17th order. However, these first high-order methods struggle with stability problems and tend to lose the ENO property without further adaptations [19]. Balsara et al. [20] recently introduced a new class of WENO schemes with adaptive order. By adaptively decreasing the order of the method, stability can be maintained while reaching higher-order in smooth areas. Furthermore, the reconstruction utilizes Legendre polynomials, which enables a more efficient calculation of the smoothness factors. Another approach to further reduce the dissipation is provided by the recently proposed class of targeted ENO (TENO) schemes [21,22]. While using the same stencil size as classical WENO schemes, the numerical dissipation is significantly reduced by an ENO-like stencil selection.

Although high-order methods require an increased amount of computational effort, the steady growth in computational power, that recently is mainly driven by the massive application of parallelization strategies, allows for extreme mesh resolutions (see e.g. Ref. [23]). Certainly, there is an increasing trade-off between reconstruction order and higher numerical resolutions in terms of absolute error level [5]. However, the combination of high-order methods with fine meshes leads to a strong decrease in numerical dissipation. Eventually, diminishing numerical dissipation fails to suppress physical instabilities of well-established verification cases for the inviscid Euler equations, which may develop symmetry breaking as numerical artifact, see e.g. the Rayleigh-Taylor instability [5,21,22,24–27] or the two-dimensional Riemann problems [28,29]. This sort of phenomena depends on the chosen scheme, on mesh resolution, but also on algorithm implementation as it is driven by roundoff errors. For moderate mesh resolutions, such symmetry breaking typically can be observed for small-scale structures near the resolution limit. However, for higher-order schemes with lower dissipation, the small-scale symmetry breaking exhibits inverse-cascade phenomena and affects the macroscopic flow evolution leading to a drastically changed flow topology [22,24].

Remacle et al. [24] already suspected floating-point inaccuracies due to roundoff errors to be the relevant mechanism behind numerical symmetry breaking for discontinuous Galerkin methods. Sutherland [29] introduced an approach to control the floating-point truncation error. Changes to physical variables that are smaller than a defined threshold are considered to be non-physical and therefore discarded. In the context of high-order (W/T)ENO-type methods in fluid mechanics, the problem has not been carefully addressed or it is qualified as induced by the low dissipation [21]. Shock or interface-driven (nearly) inviscid flows are the domain of high-resolution shock-capturing schemes. In this paper, we show that it is exactly these schemes which suffer from non-physical artifacts when they are validated for such flows.

We demonstrate the effect of inaccuracies in floating-point arithmetic for the numerical solution of the inviscid Euler equations with low-dissipative (W/T)ENO-type methods. We demonstrate the occurrence of numerical symmetry breaking and develop strategies to control and avoid this artifact. Unlike Sutherland [29], we do not reduce the accuracy range of the calculations. Instead, we show the importance of consistent algorithms that take into account the shortcomings of floating-point arithmetic systematically. Eventually, fully symmetric results can be achieved even for very low-dissipative schemes at large resolutions, thus enabling verification by grid-converged states and cross-code validation.

The remainder of the paper is organized as follows. Sections 2 and 3 briefly review the inviscid Euler equations as governing equations and the finite volume method along with the method-of-lines approach. Furthermore, the relevant relations for the HLLC approximate Riemann solver and the characteristic decomposition strategy that is applied along with the high-order stencils are presented. In Section 4, we identify numerical symmetry-breaking mechanisms and explain straightforward workarounds as well as a generalized solution strategy. In Section 5, we present simulation results for a set of benchmark cases and an extension of the implosion test case to three dimensions applying both high-order methods and high resolutions. Conclusions are drawn in Section 6.

2. Governing equations

An inviscid compressible flow evolves according to the three-dimensional Euler equations

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x + \mathbf{G}(\mathbf{U})_y + \mathbf{H}(\mathbf{U})_z = 0, \tag{1}$$

where $\mathbf{U}$ is the density of the conserved quantities mass ρ, momentum $\rho\mathbf{v} \equiv (\rho u, \rho v, \rho w)$ and total energy $E = \rho e + \frac{1}{2}\rho\mathbf{v}^2$, with e being the internal energy per unit mass.

The fluxes **F, G** and **H** are defined as

$$\mathbf{F} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(E+p) \end{pmatrix}, \quad \mathbf{G} = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ v(E+p) \end{pmatrix}, \quad \mathbf{H} = \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ w(E+p) \end{pmatrix}. \tag{2}$$

The set of equations is closed by the ideal-gas equation of state, where the pressure p is given by $p = (\gamma - 1)\rho e$ with the ratio of specific heats γ.

3. Finite volume approach

In this section, we briefly recall necessary details of the method-of-lines approach for finite volumes, which is widely used to solve systems of hyperbolic equations. We focus on aspects that are particularly important for considering floating-point inaccuracies. More detailed descriptions of the overall scheme can be found e.g. in [30,31].

The time evolution of the vector of cell-averaged conservative states $\bar{\mathbf{U}}$ is given by

$$\frac{d}{dt}\bar{\mathbf{U}}_i = \frac{1}{\Delta x}\big(\mathbf{f}_{i-\frac{1}{2},j,k} - \mathbf{f}_{i+\frac{1}{2},j,k} + \mathbf{g}_{i,j-\frac{1}{2},k} - \mathbf{g}_{i,j+\frac{1}{2},k} + \mathbf{h}_{i,j,k-\frac{1}{2}} - \mathbf{h}_{i,j,k+\frac{1}{2}}\big), \tag{3}$$

where $\Delta x (= \Delta y = \Delta z)$ is the cell size of a uniform Cartesian grid and **f, g** and **h** approximate the cell-face fluxes in x-, y- and z-direction, respectively. These fluxes are determined dimension-by-dimension from an approximate Riemann solver combined with a high-order WENO [10,18] or TENO [21] spatial reconstruction scheme. Additional volume source terms, such as gravitational acceleration, are omitted here for simplicity. The resulting ODE (3) is integrated in time using a second-order strong stability-preserving (SSP) Runge–Kutta scheme [32]. Note, that a higher-order time integration does neither trigger symmetry breaking nor affect the general results of the simulations presented in this paper and is therefore omitted to save computational cost.

3.1. HLLC approximate Riemann solver

The majority of state-of-the-art methods relies on approximate Riemann solvers since exact Riemann solvers are computationally expensive. Two common types of approximate Riemann solutions are the Roe [33] and the HLL(C) [34] approximation. With the restoration of the contact wave by Toro et al. [35], the HLLC approximation models all wave types of the Euler equations explicitly. Compared to the Roe approximation it is slightly more dissipative. Nevertheless, the better positivity-preservation of the first-order Godunov scheme with the HLLC approximation renders it favorable for problems with low density or pressures [36]. We focus on a HLLC flux formulation, although for the high-order reconstructions as used in this work, it does not fully maintain positivity, see also [37]. We emphasize that all concepts or workarounds essentially apply also to other approximate Riemann solvers.

As "HLLC solver" we define an algorithm that delivers approximate cell-face fluxes of the Riemann problem with reconstructed left and right face states, $\mathbf{U}^L$ and $\mathbf{U}^R$, respectively. Following the compact formulation of Johnsen and Colonius [38], the HLLC flux is defined as

$$\mathbf{f}^{HLLC} = \frac{1+sign(s^*)}{2}\left[\mathbf{f}^L + s^-\left(\mathbf{U}^{*L} - \mathbf{U}^L\right)\right] + \frac{1-sign(s^*)}{2}\left[\mathbf{f}^R + s^+\left(\mathbf{U}^{*R} - \mathbf{U}^R\right)\right], \tag{4}$$

where two intermediate states, $\mathbf{U}^{*L}$ and $\mathbf{U}^{*R}$, are separated by the contact wave and are determined from

$$\boldsymbol{U}^{*K} = \frac{s^K - u^K}{s^K - s^*}\begin{pmatrix} \rho^K \\ \rho^K s^* \\ \rho^K v^K \\ \rho^K w^K \\ E^K + \left(s^* - u^K\right)\left(\rho^K s^* + \frac{p^K}{s^K - u^K}\right) \end{pmatrix} \tag{5}$$

with $k = L, R$. The wave speeds s^- and s^+ are estimated according to Batten et al. [36] from

$$s^- = \min(s^L, 0), \qquad s^+ = \max(s^R, 0), \tag{6}$$

with

$$s^L = \min(\bar{u} - \bar{c}, u^L - c^L), \qquad s^R = \max(\bar{u} + \bar{c}, u^R + c^R). \tag{7}$$

$\bar{u}$ and $\bar{c}$ are determined from arithmetic averages $\bar{u} = \frac{1}{2}\left(u^L + u^R\right)$ and $\bar{c} = \frac{1}{2}\left(c^L + c^R\right)$. Alternatively, Roe averaging can be applied to determine $\bar{u}$ and $\bar{c}$, both procedures achieve nearly identical numerical results [39]. Following Einfeldt et al. [40], the contact wave speed is obtained from

$$s^* = \frac{p^R - p^L + \rho^L u^L\left(s^L - u^L\right) - \rho^R u^R\left(s^R - u^R\right)}{\rho^L\left(s^L - u^L\right) - \rho^R\left(s^R - u^R\right)}. \tag{8}$$

3.2. High-order spatial reconstruction with application of local characteristic decomposition

At the beginning of each time step, only cell-averaged conservative variables $\bar{\mathbf{U}}$ are available. For the application of the HLLC procedure as described above, the values at cell faces have to be reconstructed. Different higher-order spatial interpolations, such as WENO and TENO schemes [10,18,21], are available in the literature for this purpose. Note that if the reconstruction is performed directly on conservative variables, spurious oscillations may be introduced in the vicinity of discontinuities due to interactions of different characteristic fields [41]. The issue becomes even more relevant when the order of reconstruction is increased, and essentially non-oscillatory behavior can no longer be guaranteed. As a remedy, all cell-averaged conservative variables within the stencil range are locally projected onto characteristic space and the reconstruction of the cell-face variables is performed with these characteristic variables $\bar{\mathbf{W}}$. Afterwards, the reconstructed characteristic variables at the cell faces are projected back onto physical space and passed to the HLLC solver. This procedure is computationally costly and also challenging in terms of floating-point consistency. Nevertheless, the decomposition is required whenever spurious oscillations cannot be tolerated or positivity of density and pressure is violated.

