Algebraic Flux Correction II
Compressible Flow Problems

Dmitri Kuzmin, Matthias Möller, and Marcel Gurris

Abstract Flux limiting for hyperbolic systems requires a careful generalization of the design principles and algorithms introduced in the context of scalar conservation laws. In this chapter, we develop FCT-like algebraic flux correction schemes for the Euler equations of gas dynamics. In particular, we discuss the construction of artificial viscosity operators, the choice of variables to be limited, and the transformation of antidiffusive fluxes. An a posteriori control mechanism is implemented to make the limiter failsafe. The numerical treatment of initial and boundary conditions is discussed in some detail. The initialization is performed using an FCT-constrained L^2 projection. The characteristic boundary conditions are imposed in a weak sense, and an approximate Riemann solver is used to evaluate the fluxes on the boundary. We also present an unconditionally stable semi-implicit time-stepping scheme and an iterative solver for the fully discrete problem. The results of a numerical study indicate that the nonlinearity and non-differentiability of the flux limiter do not inhibit steady state convergence even in the case of strongly varying Mach numbers. Moreover, the convergence rates improve as the pseudo-time step is increased.

1 Introduction

The first successful finite element schemes for compressible flow problems were developed by the Swansea and INRIA groups in the 1980s. The most prominent representative of these schemes is the two-step Taylor-Galerkin method [1, 43] and its combination with FCT [42, 57, 58]. The early 1990s have witnessed the advent of edge-based data structures [6, 44, 54, 59] that offer a number of significant advan-

Dmitri Kuzmin
Applied Mathematics III, University Erlangen-Nuremberg
Cauerstr. 11, D-91058, Erlangen, Germany
e-mail: kuzmin@am.uni-erlangen.de

Matthias Möller · Marcel Gurris
Institute of Applied Mathematics (LS III), TU Dortmund
Vogelpothsweg 87, D-44227, Dortmund, Germany
e-mail: {matthias.moeller,marcel.gurris}@math.tu-dortmund.de

tages compared to the traditional element-based implementation. In the case of P_1 finite elements, the edge-based formulation is equivalent to a vertex-centered finite volume scheme [59, 60]. This equivalence makes it possible to implement approximate Riemann solvers and slope limiters in the context of finite element discretizations on simplex meshes [46, 47, 48, 49, 52, 54]. However, the resulting schemes require mass lumping and are sensitive to the orientation of mesh edges.

All classical high-resolution FEM are explicit and, therefore, subject to time step restrictions. Implicit schemes have the potential of being unconditionally stable but rely on the quality of the iterative solver for the nonlinear system. In particular, a careful linearization/preconditioning of the discrete Jacobian operator is essential. A semi-implicit solution strategy [10, 14, 67] and weak imposition of characteristic boundary conditions [18] lead to an algorithm that converges to steady state solutions at arbitrarily large CFL numbers [18, 19]. This is a remarkable result since the use of nondifferentiable limiters is commonly believed to inhibit convergence.

The development of flux-corrected transport schemes for systems of equations is more difficult than in the scalar case. A limiter designed to control the local maxima and minima of the conservative variables does not guarantee that the pressure or internal energy will stay nonnegative. Likewise, the velocity is not directly constrained and may exhibit spurious fluctuations. Since the rate of transport depends on the oscillatory velocity and pressure fields, undershoots and overshoots eventually carry over to the conservative variables. As a typical consequence, the speed of sound becomes negative, indicating that the simulation is going to crash.

In this chapter, we review some recent advances in the design of implicit algebraic flux correction schemes for the Euler equations [18, 19, 32, 33, 34, 50]. After the presentation of the standard Galerkin scheme, we discuss various forms of artificial dissipation and the above difficulties associated with flux limiting for systems of equations. In particular, we present a synchronized FCT limiter that features a node-based transformation to primitive variables and a failsafe control mechanism inspired by the recent work of Zalesak [76]. Also, we address the treatment of nonlinearities and the implementation of initial/boundary conditions. A numerical study is performed for a number of steady and unsteady inviscid flow problems in 2D.

2 The 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

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

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

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

where ρ is the density, $\mathbf{v}$ is the velocity, p is the pressure, E is the total energy, and $\mathscr{I}$ is the identity tensor. The system is closed with the equation of state

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

for an ideal polytropic gas with the heat capacity ratio γ. The default is $\gamma = 1.4$ (air).

The nonlinear system (1)–(3) can be written in the generic divergence form

$$\frac{\partial U}{\partial t} + \nabla \cdot \mathbf{F} = 0, \tag{5}$$

where

$$U = \begin{bmatrix} \rho \\ \rho\mathbf{v} \\ \rho E \end{bmatrix}, \qquad \mathbf{F} = \begin{bmatrix} \rho\mathbf{v} \\ \rho\mathbf{v}\otimes\mathbf{v} + p\mathscr{I} \\ \rho E\mathbf{v} + p\mathbf{v} \end{bmatrix} \tag{6}$$

are the vectors of conservative variables and fluxes. It can be shown that [71]

$$\mathbf{F} = \mathbf{A}U, \tag{7}$$

where $\mathbf{A} = \frac{\partial \mathbf{F}}{\partial U}$ is the Jacobian tensor associated with the quasi-linear form of (5)

$$\frac{\partial U}{\partial t} + \mathbf{A} \cdot \nabla U = 0. \tag{8}$$

Due to the hyperbolicity of the Euler equations, any directional Jacobian matrix $\mathbf{e} \cdot \mathbf{A}$ is diagonalizable and admits the factorization [24, 37, 71]

$$\mathbf{e} \cdot \mathbf{A} = R\Lambda R^{-1}, \tag{9}$$

where $\Lambda(\mathbf{e})$ is the diagonal matrix of eigenvalues and $R(\mathbf{e})$ is the matrix of right eigenvectors. In the 3D case, the eigenvalues of the 5×5 matrix $\mathbf{e} \cdot \mathbf{A}$ are given by

$$\lambda_1 = \mathbf{e} \cdot \mathbf{v} - c, \tag{10}$$

$$\lambda_2 = \lambda_3 = \lambda_4 = \mathbf{e} \cdot \mathbf{v}, \tag{11}$$

$$\lambda_5 = \mathbf{e} \cdot \mathbf{v} + c, \tag{12}$$

where $c = \sqrt{\gamma p/\rho}$ is the speed of sound. Thus, the solution to a Riemann problem is a superposition of three waves traveling at speed $\mathbf{e} \cdot \mathbf{v}$ and two waves propagating at speeds $\pm c$ relative to the gas. Closed-form expressions for the eigenvectors associated with each characteristic speed can be found, e.g., in [56].

Let $\Omega \subset \mathbb{R}^n$, $n \in \{1,2,3\}$ be a bounded domain. The solution to the unsteady Euler equations is initialized by a given distribution of all variables

$$U(\mathbf{x},t) = U_0(\mathbf{x}) \qquad \text{in } \Omega. \tag{13}$$

Given a vector of "free stream" solution values U_∞, characteristic boundary conditions of Dirichlet or Neumann type can be defined in terms of the solution to the Riemann problem with the interior state U and exterior state U_∞, see Section 11.

In general, we impose Dirichlet boundary conditions on the boundary part Γ_D

$$U = G(U, U_\infty) \qquad \text{on } \Gamma_D \tag{14}$$

and Neumann (normal flux) boundary conditions on the boundary part Γ_N

$$\mathbf{n} \cdot \mathbf{F} = F_n(U, U_\infty) \qquad \text{on } \Gamma_N, \tag{15}$$

where $\mathbf{n}$ is the unit outward normal. Note that the solution to the Riemann problem depends not only on the prescribed boundary data but also on the unknown solution.

3 Group FEM Approximation

To begin with, we discretize the Euler equations using linear or multilinear finite elements. After integration by parts, the variational formulation of (5) becomes

$$\int_\Omega \left(w \frac{\partial U}{\partial t} - \nabla w \cdot \mathbf{F} \right) \mathrm{d}\mathbf{x} + \int_\Gamma w F_n \mathrm{d}s = 0, \qquad \forall w. \tag{16}$$

Since the test function w vanishes on Γ_D, the surface integral reduces to that over Γ_N.

Within the framework of Fletcher's [16] group finite element formulation, the approximate solution $U_h \approx U$ and the numerical flux function $\mathbf{F}_h \approx \mathbf{F}$ are interpolated using the same set of piecewise-polynomial basis functions $\{\varphi_i\}$. That is,

$$U_h(\mathbf{x}, t) = \sum_j \mathrm{U}_j(t) \varphi_j(\mathbf{x}), \tag{17}$$

$$\mathbf{F}_h(\mathbf{x}, t) = \sum_j \mathbf{F}_j(t) \varphi_j(\mathbf{x}). \tag{18}$$

Inserting these approximations into the Galerkin weak form (16), one obtains a system of semi-discretized equations for the time-dependent nodal values

$$\sum_j \left(\int_\Omega \varphi_i \varphi_j \, \mathrm{d}\mathbf{x} \right) \frac{\mathrm{d}\mathrm{U}_j}{\mathrm{d}t} = \sum_j \left(\int_\Omega \nabla \varphi_i \varphi_j \mathrm{d}\mathbf{x} \right) \cdot \mathbf{F}_j - \int_\Gamma \varphi_i \mathrm{F}_n \mathrm{d}s. \tag{19}$$

By the homogeneity property (7) of the Euler fluxes, we have

$$\mathbf{F}_j = \mathbf{A}_j \mathrm{U}_j.$$

Thus, the matrix form of the semi-discrete problem can be written as follows:

$$M_C \frac{\mathrm{dU}}{\mathrm{d}t} = K\mathrm{U} + \mathrm{S}(\mathrm{U}). \tag{20}$$

The $(n+2)\times(n+2)$ blocks of the consistent mass matrix $M_C = \{\mathrm{M}_{ij}\}$ are defined by $\mathrm{M}_{ij} = m_{ij}\mathrm{I}$, where I stands for the identity matrix and

$$m_{ij} = \int_\Omega \varphi_i \varphi_j \, \mathrm{d}\mathbf{x}. \tag{21}$$

Furthermore, the vector of boundary loads associated with node i is given by

$$\mathrm{S}_i = -\int_\Gamma \varphi_i \mathrm{F}_n \mathrm{d}s, \tag{22}$$

and the formula for entries of the discrete Jacobian operator $K = \{\mathrm{K}_{ij}\}$ reads

$$\mathrm{K}_{ij} = \mathbf{c}_{ji} \cdot \mathbf{A}_j, \qquad \mathbf{c}_{ij} = \int_\Omega \varphi_i \nabla \varphi_j \, \mathrm{d}\mathbf{x}. \tag{23}$$

Since $\sum_j \varphi_j \equiv 1$, the matrix of discrete derivatives $\mathbf{C} := \{\mathbf{c}_{ij}\}$ has zero row sums

$$\sum_j \mathbf{c}_{ij} = 0. \tag{24}$$

Furthermore, integration by parts reveals that the coefficients $\mathbf{c}_{ij}$ and $\mathbf{c}_{ji}$ satisfy

$$\mathbf{c}_{ji} = -\mathbf{c}_{ij} + \int_\Gamma \varphi_i \varphi_j \mathbf{n} \, \mathrm{d}s. \tag{25}$$

The boundary term is symmetric and corresponds to an entry of the mass matrix for the surface triangulation of Γ. In the case of (multi-)linear finite elements, the basis function φ_i vanishes on Γ, unless $\mathbf{x}_i$ is a boundary node. It follows that

$$\mathbf{c}_{ji} = -\mathbf{c}_{ij}, \qquad \mathbf{c}_{ii} = 0, \qquad \mathrm{S}_i = 0 \tag{26}$$

in the interior of Ω. The above properties of the discrete gradient operator $\mathbf{C}$ play an important role in the derivation of edge-based data structures [27, 40, 60].

4 Edge-Based Representation

Properties (24) and (26) make it possible to express the components of $K\mathrm{U}$ in terms of edge contributions. The following representation is valid inside Ω

$$(K\mathrm{U})_i = \sum_{j\neq i} \mathbf{e}_{ij} \cdot (\mathbf{F}_j - \mathbf{F}_i), \qquad \mathbf{e}_{ij} = \frac{\mathbf{c}_{ji} - \mathbf{c}_{ij}}{2}. \tag{27}$$

The numerical fluxes for an edge-based implementation are defined by [34, 60]

$$(K\mathrm{U})_i = -\sum_{j\neq i} \mathrm{G}_{ij}, \qquad \mathrm{G}_{ij} = \mathbf{c}_{ij} \cdot \mathbf{F}_i - \mathbf{c}_{ji} \mathbf{F}_j. \tag{28}$$

For the derivation of the above flux decomposition for $K\mathrm{U}$, we refer to [27, 34, 60].

As shown by Roe [55], the flux difference can be linearized as follows

$$\mathbf{F}_j - \mathbf{F}_i = \mathbf{A}_{ij}(\mathrm{U}_j - \mathrm{U}_i). \tag{29}$$

The edge Jacobian matrix $\mathbf{A}_{ij} := \mathbf{A}(\rho_{ij}, \mathbf{v}_{ij}, H_{ij})$ is associated with a special set of density-averaged variables known as the *Roe mean values*

$$\rho_{ij} = \sqrt{\rho_i \rho_j}, \tag{30}$$

$$\mathbf{v}_{ij} = \frac{\sqrt{\rho_i}\mathbf{v}_i + \sqrt{\rho_j}\mathbf{v}_j}{\sqrt{\rho_i} + \sqrt{\rho_j}}, \tag{31}$$

$$H_{ij} = \frac{\sqrt{\rho_i}H_i + \sqrt{\rho_j}H_j}{\sqrt{\rho_i} + \sqrt{\rho_j}}, \tag{32}$$

where $H = E + \frac{p}{\rho}$ denotes the stagnation enthalpy. The speed of sound is given by

$$c_{ij} = \sqrt{(\gamma - 1)\left(H_{ij} - \frac{|\mathbf{v}_{ij}|^2}{2}\right)}. \tag{33}$$

By virtue of (27) and (29), the following relationship holds for internal nodes

$$\mathrm{K}_{ii} = -\sum_{j \neq i} \mathrm{K}_{ij}, \qquad \mathrm{K}_{ij} = \mathbf{e}_{ij} \cdot \mathbf{A}_{ij}, \quad j \neq i. \tag{34}$$

This representation of K_{ij} turns out to be very useful when it comes to the design of artificial viscosity operators for algebraic flux correction schemes (see the next section). However, the assembly of K should be performed using definition (23).

By the hyperbolicity of the Euler equations, the directional Roe matrix $\mathbf{e}_{ij} \cdot \mathbf{A}_{ij}$ is diagonalizable with real eigenvalues. Invoking (9), we obtain the factorization

$$\mathbf{e}_{ij} \cdot \mathbf{A}_{ij} = |\mathbf{e}_{ij}|\, \mathrm{R}_{ij} \Lambda_{ij} \mathrm{R}_{ij}^{-1}. \tag{35}$$

According to (10)–(12) the entries of the eigenvalue matrix Λ_{ij} are given by

$$\lambda_1 = v_{ij} - c_{ij}, \tag{36}$$

$$\lambda_2 = \lambda_3 = \lambda_4 = v_{ij}, \tag{37}$$

$$\lambda_5 = v_{ij} + c_{ij}. \tag{38}$$

Here c_{ij} is the speed of sound (33) for Roe's approximate Riemann solver, while

$$v_{ij} = \frac{\mathbf{e}_{ij} \cdot \mathbf{v}_{ij}}{|\mathbf{e}_{ij}|}$$

is the density-averaged velocity along the (virtual) edge connecting nodes i and j.

5 Artificial Viscosity Operators

In the chapter on algebraic flux correction for scalar conservation laws [31], we constructed a nonoscillatory low-order scheme using row-sum mass lumping

$$M_L := \mathrm{diag}\{m_i \mathrm{I}\}, \qquad m_i = \sum_j m_{ij} \tag{39}$$

and conservative postprocessing of the Galerkin operator $K = \{\mathrm{K}_{ij}\}$. For systems of conservation laws, each block K_{ij} is an $(n+2)\times(n+2)$ matrix. The blocks of the artificial diffusion operator $D := \{\mathrm{D}_{ij}\}$ are matrices of the same size. As in the scalar case, the discrete Jacobian operator is modified edge-by-edge thus:

$$\begin{aligned} &\mathrm{K}_{ii} := \mathrm{K}_{ii} - \mathrm{D}_{ij}, \;\; \mathrm{K}_{ij} := \mathrm{K}_{ij} + \mathrm{D}_{ij}, \\ &\mathrm{K}_{ji} := \mathrm{K}_{ji} + \mathrm{D}_{ij}, \; \mathrm{K}_{jj} := \mathrm{K}_{jj} - \mathrm{D}_{ij}. \end{aligned} \tag{40}$$

Replacing K with $L := K + D$, one obtains the low-order approximation to (20)

$$M_L \frac{\mathrm{dU}}{\mathrm{d}t} = L\mathrm{U} + \mathrm{S}(\mathrm{U}). \tag{41}$$

If all off-diagonal matrix blocks L_{ij} are positive semi-definite, then such a low-order scheme proves local extremum diminishing (LED) with respect to local *characteristic variables* [34]. This condition is a generalization of the LED criterion for scalar transport equations. In the case of a hyperbolic system it is less restrictive than the requirement that all off-diagonal entries of L be nonnegative.

According to (34) and (35), the negative eigenvalues of K_{ij} and K_{ji} can be eliminated by adding tensorial artificial dissipation of the form [34]

$$\mathrm{D}_{ij} = |\mathbf{e}_{ij} \cdot \mathbf{A}_{ij}| := |\mathbf{e}_{ij}|\,\mathrm{R}_{ij}|\Lambda_{ij}|\mathrm{R}_{ij}^{-1}, \tag{42}$$

where $|\Lambda_{ij}|$ is a diagonal matrix containing the absolute values of the eigenvalues.

