Algebraic Flux Correction I.
Scalar Conservation Laws

Dmitri Kuzmin[1] and Matthias Möller[2]

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

Summary. An algebraic approach to the design of multidimensional high-resolution schemes is introduced and elucidated in the finite element context. A centered space discretization of unstable convective terms is rendered local extremum diminishing by a conservative elimination of negative off-diagonal coefficients from the discrete transport operator. This modification leads to an upwind-biased low-order scheme which is nonoscillatory but overly diffusive. In order to reduce the incurred error, a limited amount of compensating antidiffusion is added in regions where the solution is sufficiently smooth. Two closely related flux correction strategies are presented. The first one is based on a multidimensional generalization of total variation diminishing (TVD) schemes, whereas the second one represents an extension of the FEM-FCT paradigm to implicit time-stepping. Nonlinear algebraic systems are solved by an iterative defect correction scheme preconditioned by the low-order evolution operator which enjoys the M-matrix property. The diffusive and antidiffusive terms are represented as a sum of antisymmetric internodal fluxes which are constructed edge-by-edge and inserted into the global defect vector. The new methodology is applied to scalar transport equations discretized in space by the Galerkin method. Its performance is illustrated by numerical examples for 2D benchmark problems.

1 Introduction

Over the past three decades that elapsed since the birth of the FCT algorithm, numerous clones and alternative high-resolution schemes based on flux/slope limiters have been proposed in the literature. Nevertheless, the developement of reliable discretization techniques for convection-dominated flows remains one of the main challenges in Computational Fluid Dynamics. As a matter of fact, a serious disadvantage of many existing numerical schemes is the lack of generality. Most of them are only suitable for structured or simplex meshes, specific space discretization (finite differences/volumes, discontinuous or linear finite element approximations) and/or explicit time-stepping.

In particular, the use of high-resolution finite element schemes is still rather uncommon in spite of their enormous potential demonstrated by Löhner and his collaborators [34],[36],[37]. The failure of the FEM to be as successful in CFD as in structural mechanics is largely due to the fact that many limiting techniques are essentially one-dimensional and do not carry over to unstructured meshes. In a series of recent publications, we derived a new family of implicit FEM-FCT schemes using a mass-conserving modification of matrices that result from the standard Galerkin discretization [22],[23],[25],[26]. In fact, this approach to the design of nonoscillatory positivity-preserving schemes can be characterized as *Algebraic Flux Correction* which is applicable to discrete transport operators of any origin. All the necessary information is provided by the magnitude, sign and position of nonzero matrix coefficients.

Furthermore, the AFC methodology forms the basis for a fully multidimensional generalization of Harten's *total variation diminishing* schemes [28]. A node-oriented flux limiter of TVD type is constructed so as to control the ratio of upstream and downstream edge contributions which are associated with the positive and negative off-diagonal coefficients of the high-order transport operator, respectively [28]. This limiting strategy resembles Zalesak's FCT algorithm [54] but its derivation is based on different premises. In addition, the proposed FEM-TVD schemes are upwind-biased, i.e., the raw antidiffusive fluxes are multiplied by the correction factor for the upwind node rather than by the minimum of those for both nodes.

In contrast to flux limiters of TVD type, flux-corrected transport methods operate at the fully discrete level. As a result, the correction factors estimated by the limiter depend on the time step, so that it is impossible to reap the benefits of the fully implicit time-stepping without sacrificing some accuracy. In order to prevent an implicit FEM-FCT discretization from becoming increasingly diffusive at large time steps, flux correction can be carried out in an iterative fashion so as to 'recycle' the rejected antidiffusion step-by-step [29]. In this chapter, we derive and compare algebraic TVD and FCT schemes for scalar convection-diffusion problems. Their performance is evaluated numerically. An extension of both AFC techniques to the Euler and Navier-Stokes equations of fluid dynamics is presented in the next two chapters.

2 Finite Element Discretization

Let us introduce the principles of algebraic flux correction in the finite element framework and keep in mind that they are also applicable to finite volume and finite difference discretizations. Consider the time-dependent continuity equation which represents a mass conservation law for a scalar quantity u

$$\frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{v}u) = 0 \qquad \text{in } \Omega, \tag{1}$$

where $\mathbf{v} = \mathbf{v}(\mathbf{x}, t)$ is a nonuniform velocity field, which is assumed to be known analytically or computed numerically from a momentum equation solved in a parallel way. The initial data are given by $u(\mathbf{x},0) = u_0(\mathbf{x})$, and boundary conditions are to be prescribed only at the inlet $\Gamma_{\text{in}} = \{\mathbf{x} \in \Gamma : \mathbf{v}\cdot\mathbf{n} < 0\}$, where $\mathbf{n}$ denotes the unit outward normal to the boundary Γ.

The weak form of equation (1) is derived by integrating the weighted residual over the domain Ω and setting the result equal to zero

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

If the boundary conditions are specified in terms of fluxes rather than actual values of u at the inlet, it is worthwhile to integrate the convective term by parts and substitute the incoming fluxes into the resulting surface integral.

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

$$u_h = \sum_j u_j \varphi_j, \qquad (\mathbf{v}u)_h = \sum_j (\mathbf{v}_j u_j)\,\varphi_j, \tag{3}$$

where φ_j denote the basis functions spanning the finite-dimensional subspace. This kind of approximation was promoted by Fletcher [13] who called it the *group finite element formulation*. It was found 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 [13]. In fact, it is even possible to use mixed interpolations. For instance, a bilinear approximation of u could be combined with its nonconforming counterpart [46] for the convective fluxes which call for the use of edge-oriented degrees of freedom.

The substitution of (3) into (2) yields the following semi-discrete problem

$$\sum_j \left[\int_\Omega \varphi_i\varphi_j\,\mathrm{d}\mathbf{x}\right]\frac{\mathrm{d}u_j}{\mathrm{d}t} + \sum_j \left[\int_\Omega \varphi_i \mathbf{v}_j\cdot\nabla\varphi_j\,\mathrm{d}\mathbf{x}\right] u_j = 0. \tag{4}$$

This gives a system of ordinary differential equations for the nodal values of the approximate solution which can be written compactly in matrix form

$$M_C \frac{\mathrm{d}u}{\mathrm{d}t} = Ku, \tag{5}$$

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

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

For fixed meshes, the coefficients m_{ij} and $\mathbf{c}_{ij}$ remain unchanged throughout the simulation and, consequently, need to be evaluated just once, during the initialization step. 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. The auxiliary coefficients $\mathbf{c}_{ij}$ correspond to the discretized space derivatives and have zero row sums, i. e., $\sum_j \mathbf{c}_{ij} = 0$ as long as the sum of the basis functions φ_j is equal to one at every point.

3 Conservative Flux Decomposition

It is common knowledge that the Galerkin FEM is globally conservative [15]. Indeed, summing equations (4) over i and taking into account that the basis functions sum to unity, one recovers the integral form of the conservation law. Therefore, the total mass of u in Ω may only change due to the boundary fluxes. At the same time, the finite element discretization of convective terms does not admit a natural decomposition into a sum of numerical fluxes from one node into another. Since most high-resolution schemes operate with such fluxes, their extension to finite elements proved to be a difficult task. Peraire et al. [45] demonstrated that a conservative flux decomposition is feasible for P_1 finite elements (piecewise-linear approximation on a simplex mesh). A similar technique was proposed by Barth [3],[4] who investigated the relationship between finite element and finite volume discretizations. The transition to an edge-based data structure reportedly offers a number of significant advantages as compared to the conventional element-based formulation. Moreover, it paves the way for a straightforward extension of many popular high-resolution schemes (including TVD) to unstructured meshes [34],[38],[42].

In [25] we developed a flux decomposition technique which is applicable to general finite element approximations on arbitrary meshes including quadrilateral and hexahedral ones. Integration by parts in the weak formulation (2) and the fact that the coefficients $\mathbf{c}_{ij}$ have zero row sums make it possible to decompose the contribution of convective terms to interior nodes into a sum of antisymmetric internodal fluxes (see appendix to this chapter)

$$(Ku)_i = -\sum_{j\neq i} g_{ij}, \qquad \text{where} \quad g_{ij} = (\mathbf{v}_i \cdot \mathbf{c}_{ij})u_i - (\mathbf{v}_j \cdot \mathbf{c}_{ji})u_j. \tag{7}$$

A promising approach to the derivation of nonoscillatory finite element methods consists in replacing the centered Galerkin flux g_{ij} by another consistent numerical flux [38]. On the other hand, it is often desirable to use an already existing finite element code based on conventional data structures. Therefore, we adopt a different strategy in the present chapter.

Due to the fact that the Galerkin method is conservative, it suffices to guarantee that all subsequent matrix manipulations to be performed at the discrete level do not violate this property. To this end, we introduce the concept of *discrete diffusion operators* [23]. They are defined as symmetric matrices

$$D = \{d_{ij}\} \qquad \text{such that} \quad d_{ij} = d_{ji} \tag{8}$$

which have zero row and column sums

$$\sum_i d_{ij} = \sum_j d_{ij} = 0. \tag{9}$$

We remark that the matrix D is typically sparse and its nonzero off-diagonal entries d_{ij} may be positive (diffusion) or negative (antidiffusion).

In the finite element framework, some well-known representatives of this important class of matrices are as follows:

- The *discrete Laplacian operator* which results from the discretization of second derivatives after integration by parts in the weak formulation
$$d_{ij} = \int_{\Omega} \nabla\varphi_i \cdot \nabla\varphi_j \, \mathrm{d}\mathbf{x}.$$
- The *streamline diffusion operator* which provides a stabilization of convective terms by artificial diffusion in the streamline direction
$$d_{ij} = \int_{\Omega} \mathbf{v}\cdot\nabla\varphi_i \; \mathbf{v}\cdot\nabla\varphi_j \, \mathrm{d}\mathbf{x}.$$
- The *mass diffusion operator* which is given by the difference between the consistent mass matrix and its lumped counterpart (see below)
$$d_{ij} = \int_{\Omega} \varphi_i(\varphi_j - \delta_{ij}) \, \mathrm{d}\mathbf{x}.$$

  Here δ_{ij} is the Kronecker delta which equals 1 if $i = j$ and 0 otherwise.

A discrete diffusion operator D applied to the vector of nodal values u yields

$$(Du)_i = \sum_j d_{ij} u_j = \sum_{j\neq i} d_{ij}(u_j - u_i) \tag{10}$$

due to the zero row sum property. Hence, the contribution of diffusive terms to node i can be decomposed into a sum of numerical fluxes:

$$(Du)_i = \sum_{j\neq i} f_{ij}, \qquad \text{where} \quad f_{ij} = d_{ij}(u_j - u_i). \tag{11}$$

The flux f_{ij} from node j into node i is proportional to the difference between the nodal values, so it leads to a steepening or flattening of solution profiles depending on the sign of the coefficient d_{ij}. Furthermore, the symmetry of the matrix D implies that $f_{ji} = -f_{ij}$ so that there is no net loss or gain of mass. The amount received by node i is subtracted from node j and vice versa.

The antisymmetric diffusive fluxes can be associated with edges of the graph which represents the sparsity pattern of the global stiffness matrix. For linear finite elements, their number equals the number of actual mesh edges, whereas multilinear and high-order FEM approximations allow for interactions of all nodes sharing the same element. As we are about to see, artificial diffusion operators satisfying the above conditions constitute a very useful tool for the design of multidimensional high-resolution schemes.

4 Design Criteria

First of all, we introduce the algebraic constraints which should be imposed on the discrete operators to prevent the formation of spurious undershoots and overshoots in the vicinity of steep gradients. Assume that the semi-discretized transport equation can be cast in the generic form

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_j \sigma_{ij} u_j, \qquad \text{where} \quad \sigma_{ii} = -\sum_{j\neq i} \sigma_{ij}. \tag{12}$$

In particular, this is feasible for our nodal ODE system (5) if M_C is replaced by its diagonal counterpart M_L resulting from the row-sum mass lumping

$$\sigma_{ij} = \frac{k_{ij}}{m_i}, \qquad \text{where} \qquad m_i = \sum_j m_{ij}, \quad M_L = \mathrm{diag}\{m_i\}$$

and the velocity field $\mathbf{v}$ is discretely divergence-free in the sense that

$$(\nabla \cdot \mathbf{v})_i = \frac{1}{m_i} \sum_j \mathbf{v}_j \cdot \mathbf{c}_{ij} = -\sum_j \sigma_{ij} = 0. \tag{13}$$

This approximation corresponds to a recovery of continuous nodal gradients by means of a lumped-mass L_2-projection. For 'compressible' flows, the sum of the coefficients σ_{ij} is nonvanishing. However, this makes no difference for the algebraic flux correction algorithms to be derived below.

Algebraic Constraint I (semi-discrete level)

If the coefficients of the numerical scheme do have zero row sums, then the right-hand side of (12) can be represented in terms of the off-diagonal ones

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_{j\neq i} \sigma_{ij}(u_j - u_i). \tag{14}$$

It was shown by Jameson [19], [20], [21] that negative coefficients in the above expression are the 'villains' responsible for the birth and growth of nonphysical oscillations. Indeed, if $\sigma_{ij} \geq 0, \ \forall j \neq i$ then the spatial discretization proves stable in the L_∞-norm due to the fact that

- maxima do not increase: $u_i = \max_j u_j \quad \Rightarrow \quad u_j - u_i \leq 0 \quad \Rightarrow \quad \frac{\mathrm{d}u_i}{\mathrm{d}t} \leq 0,$
- minima do not decrease: $u_i = \min_j u_j \quad \Rightarrow \quad u_j - u_i \geq 0 \quad \Rightarrow \quad \frac{\mathrm{d}u_i}{\mathrm{d}t} \geq 0.$

As a rule, the coefficient matrices are sparse, so that $\sigma_{ij} = 0$ 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. Therefore, semi-discrete schemes of this type are *local extremum diminishing* (LED). For three-point finite difference methods, the LED constraint reduces to Harten's TVD conditions [16]. If the homogeneous Dirichlet boundary conditions are prescribed at both endpoints, the total variation of the (piecewise-linear) approximate solution can be expressed as follows [19]

$$TV(u_h) := \sum_i |u_{i+1} - u_i| = 2\left(\sum \max u - \sum \min u\right) \tag{15}$$

and is obviously nonincreasing as long as the local maxima and minima do not grow. Therefore, one-dimensional LED schemes are necessarily total variation diminishing. At the same time, the positivity of matrix coefficients is easy to verify for arbitrary discretizations on unstructured meshes so that Jameson's LED criterion provides a very handy generalization of the TVD concepts.

Algebraic Constraint II (fully discrete level)

After the time discretization, an additional condition may need to be imposed in order to make sure that the solution values remain nonnegative if this should be the case for physical reasons. In general, a fully discrete scheme is *positivity-preserving* if it can be represented in the form

$$Au^{n+1} = Bu^n, \tag{16}$$

where $B = \{b_{ij}\}$ has no negative entries and $A = \{a_{ij}\}$ is a so-called *M-matrix* defined as a nonsingular discrete operator such that $a_{ij} \leq 0$ for $j \neq i$ and all the coefficients of its inverse are nonnegative. These properties imply that the positivity of the old solution u^n carries over to $u^{n+1} = A^{-1}Bu^n$. Here and below the superscript n denotes the time level.

As a useful byproduct, our algebraic positivity criterion yields a readily computable upper bound for admissible values of the time step $\Delta t = t^{n+1} - t^n$. In particular, the local extremum diminishing ODE system (14) discretized in time by the standard $\theta-$ scheme reads

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \theta \sum_{j \neq i} \sigma_{ij}^{n+1}(u_j^{n+1} - u_i^{n+1}) + (1-\theta) \sum_{j \neq i} \sigma_{ij}^n (u_j^n - u_i^n). \tag{17}$$

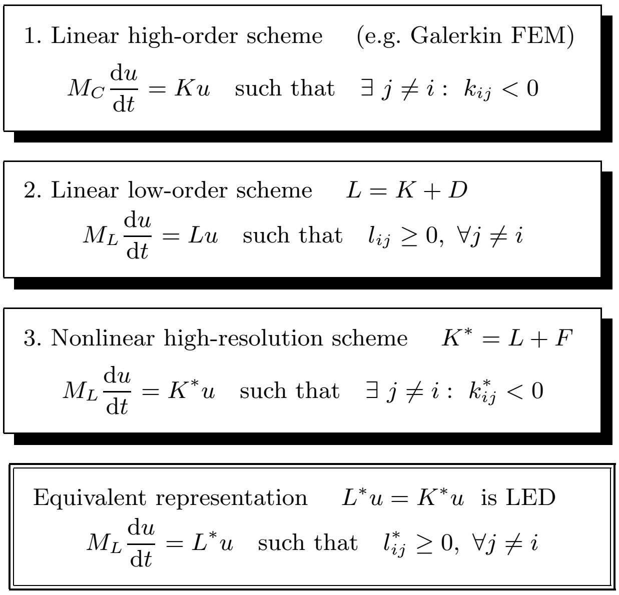
It is unconditionally positivity-preserving for $\theta = 1$ (backward Euler method) and subject to the following CFL-like condition otherwise [23],[25]

$$1 + \Delta t(1-\theta) \min_i \sigma_{ii}^n \geq 0 \qquad \text{for} \quad 0 \leq \theta < 1. \tag{18}$$

Note that this estimate is based solely on the magnitude of the diagonal coefficients σ_{ii}^n, which makes it a handy tool for adaptive time step control.

5 Algebraic Flux Correction of TVD Type

The basic idea underlying algebraic flux correction is rather simple and can be traced back to the concepts of flux-corrected transport. Roughly speaking, the governing equation is discretized in space by an arbitrary linear high-order method (e.g. central differences or Galerkin FEM) and the resulting matrices are modified *a posteriori* so as to enforce the constraints I and II imposed above. The flow chart of required algebraic manipulations is sketched in Fig. 1. The time step Δt should be chosen so as to satisfy condition (18).



