
Algebraic Flux Correction II. Compressible Euler Equations

Dmitri Kuzmin¹ and Matthias Möller²

¹ Institute of Applied Mathematics (LS III), University of Dortmund
Vogelpothsweg 87, D-44227, Dortmund, Germany
kuzmin@math.uni-dortmund.de

² matthias.moeller@math.uni-dortmund.de

Summary. Algebraic flux correction schemes of TVD and FCT type are extended to systems of hyperbolic conservation laws. The group finite element formulation is employed for the treatment of the compressible Euler equations. An efficient algorithm is proposed for the edge-by-edge matrix assembly. A generalization of Roe's approximate Riemann solver is derived by rendering all off-diagonal matrix blocks positive semi-definite. Another usable low-order method is constructed by adding scalar artificial viscosity proportional to the spectral radius of the cumulative Roe matrix. The limiting of antidiffusive fluxes is performed using a transformation to the characteristic variables or a suitable synchronization of correction factors for the conservative ones. The outer defect correction loop is equipped with a block-diagonal preconditioner so as to decouple the discretized Euler equations and solve them in a segregated fashion. As an alternative, a strongly coupled solution strategy (global BiCGSTAB method with a block-Gauß-Seidel preconditioner) is introduced for applications which call for the use of large time steps. Various algorithmic aspects including the implementation of characteristic boundary conditions are addressed. Simulation results are presented for inviscid flows in a wide range of Mach numbers.

1 Introduction

Unstructured grid finite element methods appear to be particularly attractive for the treatment of aerodynamic applications governed by the compressible Euler equations [25],[28],[35]. However, most of the algorithms currently in use are explicit and, consequently, subject to the CFL condition which becomes very restrictive in the presence of adaptive mesh refinement. For steady-state flows the computational cost can be drastically reduced by using *local time-stepping* to achieve a (more) uniform distribution of Courant numbers in the domain. For transient flows this approach is not feasible since using different time steps at different mesh points may result in a loss of 'mass' and allow shocks to move at wrong speed. Hence, there is a need for the development of truly *implicit* high-resolution finite element schemes which are unconditionally

stable and much more efficient than explicit ones for the above-mentioned class of CFD applications. At the same time, their implementation is more difficult and the actual performance strongly depends on the choice of data structures, configuration of linear solvers, efficiency of the matrix assembly, and other algorithmic details. In this chapter, we extend the implicit FEM-FCT and FEM-TVD schemes to systems of hyperbolic conservation laws, discuss the subtleties of algebraic flux correction for the Euler equations and present some iterative solution strategies for the resulting algebraic systems.

2 Compressible Euler Equations

The Euler equations of gas dynamics represent a system of conservation laws for the mass, momentum, and energy of an inviscid compressible fluid

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0, \quad (1)$$

$$\frac{\partial(\rho \mathbf{v})}{\partial t} + \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v}) + \nabla p = 0, \quad (2)$$

$$\frac{\partial(\rho E)}{\partial t} + \nabla \cdot (\rho H \mathbf{v}) = 0, \quad (3)$$

where ρ , \mathbf{v} , p , E and $H = E + p/\rho$ are the density, velocity, pressure, total energy per unit mass, and stagnation enthalpy, respectively. This first-order PDE system is a simplification of the more realistic *compressible Navier-Stokes equations* which include the effects of viscosity and heat conduction.

The internal energy is assumed to be a known function of pressure and density. For an ideal polytropic gas, we have the following equation of state

$$p = (\gamma - 1)\rho \left(E - \frac{|\mathbf{v}|^2}{2} \right). \quad (4)$$

In this relation, the constant γ is the ratio of specific heats ($\gamma = 1.4$ for air).

Introducing the vector of conservative variables U and the triple of fluxes $\mathbf{F} = (F^1, F^2, F^3)$ for each coordinate direction in the Euclidean space \mathbb{R}^3

$$U = \begin{bmatrix} \rho \\ \rho \mathbf{v} \\ \rho E \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \rho \mathbf{v} \\ \rho \mathbf{v} \otimes \mathbf{v} + p \mathcal{I} \\ \rho H \mathbf{v} \end{bmatrix}, \quad (5)$$

where \mathcal{I} stands for the identity tensor, we can represent the (three-dimensional) Euler equations in the standard divergence form as follows

$$\frac{\partial U}{\partial t} + \nabla \cdot \mathbf{F} = 0, \quad \text{where} \quad \nabla \cdot \mathbf{F} = \sum_{d=1}^3 \frac{\partial F^d}{\partial x_d}. \quad (6)$$

The chain rule yields an equivalent quasi-linear formulation in which the spatial derivatives are applied to the conservative variables rather than fluxes

$$\frac{\partial U}{\partial t} + \mathbf{A} \cdot \nabla U = 0, \quad \text{where} \quad \mathbf{A} \cdot \nabla U = \sum_{d=1}^3 A^d \frac{\partial U}{\partial x_d}. \quad (7)$$

Here $\mathbf{A} = (A^1, A^2, A^3)$ denotes the triple of Jacobian matrices such that [28]

$$F^d = A^d U, \quad A^d = \frac{\partial F^d}{\partial U}, \quad d = 1, 2, 3. \quad (8)$$

The first relation in this formula means that each flux component is a homogeneous function of the conservative variables [45].

Due to the hyperbolicity of the Euler equations, any linear combination of the three Jacobians is diagonalizable with real eigenvalues. In other words, the inner product of \mathbf{A} with an arbitrary vector $\mathbf{e} = (e_1, e_2, e_3)$ of the 3D space admits the following factorization [17],[22],[45]

$$\mathbf{e} \cdot \mathbf{A} = \sum_{d=1}^3 e_d A^d = R \Lambda(\mathbf{e}) R^{-1}, \quad (9)$$

where $\Lambda(\mathbf{e})$ is the diagonal matrix of eigenvalues and R is the matrix of right eigenvectors whereas R^{-1} is composed from the left ones. Analytical expressions for these matrices can be found, for instance, in [38]. At the same time, it is impossible to diagonalize the unidirectional Jacobians A^d simultaneously as they do not commute and, therefore, have different sets of eigenvalues [17]. This information on the spectral properties of \mathbf{A} is important for the design of numerical methods and will be utilized in what follows.

3 High-Order Scheme

The first algorithm to be presented is the basic Galerkin scheme to be endowed with our algebraic flux limiters of FCT and TVD type. As already mentioned in the Introduction, *implicit* finite element methods are still rarely used in compressible flow simulations. Therefore, matrix assembly for the Euler equations has received little attention in the literature. In this section, we show how it can be implemented in an efficient way building on Roe's linearization technique [37] for systems of hyperbolic conservation laws.

Let us start with the divergence form (6) of the Euler equations and discretize them in space leaving the time derivative continuous for the time being. As in the scalar case, we adopt the group finite element formulation. That is, both the conservative variables and the flux function are approximated in terms of their nodal values multiplied by the FEM basis functions:

$$U_h(\mathbf{x}, t) = \sum_j U_j(t) \varphi_j(\mathbf{x}), \quad \mathbf{F}_h(\mathbf{x}, t) = \sum_j \mathbf{F}_j(t) \varphi_j(\mathbf{x}). \quad (10)$$

The use of these approximations in the weak form of (6) leads to a system of semi-discretized equations for the time-dependent nodal values

$$\sum_j \left[\int_{\Omega} \varphi_i \varphi_j \, d\mathbf{x} \right] \frac{dU_j}{dt} + \sum_j \left[\int_{\Omega} \varphi_i \nabla \varphi_j \, d\mathbf{x} \right] \cdot \mathbf{F}_j = 0. \quad (11)$$

In order to solve it numerically, we need to eliminate the dependent variables \mathbf{F}_j in favor of the unknowns U_j so as to obtain an ODE system of the form

$$M_C \frac{d\mathbf{U}}{dt} = K\mathbf{U}, \quad (12)$$

where M_C is the block-diagonal mass matrix and K is a discrete counterpart of the operator $-\mathbf{A} \cdot \nabla$ for the quasi-linear formulation (7). Of course, it depends on the solution vector \mathbf{U} since the governing equations are nonlinear.

Recall that the basis functions φ_j sum to unity, so that the sum of their derivatives vanishes at every point. Therefore, the auxiliary coefficients \mathbf{c}_{ij} as defined in equation (6) of the previous chapter have zero row sums:

$$\mathbf{c}_{ij} = \int_{\Omega} \varphi_i \nabla \varphi_j \, d\mathbf{x}, \quad \mathbf{c}_{ii} = - \sum_{j \neq i} \mathbf{c}_{ij}. \quad (13)$$

It follows from (11) that the right-hand side of the five coupled equations for node i can be expressed as a sum of edge contributions

$$(K\mathbf{U})_i = - \sum_j \mathbf{c}_{ij} \cdot \mathbf{F}_j = - \sum_{j \neq i} \mathbf{c}_{ij} \cdot (\mathbf{F}_j - \mathbf{F}_i). \quad (14)$$

This sort of representation is typical of finite volume/difference schemes, which enables us to borrow many useful ideas originally developed in this context.

In his pioneering work on approximate Riemann solvers [37], Roe showed that the differences between the components of \mathbf{F} and \mathbf{U} are related by

$$\mathbf{F}_j - \mathbf{F}_i = \hat{\mathbf{A}}_{ij} (U_j - U_i), \quad (15)$$

where the triple of matrices $\hat{\mathbf{A}}_{ij} = (\hat{\mathbf{A}}_{ij}^1, \hat{\mathbf{A}}_{ij}^2, \hat{\mathbf{A}}_{ij}^3)$ corresponds to the Jacobian tensor \mathbf{A} evaluated for a special set of *density-averaged* variables

$$\hat{\rho}_{ij} = \sqrt{\rho_i \rho_j}, \quad \hat{\mathbf{v}}_{ij} = \frac{\sqrt{\rho_i} \mathbf{v}_i + \sqrt{\rho_j} \mathbf{v}_j}{\sqrt{\rho_i} + \sqrt{\rho_j}}, \quad \hat{H}_{ij} = \frac{\sqrt{\rho_i} H_i + \sqrt{\rho_j} H_j}{\sqrt{\rho_i} + \sqrt{\rho_j}} \quad (16)$$

which are called the Roe mean values. The speed of sound \hat{c}_{ij} for the intermediate state $(\hat{\rho}_{ij}, \hat{\mathbf{v}}_{ij}, \hat{H}_{ij})$ is determined as follows:

$$\hat{c}_{ij} = \sqrt{(\gamma - 1) \left(\hat{H}_{ij} - \frac{|\hat{\mathbf{v}}_{ij}|^2}{2} \right)}. \quad (17)$$

By virtue of this linearization, the right-hand side of (12) can be assembled from individual edge contributions which depend on the difference between the values of the conservative variables/fluxes at nodes i and j

$$(KU)_i \longleftarrow \mathbf{c}_{ij} \cdot (\mathbf{F}_i - \mathbf{F}_j) = \mathbf{c}_{ij} \cdot \hat{\mathbf{A}}_{ij}(U_i - U_j), \quad (18)$$

$$(KU)_j \longleftarrow \mathbf{c}_{ji} \cdot (\mathbf{F}_j - \mathbf{F}_i) = \mathbf{c}_{ji} \cdot \hat{\mathbf{A}}_{ij}(U_j - U_i). \quad (19)$$

Let us represent the linear combinations of the averaged Jacobians in terms of the *cumulative Roe matrices* A_{ij} and B_{ij} defined by

$$A_{ij} = \mathbf{a}_{ij} \cdot \hat{\mathbf{A}}_{ij}, \quad \mathbf{a}_{ij} = \frac{\mathbf{c}_{ij} - \mathbf{c}_{ji}}{2}, \quad (20)$$

$$B_{ij} = \mathbf{b}_{ij} \cdot \hat{\mathbf{A}}_{ij}, \quad \mathbf{b}_{ij} = \frac{\mathbf{c}_{ij} + \mathbf{c}_{ji}}{2}. \quad (21)$$

Integration by parts in (13) reveals that the coefficients \mathbf{c}_{ij} and \mathbf{c}_{ji} satisfy

$$\mathbf{c}_{ji} = -\mathbf{c}_{ij} + \int_{\Gamma} \varphi_i \varphi_j \mathbf{n} \, ds, \quad (22)$$

where \mathbf{n} denotes the unit outward normal to the boundary Γ . We remark that finite element basis functions have compact support. As a rule, the surface integral is equal to zero unless both nodes belong to the boundary.

By substitution of (22) into formulae (20) and (21), we obtain

$$\mathbf{a}_{ij} = \mathbf{c}_{ij} - \frac{1}{2} \mathbf{s}_{ij}, \quad \mathbf{b}_{ij} = \frac{1}{2} \mathbf{s}_{ij}, \quad (23)$$

where \mathbf{s}_{ij} is an entry of the ‘mass matrix’ for the surface triangulation

$$\mathbf{s}_{ij} = \int_{\Gamma} \varphi_i \varphi_j \mathbf{n} \, ds.$$

In the interior of the domain, \mathbf{s}_{ij} vanishes and (23) reduces to

$$\mathbf{a}_{ij} = \mathbf{c}_{ij}, \quad \mathbf{b}_{ij} = 0.$$

Consequently, just the antisymmetric part A_{ij} of the cumulative Roe matrix is to be evaluated for each interior edge. The symmetric part B_{ij} is only needed at the boundary and can sometimes be neglected (see below).

It is easy to verify that the inner products in (18)–(19) are given by

$$\mathbf{c}_{ij} \cdot \hat{\mathbf{A}}_{ij} = A_{ij} + B_{ij}, \quad \mathbf{c}_{ji} \cdot \hat{\mathbf{A}}_{ij} = -A_{ij} + B_{ij}. \quad (24)$$

Hence, the contribution of the edge ij to the right-hand side of (12) reads

$$(KU)_i \longleftarrow (A_{ij} + B_{ij})(U_i - U_j), \quad (25)$$

$$(KU)_j \longleftarrow (A_{ij} - B_{ij})(U_i - U_j). \quad (26)$$

This representation leads to a very efficient edge-based algorithm for matrix assembly since there is no need for numerical integration as long as the coefficients \mathbf{c}_{ij} are initialized and stored. The sparsity graph of the global matrix K depends solely on the underlying mesh and on the type of approximation. For systems of equations, the list of edges is the same as in the scalar case. However, there are interactions not only between basis functions for different mesh nodes but also between those for different variables. Due to this intimate coupling of degrees of freedom, each ‘coefficient’ of the discrete operator turns into a matrix of size equal to the squared number of variables.

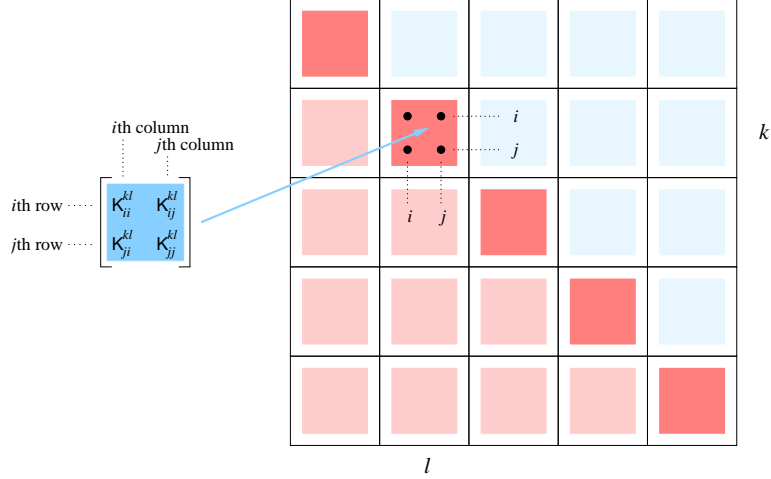


Fig. 1. Matrix assembly for the Euler equations.

If the global finite element matrix (or some parts of it) rather than its product with the solution vector \mathbf{U} needs to be assembled, this can be accomplished by evaluating the four 5×5 blocks

$$\begin{aligned} K_{ii} &= A_{ij} + B_{ij}, & K_{ij} &= -A_{ij} - B_{ij}, \\ K_{ji} &= A_{ij} - B_{ij}, & K_{jj} &= -A_{ij} + B_{ij} \end{aligned} \quad (27)$$

edge-by-edge and scattering their entries to the positions with indices i and j in the corresponding blocks of the operator K . This process is illustrated in Fig. 1. As a rule, it is not necessary to assemble and store the whole global matrix. Instead, it is possible to piece together individual edge contributions of the form (25)–(26) so as to avoid matrix-vector multiplications like $K\mathbf{U}$. Depending on the choice of smoothers/preconditioners for the iterative solver, just a few (if any) global blocks need to be generated explicitly. The block-diagonal or the upper/lower triangular part will suffice for our purposes.

