# Explicit and implicit FEM-FCT algorithms with flux linearization

Dmitri Kuzmin

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

## Abstract

A new approach to the design of flux-corrected transport (FCT) algorithms for continuous (linear/multilinear) finite element approximations of convection-dominated transport problems is pursued. The algebraic flux correction paradigm is revisited, and a family of nonlinear high-resolution schemes based on Zalesak's fully multidimensional flux limiter is considered. In order to reduce the cost of flux correction, the raw antidiffusive fluxes are linearized about an auxiliary solution computed by a high- or low-order scheme. By virtue of this linearization, the costly computation of solution-dependent correction factors is to be performed just once per time step, and there is no need for iterative defect correction if the governing equation is linear. A predictor-corrector algorithm is proposed as an alternative to the hybridization of high- and low-order fluxes. Three FEM-FCT schemes based on the Runge-Kutta, Crank-Nicolson, and backward Euler time-stepping are introduced, and a detailed comparative study is performed. Numerical results are presented for the linear convection equation and for the Euler equations of gas dynamics.

*Key words:* Computational Fluid Dynamics, convection-dominated problems, high-resolution schemes, flux-corrected transport, finite element method
*PACS:* 02.60.Cb, 02.70.Dh, 47.11.Fg

## 1 Introduction

Numerical solutions to convection-dominated flow problems are frequently corrupted by spurious oscillations or excessive numerical diffusion. The first scheme to produce satisfactory results even in the limit of pure convection was the *flux-corrected transport* (FCT) algorithm of Boris and Book [4,5].

*Email address:* kuzmin@math.uni-dortmund.de (Dmitri Kuzmin).

The basic idea behind the classical FCT methodology is as follows:

(1) Advance the solution in time by a low-order scheme that incorporates enough *numerical diffusion* to suppress undershoots and overshoots.
(2) Correct the solution using *antidiffusive fluxes* limited in such a way that no new maxima or minima can form and existing extrema cannot grow.

Predictor-corrector algorithms of this kind can be classified as diffusion-antidiffusion (DAD) methods [8]. The job of the numerical diffusion built into the low-order scheme is to enforce the positivity constraint and provide good phase accuracy. The limited antidiffusive correction is intended to reduce the amplitude errors in a local extremum diminishing manner.

A more general approach to the design of high-resolution schemes is based on blending (hybridization) of high- and low-order flux approximations. As a rule of thumb, the former is supposed to be used in regions of smoothness and the latter in the neighborhood of steep fronts. The weighting factor for the hybrid flux approximation is chosen so as to enforce physical or mathematical constraints related to some known properties of analytical solutions (positivity, monotonicity, nonincreasing total variation). A fully multidimensional FCT algorithm of this type was proposed by Zalesak [37] who constrained the positive and negative antidiffusive fluxes so as to control the net increment to the solution value at each grid point. We refer to [38] for a detailed presentation of the underlying design philosophy and further developments.

Following the advent of FCT in the 1970s, many other high-resolution schemes have been developed and backed by mathematical theory. Harten's *total variation diminishing* (TVD) methods [13] are also based on an algebraic hybridization of high- and low-order fluxes. Schemes like MUSCL, PPM, and ENO/WENO represent higher-order extensions of the Godunov method [11] that involve polynomial reconstruction from cell averages and slope limiting [3]. This geometric approach has become very popular in the context of finite difference, finite volume, and discontinuous Galerkin (DG) methods. However, slope-limited polynomial reconstruction has no natural counterpart in the realm of continuous (linear and multilinear) finite elements, whereas 'algebraic' flux correction of FCT or TVD type is still feasible [21].

The development of high-resolution FEM on the basis of FCT dates back to the explicit algorithms of Parrott and Christie [30] and Löhner *et al.* [25,26]. Several implicit FEM-FCT schemes were published by the author and his coworkers [17,19,20,29]. The rationale for the use of an implicit time discretization stems from the fact that the CFL stability condition becomes prohibitively restrictive in the case of strongly nonuniform velocity fields and/or locally refined meshes. Woodward and Colella ([35], p. 119) conclude that "adaptive grid schemes have a major drawback – they demand an implicit treatment of the flow equations.” This statement reflects a widespread prejudice that implicit schemes are computationally expensive. As a matter of fact, the cost of an implicit algorithm depends on the choice of iterative methods, parameter settings, and stopping criteria. If the time step is very small, then a good initial guess is available and 1-2 inner iterations might suffice. Thus, the cost *per time step* approaches that for a conditionally stable explicit algorithm. As the time step increases, so does the number of iterations, and more sophisticated linear algebra tools (smoothers, preconditioners) may need to be employed.

If the antidiffusive fluxes to be limited depend on the unknown solution then a nonlinear algebraic system has to be solved at every time step. Within the framework of an iterative defect correction scheme [20,21], the solution of this system is updated step-by-step, and the flux limiter is invoked at every outer iteration. Unfortunately, many flux/defect correction steps are usually required to obtain a fully converged solution, especially if the Courant number is large and the contribution of the consistent mass matrix cannot be neglected. The use of a discrete Newton method [29] makes it possible to accelerate convergence but the computational cost per time step is still rather high as compared to that of a fully explicit algorithm. This is unacceptable because the time step for FCT must be relatively small for accuracy reasons.

In order to reduce the cost of implicit flux correction, it is possible to compute the (unconstrained) high-order solution and use it to linearize the raw antidiffusive flux [19,25,29]. However, an implicit computation of the high-order predictor is expensive (or even impossible) due to the unfavorable properties of the discrete transport operator. In the present paper, we linearize the antidiffusive flux about the end-of-step solution computed by an explicit or implicit low-order scheme. This approach proves more efficient and simplifies the design of characteristic FCT schemes for nonlinear hyperbolic systems such as the Euler equations of gas dynamics. Furthermore, we apply limited antidiffusion to the low-order solution instead of modifying the algebraic system and solving it again. That is, we go back to the roots of FCT and adopt a predictor-corrector strategy of diffusion-antidiffusion type. Reportedly, such algorithms possess better phase accuracy than those based on hybridization [8].

Benchmark problems from [24,33,35] will be used to evaluate the performance of explicit and implicit FEM-FCT schemes with flux linearization about the low-order predictor. In the explicit case, the use of a TVD Runge-Kutta time-stepping method [12] insures positivity preservation, whereas no such proofs are available for the classical FEM-FCT algorithm [25,26] based on a two-step Taylor-Galerkin method. The numerical study to be presented demonstrates that the linearized Crank-Nicolson and backward Euler FCT schemes are $2-30$ times faster than their nonlinear counterparts. Moreover, the computational cost per time step is comparable to that for Runge-Kutta FCT, although no attempt was made to optimize the parameter settings for the iterative solver.

## 2 Design criteria

As a model problem, we consider the continuity equation for a scalar quantity $u(\mathbf{x},t)$ convected by the velocity field $\mathbf{v}(\mathbf{x},t)$ in a domain $\Omega$ with boundary $\Gamma$

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

Since this equation is hyperbolic, boundary conditions are required only on the inflow part $\Gamma_- \subset \Gamma$, where the velocity vector $\mathbf{v}$ is pointing into $\Omega$

$$u(\mathbf{x}) = g(\mathbf{x}), \quad \forall \mathbf{x} \in \Gamma_-. \tag{2}$$

The initial condition for the problem to be solved is given as a function of $\mathbf{x}$

$$u(\mathbf{x},0) = u_0(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega. \tag{3}$$

In many cases, the exact solution is known to be *positivity-preserving*, i.e.,

$$u_0(\mathbf{x}) \geq 0,\ \forall \mathbf{x} \in \Omega \quad \Rightarrow \quad u(\mathbf{x},t) \geq 0,\ \forall \mathbf{x} \in \Omega,\ \forall t > 0. \tag{4}$$

This property guarantees that densities, temperatures, concentrations, and the like remain nonnegative. A good numerical scheme should also preserve the sign of the initial data if this constraint is dictated by the physics.

### *2.1 Space discretization*

After the discretization in space by a finite difference, finite volume or finite element 'method of lines', the solution $u(\mathbf{x},t)$ is approximated by a set of time-dependent nodal values $u_i(t)$ that correspond to pointwise values, cell averages or coefficients of polynomial basis functions, respectively. The vector of unknowns satisfies a system of differential-algebraic equations

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

where $M_C = \{m_{ij}\}$ is the so-called *mass matrix* and $C = \{c_{ij}\}$ is the matrix of coefficients that result from the discretization of the convective term $\nabla \cdot (\mathbf{v}u)$. The coefficients of $C$ depend on the computational mesh, on the velocity, and on the numerical method. They may also depend on the unknown solution if the governing equation and/or the numerical scheme is nonlinear.

In the case of a finite element discretization, a diagonal mass matrix $M_L$ can be constructed using the technique known as *row-sum mass lumping*

$$M_L = \mathrm{diag}\{m_i\}, \qquad m_i = \sum_j m_{ij}, \quad \forall i, \tag{6}$$

where $m_{ij}$ are the coefficients of the consistent mass matrix $M_C$. The lumped-mass version of (5) is a system of ordinary differential equations

$$m_i \frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_j c_{ij} u_j. \tag{7}$$

As a rule, the coefficient matrix $C$ is sparse, so that only nearest neighbors make a nonzero contribution to the right-hand side of each equation.

In the course of discretization, the nonnegativity property (4) of the continuous problem may be lost. A semi-discrete scheme is positivity-preserving if

$$u_i(0) \geq 0, \quad \forall i \quad \Rightarrow \quad u_i(t) \geq 0, \quad \forall i, \quad \forall t > 0. \tag{8}$$

A sufficient condition for discretization (7) to be positivity-preserving is

$$m_i > 0, \quad c_{ij} \geq 0, \qquad \forall i, \quad \forall j \neq i. \tag{9}$$

Under this condition, the right-hand side of (7) is nonnegative if $u_i(t) = 0$ and $u_j(t) \geq 0$, $\forall j \neq i$. This implies that $u_i$ may only increase with time

$$c_{ii} u_i = 0, \quad c_{ij} u_j \geq 0, \quad \forall j \quad \Rightarrow \quad \frac{\mathrm{d}u_i}{\mathrm{d}t} \geq 0.$$

To avoid a common misunderstanding, we remark that the numerical solution is not forced to be positive if $\exists j \neq i$ such that $u_j(0) < 0$. Positivity preservation means that the numerical scheme cannot produce *nonphysical* negative values, i.e., undershoots. Likewise, a nonpositive solution is required to preserve its sign, so that no overshoots are generated. Source or sink terms, such as the pressure gradient in the momentum equations, require special care because they may reverse the sign of analytical and numerical solutions alike.