1. Linear high-order scheme (e.g. Galerkin FEM)

$$M_C \frac{\mathrm{d}u}{\mathrm{d}t} = Ku \quad \text{such that} \quad \exists\, j \neq i: \ k_{ij} < 0$$

2. Linear low-order scheme $L = K + D$

$$M_L \frac{\mathrm{d}u}{\mathrm{d}t} = Lu \quad \text{such that} \quad l_{ij} \geq 0, \ \forall j \neq i$$

3. Nonlinear high-resolution scheme $K^* = L + F$

$$M_L \frac{\mathrm{d}u}{\mathrm{d}t} = K^*u \quad \text{such that} \quad \exists\, j \neq i: \ k^*_{ij} < 0$$

Equivalent representation $L^*u = K^*u$ is LED

$$M_L \frac{\mathrm{d}u}{\mathrm{d}t} = L^*u \quad \text{such that} \quad l^*_{ij} \geq 0, \ \forall j \neq i$$

Fig. 1. Roadmap of AFC-TVD manipulations.

First, we perform mass lumping and transform the high-order operator K into its nonoscillatory low-order counterpart L by adding a discrete diffusion operator D designed so as to get rid of all negative off-diagonal coefficients. In the next step, excessive artificial diffusion is removed. This is accomplished by applying a limited amount of compensating antidiffusion F which depends on the local solution behavior and improves the accuracy in smooth regions. Both diffusive and antidiffusive terms admit a conservative flux decomposition so that the proposed modifications do not affect the total mass.

It is worth mentioning that the final operator K^* does have some negative off-diagonal coefficients. Nevertheless, the resulting discretization proves local extremum diminishing if (for a given solution vector u) there exists a matrix L^* such that all off-diagonal entries l^*_{ij} are nonnegative and $L^*u = K^*u$. In the remainder of this section we will dwell on the design of discrete diffusion/antidiffusion operators and introduce flux limiters of TVD type which guarantee the existence of L^* without constructing it explicitly. In much the same way, we will derive a family of implicit FCT schemes using criterion (16) to render the underlying high-order method positivity-preserving.

5.1 Discrete Upwinding

For finite difference and finite volume discretizations, the first-order accurate upwind method yields an operator L which corresponds to the least diffusive linear LED scheme. Up to now, it has been largely unclear how to construct such an optimal low-order discretization in the finite element framework. Streamline-diffusion methods like SUPG are stable but not monotonicity-preserving, whereas other upwind-biased finite element schemes resort to a finite volume approximation of convective terms [1],[2],[52]. At the same time, the LED constraint can be enforced by elimination of negative off-diagonal coefficients from the discrete transport operator. Interestingly enough, this algebraic approach to the design of 'monotone' low-order methods reduces to standard upwinding for the one-dimensional convection equation [22],[23].

As a starting point, we consider a linear high-order discretization, e. g. our semi-discrete problem (5) for the Galerkin method. After mass lumping, each nodal value u_i satisfies an ordinary differential equation of the form

$$m_i \frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_{j \neq i} k_{ij}(u_j - u_i) + \delta_i u_i, \qquad \text{where} \quad \delta_i = \sum_j k_{ij}. \tag{19}$$

The first term in the right-hand side is associated with the 'incompressible' part of the discrete transport operator K since $\delta_i u_i$ is an approximation of $-u\nabla \cdot \mathbf{v}$ (see above) which vanishes for divergence-free velocity fields and is responsible for a physical growth of local extrema otherwise. For the concomitant low-order scheme to be local extremum diminishing, all off-diagonal coefficients of the linear operator $L = K + D$ must be nonnegative. Hence, the optimal diffusion coefficients are given by

$$d_{ii} = -\sum_{j \neq i} d_{ij}, \qquad d_{ij} = \max\{0, -k_{ij}, -k_{ji}\} = d_{ji}. \tag{20}$$

By construction, $D = \{d_{ij}\}$ is a discrete diffusion operator. It follows that the difference between the resulting scheme and the original one can be represented as a sum of antisymmetric diffusive fluxes $f_{ij}^d = d_{ij}(u_j - u_i)$ between adjacent nodes whose basis functions have overlapping supports. Recall that this is sufficient to guarantee mass conservation at the algebraic level. The above manipulations lead to the desired semi-discrete scheme of low order

$$M_L \frac{\mathrm{d}u}{\mathrm{d}t} = Lu \quad \text{such that} \quad l_{ij} \geq 0, \ \forall j \neq i. \tag{21}$$

In practice, the elimination of negative off-diagonal entries is performed edge-by-edge without assembling the global matrix D. After the initialization $L := K$, we examine each pair of nonzero off-diagonal coefficients l_{ij} and l_{ji}. If the smaller one is negative, it is set equal to zero and three other entries are modified so as to restore row/column sums:

$$\begin{aligned} l_{ii} &:= l_{ii} - d_{ij}, \quad & l_{ij} &:= l_{ij} + d_{ij}, \\ l_{ji} &:= l_{ji} + d_{ij}, \quad & l_{jj} &:= l_{jj} - d_{ij}. \end{aligned} \tag{22}$$

Without loss of generality, we orient the edges of the sparsity graph so that $l_{ji} \geq l_{ij} = \max\{0, k_{ij}\}$ for the edge $\overrightarrow{ij}$. This orientation convention implies that node i is located 'upwind' and corresponds to the row number of the eliminated negative entry (if any). Furthermore, the nodes can be renumbered so as to transform L into an upper or lower triangular matrix and to design very efficient solvers/smoothers/preconditioners for the resulting linear system.

The 'postprocessing' technique described in this section will be referred to as *discrete upwinding.* Note that the LED constraint is imposed only on the incompressible part of the transport operator. The 'reactive' term $\delta_i u_i$ is not affected by artificial diffusion since $\sum_j l_{ij} = \sum_j (k_{ij} + d_{ij}) = \sum_j k_{ij}$ due to the zero row sum property of D. If the governing equation contains sources and sinks, they may need to be linearized as proposed by Patankar [44] and explained in [25],[26]. Furthermore, physical diffusion can be built into the matrix either before or after discrete upwinding. In the former case, it is automatically detected and the amount of artificial diffusion is reduced accordingly. In our experience, the TVD flux limiters to be presented below should be applied to the convective operator alone. Therefore, it is advisable to incorporate the contribution of physical diffusion into L rather than K.

Example. To elucidate the ins and outs of discrete upwinding in a rather simple setting, consider the one-dimensional counterpart of equation (1)

$$\frac{\partial u}{\partial t} + v \frac{\partial u}{\partial x} = 0, \tag{23}$$

where the velocity v is assumed to be constant and positive. The computational domain $\Omega = (a, b)$ is defined by its two endpoints, and an essential boundary condition of the Dirichlet type is prescribed at the inlet $x = a$.

This hyperbolic equation is discretized in space by the lumped-mass Galerkin method using a piecewise-linear approximation on a uniform mesh of size Δx. The corresponding 2×2 *element matrices* are given by

$$\hat{M}_L = \frac{\Delta x}{2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad \hat{K} = \frac{v}{2} \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}. \tag{24}$$

After the global matrix assembly, the central difference discretization of the convective term is recovered at interior nodes:

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = -v \, \frac{u_{i+1} - u_{i-1}}{2\,\Delta x}. \tag{25}$$

The negative coefficient in the upper-right corner of $\hat{K}$ violates the LED criterion and should be eliminated. To this end, the artificial diffusion operator $\hat{D}$ is designed to be a symmetric matrix with zero row and column sums such that the entry $\hat{l}_{12}$ of the low-order operator $\hat{L} = \hat{K} + \hat{D}$ is equal to zero

$$\hat{D} = \frac{v}{2}\begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \quad \Rightarrow \quad \hat{L} = v\begin{bmatrix} 0 & 0 \\ 1 & -1 \end{bmatrix}. \tag{26}$$

The resulting local extremum diminishing scheme is the least diffusive among linear ones. Remarkably, it is equivalent to the standard upwind method

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = -v\,\frac{u_i - u_{i-1}}{\Delta x} \tag{27}$$

and proves positivity-preserving under condition (18) which reduces to

$$v\frac{\Delta t}{\Delta x} \leq \frac{1}{1-\theta}, \qquad 0 \leq \theta < 1. \tag{28}$$

Recall that this restriction does not apply to the backward Euler time-stepping which corresponds to ‘upwinding in time’ and is only first-order accurate.

5.2 Classical TVD Methodology

Let us stay in one dimension for a while in order to review the basic concepts and principles behind Harten’s TVD schemes [16],[17]. It is well known that the numerical diffusion inherent to the upwind method is proportional to Δx so that even this ‘optimal’ LED scheme is only first-order accurate. The quality of the results can be dramatically improved by applying a nonlinear antidiffusive correction $\hat{F}(u)$ to the monotone operator $\hat{L}$. In essence, the final transport operator $\hat{K}^*(u) = \hat{L} + \hat{F}(u)$ is constructed by removing a certain fraction of the artificial diffusion which was added to the high-order discretization to suppress spurious oscillations. Specifically, we consider

$$\hat{F}(u) = -\Phi(r_i)\hat{D} \quad \Rightarrow \quad \hat{K}^*(u) = \hat{K} + [1 - \Phi(r_i)]\hat{D}, \tag{29}$$

where the flux limiter Φ determines the magnitude of admissible antidiffusion in an adaptive fashion. As a rule of thumb, the blending factor $\Phi(r_i)$ should be equal to zero in the vicinity of steep gradients and approach (or even exceed) unity in regions where the solution is sufficiently smooth.

The smoothness sensor r_i is typically defined to be the ratio of consecutive gradients which is to be evaluated at the upwind node

$$r_i = \frac{u_i - u_{i-1}}{u_{i+1} - u_i}. \tag{30}$$

Obviously, this quantity is negative at a local extremum (see Fig. 2), relatively small for smooth data and large if the solution tends to change abruptly.

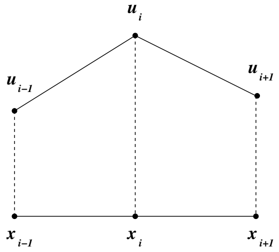


Fig. 2. Three-point stencil in one dimension.

The global matrix assembly yields a conservative finite difference scheme

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} + \frac{f_{i+1/2} - f_{i-1/2}}{\Delta x} = 0, \tag{31}$$

where the numerical fluxes $f_{i\pm1/2}$ correspond to a nonlinear combination of first- and second-order approximations to the continuous flux function vu

$$f_{i+1/2} = vu_i + \frac{v}{2}\Phi(r_i)(u_{i+1} - u_i). \tag{32}$$

The resulting semi-discretized equations can be rewritten in the form

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = c_{i-1/2}(u_{i-1} - u_i) + c_{i+1/2}(u_{i+1} - u_i), \tag{33}$$

where the pair of (possibly nonlinear) coefficients $c_{i\pm1/2}$ can be defined in many different ways. In particular, the following representation is feasible

$$c_{i-1/2} = \frac{v}{2\Delta x}\left[2 + \frac{\Phi(r_i)}{r_i} - \Phi(r_{i-1})\right], \qquad c_{i+1/2} = 0. \tag{34}$$

To satisfy the LED criterion and meet the requirements of Harten's theorem [16], the expression in the brackets must be nonnegative. A variety of flux limiters have been proposed in the literature to enforce this condition. It was shown by Jameson [19] that most of them can be interpreted and implemented as *limited average operators* $\mathcal{L}(a,b)$ such that $\Phi(r) = \mathcal{L}(1,r)$. These two-parameter functions are characterized by a number of common properties:

P1. $\mathcal{L}(a,b) = \mathcal{L}(b,a)$.

P2. $\mathcal{L}(ca,cb) = c\mathcal{L}(a,b)$.

P3. $\mathcal{L}(a,a) = a$.

P4. $\mathcal{L}(a,b) = 0 \quad \text{if} \quad ab \le 0$.

In particular, the last one ensures that $\Phi(r) = 0$ if $r \le 0$. Thus, the accuracy of a TVD discretization inevitably degrades to the first order at local extrema.

Another important implication is the symmetry of the flux limiter

$$\Phi(r) = \mathcal{L}(1, r) = r\mathcal{L}(1/r, 1) = r\Phi(1/r) \tag{35}$$

which follows from the properties (P1) and (P2). By virtue of these relations, the *antidiffusive flux* from node $i+1$ into node i is proportional to the limited average of the slopes and has the same effect as a diffusive flux from node $i-1$ provided that the coefficient $\Phi(1/r_i)$ is greater than zero:

$$\Phi(r_i)(u_{i+1} - u_i) = \mathcal{L}(u_{i+1} - u_i, u_i - u_{i-1}) = \Phi(1/r_i)(u_i - u_{i-1}). \tag{36}$$

In other words, the task of the limiter is to guarantee that the antidiffusive flux can be expressed as a 'diffusive' one for which the corresponding off-diagonal coefficient is nonnegative. This is sufficient to satisfy the LED constraint.

Note that the averaging operators $\mathcal{L}$ are applied not to the slope ratio r_i but to its nominator and denominator so that division by zero is ruled out. Some of the standard TVD limiters written in this form are as follows [19]

minmod:	$\mathcal{L}(a,b) = \mathcal{S}(a,b)\ \min\{\lvert a\rvert, \lvert b\rvert\}$
Van Leer:	$\mathcal{L}(a,b) = \mathcal{S}(a,b)\ \dfrac{2\lvert a\rvert\lvert b\rvert}{\lvert a\rvert+\lvert b\rvert}$
MC:	$\mathcal{L}(a,b) = \mathcal{S}(a,b)\ \min\left\{\frac{\lvert a+b\rvert}{2}, 2\lvert a\rvert, 2\lvert b\rvert\right\}$
superbee:	$\mathcal{L}(a,b) = \mathcal{S}(a,b)\ \max\{\min\{2\lvert a\rvert, \lvert b\rvert\}, \min\{\lvert a\rvert, 2\lvert b\rvert\}\}$

$$\text{where} \quad \mathcal{S}(a,b) = \frac{\operatorname{sign}(a) + \operatorname{sign}(b)}{2} = \begin{cases} 1 & \text{if } a > 0 \ \wedge \ b > 0, \\ -1 & \text{if } a < 0 \ \wedge \ b < 0, \\ 0 & \text{otherwise.} \end{cases}$$

The associated one-parameter limiter functions Φ yield correction factors lying in the range $[0, 2]$, whereby the integer values $0, 1, 2$ correspond to the upwind, central, and downwind approximation of the convective term, respectively.

5.3 Generalized TVD Formulation

Now let us proceed to algebraic flux correction in the multidimensional case. Recall that we discretized the continuity equation in space by the Galerkin method, performed mass lumping and transformed the high-order operator K into a low-order operator L by elimination of negative off-diagonal coefficients. This modification inevitably leads to a global loss of accuracy. According to the Godunov theorem [14], a nonoscillatory high-resolution scheme must be nonlinear even for a linear partial differential equation. On the other hand, the majority of real-life CFD applications are governed by *nonlinear* conservation laws to begin with, so that the computational overhead due to an iterative adjustment of implicit artificial diffusion is not very significant.

Following the algorithm presented for the finite difference TVD schemes, we employ an antidiffusive correction $F(u)$ to reduce the error incurred by discrete upwinding in smooth regions. The modified transport operator $K^*(u)$ for a generalized TVD method exhibits the following structure (cf. Fig. 1)

$$K^*(u) = L + F(u) = K + D + F(u), \tag{37}$$

where both D and $F(u)$ possess the properties of discrete diffusion operators.

In a practical implementation, the contribution of the nonlinear antidiffusive terms to the right-hand side of the final semi-discrete scheme

$$M_L \frac{\mathrm{d}u}{\mathrm{d}t} = K^* u \tag{38}$$

is assembled edge-by-edge from internodal fluxes. Specifically, we have

$$(Fu)_i = \sum_{j \neq i} f_{ij}^a \qquad \text{such that} \quad f_{ji}^a = -f_{ij}^a, \tag{39}$$

where the antidiffusive flux f_{ij}^a from node j into its **upwind** (in the sense of our orientation convention $l_{ji} \geq l_{ij}$) neighbor i depends on the diffusion coefficient d_{ij} for discrete upwinding and on the entry $l_{ji} = \max\{k_{ji}, k_{ji} - k_{ij}\}$ of the low-order transport operator:

$$f_{ij}^a := \min\{\Phi(r_i) d_{ij}, l_{ji}\}(u_i - u_j). \tag{40}$$

Furthermore, Φ is a standard one-parameter limiter applied to a suitable smoothness indicator r_i (to be specified below). By definition, the downwind node j receives the flux $f_{ji}^a := -f_{ij}^a$ of the same magnitude but with the opposite sign so that mass conservation is guaranteed.