4 Low-Order Scheme

To a large extent, the ability of a high-resolution scheme to withstand the formation of wiggles depends on the quality of the underlying low-order method. Let us construct it at the algebraic level following the strategy presented in the previous chapter. To this end, we perform the row-sum mass lumping and replace the original Galerkin discretization (12) by

$$M_L \frac{dU}{dt} = LU, \quad (28)$$

where M_L denotes the lumped mass matrix and L is the low-order Jacobian operator. Recall that its scalar counterpart was derived by ‘discrete upwinding’ which amounts to a conservative elimination of negative off-diagonal entries from the high-order transport operator. For systems of conservation laws, the edge contributions to the global matrix are no longer scalar quantities but matrices themselves. This leads to the following algebraic constraint:

LED principle for systems (semi-discrete level)

Let the system of ordinary differential equations for the values of the conservative variables at node i be represented in the form

$$m_i \frac{dU_i}{dt} = \sum_{j \neq i} L_{ij} (U_j - U_i). \quad (29)$$

If all off-diagonal matrix blocks L_{ij} are *positive semi-definite* (that is, their eigenvalues are nonnegative), then such a discretization is local extremum diminishing for a certain set of local *characteristic variables*. Obviously, this condition is consistent with the LED criterion for scalar conservation laws but in the case of a hyperbolic system it is much less restrictive than the requirement that all off-diagonal coefficients of L be nonnegative.

To construct a nonoscillatory low-order scheme for the compressible Euler equations, we add tensorial artificial viscosity D_{ij} and remove the symmetric part B_{ij} of the cumulative Roe matrix in (27). This gives

$$\begin{aligned} L_{ii} &= A_{ij} - D_{ij}, & L_{ij} &= -A_{ij} + D_{ij}, \\ L_{ji} &= A_{ij} + D_{ij}, & L_{jj} &= -A_{ij} - D_{ij}. \end{aligned} \quad (30)$$

These modified edge contributions are built into the blocks of L as explained above. It is worth mentioning that the high-order operator K does not need to be assembled at all. The corresponding raw antidiffusive flux reads

$$F_{ij} = - \left(M_{ij} \frac{d}{dt} + D_{ij} + B_{ij} \right) (U_j - U_i), \quad F_{ji} = -F_{ij}. \quad (31)$$

The block $M_{ij} = m_{ij}I$, where I denotes the 5×5 identity matrix, is responsible for the error induced by mass lumping. After the time discretization, one obtains an expression similar to (71) of the preceding chapter. It remains to design the matrix D_{ij} so as to enforce the generalized LED constraint.

5 Design of Artificial Viscosities

As pointed out earlier, the system of Euler equations is hyperbolic so that any linear combination of the three Jacobian matrices is diagonalizable with real eigenvalues. In particular, there exists a diagonal matrix Λ_{ij} and a regular matrix R_{ij} of right eigenvectors such that the cumulative Roe matrix A_{ij} for the edge ij admits the following factorization

$$A_{ij} = |\mathbf{a}_{ij}| R_{ij} \Lambda_{ij} R_{ij}^{-1}. \quad (32)$$

The scaling factor is given by the Euclidean norm of the coefficient vector

$$|\mathbf{a}_{ij}| = \sqrt{\mathbf{a}_{ij} \cdot \mathbf{a}_{ij}}, \quad \mathbf{a}_{ij} = (a_{ij}^1, a_{ij}^2, a_{ij}^3)$$

and the diagonal elements of the eigenvalue matrix $\Lambda_{ij} = \text{diag}\{\lambda_1, \dots, \lambda_5\}$ correspond to the characteristic speeds of wave propagation

$$\lambda_1 = \hat{v}_{ij} - \hat{c}_{ij}, \quad \lambda_2 = \lambda_3 = \lambda_4 = \hat{v}_{ij}, \quad \lambda_5 = \hat{v}_{ij} + \hat{c}_{ij}. \quad (33)$$

Here \hat{c}_{ij} is the speed of sound for Roe's linearization as defined in (17), while \hat{v}_{ij} is a 'projection' of the density-averaged velocity onto the edge ij

$$\hat{v}_{ij} = \mathbf{e}_{ij} \cdot \hat{\mathbf{v}}_{ij}, \quad \text{where} \quad \mathbf{e}_{ij} = \frac{\mathbf{a}_{ij}}{|\mathbf{a}_{ij}|}. \quad (34)$$

In the continuous case, the characteristics associated with the multiple eigenvalue $\lambda_2 = \lambda_3 = \lambda_4$ represent the trajectories of fluid particles. In addition, there are two superimposed acoustic waves traveling at speeds $\pm c$ relative to the gas. We remark that the 'upwind' direction is different for different waves since it depends on the sign of the eigenvalues. This is why we abstain from fixing the order of nodes for the edges of the sparsity graph and postpone the discussion of edge orientation for TVD limiters until section 7.2.

The characteristic decomposition (32) enables us to eliminate the negative eigenvalues in the coefficient blocks L_{ij} and L_{ji} of the discrete Jacobian by resorting to an analog of discrete upwinding. The artificial dissipation D_{ij} for the resulting 'flux difference splitting' scheme is defined as follows [19]

$$D_{ij} = |A_{ij}| = |\mathbf{a}_{ij}| R_{ij} |\Lambda_{ij}| R_{ij}^{-1}, \quad (35)$$

where the matrix $|\Lambda_{ij}|$ contains the absolute values of the eigenvalues

$$|\Lambda_{ij}| = \text{diag}\{|\lambda_1|, \dots, |\lambda_5|\}. \quad (36)$$

In one dimension, we recover Roe's approximate Riemann solver [37] which is one of the most popular discretization techniques for the Euler equations. For a comprehensive review and a comparative study of such upwind-biased low-order schemes the interested reader is referred to [28],[30].

Instead of dealing with the cumulative Roe matrix, it is possible to employ *dimensional splitting* and diagonalize the Jacobians one at a time [17],[22]

$$D_{ij} = \sum_{d=1}^3 |A_{ij}^d|, \quad \text{where } A_{ij}^d = a_{ij}^d \hat{A}_{ij}^d. \quad (37)$$

This decomposition into one-dimensional wave patterns brings about strong numerical diffusion in the crosswind direction. Moreover, the cost of evaluating D_{ij} triples in comparison to (35). On the other hand, the implementation is very simple and flux limiting in terms of characteristic variables is feasible. Many compressible flow solvers are based on this sort of splitting.

A much cheaper alternative to (35) and (37) is to add scalar dissipation proportional to the spectral radius of the Roe matrix [25],[28],[49]

$$D_{ij} = d_{ij} \mathbf{1}, \quad \text{where } d_{ij} = |\mathbf{a}_{ij}| \max_i |\lambda_i|. \quad (38)$$

Note that it affects just the diagonal blocks of the discrete Jacobian operator and is the same for all variables. This simplifies bookkeeping and reduces the cost of matrix assembly. As long as excessive artificial viscosity is removed in the course of flux correction, the computational effort required for evaluation of D_{ij} from (35) or (37) does not pay off. Surprisingly enough, a slightly overdiffusive low-order method may even be preferable due to the resulting improvement of phase accuracy [19],[49]. Last but not least, scalar dissipation can be used as a preconditioner for the approximate Riemann solver.

Remark. The insertion of artificial diffusion can be combined with a conservative flux decomposition for the term KU , which yields [18],[19]

$$(KU)_i = - \sum_{j \neq i} G_{ij}, \quad G_{ij} = \mathbf{c}_{ij} \cdot \mathbf{F}_i - \mathbf{c}_{ji} \cdot \mathbf{F}_j. \quad (39)$$

Such a representation is particularly useful for finite element codes utilizing an edge-based data structure. Moreover, it facilitates an extension of many one-dimensional discretization schemes to unstructured meshes [28],[32].

The low-order counterpart of G_{ij} can be constructed by adding a diffusive flux depending on the spectral properties of the Roe matrix

$$G_{ij}^* = G_{ij} - (D_{ij} + B_{ij})(U_j - U_i). \quad (40)$$

This modification is equivalent to using (30) in lieu of (27) during the assembly of the global Jacobian operator. Hence, the right-hand side of the semi-discrete low-order scheme (29) can be expressed as

$$(LU)_i = - \sum_{j \neq i} G_{ij}^*, \quad G_{ji}^* = -G_{ij}^* \quad (41)$$

and assembled edge-by-edge from the antisymmetric numerical fluxes G_{ij}^* if this is desirable from the viewpoint of implementation.

6 One-Dimensional Case

Before embarking on the design of flux limiters, it is instructive to apply the above low-order method to the one-dimensional Euler equations

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} = 0. \quad (42)$$

In this case, the vectors of unknown variables and fluxes reduce to

$$U = \begin{bmatrix} \rho \\ \rho v \\ \rho E \end{bmatrix}, \quad F = \begin{bmatrix} \rho v \\ \rho v^2 + p \\ \rho H v \end{bmatrix}. \quad (43)$$

The differentiation of F yields the equivalent quasi-linear form

$$\frac{\partial U}{\partial t} + A \frac{\partial U}{\partial x} = 0, \quad (44)$$

where $A = \frac{\partial F}{\partial U}$ is the Jacobian matrix. It is easy to verify that

$$A = \begin{bmatrix} 0 & 1 & 0 \\ \frac{1}{2}(\gamma - 3)v^2 & (3 - \gamma)v & \gamma - 1 \\ \frac{1}{2}(\gamma - 1)v^3 - vH & H - (\gamma - 1)v^2 & \gamma v \end{bmatrix}. \quad (45)$$

Furthermore, the characteristic decomposition of A is as follows

$$A = R \Lambda R^{-1}, \quad \Lambda = \text{diag}\{v - c, v, v + c\}, \quad (46)$$

where v is the fluid velocity and $c = \sqrt{\gamma p / \rho}$ is the local speed of sound for a polytropic gas. The ratio $M = \frac{|v|}{c}$ is called the *Mach number*.

The columns of the matrix R represent the right eigenvectors

$$R = \begin{bmatrix} 1 & 1 & 1 \\ v - c & v & v + c \\ H - vc & \frac{1}{2}v^2 & H + vc \end{bmatrix} = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3] \quad (47)$$

and the rows of its inverse R^{-1} correspond to the left ones

$$R^{-1} = \begin{bmatrix} \frac{1}{2}(b_1 + \frac{v}{c}) & \frac{1}{2}(-b_2v - \frac{1}{c}) & \frac{1}{2}b_2 \\ 1 - b_1 & b_2v & -b_2 \\ \frac{1}{2}(b_1 - \frac{v}{c}) & \frac{1}{2}(-b_2v + \frac{1}{c}) & \frac{1}{2}b_2 \end{bmatrix} = \begin{bmatrix} \mathbf{l}_1 \\ \mathbf{l}_2 \\ \mathbf{l}_3 \end{bmatrix}, \quad (48)$$

where the auxiliary coefficients b_1 and b_2 are given by

$$b_1 = b_2 \frac{v^2}{2}, \quad b_2 = \frac{\gamma - 1}{c^2}.$$

By definition, the eigenvalues and eigenvectors of A satisfy the equations

$$\mathbf{A}\mathbf{r}_k = \lambda_k \mathbf{r}_k, \quad \mathbf{l}_k \mathbf{A} = \lambda_k \mathbf{l}_k, \quad k = 1, 2, 3 \quad (49)$$

which can be written in matrix form as $\mathbf{A}\mathbf{R} = \mathbf{R}\mathbf{\Lambda}$ and $\mathbf{R}^{-1}\mathbf{A} = \mathbf{\Lambda}\mathbf{R}^{-1}$.

After the space discretization by the low-order scheme, we end up with the following ODE system for the nodal values of the conservative variables

$$m_i \frac{d\mathbf{U}_i}{dt} = \sum_{j \neq i} \mathbf{L}_{ij} (\mathbf{U}_j - \mathbf{U}_i), \quad \text{where } \mathbf{L}_{ij} = \mathbf{K}_{ij} + \mathbf{D}_{ij}. \quad (50)$$

At interior nodes, the high-order element/edge contribution is of the form

$$\mathbf{K}_{ij} = -c_{ij} \hat{\mathbf{A}}_{ij} = -\mathbf{A}_{ij}, \quad j = i \pm 1, \quad (51)$$

where $\hat{\mathbf{A}}_{ij}$ is the Jacobian matrix (45) evaluated using the Roe mean values $(\hat{\rho}_{ij}, \hat{v}_{ij}, \hat{H}_{ij})$. On a uniform mesh of linear finite elements, we have

$$m_i = \Delta x, \quad c_{ij} = \begin{cases} 1/2 & \text{for } j = i + 1, \\ -1/2 & \text{for } j = i - 1. \end{cases} \quad (52)$$

The viscous dissipation \mathbf{D}_{ij} defined in (35) and (37) simplifies to

$$\mathbf{D}_{ij} = \frac{1}{2} |\hat{\mathbf{A}}_{ij}|, \quad \text{where } |\hat{\mathbf{A}}_{ij}| = \mathbf{R}_{ij} |\mathbf{\Lambda}_{ij}| \mathbf{R}_{ij}^{-1}. \quad (53)$$

Plugging it into formula (40), we obtain the corresponding numerical flux, which is identical to that for Roe's approximate Riemann solver [37]

$$\mathbf{G}_{ij}^* = \frac{\mathbf{F}_i + \mathbf{F}_j}{2} - \frac{1}{2} |\hat{\mathbf{A}}_{ij}| (\mathbf{U}_j - \mathbf{U}_i), \quad j = i + 1. \quad (54)$$

The underlying principles and the properties of the resulting first-order scheme are described in many textbooks on gas dynamics [17],[22],[43]. We remark that Roe's method fails to recognize expansion waves and, therefore, may give rise to entropy-violating solutions (rarefaction shocks) in the vicinity of sonic points. A suitable *entropy fix*, for instance, the modification proposed by Harten and Hyman [14], makes it possible to circumvent this problem.

In the case of scalar dissipation (38), the amount of artificial viscosity is proportional to the largest in magnitude eigenvalue of $\hat{\mathbf{A}}_{ij}$

$$\mathbf{D}_{ij} = \frac{\lambda_{\max}}{2} \mathbf{I}, \quad \text{where } \lambda_{\max} = |\hat{v}_{ij}| + \hat{c}_{ij}. \quad (55)$$

Note that the diffusion coefficient $d_{ij} = \frac{\lambda_{\max}}{2}$ equals that for discrete upwinding applied to the fastest wave propagating at the characteristic speed λ_{\max} . This definition of \mathbf{D}_{ij} constitutes a cost-effective alternative to (53) and yields a perfect low-order method for the FCT algorithm [49].

7 Flux Limiting for Systems

Despite the remarkable progress made in the development of high-resolution schemes for scalar conservation laws, their extension to hyperbolic systems remains a challenging open problem. One of the main difficulties that hinder a rigorous generalization of the scalar machinery to the systems arena is the lack of reliable physical and mathematical criteria for the design of flux limiters. Some useful concepts, algorithms, and ideas are presented in this section. For an in-depth coverage of this topic, the reader is referred to [49].

It is tempting to treat the constituents of system (1)–(3) independently using a flux correction algorithm developed for scalar transport equations. Unfortunately, this naïve approach often yields rather disappointing results. In particular, a ‘blind’ adjustment of the conservative fluxes may turn out to be harmful and give rise to undershoots/overshoots in certain dependent variables, such as pressure, velocity, internal energy, entropy etc. The intricate coupling of the Euler equations makes it very difficult to monitor and control the evolution of all physically relevant quantities simultaneously. Hence, the numerical solutions are strongly influenced not only by the type of the flux limiter but also by the set of variables to which it is applied.

In light of the above, flux limiting for hyperbolic systems is more involved than that for scalar conservation laws. The design criteria should reflect the physical properties of the problem at hand and be enforced in a fail-safe way. To this end, the following strategies have been proposed [25],[49]

- flux limiting in terms of nonconservative (e.g. primitive, characteristic) variables, for which the constraints to be imposed are relevant/important;
- a proper synchronization of the correction factors for individual variables;
- an a-posteriori control and cancellation/tuning of the antidiffusive fluxes into nodes at which nonphysical solution values are detected.

These remedies are problem-dependent and require empirical input as well as a solid understanding of the underlying physics. However, the simulation results are typically rewarding. Following these guidelines, one can perform algebraic flux correction for the Euler equations as explained below.

