Journal of Computational Physics 231 (2012) 8114–8132

Contents lists available at SciVerse ScienceDirect

Journal of Computational Physics

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

New adaptive artificial viscosity method for hyperbolic systems of conservation laws

Alexander Kurganov, Yu Liu *

Mathematics Department, Tulane University, New Orleans, LA 70118, USA

ARTICLE INFO

Article history:
Received 14 April 2011
Received in revised form 1 June 2012
Accepted 26 July 2012
Available online 14 August 2012

Keywords:
Hyperbolic systems of conservation laws
Godunov-type schemes
Weak local residual
Artificial viscosity

ABSTRACT

We propose a new finite volume method for solving general multidimensional hyperbolic systems of conservation laws. Our method is based on an appropriate numerical flux and a high-order piecewise polynomial reconstruction. The latter is utilized without any computationally expensive nonlinear limiters, which are typically needed to guarantee nonlinear stability of the scheme. Instead, we enforce stability of the proposed method by adding a new *adaptive artificial viscosity*, whose coefficients are proportional to the size of the weak local residual, which is sufficiently large ($\sim\Delta$, where Δ is a discrete small scale) at the shock regions, much smaller ($\sim\Delta^{\alpha}$, where α is close to 2) near the contact waves, and very small ($\sim\Delta^4$) in the smooth parts of the computed solution.

We test the proposed scheme on a number of benchmarks for both scalar conservation laws and for one- and two-dimensional Euler equations of gas dynamics. The obtained numerical results clearly demonstrate the robustness and high accuracy of the new method.

© 2012 Elsevier Inc. All rights reserved.

1. Introduction

We study finite volume Godunov-type schemes for one-dimensional (1-D),

$$\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x = \mathbf{0} \tag{1.1}$$

and two-dimensional (2-D),

$$\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x + \mathbf{g}(\mathbf{u})_y = \mathbf{0}, \tag{1.2}$$

hyperbolic systems of conservation laws. In the above formulae, $\mathbf{u} \in \mathbb{R}^N$ is a vector of conserved quantities and $\mathbf{f}(\mathbf{u})$ and $\mathbf{g}(\mathbf{u})$ are fluxes (the hyperbolicity of the system is ensured provided the matrix $n_x\partial\mathbf{f}/\partial\mathbf{u} + n_y\partial\mathbf{g}/\partial\mathbf{u}$ have N real eigenvalues and N linearly independent corresponding eigenvectors for any $(n_x, n_y)^T \in \mathbb{R}^2$).

Godunov-type schemes form a class of projection-evolution methods, in which at each time step the computed solution is approximated by a global piecewise polynomial function (called a reconstruction and is supposed to satisfy conservation, accuracy and non-oscillatory requirements), which is evolved in time (from a current time level to the next one) according to the integral form of (1.1) (or (1.2), respectively). The latter is obtained by integrating the studied system of conservation laws over the space–time control volume of size $\Delta t \times \Delta x$ (or $\Delta t \times \Delta x \times \Delta y$, respectively), where Δt is a small temporal scale and Δx and Δy are small spatial scales. Depending on the way the control volumes are selected, the class of Godunov-type

* Corresponding author.
E-mail addresses: kurganov@math.tulane.edu (A. Kurganov), yliu3@tulane.edu (Y. Liu).

0021-9991/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jcp.2012.07.040

schemes can be split into two subclasses: upwind schemes (their prototype is the original first-order Godunov scheme [7]) and central schemes (whose prototype is the first-order Lax-Friedrichs scheme [5,26]). First-order Godunov-type schemes are obtained using the first-order piecewise constant reconstruction, while their higher-order extensions hinge on replacement of the first-order reconstruction with a higher-order one, which consists of either linear, parabolic, cubic or even higher degree polynomial pieces.

Many of the Godunov-type schemes may be written in a particularly simple semi-discrete form (method of lines), obtained by integrating (1.1) ((1.2)) over grid cells of size Δx ($\Delta x \times \Delta y$) and replacing the fluxes at the cell interfaces (flux integrals along the cell boundaries) by appropriate numerical fluxes. In the 1-D case, a semi-discrete scheme on a uniform grid with $x_j := j\Delta x$ can be written as follows. Let $x_{j\pm\frac{1}{2}} := x_j \pm \Delta x/2$ and the computational domain consists of the cells $I_j := [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$. Then, the cell averages of $\mathbf{u}(x,t)$ over I_j at time t, denoted by

$$\bar{\mathbf{u}}_j(t) \approx \frac{1}{\Delta x}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \mathbf{u}(x,t)\,dx$$

are evolved in time by numerically solving the following system of ODEs:

$$\frac{d}{dt}\bar{\mathbf{u}}_j(t) = -\frac{\mathbf{H}_{j+\frac{1}{2}}(t) - \mathbf{H}_{j-\frac{1}{2}}(t)}{\Delta x}, \tag{1.3}$$

where $\mathbf{H}_{j+\frac{1}{2}}$ is a numerical flux, obtained using the piecewise polynomial approximation, reconstructed at time t using the available cell averages $\{\bar{\mathbf{u}}_j(t)\}$. In our numerical experiments reported in Section 3, we have used the central-upwind fluxes developed in [20,21] (for completeness, both the 1-D and 2-D central-upwind fluxes are briefly reviewed in Appendixes A and B). However, we would like to stress that the proposed adaptive artificial viscosity method is not tied to any specific numerical flux and can be implemented with one's favorite flux, for which the semi-discrete scheme (1.3) is linearly stable.

It is well-known that the systems (1.1) and (1.2) admit nonsmooth solutions that may contain shocks, contact discontinuities and rarefaction waves. Therefore, when such solutions are to be captured, linearly stable methods may develop large spurious oscillations and even blow up. Thus, a good numerical method must be nonlinearly stable. Nonlinear stability of Godunov-type schemes is typically guaranteed by enforcing non-oscillatory nature of the piecewise polynomial reconstruction with the help of nonlinear limiters. However, such limiters may be very complicated and computationally expensive, especially when a high-order multidimensional scheme is to be designed (see, e.g., [38] and references therein). Alternatively, one may use less computationally expensive unlimited reconstructions, while enforcing nonlinear stability by adding artificial viscosity to the PDE system in the regions of discontinuities. Obviously, to ensure consistency of the numerical approximation, this artificial viscosity must vanish as $\Delta \to 0$, where in the 1-D case, $\Delta := \max(\Delta t, \Delta x)$ and in the 2-D case, $\Delta := \max(\Delta t, \Delta x, \Delta y)$.

The idea of adding artificial viscosity was first proposed in [45] and since then it was notably adopted in many works including [1,9,10,13–15,17,31,36,41,46] among others. The major difficulty in designing a highly accurate and robust artificial viscosity method is to make sure that a sufficient amount of stabilizing diffusion is added wherever it is needed to ensure stability, while in the rest of the computational domain the diffusion must be either switched off or small enough not to affect the high accuracy of the scheme there. At the same time, if the viscosity coefficient is too large in the areas of discontinuity, the solution will be overly smeared there. Therefore, to achieve overall high resolution, the viscosity should be added in an adaptive way using a certain indicator, which should automatically pick rough parts of the computed solution and determine the (optimal) amount of viscosity needed to be added there.

In this paper, we propose a new adaptive artificial viscosity method. In our method, the viscosity coefficients are chosen to be proportional to the size of the weak local residual (WLR), which was originally developed in [18] and then used in [3,19] as a smoothness indicator (SI) for several adaption algorithms. The key point we use here is that for a convergent numerical method of formal order r, the WLR is proportional to Δ near (nonlinear) shocks, while it is much smaller ($\sim \Delta^{\alpha}$, α is close to 2) at (linear) contact waves, and tiny in the smooth parts of the solution ($\sim \Delta^{\min(4,r+2)}$). Therefore, the artificial viscosity vanishes as one refines the grid, so that it is consistent with the original hyperbolic equations. Moreover, the rate at which the viscosity coefficients decay, allows us to achieve the main goal—to stabilize the solution at shock regions without oversmearing contact discontinuities or affecting the high resolution of smooth parts of the computed solution.

The key question one has to address to make the adaptive artificial viscosity method robust is how to select the coefficients of proportionality in such a way that the computed solution is non-oscillatory, but its discontinuous parts are well resolved, that is, they are not overly smeared. To tune the viscosity coefficients, we adopt the strategy proposed in [10]: for the problem at hand, the coefficients are first adjusted on a very coarse mesh and then used for the high-resolution computation on finer meshes.

This paper is organized as follows. In Section 2, we describe the derivation of the modified version of the WLR and use it to introduce adaptive artificial viscosity into both (1.1) and (1.2). The obtained systems are then to be solved by semi-discrete Godunov-type schemes using unlimited high-order reconstructions, which are described in Appendices A and B together with the central-upwind fluxes that have been used in our numerical experiments. In Section 3, we apply the new adaptive artificial viscosity method to a number of numerical examples in both 1-D and 2-D. We compare our method with the

central-upwind scheme combined with the fifth-order WENO reconstruction [2,37,38]. The obtained results indicate high accuracy, efficiency and robustness of our new method which seems to outperform the WENO approach.

2. New adaptive artificial viscosity method

Godunov-type schemes are projection-evolution methods, in which the approximate solution, realized by its cell averages, is first projected onto the space of piecewise polynomials, and then the obtained piecewise polynomial interpolant is evolved to the new time level using the integral form of conservation laws.

We will focus on semi-discrete schemes, in which the evolution step is performed with the help of numerical fluxes (many reliable numerical fluxes are available, see, e.g., [6,20,27,38,43]). In our numerical experiments, we have used the central-upwind fluxes from [20], which are presented in Appendices A and B.

The (formal) order of the scheme depends on the (formal) order of the piecewise polynomial reconstruction (we would like to stress that a high-order reconstruction is typically required to achieve high resolution). To ensure a non-oscillatory nature of the scheme the reconstruction can be made non-oscillatory with the help of a nonlinear limiter. A library of such limiters is available [2,6,11,12,16,27–30,32,35,38–40,43,44], however, one must realize that first, in the system case, the use of limiters cannot guarantee the scheme to be oscillation-free, and second, the limiters, especially high-order ones, may be extremely computationally expensive and cumbersome, which would dramatically reduce efficiency of the scheme.

An alternative approach for enforcing stability of the method is to introduce numerical diffusion by adding artificial viscosity term to the hyperbolic system. This idea dates back to [45], where a viscosity term was added to the momentum equation so that the solution of the new system satisfies the Rankine–Hugoniot jump condition in the shock region and has negligible effect outside the shock layer (see also [1]). The critical question is how to adaptively introduce the right amount of viscosity in the right place. It is quite easy to come up with a reliable strategy in the scalar case. In this paper, we propose a new way of adding artificial viscosity, which applies to both scalar and system cases.

2.1. One-dimensional scheme

We augment the hyperbolic system (1.1) with an adaptive artificial viscosity:

$$\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x = C(\varepsilon(\mathbf{u})\mathbf{u}_x)_x, \tag{2.1}$$

where C is a tunable positive viscosity coefficient and $\varepsilon(\mathbf{u})$ is a nonnegative quantity, whose size is automatically adjusted depending on the local properties of $\mathbf{u}$. For computed solutions, we will make $\varepsilon(\mathbf{u})$ proportional to the WLR, which is one of the key points in the construction of our new method.

We would like to stress that our goal is to derive a powerful, efficient and highly accurate method, which can be applied as a black-box solver to any hyperbolic systems of conservation laws. Therefore, the artificial diffusion term on the right-hand side (RHS) of (2.1) has a general form. For a particular problem at hand, one can add a different artificial diffusion term, which may be based on a small physical viscosity present in the underlying physical system (for instance, in the case of the Euler equations of gas dynamics, one can add Navier–Stokes type artificial diffusion terms). Advantages and disadvantages of such an approach will be studied in future work.

2.1.1. One-dimensional weak local residuals (WLR)

In this section, we derive formulae for the WLRs following the approach from [18]. We first consider the system (1.1) in the computational domain $X \times [0,T]$ and recall that by definition weak solutions of (1.1) satisfy the integral equation

$$\mathbf{E}(\mathbf{u},\phi) := \int_{t=0}^{T}\int_X \{\mathbf{u}(x,t)\phi_t(x,t) + \mathbf{f}(\mathbf{u}(x,t))\phi_x(x,t)\}\,dx\,dt + \int_X \mathbf{u}(x,0)\phi(x,0)\,dx = \mathbf{0} \tag{2.2}$$

for all test functions $\phi(x,t) \in C_0^1(X \times [0,T))$. We then assume that the point values of the computed solution $\mathbf{u}^n_{j+\frac{1}{2}}$ are available throughout the domain $X \times [0,T]$ and introduce the function $\mathbf{u}^\Delta(x,t)$ as a piecewise constant approximation of the computed solution by setting

$$\mathbf{u}^\Delta(x,t) := \mathbf{u}^n_{j+\frac{1}{2}}, \quad \text{if} \quad (x,t) \in [x_j, x_{j+1}) \times [t^{n-\frac{1}{2}}, t^{n+\frac{1}{2}}). \tag{2.3}$$

The quantity $\mathbf{E}(\mathbf{u}^\Delta,\phi)$ can then be viewed as a weak residual for $\mathbf{u}^\Delta$ with respect to ϕ. Obviously, the residual (2.3) cannot be computed in practice since ϕ is a general test function. However, following the approach in [3,18,21], one can approximate the global test function ϕ with the help of localized test functions $\phi^{n-\frac{1}{2}}_{j+\frac{1}{2}}$ and then introduce the corresponding WLRs.