Let us derive a *sufficient* condition for the FEM-TVD scheme (38) to be local extremum diminishing. If $\Phi(r_i) = 0$ or $d_{ij} = 0$, the antidiffusive flux f_{ij}^a vanishes and does not pose any hazard. Therefore, we restrict ourselves to the nontrivial case $f_{ij}^a \neq 0$ which implies that both $\Phi(r_i)$ and d_{ij} are strictly positive. Our objective is to prove the existence of a LED operator L^* which is equivalent to K^* for the given solution u (see the last box in Fig. 1). Clearly, the sensor r_i cannot be chosen arbitrarily. The symmetry property (35) of the limiter Φ makes it possible to represent the antidiffusive flux in the form

$$f_{ij}^a = \Phi(r_i) a_{ij} (u_i - u_j) = \Phi(1/r_i) a_{ij} \Delta u_{ij}, \tag{41}$$

where the (positive) *antidiffusion coefficient* a_{ij} and the *upwind difference* Δu_{ij} are defined as follows

$$a_{ij} := \min\{d_{ij}, l_{ji}/\Phi(r_i)\}, \qquad \Delta u_{ij} := r_i(u_i - u_j). \tag{42}$$

For the numerical solution to be nonoscillatory, the antidiffusive fluxes must behave as diffusive ones, cf. equation (36). The assumption $d_{ij} > 0$ implies that $k_{ij} < 0$ and $l_{ij} = 0$ for the edge $\overrightarrow{ij}$ which links an upwind node i and a downwind node j. Therefore, the edge contributions to the two components of the modified convective term K^*u in (38) can be written as

$$k^*_{ij}(u_j - u_i) = f^a_{ij}, \qquad k^*_{ji}(u_i - u_j) = l_{ji}(u_i - u_j) - f^a_{ij}. \tag{43}$$

The increment to node j is obviously of diffusive nature and satisfies the LED criterion, since the coefficient $k^*_{ji} = l_{ji} - \Phi(r_i)a_{ij}$ is nonnegative by construction (see the definition of a_{ij}). Furthermore, it follows from relation (41) that the negative off-diagonal entry $k^*_{ij} = -\Phi(r_i)a_{ij}$ of the nonlinear operator K^* is acceptable provided Δu_{ij} admits the following representation

$$\Delta u_{ij} = \sum_{k \neq i} \sigma_{ik}(u_k - u_i), \qquad \text{where} \quad \sigma_{ik} \geq 0, \quad \forall k \neq i. \tag{44}$$

In other words, the limited antidiffusive flux f^a_{ij} from node j into node i should be interpreted as a sum of diffusive fluxes contributed by other neighbors. It remains to devise a multidimensional smoothness indicator r_i and check if the corresponding upwind difference Δu_{ij} satisfies the above condition.

5.4 Slope-Limiter FEM-TVD Algorithm

For classical finite difference TVD schemes, r_i represents the slope ratio (30) at the upwind node, so that $\Delta u_{ij} = u_k - u_i$, where $k = i - 1$ is the second neighbor of node i. However, this natural definition of r_i is no longer possible in multidimensions, whereby each node interacts with more than two neighbors. A geometric approach commonly employed in the literature is to reconstruct a local one-dimensional stencil by insertion of equidistant *dummy nodes* on the continuation of each mesh edge [1],[19],[38],[39]. The difference Δu_{ij} is defined as before using the interpolated or extrapolated solution value at the dummy node k adjacent to the upwind node i. The construction of a three-point stencil for an unstructured triangular mesh is illustrated in Fig. 3.

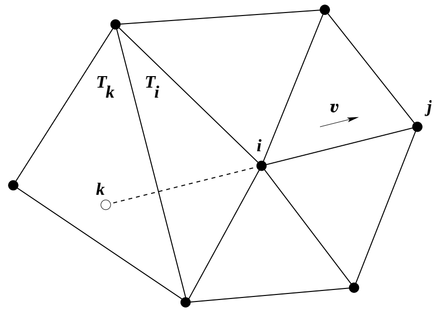


Fig. 3. Three-point stencil in two dimensions.

An evaluation of various techniques for the recovery of u_k was performed by Lyra [38]. His comparative study covers the following algorithms:

- interpolation/extrapolation using the upwind triangle T_i containing node i,
- interpolation using the actual triangle T_k containing the dummy node k,
- extrapolation using a least squares reconstruction for the nodal gradient.

Numerical experiments revealed that the solutions depend strongly on the employed strategy. The first option was proved to provide the LED property but failed to produce nonoscillatory results for some aerodynamic applications. The second procedure based on the actual triangle was favored due to the enhanced robustness as compared to the use of the adjacent triangle. However, the resulting discretization is no longer local extremum diminishing and neither is the gradient reconstruction method which corresponds to

$$\varDelta u_{ij} = (\mathbf{x}_i - \mathbf{x}_j) \cdot \nabla_h u_i, \tag{45}$$

where $\nabla_h u_i$ stands for a continuous approximation to the solution gradient at the upwind node i recovered by means of a consistent L_2-projection:

$$\nabla_h u_i = \frac{1}{m_i} \sum_{k \neq i} \mathbf{c}_{ik} (u_k - u_i). \tag{46}$$

The involved coefficients $\mathbf{c}_{ik}$ are defined in (6) for the standard Galerkin method. This approach is relatively simple to implement and more efficient than the linear interpolation techniques. However, Lyra [38] reported its performance to be quite poor and emphasized the need for the development of a more robust algorithm for the reconstruction of nodal gradients.

Let us explain why the above choice of the upwind difference may prove unsatisfactory. If any of the scalar products $\mathbf{c}_{ik} \cdot (\mathbf{x}_i - \mathbf{x}_j)$ is negative, then the formula for $\varDelta u_{ij}$ is not of the form (44), so the numerical scheme may fail to satisfy the LED criterion. To rectify this, one can employ a monotone projection operator constructed by resorting to discrete upwinding. Note that $\mathbf{c}_{ki} = -\mathbf{c}_{ik}$ for internal nodes, so that the elimination of negative off-diagonal coefficients leads to the following LED-type reconstruction procedure [27]

$$\varDelta u_{ij} = \frac{2}{m_i} \sum_{k \neq i} \max\{0, \mathbf{c}_{ik} \cdot (\mathbf{x}_i - \mathbf{x}_j)\} (u_k - u_i). \tag{47}$$

In one dimension, this kind of extrapolation corresponds to using the upwind gradient and yields $\varDelta u_{ij} = u_k - u_i$, where k is the upwind neighbor of i.

By virtue of (42), the upwind difference $\varDelta u_{ij}$ can be converted into the smoothness sensor r_i which provides an estimate of the gradient jump along the edge $\overrightarrow{ij}$. Hence, this geometric approach to the design of nonlinear LED schemes will be referred to as the *slope-limiter* FEM-TVD algorithm. It may be equipped with any standard limiter Φ. At the same time, the numerical results are rather sensitive to the alignment of the three-point stencil and to the algorithm employed to recover the solution values at the dummy nodes. Moreover, such methods are computationally expensive and may experience severe convergence problems for steady-state applications. This shortcoming was also noticed by Lyra [38] who explained it by the lack of background dissipation and indicated that the convergence rates can be improved to some extent by 'freezing' the antidiffusive terms as the solution approaches the steady state. The main reason for the insufficient robustness seems to be the unidirectional nature of stencil reconstruction and the independent limiting of antidiffusive fluxes associated with the same upwind node.

5.5 Flux-Limiter FEM-TVD Algorithm

Let us abandon the stencil reconstruction technique and design the sensor r_i in a different way. The one-dimensional convection equation (23) discretized in space by the lumped-mass Galerkin FEM or by the central difference method can be written in the form (33) where the two coefficients are given by

$$c_{i-1/2} = \frac{v}{2\Delta x} > 0, \qquad c_{i+1/2} = -\frac{v}{2\Delta x} < 0. \tag{48}$$

Remarkably, the ratio of the upwind and downwind contributions to node i

$$r_i = \frac{c_{i-1/2}(u_{i-1} - u_i)}{c_{i+1/2}(u_{i+1} - u_i)} \tag{49}$$

reduces to the slope ratio r_i defined in (30) as long as the velocity v is constant. Moreover, this interpretation leads to a conceptually different limiting strategy which guarantees the TVD property for variable velocity fields and carries over to multidimensions. As we are about to see, the new algorithm is akin to that proposed by Zalesak [54] in the framework of flux-corrected transport methods, so we adopt his notation to reflect this relationship.

In the multidimensional case, the incompressible part of the original convective term Ku can be decomposed into a sum of edge contributions with negative coefficients and a sum of those with positive coefficients

$$P_i = \sum_{j\neq i} \min\{0, k_{ij}\}(u_j - u_i), \quad Q_i = \sum_{j\neq i} \max\{0, k_{ij}\}(u_j - u_i) \tag{50}$$

which are due to mass transfer from the downstream and upstream directions, respectively. The sum P_i is composed from the *raw antidiffusive fluxes* which offset the error incurred by elimination of negative matrix entries in the course of discrete upwinding. They are responsible for the formation of spurious wiggles and must be securely limited. At the same time, the constituents of the sum Q_i are harmless since they resemble diffusive fluxes and do satisfy the LED criterion. Thus, it is natural to require that the net antidiffusive flux into node i be a limited average of the original increments P_i and Q_i.

Due to the property (P4) of TVD limiters, it is worthwhile to distinguish between the positive and negative edge contributions to both sums

$$P_i = P_i^+ + P_i^-, \qquad P_i^\pm = \sum_{j \neq i} \min\{0, k_{ij}\} \begin{matrix}\min\\ \max\end{matrix}\{0, u_j - u_i\}, \tag{51}$$

$$Q_i = Q_i^+ + Q_i^-, \qquad Q_i^\pm = \sum_{j \neq i} \max\{0, k_{ij}\} \begin{matrix}\max\\ \min\end{matrix}\{0, u_j - u_i\} \tag{52}$$

and limit the positive and negative antidiffusive fluxes separately. To this end, we pick a standard limiter Φ and compute the *nodal correction factors*

$$R_i^\pm = \Phi(Q_i^\pm / P_i^\pm) \tag{53}$$

which determine the percentage of $P_i^\pm$ that can be retained without violating the LED constraint for row i of the modified transport operator K^*. Clearly, $R_i^\pm$ does not need to be evaluated if the raw antidiffusion $P_i^\pm$ vanishes.

For each edge $\overrightarrow{ij}$ of the sparsity graph, the antidiffusive flux f_{ij}^a from its downwind node j into the upwind node i is constructed as follows:

$$f_{ij}^a := \begin{cases} \min\{R_i^+ d_{ij}, l_{ji}\}(u_i - u_j) & \text{if} \quad u_i \geq u_j, \\ \min\{R_i^- d_{ij}, l_{ji}\}(u_i - u_j) & \text{if} \quad u_i < u_j, \end{cases} \qquad f_{ji}^a := -f_{ij}^a. \tag{54}$$

Importantly, the same correction factor $R_i^\pm$ is applied to all positive/negative antidiffusive fluxes which represent the interactions of node i with its neighbors located downstream in the sense of our orientation convention.

The node-oriented limiting strategy makes it possible to control the combined effect of antidiffusive fluxes *acting in concert* rather than merely the variation of the solution along each edge. Moreover, the new limiter extracts all information from the original matrix K and does not need the coordinates of nodes or other geometric details. The equivalence of (40) and (54) reveals that the underlying smoothness indicator r_i is implicitly defined by

$$r_i = \begin{cases} Q_i^+ / P_i^+ & \text{if} \quad u_i \geq u_j, \\ Q_i^- / P_i^- & \text{if} \quad u_i < u_j. \end{cases} \tag{55}$$

It is easy to verify that $\Delta u_{ij} = r_i(u_i - u_j)$ satisfies condition (44) since all coefficients in the sum of upwind contributions $Q_i^\pm$ are nonnegative and

$$\Delta u_{ij} = \sigma_{ij} Q_i^\pm, \qquad \text{where} \quad \sigma_{ij} = \begin{matrix}\max\\ \min\end{matrix}\{0, u_i - u_j\}/P_i^\pm \geq 0. \tag{56}$$

Thus, our nonlinear semi-discrete scheme (38) proves local extremum diminishing if the antidiffusive fluxes are computed from (54). In the sequel, we will call the new algorithm the *flux-limiter* FEM-TVD method to distinguish it from the one described in the preceding section. In one dimension, both generalizations of TVD schemes reduce to their finite difference prototype.

5.6 Iterative Defect Correction

The algorithm presented so far can be classified as a *method of lines* which starts with an approximation of spatial derivatives and yields a system of coupled ordinary differential equations for the time-dependent nodal values u_i. In principle, the discretization of system (38) in time can be performed by any numerical method for solution of initial value problems. First- or second-order accuracy is sufficient for our purposes, so we can use the standard θ-scheme. Furthermore, we concentrate on implicit time-stepping methods $(0 < \theta \leq 1)$ because the implementation of the fully explicit one is straightforward. As a result, we end up with a nonlinear algebraic system of the form

$$M_L \frac{u^{n+1} - u^n}{\Delta t} = \theta K^*(u^{n+1})u^{n+1} + (1-\theta)K^*(u^n)u^n \tag{57}$$

which must be solved iteratively. According to the positivity constraint (18), the time step Δt is subject to a CFL-like condition unless $\theta = 1$.

Successive approximations to the end-of-step solution u^{n+1} can be computed e. g. by the fixed-point defect correction scheme [52]

$$u^{(m+1)} = u^{(m)} + A^{-1}r^{(m)}, \qquad m = 0, 1, 2, \dots \tag{58}$$

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

In a practical implementation, the 'inversion' of A is also performed by a suitable iterative method for solving the linear subproblem

$$A\Delta u^{(m+1)} = r^{(m)}, \qquad m = 0, 1, 2, \dots \tag{59}$$

After a certain number of inner iterations, the solution increment $\Delta u^{(m+1)}$ is applied to the last iterate, whereby u^n provides a reasonable initial guess

$$u^{(m+1)} = u^{(m)} + \Delta u^{(m+1)}, \qquad u^{(0)} = u^n. \tag{60}$$

Incidentally, the auxiliary problem (59) does not have to be solved very accurately at each outer iteration. By construction, the evolution operator

$$A = M_L - \theta \Delta t L, \qquad L = K + D \tag{61}$$

for the underlying linear LED scheme (21) enjoys the M-matrix property and constitutes an excellent preconditioner. Furthermore, the diagonal dominance of A can be enhanced by means of an implicit underrelaxation [12]. In fact, iterative defect correction preconditioned by the monotone upwind operator is frequently used to enhance the robustness of CFD solvers. This practice is to be recommended even in the linear case, since an iterative method may fail to converge if applied directly to the ill-conditioned matrix originating from a high-order discretization of the troublesome convective terms.

The defect vector and the constant right-hand side are given by

$$r^{(m)} = b^n - [A - \theta \Delta t F(u^{(m)})]u^{(m)}, \tag{62}$$

$$b^n = M_L u^n + (1-\theta)\Delta t[L + F(u^n)]u^n. \tag{63}$$

Both expressions consist of a low-order contribution augmented by limited antidiffusion of the form (39). The antidiffusive fluxes f_{ij}^a are evaluated edge-by-edge at the corresponding time level and inserted into the global vectors. If they are omitted, we recover the nonoscillatory linear scheme (21) which is overly diffusive. The task of the flux limiter is to determine how much artificial diffusion can be safely removed without violating the LED criterion. We remark that our FEM-TVD algorithm is directly applicable to steady-state problems as well as to time-dependent equations written as stationary boundary value problems in the space-time domain (see below).

5.7 Summary of the FEM-TVD Algorithm

The multidimensional flux limiter of TVD type is easy to implement in a finite element code using the conventional or the edge-based data structure. In fact, the required modifications are limited to the matrix assembly routine which is to be called repeatedly in the outer defect correction loop. Since the origin of the discrete transport operator K is immaterial, finite difference and finite volume discretizations of the form (5) are also admissible. The sequence of 'postprocessing' steps to be performed can be summarized as follows:

In a loop over edges:

1. Retrieve the entries k_{ij} and k_{ji} of the high-order transport operator.
2. Determine the artificial diffusion coefficient d_{ij} from equation (20).
3. Update the four entries of the preconditioner A as required by (22).
4. Adopt the edge orientation $\overrightarrow{ij}$ such that node i is located upwind.
5. Store the order of nodes as well as d_{ij} and l_{ji} for future reference.

In a loop over nodes:

6. Calculate the ratio of upstream/downstream contributions $Q_i^{\pm}$ and $P_i^{\pm}$.
7. Apply a TVD limiter Φ to obtain the nodal correction factors $R_i^{\pm}$.

In a loop over edges:

8. Compute the diffusive flux $f_{ij}^d = d_{ij}(u_j - u_i)$ due to discrete upwinding.
9. Check the sign of $u_i - u_j$ and evaluate the antidiffusive flux f_{ij}^a from (54).
10. Insert the corrected internodal flux $f_{ij} = f_{ij}^d + f_{ij}^a$ into the defect vector.

The performance of the above algorithm will be illustrated by the numerical examples at the end of this chapter. The slope-limiter version can be coded in a similar way using the upwind difference (47) to determine the slope ratio $r_i = \Delta u_{ij}/(u_i - u_j)$ and the corresponding correction factors $\Phi(r_i)$. Note that no loop over nodes is needed in this case. Indeed, the recovery of Δu_{ij} via stencil reconstruction is performed independently for each edge. As an alarming consequence, the contributions of other edges are not taken into account, so that the total antidiffusive flux cannot be properly controlled. However, on regular grids the numerical results are typically quite good [27].

6 Algebraic Flux Correction of FCT Type

In contrast to the algebraic TVD methods presented above, FCT algorithms of this kind are designed at the *fully discrete* level. At the same time, the basic principles of algebraic flux correction remain unchanged. The matrix manipulations to be performed (see Fig. 4) are very similar to those outlined in Fig. 1. In particular, the nonoscillatory low-order scheme is constructed by resorting to discrete upwinding and the accuracy is improved by adding nonlinear antidiffusion $f(u^{n+1}, u^n)$ controlled by the flux limiter. Criterion (16) is invoked to ensure that the positivity of u^n is inherited by the auxiliary solution $\tilde{u}$ to an explicit subproblem and carries over to u^{n+1} [22],[23],[29].