7.1 Variable Transformations

The first issue to be addressed is the choice of variables. It is essential to use the divergence form (6) of the Euler equations and maintain conservation at the discrete level. This guarantees convergence to a weak solution and prevents shocks from moving at wrong speeds. However, the numerical method should allow for a physical growth/decay of local extrema, so imposing tight solution bounds on the conservative variables is not to be recommended [49]. Instead, viable constraints can be devised and enforced making use of a local transformation of the solution variations and of the raw antidiffusive fluxes to a set of variables that are more amenable to flux correction [25].

In particular, it is not unusual that flux limiting in terms of the primitive variables (ρ, \mathbf{v}, p) produces superior results as compared to the use of the conservative or mixed variables [28],[32]. An even better (albeit costly) alternative is to constrain a set of suitably defined *characteristic variables* for the linearized hyperbolic system. A decisive advantage of this approach is that scalar discretization tools are applicable to the transformed equations and there is no need for an *ad hoc* synchronization of the flux limiter. Moreover, the physical nature of wave propagation is taken into account. Thus, a proper control of the characteristic variables is typically sufficient to guarantee that all other variables are also free of nonphysical oscillations.

For linear hyperbolic systems such that the coefficient matrix A is constant, multiplication of (44) from the left by R^{-1} and use of (46) yields

$$\frac{\partial W}{\partial t} + \Lambda \frac{\partial W}{\partial x} = 0, \quad \text{where } W := R^{-1}U. \quad (56)$$

Since Λ is a diagonal matrix, this so-called *canonical form* of system (44) consists of three decoupled convection equations

$$\frac{\partial w_k}{\partial t} + \lambda_k \frac{\partial w_k}{\partial x} = 0, \quad k = 1, 2, 3. \quad (57)$$

It follows that the evolution of the characteristic variables $W = [w_1, w_2, w_3]^T$ is described by three simple waves propagating independently of one another with velocities v and $v \pm c$. The exact solution is constant along the characteristics of (57) which are the straight lines satisfying the ODEs

$$\frac{dx}{dt} = \lambda_k, \quad \text{where } \lambda_k = \text{const}. \quad (58)$$

The solution to the original problem can be recovered as $U = RW$ or

$$U = \sum_{k=1}^3 w_k \mathbf{r}_k, \quad w_k(x, t) = w_k(x - \lambda_k t, 0). \quad (59)$$

Hence, the characteristic variables w_k are the coefficients of \mathbf{r}_k in the eigenvector expansion of U which depends only on the initial data [20].

For nonlinear hyperbolic systems, the eigenvalues and eigenvectors of the Jacobian matrix A vary in space and time. Therefore, characteristic variables can only be defined *locally*, for a given solution U . After the spatial discretization, the transformation matrices R and R^{-1} are evaluated for a ‘frozen’ set of data such as the arithmetic average or the Roe mean values for two nodes. In particular, the tensorial artificial viscosity (53) corresponds to

$$D_{ij}(U_j - U_i) = \frac{1}{2} R_{ij} |\Lambda_{ij}| \Delta W_{ij}, \quad (60)$$

where the transformed solution increment ΔW_{ij} is given by

$$\Delta W_{ij} := R_{ij}^{-1}(U_j - U_i). \quad (61)$$

This representation reveals that Roe’s approximate Riemann solver can be interpreted as discrete upwinding applied to the decoupled scalar equations for the local characteristic variables. Indeed, the amount of artificial diffusion for field k is proportional to $|\lambda_k|/2$, which is just enough to render the scalar semi-discrete scheme local extremum diminishing (see the 1D example in section 5.1 of the previous chapter). In order to calculate the corresponding correction to the original discretization, the vector of characteristic diffusive fluxes $F_{ij} = \frac{1}{2}|\Lambda_{ij}|\Delta w_{ij}$ is converted back to the conservative variables in (60). In practice, both forward and backward variable transformations can/should be performed in a computationally efficient way without generating the eigenvector matrices for each edge explicitly [28],[48]. Arguing as above, one can construct characteristic flux limiters of TVD and FCT type.

7.2 Characteristic TVD Limiter

Characteristic TVD schemes for the Euler equations date back to the 1980s [47],[48]. In the one-dimensional case, their derivation is fairly straightforward, and various *ad hoc* extensions have been used in finite element codes with considerable success [2],[10],[29],[39],[40]. The ongoing quest for a genuinely multidimensional generalization is complicated by the fact that the underlying characteristic decomposition is no longer unique. In multidimensions, the waves can travel in an infinite number of directions, so it is unclear how to transform the linearized hyperbolic system into a set of decoupled convection equations, for which robust numerical techniques are already available. Hence, one needs not only a fully multidimensional transport algorithm but also a multidimensional wave decomposition for the system at hand [41]. A few promising wave models have been proposed for the design of upwind-biased schemes based on the *fluctuation splitting* approach [3] (also referred to as *residual distribution* [7]) but their complexity is still too high.

The freedom of choosing the direction \mathbf{e} in the factorization (9) arbitrarily turns out to be a blessing and a curse at the same time. On the one hand, it provides the necessary flexibility for the development of numerical schemes. On the other hand, the definition of characteristic variables is ambiguous in multidimensions and has a strong influence on the simulation results. Recall that we constructed the artificial viscosity (35) via a diagonalization of the cumulative Roe matrix A_{ij} which was evaluated in the direction of the normalized coefficient vector \mathbf{e}_{ij} defined in (34). As shown by Lyra et al. [29], this approach is also suitable for the design of characteristic FEM-TVD schemes based on unidirectional slope limiting. On the other hand, the use of a single direction $\mathbf{e} = \mathbf{e}_{ij}$ for the definition of local characteristic variables seems to be inappropriate for *node-oriented* flux limiters which operate as follows:

1. Gather upstream/downstream edge contributions to individual nodes.
2. Compute the nodal correction factors on the basis of this information.
3. Check the sign of the antidiffusive fluxes and limit them edge-by-edge.

Obviously, the data to be collected in at the first step should correspond to the same set of characteristic variables, whereas the transformation matrices R_{ij} and R_{ij}^{-1} for the cumulative Roe matrix depend not only on the averaged solution values but also on the coefficient vector \mathbf{e}_{ij} for the edge at hand. This has led us to employ the dimensional splitting (37) and limit the characteristic variables defined by the eigenvectors of \hat{A}_{ij}^d which are independent of the mesh. Of course, this provisional wave model is not optimal, so the experienced reader is encouraged to experiment with other options.

Let us construct a characteristic TVD limiter following the strategy proposed by Yee et al. [48] and apply it to the vector of fluxes in each coordinate direction so as to impose the LED constraint on the characteristic variables. Both explicit and implicit time discretizations are feasible. As before, all algebraic modifications to be performed affect just the matrix assembly routine. The limiting process consists of the following algorithmic steps:

1. In a loop over edges, generate the right/left eigenvectors of the unidirectional Roe matrix \hat{A}_{ij}^d ($d = 1, 2, 3$) which correspond to the eigenvalues

$$\lambda_1^d = \hat{v}_{ij}^d - \hat{c}_{ij}, \quad \lambda_2^d = \lambda_3^d = \lambda_4^d = \hat{v}_{ij}^d, \quad \lambda_5^d = \hat{v}_{ij}^d + \hat{c}_{ij}. \quad (62)$$

Analytical expressions for the eigenvector matrices can be found in [17].

2. Decompose the difference between the nodal values U_j and U_i into its characteristic components via a transformation analogous to (61)

$$\Delta w_{ij} = [R_{ij}^d]^{-1}(U_j - U_i), \quad (63)$$

where the rows of $[R_{ij}^d]^{-1}$ represent the left eigenvectors of \hat{A}_{ij}^d .

3. Update the sums P_i^\pm and Q_i^\pm of downstream/upstream edge contributions to node i for the characteristic field number k as follows:

For all edges/fluxes A_{ij}^d	For boundary edges B_{ij}^d
Compute $k_{ij}^a = -a_{ij}^d \lambda_k^d$ and $\delta_a^\pm = \max_{\min} \{0, k_{ij}^a \Delta w_{ij}^k\}$ If $k_{ij}^a < 0$, then augment $P_i^\pm := P_i^\pm + \delta_a^\pm, \quad Q_j^\pm := Q_j^\pm + \delta_a^\pm$ If $k_{ij}^a > 0$, then augment $P_j^\pm := P_j^\pm + \delta_a^\pm, \quad Q_i^\pm := Q_i^\pm + \delta_a^\pm$	Compute $k_{ij}^b = -b_{ij}^d \lambda_k^d$ and $\delta_b^\pm = \max_{\min} \{0, k_{ij}^b \Delta w_{ij}^k\}$ If $k_{ij}^b < 0$, then augment $P_i^\pm := P_i^\pm + \delta_b^\pm, \quad P_j^\mp := P_j^\mp - \delta_b^\pm$ If $k_{ij}^b > 0$, then augment $Q_i^\pm := Q_i^\pm + \delta_b^\pm, \quad Q_j^\mp := Q_j^\mp - \delta_b^\pm$

4. In a loop over nodes, calculate the nodal correction factors for all local characteristic variables in each space direction

$$R_i^\pm = \Phi(Q_i^\pm / P_i^\pm). \quad (64)$$

5. Determine the number of the ‘upwind node’ *individually* for each scalar wave propagating at the speed λ_k^d relative to the edge ij

$$I = \begin{cases} i & \text{if } k_{ij}^a \leq 0, \\ j & \text{if } k_{ij}^a > 0, \end{cases} \quad (65)$$

where the coefficient k_{ij}^a is the one defined in step 3 (see the left box).

6. Limit the transformed data jumps Δw_{ij} in accordance with the adopted edge orientation for the characteristic variable at hand

$$\Delta \widehat{w}_{ij}^k = \begin{cases} R_I^+ \Delta w_{ij}^k & \text{if } \delta_a^k \geq 0, \\ R_I^- \Delta w_{ij}^k & \text{if } \delta_a^k < 0, \end{cases} \quad \delta_a^k = k_{ij}^a \Delta w_{ij}^k. \quad (66)$$

Note that the value of δ_a^k is invariant to the order of nodes due to the asymmetry of the parameters $k_{ji}^a = -k_{ij}^a$ and $w_{ji}^k = -w_{ij}^k$.

7. Add the limited antidiffusive correction to the raw diffusive flux $|\Lambda_{ij}^d| \Delta w_{ij}$ and transform the result back to the conservative variables

$$F_{ij}^d = |a_{ij}^d| R_{ij}^d |\Lambda_{ij}^d| (\Delta w_{ij} - \Delta \widehat{w}_{ij}). \quad (67)$$

8. Insert the net (anti-)diffusive flux F_{ij}^d into the right-hand side and/or the defect vector for the modified linear system by setting

$$(K^*U)_i := (K^*U)_i + F_{ij}^d, \quad (K^*U)_j := (K^*U)_j - F_{ij}^d \quad (68)$$

for K^*U initialized by KU for the original Galerkin scheme (11).

The practical implementation of the algorithm may certainly differ from that outlined above and be based on a more involved wave model or used in conjunction with a fractional-step approach such as the ADI method [48].

A few remarks are in order. In the case of scalar convection, the final antidiffusion coefficient for the edge \overrightarrow{ij} was given by $a_{ij} = \min\{R_i^\pm d_{ij}, l_{ji}\}$. This extra check is no longer necessary since

$$k_{ji}^a = -k_{ij}^a \quad \Rightarrow \quad l_{ji}^a := k_{ji}^a + d_{ij}^a = 2d_{ij}^a,$$

where $d_{ij}^a := |a_{ij}^d \lambda_k^d| = |k_{ij}^a|$ is the artificial diffusion coefficient for wave k . The LED property for row j follows from the fact that

$$k_{ji}^* := l_{ji}^a - R_i^\pm d_{ij}^a = k_{ji}^a + (1 - R_i^\pm) d_{ij}^a = (2 - R_i^\pm) d_{ij}^a \geq 0$$

as long as $R_i^\pm \leq 2$, which is the case for all standard TVD limiters. Furthermore, we found that the symmetric contributions $\delta_b^k = k_{ij}^b \Delta w_{ij}^k$ of boundary edges can be omitted at steps 5-7 (this amounts to a selective ‘prelimiting’ of the antidiffusive fluxes) but must be taken into account at step 3.

7.3 Synchronized FCT Limiter

In the FCT community, flux limiting has traditionally been performed on the conservative or primitive variables using a scalar correction factor α_{ij} for the vector F_{ij} of raw antidiffusive fluxes. As already indicated above, this kind of synchronization is mandatory unless the variables under consideration are fully decoupled. Otherwise, spurious ripples may emerge as a consequence of relative phase errors. Löhner *et al.* [24],[25] mentioned the following approaches to the design of a synchronized FCT limiter for the Euler equations:

- evaluation of the correction factors for a single ‘indicator variable’;
- using the minimum of those obtained for a certain group of variables;
- conservative flux limiting based on local variable transformations.

A general algorithm for constraining an arbitrary set of target quantities is outlined in the recent monograph by Löhner [25]. The following realization of his methodology can be used to carry out algebraic flux correction in terms of any nonconservative variables including the characteristic ones:

1. Initialize the auxiliary arrays by $P_i^\pm \equiv 0$, $Q_i^\pm \equiv 0$, $R_i^\pm \equiv 1$.
2. In a loop over edges, convert the raw antidiffusive fluxes as well as the differences between the nodal values of \tilde{U} to the new variables

$$\hat{F}_{ij} = T_{ij}F_{ij}, \quad \Delta w_{ij} = T_{ij}(\tilde{U}_j - \tilde{U}_i), \quad (69)$$

where the transformation matrix T_{ij} depends on the choice of variables and is evaluated using an appropriate average of \tilde{U}_i and \tilde{U}_j [49].

3. Perform the (optional) prelimiting of the transformed fluxes

$$\hat{F}_{ij}^k := 0 \quad \text{if } \hat{F}_{ij}^k \Delta w_{ij}^k \geq 0. \quad (70)$$

4. Augment the sums of positive/negative fluxes for each target variable

$$P_i^\pm := P_i^\pm + \max_{\min} \{0, \hat{F}_{ij}^k\}, \quad P_j^\pm := P_j^\pm + \max_{\min} \{0, -\hat{F}_{ij}^k\}. \quad (71)$$

5. Update the maximum/minimum admissible increments as follows

$$Q_i^\pm := \max_{\min} \{Q_i^\pm, \Delta w_{ij}^k\}, \quad Q_j^\pm := \max_{\min} \{Q_j^\pm, -\Delta w_{ij}^k\}. \quad (72)$$

6. In a loop over nodes, calculate the nodal correction factors

$$R_i^\pm := \min\{1, m_i Q_i^\pm / P_i^\pm\} \quad \text{if } |P_i^\pm| > 0. \quad (73)$$

7. In a loop over edges, inspect the sign of the antidiffusive fluxes and compute the final correction factors for the selected control variables

$$\alpha_{ij}^k = \begin{cases} \min\{R_i^+, R_j^-\} & \text{if } \hat{F}_{ij}^k \geq 0, \\ \min\{R_j^+, R_i^-\} & \text{if } \hat{F}_{ij}^k < 0. \end{cases} \quad (74)$$

8. Apply the synchronized weight $\hat{\alpha}_{ij} = \min_k \alpha_{ij}^k$, directly to F_{ij} or limit \hat{F}_{ij} and transform it back to the conservative variables

$$F_{ij}^* = T_{ij}^{-1} \hat{F}_{ij}^*, \quad \text{where} \quad \hat{F}_{ij}^* = \hat{\alpha}_{ij} \hat{F}_{ij}. \quad (75)$$

9. Check the result and set $F_{ij}^* := 0$ if nonphysical behavior is detected [49].
 10. Insert the limited antidiffusive flux F_{ij}^* into the right-hand side B and/or into the defect vector for the fully discrete scheme

$$B_i := B_i + F_{ij}^*, \quad B_j := B_j - F_{ij}^*. \quad (76)$$

At present, there is still a large degree of empiricism in the construction of synchronized FCT limiters, and their performance is strongly problem-dependent. Reportedly, the use of synchronization on the density and energy is to be recommended for highly dynamic flows, while the density and pressure should be controlled for steady-state applications [24],[25]. At the same time, the minimum of correction factors for all conservative variables is too restrictive. In particular, insignificant fluctuations of the crosswind velocity may cause a complete cancellation of the antidiffusive flux. Therefore, an equal treatment of all velocity components yields poor results, especially if the flow takes place in a predominant direction. Interestingly enough, the iterative limiting strategy introduced in the previous chapter is much less sensitive to the choice of indicator variables and to the synchronization procedure, since the rejected antidiffusion can be reused at subsequent flux/defect correction steps.