Here, we present the characteristic decomposition in x-direction in detail. Since the flux is computed dimension-by-dimension, the procedure in y- and z-directions is analogous. All conservative variables within the stencil range are projected onto characteristic space by

$$\bar{\mathbf{W}} = \mathbf{L}\cdot\bar{\mathbf{U}} = \mathbf{L}_1\rho + \mathbf{L}_2\rho u + \mathbf{L}_3\rho v + \mathbf{L}_4\rho w + \mathbf{L}_5 E \tag{9}$$

with $\mathbf{L} = (\mathbf{L}_1, \mathbf{L}_2, \mathbf{L}_3, \mathbf{L}_4, \mathbf{L}_5)$ being the left eigenvector matrix of the linearized system

$$\mathbf{L} = \begin{pmatrix} \frac{1}{4}\frac{q^2}{\hat{H}-\frac{1}{2}q^2} + \frac{1}{2}\frac{\hat{u}}{c} & \frac{1}{2}\frac{\hat{u}}{\hat{H}-\frac{1}{2}q^2} - \frac{1}{2c} & -\frac{1}{2}\frac{\hat{v}}{\hat{H}-\frac{1}{2}q^2} & -\frac{1}{2}\frac{\hat{w}}{\hat{H}-\frac{1}{2}q^2} & \frac{1}{2}\frac{1}{\hat{H}-\frac{1}{2}q^2} \\ -q^2 + \hat{H} & \hat{u} & \hat{v} & \hat{w} & -1 \\ \hat{v} & 0 & -1 & 0 & 0 \\ -\hat{w} & 0 & 0 & 1 & 0 \\ \frac{1}{4}\frac{q^2}{\hat{H}-\frac{1}{2}q^2} - \frac{1}{2}\frac{\hat{u}}{c} & \frac{1}{2}\frac{\hat{u}}{\hat{H}-\frac{1}{2}q^2} + \frac{1}{2c} & -\frac{1}{2}\frac{\hat{v}}{\hat{H}-\frac{1}{2}q^2} & -\frac{1}{2}\frac{\hat{w}}{\hat{H}-\frac{1}{2}q^2} & \frac{1}{2}\frac{1}{\hat{H}-\frac{1}{2}q^2} \end{pmatrix} \tag{10}$$

with $q^2 = \hat{u}^2 + \hat{v}^2 + \hat{w}^2$ and enthalpy H. Note that quantities with hat-notation "$\hat{\cdot}$" denote density-based Roe averages of the cell-averaged variables of the cells adjacent to the face.

After reconstruction, the resulting characteristic cell-face variables $\bar{\mathbf{W}}_{recon}$ are projected back onto physical space by

$$\bar{\mathbf{U}} = \mathbf{R}\cdot\bar{\mathbf{W}}_{recon} = \mathbf{R}_1 w_1 + \mathbf{R}_2 w_2 + \mathbf{R}_3 w_3 + \mathbf{R}_4 w_4 + \mathbf{R}_5 w_5, \tag{11}$$

where $\mathbf{R} = (\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3, \mathbf{R}_4, \mathbf{R}_5)$ denotes the right eigenvector matrix of the linearized system and is given by

$$\boldsymbol{R} = \begin{pmatrix} 1 & \frac{1}{\hat{H}-\frac{1}{2}q^2} & 0 & 0 & 1 \\ \hat{u} - \hat{c} & \frac{\hat{u}}{\hat{H}-\frac{1}{2}q^2} & 0 & 0 & \hat{u} + \hat{c} \\ \hat{v} & \frac{\hat{v}}{\hat{H}-\frac{1}{2}q^2} & -1 & 0 & \hat{v} \\ \hat{w} & \frac{\hat{w}}{\hat{H}-\frac{1}{2}q^2} & 0 & 1 & \hat{w} \\ \hat{H} - \hat{c}\hat{u} & \frac{\hat{H}}{\hat{H}-\frac{1}{2}q^2} - 1 & -\hat{v} & \hat{w} & \hat{H} + \hat{c}\hat{u} \end{pmatrix}. \tag{12}$$

4. Mechanisms of numerical symmetry breaking

In the following, we show how nominally symmetric problem configurations may result in asymmetric flow evolution due to floating-point errors in various steps of the numerical algorithm. For simplicity, we first consider two spatial dimensions. Strategies for three-dimensional problems, which reveal additional challenges, are presented afterwards.

We have chosen the Rayleigh-Taylor instability and a gas implosion problem as test cases given their sensitivity to small disturbances.

The Rayleigh-Taylor instability is a common test case for accuracy studies of numerical schemes [5,21,22,24], since it contains both discontinuities and complex flow structures. Two initial gas layers with different densities are exposed to gravity with unity magnitude, where the resulting acceleration is directed towards the lighter fluid. A small disturbance of the contact line triggers the instability. The set-up is chosen according to [5], where the computational domain is $[0,\ 0.25] \times [0,\ 1]$ and the interface is placed at $y = 0.5$. Initial states are given by $(\rho, u, v, p)_{y \le 0.5} = (2, 0, -0.025c\cdot\cos(8\pi x), 2y + 1)$ and $(\rho, u, v, p)_{y > 0.5} = (1, 0, -0.025c\cdot\cos(8\pi x), y + 1.5)$, where the speed of sound is $c = \sqrt{\frac{\gamma p}{\rho}}$ with $\gamma = \frac{5}{3}$. Top and bottom boundary states are fixed to (1, 0, 0, 2.5) and (2, 0, 0, 1), respectively. Symmetry boundary conditions are imposed at the left and right boundary.

The second test case describes a two dimensional implosion of a gas in a square box of size $L = 0.6$ as described by Liska and Wendroff [42]. The low-density core region forms another square box, yet its size is smaller and it is centered but rotated

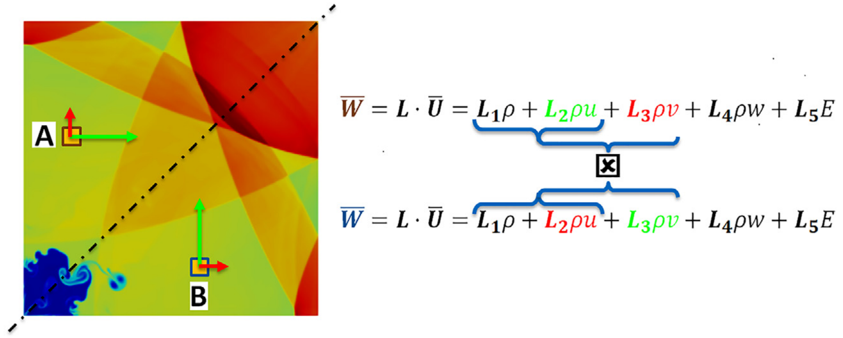

A
B
$\overline{W} = L \cdot \overline{U} = \underbrace{\underbrace{L_1\rho + L_2\rho u} + L_3\rho v} + L_4\rho w + L_5 E$
$\boxed{\times}$
$\overline{W} = L \cdot \overline{U} = \overbrace{L_1\rho + \overbrace{L_2\rho u + L_3\rho v}} + L_4\rho w + L_5 E$

Fig. 1. Illustration of the asymmetric floating-point evaluation of Eq. (9) for two symmetrically placed cells.

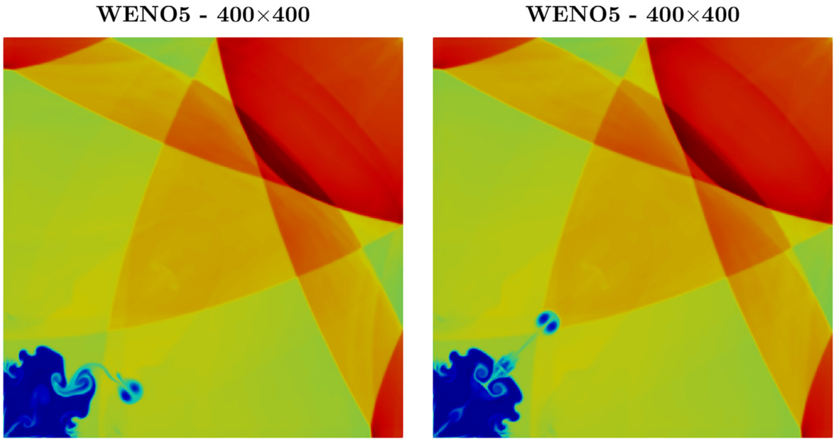


Fig. 2. Implosion test case $t = 1.0$: density contours from 0.37 (blue) to 1.18 (red); (left) straightforward implementation without sorting during characteristic decomposition; (right) adapted order of evaluations during characteristic decomposition. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