1. Linear high-order scheme (e.g. Galerkin FEM)

$$M_C \frac{u^{n+1} - u^n}{\Delta t} = \theta K u^{n+1} + (1-\theta) K u^n, \quad \exists\, j \neq i: \; k_{ij} < 0$$

2. Linear low-order scheme $L = K + D$

$$M_C \frac{u^{n+1} - u^n}{\Delta t} = \theta L u^{n+1} + (1-\theta) L u^n, \quad l_{ij} \geq 0, \; \forall j \neq i$$

3. Nonlinear high-resolution scheme $f_i = \sum_{j \neq i} f_{ij}^a$

$$M_L \frac{u^{n+1} - u^n}{\Delta t} = \theta L u^{n+1} + (1-\theta) L u^n + f(u^{n+1}, u^n)$$

Equivalent representation $A u^{n+1} = B(\tilde{u})\tilde{u}$, where

$A = M_L - \theta \Delta t L$ is an M-matrix and $b_{ij} \geq 0, \; \forall i, j$

Fig. 4. Roadmap of AFC-FCT manipulations.

6.1 High- and Low-Order Schemes

The crux of the generalized FCT methodology is to switch between the underlying high- and low-order discretizations in an adaptive fashion so as to satisfy the algebraic positivity constraint. Consider the difference between the nodal ODE systems (5) and (21) which can be formally written as

$$P(u) = (M_L - M_C)\frac{\mathrm{d}u}{\mathrm{d}t} - Du, \tag{64}$$

where the matrices $M_C - M_L$ and $D = L - K$ represent discrete diffusion operators as defined at the beginning of this chapter. Therefore, the raw antidiffusion $P(u)$ can be decomposed into a sum of internodal fluxes

$$P_i = \sum_{j\neq i} f_{ij}, \qquad f_{ij} = -\left[m_{ij}\frac{\mathrm{d}}{\mathrm{d}t} + d_{ij}\right](u_j - u_i), \quad f_{ji} = -f_{ij}. \tag{65}$$

Note that the semi-discrete flux f_{ij} contains a time derivative multiplied by an entry of the consistent mass matrix M_C. For stationary problems, the increment P_i reduces to the sum of *downwind* edge contributions (50) which are associated with the negative off-diagonal coefficients of the operator K.

After the discretization in time by the θ-scheme, the algebraic systems for the high- and low-order methods are related by the formula [23]

$$M_L\frac{u^{n+1} - u^n}{\Delta t} = \theta L u^{n+1} + (1-\theta)Lu^n + P(u^{n+1}, u^n). \tag{66}$$

The last term in the right-hand side is a fully discrete counterpart of $P(u)$. It represents the amount of compensating antidiffusion that needs to be added to the upwind-like low-order scheme to recover the original high-order one (cf. Fig. 4). It is worth mentioning that the discrete operators K, D and $L = K + D$ may depend on the solution if the velocity field does. To simplify notation, we do not indicate this dependence explicitly.

Iterative defect correction makes it possible to resolve the nonlinearities inherent to the governing equation and/or to the discretization procedure into a sequence of well-behaved linear systems of the form (59). The corresponding defect vector for the $m-$th outer iteration is given by

$$r^{(m)} = b^{(m+1)} - Au^{(m)}. \tag{67}$$

The preconditioner A is defined in (61) whereas the load vector

$$b^{(m+1)} = b^n + P(u^{(m)}, u^n) \tag{68}$$

is composed from the right-hand side for the low-order scheme

$$b^n = [M_L + (1-\theta)\Delta t L]u^n \tag{69}$$

and the sum of fully discretized raw antidiffusive fluxes such that

$$P_i^{(m)} = \sum_{j\neq i} f_{ij}^{(m)}, \qquad f_{ji}^{(m)} = -f_{ij}^{(m)}. \tag{70}$$

It follows from (65) that the flux $f_{ij}^{(m)}$ from node j into node i reads

$$f_{ij}^{(m)} = -m_{ij}[\Delta u_{ij}^{(m)} - \Delta u_{ij}^n] - \theta \Delta t d_{ij}^{(m)} \Delta u_{ij}^{(m)} - (1-\theta)\Delta t d_{ij}^n \Delta u_{ij}^n, \tag{71}$$

where the explicit and implicit antidiffusion is proportional to

$$\Delta u_{ij}^n = u_j^n - u_i^n \quad \text{and} \quad \Delta u_{ij}^{(m)} = u_j^{(m)} - u_i^{(m)}, \tag{72}$$

respectively. The implicit part must be updated in each defect correction cycle while the explicit one is to be computed just once per time step.

Substitution of expressions (61) and (67) into (58) reveals that successive approximations to the high-order solution can be computed as follows

$$Au^{(m+1)} = b^{(m+1)}, \qquad m = 0, 1, 2, \dots \tag{73}$$

By construction, the raw antidiffusive fluxes offset not only the error induced by discrete upwinding but also the numerical diffusion due to mass lumping that could not be removed in the framework of the FEM-TVD methodology. Therefore, FEM-FCT schemes are typically more accurate for strongly time-dependent problems which call for the use of the consistent mass matrix M_C. At the same time, the use of the lumped mass matrix M_L is appropriate for less dynamic ones or those being marched to a steady state. In this case, the first term in the right-hand side of (71) should be omitted.

An extra stabilization of convective terms appears to be necessary for fully explicit schemes [23]. Stabilized finite element methods are typically based on some kind of streamline diffusion which can be added explicitly, incorporated into the test function or emulated by high-order time derivatives in the Taylor series expansion [7],[8],[10]. Recall that streamline diffusion operators are of the form (8)-(9) so that decomposition (11) is feasible. We refer to Löhner *et al.* [36],[37],[34] for a presentation of the explicit FEM-FCT algorithm and restrict ourselves to implicit Galerkin schemes. They enjoy unconditional stability for $\theta \geq 0.5$ and constitute viable high-order methods as long as spurious undershoots and overshoots are precluded by a built-in flux limiter.

6.2 Basic FEM-FCT Algorithm

Due to the fact that the preconditioner A was designed to be an M-matrix, the left-hand side of (73) already satisfies the algebraic constraint (16) and so does the right-hand side if the antidiffusive correction $P(u^{n+1}, u^n)$ is omitted. Clearly, it is desirable to retain as much antidiffusion as possible without generating new extrema and accentuating already existing ones. To this end, the raw antidiffusive fluxes should be multiplied by appropriate correction factors before they are inserted into the right-hand side. This adjustment should guarantee that the discrete scheme remains positivity-preserving.

A generalized FEM-FCT formulation based on algebraic flux correction was introduced in [22],[23],[25]. The limited antidiffusive fluxes belong into the right-hand side of system (73) which is to be redefined as follows

$$b_i^{(m+1)} = b_i^n + \sum_{j \neq i} \alpha_{ij}^{(m)} f_{ij}^{(m)}, \qquad 0 \leq \alpha_{ij}^{(m)} \leq 1. \tag{74}$$

It is easy to verify that the usual FCT algorithm is recovered for $\theta = 0$. Let us leave the solution-dependent correction factors $\alpha_{ij}^{(m)}$ unspecified for the time being but draw attention to the fact that they are bounded by 1 whereas flux limiters of TVD type are allowed to accept more than 100% of the raw antidiffusive flux. Another notable distinction between the two techniques is that FCT limiters are invariant to the edge orientation. Hence, it is no longer necessary to check which of the two nodes is located upwind (cf. section 6).

As already mentioned above, the antidiffusive fluxes should be limited so as to enforce the AFC constraint (16) making use of an intermediate solution which is supposed to be positivity-preserving. Consider the subproblem

$$M_L \tilde{u}^n = b^n \tag{75}$$

such that u^n corresponds to the explicit low-order solution at the instant $t^{n+1-\theta}$ and reduces to $\tilde{u}^n = u^n$ in the case $\theta = 1$ (backward Euler method). Other time-stepping schemes preserve the positivity of u^n provided that

$$\Delta t \leq \frac{1}{1-\theta} \min_i \{-m_i/l_{ii} | \ l_{ii} < 0\}, \qquad 0 \leq \theta < 1. \tag{76}$$

This readily computable upper bound follows from our CFL-like condition (18) and can serve as the threshold parameter for an adaptive time step control. We remark that the intermediate solution $\tilde{u}^n$ is independent of the iteration counter m and does not change in the course of defect correction.

For the right-hand side $b^{(m+1)}$ to possess the desired representation, the flux limiter should guarantee the existence of a matrix $B = \{b_{ij}\}$ such that

$$b^{(m+1)} = B(\tilde{u}^n)\tilde{u}^n \qquad \text{and} \quad b_{ij} \geq 0, \quad \forall i, j. \tag{77}$$

Under this condition, the resulting scheme proves positivity-preserving since the linear system (73) for each solution update can be cast in the form (16) with $\tilde{u}^n$ in lieu of u^n. It goes without saying that the matrix $B(\tilde{u}^n)$ does not need to be constructed explicitly. We will see shortly that a new interpretation of Zalesak's limiter provides the necessary mechanism for the computation of 'optimal' correction factors for our algebraic FCT schemes.

6.3 Iterative FEM-FCT Algorithm

The main advantage of implicit FEM-FCT schemes is their ability to cope with large time steps. However, the artificial diffusion introduced by discrete upwinding is proportional to the time step, while the amount of acceptable antidiffusion depends solely on the local extrema of the auxiliary solution. Hence, a smaller percentage of the raw antidiffusive flux survives the limiting step as the local Courant number increases. To circumvent this deficiency of our basic FEM-FCT algorithm, we introduce an iterative limiting strategy which prevents the numerical solution from becoming increasingly diffusive at large time steps. A somewhat similar technique was developed by Schär and Smolarkiewicz [47] in the finite difference context but their methodology is inherently explicit so that iterative flux correction does not pay off.

The iterative FEM-FCT procedure [29] differs from the algorithm presented above in that the previously accepted antidiffusion is taken into account and only the rejected portion of the antidiffusive flux needs to be dealt with at subsequent defect correction steps. To this end, the limited antidiffusion is incorporated into the *variable* auxiliary solution $\tilde{u}^{(m)}$ which must be updated along with the right-hand side of (73) at each outer iteration

$$M_L\tilde{u}^{(m)} = b^{(m)}, \qquad b^{(0)} = b^n. \tag{78}$$

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.

Furthermore, the correction factors $\alpha_{ij}^{(m)}$ are based on the local extrema of $\tilde{u}^{(m)}$ rather than $\tilde{u}^n$ and applied to the difference between the raw antidiffusive flux $f_{ij}^{(m)}$ and the cumulative effect of previous corrections $g_{ij}^{(m)}$ which is initialized by zero at the beginning of a new time step

$$\Delta f_{ij}^{(m)} = f_{ij}^{(m)} - g_{ij}^{(m)}, \qquad g_{ij}^{(0)} = 0. \tag{79}$$

The limited flux difference is added to the sum of its predecessors

$$g_{ij}^{(m+1)} = g_{ij}^{(m)} + \alpha_{ij}^{(m)}\Delta f_{ij}^{(m)} \tag{80}$$

and inserted into the right-hand side of the linear system to be solved

$$b_i^{(m+1)} = b_i^{(m)} + \sum_{j\neq i}\alpha_{ij}^{(m)}\Delta f_{ij}^{(m)}. \tag{81}$$

At the first outer iteration, Zalesak's limiter is applied to $\Delta f_{ij}^{(0)} = f_{ij}^{(0)}$ and $\tilde{u}^{(0)} = \tilde{u}^n$ so that the load vector $b^{(1)}$ is identical to that defined in (74). As the iteration process continues, more and more antidiffusion can be built into the auxiliary solution $\tilde{u}^{(m)}$ while the remainder $\Delta f_{ij}^{(m)}$ gradually shrinks. This simplifies the task of the flux limiter and enables it to remove excessive artificial diffusion step-by-step in a positivity-preserving manner.

It is easy to verify by successive substitution that the load vector $b^{(m+1)}$ consists of the low-order contribution b^n and the limited antidiffusion accumulated in the course of iterative defect/flux correction:

$$b_i^{(m+1)} = b_i^n + \sum_{j\neq i} g_{ij}^{(m+1)}, \qquad g_{ij}^{(m+1)} = \sum_{k=0}^{m} \alpha_{ij}^{(k)} \Delta f_{ij}^{(k)}. \tag{82}$$

Moreover, the right-hand side for the high-order Galerkin discretization (68) is recovered if no limiting of $\Delta f_{ij}^{(m)}$ is performed in the $m-$th iteration

$$\alpha_{ij}^{(m)} \equiv 1 \quad \Rightarrow \quad b_i^{(m+1)} = b_i^n + \sum_{j\neq i}(g_{ij}^{(m)} + \Delta f_{ij}^{(m)}) = b_i^n + \sum_{j\neq i} f_{ij}^{(m)}.$$

The task of the flux limiter it to select the correction factors $\alpha_{ij}^{(m)}$ so as to satisfy an analog of (77) and make each solution update $\tilde{u}^{(m+1)} = M_L^{-1} b^{(m+1)}$ positivity-preserving. In the fully explicit case (forward Euler time-stepping, lumped mass matrix) just one iteration is necessary so that $u^{n+1} = \tilde{u}^{(1)}$ is the final solution. For our implicit FEM-FCT schemes, the M-matrix property of the preconditioner A ensures that $u^{(m+1)} \geq 0$ as long as $\tilde{u}^{(m+1)} \geq 0$.

6.4 Zalesak's Limiter

Let us describe Zalesak's algorithm for the computation of correction factors and check if the right-hand side satisfies the following positivity constraint:

$$b_i = m_i \tilde{u}_i + \sum_{j\neq i} \alpha_{ij} f_{ij} \geq 0 \qquad \text{if} \quad \tilde{u}_j \geq 0, \quad \forall j. \tag{83}$$

Note that both (74) and (81) are of this form, so there is no need to distinguish between the basic and iterative limiting strategy in this section.

Varying the correction factors α_{ij} between zero and unity, one can blend the high-order method with the concomitant low-order one. The latter should be used in the vicinity of steep gradients where spurious oscillations are likely to arise. The objective is to control the interplay of antidiffusive fluxes so that they cannot conspire to create or enhance a local extremum [54]. Moreover, all antidiffusive fluxes directed down the gradient of $\tilde{u}$ should be canceled from the outset to prevent the formation of plateaus amidst a steep front

$$f_{ij} := 0 \qquad \text{if} \quad f_{ij}(\tilde{u}_i - \tilde{u}_j) \leq 0. \tag{84}$$

In other words, the antidiffusive flux is not allowed to flatten the auxiliary solution. This optional *prelimiting step* can be traced back to the SHASTA scheme of Boris and Book who designed their limiter so as to reverse (rather than cancel) such 'defective' fluxes [6]. Prelimiting of the form (84) was introduced by Zalesak [54] in relations (14) and (14') of his paper. At the same time, he argued that the effect of the above amendment is marginal and cosmetic in nature since the vast majority of antidiffusive fluxes entail a steepening of the gradient. Two decades later, the virtues of prelimiting were rediscovered by DeVore [9] who explained its ramifications and demonstrated that it may lead to an appreciable improvement of simulation results. If this step is missing, the FCT limiter appears to be positivity- but not monotonicity-preserving so that solutions may be corrupted by numerical ripples of significant amplitude. This fact was also confirmed by our own numerical experiments [23]. Hence, it is advisable to prelimit the fluxes prior to the computation of α_{ij}.

In the worst case, all antidiffusive fluxes into node i have the same sign. Therefore, it is worthwhile to split the increment P_i as defined in (70) into a sum of positive contributions and a sum of negative ones, cf. (51)

$$P_i = P_i^+ + P_i^-, \qquad P_i^\pm = \sum_{j\neq i} \max_{\min}\{0, f_{ij}\}. \tag{85}$$

The maximum/minimum admissible increment depends on the solution values at the neighboring nodes that share an element/edge with node i

$$Q_i^\pm = \max_{\min} \Delta u_{ij}^\pm, \qquad \text{where} \quad \Delta u_{ij}^\pm = \max_{\min}\{0, \tilde{u}_j - \tilde{u}_i\}. \tag{86}$$

This corresponds to the following upper/lower bounds for the nodal value

$$\tilde{u}_i^{\max} = \tilde{u}_i + Q_i^+, \qquad \tilde{u}_i^{\min} = \tilde{u}_i + Q_i^-.$$

In order to prevent the formation of a spurious overshoot/undershoot, the positive/negative antidiffusive flux f_{ij} should be multiplied by

$$R_i^\pm = \begin{cases} \min\{1, m_i Q_i^\pm / P_i^\pm\} & \text{if } P_i^\pm \neq 0, \\ 1 & \text{if } P_i^\pm = 0. \end{cases} \tag{87}$$

In our experience, it makes sense to set $R_i^\pm := 1$ at the inlet, where the Dirichlet boundary conditions override the effect of any antidiffusive correction. The same adjustment can/should be performed at outflow boundaries [29],[41].

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, one should check the sign of the flux and apply the minimum of the nodal correction factors

$$\alpha_{ij} = \begin{cases} \min\{R_i^+, R_j^-\} & \text{if } f_{ij} \geq 0, \\ \min\{R_j^+, R_i^-\} & \text{if } f_{ij} < 0, \end{cases} \qquad \alpha_{ji} = \alpha_{ij}. \tag{88}$$

This choice of α_{ij} is safe enough to guarantee that $\tilde{u}_i^{\min} \leq b_i/m_i \leq \tilde{u}_i^{\max}$ so that no enhancement of local extrema takes place. Remarkably, the above algorithm is independent of the underlying discretization and can be implemented as a 'black-box' routine which computes the correction factors for a given auxiliary solution $\tilde{u}$ and an array of raw antidiffusive fluxes f_{ij}.

