High-resolution FEM-FCT schemes for multidimensional conservation laws

D. Kuzmin*, M. Möller and S. Turek

Institute of Applied Mathematics (LS III), University of Dortmund
Vogelpothsweg 87, D-44227, Dortmund, Germany

Abstract

The flux-corrected-transport paradigm is generalized to implicit finite element schemes and nonlinear systems of hyperbolic conservation laws. In the scalar case, a nonoscillatory low-order method of upwind type is derived by elimination of negative off-diagonal entries of the discrete transport operator. The difference between the discretizations of high and low order is decomposed into a sum of skew-symmetric antidiffusive fluxes. An iterative flux limiter independent of the time step is proposed for implicit schemes. The nonlinear antidiffusion is incorporated into the solution in the framework of a defect correction scheme preconditioned by the monotone low-order operator. In the case of a hyperbolic system, the global Jacobian matrix is assembled edge-by-edge without resorting to numerical integration. Its low-order counterpart is constructed by rendering all off-diagonal blocks positive definite or adding scalar artificial diffusion proportional to the spectral radius of the Roe matrix. The coupled equations are solved in a segregated manner within an outer defect correction loop equipped with a block-diagonal preconditioner. After a suitable synchronization, the correction factors evaluated for an arbitrary set of indicator variables are applied to the antidiffusive fluxes which are inserted into the global defect vector. The performance of the new algorithm is illustrated by numerical examples for scalar transport problems and the compressible Euler equations.

Key Words: convection-dominated flows; hyperbolic conservation laws; flux correction; finite elements; implicit time-stepping

1 Introduction

The flux-corrected-transport algorithm for convection-dominated flows was the first nonlinear high-resolution scheme to switch between high- and low-order discretizations in an adaptive fashion depending on the local smoothness of the solution. Its foundations were laid in the early 1970s by Boris and Book [3]. A genuinely multidimensional generalization of their celebrated SHASTA scheme was proposed by Zalesak [44] and carried over to finite elements by Löhner and his coworkers [27],[28]. For many years, their explicit FEM-FCT methodology remained unrivaled because the extension of other high-resolution methods to multidimensions and unstructured grids proved to be a rather challenging task.

*Correspondence to: kuzmin@math.uni-dortmund.de

Some finite element codes for compressible flow problems operate with numerical fluxes designed as in the finite difference context. To this end, a local one-dimensional stencil is reconstructed for each mesh edge by insertion of 'dummy nodes'. The solution values at these nodes are obtained using a suitable interpolation/extrapolation technique, and a slope limiter is invoked to control the gradients. Unfortunately, the resulting FEM discretization may still produce nonphysical undershoots and overshoots in some cases. Moreover, the use of P_1-elements is essential to the derivation of the underlying edge-based data structure as proposed by Peraire *et al.* [35]. An alternative flux decomposition procedure, which is applicable to arbitrary finite elements, is introduced in our recent publication devoted to a fully multidimensional flux limiter of TVD type [22].

In this paper, we present a generalized FEM-FCT formulation which is based on a representation of diffusive/antidiffusive terms as a sum of skew-symmetric internodal fluxes. A complete transition to a computationally efficient 'edge-based' data structure is feasible but not mandatory [19],[22]. Hence, the algorithm to be presented can be readily integrated into an existing finite element code while preserving the conventional element-by-element matrix assembly and data access. It amounts to modifying the high-order transport operator so as to enforce the M-matrix property and render the discretization local extremum diminishing. An in-depth description of the mathematical background is available in [17],[19],[20]. Our fully discrete approach distinguishes itself in that it deals with finite element matrices regardless of the underlying mesh, approximation spaces and even the number of spatial dimensions. Moreover, it can be used in conjunction with implicit time-stepping schemes, unlike the classical FEM-FCT procedure.

We start with a summary of the new methodology for a scalar convection-diffusion equation and improve the algorithm in a number of ways. First of all, we observe that Zalesak's limiter depends on the time step, so that the unconditional stability/positivity of our implicit FEM-FCT schemes cannot be duly utilized. To rectify this, we introduce an iterative limiting strategy, whereby the rejected antidiffusion can be 'recycled' and built into the numerical solution during the subsequent defect correction steps. Furthermore, we clarify the origin of spurious ripples which may develop at the boundary and propagate into the interior of the computational domain. It turned out that this alarming phenomenon first mentioned in [17] is due to failure of the standard FCT algorithm to determine correct upper and lower bounds for the solution values at the inlet and outlet. The modification proposed below eliminates the source of troubles completely.

The second part of this paper deals with a generalization of implicit FEM-FCT schemes to the Euler equations of gas dynamics. This is a continuation of the research reported in our previous publications [19],[20]. The issues to be addressed include the assembly of the global Jacobian matrix, the iterative treatment of nonlinearities, the construction of a nonoscillatory low-order scheme for systems of hyperbolic conservation laws, a proper synchronization of correction factors for decoupled variables and the implementation of boundary conditions. We will show that many elements of the mathematical theory and finite difference algorithms developed for the one-dimensional Euler equations can be incorporated into the finite element framework. Due to the fact that the system at hand is strongly nonlinear to begin with, the overhead cost incurred by the iterative flux correction is not very significant. Encouraging results obtained for a number of standard two-dimensional test problems demonstrate the potential of the new method.

2 Galerkin FEM for scalar equations

Consider a generic time-dependent conservation law for a scalar quantity u

$$\frac{\partial u}{\partial t} + \nabla \cdot \mathbf{f} = q \qquad \text{in } \Omega, \tag{1}$$

where q is a source term and $\mathbf{f}$ is a (nonlinear) flux function. Let us assume that it is composed from convective fluxes of the form $\mathbf{f}_c = \mathbf{v}u$ and diffusive fluxes of the form $\mathbf{f}_d = -\epsilon \nabla u$. The weak form of the resulting convection-diffusion equation reads

$$\int_\Omega w \left[\frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{v}u - \epsilon \nabla u) - q \right] d\mathbf{x} = 0, \qquad \forall w. \tag{2}$$

A common practice in finite element methods for conservation laws is to interpolate convective fluxes and source terms in the same way as the numerical solution

$$u = \sum_j u_j \varphi_j, \qquad \mathbf{f}_c = \sum_j (\mathbf{v}_j u_j)\, \varphi_j, \qquad q = \sum_j q_j \varphi_j, \tag{3}$$

where φ_i denote the basis functions spanning the finite-dimensional subspace. This kind of approximation was called the *group finite element formulation* by Fletcher [7] who found it to provide a very efficient treatment of nonlinear convective terms and even lead to a small gain of accuracy for the 2D Burgers equation discretized on a uniform grid.

The resulting Galerkin discretization of equation (1) reads

$$\sum_j \left[\int_\Omega \varphi_i \varphi_j \, d\mathbf{x} \right] (\dot{u}_j - q_j) + \sum_j \left[\int_\Omega (\varphi_i \mathbf{v}_j \cdot \nabla \varphi_j + \epsilon \nabla \varphi_i \cdot \nabla \varphi_j) \, d\mathbf{x} \right] u_j = 0. \tag{4}$$

It is implied that the diffusive flux vanishes at the boundary, so that the surface integral arising from the integration by parts can be omitted. Note that for most finite elements the sum of basis functions equals unity: $\sum_i \varphi_i \equiv 1$. Summing the semi-discretized equations over i, one recovers the integral form of the conservation law, which ensures that the total amount of u in Ω may only change due to boundary fluxes and internal sources or sinks. This shows that the Galerkin method enjoys the global conservation property.

The above system of ordinary differential equations for the nodal values of the approximate solution can be written compactly in matrix form

$$M_C \frac{du}{dt} = Ku + M_C\, q, \tag{5}$$

where $M_C = \{m_{ij}\}$ denotes the consistent mass matrix and $K = \{k_{ij}\}$ stands for the discrete transport operator. The matrix entries are given by

$$m_{ij} = \int_\Omega \varphi_i \varphi_j \, d\mathbf{x}, \qquad k_{ij} = -\mathbf{v}_j \cdot \mathbf{c}_{ij} - \epsilon\, s_{ij}, \tag{6}$$

where $\mathbf{c}_{ij}$ and s_{ij} result from the discretization of differential operators corresponding to the first- and second-order derivatives, respectively

$$\mathbf{c}_{ij} = \int_\Omega \varphi_i \nabla \varphi_j \, d\mathbf{x}, \qquad s_{ij} = \int_\Omega \nabla \varphi_i \cdot \nabla \varphi_j \, d\mathbf{x}. \tag{7}$$

Note that the coefficients m_{ij}, $\mathbf{c}_{ij}$, s_{ij} remain unchanged as long as the mesh is fixed. Therefore, they need to be determined just once during the initialization phase. This enables us to update the matrix K in a very efficient way by computing its entries k_{ij} from formula (6) without resorting to costly numerical integration. It is worth mentioning that this shortcut is not essential for the subsequent considerations, so that the assembly of the discrete transport operator can also be performed in the usual way.

3 Discrete upwinding

The FEM-FCT algorithm to be presented prevents the birth and growth of spurious extrema by blending the oscillatory Galerkin discretization with a monotone low-order method. The latter should be used in the vicinity of steep gradients, where nonphysical undershoots and overshoots are likely to arise. In the realm of finite differences and finite volumes, the upwind scheme is a perfect 'wiggle-killer'. At the same time, it has been largely unclear how to perform upwinding in the finite element framework. Most of the upwind-like finite element methods encountered in the literature resort to a finite volume discretization for the convective terms [2],[41]. An alternative derivation of the least diffusive positivity-preserving scheme can be carried out by adding discrete diffusion depending solely on the magnitude and position of negative matrix entries [16],[17]. This approach will be elucidated below for the scalar transport equation and extended to hyperbolic systems of conservation laws in the second part of this paper.

Let us perform mass lumping and represent the ODE for a nodal value u_i in the form

$$m_i \frac{du_i}{dt} = \sum_{j \neq i} k_{ij}(u_j - u_i) + r_i u_i + m_i q_i, \tag{8}$$

where $m_i = \sum_j m_{ij}$ and $r_i = \sum_j k_{ij}$. The first term in the right-hand side of (8) is engendered by the incompressible part of the discrete transport operator, while $r_i u_i$ is a discrete counterpart of $u \nabla \cdot \mathbf{v}$ which vanishes for divergence-free velocity fields.

For the numerical solution to be nonoscillatory even close to shocks and discontinuities, all off-diagonal coefficients of K must be nonnegative: $k_{ij} \geq 0,\ j \neq i$. This condition is necessary to enforce the M-matrix property (see below) and make the discretization *local extremum diminishing* (LED) for incompressible flows in the absence of source terms ($r_i = q_i = 0$). In this case, the semi-discrete scheme reduces to

$$\frac{du_i}{dt} = \sum_{j \neq i} c_{ij}(u_j - u_i), \qquad \text{where} \quad c_{ij} = \frac{k_{ij}}{m_i} \geq 0. \tag{9}$$

Such a discretization proves to be stable in the L_∞-norm. Indeed, if u_i is a maximum, then $u_j - u_i \leq 0,\ \forall j$, so that $\frac{du_i}{dt} \leq 0$. Hence, a maximum cannot increase, and similarly a minimum cannot decrease. As a rule, the coefficient matrices are sparse, so that $k_{ij} = 0$ unless i and j are adjacent nodes. Arguing as above, one can show that a *local* maximum cannot increase, and a *local* minimum cannot decrease. The LED criterion was introduced by Jameson [13],[14],[15] as a handy tool for the design of high-resolution schemes on unstructured meshes. It reduces to Harten's TVD conditions [9],[10] in the one-dimensional case but remains valid in multidimensions and is easy to verify.

Any discrete transport operator K can be rendered local extremum diminishing by adding a tensor of artificial diffusion $D = \{d_{ij}\}$ designed so as to eliminate its negative off-diagonal entries. The optimal diffusion coefficients are given by [17],[19]

$$d_{ii} = -\sum_{k \neq i} d_{ik}, \qquad d_{ij} = d_{ji} = \max\{0, -k_{ij}, -k_{ji}\}. \tag{10}$$

By construction, D is a *generalized diffusion operator* defined as a symmetric matrix having zero row and column sums [17]. Applying it to the vector u, we obtain

$$(Du)_i = \sum_j d_{ij} u_j = \sum_{j \neq i} d_{ij}(u_j - u_i) \tag{11}$$

due to the zero row sum. Therefore, diffusive terms can be decomposed into a sum of numerical fluxes which reduce the difference between the nodal values

$$(Du)_i = \sum_{j \neq i} f_{ij}, \qquad \text{where} \quad f_{ij} = d_{ij}(u_j - u_i). \tag{12}$$

Note that $f_{ji} = -f_{ij}$ due to the symmetry of D. Hence, the insertion of artificial diffusion does not violate the discrete conservation principle. This kind of flux decomposition is feasible for any generalized diffusion operator [17]. In the sequel, we will take advantage of this fact again and remove excessive diffusion in a mass-conserving fashion.

The elimination of negative matrix entries as proposed above yields the least diffusive LED scheme obtainable from the original Galerkin discretization. For the pure convection equation in one dimension, it is equivalent to the upwind difference method [17],[19]. Note that physical diffusion (if any) built into the coefficients k_{ij} is automatically detected and the amount of artificial diffusion is reduced accordingly. In diffusion-dominated cases, the discrete transport operators K and $L = K + D$ are identical, since the coefficients are nonnegative from the outset. Alternatively, this postprocessing technique can be applied to the convective part of K without taking the physical diffusion into account.

Discrete upwinding should be performed edge-by-edge in accordance with the sparsity structure of the finite element matrix. Let us start with the Galerkin operator $L = K$. For each pair of neighboring nodes i and j the required modification is as follows

$$\begin{aligned} l_{ii} &= l_{ii} - d_{ij}, \qquad & l_{ij} &= l_{ij} + d_{ij}, \\ l_{ji} &= l_{ji} + d_{ij}, \qquad & l_{jj} &= l_{jj} - d_{ij}. \end{aligned} \tag{13}$$

Let us emphasize that the LED constraint is imposed only on the incompressible part of the discrete transport operator K as it should be for physical reasons. The additional term $r_i u_i + m_i q_i$ in the right-hand side of (8) allows for an admissible growth and decay of local extrema due to compressibility and sources/sinks. In order to ensure that the positivity of thermodynamic variables is reproduced by the numerical solution, this term may need to be linearized as proposed by Patankar [34]. This procedure is described in our previous publications [19],[20] which should also be consulted for other algorithmic details and the underlying theory. For simplicity, we omit the source terms in what follows.

4 Defect correction

The equations at hand can be discretized in time by the standard θ-scheme. If implicit time-stepping ($0 < \theta \leq 1$) is employed, the nonlinearities inherent to the conservation law and/or to the numerical method must be treated iteratively. Let us update the solution in each outer iteration using the straightforward defect correction scheme

$$u^{(m+1)} = u^{(m)} + [A(u^{(m)})]^{-1} r^{(m)}, \qquad m = 0, 1, 2, \ldots \tag{14}$$

where $r^{(m)}$ denotes the residual vector for the m-th cycle and A is a suitably chosen 'preconditioner' which should be easy to invert. The iteration process is terminated when the norm of the defect or that of the relative changes is small enough.

