Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 226 (2007) 204–243
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Quadrature-free non-oscillatory finite volume schemes on unstructured meshes for nonlinear hyperbolic systems

Michael Dumbser [a,b,*], Martin Käser [b], Vladimir A. Titarev [b], Eleuterio F. Toro [b]

[a] *Institut für Aerodynamik und Gasdynamik, Pfaffenwaldring 21, 70550 Stuttgart, Germany*
[b] *Laboratory of Applied Mathematics, University of Trento, Via Mesiano 77, 38050 Trento, Italy*

Received 1 December 2006; received in revised form 16 March 2007; accepted 1 April 2007
Available online 11 April 2007

Abstract

In this article we present a quadrature-free essentially non-oscillatory finite volume scheme of arbitrary high order of accuracy both in space and time for solving nonlinear hyperbolic systems on unstructured meshes in two and three space dimensions. For high order spatial discretization, a WENO reconstruction technique provides the reconstruction polynomials in terms of a hierarchical orthogonal polynomial basis over a reference element. The Cauchy–Kovalewski procedure applied to the reconstructed data yields for each element a space–time Taylor series for the evolution of the state and the physical fluxes. This Taylor series is then inserted into a special numerical flux across the element interfaces and is subsequently integrated analytically in space and time. Thus, the Cauchy–Kovalewski procedure provides a natural, direct and cost-efficient way to obtain a quadrature-free formulation, avoiding the expensive numerical quadrature arising usually for high order finite volume schemes in three space dimensions. We show numerical convergence results up to sixth order of accuracy in space and time for the compressible Euler equations on triangular and tetrahedral meshes in two and three space dimensions. Furthermore, various two- and three-dimensional test problems with smooth and discontinuous solutions are computed to validate the approach and to underline the non-oscillatory shock-capturing properties of the method.

MSC: 65M12; 76M12

PACS: 02.70.Fj; 47.11.-j

Keywords: Quadrature-free finite volume schemes; Nonlinear hyperbolic systems; WENO reconstruction; ADER approach; Unstructured meshes in 2D and 3D

DOI of original article: 10.1016/j.jcp.2006.06.043.
* Corresponding author. Tel.: +49 017 09360238; fax: +49 711 68563438.
E-mail addresses: michael.dumbser@iag.uni-stuttgart.de (M. Dumbser), martin.kaeser@ing.unitn.it (M. Käser), vladimir.titarev@ing.unitn.it (V.A. Titarev), toro@ing.unitn.it (E.F. Toro).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2007.04.004

1. Introduction

The computational effort associated with numerical quadrature in high-order finite volume schemes increases drastically with the number of space dimensions. Whereas in one space dimension the element boundaries are just the interval endpoints, i.e. zero-dimensional, they are already one-dimensional edges in two space dimensions and two-dimensional faces in the three-dimensional case. Considering that the numerical flux has also to be integrated in time, one ends up with the computation of d-dimensional space–time integrals for evaluating the inter cell fluxes in a computational domain in d-dimensional space. The necessity of numerical quadrature in multiple space dimensions leads to a considerably higher number of flux evaluations for finite volume schemes compared to, for example, conservative WENO finite difference schemes [3,31], even on purely Cartesian meshes.

If we further consider volume integrals that will arise automatically in the context of partial differential equations (PDE) with source terms, one must compute additional $d + 1$-dimensional space–time integrals over the element volumes for the source term. The problem is even much more severe for discontinuous Galerkin finite element schemes [9,11–14] because of enhanced accuracy requirements versus the numerical quadrature due to the multiplication with the test functions.

Throughout this article we intend the expression *quadrature-free* to mean the avoiding of *numerical* quadrature using some sort of *analytic* integration technique. Whereas in the context of discontinuous Galerkin schemes a quadrature-free formulation was already proposed by Atkins and Shu [2] and van der Ven and van der Vegt [53] for linear and nonlinear hyperbolic systems, this idea is still quite new for high order finite volume methods. In [17] an arbitrary high order quadrature-free non-oscillatory finite volume scheme on unstructured meshes was proposed for linear hyperbolic systems in 2D and 3D. The present article deals with the extension of this method to the case of nonlinear hyperbolic systems in two and three space dimensions.

A first attempt to partially get rid of the necessary numerical quadrature in the context of finite volume schemes for nonlinear hyperbolic systems emerged from the work on the ADER (*a*rbitrary high order *der*ivatives) approach for finite volume schemes developed originally by Toro et al. [49]. The ADER approach is based on an approximate solution of derivative Riemann problems, also called generalized Riemann problems in the literature [24,7,6], where the initial condition consists of piecewise polynomials separated by a discontinuity at the element interface in contrast to the original approach of Godunov [26], where the Riemann problem was defined by piecewise constant initial conditions. Toro et al. proposed a Taylor series in time as solution for the derivative Riemann problem. The Cauchy–Kovalewski procedure is then used, whereby all mixed space–time derivatives of the state and the physical flux are obtained by repeatedly differentiating the governing PDE. All spatial derivatives, needed as input for the Cauchy–Kovalewski procedure, are determined by the solution of Riemann problems. This procedure finally yields all the coefficients of the temporal expansion for the state and the flux.

We would like to mention here that the approach proposed by Toro et al. is different from the one proposed by Harten, Engquist, Osher and Chakravarthy in their pioneering work on ENO schemes [27], where the Cauchy–Kovalewski procedure is applied in the barycenter of each element and the resulting space–time Taylor series for the state is evaluated at the Gaussian integration points in space and time at both sides of the interface. The left and right boundary-extrapolated values are then used as arguments for the Riemann solver, which subsequently computes the interaction of the two space–time Taylor series on both sides of the interface. The ADER approach of Toro and co-workers *first* computes the interaction of both sides via the solution of Riemann problems for the state and all spatial derivatives and *then* applies the Cauchy–Kovalewski procedure. Both methods, however, can be seen as approximate methods to solve the derivative Riemann problem.

Recently, Titarev and Toro [47] proposed the ADER flux expansion variant, where the numerical flux could be integrated analytically at least in time. However, they did not use all the additional information on the mixed space–time derivatives generated by the Cauchy–Kovalewski procedure in order to avoid also numerical quadrature in space.

The main goal of this article is now to fully exploit the space–time Taylor series and to subsequently construct a cost-efficient finite volume method based on this information. An important building block for the

high order accurate spatial discretization on unstructured meshes is the WENO reconstruction operator proposed by Dumbser and Käser in [17] for linear hyperbolic systems. It provides the reconstruction polynomials using hierarchical orthogonal basis functions on a reference element as typically used for discontinuous Galerkin finite element schemes. However, to ensure an essentially non-oscillatory behaviour also for nonlinear hyperbolic systems, a special reconstruction in characteristic variables is used. Once the reconstruction polynomials are obtained, all spatial derivatives at the current time level are computed in the element barycenter with respect to a space–time reference element. To be able to carry out the Cauchy–Kovalewski procedure with the data given with respect to this space–time reference element, the governing PDE has to be rewritten for the reference coordinate system. Then, the Cauchy–Kovalewski procedure is applied in order to produce the full space–time Taylor series for the state and the physical flux inside each element. Using this procedure in the element barycenter to produce a high order one-step time discretization was already proposed by Harten et al. [27]. However, their method was not quadrature-free, since they used Gaussian quadrature in time. Since they only dealt with the one-dimensional case, the problem of spatial quadrature did not arise in their context.

Having computed the space–time Taylor series of the state and the physical flux in all elements, we now propose to use a special numerical flux function at the element interfaces which is composed of a leading flux, evaluated at the space–time barycenter of the element interface plus a quadrature-free high order correction. The leading flux has to be evaluated only $d+1$ times per element, as for first and second order finite volume schemes, independently of the order of accuracy of the method. Typically, we use either the original Godunov flux [26] or HLL-type fluxes [23,29,50] as leading fluxes, but any other Riemann solver would work equally well. The higher order corrections are then obtained by a quadrature-free corrector flux where all space–time integrals are computed analytically. For this corrector flux, we typically use Roe's scheme [39], where the Roe matrix is evaluated only once at the left and right states at the space–time barycenter of the interface and then held constant, i.e. frozen, for the whole interface. Freezing the Roe matrix at the interface is necessary for analytical integration. Instead of Roe's scheme, also the HLL family of schemes may be used in the corrector flux. The signal velocities, however, must be frozen for each element interface, see also [53], where two of the HLLC signal speeds were frozen for the interface.

Our approach is similar to the Taylor quadrature developed recently in the framework of space–time DG schemes [52,53]. The difference is the fact that our space–time Taylor series is the result of the Cauchy–Kovalewski procedure applied to the reconstructed polynomials at the current time level t^n, whereas the space–time DG method directly evolves a polynomial space–time series expansion for the unknowns inside each element. The advantage of the Cauchy–Kovalewski based approach is that it remains fully explicit in time, whereas the space–time DG method is implicit. However, the latter method is unconditionally stable, which may be very useful in many CFD applications.

The outline of this paper is as follows. First, in Section 2, we briefly present the WENO reconstruction algorithm in characteristic variables for unstructured triangular and tetrahedral meshes in two and three space dimensions. Second, the algorithm for performing efficiently the Cauchy–Kovalewski procedure in a space–time reference element is presented in Section 3, for the three-dimensional nonlinear Euler equations of compressible gas dynamics. This is the main building block of our approach since it produces the necessary space–time Taylor series for the state and the flux that is necessary to do an analytical space–time integration over the element boundaries. Third, we present the fully discrete quadrature-free formulation of the scheme in Section 4. Subsequently, in Section 6, numerical convergence studies are carried out on unstructured triangular and tetrahedral meshes for smooth test problems up to sixth order of accuracy in space and time. Finally, two- and three-dimensional test cases with smooth and discontinuous solutions are presented in Section 7. A summary with conclusions and perspectives is given in Section 8.

2. Reconstruction

Finite volume methods contain as a distinguished feature a set of integral cell averages. These are sufficient for the construction of first order methods. The design of higher order finite volume methods usually requires a reconstruction (or recovery) procedure starting from cell averages and yielding higher order polynomials, see [1,4,25,37,30,41,43].

2.1. Reconstruction on unstructured meshes in two and three space dimensions using hierarchical orthogonal basis functions

The main ingredient of the spatial discretization of the proposed arbitrary high order quadrature-free finite volume scheme for nonlinear hyperbolic systems is the WENO reconstruction algorithm proposed by Dumbser and Käser in [17], where it was subsequently applied to construct a quadrature-free ADER-FV scheme for linear hyperbolic systems. The computational domain Ω is discretized by conforming elements $T^{(m)}$, indexed by a unique mono-index m ranging from 1 to the total number of elements E. The elements are chosen to be triangles in 2D and tetrahedrons in 3D. The union of all elements is called the triangulation or tetrahedrization of the domain Ω, respectively,

$$\mathscr{T}_\Omega = \bigcup_{m=1}^{E} T^{(m)}. \tag{1}$$

As usual for finite volume schemes, data is represented by the cell averages of a conserved quantity U_p inside an element $T^{(m)}$,

$$\overline{U}_p^{(m)} = \frac{1}{|T^{(m)}|} \int_{T^{(m)}} U_p \, \mathrm{d}V, \tag{2}$$

where $|T^{(m)}|$ denotes the volume of the element. In order to achieve high order of accuracy for the spatial discretization, we need to reconstruct higher order polynomials w_p from the given cell averages. We write the reconstruction polynomial for the conserved quantity U_p in element $T^{(m)}$ as

$$w_p^{(m)}(\vec{\xi}) = \hat{w}_{pl}^{(m)} \Psi_l(\vec{\xi}), \tag{3}$$

where $\vec{\xi} = (\xi, \eta, \zeta)$ are the coordinates in a reference coordinate system. Fig. 1 depicts the reference elements T_E with the element barycenters and the face barycenters. In 3D, the transformation from the physical coordinate system x–y–z into the reference coordinate system ξ–η–ζ is defined by

$$\begin{aligned} x &= X_1^{(m)} + (X_2^{(m)} - X_1^{(m)})\xi + (X_3^{(m)} - X_1^{(m)})\eta + (X_4^{(m)} - X_1^{(m)})\zeta, \\ y &= Y_1^{(m)} + (Y_2^{(m)} - Y_1^{(m)})\xi + (Y_3^{(m)} - Y_1^{(m)})\eta + (Y_4^{(m)} - Y_1^{(m)})\zeta, \\ z &= Z_1^{(m)} + (Z_2^{(m)} - Z_1^{(m)})\xi + (Z_3^{(m)} - Z_1^{(m)})\eta + (Z_4^{(m)} - Z_1^{(m)})\zeta, \end{aligned} \tag{4}$$

where $X_i^{(m)}$, $Y_i^{(m)}$ and $Z_i^{(m)}$ denote the physical vertex coordinates of the considered element $T^{(m)}$. In two space dimensions, the same transformation is valid for x and y, setting $\zeta = 0$. Throughout the whole paper we use classical tensor notation, which implies summation over each index appearing twice. Whereas the reconstructed degrees of freedom $\hat{w}_{pl}^{(m)}$ are not space-dependent, the reconstruction basis functions Ψ_l are polynomials of degree M and depend on space. The index l ranges from 0 to its maximum value $L-1$, where $L = \frac{1}{2}(M+1)(M+2)$ and $L = \frac{1}{6}(M+1)(M+2)(M+3)$ are the numbers of reconstructed degrees of freedom in 2D and 3D, respectively, depending on the order of the reconstruction. We use the hierarchical orthogonal reconstruction basis functions that are given for example in [10,15] for triangles in 2D and tetrahedrons in 3D.

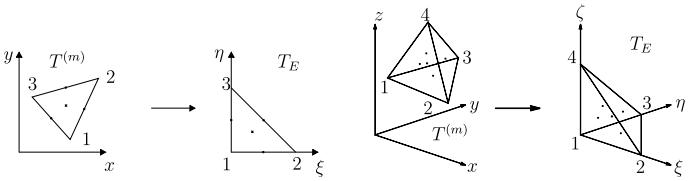


Fig. 1. Transformation from the physical triangle and tetrahedron $T^{(m)}$ to the canonical reference triangle T_E with nodes $(0,0)$, $(1,0)$ and $(0,1)$ and the canonical reference tetrahedron T_E with nodes $(0,0,0)$, $(1,0,0)$, $(0,1,0)$, $(0,0,1)$. The crosses show the element barycenters, the points show the barycenters of the element faces.

Details of the WENO reconstruction algorithm on unstructured meshes in 2D and 3D and how the degrees of freedom $\hat{w}_{pl}$ can be computed on each stencil can be found in [17]. Here we briefly list the main points for reconstruction on an element $T^{(m)}$:

First, construct the set

$$\mathscr{S}^{(m)}_{\mathrm{WENO}} = \bigcup_{i=1}^{n_s} \mathscr{S}^{(m)}_i \quad \text{with } \mathscr{S}^{(m)}_i = \bigcup_{k=1}^{n_e} T^{(j_i(k))} \tag{5}$$

of WENO reconstruction stencils. Here, k with $1 \leqslant k \leqslant n_e$ is a local index, counting the elements in each stencil and $j = j_i(k)$ is the mapping from the local index k to the global indexation of the elements in $\mathscr{T}_\Omega$. We set by definition $j_i(1) = m$ and thus the first element in each stencil ($k = 1$) is always the considered element $T^{(m)}$. The set of stencils $\mathscr{S}^{(m)}_i$ consists of one central stencil ($\mathscr{S}^{(m)}_1$) and a set of one-sided stencils ($2 \leqslant i \leqslant n_s$ with $n_s = 7$ in 2D and $n_s = 9$ in 3D) each containing a number of n_e elements that must be larger than the number of degrees of freedom L. This, unfortunately, leads to overdetermined reconstruction equations that have to be solved, but is necessary for stability reasons [4,25].

Then, transform all elements $T^{(j_i(k))}$ that constitute the reconstruction stencils into the reference coordinate system associated with the element $T^{(m)}$ for which reconstruction is to be done. Thus, $T^{(m)}$ is mapped to the reference element T_E, the other stencil elements are mapped to some neighbors $\widetilde{T}^{(j_i(k))}$ of the reference element. The stencils mapped in this manner are subsequently called $\widetilde{\mathscr{S}}^{(m)}_i$.

To obtain the reconstruction equations we require, for each reconstruction polynomial, integral conservation in all elements of a stencil. The final set of reconstruction equations on each transformed stencil $\widetilde{\mathscr{S}}^{(m)}_i$ is according to [17]:

$$\left(\int_{T_E} \Psi_l(\vec{\tilde{\xi}}(\widetilde{T}^{(j)}, \vec{\Xi}))\mathrm{d}\tilde{\xi}\,\mathrm{d}\tilde{\eta}\,\mathrm{d}\tilde{\zeta} \right) \hat{w}^{(m_i)}_{pl} = |T_E| \overline{U}^{(j_i(k))}_p, \quad \forall \widetilde{T}^{(j_i(k))} \in \widetilde{\mathscr{S}}^{(m)}_i. \tag{6}$$

Here, $\vec{\tilde{\xi}} = (\xi, \eta, \zeta)^{\mathrm{T}}$ is the vector of the coordinates in the reference coordinate system associated with element $T^{(m)}$ and $\vec{\Xi} = (\tilde{\xi}, \tilde{\eta}, \tilde{\zeta})^{\mathrm{T}}$ is the vector of coordinates in another reference coordinate system. The latter one is associated to each transformed element $\widetilde{T}^{(j_i(k))}$ in order to carry out the numerical integration of the left hand side for which we use classical multidimensional Gaussian quadrature of appropriate order. For an exhaustive overview of such multidimensional quadrature formulae see [44]. The resulting overdetermined linear equation system is solved using a least-squares algorithm which preserves exact integral conservation in the considered element $T^{(m)}$. The pseudo-inverse of the reconstruction matrix is stored for all elements and for all WENO stencils. For more details on the stencil search algorithm, on the notation and on the derivation and solution of Eq. (6) see [17].

2.2. Reconstruction in characteristic variables

In order to produce an essentially non-oscillatory reconstruction polynomial, we use a nonlinearly weighted combination of reconstruction polynomials obtained on different stencils, in which smoother reconstructions are preferred. For a history of developments concerning high order accurate polynomial recovery for finite volume methods on unstructured triangulations see the references [1,4,17,25,37,30,33,41,43], and for high order ENO and WENO methods on structured grids see for instance [3,28,31].

Whereas in [17] the reconstruction is done componentwise for linear systems, in this section we present a special *characteristic* reconstruction for nonlinear systems. We point out that the procedure described in this article is different from the usual characteristic ENO or WENO reconstruction described in the research literature on ENO and WENO schemes. Compared to the ENO scheme of Harten et al. [27], our method applies the characteristic transformation *not* onto the cell averages (i.e. the input of the reconstruction), but onto the degrees of freedom $\hat{w}_{pl}$ (i.e. the output of the reconstruction). This is cheaper for our kind of least-squares reconstruction since for a robust reconstruction on unstructured meshes we need *more* cell averages than the number of degrees of freedom finally obtained. Applying the characteristic transformation to the cell averages would result in more matrix-vector multiplications than applying the transformation only to the final degrees of freedom of the reconstruction polynomials. However, it can be easily proven that both formulations

are equivalent. Furthermore, our characteristic reconstruction also differs from the one usually used in WENO schemes since standard WENO schemes reconstruct point values. There, the characteristic decomposition is applied for each spatial Gaussian quadrature point according to the face normal vector of the face under consideration. In our procedure, we reconstruct the entire polynomial with all its coefficients. This makes it necessary to consider all possible characteristic directions simultaneously in order to finally deliver a reconstruction polynomial in conservative variables that is the combination of the smoothest reconstructions in each characteristic direction.