The right-hand side of equation (7) can be decomposed as follows

$$\sum_j c_{ij} u_j = \sum_{j \neq i} c_{ij}(u_j - u_i) + u_i \sum_j c_{ij}. \tag{10}$$

This decomposition is a discrete counterpart of the vector identity

$$\nabla \cdot (\mathbf{v}u) = \mathbf{v} \cdot \nabla u + u \nabla \cdot \mathbf{v} \tag{11}$$

as shown in the Appendix in the context of a finite element discretization.

If the coefficient matrix $C$ has zero row sums then (7) can be written as

$$m_i \frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_{j \neq i} c_{ij}(u_j - u_i). \tag{12}$$

A positivity-preserving scheme of this form proves *local extremum diminishing* (LED). If a maximum is attained at point $i$ then $u_i \geq u_j, \ \forall j \neq i$, whence

$$c_{ij}(u_j - u_i) \leq 0, \quad \forall j \neq i \quad \Rightarrow \quad \frac{\mathrm{d}u_i}{\mathrm{d}t} \leq 0.$$

Thus, a maximum cannot decrease and, similarly, a minimum cannot decrease

$$c_{ij}(u_j - u_i) \geq 0, \quad \forall j \neq i \quad \Rightarrow \quad \frac{\mathrm{d}u_i}{\mathrm{d}t} \geq 0.$$

The LED criterion was introduced by Jameson [14] as a "convenient basis for the construction of nonoscillatory schemes on both structured and unstructured meshes." Note that classical FCT algorithms [4,37] are based on essentially the same design criterion (no new maxima or minima, no growth of existing ones). In one space dimension, the LED property guarantees that the total variation of the discrete solution is a nonincreasing function of time [14]. Thus, one-dimensional LED schemes are total variation diminishing (TVD).

If the row sums of $C$ are nonzero, it is certainly incorrect to demand that the discretization be LED, since the solution of the continuous problem may develop physical maxima and minima due to compressibility effects. Thus, only the 'incompressible' part (the sum over $j \neq i$) is required to be LED. The remainder vanishes for $u_i = 0$ and, thus, poses no hazard to positivity.

### *2.2 Time discretization*

After the discretization in time, one obtains an algebraic system of the form

$$Au^{n+1} = Bu^n, \tag{13}$$

where $A = \{a_{ij}\}$ and $B = \{b_{ij}\}$ are associated with the implicit and explicit part, respectively. The superscript refers to the time level and $u_i^0 = u_i(0), \ \forall i$.

In what follows, we assume that the underlying space discretization (7) is positivity-preserving, i.e., the coefficients of $M_L$ and $C$ satisfy conditions (9). In order to guarantee that the fully discrete version (13) preserves positivity, an upper bound may need to be imposed on the time step $\Delta t = t^{n+1} - t^n$. This bound can be derived using the concept of *monotone* matrices [34,36].

A regular matrix $A$ is called monotone if $A^{-1} \geq 0$ or, equivalently,

$$Au \geq 0 \quad \Rightarrow \quad u \geq 0. \tag{14}$$

Here and below, all inequalities in which the subscripts are omitted are meant to hold componentwise, unless mentioned otherwise.

Clearly, it is impractical to compute the inverse of $A$ and check the sign of its coefficients to prove that $A$ is monotone. Instead, we will consider a special class of monotone matrices that satisfy the following conditions

- all diagonal coefficients of $A$ are positive

$$a_{ii} > 0, \quad \forall i, \tag{15}$$

- $A$ has no positive off-diagonal entries

$$a_{ij} \leq 0, \quad \forall j \neq i, \tag{16}$$

- $A$ is strictly diagonally dominant

$$\sum_j a_{ij} \geq 0, \quad \forall i. \tag{17}$$

These conditions are sufficient for $A$ to be monotone ([31], p. 19). A monotone matrix ($A^{-1} \geq 0$) that satisfies (16) is called an M-matrix [34]. The M-matrix property is widely used to prove *discrete maximum principles* (DMP) for finite element discretizations of elliptic and parabolic problems [9,15].

By virtue of (14), a solution update of the form (13) is positivity-preserving if the coefficients of $A$ satisfy (15)–(17) and $B$ has no negative entries, that is

$$b_{ij} \geq 0, \quad \forall i, \ \forall j. \tag{18}$$

Example 1. Let us discretize (7) in time by the two-level $\theta-$*scheme*

$$m_i \frac{u_i^{n+1} - u_i^n}{\Delta t} = \theta \sum_j c_{ij}^{n+1} u_j^{n+1} + (1-\theta) \sum_j c_{ij}^n u_j^n, \tag{19}$$

where $0 \leq \theta \leq 1$ is an implicitness parameter. This time discretization combines the forward Euler method ($\theta = 0$, first-order), Crank-Nicolson scheme ($\theta = \frac{1}{2}$, second-order), and backward Euler method ($\theta = 1$, first-order).

The system of equations (19) can be written in matrix form (13), where

$$A = M_L - \theta \Delta t C^{n+1}, \qquad B = M_L + (1-\theta)\Delta t C^n. \tag{20}$$

The off-diagonal coefficients of the matrices $A$ and $B$ are given by

$$a_{ij} = -\theta \Delta t c_{ij}^{n+1}, \qquad b_{ij} = (1-\theta)\Delta t c_{ij}^n, \qquad \forall j \neq i \tag{21}$$

and have the right sign ($a_{ij} \leq 0,\ b_{ij} \geq 0$) since we require that $c_{ij} \geq 0$, $\forall j \neq i$.

Conditions (15) and (17) for the diagonal coefficient $a_{ii}$ are satisfied if

$$a_{ii} = m_i - \theta \Delta t c_{ii}^{n+1} > \theta \Delta t \sum_{j \neq i} c_{ij}^{n+1} \geq 0, \qquad \forall i. \tag{22}$$

The diagonal coefficient $b_{ii}$ is nonnegative under the CFL-like condition

$$b_{ii} = m_i + (1-\theta)\Delta t c_{ii}^n \geq 0, \quad \forall i. \tag{23}$$

In summary, a discretization of the form (19) is positivity-preserving if it satisfies (9) as well as restrictions (22)–(23) on the choice of $\theta$ and $\Delta t$.

Example 2. Consider an explicit Runge-Kutta time-stepping method

$$M_L u^{(l)} = \sum_{k=0}^{l-1} \gamma_{kl} \left( M_L u^{(k)} + \theta_{kl} \Delta t C^{(k)} u^{(k)} \right), \tag{24}$$

$$u^{(0)} = u^n, \qquad u^{n+1} = u^{(L)}, \qquad l = 1, \ldots, L, \tag{25}$$

where the parameters $\gamma_{kl}$ and $\theta_{kl}$ are required to satisfy the conditions [12]

$$0 \leq \gamma_{kl} \leq 1, \quad \sum_{k=0}^{l-1} \gamma_{kl} = 1, \quad 0 \leq \theta_{kl} \leq 1. \tag{26}$$

These conditions imply that the right-hand side of (24) is a convex combination of forward Euler predictors with $\Delta t$ replaced by $\theta_{kl}\Delta t$, where $\theta_{kl} \in [0,1]$. It follows that (24) is positivity-preserving under condition (23) with $\theta = 1$.

Gottlieb and Shu [12] call the above family of time-stepping schemes TVD Runge-Kutta methods because they preserve the TVD property of the underlying space discretization in the 1D case. Other (non-TVD but linearly stable) Runge-Kutta schemes can generate spurious oscillations even if the space discretization is TVD. We refer to [12] for a numerical example.

The optimal (in terms of the time step restriction and computational cost) TVD Runge-Kutta method of second order is as follows [12]

$$M_L u^{(1)} = M_L u^n + \Delta t C^n u^n, \tag{27}$$

$$M_L u^{n+1} = \frac{1}{2} M_L u^{(1)} + \frac{1}{2}\left(M_L u^n + \Delta t C^{(1)} u^{(1)}\right). \tag{28}$$

The final solution $u^{n+1}$ represents the average of the forward Euler predictor $u^{(1)}$ and a backward Euler corrector evaluated using $u^{(1)}$ in place of $u^{n+1}$.

## 3 Algebraic flux correction

Conditions (15)–(18) and (22)–(23) may serve as algebraic constraints to be imposed on the coefficients of the numerical scheme. These inequalities are easy to check for arbitrary discretizations in space and time. In many cases, some matrix entries have wrong sign, violating the design criteria presented in the previous section. Since a one-dimensional LED scheme is TVD, it belongs to the class of *monotonicity-preserving* methods [23] which can be at most first-order accurate if the coefficients $c_{ij}$ do not depend on the unknown solution. This statement is known as the Godunov theorem [11]. Modern high-resolution schemes based on flux hybridization of diffusion-antidiffusion (DAD) circumvent this order barrier using flux/slope limiters to fit coefficients to the local solution behavior. A general framework for the design of constrained high-order approximations was presented in [18,21]. It includes a family of implicit FEM-FCT algorithms and can be classified as *algebraic flux correction.*

In this section, we explain how to enforce the positivity constraint and achieve high resolution using an algebraic flux correction scheme. A linear or multilinear finite element discretization of equation (1) can be written as

$$M_C \frac{\mathrm{d}u}{\mathrm{d}t} = Ku, \tag{29}$$

where $M_C = \{m_{ij}\}$ is the consistent mass matrix and $K = \{k_{ij}\}$ the discrete transport operator. The underlying variational formulation and a practical approach to the computation of $m_{ij}$ and $k_{ij}$ are presented in the Appendix.