Flux limiting in terms of characteristic variables requires that the diffusive and antidiffusive fluxes be defined separately for each component of $\mathbf{e}_{ij} = (e_{ij}^1, \ldots, e_{ij}^n)$ and $\mathbf{A}_{ij} = (\mathrm{A}_{ij}^1, \ldots, \mathrm{A}_{ij}^n)$. Thus, the above definition of D_{ij} should be replaced with

$$\mathrm{D}_{ij} = |e_{ij}^1 \mathrm{A}_{ij}^1| + \cdots + |e_{ij}^n \mathrm{A}_{ij}^n|. \tag{43}$$

In the 1D case, the low-order scheme with artificial viscosity of the form (42) or (43) reduces to Roe's approximate Riemann solver (see Appendix).

The cost of evaluating the Roe matrix $\mathbf{A}_{ij}$ is rather high. An inexpensive alternative is the computation of D_{ij} using the Jacobian at the arithmetic mean state

$$\mathbf{A}_{ij} := \mathbf{A}\left(\frac{\mathrm{U}_j + \mathrm{U}_i}{2}\right). \tag{44}$$

Banks et al. [5] present a numerical study of methods that use this linearization. In particular, the expected order of accuracy is verified numerically. Importantly, the replacement of the Roe mean values with the arithmetic mean does not make the scheme nonconservative if this approximation is used in the definition of D_{ij} only.

In particularly sensitive applications, the minimal artificial viscosity based on the characteristic decomposition of $\mathbf{A}_{ij}$ may fail to suppress spurious oscillations. This is unacceptable if the flux limiter relies on the assumption that the local extrema of the low-order solution constitute physically legitimate upper and lower bounds.

A possible remedy is the use of Rusanov-like scalar dissipation proportional to the fastest characteristic speed [5, 76]. The straightforward definition is

$$\mathrm{D}_{ij} = d_{ij}\mathrm{I}, \qquad d_{ij} = |\mathbf{e}_{ij}| \max_i |\lambda_i|, \tag{45}$$

where $\max_i |\lambda_i| = |\mathbf{e}_{ij}|(|v_{ij}| + c_{ij})$ is the spectral radius of the Roe matrix. In our experience, a more robust and efficient low-order scheme is obtained with [33]

$$\mathrm{D}_{ij} = \max\{d_{ij}, d_{ji}\}\mathrm{I}, \qquad d_{ij} = |\mathbf{e}_{ij} \cdot \mathbf{v}_j| + |\mathbf{e}_{ij}| c_j, \tag{46}$$

where $c_i = \sqrt{\gamma p_i/\rho_i}$ is the speed of sound at node i. In the context of implicit schemes, scalar dissipation may be used for preconditioning purposes even if tensorial artificial viscosity of the form (42) or (43) is favored for accuracy reasons.

6 Algebraic Flux Correction

The semi-discrete Galerkin scheme (20) admits a conservative splitting into the nonoscillatory low-order part (41) and an antidiffusive correction:

$$M_C \frac{\mathrm{dU}}{\mathrm{d}t} = K\mathrm{U} + \mathrm{S}(\mathrm{U}) \quad \Leftrightarrow \quad M_L \frac{\mathrm{dU}}{\mathrm{d}t} = L\mathrm{U} + \mathrm{S}(\mathrm{U}) + \mathrm{F}(\mathrm{U}), \tag{47}$$

where $\mathrm{F}(\mathrm{U})$ is the vector of raw antidiffusive fluxes. By definition of M_L and D

$$\mathrm{F}_i = \sum_{j \neq i} \mathrm{F}_{ij}, \qquad \mathrm{F}_{ij} = m_{ij} \left(\frac{\mathrm{dU}_i}{\mathrm{d}t} - \frac{\mathrm{dU}_j}{\mathrm{d}t} \right) + \mathrm{D}_{ij}(\mathrm{U}_i - \mathrm{U}_j). \tag{48}$$

In the process of flux correction, F_i is replaced with its limited counterpart

$$\bar{\mathrm{F}}_i = \sum_{j \neq i} \bar{\mathrm{F}}_{ij}, \qquad \bar{\mathrm{F}}_{ij} := \alpha_{ij} \mathrm{F}_{ij}, \qquad 0 \leq \alpha_{ij} \leq 1. \tag{49}$$

In Section 9, we discuss various generalizations of scalar limiting techniques to systems. All of them produce a constrained semi-discrete problem of the form

$$M_L \frac{\mathrm{dU}}{\mathrm{d}t} = R(\mathrm{U}), \tag{50}$$

where $R(\mathrm{U}) = L\mathrm{U} + \mathrm{S}(\mathrm{U}) + \bar{\mathrm{F}}(\mathrm{U})$ incorporates the nonlinear antidiffusive correction.

Let U^n denote the vector of solution values at the time level $t^n = n\Delta t$, where Δt is a constant time step. Integration in time by the two-level θ-scheme yields

$$M_L \frac{\mathrm{U}^{n+1} - \mathrm{U}^n}{\Delta t} = \theta R(\mathrm{U}^{n+1}) + (1-\theta) R(\mathrm{U}^n), \tag{51}$$

where $\theta \in (0,1]$ is the implicitness parameter. In the fully discrete form of (48), the time derivative $\frac{\mathrm{dU}_i}{\mathrm{d}t}$ is replaced with $\frac{\mathrm{U}_i^{n+1} - \mathrm{U}_i^n}{\Delta t}$ and $\mathrm{D}_{ij}(\mathrm{U}_i - \mathrm{U}_j)$ becomes

$$\theta \mathrm{D}_{ij}^{n+1}(\mathrm{U}_i^{n+1} - \mathrm{U}_j^{n+1}) + (1-\theta)\mathrm{D}_{ij}^n(\mathrm{U}_i^n - \mathrm{U}_j^n).$$

The structure of the constrained flux $\bar{\mathrm{F}}_{ij}$ depends on the adopted limiting strategy.

7 Solution of Nonlinear Systems

Following a common practice [10, 14, 67], we linearize the contribution of $R(\mathrm{U}^{n+1})$ to the right-hand side of (51) about U^n using the Taylor series expansion

$$R(\mathrm{U}^{n+1}) \approx R(\mathrm{U}^n) + \left(\frac{\partial R}{\partial \mathrm{U}}\right)^n (\mathrm{U}^{n+1} - \mathrm{U}^n). \tag{52}$$

Plugging this approximation into (51), one obtains the linear algebraic system

$$\left[\frac{M_L}{\Delta t} - \theta \left(\frac{\partial R}{\partial \mathrm{U}}\right)^n\right](\mathrm{U}^{n+1} - \mathrm{U}^n) = R(\mathrm{U}^n). \tag{53}$$

If the steady-state solution is of interest, we use the backward Euler method ($\theta = 1$) and gradually increase the pseudo-time step Δt. When the solution begins to approach the steady state ($R(\mathrm{U}) = 0$) , the removal of the mass matrix can greatly speed up the convergence process since (53) reduces to Newton's method

$$-\left(\frac{\partial R}{\partial \mathrm{U}}\right)^n (\mathrm{U}^{n+1} - \mathrm{U}^n) = R(\mathrm{U}^n) \tag{54}$$

in the limit of infinitely large (pseudo)-time steps. On the other hand, removing the mass matrix too soon may have an adverse effect on the convergence rates [62].

Trépanier et al. [67] found it useful to freeze the Jacobian after the residuals reach a prescribed tolerance. This can significantly reduce the cost of matrix assembly.

Neglecting the nonlinearity of $L = K + D$, we approximate the Jacobian by [18]

$$\frac{\partial R}{\partial \mathrm{U}} \approx K + D + \frac{\partial \mathrm{S}}{\partial \mathrm{U}} + \frac{\partial \bar{\mathrm{F}}}{\partial \mathrm{U}}. \tag{55}$$

If the blocks of the Galerkin transport operator K are defined by (23), the use of $K(\mathrm{U}^n)$ instead of $K(\mathrm{U}^{n+1})$ boils down to replacing the flux $\mathbf{F}_j = \mathbf{A}_j^{n+1}\mathrm{U}_j^{n+1}$ with the flux $\mathbf{F}_j = \mathbf{A}_j^{n}\mathrm{U}_j^{n+1}$. Thus, the above linearization about U^n is conservative.

Since the vector of boundary fluxes $\mathrm{S}(\mathrm{U})$ depends on the solution of a Riemann problem, its differentiation is a rather laborious process. For details, we refer to Gurris [18] who derived a formula for $\frac{\partial \mathrm{S}}{\partial \mathrm{U}}$ using a repeated application of the chain rule. His numerical study indicates that the implicit treatment of the weakly imposed boundary conditions makes it possible to achieve unconditional stability.

The use of a non-differentiable flux limiter rules out the derivation of closed-form expressions for $\frac{\partial \mathrm{F}}{\partial \mathrm{U}}$. In principle, the antidiffusive term can be differentiated numerically using finite differencing [50, 51]. However, the significant overhead cost and the sensitivity to the choice of the free parameter restrict the practical utility of this approach. Moreover, the resultant matrix is not as sparse as the low-order Jacobian since the use of limiters widens the computational stencils. For this reason, we currently favor a semi-explicit treatment of limited antidiffusion.

Instead of linearizing the nondifferentiable antidiffusive term about U^n, one can update it in an iterative fashion. Given an approximate solution $\mathrm{U}^{(m)} \approx \mathrm{U}^{n+1}$ to (53), a new approximation $\mathrm{U}^{(m+1)}$ is obtained by solving the linear system

$$J(\mathrm{U}^{(m)})(\mathrm{U}^{(m+1)} - \mathrm{U}^n) = R(\mathrm{U}^n) + \theta(\bar{\mathrm{F}}(\mathrm{U}^{(m)}) - \bar{\mathrm{F}}(\mathrm{U}^n)), \tag{56}$$

$$J(\mathrm{U}) = \tfrac{M_L}{\Delta t} - \theta\left(L(\mathrm{U}) + \tfrac{\partial \mathrm{S}}{\partial \mathrm{U}}\right). \tag{57}$$

Due to the semi-explicit treatment of $\bar{\mathrm{F}}(\mathrm{U}^{n+1})$, the so-defined defect correction scheme may converge rather slowly. However, it can be converted into a quasi-Newton method using the Anderson convergence acceleration technique [26].

The repeated evaluation of the antidiffusive term can be avoided using a linearization about the solution of the low-order system. This predictor-corrector strategy is appropriate if the transient flow behavior dictates the use of small time steps. In this case, the following algorithm [28, 33] is a cost-effective alternative to (53)

1. Calculate the end-of-step solution $\mathrm{U}^L \approx \mathrm{U}^{n+1}$ to the low-order system

$$J(\mathrm{U}^n)(\mathrm{U}^L - \mathrm{U}^n) = L(\mathrm{U}^n)\mathrm{U}^n + \mathrm{S}(\mathrm{U}^n). \tag{58}$$

2. Calculate the vector of raw antidiffusive fluxes F_{ij} linearized about U^L

$$\mathrm{F}_{ij} = m_{ij}\left(\dot{\mathrm{U}}_i^L - \dot{\mathrm{U}}_j^L\right) + \mathrm{D}_{ij}(\mathrm{U}_i^L - \mathrm{U}_j^L), \tag{59}$$

   where $\dot{\mathrm{U}}_i^L$ is a low-order approximation to the time-derivative at node i

$$\dot{\mathrm{U}}^L = M_L^{-1}[L(\mathrm{U}^L)\mathrm{U}^L + \mathrm{S}(\mathrm{U}^L)]. \tag{60}$$

3. Apply the flux limiter and calculate the final solution U^{n+1}

$$\mathrm{U}_i^{n+1} = \mathrm{U}_i^L + \frac{1}{m_i}\sum_{j\neq i}\bar{\mathrm{F}}_{ij}. \tag{61}$$

8 Solution of Linear Systems

In the 3D case, there are 5 unknowns (density, 3 momentum components, and energy) per mesh node. Hence, each linear system to be solved can be written as

$$\begin{bmatrix} J_{11} & J_{12} & J_{13} & J_{14} & J_{15} \\ J_{21} & J_{22} & J_{23} & J_{24} & J_{25} \\ J_{31} & J_{32} & J_{33} & J_{34} & J_{35} \\ J_{41} & J_{42} & J_{43} & J_{44} & J_{45} \\ J_{51} & J_{52} & J_{53} & J_{54} & J_{55} \end{bmatrix} \begin{bmatrix} \Delta u_1 \\ \Delta u_2 \\ \Delta u_3 \\ \Delta u_4 \\ \Delta u_5 \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \end{bmatrix}. \tag{62}$$

Simultaneous update of all variables is costly in terms of CPU time and memory requirements. The coupled system can be split into smaller subproblems using an iterative method of block-Jacobi or block-Gauss-Seidel type. In the former case, the new value of Δu_k is calculated using Δu_l from the last outer iteration:

$$J_{kk}\Delta u_k^{(m+1)} = r_k - \sum_{k\neq l} J_{kl}\Delta u_l^{(m)}, \qquad \Delta u^{(0)} := 0, \tag{63}$$

where m is the iteration counter and k is the subproblem index. Replacing $\Delta u_l^{(m)}$ with $\Delta u_l^{(m+1)}$ for $l < k$, one obtains the block-Gauss-Seidel method

$$J_{kk}\Delta u_k^{(m+1)} = r_k - \sum_{l>k} J_{kl}\Delta u_l^{(m)} - \sum_{l<k} J_{kl}\Delta u_l^{(m+1)}. \tag{64}$$

This segregated solution strategy is easy to implement but may require many iterations per time step. A more robust iterative solver for (62) can be designed using a Krylov-subspace or multigrid method equipped with a smoother/preconditioner that involves solution of small coupled problems on elements/patches. In the next chapter, we will use such a method to solve the discrete saddle point problem for the finite element discretization of the incompressible Navier-Stokes equations.

9 Flux Limiting for Systems

The design of flux limiters for hyperbolic systems is more involved than that for scalar conservation laws. If the density, momentum, and energy increments are limited separately, undershoots/overshoots are likely to arise in all quantities of interest. The following remedies to this problem have been proposed [41, 42, 73, 74, 76]

- synchronization of the correction factors for selected control variables;
- transformations to nonconservative (primitive, characteristic) variables;
- a posteriori control and postprocessing of the flux-corrected solution.

In the synchronized version of the FCT limiter [42, 41], all components of the raw antidiffusive flux F_{ij} are multiplied by the same correction factor α_{ij}. No syn-

chronization of α_{ij} is required if a transformation to the local characteristic variables is performed. However, this sort of flux correction is computationally expensive and requires dimensional splitting for the diffusive and antidiffusive fluxes.

Limiters that constrain the primitive (density, velocity, pressure) or characteristic variables are typically quite reliable but the involved linearizations may also cause them to fail, no matter how carefully they are designed. While it is impossible to rule out the formation of spurious maxima and minima a priori, they can be easily detected and removed at a postprocessing step. This approach was introduced by Zalesak [76] who used it to maintain the nonnegativity of pressures and internal energies in a characteristic FCT method for the compressible Euler equations.

9.1 Transformation of Variables

We begin with the presentation of a symmetric limiter for a general set of dependent quantities. In classical high-resolution schemes for the Euler equations, the required transformations between the conservative and nonconservative variables are usually performed edge-by-edge [40, 73, 74, 76]. The solution-dependent transformation matrix $\mathrm{T}_{ij} = \mathrm{T}_{ji}$ is evaluated using a suitably defined average of U_i and U_j.

A very general limiting strategy for systems was proposed by Löhner [40]. Given a tentative solution U and the corresponding vector of raw antidiffusive fluxes

$$\mathrm{F}_{ij} = [f_{ij}^{\rho}, \mathbf{f}_{ij}^{\rho v}, f_{ij}^{\rho E}]^T, \tag{65}$$

the following algorithm can be used to calculate the synchronized correction factors α_{ij} for a given set of possibly nonconservative control variables:

1. Initialize the three auxiliary arrays for the generalized Zalesak limiter

$$P_i^{\pm} := 0, \quad Q_i^{\pm} := 0, \quad R_i^{\pm} := 1. \tag{66}$$

2. For each pair of neighbor nodes, perform the local change of variables

$$\hat{\mathrm{F}}_{ij} := \mathrm{T}_{ij}\mathrm{F}_{ij}, \qquad \Delta\mathrm{W}_{ij} := \mathrm{T}_{ij}(\mathrm{U}_j - \mathrm{U}_i), \tag{67}$$

$$\hat{\mathrm{F}}_{ji} := -\hat{\mathrm{F}}_{ij}, \qquad \Delta\mathrm{W}_{ji} := -\Delta\mathrm{W}_{ij}. \tag{68}$$

3. Update the sums of positive/negative components to be limited

$$P_{i,k}^{\pm} := P_{i,k}^{\pm} + \min_{\max}\{0, \hat{f}_{ij}^k\}, \qquad P_{j,k}^{\pm} := P_{j,k}^{\pm} + \min_{\max}\{0, \hat{f}_{ji}^k\}. \tag{69}$$

4. Update the upper/lower bounds for the sum of limited increments

$$Q_{i,k}^{\pm} := \min_{\max}\left\{Q_{i,k}^{\pm}, \Delta w_{ij}^k\right\}, \qquad Q_{j,k}^{\pm} := \min_{\max}\left\{Q_{j,k}^{\pm}, \Delta w_{ji}^k\right\}. \tag{70}$$

5. Calculate the nodal correction factors for positive/negative edge contributions

$$R_{i,k}^{\pm} = \min\left\{1, \frac{\gamma_i Q_{i,k}^{\pm}}{P_{i,k}^{\pm}}\right\}, \tag{71}$$

where $\gamma_i > 0$ is a positive scaling factor ($\gamma_i = m_i/\Delta t$ for generalized FCT).