In this paper, we use

$$\phi^{n-\frac{1}{2}}_{j+\frac{1}{2}}(x,t) = B_{j+\frac{1}{2}}(x)B^{n-\frac{1}{2}}(t), \tag{2.4}$$

where

$$B_{j+\frac{1}{2}}(x) := \begin{cases} \frac{1}{2}\left(\frac{x-x_{j-1}}{\Delta x}\right)^2, & \text{if } x_{j-1} \leqslant x < x_j, \\ \frac{3}{4} - \left(\frac{x-x_{j+\frac{1}{2}}}{\Delta x}\right)^2, & \text{if } x_j \leqslant x < x_{j+1}, \\ \frac{1}{2}\left(\frac{x-x_{j+2}}{\Delta x}\right)^2, & \text{if } x_{j+1} \leqslant x < x_{j+2}, \\ 0, & \text{otherwise} \end{cases} \tag{2.5}$$

and

$$B^{n-\frac{1}{2}}(t) := \begin{cases} \frac{t-t^{n-\frac{3}{2}}}{\Delta t}, & \text{if } t^{n-\frac{3}{2}} \leqslant t < t^{n-\frac{1}{2}}, \\ \frac{t^{n+\frac{1}{2}}-t}{\Delta t}, & \text{if } t^{n-\frac{1}{2}} \leqslant t < t^{n+\frac{1}{2}}, \\ 0, & \text{otherwise} \end{cases} \tag{2.6}$$

are the quadratic and linear B-splines with the localized supports of size $|\mathrm{supp}\left(B_{j+\frac{1}{2}}\right)| = 3\Delta x$ and $|\mathrm{supp}\left(B^{n-\frac{1}{2}}\right)| = 2\Delta t$, respectively. One can easily show that for every smooth test function ϕ there exist coefficients $\beta^{n-\frac{1}{2}}_{j+\frac{1}{2}}$, which are independent of Δx and Δt and satisfy

$$\phi(x,t) = \sum_j \sum_n \beta^{n-\frac{1}{2}}_{j+\frac{1}{2}} \phi^{n-\frac{1}{2}}_{j+\frac{1}{2}}(x,t) + \mathcal{O}(\Delta^2). \tag{2.7}$$

We then substitute (2.3)–(2.6) into (2.2), use (2.7) and after a straightforward calculation obtain

$$\mathbf{E}(\mathbf{u}^\Delta, \phi) = \sum_j \sum_n \beta^{n-\frac{1}{2}}_{j+\frac{1}{2}} \mathbf{E}^{n-\frac{1}{2}}_{j+\frac{1}{2}} + \mathcal{O}(\Delta^4),$$

where the quantities

$$\mathbf{E}^{n-\frac{1}{2}}_{j+\frac{1}{2}} = \frac{1}{6}\left[\mathbf{u}^n_{j+\frac{3}{2}} - \mathbf{u}^{n-1}_{j+\frac{3}{2}} + 4\left(\mathbf{u}^n_{j+\frac{1}{2}} - \mathbf{u}^{n-1}_{j+\frac{1}{2}}\right) + \mathbf{u}^n_{j-\frac{1}{2}} - \mathbf{u}^{n-1}_{j-\frac{1}{2}}\right]\Delta x + \frac{1}{4}\left[\mathbf{f}\left(\mathbf{u}^n_{j+\frac{3}{2}}\right) - \mathbf{f}\left(\mathbf{u}^n_{j-\frac{1}{2}}\right) + \mathbf{f}\left(\mathbf{u}^{n-1}_{j+\frac{3}{2}}\right) - \mathbf{f}\left(\mathbf{u}^{n-1}_{j-\frac{1}{2}}\right)\right]\Delta t \tag{2.8}$$

can be viewed as WLRs.

Remark 2.1. Notice that since piecewise polynomial reconstructions used to compute point values of finite volume solutions are generically discontinuous, two (different) point values are available at each cell interface $x = x_{j+\frac{1}{2}}$. We can use either one or their convex combination to compute the WLR in (2.8). In our numerical experiments, we have used the value on the left-hand side of each cell interface.

One may show (see [3,18,19]) that the size of WLRs is proportional to Δ near the shock (in poorly resolved) regions, while in the smooth (highly resolved) parts of the computed solution the WLRs are several orders of magnitude smaller. Moreover, our numerical experiments demonstrate that typically

$$\|\mathbf{E}^{n-\frac{1}{2}}_{j+\frac{1}{2}}\|_\infty \sim \begin{cases} \Delta, & \text{near shock waves}, \\ \Delta^\alpha, & \text{near contact waves}, \quad 1 < \alpha \leqslant 2, \\ \Delta^p, & \text{in smooth regions}, \end{cases}$$

where $p = \min\{r+2, 4\}$ and r is the formal order of accuracy of a scheme. Here, the upper bound $p = 4$ is related to the way the computed solution is extended in (2.3) and to the particular choice of the locally supported test functions $\phi^{n-\frac{1}{2}}_{j+\frac{1}{2}}(x,t)$ used in the calculation of the WLR.

2.1.2. One-dimensional adaptive artificial viscosity method

Let us assume that at a certain time level t the approximate solution and the WLR values $\mathbf{E}_{j+\frac{1}{2}}$ are available. We then select a linearly stable numerical flux $\mathbf{H}_{j+\frac{1}{2}}$ and evolve the solution according to the semi-discrete scheme

$$\frac{d}{dt}\bar{\mathbf{u}}_j = -\frac{\mathbf{H}_{j+\frac{1}{2}} - \mathbf{H}_{j-\frac{1}{2}}}{\Delta x} + C\left(\frac{\varepsilon_{j+\frac{1}{2}}\Delta\bar{\mathbf{u}}_{j+\frac{1}{2}} - \varepsilon_{j-\frac{1}{2}}\Delta\bar{\mathbf{u}}_{j-\frac{1}{2}}}{(\Delta x)^2}\right). \tag{2.9}$$

Here, $\Delta\bar{\mathbf{u}}_{j+\frac{1}{2}} := \bar{\mathbf{u}}_{j+1} - \bar{\mathbf{u}}_j$ and $\varepsilon_{j+\frac{1}{2}} := \max(|\mathbf{E}^{(i)}_{j-\frac{1}{2}}|, |\mathbf{E}^{(i)}_{j+\frac{1}{2}}|, |\mathbf{E}^{(i)}_{j+\frac{3}{2}}|)$, where $\mathbf{E}^{(i)}_{j+\frac{1}{2}}$ is the i^{th} component of the vector so that the artificial viscosity is proportional to the values of the WLR in the vicinity of $x_{j+\frac{1}{2}}$. In principle, one can use any component of the vector $\mathbf{E}_{j+\frac{1}{2}}$, but some components might be superior depending on a problem at hand. We also notice that here, the quantities $\bar{\mathbf{u}}_j, \mathbf{H}_{j+\frac{1}{2}}$ and $\varepsilon_{j+\frac{1}{2}}$ depend on t, but we suppress this dependence for brevity.

Eq. (2.9) represents our new adaptive artificial viscosity method. The viscosity flux $C\varepsilon_{j+\frac{1}{2}}\Delta\bar{\mathbf{u}}_{j+\frac{1}{2}}/\Delta x$ is proportional to the size of the WLR in the vicinity of $x_{j+\frac{1}{2}}$. This will guarantee that a substantial amount of the numerical viscosity is added to the "rough", non-smooth parts of the solution only, while in the smooth regions the numerical viscosity is several orders of magnitude smaller, and thus the high accuracy will not be affected there. Since no limiters are going to be used, the

scheme must have enough numerical diffusion in the shock areas to prevent oscillation. This goal will be achieved by a proper choice of the viscosity coefficients C.

In the scalar case, the size of C can be estimated using the following lemma, which makes a direct connection between the amount of numerical diffusion and the total variation of the computed solution.

Lemma 2.1 ([42]). *Consider the following three-point scheme written in a viscous form:*

$$\bar{u}_j^{n+1} = \bar{u}_j^n - \lambda(f_{j+\frac{1}{2}}^n - f_{j-\frac{1}{2}}^n) + \frac{1}{2}\left(Q_{j+\frac{1}{2}}^n \Delta\bar{u}_{j+\frac{1}{2}}^n - Q_{j-\frac{1}{2}}^n \Delta\bar{u}_{j-\frac{1}{2}}^n\right), \quad \lambda = \frac{\Delta t}{\Delta x}, \tag{2.10}$$

where

$$f_{j\pm\frac{1}{2}}^n := \frac{f(\bar{u}_j^n) + f(\bar{u}_{j\pm1}^n)}{2}. \tag{2.11}$$

If $Q_{j+\frac{1}{2}}^n$ *satisfies*

$$\lambda\left|\frac{f(\bar{u}_{j+1}^n) - f(\bar{u}_j^n)}{\Delta\bar{u}_{j+\frac{1}{2}}^n}\right| \leqslant Q_{j+\frac{1}{2}}^n \leqslant 1 \tag{2.12}$$

for all $j \in \mathbb{Z}$ *and* $n \geqslant 0$, *then the scheme (2.1) is total variation diminishing (TVD).*

Let us now apply the forward Euler time discretization to (2.9). This results in the following fully discrete scheme:

$$\bar{u}_j^{n+1} = \bar{u}_j^n - \lambda\left(H_{j+\frac{1}{2}}^n - H_{j-\frac{1}{2}}^n\right) + \frac{\lambda C}{\Delta x}\left(\varepsilon_{j+\frac{1}{2}}^n \Delta\bar{u}_{j+\frac{1}{2}}^n - \varepsilon_{j-\frac{1}{2}}^n \Delta\bar{u}_{j-\frac{1}{2}}^n\right). \tag{2.13}$$

Comparing with (2.10) and noticing that $H_{j+\frac{1}{2}}^n$ is an approximation of $f_{j+\frac{1}{2}}^n$, we may obtain a rough estimate on the size of C. To this end, we use Lemma 2.1, which suggests that the adaptive artificial viscosity method should be TVD provided the following condition is satisfied:

$$\lambda\left|\frac{f(\bar{u}_{j+1}^n) - f(\bar{u}_j^n)}{\Delta\bar{u}_{j+\frac{1}{2}}^n}\right| \leqslant \frac{2\lambda C\varepsilon_{j+\frac{1}{2}}^n}{\Delta x} \leqslant 1.$$

Note that by the CFL condition $\lambda\left|\left(f(\bar{u}_{j+1}^n) - f(\bar{u}_j^n)\right)/\Delta\bar{u}_{j+\frac{1}{2}}^n\right| \leqslant 0.5$, hence we set

$$C = \frac{\Delta x}{4\lambda\varepsilon_{\max}^n}, \tag{2.14}$$

where $\varepsilon_{\max}^n = \max_j\left\{\varepsilon_{j+\frac{1}{2}}^n\right\}$. This will ensure that near the shock (where $\varepsilon_{j+\frac{1}{2}}^n \sim \varepsilon_{\max}^n$) the amount of artificial viscosity will be sufficient to stabilize the computed solution. We now examine the sharpness of the estimate (2.14) numerically.

Scalar Numerical Example. Consider the 1-D inviscid Burgers' equation,

$$u_t + \left(\frac{u^2}{2}\right)_x = 0,$$

subject to the 2π-periodic boundary conditions and the following initial condition:

$$u(x,0) = \sin x + 2.5.$$

We apply the adaptive artificial viscosity method to this initial-boundary value problem with C given by (2.14). In Fig. 2.1 (upper row), we show the solution computed on four different grids with 50, 200, 400 and 800 grid cells per period at time $t = 2$. As one can see, the solution seems to converge to the exact one, but the shock is not well-resolved even when 800 cells are used. We then significantly reduce the numerical diffusion by replacing the "theoretical" coefficient C with a 15 times smaller one (we now take $C = \Delta x/(60\lambda\varepsilon_{\max}^n)$). The obtained results, plotted in Fig. 2.1 (lower row), show a significant improvement in the resolution of the shock wave without appearance of any spurious oscillations. We conjecture that the method remains stable thanks to the numerical diffusion present at central-upwind numerical fluxes, which are—unlike the centered flux (2.11)–linearly stable and thus require much less additional artificial viscosity to be added to achieve the nonlinear stability.

This example suggests that the "theoretical" bound on C, obtained in (2.14), is not sharp at all and that even in the scalar case a better strategy should be used for selecting the viscosity coefficient C. To tune it, we adopt the approach proposed in [10]: for the problem at hand, the coefficient C is first adjusted on a very coarse grid (this makes the tuning process computationally inexpensive) and then used for high-resolution computation on finer grids. The tuning is conducted to find the value of C, which leads to sharp resolution of the discontinuities without producing large spurious oscillations. Finding an optimal C may be a very delicate issue. However, our numerical experiments clearly show that the proposed method is not too sensitive to the selected value of C, which makes our approach quite robust.

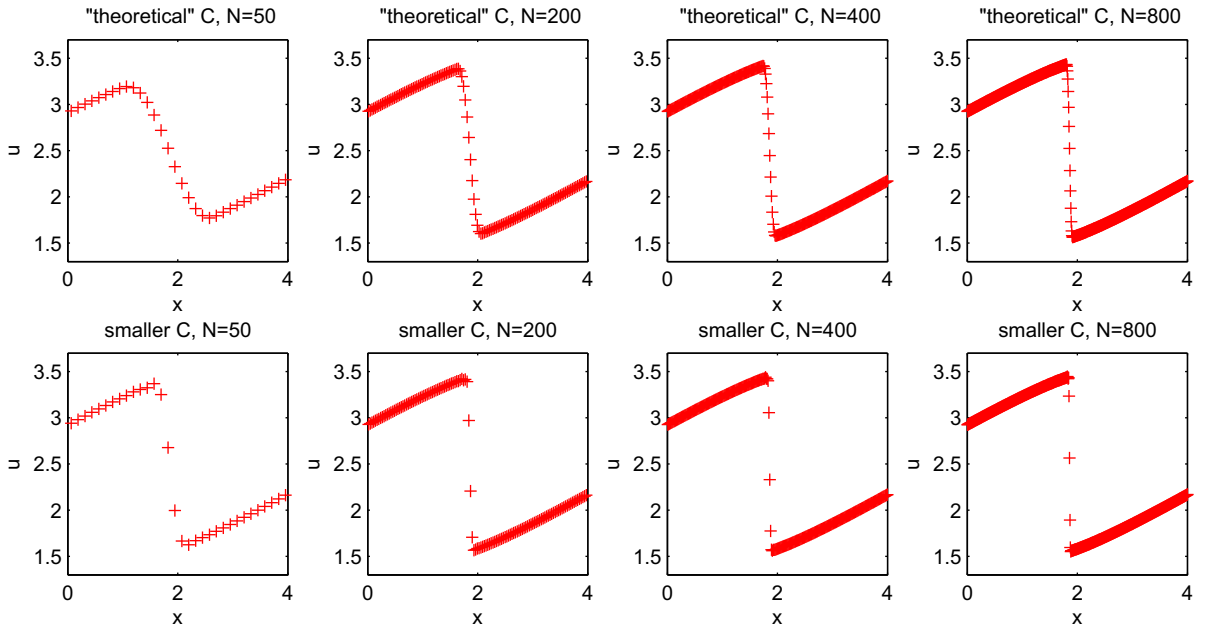

Fig. 2.1. Inviscid Burgers' equation. Numerical solution zoomed at $x \in [0,4]$ at time $t = 2$ with 50, 200, 400 and 800 grid cells per period and either the "theoretical" (upper row) or a smaller (lower row) C.

2.2. Two-dimensional scheme

To construct the 2-D adaptive artificial viscosity method, we first extend Eq. (2.1) to the case of two space dimensions:

$$\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x + \mathbf{g}(\mathbf{u})_y = C\Big[(\varepsilon(\mathbf{u})\mathbf{u}_x)_x + (\varepsilon(\mathbf{u})\mathbf{u}_y)_y\Big], \tag{2.15}$$

where, as in the 1-D case, C is a tunable positive viscosity coefficient and $\varepsilon(\mathbf{u})$ is a positive quantity, whose size is proportional to the WLR, which will be derived in Section 2.2.1.