Discretization (29) is neither LED nor positivity-preserving since $\exists m_{ij} \neq 0$, $j \neq i$ and $\exists k_{ij} < 0,\ j \neq i$. In order to fix this semi-discrete scheme, let us

- approximate the consistent mass matrix $M_C$ by its lumped counterpart $M_L$,
- eliminate all negative off-diagonal coefficients of $K$ by adding an artificial diffusion operator $D = \{d_{ij}\}$ to be designed so that $k_{ij} + d_{ij} \geq 0,\ \forall j \neq i$.

In addition, we require that $D$ be a symmetric matrix with zero row and column sums. These properties provide consistency and mass conservation.

For every pair of nonzero off-diagonal entries $k_{ij}$ and $k_{ji}$, the corresponding coefficients of the artificial diffusion operator $D$ are given by [17,19]

$$d_{ij} = \max\{-k_{ij}, 0, -k_{ji}\} = d_{ji}, \quad \forall j \neq i. \tag{30}$$

The diagonal coefficients of $D$ are defined so as to obtain zero row sums

$$d_{ii} := -\sum_{j\neq i} d_{ij}. \tag{31}$$

Symmetry follows from (30) and implies that the column sums of $D$ are zero.

In practice, there is no need to assemble the global matrix $D$. Instead, artificial diffusion can be built into the discrete transport operator $K$ by setting

$$\begin{aligned} k_{ii} &:= k_{ii} - d_{ij}, \quad & k_{ij} &:= k_{ij} + d_{ij}, \\ k_{ji} &:= k_{ji} + d_{ij}, \quad & k_{jj} &:= k_{jj} - d_{ij}. \end{aligned} \tag{32}$$

In one dimension, this manipulation transforms a linear finite element / central difference approximation into the first-order upwind difference [19].

Replacing $M_C$ by $M_L$ and $K$ by $L$, one obtains a low-order scheme of the form

$$M_L \frac{\mathrm{d}u}{\mathrm{d}t} = Lu \tag{33}$$

which is positivity-preserving but overly diffusive due to its linearity. Hence, a nonlinear antidiffusive correction is required to achieve higher accuracy.

Since both $D$ and $M_C - M_L$ are symmetric matrices with zero row and column sums, the difference between the residuals of systems (29) and (33)

$$f = (M_L - M_C)\frac{\mathrm{d}u}{\mathrm{d}t} - Du \tag{34}$$

can be decomposed into skew-symmetric internodal fluxes as explained below.

By virtue of (31), each component of the diffusive term $Du$ can be written as

$$[Du]_i = \sum_j d_{ij} u_j = \sum_{j\neq i} d_{ij}(u_j - u_i). \tag{35}$$

The error due to mass lumping (6) can be decomposed in a similar way

$$[M_C u - M_L u]_i = \sum_j m_{ij} u_j - m_i u_i = \sum_{j \neq i} m_{ij}(u_j - u_i). \tag{36}$$

Due to (35)–(36) and symmetry, the error (34) admits the decomposition

$$f_i = \sum_{j \neq i} f_{ij}, \qquad f_{ji} = -f_{ij}, \tag{37}$$

where $f_{ij}$ stands for the *raw antidiffusive flux* from node $j$ into node $i$

$$f_{ij} = \left[ m_{ij} \frac{\mathrm{d}}{\mathrm{d}t} + d_{ij} \right] (u_i - u_j). \tag{38}$$

Since the companion flux $f_{ji}$ is the negative of $f_{ij}$, what is subtracted from one node is added to another. Every pair of fluxes can be associated with an edge of the sparsity graph, i.e., with a pair of nonzero off-diagonal coefficients.

In the process of flux correction, every antidiffusive flux $f_{ij}$ is multiplied by a solution-dependent correction factor $\alpha_{ij} \in [0, 1]$ and inserted into the right-hand side of (33). The flux-corrected semi-discrete scheme reads

$$M_L \frac{\mathrm{d}u}{\mathrm{d}t} = Lu + \bar{f}, \qquad \bar{f}_i = \sum_{j \neq i} \alpha_{ij} f_{ij}. \tag{39}$$

The fluxes $f_{ij}$ and $f_{ji} = -f_{ij}$ must be limited using the same correction factor $\alpha_{ij} = \alpha_{ji}$ to maintain skew-symmetry and, hence, discrete conservation.

By construction, the high-order discretization (29) is recovered for $\alpha_{ij} = 1$ and its low-order counterpart (33) for $\alpha_{ij} = 0$. The former setting is usually acceptable in regions of smoothness. However, some artificial diffusion may need to be retained ($0 \leq \alpha_{ij} < 1$) in the neighborhood of steep gradients, where spurious undershoots and overshoots are likely to occur. The job of the flux limiter is to find a set of solution-dependent correction factors (as close to 1 as possible) such that the fully discrete scheme is positivity-preserving, at least for sufficiently small time steps. This scheme has the potential of being higher than first-order accurate since it is nonlinear in the choice of $\alpha_{ij}$.

## 4 Nonlinear FEM-FCT schemes

A family of implicit FEM-FCT algorithms was developed in [17,19,20] using the above approach to flux correction. The time discretization of (39) was performed by the $\theta-$scheme which yields an algebraic system of the form

$$[M_L - \theta\Delta t L^{n+1}]u^{n+1} = [M_L + (1-\theta)\Delta t L^n]u^n + \Delta t \bar{f}(u^{n+1}, u^n). \tag{40}$$

The fully discrete form of the raw antidiffusive flux (38) is as follows

$$\begin{aligned} f_{ij} = &[m_{ij}(u_i^{n+1} - u_j^{n+1}) - m_{ij}(u_i^n - u_j^n)]/\Delta t \\ &+ \theta d_{ij}^{n+1}(u_i^{n+1} - u_j^{n+1}) + (1-\theta)d_{ij}^n(u_i^n - u_j^n). \end{aligned} \tag{41}$$

Interestingly enough, the contribution of the consistent mass matrix consists of a truly antidiffusive implicit part and a diffusive explicit part. Mass diffusion of the form $(M_C - M_L)u^n$ has been used to construct the nonoscillatory low-order scheme within the framework of explicit FEM-FCT algorithms [25].

Note that the antidiffusive term $\bar{f}(u^{n+1}, u^n)$ depends on the unknown solution $u^{n+1}$, even for $\theta = 0$. Therefore, algebraic system (40) is nonlinear and must be solved iteratively. Consider a sequence of approximations $\{u^{(m)}\}$ to the flux-corrected end-of-step solution $u^{n+1}$. The current iterate $u^{(m)}$ can be used to update $\bar{f}$ and the matrix in the left-hand side of the linear system

$$A^{(m)}u^{(m+1)} = B^n u^n + \Delta t \bar{f}(u^{(m)}, u^n), \qquad m = 0, 1, \ldots \tag{42}$$

where $A^{(m)}$ and $B^n$ are associated with the low-order part of (40)

$$A^{(m)} = M_L - \theta\Delta t L^{(m)}, \qquad B^n = M_L + (1-\theta)\Delta t L^n. \tag{43}$$

By construction, the coefficients of these matrices satisfy positivity constraints (15)–(18) if the time step $\Delta t$ is small enough for (22)–(23) to hold.

The iteration process continues until the residuals and/or relative changes become small enough. A natural initial guess is $u^{(0)} = u^n$ but this implies $f_{ij}^{(0)} = d_{ij}^n(u_i^n - u_j^n)$. In our experience, the fixed-point iteration (42) converges faster if the fluxes $f_{ij}$ are initialized using linear extrapolation

$$f_{ij}^{(0)} = [m_{ij}(u_i^n - u_j^n) - m_{ij}(u_i^{n-1} - u_j^{n-1})]/\Delta t + d_{ij}^n(u_i^n - u_j^n). \tag{44}$$

Every solution update of the form (42) can be split into three steps

(1) Compute an explicit low-order approximation $\tilde{u}$ to $u^{n+1-\theta}$ by solving

$$M_L\tilde{u} = [M_L + (1-\theta)\Delta t L^n]u^n. \tag{45}$$

(2) Apply limited antidiffusive fluxes to the intermediate solution $\tilde{u}$

$$M_L\bar{u} = M_L\tilde{u} + \Delta t\bar{f}(u^{(m)}, u^n). \tag{46}$$

(3) Solve the linear system for the new approximation to $u^{n+1}$

$$\left[M_L - \theta\Delta t L^{(m)}\right] u^{(m+1)} = M_L\bar{u}. \tag{47}$$

It is worth mentioning that the auxiliary solution $\tilde{u}$ needs to be computed just once per time step. For $\theta < 1$, the computation of $\tilde{u}$ is positivity-preserving under the CFL-like condition (23). No time step restrictions apply to the fully implicit backward Euler version ($\theta = 1$) because $\tilde{u} = u^n$ in this case.

Flux limiting in the second step is performed using Zalesak's FCT algorithm to be presented in the next section. It is designed to insure that $\bar{u} \geq 0$ for $\tilde{u} \geq 0$. The last step (47) preserves positivity under condition (22) that insures the diagonal dominance of the left-hand side matrix. In summary,

$$u^n \geq 0 \quad \Rightarrow \quad \tilde{u} \geq 0 \quad \Rightarrow \quad \bar{u} \geq 0 \quad \Rightarrow \quad u^{(m+1)} \geq 0 \tag{48}$$

provided that the time step $\Delta t$ satisfies (22) and (23) for a given $\theta \in (0, 1]$.

Remark. The above FEM-FCT algorithm may be used in conjunction with the Crank-Nicolson or backward Euler time-stepping. The forward Euler version ($\theta = 0$) is not to be recommended because the instability of the high-order scheme triggers aggressive flux limiting that may result in a significant distortion of solution profiles. For this reason, a TVD Runge-Kutta scheme, such as (27)–(28), should be employed if a fully explicit solution strategy is preferred.

## 5 Multidimensional Zalesak limiter

Flux correction may be required even if there is no threat to positivity. It might happen that $f_{ij}$ has the same sign as $\tilde{u}_j - \tilde{u}_i$. Such fluxes flatten solution profiles instead of steepening them. As a consequence, numerical ripples may develop within the bounds imposed on the flux-corrected solution [6].