In more detail, we apply our WENO algorithm to each characteristic variable of the system, depending on the direction. On the unstructured meshes considered here we use the face normal vectors $\vec{n}_e$ as characteristic directions, where e denotes the number of the edge for triangles in 2D and the number of the face for tetrahedrons in 3D. On each stencil $\widetilde{\mathscr{S}}_i^{(m)}$, the degrees of freedom defining the polynomial for each characteristic direction $\vec{n}_e$ are

$$\hat{c}_{pl}^{(i,\vec{n}_e)} = (R_{pq}^{\vec{n}_e})^{-1} \cdot \hat{w}_{ql}^{(m_i)}, \tag{7}$$

where $R_{pq}^{\vec{n}_e}$ is the matrix of right eigenvectors of the Jacobian matrix in direction $\vec{n}_e$ of the relevant hyperbolic system, evaluated for each face normal $\vec{n}_e$ at the arithmetic average $\frac{1}{2}(\overline{U}_p^{(m)} + \overline{U}_p^{(k_e)})$ of the cell averages of the considered element $T^{(m)}$ and the associated neighbor element $T^{(k_e)}$ at face number $0 \leqslant e \leqslant N_E$, where N_E is the number of faces and where we additionally define the zeroth set of characteristic variables to be equal to the conservative variables themselves, i.e. $R_{pq}^{\vec{n}_0} = \delta_{pq}$, with the classical Kronecker symbol δ_{pq}. We note that in the following Eqs. (8)–(14) there is *no* automatic summation induced over index p. First, we compute the smoothness σ of each characteristic polynomial from the quadratic functional according to [17],

$$\sigma_p^{(i,\vec{n}_e)} = \hat{c}_{pl}^{(i,\vec{n}_e)} \cdot \Sigma_{lm} \cdot \hat{c}_{pm}^{(i,\vec{n}_e)}. \tag{8}$$

Here, Σ_{lm} is the universal mesh-independent oscillation indicator matrix that depends only on the desired polynomial degree M of the reconstruction. The non-normalized nonlinear weights $\tilde{\omega}_p^{(i,\vec{n}_e)}$ are then functions of the linear weights λ_i and the previously computed oscillation indicators $\sigma_p^{(i,\vec{n}_e)}$. Since the final weights $\omega_p^{(i,\vec{n}_e)}$ must sum to 1, the non-normalized weights $\tilde{\omega}_p^{(i,\vec{n}_e)}$ must be normalized with their sum. One therefore obtains

$$\omega_p^{(i,\vec{n}_e)} = \frac{\tilde{\omega}_p^{(i,\vec{n}_e)}}{\sum_{s=1}^{n_s} \tilde{\omega}_p^{(s,\vec{n}_e)}}, \quad \text{with } \tilde{\omega}_p^{(i,\vec{n}_e)} = \frac{\lambda_i}{(\epsilon + \sigma_p^{(i,\vec{n}_e)})^r}. \tag{9}$$

The linear weights are chosen according to [17], putting a large linear weight $\lambda_1 \gg 1$ on the central stencil and $\lambda_i = 1$ elsewhere. Finally, we obtain for each characteristic variable for each characteristic direction $\vec{n}_e$ a reconstruction polynomial of degree M that is obtained from a nonlinear combination of the reconstruction polynomials on all stencils $\widetilde{\mathscr{S}}_i^{(m)}$:

$$\hat{c}_{pl}^{(\mathrm{WENO},\vec{n}_e)} = \sum_{i=1}^{n_s} \omega_p^{(i,\vec{n}_e)} \hat{c}_{pl}^{(i,\vec{n}_e)}, \quad \forall 0 \leqslant e \leqslant N_E. \tag{10}$$

Since in our approach the final result of the reconstruction operator is one single reconstruction polynomial defined in the whole element, we still have to perform a selection amongst the various characteristic reconstruction polynomials obtained by (10). Therefore, we transform all characteristic WENO polynomials back to conservative variables, i.e.

$$\hat{w}_{pl}^{(\mathrm{WENO},\vec{n}_e)} = R_{pq}^{\vec{n}_e} \cdot \hat{c}_{ql}^{(\mathrm{WENO},\vec{n}_e)}, \quad \forall 0 \leqslant e \leqslant N_E, \tag{11}$$

and then choose for each conservative variable the smoothest polynomial obtained from all the different characteristic reconstructions. First, the oscillation indicators for the polynomials (11) are computed as in (12),

$$\sigma_p^{\vec{n}_e} = \hat{w}_{pl}^{(\mathrm{WENO},\vec{n}_e)} \cdot \Sigma_{lm} \cdot \hat{w}_{pm}^{(\mathrm{WENO},\vec{n}_e)}, \tag{12}$$

then the smoothest one is selected according to an ENO-type algorithm:

$$\omega_p^{\vec{n}_e} = \begin{cases} 1 & \text{if } \sigma_p^{\vec{n}_e} = \min(\sigma_p^{\vec{n}_0}, \sigma_p^{\vec{n}_1}, \ldots, \sigma_p^{\vec{n}_{N_E}}), \\ 0 & \text{else.} \end{cases} \tag{13}$$

The final reconstruction polynomial for the conservative variables, valid in the whole element $T^{(m)}$, is then given by

$$\hat{w}_{pl}^{\mathrm{WENO}} = \sum_{e=0}^{N_E} \omega_p^{\vec{n}_e} \hat{w}_{pl}^{(\mathrm{WENO},\vec{n}_e)}. \tag{14}$$

An important feature of the proposed WENO reconstruction algorithm is that it produces the coefficients $\hat{w}_{pl}^{\mathrm{WENO}}$ of the reconstructed polynomial in each element $T^{(m)}$. It furthermore avoids the problem of negative weights [30,41], since the linear weights λ_i are simply chosen such as to put a very large weight $\lambda_1 \gg 1$ on the central stencil, which should preferably be used in smooth regions of the numerical solution because it provides the stability and accuracy of the method, and $\lambda_i = 1$ for all the other stencils with $i \neq 1$. We must note that this simple method to avoid the negative weights has the cost that the optimal order of accuracy of standard WENO schemes [31,3] can not be reached. With this information we now can compute all spatial derivatives of the conserved variables at the element barycenter $T^{(m)}$ by performing a change of the basis via L_2 projection. The spatial derivatives are needed later for the Cauchy–Kovalewski procedure. We first define a set of space–time basis functions

$$\theta_{ijkl}(\vec{\xi}, \tau) = \frac{(\xi - \xi_{\mathrm{BC}})^i (\eta - \eta_{\mathrm{BC}})^j (\zeta - \zeta_{\mathrm{BC}})^k \tau^l}{i!j!k!l!}, \tag{15}$$

which correspond to the polynomials associated with a space–time Taylor series of maximum degree M expanded at the barycenter $\vec{\xi}_{\mathrm{BC}} = (\xi_{\mathrm{BC}}, \eta_{\mathrm{BC}}, \zeta_{\mathrm{BC}})$ of each element and expanded at the current time t^n. The coordinate vectors for the element barycenter are $\vec{\xi}_{\mathrm{BC}} = (\frac{1}{3}, \frac{1}{3})$ in 2D and $\vec{\xi}_{\mathrm{BC}} = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4})$ in 3D, respectively. For the range of the indices i, j, k, l we have $0 \leqslant i, j, k, l \leqslant M$ and $0 \leqslant i + j + k + l \leqslant M$. Furthermore, in (15) we have introduced the normalized time

$$\tau = \frac{t - t^n}{\Delta t}. \tag{16}$$

The numerical solution depending on space and time is written in this basis as

$$U_p(\vec{\xi}, \tau) = \frac{\partial^{i+j+k+l}}{\partial \xi^i \partial \eta^i \partial \zeta^i \partial \tau^l} U_p \theta_{ijkl}(\vec{\xi}, \tau) := \widehat{U}_{pijkl} \theta_{ijkl}(\vec{\xi}, \tau). \tag{17}$$

Then, the transformation between the purely spatial reconstruction basis $\Psi_l(\vec{\xi})$ and the zeroth time derivative level of the space–time Taylor basis $\theta_{ijkl}(\vec{\xi}, \tau)$ is carried out by standard L_2 projection,

$$\langle \Psi_n, \Psi_m \rangle \hat{w}_{pm}^{\mathrm{WENO}} = \langle \Psi_n, \theta_{ijk0} \rangle \widehat{U}_{pijk0}, \tag{18}$$

where $\langle ., . \rangle$ is the standard inner product of two functions over the reference element T_E. Hence, the purely spatial derivatives of the conserved quantities U_p follow from the solution of (18) directly as

$$\widehat{U}_{pijk0} = (\langle \Psi_n, \theta_{ijk0} \rangle^{-1} \langle \Psi_n, \Psi_m \rangle) \hat{w}_{pm}^{\mathrm{WENO}}. \tag{19}$$

The matrix on the right hand side does not depend on the mesh and can be computed once and then stored. Since it is a sparse matrix, the change of basis can be performed very efficiently.

3. The Cauchy–Kovalewski procedure in the reference element for the three-dimensional Euler equations

The reconstruction yields, for each element, a polynomial distribution at a given time level t^n. The resulting discontinuities at the element faces are then resolved via the solution of generalized (or derivative) Riemann problems. The solution of such local Cauchy problems involves Taylor series expansions and the substitution of time derivatives by spatial derivatives via the Cauchy–Kovalewski procedure. In this section, we will present a new approach for this procedure, which is particularly suited for unstructured meshes.

It is a particular and very important feature of triangular and tetrahedral elements that the mapping from the physical x–y–z coordinate system to the ξ–η–ζ reference coordinate system leads to a *constant* Jacobian matrix for the mapping. This allows us to rewrite the governing PDE easily in terms of the space–time derivatives in the reference system. For general quadrilaterals and hexahedrons as well as for elements with curved boundaries, this transformation of the PDE is still possible, but the spatially varying Jacobian matrix of the mapping must then be taken into account later inside the Cauchy–Kovalewski procedure by applying the product rule.

For the case of a mapping with constant Jacobian matrix, such as (4), we can write a generic nonlinear hyperbolic system of conservation laws

$$\frac{\partial}{\partial t}U_p + \frac{\partial}{\partial x}f_p + \frac{\partial}{\partial y}g_p + \frac{\partial}{\partial z}h_p = S_p \tag{20}$$

directly as

$$\frac{\partial}{\partial \tau}U_p + \Delta t\left(\frac{\partial}{\partial \xi}f_p^* + \frac{\partial}{\partial \eta}g_p^* + \frac{\partial}{\partial \zeta}h_p^*\right) = \Delta t S_p \tag{21}$$

with

$$f^* = f\xi_x + g\xi_y + h\xi_z, \quad g^* = f\eta_x + g\eta_y + h\eta_z, \quad h^* = f\zeta_x + g\zeta_y + h\zeta_z. \tag{22}$$

S_p is a vector of source terms corresponding to some volume force. For the nonlinear Euler equations, the state vector of the conserved variables U_p and the vectors of the nonlinear fluxes f_p, g_p and h_p are:

$$U_p = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix} \quad f_p = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(\rho E + p) \end{pmatrix} \quad g_p = \begin{pmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho vw \\ v(\rho E + p) \end{pmatrix} \quad h_p = \begin{pmatrix} \rho w \\ \rho wu \\ \rho wv \\ \rho w^2 + p \\ w(\rho E + p) \end{pmatrix}. \tag{23}$$

The system is closed by the equation of state of an ideal gas,

$$p = (\gamma - 1)\left(\rho E - \frac{1}{2}\rho(u^2 + v^2 + w^2)\right). \tag{24}$$

Here, $\gamma = 1.4$ is the ratio of specific heats. The two-dimensional case is obtained by setting $\frac{\partial}{\partial z} = 0$ and $w = 0$.

Although it may initially seem to be extremely cumbersome to do the Cauchy–Kovalewski procedure for nonlinear three-dimensional hyperbolic systems, there exists a very efficient algorithm at least for some particular nonlinear systems. It was originally developed by Dyson [22] for the Euler equations and was then modified by Dumbser and Munz [21,20] and Taube et al. [46] in their work on ADER-DG schemes for the Euler and MHD equations. However, in the previous references, the algorithm was always applied in physical coordinates. The method is based on the generalized Leibniz rule, which gives the space–time derivatives of arbitrary order for a product of multivariate scalar functions f_1 and f_2:

$$\begin{aligned} \mathscr{L}^{(a,b,c,d)}(f_1, f_2) &:= \frac{\partial^{a+b+c+d}(f_1(\xi,\eta,\zeta,\tau)f_2(\xi,\eta,\zeta,\tau))}{\partial\xi^a\partial\eta^b\partial\zeta^c\partial\tau^d} \\ &= \sum_{i=0}^{a}\sum_{j=0}^{b}\sum_{k=0}^{c}\sum_{l=0}^{d}\left[\Pi_{bc}\cdot\frac{\partial^{(a-i)+(b-j)+(c-k)+(d-l)}f_1}{\partial\xi^{(a-i)}\partial\eta^{(b-j)}\partial\zeta^{(c-k)}\partial\tau^{(d-l)}}\cdot\frac{\partial^{i+j+k+l}f_2}{\partial\xi^i\partial\eta^j\partial\zeta^k\partial\tau^l}\right], \end{aligned} \tag{25}$$

with Π_{bc} the product of the binomial coefficients

$$\Pi_{bc} = \binom{a}{i}\binom{b}{j}\binom{c}{k}\binom{d}{l}. \tag{26}$$

This rule alone is not yet sufficient to treat the nonlinear Euler equations, since there appears for example a term like $(\rho u)u$ in the momentum flux, but u and its derivatives are not known a priori, but only those of the conservative variables ρ and ρu. The key now consists in reformulating the Leibniz rule (25) such as to

calculate all the space–time derivatives of the auxiliary variable u from the known space–time derivatives of ρ and $\rho\, u$ and all previously computed space–time derivatives of lower order of the auxiliary variable u. This is achieved by a modified Leibniz rule of the form

$$\mathscr{L}_{**}^{(a,b,c,d)}(f_1 f_2, f_1, f_2) := \frac{\partial^{a+b+c+d} f_2(\xi,\eta,\zeta,\tau)}{\partial\xi^a \partial\eta^b \partial\zeta^c \partial\tau^d} = \frac{1}{f_1}\cdot\left[\frac{\partial^{a+b+c+d}(f_1(\xi,\eta,\zeta,\tau) f_2(\xi,\eta,\zeta,\tau))}{\partial\xi^a \partial\eta^b \partial\zeta^c \partial\tau^d} - \mathscr{L}_{*}^{(a,b,c,d)}(f_1, f_2)\right], \tag{27}$$

with $f_1 \neq 0$. The operator $\mathscr{L}_{*}^{(a,b,c,d)}(f_1, f_2)$ only contains lower order derivatives of f_2 and is defined as the original Leibniz rule (25) except of the last term in the sum:

$$\mathscr{L}_{*}^{(a,b,c,d)}(f_1, f_2) = \underbrace{\sum_{i=0}^{a}\sum_{j=0}^{b}\sum_{k=0}^{c}\sum_{l=0}^{d}}_{i+j+k+l\neq a+b+c+d}\left[\Pi_{bc}\cdot\frac{\partial^{(a-i)+(b-j)+(c-k)+(d-l)} f_1}{\partial\xi^{(a-i)}\partial\eta^{(b-j)}\partial\zeta^{(c-k)}\partial\tau^{(d-l)}}\cdot\frac{\partial^{i+j+k+l} f_2}{\partial\xi^{i}\partial\eta^{j}\partial\zeta^{k}\partial\tau^{l}}\right],$$
$$\text{with } \mathscr{L}_{*}^{(0,0,0,0)}(f_1, f_2) := 0. \tag{28}$$

For a simple illustration of this strategy see [21]. Using the operators (25)–(28) we can now tackle the Cauchy–Kovalewski procedure for the nonlinear three-dimensional Euler equations written in the reference element, see Eq. (21). We suppose that all pure spatial derivatives with respect to ξ, η and ζ of the conservative variables ρ, ρu, ρv, ρw and ρE are already computed in the element barycenter according to Eq. (19).

Starting from zeroth order spatial and temporal derivatives and successively increasing the spatial and the temporal derivative order, the Cauchy–Kovalewski procedure can be written in an unrolled recursive manner. At each time-derivative level, first the primitive variables u, v and w are computed from the vector of conserved quantities U_p using the operator $\mathscr{L}_{**}$. Then, the auxiliary variables ρu^2, ρv^2, ρw^2, ρuv ρuw and ρvw are computed using the generalized Leibniz rule $\mathscr{L}$, from the first three of which the pressure p can be subsequently computed. After applying the operator $\mathscr{L}$ to compute $u(\rho E + p)$, $v(\rho E + p)$ and $w(\rho E + p)$, the spatio-temporal derivatives of the nonlinear fluxes according to (23) can be computed. Inserting these into the definition (22) finally allows to compute the time-derivatives of successive order of the conservative variables U using the governing PDE written in the reference element (21). We emphasize that this algorithm is completely generic and can be run up to *any* given degree of the reconstruction polynomials M.

In order to present the whole algorithm more clearly, in Appendix A we list the entire Fortran code of a subroutine called `NonLinearCK_Euler3D`, which performs the Cauchy–Kovalewski procedure in our method for the homogeneous case without source terms. For the incorporation of nonlinear source terms, for example gravitational volume forces, see [20] for more details. In the case of externally imposed source terms that depend only on space and time, all mixed space–time derivatives of S_p have to be added in the last loop, see the comments in Appendix A.

When entering the subroutine, the subarray `W(1:5,:,:,:,0)` contains as input all purely spatial derivatives with respect to ξ, η and ζ of the conservative variables. The first index of `W` refers to the variable number as given in the comments of the code, the second, third and fourth indices refer to the order of the ξ, η and ζ derivatives, respectively, and the last index refers to the derivatives with respect to the reference time τ. While running the algorithm, the array `W` is filled with all mixed space–time derivatives of the conservative variables and the introduced auxiliary variables, from which the mixed space–time derivatives of the fluxes are obtained. The set of fluxes f_p^*, g_p^*, h_p^* is used according to (23) to compute the time derivatives of the next level for U_p. The mixed space–time derivatives of the nonlinear fluxes f_p, g_p, h_p are returned to the calling subroutine in the arrays `FCx`, `FCy` and `FCz`. All mixed space–time derivatives of the conservative variables are returned in the subarray `W(1:5,:,:,:,:)`.

As final result of the Cauchy–Kovalewski procedure, we hence obtain the full set of coefficients $\widehat{U}_{pijkl}$, $\hat{f}_{pijkl}$, $\hat{g}_{pijkl}$ and $\hat{h}_{pijk}$ of the spatio-temporal Taylor series for the conservative variables *and* the nonlinear fluxes, expanded about the current time t^n and the element barycenter $\vec{\xi}_{\mathrm{BC}}$:

$$U_p(\vec{\xi},\tau) = \widehat{U}_{pijkl}\cdot\theta_{ijkl}(\vec{\xi},\tau), \quad f_p(\vec{\xi},\tau) = \hat{f}_{pijkl}\cdot\theta_{ijkl}(\vec{\xi},\tau),$$
$$g_p(\vec{\xi},\tau) = \hat{g}_{pijkl}\cdot\theta_{ijkl}(\vec{\xi},\tau), \quad h_p(\vec{\xi},\tau) = \hat{h}_{pijkl}\cdot\theta_{ijkl}(\vec{\xi},\tau). \tag{29}$$

4. The fully discrete formulation of the quadrature-free ADER-FV scheme

As it is usual for finite volume schemes, the conservation law (20) is integrated over each element $T^{(m)}$ in space using Gauss' theorem and in the time interval $[t^n; t^n + \Delta t]$,

$$\int_{t^n}^{t^n+\Delta t}\int_{T^{(m)}}\frac{\partial}{\partial t}U_p\,\mathrm{d}V\,\mathrm{d}t + \int_{t^n}^{t^n+\Delta t}\int_{\partial T^{(m)}}(f_p\ g_p\ h_p)\vec{n}\,\mathrm{d}S\,\mathrm{d}t = \int_{t^n}^{t^n+\Delta t}\int_{T^{(m)}}S_p\mathrm{d}V\,\mathrm{d}t. \tag{30}$$

It can be easily shown, that the Euler equations are rotationally invariant [48], so the boundary integral term in (30) can be written in a much simpler form after transformation to an edge-aligned local coordinate system. The transformation from the original variables U_p to the edge-aligned variables U'_p and its inverse are

$$U_p = T_{pq}U'_q, \quad U'_p = T_{pq}U_p, \tag{31}$$

with

$$T_{pq} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & n_x & s_x & t_x & 0 \\ 0 & n_y & s_y & t_y & 0 \\ 0 & n_z & s_z & t_z & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \tag{32}$$

where $\vec{n} = (n_x, n_y, n_z)^{\mathrm{T}}$ is the unit normal vector on the surface $\partial T^{(m)}$ of the element and $\vec{s} = (s_x, s_y, s_z)^{\mathrm{T}}$ and $\vec{t} = (t_x, t_y, t_z)^{\mathrm{T}}$ are two unit vectors that are orthogonal to each other and to $\vec{n}$. Integrating the first term in (30) in space and time and making use of the rotational invariance of the Euler equations, we get

$$|T^{(m)}|(\overline{U}_p(t^{n+1}) - \overline{U}_p(t^n)) = \int_{t^n}^{t^n+\Delta t}\int_{T^{(m)}}S_p(\vec{\xi},t)\,\mathrm{d}V\,\mathrm{d}t - s$$
$$-\sum_{e=1}^{N_E}\int_0^1\int_{\partial T_e^{(m)}}T_{pq}f_p^h(U_q'^{(m)}(\vec{\xi},\tau), U_q'^{(k_e)}(\vec{\xi},\tau), f_p'^{(m)}(\vec{\xi},\tau)), f_p'^{(k_e)}(\vec{\xi},\tau))\mathrm{d}S\,\mathrm{d}\tau, \tag{33}$$

where $N_E = d + 1$ denotes the number of element faces in d space dimensions and $|T^{(m)}|$ the element volume. For computing the boundary integral in Eq. (33) a numerical flux f_p^h depending on the left and right states and fluxes at the element interface number e between element $T^{(m)}$ and its neighbor $T^{(k_e)}$ has been introduced since