In a practical implementation, the 'inversion' of A is also performed by some iterative procedure. Hence, a certain number of inner iterations per cycle is required to solve the linear subproblem for the solution increment which reads

$$A(u^{(m)}) \Delta u^{(m)} = r^{(m)}. \tag{15}$$

Afterwards, the correction $\Delta u^{(m)}$ is applied to the last iterate

$$u^{(m+1)} = u^{(m)} + \Delta u^{(m)}, \qquad u^{(0)} = u^n. \tag{16}$$

Note that the auxiliary problem (15) does not have to be solved very accurately at each outer iteration. A moderate improvement of the residual (1-2 digits) is sufficient to obtain a good overall accuracy. The low-order evolution operator

$$A(u^{(m)}) = M_L - \theta \Delta t L(u^{(m)}), \qquad \text{where} \quad M_L = \text{diag}\{m_i\}, \tag{17}$$

constitutes an excellent preconditioner. It can be readily verified that A is an M-matrix, which makes it amenable to iterative solution. Furthermore, the diagonal dominance of A can be enhanced by using an implicit underrelaxation strategy [6]. It will be noted that defect correction preconditioned by the monotone upwind operator is widely used to enhance the robustness of CFD solvers even in the linear case. This is due to the fact that an iterative method may fail to converge if applied directly to the ill-conditioned matrix originating from a high-order discretization of the bad-behaved convective terms.

The defect vector $r^{(m)}$ can be designed so as to obtain the standard Galerkin approximation, the diffusive low-order solution or a nonlinear combination thereof. The finite element discretizations of high and low order are related by the formula

$$r^{(m)} = b^n + f(u^n, u^{(m)}) - A(u^{(m)}) u^{(m)}, \tag{18}$$

where b^n represents the right-hand side for the low-order scheme

$$b^n = M_L u^n + (1 - \theta) \Delta t L(u^n) u^n, \tag{19}$$

while the *antidiffusion* required to recover the high-order solution is given by

$$\begin{aligned} f(u^n, u^{(m)}) &= [(M_C - M_L) - (1 - \theta) \Delta t D(u^n)] u^n \\ &- [(M_C - M_L) + \theta \Delta t D(u^{(m)})] u^{(m)}. \end{aligned} \tag{20}$$

Here $D = L - K$ denotes the artificial diffusion operator which transforms the oscillatory high-order transport operator into the monotone low-order one as explained above.

By construction, D is a symmetric matrix with zero row and column sums, and so is the operator $M_C - M_L$ which is sometimes referred to as 'mass diffusion'. Therefore, both expressions in the brackets represent discrete diffusion operators, so that the antidiffusive terms can be decomposed into skew-symmetric internodal fluxes of the form

$$f_{ij}^{(m)} = [m_{ij} - (1-\theta)\Delta t d_{ij}^n]\ (u_j^n - u_i^n) - [m_{ij} + \theta \Delta t d_{ij}^{(m)}]\ (u_j^{(m)} - u_i^{(m)}) = -f_{ji}^{(m)}. \quad (21)$$

These raw antidiffusive fluxes offset the error induced by mass lumping and discrete upwinding. In the fully explicit case ($\theta = 0$) the convective terms must be stabilized by means of a streamline diffusion operator which also admits a flux decomposition [17],[19]. In this paper we restrict ourselves to implicit finite element methods which do not require any extra stabilization as far as the discretization is concerned. Linear Galerkin schemes of this type are of little use, as they are prone to nonphysical oscillations and lead to finite element matrices with extremely unfavorable properties. At the same time, they constitute viable high-order methods for the flux-corrected-transport algorithm.

5 Iterative FEM-FCT formulation

The essence of flux correction consists in adding as much antidiffusion as possible without generating spurious wiggles. In particular, it is important to guarantee that quantities like densities, temperatures, concentrations etc. remain strictly nonnegative. It is obvious that LED methods do satisfy this physically motivated constraint at the semi-discrete level. However, the influence of the time discretization must also be taken into account. A fully discrete scheme is positivity-preserving if it can be represented in the form

$$Au^{n+1} = Bu^n, \qquad u^n \geq 0, \quad (22)$$

where A is an M-matrix and all entries of B are nonnegative [18],[19]. Building on this positivity criterion, we will derive a nonlinear high-resolution scheme.

To begin with, let us analyze the defect correction method introduced above for the mixture of high- and low-order discretizations. The substitution of (18) into (14) gives

$$A(u^{(m)})u^{(m+1)} = b^n + f(u^n, u^{(m)}). \quad (23)$$

Recall that the low-order preconditioner A was designed to be an M-matrix. Hence, it suffices to verify the positivity condition for the right-hand side. To this end, we introduce an auxiliary quantity $\tilde{u}^n$ as the solution to the explicit subproblem

$$M_L \tilde{u}^n = M_L u^n + (1-\theta)\Delta t L(u^n)u^n. \quad (24)$$

In fact, $\tilde{u}^n$ corresponds to an intermediate solution computed at the time instant $t^{n+1-\theta}$ by the explicit low-order method. Note that $\tilde{u}^n = u^n$ in the fully implicit case $\theta = 1$. Other time-stepping schemes preserve the positivity of u^n provided that [17],[19]

$$\Delta t \leq \frac{1}{1-\theta} \min_i \{-m_i/l_{ii} |\ l_{ii} < 0\}. \quad (25)$$

This readily computable upper bound follows from the positivity condition set forth above and can serve as a threshold value for an adaptive time step control.

According to (22), the positivity of u^n and $\tilde{u}^n$ is inherited by $u^{(m+1)}$ if we drop the second term in the right-hand side of (23) and end up with $b^n = M_L \tilde{u}^n$. On the other hand, the accuracy of the finite element discretization can be dramatically improved if we retain a certain portion of the antidiffusive fluxes in regions where the solution is sufficiently smooth. The family of implicit FEM-FCT schemes proposed in [16],[17] is based on the following representation of the right-hand side

$$b_i^{(m+1)} = b_i^n + \sum_{j \neq i} \alpha_{ij}^{(m)} f_{ij}^{(m)}, \qquad \alpha_{ij}^{(m)} = \alpha_{ij}(\tilde{u}^n, f_{ij}^{(m)}). \tag{26}$$

The involved correction factors $\alpha_{ij}^{(m)}$ depend on the auxiliary solution $\tilde{u}^n$ and on the interplay of raw antidiffusive fluxes. They are determined using the universal limiting strategy presented in the next section. Roughly speaking, the flux correction step makes sure that there exists a matrix B with no negative entries such that $b^{(m+1)} = B\tilde{u}^n$. This enables us to represent the discrete scheme in the desired form (22) with $\tilde{u}^n$ in lieu of u^n.

Note that $\tilde{u}^n$ is independent of the iteration number m and does not have to be recalculated in the course of defect correction. Unfortunately, there is a price to be paid for this convenience. Namely, the numerical solution becomes increasingly diffusive at large time steps. Equation (21) reveals that the spatial contribution to the raw antidiffusive flux is proportional to the time step. At the same time, the amount of acceptable antidiffusion depends solely on $\tilde{u}^n$, so that the flux must be drastically curtailed as its magnitude increases with the time step. This is a serious drawback, since the ability to operate with large time steps was the main reason for using an implicit scheme in the first place.

In order to alleviate the dependence of the final solution on the time step, we resort to an iterative FEM-FCT formulation, whereby only the rejected portion of the antidiffusive flux needs to be dealt with. This prevents the flux limiter from returning to the worst case scenario and makes the choice of correction factors at one particular iteration less critical. A somewhat similar technique was developed by Schär and Smolarkiewicz [37] in the finite difference context. However, their flux correction formalism is inherently explicit, so that the advantages of using an iterative procedure are rather questionable for efficiency reasons. In addition, Schär and Smolarkiewicz determine the upper and lower bounds using the old solution rather than the low-order one. This practice may turn out to be rather dangerous, especially for problems with sink terms [17],[19].

In our new approach, the accepted antidiffusion is incorporated into the intermediate solution $\tilde{u}^{(m)}$ which is no longer fixed and must be updated along with the right-hand side $b^{(m)}$ for the previous defect correction step

$$M_L \tilde{u}^{(m)} = b^{(m)}, \qquad b^{(0)} = b^n. \tag{27}$$

Recall that M_L is a diagonal matrix so that no linear system has to be solved and the overhead cost associated with the computation of $\tilde{u}^{(m)}$ is negligible.

The limiting strategy remains unchanged except for the fact that the correction factors are based on the local extrema of $\tilde{u}^{(m)}$ rather than $\tilde{u}^n$ and applied to the difference between the raw antidiffusive fluxes and the net effect of previous corrections

$$\Delta f_{ij}^{(m)} = f_{ij}^{(m)} - g_{ij}^{(m)}, \qquad \alpha_{ij}^{(m)} = \alpha_{ij}(\tilde{u}^{(m)}, \Delta f_{ij}^{(m)}). \tag{28}$$

Subsequently, the limited flux difference is added to the sum of its predecessors

$$g_{ij}^{(m+1)} = g_{ij}^{(m)} + \alpha_{ij}^{(m)} \Delta f_{ij}^{(m)} = \sum_{k=0}^{m} \alpha_{ij}^{(k)} \Delta f_{ij}^{(k)}, \qquad g_{ij}^{(0)} = 0 \tag{29}$$

and inserted into the global load vector for the next iteration

$$b_i^{(m+1)} = b_i^{(m)} + \sum_{j \neq i} \alpha_{ij}^{(m)} \Delta f_{ij}^{(m)}. \tag{30}$$

In the special case $\alpha_{ij}^{(m)} \equiv 1$, successive substitution yields

$$b_i^{(m+1)} = b_i^n + \sum_{j \neq i} (g_{ij}^{(m)} + \Delta f_{ij}^{(m)}) = b_i^n + \sum_{j \neq i} f_{ij}^{(m)}. \tag{31}$$

Therefore, our iterative FEM-FCT technique proves to be consistent in the sense that it reduces to the standard Galerkin discretization if no limiting is performed.

It is worth mentioning that (30) is of the same form as (26) and even equivalent to it for $m = 0$. As before, the flux limiter ensures that $b^{(m+1)} = B\tilde{u}^{(m)}$ for some $B \geq 0$, so that the transformation $\tilde{u}^{(m+1)} = M_L^{-1} B \tilde{u}^{(m)}$ as defined by the inverse diffusion problem (27) is positivity-preserving. Note that the classical FCT algorithm with $u^{n+1} = \tilde{u}^{(1)}$ is recovered in the fully explicit case (forward Euler time-stepping, lumped mass matrix). As the iteration process continues, more and more antidiffusion can be built into the intermediate solution. At the same time, the task of the flux limiter simplifies, because the remainder $\Delta f_{ij}^{(m)}$ shrinks and a larger percentage of it can be accepted. This is in contrast to the old approach, whereby the limiting procedure always starts from scratch without taking the outcome of previous flux correction steps into account.

In either case, the defect vector for the linear system (15) is redefined as

$$r^{(m)} = b^{(m+1)} - A(u^{(m)})u^{(m)}, \tag{32}$$

where $b^{(m+1)}$ consists of the low-order part b^n and nonlinear antidiffusion. It remains to describe the algorithm for the computation of correction factors and explain how it works.

6 Limiting strategy

The flux limiter is a key element of the FEM-FCT paradigm. As already pointed out above, it is supposed to adjust the correction factors $\alpha_{ij} = \alpha_{ji}$ so as to guarantee that

$$b_i = m_i \tilde{u}_i + \sum_{j \neq i} \alpha_{ij} f_{ij} \geq 0 \qquad \text{if} \quad \tilde{u} \geq 0. \tag{33}$$

Varying α_{ij} between zero and unity, one obtains the diffusive low-order solution, the oscillatory high-order solution or something in-between. If $\tilde{u}_i$ is a local extremum, then the antidiffusive fluxes trying to enhance it must be canceled completely by setting $\alpha_{ij} = 0$. Otherwise, the following representation is feasible [16],[19]

$$b_i = (m_i - c_i)\tilde{u}_i + c_i \tilde{u}_k, \qquad \text{where} \quad c_i = \frac{\sum_{j \neq i} \alpha_{ij} f_{ij}}{\tilde{u}_k - \tilde{u}_i} \tag{34}$$

and k is the number of a neighboring node at which a local extremum is attained.

A sufficient condition for the positivity constraint to be satisfied is given by the double inequality $m_i \geq c_i \geq 0$. Let $\tilde{u}_i^{\max}$ and $\tilde{u}_i^{\min}$ denote the maximum and minimum solution values at the stencil S_i which consists of node i and its nearest neighbors

$$\tilde{u}_i^{\substack{\max \\ \min}} = \begin{matrix}\max \\ \min\end{matrix} \tilde{u}_j, \qquad j \in S_i. \tag{35}$$

To make sure that $c_i \geq 0$, we adopt $\tilde{u}_k = \tilde{u}_i^{\max}$ if the net antidiffusive flux $\sum_{j \neq i} \alpha_{ij} f_{ij}$ is positive and $\tilde{u}_k = \tilde{u}_i^{\min}$ otherwise. The remaining condition $m_i \geq c_i$ must be enforced by tuning the correction factors. This can be accomplished by invoking Zalesak's FCT limiter which was originally derived using heuristic arguments and applicable only in the framework of an explicit time discretization.

Antidiffusive fluxes directed down the gradient of $\tilde{u}$ tend to flatten solution profiles and should be canceled completely at the outset of the flux correction process

$$f_{ij} := 0, \quad \text{if} \quad f_{ij}(\tilde{u}_i - \tilde{u}_j) \leq 0. \tag{36}$$

In other words, an antidiffusive flux should not be allowed to act as a diffusive one. This optional *prelimiting step* can be traced back to the original SHASTA scheme [3]. Zalesak mentioned it as well but argued that the resulting improvement is marginal and cosmetic in nature [44]. However, without this amendment his multidimensional limiter is positivity- but not monotonicity-preserving, so that the numerical solution may exhibit various artifacts in the vicinity of shocks and discontinuities [4],[17],[19].

Following Zalesak [44] and Löhner [27], we introduce the auxiliary quantities

$$P_i^{\pm} = \frac{1}{m_i} \sum_{j \neq i} \begin{matrix}\max \\ \min\end{matrix} \{0, f_{ij}\}, \qquad Q_i^{\pm} = \tilde{u}_i^{\substack{\max \\ \min}} - \tilde{u}_i, \tag{37}$$

which represent the sum of all positive/negative antidiffusive fluxes into node i and the distance to the local extremum, respectively. In the worst case, all contributions are of the same sign, so that the flux f_{ij} should be multiplied by

$$R_i^{\pm} = \begin{cases} \min\{1, Q_i^{\pm}/P_i^{\pm}\}, & \text{if } P_i^{\pm} \neq 0, \\ 1, & \text{if } P_i^{\pm} = 0. \end{cases} \tag{38}$$

Recall that a positive flux f_{ij} into node i is always balanced by a negative flux $f_{ji} = -f_{ij}$ into node j and vice versa. Hence, the limiter must inspect the sign of the flux and take the minimum of the nodal correction factors

$$\alpha_{ij} = \begin{cases} \min\{R_i^+, R_j^-\}, & \text{if } f_{ij} \geq 0, \\ \min\{R_j^+, R_i^-\}, & \text{if } f_{ij} < 0. \end{cases} \tag{39}$$

Note that $Q_i^{\pm} = 0$ implies $R_i^{\pm} = 0$ and $\alpha_{ij} = 0$. Hence, any enhancement of local extrema is prohibited by the limiter. Furthermore, the following estimate holds [17],[19]

$$m_i Q_i^- \leq m_i R_i^- P_i^- \leq \sum_{j \neq i} \alpha_{ij} f_{ij} \leq m_i R_i^+ P_i^+ \leq m_i Q_i^+. \tag{40}$$

This proves that the corrected antidiffusive fluxes satisfy the constraint $m_i \geq c_i$.