Zalesak was the first to explore the potential of characteristic variables in the realm of FCT methods [49]. His characteristic version of the Boris-Book limiter is remarkably robust (due to a ‘fail-safe’ control mechanism) and yields impressive results for the one-dimensional Euler equations. For an in-depth presentation of the underlying design principles, the reader is referred to [50]. Unfortunately, no fully multidimensional extension is available to date. Since Zalesak’s limiter is node-oriented, the use of operator splitting is currently unavoidable for the reasons explained above. In fact, the situation is even more difficult than in the case of TVD algorithms since the raw antidiffusive fluxes are evaluated in terms of U^n and/or U^{n+1} , whereas the transformation matrices for the limiting step depend on the intermediate solution \tilde{U} .

The time-dependent nature of FCT suggests a fractional-step approach to the limiting of characteristic variables in the framework of a ‘Cartesian’ wave model. A very simple way to decouple the components of \mathbf{F} in system (6) is to invoke the first-order accurate Marchuk-Yanenko splitting

$$\frac{\partial U_d}{\partial t} + \frac{\partial F^d}{\partial x_d} = 0 \quad \text{in } (t^n, t^{n+1}) \quad d = 1, 2, 3. \quad (77)$$

The final solution to each subproblem serves as initial data for the next one

$$U_d^n = U_{d-1}^{n+1}, \quad U_0^{n+1} = U^n, \quad U^{n+1} = U_3^{n+1}.$$

Due to the splitting of spatial derivatives and of the associated Jacobians, the transformation matrices $T_{ij} = [R_{ij}^d]^{-1}$ provide a suitable definition of local characteristic variables for Zalesak's limiter. We remark that no synchronization of the nodal correction factors is necessary in this case. Furthermore, the employed operator splitting does not rule out the use of unstructured meshes, since the limiting procedure remains fully multidimensional. However, the solution of three subproblems per time step and costly variable transformations impose a heavy burden in terms of CPU time. Therefore, the advantages of flux limiting on characteristic variables are rather dubious so far.

8 Iterative Solution Strategies

The nonlinear algebraic systems resulting from an implicit FEM-FCT or FEM-TVD discretization of the Euler equations exhibit the same structure as their scalar counterparts. This enables us to design iterative solvers building on the defect correction technique described previously. Let the successive approximations to the desired solution U^{n+1} be updated as follows

$$U^{(m+1)} = U^{(m)} + [A(U^{(m)})]^{-1}R^{(m)}, \quad m = 0, 1, 2, \dots \quad (78)$$

where $A(U^{(m)})$ is a suitable preconditioner. A good candidate is the matrix for system (28) discretized in time by the standard θ -scheme

$$A(U^{(m)}) = M_L + \theta \Delta t L(U^{(m)})U^{(m)}, \quad 0 < \theta \leq 1. \quad (79)$$

It is worthwhile to construct $A(U^{(m)})$ using scalar dissipation (38) no matter what form of D_{ij} is adopted for the artificial viscosity built into $R^{(m)}$.

The constant right-hand side for the low-order method is given by

$$B^n = M_L + (1 - \theta) \Delta t L(U^n)U^n. \quad (80)$$

It is augmented by limited antidiffusion and incorporated into the defect

$$R^{(m)} = B^n + F(U^{(m)}, U^n) - A(U^{(m)})U^{(m)} \quad (81)$$

which vanishes for the exact solution of the discrete problem. It is essential to guarantee that $R^{(m)}$ does approach zero in an actual simulation. This is especially important in the case of implicit TVD-like methods which satisfy the imposed criteria only upon convergence and may produce nonphysical solution values if the number of outer iterations is insufficient.

Since the inversion of the global matrix $A(U^{(m)})$ is prohibitively expensive, the following two-step implementation of (78) is to be employed:

$$A(U^{(m)})\Delta U^{(m+1)} = R^{(m)}, \quad m = 0, 1, 2, \dots \quad (82)$$

$$U^{(m+1)} = U^{(m)} + \Delta U^{(m+1)}, \quad U^{(0)} = U^n. \quad (83)$$

The linear system to be solved at the first step can be written as

$$\begin{bmatrix} A_{11}^{(m)} & A_{12}^{(m)} & A_{13}^{(m)} & A_{14}^{(m)} & A_{15}^{(m)} \\ A_{21}^{(m)} & A_{22}^{(m)} & A_{23}^{(m)} & A_{24}^{(m)} & A_{25}^{(m)} \\ A_{31}^{(m)} & A_{32}^{(m)} & A_{33}^{(m)} & A_{34}^{(m)} & A_{35}^{(m)} \\ A_{41}^{(m)} & A_{42}^{(m)} & A_{43}^{(m)} & A_{44}^{(m)} & A_{45}^{(m)} \\ A_{51}^{(m)} & A_{52}^{(m)} & A_{53}^{(m)} & A_{54}^{(m)} & A_{55}^{(m)} \end{bmatrix} \begin{bmatrix} \Delta U_1^{(m+1)} \\ \Delta U_2^{(m+1)} \\ \Delta U_3^{(m+1)} \\ \Delta U_4^{(m+1)} \\ \Delta U_5^{(m+1)} \end{bmatrix} = \begin{bmatrix} R_1^{(m)} \\ R_2^{(m)} \\ R_3^{(m)} \\ R_4^{(m)} \\ R_5^{(m)} \end{bmatrix}. \quad (84)$$

Here the row index refers to the vector of unknowns for one conservative variable (density, momentum component or energy) which interacts with the others via the off-diagonal blocks of the preconditioner $A(U^{(m)})$.

Simultaneous solution of these equations gobbles up a lot of resources in terms of CPU time and memory requirements. Instead, the constituents of the discretized Euler equations can be decoupled and treated separately or, better yet, in parallel. This can be accomplished by switching to a block-diagonal preconditioner. Let us consider the *block-Jacobi method* [5] obtained from (84) by setting $A_{kl}^{(m)} := \delta_{kl} A_{kl}^{(m)}$, where δ_{kl} is the Kronecker delta. In this case, the global update (83) reduces to a sequence of scalar subproblems

$$A_{kk}^{(m)} \Delta U_k^{(m+1)} = R_k^{(m)}, \quad k = 1, \dots, 5. \quad (85)$$

$$U_k^{(m+1)} = U_k^{(m)} + \Delta U_k^{(m+1)}, \quad U_k^{(0)} = U_k^n, \quad (86)$$

whereby a (small) number of inner iterations is performed to compute the solution increment $\Delta U_k^{(m+1)}$ for each variable. An implicit or explicit under-relaxation [12] may be applied at the first and second step, respectively.

This segregated algorithm is very easy to implement and its performance is satisfactory as long as the time step remains relatively small. However, the nonlinear convergence rates deteriorate and the simulation may even fail completely as the time step increases. In particular, a time-marching technique combined with (85)-(86) may experience severe convergence problems when applied to steady transonic flows. This can be attributed to the fact that the *characteristic condition number* (the ratio of the largest to the smallest wave speed) is very high for Mach numbers approaching zero or unity. As a result, the Euler equations become very stiff and call for the use of local preconditioning [6] and/or a strongly coupled solution strategy.

A much more robust iterative solver can be devised on the basis of a Krylov-subspace or multigrid method for the **global** system (84). To avoid the storage of all matrix blocks and ‘invert’ one block at a time, it is worthwhile to use a smoother/preconditioner of block-Jacobi or block-Gauß-Seidel type. For an abstract system $AX = B$ having the same structure as (84), the latter updates the components of the vector x row-by-row as follows

$$A_{kk} X_k^{(m+1)} = B_k - \sum_{l < k} A_{kl} X_l^{(m+1)} - \sum_{l > k} A_{kl} X_l^{(m)}. \quad (87)$$

This splitting differs from the block-Jacobi method in that the latest solution values are used for the evaluation of the right-hand side. In either case, only the diagonal blocks of the global matrix need to be ‘inverted’ approximately in the inner iteration process. However, the block-Gauß-Seidel method is preferable as it yields better convergence rates at no additional cost [5].

A preliminary evaluation of BiCGSTAB and multigrid solvers equipped with this (weakly) coupled relaxation technique indicates that they are far superior to the algorithm (85)-(86) when it comes to simulation of (quasi-) stationary flows [33],[34]. On the other hand, the segregated approach or even a fully explicit time-stepping is appropriate for transient applications since the time step must remain rather small for accuracy reasons. Due to the tradeoff between robustness and efficiency, the choice of the ‘best’ solution strategy is highly problem-dependent. Therefore, a general-purpose CFD code should consist of several modules optimized for different flow regimes [44].

9 Implementation of Boundary Conditions

The numerical treatment of boundary conditions for the Euler equations is an issue of utmost importance and its influence on the simulation results should not be underestimated. A comprehensive coverage of this topic is available in a number of textbooks [10],[13],[17],[45]. This section is intended to give an insight into some theoretical and numerical aspects pertinent to the implementation of *characteristic boundary conditions* in a finite element code.

For a hyperbolic system that consists of N_v partial differential equations, $N_p \leq N_v$ physical boundary conditions (PBC) plus $N_n = N_v - N_p$ numerical boundary conditions (NBC) are to be prescribed. The former must secure the existence and uniqueness of the exact solution, while the latter are supposed to ensure that various perturbations generated in the interior of the computational domain leave it without being reflected at the boundaries [17]. Hence, a proper combination of PBC and NBC must be imposed by means of some extra matrix/vector manipulations to be described below.

9.1 Physical Boundary Conditions

The number of physical boundary conditions N_p depends on the pattern of wave propagation. For simplicity, consider a local coordinate system such that the unit outward normal is aligned with the x -axis. The characteristic variables associated with the direction $\mathbf{n} = (\pm 1, 0, 0)$ are given by [45]

$$W = \left[v_n - \frac{2c}{\gamma - 1}, s, v_2, v_3, v_n + \frac{2c}{\gamma - 1} \right]^T, \quad (88)$$

where $v_n = \mathbf{n} \cdot \mathbf{v}$ stands for the normal velocity, c is the speed of sound and $s = c_v \log(p/\rho^\gamma)$ is the entropy defined up to an additive constant.

The canonical transformation (56) of the linearized Euler equations yields a sequence of decoupled convection equations for the five components of the vector W which are called the *Riemann invariants*. These scalar quantities are transported along the characteristic curves at the respective speeds

$$\lambda_1 = v_n - c, \quad \lambda_2 = \lambda_3 = \lambda_4 = v_n, \quad \lambda_5 = v_n + c \quad (89)$$

which represent the eigenvalues of $A_n = \mathbf{n} \cdot \mathbf{A}$. For a pure convection equation, boundary conditions are required only at the inlet. Hence, N_p equals the number of negative eigenvalues which correspond to the incoming characteristics. It can readily be seen that the direction of wave propagation depends not only on the sign of the velocity v_n but also on the local Mach number $M = |v_n|/c$. Therefore, several different situations need to be considered:

Supersonic open boundary $|v_n| > c$

At the inlet $v_n < 0$ and all eigenvalues are negative; the values of all characteristic variables must be specified. At the outlet $v_n > 0$ and all eigenvalues are positive; no physical boundary conditions may be imposed.

Subsonic open boundary $|v_n| < c$

At the inlet $v_n < 0$ so that λ_5 is positive while the other eigenvalues are negative; all Riemann invariants except w_5 must be specified. At the outlet $v_n > 0$ so that the first eigenvalue is negative, whereas the other ones are positive. Hence, just the boundary value of w_1 is to be prescribed.

Solid wall boundary $v_n = 0$

At the solid wall, the normal velocity component must be constrained to zero. Under this *no-penetration* or *free slip* condition, there is no convective flux through the boundary. All eigenvalues except $\lambda_1 = -c$ are nonnegative so that only w_1 must be specified. This Riemann invariant corresponds to an acoustic wave entering the domain at the speed of sound.

In practice, physical boundary conditions are typically imposed on the primitive variables or given in terms of input data like the total enthalpy, entropy, temperature or inclination angle. The problem remains well-posed as long as the number of boundary conditions is correct.

9.2 Numerical Boundary Conditions

The need for numerical boundary conditions stems from the fact that the actual problem to be solved is formulated in terms of the conservative variables rather than Riemann invariants. Therefore, it is impossible to impose the Dirichlet boundary conditions in the usual way. It is common practice to recover the boundary values by changing to the characteristic variables, evaluating the incoming Riemann invariants from the physical boundary conditions and extrapolating the outgoing ones from the interior of the domain [17],[45],[46]. The inverse transformation yields the desired values of the conservative variables which can be used to compute the numerical fluxes for a

finite volume method or treated as Dirichlet boundary conditions for a finite difference scheme. However, the extrapolation of Riemann invariants lacks a rigorous justification and is expensive to perform on unstructured meshes. Since we are not aware of any publication regarding the implementation of characteristic boundary conditions for an implicit finite element method, we deem it appropriate to go into some technical details.

In order to *predict* the solution values at boundary nodes, we modify the preconditioner $A(U^{(m)}) = \{a_{ij}^{kl}\}$ by picking out the corresponding rows and setting their off-diagonal entries equal to zero. That is, if node i belongs to the boundary, then $a_{ij}^{kl} := 0, \forall j \neq i, \forall l \neq k$. This enables us to update the components of the vector U_i explicitly prior to solving the linear system (84). To this end, we divide the components of the nodal defect vector $R_i^{(m)}$ by the diagonal entries of the preconditioner and increment the old iterate

$$U_i^* = U_i^{(m)} + A_{ii}^{-1} R_i^{(m)}, \quad A_{ii} = \text{diag} \{a_{ii}^{kk}\}. \quad (90)$$

Next, the provisional solution U_i^* is transformed to the local characteristic variables w_i (see appendix). Recall that the number of physical boundary conditions is equal to the number of incoming characteristics. Hence, those components of the vector w_i^* which correspond to negative eigenvalues of the Jacobian $A_n = \mathbf{n} \cdot \mathbf{A}$ can be replaced by their exact values evaluated from the given input data. The outgoing Riemann invariants, which are associated with numerical boundary conditions, remain unchanged. Finally, we convert the corrected vector w_i^{**} back to the conservative variables, assign the result to U_i^{**} and nullify all entries of the defect vector $R_i^{(m)}$ so that

$$\Delta U_i^{(m+1)} = A_{ii}^{-1} R_i^{(m)} = 0 \quad \Rightarrow \quad U_i^{(m+1)} = U_i^{**}. \quad (91)$$

The flow chart of variable transformations and manipulations to be performed for the boundary nodes before the solution of system (84) is displayed in Fig. 2. In essence, the *corrected* values U_i^{**} represent standard Dirichlet boundary conditions for the end-of-step solution $U_i^{(m+1)}$. Note that there is no need for an *ad hoc* extrapolation of data from the interior. Further implementation details are presented in the appendix to this chapter.

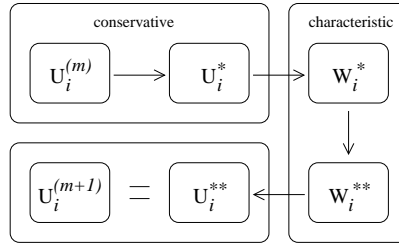


Fig. 2. Variable transformations for boundary nodes.

10 Numerical Examples

Below we apply our algebraic flux correction schemes to a few academic test problems that encompass both stationary and time-dependent inviscid flows at different Mach numbers. The computational results presented in this section provide a preliminary validation of the new algorithms and illustrate their characteristic features for an admittedly limited range of applications. At the current stage of development, there are still many aspects that call for further investigation. In particular, it is rather difficult to predict how the flux limiters and iterative solvers would perform on highly unstructured and possibly anisotropic meshes for industrial flows in complex geometries. Some fine-tuning may turn out to be necessary but the numerical examples that follow indicate that (semi-) implicit FEM-FCT and FEM-TVD methods constitute a promising approach to compressible flow simulations.