Table 1
Face definition on triangles and tetrahedrons

Face	Points		
Triangles (2D)			
1	1		2
2	2		3
3	3		1
Face	Points		
Tetrahedrons (3D)			
1	1	3	2
2	1	2	4
3	1	4	3
4	2	3	4

the reconstructed solution usually may jump across the element interfaces. We use the following numerical flux at the element interfaces:

$$\begin{aligned} & f_p^h(U_q^{\prime(m)}(\vec{\xi},\tau), U_q^{\prime(k_e)}(\vec{\xi},\tau), f_p^{\prime(m)}(\vec{\xi},\tau)), f_p^{\prime(k_e)}(\vec{\xi},\tau)) \\ & \quad = f_p^0(U_q^{\prime(m)}(\bar{\beta},\bar{\tau}), U_q^{\prime(k_e)}(\bar{\beta},\bar{\tau})) - f_p^c(U_q^{\prime(m)}(\bar{\beta},\bar{\tau}), U_q^{\prime(k_e)}(\bar{\beta},\bar{\tau}), f_p^{\prime(m)}(\bar{\beta},\bar{\tau})), f_p^{\prime(k_e)}(\bar{\beta},\bar{\tau})) \\ & \qquad + f_p^c(U_q^{\prime(m)}(\vec{\xi},\tau), U_q^{\prime(k_e)}(\vec{\xi},\tau), f_p^{\prime(m)}(\vec{\xi},\tau)), f_p^{\prime(k_e)}(\vec{\xi},\tau)), \end{aligned} \tag{34}$$

where a Roe-type corrector flux $f_p^{\mathrm{c,Roe}}$ is defined as

$$\begin{aligned} & f_p^{\mathrm{c,Roe}}(U_q^{\prime(m)}(\vec{\xi},\tau), U_q^{\prime(k_e)}(\vec{\xi},\tau), f_p^{\prime(m)}(\vec{\xi},\tau)), f_p^{\prime(k_e)}(\vec{\xi},\tau)) \\ & \quad = \frac{1}{2}(f_p^{\prime(m)}(\vec{\xi},\tau) + f_p^{\prime(k_e)}(\vec{\xi},\tau)) - \frac{1}{2}|\widetilde{A}_{pq}|(U_q^{\prime(k_e)}(\vec{\xi},\tau) - U_q^{\prime(m)}(\vec{\xi},\tau)). \end{aligned} \tag{35}$$

Here, $\bar{\tau} = \frac{1}{2}$ is the time step barycenter and $\bar{\beta} = (\bar{\xi},\bar{\eta},\bar{\zeta}) = \vec{\xi}(\bar{\chi}_1,\bar{\chi}_2)$ is the spatial barycenter of the element interface, parametrized by the local face parameters $\bar{\chi}_1 = \frac{1}{2}$ in 2D and $(\bar{\chi}_1,\bar{\chi}_2) = (\frac{1}{3},\frac{1}{3})$ in 3D, see Table 2 for the computation of the volume coordinates $\vec{\xi}$ in function of the local face parameters χ_1 and χ_2. Note that the space–time dependent corrector flux in (34) is a *linear* function of its four arguments, the last two of which are the left and right physical fluxes in normal direction $f_q^{\prime(m)}$ as obtained from the result (29) of the Cauchy–Kovalewski procedure:

$$f_p^{\prime(m)}(\vec{\xi},t) = \hat{f}_{pijkl}^{\prime(m)} \cdot \theta_{ijkl}(\vec{\xi},\tau), \tag{36}$$

with

$$\hat{f}_{pijkl}^{\prime(m)} = n_x \hat{f}_{pijkl}^{(m)} + n_y \hat{g}_{pijkl}^{(m)} + n_z \hat{h}_{pijkl}^{(m)}. \tag{37}$$

The term $|\widetilde{A}_{pq}|$ in the corrector flux denotes the absolute value of the Roe matrix [39] of the boundary extrapolated left and right states evaluated at the space–time barycenter of the element interface. Here, the absolute value operator of a matrix M_{pq} is defined by applying the absolute value operator componentwise on the diagonal matrix of eigenvalues Λ_{pq}, i.e. $|M_{pq}| = R_{pr}|\Lambda_{rs}|R_{sq}^{-1}$, where R_{pr} is the matrix of the right eigenvectors. For explicit expressions of the Roe flux see e.g. [48].

The numerical flux depends on both space and time, since we have to integrate over the element face in space and time. The first term f_p^0 on the right hand side of (34) is the numerical flux of leading order, evaluated at the space–time barycenter of the face. Here, any Riemann solver can be used. For an overview of Riemann solvers see for example [48]. The second term in (34) corresponds to the Roe flux at the space–time barycenter of the interface, where the leading order flux is evaluated. This term is necessary for consistency. The last term corresponds to the Roe flux, where the Roe matrix is, however, kept constant on the whole face and does not depend on space and time. It is evaluated once at the left and right state entering the leading order flux and then its value is 'frozen'. This is very important since otherwise, the flux could not be integrated analytically in space and time. Thus, the whole numerical flux can be interpreted as the sum of a leading flux f_p^0 and a Roe-type space–time dependent corrector flux. Due to its evaluation in the space–time barycenter, which corresponds to the Gaussian integration point for finite volume schemes up to second order of accuracy, the leading flux is up to second order accurate in space and time, whereas the Roe-type correction vanishes for first and second order finite volume schemes where the method reduces to the MUSCL scheme [55,56]. The correction intervenes only for schemes of order higher than two, where one spatio-temporal Gaussian integration point would no longer be enough to provide a formally accurate scheme.

Table 2
2D and 3D volume coordinates $\vec{\xi}^{(e)}$ as function of the face parameters χ_1 and χ_2

e	Triangles (2D)			Tetrahedrons (3D)			
	1	2	3	1	2	3	4
$\xi^{(j)}(\chi_1,\chi_2)$	χ_1	$1-\chi_1$	0	χ_2	χ_1	0	$1-\chi_1-\chi_2$
$\eta^{(j)}(\chi_1,\chi_2)$	0	χ_1	$1-\chi_1$	χ_1	0	χ_2	χ_1
$\zeta^{(j)}(\chi_1,\chi_2)$				0	χ_2	χ_1	χ_2

Table 3
Transformation of the face parameters χ_1 and χ_2 of the tetrahedron's face to the face parameters $\tilde{\chi}_1$ and $\tilde{\chi}_2$ in the neighbor tetrahedron according to the three possible orientations (h) of the neighbor face

h	1	2	3
$\tilde{\chi}_1^{(h)}(\chi_1,\chi_2)$	χ_2	$1-\chi_1-\chi_2$	χ_1
$\tilde{\chi}_2^{(h)}(\chi_1,\chi_2)$	χ_1	χ_2	$1-\chi_1-\chi_2$

Since the computation of the absolute value of the Roe matrix is quite expensive and because the Roe flux without modification is not an entropy flux, a more robust and faster corrector flux of the HLL-type $f_p^{c,\mathrm{HLL}}$ can be defined as follows:

$$f_p^{c,\mathrm{HLL}}(U_q'^{(m)}(\vec{\xi},\tau),U_q'^{(k_e)}(\vec{\xi},\tau),f_p'^{(m)}(\vec{\xi},\tau)),f_p'^{(k_e)}(\vec{\xi},\tau))$$
$$=\frac{1}{s_R-s_L}[s_R f_p'^{(m)}(\vec{\xi},\tau)-s_L f_p'^{(k_e)}(\vec{\xi},\tau)+s_L s_R(U_q'^{(k_e)}(\vec{\xi},\tau)-U_q'^{(m)}(\vec{\xi},\tau))]. \tag{38}$$

Here, the estimates of the signal velocities s_L and s_R are given as follows:

$$s_R=\max(0,u_R'+c_R,\tilde{u}'+\tilde{c}),\quad s_L=\min(0,u_L'-c_L,\tilde{u}'-\tilde{c}). \tag{39}$$

As for the Roe matrix, the velocities in Eq. (39) are only evaluated once at the space–time barycenter of the element interface and are then kept constant for the whole interface. This approach is very similar to the Taylor quadrature developed in [53] for second order accurate space–time DG methods.

Due to the linearity of the space–time dependent corrector flux in its arguments and the availability of their Taylor series expansion in space and time, we can integrate the numerical flux analytically in space and time over the space–time reference element interface. We can furthermore apply the integration first to the arguments and then compute the corrector flux with the space–time integrated arguments

$$\begin{aligned}
&\int_0^1\int_{\partial T_e^{(m)}}(U_p'^{(m)}(\vec{\xi},\tau)-U_p'^{(m)}(\bar{\beta},\bar{\tau}))\mathrm{d}S\,\mathrm{d}\tau=\widehat{U'}^{(m)}_{pijkl}F^{-,e}_{ijkl},\\
&\int_0^1\int_{\partial T_e^{(m)}}(U_p'^{(k_e)}(\vec{\xi},\tau)-U_p'^{(k_e)}(\bar{\beta},\bar{\tau}))\mathrm{d}S\,\mathrm{d}\tau=\widehat{U'}^{(k_e)}_{pijkl}F^{+,e^+,h^+}_{ijkl},\\
&\int_0^1\int_{\partial T_e^{(m)}}(f_p'^{(m)}(\vec{\xi},\tau)-f_p'^{(m)}(\bar{\beta},\bar{\tau}))\mathrm{d}S\,\mathrm{d}\tau=\widehat{f'}^{(m)}_{pijkl}F^{-,e}_{ijkl},\\
&\int_0^1\int_{\partial T_e^{(m)}}(f_p'^{(k_e)}(\vec{\xi},t)-f_p'^{(k_e)}(\bar{\beta},\bar{\tau}))\mathrm{d}S\,\mathrm{d}\tau=\widehat{f'}^{(k_e)}_{pijkl}F^{+,e^+,h^+}_{ijkl},
\end{aligned} \tag{40}$$

using the following mesh-independent flux matrices, which can be precomputed once and then stored:

$$F^{-,e}_{ijkl}=\int_0^1\int_{\partial(T_E)_e}(\theta_{ijkl}(\vec{\xi}^{(e)}(\chi_1,\chi_2),\tau)-\theta_{ijkl}(\vec{\xi}(\bar{\chi}_1,\bar{\chi}_2),\bar{\tau}))\mathrm{d}\chi_1\,\mathrm{d}\chi_2\,\mathrm{d}\tau, \tag{41}$$

$$F^{+,e^+,h}_{ijkl}=\int_0^1\int_{\partial(T_E)_e}(\theta_{ijkl}(\vec{\xi}^{(e^+)}(\tilde{\chi}_1^{(h)},\tilde{\chi}_2^{(h)}),\tau)-\theta_{ijkl}(\vec{\xi}^{(e^+)}(\bar{\chi}_1,\bar{\chi}_2),\bar{\tau}))\mathrm{d}\chi_1\,\mathrm{d}\chi_2\,\mathrm{d}\tau. \tag{42}$$

Here, e and e^+ are the local numbers of the considered common element interface between elements $T^{(m)}$ and $T^{(k_e)}$ as seen from each element, respectively, and h takes into account the possibly different orientations of two tetrahedral faces due to rotation (see Table 3) and has a meaning only in the three-dimensional case. The element faces, as defined in Table 1, are parametrized by the face parameters χ_1 and χ_2, which are mapped to the volume coordinates $\vec{\xi}$ by a mapping function depending on the face number and the orientation. For more details on the computation of the flux matrices in 2D and 3D, see [17].

The final fully discrete form of the quadrature-free finite volume scheme reads as follows:

$$|T^{(m)}|(\overline{U}_p(t^{n+1}) - \overline{U}_p(t^n)) = \Delta t|J| \int_0^1 \int_{T_E} S_p(\vec{\xi},\tau)\mathrm{d}V\,\mathrm{d}\tau - \sum_{e=1}^{N_E} \Delta t|\partial T_e^{(m)}| T_{pq} \cdot f_p^0(U_q'^{(m)}(\bar{\beta},\bar{\tau}), U_q'^{(k_e)}(\bar{\beta},\bar{\tau})) - \sum_{e=1}^{N_E} \Delta t|S_e| T_{pq} f_q^c(\widehat{U'}^{(m)}_{rijkl} F^{-,e}_{ijkl}, \widehat{U'}^{(k_e)}_{rijkl} F^{+,e^+,h^+}_{ijkl}, \widehat{f'}^{(m)}_{rijkl} F^{-,e}_{ijkl}, \widehat{f'}^{(k_e)}_{rijkl} F^{+,e^+,h^+}_{ijkl}). \quad (43)$$

The symbols $|S_e|$ are the Jacobi determinants of the faces which are equal to the edge length $|\partial T_e^{(m)}|$ in 2D and equal to half of the triangular face surface $|\partial T_e^{(m)}|$ in 3D. We emphasize that the proposed quadrature-free finite volume scheme only needs one leading flux evaluation per element face and time step, independent of the chosen order of accuracy. For schemes of order greater than two, just one additional corrector flux evaluation is necessary per element face and time step.

In the following, we would like to summarize the necessary steps of the whole algorithm described in this section to perform the update of the cell averages from time t^n to time $t^n + \Delta t$:

(1) Apply the characteristic WENO reconstruction algorithm described in Section 2 to the mean values $\overline{U}_p$ in order to obtain the high order non-oscillatory reconstruction polynomials $w_p(\vec{\xi})^{(m)}$ as defined in (3).
(2) Perform the change of basis algorithm (19) in order to compute the spatial derivatives of the conservative variables at time t^n in the spatial barycenter of each space–time reference element.
(3) Insert the barycentric spatial derivatives of U_p obtained in step 2 into the Cauchy–Kovalewski procedure as shown in detail in the program listing given in Appendix A. This yields as a result the coefficients of the space–time Taylor series in the space–time reference element T_E for the state U_p and the nonlinear physical fluxes f_p, g_p and h_p.
(4) Rotate the coefficients of the Taylor series for the state and the physical fluxes in the edge-aligned coordinate system according to (31) and (36).
(5) Compute the leading fluxes f_p^0 and the Roe matrix $\widetilde{A}_{pq}$ or the signal velocities s_L and s_R at the space–time barycenter $(\bar{\beta},\bar{\tau})$ and assume them to be constant on the face. Multiply the degrees of freedom of the spatio-temporal Taylor series obtained in step 4 for the element and its neighbors with the corresponding flux matrices and compute the corrector flux f_p^c according to (35) or (38).
(6) If necessary, compute the space–time integral of externally imposed source terms using Gaussian quadrature.
(7) Update the cell averages according to (43).

Note that the computation of the corrector fluxes is only needed for finite volume schemes of order of accuracy higher than two. Since our method can achieve theoretically any order of accuracy in space and time, we will refer to it as quadrature-free ADER finite volume (QF ADER-FV) scheme, since it fully exploits all the derivative information obtained from the reconstruction.

5. On the implementation of boundary conditions and MPI parallelization

There is a variety of physically meaningful boundary conditions for the compressible Euler equations. However, the four most important types are: periodic boundary conditions, reflective (solid slip-wall) boundary conditions, inflow boundaries and transmissive boundaries. An important point in the context of finite volume schemes is the generation of appropriate reconstruction stencils at the boundary of the computational domain, except for periodic boundary conditions. Throughout this article, except for periodic boundary conditions, we choose generally one-sided stencils, i.e. stencils that lie completely inside the computational domain. Without changing the stencil search algorithm described in [17], the stencils at the boundary are simply generated by adding to the stencil recursively the available direct neighbors of the elements already in the stencil until the required number of elements n_e is reached, starting always with the central element for which reconstruction is to be performed. This automatically leads to one-sided stencils at the boundary of Ω.

For the flux computation at the boundary, we then solve so-called inverse Riemann problems. This means that at a boundary interface we have to prescribe some *data outside* the computational domain as argument for the numerical flux to obtain the correct numerical flux *at* the interface. In other words, we must find the initial condition for the corresponding Riemann problem at the boundary such that its solution at the interface yields the physically required condition. For inflow boundaries, all values for density, velocity and pressure at the boundary $\partial\Omega$ are prescribed. The Riemann solver then automatically chooses the correct interface values due to upwinding. For transmissive boundaries, the unknown data outside the domain (outer data) is simply chosen to be equal to the reconstructed data inside the computational domain (inner data). Finally, for solid slip-wall boundary conditions we mirror the normal velocity component i.e. we change the sign of the normal velocity for the outer data. Then we copy all other velocity components as well as density and pressure from the inner data. This procedure assures the resulting Riemann problem at the boundary to be symmetric and to yield exactly a zero normal velocity at the boundary. The correct pressure on the wall boundary will then be automatically computed by the numerical flux. The performance of one-sided reconstruction together with the solution of inverse Riemann problems at reflecting walls has been studied for linear hyperbolic systems in detail for ADER finite volume schemes in [16], where the method has been validated using analytic reference solutions and where it has been also compared with standard and staggered-grid finite difference methods. We note that the solution of inverse Riemann problems is not equivalent to the method of ghost-points that is commonly used to impose boundary conditions for high-order finite difference schemes.

For curved boundaries, high order boundary representation must be used in order to achieve the desired order of accuracy, see [4,37]. At curved walls, we drop the quadrature-free approach and use classical multi-dimensional Gaussian quadrature formulae in space and time [44] to compute the flux integrals. Since this is only done on the boundaries, it does not lead to a significant increase in CPU time.

At this point we also would like to make some remarks on the MPI parallelization of ADER finite volume schemes on modern massively parallel systems using the MPI standard. Since the ADER-FV method leads to an explicit one-step scheme in time, the total communication overhead is considerably low compared to methods using high order Runge–Kutta time integration because data has to be exchanged only at the beginning of each time step. Then, each sub domain can evolve the solution independently of its neighbor domains. Unstructured mesh partitioning is done with the free METIS software package described in [32]. For our parallel implementation, we decided to split the MPI communications for ADER-FV into two parts. Before reconstruction can be done, the finite volume scheme must first exchange the necessary cell averages $\overline{U}_p$ required for the reconstruction procedure and subsequently each sub-domain can perform the reconstruction. In a second step, the reconstructed degrees of freedom $\hat{w}_{pl}$ are exchanged only between the direct neighbors of

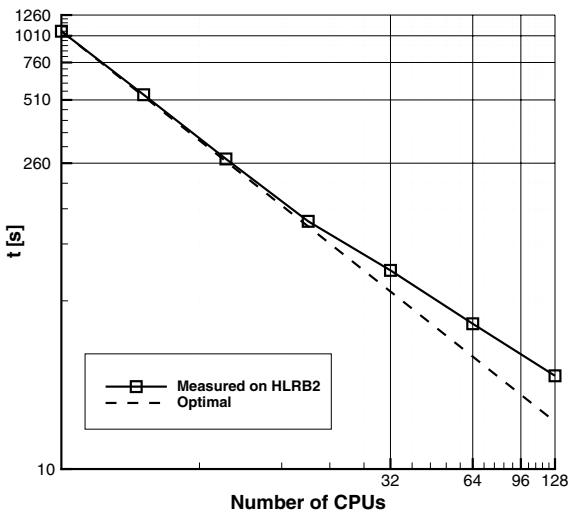


Fig. 2. MPI speedup measured on the HLRB2 supercomputer in München, Germany, using the fourth order ADER-FV scheme on an unstructured tetrahedral mesh in 3D. The solid line shows the wallclock time needed for a fixed problem on 2, 4, 8, 16, 32, 64 and 128 CPUs, respectively. The dashed line shows the theoretical optimum for comparison, supposing 100% MPI efficiency.

a sub-domain boundary. To assess the parallel efficiency, we run a fixed three-dimensional problem using the fourth order quadrature-free ADER finite volume scheme on an unstructured tetrahedral mesh with 69,120 elements. The problem is the same as used for the three-dimensional convergence studies presented in Section 6.2, corresponding to the $N_G = 24$, $M = 3$ case shown in Table 6. In Fig. 2 we show that our code scales quite well up to 128 CPUs. We emphasize that we used a very simple MPI implementation that does not use any special technique to hide MPI latency times.

Typical runs of our parallel ADER-FV code on the HLRB2 supercomputer of the *Leibniz Rechenzentrum* (LRZ) in München, Germany, use between 64 and 128 CPUs. For our largest computation, shown in Section 7.10 of this article, we used 1024 CPUs obtaining a parallel efficiency of about 40%.

6. Numerical convergence studies

6.1. Convergence studies in two dimensions

We consider the smooth two-dimensional example of a convected isentropic vortex given for example by Hu and Shu in [30]. The governing equations are the nonlinear two-dimensional Euler equations (23) with the equation of state (24) and without source term. The initial condition is a linear superposition of a homogeneous background field and some perturbations δ:

$$(\rho, u, v, p) = (1 + \delta\rho, 1 + \delta u, 1 + \delta v, 1 + \delta p). \tag{44}$$

The perturbations of the velocity components u and v as well as the perturbations of entropy $S = \frac{p}{\rho^\gamma}$ and temperature T of the vortex are given by

$$\begin{pmatrix} \delta u \\ \delta v \end{pmatrix} = \frac{\epsilon}{2\pi} e^{\frac{1-r^2}{2}} \begin{pmatrix} -(y-5) \\ (x-5) \end{pmatrix}, \quad \delta S = 0, \quad \delta T = -\frac{(\gamma-1)\epsilon^2}{8\gamma\pi^2} e^{1-r^2}, \tag{45}$$

with $r^2 = (x-5)^2 + (y-5)^2$ and the vortex strength $\epsilon = 5$. If we define the relationship between density, pressure and static temperature in a nondimensional fashion so that the gas constant becomes equal to unity, we obtain the following perturbations of the primitive variables density and pressure:

$$\delta\rho = (1 + \delta T)^{\frac{1}{\gamma-1}} - 1, \quad \delta p = (1 + \delta T)^{\frac{\gamma}{\gamma-1}} - 1. \tag{46}$$

The computational domain is $\Omega = [0;10] \times [0;10]$ and four periodic boundary conditions are imposed. After one period of $t = 10$, the exact solution is given by the initial condition (44). For measuring the error between the numerical solution u_h and the exact solution u_e, we first apply the reconstruction operator in order to get w_h from u_h and then we use the continuous L^p-norms

$$\|w_h - u_e\|_{L^p(\Omega)} = \frac{1}{|\Omega|} \left(\int_\Omega |w_h - u_e|^p \mathrm{d}V \right)^{\frac{1}{p}}, \tag{47}$$

in which the integration has been approximated using Gaussian integration formulae with twice the order of accuracy of the numerical scheme. The area of the computational domain is $|\Omega| = 100$. The L_∞ norm has been approximated by taking the maximum error arising in any of the Gaussian integration points without division by $|\Omega|$.

