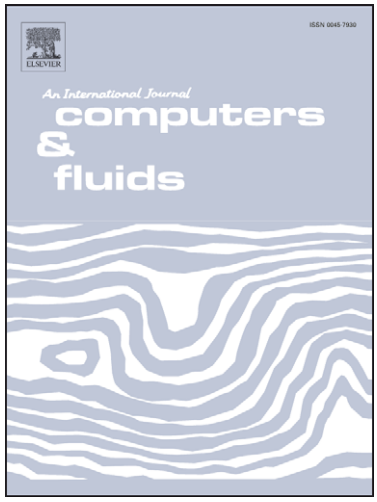
Computers & Fluids, Volume 46, Issue 1, July 2011, Pages 467–471
10th ICFD Conference Series on Numerical Methods for Fluid Dynamics (ICFD 2010)

Accepted Manuscript



Uniformly high-order schemes on arbitrary unstructured meshes for advection-diffusion equations

V.A. Titarev, D. Drikakis

PII: S0045-7930(10)00276-8
DOI: 10.1016/j.compfluid.2010.10.008
Reference: CAF 1419

To appear in: *Computers & Fluids*

Received Date: 14 May 2010
Revised Date: 14 October 2010
Accepted Date: 14 October 2010

Please cite this article as: Titarev, V.A., Drikakis, D., Uniformly high-order schemes on arbitrary unstructured meshes for advection-diffusion equations, *Computers & Fluids* (2010), doi: 10.1016/j.compfluid.2010.10.008

Uniformly high-order schemes on arbitrary unstructured meshes for advection-diffusion equations

V.A. Titarev*, D. Drikakis*

Department of Fluid Mechanics and Computational Science, Cranfield University, Cranfield, UK, MK43 0AL

Abstract

The paper presents a linear high-order method for advection-diffusion conservation laws on three dimensional mixed-element unstructured meshes. The key ingredient of the method is a reconstruction procedure in local computational coordinates. Numerical results illustrate the convergence rates for the linear equation and a non-linear hyperbolic system with diffusion terms for various types of meshes.

Keywords: unstructured, tetrahedral, hexahedral, prismatic, pyramidal, mixed-element, very high-order, diffusive

1. Introduction

Recent years have seen a rapid development of very high-order Godunov-type methods for tetrahedral meshes [2, 10, 1]. Such methods find their application primarily in numerical modelling of long-time evolution problems with complicated flow structures, e.g., aeroacoustics, seismology, tur-

*Corresponding author

Email addresses: v.a.titarev@cranfield.ac.uk, titarev@mail.ru (V.A. Titarev), d.drikakis@cranfield.ac.uk (D. Drikakis)

bulence modelling. However, in practical applications of computational fluid dynamics the present approaches based on purely tetrahedral meshes result in excessive computational costs and poor resolution of thin boundary layers. The above motivates the development of the methods which are applicable to advection-diffusion equations and can utilize arbitrary unstructured meshes, consisting of elements of various types.

Recently, a family of new very high-order weighted essentially non-oscillatory (WENO) methods for the two- and three-dimensional mixed-element unstructured meshes has been put forward in [6, 5]. These methods are direct extensions of [2, 1] and are based on the reconstruction procedure applicable to various polyhedral elements, such as hexahedral, prismatic and pyramidal cells. The computational examples provided for the compressible Euler equations demonstrate that the new methods maintain very high order of spatial accuracy across interfaces between cells of different types and in the same time produce essentially non-oscillatory profiles across discontinuous solutions and regions of high gradients of the solution.

The motivation of the present work is to extend the schemes from [6, 5] to deal with advection-diffusion problems with source terms. The approach is based on the use of the polynomial reconstruction in the local computational coordinates for both convective and diffusive terms of the equations and results in the methods of theoretically arbitrary order of spatial accuracy. The temporal evolution is carried out by means of a Runge-Kutta method, although other approaches are also possible. Since the goal of the present paper is to develop the basic methodology, the presentation is limited to the linear (single-stencil) schemes only. The essentially non-oscillatory version

of the resulting method can be constructed in a fairly straightforward way using ideas from [6, 5]. The calculations are carried out in Cartesian geometries using a variety of mixed-element meshes. The numerical results are provided for scalar advection-diffusion equation with a source term and for the compressible Euler equations with added heat conduction effects.