We would like to emphasize that similarly to the 1-D case, the artificial diffusion added on the RHS of (2.15) has a general form, which is not tied to any particular application.

2.2.1. Two-dimensional weak local residuals

In this section, we extend the WLR from one to two space dimensions following the approach from [19]. We consider the system (1.2) in the computational domain $\Omega \times [0,T]$ and recall that by definition weak solutions of (1.2) satisfy the integral equation

$$\mathbf{E}(\mathbf{u},\phi) := \int_{t=0}^{\infty}\int_{\Omega}\big\{\mathbf{u}(x,y,t)\phi_t(x,y,t) + \mathbf{f}(\mathbf{u}(x,y,t))\phi_x(x,y,t) + \mathbf{g}(\mathbf{u}(x,y,t))\phi_y(x,y,t)\big\}\,dxdydt + \int_{\Omega}\phi(x,y,0)\mathbf{u}(x,y,0)\,dxdy = 0 \tag{2.16}$$

for all test functions $\phi(x,y,t) \in C_0^1(\Omega \times [0,T))$. In complete analogy with the 1-D case, we use the piecewise constant function

$$\mathbf{u}^{\Delta}(x,y,t) := \mathbf{u}^n_{j+\frac{1}{2},k+\frac{1}{2}}, \quad \text{if} \quad (x,y,t) \in [x_j,x_{j+1}) \times [y_k,y_{k+1}) \times [t^{n-\frac{1}{2}},t^{n+\frac{1}{2}}) \tag{2.17}$$

to approximate the computed solution and define the localized test functions by

$$\phi^n_{j+\frac{1}{2},k+\frac{1}{2}}(x,y,t) := \frac{1}{\Delta}B_{j+\frac{1}{2}}(x)B_{k+\frac{1}{2}}(y)B^{n-\frac{1}{2}}(t). \tag{2.18}$$

Here, as before, $\Delta := \max(\Delta t, \Delta x, \Delta y)$, the B-splines $B_{j+\frac{1}{2}}$ and $B^{n-\frac{1}{2}}$ are given by (2.5) and (2.6), respectively, and $B_{k+\frac{1}{2}}$ is defined in a similar way:

$$B_{k+\frac{1}{2}}(y) := \begin{cases} \frac{1}{2}\left(\frac{y-y_{k-1}}{\Delta y}\right)^2, & \text{if } y_{k-1} \leqslant y < y_k, \\ \frac{3}{4} - \left(\frac{y-y_{k+\frac{1}{2}}}{\Delta y}\right)^2, & \text{if } y_k \leqslant y < y_{k+1}, \\ \frac{1}{2}\left(\frac{y-y_{k+2}}{\Delta y}\right)^2, & \text{if } y_{k+1} \leqslant y < y_{k+2}, \\ 0, & \text{otherwise}, \end{cases}$$

After plugging (2.17) and (2.18) into (2.16), a straightforward calculation yields the 2-D version of the WLR:

$$\mathbf{E}^{n-\frac12}_{j+\frac12,k+\frac12}=\frac{1}{36\Delta}\Delta x\Delta y\,\mathcal{U}^{n-\frac12}_{j+\frac12,k+\frac12}+\frac{1}{12\Delta}\Big(\Delta y\Delta t\,\mathcal{F}^{n-\frac12}_{j+\frac12,k+\frac12}+\Delta x\Delta t\,\mathcal{G}^{n-\frac12}_{j+\frac12,k+\frac12}\Big),\tag{2.19}$$

where

$$\begin{aligned}\mathcal{U}^{n-\frac12}_{j+\frac12,k+\frac12}=&\Big[\mathbf{u}^n_{j+\frac32,k+\frac32}-\mathbf{u}^{n-1}_{j+\frac32,k+\frac32}+\mathbf{u}^n_{j+\frac32,k-\frac12}-\mathbf{u}^{n-1}_{j+\frac32,k-\frac12}+\mathbf{u}^n_{j-\frac12,k+\frac32}-\mathbf{u}^{n-1}_{j-\frac12,k+\frac32}+\mathbf{u}^n_{j-\frac12,k-\frac12}-\mathbf{u}^{n-1}_{j-\frac12,k-\frac12}\Big]\\&+4\Big[\mathbf{u}^n_{j+\frac32,k+\frac12}-\mathbf{u}^{n-1}_{j+\frac32,k+\frac12}+\mathbf{u}^n_{j-\frac12,k+\frac12}-\mathbf{u}^{n-1}_{j-\frac12,k+\frac12}+\mathbf{u}^n_{j+\frac12,k+\frac32}-\mathbf{u}^{n-1}_{j+\frac12,k+\frac32}+\mathbf{u}^n_{j+\frac12,k-\frac12}-\mathbf{u}^{n-1}_{j+\frac12,k-\frac12}\Big]+16\Big[\mathbf{u}^n_{j+\frac12,k+\frac12}-\mathbf{u}^{n-1}_{j+\frac12,k+\frac12}\Big],\end{aligned}$$

$$\begin{aligned}\mathcal{F}^{n-\frac12}_{j+\frac12,k+\frac12}=&\Big[\mathbf{f}\Big(\mathbf{u}^n_{j+\frac32,k+\frac32}\Big)-\mathbf{f}\Big(\mathbf{u}^n_{j-\frac12,k+\frac32}\Big)+\mathbf{f}\Big(\mathbf{u}^n_{j+\frac32,k-\frac12}\Big)-\mathbf{f}\Big(\mathbf{u}^n_{j-\frac12,k-\frac12}\Big)+\mathbf{f}\Big(\mathbf{u}^{n-1}_{j+\frac32,k+\frac32}\Big)-\mathbf{f}\Big(\mathbf{u}^{n-1}_{j-\frac12,k+\frac32}\Big)+\mathbf{f}\Big(\mathbf{u}^{n-1}_{j+\frac32,k-\frac12}\Big)-\mathbf{f}\Big(\mathbf{u}^{n-1}_{j-\frac12,k-\frac12}\Big)\Big]\\&+4\Big[\mathbf{f}\Big(\mathbf{u}^n_{j+\frac32,k+\frac12}\Big)-\mathbf{f}\Big(\mathbf{u}^n_{j-\frac12,k+\frac12}\Big)+\mathbf{f}\Big(\mathbf{u}^{n-1}_{j+\frac32,k+\frac12}\Big)-\mathbf{f}\Big(\mathbf{u}^{n-1}_{j-\frac12,k+\frac12}\Big)\Big]\end{aligned}$$

and

$$\begin{aligned}\mathcal{G}^{n-\frac12}_{j+\frac12,k+\frac12}=&\Big[\mathbf{g}\Big(\mathbf{u}^n_{j+\frac32,k+\frac32}\Big)-\mathbf{g}\Big(\mathbf{u}^n_{j+\frac32,k-\frac12}\Big)+\mathbf{g}\Big(\mathbf{u}^n_{j-\frac12,k+\frac32}\Big)-\mathbf{g}\Big(\mathbf{u}^n_{j-\frac12,k-\frac12}\Big)+\mathbf{g}\Big(\mathbf{u}^{n-1}_{j+\frac32,k+\frac32}\Big)-\mathbf{g}\Big(\mathbf{u}^{n-1}_{j+\frac32,k-\frac12}\Big)+\mathbf{g}\Big(\mathbf{u}^{n-1}_{j-\frac12,k+\frac32}\Big)-\mathbf{g}\Big(\mathbf{u}^{n-1}_{j-\frac12,k-\frac12}\Big)\Big]\\&+4\Big[\mathbf{g}\Big(\mathbf{u}^n_{j+\frac12,k+\frac32}\Big)-\mathbf{g}\Big(\mathbf{u}^n_{j+\frac12,k-\frac12}\Big)+\mathbf{g}\Big(\mathbf{u}^{n-1}_{j+\frac12,k+\frac32}\Big)-\mathbf{g}\Big(\mathbf{u}^{n-1}_{j+\frac12,k-\frac12}\Big)\Big].\end{aligned}$$

Notice that as in the 1-D case, piecewise polynomial reconstructions used to compute point values of finite volume solutions are typically discontinuous, four (different) point values are available at each cell corner $(x_{j+\frac12},y_{k+\frac12})$. We can use any of these values or their convex combination to compute the WLR in (2.19). In our numerical experiments, we have used the value in the northeast corner of cell (j,k).

It is easy to show (see [19]; notice however that here, we have rescaled the 2-D WLRs by a factor of $1/\Delta$) that the size of WLRs is proportional to Δ near shocks (in poorly resolved) regions, while in the smooth (highly resolved) parts of the computed solution the WLRs are several orders of magnitude smaller. Our numerical experiments indicate that typically

$$\|\mathbf{E}^{n-\frac12}_{j,k}\|_\infty\sim\begin{cases}\Delta, & \text{near shock waves},\\ \Delta^\alpha, & \text{near contact waves}, \quad 1<\alpha\leqslant 2,\\ \Delta^p, & \text{in smooth regions},\end{cases}$$

where $p=\min\{r+2,4\}$ and r is the formal order of accuracy of a scheme.

Remark 2.2. Notice that our 1-D WLR (2.8) is different from the one developed in [18], and the 2-D WLR (2.19) is different from the one proposed in [19] not only by a factor of $1/\Delta$. Here, the size of the support of the localized test functions is $3\Delta x\times 2\Delta t$ ($3\Delta x\times 3\Delta y\times 2\Delta t$ for 2-D), while the WLRs in [18,19] were obtained using the localized test functions with a larger ($3\Delta x\times 3\Delta t$ and $3\Delta x\times 3\Delta y\times 3\Delta t$, respectively) support. We would like to point out that the original version of the WLRs are formally more accurate and they can be used in our adaptive artificial viscosity method provided the time step Δt remains fixed. However, the current version of the WLRs allows one to vary Δt according to the CFL condition and it is thus more efficient. Also, the use of data from only two time levels instead of the three ones reduces the computational storage requirements of the method. On the other hand, we have conducted a large number of numerical experiment that clearly indicate that the difference in the achieved resolution is negligible.

2.2.2. Two-dimensional adaptive artificial viscosity method

In 2-D, we assume that the approximate solution $\bar u_{j,k}$ and the WLRs $E_{j+\frac12,k+\frac12}$ are known at certain time level t, and then we evolve the solution according to the 2-D semi-discrete scheme with linearly stable numerical fluxes $\mathbf{H}^x_{j+\frac12,k}$ and $\mathbf{H}^y_{j,k+\frac12}$:

$$\frac{d}{dt}\bar{\mathbf{u}}_{j,k}=-\frac{\mathbf{H}^x_{j+\frac12,k}-\mathbf{H}^x_{j-\frac12,k}}{\Delta x}-\frac{\mathbf{H}^y_{j,k+\frac12}-\mathbf{H}^y_{j,k-\frac12}}{\Delta y}+C\left(\frac{\varepsilon^x_{j+\frac12,k}\Delta\bar{\mathbf{u}}_{j+\frac12,k}-\varepsilon^x_{j-\frac12,k}\Delta\bar{\mathbf{u}}_{j-\frac12,k}}{(\Delta x)^2}+\frac{\varepsilon^y_{j,k+\frac12}\Delta\bar{\mathbf{u}}_{j,k+\frac12}-\varepsilon^y_{j,k-\frac12}\Delta\bar{\mathbf{u}}_{j,k-\frac12}}{(\Delta y)^2}\right).\tag{2.20}$$

Here, C is as before a positive constant to be tuned,

$$\bar{\mathbf{u}}_{j,k}\approx\frac{1}{\Delta x\Delta y}\int_{y_{k-\frac12}}^{y_{k+\frac12}}\int_{x_{j-\frac12}}^{x_{j+\frac12}}\mathbf{u}(x,y,t)\,dx\,dy$$

is a computed cell average, $\Delta\bar{\mathbf{u}}_{j+\frac12,k}:=\bar{\mathbf{u}}_{j+1,k}-\bar{\mathbf{u}}_{j,k}$ and $\Delta\bar{\mathbf{u}}_{j,k+\frac12}:=\bar{\mathbf{u}}_{j,k+1}-\bar{\mathbf{u}}_{j,k}$.

As in the 1-D case, the key idea of the adaptive artificial viscosity method is to take $\varepsilon^x_{j+\frac12,k}$ and $\varepsilon^y_{j,k+\frac12}$ to be proportional to WLRs. To guarantee that enough artificial numerical viscosity is applied to rough parts of the solution, we take

$$\begin{aligned}\varepsilon^x_{j+\frac{1}{2},k} &= \max_{j,k}\Big\{|\mathbf{E}^{(i)}_{j-\frac{1}{2},k-\frac{1}{2}}|,|\mathbf{E}^{(i)}_{j+\frac{1}{2},k-\frac{1}{2}}|,|\mathbf{E}^{(i)}_{j+\frac{3}{2},k-\frac{1}{2}}|,|\mathbf{E}^{(i)}_{j-\frac{1}{2},k+\frac{1}{2}}|,|\mathbf{E}^{(i)}_{j+\frac{1}{2},k+\frac{1}{2}}|,|\mathbf{E}^{(i)}_{j+\frac{3}{2},k+\frac{1}{2}}|\Big\},\\ \varepsilon^y_{j,k+\frac{1}{2}} &= \max_{j,k}\Big\{|\mathbf{E}^{(i)}_{j-\frac{1}{2},k-\frac{1}{2}}|,|\mathbf{E}^{(i)}_{j-\frac{1}{2},k+\frac{1}{2}}|,|\mathbf{E}^{(i)}_{j-\frac{1}{2},k+\frac{3}{2}}|,|\mathbf{E}^{(i)}_{j+\frac{1}{2},k-\frac{1}{2}}|,|\mathbf{E}^{(i)}_{j+\frac{1}{2},k+\frac{1}{2}}|,|\mathbf{E}^{(i)}_{j+\frac{1}{2},k+\frac{3}{2}}|\Big\},\end{aligned} \tag{2.21}$$

where $\mathbf{E}^{(i)}_{j+\frac{1}{2},k+\frac{1}{2}}$ is the ith component of $\mathbf{E}_{j+\frac{1}{2},k+\frac{1}{2}}$. In general, one can choose any component i to compute $\varepsilon^x_{j+\frac{1}{2},k}$ and $\varepsilon^y_{j,k+\frac{1}{2}}$ in (2.21), but some components might be superior depending on a problem at hand. One should choose the component that can pick up the discontinuous part of the solution.

Finally, C in (2.20) is a positive viscosity coefficient, which should be selected to control the quality of the computed solution. To optimize the selection of C, we first tune C on a coarse mesh and then use it for finer mesh computations.