It remains to prove that the right-hand side b satisfies (83) for a nontrivial parameter constellation such that the antidiffusive correction to node i is nonvanishing. Let k be the number of a neighboring node such that [22],[23]

$$b_i = m_i \tilde{u}_i + c_i Q_i = (m_i - c_i)\tilde{u}_i + c_i \tilde{u}_k, \tag{89}$$

where the auxiliary quantities c_i and Q_i are defined as follows

$$c_i = \frac{\sum_{j \neq i} \alpha_{ij} f_{ij}}{Q_i}, \qquad Q_i = \begin{cases} Q_i^+ & \text{if } \sum_{j\neq i} \alpha_{ij} f_{ij} > 0, \\ Q_i^- & \text{if } \sum_{j\neq i} \alpha_{ij} f_{ij} < 0. \end{cases} \tag{90}$$

Note that division by zero is ruled out since $R_i^\pm = 0$ and all positive/negative antidiffusive fluxes into node i are canceled completely in this case:

$$f_{ij} := 0 \qquad \text{if} \quad (Q_i^+ = 0 \ \wedge \ f_{ij} > 0) \ \vee \ (Q_i^- = 0 \ \wedge \ f_{ij} < 0).$$

By definition, the coefficient c_i is nonnegative and it is easy to verify that the following estimate holds by virtue of relations (85)–(88)

$$m_i Q_i^- \leq m_i R_i^- P_i^- \leq \sum_{j \neq i} \alpha_{ij} f_{ij} \leq m_i R_i^+ P_i^+ \leq m_i Q_i^+. \tag{91}$$

Thus, the limited antidiffusive fluxes satisfy the double inequality $m_i \geq c_i \geq 0$ and it follows from (89) that b_i is nonnegative for $\tilde{u}_i \geq 0$ and $\tilde{u}_k \geq 0$. Moreover, the positivity of both coefficients in this representation of the right-hand side proves the existence of the matrix $B(\tilde{u})$ in the last box of Fig. 4.

6.5 Clipping and Terracing

Like any other numerical method, the FCT algorithm involves a certain degree of empiricism in the reconstruction of data so that the approximate solutions may exhibit various artefacts such as *clipping* and *terracing* [43]. The former phenomenon refers to a smearing of sharp peaks in the convected profile due to the fact that they cannot be properly resolved on the given mesh and their resurrection is prohibited by the flux limiter. Zalesak [54] managed to alleviate peak clipping by using the old solution u^n along with $\tilde{u}$ in the estimation of the solution bounds. However, this practice is not to be recommended if local extrema decay with time due to physical effects such as compression, expansion and sources/sinks. In this case, the use of information from the previous time step may result in an undershoot/overshoot [23],[25],[43].

Terracing manifests itself in a distortion of smooth profiles and represents 'an integrated, nonlinear effect of residual phase errors' [43] or, loosely speaking, 'the ghosts of departed ripples' [5]. In particular, this problem frequently occurs at outflow boundaries as illustrated in Fig. 5 (left) for the 1D convection equation (23) with $v = 1$ and initial data $u^0 = x$ in $\Omega = [0, 1]$. The linear function, for which flux correction is actually redundant and the standard Galerkin method would produce excellent results, degenerates into a broken line. The alternating steepening/flattening of the gradient leads to a formation of spurious plateaus. This indicates that FCT algorithms are not *linearity-preserving* and introduce too much antidiffusion in some cases.

Some preliminary speculations regarding the cause and cure of terracing at inflow and outflow boundaries can be found in [23],[29]. In particular, it turns out that a boundary value $\tilde{u}_i$ can be *misinterpreted* as a local extremum [41]. Indeed, it is only compared to the solution values at the neighboring nodes and no information about the solution behavior beyond the (artificial) open boundary is available. The erroneous cancellation of antidiffusive fluxes at the inlet/outlet entails a redistribution of mass in the interior of the domain and eventually leads to the formation of terraces. To illustrate this effect, a heuristic *lever model* was introduced in [41]. 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 Fig. 5 (right).

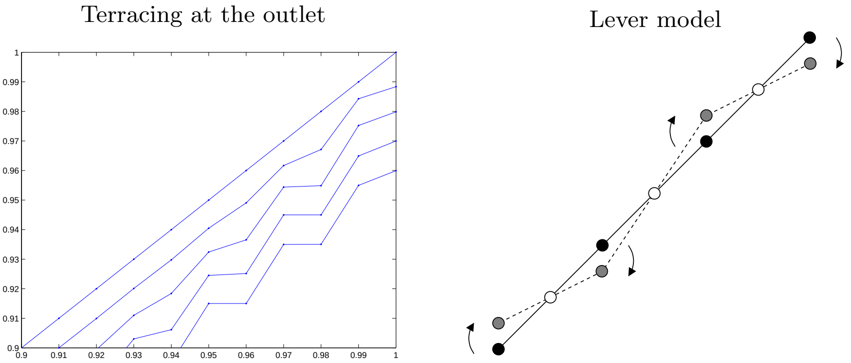


Fig. 5. Pathological behavior of FCT.

Interestingly enough, the ripples disappear if we set $R_i^\pm := 1$ for nodes belonging to the inflow and outflow boundaries (see above). At the inlet, this adjustment is admissible since the boundary values of the solution are fixed. At the outlet, it amounts to limiting the antidiffusive flux using the correction factor for the *upwind node* as in the case of TVD methods which are largely immune to terracing. The synchronization of nodal correction factors in (88) makes the FCT algorithm invariant to the flow direction and, therefore, vulnerable to incorrect upper/lower bounds for the node located downstream. Hence, the use of an upwind-biased limiting strategy is preferable in the vicinity of open boundaries and local extrema, where terracing is likely to occur as an aftermath of peak clipping. Due to conservation, the clipped mass must be distributed between the neighboring nodes and may easily go astray.

Unfortunately, a complete analysis of clipping and terracing is not available to date and it is not quite clear how to combat these artefacts. An important prerequisite seems to be the use of background diffusion [35]. It is hoped that a further investigation of Zalesak's limiter in the framework of algebraic flux correction and a detailed comparison of FCT with other high-resolution schemes will make it possible to find an effective remedy.

6.6 Summary of the FEM-FCT Algorithm

The new FEM-FCT methodology represents a generalization of the explicit algorithm proposed by Löhner et al. [36]. Obviously, it has a lot in common with the FEM-TVD approach introduced in the first part of this chapter. However, the construction and limiting of antidiffusive fluxes is performed *after* the discretization in time so that the implementation is slightly different. The basic steps and the corresponding parts of the code are as follows:

In the matrix assembly routine:

1. Retrieve the entries k_{ij} and k_{ji} of the high-order transport operator.
2. Determine the artificial diffusion coefficient d_{ij} from equation (20).
3. Update the four entries of the preconditioner A as required by (22).
4. Substitute the explicit diffusive flux into b^n (at the first iteration).
5. Evaluate/increment the raw antidiffusive flux given by (71) or (79).

In the flux correction module:

6. Initialize/update the auxiliary solution $\tilde{u}$ according to (75) or (78).
7. Use Zalesak's limiter to compute the correction factors from (88).
8. Insert the limited antidiffusive fluxes into (74) or (80)–(81).

In the defect correction loop:

9. Solve the linear system (73) or (59)–(60) with $r^{(m)}$ defined in (67).
10. Check convergence and proceed to the next iteration or time step.

As far as the time discretization is concerned, the second-order accurate Crank-Nicolson scheme is to be recommended for transient problems. In this case, the time step must remain relatively small to capture the evolution details and satisfy condition (76). Hence, the basic FEM-FCT algorithm is almost as accurate as the iterative one and certainly more efficient. On the other hand, the solution of steady-state problems by *pseudo-time-stepping* calls for the fully implicit treatment. The backward Euler method, which is only first-order accurate, is also appropriate if a nonuniform distribution of Courant numbers (e.g. due to mesh refinement or a strongly varying velocity field) makes the CFL condition too restrictive. Fully implicit FEM-FCT schemes are unconditionally positivity-preserving and the iterative formulation should be preferred to prevent loss of accuracy at large time steps.

7 Numerical Examples

The examples that follow illustrate the influence of the time discretization and of the flux limiter on the numerical results. An implicit time-stepping is employed in all cases. The performance of the flux-corrected Lax-Wendroff method (explicit, second-order accurate) was studied in [23], where many additional examples for the basic FEM-FCT algorithm can be found. Furthermore, only linear (triangular) and bilinear (quadrilateral) finite elements are considered in this chapter. Three-dimensional simulation results obtained using a (discontinuous) rotated bilinear approximation are presented in [28],[53].

The convergence of one-dimensional FCT and TVD schemes was investigated in [41] and [50], respectively. In particular, the effective order of accuracy p was estimated from the difference between the errors for the solutions computed on two sufficiently fine meshes. It can be shown that [12],[33]

$$p \approx \log_2(E_h/E_{2h}),$$

where E_h is the norm of the error on a uniform mesh with spacing h between the grid points. As reported in [41], the actual convergence rates depend on the discretization procedure and on smoothness of the exact solution. For details, the interested reader is referred to the original publications.

7.1 Solid Body Rotation

Rotation of solid bodies is frequently used to evaluate and compare numerical schemes for convection-dominated problems. A classical example is Zalesak's slotted cylinder test [54] which is intended to assess the ability of the method to cope with steep gradients and reproduce small-scale features. In order to examine the resolution of both smooth and discontinuous profiles, we consider an extended version of this 2D benchmark as proposed by LeVeque [33].

Let a slotted cylinder, a sharp cone and a smooth hump be exposed to the nonuniform velocity field $\mathbf{v} = (0.5-y, x-0.5)$ and undergo a counterclockwise rotation about the center of the unit square $\Omega = (0,1) \times (0,1)$. The initial configuration for this test is depicted in Fig. 6. Each solid body lies within a circle of radius $r_0 = 0.15$ centered at a point with Cartesian coordinates (x_0, y_0). In the rest of the domain, the solution is initialized by zero.

The shapes of the three bodies can be expressed in terms of the normalized distance function for the respective reference point (x_0, y_0)

$$r(x,y) = \frac{1}{r_0}\sqrt{(x-x_0)^2 + (y-y_0)^2}.$$

The center of the slotted cylinder is located at $(x_0, y_0) = (0.5, 0.75)$ and its geometry in the circular region $r(x,y) \leq 1$ is given by

$$u(x,y,0) = \begin{cases} 1 & \text{if } |x-x_0| \geq 0.025 \vee y \geq 0.85, \\ 0 & \text{otherwise.} \end{cases}$$

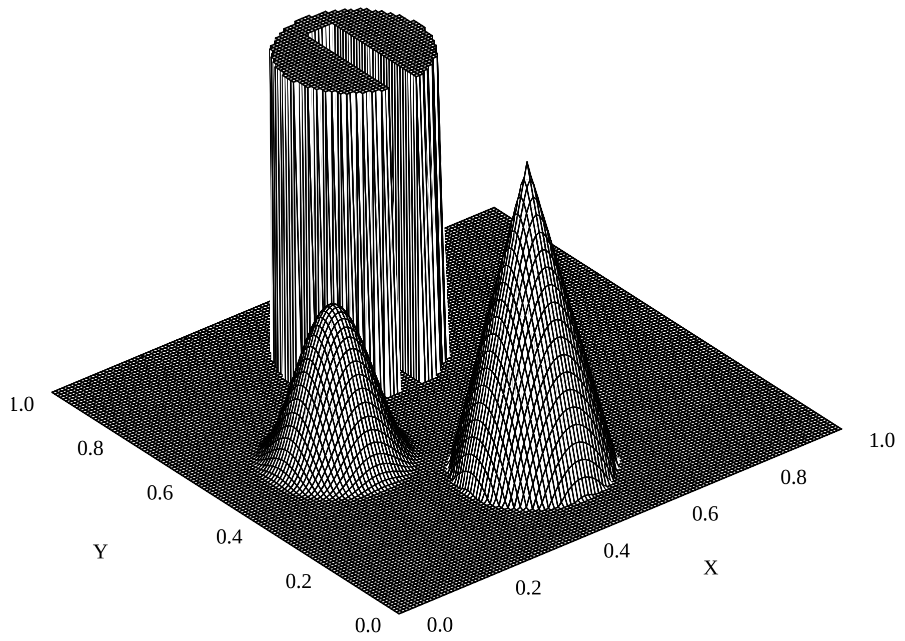


Fig. 6. Initial data and exact solution at $t = 2\pi$.

The corresponding analytical expression for the conical body reads

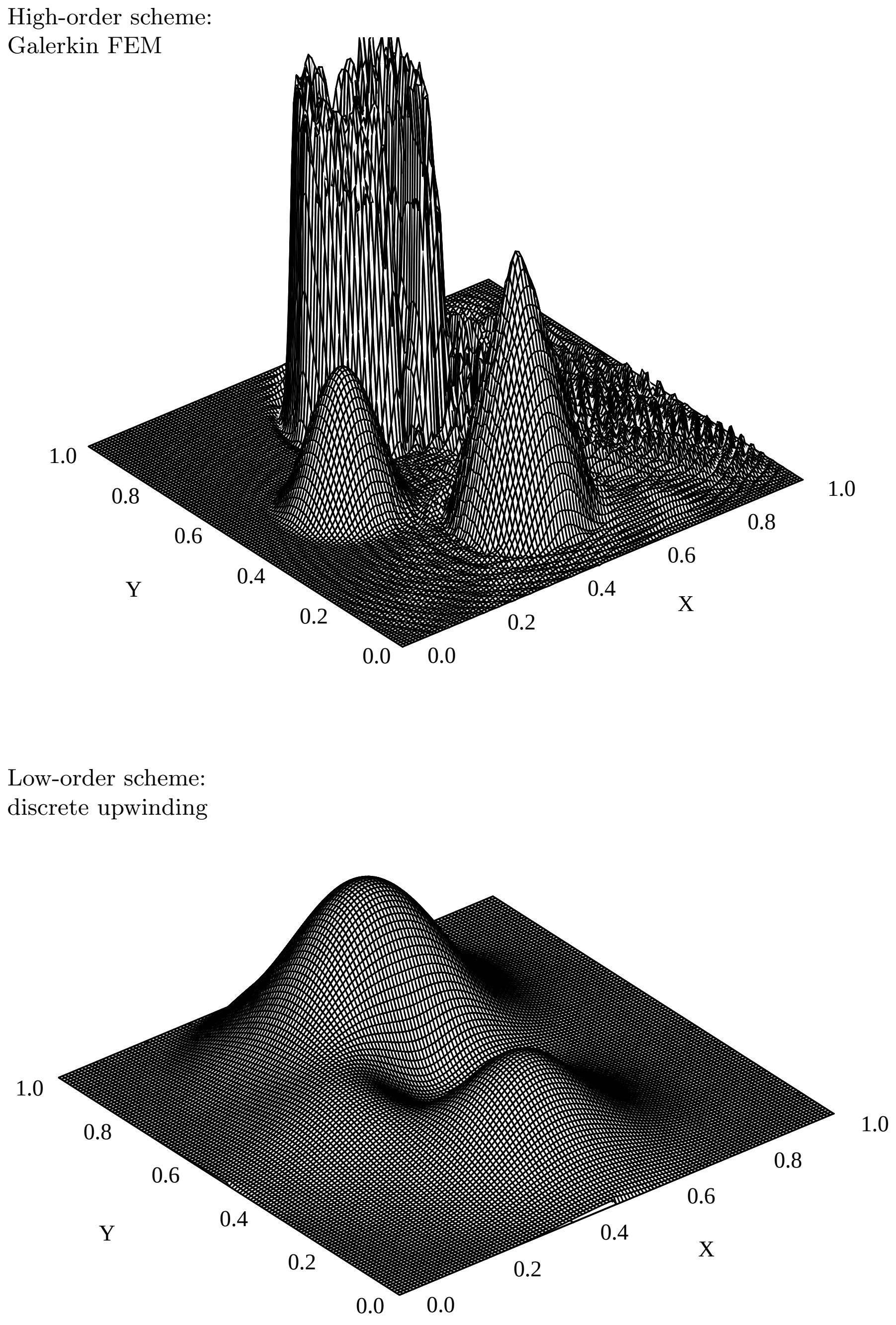
$$u(x, y, 0) = 1 - r(x, y), \qquad (x_0, y_0) = (0.5, 0.25),$$

whereas the shape and location of the hump at $t = 0$ are as follows

$$u(x, y, 0) = 0.25[1 + \cos(\pi \min\{r(x, y), 1\})], \quad (x_0, y_0) = (0.25, 0.5).$$

After one full revolution ($t = 2\pi$) the exact solution to the pure convection equation (1) coincides with the initial data. To expose the deficiencies of linear discretizations, we present the numerical results produced by the standard Galerkin method and its low-order counterpart (discrete upwinding) in Fig. 7. These solutions were computed on a mesh of 128×128 bilinear elements using the second-order accurate Crank-Nicolson time-stepping with $\Delta t = 10^{-3}$. Sure enough, the original high-order scheme reproduces the cone and hump very well but gives rise to spurious wiggles that can be traced to the slotted cylinder. On the other hand, the low-order solution is nonoscillatory but its quality is extremely poor due to the devastating effect of artificial diffusion.