7 Boundary treatment

It was reported in [17] that an FCT algorithm based on Zalesak's limiter may malfunction for smooth solutions having a nonvanishing gradient at the boundary. In this case, small ripples may pop up at the outlet and propagate into the interior of the domain as illustrated in Figure 1 (left) for the one-dimensional convection equation with $v = 1,\ u^0 = x$. Similar problems occur at the inlet but their detrimental effect is alleviated by the Dirichlet boundary condition. Some preliminary speculations regarding the origin of spurious ripples can be found in [17]. Recently, an explanation of this phenomenon was given by Möller [32] who observed that formula (35) yields a poor estimate of upper/lower bounds at inflow and outflow boundaries. Namely, $\tilde{u}_i$ can be misinterpreted as a local extremum because it is only compared to nodal values of the solution *within the domain*. As long as no information regarding the solution behavior beyond the open boundary is available, it is impossible to tell whether or not $\tilde{u}_i^{\max}$ would be an extremum for the extended domain.

As a rule of thumb, antidiffusive fluxes should neither create new minima/maxima, nor accentuate already existing ones [44]. Ironically, this fundamental law of flux correction turns out to be the cause of troubles at the boundary. The faulty flux limiter erroneously anticipates the enhancement of a local extremum and switches to the diffusive low-order solution, which entails a redistribution of mass due to the discrete conservation principle. Changing the solution value at one node is accompanied by an adjustment of those at the neighboring nodes so that the total mass remains unchanged. To illustrate this effect, the so-called *lever model* was introduced in [32]. Let the piecewise-linear solution be represented by levers of variable length hinged at their midpoints, which correspond to the element mean values, and connected continuously with one another. Pulling down the rightmost lever results in a shearing force which affects the slopes of all components as shown in Figure 1 (right). Furthermore, the implications of the optional prelimiting step (36) can be explained in a similar way. Roughly speaking, diffusive fluxes rotate the levers in the wrong direction, which results in the formation of kinks and flat plateaus amidst a steep front. For further details, the interested reader is referred to [32].

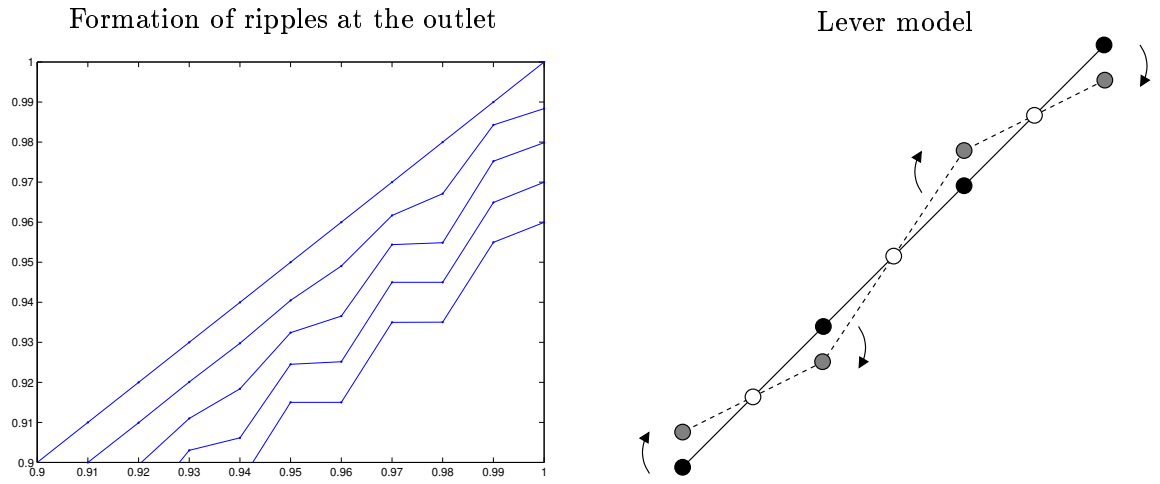


Figure 1. Pathological behavior of Zalesak's limiter.

To cure the pathological behavior of Zalesak's limiter, we reset the nodal correction factors which are based on questionable upper/lower bounds

$$R_i^{\pm} := 1 \qquad \forall i \in N_b, \tag{41}$$

where N_b denotes the set of nodes belonging to the inflow and outflow boundaries. At the inlet, it is reasonable to do so anyway because the essential boundary condition overrides the contribution of incoming antidiffusive fluxes. At the outlet, the off-diagonal coefficients of the high-order transport operator are typically nonnegative [17],[22] so this modification does not pose any hazard to positivity either. In addition, any undershoots or overshoots generated there would be immediately convected away. The newly introduced *postlimiting step* should follow (38) and precede (39) in a practical implementation. This simple remedy produces the desired effect at virtually no additional cost.

8 Summary of the FEM-FCT algorithm

The revised Zalesak's limiter completes our iterative FEM-FCT formulation which is applicable to nonlinear conservation laws, multidimensional problems, unstructured finite element meshes and implicit time-stepping schemes. It can be implemented as a black-box postprocessing routine which operates with discrete operators regardless of their origin. The modifications are introduced in a sweep over numerical edges which correspond to pairs of basis functions with overlapping supports. The list of edges is determined by the sparsity pattern of the finite element matrix. Specifically, each edge $\vec{ij}$ is associated with a nonzero off-diagonal coefficient a_{ij} in the upper/lower triangular part. The main algorithmic steps to be performed in the scalar case can be summarized as follows:

In the matrix assembly routine:

1. Retrieve/compute the entries k_{ij} and k_{ji} of the high-order transport operator.
2. Determine the corresponding artificial diffusion coefficient d_{ij} from equation (10).
3. Substitute the diffusive flux (12) into the right-hand side b^n (at the first iteration).
4. Update the four entries of the upwind-biased preconditioner A as required by (13).
5. Compute the antidiffusive flux (21) or (28) to be limited and inserted into $b^{(m+1)}$.

In the flux correction module:

6. Initialize/update $\tilde{u}$ by solving the explicit subproblem (24) or (27).
7. Invoke Zalesak's limiter with pre- and postlimiting to compute α_{ij} from (39).
8. Insert the limited antidiffusive fluxes into expressions (26) or (29),(30).

In the defect correction loop:

9. Solve the linear system (15) with the defect vector $r^{(m)}$ given by (32).
10. Update the solution according to (16) and proceed to the next iteration.

In what follows, we generalize the above algorithm to systems of hyperbolic conservation laws and elucidate its implementation for the Euler equations of gas dynamics.

9 Euler equations

Compressible flows at high velocities are governed by the Euler equations which represent a system of conservation laws for the mass, momentum and energy of an inviscid fluid

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0, \tag{42}$$

$$\frac{\partial (\rho \mathbf{v})}{\partial t} + \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v}) + \nabla p = 0, \tag{43}$$

$$\frac{\partial (\rho E)}{\partial t} + \nabla \cdot (\rho h \mathbf{v}) = 0, \tag{44}$$

where ρ, $\mathbf{v}$, p, E and $h = E + p/\rho$ stand for the density, velocity, pressure, total energy per unit mass and stagnation enthalpy, respectively. This system is complemented by an equation of state which relates the energy, pressure and density, for instance

$$p = (\gamma - 1)\rho \left(E - \frac{|\mathbf{v}|^2}{2} \right). \tag{45}$$

Here γ denotes the ratio of specific heats for a polytropic gas ($\gamma = 1.4$ for air).

Introducing the vector of conservative variables U and the triple of fluxes for each coordinate direction $\mathbf{F} = (F^1, F^2, F^3)$ defined as follows

$$U = \begin{bmatrix} \rho \\ \rho v_1 \\ \rho v_2 \\ \rho v_3 \\ \rho E \end{bmatrix}, \quad F^1 = \begin{bmatrix} \rho v_1 \\ \rho v_1^2 + p \\ \rho v_1 v_2 \\ \rho v_1 v_3 \\ \rho h v_1 \end{bmatrix}, \quad F^2 = \begin{bmatrix} \rho v_2 \\ \rho v_1 v_2 \\ \rho v_2^2 + p \\ \rho v_2 v_3 \\ \rho h v_2 \end{bmatrix}, \quad F^3 = \begin{bmatrix} \rho v_3 \\ \rho v_1 v_3 \\ \rho v_2 v_3 \\ \rho v_3^2 + p \\ \rho h v_3 \end{bmatrix} \tag{46}$$

we can represent the Euler equations in the standard divergence form

$$\frac{\partial U}{\partial t} + \nabla \cdot \mathbf{F} = 0, \qquad \text{where} \quad \nabla \cdot \mathbf{F} = \sum_{d=1}^{3} \frac{\partial F^d}{\partial x_d}. \tag{47}$$

Furthermore, the chain rule yields a quasi-linear formulation, in which the spatial derivatives are applied to the conservative variables rather than fluxes. The resulting system can be written in terms of the Jacobian matrices $\mathbf{A} = (A^1, A^2, A^3)$ such that [12]

$$\frac{\partial U}{\partial t} + \mathbf{A} \cdot \nabla U = 0, \qquad \text{where} \quad \mathbf{A} \cdot \nabla U = \sum_{d=1}^{3} A^d \frac{\partial U}{\partial x_d}. \tag{48}$$

It is worth mentioning that the fluxes are homogeneous functions of the conservative variables, so that the relationship between the components of $\mathbf{F}$ and $\mathbf{A}$ is given by

$$F^d = A^d U, \qquad A^d = \frac{\partial F^d}{\partial U}, \qquad d = 1, 2, 3. \tag{49}$$

Due to the hyperbolicity of the Euler equations, any linear combination of the three Jacobians is diagonalizable with real eigenvalues. This is a very useful property which will lead us to a natural generalization of the discrete upwinding technique.

10 Galerkin matrix assembly

Implicit finite element methods are rarely used for numerical simulation of compressible flows. Therefore, the issue of efficient matrix assembly has received little attention in the literature. In this section, we present an edge-based assembly technique for the standard Galerkin discretization of the Euler equations. Let us start with the divergence form (47) and interpolate the fluxes using the group finite element formulation which yields

$$\sum_j \left[\int_\Omega \varphi_i \varphi_j \, d\mathbf{x}\right] \frac{dU_j}{dt} + \sum_j \left[\int_\Omega \varphi_i \nabla \varphi_j \, d\mathbf{x}\right] \cdot \mathbf{F}_j = 0. \tag{50}$$

In order to obtain a (non-)linear system of the form $AU = B$ after an implicit time discretization, we should represent the semi-discrete problem as follows

$$M_C \frac{dU}{dt} = KU, \tag{51}$$

where M_C is the block-diagonal mass matrix for the coupled system and K is a discrete counterpart of the operator $-\mathbf{A} \cdot \nabla$ for the quasi-linear formulation (48).

Recall that the basis functions sum to unity, so that the sum of their derivatives vanishes. Hence, the coefficients $\mathbf{c}_{ij}$ defined in (7) satisfy $\mathbf{c}_{ii} = -\sum_{j \neq i} \mathbf{c}_{ij}$ and it follows from (50) that the right-hand side of the five coupled equations for node i is given by

$$(KU)_i = -\sum_j \mathbf{c}_{ij} \cdot \mathbf{F}_j = -\sum_{j \neq i} \mathbf{c}_{ij} \cdot (\mathbf{F}_j - \mathbf{F}_i). \tag{52}$$

Furthermore, it was shown in Roe's pioneering work on approximate Riemann solvers for hyperbolic conservation laws [36] that $\mathbf{F}_j - \mathbf{F}_i = \hat{\mathbf{A}}_{ij}(U_j - U_i)$, where $\hat{\mathbf{A}}_{ij} = (\hat{A}^1_{ij}, \hat{A}^2_{ij}, \hat{A}^3_{ij})$ corresponds to the continuous Jacobian $\mathbf{A}$ evaluated at the intermediate state

$$\hat{\rho}_{ij} = \sqrt{\rho_i \rho_j}, \qquad \hat{\mathbf{v}}_{ij} = \frac{\sqrt{\rho_i}\mathbf{v}_i + \sqrt{\rho_j}\mathbf{v}_j}{\sqrt{\rho_i} + \sqrt{\rho_j}}, \qquad \hat{h}_{ij} = \frac{\sqrt{\rho_i} h_i + \sqrt{\rho_j} h_j}{\sqrt{\rho_i} + \sqrt{\rho_j}}. \tag{53}$$

The density-averaged quantities $\hat{\rho}_{ij}$, $\hat{\mathbf{v}}_{ij}$ and $\hat{h}_{ij}$ are called the Roe mean values.

In light of the above, equation (52) admits the following representation

$$(KU)_i = -\sum_{j \neq i} \mathbf{c}_{ij} \cdot \hat{\mathbf{A}}_{ij}(U_j - U_i), \quad \text{where} \quad \mathbf{c}_{ij} \cdot \hat{\mathbf{A}}_{ij} = \sum_{d=1}^{3} c^d_{ij} \hat{A}^d_{ij}. \tag{54}$$

The dot product can be interpreted as a 'projection' of the triple $\hat{\mathbf{A}}_{ij}$ onto the numerical edge $\vec{ij}$. For our purposes, it is expedient to introduce the splitting $\mathbf{c}_{ij} \cdot \hat{\mathbf{A}}_{ij} = -(\mathrm{A}_{ij} + \mathrm{B}_{ij})$, where the two components of the *cumulative Roe matrix* are defined by [20]

$$\mathrm{A}_{ij} = \mathbf{a}_{ij} \cdot \hat{\mathbf{A}}_{ij}, \qquad \mathbf{a}_{ij} = -\frac{\mathbf{c}_{ij} - \mathbf{c}_{ji}}{2}, \tag{55}$$

$$\mathrm{B}_{ij} = \mathbf{b}_{ij} \cdot \hat{\mathbf{A}}_{ij}, \qquad \mathbf{b}_{ij} = -\frac{\mathbf{c}_{ij} + \mathbf{c}_{ji}}{2}. \tag{56}$$

A similar decomposition can be performed for the contribution of node i to $(KU)_j$

$$\mathbf{c}_{ji} \cdot (\mathbf{F}_j - \mathbf{F}_i) = \mathbf{c}_{ji} \cdot \hat{\mathbf{A}}_{ij}(U_j - U_i), \quad \text{where} \quad \mathbf{c}_{ji} \cdot \hat{\mathbf{A}}_{ij} = \mathrm{A}_{ij} - \mathrm{B}_{ij}. \tag{57}$$

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

$$\mathbf{c}_{ij} + \mathbf{c}_{ji} = \int_\Gamma \mathbf{n}\, \varphi_i \varphi_j \, ds, \tag{58}$$

where $\mathbf{n}$ denotes the outward unit normal to the boundary. Thus, we have

$$\mathbf{a}_{ij} = -\mathbf{c}_{ij} + \frac{1}{2}\int_\Gamma \mathbf{n}\, \varphi_i \varphi_j \, ds, \qquad \mathbf{b}_{ij} = -\frac{1}{2}\int_\Gamma \mathbf{n}\, \varphi_i \varphi_j \, ds. \tag{59}$$

For typical finite elements (linear and multilinear) the surface integral is nonzero only if both nodes belong to the boundary. Hence, $\mathbf{a}_{ij} = -\mathbf{c}_{ij}$ in the interior of the domain, while $\mathbf{b}_{ij} = 0$ and its contribution B_{ij} to the local Jacobian vanishes. This means that just one cumulative Roe matrix, namely A_{ij}, needs to be evaluated for each interior edge. At the boundary, one obtains $\mathbf{b}_{ij} = -\mathbf{n}\, t_{ij}/2$, where $t_{ij} = \int_\Gamma \varphi_i \varphi_j \, ds$ corresponds to an off-diagonal entry of the mass matrix for the surface triangulation. The Roe matrices for boundary edges consist of a skew-symmetric part A_{ij} and a symmetric part B_{ij}.