6. Determine the edge correction factors for the given quantity of interest

$$\alpha_{ij}^k = \min\{R_{ij}^k, R_{ji}^k\}, \qquad R_{ij}^k = \begin{cases} R_{i,k}^+, & \text{if } \hat{f}_{ij}^k \geq 0, \\ R_{i,k}^-, & \text{if } \hat{f}_{ij}^k < 0. \end{cases} \tag{72}$$

7. Multiply all components of F_{ij} and F_{ji} by the synchronized correction factor

$$\alpha_{ij} = \min_k \alpha_{ij}^k. \tag{73}$$

Instead of calculating α_{ij}^k independently and taking the minimum, one can redefine α_{ij}^k as the correction factor for the raw antidiffusive flux [33]

$$\mathrm{F}_{ij}^k := \alpha_{ij}^{k-1} \mathrm{F}_{ij}^{k-1}. \tag{74}$$

This sequential limiting procedure amounts to the multiplication of $\mathrm{F}_{ij}^0 := \mathrm{F}_{ij}$ by

$$\alpha_{ij} = \alpha_{ij}^k \cdot \alpha_{ij}^{k-1} \cdot \ldots \cdot \alpha_{ij}^1. \tag{75}$$

In contrast to (73), the result depends on the order in which the correction factors α_{ij}^k are calculated. However, the raw antidiffusive fluxes (74) already include the net effect of previous corrections, which makes the limiter less diffusive.

In our experience, averaging across shocks and contact discontinuities may give rise to unbounded solutions in some particularly sensitive problems. This has led us to prefer a node-based approach to the transformation of variables for the synchronized flux limiter [33]. In the revised version, we replace (67) and (68) with

$$\hat{\mathrm{F}}_{ij} := \mathrm{T}_i \mathrm{F}_{ij}, \qquad \Delta \mathrm{W}_{ij} := \mathrm{T}_j \mathrm{U}_j - \mathrm{T}_i \mathrm{U}_i, \tag{76}$$

$$\hat{\mathrm{F}}_{ji} := -\mathrm{T}_j \mathrm{F}_{ij}, \qquad \Delta \mathrm{W}_{ji} := -\Delta \mathrm{W}_{ij}. \tag{77}$$

Since the transformation matrices T_i and T_j are generally different, the transformed antidiffusive fluxes are no longer skew-symmetric, i.e., $\hat{\mathrm{F}}_{ji} \neq -\hat{\mathrm{F}}_{ij}$. However, the flux-limited scheme remains conservative since the synchronized correction factor α_{ij} is applied to the vector of original fluxes (65). It is neither necessary nor desirable to require that the increments to nonconservative variables be skew-symmetric.

The node-based approach makes the limiter more robust. First, the transformation matrix T_i is the same for all antidiffusive fluxes into node i. Second, the upper and lower bounds are defined using the correct nodal values of the nonconservative variables. Moreover, the revised algorithm requires less arithmetic operations.

9.2 Limiting Primitive Variables

In this section, we describe the synchronized FCT limiter with node-based transformations to the primitive variables. The flux-corrected solution is given by

$$m_i \mathrm{U}_i = m_i \mathrm{U}_i^L + \sum_{j \neq i} \alpha_{ij} \mathrm{F}_{ij}, \tag{78}$$

where U^L denotes the low-order predictor. To calculate α_{ij}, we define [33]

$$\mathbf{v}_i = \frac{(\rho \mathbf{v})_i}{\rho_i}, \qquad p_i = (\gamma - 1) \left[(\rho E)_i - \frac{|(\rho \mathbf{v})_i|^2}{2\rho_i} \right], \tag{79}$$

$$\mathbf{f}_{ij}^v = \frac{\mathbf{f}_{ij}^{\rho v} - \mathbf{v}_i f_{ij}^\rho}{\rho_i}, \qquad f_{ij}^p = (\gamma - 1) \left[f_{ij}^{\rho E} + \frac{|\mathbf{v}_i|^2}{2} f_{ij}^\rho - \mathbf{v}_i \cdot \mathbf{f}_{ij}^{\rho v} \right]. \tag{80}$$

Let u_i^L be the low-order approximation to ρ, v, or p. The raw antidiffusive 'flux' from node j into node i is denoted by f_{ij}^u. In accordance with the FCT philosophy, the choice of the correction factor α_{ij}^u must ensure that the limited antidiffusive correction does not increase the local maxima and minima of u^L. The node-based approach to computation of α_{ij}^u involves the following algorithmic steps [33]:

1. Compute the sums of positive/negative antidiffusive increments to node i

$$P_i^+ = \sum_{j \neq i} \max\{0, f_{ij}^u\}, \qquad P_i^- = \sum_{j \neq i} \min\{0, f_{ij}^u\}. \tag{81}$$

2. Compute the distance to a local maximum/minimum of the low-order solution

$$Q_i^+ = u_i^{\max} - u_i^L, \qquad Q_i^- = u_i^{\min} - u_i^L. \tag{82}$$

3. Compute the nodal correction factors for the net increment to node i

$$R_i^\pm := \min \left\{ 1, \frac{m_i Q_i^\pm}{\Delta t P_i^\pm} \right\}. \tag{83}$$

4. Define $\alpha_{ij}^u = \alpha_{ji}^u$ so as to satisfy the LED constraints for nodes i and j

$$\alpha_{ij}^u = \min\{R_{ij}, R_{ji}\}, \qquad R_{ij} = \begin{cases} R_i^+, & \text{if } f_{ij}^u \geq 0, \\ R_i^-, & \text{if } f_{ij}^u < 0. \end{cases} \tag{84}$$

If all primitive variables are selected for limiting, the synchronized correction factor α_{ij} for the explicit solution update (78) can be defined as [32, 41, 42]

$$\alpha_{ij} = \min\{\alpha_{ij}^\rho, \alpha_{ij}^v, \alpha_{ij}^p\} \tag{85}$$

or

$$\alpha_{ij} = \alpha_{ij}^\rho \alpha_{ij}^v \alpha_{ij}^p. \tag{86}$$

In the multidimensional case, small velocity fluctuations in the crosswind direction may result in the cancellation of the entire flux. To avoid this, we set $\alpha_{ij}^{v} := 1$ or define α_{ij}^{v} as the correction factor for the streamline velocity [33].

Since the change of variables in (79) and (80) involves a linearization about U_i^L, there is no guarantee that the flux-corrected solution given by (61) will stay within the original bounds, especially in the presence of large jumps. Therefore, our FCT limiting strategy includes a postprocessing step in which all undershoots and overshoots are detected and removed. The first 'failsafe' flux limiter of this kind was proposed by Zalesak (see [76], pp. 36 and 56). His recipe is very simple: "if, after flux limiting, either the density or the pressure in a cell is negative, all the fluxes into that cell are set to their low order values, and the grid point values are recalculated." It is tacitly assumed that the low-order solution is free of nonphysical values.

A similar approach can be used to enforce **local** FCT constraints in a failsafe manner [33]. The flux-corrected value u_i of the control variable u is acceptable if

$$u_i^{\min} \leq u_i \leq u_i^{\max}. \tag{87}$$

If any quantity of interest (density, velocity, pressure) has an undershoot/overshoot at node i, then a fixed percentage of the added antidiffusive fluxes $\alpha_{ij}\mathrm{F}_{ij}$ and $\alpha_{ji}\mathrm{F}_{ji}$ is removed until the offense is eliminated [33]. The number of correction cycles N depends on the effort invested in the calculation of α_{ij}. If the synchronized FCT limiter is applied to all primitive variables, then undershoots and overshoots are an exception, so that $N = 1$ is optimal. On the other hand, 3-5 cycles may be appropriate in the case $\alpha_{ij} = \alpha_{ij}^{\rho}$ or $\alpha_{ij} = \alpha_{ij}^{p}$. The choice of N affects only the amount of rejected antidiffusion. The bounds of the low-order solution are guaranteed to be preserved even for $\alpha_{ij} \equiv 1$. Hence, the failsafe corrector can not only reinforce but also replace the synchronized FCT limiter, as demonstrated by the numerical study in [33].

9.3 Limiting Characteristic Variables

The idea of flux limiting in terms of local characteristic variables dates back to the work of Yee et al. [73, 74] on total variation diminishing (TVD) schemes for the Euler equations. The traditional approach to implementation of such high-resolution schemes in edge-based finite element codes is based on the reconstruction of local 1D stencils [2, 9, 40, 46, 57, 58]. The development of a genuinely multidimensional characteristic limiter is complicated by the fact that the eigenvalues and eigenvectors of the Jacobian matrices $\mathbf{e}_{ij} \cdot \mathbf{A}_{ij}$ depend on the orientation of $\mathbf{e}_{ij}$, whereas all components of the sums $P_i^{\pm}$ must correspond to the same set of local characteristic variables. For this reason, we use artificial viscosity of the form (43) and limit the antidiffusive fluxes associated with each coordinate direction independently.

In contrast to the synchronized FCT algorithm for primitive variables, it is worthwhile to use different correction factors for different waves. In this case, an edge-based transformation of variables is required to keep the scheme conservative.

The multiplication by the matrix of left eigenvectors $\mathrm{L}_{ij} = \mathrm{R}_{ij}^{-1}$ of a directional Jacobian A_{ij}^d, $1 \le d \le n$ transforms $\mathrm{U}_j - \mathrm{U}_i$ into the characteristic difference

$$\Delta \mathrm{W}_{ij} = \mathrm{R}_{ij}^{-1}(\mathrm{U}_j - \mathrm{U}_i).$$

Since the local characteristic variables are essentially decoupled, the components of $\Delta \mathrm{W}_{ij}$ can be limited separately. If a one-sided limiting strategy is adopted, the sign of the eigenvalue λ_k determines the upwind direction for the k-th wave. Let

$$I = \begin{cases} i, & \text{if } \lambda_k \ge 0, \\ j, & \text{if } \lambda_k < 0. \end{cases} \tag{88}$$

In the process of flux limiting, a nodal correction factor $R_{I,k}^{\pm}$ is applied to Δw_{ij}^k

$$\widehat{\Delta w_{ij}^k} = \begin{cases} R_{I,k}^{+} \Delta w_{ij}^k, & \text{if } \Delta w_{ij}^k \le 0, \\ R_{I,k}^{-} \Delta w_{ij}^k, & \text{if } \Delta w_{ij}^k > 0. \end{cases} \tag{89}$$

The multiplication by the matrix of right eigenvectors transforms the remaining artificial viscosity (if any) to the conservative variables. The flux to be added is

$$\Phi(e_{ij}^d \mathrm{A}_{ij}^d, \mathrm{U}_j - \mathrm{U}_i) := |e_{ij}^d| \mathrm{R}_{ij} |\Lambda_{ij}| (\Delta \mathrm{W}_{ij} - \widehat{\Delta \mathrm{W}}_{ij}). \tag{90}$$

Clearly, the use of dimensional splitting makes this sort of algebraic flux correction more expensive than the synchronized FCT algorithm. However, flux limiting in terms of local characteristic variables is very reliable and produces accurate results. We refer to Zalesak [76] for a presentation of characteristic FCT limiters.

10 Constrained Initialization

The initialization of data is an important ingredient of numerical algorithms for systems of conservation laws. If the initial data are prescribed analytically, it is essential to guarantee that the numerical solution has the right total mass, momentum, and energy when the simulation begins. The pointwise definition of nodal values

$$U_i^0 = U_0(\mathbf{x}_i) \tag{91}$$

is generally nonconservative. This may result in significant errors if the computational mesh is too coarse in regions where U_0 is discontinuous. On the other hand, conservative high-order projections tend to produce undershoots and overshoots.

The first use of FCT in the context of constrained data projection (initialization, interpolation, remapping) dates back to the work of Smolarkiewicz and Grell [63] who introduced a class of nonconservative monotone interpolation schemes. Conservative FCT interpolations were employed by Váchal and Liska [69] and Liska

et al. [39]. Farrell et al. [12] introduced a bounded L^2 projection operator for globally conservative interpolation between unstructured meshes. In the monograph by Löhner ([40], pp. 257–260), the FCT limiter is applied to the difference between the consistent and lumped-mass L^2 projections. The latter serves as the low-order method that satisfies the maximum principle for linear finite elements [12].

A general approach to synchronized FCT projections for systems of conserved variables was presented in [33]. Let U denote the initial data or a numerical solution from an arbitrary finite element space. The standard L^2 projection is defined by

$$\int_\Omega w_h U_h^H \,\mathrm{d}\mathbf{x} = \int_\Omega w_h U \,\mathrm{d}\mathbf{x}, \qquad \forall w_h. \tag{92}$$

The nodal values of the high-order approximation U_h^H satisfy the linear system

$$M_C \mathrm{U}^H = \mathrm{R}, \tag{93}$$

where $M_C = \{m_{ij}\mathrm{I}\}$ is the consistent mass matrix and R is the load vector

$$\mathrm{R}_i = \int_\Omega \varphi_i U \,\mathrm{d}\mathbf{x}. \tag{94}$$

If the functions φ_i and U are defined on different meshes, numerical integration can be performed using a *supermesh* that represents the union of the two meshes [12].

The lumped-mass approximation to (93) is a linear system with a diagonal matrix

$$M_L \mathrm{U}^L = \mathrm{R}. \tag{95}$$

The so-defined low-order solution U_h^L has the same 'mass' as U_h^H but is free of undershoots and overshoots, at least in the case of linear finite elements [12].

The difference between U_i^H and U_i^L admits the conservative flux decomposition

$$\mathrm{U}_i^H = \mathrm{U}_i^L + \sum_{j\neq i} \mathrm{F}_{ij}, \qquad \mathrm{F}_{ij} = m_{ij}(\mathrm{U}_i^H - \mathrm{U}_j^H). \tag{96}$$

The process of flux limiting involves the same algorithmic steps as the FEM-FCT scheme for the Euler equations. The use of failsafe postprocessing is optional.

11 Boundary Conditions

The implementation of boundary conditions for the Euler equations is an issue of utmost importance. The solution to a hyperbolic system is a superposition of several waves traveling in certain directions at finite speeds. Hence, the proper choice of boundary conditions depends on the wave propagation pattern [17, 24, 61, 71]. In this section, we review the underlying theory and discuss the numerical treatment of characteristic boundary conditions in an implicit finite element formulation.

11.1 Physical Boundary Conditions

The number of physical boundary conditions (PBC) to be imposed is determined using a transformation to the local characteristic variables associated with the unit outward normal $\mathbf{n}$. The result is a set of five decoupled convection equations

$$\frac{\partial w_k}{\partial t} + \lambda_k \frac{\partial w_k}{\partial n} = 0, \qquad k = 1, \ldots, 5, \tag{97}$$

where w_k are the so-called *Riemann invariants* and λ_k are the eigenvalues of the directional Jacobian $\mathbf{n} \cdot \mathbf{A}$. The matrix-vector form of system (97) reads

$$\frac{\partial W}{\partial t} + \Lambda \frac{\partial W}{\partial n} = 0. \tag{98}$$

The matrix $\Lambda = \text{diag}\{\lambda_1, \ldots, \lambda_5\}$ and vector $W = [w_1, \ldots, w_5]^T$ are given by [71]

$$\Lambda = \text{diag}\{v_n - c, v_n, v_n, v_n, v_n + c\} \tag{99}$$

and

$$W = \left[v_n - \frac{2c}{\gamma - 1}, s, v_\xi, v_\eta, v_n + \frac{2c}{\gamma - 1} \right]^T. \tag{100}$$

Here $v_n = \mathbf{n} \cdot \mathbf{v}$ is the normal velocity, v_ξ and v_η are the two components of the tangential velocity $\boldsymbol{\tau} \cdot \mathbf{v}$, c is the speed of sound, and $s = c_v \log\left(\frac{p}{\rho^\gamma}\right)$ is the entropy.

Since the evolution of the Riemann invariants is governed by pure convection equations, a boundary condition is required for each incoming wave. Hence, the number of PBC equals the number of negative eigenvalues N_λ. By virtue of (99), the sign of λ_k depends on v_n, as well as on the local Mach number

$$M = \frac{|v_n|}{c}.$$

The following types of boundaries may need to be considered when it comes to formulating a well-posed boundary-value problem for the Euler equations:

- Supersonic inlet: $v_n < 0,\ M > 1$. All eigenvalues are negative, so $N_\lambda = 5$.
- Supersonic outlet: $v_n > 0,\ M > 1$. All eigenvalues are positive, so $N_\lambda = 0$.
- Subsonic inlet: $v_n < 0,\ M < 1$. Only $\lambda_5 = v_n + c$ is nonnegative, so $N_\lambda = 4$.
- Subsonic outlet: $v_n > 0,\ M < 1$. Only $\lambda_1 = v_n - c$ is negative, so $N_\lambda = 1$.
- Solid wall boundary: $v_n = M = 0$. Only $\lambda_1 = -c$ is negative, so $N_\lambda = 1$

In many cases, the N_λ boundary conditions are given in terms of the conservative or primitive variables. It is also possible to prescribe the total enthalpy, entropy, temperature, or inclination angle. These data define the "free stream" state U_∞ for the computation of the Dirichlet/Neumann boundary conditions (14) and (15).

11.2 Numerical Boundary Conditions

The need for numerical boundary conditions (NBC) arises whenever $0 < N_\lambda < 5$ so that the boundary values and normal fluxes cannot be determined using the prescribed PBC alone. The missing information is obtained by solving a Riemann problem. The internal state U is defined as the numerical solution to the Euler equations at the given point. The external state U_∞ can be obtained as follows [14, 61, 71]:

1. Convert the given numerical solution U to the Riemann invariants W.
2. Set $W_\infty := W$ and overwrite the incoming Riemann invariants by PBC.
3. Given the modified vector W_∞, calculate the free stream values U_∞.