The numerical convergence rates for the nonlinear ADER-FV schemes up to sixth order of accuracy are given in Table 4. The errors presented are those for the density ρ. Similar results are obtained for the other primitive flow quantities u, v and p. The CPU time given in the tables is the one obtained with the serial version of the code on one processor of an Intel Xeon dual-processor workstation with 2×3.6 GHz and 4 GB of RAM. Computations have been performed on a sequence of red-refined triangular grids, see Fig. 3. The leftmost grid in Fig. 3 is defined to be the reference grid with the characteristic mesh length $h = h_0$. We note that our triangular grids are very similar to the irregular grids used by Hu and Shu [30] and hence the results are comparable. For convenience, the results presented in [30] are repeated in Table 5. However, we emphasize that we compute the solution up to $t = 10$, which is *five times* the output time used in [30], where the error norms are computed already at $t = 2.0$. We note that on finer grids the numerical errors obtained with the

Table 4
Numerical convergence results obtained with ADER-FV schemes from third to sixth order in space and time for the two-dimensional vortex test case at $t = 10.0$

$\frac{h_0}{h}$	L^∞	L^1	L^2	$\mathcal{O}_{L^\infty}$	$\mathcal{O}_{L^1}$	$\mathcal{O}_{L^2}$	t_{CPU} (s)
ADER-FV $\mathcal{O}3$ ($M = 2$)							
2	4.1966E−01	2.5803E−02	5.5744E−03				1
4	2.0603E−01	1.0377E−02	2.5127E−03	1.0	1.3	1.1	9
8	3.9381E−02	2.0124E−03	4.7131E−04	2.4	2.4	2.4	72
16	6.4677E−03	3.8149E−04	8.4719E−05	2.6	2.4	2.5	583
32	8.8072E−04	5.2530E−05	1.1666E−05	2.9	2.9	2.9	4607
ADER-FV $\mathcal{O}4$ ($M = 3$)							
2	3.7427E−01	2.0632E−02	4.7927E−03				2
4	5.2403E−02	4.1394E−03	7.4081E−04	2.8	2.3	2.7	14
8	1.0180E−02	4.5537E−04	8.6607E−05	2.4	3.2	3.1	114
16	3.6210E−04	2.5185E−05	4.5212E−06	4.8	4.2	4.3	910
32	1.6601E−05	1.0891E−06	1.8424E−07	4.4	4.5	4.6	7188
ADER-FV $\mathcal{O}5$ ($M = 4$)							
2	3.4130E−01	1.8162E−02	4.2424E−03				3
4	4.3610E−02	2.8756E−03	5.4369E−04	3.0	2.7	3.0	21
8	8.4151E−03	3.6375E−04	7.6764E−05	2.4	3.0	2.8	172
16	2.9109E−04	1.6616E−05	3.6625E−06	4.9	4.5	4.4	1364
32	1.0793E−05	5.7088E−07	1.3018E−07	4.8	4.9	4.8	11,010
ADER-FV $\mathcal{O}6$ ($M = 5$)							
2	2.1257E−01	1.9073E−02	2.9774E−03				3
4	3.7012E−02	2.2336E−03	3.6602E−04	2.5	3.1	3.0	32
8	1.2839E−03	9.6264E−05	1.7198E−05	4.8	4.5	4.4	261
16	3.4407E−05	1.6378E−06	3.5529E−07	5.2	5.9	5.6	2122
32	5.3451E−07	2.7486E−08	4.7517E−09	6.0	5.9	6.2	16,975

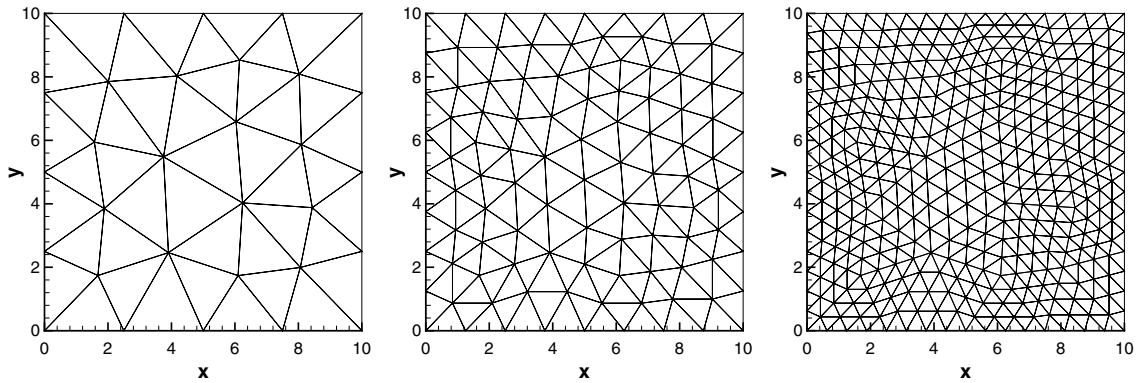


Fig. 3. Sequence of red-refined triangular meshes used for the two-dimensional convergence studies.

third order ADER-FV scheme are lower than the third order Runge–Kutta WENO finite volume scheme, despite the much longer simulation time. The fourth order ADER-FV scheme still reaches lower error norms after $t = 10$ than the WENO scheme of Hu and Shu after $t = 2$. However, we stress that in [30] only a third order TVD Runge–Kutta time stepping method has been used so that the time step was reduced proportional to $h^{4/3}$ in order to get fourth order of accuracy, whereas we run the ADER-FV scheme with a fixed Courant number of 0.5 and a time step proportional to h. The desired convergence rates at fixed Courant number are still reached for the fifth and sixth order ADER-FV schemes, see Table 4. To our knowledge, this very high order of accuracy has not yet been reached with any other finite volume scheme for the compressible Euler equations on unstructured meshes.

Table 5
Numerical convergence results given by Hu and Shu in [30] for third and fourth order WENO finite volume schemes for the two-dimensional vortex test case at $t = 2.0$

TVD Runge–Kutta WENO $\mathcal{O}3$ [30]					TVD Runge–Kutta WENO $\mathcal{O}4$ [30]			
$\frac{h_0}{h}$	L^∞	L^1	$\mathcal{O}_{L^\infty}$	$\mathcal{O}_{L^1}$	L^∞	L^1	$\mathcal{O}_{L^\infty}$	$\mathcal{O}_{L^1}$
2	3.33E−001	2.1200E−02			2.1400E−01	1.8400E−02		
4	2.27E−001	1.2800E−02	0.6	0.7	3.4300E−02	2.8000E−03	2.6	2.7
8	6.85E−002	3.8400E−03	1.7	1.7	6.5700E−03	2.1200E−04	3.7	3.7
16	3.02E−002	8.3200E−04	2.2	2.2	5.9100E−04	1.0900E−05	4.3	4.3
32	5.64E−003	1.2600E−04	2.7	2.7	1.9700E−05	3.7600E−07	4.9	4.9

6.1.1. Computational cost evaluation

To give the reader a possibility to assess the computational effort associated with the presented ADER-FV methods for the two dimensional compressible Euler equations in a fully unstructured framework in comparison to ADER-FV schemes on Cartesian meshes [47] and in comparison to the classical WENO finite difference schemes [31], we present the following quantitative cost comparison: all CPU times have been obtained for the two-dimensional compressible Euler equations using the same compiler on the same Intel Xeon workstation using one single CPU with 3.6 GHz and 4 GB of RAM. The given CPU times are defined as the total time needed for a complete time step of a single element, including reconstruction, flux computation and time integration for all variables of the Euler equations. We measured the total CPU time necessary per element update to be 45 μs for the Cartesian third order ADER-FV code of Titarev and Toro [47] and 75 μs for the unstructured third order ADER-FV schemes presented in this paper. Furthermore, the fourth order Cartesian ADER-FV approach requires 112 μs whereas the fourth order unstructured ADER-FV method presented in this paper needs 120 μs per element update. The classical fifth order WENO finite difference method [31] using third order TVD Runge–Kutta time discretization needed 15 μs per element update on a Cartesian grid. The increasing efficiency of the higher order unstructured ADER-FV algorithm presented in this article compared to the Cartesian ADER-FV method could be due to a more efficient implementation of the Cauchy–Kovalewski procedure as well as the quadrature-free implementation. However, we note that for such complex algorithms, many other implementation details must also be taken into consideration. When comparing structured and unstructured approaches it is important to mention that to mesh the same computational domain with either triangles or quadrilaterals of equal maximal edge length, one needs about *two times* more triangles than quadrilaterals and simultaneously one must accept also a more restrictive explicit time step restriction since the incircle diameter, which is used as a characteristic length scale for the CFL criterion, can be about a factor of two smaller in irregular triangles compared to regular rectangles of equal maximal edge length.

6.2. Convergence studies in three dimensions

In order to study the convergence behaviour of our method for the three dimensional compressible Euler equations we *create* a smooth unsteady test case with exact reference solution by prescribing a vector $U_p(\vec{x}, t)$ which when substituted into (20) produces a modified Euler system with a source term. Note that for the test to be useful the method must be also capable of computing solutions to inhomogeneous problems, that is with a non-vanishing right-hand side, to the required order of accuracy.

Therefore, we now solve the three-dimensional Euler equations with source terms, as given by (20)–(23), in the domain $\Omega_{3D} = [-0.5; 0.5]^3$ with six periodic boundary conditions and the following space–time dependent source term:

$$\begin{pmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \end{pmatrix} (\vec{x}, t) = \begin{pmatrix} \omega A_0 \sin(\omega t - \vec{k}\vec{x}) \\ -k_x A_0 \sin(\omega t - \vec{k}\vec{x}) \\ -k_y A_0 \sin(\omega t - \vec{k}\vec{x}) \\ -k_z A_0 \sin(\omega t - \vec{k}\vec{x}) \\ \frac{1}{\gamma - 1} \omega A_0 \sin(\omega t - \vec{k}\vec{x}) \end{pmatrix}. \tag{48}$$

The exact solution to the problem, serving also as initial condition, is

$$\begin{pmatrix} \rho \\ \vec{V} \\ p \end{pmatrix}(\vec{x},t) = \begin{pmatrix} 2 + A_0 \sin(\omega t - \vec{k}\vec{x}) \\ \vec{0} \\ 2 + A_0 \sin(\omega t - \vec{k}\vec{x}) \end{pmatrix}, \tag{49}$$

inserting (49) and (48) into (23). We note that the solution is simple in structure but non-trivial. In particular, during a numerical simulation the velocity components will show values different from zero due to numerical errors since numerical schemes will in general not be able to balance exactly the nonlinear convective flux with the source term. Therefore, this example is also a very useful validation of the capability of our method to balance the source term with the fluxes. The constants are set to be $A_0 = 1.0$ and $\vec{k} = (k_x, k_y, k_z)$ with $k_x = k_y = k_z = \omega = 2\pi$. The convergence rates for our proposed quadrature-free ADER finite volume schemes from third to sixth order of accuracy in space and time are presented in Table 6 for the velocity in x-direction, i.e. for the primitive variable u. Similar results are obtained for all the other variables. The error norms are computed in the same way as for the two dimensional convergence studies. We use the characteristic WENO reconstruction with the following parameters: $r = 4$, $\lambda_1 = 10^5$ and $\epsilon = 10^{-14}$. For the convergence studies, the exact Riemann solver is used for the leading flux, together with the Roe-type corrector. The CPU times that are given in Table 6 are for the parallel version of the code running on four Intel Itanium Madison CPUs (each with 1.6 GHz clock speed and 8 GB of RAM) of the HLRB2 supercomputer at LRZ München. The numerical convergence study on unstructured three dimensional tetrahedral meshes has the twofold purpose of verifying

Table 6
Numerical convergence results obtained with ADER-FV schemes from third to sixth order in space and time for the three-dimensional test case after $t = 0.25$

N_G	L^∞	L^1	L^2	$\mathcal{O}_{L^\infty}$	$\mathcal{O}_{L^1}$	$\mathcal{O}_{L^2}$	t_{CPU} (s)
ADER-FV $\mathcal{O}3$ ($M = 2$)							
8	2.3265E−02	8.7869E−03	1.0185E−02				4
12	8.0369E−03	2.4689E−03	2.9695E−03	2.6	3.1	3.0	17
16	3.5522E−03	1.0640E−03	1.2935E−03	2.8	2.9	2.9	54
20	1.7930E−03	5.4621E−04	6.6247E−04	3.1	3.0	3.0	130
24	9.9836E−04	3.1449E−04	3.8055E−04	3.2	3.0	3.0	260
32	4.1918E−04	1.3316E−04	1.6143E−04	3.0	3.0	3.0	820
ADER-FV $\mathcal{O}4$ ($M = 3$)							
8	6.0763E−03	1.5265E−03	1.9928E−03				8
12	1.2779E−03	2.8327E−04	3.8575E−04	3.8	4.2	4.0	33
16	4.1355E−04	8.6796E−05	1.1926E−04	3.9	4.1	4.1	109
20	1.7613E−04	3.5111E−05	4.8176E−05	3.8	4.1	4.1	262
24	8.8666E−05	1.6933E−05	2.3075E−05	3.8	4.0	4.0	524
32	2.9091E−05	5.3556E−06	7.2376E−06	3.9	4.0	4.0	1681
ADER-FV $\mathcal{O}5$ ($M = 4$)							
4	8.2030E−02	2.9604E−02	3.6341E−02				1
8	3.4650E−03	1.4997E−03	1.7384E−03	4.6	4.3	4.4	13
12	5.9061E−04	1.9677E−04	2.3619E−04	4.4	5.0	4.9	71
16	1.4233E−04	4.6959E−05	5.6289E−05	4.9	5.0	5.0	231
20	4.4338E−05	1.5301E−05	1.8369E−05	5.2	5.0	5.0	558
24	2.0926E−05	6.1352E−06	7.4011E−06	4.1	5.0	5.0	1137
ADER-FV $\mathcal{O}6$ ($M = 5$)							
4	3.1815E−02	1.1310E−02	1.3915E−02				1
8	1.3728E−03	1.4197E−04	1.7917E−04	4.5	6.3	6.3	20
12	2.1289E−04	1.0618E−05	1.6858E−05	4.6	6.4	5.8	143
16	3.1281E−05	1.6605E−06	2.3375E−06	6.7	6.4	6.9	467
20	7.4469E−06	4.1553E−07	6.1345E−07	6.4	6.2	6.0	1153
24	2.2733E−06	1.3416E−07	1.9236E−07	6.5	6.2	6.4	2304

the designed order of accuracy of the quadrature-free unstructured ADER finite volume schemes proposed in this paper, as well as to validate the MPI parallelization of our code. From our results we may conclude that the method converges with the designed order of accuracy and that for very high levels of accuracy the higher order methods are more efficient than the lower order schemes. On the finest mesh we almost reach machine precision for single precision computations which indicates that our very high order schemes are capable of maintaining an excellent balance of the source term and the nonlinear convection terms. The good balancing properties come out automatically from the high order framework and the Cauchy–Kovalewski procedure, through which the numerical flux takes into account the presence of the source term.

7. Applications in two and three space dimensions

The governing equations solved for all the following numerical examples are the nonlinear Euler equations of compressible gasdynamics, see (20)–(23).

7.1. Subsonic flow past a circular cylinder in two dimensions

We first compute the subsonic steady-state flow past a circular cylinder with radius $R = 1$ in two space dimensions. The computational domain is chosen to be $\Omega = [-10;10] \times [-10;10]$, where the circular cylinder has been cut out. The inflow condition is given by the Mach number $M_\infty = 0.1$, the density $\rho_\infty = 1$ and the pressure $p_\infty = 1/\gamma$ with $\gamma = 1.4$. The incoming flow is parallel to the x-axis. In this subsonic flow regime the potential flow approximation is valid and therefore serves as an excellent reference solution. The radial and tangential velocity components v_r and v_θ as well as the pressure field p and the pressure coefficient c_p of the potential flow are given analytically by

$$v_r = M_\infty \sqrt{\frac{\gamma p_\infty}{\rho_\infty}} \left(1 - \frac{R^2}{r^2}\right) \cos\theta, \quad v_\theta = -M_\infty \sqrt{\frac{\gamma p_\infty}{\rho_\infty}} \left(1 + \frac{R^2}{r^2}\right) \sin\theta, \tag{50}$$

$$p(r,\theta) = p_\infty \left(1 + \frac{1}{2}\gamma M_\infty^2\right) - \frac{1}{2}[v_r^2 + v_\theta^2], \quad c_p = \frac{p - p_\infty}{\frac{1}{2}\gamma p_\infty M_\infty^2}. \tag{51}$$

The computation is carried out on an irregular unstructured triangular mesh where the cylinder surface is discretized using 60 elements. The total number of triangles is 6576. We choose an ADER-FV scheme using WENO reconstruction of polynomial degree $M = 3$. The curved cylinder wall is intentionally represented by piecewise linear segments in this case to compare the behaviour of our approach with other methods. For the initial condition we take the constant inflow everywhere in the domain. The computation was performed up to the final time of $t = 200$, where the residuals have decreased sufficiently after 6500 iterations. The triangular mesh together with the c_p distribution of our numerical results as well as a quantitative comparison with the potential flow model is depicted in Fig. 5, where the pressure coefficient c_p has been plotted over the azimuthal angle θ on three concentric circles with radii $r = 1$, $r = 2$ and $r = 3$. We note that the potential flow is symmetric with respect to the coordinate axes and has two stagnation points. Our numerical simulation respects this symmetry quite well and the c_p distribution is in good agreement with the reference solution. We remark that high order discontinuous Galerkin and spectral finite volume schemes may have severe problems with this test case on unstructured meshes when no curvilinear boundary representation is used, see [5,20,57]. In these references, unphysical and unsteady vortex shedding behind the cylinder was reported using high order schemes combined with piecewise linear boundaries. In contrast, high order finite volume schemes seem to be less sensitive to this problem.

7.2. Ringleb flow

The flow problem considered in this section is the transonic Ringleb flow [8], which is one of the few analytical solutions known for the steady compressible two-dimensional Euler equations. It is computed in the hodograph (V–θ) plane, where V is the velocity magnitude and θ is the angle of the flow with respect to

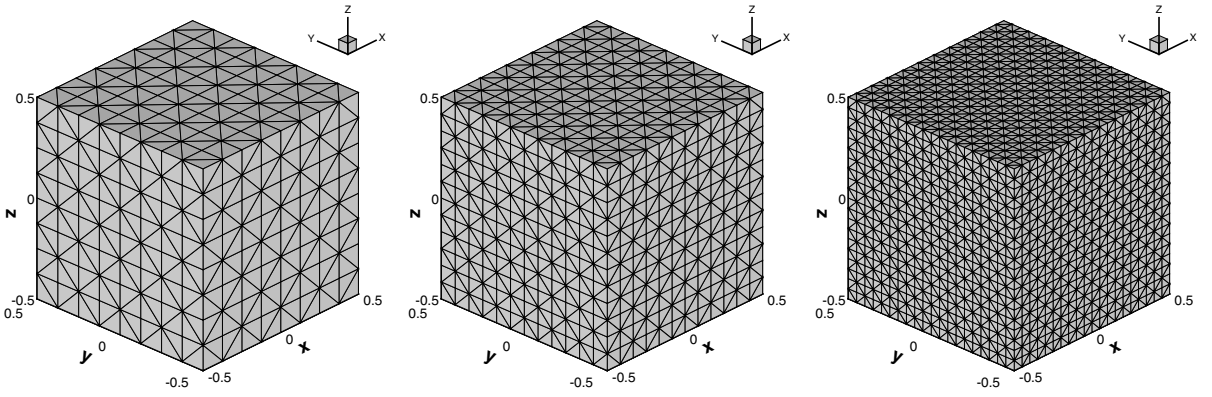


Fig. 4. Sequence of regular tetrahedral meshes used for the three-dimensional convergence studies.