According to (54)–(57), the contribution of the edge $\vec{ij}$ to the term KU reads

$$(\mathrm{A}_{ij} + \mathrm{B}_{ij})(U_j - U_i) \quad \longrightarrow \quad (KU)_i \tag{60}$$

$$(\mathrm{A}_{ij} - \mathrm{B}_{ij})(U_j - U_i) \quad \longrightarrow \quad (KU)_j \tag{61}$$

This representation leads to a very efficient edge-based algorithm for matrix assembly. Now as before, there is no need for numerical integration as long as the coefficients $\mathbf{c}_{ij}$ are initialized and stored. The graph representing the connectivity of the global matrix depends solely on the underlying mesh and on the type of finite element approximation. For systems of equations, the array of edges remains the same as in the scalar case. However, there are interactions not only between basis functions for different nodes but also between basis functions for different variables. Hence, each coefficient of the discrete operator turns into a matrix of size equal to the squared number of variables.

It can be readily inferred from (60)–(61) that the four 5×5 blocks contributed by the numerical edge $\vec{ij}$ to the global Jacobian matrix $K \in \mathbb{R}^{5N \times 5N}$ are given by

$$\begin{aligned} \mathrm{K}_{ii} &= -\mathrm{A}_{ij} - \mathrm{B}_{ij}, & \mathrm{K}_{ij} &= \mathrm{A}_{ij} + \mathrm{B}_{ij}, \\ \mathrm{K}_{ji} &= -\mathrm{A}_{ij} + \mathrm{B}_{ij}, & \mathrm{K}_{jj} &= \mathrm{A}_{ij} - \mathrm{B}_{ij}. \end{aligned} \tag{62}$$

These local Jacobians are evaluated edge-by-edge and their entries K_{ij}^{kl} $(k, l = 1, \ldots, 5)$ are scattered to the corresponding positions in the 25 blocks $K_{kl} \in \mathbb{R}^{N \times N}$. The assembly process is illustrated in Figure 2. In practice, it is not necessary to assemble and store the gigantic global matrix completely. Due to the nonlinearity of the Euler equations, the use of an iterative solution technique is indispensable. If we employ the defect correction method with a block-diagonal preconditioner (see below) then just the five blocks K_{kk} will need to be generated explicitly. The remaining part of the matrix will be represented by the edge contributions (60)–(61) to be inserted directly into the defect vector.

11 Artificial viscosity

To a large extent, the ability of a FEM-FCT algorithm to withstand the formation of wiggles depends on the quality of the underlying low-order method. In order to get rid of oscillations, we perform mass lumping and replace the high-order system (51) by

$$M_L \frac{dU}{dt} = LU, \tag{63}$$

where M_L denotes the lumped mass matrix and L is the low-order Jacobian operator. Recall that we derived its scalar counterpart by elimination of negative off-diagonal entries from the Galerkin operator. Now this discrete upwinding technique needs to be extended to hyperbolic systems. In this case, the edge contributions to the global finite element matrix are no longer scalar quantities but matrices themselves. As a natural generalization of the LED criterion, we require that all off-diagonal matrix blocks be positive definite.

To achieve the desired effect, we add tensorial artificial viscosity $\mathrm{D}_{ij} \in \mathbb{R}^{5\times 5}$ to the Roe matrices in (62). This straightforward transformation leads to

$$\begin{aligned} \mathrm{L}_{ii} &= -\mathrm{A}_{ij} - \mathrm{D}_{ij}, & \mathrm{L}_{ij} &= \mathrm{A}_{ij} + \mathrm{D}_{ij}, \\ \mathrm{L}_{ji} &= -\mathrm{A}_{ij} + \mathrm{D}_{ij}, & \mathrm{L}_{jj} &= \mathrm{A}_{ij} - \mathrm{D}_{ij}, \end{aligned} \tag{64}$$

which corresponds to (13) in the scalar case. The modified edge contributions are built into the blocks of L as explained above. In essence, D_{ij} supersedes the symmetric part B_{ij} of the cumulative Roe matrix which belongs into the raw antidiffusive flux

$$\mathrm{F}_{ij} = -\left(\mathrm{M}_{ij}\frac{d}{dt} + \mathrm{D}_{ij} - \mathrm{B}_{ij}\right)(U_j - U_i), \qquad \mathrm{F}_{ji} = -\mathrm{F}_{ij}. \tag{65}$$

The block $\mathrm{M}_{ij} = m_{ij} I$, where I denotes the 5×5 identity matrix, is responsible for the error induced by mass lumping. After the time discretization, one obtains an expression similar to (21). It remains to design D_{ij} so as to satisfy the generalized positivity constraint.

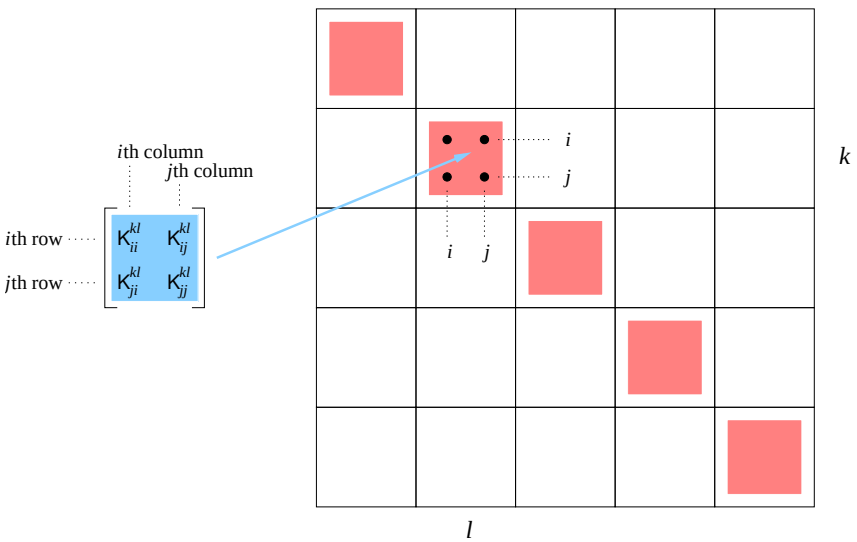


Figure 2. Edge-based matrix assembly for the Euler equations.

As already mentioned, the system of Euler equations is hyperbolic, so that any linear combination of the three Jacobian matrices is diagonalizable with real eigenvalues. Thus, there exists a diagonal matrix $\Lambda(\mathbf{a}_{ij})$ and a regular matrix $R(\mathbf{a}_{ij})$ of right eigenvectors such that the cumulative Roe matrix A_{ij} admits the following factorization

$$\mathrm{A}_{ij} = R(\mathbf{a}_{ij})\Lambda(\mathbf{a}_{ij})R(\mathbf{a}_{ij})^{-1}, \tag{66}$$

where diagonal entries of the matrix

$$\Lambda(\mathbf{a}_{ij}) = |\mathbf{a}_{ij}|\mathrm{diag}\left\{\hat{v}_{ij} - \hat{c}_{ij}, \hat{v}_{ij}, \hat{v}_{ij}, \hat{v}_{ij}, \hat{v}_{ij} + \hat{c}_{ij}\right\} \tag{67}$$

are proportional to the eigenvalues of A_{ij}. The scaling factor $|\mathbf{a}_{ij}|$ stands for the Euclidean norm of the coefficient vector $\mathbf{a}_{ij}$, while the auxiliary quantities

$$\hat{v}_{ij} = \frac{\mathbf{a}_{ij} \cdot \hat{\mathbf{v}}_{ij}}{|\mathbf{a}_{ij}|}, \qquad \hat{c}_{ij} = \sqrt{(\gamma - 1)\left(\hat{h}_{ij} - \frac{|\hat{\mathbf{v}}_{ij}|^2}{2}\right)} \tag{68}$$

represent the 'projection' of the density-averaged velocity $\hat{\mathbf{v}}_{ij}$ onto the numerical edge $\vec{ij}$ and the local speed of sound for this edge, respectively.

Making use of the characteristic decomposition (66), we can eliminate the negative eigenvalues of A_{ij} and define the tensor D_{ij} as follows

$$\mathrm{D}_{ij} = |\mathrm{A}_{ij}| = R(\mathbf{a}_{ij})\,|\Lambda(\mathbf{a}_{ij})|\,R(\mathbf{a}_{ij})^{-1}, \tag{69}$$

where the matrix $|\Lambda(\mathbf{a}_{ij})|$ contains the absolute values of the eigenvalues

$$|\Lambda(\mathbf{a}_{ij})| = |\mathbf{a}_{ij}|\mathrm{diag}\left\{|\hat{v}_{ij} - \hat{c}_{ij}|, |\hat{v}_{ij}|, |\hat{v}_{ij}|, |\hat{v}_{ij}|, |\hat{v}_{ij} + \hat{c}_{ij}|\right\}. \tag{70}$$

This kind of artificial viscosity is frequently employed to construct upwind-biased finite difference schemes for hyperbolic systems [12],[24],[31]. The approach presented in this paper enables us to incorporate it into a finite element discretization and obtain an analog of Roe's approximate Riemann solver based on flux difference splitting.

As an alternative, discrete upwinding for systems of equations can be performed in the framework of the conservative flux decomposition outlined in [19],[20],[22]

$$(KU)_i = \sum_{j \neq i} \mathrm{G}_{ij}, \qquad \text{where} \quad \mathrm{G}_{ij} = \mathbf{c}_{ji} \cdot \mathbf{F}_j - \mathbf{c}_{ij} \cdot \mathbf{F}_i. \tag{71}$$

Note that the so-defined *Galerkin fluxes* are skew-symmetric: $\mathrm{G}_{ji} = -\mathrm{G}_{ij}$. On a uniform 1D mesh, $c_{ji} = -c_{ij} = 1/2$ for linear finite elements, so that $\mathrm{G}_{ij} = (F_i + F_j)/2$. This is due to the well-known fact that the Galerkin method results in a central difference type approximation. The replacement of G_{ij} by the consistent numerical flux

$$\mathrm{G}^*_{ij} = \mathrm{G}_{ij} + \frac{1}{2}\mathrm{D}_{ij}(U_j - U_i) \tag{72}$$

yields the same nonoscillatory low-order scheme as the one derived above [19],[20].

Our generalization of Roe's approximate Riemann solver constitutes a viable method *per se* but it results in considerable overhead costs and is not to be recommended as a low-order scheme for the FEM-FCT algorithm. A much cheaper alternative is to add scalar dissipation proportional to the spectral radius of the Roe matrix [19],[20]

$$\mathrm{D}_{ij} = d_{ij} I, \tag{73}$$

where $d_{ij} = |\mathbf{a}_{ij}|(|\hat{v}_{ij}| + \hat{c}_{ij})$ is the largest (in magnitude) eigenvalue of A_{ij}. It is worth mentioning that the resulting artificial diffusion operator needs to be applied only to the five diagonal blocks of the finite element matrix. Moreover, it is the same for all components, which simplifies bookkeeping and reduces the computational cost. As long as excessive artificial viscosity is removed in the course of flux correction, a slightly better accuracy of the costly Riemann solver based on (69) does not pay off. Surprisingly enough, a slightly overdiffusive low-order method may even produce better results in the framework of FEM-FCT because of an improvement in the phase accuracy [19]. Therefore, the definition of D_{ij} in formula (73) is preferable from the computational viewpoint.

12 Generalized FEM-FCT algorithm

After the discretization in time by the standard θ-scheme, the methods of high and low order can be combined in the framework of a defect correction loop as explained above for the scalar case. The generalization of (14) to systems of equations reads

$$U^{(m+1)} = U^{(m)} + [A(U^{(m)})]^{-1} R^{(m)}, \qquad m = 0, 1, 2, \ldots \tag{74}$$

The global defect vector $R^{(m)}$ for the Galerkin discretization is given by

$$R^{(m)} = B^n + F(U^n, U^{(m)}) - A(U^{(m)})U^{(m)}. \tag{75}$$

As before, B^n denotes the constant right-hand side for the low-order scheme

$$B^n = M_L U^n + (1-\theta)\Delta t L(U^n)U^n, \qquad L = K + D. \tag{76}$$

Piecing together the contributions of raw antidiffusive fluxes (65), we obtain

$$\begin{aligned} F(U^n, U^{(m)}) &= [(M_C - M_L) - (1-\theta)\Delta t D(U^n)]U^n \\ &- [(M_C - M_L) + \theta \Delta t D(U^{(m)})]U^{(m)}. \end{aligned} \tag{77}$$

In a practical implementation, we assemble the vectors B^n and $R^{(m)}$ edge-by-edge without generating the involved global matrices. The preconditioner $A(U^{(m)})$ and the array of raw antidiffusive fluxes $\mathrm{F}_{ij}^{(m)}$ are updated in the same matrix assembly routine. The linear system to be solved at each outer iteration can be written symbolically as

$$\begin{bmatrix} A_{11}^{(m)} & A_{12}^{(m)} & A_{13}^{(m)} & A_{14}^{(m)} & A_{15}^{(m)} \\ A_{21}^{(m)} & A_{22}^{(m)} & A_{23}^{(m)} & A_{24}^{(m)} & A_{25}^{(m)} \\ A_{31}^{(m)} & A_{32}^{(m)} & A_{33}^{(m)} & A_{34}^{(m)} & A_{35}^{(m)} \\ A_{41}^{(m)} & A_{42}^{(m)} & A_{43}^{(m)} & A_{44}^{(m)} & A_{45}^{(m)} \\ A_{51}^{(m)} & A_{52}^{(m)} & A_{53}^{(m)} & A_{54}^{(m)} & A_{55}^{(m)} \end{bmatrix} \begin{bmatrix} \Delta u_1^{(m+1)} \\ \Delta u_2^{(m+1)} \\ \Delta u_3^{(m+1)} \\ \Delta u_4^{(m+1)} \\ \Delta u_5^{(m+1)} \end{bmatrix} = \begin{bmatrix} r_1^{(m)} \\ r_2^{(m)} \\ r_3^{(m)} \\ r_4^{(m)} \\ r_5^{(m)} \end{bmatrix}. \tag{78}$$

The structure and the condition number of A have a great deal of influence on the computational effort required to solve the system at hand. Working with a global matrix which is full at the block level is prohibitively expensive. Therefore, it is advisable to use a block-Jacobi preconditioner such that $A_{kl}^{(m)} = 0$ for $l \neq k$. In this case, it is sufficient to assemble and 'invert' the five diagonal blocks which we define similarly to (17)

$$A_{kk}^{(m)} = M_{kk} - \theta \Delta t L_{kk}^{(m)}, \qquad k = 1, \ldots, 5. \tag{79}$$

Here M_{kk} denotes the lumped mass matrix restricted to one variable and L_{kk} is a diagonal block of the low-order Jacobian operator. This choice of the preconditioner A essentially decouples the constituents of the compressible Euler equations and makes it possible to treat them one at a time or, better yet, in parallel. The unwieldy linear system (78) reduces to a sequence of well-behaved scalar subproblems

$$A_{kk}^{(m)} \Delta u_k^{(m)} = r_k^{(m)}, \qquad k = 1, \ldots, 5. \tag{80}$$

$$u_k^{(m+1)} = u_k^{(m)} + \Delta u_k^{(m)}, \qquad u_k^{(0)} = u_k^n. \tag{81}$$

As a result, each conservative variable can be advanced separately by solving a problem of the form (15)–(16) with limited antidiffusion built into the defect vector. Flux correction can be implemented following the FEM-FCT algorithm for scalar conservation laws.