The paper is organized as follows. In Section 2, a detailed explanation of polynomial reconstruction procedure for a scalar variable is provided. Section 3 is devoted to the application of this procedure to the construction of the numerical method. Section 4 presents numerical results and conclusions are drawn in Section 5.

2. Polynomial reconstruction on mixed-element meshes

In this section a reconstruction procedure for mixed-element unstructured meshes in three-space dimensions is described. Without loss of generality the idea can be explained as applied to a scalar variable $u(x, y, z)$. Suppose that the spatial computational domain is discretized by conforming elements V_i of the volume $|V_i|$, indexed by a unique mono-index i. The center of the element has coordinates $\mathbf{x}_i = (x_i, y_i, z_i)$. The elements considered are of hexahedral, tetrahedral, pyramidal and prismatic shapes. The main goal of the reconstruction procedure is to build a high-order polynomial $p_i(x, y, z)$ such that it has the same cell average as u on the target cell V_i and in the same time approximates the point-wise values of u inside the cell with the given order of accuracy r:

$$u(x, y, z) = p_i(x, y, z) + \text{const} \cdot h^r, \quad h \approx |V_i|^{1/3} \tag{1}$$

To build up this polynomial, the reconstruction procedure will use the cell averages of $u(x, y, z)$ on the target cell V_i as well as averages $\bar{u}_m$ from the reconstruction stencil formed by neighboring cells V_m.

In order to simplify the notation, in this section the global spatial index i is omitted and the local numbering of cells is introduced. The reconstruction problem can thus be reformulated as follows: for a target cell V_0 build a high-order polynomial $p(x, y, z)$ so that its spatial average on cell V_0 is equal to u_0 and approximates point-wise values of u with rth order of accuracy.

In general, the reconstruction can be carried out in the physical coordinates $\mathbf{x} = (x, y, z)$, taking special measures against scaling effects. A more elegant and computationally accurate approach, however, is to use the so-called reference coordinate system (ξ, η, ζ), as is done in [2] for triangular (2D) and tetrahedral (3D) elements. Here this transformation technique is extended to deal with general mixed-element mesh elements.

The basic steps of the transformation procedure can be summarized as follows. Firstly, if the cell V_0 is not tetrahedral, it is decomposed into tetrahedrons. The number of these tetrahedral elements depends on the type of element V_0 and is equal to two for a pyramid, three for a prism and finally six for a hexahedral cell. Next, the linear transformation from the physical coordinate system (x, y, z) into the reference coordinate system (ξ, η, η) is defined using one of the tetrahedral elements resulting from the decomposition. Finally, the cell V_0 and other cells in the reconstruction stencil are transformed into the reference space.

Let $\mathbf{w}_1 = (x_1, y_1, z_1)$, $\mathbf{w}_2 = (x_2, y_2, z_2)$, $\mathbf{w}_3 = (x_3, y_3, z_3)$, $\mathbf{w}_4 = (x_4, y_4, z_4)$ be the four vertices of one of the tetrahedral elements the target element V_0

consists of. The transformation from the Cartesian coordinates (x, y, z) into a reference space (ξ, η, ζ) is defined as

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} + J_i \cdot \begin{pmatrix} \xi \\ \eta \\ \zeta \end{pmatrix}, \quad J_i = \begin{bmatrix} x_2 - x_1 & x_3 - x_1 & x_4 - x_1 \\ y_2 - y_1 & y_3 - y_1 & y_4 - y_1 \\ z_2 - z_1 & z_3 - z_1 & z_4 - z_1 \end{bmatrix} \quad (2)$$

with J_i is the Jacobian matrix of the transformation, dependent on the cell V_i. Eq. (2) defines both the direct and inverse mappings from $\boldsymbol{\xi} = (\xi, \eta, \zeta)$ into $\boldsymbol{x} = (x, y, z)$, which are denoted as

$$\mathbf{x} = \boldsymbol{x}(\boldsymbol{\xi}), \quad \boldsymbol{\xi} = \boldsymbol{\xi}(\mathbf{x}) \quad (3)$$

Via the inverse mapping the element V_0 can be transformed to the element V_0' in the reference coordinate system. Note that for the uniform (Cartesian) hexahedral mesh the transformed element V_0' is just a unit cube in the reference space (ξ, η, ζ), whereas for a general hexahedral element as well as for prismatic and pyramidal cells four of vertices of the transformed element will be from the unit cube.