In contrast to cell-centered finite volume methods, there is no need for extrapolation because the values of U_h are readily available at each boundary point.

The right-hand side $G(U,U_\infty)$ of the Dirichlet boundary condition (14) is defined as the exact or approximate solution to the boundary Riemann problem associated with the states U and U_∞. Likewise, the normal flux $F_n(U,U_\infty)$ for the Neumann boundary condition (14) can be calculated using Toro's [65] exact Riemann solver or Roe's approximate Riemann solver [55]. The latter approach yields

$$F_n(U,U_\infty) = \mathbf{n}\cdot\frac{\mathbf{F}(U)+\mathbf{F}(U_\infty)}{2} - \frac{1}{2}|\mathbf{n}\cdot\mathbf{A}(U,U_\infty)|(U_\infty - U), \tag{101}$$

where $\mathbf{A}(U,U_\infty)$ is the Roe matrix for the states U and U_∞. This approach to weak imposition of characteristic boundary conditions is closely related to their numerical treatment in finite volume and discontinuous Galerkin methods [10, 67].

11.3 Practical Implementation

In a practical implementation, it is worthwhile to initialize W_∞ by the vector of free stream values and overwrite the Riemann invariants associated with nonnegative eigenvalues by the corresponding components of W. Such an algorithm is well-suited for boundaries of any type since it determines the direction of wave propagation and the upstream values of the characteristic variables automatically.

The transformation of the internal state U to the vector of Riemann invariants W is performed using definition (100). The inverse transformation is given by [61, 71]

$$\rho = \left[\frac{c^2}{\gamma}\exp\left(-\frac{w_2}{c_v}\right)\right]^{\frac{1}{\gamma-1}}, \tag{102}$$

$$\rho\mathbf{v} = \rho(v_n\mathbf{n} + v_\xi\boldsymbol{\tau}_\xi + v_\eta\boldsymbol{\tau}_\eta), \tag{103}$$

$$\rho E = \frac{p}{\gamma-1} + \frac{\rho}{2}(v_n^2 + v_\xi^2 + v_\eta^2), \tag{104}$$

where $\boldsymbol{\tau}_\xi$ and $\boldsymbol{\tau}_\eta$ are two unit vectors spanning the tangential plane, and

$$v_n = \frac{w_5 - w_1}{2}, \quad v_\xi = w_3, \quad v_\eta = w_4,$$
$$c = \frac{\gamma - 1}{4}(w_5 - w_1), \qquad p = \frac{\rho c^2}{\gamma}.$$

If the physical boundary conditions are given in terms of primitive variables or other quantities, a conversion to the Riemann invariants is required. The practical implementation of such boundary conditions depends on the type of the boundary.

11.3.1 Open Boundary Conditions

At a supersonic inlet, the free stream values of the conservative variables U_∞ can be prescribed without transforming to the Riemann invariants. At a supersonic outlet, the exterior state is given by $U_\infty = U$ so that the Roe flux (101) reduces to

$$F_n(U,U) = \mathbf{n} \cdot \mathbf{F}(U).$$

At a subsonic inlet, it is common to prescribe the density ρ_{in}, pressure p_{in}, and tangential velocity $\boldsymbol{\tau} \cdot \mathbf{v}_{in}$. In this case, the Riemann invariants w_3 and w_4 are given, whereas $w_2 = c_v \log(\frac{p_{in}}{\rho_{in}^\gamma})$ is computable. The last incoming Riemann invariant is

$$w_1 = w_5 - \frac{4}{\gamma - 1}\sqrt{\frac{\gamma p_{in}}{\rho_{in}}}. \tag{105}$$

In the case of a subsonic outlet with a prescribed exit pressure p_{out}, we have [61]

$$w_1 = w_5 - \frac{4}{\gamma - 1}\sqrt{\frac{\gamma p_{out}}{\rho_{out}}}, \tag{106}$$

where ρ_{out} depends on the calculated interior density ρ and pressure p as follows:

$$\rho_{out} = \rho \left(\frac{p_{out}}{p} \right)^{\frac{1}{\gamma}}.$$

The outgoing Riemann invariant w_5 is evaluated using the trace of the finite element solution. The open boundary conditions (105) and (106) are generally regarded as more physical than a prescribed upstream value of the Riemann invariant w_1.

11.3.2 Wall Boundary Conditions

At a solid surface, there is no convective flux across the boundary. Hence, the normal velocity v_n must vanish. The so-defined *no-penetration / free slip* condition

$$\mathbf{n} \cdot \mathbf{v} = 0 \tag{107}$$

constrains a linear combination of the three velocity components. The numerical implementation of this condition in an implicit scheme presents a considerable difficulty if the boundary is not aligned with the axes of the coordinate system.

In finite element methods for incompressible flow problems, the free slip condition (107) is usually imposed in the strong sense using element-by-element transformations to a local reference frame spanned by the normal and tangential vectors [7, 11]. The same effect can be achieved using an iterative projection of the velocity vector on the tangential plane [32]. However, the semi-explicit treatment of the wall boundary condition slows down the iterative solver and may result in a lack of robustness. Therefore, a fully implicit treatment is to be preferred.

In the weak form of the free slip condition, the free stream values for the computation of $F_n(U,U_\infty)$ are calculated using the mirror (reflection) condition

$$\mathbf{n}\cdot(\mathbf{v}_\infty+\mathbf{v})=0.$$

The density, tangential velocity, and total energy remain unchanged. Thus

$$U_\infty=\begin{bmatrix}\rho\\ \rho\mathbf{v}_\infty\\ \rho E\end{bmatrix},\qquad \mathbf{v}_\infty=\mathbf{v}-2\mathbf{n}(\mathbf{v}\cdot\mathbf{n}).$$

Another popular weak form of the zero flux boundary condition is given by

$$F_n=\begin{bmatrix}0\\ \mathbf{n}p\\ 0\end{bmatrix}. \tag{108}$$

This version does not involve the solution of a Riemann problem and has been used in FEM codes with considerable success [4, 10, 60]. However, the Roe flux (101) constitutes a more physical wall boundary condition than (108). In any case, the weak imposition of the free slip condition may give rise to a nonzero normal velocity on the wall. This problem can be fixed by adding a penalty term [18].

11.3.3 Calculation of Surface Integrals

The imposition of natural boundary conditions requires the evaluation of the numerical flux $F_n(U,U_\infty)$ at each quadrature point $\hat{\mathbf{x}}_i$. The exterior state U_∞ is associated with a ghost node $\hat{\mathbf{x}}_{i,\infty}$ located on the other side of the boundary. The ghost nodes provide the free stream values of the Riemann invariants and play the same role as *image cells* in cell-centered finite volume schemes for the Euler equations [67].

If a curved boundary is approximated using isoparametric (linear or bilinear) finite elements, then the normal vector $\mathbf{n}$ is generally discontinuous at the vertices and edges of the surface triangulation. The boundary integrals can be assembled element-by-element using the unique normal to the boundary of each cell [18]. However, the value of $F_n(U,U_\infty)$ at $\hat{\mathbf{x}}_i$ should be obtained by interpolating the (unique)

nodal values to ensure consistency with the group FEM approximation (18). Otherwise, numerical side effects may arise in the boundary layer and pollute the solution in the interior of the computational domain. As a remedy, a unique normal direction can be determined using a suitable averaging procedure [11, 50] or an analytical description of the curved boundary. For a detailed discussion of solid wall boundary conditions in curved geometries, we refer to Krivodonova and Berger [25].

12 Numerical Examples

The results presented in this section illustrate some properties of our algebraic flux correction schemes for the Euler equations. We consider a suite of 2D benchmark problems covering a relatively wide range of Mach numbers and boundary conditions. The objective of the below numerical study is to investigate the dependence of the error on the mesh size h and on the choice of the limiting strategy.

The accuracy of a numerical solution $u_h \approx u$ is measured in the global norms

$$E_1(u,h) = \sum_i m_i |u(\mathbf{x}_i) - u_i| \approx \|u - u_h\|_1, \tag{109}$$

$$E_2(u,h) = \sqrt{\sum_i m_i |u(\mathbf{x}_i) - u_i|^2} \approx \|u - u_h\|_2. \tag{110}$$

The rate of grid convergence is illustrated by the expected order of accuracy

$$p = \log_2 \left(\frac{E_i(u,2h)}{E_i(u,h)} \right), \qquad i = 1,2. \tag{111}$$

To begin with, we will evaluate the performance of the linearized FCT algorithm for unsteady compressible flow problems [28]. Next, we will investigate the convergence behavior of a characteristic TVD-like limiter for steady-state computations [29]. In this work, stationary solutions are obtained using pseudo-time-stepping. For additional numerical examples, the interested reader is referred to [18, 33].

12.1 Shock Tube Problem

Sod's shock tube problem [64] is a standard benchmark for the unsteady Euler equations. The domain $\Omega = (0,1)$ is initially separated by a membrane into two sections. When the membrane is removed, the gas begins to flow into the region of lower pressure. The initial condition for the nonlinear Riemann problem is given by

$$\begin{bmatrix} \rho_L \\ \mathbf{v}_L \\ p_L \end{bmatrix} = \begin{bmatrix} 1.0 \\ 0.0 \\ 1.0 \end{bmatrix}, \qquad \begin{bmatrix} \rho_R \\ \mathbf{v}_R \\ p_R \end{bmatrix} = \begin{bmatrix} 0.125 \\ 0.0 \\ 0.1 \end{bmatrix}, \tag{112}$$

where the subscripts refer to the subdomains $\Omega_L = (0, 0.5)$ and $\Omega_R = (0.5, 1)$. The reflective wall boundary conditions are prescribed at the endpoints of Ω.

The removal of the membrane at $t = 0$ releases a shock wave that propagates 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. 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 traveling at different speeds [35].

The dashed lines in Fig. 1 show the exact solution to the Riemann problem (112) at the final time $T = 0.231$. This solution was calculated using the exact Riemann solver HE-E1RPEXACT [66]. The numerical solution U_h^0 was initialized by means of the FCT-constrained data projection (96) and advanced in time by the semi-implicit Crank-Nicolson scheme with the time step $\Delta t = h/10$. For each algorithm under consideration, a grid convergence study was performed on a sequence of uniform grids with mesh spacing $h = 1/N$ for $N = 100$, 200, 400, 800, 1600, 3200.

All numerical solutions shown in Fig. 1a-h were calculated on a uniform mesh of 100 linear finite elements. The results produced by the low-order schemes ($\alpha_{ij} = 0$) are nonosillatory but the excess numerical diffusion gives rise to strong smearing of the moving fronts. In this example, Roe's approximate Riemann solver (Fig. 1a) performs slightly better than the Rusanov scalar dissipation (Fig. 1b).

The linearized FCT algorithm (58)–(61) produced the snapshots displayed in Figs. 1c-f. In this study, the correction factors α_{ij} were calculated via sequential limiting of the control variables listed in the parentheses. Similar results were obtained with synchronization of the form (73). The computation of the low-order predictor using Roe's formula (42) was found to generate undershoots/overshoots that carry over to the FCT solution even if the limiter is applied to *all* primitive variables (Fig. 1c). Synchronized limiting of all conservative variables was the only FCT method to produce satisfactory results (not shown here, see Table 2) with the Roe-type low-order scheme. In the case of the Rusanov scalar dissipation, nonoscillatory solutions were obtained with the density-pressure FCT limiter (Fig. 1d).

The failsafe control of density and pressure (see Section 9.2) makes the solutions less sensitive to the choice of control variables for the base limiter. The nonoscillatory results shown in Fig. 1e-f were obtained using the density-pressure postpocessing for the Roe-FCT scheme with $\alpha_{ij}(\rho, p)$ and $\alpha_{ij}(\rho, \rho E)$, respectively. The results in Figs. 1g-h indicate that it is even possible to deactivate the main limiter, i.e., set $\alpha_{ij} := 1$ and remove (a fraction of) the antidiffusive flux in regions where the local FCT constraints (87) are violated. However, this practice is not generally recommended since it might trigger aggressive limiting at the postprocessing step.

In contrast to high-resolution schemes of TVD type, the raw antidiffusive flux (59) includes a contribution of the consistent mass matrix. The lumped-mass version ($\dot{\mathrm{U}}^L := 0$) of the FCT algorithm produces the solution shown in Fig. 1h. The superior phase accuracy of the consistent-mass Galerkin discretization justifies the additional effort invested in the computation of the approximate time derivative (60).

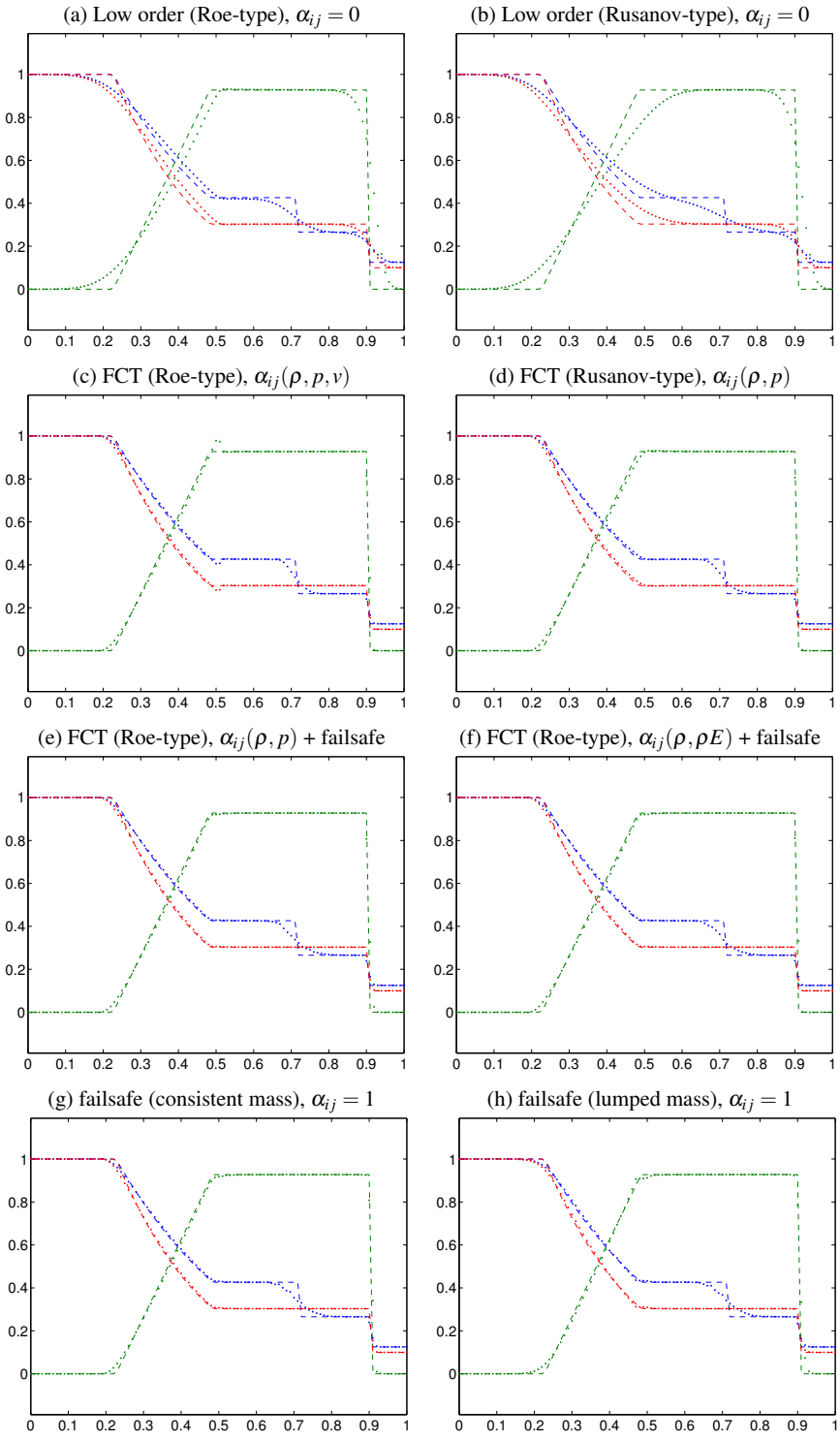


Fig. 1 Shock tube problem: $h = 10^{-2}$, $\Delta t = 10^{-3}$. Snapshots of the density (blue), velocity (green), and pressure (red) distribution at the final time $T = 0.231$.

The error norms for the density, pressure, and velocity fields calculated with the above algorithms are listed in Tables 1-4. The expected order of accuracy p was estimated by formula (111) using the solutions computed on the two finest meshes. As expected, the largest errors are observed for the low-order approximations. The accuracy of Roe's approximate Riemann solver is marginally better than that of the Rusanov scheme. The rate of grid convergence for the density approaches $p = 2/3$, which is in good agreement with the results presented in [5]. Tables 2-4 confirm that the linearized FCT algorithm converges much faster than the underlying low-order scheme. The expected order of accuracy attains values in the range 0.9–1.1.

The presented grid convergence study sheds some light on various aspects of flux limiting for the unsteady Euler equations. As a rule of thumb, constraining the density and pressure or total energy is a good choice in the context of synchronous FCT. The failsafe feature improves the robustness of the algorithm but may increase the amount of numerical diffusion. To achieve optimal phase accuracy for time-dependent problems, the raw antidiffusive flux must include the contribution of the consistent mass matrix. Of course, the overall performance of the algorithm also depends on the accuracy of the time-stepping scheme and on the time step size.

Table 1 Shock tube problem: grid convergence of the low-order schemes ($\alpha_{ij} = 0$).