In the original paper by Boris and Book [4] and some other FCT algorithms [27], the sign of a 'defective' antidiffusive flux is reversed, and its amplitude is limited in the usual way. This trick results in a sharp resolution of discontinuities but may distort a smooth profile, which is clearly undesirable. A safer remedy is to cancel $f_{ij}$ if it is directed down the gradient of $\tilde{u}$. That is, set

$$f_{ij} := 0, \quad \text{if} \quad f_{ij}(\tilde{u}_j - \tilde{u}_i) > 0. \tag{49}$$

This optional adjustment is called 'prelimiting' because it must be performed prior to the computation of the correction factors $\alpha_{ij}$ and flux limiting [6,37].

In the case of a finite element discretization, the contribution of the consistent mass matrix insures better phase accuracy but it may reverse the sign of $f_{ij}$, as defined in (41), or significantly increase its magnitude. For particularly sensitive problems, the following *minmod prelimiting* strategy is in order

$$f_{ij} := \textsc{minmod}\{f_{ij}, d_{ij}(\tilde{u}_i - \tilde{u}_j)\}. \tag{50}$$

The MINMOD function returns zero if the two arguments have different signs. Otherwise, the argument with the smaller magnitude is returned. The default value of the nonnegative coefficient $d_{ij}$ is given by equation (30).

In the context of multidimensional FCT algorithms [37], the formula for the correction factors $\alpha_{ij}$ should insure that antidiffusive fluxes *acting in concert* shall not cause the solution value $\bar{u}_i$ to exceed some maximum value $\tilde{u}_i^{\max}$ or fall below some minimum value $\tilde{u}_i^{\min}$. Assuming the worst-case scenario, positive and negative fluxes should be limited separately, as proposed by Zalesak [37]

(1) Compute the sums of positive/negative antidiffusive fluxes into node $i$

$$P_i^+ = \sum_{j\neq i} \max\{0, f_{ij}\}, \qquad P_i^- = \sum_{j\neq i} \min\{0, f_{ij}\}. \tag{51}$$

(2) Compute the distance to a local extremum of the auxiliary solution

$$Q_i^+ = \max\{0, \max_{j\neq i}(\tilde{u}_j - \tilde{u}_i)\}, \qquad Q_i^- = \min\{0, \min_{j\neq i}(\tilde{u}_j - \tilde{u}_i)\}. \tag{52}$$

(3) Compute the nodal correction factors for the net increment to node $i$

$$R_i^+ = \min\left\{1, \frac{m_i Q_i^+}{\Delta t P_i^+}\right\}, \qquad R_i^- = \min\left\{1, \frac{m_i Q_i^-}{\Delta t P_i^-}\right\}. \tag{53}$$

(4) Limit the raw antidiffusive fluxes $f_{ij}$ and $f_{ji}$ in a symmetric fashion

$$\alpha_{ij} = \begin{cases} \min\{R_i^+, R_j^-\}, & \text{if } f_{ij} > 0, \\ \min\{R_i^-, R_j^+\}, & \text{otherwise.} \end{cases} \tag{54}$$

This limiting strategy guarantees that (46) is positivity-preserving since

$$\tilde{u}_i^{\min} = \tilde{u}_i + Q_i^- \leq \bar{u}_i \leq \tilde{u}_i + Q_i^+ = \tilde{u}_i^{\max}. \tag{55}$$

Note that the nodal correction factors $R_i^\pm$ as defined in (53) are inversely proportional to $\Delta t$. Hence, a larger portion of $f_{ij}$ is retained as the time step is refined. This is the reason for the success of FCT in transient computations. On the other hand, the use of large time steps results in a loss of accuracy, and severe convergence problems are observed in the steady state limit.

A well-known problem associated with flux limiting of FCT type is *clipping* [5,37]. In accordance with the FCT design philosophy, the growth of local extrema is suppressed by setting $\alpha_{ij} = 0$ whenever $Q_i^{\pm} = 0$. As a consequence, peaks lose a little bit of amplitude at each time step. TVD methods [13] also reduce to a monotone low-order scheme at local extrema but some geometric high-resolution schemes, such as ENO/WENO, are free of this drawback [3].

## 6 Linearization of antidiffusive fluxes

A major drawback of the nonlinear FEM-FCT algorithm is the need to recompute the raw antidiffusive fluxes (41) and the correction factors (54) after every solution update. Defect correction of the form (42) keeps intermediate solutions positivity-preserving and insures fast convergence of inner iterations due to the M-matrix property of the low-order operator $A^{(m)}$. Unfortunately, the lagged treatment of limited antidiffusion results in slow convergence of outer iterations. At large time steps, as many as 50 sweeps may be required to obtain a fully converged solution (see the numerical examples below). The lumped-mass version, which corresponds to (41) with $m_{ij} = 0$, converges faster but is not to be recommended for strongly time-dependent problems [29]. The convergence of outer iterations can be accelerated using a discrete Newton method [29] to solve (40). However, the costly assembly of the (approximate) Jacobian operator is unlikely to pay off for transient computations that call for the use of small time steps and, in many cases, explicit algorithms.

The cost of an implicit FEM-FCT algorithm can be significantly reduced using a suitable linearization technique. For instance, it is possible to approximate $u^{n+1}$ in formula (41) by the solution $u^H$ of the high-order system

$$[M_C - \theta \Delta t K^{n+1}] u^H = [M_C + (1-\theta)\Delta t K^n] u^n. \tag{56}$$

In this case, the right-hand side of (47) needs to be assembled just once per time step. If the continuous problem is linear, then the left-hand side matrix does not change either, and just one iteration of the form (47) is required to obtain the end-of-step solution $u^{n+1}$. Thus, the computational effort reduces to one call of Zalesak's limiter and two linear systems per time step [19,29]. If the governing equation is nonlinear, so are the two systems to be solved.

In practice, it is usually much more expensive to solve (56) than (47). The main reason is the lack of the M-matrix property which has an adverse effect on the convergence of linear solvers. As the time step increases, inner iterations may fail to converge, even if advanced linear algebra tools are employed. A robust alternative to the brute-force approach is to compute the high-order predictor $u^H$ using fixed-point iteration (42) with $\alpha_{ij} \equiv 1$. However, this strategy is inefficient since the flux-limited version of (42) tends to converge faster [29].

Another possibility is to linearize $f_{ij}$ about the solution of the low-order system

$$[M_L - \theta\Delta t L^{n+1}]u^L = [M_L + (1-\theta)\Delta t L^n]u^n. \tag{57}$$

Unlike (57), this system can be solved efficiently but the flux-corrected solution $u^{n+1}$ computed by (45)–(47) turns out too diffusive (see [32], Section 5.2).

A third approach to the design of linearized FEM-FCT schemes is to compute a “transported and diffused” end-of-step solution $u^L$ and correct it explicitly, as in the case of classical diffusion-antidiffusion (DAD) methods [4,8]

$$M_L u^{n+1} = M_L u^L + \Delta t \bar{f}(u^L, u^n). \tag{58}$$

The provisional solution $u^L$ can be calculated using (57) or any other time discretization of (33), e.g., the explicit TVD Runge-Kutta method (27)–(28).

The raw antidiffusive fluxes for the flux correction step (58) are defined as

$$f_{ij} = m_{ij}(\dot{u}_i^L - \dot{u}_j^L) + d_{ij}^{n+1}(u_i^L - u_j^L), \tag{59}$$

where $\dot{u}^L$ denotes a finite difference approximation of the time derivative. This quantity can be computed, e.g., using the leapfrog method as applied to (29)

$$\dot{u}^L = M_C^{-1}[K^{n+1}u^L]. \tag{60}$$

Since the inverse of $M_C$ is full, we compute successive approximations to $\dot{u}^L$ using Richardson's iteration preconditioned by the lumped mass matrix [7]

$$\dot{u}^{(m+1)} = \dot{u}^{(m)} + M_L^{-1}[K^{n+1}u^L - M_C\dot{u}^{(m)}], \qquad m = 0, 1, \ldots \tag{61}$$

starting with $\dot{u}^{(0)} = 0$ or $\dot{u}^{(0)} = (u^L - u^n)/\Delta t$. Convergence is usually very fast (1-5 iterations) since the consistent mass matrix $M_C$ is well-conditioned.

In fact, it is not necessary to solve (60) very accurately. Even the lumped-mass version $\dot{u}^L = M_L^{-1}[K^{n+1}u^L]$ or the corresponding low-order approximation

$$\dot{u}^L = M_L^{-1}[L^{n+1}u^L] \tag{62}$$

can be used to predict the antidiffusive flux $f_{ij}$. A numerical study of linearized FEM-FCT algorithms based on (60) and (62) will be presented in Section 7.

The new linearization strategy offers a number of significant advantages. First, the low-order predictor $u^L$ can be calculated by an arbitrary (explicit or implicit) time-stepping method. In the case of an implicit algorithm, iterative solvers are fast due to the M-matrix property of the low-order operator. Second, the leapfrog time discretization of the antidiffusive flux is second-order accurate with respect to the time level $t^{n+1}$ on which $u^L$ and $f_{ij}$ are defined. Third, instead of three different solutions ($u^n$, $u^{n+1}$, and $\tilde{u}$) only the smooth predictor $u^L$ is involved in the computation of $f_{ij}$ and $\alpha_{ij}$ for (58). No prelimiting is required for the lumped-mass version ($\dot{u}^L = 0$) since $f_{ij}^L = d_{ij}^{n+1}(u_i^L - u_j^L)$ is truly antidiffusive. For $\dot{u}^L \neq 0$, this flux provides the upper bound for minmod prelimiting (50). Last but not least, all components of (59) admit dimensional splitting, which makes it possible to design a characteristic FEM-FCT algorithm for multidimensional hyperbolic systems as explained below.

## 7 Flux limiting for hyperbolic systems

So far we have discussed algebraic flux correction of FCT type in the context of a scalar transport equation. In this section, we outline an extension to (nonlinear) hyperbolic systems that can be written in the generic form

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F} = 0, \tag{63}$$

where $\mathbf{u}$ is the vector of state variables and $\mathbf{F}$ is the matrix of flux functions.