Remark 2.3. The resulting ODE systems (2.9) and (2.20) should be solved by a stable and sufficiently accurate ODE solver. In the numerical examples reported in Section 3.3, we have used the third-order SSP Runge–Kutta method from [8].

Remark 2.4. Since the formulae (2.8) and (2.19) require the computed solution at both current and previous time levels, the adaptive artificial viscosity method can only be used from the second time step. To obtain a solution at the first time step, one can use a high-resolution method, which is stabilized with the help of a nonlinear limiter.

3. Numerical examples

In this section, we illustrate the performance of our adaptive artificial numerical viscosity method on several 1-D and 2-D examples. We use the central-upwind fluxes and unlimited high-order piecewise polynomial reconstructions described in Appendices A and B. We use the first order extrapolation for the boundary. We compare our results with the results obtained by the same central-upwind fluxes, but with the fifth-order componentwise WENO reconstruction from [2,37,38] and without the adaptive artificial viscosity.

In all of the examples below, we have used a uniform spatial mesh with $\Delta x = \Delta y = h$. We use the ghost cell technique to obtain the absorbing boundary conditions needed to solve initial value problems on finite computational domains. All of the required ghost values have been obtained using the simplest zero-order extrapolation. The CFL number has been taken in the range of $0.45 - 0.5$. When the equation of gas dynamics are solved, the viscosity coefficients ε in (2.9) and (2.20) are computed based on the first (density) component of the WLRs. Our choice is motivated by the fact that in the density equation the flux is simply ρu (see equation Eq. (3.1) below) and thus the computational cost of the resulting method is minimized.

The viscosity coefficient C is tuned in every numerical example below according to the strategy described on page 8. For brievity of the presentation, the tuning process is shown in Example 3.1 only.

3.1. Convergence test

We use the 2-D transport equation to demonstrate convergence of the proposed adaptive artificial viscosity method. We consider the following initial-boundary value problem:

$$\begin{cases} u_t + 2u_x + u_y = 0,\\ u(x,y,0) = \sin(\pi(x+y)) + \frac{1}{2}\cos(2\pi(x-y)),\\ u(x,y,t) = u(x+2,y,t) = u(x,y+2,t).\end{cases}$$

We fix the time step to be either $\Delta t = 0.25h$ or $\Delta t = h^2$, compute the corresponding numerical solutions at time $t = 2$ and measure their L^∞-errors:

$$e_\infty := \max_{j,k}\left|\overline{u}_{j,k}(2) - \frac{1}{h^2}\int_{y_{k-\frac{1}{2}}}^{y_{k+\frac{1}{2}}}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} u(x,y,0)\,dx\,dy\right|.$$

The results are presented in Table 1. Clearly, if $\Delta t = 0.25h$, which corresponds to the CFL number 0.5, the time discretization errors dominate and our adaptive artificial viscosity method exhibits the third-order convergence. When $\Delta t = h^2$, then

Table 1
L^∞-errors and experimental rates of convergence.

h	Δt (h)	e_∞	Order	Δt	e_∞	Order
2/50	0.25	1.042E−3	–	h^2	3.245E−4	–
2/100	0.25	1.004E−4	3.38	h^2	1.026E−5	4.98
2/200	0.25	1.161E−5	3.11	h^2	3.241E−7	4.99
2/400	0.25	1.421E−6	3.03	h^2	1.044E−8	4.96

the spatial errors dominate, and the experimental order of convergence increases to the fifth one (even though our 2-D method is formally fourth-order only, see Appendix B).

3.2. One-dimensional Euler equations of gas dynamics

We numerically solve the Euler equations of gas dynamics:

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho u\\ E\end{pmatrix}+\frac{\partial}{\partial x}\begin{pmatrix}\rho u\\ \rho u^2+p\\ u(E+p)\end{pmatrix}=\mathbf{0}. \tag{3.1}$$

Here, ρ, u, p and E are the density, the velocity, the pressure and the total energy, respectively. The equation of state is $p=(\gamma-1)\cdot\left(E-\frac{1}{2}\rho u^2\right)$ and we take $\gamma=1.4$.

Example 3.1. We solve the Euler Eq. (3.1) subject to the following Riemann initial data [43]:

$$(\rho,u,p)=\begin{cases}(5.99924,19.5975,460.894), & \text{for } x<0.4,\\ (5.99242,-6.19633,46.0950), & \text{for } x>0.4.\end{cases}$$

The exact self-similar solution consists of a left facing shock traveling slowly to the right, a contact wave moving to the right, and a right traveling shock wave. We use our adaptive artificial viscosity method to compute the solution at time $t=0.035$ with $h=1/200$. The obtained results are compared with the exact solution, see [43].

In this example, we take different values of the viscosity coefficient C to demonstrate dependence of the adaptive artificial viscosity method on this parameter. We begin with $C=0$. The obtained density is shown on the top left of Fig. 3.1. As one can see, even though the computed density is very oscillatory, the contact wave is quite sharply resolved. We then increase C and take $C=5,10,15,30$ and 50. From the results, shown in the rest of Fig. 3.1, we may conclude that $C=10$ seems to be an optimal value.

We then verify that $C=10$ also works on finer grids. To this end, we refine the mesh to $h=1/400,1/800,1/1600,1/3200$ and show the obtained results in Fig. 3.2. As one can clearly observe, the numerical solution converges to the exact one.

Finally, we compare our method with the central-upwind scheme with the fifth-order WENO5 reconstruction. The density computed by WENO5 with $h=1/200$ is plotted in Fig. 3.3. It seems that our result is as good as the result obtained by WENO5, but our method is less computationally expensive (see Section 3.4).

Example 3.2 (*Lax problem, [26]*). We numerically solve (3.1) with another set of Riemann initial data:

$$(\rho,u,p)=\begin{cases}(0.445,0.698,3.528), & \text{for } x<0.5,\\ (0.5,0,0.571), & \text{for } x>0.5.\end{cases}$$

We take $h=1/200$ and compute the solution at time $t=0.16$ with both the adaptive artificial viscosity method and the central-upwind scheme with the WENO5 reconstruction.

The density component of the computed solutions is plotted in Fig. 3.4. As one can clearly see, in this example, the adaptive artificial viscosity method produces a sharper and yet less oscillatory results.

The viscosity coefficient was optimized on a coarse grid and was selected to be $C=45$. In Fig. 3.5, we show that this viscosity coefficient leads to satisfactory (though a little oscillatory) results on finer meshes as well.

Example 3.3 (*Toro's "123 problem" [43]*). In this example, the Riemann initial data are

$$(\rho,u,p)=\begin{cases}(1.0,-2.0,0.4), & \text{for } x<0.5,\\ (1.0,2.0,0.4), & \text{for } x>0.5.\end{cases}$$

The exact solution consists of two strong symmetric rarefaction waves and a trivial stationary contact discontinuity. Between the two rarefaction waves, the density is very small (close to vacuum), which makes it a suitable test for assessing numerical methods for low-density flows.

In this example, the central-upwind scheme with the WENO5 reconstruction fails (because of the appearance of negative values of computed density and/or pressure), while the adaptive artificial viscosity method performs quite well. We take $h=1/200,\ 1/400,\ 1/800$ and $1/1600$ and compute the solution at time $t=0.15$. As in the previous examples, the viscosity coefficient is tuned on the coarse grid and is taken as $C=15$. In Fig. 3.6, one can clearly see that this value of C leads to accurate results on finer grids as well.

3.3. Two-dimensional Euler equations of gas dynamics

We now consider the 2-D Euler equations of gas dynamics:

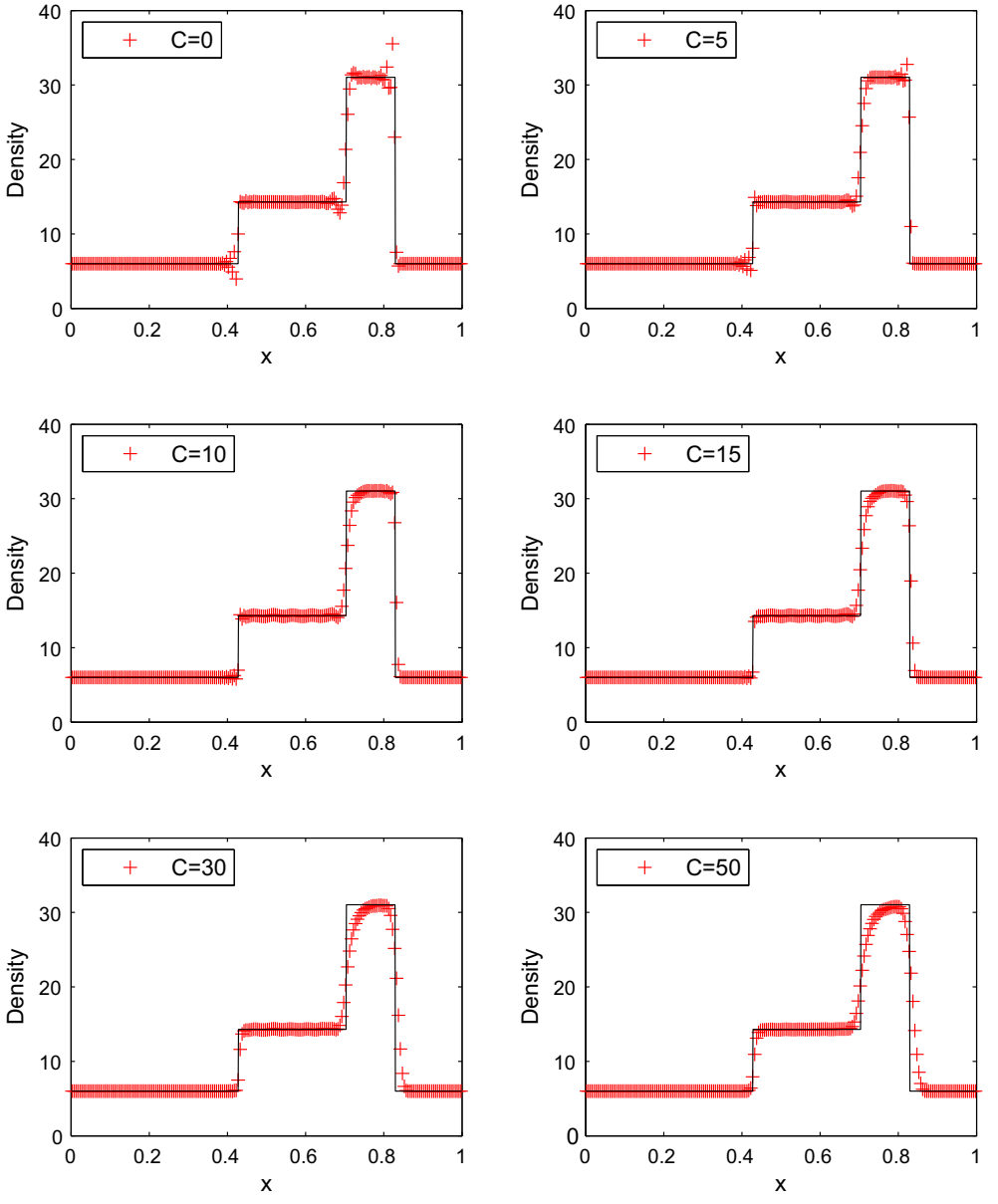

Fig. 3.1. Example 3.1: Density computed by the adaptive artificial viscosity method with different values of C. Solid line represents the exact solution.

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ E\end{pmatrix}+\frac{\partial}{\partial x}\begin{pmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ u(E+p)\end{pmatrix}+\frac{\partial}{\partial y}\begin{pmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ v(E+p)\end{pmatrix}=0. \tag{3.2}$$

Here, ρ, u, v, p and E are the density, the x- and y-velocities, the pressure and the total energy, respectively. The system is closed using the equation of state for an ideal gas, $p=(\gamma-1)\cdot\left[E-\frac{\rho}{2}(u^2+v^2)\right]$, and we take $\gamma=1.4$. In our numerical examples, the computational domain is $[0,1]\times[0,1]$, and we use a uniform 400×400 grid.

We test the proposed adaptive artificial viscosity method on 2-D Riemann initial data, which have the following general form:

$$(p,\rho,u,v)(x,y,0)=\begin{cases}(p_1,\rho_1,u_1,v_1), & \text{if} \quad x>0.5 \quad \text{and} \quad y>0.5,\\ (p_2,\rho_2,u_2,v_2), & \text{if} \quad x<0.5 \quad \text{and} \quad y>0.5,\\ (p_3,\rho_3,u_3,v_3), & \text{if} \quad x<0.5 \quad \text{and} \quad y<0.5,\\ (p_4,\rho_4,u_4,v_4), & \text{if} \quad x>0.5 \quad \text{and} \quad y<0.5.\end{cases}$$

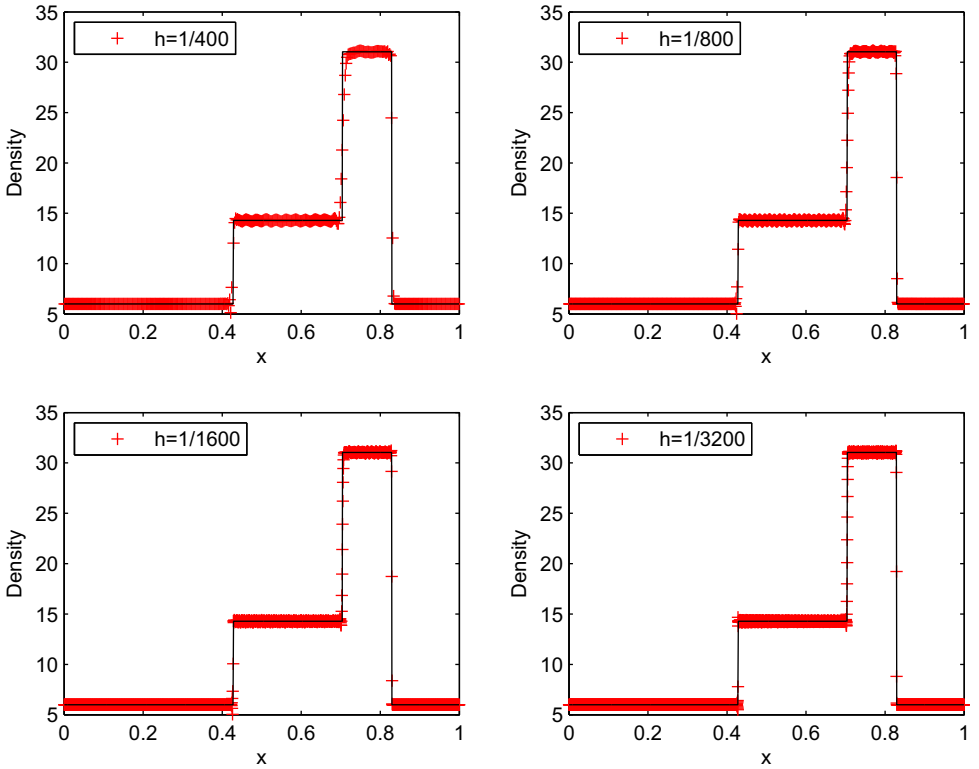

Fig. 3.2. Example 3.1: Density computed by the adaptive artificial viscosity method with $C = 10$ on different grids. Solid line represents the exact solution.