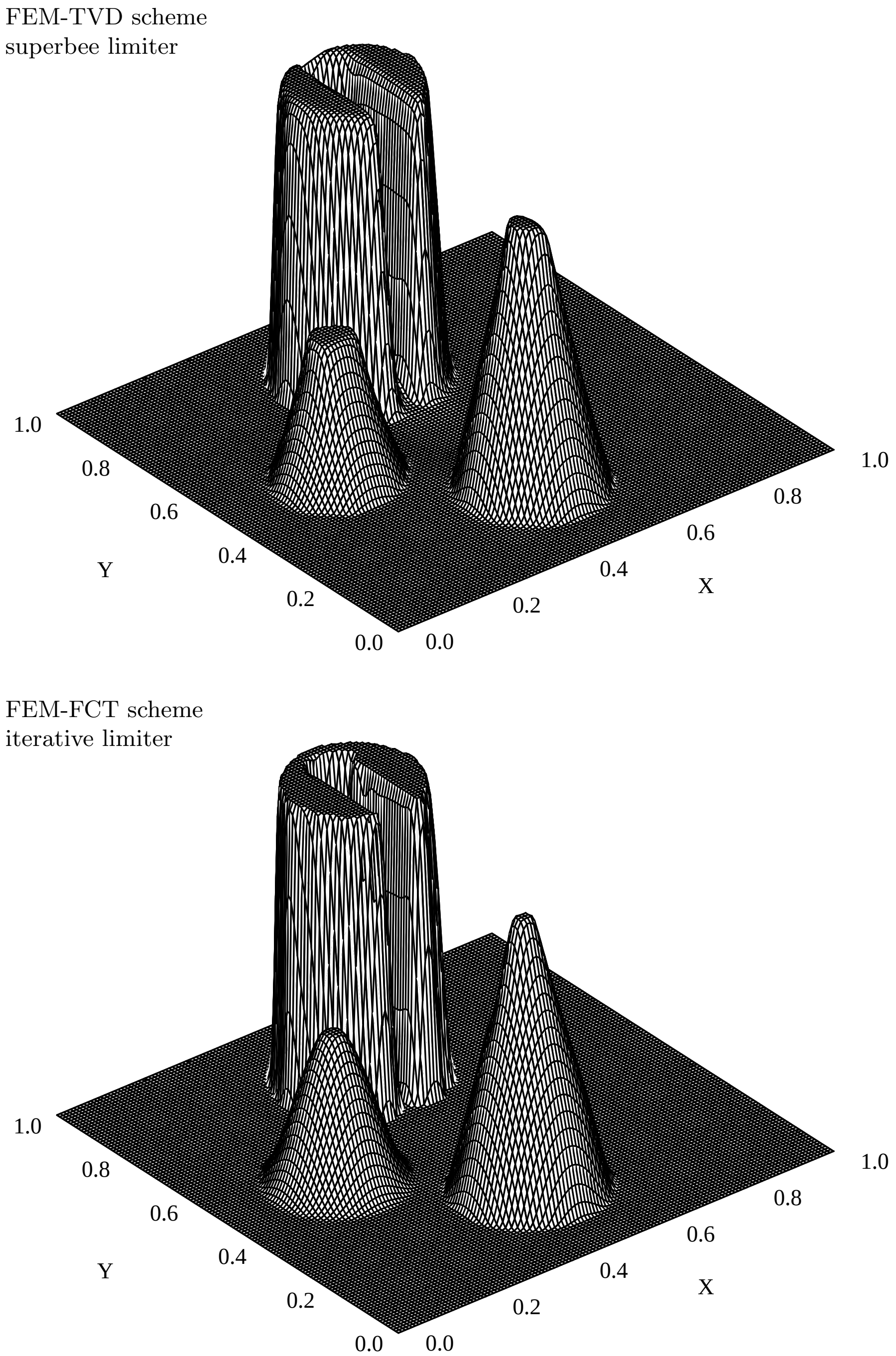
A nonlinear combination of these imperfect linear methods in the framework of algebraic flux correction yields the numerical solutions shown in Fig. 8. The nonphysical oscillations disappear and the resolution of the three bodies is still remarkably crisp. In the case of the FEM-TVD method, the narrow



High-order scheme:
Galerkin FEM

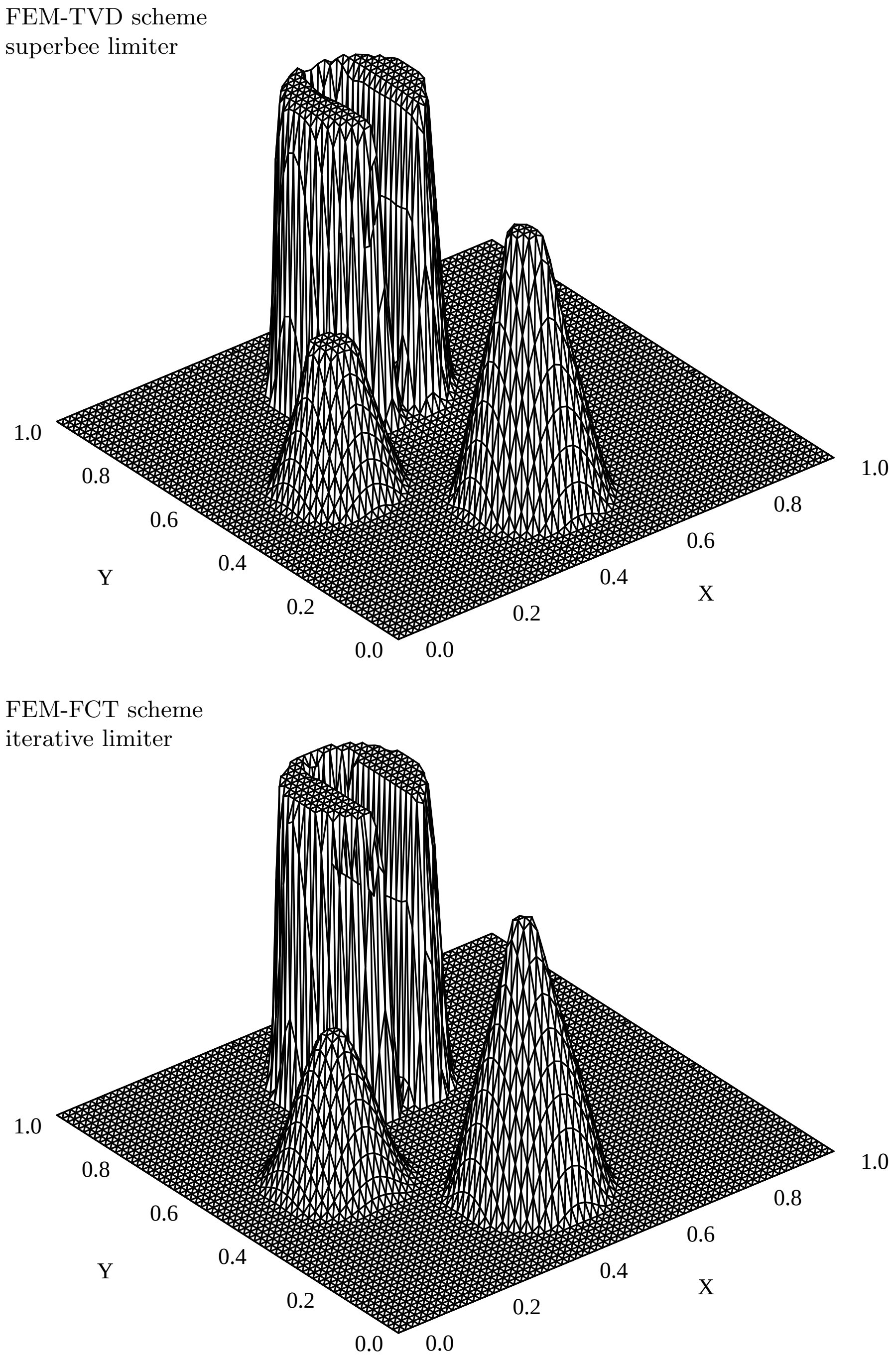
Low-order scheme:
discrete upwinding

Fig. 7. Solid body rotation: shortcomings of linear methods.


FEM-TVD scheme
superbee limiter

FEM-FCT scheme
iterative limiter

Fig. 8. Solid body rotation on a quadrilateral mesh.


FEM-TVD scheme
superbee limiter

FEM-FCT scheme
iterative limiter

Fig. 9. Solid body rotation on a triangular mesh.

bridge of the cylinder is largely preserved but some erosion of the ridges is observed. The overall accuracy is acceptable since the numerical diffusion due to mass lumping is alleviated to some extent by the strongly antidiffusive superbee limiter. At the same time, the excessive antidiffusion entails an artificial steepening of the gradients as well as a gradual flattening of the two peaks, which can be interpreted as a weak form of 'clipping' and 'terracing'. Other TVD limiters are more diffusive (see below) so that the lumping error is aggravated and a pronounced smearing of the solution profiles ensues.

For this strongly time-dependent test problem, the iterative FEM-FCT algorithm performs much better, which can be attributed to the use of the consistent mass matrix. The prelimiting of antidiffusive fluxes was found to be essential. If this optional step is omitted, the ridges of the cylinder are corrupted by harmless but optically disturbing kinks. The (inevitable) peak clipping for the cone does not exceed 10% and the hump is reproduced almost exactly. Similar results are obtained using a piecewise-linear approximation on the mesh constructed from the former one by subdivision of each quadrilateral into two triangles. For visualization purposes, the solutions shown in Fig. 9 were output on a coarser mesh of 64 × 64 × 2 triangular elements.

7.2 Swirling Flow Problem

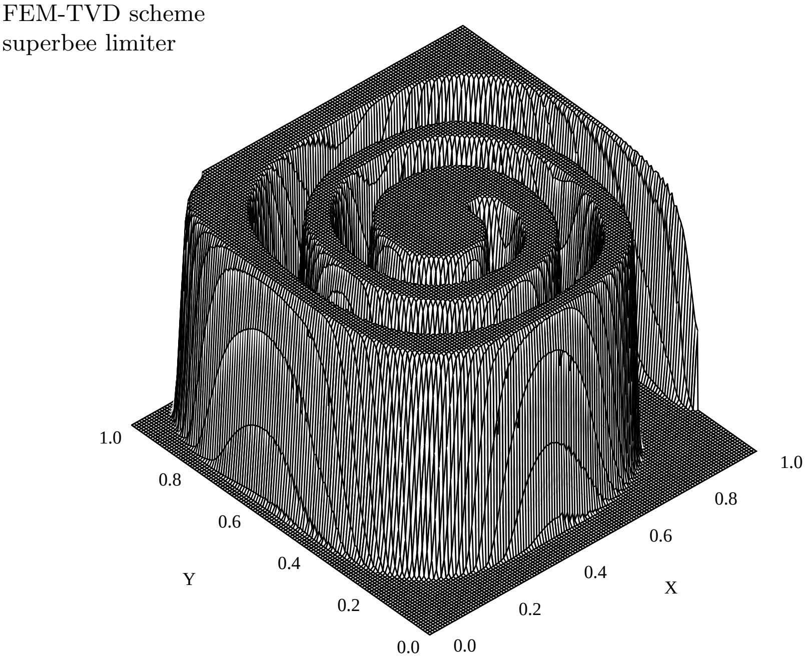
Another challenging test problem introduced by LeVeque [33] deals with a swirling deformation of the initial data by the incompressible velocity field shown in Fig. 12. The two velocity components are given by

$$v_x = \sin^2(\pi x)\sin(2\pi y), \quad v_y = -\sin^2(\pi y)\sin(2\pi x).$$

The initial condition is a discontinuous function which equals unity within a circular sector of $\pi/2$ radians and zero elsewhere:

$$u(x,y,0) = \begin{cases} 1 & \text{if } (x-1)^2 + (y-1)^2 < 0.64, \\ 0 & \text{otherwise.} \end{cases}$$

The computational results obtained at time $t = 2.5$ using the same parameter settings as in the previous example are shown in Fig. 10–11. In the course of deformation, the mass distribution assumes a complex spiral shape which is nicely resolved by both algebraic flux correction schemes under consideration. The imposed constraints are satisfied and the approximate solutions remain bounded by zero and one regardless of the mesh type. Again, the better accuracy of FEM-FCT as compared to FEM-TVD is due to the dynamic nature of the problem at hand. By construction, the latter algorithm is independent of the time discretization and lends itself to the treatment of steady-state applications. As already mentioned, the optimal degree of implicitness is also problem-dependent. In this example, the Crank-Nicolson time-stepping was selected, since Δt must be chosen impractically small for comparable results to be produced by the fully implicit backward Euler method [23],[25].



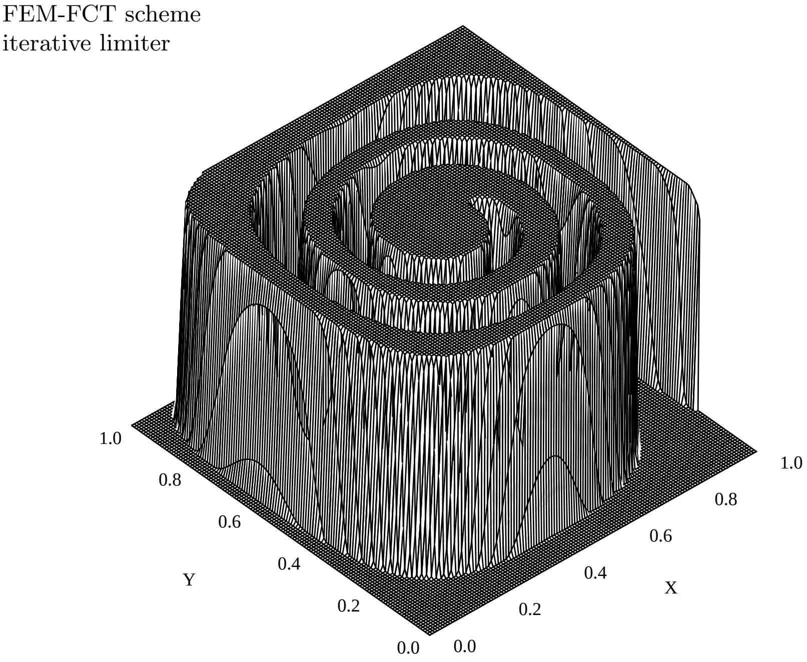
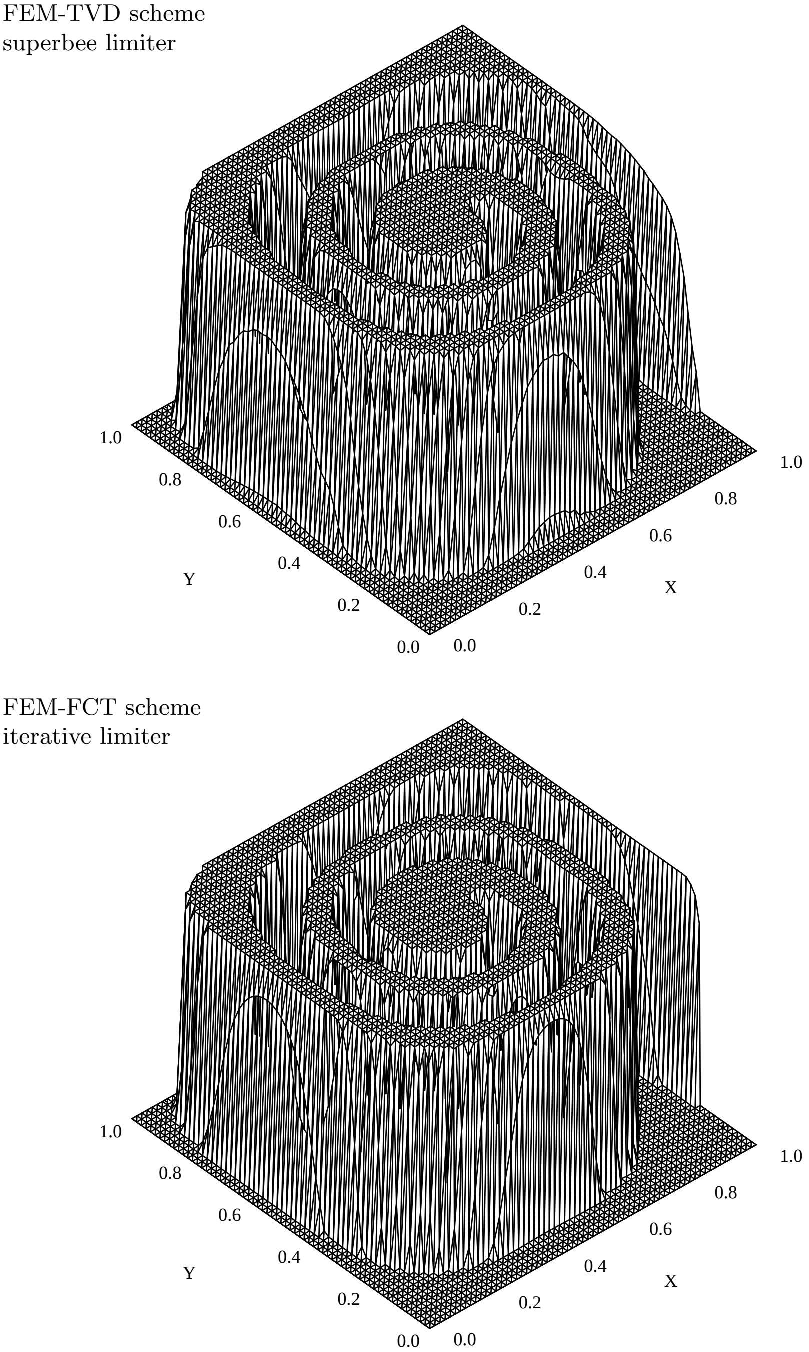


Fig. 10. Swirling deformation on a quadrilateral mesh.


FEM-TVD scheme
superbee limiter

FEM-FCT scheme
iterative limiter

Fig. 11. Swirling deformation on a triangular mesh.

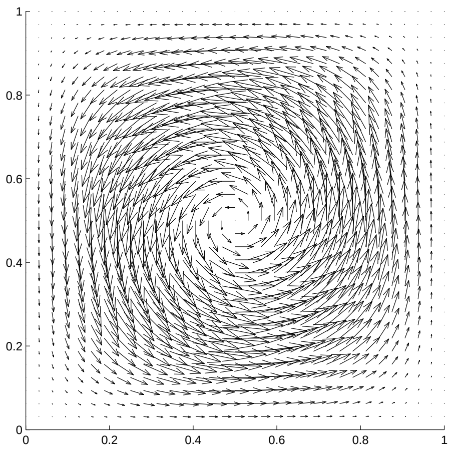


Fig. 12. Velocity field for the swirling flow.