A system of particular importance is the Euler equations of gas dynamics

$$\frac{\partial}{\partial t}\begin{bmatrix} \rho \\ \rho\mathbf{v} \\ \rho E \end{bmatrix} + \nabla \cdot \begin{bmatrix} \rho\mathbf{v} \\ \rho\mathbf{v}\otimes\mathbf{v} + p\mathcal{I} \\ (\rho E + p)\mathbf{v} \end{bmatrix} = 0. \tag{64}$$

In the case of an ideal polytropic gas, the pressure $p$ is related to the density $\rho$, velocity $\mathbf{v}$, and total energy $E$ by the equation of state

$$p = (\gamma - 1)\rho\left(E - \frac{|\mathbf{v}|^2}{2}\right) \tag{65}$$

in which $\gamma$ is the constant ratio of specific heats. The default is $\gamma = 1.4$ (air).

The group finite element approximation [10] of (63) can be represented in terms of discretized derivatives $c_{ij}^d$ with respect to $x_d$ (see Appendix)

$$\sum_j m_{ij}\frac{\mathrm{d}\mathbf{u}_j}{\mathrm{d}t} = \sum_d \sum_{j\neq i} c_{ij}^d(\mathbf{F}_j^d - \mathbf{F}_i^d), \tag{66}$$

where $\mathbf{F}_i^d$ is the flux in coordinate direction $x_d$. In the case of the Euler equations, $\mathbf{u}_i = [\rho_i, (\rho\mathbf{v})_i, (\rho E)_i]$, and there exists a matrix $\mathbf{A}_{ij}^d$ such that [23]

$$c_{ij}^d(\mathbf{F}_j^d - \mathbf{F}_i^d) = \mathbf{A}_{ij}^d(\mathbf{u}_j - \mathbf{u}_i). \tag{67}$$

This matrix represents an averaged Jacobian operator and is diagonalizable

$$\mathbf{A}_{ij}^d = \mathbf{R}_{ij}^d\mathbf{\Lambda}_{ij}^d[\mathbf{R}_{ij}^d]^{-1}, \tag{68}$$

where $\mathbf{\Lambda}_{ij}^d = \mathrm{diag}\{\lambda_k^d\}$ is the matrix of eigenvalues and $\mathbf{R}_{ij}$ is the matrix of right eigenvectors. A generalization of (30) suggests that artificial viscosity be designed so as to cancel the contribution of negative eigenvalues [20,22]

$$\mathbf{D}_{ij}^d = \mathbf{R}_{ij}^d|\mathbf{\Lambda}_{ij}^d|[\mathbf{R}^d]^{-1}. \tag{69}$$

A cheaper alternative and a usable preconditioner is scalar dissipation

$$\mathbf{D}_{ij}^d = \max_k |\lambda_k^d|\mathbf{I}, \tag{70}$$

where $\mathbf{I}$ is the identity matrix of the same size as $\mathbf{A}_{ij}^d$. In this case, the diffusion coefficient is the same for all variables. The resulting low-order scheme

$$m_i\frac{\mathrm{d}\mathbf{u}_i}{\mathrm{d}t} = \sum_d \sum_{j\neq i}[\mathbf{A}_{ij}^d + \mathbf{D}_{ij}^d](\mathbf{u}_j - \mathbf{u}_i) \tag{71}$$

is more diffusive than that based on (69) but this is acceptable (or even desirable [38]) because excessive artificial diffusion needs to be removed anyway.

Unfortunately, the nonlinearity of the Euler equations makes it impossible to prove that (71) is local extremum diminishing and/or positivity-preserving. However, in most cases it is sufficiently dissipative to suppress nonphysical oscillations in the neighborhood of shocks and contact discontinuities.

At the semi-discrete level, the raw antidiffusive flux is defined as follows

$$\mathbf{f}_{ij} = \left[m_{ij}\mathbf{I}\frac{\mathrm{d}}{\mathrm{d}t} + \sum_d \mathbf{D}_{ij}^d\right](\mathbf{u}_i - \mathbf{u}_j). \tag{72}$$

Flux limiting for systems of equations is a nontrivial task. It can be performed in terms of conservative, primitive or characteristic variables. The use of conservative or primitive variables requires a suitable synchronization of correction factors [25]. In other words, the same coefficient $\alpha_{ij}$ should be applied to all components of $\mathbf{f}_{ij}$. A general algorithm for 'limiting any set of quantities' can be found in [27]. Zalesak [38] was the first to design a 'fail-safe' characteristic FCT limiter. In the rest of this section, we extend his ideas to explicit and implicit FEM-FCT algorithms with flux linearization.

As before, the first step is to compute a low-order predictor $\mathbf{u}^L$ from a fully discrete version of (71). In the case of implicit time-stepping, the nonlinear algebraic system can be solved, e.g., by the defect correction scheme described in [20,22]. The next step is to evaluate the raw antidiffusive fluxes

$$\mathbf{f}_{ij}^d = m_{ij}(\dot{\mathbf{u}}_i^d - \dot{\mathbf{u}}_j^d) + \mathbf{D}_{ij}^d(\mathbf{u}_i^L - \mathbf{u}_j^L), \tag{73}$$

where $\dot{\mathbf{u}}_i^d$ approximates the time rate of change due to fluxes in direction $x_d$. This quantity can be calculated, e.g., by the low-order scheme as in (62)

$$\dot{\mathbf{u}}_i^d = \frac{1}{m_i}\sum_{j\neq i}[\mathbf{A}_{ij}^d + \mathbf{D}_{ij}^d](\mathbf{u}_j^L - \mathbf{u}_i^L). \tag{74}$$

For each space direction $x_d$, factorization (68) can be used to define a set of local characteristic variables for flux correction [22,38]. Every sweep of the characteristic FCT limiter consists of the following algorithmic steps, where all operations on boldface quantities are performed componentwise

(1) Reset auxiliary arrays for Zalesak's limiter: $\mathbf{P}_i^\pm \equiv \mathbf{0},\ \mathbf{Q}_i^\pm \equiv \mathbf{0},\ \mathbf{R}_i^\pm \equiv \mathbf{0}$.

(2) In a loop over edges $\{i, j\}$, transform the solution differences and the raw antidiffusive fluxes to the characteristic variables for direction $x_d$

$$\Delta\mathbf{w}_{ij}^d = [\mathbf{R}_{ij}^d]^{-1}(\mathbf{u}_j^L - \mathbf{u}_i^L), \qquad \mathbf{g}_{ij}^d = [\mathbf{R}_{ij}^d]^{-1}\mathbf{f}_{ij}^d. \tag{75}$$

(3) Perform the optional prelimiting of the transformed fluxes and update

$$\mathbf{P}_i^\pm := \mathbf{P}_i^\pm + \substack{\max\\\min}\{0, \mathbf{g}_{ij}^d\}, \qquad \mathbf{P}_j^\pm := \mathbf{P}_j^\pm + \substack{\max\\\min}\{0, -\mathbf{g}_{ij}^d\}. \tag{76}$$

$$\mathbf{Q}_i^\pm := \substack{\max\\\min}\left\{\mathbf{Q}_i^\pm, \Delta\mathbf{w}_{ij}^d\right\}, \qquad \mathbf{Q}_j^\pm := \substack{\max\\\min}\left\{\mathbf{Q}_j^\pm, -\Delta\mathbf{w}_{ij}^d\right\}. \tag{77}$$

(4) Compute the nodal correction factors for positive and negative increments

$$\mathbf{R}_i^+ := \min\left\{1, \frac{m_i\mathbf{Q}_i^+}{\Delta t\mathbf{P}_i^+}\right\}, \qquad \mathbf{R}_i^- := \min\left\{1, \frac{m_i\mathbf{Q}_i^-}{\Delta t\mathbf{P}_i^-}\right\}. \tag{78}$$

(5) Limit the antidiffusive fluxes and transform to the conservative variables

$$\bar{\mathbf{f}}_{ij}^d = \mathbf{R}_{ij}^d \bar{\mathbf{g}}_{ij}^d, \qquad \bar{\mathbf{g}}_{ij}^d = \begin{cases} \min\{\mathbf{R}_i^+, \mathbf{R}_j^-\}\mathbf{g}_{ij}^d, & \text{if } \mathbf{g}_{ij}^d > \mathbf{0}, \\ \min\{\mathbf{R}_j^+, \mathbf{R}_i^-\}\mathbf{g}_{ij}^d, & \text{otherwise.} \end{cases} \tag{79}$$

(6) Check the result and set $\bar{\mathbf{f}}_{ij}^d := \mathbf{0}$ if nonphysical behavior is detected [38].

(7) Apply the sum of limited antidiffusive fluxes to the low-order predictor

$$\mathbf{u}_i^L := \mathbf{u}_i^L + \frac{1}{m_i}\sum_{j\neq i} \bar{\mathbf{f}}_{ij}^d. \tag{80}$$

Further information regarding the implementation of characteristic flux limiters and boundary conditions for the Euler equations can be found in [20,22].

## 8 Numerical examples

A properly designed numerical algorithm should be capable of resolving both smooth and discontinuous profiles without excessive smoothing or steepening. To assess the accuracy and efficiency of our FEM-FCT schemes, we apply them to several benchmark problems [24,33,35] that we feel are representative and challenging enough to predict how the algorithms under investigation would behave in real-life applications. Analytical and/or numerical solutions are available for each test, which makes it possible to compare the results to those produced by FCT and other high-resolution schemes [21,24,35,38].

In the comparative study that follows, we consider FEM-FCT algorithms based on the TVD Runge-Kutta, Crank-Nicolson, and backward Euler time-stepping. These algorithms will be denoted by RK-FCT-L, CN-FCT-L, and BE-FCT-L, respectively. The last digit refers to the type of linearization. The nonlinear version, as presented in Section 4, and its linearization about the solution $u^H$ of (56) correspond to L=1 and L=2, respectively. Both of these algorithms are based on the $\theta-$scheme, so RK-FCT-1 and RK-FCT-2 are not available. The new approach (59) to flux linearization is denoted by L=3 if $\dot{u}^L$ is computed from (60) and by L=4 if (62) is employed. In the former case, the mass matrix $M_C$ is 'inverted' using 5 iterations of the form (61). However, almost the same accuracy can be achieved by the three-pass algorithm [7].

By default, systems (47), (56) and (57) are solved by the Gauss-Seidel method. BiCGSTAB with ILU preconditioning and Cuthill-McKee renumbering is invoked to speed up convergence at large time steps. All computations are performed on a laptop computer using the Intel Fortran Compiler for Linux.