10.1 Shock Tube Problem

Our first example is the classical shock tube problem which exhibits an interesting flow structure and is widely used in CFD for testing purposes [20],[42]. Its physical prototype is a closed tube filled with inviscid gas which is initially at rest and separated by a membrane into the regions of high and low pressure. The initial conditions for the Riemann problem are given by

$$\begin{bmatrix} \rho_L \\ \mathbf{v}_L \\ p_L \end{bmatrix} = \begin{bmatrix} 1.0 \\ 0.0 \\ 1.0 \end{bmatrix} \quad \text{for } x \in [0, 0.5], \quad \begin{bmatrix} \rho_R \\ \mathbf{v}_R \\ p_R \end{bmatrix} = \begin{bmatrix} 0.125 \\ 0.0 \\ 0.1 \end{bmatrix} \quad \text{for } x \in (0.5, 1].$$

The rupture of the membrane at $t = 0$ triggers a *normal shock wave* which moves to the right with velocity satisfying the Rankine-Hugoniot conditions. All of the primitive variables are discontinuous across the shock which is followed by a *contact discontinuity*. The latter represents a moving interface between the regions of different densities but constant velocity and pressure. At the same time, a *rarefaction wave* propagates in the opposite direction providing a smooth transition to the original values of the state variables in the left part of the domain. Hence, the flow pattern in the shock tube is characterized by three waves travelling at different speeds [20].

The snapshots shown in Fig. 3 correspond to the time instant $t = 0.231$. The one-dimensional Euler equations were discretized using 100 linear finite elements and the Crank-Nicolson time-stepping with $\Delta t = 10^{-3}$. The analytical solution is designated by the dashed lines in each plot. It was obtained using the technique described by Anderson [1]. The numerical solutions of low order (upper diagrams) are seen to be nonoscillatory and qualitatively correct but their accuracy leaves a lot to be desired. As a matter of fact, Roe's approximate Riemann solver (53) performs slightly better than scalar dissipation (55) but also results in a strong smearing of the moving fronts.

The removal of excessive artificial diffusion in the course of flux correction yields a dramatic improvement of the overall accuracy (see Fig. 3, bottom). The employed FEM-TVD scheme was equipped with the superbee limiter, while the companion FEM-FCT method was implemented following the algorithm proposed by Zalesak [49]. In either case, the low-order preconditioner was based on scalar dissipation and flux limiting was carried out in terms of the characteristic variables. The use of the conservative ones is also feasible provided that the correction factors are properly synchronized. Simulation results for the two-dimensional shock tube problem solved by FCT with synchronization on the density and energy are reported in [19]. If the minimum of correction factors for all conservative variables is adopted, then the basic limiter yields overly diffusive results, whereas the iterative version performs remarkably well [33]. Indeed, a ‘cautious’ limiting strategy does not degrade the accuracy as long as the rejected antidiffusion can be recycled.

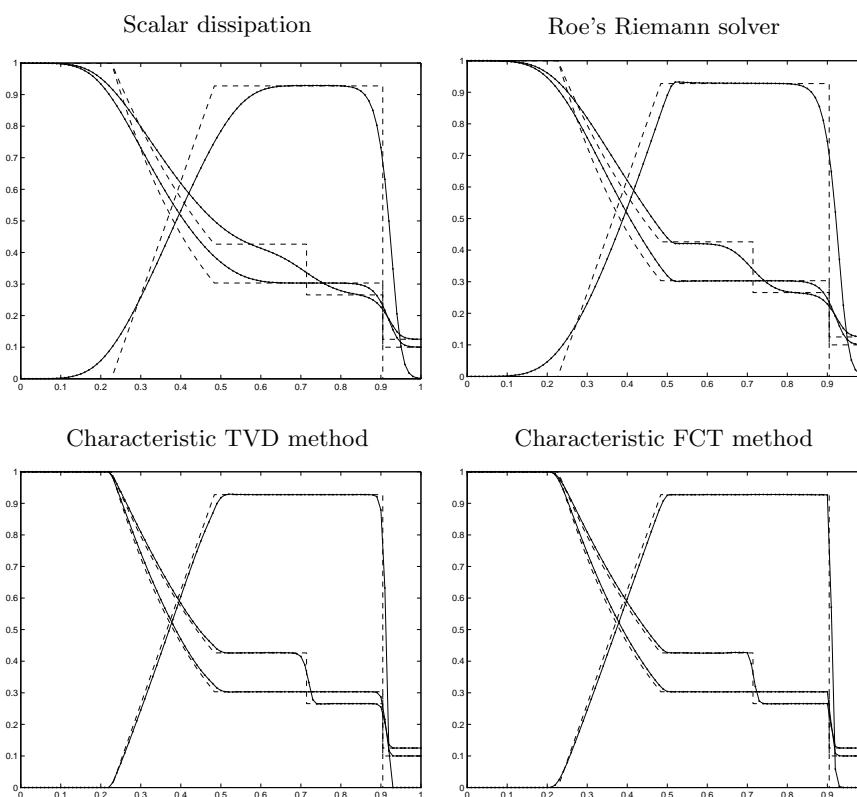


Fig. 3. Shock tube problem: numerical solutions at $t = 0.231$.

10.2 Radially Symmetric Riemann Problem

Another transient test problem proposed by LeVeque [21] represents a counterpart of the shock tube problem in polar coordinates. Before an impulsive start, the square domain $\Omega = (-0.5, 0.5) \times (-0.5, 0.5)$ is separated by an imaginary membrane into two subregions: $\Omega_L = \{(x, y) \in \Omega \mid r = \sqrt{x^2 + y^2} < 0.13\}$ and $\Omega_R = \Omega \setminus \Omega_L$. The gas is initially at rest, whereby its pressure and density are higher within the circle Ω_L than outside of it:

$$\begin{bmatrix} \rho_L \\ \mathbf{v}_L \\ p_L \end{bmatrix} = \begin{bmatrix} 2.0 \\ 0.0 \\ 15.0 \end{bmatrix} \text{ in } \Omega_L, \quad \begin{bmatrix} \rho_R \\ \mathbf{v}_R \\ p_R \end{bmatrix} = \begin{bmatrix} 1.0 \\ 0.0 \\ 1.0 \end{bmatrix} \text{ in } \Omega_R.$$

The abrupt removal of the membrane at $t = 0$ gives rise to a radially expanding shock wave which is induced by the pressure difference. The challenge of this benchmark is to capture the moving shock and to make sure that the numerical solution of the Riemann problem remains radially symmetric.

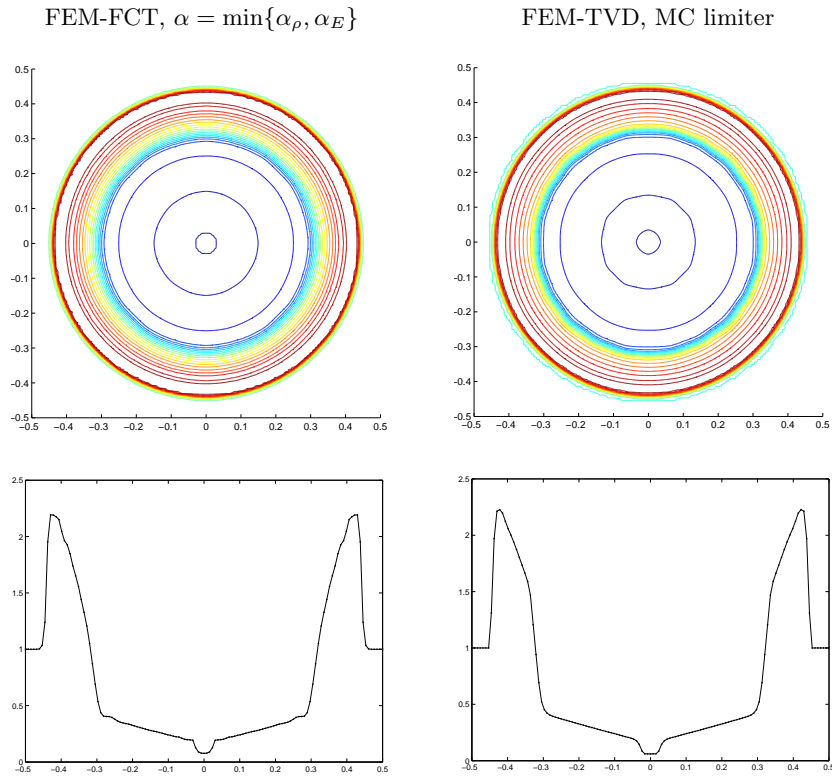


Fig. 4. Radially symmetric density distribution at $t = 0.13$.

The simulation results depicted in Fig. 4 were obtained on a uniform mesh of 128×128 bilinear elements by the characteristic FEM-TVD method (left) and by the iterative FEM-FCT algorithm with flux synchronization on the density and energy (right). As before, the discretization in time was performed by the Crank-Nicolson scheme and the constant time step $\Delta t = 10^{-3}$ was employed. The contour lines for the density distribution at $t = 0.13$ (upper diagrams) have the form of concentric circles, so that both high-resolution schemes succeed in perserving the radial symmetry of initial data. The lower diagrams show the density profiles along the x -axis. These solutions are in a very good agreement with those computed by the public-domain software package CLAWPACK [23] using the same mesh and time step.

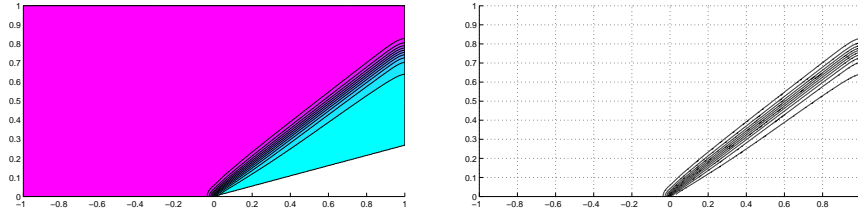
10.3 Compression Corner

To illustrate the potential of the fully implicit backward Euler time-stepping, we consider a steady two-dimensional supersonic flow over a wedge which may imitate e.g. the tip of a projectile. This configuration is sometimes referred to as *compression corner* and constitutes an excellent test problem because it admits an analytical solution in the framework of the oblique shock theory [1]. The originally uniform supersonic flow preserves its freestream characteristics until it reaches the wedge which deflects it upward through an angle θ . Provided that θ is not too large, the change in the flow direction takes place across a shock wave which has the form of a straight line emanating from the tip of the wedge and running oblique to the original flow direction.

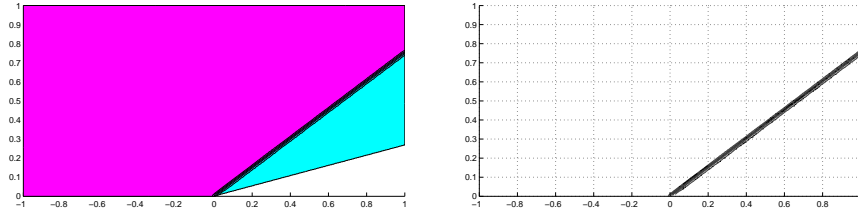
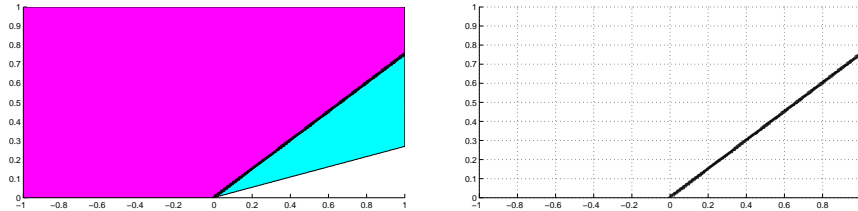
Given the inflow Mach number $M_1 = 2.5$ and the deflection angle $\theta = 15^\circ$, one can determine the downstream Mach number $M_2 = 1.87$ and the shock inclination angle $\beta = 36.94^\circ$ as explained, e.g., in [1]. A detailed description of this test case is available in the *CFD Verification and Validation Database* of the NPARC Alliance [36]. The simulation results presented in Fig. 5 were computed on a boundary-fitted mesh of 128×128 bilinear elements and interpolated onto a Cartesian background mesh for visualization purposes. All these solutions were marched to the steady state by the backward Euler method. Although it is unconditionally stable, the magnitude of admissible time steps depends on the robustness of the iterative solver. In particular, the segregated algorithm (85)-(86) cannot be operated at $\Delta t > 10^{-2}$ unless a strong under-relaxation is performed. By contrast, time steps as large as $\Delta t = 1.0$ can be used in conjunction with the coupled solution strategy [33].

The upper diagrams show the distribution of Mach numbers predicted by the low-order method based on scalar dissipation. The upstream and downstream Mach numbers M_1 and M_2 are correct but the transition between them is anything else but discontinuous. The oblique shock is strongly smeared by numerical diffusion and resembles a ‘rarefaction wave’. At the same time, this low-order solution provides a reasonable initial guess for the FEM-TVD and FEM-FCT algorithms. Let us examine the moderately diffusive MC limiter

Low-order method, scalar dissipation



FEM-TVD scheme, MC limiter

FEM-FCT scheme, $\alpha = \min\{\alpha_\rho, \alpha_E\}$ **Fig. 5.** Compression corner: $M_1 = 2.5$, $\theta = 15^\circ$.

and the iterative Zalesak limiter with the ρ - E synchronization of the correction factors. After a sufficient number of outer defect correction steps, a significant improvement is observed in the resolution of the stationary shock (see Fig. 5). In fact, the results produced by both high-resolution schemes are even superior to the reference solution from the NPARC database [36].

10.4 Prandtl-Meyer Expansion

Our next example deals with a steady supersonic flow being deflected downward rather than upward. In this case, the flow behavior is quite different from the one discussed above for the compression corner. As the gas reaches the kink, it starts spreading and the flow characteristics change smoothly across the so-called *Prandtl-Meyer expansion* wave. The streamlines gradually bend downward and eventually become parallel to the lower wall as they leave the rarefaction fan which separates the regions of uniform flow. All flow properties adjust themselves continuously across the rarefaction wave except for

the midpoint at which the wall geometry changes abruptly. The analytical solution to this problem and further details can be found in [1].

In accordance with the NPARC setup for this benchmark [36], we adopt the freestream Mach number $M_1 = 2.5$ and the deflection angle $\theta = 15^\circ$. The resulting expansion fan is composed from an infinite number of iso-Mach lines lying in the angular sector bounded by the angles $\mu_1 = 23.58^\circ$ (upstream) and $\mu_2 = 18.0^\circ$ (downstream) as measured from the ramp portion of the wall. On exit from the rarefaction wave, the analytical value of the Mach number equals $M_2 = 3.24$. Figure 6 displays the low-order solution (top) versus those produced by the FEM-TVD (middle) and FEM-FCT (bottom) schemes. All parameter settings are the same as in the previous example and so is the relative performance of the methods under consideration.

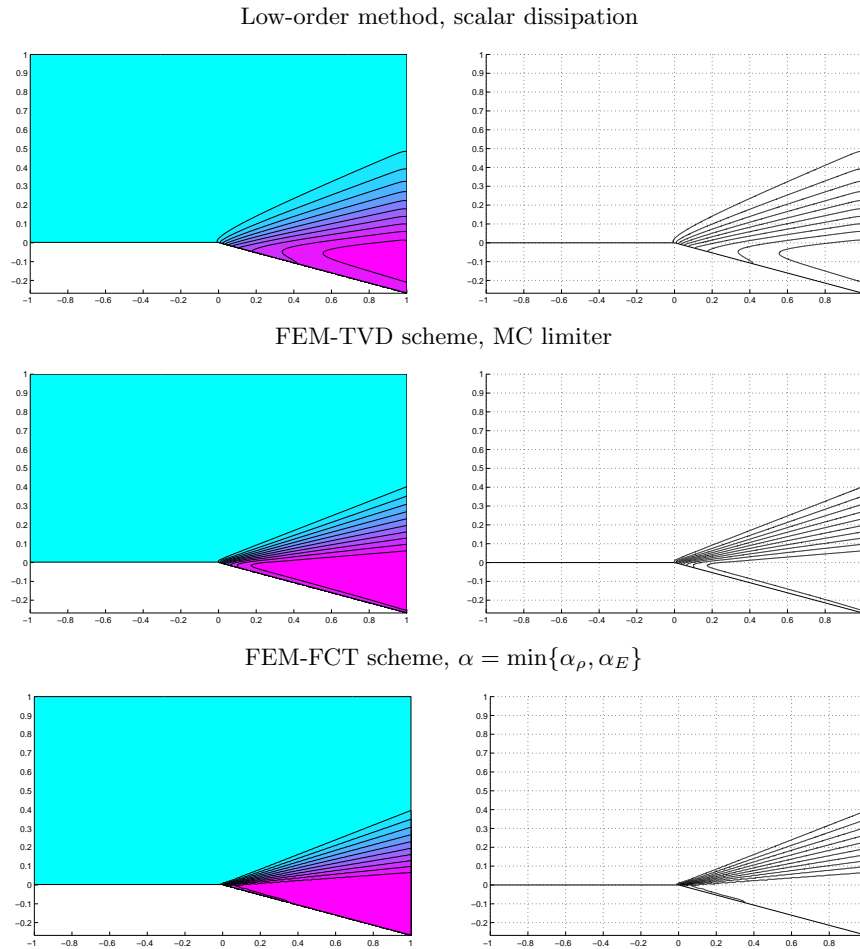


Fig. 6. Prandtl-Meyer expansion: $M_1 = 2.5$, $\theta = 15^\circ$.