7.3 Rotation of a Gaussian Hill

The third test problem devised by Lapin [30] makes it possible to assess the magnitude of artificial diffusion due to the discretization in space and time. This can be accomplished via statistical analysis of numerical solutions to the nonstationary convection-diffusion equation

$$\frac{\partial u}{\partial t} + \mathbf{v} \cdot \nabla u = \epsilon \Delta u \qquad \text{in } \Omega = (-1,1) \times (-1,1), \tag{92}$$

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

The initial condition to be imposed is given by $u(x,y,0) = \delta(x_0, y_0)$, where δ is the Dirac delta function. In a practical implementation, it is impossible to initialize the solution by a singular function. Instead, the whole mass should be concentrated at a single node of the computational mesh. The integral of the discrete solution over the domain Ω equals the sum of nodal values multiplied by the diagonal entries of the lumped mass matrix

$$\int_\Omega u_h \, \mathrm{d}\mathbf{x} = \int_\Omega \sum_j u_j \varphi_j \, \mathrm{d}\mathbf{x} = \sum_i m_i u_i.$$

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. The solution is a rotating 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 peaks of the approximate solution may certainly deviate from this destination. Their actual position can be calculated as the mathematical expectation of the center of mass, whereby the probability density u_h is obtained by solving equation (92) by the numerical scheme to be evaluated

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

The quality of approximation depends on the standard deviation

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

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

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

This statistical quantity provides an excellent estimate of numerical diffusion inherent to the finite element scheme under consideration.

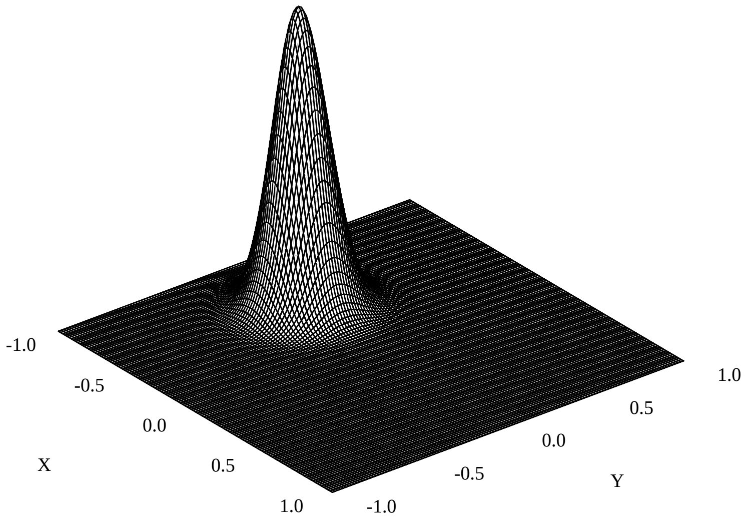
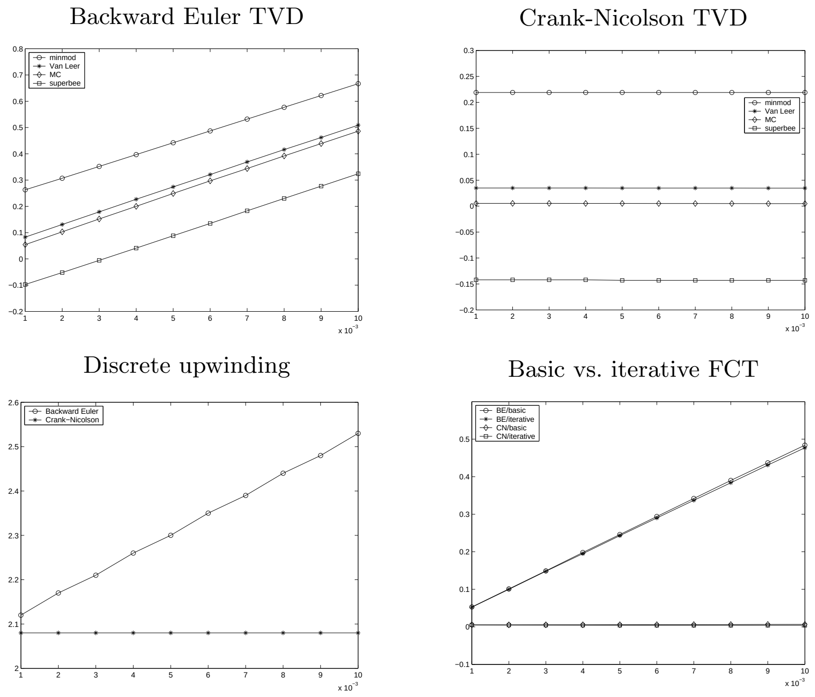


Fig. 13. Snapshot of the Gaussian hill at $t = 2.5\,\pi$.

Let us start with the analytical solution corresponding to $x_0 = 0$, $y_0 = 0.5$, and $t_0 = 0.5\pi$. As the Gaussian hill moves around the origin, it is being gradually smeared by diffusion and the height of the peak decreases accordingly. A snapshot taken after one full revolution ($t = 2.5\pi$) is displayed in Fig. 13. In the 'picture norm', the numerical results produced by our FEM-TVD and FEM-FCT schemes on a Cartesian mesh (using the same number of elements and time step as before) are virtually indistinguisable from the exact solution. However, a detailed analysis reveals that the value of the global maximum differs from case to case and so does the numerical variance σ_h.

The knowledge of the exact variance σ enables us to assess the total amount of numerical diffusion for different limiting techniques and to estimate the share of the temporal error. To this end, the values of $\Delta\sigma_{\mathrm{rel}}$ are plotted versus the time step in Fig. 14. If the first-order accurate backward Euler method is employed, the temporal part of the relative variance error dominates at large time steps and decreases linearly as Δt is refined. For the second-order accurate Crank-Nicolson scheme, the magnitude of the (anti-)diffusive error is largely invariant to the time step. This is why the corresponding lines are almost horizontal. In this case, the accuracy of the space discretization is the critical factor so that the choice of the flux limiter plays a key role.



Backward Euler TVD

Crank-Nicolson TVD

Discrete upwinding

Basic vs. iterative FCT

Fig. 14. Gaussian hill: relative variance error vs. the time step.

By far the most diffusive solutions are produced by discrete upwinding, whereas the use of algebraic flux correction leads to a dramatic improvement. However, a comparison of the relative variance errors for standard TVD limiters reveals considerable differences in their performance. The most diffusive limiter is minmod followed by Van Leer. The MC limiter outperforms both of them and is more suitable for the treatment of smooth profiles than Roe's superbee limiter. The latter turns out to be slightly underdiffusive so that $\Delta\sigma_{\mathrm{rel}}$ is negative if the spatial discretization error prevails. Although all four limiters qualify for CFD simulations, the 'right' one may be as difficult to select as the free parameter for classical artificial viscosity methods.

As expected, the iterative FCT algorithm is able to accommodate more antidiffusion than the basic limiter but the gain of accuracy is marginal in the range of time steps considered in this example. Note that the curves for the two versions almost meet at $\Delta t = 10^{-2}$ but begin to separate as Δt approaches 10^{-3}. This demonstrates that the Courant number must be 'large' for significant advantages to accrue from the iterative strategy. Its potential can be utilized to the full extent only if the time derivative is relatively small and the temporal accuracy can be sacrificed in favor of efficiency.

7.4 Stationary Convection-Diffusion

Algebraic flux correction schemes can be applied to stationary problems directly (TVD only) or in conjunction with a pseudo-time-stepping technique. In the latter case, the steady-state solution is obtained by marching into the stationary limit of the associated time-dependent problem, whereby the evolution details are immaterial. In essence, the time step represents an artificial parameter for the iterative solver and should be chosen as large as possible to reduce the computational cost. The CFL condition prevents explicit schemes from operating with large time steps, which makes them rather inefficient for such applications. This can be rectified to some extent by resorting to local time-stepping but an implicit time discretization is preferable.

In light of the above, the backward Euler method, which is not to be recommended for transient problems, lends itself to the treatment of steady and creeping flows. Let us apply the fully implicit FCT and TVD schemes to the stationary convection-diffusion equation

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

where $\mathbf{v} = (\cos 10^{\mathrm{o}}, \sin 10^{\mathrm{o}})$ is the constant velocity and $\epsilon = 10^{-3}$ is the diffusion coefficient. The concomitant boundary conditions read

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

$$u(x,0) = 0, \qquad u(1,y) = 0.$$

The solution to this singularly perturbed 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 guess for the desired stationary solution is given by

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

It is worthwhile to start with discrete upwinding and use the converged low-order solution as initial data for the nonlinear high-resolution scheme. Even this crude approximation provides a good starting point, so that the extra cost due to the assembly and limiting of antidiffusive fluxes is insignificant.

The numerical solutions depicted in Fig. 15 show that the backward Euler FEM-TVD method is capable of producing nonoscillatory solutions with a sharp resolution of steep fronts and boundary layers. The upper diagram was computed on a grid of 64×64 bilinear elements using the MC limiter. The mesh employed for the lower one consists of as few as 480 elements and is refined in regions where the solution gradients are large. Due to the mesh refinement and a special alignment of the grid lines, the accuracy is comparable to that achieved on the uniform mesh at a much higher computational cost.

The results produced by the FEM-FCT algorithm are presented in Fig. 16. The time step $\Delta t = 1.0$ (Courant number $\nu = 64$) was intentionally chosen to be very large so as to expose the differences between the basic and iterative versions. The former (see the upper diagram) exhibits excessive smearing in the vicinity of the boundary layer, which compromises the benefits offered by the unconditionaly stable backward Euler time-stepping. The lower diagram demonstrates that iterative flux correction is free of this drawback.

7.5 Convection in Space-Time

Our last example deals with the pure convection equation (23) discretized by central differences in conjunction with the leapfrog time-stepping

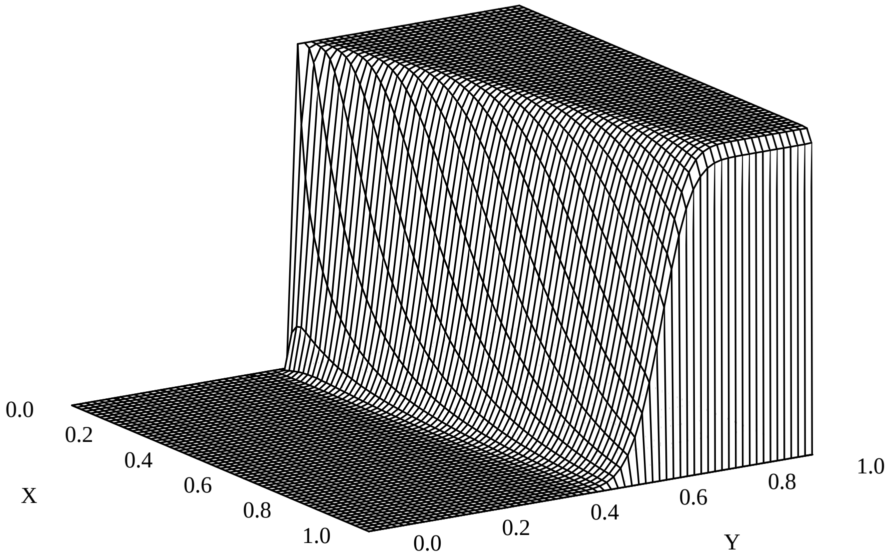
$$\frac{u_i^{n+1} - u_i^{n-1}}{2\Delta t} + v\frac{u_{i+1}^n - u_{i+1}^n}{2\Delta x} = 0.$$

At the boundaries of the space-time domain $\Omega = (0, 1) \times (0, 0.5)$ we switch to a (first-order accurate) one-sided finite difference approximation.

Instead of advancing the numerical solution in time step-by-step as usual, let us write all equations in the matrix form $Ku = 0$ and apply our algebraic TVD method to this linear high-order system. In essence, equation (23) is treated as its two-dimensional counterpart (1) with $\mathbf{x} = (x, t)$ and $\mathbf{v} = (1, 1)$. The following initial/boundary conditions are imposed at the 'inlet'

$$u(0, t) = 0, \quad u(x, 0) = \begin{cases} 1 & \text{if } (0.1 \leq x \leq 0.2) \vee (0.3 \leq x \leq 0.4), \\ 0 & \text{otherwise.} \end{cases}$$

64 × 64 Q_1 elements



20 × 24 Q_1 elements

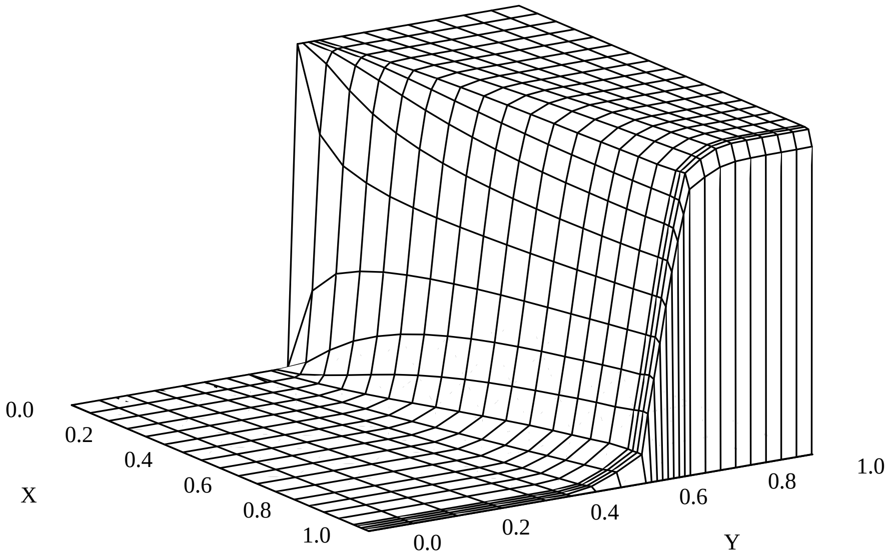
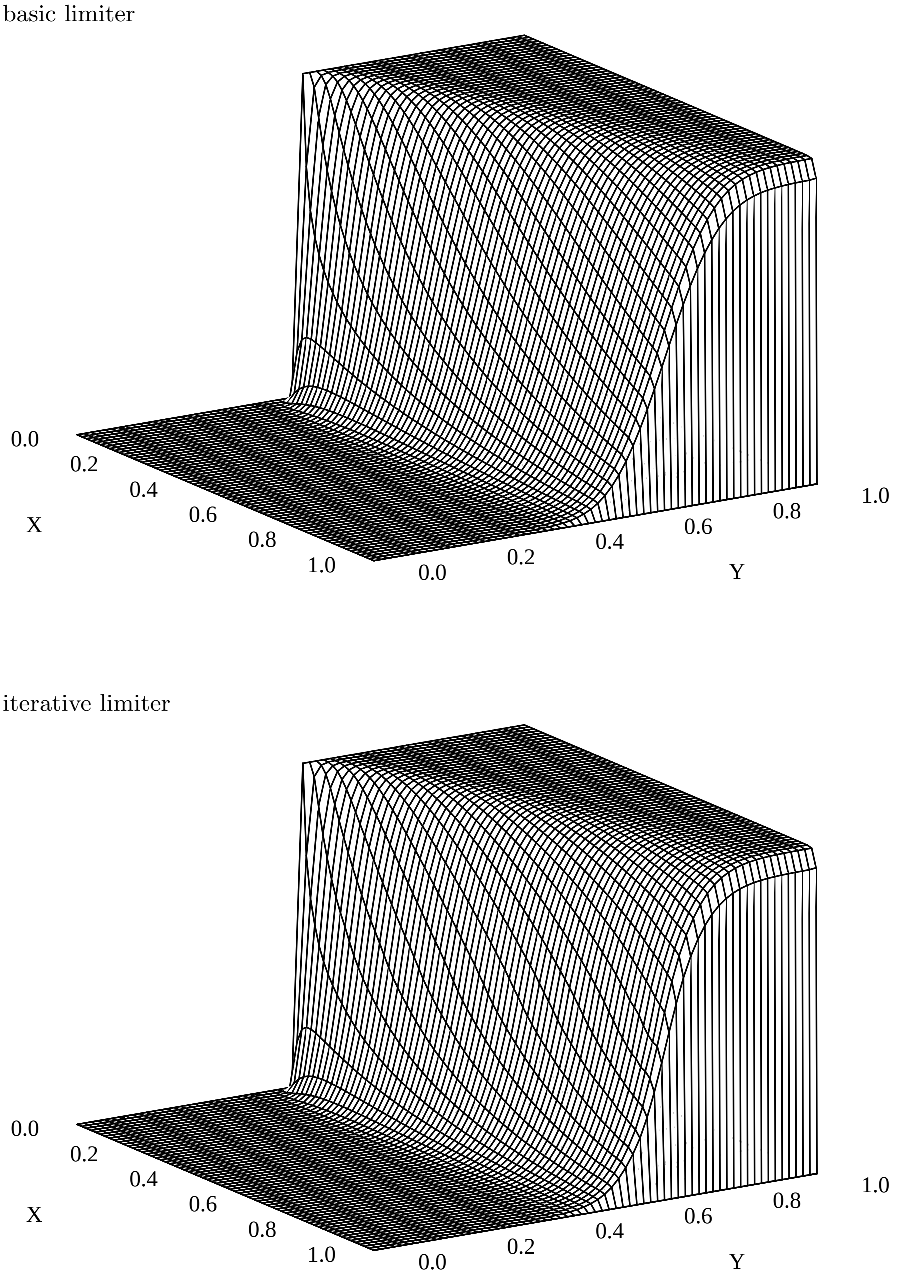