An important remark is in order regarding the choice of correction factors for systems of equations. It turns out that an independent limiting of strongly coupled variables may produce poor results in some cases. Therefore, it is worthwhile to equalize the correction factors for each edge and apply the common value $\alpha_{ij} = f(\alpha_{ij}^1, \ldots, \alpha_{ij}^5)$ to the whole vector of antidiffusive fluxes $\mathrm{F}_{ij} \in \mathbb{R}^5$. This trick often leads to a dramatic improvement which can be attributed to the fact that the phase errors become synchronized [27]. Nevertheless, there is still a large degree of empiricism in the construction of such flux limiters, and their performance is highly problem-dependent.

Flux correction for the system of Euler equations was addressed by Löhner [27],[29]. He mentioned the following approaches to the design of a synchronized limiter:

- using correction factors associated with a single 'indicator variable',
- taking the minimum of those obtained for a certain group of variables,
- flux limiting in terms of arbitrary (nonconservative) variables.

According to [29], the combination of limiters for the density and energy is to be recommended for highly dynamic flows characterized by propagating and/or interacting shock waves. The minimum of correction factors for the density and pressure is reported to be appropriate for steady-state problems. At the same time, the minimum of those for all conservative variables is far too restrictive in the framework of a standard FCT algorithm. In particular, minor fluctuations in the crosswind velocity may result in a complete cancellation of the antidiffusive flux. Therefore, an equal treatment of all velocity components does not make sense, especially if the flow takes place in a predominant direction. It is worth mentioning that the iterative FEM-FCT limiter introduced in this paper is much less sensitive to the choice of the indicator variables and of the synchronization procedure, since the rejected antidiffusion can be reused at subsequent flux/defect correction steps. The prelimiting of antidiffusive fluxes also proves to be an important prerequisite.

To compute the correction factors for variables other than the ones being solved for, we convert the auxiliary solution $\tilde{U}_i$ and the fluxes F_{ij} into [20]

$$\tilde{U}_i' := T(\tilde{U})\tilde{U}_i, \qquad \mathrm{F}_{ij}' := T(\tilde{U})\mathrm{F}_{ij}, \tag{82}$$

where $T(\tilde{U}) \in \mathbb{R}^{5\times 5}$ is a suitable transformation matrix. Then we invoke Zalesak's limiter and apply the synchronized correction factor α_{ij} to the original flux vector F_{ij}. A general algorithm for the construction of a flux limiter operating with an arbitrary set of quantities is outlined in the book by Löhner [29]. Flux limiting in terms of characteristic variables appears to be particularly attractive. This approach has been widely used in the literature on TVD-like methods for systems of hyperbolic conservation laws [2],[5],[38],[39].

13 Implementation of boundary conditions

The specification and implementation of boundary conditions for the Euler equations has always been a bit of a mystery. A comprehensive presentation of the theoretical aspects is available in a number of popular CFD textbooks [8],[12],[42]. Boundary conditions can be classified into physical and numerical ones. Physical boundary conditions (PBC) are required to obtain a well-posed problem, whereas numerical boundary conditions (NBC) are responsible for the stability and convergence of the algorithm. It is known that the tradeoff between PBC and NBC depends on the propagation properties of the hyperbolic system. If N_v is the total number of variables and N_p is the number of incoming characteristics, then N_p physical and $N_n = N_v - N_p$ numerical boundary conditions are to be prescribed. At the inlet and outlet, N_p and N_n depend on the flow regime as follows:

	Inflow boundary						Outflow boundary					
	subsonic			supersonic			subsonic			supersonic		
	1D	2D	3D	1D	2D	3D	1D	2D	3D	1D	2D	3D
N_p	2	3	4	3	4	5	1	1	1	0	0	0
N_n	1	1	1	0	0	0	2	3	4	3	4	5

At the solid wall, the normal velocity component must be set equal to zero: $\mathbf{v}\cdot\mathbf{n} = 0$. This *no-penetration* or *free slip* boundary condition prevents the smuggling of data into or out of the computational domain by convective fluxes.

As a rule, some or all boundary conditions are specified in terms of nonconservative variables like the total enthalpy, entropy, pressure or deflection angle. Therefore, it is impossible to implement them as usual Dirichlet boundary conditions. It is common practice to recover the desired boundary values by changing to the characteristic variables, evaluating the incoming Riemann invariants from the physical boundary conditions and extrapolating the outgoing ones from the interior of the domain [12],[42],[43]. The inverse transformation yields the values of the conservative variables at the boundary which can be used to compute the numerical fluxes for a finite volume method or treated as Dirichlet boundary conditions for a finite difference method. Unfortunately, the extrapolation of Riemann invariants lacks a proper justification and is expensive to perform on unstructured meshes. In addition, we are not aware of any publication devoted to the implementation of characteristic boundary conditions for an implicit finite element method. Let us briefly describe the simple technique we developed for this purpose.

In order to *predict* the solution values at boundary nodes, we modify the five diagonal blocks $A_{kk}^{(m)}$ of our preconditioner $A(U^{(m)}) = \{a_{ij}^{kl}\}$ by picking out the corresponding rows and setting their off-diagonal entries equal to zero. In other words, if node i belongs to the boundary, then $a_{ij}^{kl} = 0,\ \forall j \neq i,\ \forall l \neq k$. This enables us to update the components of the vector $U_i = [u_{1,i}, \ldots, u_{5,i}]^T$ explicitly prior to solving the linear system (80). To this end, we simply divide the components of the nodal defect vector $R_i^{(m)} = [r_{1,i}^{(m)}, \ldots, r_{5,i}^{(m)}]^T$ by the diagonal entries of the preconditioner and increment the old iterate

$$u_{k,i}^* = u_{k,i}^{(m)} + r_{k,i}^{(m)}/a_{ii}^{kk}\ , \qquad k = 1, \ldots, 5. \tag{83}$$

Next, we transform the provisional solution U_i^* to the vector of Riemann invariants W_i. Recall that the number of physical boundary conditions is equal to the number of incoming characteristics. Therefore, we can evaluate the incoming Riemann invariants analytically and substitute the exact values for the predicted ones which generally do not satisfy the imposed PBC. The outgoing Riemann invariants, which are associated with the NBC, remain unchanged. Finally, we convert the modified vector W_i^* back to the conservative variables, assign the result to $U_i^{(m)}$ and nullify all components of the defect vector $R_i^{(m)}$. The flow chart for the algebraic manipulations to be performed for boundary nodes before the solution of scalar subproblems (80)–(81) is as follows

$$U_i^{(m)} \longrightarrow U_i^* \longrightarrow W_i \longrightarrow W_i^* \longrightarrow U_i^{(m)}, \qquad R_i^{(m)} := 0. \tag{84}$$

The last modification along with the fact that only the diagonal entry of row i in the matrices $A_{kk}^{(m)}$ is nonzero implies that $\Delta U_i^{(m)} = 0$ and $U_i^{(m+1)} = U_i^{(m)}$. In essence, the *corrected* values $U_i^{(m)}$ act as Dirichlet boundary conditions for the end-of-step solution. Note that there is no need for an *ad hoc* extrapolation of data from the interior.

In principle, all boundary conditions can be handled in this way. However, a shortcut is feasible for the free slip condition. The predicted momentum $\mathbf{w}_i = [u_{2,i}^*, u_{3,i}^*, u_{4,i}^*]^T$ can be projected onto the tangent plane without changing to the Riemann invariants:

$$\mathbf{w}_i^* := \mathbf{w}_i - (\mathbf{w}_i \cdot \mathbf{n}_i)\mathbf{n}_i, \tag{85}$$

where $\mathbf{n}_i$ denotes the outward unit normal at node i. This gives the three momentum components for the corrected $U_i^{(m)}$, whereas the density and energy are provided by U_i^*. Yet another option is to apply the above modification to the defect for the momentum equation before computing $U_i^{(m)} := U_i^*$ from equation (83).

14 Numerical examples

In the remainder of this paper, we apply our iterative FEM-FCT schemes to a number of two-dimensional test problems. The generality of the new approach makes it possible to perform flux correction in 1D and 3D using the same postprocessing routine. Three-dimensional simulation results for bubbly flows in gas-liquid reactors can be found in [21]. In the case $\theta = 0$, our algorithm is largely equivalent to the classical FEM-FCT procedure and iterative flux correction is impractical. Therefore, we restrict ourselves to implicit time-stepping in what follows. Numerical results for the flux-corrected Lax-Wendroff scheme can be found in [17],[18]. For many more examples and a numerical analysis of the discretization error for our FEM-FCT methods the reader is referred to [32].

14.1 Solid body rotation

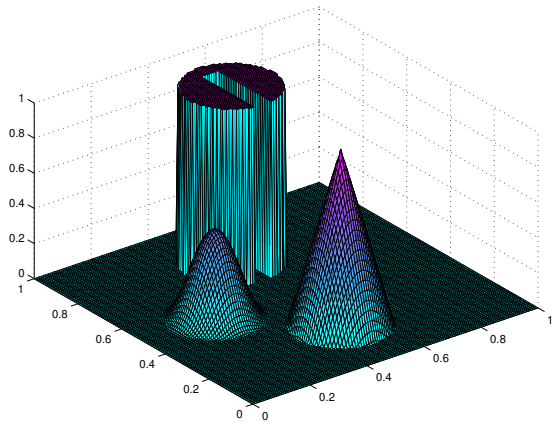
Rotation of solid bodies with discontinuities and small scale features is frequently used as a challenging test problem for transport algorithms. In the first example, we consider the benchmark configuration proposed by LeVeque [26]. It is intended to examine the ability of a numerical method to reproduce both discontinuous and smooth profiles. To this end, a slotted cylinder, a cone and a smooth hump are exposed to the nonuniform velocity field $\mathbf{v} = (0.5 - y, x - 0.5)$ and undergo a counterclockwise rotation about the center of the square domain $\Omega = (0,1) \times (0,1)$. Each of these bodies lies within a circle of radius $r_0 = 0.15$ centered at a point with Cartesian coordinates (x_0, y_0).

The exact solution to the pure convection equation after each full revolution matches the initial data depicted in Figure 3 (left). Let us introduce the normalized distance function $r(x,y) = \frac{1}{r_0}\sqrt{(x-x_0)^2 + (y-y_0)^2}$. It follows that $u(x,y,0) = 0$ for $r(x,y) > 1$. Elsewhere, the reference shape of the three bodies is given by

$$\begin{array}{lll}
\text{Cylinder:} & (x_0,y_0) = (0.5, 0.75), & u(x,y,0) = \begin{cases} 1, & \text{if } |x - x_0| \geq 0.025 \ \vee \ y \geq 0.85, \\ 0, & \text{otherwise.} \end{cases} \\
\text{Cone:} & (x_0,y_0) = (0.5, 0.25), & u(x,y,0) = 1 - r(x,y). \\
\text{Hump:} & (x_0,y_0) = (0.25, 0.5), & u(x,y,0) = 0.25[1 + \cos(\pi \min\{r(x,y), 1\})].
\end{array}$$

The numerical solution at $t = 2\pi$ produced by the iterative FEM-FCT algorithm is shown in Figure 3 (right). It was computed on a uniform mesh of 128×128 bilinear elements using the Crank-Nicolson time-stepping with $\Delta t = 10^{-3}$. Triangular finite elements yield virtually identical results [19]. No spurious wiggles come into being and the resolution of discontinuities is remarkably crisp. Even the narrow bridge of the cylinder is nicely preserved and the fill-in of the slot is insignificant. The inevitable 'peak clipping' for the cone does not exceed 10% while the hump is reproduced almost exactly. The prelimiting of antidiffusive fluxes has proved to be essential for this test. If this optional step is omitted, the sheer ridges of the cylinder are corrupted by optically disturbing kinks.

Initial data / exact solution



Iterative FEM-FCT, $\theta = 1/2$.

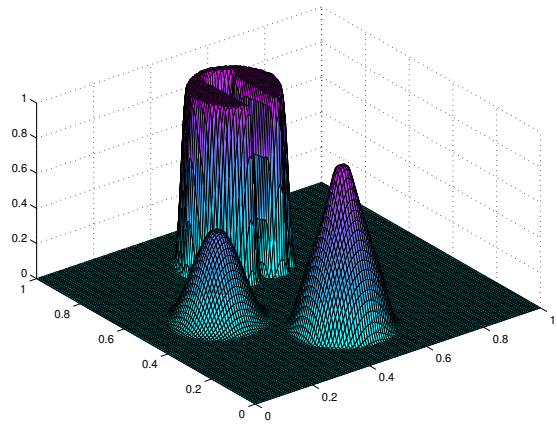


Figure 3. Solid body rotation, 128×128 bilinear elements, $t = 2\pi$.

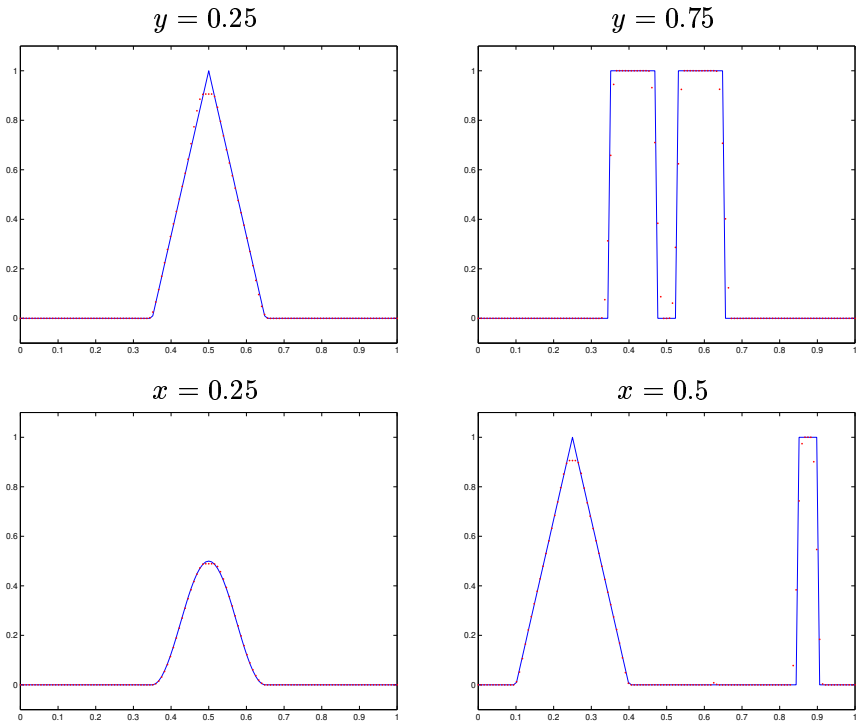


Figure 4. Cutlines for the FEM-FCT solution at $t = 2\pi$.