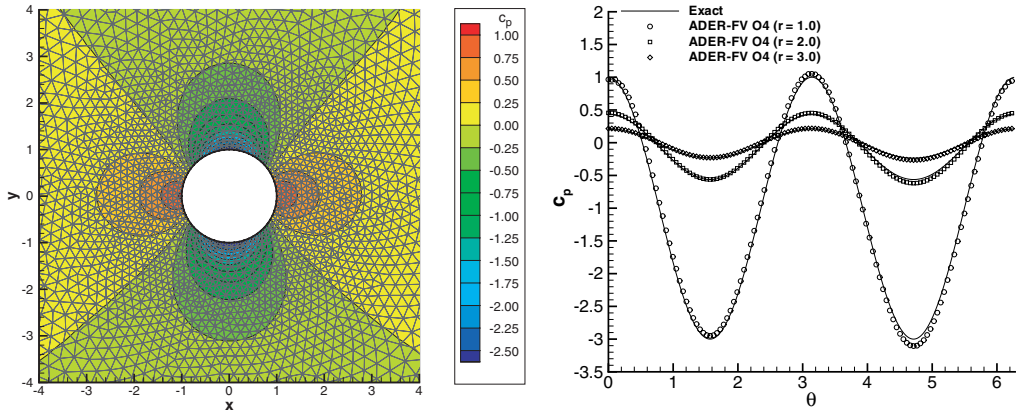


Fig. 5. Mesh and c_p distribution for the subsonic flow past a circular cylinder obtained with an ADER-FV scheme ($M = 3$) using nonlinear WENO reconstruction (left) and comparison of c_p with the potential flow solution (right).

the x-axis. The stream function Ψ of the Ringleb flow is given in the hodograph plane by the simple expression $\Psi = \sin\theta/V$, which results in the following analytical expressions for the streamlines in the physical x–y plane:

$$x = \frac{1}{2}\frac{1}{\rho}\left(\frac{1}{V^2} - \frac{2}{k^2}\right) + \frac{J}{2}, \quad y = \pm\frac{1}{k\rho V}\sqrt{1 - \left(\frac{V^2}{k^2}\right)}, \tag{52}$$

with the definitions

$$J = \frac{1}{a} + \frac{1}{3a^3} + \frac{1}{5a^5} - \frac{1}{2}\ln\frac{1+a}{1-a}, \tag{53}$$

$$a = \sqrt{1 - \frac{\gamma - 1}{2}V^2}, \quad \rho = a^{\frac{2}{\gamma-1}}, \quad k = \frac{1}{\Psi}. \tag{54}$$

It is easy to check that the isotach lines are circles and verify the condition

$$\left(x - \frac{J}{2}\right)^2 + y^2 = \frac{1}{4\rho^2 V^4}. \tag{55}$$

The geometry of the problem is chosen identical to the one proposed in [57]. It is defined by two solid walls that are obtained from two streamlines according to (52) setting $\gamma = 1.4$, $k_1 = 0.7$ and $k_2 = 1.2$. The inflow and outflow boundary conditions, where the exact solution of the Ringleb flow is imposed, are given by the

corresponding segments of the isotach line for $V = 0.5$. The computation is perf
cessively refined irregular triangular meshes, see Fig. 6, with the same resolution
gular grids used in [57]. Since the flow is completely smooth in the whole computa
analytic solution is exact, we provide also a quantitative validation of our schem
obtained for all flow quantities using ADER-FV schemes with nonlinear WENO r
degree $M = 3$ in characteristic variables, choosing $n_e = 1.25L$, $\epsilon = 10^{-14}$, $r = 2$ a
flux together with the Roe-type corrector is used. The exact solution serves as ini
lem. Two computations are performed: one with piecewise linear boundary repres
wise cubic boundaries. Concerning practical implementation, we give up the q
those boundaries which are curvilinear, using classical Gaussian quadrature for
evaluate the flux integrals, see [4,37]. Data at the boundaries are given by the Cau
The proposed quadrature-free approach is still used on all interior edges so that th
to a significant increase of computational effort.

For the piecewise linear boundary we cannot expect convergence rates to be
which is also directly confirmed by the first part of the convergence results give
would like to make the following two remarks: first, complex solid boundaries ari
are not always smooth, but may also be rough, which means that coarse elements c
order polynomial mapping may not be an ideal choice. Second, it is very importan
at least able to maintain the converged steady state solution of this difficult test ca
piecewise linear boundary approximation, see Fig. 7. In contrast, for example th
volume method fails for this test case if no high order curvilinear boundary discretiz
fail for high order boundaries on coarse meshes, see [57]. All our computations ha
long period of 10^4 time steps to be able to detect any divergent behaviour of the r
coarsest mesh, which is the most critical one concerning the preservation of the s
numerical solution remained perfectly steady even after 10^5 time steps. For compa
sented in [57] were carried out over 8000 time steps and it was reported that the
mesh started already after about 2000 time steps.

To reach the full formal order of accuracy, also the boundaries of the computat
resented with higher order polynomial approximations, consistent with the order
inside the computational domain, see [4,37]. This is clearly confirmed by the secon
that with piecewise cubic boundary representation the errors on the intermediate me
than on the finest mesh with piecewise linear boundaries. However, we remark aga
istic applications may not always be as smooth as the ones used here in this test c

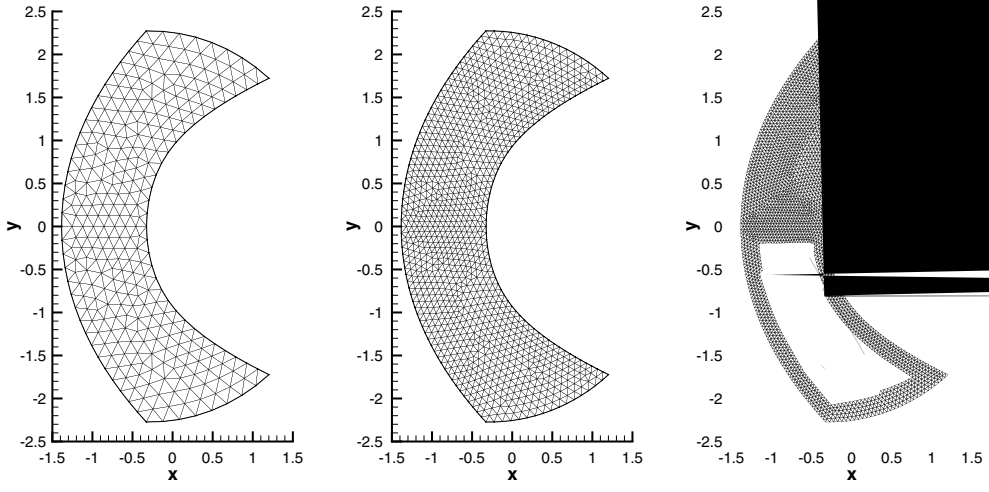


Fig. 6. Mesh sequence used for the Ringleb flow computations.

Fig. 7. Mach contour lines for the Ringleb flow after 10,000 time steps on the mesh sequence of Fig. 6 using ADER-FV ($M = 3$) with piecewise linear boundaries.

The undisturbed upstream temperature T_0 can be computed from the ideal gas law $p = \rho RT$, where we set the gas constant to $R = 1$. The temperature inside the vortex is then given as the solution of the following ordinary differential equation:

$$\frac{\mathrm{d}T}{\mathrm{d}r} = \frac{\gamma - 1}{R\gamma} \frac{v_\phi(r)^2}{r}. \tag{58}$$

We furthermore define the Mach number $M_v = v_m/c_0$ to characterize the vortex strength where c_0 is the speed of sound on the upstream side of the shock. The parameters used for the following computations are: $a = 0.075$, $b = 0.175$, $M_s = 1.5$ and $M_v = 0.7$. The undisturbed upstream values outside the vortex are $\rho_0 = 1$, $u_0 = 1.5\sqrt{\gamma}$ and $p_0 = 1$. The downstream (post-shock) values are given according to the Rankine–Hugoniot relations as follows: $\rho_1 = \frac{9\gamma+9}{9\gamma-1}$, $u_1 = \frac{(9\gamma-1)}{6(\gamma+1)}\sqrt{\gamma}$ and $p_1 = \frac{7\gamma+2}{2\gamma+2}$.

A contour plot of the density of the initial condition is given in Fig. 8, where a typical unstructured mesh for this problem with characteristic mesh length $h = 1/50$ is also depicted. The numerical results obtained at $t = 0.7$ on triangular meshes with $h = 1/200$ and $h = 1/500$ using ADER finite volume schemes from second to fourth order of accuracy can be seen in Fig. 9. The results on the fine mesh using schemes with higher order than two are in very good agreement with the solution given in [38], obtained with a third order ENO scheme on a Cartesian grid with a mesh spacing of $h = 1/512$. For a physical discussion of the results, see [38] and references therein, since in this article we focus mainly on the numerical aspects of the proposed ADER-FV schemes. From Fig. 9 one can clearly see the drastic increase of the resolution of the acoustic waves generated in this problem when using high order schemes. It is clear that second order TVD methods, which are currently the state-of-the-art methods used in most finite volume based unstructured CFD solvers, are not able to capture the emitted sound field and the discontinuous flow structures of this problem simultaneously.

7.4. Interaction of a shock wave with a wedge in two dimensions

To validate the correct treatment of solid wall boundary conditions we now apply the proposed quadrature-free ADER-FV scheme to a flow that involves the interaction of a mild shock wave with a two dimensional wedge. For this test case, excellent experimental reference data is available in form of Schlieren photographs [54,40].

The computational domain Ω has the extent $\Omega = [-2;6] \times [-3;3]$, where the tip of a wedge with length $L = 1$ and height $H = 1$ is placed at $x = 0$. On all three edges of the wedge as well as on the top and on the bottom of

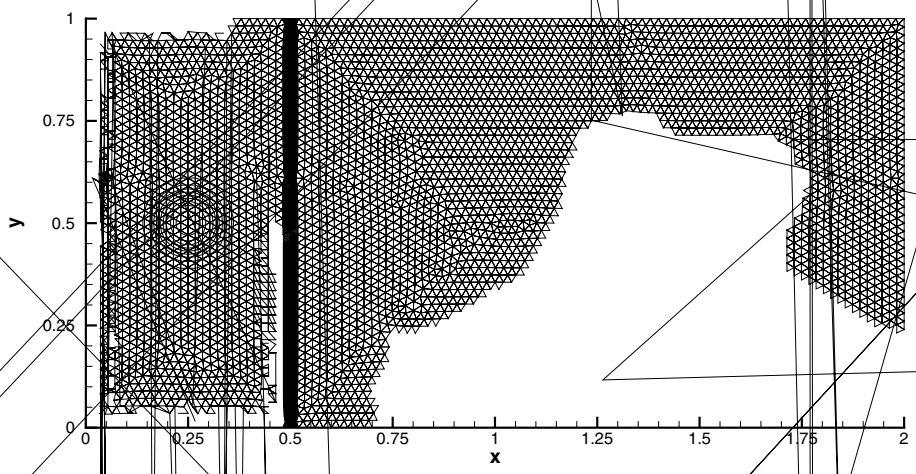


Fig. 8. Unstructured mesh ($h = 1/50$) and 30 equally spaced density contours from 0.5 to 2 showing the initial condition of the shock–vortex interaction problem.

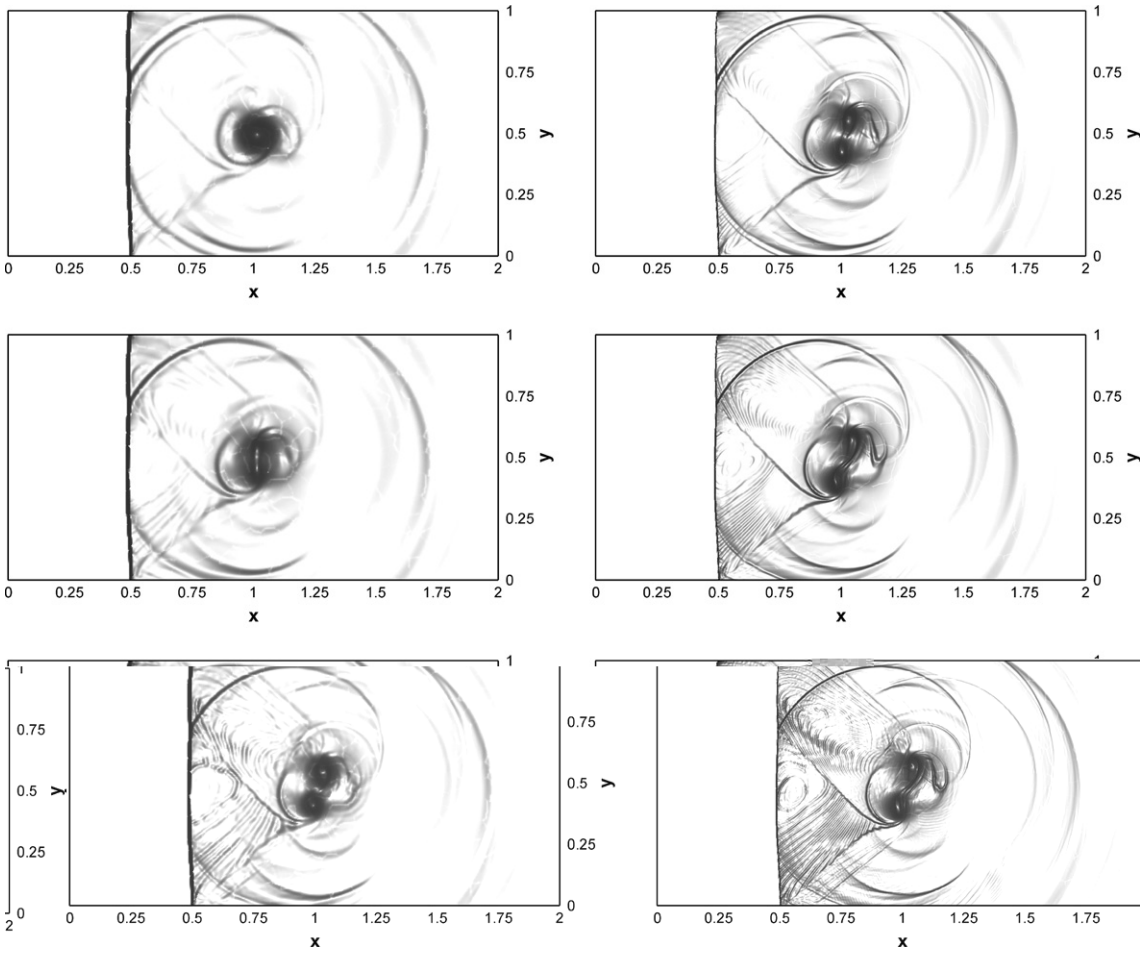


Fig. 9. Numerical Schlieren images at time $t = 0.7$ for the shock–vortex interaction problem using ADER-FV schemes from second to fourth order of accuracy (top to bottom) and a mesh spacing of $h = 1/200$ (left) and $h = 1/500$ (right), respectively.

the computational domain we impose solid (reflective) wall boundary conditions. On the left and on the right, the exact solution according to the Rankine–Hugoniot conditions is imposed as inflow boundary condition.

The initial condition for the shock wave with shock Mach number $M_s = 1.3$ is given according to the Rankine–Hugoniot relations by

$$(\rho, u, v, p)(\vec{x}, 0) = \begin{cases} (2.122, 0.442, 0.0, 1.805) & \text{if } x < -1.0, \\ (1.4, 0.0, 0.0., 1.0) & \text{if } x \geqslant -1.0. \end{cases} \tag{59}$$

We use a triangular mesh of a characteristic mesh spacing of $h = 1/100$ and apply the third and fourth order version of the quadrature-free ADER-FV scheme with HLLE leading flux and HLL-type corrector. The mesh consists of about 1.2 million triangles. The numerical Schlieren images obtained with the fourth order method are depicted in Fig. 10. Comparing our results qualitatively with the Schlieren images produced by Schardin [40] we note an excellent agreement. All the reflected and refracted waves as well as the vortices shed from the wedge are resolved correctly. Our set of images corresponds to pictures 2, 5, 8 and 17 shown in [40]. Since our computational domain has a larger extent in y-direction than the wind tunnel used for the experiments, the reflection from the upper and lower walls occur later in our computation than in the experiments. The numerical results are in excellent agreement with the experimental reference data. The same results have been obtained with the third order scheme, so that the third order results are not shown here.

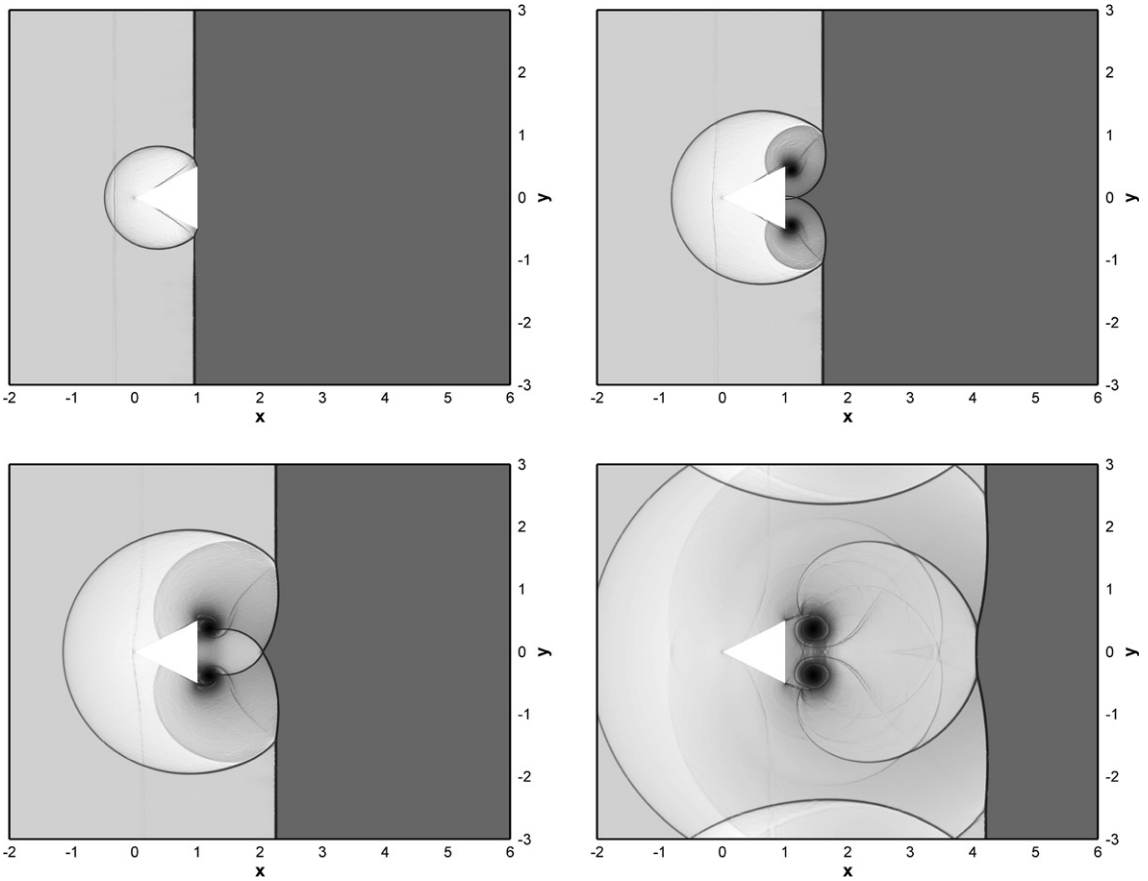


Fig. 10. Numerical Schlieren images for the shock–wedge interaction problem in 2D using the fourth order quadrature-free ADER-FV scheme on an unstructured triangular grid with mesh spacing $h = 1/100$. Output times from top left to bottom right: $t = 1.5$, $t = 2.0$, $t = 2.5$, $t = 4.0$.

7.5. Double Mach reflection problem in two dimensions

Whereas the discontinuities in the previous shock–vortex and shock–wedge interaction examples were rather mild, we now consider a more difficult and very well-known test case, the so-called double Mach reflection problem, originally proposed by Woodward and Colella in [58]. It exhibits very strong discontinuities, wall-bounded flows and furthermore develops rich small-scale structures in time that are difficult to resolve. The test consists in a moving shock wave (shock Mach number $M_s = 10$) that hits a 30° ramp. The initial condition for this problem is given by the Rankine–Hugoniot conditions as follows:

$$(\rho, u, v, p)(\vec{x}, 0) = \begin{cases} (8.0, 8.25, 0.0, 116.5) & \text{if } x < 0.1, \\ (1.4, 0.0, 0.0., 1.0) & \text{if } x \geqslant 0.1. \end{cases} \tag{60}$$

Due to our unstructured formulation we are able to compute the problem directly in its original physical setup, whereas most of the authors of other publications on this test case rotate the ramp such that it becomes parallel to the x-axis and then specify a special boundary condition on the top for the incoming shock wave. After these modifications, the problem can be computed also on a Cartesian grid. On triangular meshes, one does not need such devices since the geometry can be easily meshed automatically by the mesh generator, simply specifying the desired triangular mesh spacing. We solve the problem with the third and fourth order version of our quadrature-free finite volume schemes using the following characteristic triangular mesh spacings: $h = 1/50$, $h = 1/100$, $h = 1/200$ and $h = 1/400$. Solid wall boundary conditions are imposed on the ramp

and on the upper boundary ($y = 2$). Our results for the density (31 equidistant contour levels from 1.5 to 21.5) are depicted in Figs. 11 and 12. The contour lines agree qualitatively with the results shown in previous publications on this test problem [3,30,31,47]. For the case $h = 1/50$ also the unstructured triangular mesh is depicted, see the top picture in Fig. 11.