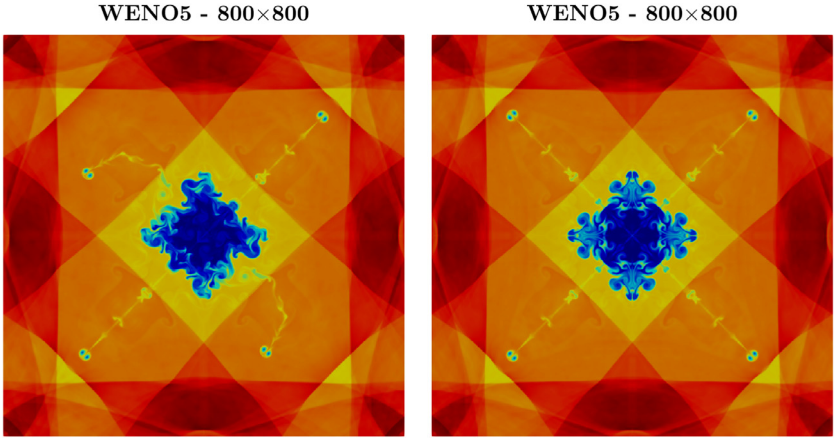


Fig. 3. Implosion test case $t = 2.5$: density contours from 0.4 (blue) to 1.05 (red); (left) implementation without brackets for contact wave speed calculation; (right) implementation with brackets for contact wave speed calculation. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

by 45º. The corner points of the inner square box are located at $(\pm 0.15, 0)$ and $(0, \pm 0.15)$. Symmetry boundary conditions are applied at all boundaries. Initially, both fluids are at rest with the initial states $(\rho, u, v, p)_{core} = (0.125, 0, 0, 0.14)$ and $(\rho, u, v, p)_{outer} = (1.0, 0, 0, 1.0)$ and $\gamma = 1.4$. For simplicity, this problem is usually studied by employing the two axial symmetries. The reduced problem still reveals an inherent symmetry along the diagonal axis. The evolution of the jet developing along this axis is very sensitive to small disturbances and can be easily deflected, especially for long-time simulations up to $t = 2.5$. Once the symmetry is slightly perturbed, frequent shock-wave interactions lead to a strong increase of asymmetry.

4.1. Lack of associativity

Standard requirements of modern programming languages ensure that summation and multiplication are commutative for two components, i.e. the result given in floating-point notation equals the correctly rounded analytic solution. However, if more than two components are summed up or multiplied, the final floating-point value typically depends on the order of operations, i.e. associativity is lost

$$(a + b) + c \neq a + (b + c). \tag{13}$$

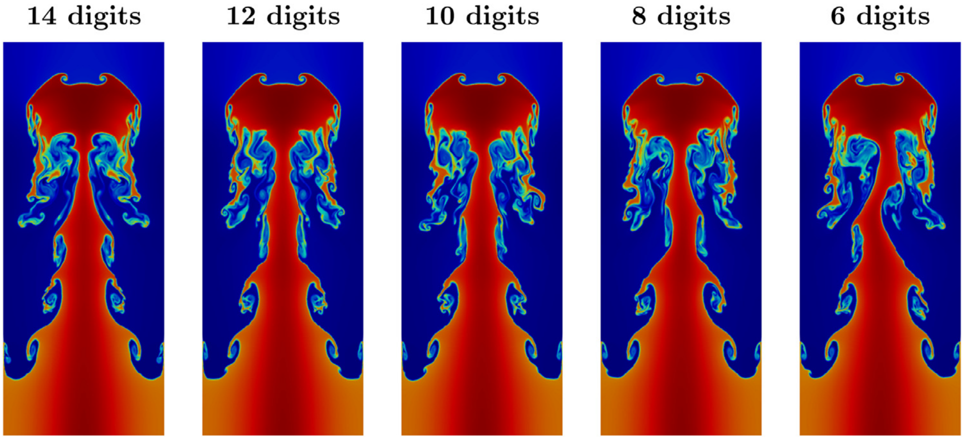


Fig. 4. Rayleigh-Taylor instability $t = 1.95$: density contours from 0.85 (blue) to 2.25 (red); symmetry breaking due to limited representation of π to the given number of digits with a resolution of 256 × 1024 using a TENO5 stencil. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

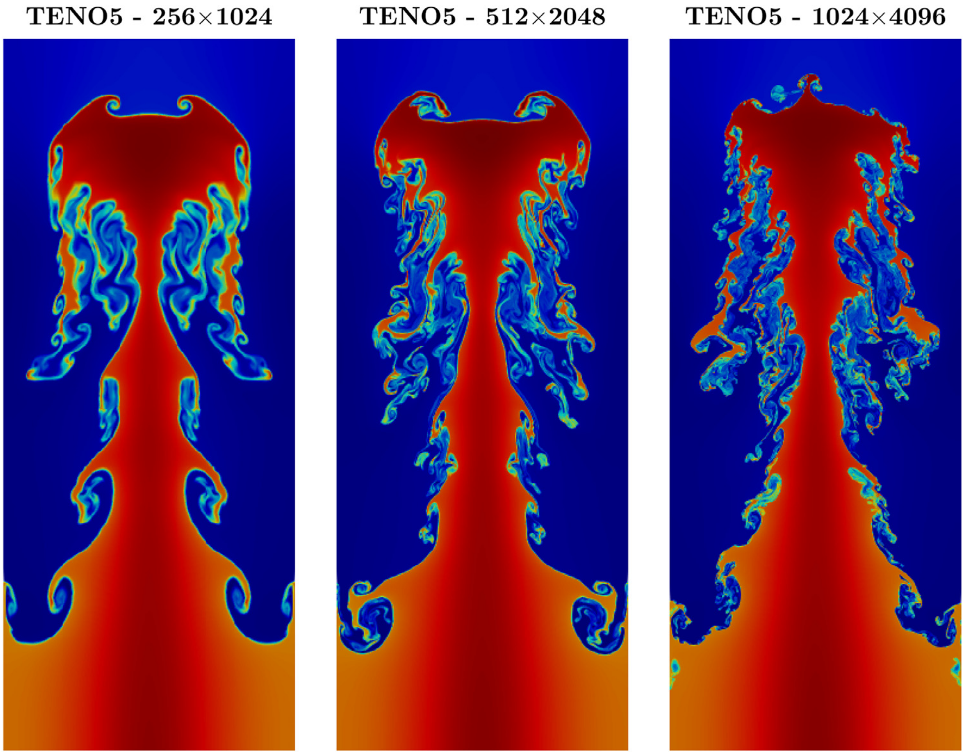


Fig. 5. Rayleigh-Taylor instability $t = 1.95$: density contours from 0.85 (blue) to 2.25 (red); symmetry breaking due to cosine evaluation inaccuracy. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Numerical algorithms build upon a series of such elementary operations. Accordingly, effects of truncation errors cannot be suppressed completely, however, we found that only few recurring algorithmic patterns trigger asymmetric flow behavior that leads to macroscopic flow deviations. These patterns occur essentially for all employed numerical discretization schemes.

The simulation of the implosion test case using low-dissipative schemes is an illustrative example for symmetry breaking along the diagonal line, see Fig. 1. The effect that triggers this asymmetry can be traced back to operations where directionally dependent quantities are subject to summation or multiplication operations, e.g. for the velocity components. Considering two isolated cells symmetric to the diagonal line, see Fig. 1, and a perfectly symmetric flow at this time instant, the velocity component in x-direction of the cell A equals the velocity component in y-direction of cell B and vice versa. Together with the lack of associativity, Eq. (13), the characteristic projection in Eq. (9) is prone to floating-point round-off errors, and small-scale disturbances are triggered. These disturbances are amplified by physical instability mechanisms of the inviscid flow (e.g. Kelvin-Helmholtz instabilities) and deflect the diagonal jet in the given example. Aiming for preserving symmetry we have to ensure that the algorithmic pattern is dimensionally invariant. Such patterns with explicit directional dependencies together with more than two components occur with the characteristic decomposition in high-resolution schemes.

One straightforward solution to handle the issue is to enforce the order of the evaluation of matrix multiplications by inserting appropriate brackets into the source code. This will force the compiler to handle the operations consistently for any cell. In 2D, Eq. (9) becomes

$$\bar{\mathbf{W}} = \mathbf{L} \cdot \bar{\mathbf{U}} = \mathbf{L}_1 \rho + (\mathbf{L}_2 \rho u + \mathbf{L}_3 \rho v) + \mathbf{L}_4 E. \tag{14}$$

The actual implementation, however, is cumbersome as Eq. (14) is part of a matrix multiplication that involves loops.

Alternatively, we can order the evaluation of Eq. (14) such that the sum of momenta is computed first, and the other terms are added subsequently, i.e.

$$\bar{\mathbf{W}} = \mathbf{L} \cdot \bar{\mathbf{U}} = (\mathbf{L}_2 \rho u + \mathbf{L}_3 \rho v) + \mathbf{L}_1 \rho + \mathbf{L}_4 E. \tag{15}$$

The same procedure is applied to Eq. (11).

When the directional dependence of the characteristic projection is eliminated, the expected perfectly diagonal post-implosion jet is observed, see Fig. 2.

A second example is shown to clarify the symmetry-breaking mechanism for a different part of the solver. The same implosion problem is now calculated on the full domain. The loss of symmetry in the second and fourth quadrant, as shown in the left frame of Fig. 3, is caused by an inconsistent calculation of the wave-speed s^* in the HLLC solver. By enforcing proper summations using additional brackets, Eq. (8) becomes

$$s^* = \frac{\left(p^R - p^L\right) + \left[\rho^L u^L \left(s^L - u^L\right) - \rho^R u^R \left(s^R - u^R\right)\right]}{\rho^L \left(s^L - u^L\right) - \rho^R \left(s^R - u^R\right)}. \tag{16}$$

Remarkably, this marginal modification removes these artifacts, see the right frame of Fig. 3. The only difference between both simulations is the introduction of brackets in one single line of code.

Note, that the driving mechanism behind symmetry breaking in both cases is the lack of associativity. Despite the fact that all results in this paper are demonstrated for one specific type of flow solver (HLLC) for simplicity, we emphasize that the underlying principles and derived best-practice guidelines are not limited to these methods, but apply for high-resolution schemes for flow solvers in general.