To examine the numerical results with extra scrutiny and facilitate comparison with those obtained by LeVeque [26] using a TVD method, four cutlines are presented in Figure 4. The solid lines designate the analytical solution while the dots represent the nodal values of the numerical one. It can be seen that our iterative FCT limiter does a really nice job. Moreover, the results are much more accurate than those produced by the TVD method equipped with the superbee limiter which introduces too much antidiffusion and tends to steepen smooth profiles [26]. In our experience, FCT is usually superior to TVD for strongly time-dependent problems which call for the use of the consistent mass matrix. At the same time, TVD schemes are to be recommended for less dynamic flows, for which mass lumping is appropriate [22]. The optimal choice of the time-stepping scheme also depends on the dynamics of the flow. In this example, the Crank-Nicolson method was selected because the fully implicit backward Euler scheme is only first order accurate and turns out to be diffusive at large time steps [17],[18],[19].

14.2 Rotation of a Gaussian hill

The second test case proposed by Lapin [23] makes it possible to evaluate the magnitude of artificial diffusion due to the discretization in space and time. This can be accomplished by applying certain statistical tools to the convection-diffusion equation

$$\frac{\partial u}{\partial t} + \mathbf{v} \cdot \nabla u = \epsilon \Delta u \quad \text{in } \Omega = (-1,1) \times (-1,1), \tag{86}$$

where $\mathbf{v} = (-y, x)$ is the velocity field and $\epsilon = 10^{-3}$ is the physical diffusion coefficient.

The initial condition to be imposed is given by $u(x,y,0) = \delta(x_0, y_0)$, where δ stands for the Dirac delta function. Clearly, it is impossible to initialize the solution by a singular function in a practical implementation. Instead, it is reasonable to concentrate the whole mass at a single node. The integral of a discrete function over the domain Ω can be computed as the sum of nodal values multiplied by the entries of the lumped mass matrix: $\int_\Omega u_h \, d\mathbf{x} = \int_\Omega \sum_i u_i \varphi_i \, d\mathbf{x} = \sum_i m_i u_i$. The total mass of a delta function equals unity. Hence, one should find node i closest to the peak location (x_0, y_0) and set $u_i^0 = 1/m_i$, $u_j^0 = 0, \ j \neq i$. Alternatively, one can start with the exact solution at a time $t_0 > 0$.

In the rotating Lagrangian reference frame, the convective term vanishes and the resulting diffusion problem can be solved analytically. It can be readily verified that the exact solution of (86) is a Gaussian hill defined by the normal distribution function

$$u(x,y,t) = \frac{1}{4\pi\epsilon t} \, e^{-\frac{r^2}{4\epsilon t}}, \qquad r^2 = (x-\hat{x})^2 + (y-\hat{y})^2,$$

where $\hat{x}$ and $\hat{y}$ denote the time-dependent peak coordinates

$$\hat{x}(t) = x_0 \cos t - y_0 \sin t, \qquad \hat{y}(t) = -x_0 \sin t + y_0 \cos t.$$

The actual peak coordinates for a numerical approximation may be quite different. They can be calculated as the mathematical expectation of the center of mass under the probability distribution with density u_h given by the finite element solution

$$\hat{x}_h(t) = \int_\Omega x u_h(x,y,t) \, d\mathbf{x}, \qquad \hat{y}_h(t) = \int_\Omega y u_h(x,y,t) \, d\mathbf{x}.$$

The quality of approximation can be assessed by considering the standard deviation

$$\sigma_h^2(t) = \int_\Omega r_h^2 u_h(x,y,t) \, d\mathbf{x}, \qquad r_h^2 = (x-\hat{x}_h)^2 + (y-\hat{y}_h)^2,$$

which quantifies the rate of smearing caused by both physical and the numerical diffusion. Due to all sorts of discretization errors, σ_h^2 may differ considerably from the exact value $\sigma^2 = 4\epsilon t$. This discrepancy represented by the relative variance error

$$\Delta\sigma_{\text{rel}} = \frac{\sigma_h^2 - \sigma^2}{\sigma^2} = \frac{\sigma_h^2}{4\epsilon t} - 1$$

serves as an excellent indicator of numerical diffusion inherent to the discretization scheme.

Let us start with the analytical solution corresponding to $x_0 = 0$, $y_0 = 0.5$ and $t_0 = 0.5\,\pi$. Figure 5 depicts the exact and numerical solution after one full revolution of the Gaussian hill. The mesh size and time step are the same as in the previous example. The iterative FEM-FCT method with Crank-Nicolson time-stepping proves its worth. Note that no peak clipping takes place. On the contrary, the global maximum reaches 10.1519 as compared to 10.1360 for the exact solution. As the Gaussian hill moves around the origin, it is being gradually smeared by diffusion. The influence of the time step Δt on the value of $\Delta\sigma_{\text{rel}}$ is illustrated in Figure 6. This diagram enables us to assess the total amount of numerical diffusion and to estimate the share of the temporal error.

Exact solution, $||u||_\infty = 10.1360$ Iterative FEM-FCT, $||u||_\infty = 10.1519$

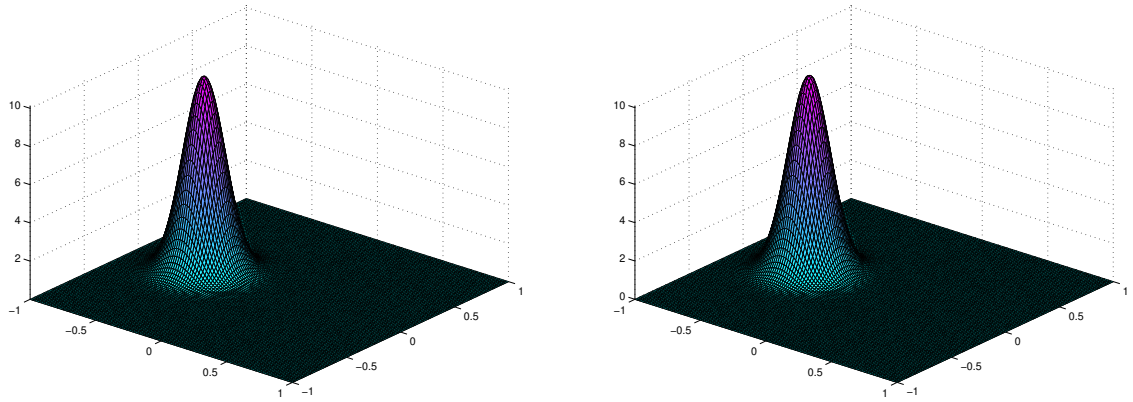


Figure 5. Rotation of a Gaussian hill, 128×128 bilinear elements, $t = 2.5\,\pi$.

If the first-order accurate backward Euler method is employed, the temporal part of the relative variance error plays an important role at large time steps and decreases linearly as the time step is refined. This is also the case for the second-order accurate Crank-Nicolson scheme but the temporal discretization error is obviously much smaller than that for the backward Euler method. As expected, the new (iterative) limiter is able to accommodate more antidiffusion than the old (non-iterative) one. In fact, $\Delta\sigma_{\text{rel}}$ may even become negative if the spatial discretization error prevails. This explains the above-mentioned enhancement of the peak in Figure 5. However, the differences between the performance of the two FEM-FCT versions are marginal in the range of time steps considered in this example. The Courant number must be 'large' for the advantages of the iterative approach to become pronounced. Hence, the potential of the new limiter can only be utilized to the full extent if the time derivative of the transported quantity is relatively small and temporal accuracy can be sacrificed in favor of the unconditional positivity offered by the fully implicit FEM-FCT method.

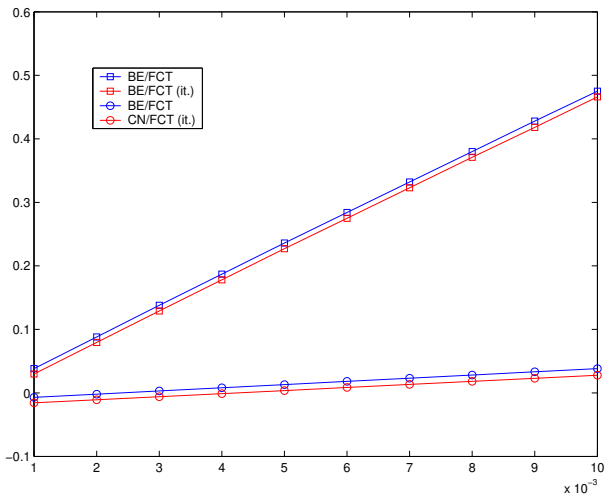


Figure 6. Gaussian hill: relative variance error vs. the time step.

14.3 Steady-state convection-diffusion

The proposed FEM-FCT algorithm can be applied to stationary problems in conjunction with a pseudo-time-stepping technique, whereby the steady-state solution is obtained by marching into the stationary limit of the associated time-dependent problem. Evolution details are immaterial in this case, since the time step is merely an artificial parameter which determines the convergence rates. Hence, it is desirable to choose time steps as large as possible, so as to reduce the computational cost. The restrictive CFL condition prevents explicit schemes from operating with large time steps and makes them too inefficient for our purposes. This drawback can be rectified to some extent by resorting to local time-stepping but it is obvious that steady-state problems call for an implicit treatment.

In light of the above, the fully implicit backward Euler method, which was found to be quite diffusive for transient problems, constitutes an excellent iterative solver for steady or creeping flows. Let us investigate the numerical behavior of the BE/FCT scheme for the singularly perturbed convection-diffusion equation

$$\mathbf{v} \cdot \nabla u - \epsilon \Delta u = 0 \qquad \text{in } \Omega = (0,1) \times (0,1),$$

where $\mathbf{v} = (\cos 10^o, \sin 10^o)$ and $\epsilon = 10^{-3}$. The concomitant boundary conditions read

$$\frac{\partial u}{\partial y}(x,1) = 0, \qquad u(x,0) = u(1,y) = 0, \qquad u(0,y) = \begin{cases} 1, & y \geq 0.5, \\ 0, & y < 0.5. \end{cases}$$

The solution to this elliptic problem is characterized by the presence of a sharp front next to the line $x = 1$. The boundary layer develops because the solution of the reduced problem ($\epsilon = 0$) does not satisfy the homogeneous Dirichlet boundary condition.

A reasonable initial approximation for the pseudo-time-stepping loop is given by

$$u(x,y,0) = \begin{cases} 1 - x, & y \geq 0.5, \\ 0, & y < 0.5. \end{cases}$$

It is worthwhile to start with the discrete upwind scheme and use the converged low-order solution as initial data for the time-dependent FEM-FCT algorithm. This 'educated guess' should be close enough to the steady-state limit. Hence, the computational overhead due to the assembly and limiting of antidiffusive fluxes will be insignificant.

The numerical solutions depicted in Figure 7 were computed on a Cartesian grid of 64×64 bilinear elements. Both of them were produced by FEM-FCT with backward Euler time-stepping. The time step $\Delta t = 0.1$ was intentionally chosen to be very large to expose the devastating effect of numerical diffusion for the non-iterative formulation. The pronounced smearing in the vicinity of the boundary layer (see the left diagram) seriously compromises the benefits of unconditional stability/positivity which were the main reason for using an implicit time discretization in the first place. Iterative flux correction performs much better, as demonstrated by the right diagram. Both the front and the boundary layer are resolved sharply and the solution is completely free of oscillations. The time step must be drastically reduced ($\Delta t \approx 10^{-3}$) for the non-iterative limiter to be competitive. An adaptive grid refinement makes it possible to achieve comparable accuracy on a much coarser mesh consisting of as few as 160 quadrilateral elements [19].

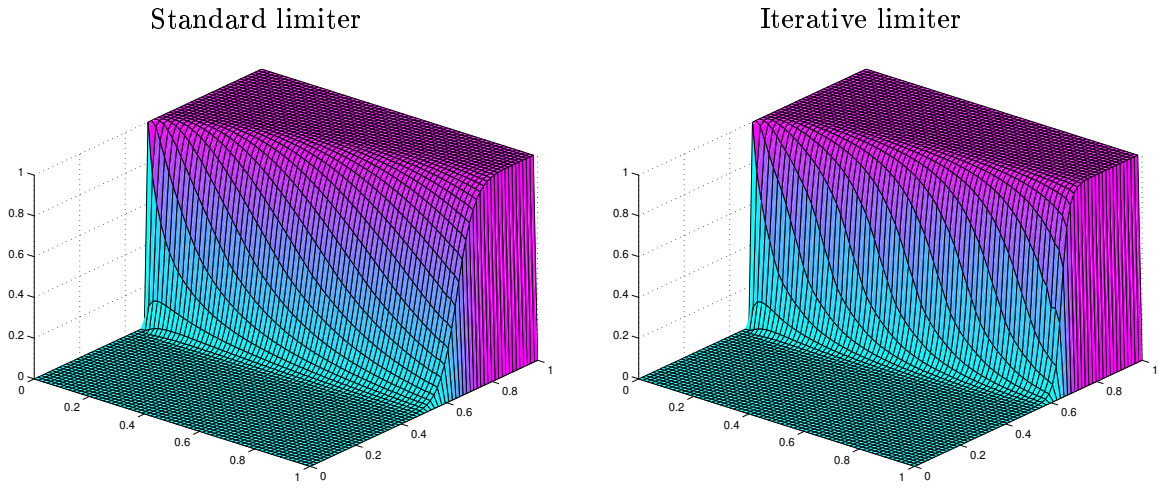


Figure 7. Steady-state convection-diffusion. BE/FCT, $\Delta t = 0.1$.

14.4 Shock tube problem

The first time-dependent example for the compressible Euler equations is the well-known shock tube problem [24],[40]. Its physical prototype is a closed tube filled with gas which is initially at rest and separated by a membrane into the regions of high and low pressure. Specifically, the initial conditions for the Riemann problem to be solved are given by

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

After the membrane is removed, the flow structure in the shock tube is characterized by three dynamically moving waves. A shock wave sets off for the region of lower pressure with velocity v_s satisfying the Rankine-Hugoniot conditions. All of the primitive variables are discontinuous across the shock. The pressure jump propels the mass in the same direction with velocity v_p. The moving interface between the regions of different densities but constant velocity and pressure represents a contact discontinuity. Finally, a rarefaction wave propagates in the opposite direction providing a smooth transition to the original values of the state variables in the region of high pressure.