	Roe's scheme			Rusanov's scheme		
h	$E_1(\rho,h)$	$E_1(u,h)$	$E_1(p,h)$	$E_1(\rho,h)$	$E_1(u,h)$	$E_1(p,h)$
1/100	2.1788e-02	4.2199e-02	1.9789e-02	2.8687e-02	5.4016e-02	2.6282e-02
1/200	1.3969e-02	2.4904e-02	1.1995e-02	1.9468e-02	3.2518e-02	1.6138e-02
1/400	8.8233e-03	1.3737e-02	7.0753e-03	1.2659e-02	1.8557e-02	9.6411e-03
1/800	5.5562e-03	7.5261e-03	4.1293e-03	8.1083e-03	1.0478e-02	5.6589e-03
1/1600	3.4900e-03	4.0920e-03	2.3835e-03	5.1423e-03	5.8427e-03	3.2668e-03
1/3200	2.2003e-03	2.2017e-03	1.3609e-03	3.2579e-03	3.2178e-03	1.8593e-03
	$p = 0.67$	$p = 0.89$	$p = 0.81$	$p = 0.66$	$p = 0.86$	$p = 0.81$

Table 2 Shock tube problem: grid convergence of FCT without failsafe correction.

	Roe-type predictor, $\alpha_{ij}(\rho,\rho E,\rho v)$			Rusanov-type predictor, $\alpha_{ij}(\rho,p)$		
h	$E_1(\rho,h)$	$E_1(u,h)$	$E_1(p,h)$	$E_1(\rho,h)$	$E_1(u,h)$	$E_1(p,h)$
1/100	7.2976e-03	1.1244e-02	5.6132e-03	9.2527e-03	1.0041e-02	4.6990e-03
1/200	3.8044e-03	6.8249e-03	2.9742e-03	5.1909e-03	6.2159e-03	2.5124e-03
1/400	1.9693e-03	3.3300e-03	1.4743e-03	2.8313e-03	3.0024e-03	1.2358e-03
1/800	1.0334e-03	1.5903e-03	7.2550e-04	1.4237e-03	1.4209e-03	6.0422e-04
1/1600	5.3461e-04	7.3201e-04	3.5412e-04	7.0374e-04	6.4491e-04	2.9243e-04
1/3200	2.8770e-04	3.3918e-04	1.7761e-04	3.5707e-04	2.9345e-04	1.4587e-04
	$p = 0.89$	$p = 1.11$	$p = 1.00$	$p = 0.98$	$p = 1.13$	$p = 1.00$

Table 3 Shock tube problem: grid convergence of FCT with failsafe correction.

	Roe-type predictor, $\alpha_{ij}(\rho,p)$			Roe-type predictor, $\alpha_{ij}(\rho,\rho E)$		
h	$E_1(\rho,h)$	$E_1(u,h)$	$E_1(p,h)$	$E_1(\rho,h)$	$E_1(u,h)$	$E_1(p,h)$
1/100	8.4389e-03	9.0475e-03	4.2509e-03	7.9186e-03	8.8199e-03	4.2180e-03
1/200	5.0820e-03	5.7128e-03	2.2982e-03	4.7613e-03	5.6378e-03	2.2879e-03
1/400	3.0545e-03	2.7739e-03	1.1361e-03	2.6349e-03	2.7383e-03	1.1309e-03
1/800	1.8737e-03	1.3183e-03	5.5785e-04	1.4212e-03	1.2924e-03	5.5345e-04
1/1600	1.1502e-03	5.9718e-04	2.7068e-04	7.0388e-04	5.8257e-04	2.6794e-04
1/3200	6.6805e-04	2.7102e-04	1.3611e-04	3.5656e-04	2.6259e-04	1.3413e-04
	$p = 0.78$	$p = 1.14$	$p = 0.99$	$p = 0.98$	$p = 1.15$	$p = 1.00$

Table 4 Shock tube problem: grid convergence of Roe-type failsafe FCT for $\alpha_{ij} = 1$.

	consistent mass matrix			lumped mass matrix		
h	$E_1(\rho,h)$	$E_1(u,h)$	$E_1(p,h)$	$E_1(\rho,h)$	$E_1(u,h)$	$E_1(p,h)$
1/100	8.4725e-03	9.2123e-03	4.3338e-03	8.9680e-03	1.0579e-02	5.0899e-03
1/200	5.1763e-03	5.8569e-03	2.3466e-03	5.4849e-03	6.3070e-03	2.6797e-03
1/400	3.0879e-03	2.8643e-03	1.1668e-03	3.2170e-03	3.0904e-03	1.3348e-03
1/800	1.9700e-03	1.3755e-03	5.7960e-04	1.9142e-03	1.4843e-03	6.5940e-04
1/1600	1.2025e-03	6.3730e-04	2.8494e-04	1.1215e-03	6.9581e-04	3.2554e-04
1/3200	7.5109e-04	3.0126e-04	1.4721e-04	6.4676e-04	3.3016e-04	1.6544e-04
	$p = 0.68$	$p = 1.08$	$p = 0.95$	$p = 0.79$	$p = 1.08$	$p = 0.98$

12.2 Radially Symmetric Riemann Problem

The second transient benchmark [36] is a radially symmetric 2D counterpart of the shock tube problem. Before an impulsive start, an imaginary membrane separates the square domain $\Omega = (-0.5,0.5) \times (-0.5,0.5)$ into the inner circle

$$\Omega_L = \{(x,y) \in \Omega \,|\, \sqrt{x^2+y^2} < 0.13\}$$

and the complement $\Omega_R = \Omega \backslash \Omega_L$. Reflective boundary conditions are prescribed on the boundary of Ω. The gas is initially at rest. Higher pressure and density are maintained inside Ω_L than outside. The interior and exterior states are given by

$$\begin{bmatrix} \rho_L \\ \mathbf{v}_L \\ p_L \end{bmatrix} = \begin{bmatrix} 2.0 \\ 0.0 \\ 15.0 \end{bmatrix}, \qquad \begin{bmatrix} \rho_R \\ \mathbf{v}_R \\ p_R \end{bmatrix} = \begin{bmatrix} 1.0 \\ 0.0 \\ 1.0 \end{bmatrix}.$$

The abrupt removal of the membrane at $t = 0$ gives rise to a radially expanding shock wave driven by the pressure difference. The challenge of this test is to capture the moving discontinuities while preserving the radial symmetry of the solution.

All computations are performed using linear finite elements on unstructured meshes constructed via regular subdivision of the coarse mesh depicted in Fig. 2.

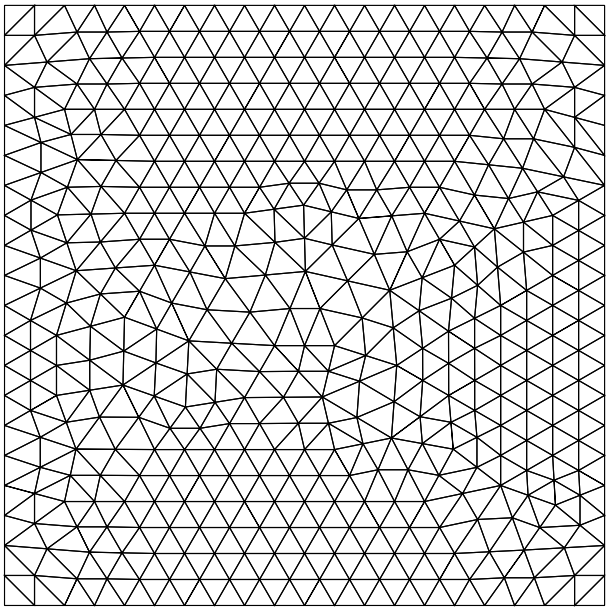


Fig. 2 Radially symmetric Riemann problem: coarse mesh, 824 triangles, 453 vertices.

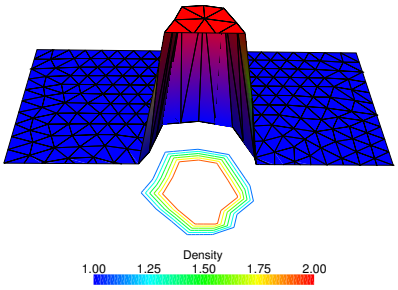
As explained in Section 10, it is advisable to initialize the numerical solution in a conservative manner. The total mass and energy of the initial data are given by

$$\int_\Omega \rho \, \mathrm{d}\mathbf{x} = 1 + (0.13)^2 \cdot \pi \approx 1.05309291584567,$$
$$\int_\Omega \rho E \, \mathrm{d}\mathbf{x} = 2.5 + 35 \cdot (0.13)^2 \pi \approx 4.35825205459836.$$

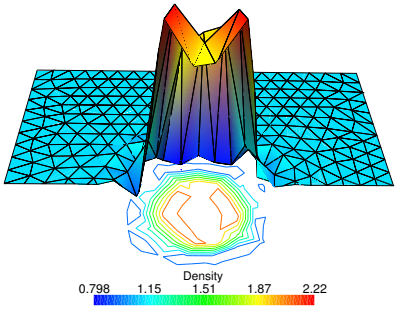
Since the exact solution is discontinuous, the load vector (94) was assembled using adaptive cubature formulas [70]. The density profiles produced by 4 different initialization techniques are shown in Figs. 3a-d. It can readily be seen that the consistent L^2 projection fails to preserve the bounds of the initial data, while its lumped counterpart gives rise to significant numerical diffusion. The synchronized FCT limiter (96) with $\alpha_{ij} = \alpha_{ij}(\rho, p)$ makes it possible to achieve a crisp resolution of the discontinuous initial profile without generating undershoots or overshoots.

Table 5 reveals that the pointwise initialization of nodal values is nonconservative. The consistent-mass L^2 projection preserves the total mass and energy but the initial density exhibits undershoots and overshoots of about 20%. Moreover, the initial pressure attains negative values, which results in an immediate crash of the code. In contrast, the nodal values obtained with the lumped-mass L^2 projection and the flux-corrected version satisfy $1.0 \leq \rho_h^0 \leq 2.0$ and $1.0 \leq p_h^0 \leq 15.0$ as desired.

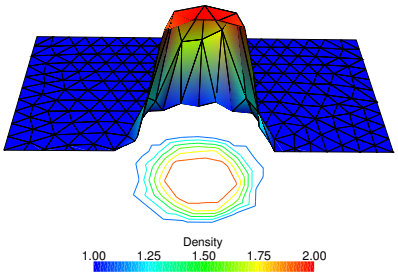
(a) Pointwise initialization



(b) Consistent L^2 projection, $\alpha_{ij} = 1$



(c) lumped-mass L^2 projection, $\alpha_{ij} = 0$



(d) constrained L^2 projection, $\alpha_{ij}(\rho, p)$

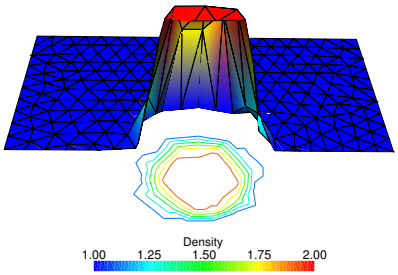
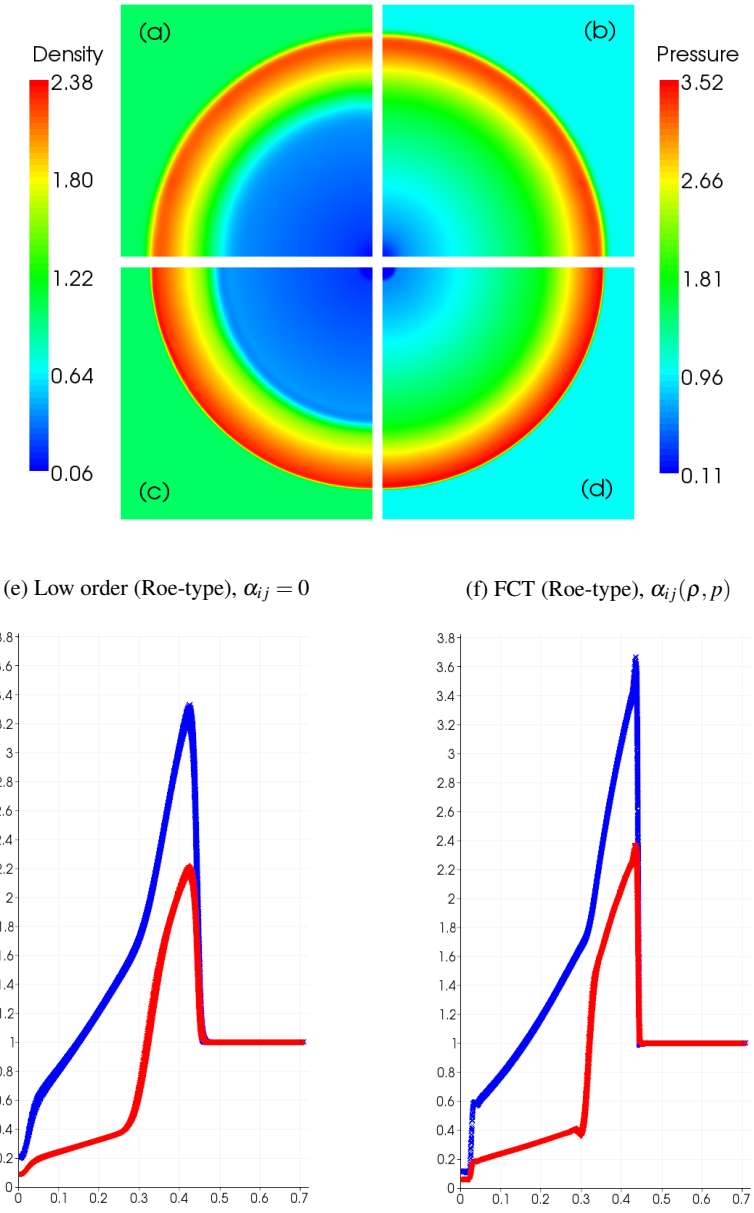


Fig. 3 Radially symmetric Riemann problem: initial density ρ_h^0 on the coarse mesh.

The evolution of the numerical solution initialized by the constrained L^2 projection was studied on the mesh obtained with 4 global refinements. The Crank-Nicolson time-stepping was employed with $\Delta t = 2 \cdot 10^{-3}$. Figures 4a-d display snapshots of the density (left) and pressure (right) at the final time $T = 0.13$. These solutions were obtained using the Roe tensorial dissipation and linearized FCT with the density-pressure limiter. Remarkably, both the low-order solution (top) and its flux-corrected counterpart (bottom) preserve the radial symmetry on the unstructured mesh. The symmetry plots shown in Figs. 4e-f show the nodal values $\rho_i = \rho_h(x_i, y_i)$

Table 5 Radially symmetric Riemann problem: constrained initialization on the coarse mesh.

	$\int_\Omega \rho_h^0 \, \mathbf{dx}$	$\int_\Omega (\rho E)_h^0 \, \mathbf{dx}$	$\min(\rho_h^0)$	$\max(\rho_h^0)$	$\min(p_h^0)$	$\max(p_h^0)$
(a)	1.04799	4.17949	1.0	2.0	1.0	15.0
(b)	1.05309	4.35823	7.9845e-01	2.2239e+00	-1.8216e+00	1.8135e+01
(c), (d)	1.05309	4.35823	1.0	2.0	1.0	15.0



(e) Low order (Roe-type), $\alpha_{ij} = 0$ (f) FCT (Roe-type), $\alpha_{ij}(\rho, p)$

Fig. 4 Radially symmetric Riemann problem: density (red) and pressure (blue) at $T = 0.13$.

and $p_i = p_h(x_i, y_i)$ versus distance to the origin. The presented results are in a good agreement with the reference solutions computed using CLAWPACK [38].

12.3 Double Mach Reflection

A more challenging test for the unsteady Euler equations is the double Mach reflection problem of Woodward and Colella [72] . In this benchmark, a Mach 10 shock impinges on a reflecting wall at the angle of 60° degrees. The computational domain is the rectangle $\Omega = (0,4) \times (0,1)$. The following pre-shock and post-shock values of the flow variables are used to define the initial and boundary conditions [3]

$$\begin{bmatrix} \rho_L \\ u_L \\ v_L \\ p_L \end{bmatrix} = \begin{bmatrix} 8.0 \\ 8.25\cos(30^\circ) \\ -8.25\sin(30^\circ) \\ 116.5 \end{bmatrix}, \qquad \begin{bmatrix} \rho_R \\ u_R \\ v_R \\ p_R \end{bmatrix} = \begin{bmatrix} 1.4 \\ 0.0 \\ 0.0 \\ 1.0 \end{bmatrix}. \tag{113}$$

Initially, the post-shock values are prescribed in $\Omega_L = \{(x,y) \mid x < 1/6 + y/\sqrt{3}\}$ and the pre-shock values in $\Omega_R = \Omega \backslash \Omega_L$. The reflecting wall corresponds to $1/6 \le x \le 4$ and $y = 0$. No boundary conditions are required along the line $x = 4$. On the rest of the boundary, the post-shock conditions are prescribed for $x < 1/6 + (1 + 20t)/\sqrt{3}$ and the pre-shock conditions elsewhere [3]. The so-defined values along the top boundary describe the exact motion of the initial Mach 10 shock.

The density fields (30 isolines) depicted in Figs. 5-7 were computed using bilinear finite elements on a sequence of structured meshes with equidistant grid spacings $h = 1/64$, $1/128$, $1/256$, and $1/512$. Integration in time was performed until $T = 0.2$ by the Crank-Nicolson scheme with the time step $\Delta t = 64h \cdot 10^{-4}$. The low-order solution displayed in Fig. 5 was calculated using the Roe-type artificial viscosity. Due to strong numerical diffusion, the complex interplay of incident, reflected, and Mach stem shock waves is resolved rather poorly, and so is the slipstream at the triple point. The use of FCT with synchronized limiting on primitive (Fig. 6) or conservative (Fig. 7) variables yields a marked improvement without producing 'staircase structures' or other artefacts observed by Woodward and Colella [72].