10.5 Flow Past a Cylinder

In what follows, we stick to the characteristic FEM-TVD algorithm based on the MC limiter and apply it to a number of flow problems in non-Cartesian geometries so as to demonstrate the flexibility of the finite element approach. The first example is a steady supersonic flow past a circular cylinder placed in the center of a rectangular domain [32]. Specifically, the two-dimensional Euler equations are solved in $\Omega = \{(x, y) \in (-2.5, 2.5) \times (-5, 5) : x^2 + y^2 > 0.25\}$ and the freestream Mach number $M = 3$ is prescribed at the left border.

To generate the hierarchical quad-tree data structures for the geometric multigrid solver, the elements of a manually generated coarse mesh are successively subdivided into four subelements. The body-fitted computational mesh shown in Fig. 7 (left) consists of 15,104 bilinear elements and corresponds to the third refinement level for the underlying coarse grid. Note that a local mesh refinement is performed in the wake of the cylinder. The Mach number distribution (right diagram) gives an insight into the flow structure which

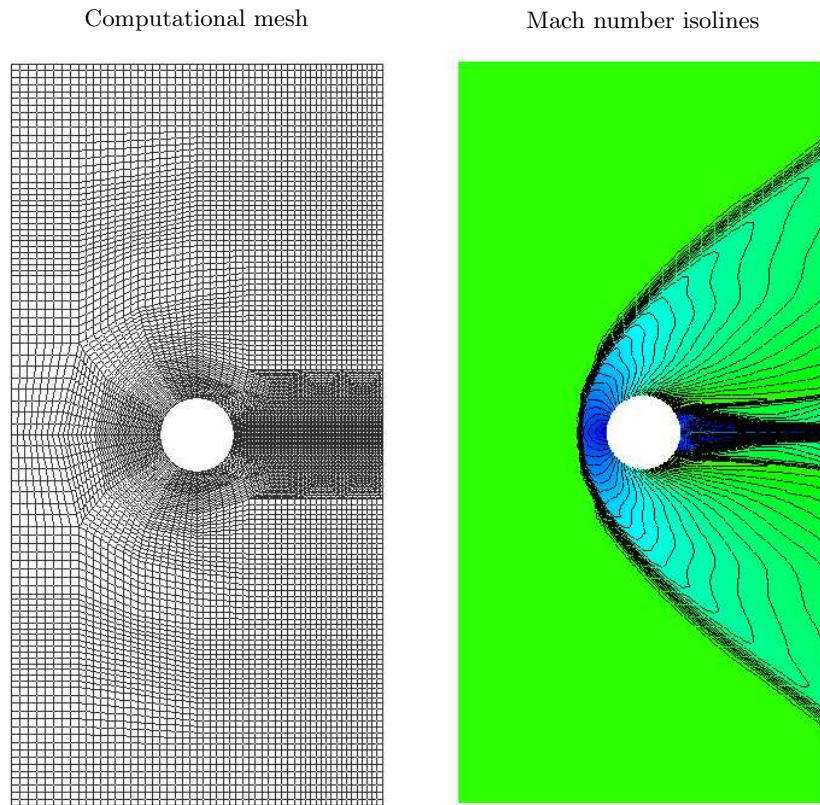


Fig. 7. Steady flow past a cylinder: FEM-TVD scheme, Q_1 elements.

features a strong bow shock as well as stagnation and rarefaction zones. Contrary to the numerical study published in [28],[32] no stability or convergence problems were encountered during this simulation.

On the other hand, the sensitivity of the computational results to the employed mesh was confirmed by our numerical experiments. The unstructured ‘coarse’ triangulation displayed in Fig. 8 contains 2,942 elements (1,561 nodes) which amounts to 47,072 elements (23,896 nodes) after two refinement steps. In this case, the mesh size is fairly uniform, except near the internal boundary. The resulting flow pattern agrees well with that presented in Fig. 7 but some differences can be observed in the critical zone downstream of the cylinder, where the numerical solution is highly mesh-dependent [28]. This discrepancy can be attributed to the fact that separated flows admit nonunique solutions in the framework of an inviscid gas model unless additional empirical hypotheses are employed [4]. Therefore, such applications call for the use of the compressible Navier-Stokes equations rather than the Euler equations.

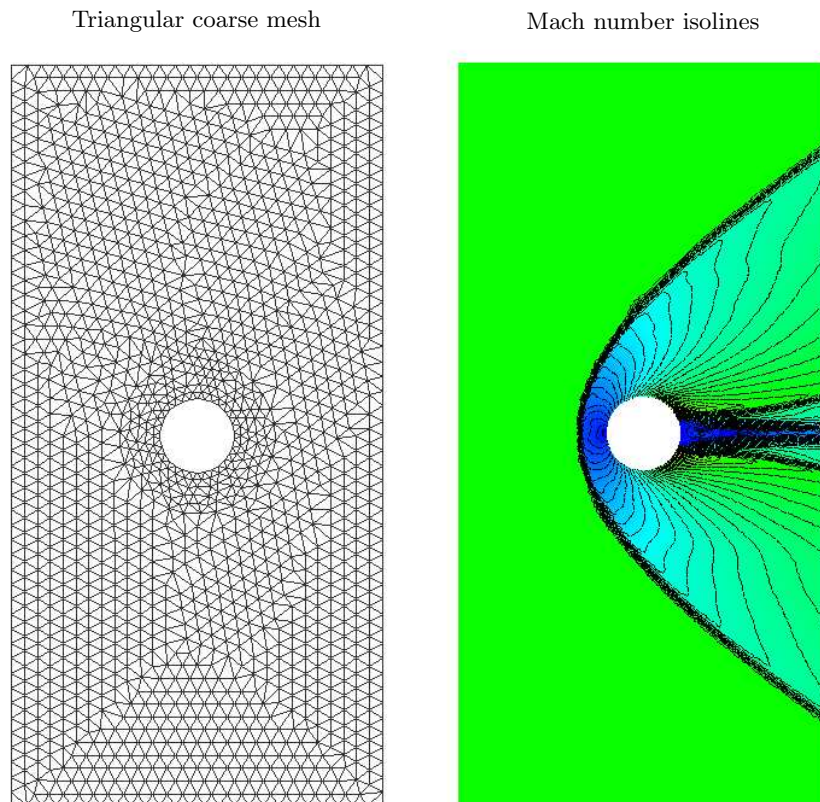


Fig. 8. Steady flow past a cylinder: FEM-TVD scheme, P_1 elements.

10.6 Supersonic Scramjet Inlet

Another interesting benchmark configuration was introduced in [8]. A steady supersonic flow enters a narrowing channel at $M = 3$ and impinges on two sharp-cornered internal obstacles, which gives rise to multiple shock wave reflections. The computational domain Ω is delimited by three piecewise-linear boundary components passing through the following set of points [11]

	<i>upper wall</i>						<i>upper obstacle</i>				
x_i	0.0	0.4	4.9	12.6	14.25	16.9	4.9	8.9	9.4	14.25	12.6
y_i	3.5	3.5	2.9	2.12	1.92	1.7	1.4	0.5	0.5	1.2	1.4

The ‘scramjet inlet’ is symmetric so that the coordinates of the corner points for the lower wall/obstacle are given by $(x_i, -y_i)$. As in the previous example, a top-down approach to mesh generation is adopted. The underlying coarse mesh shown in Fig. 9 consists of 1,033 triangles (616 vertices) and the number of elements quadruples after each global refinement step.

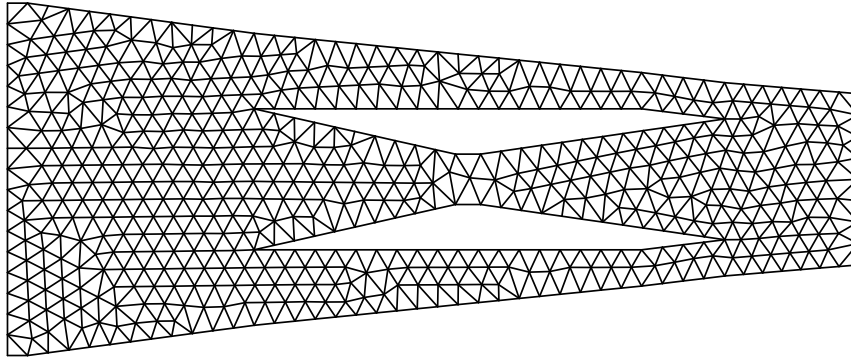


Fig. 9. Scramjet inlet: triangular coarse mesh.

The stationary solutions displayed in Fig. 10 were produced by the fully implicit FEM-TVD method via time marching. The upper diagram depicts the Mach number distribution which was computed using 66,112 linear finite elements (33,859 nodes). Four times as many triangles (133,831 nodes) were employed to obtain the lower diagram. It can be seen that strong shocks are reproduced reasonably well but the interplay of weak ones is difficult to capture even on the finer mesh. Like in many other aerodynamic applications, there is a need for **adaptive** mesh refinement which can be readily realized in a finite element code if appropriate error estimators/indicators are available. Interestingly enough, the presented Mach number contours for the global refinement levels 3 and 4 resemble the simulation results of Díaz et al. [8] on the initial mesh and after one adaptation step, respectively.

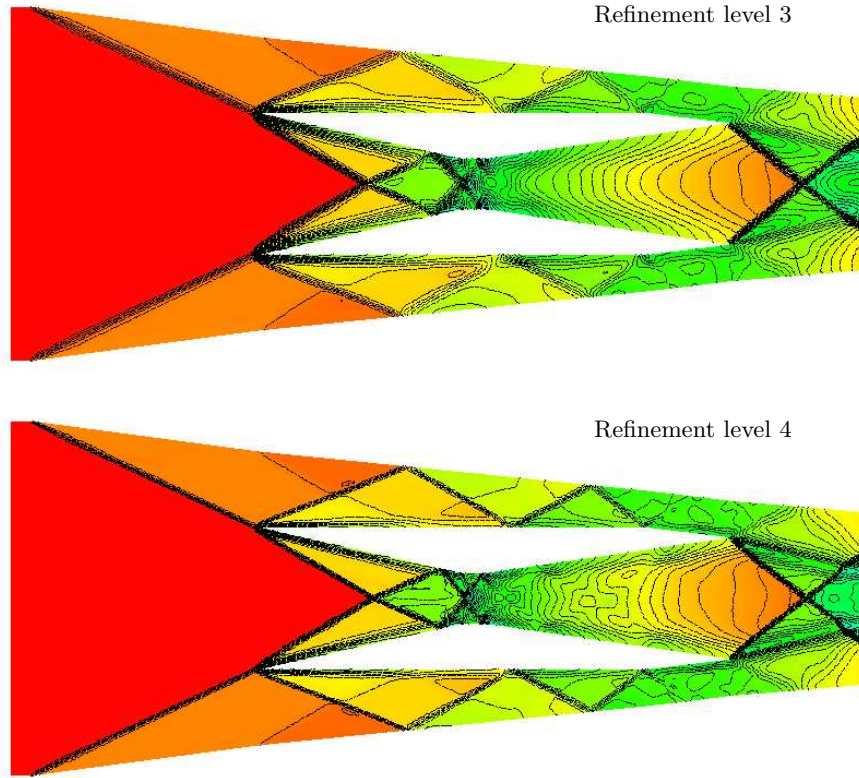


Fig. 10. Scramjet inlet: characteristic FEM-TVD method, MC limiter.

10.7 NACA 0012 Airfoil

Finally, let us deal with external flows past a NACA 0012 airfoil at various freestream conditions. The upper and lower surfaces of this symmetric profile are given in the form $y = \pm f(x)$, respectively, where $x \in [0, 1.00893]$ and

$$f(s) = 0.6 \left(0.2969s^{1/2} - 0.126s - 0.3516s^2 + 0.2843s^3 - 0.1015s^4 \right).$$

The compressible Euler equations are to be solved in a bounded domain, whose artificial far-field boundary is a circle of radius 10 centered at the tip of the airfoil. The first coarse mesh for this simulation is displayed in Fig. 11. It numbers 5,828 triangular elements and 2,973 vertices. A close-up of the central part shows the mesh details in the vicinity of the internal boundary. As an alternative, we consider a quadrilateral coarse mesh which consists of 3,000 elements and exhibits an unstructured tessellation pattern (see Fig. 12). In either case, the final mesh is constructed from the coarse one by two quad-tree refinements, which gives 93,248 triangles or 48,000 quadrilaterals.

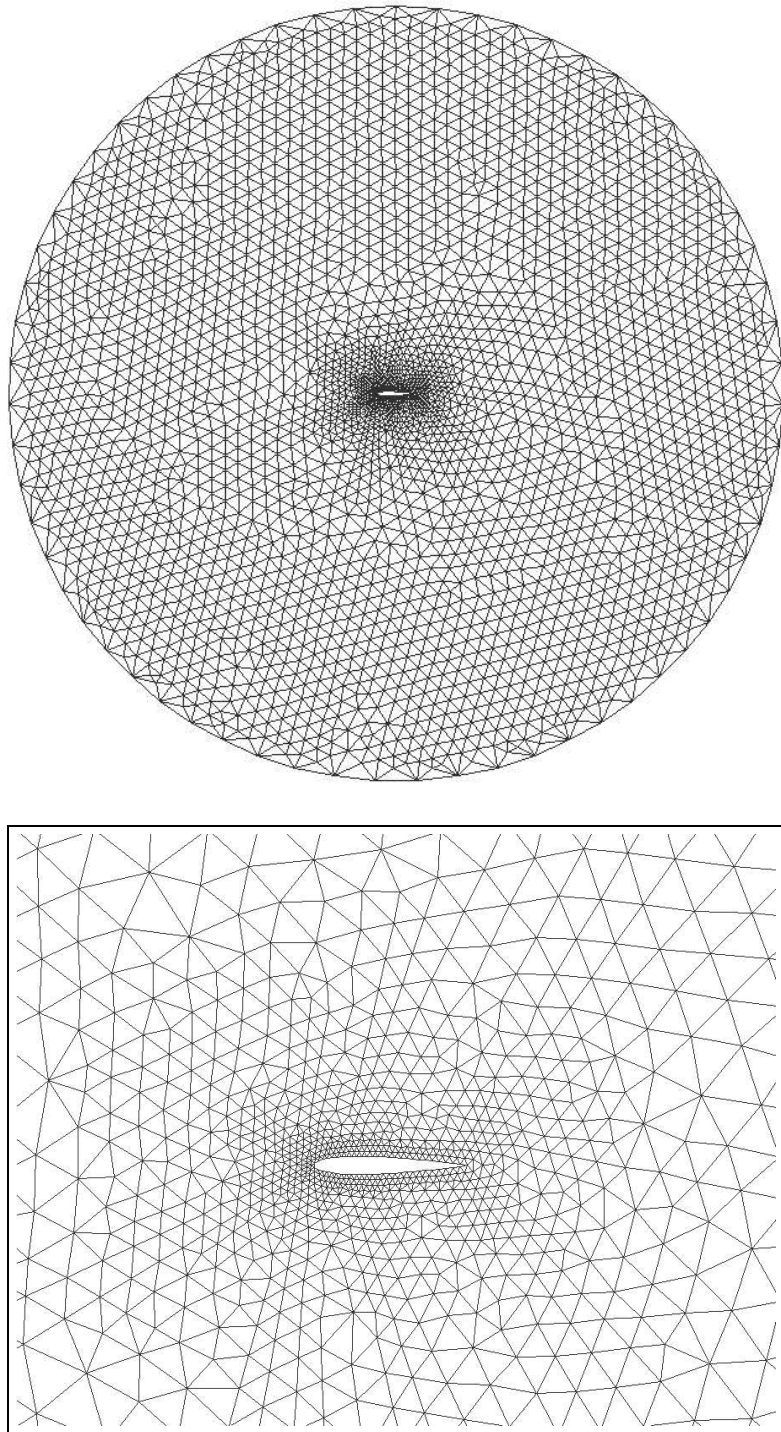


Fig. 11. Triangular coarse mesh for a NACA 0012 airfoil.