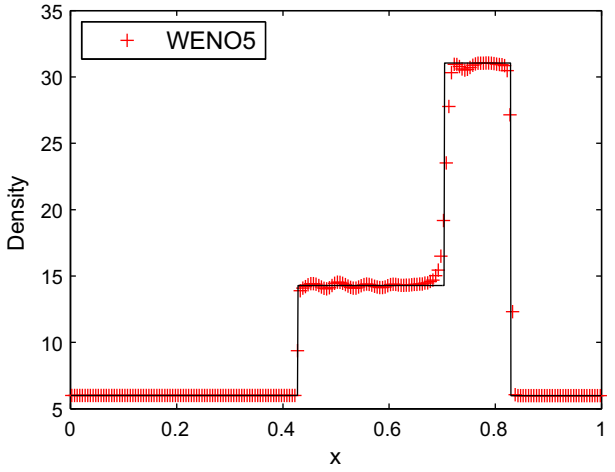

Fig. 3.3. Example 3.1: Density computed by the central-upwind scheme with WENO5 reconstruction ($h = 1/200$). The solid line represents the exact solution.

There are 19 genuinely different configurations separated by three types of 1-D centered waves, which are rarefaction, shock and contact waves (consult [33,34] for details). We will only show the results for the following two configurations only:

Configuration 3, [25] (Four 1-D shock waves):

$$p_2 = 0.3 \quad \rho_2 = 0.5323 \quad p_1 = 1.5 \quad \rho_1 = 1.5$$
$$u_2 = 1.206 \quad v_2 = 0 \quad u_1 = 0 \quad v_1 = 0$$
$$p_3 = 0.029 \quad \rho_3 = 0.138 \quad p_4 = 0.3 \quad \rho_4 = 0.5323$$
$$u_3 = 1.206 \quad v_3 = 1.206 \quad u_4 = 0 \quad v_4 = 1.206$$

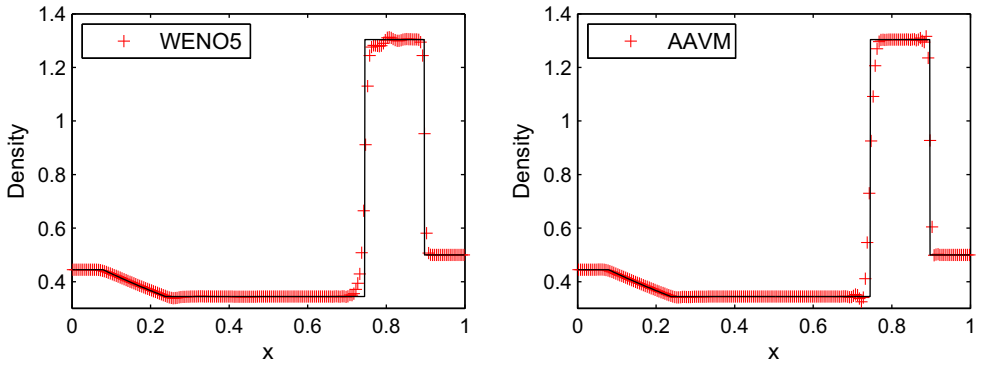

Fig. 3.4. Example 3.2: Density computed by the central-upwind scheme with WENO5 reconstruction, left, and the adaptive artificial viscosity method (AAVM), right. Solid line represents the exact solution.

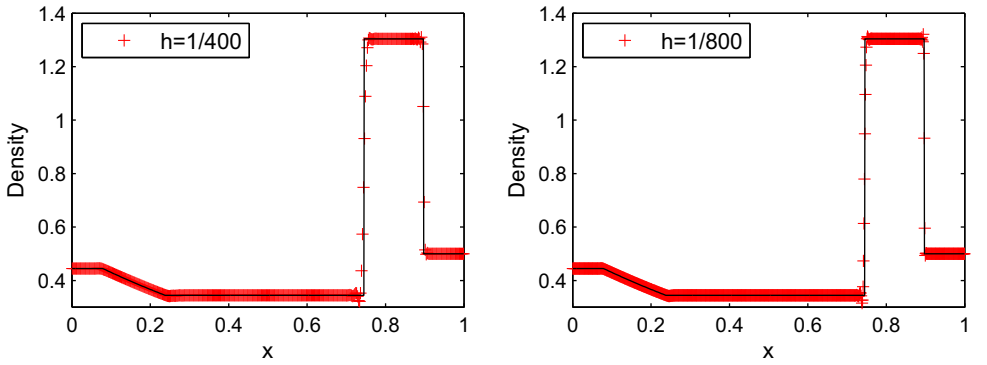

Fig. 3.5. Example 3.2: Density computed by the adaptive artificial viscosity method on different grids. Solid line represents the exact solution.

Configuration 19, [25] (One shock, two contacts and one rarefaction wave):

$$p_2 = 1 \quad \rho_2 = 2 \quad p_1 = 1 \quad \rho_1 = 1$$
$$u_2 = 0 \quad v_2 = -0.3 \quad u_1 = 0 \quad v_1 = 0.3$$
$$p_3 = 0.4 \quad \rho_3 = 1.0625 \quad p_4 = 0.4 \quad \rho_4 = 0.5197$$
$$u_3 = 0 \quad v_3 = 0.2145 \quad u_4 = 0 \quad v_4 = -0.4259$$

We test the proposed 2-D adaptive artificial viscosity method and compare the obtained results with those computed with the help of the central-upwind scheme coupled with the 2-D WENO5 reconstruction performed in the dimension-by-dimension manner (see [37] for the reconstruction details).

Figs. 3.7 (Configuration 3) and 3.9 (Configuration 19) show the contour plots of the density (left) and the corresponding values of the WLR (right) computed at time $t = 0.3$ by our adaptive artificial viscosity method. The WENO5 results are shown in Figs. 3.8 and 3.10, respectively. As one can see, the adaptive artificial viscosity method achieves much better resolution than the central-upwind scheme with the WENO5 reconstruction. Moreover, our method is more efficient than the WENO5 approach (see Section 3.4).

We note that as in the 1-D numerical examples, the viscosity coefficient C was tuned on a coarse grid (with 50×50 uniform cells) and was then selected to be equal to 1 (Configuration 3) or 2 (Configuration 19).

3.4. Efficiency test

We now check the efficiency of the adaptive artificial viscosity method by comparing its computational cost with the cost of the central-upwind scheme with the fifth-order WENO5 reconstruction.

We first take the 1-D Example 3.1 and run both codes with a fixed spatial grid ($\Delta x = 1/200$) and fixed time steps ($\Delta t = \Delta x/80$). The CPU times are 0.227 s for the adaptive artificial viscosity method and 0.542 s for the central-upwind scheme with the WENO5 reconstruction. As one can see, the adaptive artificial viscosity method is about 58.1% faster.

We then proceed with the 2-D Example (*Configuration* 3 from Section 3.3) and once again use a fixed spatial grid ($h = \Delta x = \Delta y = 1/200$) and fixed time steps ($\Delta t = h/12$). The CPU times are 646.974 s for the adaptive artificial viscosity

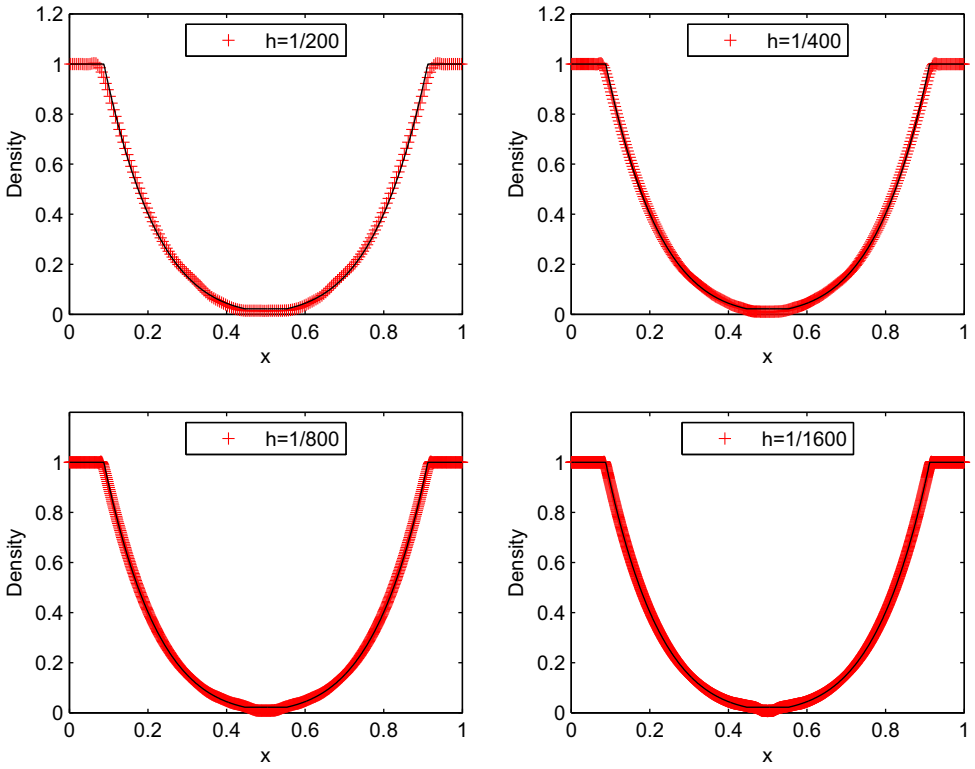

Fig. 3.6. Example 3.3: Density computed by the adaptive artificial viscosity method with $C = 15$. Solid line represents the exact solution.

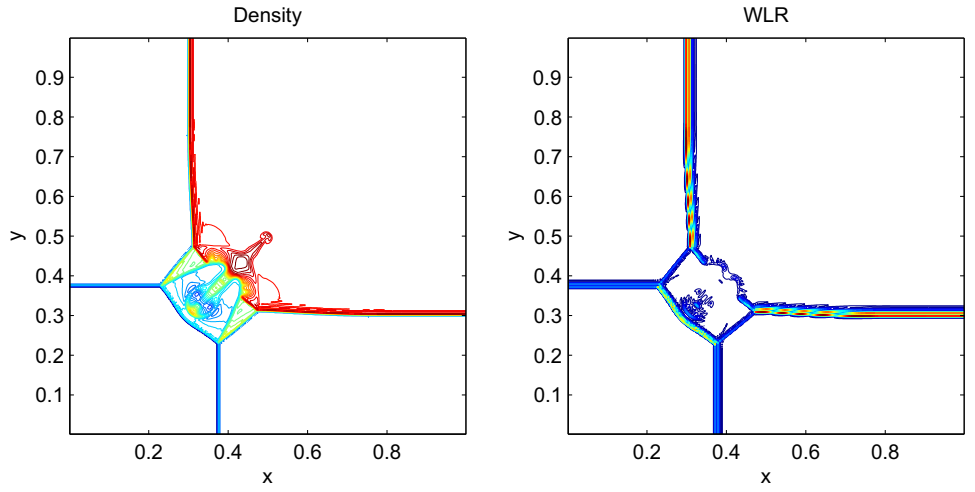

Fig. 3.7. Configuration 3: Contour plots (32 contours) of the density computed by the adaptive artificial viscosity method with $C = 1$ (left) and the corresponding WLR (right).

method and 1408.503 s for the central-upwind scheme with the WENO5 reconstruction. Thus, in this 2-D example, the adaptive artificial viscosity method is about 54.1% faster.

We note that the adaptive artificial viscosity method has to be first tested on a coarser grid to optimize the viscosity coefficient C, which increases the total computational cost of the adaptive artificial viscosity method. However, if the coarse grid is much coarser than the fine one, the additional computational cost is very small. For instance, in the 2-D case, if the fine grid is with $\Delta x = \Delta y = 1/400$ and the coarse one is with $\Delta x = \Delta y = 1/50$ (as it was the case in the examples brought in Section 3.3), then each coarse grid experiment will only add about $(1/8)^3 = 1/512$ of the total computational time and say, 10 experiments needed to optimize C will only increase the total computational cost of the adaptive artificial viscosity method by about 2%.

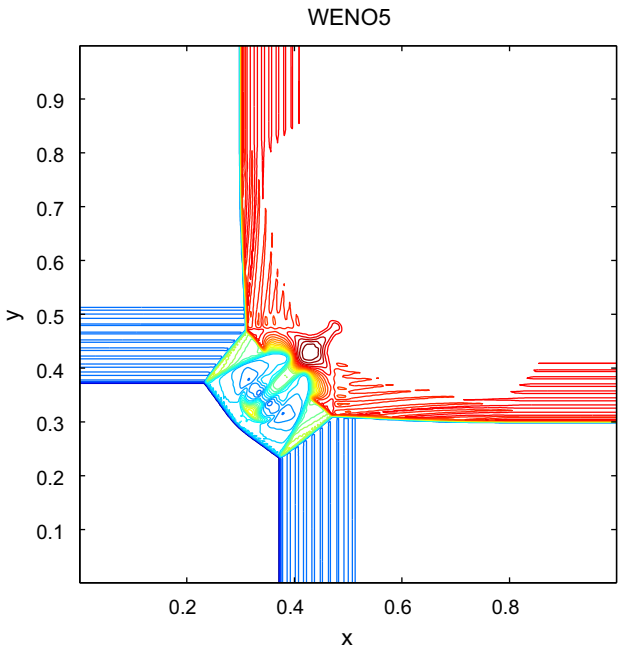

Fig. 3.8. Configuration 3: Contour plot (32 contours) of the density computed by the central-upwind scheme with the WENO5 reconstruction.