The slip surfaces present in this test case may become unstable on sufficiently refined meshes and may start rolling. Although the form of the roll-up and the resulting small-scale structures are unphysical due to the lack of the *physical* molecular viscosity, the amount of rolling is believed to be a qualitative indicator of the amount of *numerical* viscosity introduced by the scheme. In Fig. 13 we therefore show a zoom into the roll-up region obtained with the third order version of our method on the left and obtained with the fourth order variant on the right on the finest mesh with $h = 1/400$. It can be clearly seen in Fig. 13 that on the same mesh the amount of rolling is much larger for the fourth order scheme than for the third order method, underlining the benefits of high order schemes in test cases with rich small scale structures.

We note that the HLLE flux was used as leading flux, in connection with the HLL-type corrector. This seems to be necessary for this test case with very strong shock waves, since the original Godunov flux together with the Roe-type corrector was subject to both very well-known shock wave instabilities present even in first

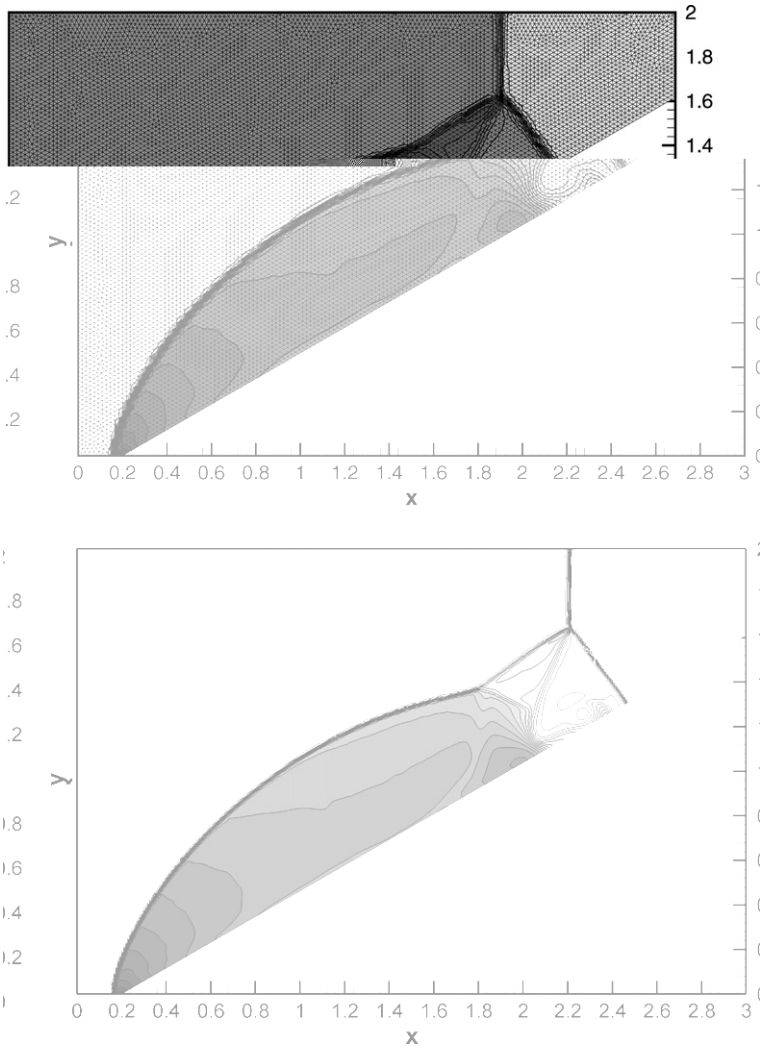


Fig. 11. Density contour levels obtained for the double Mach reflection problem with third order quadrature-free ADER-FV schemes. The mesh spacings are $h = 1/50$ (top) and $h = 1/100$ (bottom). The unstructured triangular mesh is depicted for the case $h = 1/50$.

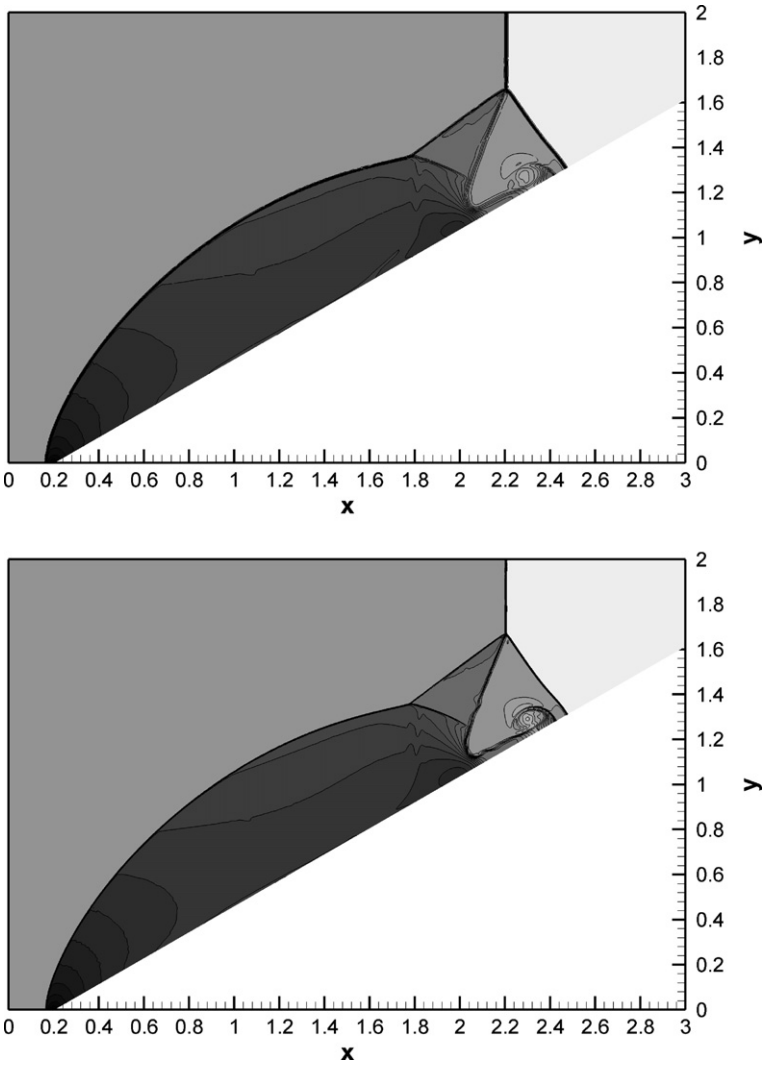


Fig. 12. Density contour levels obtained for the double Mach reflection problem with third order quadrature-free ADER-FV schemes. The mesh spacings are $h = 1/200$ (top) and $h = 1/400$ (bottom).

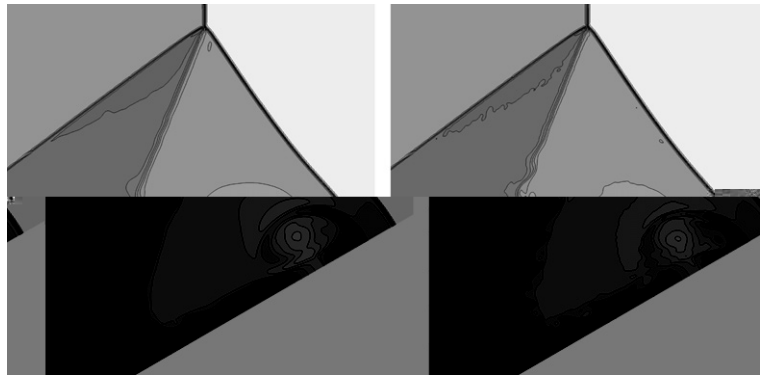


Fig. 13. Zoom into the double Mach reflection problem on the finest mesh ($h = 1/400$) using third order (left) and fourth order (right) quadrature-free ADER-FV schemes.

rder finite volume computations with certain Riemann solvers, namely the kinked Mach stem and he carbuncle phenomenon, see for example [19]. These two shock wave insta hich mainly depend on he choice of the Riemann solver, were even observable in our high order c ns on unstructured triangular meshes in those regions of the mesh where many edges are aligned ow.

6. Shock tube problems in three dimensions

In this section we consider classical one-dimensional shock tube problem computed in a fully three-dimensional setting. We choose a computational domain of the shap r cylinder, as used in real shock tube wind tunnels. The domain has the length $L = 1$, ranging f 5 to $x = +0.5$, with a uniform cylinder radius of $R = 0.1$. The irregular unstructured tetrahedral our computations is depicted in Fig. 14 and contains 60507 tetrahedral elements with a typical f 0.01. This corresponds to an equivalent one-dimensional resolution of 100 cells. We solve t imensional Euler equations without source term with the following initial condition:

$$(\rho, u, v, w, p)(\vec{x}, 0) = \begin{cases} (\rho_L, u_L, 0, 0, p_L) & \text{if } x < 0, \\ (\rho_R, u_R, 0, 0, p_R) & \text{if } x > 0. \end{cases} \tag{61}$$

The values of the left and right initial states for the various test cases are g The first case corresponds to a modification of the standard Sod test case, proposed in [4 sonic point in he rarefaction fan that usually leads to the well-known sonic glitch problem mann solvers sed in first order finite volume schemes. The second test case is also known a ube problem nd was proposed by Lax in [34] and is often computed in the research lit rder WENO chemes. Test case 3 was introduced to show on the one hand the capability of t d to capture iscontinuities that are very close to each other and on the other hand to deal w ssure jumps without producing negative pressure values. Note that the pressure in the initial jumps over ve orders of magnitude from 10^3 down to 10^{-2}. The fourth and last shock tu the particular feature of a very slowly moving shock wave that can cause problems for n d that can ead to spurious oscillations. More detailed information on the setup and the p st cases 3 nd 4 can be found in [48]. For the first and second test case we use the third an adrature-ree finite volume schemes proposed previously in this article and for test cases t he third rder scheme since the adjacent discontinuities in test three and the slowly mo caused rouble for the fourth order method. Probably, for these more severe test cases a ng MP-VENO technique [45,3] should be implemented. A Courant number of CFL lest in-sphere diameter appearing in the entire computational domain, was used in all t ations resented in this section we used the exact Riemann solver for the leading flu rector iven by Eq. (35). The results depicted in Fig. 15 show the reconstructed solu ρ on 00 equidistant sample points taken on the x-axis ($y = z = 0$) at the final outpu le 8. enerally we note that in all test cases only relatively few oscillations are visible ially on-oscillatory character of the proposed nonlinear reconstruction schemes. Fu ties

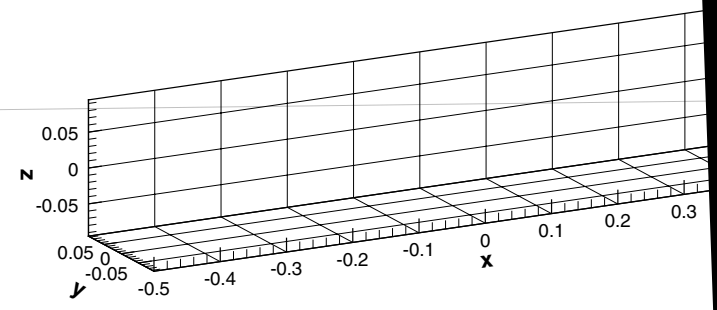


Table 8
Initial states left and right and simulation end times for the 3D shock tube problems

Test case	ρ_L	u_L	p_L	ρ_R	u_R	p_R	t_{end}
1	1.0	0.75	1.0	0.125	0.0	0.1	0.20
2	0.445	0.698	3.528	0.5	0.0	0.571	0.14
3	1.0	0.0	1000.	1.0	0.0	0.01	0.012
4	5.99,924	19.5975	460.895	5.99242	−6.19633	46.0950	0.035

are well resolved within only few intermediate points. In particular, we note that for test case 1 the sonic glitch problem, known for many first order methods, is not present in our third and fourth order computations. It is known that non-entropy satisfying fluxes, especially linearized Riemann solvers, will produce an unphysical rarefaction shock in this point. Also the Lax problem (test case number 2) does not cause many problems for our proposed third and fourth order quadrature-free ADER-FV method with characteristic WENO reconstruction. There are only very small oscillations visible between the contact discontinuity and the shock, which is very well resolved within two points. Our method also behaves very robustly for the more severe test case 3. Only a very small overshoot is visible between the two discontinuities. For the last test case 4 we get good non-oscillatory results for the slowly moving shock wave, see Fig. 15, and only some small oscillations behind the strong right-moving shock.

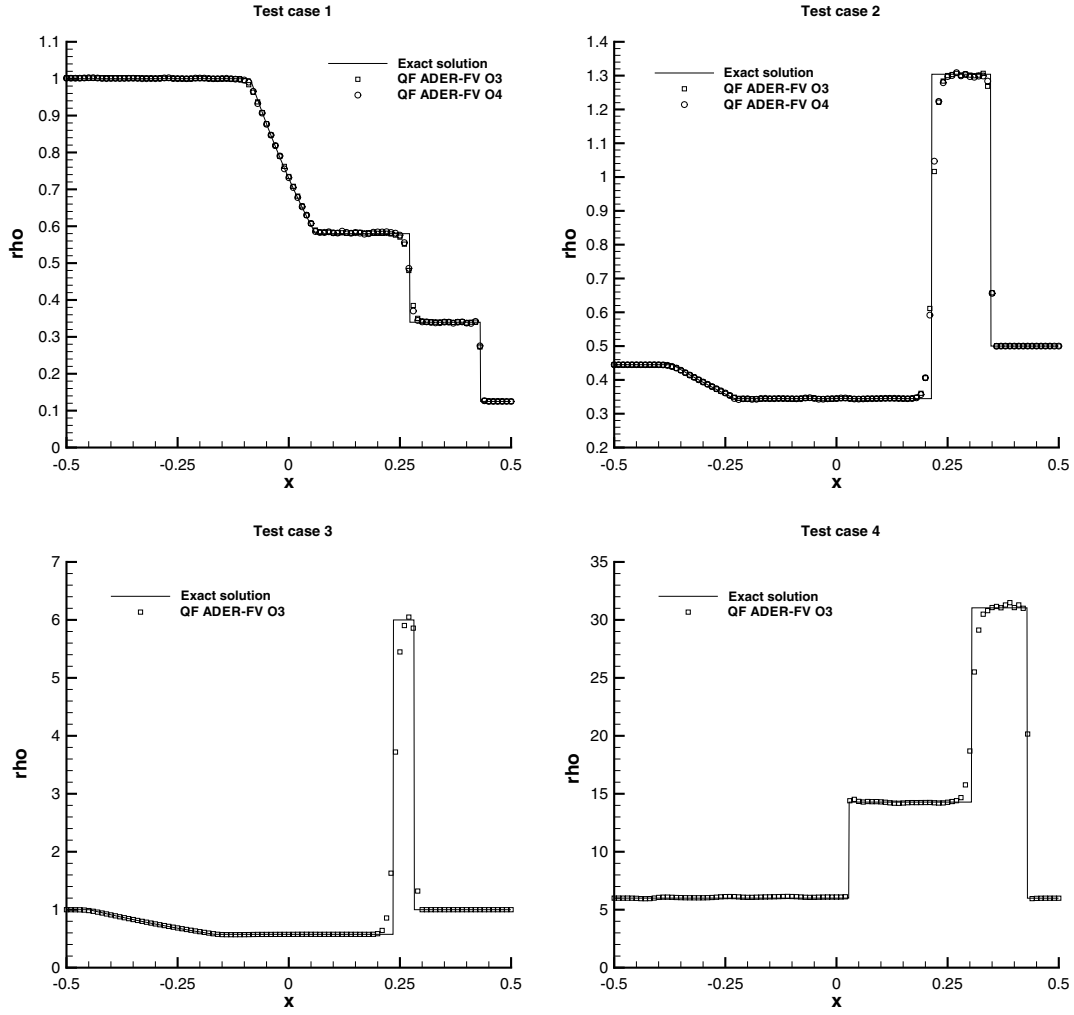


Fig. 15. Results for the 3D shock tube problems computed on the irregular tetrahedral mesh shown in Fig. 14.

Generally, we conclude that the our proposed nonlinear schemes can handle well also solutions that are mainly determined by (strong) shock waves and other discontinuities.

7.7. Shock–density interaction in three dimensions

A very important one-dimensional test case, which will be presented in this section, was proposed by Shu and Osher [42] and was developed to show the advantages of high order schemes. We compute this problem solving the full three-dimensional Euler equations on a three-dimensional computational domain Ω that is defined by $\Omega = [-5;5] \times [-0.075;0.075]^2$. Ω is discretized using a regular tetrahedral mesh as shown in Fig. 4 for the convergence studies. We use a characteristic edge length of 0.025 resulting in 72,000 tetrahedrons. This corresponds to an equivalent one-dimensional resolution of 400 elements in the x-direction.

The solution of this example contains simultaneously a strong shock wave, smooth nonlinear acoustic waves and also entropy waves of high frequency, which are difficult to resolve with low order methods. The initial condition is given by

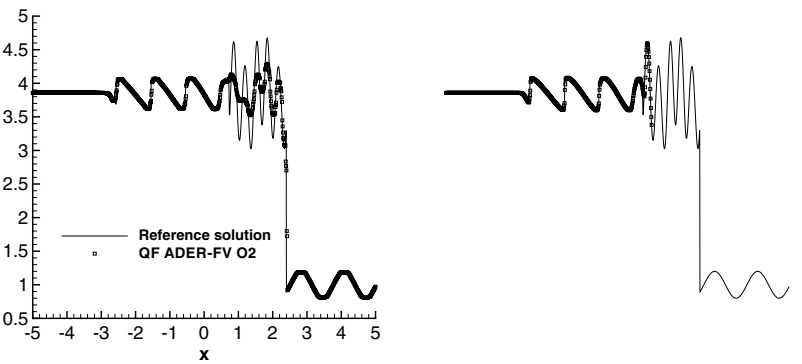
$$(\rho, u, v, w, p)(\vec{x}, 0) = \begin{cases} (3.8571, 2.6294, 0, 0, 10.333) & \text{if } x < -4, \\ (1 + 0.2\sin(5x), 0, 0, 0, 1) & \text{if } x \geqslant -4, \end{cases} \tag{62}$$

which causes the shock to move at a Mach number of $M = 3.0$ into the sinusoidal density fluctuation. We use the proposed quadrature-free ADER-FV schemes of second and fourth order of accuracy with the HLLE flux as leading flux and the HLL-type corrector, as given by Eq. (38). Simulations are performed until the final output time $t = 1.8$. The solution is depicted in Fig. 16 evaluating the reconstruction polynomials w_p for the density ρ on the x-axis ($y = z = 0$) using 1000 equidistantly spaced sample points. Thus, the plots shown in Fig. 16 are oversampled and one can also see the behaviour of the reconstruction polynomials inside the elements.

We clearly see that the resolution of the second order schemes is not sufficient to resolve the high frequency entropy waves. Looking closely at Fig. 16 one can see a TVD-like behaviour of our second order reconstruction since extrema are clipped. The resolution of the fourth order ADER-FV scheme is much better and the smooth high frequency waves behind the shock wave are captured very well. However, albeit the very good resolution of the fourth order scheme, no spurious oscillations are visible in the vicinity of the shock wave. Our numerical solution is in almost perfect agreement with the reference solution which was computed in 1D using a second order TVD finite volume scheme and 10,000 elements.

7.8. Three-dimensional explosion problem

Whereas the previous three dimensional computations were still purely one-dimensional test problems, although computed in a fully three-dimensional setting, in this section we solve a genuinely three dimensional test case, for which, however, a spherically symmetric reference solution can still be computed. The governing



equations are the Euler equations in three space dimensions and the computational domain is a sphere of radius $R = 1$, with the following initial condition:

$$(\rho, u, v, w, p)(\vec{x}, 0) = \begin{cases} (1.0, 0, 0, 0, 1.0) & \text{if } \|\vec{x}\| < 0.4, \\ (0.125, 0, 0, 0, 0.1) & \text{if } \|\vec{x}\| \geqslant 0.4. \end{cases} \tag{63}$$

This test problem can be interpreted as a three-dimensional version of the Sod shock tube problem and is discussed further in [48]. Exploiting the rotational symmetry of this test case, one can reduce the problem to a one-dimensional system of PDEs with source term, that can be solved on a very fine mesh in order to produce a reliable reference solution [48]. We solve the problem on an irregular tetrahedral mesh with smallest edge length 2×10^{-2}, which corresponds to an equivalent spatial resolution of 100^3 elements on a structured Cartesian grid. Our tetrahedral mesh used in this computation contains about 2.3 million tetrahedrons and is depicted in Fig. 17. We solve the problem using second, third and fourth order quadrature-free ADER-FV schemes with characteristic WENO reconstruction. The HLLE flux is used as leading flux together with the HLL-type corrector. The computations were performed on the HLRB2 supercomputer of the LRZ in München, Germany, using 128 Intel Itanium2 Madison 9 M processors, each with 4 GB of RAM and 1.6 GHz clock speed. The wallclock times used for the second, third and fourth order simulation were 33 min, 54 min and 100 min, respectively. The results on the positive x-axis are depicted in Fig. 18 showing the reconstruction polynomial for the density ρ on 100 equidistantly distributed sample points. The reference solution was computed solving the equivalent one-dimensional PDE with source term using a second order TVD scheme and 10,000 elements. One can clearly see the essentially non-oscillatory behaviour of the proposed quadrature-free ADER-FV schemes with characteristic WENO reconstruction as well as the increase in resolution for the higher order methods even in the direct vicinity of the discontinuities. We emphasize that in particular the beginning and the end of the rarefaction fan are difficult to capture and that especially here, the higher order schemes perform better.