4.2. Generalized solution strategy

The simple workarounds that were presented in detail in the previous section demonstrate the fundamental mechanisms, however, they are only applicable to two-dimensional situations. They do not work for more complex problems, especially not for three dimensions. This can be demonstrated exemplarily for the calculation of the velocity magnitude q. Since three components have to be summed, three different floating-point results

$$q^2 = \begin{cases} (v^2 + w^2) + u^2, \\ (w^2 + u^2) + v^2, \\ (u^2 + v^2) + w^2 \end{cases} \tag{17}$$

can be obtained. The same problem arises in several other parts of the algorithm, where subterms related to the three Cartesian directions have to be summed up, e.g. in Eq. (3) the fluxes in x, y, and z-direction. This summation of direction-related terms can be found for a wide range of other algorithms in CFD codes including curvature calculations, multiresolution approaches and level-set methods.

For certain operations, where the preferential direction is known a priori, the simple bracketing strategy is applicable such as

$$q^2 = \begin{cases} (v^2 + w^2) + u^2 & \text{for flux calculation in x-direction,} \\ (w^2 + u^2) + v^2 & \text{for flux calculation in y-direction,} \\ (u^2 + v^2) + w^2 & \text{for flux calculation in z-direction.} \end{cases} \tag{18}$$

However, generalization of this procedure under relaxed conditions is not possible. Instead, we propose to use a consistency-ensuring function for all relevant summations in the code. Consistency means in this context that the result of the summation is identical for all cases of Eq. (17). Thus, the consistency-ensuring summation restores associativity for floating-point numbers. This tailored summation takes the sum of three floating-point values a, b and c according to

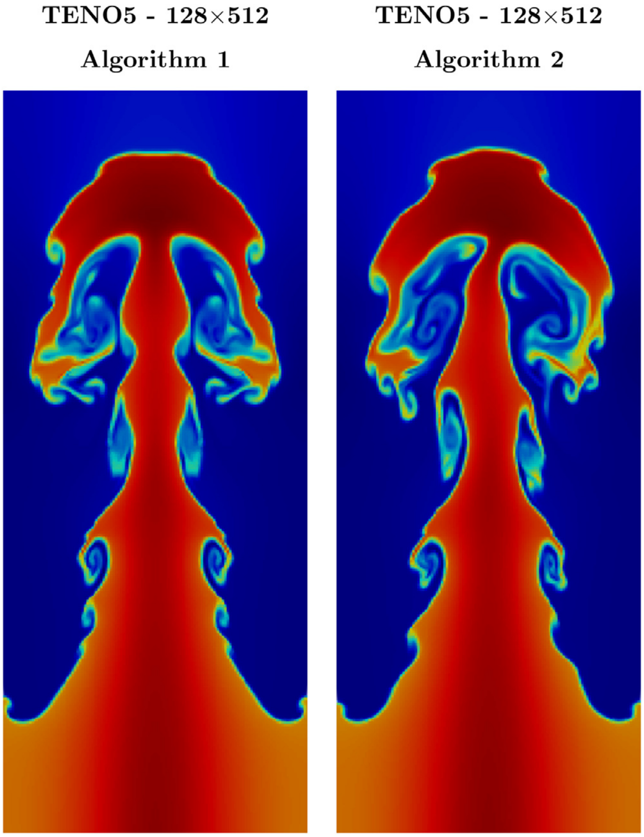


Fig. 6. Rayleigh-Taylor instability $t = 1.95$: density contours from 0.85 (blue) to 2.25 (red); (left) straightforward implementation of Roe-averaging procedure; (right) efficient implementation of Roe-averaging procedure. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$SUM_{consistent} = \frac{1}{2}\left(\max(s_1, s_2, s_3) + \min(s_1, s_2, s_3)\right) \tag{19}$$

with

$$\begin{aligned} s_1 &= (a + b) + c, \\ s_2 &= (c + a) + b, \\ s_3 &= (b + c) + a, \end{aligned} \tag{20}$$

and four floating-point values a, b, c and d with

$$\begin{aligned} s_1 &= (a + b) + (c + d), \\ s_2 &= (c + a) + (b + d), \\ s_3 &= (b + c) + (a + d). \end{aligned} \tag{21}$$

Note, this procedure does not necessarily provide the floating-point representation of the exact analytical result, but it delivers a consistent result that is independent of the sequence of terms. Hence, the evaluation of the sum becomes directionally independent. Extension to higher numbers of summands is straightforward, yet in our experience not required for common applications. An alternative, well-known approach to avoid errors in the sum of numbers with different order of magnitude is to sort the values before adding them up. In principle, this would also work here, however, we compared the run time of both procedures and found our solution to be around 30 times faster than the application of “std::sort” in C++. The overall performance impact of the

```
//compute density-based Roe averages according to Eq. (24)
double sqrt_rho_i   = std::sqrt(rho_i);
double sqrt_rho_ip1 = std::sqrt(rho_ip1);
double divisor      = 1.0 / ( sqrt_rho_i + sqrt_rho_ip1 );

double u_roe_avg = ((u_i * sqrt_rho_i) + (u_ip1 * sqrt_rho_ip1))
                    * divisor;
double v_roe_avg = ((v_i * sqrt_rho_i) + (v_ip1 * sqrt_rho_ip1))
                    * divisor;
double w_roe_avg = ((w_i * sqrt_rho_i) + (w_ip1 * sqrt_rho_ip1))
                    * divisor;
double H_roe_avg = ((H_i * sqrt_rho_i) + (H_ip1 * sqrt_rho_ip1))
                    * divisor;
```

C++ Code Example 1. A straightforward algorithm of the Roe-averaging procedure.

```
//compute density-based Roe averages according to Eq. (25)
double sqrt_rho     = std::sqrt(rho_ip1 * one_rho_i);
double divisor      = 1.0 / ( sqrt_rho + 1.0 );

double u_roe_avg = (u_i + (u_ip1 * sqrt_rho)) * divisor;
double v_roe_avg = (v_i + (v_ip1 * sqrt_rho)) * divisor;
double w_roe_avg = (w_i + (w_ip1 * sqrt_rho)) * divisor;
double H_roe_avg = (H_i + (H_ip1 * sqrt_rho)) * divisor;
```

C++ Code Example 2. An efficient algorithm of the Roe-averaging procedure.

consistency-managed sums for the simulations shown in this paper is never higher than 10%.

4.3. *Influence of initial disturbance*

Although it seems obvious that perturbations in the initial condition can affect the simulation result, here we want to raise awareness of unintentionally initialized asymmetries.

Consider the classical Rayleigh-Taylor instability where the velocity field is perturbed according to

$$v = -0.025c \cdot \cos(8\pi x). \tag{22}$$

We want to highlight the effect of the accuracy of π when evaluating Eq. (22). In Fig. 4 we show the Rayleigh-Taylor instability at $t = 1.95$ for a resolution of 256×1024 and varying accuracy of the numerical representation of π. Only from 12 digits, the error becomes negligible and the Rayleigh-Taylor instability seems symmetric. Another well-known effect is an inaccurate cosine evaluation with C++ routines, shown in Fig. 5. The fact that $cos(\pi - \epsilon) \neq \cos(\pi + \epsilon)$ is known in the computer science community, yet little attention is payed to its consequences in computational fluid dynamics. Again, the symmetry breaking in Fig. 5 is simply induced by the violation of cosine-symmetry and could lead to wrong interpretations with respect to the quality of numerical schemes. As a remedy, we enforced symmetric initial conditions by

$$v = \begin{cases} -0.025c \cdot \cos(8\pi x), & \text{if } x < 0.125, \\ -0.025c \cdot \cos\left[8\pi\,(0.25 - x)\right], & \text{otherwise} \end{cases} \tag{23}$$

and observed perfectly symmetric results even for unprecedentedly large resolutions for the Rayleigh-Taylor instability as compared to literature (see next section).

4.4. *Caveats of performance optimization*

In this last example, we demonstrate the effect of code optimizations with emphasis on runtime minimization. Given a low-dissipative scheme with careful implementation following our previous guideline, round-off errors can still be triggered from code optimization efforts.

We consider a scheme which involves the density-based Roe-averaging procedures as required for the characteristic decomposition, see Eqs. (10) and (12). Following Roe [33], the average is defined as

$$\hat{k} = \frac{\bar{k}_i \cdot \sqrt{\rho_i} + \bar{k}_{i+1} \cdot \sqrt{\rho_{i+1}}}{\sqrt{\rho_i} + \sqrt{\rho_{i+1}}}, \tag{24}$$

where k is replaced by the averaged quantity of interest, e.g. u, v, w or H. A straightforward algorithm is given in Example 1.