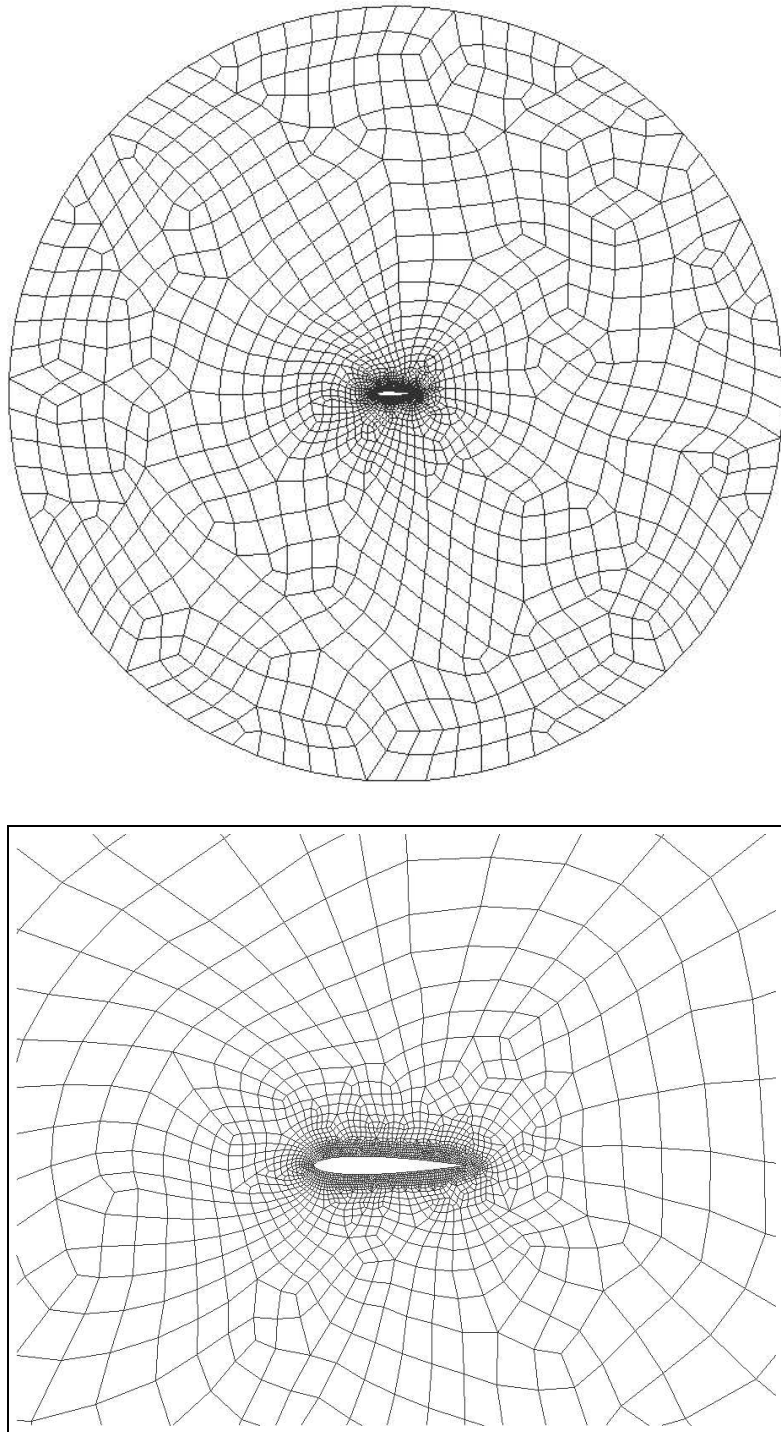


Fig. 12. Quadrilateral coarse mesh for a NACA 0012 airfoil.

The density and pressure are initialized by unity and steady-state solutions are sought for the following far-field Mach numbers and angles of attack

Figure 13	$M_\infty = 0.5, \alpha = 0^\circ$	subsonic flow
Figure 14	$M_\infty = 0.8, \alpha = 1.25^\circ$	transonic flow
Figure 15	$M_\infty = 0.95, \alpha = 0^\circ$	transonic flow

A comparison with the simulation results reported in the literature [11],[15] reveals that the numerical solutions presented in Fig. 13-15 are quite accurate, even though no adaptive mesh refinement was performed. Weighted a posteriori error estimates for an adaptive finite element discretization of the Euler equations can be derived as described in [15]. Clearly, even low-order methods yield reasonable results on a suitably refined mesh. The goal of our numerical study was to evaluate the performance of the new algorithm on a nonuniform mesh constructed without using any knowledge of the solution. On the other hand, the advent of self-adaptive FEM-TVD and FEM-FCT schemes would make it possible to exploit the advantages of unstructured grids to the full extent and create a very powerful CFD tool in the long run.

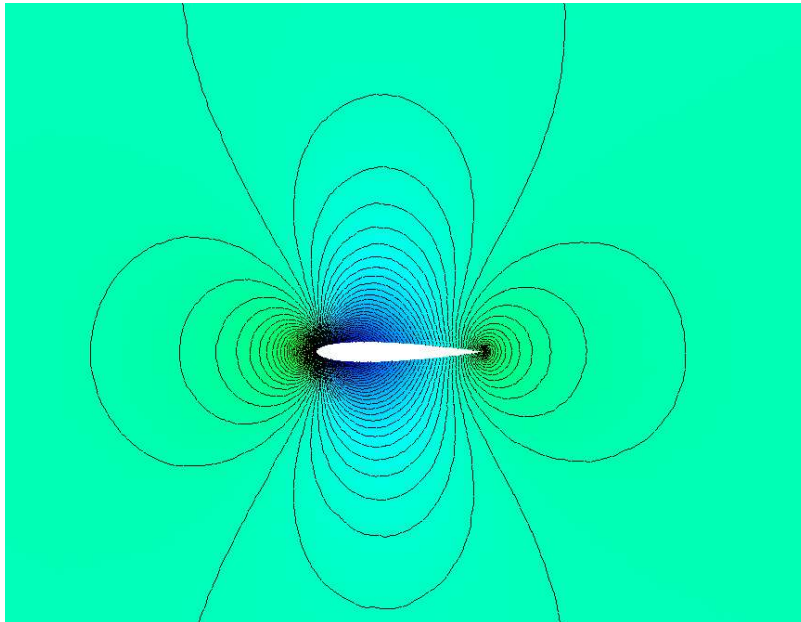
11 Conclusions and Outlook

In the present chapter, we extended the algebraic flux correction paradigm to the compressible Euler equations and proposed a practical algorithm for implementation of the resulting high-resolution schemes in a finite element code. The theory of hyperbolic conservation laws was occasionally revisited to make the exposition as self-contained as possible. At the same time, emphasis was laid on various numerical aspects including but not limited to

- edge-by-edge matrix assembly for implicit Galerkin discretizations;
- construction of approximate Riemann solvers on unstructured grids;
- flux correction, synchronization and the choice of control variables;
- segregated versus coupled solution methods for algebraic systems;
- implementation of characteristic boundary conditions.

Our algebraic FCT and TVD methods were shown to prove their worth in the systems arena. However, there are still many open problems and unexplored areas that call for further research. In particular, characteristic flux limiters would benefit from a fully multidimensional wave model. For implicit time-stepping to be profitable, the combination of linear algebra tools should be adapted to the underlying discretization. A strongly coupled iterative solver can be developed in the framework of the FMG-FAS multigrid procedure which lends itself to the treatment of the stationary Euler equations [16],[27]. An analog of the local MPSC smoother [44] offers a promising alternative to the block-iterative or ‘collective’ (pointwise-coupled) Gauß-Seidel relaxation. The marriage of algebraic flux correction and high-performance computing techniques for large-scale applications is discussed in the next chapter.

Pressure isolines, FEM-TVD scheme, P_1 elements



Pressure isolines, FEM-TVD scheme, Q_1 elements

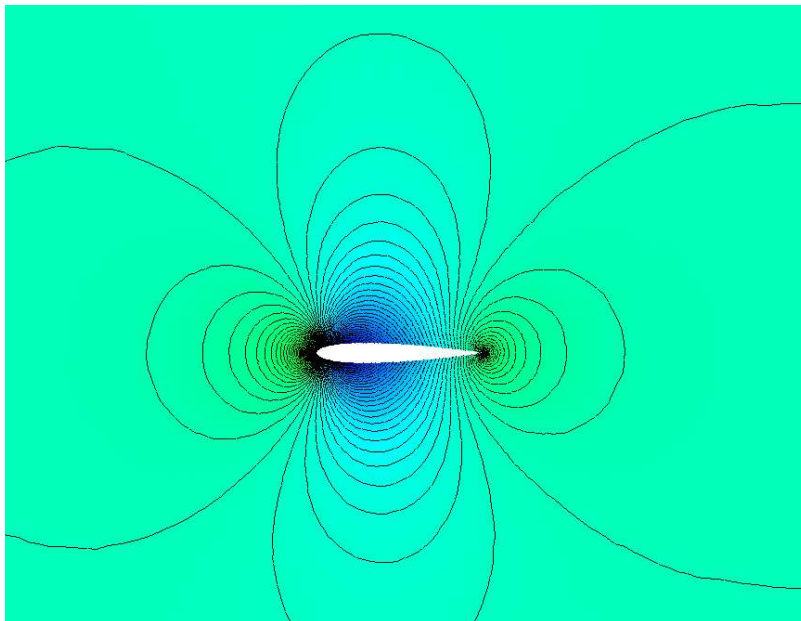
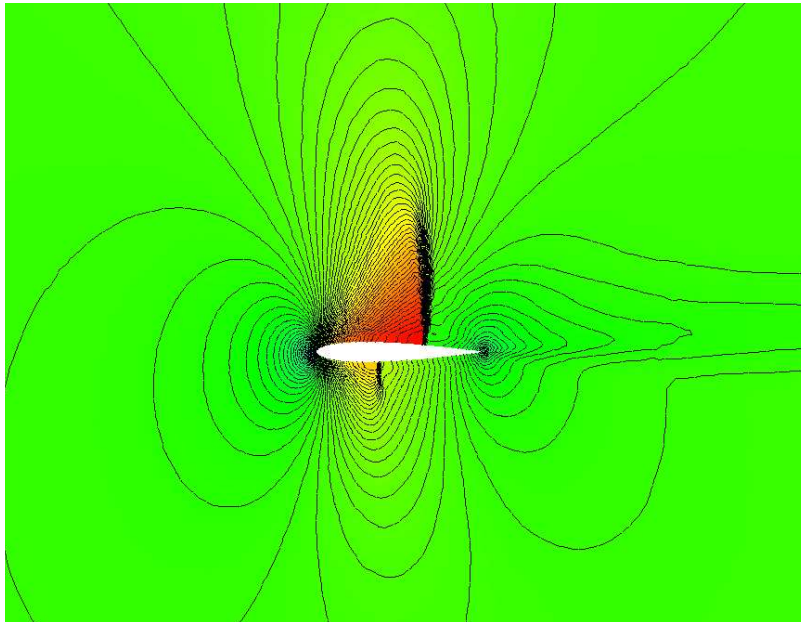


Fig. 13. Flow past a NACA 0012 airfoil at $M_\infty = 0.5$, $\alpha = 0^\circ$.

Mach number isolines, FEM-TVD scheme, P_1 elements



Mach number isolines, FEM-TVD scheme, Q_1 elements

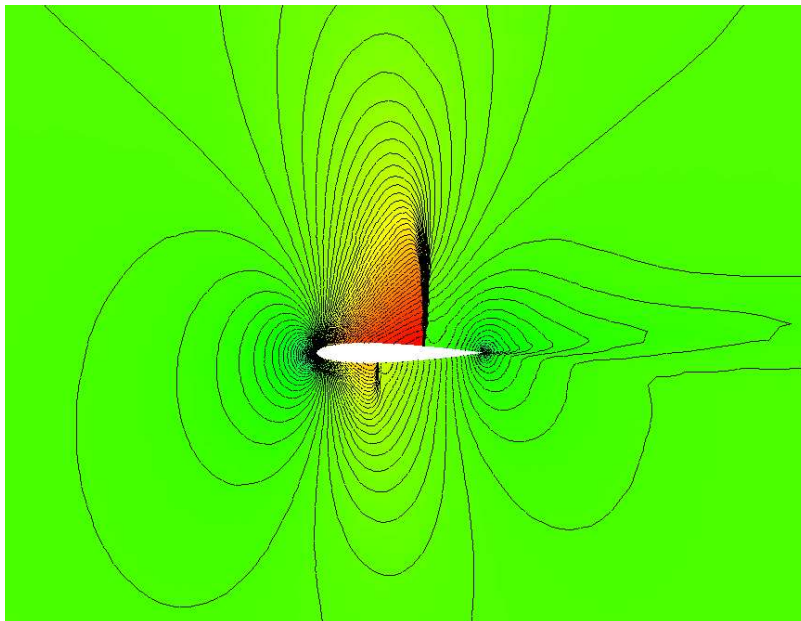
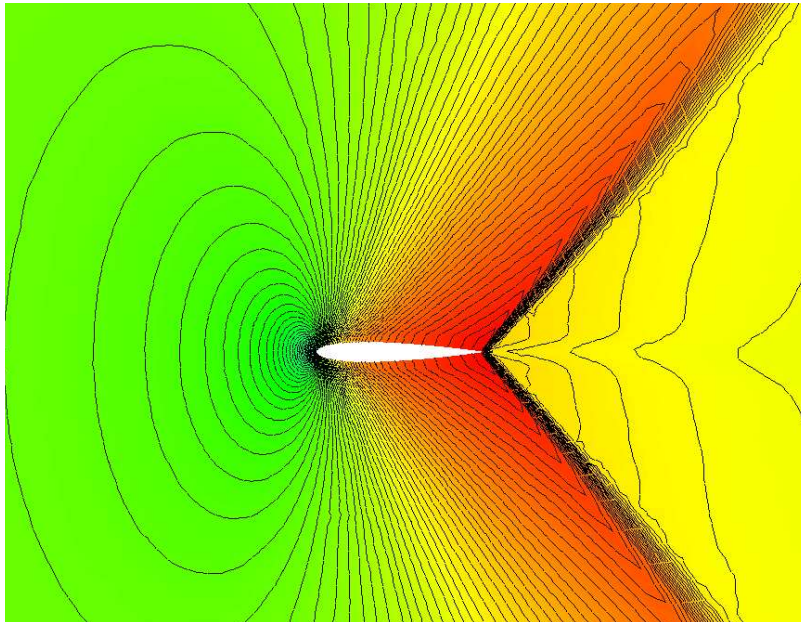


Fig. 14. Flow past a NACA 0012 airfoil at $M_\infty = 0.8$, $\alpha = 1.25^\circ$.

Mach number isolines, FEM-TVD scheme, P_1 elements



Mach number isolines, FEM-TVD scheme, Q_1 elements

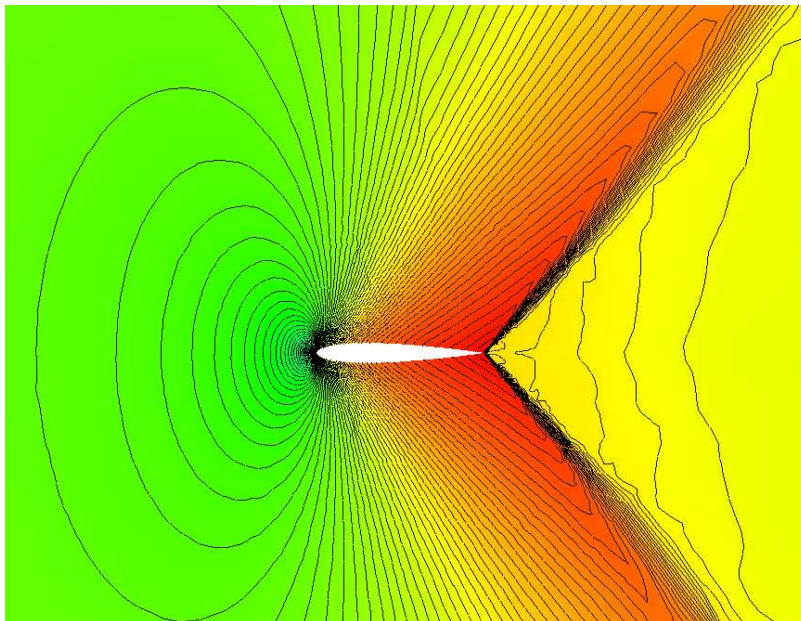


Fig. 15. Flow past a NACA 0012 airfoil at $M_\infty = 0.95$, $\alpha = 0^\circ$.