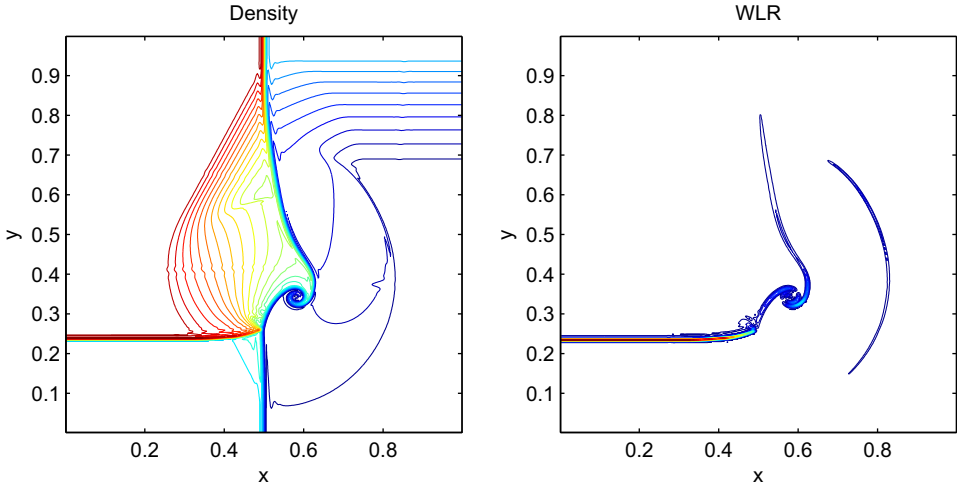

Fig. 3.9. Configuration 19: Contour plots (29 contours) of the density computed by the adaptive artificial viscosity method with $C = 2$ (left), and the corresponding WLR (right).

Appendix A. One-dimensional semi-discrete central-upwind scheme

In this section, we give a brief description of the Godunov-type central-upwind scheme for 1-D hyperbolic systems of conservation laws (1.1), and we refer readers to [20–22,24] for the derivation of the scheme and more details.

The semi-discrete central-upwind scheme is obtained by using the central-upwind flux,

$$\mathbf{H}_{j+\frac{1}{2}}(t)=\frac{a^+_{j+\frac{1}{2}}\mathbf{f}(\mathbf{u}^-_{j+\frac{1}{2}})-a^-_{j+\frac{1}{2}}\mathbf{f}(\mathbf{u}^+_{j+\frac{1}{2}})}{a^+_{j+\frac{1}{2}}-a^+_{j+\frac{1}{2}}}+a^+_{j+\frac{1}{2}}a^-_{j+\frac{1}{2}}\left[\frac{\mathbf{u}^+_{j+\frac{1}{2}}-\mathbf{u}^-_{j+\frac{1}{2}}}{a^+_{j+\frac{1}{2}}-a^-_{j+\frac{1}{2}}}-\mathbf{q}_{j+\frac{1}{2}}\right], \tag{A.1}$$

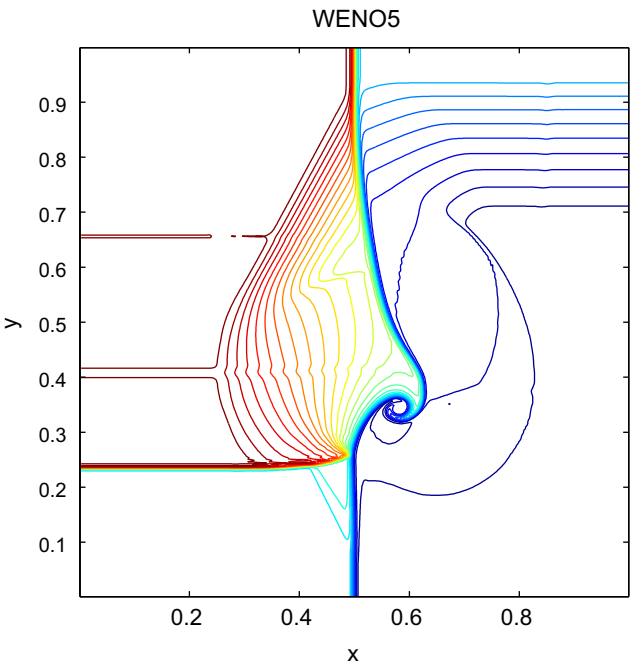

Fig. 3.10. Configuration 19: Contour plot (29 contours) of the density computed by the central-upwind scheme with the WENO5 reconstruction.

In (A.1), $\mathbf{u}^{\pm}_{j+\frac{1}{2}}$ denote the right- and left-sided values of the piecewise polynomial reconstruction $\widetilde{\mathbf{u}}(\cdot,t)=\sum_j \mathbf{p}_j(\cdot,t)\chi_j(\cdot)$ at the cell interface $x=x_{j+\frac{1}{2}}$, namely, $\mathbf{u}^{+}_{j+\frac{1}{2}}=\mathbf{p}_{j+1}(x_{j+\frac{1}{2}})$ and $\mathbf{u}^{-}_{j+\frac{1}{2}}=\mathbf{p}_j(x_{j+\frac{1}{2}})$. Here, χ_j is a characteristic function of the cell $I_j=(x_{j-\frac{1}{2}},x_{j+\frac{1}{2}})$ and $\mathbf{p}_j$ are polynomial pieces reconstructed from the cell averages $\{\bar{\mathbf{u}}_j(t)\}$, available at time level t. The reconstruction, performed in a componentwise manner, must be conservative and sufficiently accurate. In our numerical examples, we have used a fifth-order piecewise polynomial reconstruction consisting of fourth-degree polynomial pieces $\mathbf{p}_j(x)$, designed to satisfy the following conservation properties:

$$\frac{1}{\Delta x}\int_{I_j}\mathbf{p}_j(x)\,dx=\bar{\mathbf{u}}_j,\quad \frac{1}{\Delta x}\int_{I_{j\pm1}}\mathbf{p}_j(x)\,dx=\bar{\mathbf{u}}_{j\pm1},\quad \frac{1}{\Delta x}\int_{I_{j\pm2}}\mathbf{p}_j(x)\,dx=\bar{\mathbf{u}}_{j\pm2}.$$

The one-sided point values of this reconstruction at $x_{j+\frac{1}{2}}$ are

$$\mathbf{u}^{+}_{j+\frac{1}{2}}=\frac{1}{60}\left(-3\bar{\mathbf{u}}_{j-1}+27\bar{\mathbf{u}}_j+47\bar{\mathbf{u}}_{j+1}-13\bar{\mathbf{u}}_{j+2}+2\bar{\mathbf{u}}_{j+3}\right),$$
$$\mathbf{u}^{-}_{j+\frac{1}{2}}=\frac{1}{60}\left(2\bar{\mathbf{u}}_{j-2}-13\bar{\mathbf{u}}_{j-1}+47\bar{\mathbf{u}}_j+27\bar{\mathbf{u}}_{j+1}-3\bar{\mathbf{u}}_{j+2}\right).$$

In the case of a convex flux (a central-upwind scheme for the systems with non-convex fluxes was developed in [23]), the one-sided local speeds of propagation used in (A.1) can be estimated by

$$\begin{aligned}a^{+}_{j+\frac{1}{2}}&=\max\left\{\lambda_N\left(\frac{\partial\mathbf{f}}{\partial\mathbf{u}}\left(\mathbf{u}^{-}_{j+\frac{1}{2}}\right)\right),\lambda_N\left(\frac{\partial\mathbf{f}}{\partial\mathbf{u}}\left(\mathbf{u}^{+}_{j+\frac{1}{2}}\right)\right),0\right\},\\ a^{-}_{j+\frac{1}{2}}&=\min\left\{\lambda_1\left(\frac{\partial\mathbf{f}}{\partial\mathbf{u}}\left(\mathbf{u}^{-}_{j+\frac{1}{2}}\right)\right),\lambda_1\left(\frac{\partial\mathbf{f}}{\partial\mathbf{u}}\left(\mathbf{u}^{+}_{j+\frac{1}{2}}\right)\right),0\right\},\end{aligned}\tag{A.2}$$

where, $\lambda_1<\lambda_2<\cdots<\lambda_N$ are the N eigenvalues of the Jacobian $\frac{\partial\mathbf{f}}{\partial\mathbf{u}}$.

Finally, the built-in anti-diffusion term $\mathbf{q}_{j+1/2}$ in (A.1) is given by (see [20] for its derivation)

$$\mathbf{q}_{j+\frac{1}{2}}=\frac{1}{a^{+}_{j+\frac{1}{2}}-a^{-}_{j+\frac{1}{2}}}\operatorname{minmod}\left(\mathbf{u}^{+}_{j+\frac{1}{2}}-\mathbf{u}^{*}_{j+\frac{1}{2}},\mathbf{u}^{*}_{j+\frac{1}{2}}-\mathbf{u}^{-}_{j+\frac{1}{2}}\right),\tag{A.3}$$

where the minmod function, defined as

$$\operatorname{minmod}(z_1,z_2)=\frac{\operatorname{sgn}(z_1)+\operatorname{sgn}(z_2)}{2}\min(|z_1|,|z_2|)$$

is applied in (A.3) in a componentwise manner, and the intermediate value $\mathbf{u}^*_{j+\frac12}$ is given by

$$\mathbf{u}^*_{j+\frac12}=\frac{a^+_{j+\frac12}\mathbf{u}^+_{j+\frac12}-a^-_{j+\frac12}\mathbf{u}^-_{j+\frac12}-\left\{\mathbf{f}\left(\mathbf{u}^+_{j+\frac12}\right)-\mathbf{f}\left(\mathbf{u}^-_{j+\frac12}\right)\right\}}{a^+_{j+\frac12}-a^-_{j+\frac12}}. \tag{A.4}$$

Remark 3.1. A fully discrete central-upwind scheme will be obtained by solving the ODE system (1.3), (A.1) by a stable ODE solver of an appropriate order. In the reported numerical examples, we have used the third-order strong stability preserving Runge–Kutta solver ([8]).

Remark 3.2. Notice that the (formal) order of the scheme (1.3), (A.1)–(A.4) depends on the order of the piecewise polynomial reconstruction used to evaluate $\mathbf{u}^\pm_{j+\frac12}$ and on the order of the ODE solver used to integrate (1.3), see [21,20,22].

Remark 3.3. Note that the quantities $\mathbf{u}^\pm_{j+\frac12}$, $\mathbf{p}_j$, $a^\pm_{j+\frac12}$, $\mathbf{q}^\pm_{j+\frac12}$ and $\mathbf{u}^*_{j+\frac12}$ used above, depend on t, but we have suppressed this dependence to simplify the notation.

Appendix B. Two-dimensional semi-discrete central-upwind scheme

In this section, we briefly describe the 2-D semi-discrete central-upwind scheme from [21]. Its derivation and more detailed description can be found in [20–22].

Let us consider the 2-D hyperbolic systems of conservation law (1.2) and let $\bar{\mathbf{u}}_{j,k}$ denote the computed cell averages of $\mathbf{u}$ at time t

$$\bar{\mathbf{u}}_{j,k}\approx\frac{1}{\Delta x\Delta y}\int_{x_{j-\frac12}}^{x_{j+\frac12}}\int_{y_{k-\frac12}}^{y_{k+\frac12}}\mathbf{u}(x,y,t)\,dy\,dx.$$

The 2-D semi-discrete central-upwind scheme from [21] has the following flux form:

$$\frac{d}{dt}\bar{\mathbf{u}}_{j,k}(t)=-\frac{\mathbf{H}^x_{j+\frac12,k}(t)-\mathbf{H}^x_{j-\frac12,k}(t)}{\Delta x}-\frac{\mathbf{H}^y_{j,k+\frac12}(t)-\mathbf{H}^y_{j,k-\frac12}(t)}{\Delta y}, \tag{B.1}$$

where the fourth-order numerical fluxes are

$$\begin{aligned}\mathbf{H}^x_{j+\frac12,k}(t):=&\frac{a^+_{j+\frac12,k}}{6\left(a^+_{j+\frac12,k}-a^-_{j+\frac12,k}\right)}\left[\mathbf{f}(\mathbf{u}^{\mathrm{NE}}_{j,k})+4\mathbf{f}(\mathbf{u}^{\mathrm{E}}_{j,k})+\mathbf{f}(\mathbf{u}^{\mathrm{SE}}_{j,k})\right]-\frac{a^-_{j+\frac12,k}}{6\left(a^+_{j+\frac12,k}-a^-_{j+\frac12,k}\right)}\left[\mathbf{f}(\mathbf{u}^{\mathrm{NW}}_{j+1,k})+4\mathbf{f}(\mathbf{u}^{\mathrm{W}}_{j+1,k})+\mathbf{f}(\mathbf{u}^{\mathrm{SW}}_{j+1,k})\right]\\&+\frac{a^+_{j+\frac12,k}a^-_{j+\frac12,k}}{6\left(a^+_{j+\frac12,k}-a^-_{j+\frac12,k}\right)}\left[\mathbf{u}^{\mathrm{NW}}_{j+1,k}-\mathbf{u}^{\mathrm{NE}}_{j,k}+4(\mathbf{u}^{\mathrm{W}}_{j+1,k}-\mathbf{u}^{\mathrm{E}}_{j,k})+\mathbf{u}^{\mathrm{SW}}_{j+1,k}-\mathbf{u}^{\mathrm{SE}}_{j,k}\right]\end{aligned}$$

and

$$\begin{aligned}\mathbf{H}^y_{j,k+\frac12}(t):=&\frac{b^+_{j,k+\frac12}}{6\left(b^+_{j,k+\frac12}-b^-_{j,k+\frac12}\right)}\left[\mathbf{g}(\mathbf{u}^{\mathrm{NW}}_{j,k})+4\mathbf{g}(\mathbf{u}^{\mathrm{N}}_{j,k})+\mathbf{g}(\mathbf{u}^{\mathrm{NE}}_{j,k})\right]-\frac{b^-_{j,k+\frac12}}{6\left(b^+_{j,k+\frac12}-b^-_{j,k+\frac12}\right)}\left[\mathbf{g}(\mathbf{u}^{\mathrm{SW}}_{j,k+1})+4\mathbf{g}(\mathbf{u}^{\mathrm{S}}_{j,k+1})+\mathbf{g}(\mathbf{u}^{\mathrm{SE}}_{j,k+1})\right]\\&+\frac{b^+_{j,k+\frac12}b^-_{j,k+\frac12}}{6\left(b^+_{j,k+\frac12}-b^-_{j,k+\frac12}\right)}\left[\mathbf{u}^{\mathrm{SW}}_{j,k+1}-\mathbf{u}^{\mathrm{NW}}_{j,k}+4(\mathbf{u}^{\mathrm{S}}_{j,k+1}-\mathbf{u}^{\mathrm{N}}_{j,k})+\mathbf{u}^{\mathrm{SE}}_{j,k+1}-\mathbf{u}^{\mathrm{NE}}_{j,k}\right].\end{aligned}$$