To quantify the difference between the (analytical or numerical) reference solution $u$ and its approximation $u_h$, we introduce the discrete error norms

$$E_1 = \sum_i m_i |u(x_i, y_i) - u_i| \approx \int_\Omega |u - u_h| \, \mathrm{d}x = ||u - u_h||_1, \tag{81}$$

$$E_2 = \sqrt{\sum_i m_i |u(x_i, y_i) - u_i|^2} \approx \sqrt{\int_\Omega |u - u_h|^2 \, \mathrm{d}x} = ||u - u_h||_2, \tag{82}$$

where $m_i$ are the diagonal coefficients of the lumped mass matrix $M_L$. The goal of the numerical study to be presented was to investigate how the above errors and the CPU times for explicit and implicit FEM-FCT schemes depend on the mesh size $h$, time step $\Delta t$, and linearization technique (if any).

### *8.1 Solid body rotation*

A standard test problem for high-resolution schemes as applied to the linear convection equation (1) in 2D is *solid body rotation* [24,37]. Consider the computational domain $\Omega = (0,1) \times (0,1)$ and the incompressible velocity field

$$\mathbf{v}(x, y) = (0.5 - y, x - 0.5) \tag{83}$$

which corresponds to a counterclockwise rotation about the center $(0.5, 0.5)$ of $\Omega$. After each full revolution ($t = 2\pi k$), the exact solution of (1) coincides with the initial data as defined in [23] and depicted in Fig. 1. The challenge of this test is to preserve the shape of the rotating bodies as far as possible.

At $t = 0$, 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, and the homogeneous Dirichlet boundary condition $g = 0$ is prescribed at the inlets in accordance with (2).

The initial shapes of the three bodies can be expressed in terms of a normalized distance to the corresponding reference point $(x_0, y_0)$ in $\Omega$

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

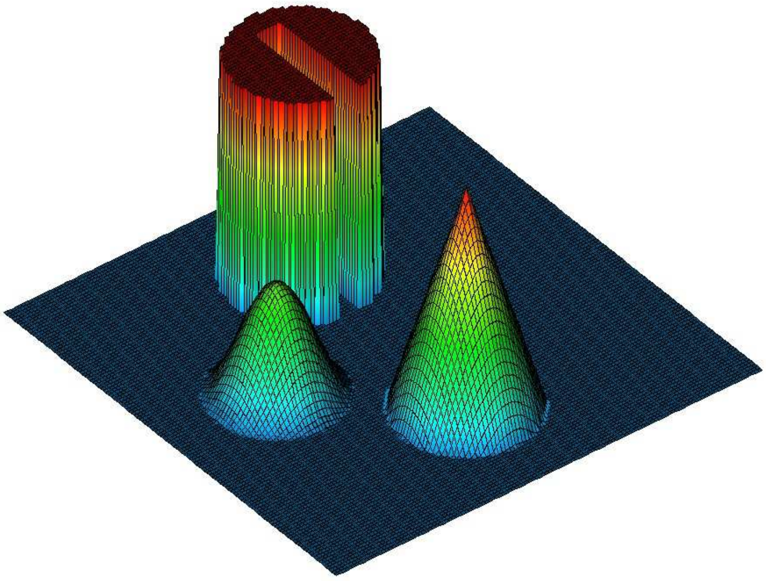


Fig. 1. Initial data / exact solution at the final time.

The formula that describes the shape of the slotted cylinder in the circle $r(x,y) \leq 1$ associated with $(x_0, y_0) = (0.5, 0.75)$ is as follows

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

The cone is centered at $(x_0, y_0) = (0.5, 0.25)$ and its geometry is given by

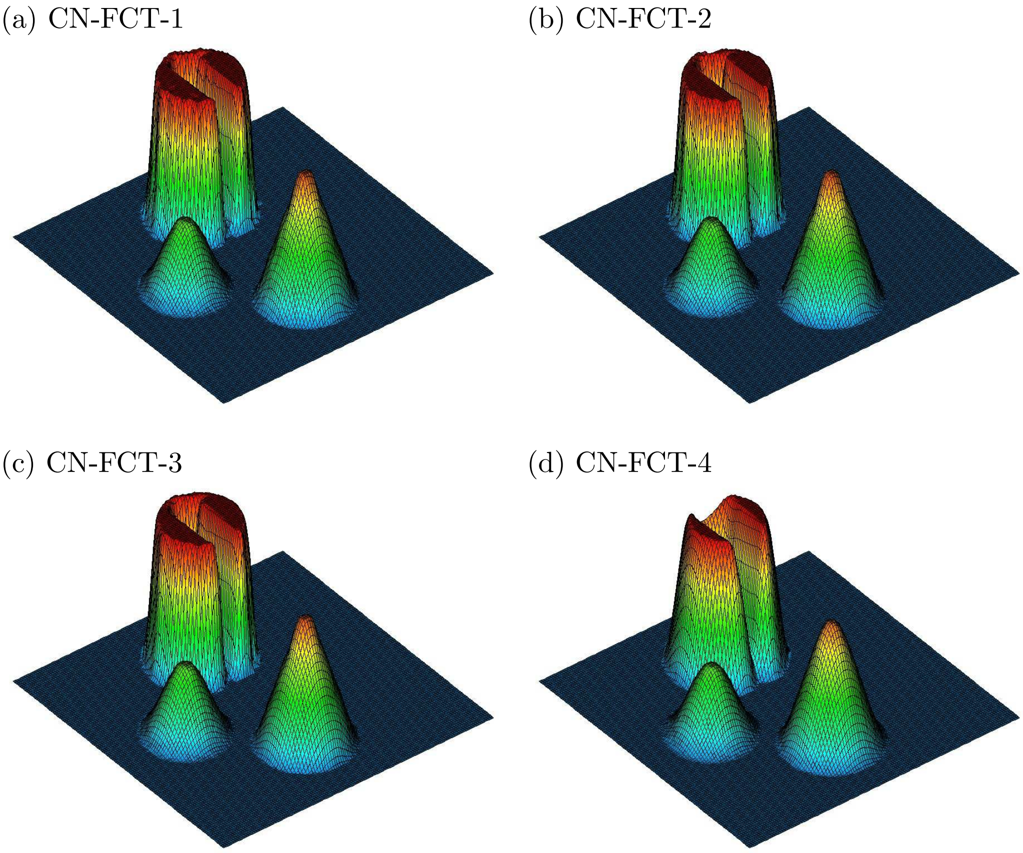
$$u(x,y,0) = 1 - r(x,y).$$

The initial location and shape of the smooth hump are: $(x_0, y_0) = (0.25, 0.5)$,

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

The numerical results produced by the four Crank-Nicolson FEM-FCT schemes after one full revolution ($t = 2\pi$) are presented in Fig. 2. The pictures display the shapes of the numerical solutions computed on a uniform quadrilateral mesh using $128 \times 128$ bilinear elements and the time step $\Delta t = 10^{-3}$. Prelimiting of the form (49) was invoked in CN-FCT-1 through CN-FCT-3, whereas CN-FCT-4 was found to produce ripple-free solutions without prelimiting.

The diagrams in Fig. 3 depict the convergence history for (81) and the total CPU time as a function of the mesh size $h$. It can be seen that the linearized schemes CN-FCT-2 and CN-FCT-3 are almost as accurate as CN-FCT-1. The norms of the error for CN-FCT-4 are larger on all meshes but decrease at a faster rate. Due to the presence of a discontinuous profile, none of the four



(a) CN-FCT-1

(b) CN-FCT-2

(c) CN-FCT-3

(d) CN-FCT-4

Fig. 2. Solid body rotation, CN-FCT schemes, $\mathcal{Q}_1$ elements, $\Delta t = 10^{-3},\ t = 2\pi$.

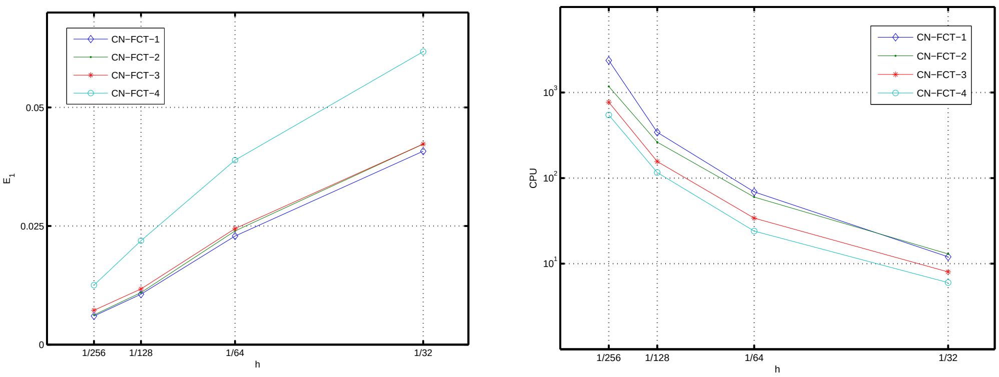


Fig. 3. Solid body rotation, convergence history and CPU times for CN-FCT.

schemes exhibits the theoretically possible second order of convergence. This slowdown is due to flux limiting and (mostly) due to the fact that *a priori* error estimates for the underlying high-order discretization are only valid for smooth solutions. A comparison of the CPU times illustrates the gain of efficiency offered by the new algorithms CN-FCT-3 and CN-FCT-4. The cost of CN-FCT-2 is significantly higher and even exceeds that for CN-FCT-1 on the coarsest mesh due to the slow convergence of the linear solver for the illconditioned high-order system (56). In the case of CN-FCT-1, flux/defect correction of the form (42) was performed as long as required to make the residual of (40) scaled by the norm of the right-hand side smaller than $\epsilon = 10^{-5}$.

The results in Tables 1–3 illustrate the performance of different time-stepping methods and linearization techniques. The errors and CPU times are measured for the numerical solutions computed on the finest mesh ($h = 1/128$). The entry in the last column is the average number of outer iterations (42) required to reach the above tolerance for CN-FCT-1. In the case of linearized FCT schemes, there is no need for iterative defect correction, so NIT=1. For time steps as small as $\Delta t = 10^{-3}$, the second-order accurate RK-FCT and CN-FCT schemes produce essentially the same results, whereas the errors for the first-order accurate BE-FCT version are approximately twice as large (see Table 1). It can be seen that the new approach to flux linearization reduces the difference between the cost (per time step) of explicit and implicit FEM-FCT algorithms. A further gain of efficiency can be achieved using preconditioned Richardson's iteration of the form (61) to update the solution in a fully explicit way.