For performing the reconstruction on the target element V_0, the so-called central reconstruction stencil $\mathcal{S}$ is formed which consists of $M + 1$ elements, including the target element V_0:

$$\mathcal{S} = \bigcup_{m=0}^{M} V_m$$

where the local index m counts the elements in the stencil $\mathcal{S}$. The stencil is build up recursively by adding the direct side neighbors of the element V_0 and all of the elements already existing in the stencil, until the desired number of elements is reached. The neighbors are added irrespective of their

shape and efforts are made to have the stencil which is as symmetric as possible. The inverse mapping (3) is then applied to all the elements V_m from the reconstruction stencil $\mathcal{S}$ and the transformed elements and stencil are denoted as E'_m and $\mathcal{S}'$, respectively:

$$\mathcal{S}' = \bigcup_{m=0}^{M} V'_m$$

Fig. 1 shows an example of the stencil the physical coordinate system for the third order reconstruction and prismatic mesh. The stencil consists of 20 cells, which are symmetrically grouped around the target cell V_i.

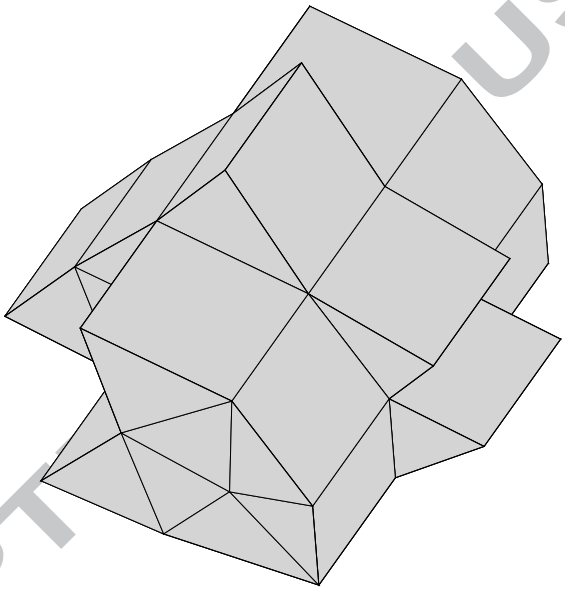


Figure 1: An example of a central stencil of the third-order spatial reconstruction.

The reconstruction polynomial at the transformed cell V'_0 is sought as an expansion over local polynomial basis functions $\phi_k(\xi, \eta, \zeta)$:

$$p(\xi, \eta, \zeta) = \sum_{k=0}^{K} a_k \phi_k(\xi, \eta, \zeta) = \bar{u}_0 + \sum_{k=1}^{K} a_k \phi_k(\xi, \eta, \zeta) \tag{4}$$

where a_k are degrees of freedom and the upper index in the summation of expansion K is related to the degree of the polynomial r by the expression

$$K = \frac{1}{2}(r+1)(r+2)(r+3) - 1.$$

The basis functions ϕ_k are constructed in such a way that condition (1) is satisfied identically irrespective of values of degrees of freedom:

$$\begin{gathered}\phi_k(\xi,\eta,\zeta) \equiv \psi_k(\xi,\eta,\zeta) - \frac{1}{|V_0'|}\int\limits_{V_0'} \psi_k \, d\xi d\eta d\zeta, \quad k = 1, 2, \ldots \\ \{\psi_k\} = \xi, \quad \eta, \quad \zeta, \quad \xi^2, \quad \eta^2, \quad \zeta^2, \quad \xi\cdot\eta, \ldots\end{gathered} \tag{5}$$