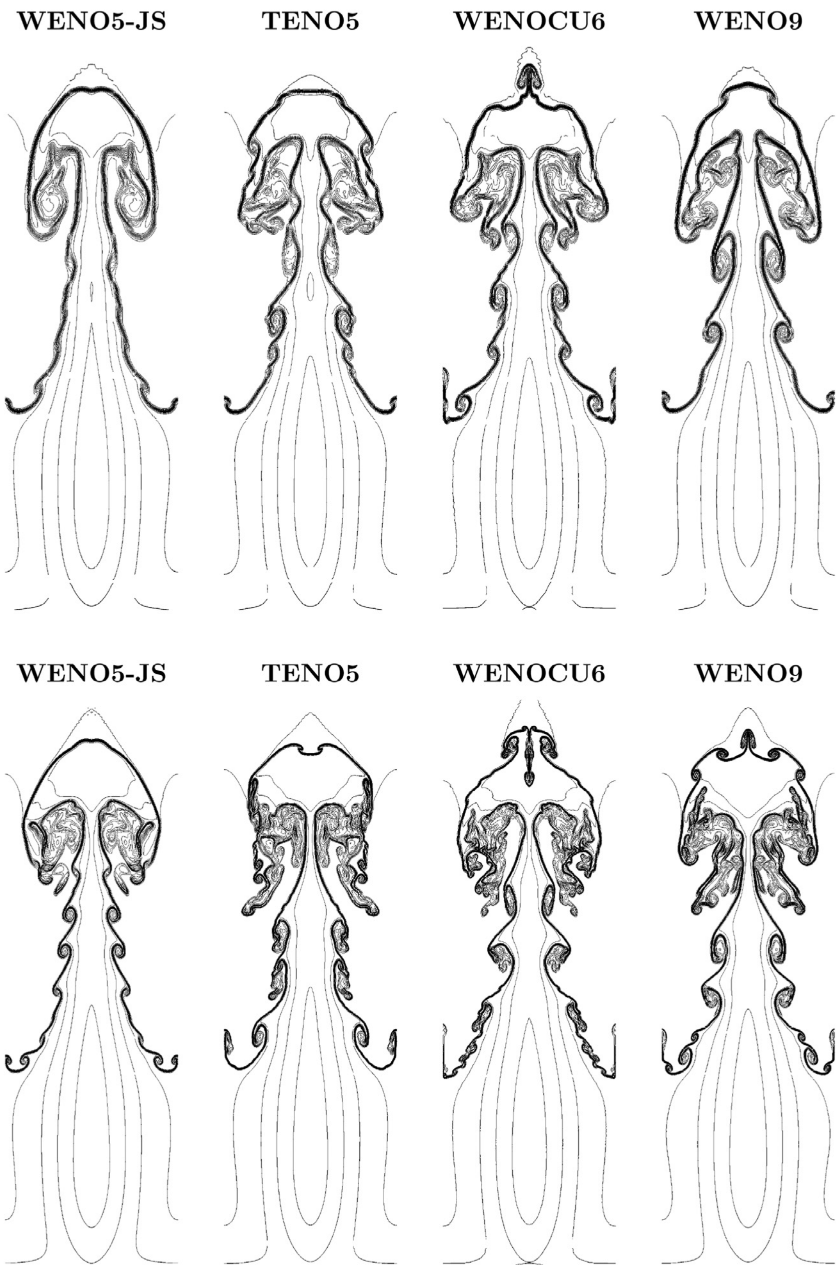


Fig. 7. Rayleigh-Taylor instability: density contours; (top) 128 × 512; (bottom) 256 × 1024.

From a computational efficiency point of view, the operation count of this implementation can be reduced. The evaluation of

$$\hat{k} = \frac{\bar{k}_i + \bar{k}_{i+1} \cdot \sqrt{\frac{\rho_{i+1}}{\rho_i}}}{\sqrt{\frac{\rho_{i+1}}{\rho_i}} + 1} \tag{25}$$

is analytically identical to Eq. (24), yet computationally preferable for the saving of one costly root function evaluation. However, this popular rearrangement causes asymmetry as demonstrated in the following. The optimized C++ code is shown in Example 2.

In Fig. 6, we show two snapshots of the density contours from a Rayleigh-Taylor instability simulation using the implementation of Eq. (24) (left figure) and Eq. (25) (right figure), respectively. Both simulations are performed with TENO5 ($C_T = 10^{-5}$) using 128 × 512 cells and the density is shown at $t = 1.95$. Obviously, the optimized algorithm strongly affects the quality of the results and initiates symmetry breaking. Successive mathematical operations, especially within the costly root function, lead to a fast amplification of floating-point errors that affects the overall flow evolution even for moderate resolutions.

5. High resolution examples and discussion

We have reported several strategies to improve the solver implementations for the Euler equations with symmetry preserving property. In this section, we demonstrate their efficiency for modern low-dissipative numerical schemes together with large spatial resolutions. The Rayleigh-Taylor instability problem, the implosion problem and a classical Riemann problem are computed

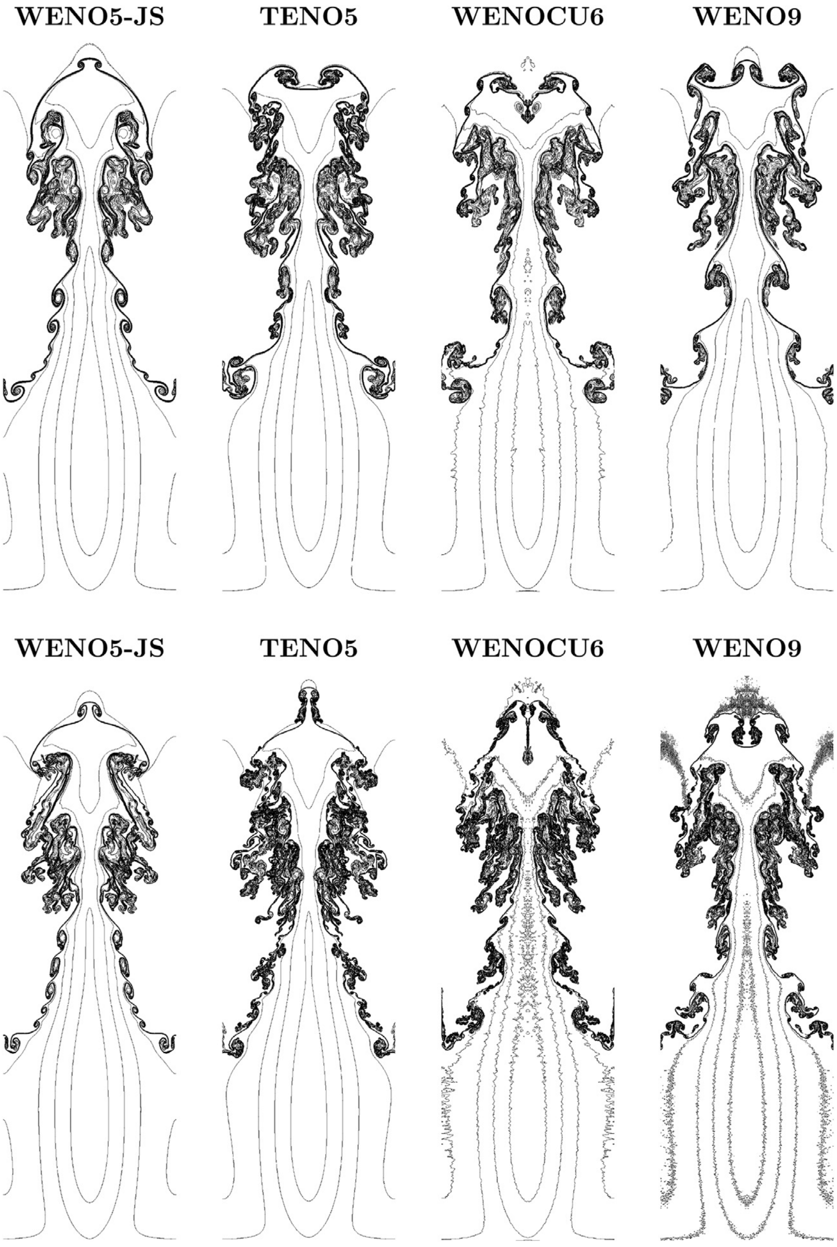


Fig. 8. Rayleigh-Taylor instability $t = 1.95$: density contours; (top) 512 × 2048; (bottom) 1024 × 4096.

in two-dimensions with unprecedented resolution and different high-order stencils. The results are compared to literature data and show very good agreement. Finally, we present a new three-dimensional implosion test case, which is directly obtained by the extension of the two-dimensional case of Liska and Wendroff [42]. The high number of inherent symmetries together with the high number of wave interactions makes this case particularly interesting and challenging to study the directional independence of the framework in three dimensions.

The inviscid Euler equations are formally scale-free due to the infinite Reynolds number, thus numerical schemes with vanishing dissipation predict an infinite cascade evolution of small structures. Numerical dissipation retards small disturbance amplification, thus a delayed evolution of asymmetries is observed. For this reason, often in literature the onset of an asymmetric Rayleigh-Taylor instability evolution is used as an indicator for the amount of artificial numerical dissipation [5,21,22]. In this work, we demonstrate that numerical dissipation does basically determine the growth rate of either initially present or algorithmically induced asymmetric disturbances but does not trigger them itself. In other words, visible symmetry breaking is an indicator of insufficiently damped artificial disturbances rather than the result of some underlying physics that is observable due to higher-order methods.

The CFL number is set to 0.6 for all test cases. Calculations were performed on the SuperMUC Petascale System using CPUs of the type Xeon E5-2697 v3 (Haswell). The C++ source code is compiled using the Intel compiler (ICC) version 16.0.4.