7.9. Low Mach number flow past a sphere

In the previous three-dimensional examples, we compared our unstructured quadrature-free finite volume schemes with exact solutions available for one-dimensional shock tube problems exhibiting discontinuous

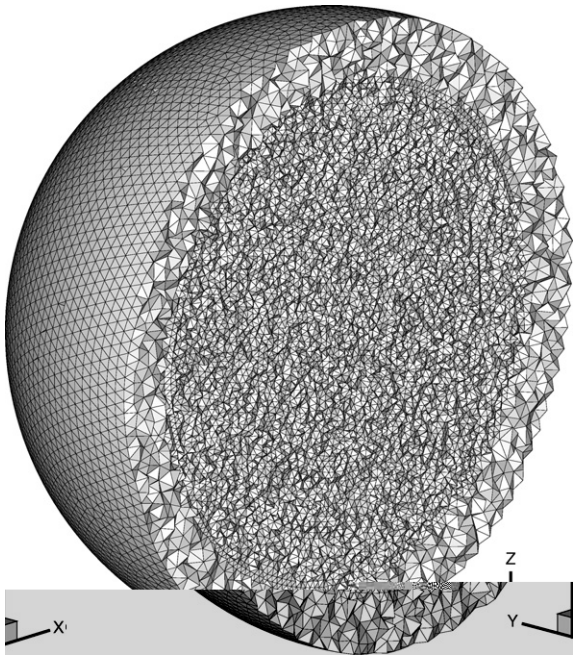


Fig. 17. Spherical tetrahedral mesh used for the three-dimensional explosion test problem.

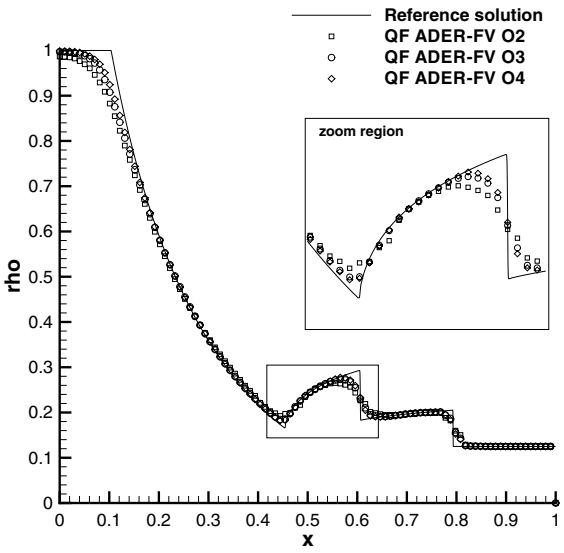


Fig. 18. Results on the positive x-axis for the 3D explosion test problem.

solutions. However, many problems arising in computational fluid dynamics (CFD) are not always and necessarily dominated by shock waves or other discontinuities, but by rather smooth flow features appearing at moderate to very low Mach numbers, such as airflow around cars, gliders and dirigibles or air flows through ducts and cooling fans.

This section is therefore dedicated to an application of the proposed method to the low Mach number flow around a sphere. The sphere has a radius of $R = 1$ and the computational domain is $\Omega = [-10;10]^3$. The incoming flow, which we also use as an initial condition for this test problem, is parallel to the positive x-axis and has the following properties: $M_\infty = 10^{-2}$, $\gamma = 1.4$, $\rho_\infty = 1$, and $p_\infty = 1/\gamma$, $c_\infty = 1$, $u_\infty = M_\infty c_\infty = 10^{-2}$. The exact solution of the potential flow around the sphere, which is a very good theoretical approximation for this problem, is given in the xy-plane for the angular and radial velocities as follows:

$$v_\phi(r,\phi) = -u_\infty \sin(\phi)\left(1 + \frac{1}{2}\left(\frac{R}{r}\right)^3\right), \quad v_r(r,\phi) = u_\infty \cos(\phi)\left(1 - \left(\frac{R}{r}\right)^3\right). \tag{64}$$

We emphasize that the reference solution is *steady* and has *two* stagnation points on the x-axis at $\vec{x}_1 = (-1,0,0)$ and at $\vec{x}_2 = (1,0,0)$. It is furthermore symmetric with respect to the x-axis. Bassi and Rebay [5] and Dumbser and Munz [20] have shown in a two-dimensional setting that high order unstructured discontinuous Galerkin schemes that do not use a curved boundary approximation will fail on this test problem since the numerical solution will become unsteady, non-symmetric and will not exhibit the two stagnation points.

We now solve this problem using a fourth order quadrature-free ADER-FV scheme with characteristic WENO reconstruction in order to show that the method also behaves very well in this difficult low Mach number test case. We emphasize that no curved boundary representation is used here. The sphere is simply approximated by the triangular surfaces of the adjacent tetrahedrons. The unstructured mesh used for this computation contains 20,1722 tetrahedrons. The problem was solved again on 128 CPUs of the HLRB2 supercomputer in München, consuming 7.25 h wallclock time for 22,470 time steps. The velocity iso-surfaces and some streamlines computed from our numerical solution are shown in Fig. 19. The depicted solution is perfectly converged to steady state and we clearly can see the two stagnation points and the expected symmetry properties. In order to quantify these results, the reconstructed solution w_p is evaluated on circles with different radii ($r = 1.0$, $r = 1.5$ and $r = 2.0$) on 90 equidistantly spaced sample points in the xy-plane and in the yz-plane, see the left and the right sub-figures in Fig. 20, respectively. The agreement between the numerical solution and the reference solution is almost perfect. In the left sub-figure, we show the reconstructed angular velocity in the xy-plane and in the right sub-figure we show the same quantity in the yz-plane ($\phi = 90°$). Due to symmetry, the quantities should not depend on the second azimuthal angle θ and thus should remain constant.

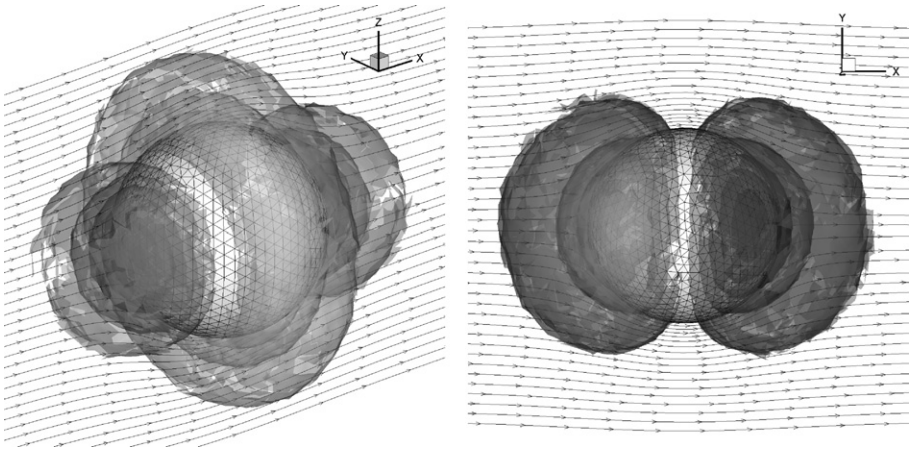
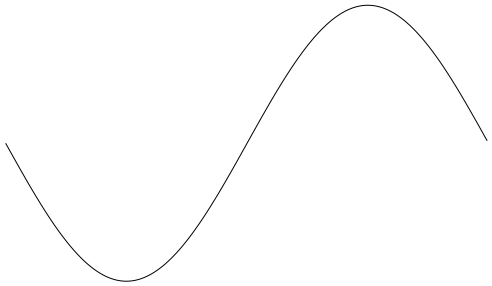


Fig. 19. Velocity iso-surfaces and streamlines around the sphere. Left: u – iso-surfaces. Right: w – iso-surfaces.



7.10. Interaction of a shock wave with a half cone in three dimensions

The setup of this test case is similar to the two dimensional shock–wedge interaction problem studied in Section 7.4.

In the x-direction the computational domain Ω covers the interval $[-1.5;3]$, and in the $y-z$ plane it consists of a half circle of radius $R = 2.25$ with its center on the x-axis. A circular half cone with length $L = 1$, tip radius $R_1 = 0.02$ and foot radius $R_2 = 0.5$ is placed with its tip on the origin. Solid wall boundary conditions are imposed everywhere on the cone and on the mantle surface of the half cylinder that constitutes the boundary of the computational domain. On the top ($x = -1.5$) and on the bottom ($x = 3$) of the half cylinder, the exact solution according to the Rankine–Hugoniot conditions of a shock wave with shock Mach number $M_s = 1.3$ is imposed. The initial condition that produces this shock wave is identical to the one given in Eq. (59) in Section 7.4, setting the third velocity component $w = 0$ everywhere. The computational mesh used for this computation, which is by far the largest presented in this article, consists of 9.3 million tetrahedrons. A view of the mesh is depicted in Fig. 21, where we do not show tetrahedrons with $y < 0$ and $x < -0.5$ to be able to visualize also the mesh in the interior of the computational domain. We emphasize that the unstructured mesh generation for this large-scale application took only about 20 min with very few manual user interactions. The simulation was performed using a third order ADER-FV scheme on 1024 CPUs of the HLRB2 supercomputer at the *Leibniz Rechenzentrum* (LRZ) in München, Germany. The computation until the final

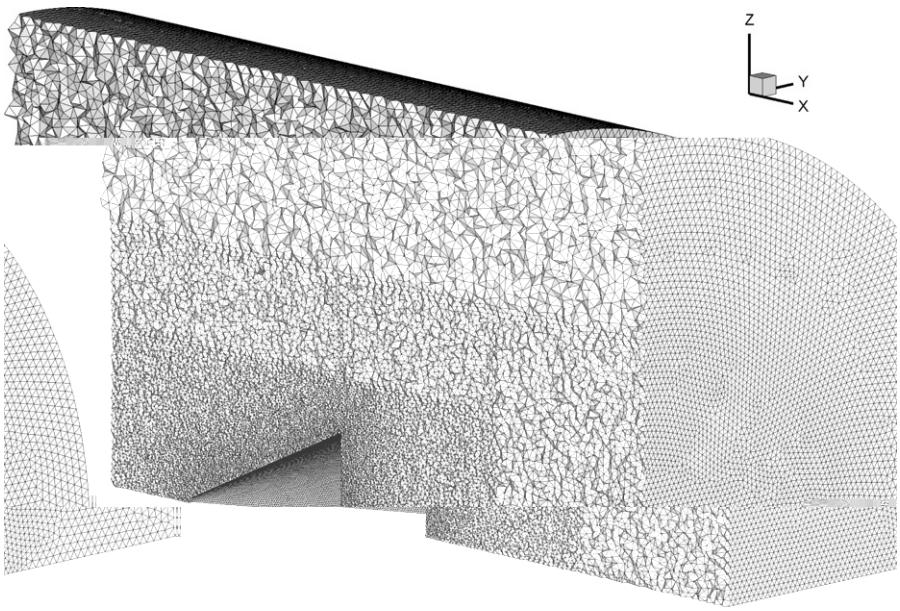


Fig. 21. View of the unstructured tetrahedral mesh with 9.3 million elements used for the computation of the shock-half cone interaction problem.

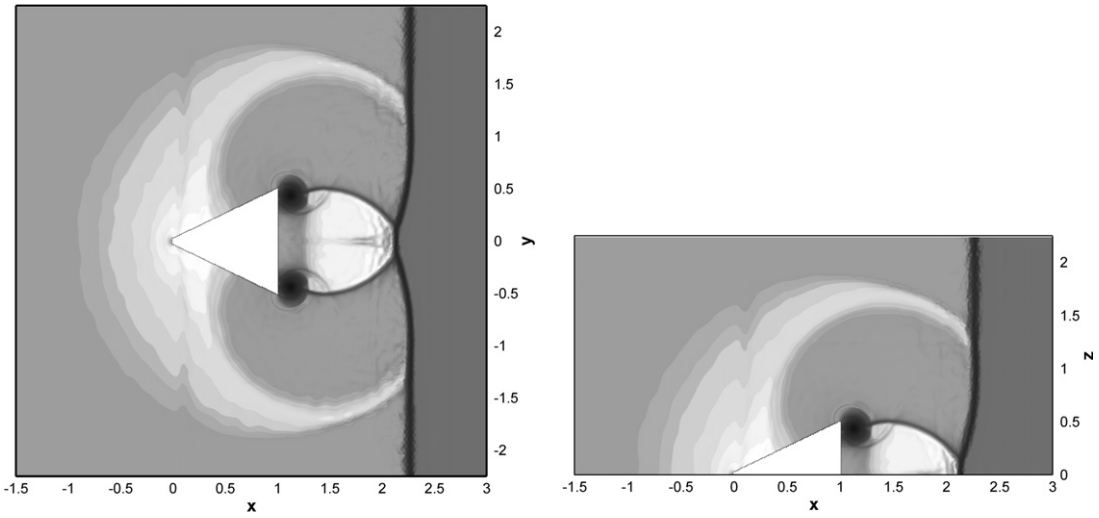


Fig. 22. Numerical Schlieren images at $t = 2.5$ for the shock-half cone interaction problem in 3D using the third order quadrature-free ADER-FV scheme on an unstructured tetrahedral mesh. Left: Cut in x–y-plane. Right: Cut in x–z-plane.

output time $t = 2.5$ took 15 h wallclock time for 7134 time steps. To facilitate the comparison with the two-dimensional results, we show two cuts in the x–y- and in the x–z-plane, respectively, in Fig. 22. The results agree qualitatively with the two-dimensional results. We note, however, that the reflected wave in three space dimensions is much weaker than in two space dimensions. Whereas in 2D a sharp interface was present, in 3D we notice an almost smooth transition. Two three-dimensional views of the cone together with density iso-surfaces and streamlines can finally be seen in Figs. 23 and 24, where the torus-shaped vortex shedding behind the half cone can be seen.

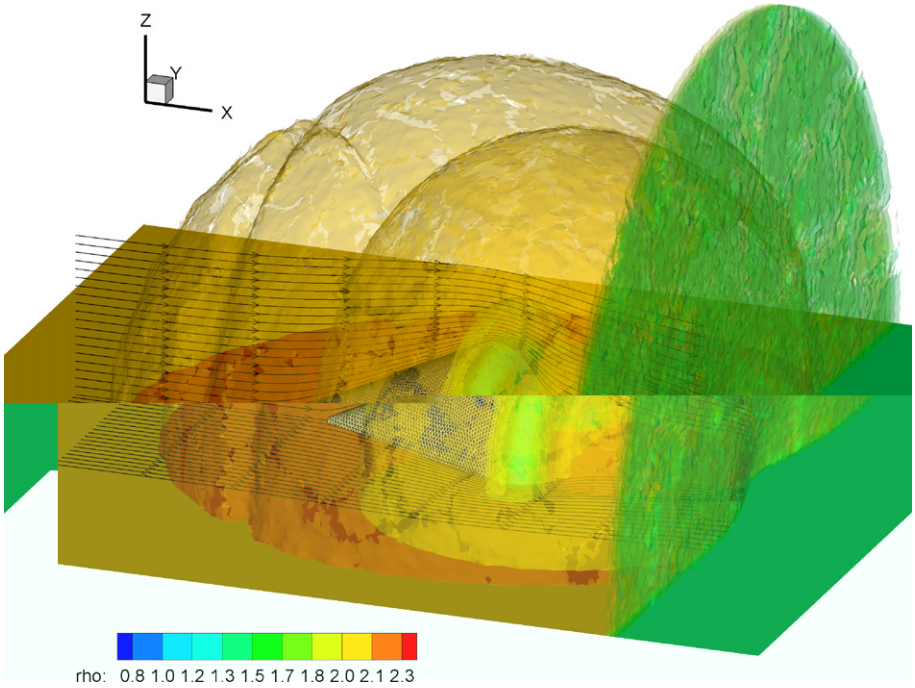


Fig. 23. Density contour surfaces and streamlines at $t = 2.5$ for the shock-half cone interaction problem seen from behind the cone. On can clearly see the refracted primary shock wave, the reflected waves and the vortex half ring shedded from the half cone.

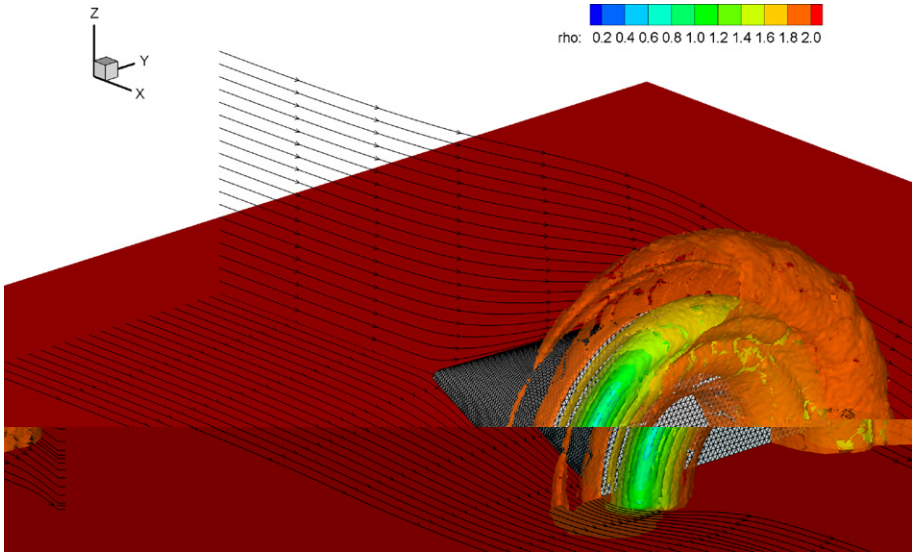


Fig. 24. Zoom into the streamlines and the density contour surfaces of the vortex half ring at $t = 2.5$ for the shock-half cone interaction problem seen from behind the cone.

8. Summary and conclusions

To our knowledge, the scheme proposed in this article is the first high order non-oscillatory quadrature-free finite volume scheme for nonlinear hyperbolic systems on unstructured triangular and tetrahedral meshes in two and three space dimensions. Also the technique for applying the proposed type of characteristic WENO reconstruction yielding the whole polynomial information in each element on unstructured meshes is new. The Cauchy–Kovalewski procedure as a main building block applied onto the reconstructed non-oscillatory piece-

wise polynomials directly provides a space–time Taylor series for the conserved quantities and the physical fluxes. This information is sufficient to construct a highly accurate space–time dependent numerical upwind flux that can be subsequently integrated analytically in space and time and thus allows the construction of a high order accurate quadrature-free one-step finite volume scheme. This was achieved by a splitting of the flux into a leading term evaluated at the space–time barycenters of the element interfaces and a space–time dependent corrector flux, which is a linear function of *four* arguments instead of *two*, as is usually the case for numerical flux functions used in the context of finite volume schemes. Whereas the particular feature of the possibility to construct a one-step time stepping scheme was already previously used in the research literature, as for example in the original work of Harten et al. on ENO schemes [27] and furthermore in the family of ADER-FV and ADER-DG schemes [17,21,33,47,49,51], the use of the Cauchy–Kovalewski procedure to avoid numerical quadrature in space and time is new. For this purpose the system of governing PDEs is rewritten in the coordinate system of the reference element which allows to compute the space–time integrals appearing in our method in a mesh-independent way, once and for all, analytically. A similar approach of obtaining a quadrature-free formulation via the Cauchy–Kovalewski procedure may also be feasible in the framework of discontinuous Galerkin finite element schemes. This will be explored in future work.