The unknown degrees of freedom a_k are found by requiring that for each cell V_m', $m = 1, \ldots M$, from the stencil $\mathcal{S}'$ the cell average of the reconstruction polynomial $p(\xi,\eta,\zeta)$ be equal to the cell average of the solution $\bar{u}_m$:

$$\int\limits_{V_m'} p(\xi,\eta,\zeta)\, d\xi d\eta d\zeta = \bar{u}_0|V_m'| + \sum_{k=1}^{K}\int\limits_{V_m'} a_k\phi_k \, d\xi d\eta d\zeta = u_m|V_m'|. \tag{6}$$

In general, at least K cells in the stencil are needed, different from the target cell V_0. However, the use of the minimum possible number of cells in the stencil $M \equiv K$ results in a scheme which may become unstable on general meshes. It is therefore recommended to use more cells in the stencil then the minimal required number [2]. Although on purely tetrahedral meshes it is usually sufficient to use 50% more cells, on mixed-element meshes it may be safer to increase the stencil further. We typically select $M = (2\ldots3)\cdot K$ depending on the cell type and order of spatial accuracy. The resulting overdetermined system (6) is solved by means of the least-square procedure.

$$f_{nj} \approx \sum_{\beta} \hat{f}_\beta \omega_\beta |A_j|,$$

where

$$\hat{f}_\beta = \frac{1}{2} V_n \left((1 + \text{sign}(V_n)) u_\beta^- + (1 - \text{sign}(V_n)) u_\beta^+ \right).$$

The calculation of the diffusive intercell fluxes f_{nj}^v requires the knowledge of the spatial derivatives of the solution at Gaussian integration points of the the cell face. For a given cell V_i they are obtained from the derivatives of the reconstruction polynomials with respect to local computational coordinates as follows:

$$\begin{pmatrix} \dfrac{\partial u}{\partial x} \\ \dfrac{\partial u}{\partial y} \\ \dfrac{\partial u}{\partial z} \end{pmatrix}_i^n = J_i^T \begin{pmatrix} \dfrac{\partial p(\xi,\eta,\zeta)}{\partial \xi} \\ \dfrac{\partial p(\xi,\eta,\zeta)}{\partial \eta} \\ \dfrac{\partial p(\xi,\eta,\zeta)}{\partial \zeta} \end{pmatrix}_i \tag{7}$$

where J_i^T is the transpose of the transformation matrix defined in (2). Denote by ∇u_β^-, ∇u_β^+ the boundary extrapolated values of the gradients of the solution. Then the diffusive flux for face j of cell V_i is given by

$$f_{nj}^v \approx \sum_\beta \hat{g}_\beta \omega_\beta |A_j|, \quad \hat{g}_\beta = \frac{1}{2} \lambda (\nabla u_\beta^- + \nabla u_\beta^+).$$

Finally, the numerical source term S_i is calculated by decomposing the cell V_i into tetrahedrons and then a using standard ten-point volume Gaussian quadrature:

$$S_i \approx \frac{1}{|V_i|} \sum_\gamma S(\boldsymbol{x}_\gamma, t, p_i(\boldsymbol{x}_\gamma, t)) \omega_\gamma,$$

where $\boldsymbol{x}_\gamma$ and ω_γ are nodes and weights of the resulting composite quadrature rule.

3. Numerical method

3.1. The framework

Consider the three-dimensional compressible Euler equations with heat conduction effects:

$$\frac{\partial}{\partial t}\mathbf{U} + \frac{\partial}{\partial x}(\mathbf{F} - \mathbf{F}^v) + \frac{\partial}{\partial y}(\mathbf{G} - \mathbf{G}^v) + \frac{\partial}{\partial z}(\mathbf{H} - \mathbf{H}^v) = \mathbf{S}, \tag{8}$$

where $\mathbf{U}$ is the vector of conserved variables, $\mathbf{F}$, $\mathbf{G}$, $\mathbf{H}$ and $\mathbf{F}^v$, $\mathbf{G}^v$, $\mathbf{H}^v$ are convective and heat conduction flux vectors in x, y and z coordinate directions respectively; $\mathbf{S}$ is the source term:

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

$$\mathbf{F}^v = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \lambda \dfrac{\partial T}{\partial x} \end{pmatrix}, \quad \mathbf{G}^v = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \lambda \dfrac{\partial T}{\partial y} \end{pmatrix}, \quad \mathbf{H}^v = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \lambda \dfrac{\partial T}{\partial z} \end{pmatrix}.$$

Here ρ is density, u,v,w velocity components in the x, y and z directions, respectively, p pressure, $E = p/(\gamma - 1) + (1/2)\rho(u^2 + v^2 + w^2)$ total energy, λ thermal conductivity coefficient, assumed to be constant here; $T = p/\rho$ gas temperature, γ the ratio of specific heats ($\gamma = 1.4$).