To compare the performance of the low-order methods based on discrete upwinding for hyperbolic systems (69) and scalar artificial viscosity (73), we present the one-dimensional simulation results in Figure 8. The snapshots correspond to the time instant $t = 0.231$. The Euler equations were discretized using 100 linear finite elements and the Crank-Nicolson time-stepping with $\Delta t = 10^{-3}$. The analytical solution represented by the dotted line was obtained using the technique described by Anderson [1]. Both numerical solutions are completely free of oscillations and qualitatively correct but their accuracy leaves a lot to be desired. The smearing introduced by scalar dissipation is seen to be stronger than that for the finite element counterpart of Roe's approximate Riemann solver but the differences are quite small. In what follows, we will stick to the former approach because it is more economical. In addition, some (but not too much) extra diffusion turns out to be beneficial as long as it can be removed in the antidiffusive step [19].

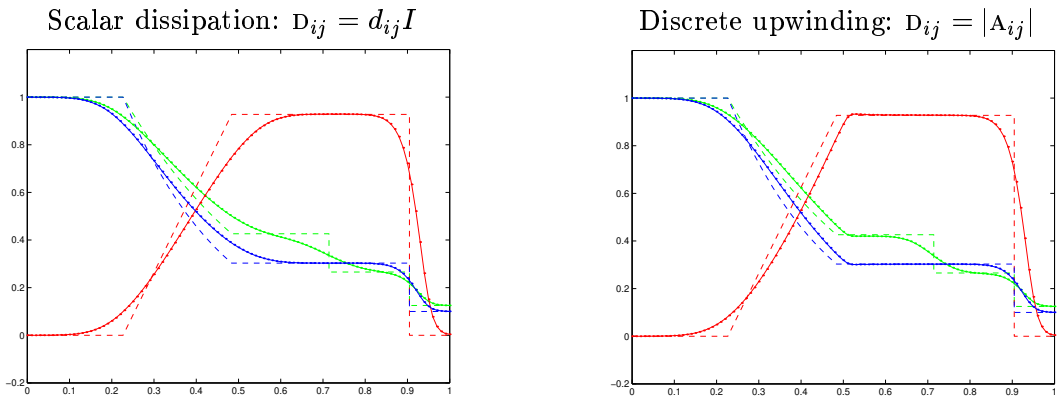


Figure 8. Shock tube problem in 1D. Low-order solutions, $t = 0.231$.

The two-dimensional results displayed in Figure 9 were computed by the iterative CN/FCT algorithm using 128×128 bilinear elements and $\Delta t = 10^{-3}$. The underlying low-order method was constructed by adding scalar dissipation proportional to the spectral radius of the Roe matrix. Following the guidelines provided by Löhner [27],[29], we performed the synchronization of Zalesak's limiter by taking the minimum of correction factors for the density and energy. The solution along the line $y = 0.5$ (see the 1D diagram in the lower right corner) demonstrates that the resolution of the shock and of the contact

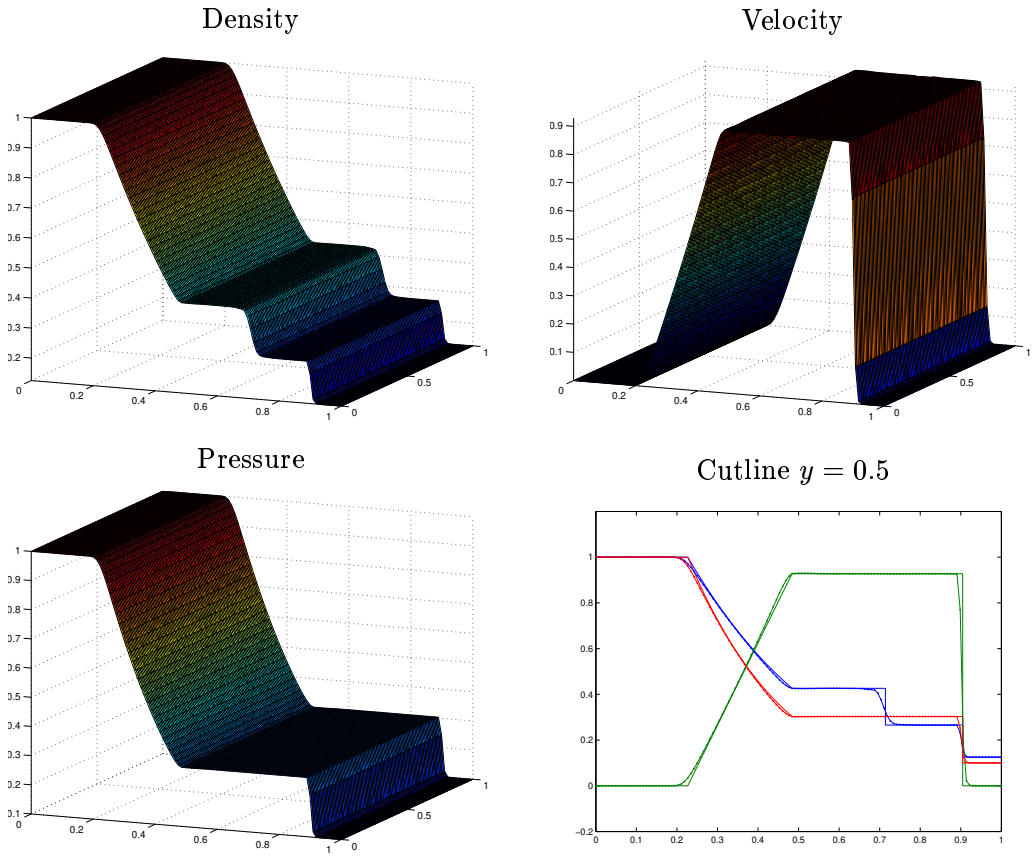


Figure 9. Shock tube problem in 2D. Iterative FEM-FCT scheme, $t = 0.231$.

discontinuity is dramatically improved as compared to the low-order solutions in Figure 8. Moreover, no spurious undershoots or overshoots are observed, even though the correction factors for the momentum were left out of consideration in the limiting process. Due to the fact that the time step must remain small for accuracy reasons in this example, the non-iterative FEM-FCT method yields very similar results. However, it becomes rather diffusive if the minimum of correction factors for all conservative variables is employed for synchronization. In this case, the iterative limiter performs much better because the rejected antidiffusion can be built into the intermediate solution step-by-step [32].

15 Radially symmetric Riemann problem

The second transient benchmark was proposed by LeVeque [25] to assess the ability of numerical methods to preserve radial symmetry. In essence, it represents a counterpart of the shock tube problem in polar coordinates. Before an impulsive start, the unit square $\Omega = (-0.5, 0.5) \times (-0.5, 0.5)$ is separated by an imaginary membrane into two subregions: $\Omega_L = \{(x, y) \in \Omega : r = \sqrt{x^2 + y^2} < 0.13\}$ and $\Omega_R = \Omega \backslash \Omega_L$. The gas is initially at rest, whereby its pressure and density are higher within the circle Ω_L than outside of it:

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

The abrupt removal of the membrane at $t = 0$ triggers a radially expanding shock wave which is induced by the pressure difference. The objective is to capture the moving shock and to make sure that the numerical solution remains radially symmetric.

The simulation results depicted in Figure 10 were obtained using the same method, mesh and time step as in the previous example. Note that the contour lines for the density at $t = 0.13$ (see the left diagram) have the form of concentric circles, which demonstrates that our FEM-FCT algorithm does preserve the symmetry. The 1D plot on the right shows the density distribution along the $x-$axis for the same solution (dotted line) and for the one computed on a much finer mesh with more than one million nodes (solid line). A good agreement is observed between the two curves. The 'exact' solution can be derived by solving a one-dimensional Riemann problem with geometric source terms [25].

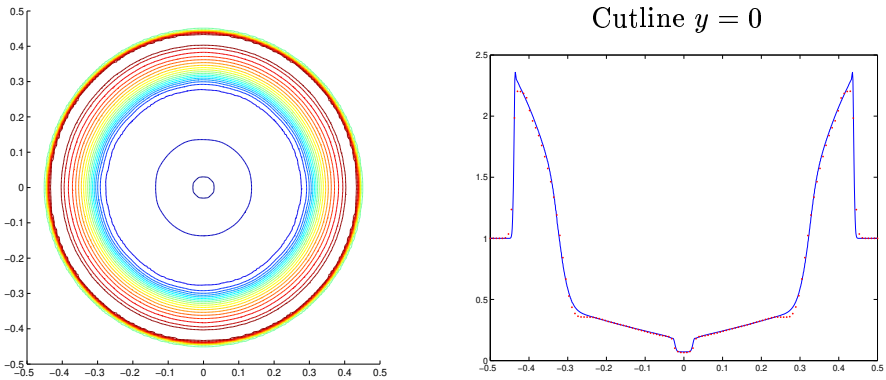


Figure 10. Radially symmetric RP. Density distribution at $t = 0.13$.

15.1 Compression corner

To illustrate the potential of the fully implicit FEM-FCT algorithm for the Euler equations, we apply it to a steady two-dimensional supersonic flow over a wedge which may imitate e.g. the tip of a projectile. This standard configuration is sometimes referred to as *compression corner* and constitutes an excellent test problem because it can be solved analytically in the framework of the *oblique shock theory* [1]. The originally uniform supersonic flow under consideration preserves its free-stream characteristics until it reaches the wedge which deflects it upward through an angle θ. Provided that θ is not too large, the change in the flow direction takes place across a shock wave which has the form of a straight line emanating from the tip of the wedge and running oblique to the original flow direction. All the streamlines experience the same deflection at the shock, so that the flow behind it is also uniform and parallel to the surface of the wedge. The Mach number decreases, whereas the pressure, density and temperature increase across the shock wave. The so-called $\theta - \beta - M$ *relation* makes it possible to express the deflection angle θ in terms of the shock inclination angle β and the upstream Mach number M_1 as follows [1]

$$\tan\theta = 2\cot\beta\frac{M_1^2\sin^2\beta - 1}{M_1^2(\gamma + \cos 2\beta) + 2}.$$

Conversely, one can determine the angle β from this formula if θ and M_1 are given.

In particular, the combination $M_1 = 2.5$, $\theta = 15^\circ$ yields a weak shock with $\beta = 36.94^\circ$. The downstream Mach number $M_2 = 1.87$ can be evaluated as explained in [1]. This test case is well documented in the *CFD Verification and Validation* database of the NPARC Alliance [33] where additional information and comparison data can be found. Our simulation results presented in Figure 11 were computed on a boundary-fitted mesh of 128×128 bilinear elements and interpolated onto a Cartesian mesh *afterwards* for visualization purposes. The same kind of postprocessing was employed in all the remaining examples. Since the problem at hand is stationary, it is worthwhile to use the backward Euler method for the discretization in time (see above). Although it is unconditionally stable for scalar equations, we had to use a moderate time step $\Delta t = 10^{-2}$ or a strong underrelaxation to secure the convergence of outer iterations. It is not surprising that our segregated algorithm for the Euler equations is insufficiently robust for steady flows which call for the use of a coupled solution technique [11],[41].

The distribution of Mach numbers predicted by the low-order method based on scalar dissipation looks more or less reasonable (see Figure 11, top). The upstream and downstream Mach numbers M_1 and M_2 are correct but the transition between them is anything else but discontinuous. The oblique shock is strongly smeared by numerical diffusion and actually resembles a 'rarefaction wave'. At the same time, this overly diffusive solution serves as a good initial guess for the FEM-FCT solver. Even the non-iterative formulation brings about a dramatic improvement in comparison with the low-order method. Figure 11 (middle) demonstrates that the shock is resolved sharply within 5-6 elements. However, the accuracy of the results deteriorates as the time step is increased, whereas the iterative FEM-FCT limiter is free of this drawback. It yields the numerical solution shown in Figure 11 (bottom), where the shock is resolved within as few as 3-4 elements. Obviously, this gain of accuracy is only recorded if the number of outer iterations is sufficiently large for the rejected antidiffusion to be effectively recycled.

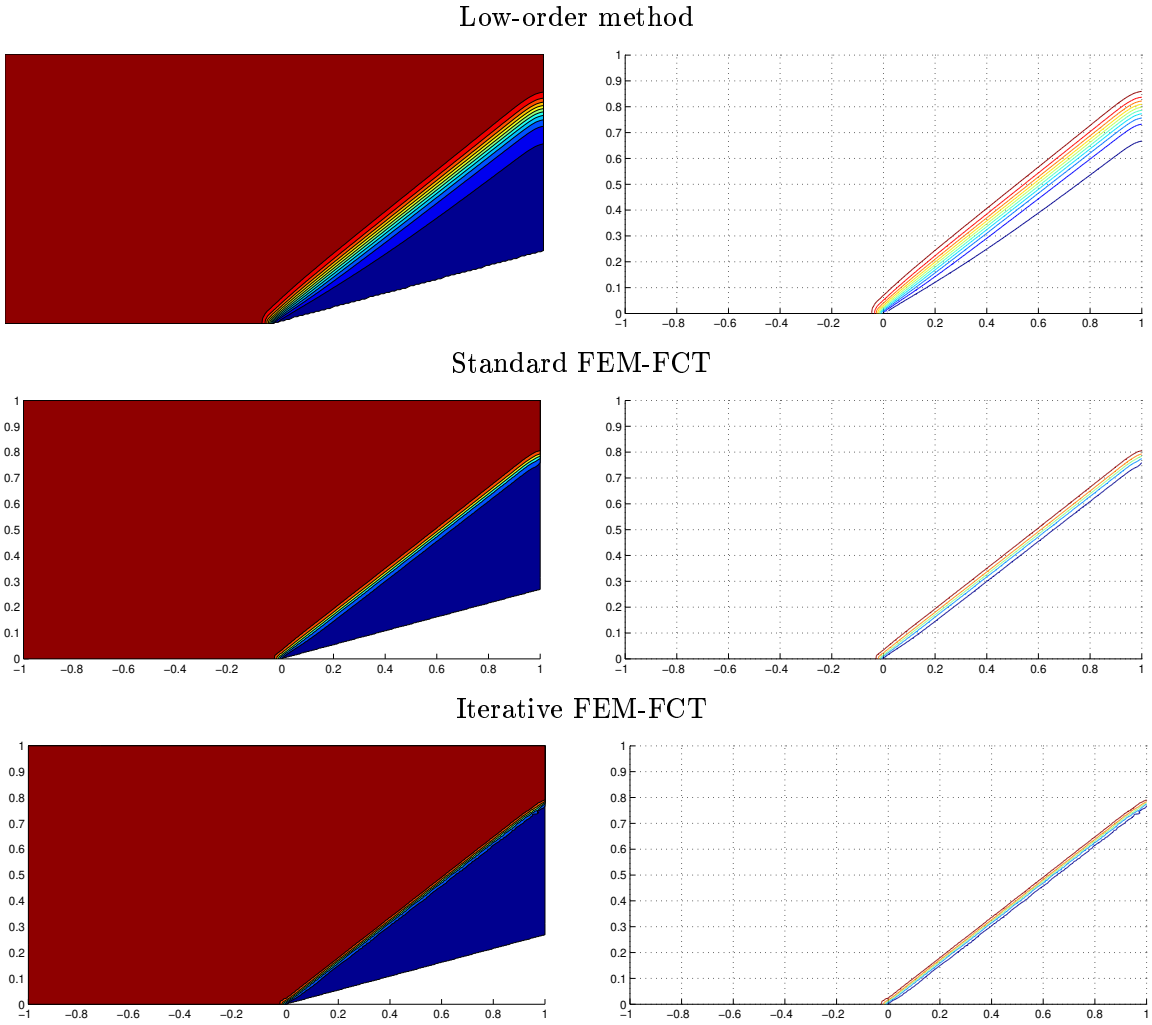


Figure 11. Compression corner. Oblique shock at $M_1 = 2.5$, $\theta = 15^\circ$.