12.4 GAMM Channel

In the remainder of this section, we present the results of a numerical study for the stationary Euler equations. To begin with, we simulate the steady transonic flow in the GAMM channel with a 10% circular bump. For a detailed description of this popular benchmark, we refer to Feistauer et al. [13]. The gas enters the channel at free stream Mach number $M_\infty = 0.67$ and accelerates to supersonic velocities as it flows over the bump. The Mach number varies between approximately 0.22 and 1.41. An isolated shock wave forms in the local supersonic region. The inlet and outlet lie in the region of subsonic flow. Hence, the results are sensitive to the choice of physical and numerical boundary conditions. This makes the GAMM channel problem rather challenging when it comes to computing steady-state solutions.

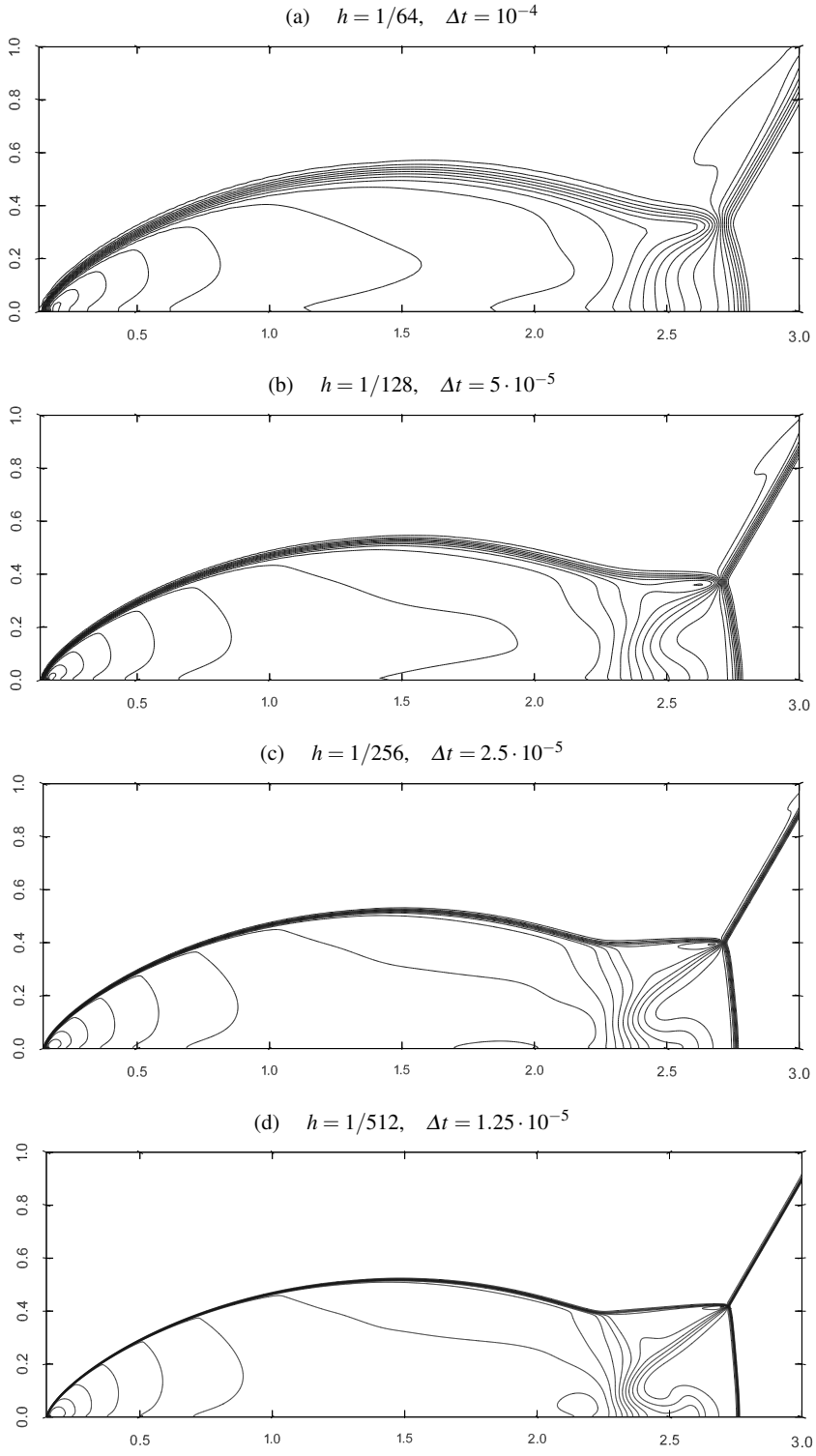


Fig. 5 Double Mach reflection: isodensity contours at $T = 0.2$; low-order scheme, $\alpha_{ij} = 0$.

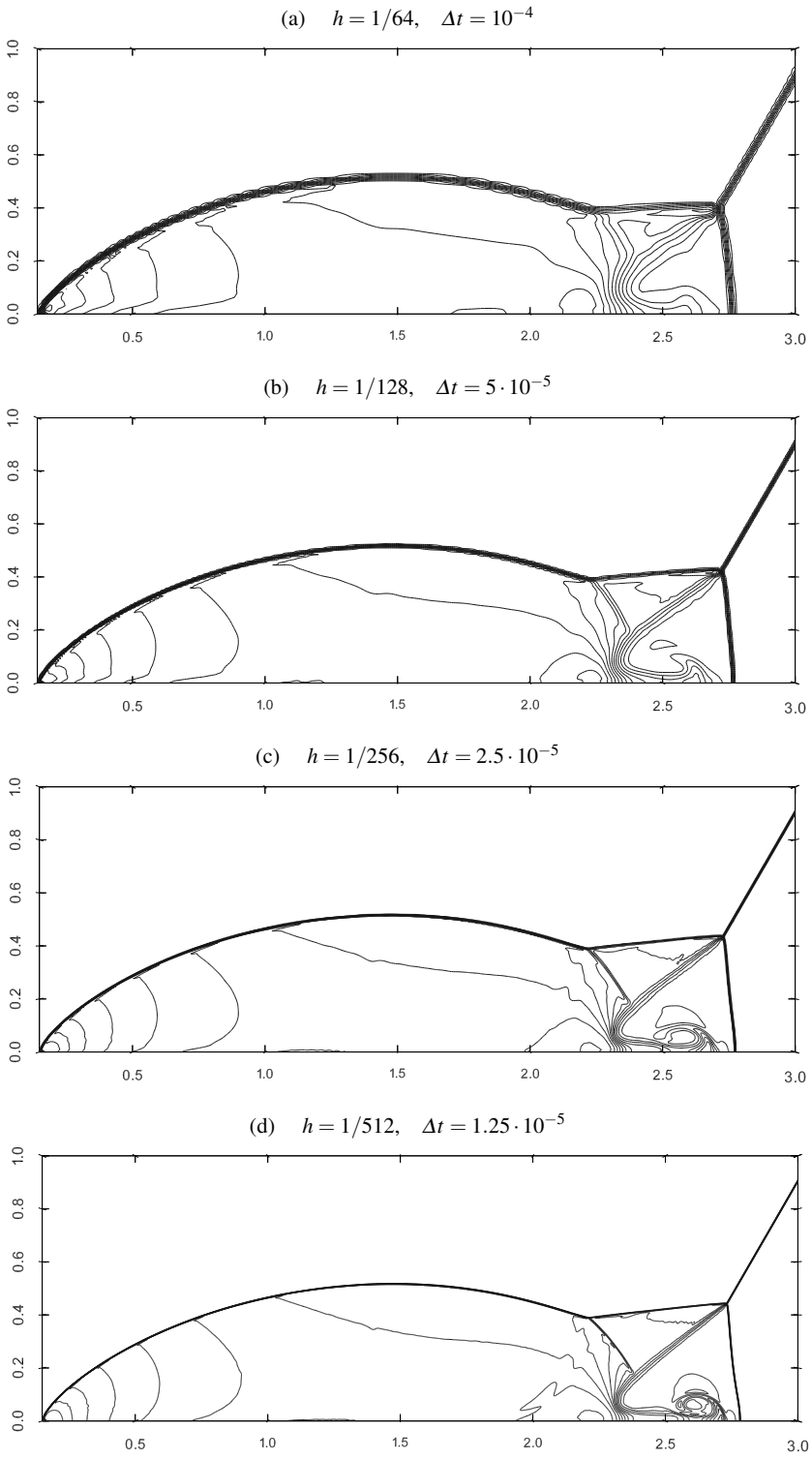


Fig. 6 Double Mach reflection: isodensity contours at $T = 0.2$; unsafe FCT^{Roe}, $\alpha_{ij}(\rho, p)$.

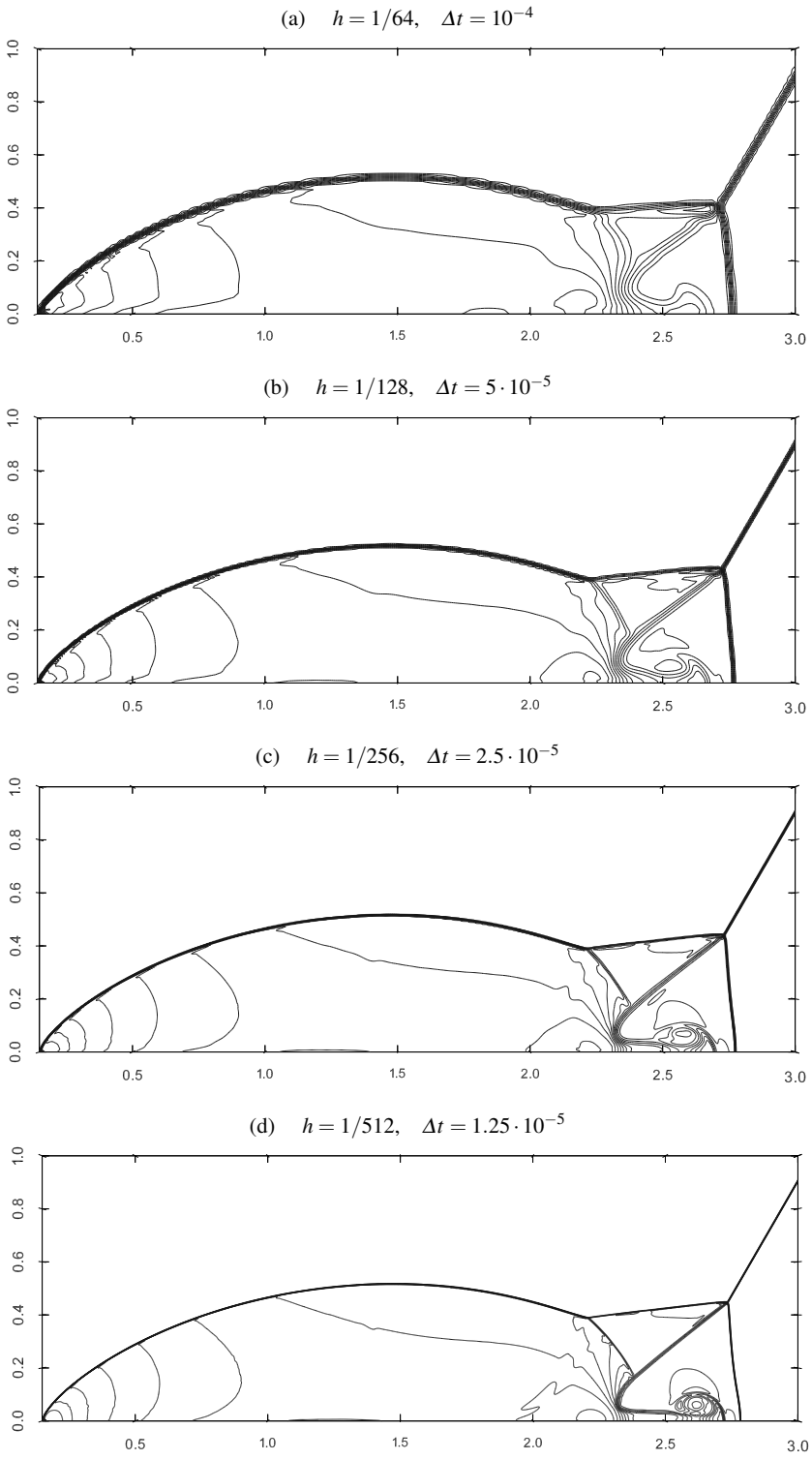


Fig. 7 Double Mach reflection: isodensity contours at $T = 0.2$; unsafe FCT^{Roe}, $\alpha_{ij}(\rho, \rho E)$.

Unless mentioned otherwise, the free stream boundary values for all stationary benchmark problems are given in the following dimensionless form [61]

variable	free stream value
ρ_∞	1
u_∞	M_∞
v_∞	0
p_∞	$\frac{1}{\gamma}$
E_∞	$\frac{M_\infty^2}{2}+\frac{1}{\gamma(\gamma-1)}$

The unstructured triangular mesh shown in Fig. 8b is successively refined to construct the computational mesh for the GAMM channel. Table 6 lists the number of vertices (NVT) and elements (NEL) for up to 5 quad-tree refinements. The stationary Mach number distribution computed with an algebraic flux correction scheme of TVD type [18, 19, 29] on mesh level 6 is presented in Fig. 8a. It can readily be seen that the resolution of the shock wave is rather crisp and nonoscillatory.

The numerical solution to the Euler equations was initialized by the above free stream values and marched to the steady state using pseudo-time-stepping in conjunction with the semi-implicit linearization procedure (see Section 7). At the initial stage, we neglect the nonlinear antidiffusive term and begin with the inexpensive computation of a low-order predictor. When the residuals of the low-order scheme reach the prescribed tolerance, the limited antidiffusive correction is switched on, and the iteration process continues until convergence to a stationary solution.

During the startup phase, the pseudo-time-stepping scheme runs at the moderately large CFL number $\nu = 100$. When the relative residual falls below 10^{-2}, the linearization becomes sufficiently accurate, and ν can be chosen arbitrarily large. In our experience, the semi-implicit algorithm converges even for $\nu = \infty$.

Figure 9 presents the convergence history for the flux-corrected Galerkin scheme and its low-order counterpart. In either case, the Neumann-type boundary conditions are imposed in a weak sense. The log-scale plots show the residual of the nonlinear system versus the number of pseudo-time steps for various CFL numbers. The employed mesh (refinement level 4) contains a total of $9,577$ vertices.

Remarkably, convergence to the steady-state solution accelerates as the CFL number increases. In the case of the low-order scheme, $\nu = \infty$ delivers the best convergence rates, whereby the norm of the residual falls below 10^{-12} after just ten iterations. Small values of the CFL number imply slow convergence, whereas fast and almost monotone error reduction is observed for large pseudo-time steps.

The flux-corrected Galerkin scheme exhibits a similar convergence behavior but requires a larger number of nonlinear iterations. As the CFL number is increased, the convergence rates improve until the threshold $\nu = 100$ is reached. A further increase of the pseudo-time step does not result in faster convergence. In contrast to the findings of Trépanier et al. [67], the rate of convergence does not deteriorate but stays approximately the same for all $\nu \geq 100$. However, the lagged treatment of the non-differentiable antidiffusive term and the oscillatory behavior of the correction

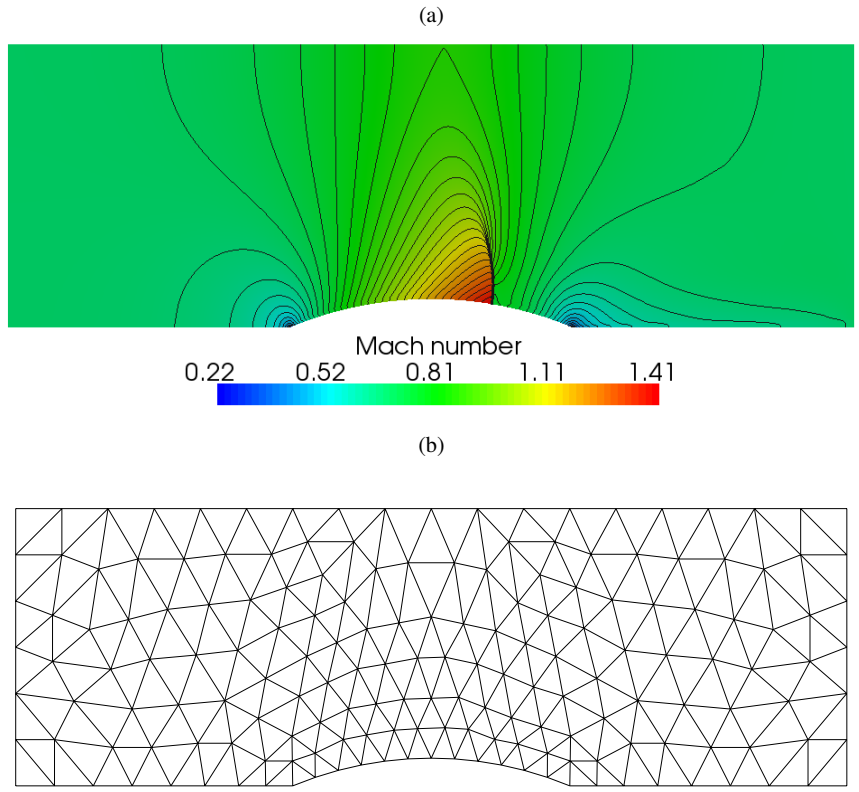


Fig. 8 GAMM channel: (a) stationary Mach number distribution and (b) the coarse grid.