Spatial integration of (8) over a mesh element V_i yields the following semi-discrete finite-volume method:

$$\frac{d}{dt}\mathbf{U}_i + \frac{1}{|V_i|}\oint\limits_{\partial V_i} \mathbf{F}_n dA = \frac{1}{|V_i|}\oint\limits_{\partial V_i} \mathbf{F}_{vn} dA + \mathbf{S}_i,$$

$$\mathbf{F}_n = \mathbf{F}n_x + \mathbf{G}n_y + \mathbf{H}n_z, \quad \mathbf{F}_n^v = \mathbf{F}^v n_x + \mathbf{G}^v n_y + \mathbf{H}^v n_z, \tag{9}$$

$$\mathbf{S}_i = \frac{1}{|V_i|}\int\limits_{V_i} \mathbf{S}(x, y, z, t, \mathbf{U})\, dxdydz,$$

where $\mathbf{U}_i(t)$ are the cell averages of the solution at time t, $\mathbf{F}_n$, $\mathbf{F}_n^v$ - projection of the convective and heat conduction flux tensors on the face normal direction. The flux integrals over the element boundary ∂V_i are split into the sum of integrals over each face A_j resulting in the concise form of the semi-discrete method (9):

$$\frac{d}{dt}\mathbf{U}_i = \mathbf{R}_i, \tag{10}$$

where

$$\mathbf{R}_i = -\frac{1}{|V_i|}\sum_{j=1}^{L}(\mathbf{F}_{nj} - \mathbf{F}_{nj}^v) + \mathbf{S}_i, \quad \mathbf{F}_{nj} = \int\limits_{A_j} \mathbf{F}_n dA, \quad \mathbf{F}_{nj}^v = \int\limits_{A_j} \mathbf{F}_n^v dA.$$

The temporal derivative in (10) is approximated by the third-order TVD Runge-Kutta method [4]. The time step Δt is selected according to the formula

$$\min_i \left(\frac{w_i}{h_i} + \frac{1}{2}\frac{\lambda}{h_i^2}\right)\Delta t \le K \tag{11}$$

where w_i is an estimate of the maximum (in absolute value) propagation speed in cell V_i, $K \le 1/3$ is the CFL number, h_i is the characteristic length (diameter) of the element V_i.

The description of the scheme is complete once a procedure to calculate convective $\mathbf{F}_{nj}$, $\mathbf{F}_{nk}^v$, fluxes as well as the numerical source term $\mathbf{S}_i$ is specified.

3.2. Calculation of fluxes and numerical source term

The exact integral expressions for numerical fluxes are approximated by a suitable Gaussian numerical quadrature:

$$\mathbf{F}_{nj} \approx \sum_{\beta} \mathbf{F}_n \left(\mathbf{U}(\mathbf{x}_\beta, t)\right) \omega_\beta |A_j|, \quad \mathbf{F}^v_{nj} \approx \sum_{\beta} \mathbf{F}^v_n \left(\mathbf{U}(\mathbf{x}_\beta, t)\right) \omega_\beta |A_j| \tag{12}$$

Calculation of convective numerical fluxes in (12) requires the knowledge of point-wise values of the unknown vector of conserved variables $\mathbf{U}$ at the Gaussian points. These values are obtained by the reconstruction procedure on mixed-element meshes, explained in the previous section and applied here for each component of $\mathbf{U}$. The reconstruction produces the high-order vector polynomials $\mathbf{P}_i(\xi, \eta, \zeta)$ defined in the local reference coordinate system of each element V_i. Since these polynomials are different, at each Gaussian point β in the expression for the numerical flux (12) for the face A_j of cell V_i two approximate values for $\mathbf{U}$ exist. The first value $\mathbf{U}^-_\beta$ corresponds to the spatial limit to the cell boundary from inside the cell V_i and is given by the reconstruction polynomial $\mathbf{P}_i$. The second value $\mathbf{U}^+_\beta$ corresponds to the spatial limit from outside the element and is obtained by using the reconstruction polynomial of the neighboring element $V_{i'}$. For convective fluxes the resulting discontinuity is resolved by replacing the physical flux at each Gaussian integration point by a monotone function of left and right boundary extrapolated values so that the expression for $\mathbf{F}_{nj}$ takes the form:

$$\mathbf{F}_{nj} \approx \sum_{\beta} \hat{\mathbf{F}} \left(\mathbf{U}^-_\beta, \mathbf{U}^+_\beta\right) \omega_\beta |A_j| \tag{13}$$

The function $\hat{\mathbf{F}} \left(\mathbf{U}^-_\beta, \mathbf{U}^+_\beta\right)$ is called the Riemann solver, or a building block of a high-order scheme. Here, we use the HLLC Riemann solver [8].

In order to calculate the diffusive numerical fluxes, firstly for each spatial cell V_i a reconstruction polynomial for the temperature field is constructed using the same stencil as for the conserved vector. Next, the boundary extrapolated values of the temperature gradients of the solution ∇T_β^-, ∇T_β^+ are obtained from the reconstruction polynomial using the expressions (7). The diffusive flux for face j of cell V_i is given by

$$\mathbf{F}_{nj}^v \approx \sum_\beta \hat{\mathbf{F}}_\beta^v \omega_\beta |A_j|,$$

where

$$\hat{\mathbf{F}}_\beta^v = \lambda \begin{pmatrix} 0 \\ \dots \\ \frac{1}{2}(\nabla T_\beta^- + \nabla T_\beta^+) \end{pmatrix}.$$

Note that in extending the present linear scheme to the non-linear (essentially non-oscillatory) version the WENO reconstruction should be used for computing the convective flux only (12). The use of any limited spatial reconstruction of TVD or WENO type for computing the diffusion flux (13) can lead to wrong results. For more details and possible approaches for incorporating non-linear reconstruction into the diffusion flux see [3, 7].

Finally, the numerical source term $\mathbf{S}_i$ is calculated by decomposing the cell V_i into tetrahedrons and then a using standard ten-point volume Gaussian quadrature:

$$\boldsymbol{S}_i \approx \frac{1}{|V_i|} \sum_\gamma \mathbf{S}(\boldsymbol{x}_\gamma, t, \mathbf{P}_i(\boldsymbol{x}_\gamma, t))\omega_\gamma,$$

where again the index γ refers to the Gaussian points and weights.

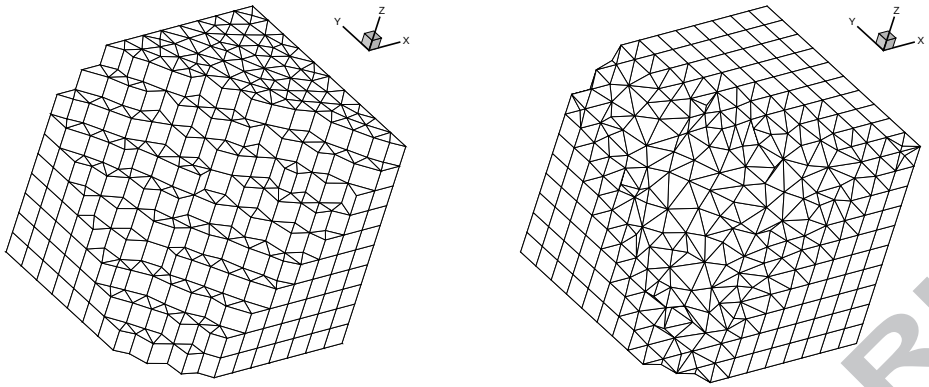


Figure 2: Examples of prismatic (left) and hybrid (right) computational meshes.