This feature of the proposed method is of particular importance on unstructured tetrahedral meshes in three space dimensions, where numerical quadrature could become very expensive for high order finite volume schemes. We emphasize that the original WENO reconstruction that reconstructs point-values in all Gaussian quadrature points could not be used to construct a quadrature-free finite volume method since for this purpose the whole polynomial information is necessary inside the elements. We emphasize that with the quadrature-free approach only one leading flux and one corrector flux has to be computed for each element face. This leads to a total of *eight* flux evaluations for tetrahedral elements in 3D per element per time step, independent of the order of the method. In comparison, using a standard fifth order WENO scheme together with a third order TVD Runge–Kutta time stepping method, one would need at least seven spatial Gaussian integration points per face [44] and three Runge–Kutta stages in time, leading to 21 flux evaluations per element face and time step. On three-dimensional tetrahedral meshes, this method would require 84 numerical flux evaluations per element per time-step and 84 nonlinear WENO reconstructions in characteristic variables, since the standard WENO scheme reconstructs point values [30] and not entire polynomials. In our approach, we only need *one* nonlinear WENO reconstruction per element and time step, but this reconstruction is more expensive than the point-wise WENO reconstruction since we have to reconstruct all polynomial coefficients and we must consider all characteristic directions simultaneously. We remark furthermore that standard pointwise WENO reconstruction schemes have the advantage of obtaining a higher order of accuracy than our method considering the union of all stencils. Although we think to have reduced the computational effort for the nonlinear WENO reconstruction as much as possible, it is still the most expensive part of the whole algorithm, consuming about 75% of the CPU time. Compared with the computational effort needed for reconstruction, the CPU time requirements for the Cauchy–Kovalewski procedure of about 15% of the total CPU time are relatively low. The remaining time (about 10%) is spent in the evaluation of the quadrature-free formulation of the space–time integrals of the numerical flux.

We have shown the versatility of the proposed method on a large number of test cases computed on unstructured triangular and tetrahedral meshes in two and three space dimensions, respectively. The high order quadrature-free ADER finite volume scheme can be applied at the same time to low Mach number flow and steady state problems as well as to very high Mach number flow. Since unstructured mesh generation can be done almost automatically, the presented method could possibly close the gap in modern computational fluid dynamics (CFD) and computational aeroacoustics (CAA) between the well-developed highly accurate finite-difference solvers needing structured grids on the one hand and the usually only second order unstructured finite volume solvers on the other hand. A possible future extension of this method is the incorporation of diffusion terms for the solution of the compressible Navier-Stokes equations. The final aim of our research is to provide highly accurate direct numerical simulations (DNS) and large eddy simulations (LES) on unstructured meshes in very complex geometries. To improve the computational efficiency of the method further research will be carried out to implement a time-accurate local time stepping technique based on the information generated by the Cauchy–Kovalewski procedure. This approach has already been successfully implemented recently for Cauchy–Kovalewski based discontinuous Galerkin methods [18,35]. Because of the

many points in common between ADER-DG and ADER-FV schemes, time-accurate local time stepping may be also included in the unstructured quadrature-free ADER-FV method.

Acknowledgments

The first two authors were financed by the *Deutsche Forschungsgemeinschaft* (DFG) in the framework of the *DFG Forschungsstipendium* (DU 1107/1-1) and the *Emmy Noether Programm* (KA 2281/1-1). The authors are especially grateful to the Leibniz Rechenzentrum in München, Germany, for helpful advice and for the use of the supercomputing facilities. Last but not least the authors would like to thank the two anonymous referees for their helpful questions and remarks that helped to improve the quality of this article.

Appendix A. A Fortran example code for the Cauchy–Kovalewski procedure for the 3D Euler equations written in reference coordinates

```
SUBROUTINE NonLinearCK_Euler3D(W,FCx,FCy,FCz,M,BC,gamma,Jacobi,dt)
  IMPLICIT NONE
  ! Argument list declaration
  INTEGER :: M                                  ! M = max(deg(Psi_l))
  REAL    :: FCx(1:5,0:M,0:M,0:M,0:M)           ! f
  REAL    :: FCy(1:5,0:M,0:M,0:M,0:M)           ! g
  REAL    :: FCz(1:5,0:M,0:M,0:M,0:M)           ! h
  REAL    :: W(1:18,0:M,0:M,0:M,0:M)            ! U_p & aux. variables
  REAL    :: BC(0:M,0:M)                        ! Binomial coeff.
  REAL    :: gamma                              ! gamma = c_p/c_v
  REAL    :: Jacobi(3,3)                        ! dXi_i/dx_j
  REAL    :: dt                                 ! time step Delta t
  ! Local variables
  REAL    :: V6,V7,V8,V9,PiBC,rhoEp,E_kin       ! Temp. variables
  REAL    :: FCxi(1:5,0:M,0:M,0:M,0:M)          ! f*
  REAL    :: FCet(1:5,0:M,0:M,0:M,0:M)          ! g*
  REAL    :: FCzt(1:5,0:M,0:M,0:M,0:M)          ! h*
  INTEGER :: a,b,c,d,i,j,k,l                    ! Loop counters
  ! Numbering of the conservative and auxiliary variables in the array W
  ! 1  : rho          2  : rho*u        3  : rho*v        4 : rho*u   5 : rho*E
  ! 6  : u            7  : v            8  : w            9 : p
  ! 10 : (rho*u)*u     11 : (rho*v)*v     12 : (rho*w)*w
  ! 13 : (rho*u)*v     14 : (rho*u)*w     15 : (rho*v)*w
  ! 16 : u*(rho*E+p)   17 : v*(rho*E+p)   18 : w*(rho*E+p)
  INTENT(INOUT)  ::  W
  INTENT(OUT)    ::  FCx,FCy,FCz
  INTENT(IN)     ::  M, BC, gamma, Jacobi
  DO d=0,M          ! Successively increase time derivative level
    ! Compute aux. variables 6 - 8 using the operator L_**
    DO c=0,M-d; DO b=0,M-d-c; DO a=0,M-d-c-b
      V6 = 0.; V7 = 0.; V8 = 0.; V9 = 0.
      W(6:9,a,b,c,d) = 0.     ! Set last component to zero
      DO l=0,d; DO k=0,c; DO j=0,b; DO i=0,a
        PiBC=BC(a,i)*BC(b,j)*BC(c,k)*BC(d,l)
        V6 = V6 + PiBC*W(1,a-i,b-j,c-k,d-l)*W(6,i,j,k,l)
        V7 = V7 + PiBC*W(1,a-i,b-j,c-k,d-l)*W(7,i,j,k,l)
        V8 = V8 + PiBC*W(1,a-i,b-j,c-k,d-l)*W(8,i,j,k,l)
      ENDDO; ENDDO; ENDDO; ENDDO
      ! Compute aux. variables 6-9
      W(6,a,b,c,d)=(W(2,a,b,c,d) - V6)/W(1,0,0,0,0)     ! (rho*u)/rho
      W(7,a,b,c,d)=(W(3,a,b,c,d) - V7)/W(1,0,0,0,0)     ! (rho*v)/rho
      W(8,a,b,c,d)=(W(4,a,b,c,d) - V8)/W(1,0,0,0,0)     ! (rho*w)/rho
    ENDDO; ENDDO; ENDDO
    ! Compute aux. variables 10-15 with the generalized Leibniz rule
    DO c=0,M-d; DO b=0,M-d-c; DO a=0,M-d-c-b
```

```
      W(10:18,a,b,c,d) = 0.
      DO l=0,d; DO k=0,c; DO j=0,b; DO i=0,a
        PiBC=BC(a,i)*BC(b,j)*BC(c,k)*BC(d,l)
        W(10,a,b,c,d) = W(10,a,b,c,d) + PiBC*W(2,a-i,b-j,c-k,d-l)*W(6,i,j,k,l)
        W(11,a,b,c,d) = W(11,a,b,c,d) + PiBC*W(3,a-i,b-j,c-k,d-l)*W(7,i,j,k,l)
        W(12,a,b,c,d) = W(12,a,b,c,d) + PiBC*W(4,a-i,b-j,c-k,d-l)*W(8,i,j,k,l)
        W(13,a,b,c,d) = W(13,a,b,c,d) + PiBC*W(2,a-i,b-j,c-k,d-l)*W(7,i,j,k,l)
        W(14,a,b,c,d) = W(14,a,b,c,d) + PiBC*W(2,a-i,b-j,c-k,d-l)*W(8,i,j,k,l)
        W(15,a,b,c,d) = W(15,a,b,c,d) + PiBC*W(3,a-i,b-j,c-k,d-l)*W(8,i,j,k,l)
      ENDDO; ENDDO; ENDDO; ENDDO
      ! Compute the pressure
      E_kin = 0.5*(W(10,a,b,c,d)+W(11,a,b,c,d)+W(12,a,b,c,d))
      W( 9,a,b,c,d) =(gamma-1)*( W(5,a,b,c,d) - E_kin )
    ENDDO; ENDDO; ENDDO
    ! Compute aux. variables 16-18 with the generalized Leibniz rule
    DO c=0,M-d; DO b=0,M-d-c; DO a=0,M-d-c-b
      W(16:18,a,b,c,d) = 0.
      DO l=0,d; DO k=0,c; DO j=0,b; DO i=0,a
        PiBC=BC(a,i)*BC(b,j)*BC(c,k)*BC(d,l)
        rhoEp = W(5,i,j,k,l) + W(9,i,j,k,l)
        W(16,a,b,c,d)=W(16,a,b,c,d)+PiBC*W(6,a-i,b-j,c-k,d-l)*rhoEp
        W(17,a,b,c,d)=W(17,a,b,c,d)+PiBC*W(7,a-i,b-j,c-k,d-l)*rhoEp
        W(18,a,b,c,d)=W(18,a,b,c,d)+PiBC*W(8,a-i,b-j,c-k,d-l)*rhoEp
      ENDDO; ENDDO; ENDDO; ENDDO
    ENDDO; ENDDO; ENDDO
    ! Compute convective fluxes f, g, h and transformed fluxes f*, g*, h*
    DO c=0,M-d; DO b=0,M-d-c; DO a=0,M-d-c-b
      FCx(1,a,b,c,d)  = W( 2,a,b,c,d); FCx(3,a,b,c,d) = W(13,a,b,c,d)
      FCx(4,a,b,c,d)  = W(14,a,b,c,d); FCx(5,a,b,c,d) = W(16,a,b,c,d)
      FCx(2,a,b,c,d)  = W(10,a,b,c,d)+W( 9,a,b,c,d)
      FCy(1,a,b,c,d)  = W( 3,a,b,c,d); FCy(2,a,b,c,d) = W(13,a,b,c,d)
      FCy(4,a,b,c,d)  = W(15,a,b,c,d); FCy(5,a,b,c,d) = W(17,a,b,c,d)
      FCy(3,a,b,c,d)  = W(11,a,b,c,d)+W( 9,a,b,c,d)
      FCz(1,a,b,c,d)  = W( 4,a,b,c,d); FCz(2,a,b,c,d) = W(14,a,b,c,d)
      FCz(3,a,b,c,d)  = W(15,a,b,c,d); FCz(5,a,b,c,d) = W(18,a,b,c,d)
      FCz(4,a,b,c,d)  = W(12,a,b,c,d)+W( 9,a,b,c,d)
      FCxi(:,a,b,c,d) = FCx(:,a,b,c,d)*Jacobi(1,1)+ &
                        FCy(:,a,b,c,d)*Jacobi(1,2)+FCz(:,a,b,c,d)*Jacobi(1,3)
      FCet(:,a,b,c,d) = FCx(:,a,b,c,d)*Jacobi(2,1)+ &
                        FCy(:,a,b,c,d)*Jacobi(2,2)+FCz(:,a,b,c,d)*Jacobi(2,3)
      FCzt(:,a,b,c,d) = FCx(:,a,b,c,d)*Jacobi(3,1)+ &
                        FCy(:,a,b,c,d)*Jacobi(3,2)+FCz(:,a,b,c,d)*Jacobi(3,3)
    ENDDO; ENDDO; ENDDO
    ! Inserting the fluxes (and source terms) into the governing equations
    DO c=0,M-1-d; DO b=0,M-1-d-c; DO a=0,M-1-d-c-b
      W(1:5,a,b,c,d+1) = - dt*FCxi(1:5,a+1,b,c,d) - dt*FCet(1:5,a,b+1,c,d) &
                         - dt*FCzt(1:5,a,b,c+1,d)     ! + dt*S(1:5,a,b,c,d)
    ENDDO; ENDDO; ENDDO
  END DO
END SUBROUTINE NonLinearCK_Euler3D
```

References

[1] R. Abgrall, On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation, Journal of Computational Physics 144 (1994) 45–58.

[2] H. Atkins, C.W. Shu, Quadrature-free implementation of the discontinuous Galerkin method for hyperbolic equations, AIAA Journal 36 (1998) 775–782.

[3] D. Balsara, C.W. Shu, Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy, Journal of Computational Physics 160 (2000) 405–452.

[4] T.J. Barth, P.O. Frederickson, Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction. AIAA paper no. 90-0013, 28th Aerospace Sciences Meeting January 1990.

[5] F. Bassi, S. Rebay, High-order accurate discontinuous finite element solution of the 2D Euler equations, Journal of Computational Physics 138 (1997) 251–285.
[6] M. Ben-Artzi, J. Falcovitz, A second-order Godunov-type scheme for compressible fluid dynamics, Journal of Computational Physics 55 (1984) 1–32.
[7] A. Bourgeade, P. LeFloch, P.A. Raviart, An asymptotic expansion for the solution of the generalized riemann problem. Part II: application to the gas dynamics equations, Annales de l'institut Henri Poincaré (C) Analyse non linéaire 6 (1989) 437–480.
[8] G. Chiocchia, Exact solutions to transonic and supersonic flows. AGARD Advisory Report AR-211, 1985.
[9] B. Cockburn, S. Hou, C.W. Shu, The Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case, Mathematics of Computation 54 (1990) 545–581.
[10] B. Cockburn, G.E. Karniadakis, C.W. Shu, Discontinuous Galerkin methods, Lecture Notes in Computational Science and Engineering, Springer, 2000.
[11] B. Cockburn, S.Y. Lin, C.W. Shu, TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one dimensional systems, Journal of Computational Physics 84 (1989) 90–113.
[12] B. Cockburn, C.W. Shu, TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework, Mathematics of Computation 52 (1989) 411–435.
[13] B. Cockburn, C.W. Shu, The Runge–Kutta local projection P1-Discontinuous Galerkin finite element method for scalar conservation laws, Mathematical Modelling and Numerical Analysis 25 (1991) 337–361.
[14] B. Cockburn, C.W. Shu, The Runge–Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems, Journal of Computational Physics 141 (1998) 199–224.
[15] M. Dubiner, Spectral methods on triangles and other domains, Journal of Scientific Computing 6 (1991) 345–390.
[16] M. Dumbser, M. Käser, J. de la Puente, Arbitrary high-order finite volume schemes for seismic wave propagation on unstructured meshes in two and three space dimensions, Geophysical Journal International, in press.
[17] M. Dumbser, M. Käser, Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems, Journal of Computational Physics 221 (2007) 693–723.
[18] M. Dumbser, M. Käser, E.F. Toro, An arbitrary high order discontinuous Galerkin method for elastic waves on unstructured meshes V: Local time stepping and *p*-adaptivity, Geophysical Journal International, in press.
[19] M. Dumbser, J.M. Moschetta, J. Gressier, A matrix stability analysis of the carbuncle phenomenon, Journal of Computational Physics 197 (2004) 647–670.
[20] M. Dumbser, C.D. Munz, On source terms and boundary conditions using arbitrary high order discontinuous Galerkin schemes, International Journal of Applied Mathematics and Scientific Computing, in press.
[21] M. Dumbser, C.D. Munz, Building blocks for arbitrary high order discontinuous Galerkin schemes, Journal of Scientific Computing 27 (2006) 215–230.
[22] R.W. Dyson, Technique for very high order nonlinear simulation and validation. Technical Report TM-2001-210985, NASA, 2001.
[23] B. Einfeldt, C.D. Munz, P.L. Roe, B. Sjögreen, On Godunov-type methods near low densities, Journal of Computational Physics 92 (1991) 273–295.
[24] P. Le Floch, P.A. Raviart, An asymptotic expansion for the solution of the generalized riemann problem. Part I: General theory, Annales de l'institut Henri Poincaré (C) Analyse non linéaire 5 (1988) 179–207.
[25] O. Friedrich, Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids, Journal of Computational Physics 144 (1998) 194–212.
[26] S.K. Godunov, Finite difference methods for the computation of discontinuous solutions of the equations of fluid dynamics, Mat. Sb. 47 (1959) 271–306.
[27] A. Harten, B. Engquist, S. Osher, S. Chakravarthy, Uniformly high order essentially non-oscillatory schemes, III, Journal of Computational Physics 71 (1987) 231–303.
[28] A. Harten, B. Engquist, S. Osher, S.R. Chakravarthy, Uniformly high order accurate essentially non-oscillatory schemes III, Journal of Computational Physics 71 (1987) 231–303.
[29] A. Harten, P.D. Lax, B. van Leer, On upstream differencing and Godunov-type schemes for hyperbolic conservation laws, SIAM Review 25 (1) (1983) 35–61.
[30] C. Hu, C.W. Shu, Weighted essentially non-oscillatory schemes on triangular meshes, Journal of Computational Physics 150 (1999) 97–127.
[31] G.-S. Jiang, C.W. Shu, Efficient implementation of weighted ENO schemes, Journal of Computational Physics (1996) 202–228.
[32] G. Karypis, V. Kumar, Multilevel k-way partitioning scheme for irregular graphs, J. Parallel Distrib. Comput. 48 (1998) 96–129.
[33] M. Käser, A. Iske, ADER schemes on adaptive triangular meshes for scalar conservation laws, Journal of Computational Physics 205 (2005) 486–508.
[34] P.D. Lax, Weak solutions of nonlinear hyperbolic equations and their numerical approximation, Comm. Pure Appl. Math. 7 (1954) 159–193.
[35] F. Loercher, G. Gassner, C.D. Munz, A discontinuous Galerkin scheme based on a space–time expansion. i. inviscid compressible flow in one space dimension, Journal of Scientific Computing, in press.
[36] J.M. Moschetta, J. Gressier, A cure for the sonic point glitch, International Journal of Computational Fluids Dynamics 13 (2000) 143–159.
[37] C. Ollivier-Gooch, M. Van Altena, A high-order-accurate unstructured mesh finite-volume scheme for the advection-diffusion equation, Journal of Computational Physics 181 (2002) 729–752.

[38] A. Rault, G. Chiavassa, R. Donat, Shock–vortex interactions at high mach numbers, Journal of Scientific Computing 19 (2003) 347–371.
[39] P.L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, Journal of Computational Physics 43 (1981) 357–372.
[40] H. Schardin, in: Proc. VII Int. Cong. High Speed Photg., Darmstadt, O. Helwich Verlag, 1965, pp. 113–119.
[41] J. Shi, C. Hu, C.W. Shu, A technique of treating negative weights in WENO schemes, Journal of Computational Physics 175 (2002) 108–127.
[42] C.W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock capturing schemes II, Journal of Computational Physics 83 (1989) 32–78.
[43] T. Sonar, On the construction of essentially non-oscillatory finite volume approximations to hyperbolic conservation laws on general triangulations: polynomial recovery, accuracy and stencil selection, Computer methods in applied mechanics and engineering 140 (1997) 157–181.
[44] A.H. Stroud, Approximate Calculation of Multiple Integrals, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1971.
[45] A. Suresh, H.T. Huynh, Accurate monotonicity-preserving schemes with Runge–Kutta time stepping, Journal of Computational Physics 136 (1997) 83–99.
[46] A. Taube, M. Dumbser, D. Balsara, C.-D. Munz, Arbitrary high order discontinuous Galerkin schemes for the magnetohydrodynamic equations, Journal of Scientific Computing 30 (2007) 441–464.
[47] V.A. Titarev, E.F. Toro, ADER schemes for three-dimensional nonlinear hyperbolic systems, Journal of Computational Physics 204 (2005) 715–736.
[48] E.F. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics, second ed., Springer, 1999.
[49] E.F. Toro, R.C. Millington, L.A.M. Nejad, Towards very high order Godunov schemes, in: E.F. Toro (Ed.), Godunov Methods, Theory and Applications, Kluwer/Plenum Academic Publishers, 2001, pp. 905–938.
[50] E.F. Toro, M. Spruce, W. Speares, Restoration of the contact surface in the Harten-Lax-van Leer Riemann solver, Journal of Shock Waves 4 (1994) 25–34.
[51] E.F. Toro, V.A. Titarev, Solution of the generalized Riemann problem for advection-reaction equations, Proc. Roy. Soc. London (2002) 271–281.
[52] J.J.W. van der Vegt, H. van der Ven, Space-time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows I. general formulation, Journal of Computational Physics 182 (2002) 546–585.
[53] H. van der Ven, J.J.W. van der Vegt, Space-time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows II. efficient flux quadrature, Comput. Methods Appl. Mech. Engrg. 191 (2002) 4747–4780.
[54] M. van Dyke, An album of fluid motion, The Parabolic Press, 2005.
[55] B. van Leer, Towards the ultimate conservative difference scheme II: Monotonicity and conservation combined in a second order scheme, Journal of Computational Physics 14 (1974) 361–370.
[56] B. van Leer, Towards the ultimate conservative difference scheme V: A second order sequel to Godunov's method, Journal of Computational Physics 32 (1979) 101–136.
[57] Z.J. Wang, Y. Liu, Extension of the spectral volume method to high-order boundary representation, Journal of Computational Physics 211 (2006) 154–178.
[58] P. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, Journal of Computational Physics 54 (1984) 115–173.