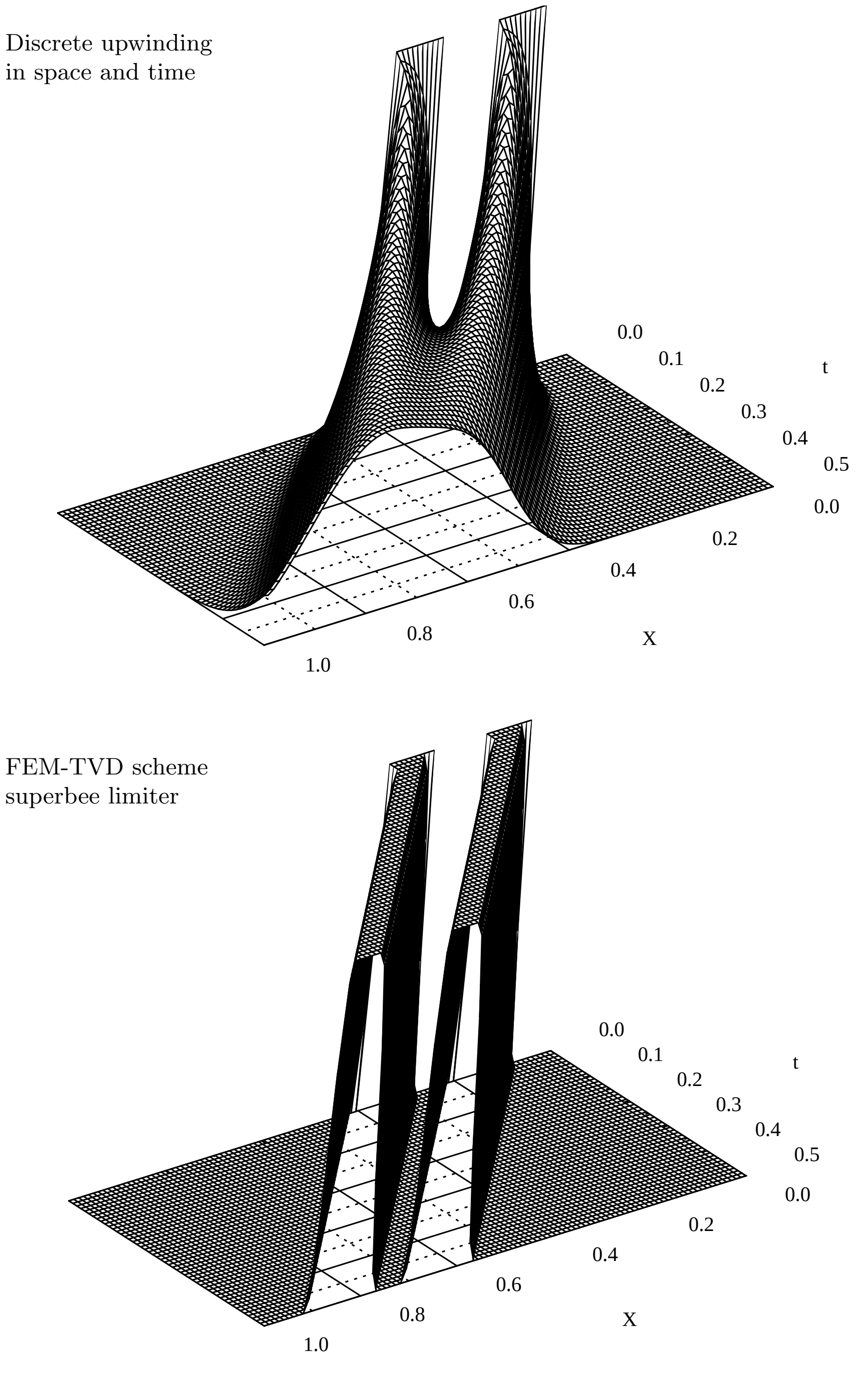
Fig. 15. Stationary solutions produced by the FEM-TVD scheme.



basic limiter

iterative limiter

Fig. 16. Stationary solutions produced by the FEM-FCT schemes.



Discrete upwinding in space and time

FEM-TVD scheme superbee limiter

Fig. 17. Convection in the space-time domain $\Omega = (0, 1) \times (0, 0.5)$.

Discrete upwinding applied to the matrix K yields the linear low-order method which corresponds to the upwind difference approximation of spatial derivatives and the backward Euler time-stepping. The computational results obtained on a uniform space-time mesh with $\Delta x = \Delta t = 10^{-2}$ are displayed in Fig. 17 (top). The perspective chosen for visualization is such that the solution of equation (23) at time $t = 0.5$ appears in the foreground. Ideally, it should be a copy of the discontinuous initial data shifted by $vt = 0.5$ along the $x-$axis. However, it can be seen that numerical diffusion rapidly destroys the two rectangular pulses and fills the narrow gap between them.

Using this overly diffusive solution as initial guess for the iterative defect correction scheme, we add compensating antidiffision controlled by the superbee limiter. The resulting FEM-TVD solution is shown in Fig. 17 (bottom). It preserves the initial profiles very well and is devoid of spurious oscillations inspite of the fact that the Courant number $\nu = v\Delta t/\Delta x = 1$ for this simulation. Thus, it is possible to circumvent the restrictive CFL-like condition (18) and construct nonlinear space-time discretizations of high order.

8 Conclusions and Outlook

A new class of high-resolution finite element schemes was presented. Node-oriented flux limiters of FCT and TVD type were applied at the algebraic level so as to render the underlying Galerkin discretization local extremum diminishing and positivity-preserving. The proposed methodology is very flexible and can be readily integrated into existing CFD software as a modular extension to the matrix assembly routine. Remarkably, algebraic flux correction is applicable to arbitrary discretizations in space and time (explicit and implicit time-stepping, finite elements/differences/volumes, Cartesian and unstructured meshes) and portable to higher dimensions. In fact, the same 'post-processing' routine can be used in 1D, 2D, and 3D implementations.

The generality of our algebraic approach makes it a valuable design tool and suggests many directions for further research. In particular, an extension to higher-order finite elements is feasible but nontrivial. For superlinear approximations, even the mass matrix and the discrete Laplacian operator may have negative off-diagonal coefficients, and it is unclear whether or not they should be eliminated in the course of discrete upwinding. It might be worthwhile to introduce some stabilizing background diffusion so as to reduce phase errors and alleviate terracing. For instance, the fourth-order accurate CNTG scheme [11], which represents a generalization of the method used in the 'reversible' FCT algorithm [5], would be a good candidate for the linear high-order scheme. Furthermore, it would be interesting to investigate how algebraic flux correction performs in the realm of discontinuous Galerkin methods. Last but not least, there is a need for the development of robust and efficient iterative solvers for nonsymmetric algebraic systems that result from an implicit LED discretization of the troublesome convective terms.

A. Galerkin Flux Decomposition

In this appendix, we present the derivation of numerical fluxes for the finite element discretization of the generic conservation law

$$\frac{\partial u}{\partial t} + \nabla \cdot \mathbf{f} = 0 \qquad \text{in } \Omega.$$

The flux function $\mathbf{f}$ may depend on the solution u in a nonlinear way.

Using the divergence theorem to perform integration by parts in the weak form of this equation, we obtain the integral relation

$$\int_\Omega w \frac{\partial u}{\partial t}\,\mathrm{d}\mathbf{x} - \int_\Omega \nabla w \cdot \mathbf{f}\,\mathrm{d}\mathbf{x} + \int_\Gamma w\ \mathbf{f} \cdot \mathbf{n}\,\mathrm{d}s = 0, \qquad \forall w.$$

The approximate solution to this variational problem is sought in the form

$$u_h(\mathbf{x}, t) = \sum_j u_j(t)\varphi_j(\mathbf{x}).$$

Consider the group finite element formulation [13] which consists in using the same interpolation for the flux function

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

Rendering the residual orthogonal to each basis function φ_i of the *ansatz* space gives a system of semi-discretized equations for the nodal values

$$\sum_j \left[\int_\Omega \varphi_i\varphi_j\,\mathrm{d}\mathbf{x}\right]\frac{\mathrm{d}u_j}{\mathrm{d}t} - \sum_j \left[\int_\Omega \nabla\varphi_i\varphi_j\,\mathrm{d}\mathbf{x} - \int_\Gamma \varphi_i\varphi_j\,\mathbf{n}\,\mathrm{d}s\right]\cdot \mathbf{f}_j = 0.$$

The integrals containing the product of basis functions represent entries of the mass matrices for the volume and surface triangulation

$$m_{ij} = \int_\Omega \varphi_i\varphi_j\,\mathrm{d}\mathbf{x}, \qquad \mathbf{s}_{ij} = \int_\Gamma \varphi_i\varphi_j\,\mathbf{n}\,\mathrm{d}s.$$

Furthermore, the volume integrals that result from the discretization of spatial derivatives can be written in the notation of section 2 as follows

$$\left[\int_\Omega \nabla\varphi_i\varphi_j\,\mathrm{d}\mathbf{x}\right]\cdot \mathbf{f}_j = \mathbf{c}_{ji}\cdot\mathbf{f}_j, \qquad \mathbf{c}_{ij} = \int_\Omega \varphi_i\nabla\varphi_j\,\mathrm{d}\mathbf{x}.$$

Recall that the coefficient matrix $\{\mathbf{c}_{ij}\}$ has zero row sums, which enables us to express its diagonal entries in terms of the off-diagonal ones

$$\sum_j \mathbf{c}_{ij} = 0 \quad \Rightarrow \quad \mathbf{c}_{ii} = -\sum_{j\neq i}\mathbf{c}_{ij}.$$

It follows that the ODEs at hand can be cast into the conservation form

$$\sum_j \left[m_{ij} \frac{\mathrm{d}u_j}{\mathrm{d}t} + \mathbf{s}_{ij} \cdot \mathbf{f}_j \right] + \sum_{j \neq i} g_{ij} = 0,$$

where the interior part of the discretized divergence term is assembled from the *Galerkin fluxes* associated with edges of the sparsity graph

$$g_{ij} = \mathbf{c}_{ij} \cdot \mathbf{f}_i - \mathbf{c}_{ji} \cdot \mathbf{f}_j, \qquad g_{ji} = -g_{ij}.$$

These antisymmetric fluxes are responsible for the bilateral mass exchange between two neighboring nodes, whereby no mass is created or destroyed artificially in the interior of the domain. The total amount of u may only change due to the external feed $\mathbf{s}_{ij} \cdot \mathbf{f}_j$ which is equal to zero unless the basis functions for both nodes are nonvanishing on the boundary.

The above flux decomposition makes it possible to extend the wealth of upwinding techniques and slope limiters available for the data structure of Peraire *et al.* [45] to arbitrary Galerkin discretizations. The interested reader is referred to [34],[38],[40] for the mathematical and algorithmic background of such high-resolution finite element schemes that were originally developed for piecewise-linear approximations on triangular meshes.

References

1. P. Arminjon and A. Dervieux, Construction of TVD-like artificial viscosities on 2-dimensional arbitrary FEM grids. *INRIA Research Report* **1111** (1989).
2. K. Baba and M. Tabata, On a conservative upwind finite element scheme for convective diffusion equations. *RAIRO Numerical Analysis* **15** (1981) 3–25.
3. T. J. Barth, Numerical aspects of computing viscous high Reynolds number flows on unstructured meshes. Technical report 91-0721, *AIAA paper*, 1991.
4. T. J. Barth, Aspects of unstructured grids and finite volume solvers for the Euler and Navier-Stokes equations. In *von Karman Institute for Fluid Dynamics Lecture Series* Notes 1994-05, Brussels, 1994.
5. D. L. Book, The Conception, Gestation, Birth, and Infancy of FCT. In this volume.
6. 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.
7. A. N. Brooks and T. J. R. Hughes, Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. Engrg.* **32** (1982) 199-259.
8. G. F. Carey and B. N. Jiang, Least-squares finite elements for first-order hyperbolic systems. *Int. J. Numer. Meth. Fluids* **26** (1988) 81–93.
9. C. R. DeVore, An improved limiter for multidimensional flux-corrected transport. *NASA Technical Report* AD-A360122 (1998).
10. J. Donea, L. Quartapelle and V. Selmin, An analysis of time discretization in the finite element solution of hyperbolic problems. *J. Comput. Phys.* **70** (1987) 463–499.
11. 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).
12. J. H. Ferziger and M. Peric, *Computational Methods for Fluid Dynamics.* Springer, 1996.
13. C. A. J. Fletcher, The group finite element formulation. *Comput. Methods Appl. Mech. Engrg.* **37** (1983) 225-243.
14. S. K. Godunov, Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics. *Mat. Sbornik* **47** (1959) 271-306.
15. P. Hansbo, Aspects of conservation in finite element flow computations. *Comput. Methods Appl. Mech. Engrg.* **117** (1994) 423-437.
16. A. Harten, High resolution schemes for hyperbolic conservation laws. *J. Comput. Phys.* **49** (1983) 357–393.
17. A. Harten, On a class of high resolution total-variation-stable finite-difference-schemes. *SIAM J. Numer. Anal.* **21** (1984) 1-23.
18. C. Hirsch, *Numerical Computation of Internal and External Flows. Vol. II: Computational Methods for Inviscid and Viscous Flows.* John Wiley & Sons, Chichester, 1990.
19. 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. *Int. Journal of CFD* **4** (1995) 171-218.
20. A. Jameson, Computational algorithms for aerodynamic analysis and design. *Appl. Numer. Math.* **13** (1993) 383-422.

21. A. Jameson, Positive schemes and shock modelling for compressible flows. *Int. J. Numer. Meth. Fluids* **20** (1995) 743–776.
22. D. Kuzmin, Positive finite element schemes based on the flux-corrected transport procedure. In: K. J. Bathe (ed.), *Computational Fluid and Solid Mechanics*, Elsevier, 887-888 (2001).
23. D. Kuzmin and S. Turek, Flux correction tools for finite elements. *J. Comput. Phys.* **175** (2002) 525-558.
24. 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 Italy, 2002, 133-143.
25. D. Kuzmin, M. Möller and S. Turek, Multidimensional FEM-FCT schemes for arbitrary time-stepping. *Int. J. Numer. Meth. Fluids* **42** (2003) 265-295.
26. D. Kuzmin, M. Möller and S. Turek, Implicit flux-corrected transport algorithm for finite element simulation of the compressible Euler equations. To appear in: M. Krizek *et al.* (eds), *Finite Element Methods: Fifty Years of Conjugate Gradients*, Proc. Conf., Univ. of Jyväskylä, 2002. Springer, Berlin, 2004.
27. D. Kuzmin and S. Turek, Finite element discretization tools for gas-liquid flows. In: M. Sommerfeld (ed.), *Bubbly Flows: Analysis, Modelling and Calculation*, Springer, 2004, 191-201.
28. D. Kuzmin and S. Turek, High-resolution FEM-TVD schemes based on a fully multidimensional flux limiter. Technical report **229**, University of Dortmund, 2003. To appear in *J. Comput. Phys.*
29. D. Kuzmin, M. Möller and S. Turek, High-resolution FEM-FCT schemes for multidimensional conservation laws. Technical report **231**, University of Dortmund, 2003. To appear in *Comput. Methods Appl. Mech. Engrg.*
30. A. Lapin, University of Stuttgart. Private communication.
31. P. D. Lax, *Systems of Conservation Laws and Mathematical Theory of Shock Waves*. SIAM Publications, Philadelphia, 1973.
32. R. J. LeVeque, *Numerical Methods for Conservation Laws*. Birkhäuser, 1992.
33. R. J. LeVeque, High-resolution conservative algorithms for advection in incompressible flow. *Siam J. Numer. Anal.* **33** (1996) 627–665.
34. R. Löhner, *Applied CFD Techniques: An Introduction Based on Finite Element Methods*. Wiley, 2001.
35. R. Löhner and J. D. Baum, 30 years of FCT: status and directions. In this volume.
36. 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.
37. 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.
38. P. R. M. Lyra, *Unstructured Grid Adaptive Algorithms for Fluid Dynamics and Heat Conduction*. PhD thesis, University of Wales, Swansea, 1994.
39. P. R. M. Lyra, K. Morgan, J. Peraire and J. Peiro, TVD algorithms for the solution of the compressible Euler equations on unstructured meshes. *Int. J. Numer. Meth. Fluids* **19** (1994) 827–847.
40. P. R. M. Lyra and K. Morgan, A review and comparative study of upwing biased schemes for compressible flow computation. III: Multidimensional extension on unstructured grids. *Arch. Comput. Methods Eng.* **9** (2002) no. 3, 207-256.

41. M. Möller, *Hochauflösende FEM-FCT-Verfahren zur Diskretisierung von konvektionsdominanten Transportproblemen mit Anwendung auf die kompressiblen Eulergleichungen.* Diploma thesis, University of Dortmund, 2003.
42. K. Morgan and J. Peraire, Unstructured grid finite element methods for fluid mechanics. *Reports on Progress in Physics*, **61** (1998), no. 6, 569-638.
43. E. S. Oran and J. P. Boris, *Numerical Simulation of Reactive Flow.* 2nd edition, Cambridge University Press, 2001.
44. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow.* McGraw-Hill, New York, 1980.
45. 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).
46. R. Rannacher and S. Turek, A simple nonconforming quadrilateral Stokes element. *Numer. Meth. PDEs* **8** (1992), no. 2, 97-111.
47. C. Schär and P. K. Smolarkiewicz, A synchronous and iterative flux-correction formalism for coupled transport equations. *J. Comput. Phys.* **128** (1996) 101–120.
48. V. Selmin, Finite element solution of hyperbolic equations. I. One-dimensional case. *INRIA Research Report* **655** (1987).
49. V. Selmin, Finite element solution of hyperbolic equations. II. Two-dimensional case. *INRIA Research Report* **708** (1987).
50. A. Sokolichin, *Mathematische Modellbildung und numerische Simulation von Gas-Flüssigkeits-Blasenströmungen.* Habilitation thesis, University of Stuttgart, 2003.
51. P. K. Sweby, High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM J. Numer. Anal.* **21** (1984), 995–1011.
52. S. Turek, *Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approach*, LNCSE $\underline{6}$, Springer, 1999.
53. S. Turek, *Algebraic Flux Correction III. Incompressible Flow Solvers.* In this volume.
54. S. T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids. *J. Comput. Phys.* **31** (1979) 335–362.