4. Numerical examples

In this section the numerical results of the 3rd order method, considered by the authors to be the most practical one, are presented. The calculations are carried out in a cube $[0,1]^3$ with periodic boundary conditions using a fixed Courant number $K = 0.3$. The unstructured meshes utilized in calculations are constructed as follows. First the number of cells N_{edge} over each edge of the cube is specified. Then, for each N_{edge} three sequences of meshes are constructed: purely hexahedral, purely prismatic as well as a mixed-element mesh, consisting of tetrahedral and pyramidal elements. Each mesh in the sequence is obtained using $(N_{\text{edge}})^k$, $k = 1, 2, 3$, cells over each edge of the cube. The total number of computational cells in the domain is then denoted as N_{tot} and is a function of both N_{edge} and the type of elements used inside the computational domain.

In the convergence studies, three levels of refinement are used, corresponding to the starting mesh with $N_{edge} = 10$. The summary of total number of

Table 1: Mesh summary

N_{edge}	Hexahedral mesh	Prismatic mesh	Hybrid mesh
10	10^3	3.4×10^3	1.1×10^4
20	8×10^3	2.8×10^4	6.9×10^4
40	6.4×10^4	2.2×10^5	4.0×10^5

elements for each mesh type can be found in Table 1. For example, for $N_{edge} = 10$ the hexahedral mesh consists of 1000 cells, the prismatic mesh consists of 3420 prims whereas the hybrid mesh contains 10364 tetrahedrons and 600 pyramids. Cutaway sections of the corresponding prismatic and hybrid meshes are shown on Fig. 2.

The convergence studies for the system of conservation laws (8) rely on the method of manufactured solutions. The exact (manufactured) solution of the equations is prescribed as

$$T(x, y, z, t) = 1 + \varepsilon(\sin(2\pi x) + \sin(2\pi y) + \sin(2\pi z) + \sin(t)),$$

$$\rho = 1, \quad u = v = w = 0, \quad p = \rho T \equiv T, \quad \varepsilon = \frac{1}{10}.$$

The initial condition of the problem is obtained by setting $t = 0$ in the manufactured solution. The expression for the vector source term $\mathbf{S}$ is obtained by inserting the exact solution into the homogenous form of the equations (8), which result in the following:

$$\mathbf{S} = \left(0, \quad 2\pi\varepsilon\cos(2\pi x), \quad 2\pi\varepsilon\cos(2\pi y), \quad 2\pi\varepsilon\cos(2\pi z), \quad \varepsilon\frac{\cos(t)}{\gamma - 1}\right)^T$$

In convergence studies the values of the error ϵ_k corresponding to mesh k are calculated at output time $t = 0.1$ and are used to estimate the conver-

gence rate r. The conventional estimate, stemming for the structured-mesh studies, uses the number of cell along the edge of the computational domain to characterize each mesh resolution and is given by

$$r = -\frac{\log\frac{\epsilon_k}{\epsilon_{k+1}}}{\log 2} \tag{14}$$

Table 2 shows the resulting convergence rates. It is observed that the performance of the method depends on the type of the mesh. The highest convergence rate is attained on the hexahedral mesh whereas the hybrid tetrahedral-pyramidal mesh is the most difficult to handle. However, since the total number of spatial cells N_{tot} for the hybrid mesh grows slower than N_{edge}^3, the so-computed convergence rates may underestimate the actual performance of the method, which is seen in the results for hybrid meshes.

Another way of estimating the convergence rate of the method on hybrid unstructured meshes is to take $N_{tot}^{1/3}$ as an estimate of the mesh resolution in each spatial direction and use it for computing the convergence rate according to the formula:

$$r = -\frac{\log\frac{\epsilon_k}{\epsilon_{k+1}}}{\log\frac{m_k}{m_{k+1}}}, \quad m_k = \left(N_{tot}^{1/3}\right)_k \tag{15}$$

Table 3 shows the corresponding convergence rates. It is observed that for the hybrid mesh the convergence rates have now improved significantly and approaches the theoretically expected third-order accuracy. The results for hexahedral and prismatic meshes have not changed since for these meshes m_k is proportional to N_{edge}.

Table 2: Convergence study for the compressible flow equations (8), using expression (14) for the convergence rate. Given are the errors of the total energy E.