Table 2 demonstrates that the errors for BE-FCT become disproportionately large as compared to those for RK-FCT and CN-FCT as we increase $\Delta t$ by a factor of 10. Since the backward Euler method is equivalent to the first-order backward difference approximation of the time derivative, it turns out overly diffusive at large time steps. The main advantage of BE-FCT is its unconditional stability and positivity preservation for arbitrary time steps. The poor accuracy of the results in Table 3 indicates that no time-accurate solutions can be obtained with time steps that violate the CFL stability condition in the whole domain. However, if the Courant number $\nu = |\mathbf{v}|\frac{\Delta t}{h}$ exceeds unity only in small subdomains, where the velocity is unusually large and/or adaptive mesh refinement is performed, then a local loss of accuracy is acceptable if the use of large time steps would make the computation much more efficient.

Table 1
Solid body rotation: results for $h = 1/128,\ \Delta t = 10^{-3},\ t = 2\pi$.

| | $E_1$ | $E_2$ | CPU | NIT |
|---|---|---|---|---|
| RK-FCT-3 | 1.1754e-2 | 5.9882e-2 | 127 | 1.0 |
| RK-FCT-4 | 2.1913e-2 | 8.3066e-2 | 84 | 1.0 |
| CN-FCT-1 | 1.0622e-2 | 5.6411e-2 | 343 | 3.5 |
| CN-FCT-2 | 1.0980e-2 | 5.7370e-2 | 263 | 1.0 |
| CN-FCT-3 | 1.1729e-2 | 5.9818e-2 | 156 | 1.0 |
| CN-FCT-4 | 2.1902e-2 | 8.3045e-2 | 116 | 1.0 |
| BE-FCT-1 | 1.9818e-2 | 7.5392e-2 | 280 | 2.5 |
| BE-FCT-2 | 2.0069e-2 | 7.5862e-2 | 255 | 1.0 |
| BE-FCT-3 | 2.1131e-2 | 7.9686e-2 | 155 | 1.0 |
| BE-FCT-4 | 2.7443e-2 | 9.2886e-2 | 110 | 1.0 |

Table 2
Solid body rotation: results for $h = 1/128,\ \Delta t = 10^{-2},\ t = 2\pi$.

| | $E_1$ | $E_2$ | CPU | NIT |
|---|---|---|---|---|
| RK-FCT-3 | 1.8289e-2 | 7.5075e-2 | 13 | 1.0 |
| RK-FCT-4 | 2.4417e-2 | 8.8419e-2 | 8 | 1.0 |
| CN-FCT-1 | 1.2867e-2 | 6.2870e-2 | 173 | 19.7 |
| CN-FCT-2 | 1.3552e-2 | 6.5033e-2 | 27 | 1.0 |
| CN-FCT-3 | 1.7018e-2 | 7.3535e-2 | 17 | 1.0 |
| CN-FCT-4 | 2.3676e-2 | 8.7242e-2 | 13 | 1.0 |
| BE-FCT-1 | 5.5943e-2 | 1.3651e-1 | 155 | 15.9 |
| BE-FCT-2 | 5.6119e-2 | 1.3675e-1 | 36 | 1.0 |
| BE-FCT-3 | 5.7247e-2 | 1.3966e-1 | 17 | 1.0 |
| BE-FCT-4 | 5.8198e-2 | 1.4102e-1 | 13 | 1.0 |

Table 3
Solid body rotation: results for $h = 1/128,\ \Delta t = 10^{-1},\ t = 2\pi$.

| | $E_1$ | $E_2$ | CPU | NIT |
|---|---|---|---|---|
| CN-FCT-1 | 7.3711e-2 | 1.6587e-1 | 54 | 35.0 |
| BE-FCT-1 | 1.0519e-1 | 2.0244e-1 | 92 | 51.3 |
| BE-FCT-3 | 1.0504e-1 | 2.0250e-1 | 4 | 1.0 |
| BE-FCT-4 | 1.0506e-1 | 2.0251e-1 | 3 | 1.0 |

No results for RK-FCT and linearized CN-FCT are presented in Table 3 since these schemes turn out to be unstable for the time step $\Delta t = 10^{-1}$ that exceeds the upper bound imposed by the CFL-like condition (23). CN-FCT-1 remains stable and more accurate than BE-FCT but the solution is not guaranteed to be positivity-preserving. The results for BE-FCT-2 are missing due to the failure of the BiCGSTAB solver as applied to the high-order system (56). At large time steps, the cost of a nonlinear FEM-FCT algorithm becomes very high due to slow convergence of inner and outer iterations. In the case of BE-FCT-1, more than 50 flux/defection steps are required to advance the solution from one time level to the next in Table 3. Flux linearization makes it possible to obtain the same results 30 times faster using BE-FCT-3 or BE-FCT-4.

### *8.2 Swirling flow*

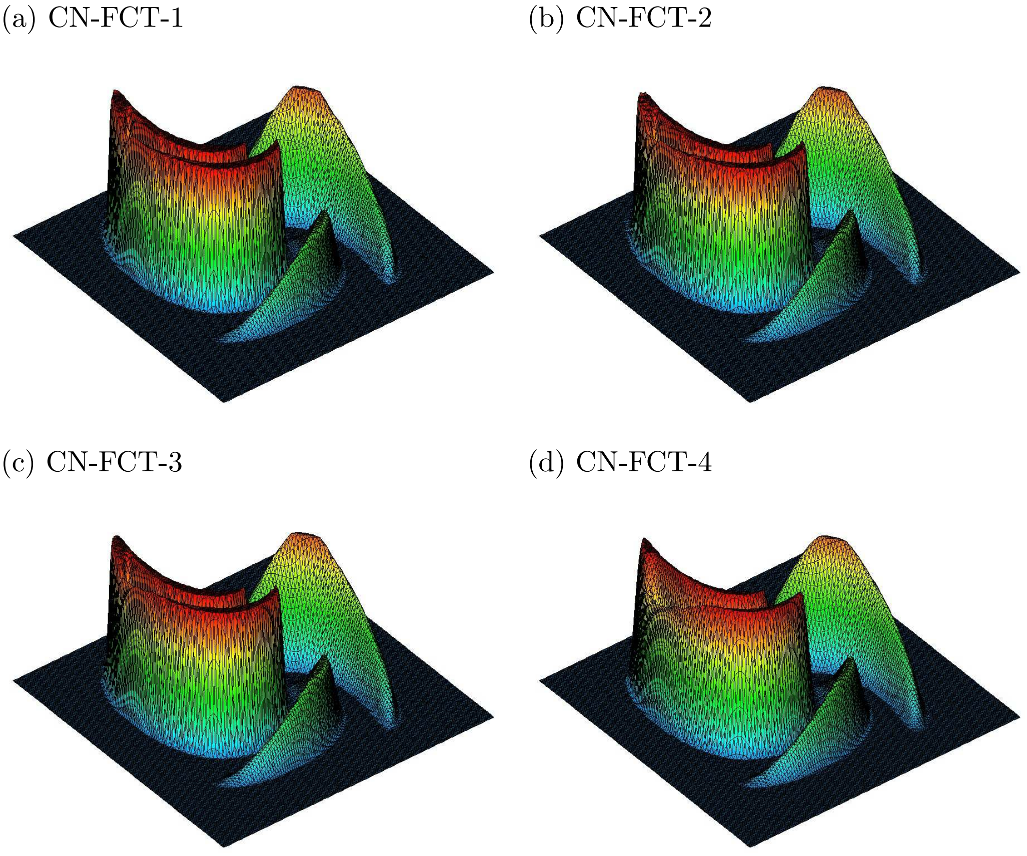
In the next test, we consider the same equation, the same domain, and the same initial data as before but the velocity field is given by the formula [24]

$$\mathbf{v}(x, y, t) = (\sin^2(\pi x)\sin(2\pi y)g(t), -\sin^2(\pi y)\sin(2\pi x)g(t)), \tag{84}$$

where $g(t) = \cos(\pi t/T)$ on the time interval $0 \leq t \leq T$. This time-dependent velocity field describes a swirling deformation flow that provides a more severe test than solid body rotation with a constant angular velocity.

Since $\mathbf{v} = (0, 0)$ on the boundaries of the unit square, no boundary conditions need to be prescribed in the case of pure convection. The function $g(t)$ is designed so that the flow slows down and eventually reverses its direction in such a way that the initial profile is recovered at time $t = T$. Thus, the analytical solution at $t = T$ is available and reproduces the configuration depicted in Fig. 1 although the flow field has a fairly complicated structure.