5.1. Rayleigh-Taylor instability

We start evaluating the Rayleigh-Taylor instability test case, which is already described in Section 4. This test case is widely

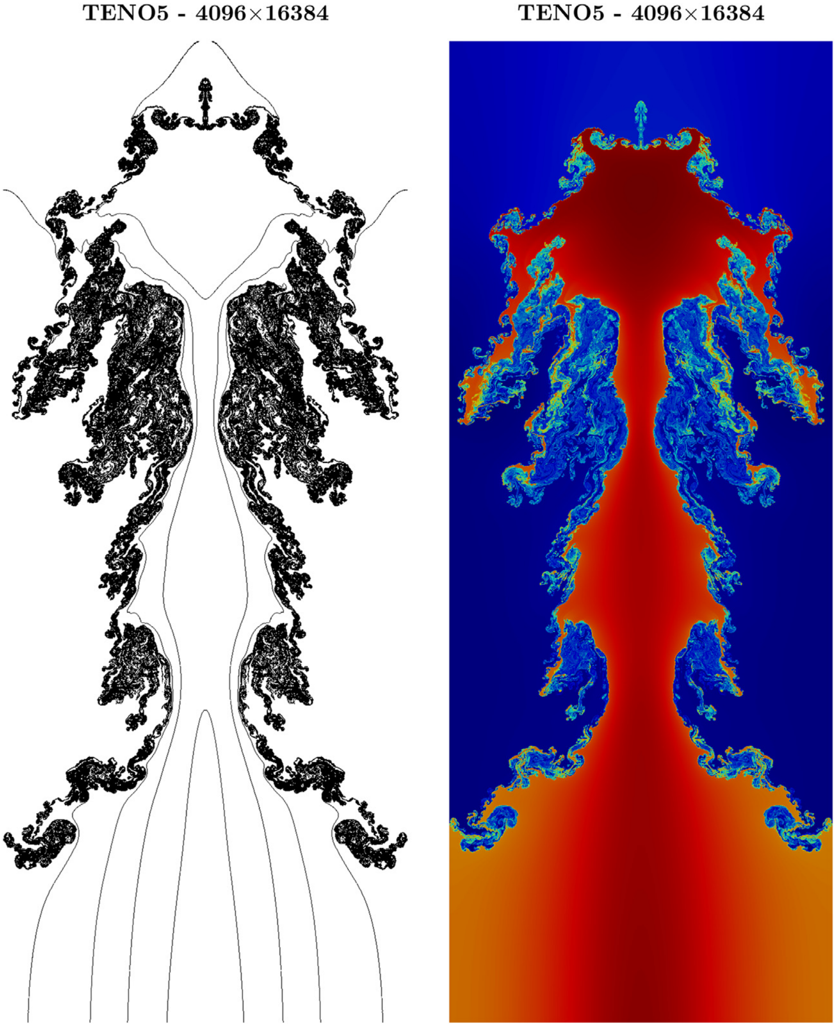


Fig. 9. Rayleigh-Taylor instability $t = 1.95$: (left) density contours; (right) color density map (blue= 0.85 to red= 2.25). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

used for verification of numerical methods. The resolution and richness of vortical structures of this test case can be utilized as a measure of the numerical dissipation introduced by the method. The loss of symmetry in this test case is an effect of reduced numerical dissipation as shown by different authors [5,21,22,24,26]. Remacle et al. [24] supposed that the only cause of the asymmetric behavior of the flow might be due to roundoff errors, however, they did not achieve symmetric results for higher resolutions.

We simulated the Rayleigh-Taylor instability with increasingly high resolution applying four different high-order stencils, namely the WENO5-JS [10], the TENO5 ($C_T = 10^{-5}$) [21], the WENOCU6 [43] and the WENO9 [18] stencil. The resulting density contours are shown in Figs. 7 and 8 for four different resolutions 128×512, 256×1024, 512×2048 and 1024×4096. 15 contour lines are equally spaced from $\rho = 0.952269$ to $\rho = 2.14589$ according to [5]. The final simulation time is $t = 1.95$. Unlike in Fig. 2 in [5], Fig. 20 in [21] and Fig. 20 in [22], where asymmetric results are presented for high-order methods, the symmetry is exactly maintained up to floating-point precision. To our best knowledge, there is no result of the Rayleigh-Taylor instability reported in literature for resolutions beyond 512×2048.

Fig. 9 shows the Rayleigh-Taylor instability at $t = 1.95$ for an extreme resolution of $4096 \times 16,384$ using the TENO5 scheme. Symmetry is preserved quantitatively up to floating-point precision. Thus, the consistent floating-point arithmetic in our framework enables the usage of high-order methods combined with highly resolved meshes without any numerical symmetry breaking.

5.2. 2D implosion test case

The two-dimensional implosion test case of Liska and Wendroff [42] is a challenging test case due to the presence of non-grid aligned shocks and the appearance of axisymmetric jets, see Section 4 for a more detailed problem description. This test case is often used for code validation in the astrophysics community [29,44]. Sutherland [29] reports the problem of floating-point inaccuracies and handles this by a controlled rounding procedure with a non-physical threshold. Fluctuations smaller than this limit are regarded as floating-point noise and intentionally eliminated. However, following the implementation principles of Section 4, we achieve strong symmetry preservation without the usage of any additional error-cancellation procedure, and furthermore we are able to maintain the full precision range. In [29,42,44] only the first quadrant of the implosion problem is simulated. The symmetry around the diagonal of the first quadrant does not necessarily ensure the symmetry around the diagonals of all other quadrants

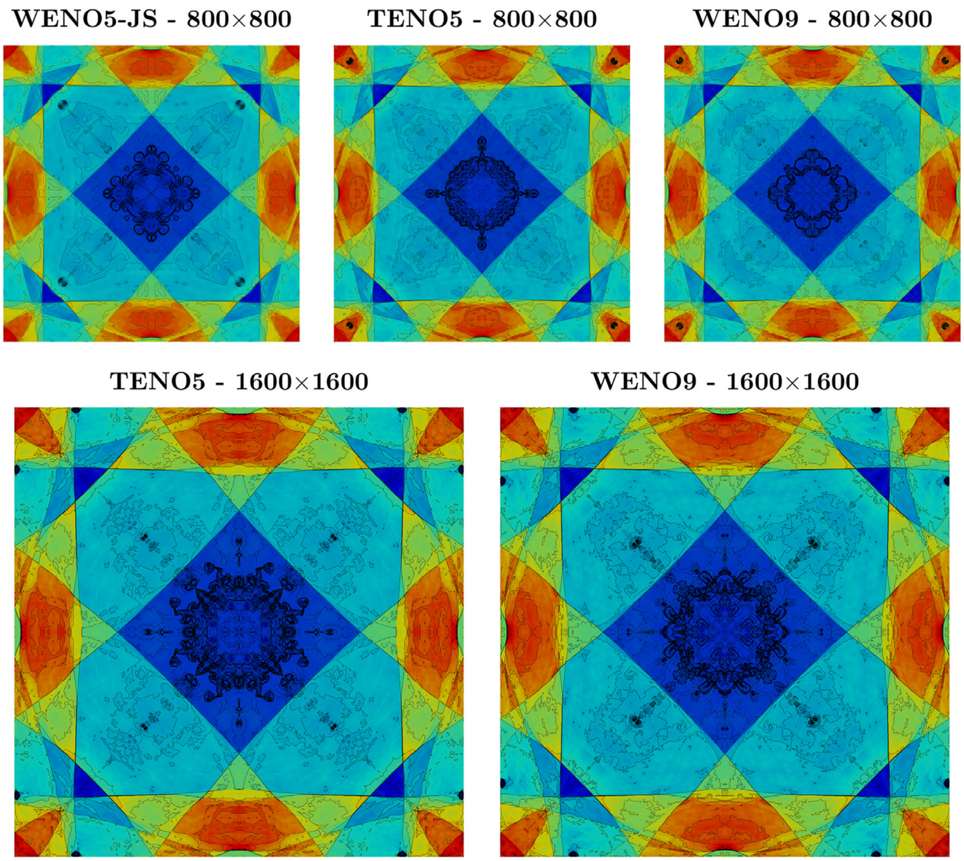


Fig. 10. Implosion $t = 2.5$: color pressure map (blue= 0.7 to red= 1.15) is overlaid by 31 density contours (0.35 to 1.1). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

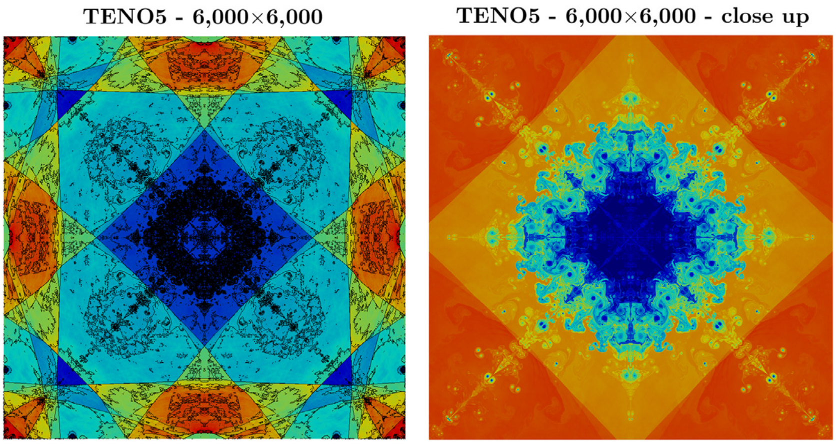
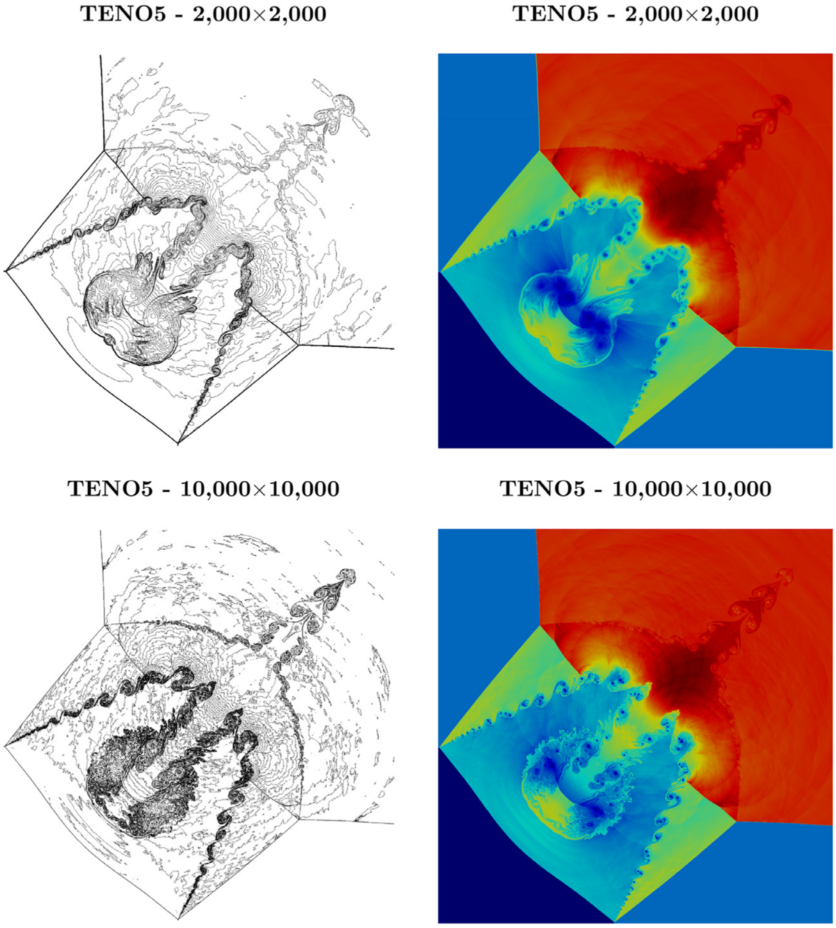


Fig. 11. Implosion $t = 2.5$: (left) color pressure map (blue= 0.7 to red= 1.15) is overlaid by 31 density contours (0.35 to 1.1); (right) color density map of the inner region $[0.15, 0.45] \times [0.15, 0.45]$ (blue= 0.4 to red= 1.0) . (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

as shown in Fig. 3. Thus, here we simulated the full problem domain.