A. Characteristic Boundary Conditions

In this appendix, we elaborate on the implementation of numerical boundary conditions for the Euler equations. The necessary background is presented in section 9. Let us omit the subscript i and denote the predicted values of the density, momentum, and total energy by ρ_* , $(\rho\mathbf{v})_*$ and $(\rho E)_*$, respectively. The velocity \mathbf{v}_* , pressure p_* , and speed of sound c_* are given by

$$\mathbf{v}_* = \frac{(\rho\mathbf{v})_*}{\rho_*}, \quad p_* = (\gamma - 1) \left[(\rho E)_* - \rho_* \frac{|\mathbf{v}_*|^2}{2} \right], \quad c_* = \sqrt{\frac{\gamma p_*}{\rho_*}}.$$

These auxiliary quantities are required for the computation of the provisional Riemann invariants w^* from an analog of (88). The correction to be performed depends on the number and type of physical boundary conditions.

Supersonic open boundary

At a supersonic inlet, the boundary values of all conservative variables can be prescribed without conversion to the Riemann invariants. At a supersonic outlet, no boundary conditions at all are required.

Subsonic inflow boundary

The analytical values of $w_1^{**}, \dots, w_4^{**}$ are obtained from the physical boundary conditions. The first two Riemann invariants are plugged into the relations

$$w_1^{**} = \mathbf{n} \cdot \mathbf{v}_{**} - \frac{2c_{**}}{\gamma - 1}, \quad w_2^{**} = c_v \log \left(\frac{p_{**}}{\rho_{**}^\gamma} \right)$$

for the new boundary values of the normal velocity, speed of sound, pressure, and density. The second pair of parameters yields the tangential velocity

$$\mathbf{v}_\tau = \mathbf{v}_\tau(w_3^{**}, w_4^{**}).$$

The outgoing Riemann invariant retains its predicted value w_5^* so that

$$w_5^{**} = \mathbf{n} \cdot \mathbf{v}_{**} + \frac{2c_{**}}{\gamma - 1} = \mathbf{n} \cdot \mathbf{v}_* + \frac{2c_*}{\gamma - 1} = w_5^*.$$

This gives a system of five equations for five independent variables. Specifically, the components of the vector U^{**} can be recovered as follows

$$\begin{aligned} c_{**} &= (\gamma - 1) \frac{w_5^* - w_1^{**}}{4}, & \rho_{**} &= \left[\frac{c_{**}^2}{\gamma} \exp(-w_2^{**}/c_v) \right]^{\frac{1}{\gamma-1}}, \\ \mathbf{v}_{**} &= \mathbf{v}_\tau + \frac{w_1^{**} + w_5^*}{2} \mathbf{n}, & (\rho\mathbf{v})_{**} &= \rho_{**} \mathbf{v}_{**}, \\ (\rho E)_{**} &= \rho_{**} \left[\frac{c_{**}^2}{\gamma(\gamma - 1)} + \frac{|\mathbf{v}_{**}|^2}{2} \right], & U^{**} &= [\rho_{**}, (\rho\mathbf{v})_{**}, (\rho E)_{**}]^T. \end{aligned}$$

Subsonic outflow boundary

The incoming Riemann invariant w_1^* is overwritten by the imposed boundary condition $w_1^{**} = \mathbf{n} \cdot \mathbf{v}_{**} - \frac{2c_{**}}{\gamma-1}$. All other characteristic variables retain their boundary values computed at the predictor step. In particular, we have

$$\begin{aligned} w_2^{**} &= c_v \log(p_{**}/\rho_{**}^\gamma) = c_v \log(p_*/\rho_*^\gamma) = w_2^*, \\ w_5^{**} &= \mathbf{n} \cdot \mathbf{v}_{**} + \frac{2c_{**}}{\gamma-1} = \mathbf{n} \cdot \mathbf{v}_* + \frac{2c_*}{\gamma-1} = w_5^* \end{aligned}$$

and the tangential velocity \mathbf{v}_τ is defined by w_3^* and w_4^* . This information is sufficient to compute the nodal values of the conservative variables

$$\begin{aligned} c_{**} &= (\gamma-1) \frac{w_5^{**} - w_1^{**}}{4}, & \rho_{**} &= \left[\frac{c_{**}^2}{\gamma} \exp(-w_2^*/c_v) \right]^{\frac{1}{\gamma-1}}, \\ \mathbf{v}_{**} &= \mathbf{v}_* + \left[\frac{w_1^{**} + w_5^{**}}{2} - \mathbf{n} \cdot \mathbf{v}_* \right] \mathbf{n}, & (\rho \mathbf{v})_{**} &= \rho_{**} \mathbf{v}_{**}, \\ (\rho E)_{**} &= \rho_{**} \left[\frac{c_{**}^2}{\gamma(\gamma-1)} + \frac{|\mathbf{v}_{**}|^2}{2} \right], & \mathbf{U}^{**} &= [\rho_{**}, (\rho \mathbf{v})_{**}, (\rho E)_{**}]^T. \end{aligned}$$

As a rule, the pressure p_{**} rather than the Riemann invariant w_1^{**} is specified at a subsonic outlet. In this case, the transformation formulae read

$$\begin{aligned} \rho_{**} &= \rho_* \left(\frac{p_{**}}{p_*} \right)^{\frac{1}{\gamma}}, & c_{**} &= \sqrt{\frac{\gamma p_{**}}{\rho_{**}}}, \\ \mathbf{v}_{**} &= \mathbf{v}_* + \frac{2(c_* - c_{**})}{\gamma-1} \mathbf{n}, & (\rho \mathbf{v})_{**} &= \rho_{**} \mathbf{v}_{**}, \\ (\rho E)_{**} &= \frac{p_{**}}{\gamma-1} + \rho_{**} \frac{|\mathbf{v}_{**}|^2}{2}, & \mathbf{U}^{**} &= [\rho_{**}, (\rho \mathbf{v})_{**}, (\rho E)_{**}]^T. \end{aligned}$$

Solid wall boundary

At an impervious solid wall, the normal velocity v_n must vanish. Taking this physical boundary condition into account, we obtain the expression

$$w_5^{**} = \frac{2c_{**}}{\gamma-1} = \mathbf{n} \cdot \mathbf{v}_* + \frac{2c_*}{\gamma-1} = w_5^*.$$

Furthermore, $v_n = 0$ implies that the entropy and the tangential velocity components do not cross the boundary. Therefore, their original values should not be corrected and the resulting conversion rules are as follows

$$\begin{aligned} c_{**} &= c_* + \frac{\gamma-1}{2} \mathbf{n} \cdot \mathbf{v}_*, & \rho_{**} &= \left[\frac{c_{**}^2}{\gamma} \frac{\rho_*^\gamma}{p_*} \right]^{\frac{1}{\gamma-1}}, \\ \mathbf{v}_{**} &= \mathbf{v}_* - (\mathbf{n} \cdot \mathbf{v}_*) \mathbf{n}, & (\rho \mathbf{v})_{**} &= \rho_{**} \mathbf{v}_{**}, \\ (\rho E)_{**} &= \rho_{**} \left[\frac{c_{**}^2}{\gamma(\gamma-1)} + \frac{|\mathbf{v}_{**}|^2}{2} \right], & \mathbf{U}^{**} &= [\rho_{**}, (\rho \mathbf{v})_{**}, (\rho E)_{**}]^T. \end{aligned}$$

The second line in this group of formulae corresponds to a projection of the predicted momentum onto the tangent plane. This modification is sufficient to enforce the free slip boundary condition $\mathbf{n} \cdot \mathbf{v}_{**} = 0$. Alternatively, it can be applied to the defect for the momentum equation before computing U^* from (90). In either case, the correction of ρ_* and $(\rho E)_*$ is optional.

If the initial data fail to satisfy the free slip condition, the abrupt change of the normal velocity results in an *impulsive start* which is physically impossible due to inertia. Hence, various numerical difficulties and poor convergence are to be expected. This problem was investigated by Lyra [28],[32] who proposed the following relaxation procedure for the freestream condition

$$\mathbf{n} \cdot \mathbf{v}^{n+1} = (1 - \omega)\mathbf{n} \cdot \mathbf{v}^n, \quad 0 \leq \omega < 1.$$

In essence, the fluid is allowed to seep through the wall during the startup but the normal velocity is gradually driven to zero as the flow evolves. The relaxation parameter ω determines the rate at which it is adjusted.

Acknowledgments

The authors would like to thank Stefan Turek for fruitful discussions and Jens Acker for assistance with mesh generation.

References

1. J. D. Anderson, Jr., *Modern Compressible Flow*, McGraw-Hill, 1990.
2. P. Arminjon and A. Dervieux, Construction of TVD-like artificial viscosities on 2-dimensional arbitrary FEM grids. *INRIA Research Report 1111* (1989).
3. J.-C. Carette, H. Deconinck, H. Paillère and P.L. Roe, Multidimensional upwinding: Its relation to finite elements. *Int. J. Numer. Methods Fluids* **20** (1995) no. 8-9, 935-955.
4. G. G. Cherny, *Gas Dynamics* (in Russian). Nauka, Moscow, 1988.
5. R. Codina and M. Cervera, Block-iterative algorithms for nonlinear coupled problems In: *Advanced computational methods in structural mechanics*, Chapter 7. Eds. M. Papadrakakis and G. Bueda. Theory and Engineering Applications of Computational Methods CIMNE, Barcelona, 1996.
6. D. L. Darmofal and B. Van Leer, Local preconditioning: Manipulating mother nature to fool father time. In: D. A. Caughey et al. (eds), *Frontiers of computational fluid dynamics*, Singapore: World Scientific Publishing, 1998, 211-239.
7. H. Deconinck, H. Paillère, R. Struijs and P.L. Roe, Multidimensional upwind schemes based on fluctuation-splitting for systems of conservation laws. *Comput. Mech.* **11** (1993) no. 5-6, 323-340.
8. M. J. Díaz, F. Hecht and B. Mohammadi, New Progress in Anisotropic Grid Adaptation for Inviscid and Viscous Flows Simulations. In *Proceedings of the 4th Annual International Meshing Roundtable*. Sandia National Laboratories, 1995.

9. J. Donea and A. Huerta, *Finite Element Methods for Flow Problems*. Wiley, 2003.
10. J. Donea, V. Selmin and L. Quartapelle, Recent developments of the Taylor-Galerkin method for the numerical solution of hyperbolic problems. *Numerical methods for fluid dynamics III*, Oxford, 171-185 (1988).
11. M. Feistauer, J. Felcman and I. Straškraba, *Mathematical and Computational Methods for Compressible Flow*. Clarendon Press, Oxford, 2003.
12. J.H. Ferziger and M. Peric, *Computational Methods for Fluid Dynamics*. Springer, 1996.
13. C. A. J. Fletcher, *Computational Techniques for Fluid Dynamics*. Springer, 1988.
14. A. Harten and J.M. Hyman, Self adjusting grid methods for one-dimensional hyperbolic conservation laws. *J. Comput. Phys.* **50** (1983) 235–269.
15. R. Hartmann and P. Houston, Adaptive Discontinuous Galerkin Finite Element Methods for the Compressible Euler Equations. *J. Comput. Phys.* **183** (2002) 508-532.
16. P.W. Hemker and B. Koren, Defect correction and nonlinear multigrid for steady Euler equations. In: W.G. Habashi and M.M. Hafez (eds), *Computational fluid dynamics techniques*. London: Gordon and Breach Publishers, 1995, 699–718.
17. C. Hirsch, *Numerical Computation of Internal and External Flows. Vol. II: Computational Methods for Inviscid and Viscous Flows*. John Wiley & Sons, Chichester, 1990.
18. D. Kuzmin, M. Möller and S. Turek, Multidimensional FEM-FCT schemes for arbitrary time-stepping. *Int. J. Numer. Meth. Fluids* **42** (2003) 265-295.
19. D. Kuzmin, M. Möller and S. Turek, High-resolution FEM-FCT schemes for multidimensional conservation laws. Technical report **231**, University of Dortmund, 2003. To appear in: *Comput. Methods Appl. Mech. Engrg.*
20. R. J. LeVeque, *Numerical Methods for Conservation Laws*. Birkhäuser, 1992.
21. R. J. LeVeque, Simplified multi-dimensional flux limiting methods. *Numerical Methods for Fluid Dynamics IV* (1993) 175–190.
22. R. J. LeVeque, *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, 2003.
23. R. J. LeVeque, CLAWPACK – Conservation LAWs PACKage, available on the Web at the URL <http://www.amath.washington.edu/~claw/>.
24. R. Löhner, K. Morgan, J. Peraire and M. Vahdati, Finite element flux-corrected transport (FEM-FCT) for the Euler and Navier-Stokes equations. *Int. J. Numer. Meth. Fluids* **7** (1987) 1093–1109.
25. R. Löhner, *Applied CFD Techniques: An Introduction Based on Finite Element Methods*. Wiley, 2001.
26. R. Löhner and J.D. Baum, 30 years of FCT: status and directions. In this volume.
27. J. F. Lynn, *Multigrid Solution of the Euler Equations with Local Preconditioning*. PhD thesis, University of Michigan, 1995.
28. P. R. M. Lyra, *Unstructured Grid Adaptive Algorithms for Fluid Dynamics and Heat Conduction*. PhD thesis, University of Wales, Swansea, 1994.
29. P. R. M. Lyra, K. Morgan, J. Peraire and J. Peiro, TVD algorithms for the solution of the compressible Euler equations on unstructured meshes. *Int. J. Numer. Meth. Fluids* **19** (1994) 827–847.

30. P. R. M. Lyra and K. Morgan, A review and comparative study of upwinding biased schemes for compressible flow computation. I: 1-D first-order schemes. *Arch. Comput. Methods Eng.* **7** (2000) no. 1, 19-55.
31. P. R. M. Lyra and K. Morgan, A review and comparative study of upwinding biased schemes for compressible flow computation. II: 1-D higher-order schemes. *Arch. Comput. Methods Eng.* **7** (2000) no. 3, 333-377.
32. P. R. M. Lyra and K. Morgan, A review and comparative study of upwinding biased schemes for compressible flow computation. III: Multidimensional extension on unstructured grids. *Arch. Comput. Methods Eng.* **9** (2002) no. 3, 207-256.
33. M. Möller, *Hochauflösende FEM-FCT-Verfahren zur Diskretisierung von konvektionsdominanten Transportproblemen mit Anwendung auf die kompressiblen Eulergleichungen*. Diploma thesis, University of Dortmund, 2003.
34. M. Möller, D. Kuzmin and S. Turek, Implicit FEM-FCT algorithm for compressible flows. To appear in: Proceedings of the *European Conference on Numerical Mathematics and Advanced Applications*, August 18-22, 2003, Prague.
35. K. Morgan and J. Peraire, Unstructured grid finite element methods for fluid mechanics. *Reports on Progress in Physics*, **61** (1998), no. 6, 569-638.
36. NPARC Alliance, Computational Fluid Dynamics (CFD) Verification and Validation Web Site: <http://www.grc.nasa.gov/WWW/wind/valid/>.
37. P. L. Roe, Approximate Riemann solvers, parameter vectors and difference schemes. *J. Comput. Phys.* **43** (1981) 357-372.
38. A. Rohde, Eigenvalues and eigenvectors of the Euler equations in general geometries. *AIAA Paper* 2001-2609.
39. V. Selmin, Finite element solution of hyperbolic equations. I. One-dimensional case. *INRIA Research Report* **655** (1987).
40. V. Selmin, Finite element solution of hyperbolic equations. II. Two-dimensional case. *INRIA Research Report* **708** (1987).
41. D. Sidilkover, A genuinely multidimensional upwind scheme and efficient multigrid solver for the compressible Euler equations, *ICASE Report* No. 94-84, 1994.
42. G. Sod, A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *J. Comput. Phys.* **27** (1978) 1-31.
43. E. F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer, 1999.
44. S. Turek, *Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approach*, LNCSE **6**, Springer, 1999.
45. P. Wesseling, *Principles of Computational Fluid Dynamics*. Springer, 2001.
46. H. C. Yee, Numerical approximations of boundary conditions with applications to inviscid gas dynamics. *NASA report* TM-81265, 1981.
47. H. C. Yee, Construction of explicit and implicit symmetric TVD schemes and their applications. *J. Comput. Phys.* **43** (1987) 151-179.
48. H. C. Yee, R. F. Warming and A. Harten, Implicit Total Variation Diminishing (TVD) schemes for steady-state calculations. *J. Comput. Phys.* **57** (1985) 327-360.
49. S. T. Zalesak, On the use of characteristic variables in the design of Flux-Corrected Transport (FCT) Algorithms. Preprint, 2003.
50. S. T. Zalesak, On the design of Flux-Corrected Transport (FCT) Algorithms. In this volume.