N_{edge}	L_0 error	L_0 rate	L_1 error	L_1 rate
		Hexahedral mesh		
10	1.10×10^{-2}		3.86×10^{-3}	
20	1.75×10^{-3}	2.65	5.22×10^{-4}	2.89
40	1.99×10^{-4}	3.13	5.88×10^{-5}	3.15
		Prismatic mesh		
10	5.09×10^{-3}		1.88×10^{-3}	
20	6.57×10^{-4}	2.95	2.51×10^{-4}	2.90
40	7.51×10^{-5}	3.13	2.98×10^{-5}	3.07
		Hybrid mesh		
10	1.72×10^{-3}		4.13×10^{-4}	
20	4.38×10^{-4}	1.98	7.31×10^{-5}	2.50
40	7.87×10^{-5}	2.48	1.23×10^{-5}	2.58

5. Conclusions

This paper has focused on the development of uniformly high-order accurate finite-volume schemes for arbitrary-element three-dimensional unstructured meshes as applied to the advection-diffusion equations. A detailed description of the reconstruction step of schemes as well as calculation of convection and diffusive fluxes and the source term is provided allowing for its practical implementation by the reader. The presented numerical results demonstrate that the new scheme achieves the designed third-order order of

Table 3: Convergence study for the compressible flow equations (8), using expression (14) for the convergence rate. Given are the errors of the total energy E.

$N_{total}^{1/3}$	L_0 error	L_0 rate	L_1 error	L_1 rate
	Hexahedral mesh			
10	1.10×10^{-2}		3.86×10^{-3}	
20	1.75×10^{-3}	2.65	5.22×10^{-4}	2.89
40	1.99×10^{-4}	3.13	5.88×10^{-5}	3.15
	Prismatic mesh			
15	5.09×10^{-3}		1.88×10^{-3}	
30	6.57×10^{-4}	2.95	2.51×10^{-4}	2.90
60	7.51×10^{-5}	3.13	2.98×10^{-5}	3.07
	Hybrid mesh			
22	1.72×10^{-3}		4.13×10^{-4}	
41	4.38×10^{-4}	2.20	7.31×10^{-5}	2.78
73	7.87×10^{-5}	2.97	1.23×10^{-5}	3.10

accuracy for arbitrary elements. Further work will include the incorporation of recent WENO approaches for spatial discretization [2, 6, 5] as well as ADER-type time evolution methods [9, 2].

[1] Dumbser, M., 2010. Arbitrary High Order PNPM Schemes on Unstructured Meshes for the Compressible Navier-Stokes Equations. Computers & Fluids 39 (1), 60–76.

[2] Dumbser, M., Käser, M., Titarev, V., Toro, E., 2007. Quadrature-free

non-oscillatory finite volume schemes on unstructured meshes for nonlinear hyperbolic systems. Journal of Computational Physics 226, 204–243.

[3] Gassner, G., Lörcher, F., Munz, C., 2007. A contribution to the construction of diffusion fluxes for finite volume and discontinuous Galerkin schemes. Journal of Computational Physics 224, 1049–1063.

[4] Jiang, G., Shu, C., 1996. Efficient implementation of weighted ENO schemes. J. Comput. Phys. 126, 202–212.

[5] P. Tsoutsanis, V.A. Titarev, D. D., 2010. WENO schemes on arbitrary mixed-element unstructured meshes in three space dimensions. Journal of Computational Physics, *submitted*.

[6] Titarev, V., Tsoutsanis, P., Drikakis, D., 2010. WENO schemes for mixed-element unstructured meshes. Communications in Computational Physics 8 (3), 585–609.

[7] Toro, E., Hidalgo, A., 2009. ADER finite volume schemes for nonlinear reaction-diffusion equations. Applied Numerical Mathematics 59, 73–100.

[8] Toro, E., Spruce, M., Speares, W., 1994. Restoration of the contact surface in the Harten-Lax-van Leer Riemann solver. Journal of Shock Waves 4, 25–34.

[9] Toro, E., Titarev, V., 2006. Derivative Riemann solvers for systems of conservation laws and ADER methods. J. Comput. Phys. 212 (1), 150–165.

[10] Zhang, Y.-T., Shu, C.-W., 2009. Third order WENO scheme on three dimensional tetrahedral meshes. Communications in Computational Physics 5 (2-4), 836–848.