Results for the final pressure and density distribution are given in Fig. 10. The lowest presented resolution is chosen according to [42] and delivers comparable results for the WENO5-JS stencil. For higher resolutions, the jet moves further in diagonal direction and reaches the end of the domain, where it is then split into two parts. Thus, the progress of the jet can be used as a measure of the magnitude of numerical dissipation. The TENO5 stencil is able to achieve a result close to the one of the WENO9 stencil. Despite the smaller size and reduced computational cost of the TENO5 stencil as compared to the WENO9 stencil, the level of numerical dissipation is similar. Thus, it is more efficient to use the TENO5 stencil to achieve a desired level of numerical dissipation.

Again, the robustness of the implementation is tested by further increasing the resolution up to $36 \cdot 10^6$ cells. The resulting pressure and density distribution is shown in Fig. 11. Due to the absence of any physical viscosity, no convergence can be expected and the



TENO5 - 2,000×2,000

TENO5 - 2,000×2,000

TENO5 - 10,000×10,000

TENO5 - 10,000×10,000

Fig. 12. 2D Riemann case at $t = 1.1$ in the region $[0, 1.2] \times [0, 1.2]$: (left) 32 density contour levels from 0.15 to 1.7; (right) color density map (blue= 0.14 to red= 1.75). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

reduction of the numerical dissipation with increasing resolution leads to finer and finer structures. All symmetries are quantitatively maintained to floating-point precision at the end of the simulation.

5.3. 2D Riemann case

Schulz-Rinne et al. [45] proposed a series of two-dimensional Riemann test cases with systematic permutations of shockwaves and expansion waves running in x- and y-direction, respectively. Here, we focus on the four-shock setup that was already studied by several groups [28,29,42]. However, similarly to Balsara et al. [28], we modify the domain size to $[0, 2] \times [0, 2]$ and simulate the problem until $t = 1.1$. This is advantageous since the relevant region now covers more than one quarter of the full domain at the end of the simulation. In literature, the initial quantities are rounded after five digits, resulting in measurable artifacts of the complex wave system. Here, we apply the exact Rankine-Hugoniot jump relations to obtain

$$\begin{array}{llllll}
\rho = 1.5, & u = 0, & v = 0, & P = 1.5, & \text{for } x > 1, & y > 1, \\
\rho = \frac{33}{62}, & u = \frac{4}{\sqrt{11}}, & v = 0, & P = .3, & \text{for } x < 1, & y > 1, \\
\rho = \frac{77}{558}, & u = \frac{4}{\sqrt{11}}, & v = \frac{4}{\sqrt{11}}, & P = \frac{9}{310}, & \text{for } x < 1, & y < 1, \\
\rho = \frac{33}{62}, & u = 0, & v = \frac{4}{\sqrt{11}}, & P = .3, & \text{for } x > 1, & y < 1.
\end{array} \tag{26}$$

For this test case, a cutoff value of $C_T = 10^{-4}$ was used for the TENO5 stencil.

The resulting density field (contours and color maps) is shown in Fig. 12 for a section of the computational domain. The two resolutions are chosen based on the reference configuration, see Fig. 11 of [28], and as example of an extreme resolution with $100 \cdot 10^6$ cells. Contrary to [28], here the diagonal symmetry of the guitar-like shape of the jet is completely maintained. The same is also shown for an extreme resolution together with the expected increased richness of the fine structures.

5.4. 3D implosion test case

Finally, we tested the symmetry preserving property of the framework in three dimensions. For this purpose, the two-dimensional test case of Liska and Wendroff [42] (Section 5.2) is extended to three dimensions in a straightforward way using an octahedral-shaped low-pressure area at the center of a cubical domain of size $[0, 0.6] \times [0, 0.6] \times [0, 0.6]$. The six corner points of the octahedron are placed at $(0.3 \pm 0.15, 0.3, 0.3)$, $(0.3, 0.3 \pm 0.15, 0.3)$ and $(0.3, 0.3, 0.3 \pm 0.15)$. With this setup, each of the eight octants of the coordinate system, which is centered in the middle of the octahedron, has identical flow conditions. Within each octant, the problem can be further split into six symmetric subdomains. Thus, the problem can be fully described by one-48th of the whole domain. In order to avoid an asymmetric initialization, the inner

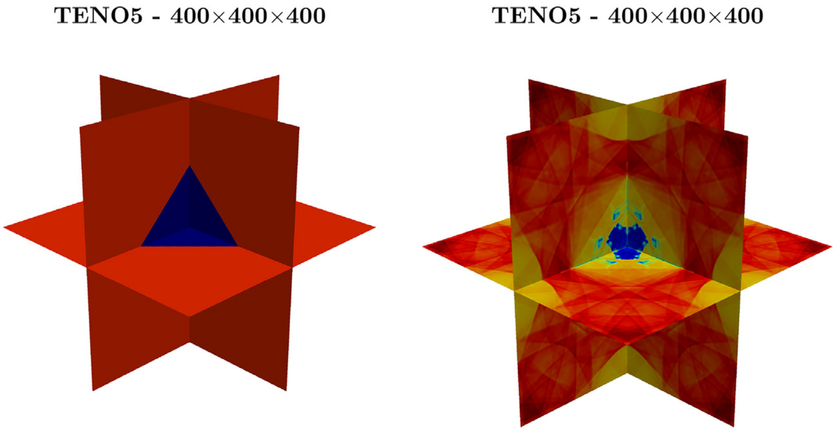


Fig. 13. 3D implosion: color density map (blue= 0.4 to red= 1.13) for $t = 0$ and $t = 2.5$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

low-pressure domain is given by

$$\begin{aligned}
(x-0.3)+(y-0.3)+(z-0.3) &< 0.15+10^{-10} \quad \cap \\
(x-0.3)+(y-0.3)-(z-0.3) &< 0.15+10^{-10} \quad \cap \\
(x-0.3)-(y-0.3)+(z-0.3) &< 0.15+10^{-10} \quad \cap \\
(x-0.3)-(y-0.3)-(z-0.3) &< 0.15+10^{-10} \quad \cap \\
-(x-0.3)+(y-0.3)+(z-0.3) &< 0.15+10^{-10} \quad \cap \\
-(x-0.3)+(y-0.3)-(z-0.3) &< 0.15+10^{-10} \quad \cap \\
-(x-0.3)-(y-0.3)+(z-0.3) &< 0.15+10^{-10} \quad \cap \\
-(x-0.3)-(y-0.3)-(z-0.3) &< 0.15+10^{-10}
\end{aligned} \tag{27}$$

with $(\rho, u, v, p)_{inner} = (0.125, 0, 0, 0.14)$, $(\rho, u, v, p)_{outer} = (1.0, 0, 0, 1.0)$ and $\gamma = 1.4$. Symmetry boundary conditions are applied at each side of the domain.

In Fig. 13, density contours are shown for both the initial time step $t = 0$ and the final time step $t = 2.5$ using a resolution of 400^3 cells. We monitored the floating-point representation of the state values in all 48 subdomains and found exact agreement for all simulation times. Furthermore, a jet along each diagonal similar to the one obtained in two dimensions can be detected using a density contour as shown in Fig. 14. Note, that each of the jets consists of six perfectly symmetric parts.

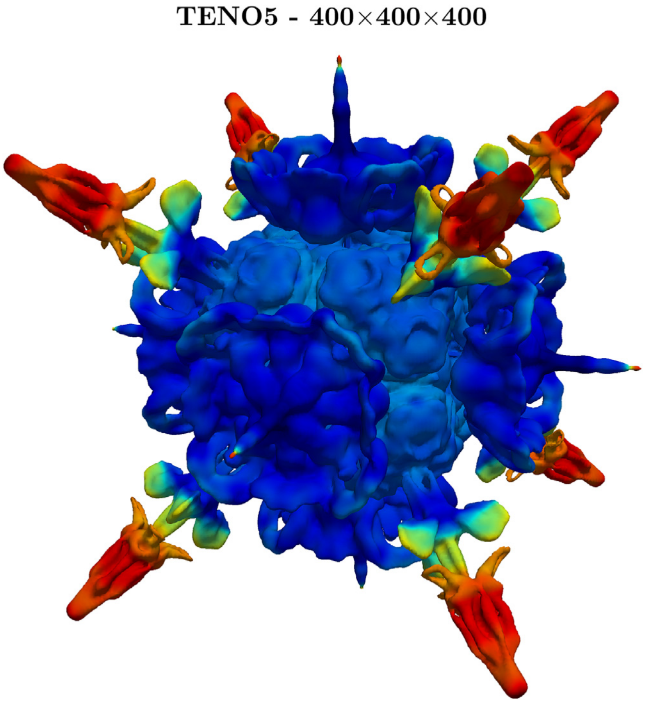


Fig. 14. 3D implosion $t = 2.5$: density contour line for the density value 0.86 colored according to pressure values (blue= 0.8 to red= 1.0). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

6. Conclusion

Various examples of symmetry breaking using low-dissipative schemes are reported in literature. Often this effect is used to judge the quality, i.e. the effect of numerical dissipation, of a scheme. Clearly, high-order discretization schemes with highly resolved meshes allow to decrease numerical viscosity effects considerably. Thus, damping of the inherent physical instabilities in the inviscid Euler equations is now insufficient, and the onset of symmetry breaking with such low-dissipative schemes first is a manifestation of algorithmic imperfections rather than of the low dissipation of the numerical scheme.