So far we have only dealt with structured meshes, so that one of the main advantages of the discrete FEM-FCT approach still awaits numerical verification. In order to study the performance of the new algorithm on unstructured meshes, let us perform an adaptive mesh refinement by clustering the grid points in the vicinity of the stationary shock. The quadrilateral elements of the coarse mesh shown in Figure 12 are successively subdi-

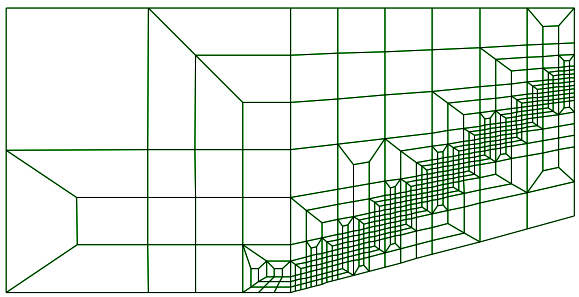


Figure 12. Coarse mesh with local refinement.

vided into four subelements which yields a computational mesh with about 10,000 vertices after two refinements. This top-down approach enables us to generate the hierarchical quad-tree data structures needed by our geometric multigrid solver for linear systems [41]. The resulting numerical solution to the compression corner problem is shown in Figure 13. It exhibits superb accuracy and remains absolutely nonoscillatory, which confirms that the iterative FEM-FCT method is applicable to unstructured meshes and non-rectangular finite elements. Furthermore, the fully implicit pseudo-time-stepping proves to be appropriate for the steady Euler equations and an adaptive time step control [41] is feasible. It is worth mentioning that our simulation results produced by FEM-FCT appear to be superior to the 'reference solution' from the NPARC database [33].

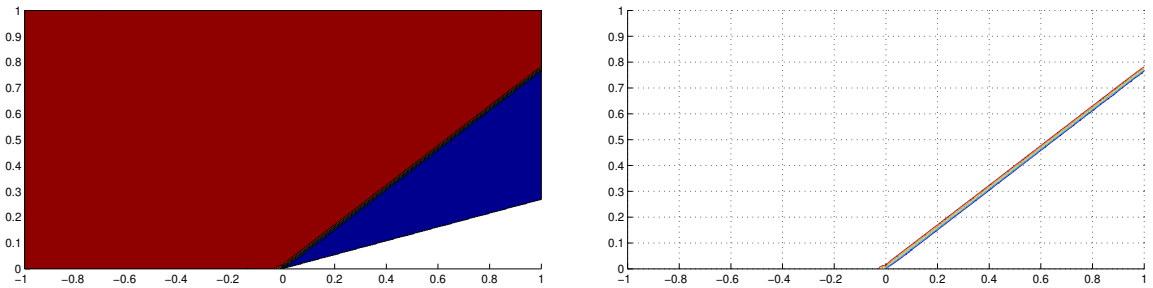


Figure 13. Adaptive FEM-FCT solution, $M_1 = 2.5$, $\theta = 15°$.

15.2 Prandtl-Meyer expansion

Our last example deals with a steady supersonic flow being deflected downward rather than upward. In this case, the flow behavior is quite different from the one discussed above for the compression corner. As the gas reaches the kink, it starts spreading and the flow characteristics change smoothly across the so-called *Prandtl-Meyer expansion* wave. The streamlines gradually bend downward and eventually become parallel to the lower wall as they leave the rarefaction fan which separates the regions of uniform flow. All flow properties adjust themselves continuously across the rarefaction wave except for the critical point at which the wall geometry changes abruptly. The Mach number increases, while the pressure, density and temperature decrease. The analytical solution to this problem and further details can be found in the book by Anderson [1].

To comply with the NPARC setup for this benchmark [33], we adopt the free-stream Mach number $M_1 = 2.5$ and the deflection angle $\theta = 15°$. The resulting expansion fan is composed from an infinite number of iso-Mach lines lying in the angular sector bounded by the lines corresponding to $\mu_1 = 23.58°$ upstream and $\mu_2 = 18.0°$ downstream. On exit from the rarefaction wave, the Mach number equals $M_2 = 3.24$ for the exact solution. The objective is to investigate the ability of the discretization scheme to reproduce smooth transitions as opposed to shocks. Let us return to the settings used to obtain the results in Figure 11. The numerical solutions produced by the same algorithms applied to the Prandtl-Meyer expansion problem are presented in Figure 14. The numerical diffusion inherent to the low-order method (top) leads to a pronounced smearing of the expansion wave, whereas both FEM-FCT solutions (middle/bottom) are seen to be very accurate. They are virtually identical, since no flux limiting is necessary in smooth regions.

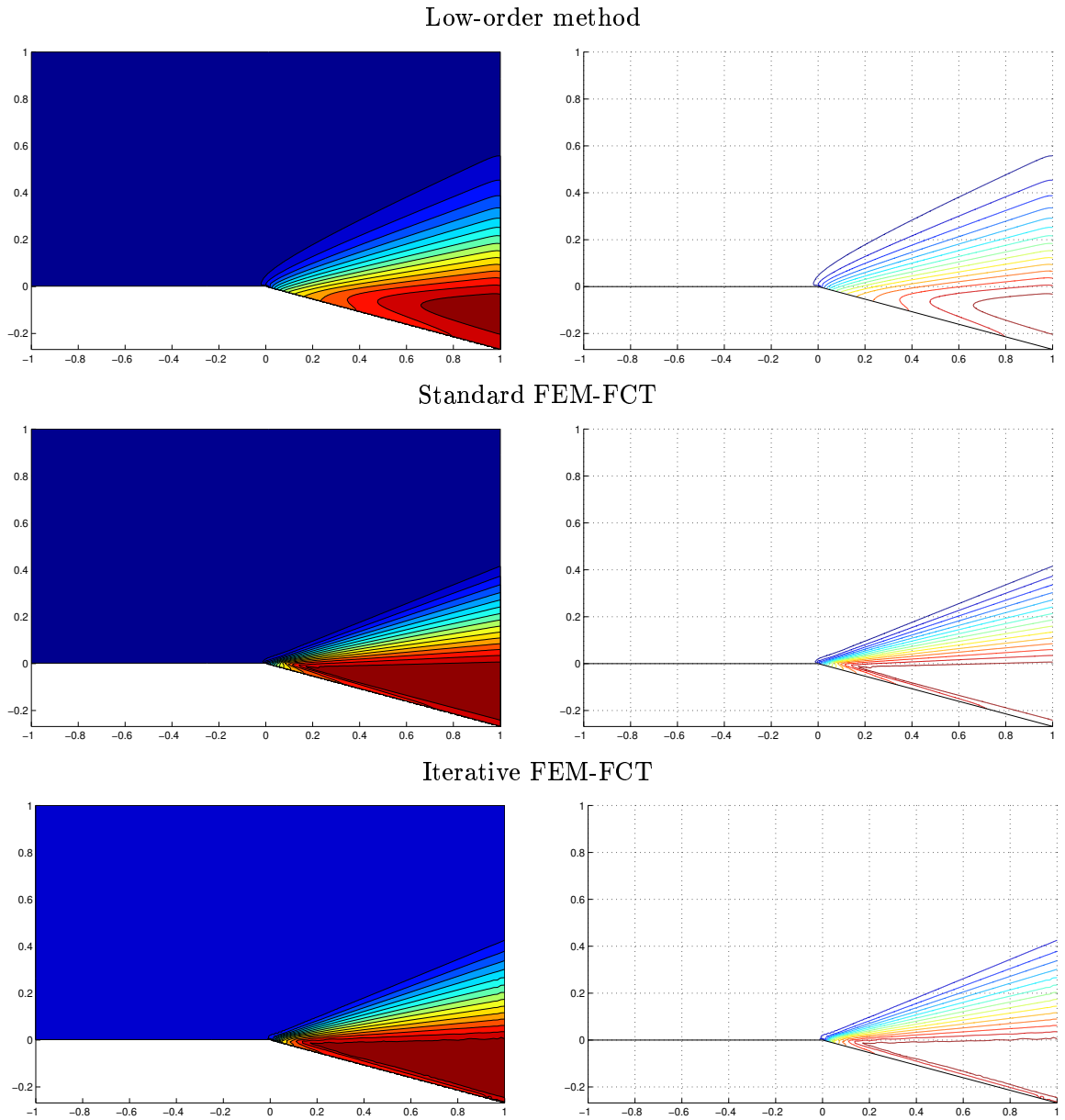


Figure 14. Prandtl-Meyer expansion at $M_1 = 2.5$, $\theta = 15^\circ$.

16 Conclusions

An iterative FEM-FCT algorithm for multidimensional conservation laws was presented. Some deficiencies of the original formulation were exposed and measures were taken to fill the gaps. Zalesak's limiter was embedded into a nonlinear defect correction loop. An extension of the scalar FCT methodology to the compressible Euler equations was carried out making use of the one-dimensional theory and flux difference splitting tools developed in the finite difference framework. The need for a proper synchronization of correction factors was emphasized. An efficient way to perform the matrix assembly was proposed and other relevant implementation aspects were discussed in detail. Implicit time discretization appears to be very attractive for handling stationary problems and/or circumventing

the stability condition. However, our segregated approach to the solution of the Euler equations is hardly optimal for steady flows. The convergence rates can be much improved by using a coupled solution technique and a FMG-FAS multigrid method [11],[30]. An analog of the *local MPSC* smoother for the incompressible Navier-Stokes equations [41] seems to be a superior alternative to the commonly employed *collective Gauß-Seidel relaxation*. The development of robust and efficient iterative solvers for implicit FEM-FCT schemes constitutes an important direction for further research.

References

[1] J.D. Anderson, Jr., *Modern Compressible Flow*, McGraw-Hill, 1990.

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

[3] J.P. Boris and D.L. Book, Flux-corrected transport. I. SHASTA, A fluid transport algorithm that works. *J. Comput. Phys.* **11** (1973) 38–69.

[4] C.R. DeVore, An improved limiter for multidimensional flux-corrected transport. *NASA Technical Report* AD-A360122 (1998).

[5] 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).

[6] J.H. Ferziger and M. Peric, *Computational Methods for Fluid Dynamics*. Springer, 1996.

[7] C.A.J. Fletcher, The group finite element formulation. *Comput. Methods Appl. Mech. Engrg.* **37** (1983) 225-243.

[8] C.A.J. Fletcher, *Computational Techniques for Fluid Dynamics*. Springer, 1988.

[9] A. Harten, High resolution schemes for hyperbolic conservation laws, *J. Comput. Phys.* **49** (1983) 357–393.

[10] A. Harten, On a class of high resolution total-variation-stable finite-difference-schemes. *SIAM J. Numer. Anal* 21 (1984) 1-23.

[11] P.W. Hemker and B. Koren, Defect correction and nonlinear multigrid for steady Euler equations. In: W.G. Habashi and M.M. Hafez (ed.). *Computational fluid dynamics techniques*. London: Gordon and Breach Publishers, 1995, 699–718.

[12] C. Hirsch, *Numerical Computation of Internal and External Flows*. John Wiley & Sons, Chichester, 1990.

[13] A. Jameson, Analysis and design of numerical schemes for gas dynamics 1. Artificial diffusion, upwind biasing, limiters and their effect on accuracy and multigrid convergence, *International Journal of Computational Fluid Dynamics* **4** (1995) 171-218.

[14] A. Jameson, Computational algorithms for aerodynamic analysis and design. *Appl. Numer. Math.* **13** (1993) 383-422.

[15] A. Jameson, Positive schemes and shock modelling for compressible flows. *Int. J. Numer. Meth. Fluids* **20** (1995) 743–776.

[16] D. Kuzmin, Positive finite element schemes based on the flux-corrected transport procedure, In: *Computational Fluid and Solid Mechanics*, Elsevier, 887-888 (2001).

[17] D. Kuzmin and S. Turek, Flux correction tools for finite elements. *J. Comput. Phys.* **175** (2002) 525-558.

[18] D. Kuzmin and S. Turek, Explicit and implicit high-resolution finite element schemes based on the Flux-Corrected-Transport algorithm. In: F. Brezzi et al. (Eds.), Proceedings of the 4th European Conference on Numerical Mathematics and Advanced Applications, Springer-Verlag Italia, 2002.

[19] D. Kuzmin, M. Möller and S. Turek, Multidimensional FEM-FCT schemes for arbitrary time-stepping. Technical report No. 215, University of Dortmund, 2002. To appear in *Int. J. Numer. Meth. Fluids*.

[20] D. Kuzmin, M. Möller and S. Turek, Implicit flux-corrected transport algorithm for finite element simulation of the compressible Euler equations. Technical report No. 221, University of Dortmund, 2002. To appear in: Proceedings of the Conference *Finite Element Methods: 50 Years of Conjugate Gradients*, University of Jyväskylä, Finland, June 11-12, 2002.

[21] D. Kuzmin and S. Turek, Finite element discretization and iterative solution techniques for multiphase flows in gas-liquid reactors. Technical report No. 222, University of Dortmund, 2002. To appear in: Proceedings of the Conference *Finite Element Methods: 50 Years of Conjugate Gradients*, University of Jyväskylä, Finland, June 11-12, 2002.

[22] D. Kuzmin and S. Turek, High-resolution FEM-TVD schemes based on a fully multidimensional flux limiter. Submitted to *J. Comput. Phys.*

[23] A. Lapin, University of Stuttgart. Private communication.

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

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

[26] R. J. LeVeque, High-resolution conservative algorithms for advection in incompressible flow. *Siam J. Numer. Anal.* **33** (1996) 627–665.

[27] 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.

[28] R. Löhner, K. Morgan, M. Vahdati, J. P. Boris and D. L. Book, FEM-FCT: combining unstructured grids with high resolution. *Commun. Appl. Numer. Methods* **4** (1988) 717–729.

[29] R. Löhner, *Applied CFD Techniques*. Wiley, 2001.

[30] J.F. Lynn, *Multigrid Solution of the Euler Equations with Local Preconditioning*. PhD thesis, University of Michigan, 1995.

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

[32] M. Möller, *Hochauflösende FEM-FCT-Verfahren zur Diskretisierung von konvektionsdominanten Transportproblemen mit Anwendung auf die kompressiblen Eulergleichungen*. Diploma thesis, University of Dortmund, 2003.

[33] NPARC Alliance, Computational Fluid Dynamics (CFD) Verification and Validation Web Site: *http://www.grc.nasa.gov/WWW/wind/valid/*

[34] S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*. McGraw-Hill, 1980.

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

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

[37] C. Schär and P. K. Smolarkiewicz, A synchronous and iterative flux-correction formalism for coupled transport equations. *J. Comput. Phys.* **128** (1996) 101–120.

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

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

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

[41] S. Turek, *Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approach*, LNCSE 6, Springer, 1999.

[42] P. Wesseling, *Principles of Computational Fluid Dynamics*. Springer, 2001.

[43] H. C. Yee, Numerical approximations of boundary conditions with applications to inviscid gas dynamics. *NASA report* TM-81265, 1981.

[44] S. T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids. *J. Comput. Phys.* **31** (1979) 335–362.