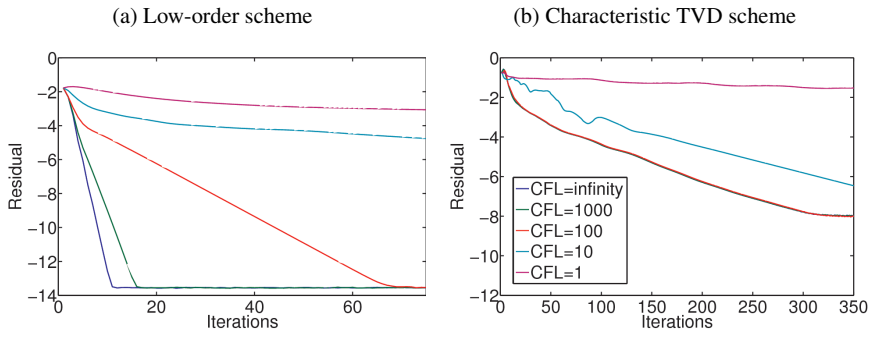


Fig. 9 GAMM channel: convergence history for various CFL numbers on mesh level 4.

factors produced by the limiter impose an upper bound on the rate of convergence. A better preconditioning of the discrete Jacobian operator and/or the use of convergence acceleration technique are likely to yield a further gain of efficiency.

Table 6 GAMM channel: relative L^2 errors and grid convergence rates.

Level	NVT	NEL	E_2^{Low}	p^{Low}	E_2^{Lim}	p^{Lim}
1	176	292	$5.47 \cdot 10^{-2}$	0.59	$3.05 \cdot 10^{-2}$	0.56
2	643	1168	$3.64 \cdot 10^{-2}$	0.64	$2.07 \cdot 10^{-2}$	1.04
3	2453	4672	$2.34 \cdot 10^{-2}$	0.59	$1.01 \cdot 10^{-2}$	0.99
4	9577	18688	$1.55 \cdot 10^{-2}$	0.61	$5.07 \cdot 10^{-3}$	1.45
5	37841	18688	$1.01 \cdot 10^{-2}$		$1.85 \cdot 10^{-3}$	
6	150433	299008				

The results of a grid convergence study for stationary solutions to the GAMM channel problem are presented in Table 6. The relative L^2 error defined as

$$E_2^{rel} = \frac{\|U_h - U\|_2}{\|U\|_2} \tag{114}$$

is calculated using the reference solution U computed on mesh level 6. The effective order of accuracy is $p \approx 0.6$ for the low-order predictor and $p \approx 1.0$ for the high-resolution scheme. The higher accuracy of the flux-corrected solution justifies the additional computational effort. The errors generated near the shock can be reduced using adaptive mesh refinement based on a goal-oriented error estimate [30].

A proper implementation of boundary conditions is crucial for the overall accuracy of a numerical scheme for the Euler equations. Errors caused by an inappropriate boundary treatment may propagate into the interior of the domain and inhibit convergence to steady-state solutions. For an in-depth numerical study of the boundary conditions for the GAMM channel problem, we refer to Gurris et al. [18, 19]. It turns out that the fully implicit treatment of weakly imposed boundary conditions (see Section 11) leads to a much more robust and efficient implementation than the predictor-corrector algorithm described in the first edition of this chapter [32].

13 NACA 0012 Airfoil

In the next example, we simulate the steady gas flow past a NACA 0012 airfoil. The upper and lower surfaces are given by the function $f^{\pm} : [0, 1.00893] \mapsto \mathbb{R}$ with

$$f^{\pm}(x) = \pm 0.6 \left(0.2969\sqrt{x} - 0.126x - 0.3516x^2 - 0.1015x^4\right). \tag{115}$$

We consider three test configurations labeled Case I-III. The corresponding values of the free stream Mach number M_∞ and inclination angle α are listed in Table 7.

Table 7 NACA 0012 airfoil: Test cases

Case	α	M_∞
I	2°	0.63
II	1.25°	0.8
III	1°	0.85

The outer boundary of the computational domain is a circle of radius 10 centered at the tip of the airfoil. The unstructured coarse mesh and a zoom of the reference solution for Case II are displayed in Fig. 10. The stationary Mach number distribution is in a good agreement with the numerical results presented in [13, 23, 32, 53].

The low-order solution is initialized by the free stream values, and a few iterations with the CFL number $\nu = 10$ are performed before increasing the pseudo-time step. As before, the low-order predictor serves as an initial guess for the algebraic flux correction scheme equipped with the characteristic limiter of TVD type.

The nonlinear convergence history for mesh level 2 and the results of a grid convergence study for Case 2 are presented in Fig. 11 and Table 8, respectively. As in the previous example, the semi-implicit pseudo-time-stepping scheme converges faster as the CFL number is increased. In the case of $\nu = \infty$, the residual falls below 10^{-15} in 20 iterations. The high-resolution scheme exhibits similar convergence behavior, although the total number of iterations is much larger. It takes approximately 200 iterations for the residual to reach the tolerance 10^{-8}. Increasing the CFL numbers beyond the threshold $\nu = 100$ yields just a marginal improvement. The errors for $\nu = 100$, 1000, and ∞ are almost identical but considerably smaller than those for $\nu = 1$ and 10. The effective order of accuracy is about 0.5 for the low-order scheme and 1.0 for the characteristic FEM-TVD scheme (see Table 8).

The drag and lift coefficients for all test cases are displayed in Table 9. They agree well with the available reference data [8, 15, 53], although the lift is slightly underestimated. This fact can be attributed to the relatively small size of the computational domain. It was shown in [8, 53] that the value of the lift coefficient tends to increase with the distance to the artificial far field boundary. The results presented therein were computed with far field distances of up to 2048 chords, while the far field boundary of our domain is located as few as 10 chords away from the airfoil.

Table 8 NACA 0012 airfoil: relative L^2 errors and grid convergence rates.

Level	NVT	NEL	E_2^{Low}	p^{Low}	E_2^{Lim}	p^{Lim}
1	2577	4963	$4.08 \cdot 10^{-2}$	0.51	$1.68 \cdot 10^{-3}$	1.02
2	10117	19852	$2.86 \cdot 10^{-2}$	0.46	$8.27 \cdot 10^{-4}$	1.02
3	40086	79408	$2.08 \cdot 10^{-2}$		$2.68 \cdot 10^{-4}$	
4	159580	317632				

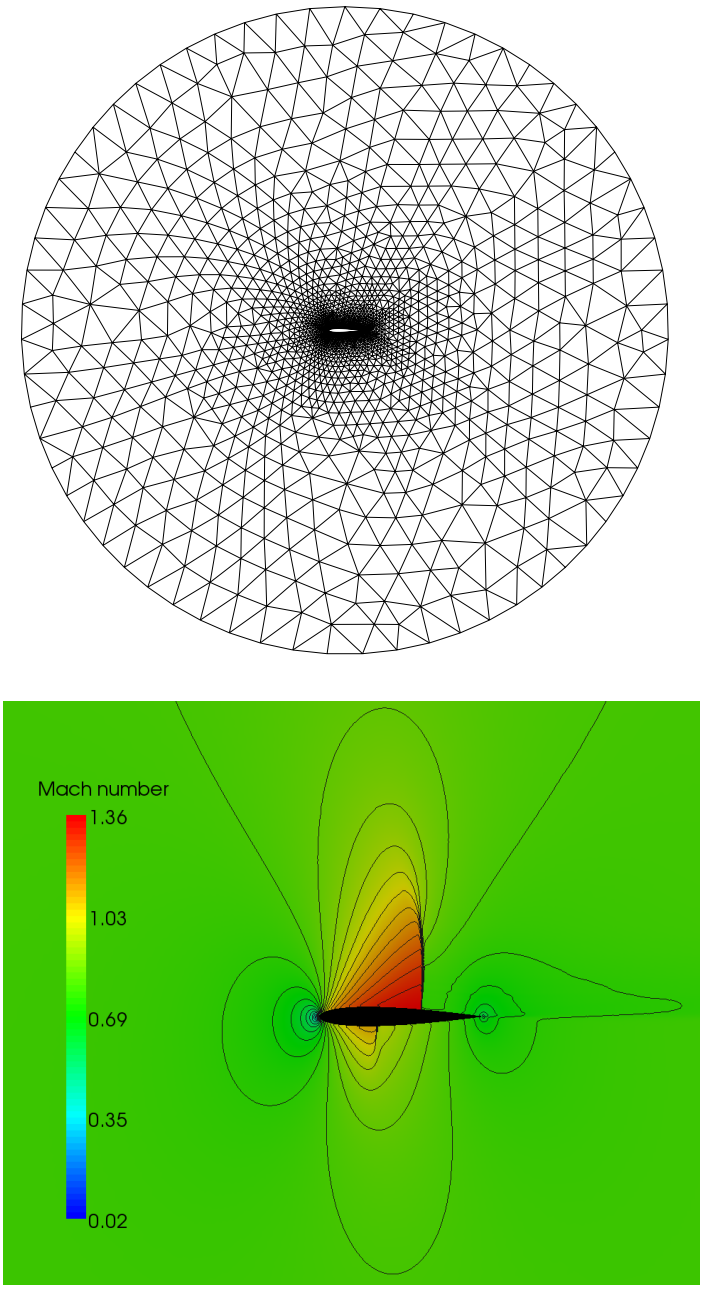


Fig. 10 NACA 0012 airfoil: coarse mesh and the Mach number distribution (zoom).

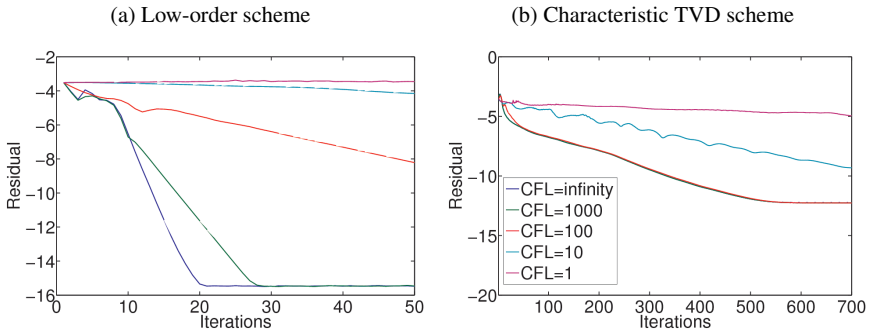


Fig. 11 NACA 0012 airfoil: convergence history for various CFL numbers on mesh level 2.

Table 9 NACA 0012 airfoil: drag and lift coefficients for all configurations.

(a) Case I

Level	C_D	C_L
1	$2.8194 \cdot 10^{-3}$	0.2791
2	$3.5473 \cdot 10^{-4}$	0.2977
3	$1.2927 \cdot 10^{-4}$	0.3071
4	$1.1355 \cdot 10^{-4}$	0.3120

(b) Case II

Level	C_D	C_L
1	$2.0043 \cdot 10^{-2}$	0.3065
2	$1.9198 \cdot 10^{-2}$	0.3169
3	$1.9501 \cdot 10^{-2}$	0.3199
4	$1.9933 \cdot 10^{-2}$	0.3200

(c) Case III

Level	C_D	C_L
1	$5.2434 \cdot 10^{-2}$	0.3205
2	$5.3217 \cdot 10^{-2}$	0.3400
3	$5.4087 \cdot 10^{-2}$	0.3441
4	$5.4636 \cdot 10^{-2}$	0.3436

14 Converging-Diverging Nozzle

In the last numerical example, we simulate the transonic flow in a converging-diverging nozzle. The free slip boundary condition (107) is prescribed on the upper and lower walls of the nozzle defined by the function $g^{\pm} : [-2,8] \mapsto \mathbb{R}$ with [23]

$$g^{\pm}(x) = \begin{cases} \pm 1 & \text{if} \quad -2 \le x \le 0, \\ \pm \frac{\cos\left(\frac{\pi x}{2}\right)+3}{4} & \text{if} \quad 0 < x \le 4, \\ \pm 1 & \text{if} \quad 4 < x \le 8. \end{cases} \tag{116}$$

At the subsonic inlet ($x = -2,\ -1 \le y \le 1$), the free stream Mach number equals $M_\infty = 0.3$. To facilitate comparison with the results presented by Hartmann and Houston [23], we define the free stream pressure as $p_\infty = 1$ rather than $p_\infty = \frac{1}{\gamma}$. At the subsonic outlet ($x = 8,\ -1 \le y \le 1$), the exit pressure $p_{out} = \frac{2}{3}$ is prescribed as explained in Section 11.3.1. As the nozzle converges, the gas is accelerated to supersonic velocities. After entering the diverging part, the flow begins to decelerate and passes through a shock before becoming subsonic again [23].

A mesh of bilinear elements is generated from a structured coarse mesh using global refinements. The numbers of vertices and elements for 7 levels of refinement are listed in Table 10. Figure 12 displays the numerical solution computed on mesh level 7. There is a good agreement with the results obtained by Hartmann [22].

Table 10 Converging-diverging nozzle: mesh properties.

Level	NVT	NEL
1	33	20
2	105	80
3	369	320
4	1377	1280
5	5313	5120
6	20865	20480
7	82689	81920

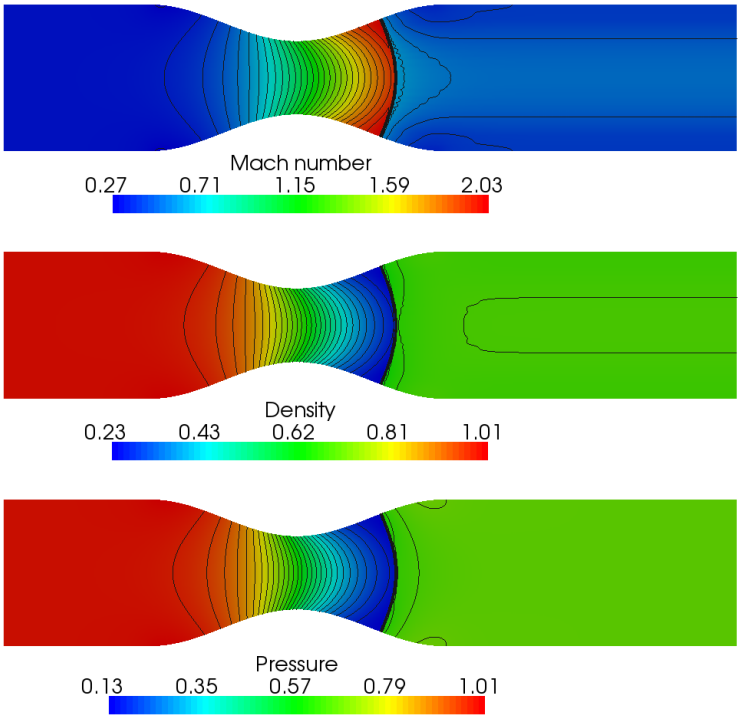


Fig. 12 Converging-diverging nozzle: FEM-TVD solution on mesh level 7.

To assess the numerical error in the outlet boundary condition $p_{\text{out}} = \frac{2}{3}$, we present the pressure distribution at the outlet Γ_{out} in Fig. 13. The relative L^2 error

$$E_2^{\text{out}} = \frac{\|p - p_{\text{out}}\|_{2,\Gamma_{out}}}{\|p_{\text{out}}\|_{2,\Gamma_{out}}} \tag{117}$$

and the effective order of accuracy p^{out} for mesh levels 5-7 are listed in Table 11, where NVT_{out} is the number of nodes at the outlet. It can be seen that the errors

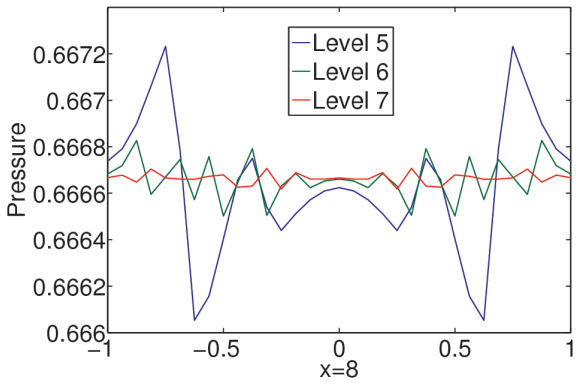


Fig. 13 Converging-diverging nozzle: exit pressure distribution.

are very small, even on a relatively coarse mesh. As the mesh is refined, the errors shrink. This illustrates the consistency of the proposed boundary treatment.

Table 11 Grid convergence study: outflow boundary condition.

Level	NVT	NEL	NVT_{out}	E_2^{out}	p^{out}
5	5313	5120	33	$2.50 \cdot 10^{-3}$	1.32
6	20865	20480	65	$1.00 \cdot 10^{-3}$	1.12
7	82689	81920	129	$4.62 \cdot 10^{-4}$	

15 Conclusions

This chapter sheds some light on the aspects of algebraic flux correction for systems of conservation laws. We extended the scalar limiting machinery to the compressible Euler equations and discussed various implementation details (initial and boundary conditions, linearization techniques, iterative solvers etc). Furthermore, we presented a new approach to constraining the primitive variables in synchronized FCT algorithms. It differs from other flux limiters for systems in that the transformation of variables is performed node-by-node rather than edge-by-edge. The generalized Zalesak limiter was equipped with a simple failsafe corrector designed to preserve the bounds of the low-order solution. A numerical study was performed to illustrate the practical utility of the proposed limiting techniques for the Euler equations.

In summary, flux limiting for hyperbolic systems may require (i) a careful choice of the variables to be controlled, (ii) a suitable synchronization of the correction factors, and (iii) a mechanism that makes it possible to 'undo' the antidiffusive cor-

rection whenever it turns out to be harmful. The accuracy and efficiency of the code depend on the employed linearizations. Moreover, the implementation of characteristic boundary conditions can make or break the numerical algorithm. All of these issues must be taken into account when it comes to solving real-life problems.