We have identified floating-point truncation errors as the main source to trigger these instabilities. Due to their exponential growth the negligibly small initial disturbances grow rapidly during the simulation and may even dominate the final huge-scale topology of the simulation result. We show that such symmetry-breaking is not a physical result of highly accurate high-resolution schemes, but rather the result of algorithmic artifacts such as the lack of associativity, which no longer are hidden by numerical dissipation.

We present implementation strategies to evaluate consistently floating-point arithmetic within a finite volume solver as solution to the prescribed problem. A generalized procedure is provided to guarantee directional independence that allows for maintaining the inherent symmetries as shown for a broad range of test cases both in two and three dimensions. The symmetry of the state variables can now be maintained exactly up to floating-point precision independently of resolution.

The simulation of inviscid or very-large Reynolds number flows with extreme resolutions will become increasingly common with the evolution from peta- to exascaling computing power. In this paper we have addressed an important aspect of the increasing relevance of implementation verification that accompanies this evolution.

Declaration of competing interest

The authors declare that they do not have any conflict of interests.

Acknowledgments

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 667483). The authors gratefully acknowledge the Gauss Centre for Supercomputing e.V. for funding this project by providing computing time on the GCS Supercomputer SuperMUC at Leibniz Supercomputing Centre.

References

[1] Hadjadj A, Kudryavtsev A. Computation and flow visualization in high-speed aerodynamics. J Turbul 2005;6:N16.
[2] Bermejo-Moreno I, Campo L, Larsson J, Bodart J, Helmer D, Eaton JK. Confinement effects in shock wave/turbulent boundary layer interactions through wall-modelled large-eddy simulations. J Fluid Mech 2014;758:5–62.
[3] Kannan K, Kedelty D, Herrmann M. An in-cell reconstruction finite volume method for flows of compressible immiscible fluids. J Comput Phys 2018;373:784–810.
[4] Meng JC, Colonius T. Numerical simulation of the aerobreakup of a water droplet. J Fluid Mech 2018;835:1108–35.
[5] Shi J, Zhang Y-T, Shu C-W. Resolution of high order WENO schemes for complicated flow structures. J Comput Phys 2003;186(2):690–6.
[6] Cheng T, Lee K. Numerical simulations of underexpanded supersonic jet and free shear layer using WENO schemes. Int J Heat Fluid Flow 2005;26(5):755–70.
[7] Zhang Y-T, Shi J, Shu C-W, Zhou Y. Numerical viscosity and resolution of high-order weighted essentially nonoscillatory schemes for compressible flows with high reynolds numbers. Phys Rev E 2003;68(4):046709.
[8] Harten A, Engquist B, Osher S, Chakravarthy SR. Uniformly high order accurate essentially non-oscillatory schemes, III. J Comput Phys 1987;71(2):231–303.
[9] Liu X-D, Osher S, Chan T. Weighted essentially non-oscillatory schemes. J Comput Phys 1994;115(1):200–12.
[10] Jiang G-S, Shu C-W. Efficient implementation of weighted ENO schemes. J Comput Phys 1996;126(1):202–28.
[11] Zhang P, Wong SC, Shu C-W. A weighted essentially non-oscillatory numerical scheme for a multi-class traffic flow model on an inhomogeneous highway. J Comput Phys 2006;212(2):739–56.
[12] Carrillo JA, Gamba IM, Majorana A, Shu C-W. A WENO-solver for the transients of Boltzmann–poisson system for semiconductor devices: performance and comparisons with monte carlo methods. J Comput Phys 2003;184(2):498–525.
[13] Filbet F, Shu C-W. Approximation of hyperbolic models for chemosensitive movement. SIAM J Sci Comput 2005;27(3):850–72.
[14] Aràndiga F, Belda AM. Weighted ENO interpolation and applications. Commun Nonlinear Sci Numer Simulat 2004;9(2):187–95.
[15] Shu C-W. High order weighted essentially nonoscillatory schemes for convection dominated problems. SIAM Review 2009;51(1):82–126.
[16] Henrick AK, Aslam TD, Powers JM. Mapped weighted essentially non-oscillatory schemes: achieving optimal order near critical points. J Comput Phys 2005;207(2):542–67.
[17] Borges R, Carmona M, Costa B, Don WS. An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws. J Comput Phys 2008;227(6):3191–211.
[18] Balsara DS, Shu C-W. Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy. J Comput Phys 2000;160(2):405–52.
[19] Gerolymos G, Sénéchal D, Vallet I. Very-high-order WENO schemes. J Comput Phys 2009;228(23):8481–524.
[20] Balsara DS, Garain S, Shu C-W. An efficient class of WENO schemes with adaptive order. J Comput Phys 2016;326:780–804.
[21] Fu L, Hu XY, Adams NA. A family of high-order targeted ENO schemes for compressible-fluid simulations. J Comput Phys 2016;305:333–59.
[22] Fu L, Hu XY, Adams NA. Targeted ENO schemes with tailored resolution property for hyperbolic conservation laws. J Comput Phys 2017;349:97–121.
[23] Rossinelli D., Hejazialhosseini B., Hadjidoukas P., Bekas C., Curioni A., Bertsch A., et al. 11 PFLOP/S simulations of cloud cavitation collapse. Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis2013;:1–13.
[24] Remacle J-F, Flaherty JE, Shephard MS. An adaptive discontinuous Galerkin technique with an orthogonal basis applied to compressible flow problems. SIAM Rev. 2003;45(1):53–72.
[25] Xu Z, Shu C-W. Anti-diffusive flux corrections for high order finite difference WENO schemes. J Comput Phys 2005;205(2):458–85.
[26] Sun Z-S, Luo L, Ren Y-X, Zhang S-Y. A sixth order hybrid finite difference scheme based on the minimized dispersion and controllable dissipation technique. J Comput Phys 2014;270:238–54.
[27] Delin C, Zhongguo S, Zhu H, Guang X. Improvement of the weighted essentially nonoscillatory scheme based on the interaction of smoothness indicators. Int J Numer Methods Fluids 2017;85(12):693–711.
[28] Balsara DS, Dumbser M, Abgrall R. Multidimensional HLLC Riemann solver for unstructured meshes–with application to euler and MHD flows. J Comput Phys 2014;261:172–208.
[29] Sutherland RS. A new computational fluid dynamics code i: Fyris alpha. Astrophys Space Sci 2010;327(2):173–206.
[30] Toro EF. Riemann solvers and numerical methods for fluid dynamics: a practical introduction. Springer Science & Business Media; 2013.
[31] LeVeque RJ. Finite volume methods for hyperbolic problems, 31. Cambridge University Press; 2002.
[32] Gottlieb S, Shu C-W, Tadmor E. Strong stability-preserving high-order time discretization methods. SIAM Rev 2001;43(1):89–112.
[33] Roe PL. Approximate riemann solvers, parameter vectors, and difference schemes. J Comput Phys 1981;43(2):357–72.
[34] Harten A, Lax PD, Leer Bv. On upstream differencing and godunov-type schemes for hyperbolic conservation laws. SIAM Rev 1983;25(1):35–61.
[35] Toro EF, Spruce M, Speares W. Restoration of the contact surface in the HLL-Riemann solver. Shock Waves 1994;4(1):25–34.
[36] Batten P, Clarke N, Lambert C, Causon DM. On the choice of wavespeeds for the HLLC riemann solver. SIAM J Sci Comput 1997;18(6):1553–70.
[37] Hu XY, Adams NA, Shu C-W. Positivity-preserving method for high-order conservative schemes solving compressible euler equations. J Comput Phys 2013;242:169–80.
[38] Johnsen E, Colonius T. Implementation of WENO schemes in compressible multicomponent flow problems. J Comput Phys 2006;219(2):715–32.
[39] Coralic V, Colonius T. Finite-volume WENO scheme for viscous compressible multicomponent flows. J Comput Phys 2014;274:95–121.
[40] Einfeldt B, Munz C-D, Roe PL, Sjögreen B. On godunov-type methods near low densities. J Comput Phys 1991;92(2):273–95.
[41] Qiu J, Shu C-W. On the construction, comparison, and local characteristic decomposition for high-order central WENO schemes. J Comput Phys 2002;183(1):187–209.
[42] Liska R, Wendroff B. Comparison of several difference schemes on 1D and 2D test problems for the euler equations. SIAM J Sci Comput 2003;25(3):995–1017.
[43] Hu X, Wang Q, Adams NA. An adaptive central-upwind weighted essentially non-oscillatory scheme. J Comput Phys 2010;229(23):8952–65.
[44] Schneider EE, Robertson BE. CHOLLA: A new massively parallel hydrodynamics code for astrophysical simulation. Astrophys J Supplement Ser 2015;217(2):24.
[45] Schulz-Rinne CW, Collins JP, Glaz HM. Numerical solution of the Riemann problem for two-dimensional gas dynamics. SIAM J Sci Comput 1993;14(6):1394–414.