Here, $\mathbf{u}^{\mathrm{E}}_{j,k}$, $\mathbf{u}^{\mathrm{W}}_{j,k}$, $\mathbf{u}^{\mathrm{N}}_{j,k}$, $\mathbf{u}^{\mathrm{S}}_{j,k}$, $\mathbf{u}^{\mathrm{NE}}_{j,k}$, $\mathbf{u}^{\mathrm{NW}}_{j,k}$, $\mathbf{u}^{\mathrm{SE}}_{j,k}$ and $\mathbf{u}^{\mathrm{SW}}_{j,k}$ are reconstructed point values along the boundary of the cell $I_{j,k}:=(x_{j-\frac12},x_{j+\frac12})\times(y_{k-\frac12},y_{k+\frac12})$:

$$\begin{aligned}&\mathbf{u}^{\mathrm{E}}_{j,k}=\widetilde{\mathbf{u}}(x_{j+\frac12},y_k),\quad \mathbf{u}^{\mathrm{W}}_{j,k}=\widetilde{\mathbf{u}}(x_{j-\frac12},y_k),\quad \mathbf{u}^{\mathrm{N}}_{j,k}=\widetilde{\mathbf{u}}(x_j,y_{k+\frac12}),\quad \mathbf{u}^{\mathrm{S}}_{j,k}=\widetilde{\mathbf{u}}(x_j,y_{k-\frac12}),\\&\mathbf{u}^{\mathrm{NE}}_{j,k}=\widetilde{\mathbf{u}}(x_{j+\frac12},y_{k+\frac12}),\quad \mathbf{u}^{\mathrm{NW}}_{j,k}=\widetilde{\mathbf{u}}(x_{j-\frac12},y_{k+\frac12}),\quad \mathbf{u}^{\mathrm{SE}}_{j,k}=\widetilde{\mathbf{u}}(x_{j+\frac12},y_{k-\frac12}),\quad \mathbf{u}^{\mathrm{SW}}_{j,k}=\widetilde{\mathbf{u}}(x_{j-\frac12},y_{k-\frac12}).\end{aligned}$$

In our numerical experiments, these point values have been obtained using a conservative unlimited fourth-order piecewise polynomial reconstruction $\widetilde{\mathbf{u}}(\cdot,\cdot,t)=\sum_{j,k}\mathbf{p}_{j,k}(\cdot,\cdot,t)\chi_{j,k}(\cdot,\cdot)$, where $\chi_{j,k}$ is a characteristic function of the cell $I_{j,k}$ and the 13 coefficients $(p_{j,k},(p_x)_{j,k},(p_y)_{j,k}$, etc.) of each polynomial piece

$$\begin{aligned}\mathbf{p}_{j,k}(x,y) &= p_{j,k} + (p_x)_{j,k}(x-x_j) + (p_y)_{j,k}(y-y_k)\\ &+\frac{1}{2}(p_{xx})_{j,k}(x-x_j)^2 + (p_{xy})_{j,k}(x-x_j)(y-y_k) + \frac{1}{2}(p_{yy})_{j,k}(y-y_k)^2\\ &+\frac{1}{6}(p_{xxx})_{j,k}(x-x_j)^3 + \frac{1}{2}(p_{xxy})_{j,k}(x-x_j)^2(y-y_k)\\ &+\frac{1}{2}(p_{xyy})_{j,k}(x-x_j)(y-y_k)^2 + \frac{1}{6}(p_{yyy})_{j,k}(y-y_k)^3\\ &+\frac{1}{24}(p_{xxxx})_{j,k}(x-x_j)^4 + \frac{1}{4}(p_{xxyy})_{j,k}(x-x_j)^2(y-y_k)^2 + \frac{1}{24}(p_{yyyy})_{j,k}(y-y_k)^4\end{aligned}$$

are calculated from the following 13 conservation requirements (see [4]):

$$\begin{aligned}&\iint_{I_{j,k}} \mathbf{p}_{j,k}(x,y)\,dx\,dy = \bar{\mathbf{u}}_{j,k}, \quad \iint_{I_{j\pm1,k\pm1}} \mathbf{p}_{j,k}(x,y)\,dx\,dy = \bar{\mathbf{u}}_{j\pm1,k\pm1},\\ &\iint_{I_{j\pm1,k}} \mathbf{p}_{j,k}(x,y)\,dx\,dy = \bar{\mathbf{u}}_{j\pm1,k}, \quad \iint_{I_{j,k\pm1}} \mathbf{p}_{j,k}(x,y)\,dx\,dy = \bar{\mathbf{u}}_{j,k\pm1},\\ &\iint_{I_{j\pm2,k}} \mathbf{p}_{j,k}(x,y)\,dx\,dy = \bar{\mathbf{u}}_{j\pm2,k}, \quad \iint_{I_{j,k\pm2}} \mathbf{p}_{j,k}(x,y)\,dx\,dy = \bar{\mathbf{u}}_{j,k\pm2}.\end{aligned}$$

A straightforward calculation shows that the reconstructed point values are

$$\begin{aligned}\mathbf{u}^{\mathrm{E}}_{j,k} &= \frac{1}{5760}(192\bar{\mathbf{u}}_{j-2,k} + 40\bar{\mathbf{u}}_{j-1,k-1} - 1328\bar{\mathbf{u}}_{j-1,k} + 40\bar{\mathbf{u}}_{j-1,k+1} + 27\bar{\mathbf{u}}_{j,k-2} - 308\bar{\mathbf{u}}_{j,k-1} + 5074\bar{\mathbf{u}}_{j,k} - 308\bar{\mathbf{u}}_{j,k+1} + 27\bar{\mathbf{u}}_{j,k+2}\\ &\quad - 80\bar{\mathbf{u}}_{j+1,k-1} + 2752\bar{\mathbf{u}}_{j+1,k} - 80\bar{\mathbf{u}}_{j+1,k+1} - 288\bar{\mathbf{u}}_{j+2,k}),\\ \mathbf{u}^{\mathrm{W}}_{j,k} &= \frac{1}{5760}(-288\bar{\mathbf{u}}_{j-2,k} - 80\bar{\mathbf{u}}_{j-1,k-1} + 2752\bar{\mathbf{u}}_{j-1,k} - 80\bar{\mathbf{u}}_{j-1,k+1} + 27\bar{\mathbf{u}}_{j,k-2} - 308\bar{\mathbf{u}}_{j,k-1} + 5074\bar{\mathbf{u}}_{j,k} - 308\bar{\mathbf{u}}_{j,k+1} + 27\bar{\mathbf{u}}_{j,k+2}\\ &\quad + 40\bar{\mathbf{u}}_{j+1,k-1} - 1328\bar{\mathbf{u}}_{j+1,k} + 40\bar{\mathbf{u}}_{j+1,k+1} + 192\bar{\mathbf{u}}_{j+2,k}),\\ \mathbf{u}^{\mathrm{N}}_{j,k} &= \frac{1}{5760}(27\bar{\mathbf{u}}_{j-2,k} + 40\bar{\mathbf{u}}_{j-1,k-1} - 308\bar{\mathbf{u}}_{j-1,k} - 80\bar{\mathbf{u}}_{j-1,k+1} + 192\bar{\mathbf{u}}_{j,k-2} - 1328\bar{\mathbf{u}}_{j,k-1} + 5074\bar{\mathbf{u}}_{j,k} + 2752\bar{\mathbf{u}}_{j,k+1} - 288\bar{\mathbf{u}}_{j,k+2}\\ &\quad + 40\bar{\mathbf{u}}_{j+1,k-1} - 308\bar{\mathbf{u}}_{j+1,k} - 80\bar{\mathbf{u}}_{j+1,k+1} + 27\bar{\mathbf{u}}_{j+2,k}),\\ \mathbf{u}^{\mathrm{S}}_{j,k} &= \frac{1}{5760}(27\bar{\mathbf{u}}_{j-2,k} - 80\bar{\mathbf{u}}_{j-1,k-1} - 308\bar{\mathbf{u}}_{j-1,k} + 40\bar{\mathbf{u}}_{j-1,k+1} - 288\bar{\mathbf{u}}_{j,k-2} + 2752\bar{\mathbf{u}}_{j,k-1} + 5074\bar{\mathbf{u}}_{j,k} - 1328\bar{\mathbf{u}}_{j,k+1} + 192\bar{\mathbf{u}}_{j,k+2}\\ &\quad - 80\bar{\mathbf{u}}_{j+1,k-1} - 308\bar{\mathbf{u}}_{j+1,k} + 40\bar{\mathbf{u}}_{j+1,k+1} + 27\bar{\mathbf{u}}_{j+2,k}),\\ \mathbf{u}^{\mathrm{NE}}_{j,k} &= \frac{1}{180}(6\bar{\mathbf{u}}_{j-2,k} + 5\bar{\mathbf{u}}_{j-1,k-1} - 34\bar{\mathbf{u}}_{j-1,k} - 10\bar{\mathbf{u}}_{j-1,k+1} + 6\bar{\mathbf{u}}_{j,k-2} - 34\bar{\mathbf{u}}_{j,k-1} + 107\bar{\mathbf{u}}_{j,k} + 71\bar{\mathbf{u}}_{j,k+1} - 9\bar{\mathbf{u}}_{j,k+2} - 10\bar{\mathbf{u}}_{j+1,k-1}\\ &\quad + 71\bar{\mathbf{u}}_{j+1,k} + 20\bar{\mathbf{u}}_{j+1,k+1} - 9\bar{\mathbf{u}}_{j+2,k}),\\ \mathbf{u}^{\mathrm{NW}}_{j,k} &= \frac{1}{180}(-9\bar{\mathbf{u}}_{j-2,k} - 10\bar{\mathbf{u}}_{j-1,k-1} + 71\bar{\mathbf{u}}_{j-1,k} + 20\bar{\mathbf{u}}_{j-1,k+1} + 6\bar{\mathbf{u}}_{j,k-2} - 34\bar{\mathbf{u}}_{j,k-1} + 107\bar{\mathbf{u}}_{j,k} + 71\bar{\mathbf{u}}_{j,k+1} - 9\bar{\mathbf{u}}_{j,k+2} + 5\bar{\mathbf{u}}_{j+1,k-1}\\ &\quad - 34\bar{\mathbf{u}}_{j+1,k} - 10\bar{\mathbf{u}}_{j+1,k+1} + 6\bar{\mathbf{u}}_{j+2,k}),\\ \mathbf{u}^{\mathrm{SE}}_{j,k} &= \frac{1}{180}(6\bar{\mathbf{u}}_{j-2,k} - 10\bar{\mathbf{u}}_{j-1,k-1} - 34\bar{\mathbf{u}}_{j-1,k} + 5\bar{\mathbf{u}}_{j-1,k+1} - 9\bar{\mathbf{u}}_{j,k-2} + 71\bar{\mathbf{u}}_{j,k-1} + 107\bar{\mathbf{u}}_{j,k} - 34\bar{\mathbf{u}}_{j,k+1} + 6\bar{\mathbf{u}}_{j,k+2} + 20\bar{\mathbf{u}}_{j+1,k-1}\\ &\quad + 71\bar{\mathbf{u}}_{j+1,k} - 10\bar{\mathbf{u}}_{j+1,k+1} - 9\bar{\mathbf{u}}_{j+2,k}),\\ \mathbf{u}^{\mathrm{SW}}_{j,k} &= \frac{1}{180}(-9\bar{\mathbf{u}}_{j-2,k} + 20\bar{\mathbf{u}}_{j-1,k-1} + 71\bar{\mathbf{u}}_{j-1,k} - 10\bar{\mathbf{u}}_{j-1,k+1} - 9\bar{\mathbf{u}}_{j,k-2} + 71\bar{\mathbf{u}}_{j,k-1} + 107\bar{\mathbf{u}}_{j,k} - 34\bar{\mathbf{u}}_{j,k+1} + 6\bar{\mathbf{u}}_{j,k+2}\\ &\quad - 10\bar{\mathbf{u}}_{j+1,k-1} - 34\bar{\mathbf{u}}_{j+1,k} + 5\bar{\mathbf{u}}_{j+1,k+1} + 6\bar{\mathbf{u}}_{j+2,k}).\end{aligned} \tag{B.2}$$

Notice that one can substantially reduce the computational cost of evaluating the above point values by utilizing the symmetry and the common terms in (B.2). To this end, we define the following discrete operators:

$$
\begin{aligned}
&\sigma_1^x\bar{\mathbf{u}}_{j,k} := \bar{\mathbf{u}}_{j-1,k} + \bar{\mathbf{u}}_{j+1,k}, \quad \sigma_2^x\bar{\mathbf{u}}_{j,k} := \bar{\mathbf{u}}_{j-2,k} + \bar{\mathbf{u}}_{j+2,k},\\
&\sigma_1^y\bar{\mathbf{u}}_{j,k} := \bar{\mathbf{u}}_{j,k-1} + \bar{\mathbf{u}}_{j,k+1}, \quad \sigma_2^y\bar{\mathbf{u}}_{j,k} := \bar{\mathbf{u}}_{j,k-2} + \bar{\mathbf{u}}_{j,k+2},\\
&\sigma_1^{xy}\bar{\mathbf{u}}_{j,k} := \sigma_1^x\bar{\mathbf{u}}_{j,k} + \sigma_1^y\bar{\mathbf{u}}_{j,k}, \quad \sigma_2^{xy}\bar{\mathbf{u}}_{j,k} := \sigma_2^x\bar{\mathbf{u}}_{j,k} + \sigma_2^y\bar{\mathbf{u}}_{j,k},\\
&D_1^x\bar{\mathbf{u}}_{j,k} := \bar{\mathbf{u}}_{j+1,k} - \bar{\mathbf{u}}_{j-1,k}, \quad D_2^x\bar{\mathbf{u}}_{j,k} := \bar{\mathbf{u}}_{j+2,k} - \bar{\mathbf{u}}_{j-2,k},\\
&D_1^y\bar{\mathbf{u}}_{j,k} := \bar{\mathbf{u}}_{j,k+1} - \bar{\mathbf{u}}_{j,k-1}, \quad D_2^y\bar{\mathbf{u}}_{j,k} := \bar{\mathbf{u}}_{j,k+2} - \bar{\mathbf{u}}_{j,k-2},\\
&\sigma_d\bar{\mathbf{u}}_{j,k} := \bar{\mathbf{u}}_{j-1,k-1} + \bar{\mathbf{u}}_{j+1,k+1} + \bar{\mathbf{u}}_{j+1,k-1} + \bar{\mathbf{u}}_{j-1,k+1}
\end{aligned}
\tag{B.3}
$$

and then the following auxiliary quantities, whose dependence on j and k we omit to simplify the notation:

$$
\begin{aligned}
&C_1 := (7084\bar{\mathbf{u}}_{j,k} - 368\sigma_1^{xy}\bar{\mathbf{u}}_{j,k} + 27\sigma_2^{xy}\bar{\mathbf{u}}_{j,k} + 10\sigma_d\bar{\mathbf{u}}_{j,k})/5760,\\
&C_2 := (36D_1^x\bar{\mathbf{u}}_{j,k} - 5D_2^x\bar{\mathbf{u}}_{j,k} - D_1^x\bar{\mathbf{u}}_{j,k+1} - D_1^x\bar{\mathbf{u}}_{j,k-1})/96,\\
&C_3 := (36D_1^y\bar{\mathbf{u}}_{j,k} - 5D_2^y\bar{\mathbf{u}}_{j,k} - D_1^y\bar{\mathbf{u}}_{j+1,k} - D_1^y\bar{\mathbf{u}}_{j-1,k})/96,\\
&C_4 := (38\sigma_1^x\bar{\mathbf{u}}_{j,k} - 3\sigma_2^x\bar{\mathbf{u}}_{j,k} + 2\sigma_1^y\bar{\mathbf{u}}_{j,k} - \sigma_d\bar{\mathbf{u}}_{j,k} - 70\bar{\mathbf{u}}_{j,k})/192,\\
&C_5 := (38\sigma_1^y\bar{\mathbf{u}}_{j,k} - 3\sigma_2^y\bar{\mathbf{u}}_{j,k} + 2\sigma_1^x\bar{\mathbf{u}}_{j,k} - \sigma_d\bar{\mathbf{u}}_{j,k} - 70\bar{\mathbf{u}}_{j,k})/192,\\
&C_6 := (D_1^x\bar{\mathbf{u}}_{j,k+1} - D_1^x\bar{\mathbf{u}}_{j,k-1})/16,\\
&C_7 := (D_1^y\bar{\mathbf{u}}_{j+1,k} + D_1^y\bar{\mathbf{u}}_{j-1,k} - 2D_1^x\bar{\mathbf{u}}_{j,k})/32,\\
&C_8 := (D_1^x\bar{\mathbf{u}}_{j,k+1} + D_1^x\bar{\mathbf{u}}_{j,k-1} - 2D_1^x\bar{\mathbf{u}}_{j,k})/32,\\
&C_9 := (D_2^x\bar{\mathbf{u}}_{j,k} - 2D_1^x\bar{\mathbf{u}}_{j,k})/96, \quad C_{10} := (D_2^y\bar{\mathbf{u}}_{j,k} - 2D_1^y\bar{\mathbf{u}}_{j,k})/96,\\
&C_{11} := (4\bar{\mathbf{u}}_{j,k} - 2\sigma_1^{xy}\bar{\mathbf{u}}_{j,k} + \sigma_d\bar{\mathbf{u}}_{j,k})/64,\\
&C_{12} := (6\bar{\mathbf{u}}_{j,k} - 4\sigma_1^x\bar{\mathbf{u}}_{j,k} + \sigma_2^x\bar{\mathbf{u}}_{j,k})/384, \quad C_{13} := (6\bar{\mathbf{u}}_{j,k} - 4\sigma_1^y\bar{\mathbf{u}}_{j,k} + \sigma_2^y\bar{\mathbf{u}}_{j,k})/384.
\end{aligned}
\tag{B.4}
$$

At the end, we rewrite (B.2) as

$$
\begin{aligned}
&\mathbf{u}_{j,k}^{\mathrm{E}} = C_1 + C_2 + C_4 + C_9 + C_{12}, \mathbf{u}_{j,k}^{\mathrm{W}} = C_1 - C_2 + C_4 - C_9 + C_{12},\\
&\mathbf{u}_{j,k}^{\mathrm{N}} = C_1 + C_3 + C_5 + C_{10} + C_{13}, \mathbf{u}_{j,k}^{\mathrm{S}} = C_1 - C_3 + C_5 - C_{10} + C_{13},\\
&\mathbf{u}_{j,k}^{\mathrm{NE}} = \mathbf{u}_{j,k}^{\mathrm{E}} + C_3 + C_5 + C_6 + C_7 + C_8 + C_{10} + C_{11} + C_{13},\\
&\mathbf{u}_{j,k}^{\mathrm{SE}} = \mathbf{u}_{j,k}^{\mathrm{E}} - C_3 + C_5 - C_6 - C_7 + C_8 - C_{10} + C_{11} + C_{13},\\
&\mathbf{u}_{j,k}^{\mathrm{NW}} = \mathbf{u}_{j,k}^{\mathrm{W}} + C_3 + C_5 - C_6 + C_7 - C_8 + C_{10} + C_{11} + C_{13},\\
&\mathbf{u}_{j,k}^{\mathrm{SW}} = \mathbf{u}_{j,k}^{\mathrm{W}} - C_3 + C_5 + C_6 - C_7 - C_8 - C_{10} + C_{11} + C_{13}.
\end{aligned}
\tag{B.5}
$$

Notice that (B.3)–(B.5) is equivalent to (B.2), but the number of multiplications and divisions is now reduced from 112 to 39.

Finally, $a^{\pm}_{j+\frac{1}{2},k}$ and $b^{\pm}_{j,k+\frac{1}{2}}$ are the one-sided local propagation speeds in the x- and y-directions, respectively. In the case of convex fluxes, they can be estimated by

$$
\begin{aligned}
a^{+}_{j+\frac{1}{2},k} &:= \max\left\{\lambda_N\left(\frac{\partial\mathbf{f}}{\partial\mathbf{u}}\left(\mathbf{u}^{\mathrm{W}}_{j+1,k}\right)\right), \lambda_N\left(\frac{\partial\mathbf{f}}{\partial\mathbf{u}}\left(\mathbf{u}^{\mathrm{E}}_{j,k}\right)\right), 0\right\},\\
a^{-}_{j+\frac{1}{2},k} &:= \min\left\{\lambda_1\left(\frac{\partial\mathbf{f}}{\partial\mathbf{u}}\left(\mathbf{u}^{\mathrm{W}}_{j+1,k}\right)\right), \lambda_1\left(\frac{\partial\mathbf{f}}{\partial\mathbf{u}}\left(\mathbf{u}^{\mathrm{E}}_{j,k}\right)\right), 0\right\},\\
b^{+}_{j,k+\frac{1}{2}} &:= \max\left\{\lambda_N\left(\frac{\partial\mathbf{g}}{\partial\mathbf{u}}\left(\mathbf{u}^{\mathrm{S}}_{j,k+1}\right)\right), \lambda_N\left(\frac{\partial\mathbf{g}}{\partial\mathbf{u}}\left(\mathbf{u}^{\mathrm{N}}_{j,k}\right)\right), 0\right\},\\
b^{-}_{j,k+\frac{1}{2}} &:= \min\left\{\lambda_1\left(\frac{\partial\mathbf{g}}{\partial\mathbf{u}}\left(\mathbf{u}^{\mathrm{S}}_{j,k+1}\right)\right), \lambda_1\left(\frac{\partial\mathbf{g}}{\partial\mathbf{u}}\left(\mathbf{u}^{\mathrm{N}}_{j,k}\right)\right), 0\right\},
\end{aligned}
$$

where $\lambda_1 < \lambda_2 < \cdots < \lambda_N$ are the N eigenvalues of the Jacobians $\frac{\partial\mathbf{f}}{\partial\mathbf{u}}$ and $\frac{\partial\mathbf{g}}{\partial\mathbf{u}}$, respectively.

Remark 3.4. A fully discrete 2-D central-upwind scheme can be obtained by solving the ODE system (B.1) with one's favorite ODE solver. In this paper, we have used the third-order SSP Runge–Kutta solver [8].

Acknowledgement

The work of A. Kurganov was supported in part by the NSF Grant DMS-1115718.

References

[1] E. Caramana, M. Shashkov, P. Whalen, Formulations of artificial viscosity for multi-dimensional shock wave computations, J. Comput. Phys. 144 (1998) 70–97.

[2] B. Cockburn, C. Johnson, C.-W. Shu, E. Tadmor, Advanced numerical approximation of nonlinear hyperbolic equations, Lecture Notes in Mathematics, 1697, Springer-Verlag, Berlin, 1998.

[3] L. Constantin, A. Kurganov, Adaptive central-upwind schemes for hyperbolic systems of conservation laws, in: Hyperbolic Problems: Theory, Numerics, Applications (Osaka, 2004), Yokohama Publishers, 2006, pp. 95–103.
[4] J. Dewar, A. Kurganov, M. Leopold, New residual-based adaption indicator for compressible Euler equations and its application for scheme adaption, in preparation.
[5] K. Friedrichs, Symmetric hyperbolic linear differential equations, Commun. Pure Appl. Math. 7 (1954) 345–392.
[6] E. Godlewski, P.-A. Raviart, Numerical approximation of hyperbolic systems of conservation laws, Applied Mathematical Sciences, 118, Springer-Verlag, New York, 1996.
[7] S. Godunov, A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics, Mat. Sb. (N.S.) 47 (89) (1959) 271–306.
[8] S. Gottlieb, C.-W. Shu, E. Tadmor, Strong stability-preserving high-order time discretization methods, SIAM Rev. 43 (2001) 89–112 (electronic).
[9] J.-L. Guermond, R. Pasquetti, Entropy-based nonlinear viscosity for Fourier approximations of conservation laws, C.R. Math. Acad. Sci. Paris 346 (2008) 801–806.
[10] J.-L. Guermond, R. Pasquetti, B. Popov, Entropy viscosity method for nonlinear conservation laws, J. Comput. Phys. 230 (2011) 4248–4267.
[11] A. Harten, High resolution schemes for hyperbolic conservation laws, J. Comput. Phys. 49 (1983) 357–393.
[12] A. Harten, B. Engquist, S. Osher, S. Chakravarthy, Uniformly high-order accurate essentially nonoscillatory schemes. III, J. Comput. Phys. 71 (1987) 231–303.
[13] R. Hartmann, P. Houston, Adaptive discontinuous Galerkin finite element methods for nonlinear hyperbolic conservation laws, SIAM J. Sci. Comput. 24 (2002) 979–1004 (electronic).
[14] R. Hartmann, P. Houston, Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations, J. Comput. Phys. 183 (2002) 508–532.
[15] T. Hughes, M. Mallet, A new finite element formulation for computational fluid dynamics. IV. A discontinuity-capturing operator for multidimensional advective-diffusive systems, Comput. Methods Appl. Mech. Eng. 58 (1986) 329–336.
[16] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys. 126 (1996) 202–228.
[17] C. Johnson, A. Szepessy, P. Hansbo, On the convergence of shock-capturing streamline diffusion finite element methods for hyperbolic conservation laws, Math. Comput. 54 (1990) 107–129.
[18] S. Karni, A. Kurganov, Local error analysis for approximate solutions of hyperbolic conservation laws, Adv. Comput. Math. 22 (2005) 79–99.
[19] S. Karni, A. Kurganov, G. Petrova, A smoothness indicator for adaptive algorithms for hyperbolic systems, J. Comput. Phys. 178 (2002) 323–341.
[20] A. Kurganov, C.-T. Lin, On the reduction of numerical dissipation in central-upwind schemes, Commun. Comput. Phys. 2 (2007) 141–163.
[21] A. Kurganov, S. Noelle, G. Petrova, Semi-discrete central-upwind scheme for hyperbolic conservation laws and Hamilton–Jacobi equations, SIAM J. Sci. Comput. 23 (2001) 707–740.
[22] A. Kurganov, G. Petrova, A third-order semi-discrete genuinely multidimensional central scheme for hyperbolic conservation laws and related problems, Numer. Math. 88 (2001) 683–729.
[23] A. Kurganov, G. Petrova, B. Popov, Adaptive semi-discrete central-upwind schemes for nonconvex hyperbolic conservation laws, SIAM J. Sci. Comput. 29 (2007) 2381–2401.
[24] A. Kurganov, E. Tadmor, New high resolution central schemes for nonlinear conservation laws and convection-diffusion equations, J. Comput. Phys. 160 (2000) 241–282.
[25] A. Kurganov, E. Tadmor, Solution of two-dimensional riemann problems for gas dynamics without riemann problem solvers, Numer. Methods Partial Diff. Equat. 18 (2002) 584–608.
[26] P. Lax, Weak solutions of nonlinear hyperbolic equations and their numerical computation, Comm. Pure Appl. Math. 7 (1954) 159–193.
[27] R. LeVeque, Finite Volume Methods for Hyperbolic Problems, Cambridge Texts in Applied Mathematics, Cambridge University Press, Cambridge, 2002.
[28] D. Levy, G. Puppo, G. Russo, Central WENO schemes for hyperbolic systems of conservation laws, M2AN Math, Model. Numer. Anal. 33 (1999) 547–571.
[29] D. Levy, G. Puppo, G. Russo, Compact central WENO schemes for multidimensional conservation laws, SIAM J. Sci. Comput. 22 (2000) 656–672 (electronic).
[30] X.-D. Liu, S. Osher, T. Chan, Weighted essentially non-oscillatory schemes, J. Comput. Phys. 115 (1994) 200–212.
[31] J. Monaghan, R. Gingold, Shock simulation by the particle method SPH, J. Comput. Phys. 52 (1983) 374–389.
[32] H. Nessyahu, E. Tadmor, Nonoscillatory central differencing for hyperbolic conservation laws, J. Comput. Phys. 87 (1990) 408–463.
[33] C. Schulz-Rinne, Classification of the Riemann problem for two-dimensional gas dynamics, SIAM J. Math. Anal. 24 (1993) 76–88.
[34] C. Schulz-Rinne, J. Collins, H. Glaz, Numerical solution of the Riemann problem for two-dimensional gas dynamics, SIAM J. Sci. Comput. 14 (1993) 1394–1414.
[35] S. Serna, A. Marquina, Power ENO methods: a fifth-order accurate weighted power ENO method, J. Comput. Phys. 194 (2004) 632–658.
[36] A. Shchepetkin, J. McWilliams, Quasi-monotone advection schemes based on explicit locally adaptive dissipation, Month. Weather Rev. 126 (1998) 1541–1580.
[37] J. Shi, C. Hu, C.-W. Shu, A technique of treating negative weights in WENO schemes, J. Comput. Phys. 175 (2002) 108–127.
[38] C.-W. Shu, High order weighted essentially nonoscillatory schemes for convection dominated problems, SIAM Rev. 51 (2009) 82–126.
[39] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, J. Comput. Phys. 77 (1988) 439–471.
[40] P. Sweby, High resolution schemes using flux limiters for hyperbolic conservation laws, SIAM J. Numer. Anal. 21 (1984) 995–1011.
[41] A. Szepessy, Convergence of a shock-capturing streamline diffusion finite element method for a scalar conservation law in two space dimensions, Math. Comput. 53 (1989) 527–545.
[42] E. Tadmor, The large-time behavior of the scalar, genuinely nonlinear Lax-Friedrichs scheme, Math. Comput. 43 (1984) 353–368.
[43] E. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction, third ed., Springer-Verlag, Berlin, Heidelberg, 2009.
[44] B. van Leer, Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method, J. Comput. Phys. 32 (1979) 101–136.
[45] J. von Neumann, R. Richtmyer, A method for the numerical calculation of hydrodynamic shocks, J. Appl. Phys. 21 (1950) 232–237.
[46] M. Wilkins, Use of artificial viscosity in multidimensional fluid dynamic calculations, J. Comput. Phys. 36 (1980) 281–303.