Acknowledgments

The authors would like to thank Stefan Turek (Dortmund University of Technology), John Shadid (Sandia National Laboratories), and Mikhail Shashkov (Los Alamos National Laboratory) for many stimulating discussions and useful suggestions.

References

[1] F. Angrand and A. Dervieux, Some explicit triangular finite element schemes for the Euler equations. *Int. J. Numer. Methods Fluids* **4** (1984) 749–764.

[2] P. Arminjon and A. Dervieux, Construction of TVD-like artificial viscosities on 2-dimensional arbitrary FEM grids. *INRIA Research Report* **1111** (1989).

[3] Athena test suite, http://www.astro.virginia.edu/VITA/ATHENA/dmr.html.

[4] N. Balakrishnan and G. Fernandez, Wall boundary conditions for inviscid compressible flows on unstructured meshes. *Int. J. Numer. Methods Fluids* **28** (1998) 1481–1501.

[5] J.W. Banks, W.D. Henshaw and J.N. Shadid, An evaluation of the FCT method for high-speed flows on structured overlapping grids. *J. Comput. Phys.* **228** (2009) 5349–5369.

[6] T.J. Barth, Numerical aspects of computing viscous high Reynolds number flows on unstructured meshes. *AIAA Paper* 91-0721 (1991).

[7] M. Behr, On the application of slip boundary condition on curved boundaries, *Int. J. Numer. Methods Fluids* **45** (2004) 43–51.

[8] D. De Zeeuw and K.G. Powell, An adaptive refined Cartesian mesh solver for the Euler equations, *J. Comp. Phys.* **104** (1993) 56–68.

[9] 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).

[10] V. Dolejší and M. Feistauer, A semi-implicit discontinuous Galerkin finite element method for the numerical solution of inviscid compressible flow. *J. Comp. Phys.* **198** (2004) 727–746.

[11] M.S. Engelman, R.L. Sani and P.M. Gresho, The implementation of normal and/or tangential boundary conditions in finite element codes for incompressible fluid flow. *Int. J. Numer. Methods Fluids* **2** (1982) 225–238.

[12] P.E. Farrell, M.D. Piggott, C.C. Pain, G.J. Gorman and C.R. Wilson, Conservative interpolation between unstructured meshes via supermesh construction. *Comput. Methods Appl. Mech. Engrg.* **198** (2009) 2632–2642.

[13] M. Feistauer, J. Felcman and I. Straškraba, *Mathematical and Computational Methods for Compressible Flow*. Clarendon Press, Oxford, 2003.
[14] M. Feistauer and V. Kučera, On a robust discontinuous Galerkin technique for the solution of compressible flow. *J. Comp. Phys.* **224** (2007) 208–231.
[15] L. Fezoui and B. Stoufflet, A class of implicit upwind schemes for Euler simulations with unstructured meshes, *J. Comp. Phys.* **84** (1989) 174–206.
[16] C. A. J. Fletcher, The group finite element formulation. *Comput. Methods Appl. Mech. Engrg.* **37** (1983) 225–243.
[17] J.-M. Ghidaglia and F. Pascal, On boundary conditions for multidimensional hyperbolic systems of conservation laws in the finite volume framework. Technical Report, ENS de Cachan, 2002.
[18] M. Gurris, *Implicit Finite Element Schemes for Compressible Gas and Particle-Laden Gas Flows*. PhD Thesis, Dortmund University of Technology, 2010.
[19] M. Gurris, D. Kuzmin and S. Turek, Implicit finite element schemes for the stationary compressible Euler equations. *Int. J. Numer. Methods Fluids* (2011), doi: *10.1002/fld.2532*.
[20] A. Harten, On a class of high-resolution total-variation-stable finite-difference schemes. *SIAM J. Numer. Anal.* **21** (1984) 1–23.
[21] A. Harten and J. M. Hyman, Self adjusting grid methods for one-dimensional hyperbolic conservation laws. *J. Comput. Phys.* **50** (1983) 235–269.
[22] R. Hartmann, Homepage http://www.numerik.uni-hd.de/~hartmann/.
[23] R. Hartmann and P. Houston, Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations, *J. Comp. Phys.* **183** (2002) 508–532.
[24] C. Hirsch, *Numerical Computation of Internal and External Flows. Vol. II: Computational Methods for Inviscid and Viscous Flows*. John Wiley & Sons, Chichester, 1990.
[25] L. Krivodonova and M. Berger, High-order accurate implementation of solid wall boundary conditions in curved geometries. *J. Comput. Phys.* **211** (2006) 492–512.
[26] D. Kuzmin, Linearity-preserving flux correction and convergence acceleration for constrained Galerkin schemes. To appear in *J. Comput. Appl. Math.* (2012).
[27] D. Kuzmin, *A Guide to Numerical Methods for Transport Equations*, University Erlangen-Nuremberg, 2010, http://www.mathematik.uni-dortmund.de/~kuzmin/Transport.pdf.
[28] D. Kuzmin, Explicit and implicit FEM-FCT algorithms with flux linearization. *J. Comput. Phys.* **228** (2009) 2517–2534.
[29] D. Kuzmin, Algebraic flux correction for finite element discretizations of coupled systems. In: E. Oñate, M. Papadrakakis, B. Schrefler (eds) *Computational Methods for Coupled Problems in Science and Engineering II*, CIMNE, Barcelona, 2007, 653–656.

[30] D. Kuzmin and M. Möller, Goal-oriented mesh adaptation for flux-limited approximations to steady hyperbolic problems. *J. Comput. Appl. Math.* **233** (2010) 3113–3120.

[31] D. Kuzmin and M. Möller, Algebraic flux correction I. Scalar conservation laws. Chapter 6 in the first edition of this book: Springer, 2005, 155–206.

[32] D. Kuzmin and M. Möller, Algebraic flux correction II. Compressible Euler equations. Chapter 7 in the first edition of this book: Springer, 2005, 207–250.

[33] D. Kuzmin, M. Möller, J.N. Shadid and M. Shashkov: Failsafe flux limiting and constrained data projections for equations of gas dynamics. *J. Comput. Phys.* **229** (2010) 8766–8779.

[34] D. Kuzmin, M. Möller and S. Turek, High-resolution FEM-FCT schemes for multidimensional conservation laws. *Computer Methods Appl. Mech. Engrg.* **193** (2004) 4915–4946.

[35] R. J. LeVeque, *Numerical Methods for Conservation Laws*. Birkhäuser, 1992.

[36] R. J. LeVeque, Simplified multi-dimensional flux limiting methods. *Numerical Methods for Fluid Dynamics* **IV** (1993) 175–190.

[37] R. J. LeVeque, *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, 2003.

[38] R. J. LeVeque, CLAWPACK – Conservation LAWs PACKage, available at the URL http://www.amath.washington.edu/~claw/.

[39] R. Liska, M. Shashkov, P. Váchal and B. Wendroff, Optimization-based synchronized Flux-Corrected Conservative interpolation (remapping) of mass and momentum for Arbitrary Lagrangian-Eulerian methods. *J. Comput. Phys.* **229** (2010) 1467–1497.

[40] R. Löhner, Applied CFD Techniques: An Introduction Based on Finite Element Methods. Second Edition, John Wiley & Sons, 2008.

[41] R. Löhner and J.D. Baum, 30 years of FCT: Status and directions. Chapter 5 in the first edition of this book: Springer, 2005, 131–154.

[42] 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.

[43] R. Löhner, K. Morgan and O.C. Zienkiewicz, An adaptive finite element procedure for compressible high speed flows. *Comput. Methods Appl. Mech. Engrg.* **51** (1985) 441–465.

[44] H. Luo, J.D. Baum and R. Löhner, Numerical solution of the Euler equations for complex aerodynamic configurations using an edge-based finite element scheme. AIAA-93-2933 (1993).

[45] P. R. M. Lyra, *Unstructured Grid Adaptive Algorithms for Fluid Dynamics and Heat Conduction*. PhD thesis, University of Wales, Swansea, 1994.

[46] 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.

[47] P. R. M. Lyra and K. Morgan, A review and comparative study of upwind biased schemes for compressible flow computation. I: 1-D first-order schemes. *Arch. Comput. Methods Eng.* **7** (2000) no. 1, 19–55.

[48] P. R. M. Lyra and K. Morgan, A review and comparative study of upwind biased schemes for compressible flow computation. II: 1-D higher-order schemes. *Arch. Comput. Methods Eng.* **7** (2000) no. 3, 333–377.

[49] P. R. M. Lyra and K. Morgan, A review and comparative study of upwind biased schemes for compressible flow computation. III: Multidimensional extension on unstructured grids. *Arch. Comput. Methods Eng.* **9** (2002) no. 3, 207–256.

[50] M. Möller, *Adaptive High-Resolution Finite Element Schemes*. PhD thesis, Dortmund University of Technology, 2008.

[51] M. Möller, Efficient solution techniques for implicit finite element schemes with flux limiters. *Int. J. Numer. Methods Fluids* **55** (2007) 611–635.

[52] K. Morgan and J. Peraire, Unstructured grid finite element methods for fluid mechanics. *Reports on Progress in Physics*, **61** (1998), no. 6, 569–638.

[53] A. Nejat, *A Higher-Order Accurate Unstructured Finite Volume Newton-Krylov Algorithm for Inviscid Compressible Flows*, PhD Thesis, Vancouver, 2007.

[54] J. Peraire, M. Vahdati, J. Peiro, K. Morgan, The construction and behaviour of some unstructured grid algorithms for compressible flows. *Numerical Methods for Fluid Dynamics* **IV**, Oxford University Press, 1993, 221–239.

[55] P. L. Roe, Approximate Riemann solvers, parameter vectors and difference schemes. *J. Comput. Phys.* **43** (1981) 357–372.

[56] A. Rohde, Eigenvalues and eigenvectors of the Euler equations in general geometries. *AIAA Paper* 2001-2609, (2001).

[57] V. Selmin, Finite element solution of hyperbolic equations. I. One-dimensional case. *INRIA Research Report* **655** (1987).

[58] V. Selmin, Finite element solution of hyperbolic equations. II. Two-dimensional case. *INRIA Research Report* **708** (1987).

[59] V. Selmin, The node-centred finite volume approach: bridge between finite differences and finite elements. *Comput. Methods Appl. Mech. Engrg.* **102** (1993) 107–138.

[60] V. Selmin and L. Formaggia, Unified construction of finite element and finite volume discretizations for compressible flows. *Int. J. Numer. Methods Engrg.* **39** (1996) 1–32.

[61] R. A. Shapiro, *Adaptive Finite Element Solution Algorithm for the Euler Equations*. Notes on Numerical Fluid Mechanics **32**, Vieweg, 1991.

[62] T.M. Smith, R.W. Hooper, C.C. Ober and A.A. Lorber, Intelligent nonlinear solvers for computational fluid dynamics. Conference Paper, Presentation at the 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno NV, January 2006.

[63] P. K. Smolarkiewicz and G. A. Grell, A class of monotone interpolation schemes. *J. Comput. Phys.* **101** (1992) 431–440.

[64] G. Sod, A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *J. Comput. Phys.* **27** (1978) 1–31.

[65] E. F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer, 1999.

[66] E. F. Toro, *NUMERICA, A Library of Source Codes for Teaching, Research and Applications*. Numeritek Ltd., http://www.numeritek.com, 1999.
[67] J.-Y. Trépanier, M. Reggio and D. Ait-Ali-Yahia, An implicit flux-difference splitting method for solving the Euler equations on adaptive triangular grids. *Int. J. Num. Meth. Heat Fluid Flow* **3** (1993) 63–77.
[68] S. Turek, *Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approach*, LNCSE 6, Springer, 1999.
[69] P. Váchal and R. Liska, Sequential Flux-Corrected Remapping for ALE Methods. In: A. Bermudez de Castro, D. Gomez, P. Quintela, and P. Salgado (eds.) *Numerical Mathematics and Advanced Applications* (ENUMATH 2005). Springer, 2006, pp. 671-679.
[70] W. Vogt, *Adaptive Verfahren zur numerischen Quadratur und Kubatur*. Preprint No. M 1/06, IfMath TU Ilmenau, 2006.
[71] P. Wesseling, *Principles of Computational Fluid Dynamics*. Springer, 2001.
[72] P. R. Woodward and P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks. *J. Comput. Phys.* **54** (1984) 115–173.
[73] H. C. Yee, Construction of explicit and implicit symmetric TVD schemes and their applications. *J. Comput. Phys.* **43** (1987) 151–179.
[74] 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.
[75] S. T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids. *J. Comput. Phys.* **31** (1979) 335–362.
[76] S. T. Zalesak, The design of Flux-Corrected Transport (FCT) algorithms for structured grids. Chapter 2 in the first edition of this book: Springer, 2005, 29–78.

Appendix

In this Appendix, we derive the artificial diffusion operator for the piecewise-linear Galerkin approximation to the one-dimensional Euler equations

$$\frac{\partial U}{\partial t}+\frac{\partial F}{\partial x}=0. \tag{118}$$

In the 1D case, we have

$$U=\begin{bmatrix}\rho\\ \rho v\\ \rho E\end{bmatrix},\qquad F=\begin{bmatrix}\rho v\\ \rho v^2+p\\ \rho H v\end{bmatrix}. \tag{119}$$

The differentiation of F by the chain rule yields the equivalent quasi-linear form

$$\frac{\partial U}{\partial t}+A\frac{\partial U}{\partial x}=0, \tag{120}$$

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

$$A=\begin{bmatrix} 0 & 1 & 0 \\ \frac{1}{2}(\gamma-3)v^2 & (3-\gamma)v & \gamma-1 \\ \frac{1}{2}(\gamma-1)v^3-vH & H-(\gamma-1)v^2 & \gamma v \end{bmatrix}. \tag{121}$$

The eigenvalues and right/left eigenvectors of A satisfy the system of equations

$$A\mathbf{r}_k=\lambda_k\mathbf{r}_k, \qquad \mathbf{l}_kA=\lambda_k\mathbf{l}_k, \qquad k=1,2,3 \tag{122}$$

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

$$A=R\Lambda R^{-1}, \qquad \Lambda=\mathrm{diag}\{v-c,v,v+c\} \tag{123}$$

in accordance with (9). The matrices of eigenvalues and eigenvectors are given by

$$\Lambda=\mathrm{diag}\{v-c,v,v+c\}, \tag{124}$$

$$R=\begin{bmatrix} 1 & 1 & 1 \\ v-c & v & v+c \\ H-vc & \frac{1}{2}v^2 & H+vc \end{bmatrix}=[\mathbf{r}_1,\mathbf{r}_2,\mathbf{r}_3], \tag{125}$$

and

$$R^{-1}=\begin{bmatrix} \frac{1}{2}\left(b_1+\frac{v}{c}\right) & \frac{1}{2}\left(-b_2v-\frac{1}{c}\right) & \frac{1}{2}b_2 \\ 1-b_1 & b_2v & -b_2 \\ \frac{1}{2}\left(b_1-\frac{v}{c}\right) & \frac{1}{2}\left(-b_2v+\frac{1}{c}\right) & \frac{1}{2}b_2 \end{bmatrix}=\begin{bmatrix} \mathbf{l}_1 \\ \mathbf{l}_2 \\ \mathbf{l}_3 \end{bmatrix}, \tag{126}$$

where

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

On a uniform mesh of linear finite elements, the coefficients of the lumped mass matrix M_L and of the discrete gradient operator C are given by

$$m_i=\Delta x, \qquad c_{ij}=\begin{cases} 1/2, & j=i+1, \\ -1/2, & j=i-1. \end{cases} \tag{127}$$

The lumped-mass Galerkin approximation is equivalent to the central difference scheme which can be written in the generic conservative form

$$\frac{\mathrm{dU}_i}{\mathrm{d}t}+\frac{\mathrm{F}_{i+1/2}-\mathrm{F}_{i-1/2}}{\Delta x}=0, \tag{128}$$

where

$$\mathrm{F}_{i+1/2}=\frac{\mathrm{F}_i+\mathrm{F}_{i+1}}{2}.$$

The numerical flux of the low-order scheme with $\mathrm{D}_{i+1/2}$ defined by (42) is

$$\mathrm{F}_{i+1/2} = \frac{\mathrm{F}_i + \mathrm{F}_{i+1}}{2} - \frac{1}{2}|\mathrm{A}_{i+1/2}|(\mathrm{U}_{i+1} - \mathrm{U}_i), \tag{129}$$

where $\mathrm{A}_{i+1/2}$ is the 1D Roe matrix. The so-defined approximation is known as Roe's approximate Riemann solver [55]. A detailed description of this first-order scheme can be found in many textbooks on gas dynamics [24, 37, 65]. Roe's method fails to recognize expansion waves and, therefore, may give rise to entropy-violating solutions (rarefaction shocks) in the neighborhood of sonic points. Hence, some additional numerical diffusion may need to be applied in regions where one of the characteristic speeds approaches zero [20, 21]. This trick is called an *entropy fix*.

The use of scalar dissipation (46) leads to a Rusanov-like low-order scheme with

$$\mathrm{F}_{i+1/2} = \frac{\mathrm{F}_i + \mathrm{F}_{i+1}}{2} - \frac{a_{i+1/2}}{2}(\mathrm{U}_{i+1} - \mathrm{U}_i), \tag{130}$$

where $a_{i+1/2}$ denotes the fastest characteristic speed. Zalesak [76] defines it as

$$a_{i+1/2} = \frac{|v_i| + |v_{i+1}|}{2} + \frac{c_i + c_{i+1}}{2}.$$

For reasons explained in [5], our definition of the Rusanov flux (130) is based on

$$a_{i+1/2} := \max\{|v_i| + c_i, |v_{i+1}| + c_{i+1}\}.$$

This formula yields a very robust and efficient low-order method for FCT [33].