The numerical solutions in Fig. 4–5 were computed by the four CN-FCT schemes using linear finite elements and $\Delta t = 10^{-3}$. The underlying triangular mesh has the same vertices and twice as many cells as its quadrilateral counterpart with $h = 1/128$. The snapshots presented in Fig. 4 correspond to the time of maximum deformation $t = T/2$ and those in Fig. 5 to the final time $T = 1.5$. Although the solution undergoes significant deformations in the course of simulation, the shape of the initial data is recovered fairly well. As in the case of solid body rotation, erosion of the slotted cylinder is stronger for CN-FCT-4 than for the other three schemes. On the other hand, the artificial

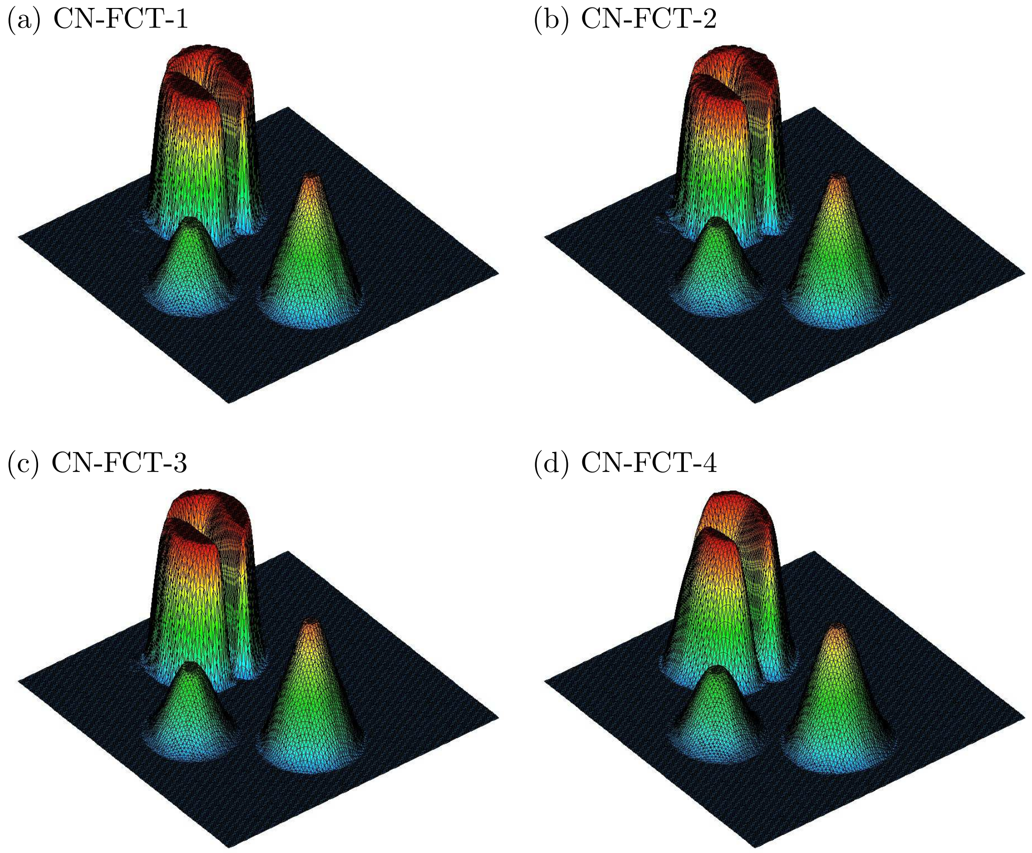

(a) CN-FCT-1

(b) CN-FCT-2

(c) CN-FCT-3

(d) CN-FCT-4

Fig. 4. Swirling deformation, CN-FCT schemes, $\mathcal{P}_1$ elements, $\Delta t = 10^{-3}$, $t = 0.75$.


(a) CN-FCT-1

(b) CN-FCT-2

(c) CN-FCT-3

(d) CN-FCT-4

Fig. 5. Swirling deformation, CN-FCT schemes, $\mathcal{P}_1$ elements, $\Delta t = 10^{-3}$, $t = 1.5$.

steepening of smooth profiles is alleviated since the linearized antidiffusive flux is smooth and does not need to be prelimited, at least in this particular test.

The error norms and CPU times for all FEM-FCT algorithms as applied to swirling flow are presented in Tables 4–6. Since the velocity field is time-dependent, the coefficients $k_{ij}$ and $d_{ij}$ need to be updated after each time step. An efficient approach to matrix update for time-dependent problems is described in the Appendix. Since matrix assembly claims a larger share of the CPU time, the difference between the cost of explicit and implicit schemes is smaller than in the case of a stationary velocity field. In Table 4, the differences between the CPU times for RK-FCT-4, CN-FCT-4, and BE-FCT-4 are marginal since the convergence of the Gauss-Seidel solver is very fast.

At intermediate and large time steps, the convergence of implicit solvers slows down but this is the price to be paid for robustness. Tables 5–6 summarize the results for $\Delta t = 10^{-2}$ and $\Delta t = 10^{-3}$. Both explicit RK-FCT algorithms turned out unstable, and the linear solver for BE-FCT-2 failed in the latter test. Again, the new approach to flux linearization proved very efficient as compared to the nonlinear version and the one based on a high-order predictor. The associated loss of accuracy is acceptable, especially in the case of BE-FCT.

Table 4
Swirling deformation: results for $h = 1/128,\ \Delta t = 10^{-3},\ t = 1.5$.

| | $E_1$ | $E_2$ | CPU | NIT |
|---|---|---|---|---|
| RK-FCT-3 | 1.4440e-2 | 6.6023e-2 | 45 | 1.0 |
| RK-FCT-4 | 2.4558e-2 | 8.9130e-2 | 36 | 1.0 |
| CN-FCT-1 | 1.2043e-2 | 5.8858e-2 | 122 | 5.6 |
| CN-FCT-2 | 1.3049e-2 | 6.1370e-2 | 60 | 1.0 |
| CN-FCT-3 | 1.4300e-2 | 6.5626e-2 | 50 | 1.0 |
| CN-FCT-4 | 2.4493e-2 | 8.8983e-2 | 42 | 1.0 |
| BE-FCT-1 | 2.4606e-2 | 8.4485e-2 | 112 | 4.8 |
| BE-FCT-2 | 2.5185e-2 | 8.5713e-2 | 60 | 1.0 |
| BE-FCT-3 | 2.5334e-2 | 8.5644e-2 | 49 | 1.0 |
| BE-FCT-4 | 3.1814e-2 | 1.0039e-1 | 41 | 1.0 |

Table 5
Swirling deformation: results for $h = 1/128,\ \Delta t = 10^{-2},\ t = 1.5$.

| | $E_1$ | $E_2$ | CPU | NIT |
|---|---|---|---|---|
| CN-FCT-1 | 2.2380e-2 | 8.1277e-2 | 38 | 17.9 |
| CN-FCT-2 | 2.3670e-2 | 8.4051e-2 | 10 | 1.0 |
| CN-FCT-3 | 2.4119e-2 | 8.6538e-2 | 6 | 1.0 |
| CN-FCT-4 | 2.8809e-2 | 9.6268e-2 | 5 | 1.0 |
| BE-FCT-1 | 6.4479e-2 | 1.4867e-1 | 53 | 21.1 |
| BE-FCT-2 | 6.4621e-2 | 1.4885e-1 | 11 | 1.0 |
| BE-FCT-3 | 6.3877e-2 | 1.4760e-1 | 6 | 1.0 |
| BE-FCT-4 | 6.4827e-2 | 1.4907e-1 | 5 | 1.0 |

Table 6
Swirling deformation: results for $h = 1/128,\ \Delta t = 10^{-1},\ t = 1.5$.

| | $E_1$ | $E_2$ | CPU | NIT |
|---|---|---|---|---|
| CN-FCT-1 | 6.3013e-2 | 1.3422e-1 | 12 | 24.1 |
| CN-FCT-3 | 6.4958e-2 | 1.3829e-1 | 1 | 1.0 |
| CN-FCT-4 | 6.3189e-2 | 1.3556e-1 | 1 | 1.0 |
| BE-FCT-1 | 1.1173e-1 | 2.0886e-1 | 11 | 17.7 |
| BE-FCT-3 | 1.1155e-1 | 2.0870e-1 | 1 | 1.0 |
| BE-FCT-4 | 1.1155e-1 | 2.0870e-1 | 1 | 1.0 |

### *8.3 Shock tube problem*

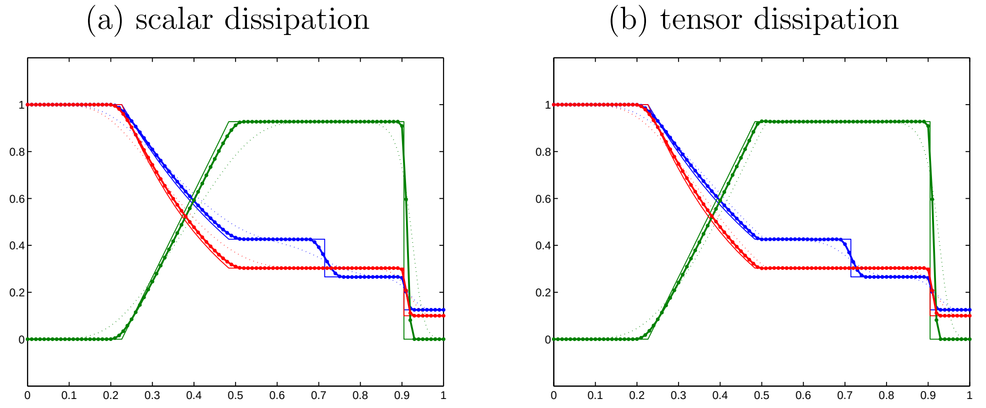
Sod's shock tube problem [33] has become a standard benchmark for high-resolution schemes, as applied to the one-dimensional Euler equations. The physical process to be simulated is the flow of gas in a tube initially separated by a membrane into two sections. Reflective boundary conditions are prescribed at the endpoints of $\Omega = (0,1)$. The initial condition for the nonlinear Riemann problem to be solved is given in terms of the primitive variables

$$\begin{bmatrix} \rho_L \\ v_L \\ p_L \end{bmatrix} = \begin{bmatrix} 1.0 \\ 0.0 \\ 1.0 \end{bmatrix}, \qquad \begin{bmatrix} \rho_R \\ v_R \\ p_R \end{bmatrix} = \begin{bmatrix} 0.125 \\ 0.0 \\ 0.15 \end{bmatrix}, \tag{85}$$

where the subscripts refer to the subdomains $\Omega_L = (0, 0.5)$ and $\Omega_R = (0.5.1)$.

The solid lines in Fig. 6 correspond to a snapshot of the analytical solution at $t = 0.231$. This solution can be constructed, for example, using the technique described by Anderson [1]. It features a shock wave, a contact discontinuity, and a rarefaction wave. The presented numerical solutions are computed using CN-FCT-1 with 100 linear finite elements $\Delta t = 10^{-3}$. The dotted lines correspond to the low-order approximation based on (71) and (69) or (70). It can be seen that scalar dissipation (SD, left diagram) produces a more diffusive solution than tensor dissipation (TD, right diagram) but the differences between their flux-limited counterparts (bullet-marked solid lines) are not so pronounced. The error norms for all solutions are presented in Table 7.

Flux correction was performed in terms of local characteristic variables by the algorithm outlined in Section 7. In this particular test, the prelimiting of raw antidiffusive fluxes turned out to be unnecessary and was deactivated. The nonlinear algebraic system associated with the Crank-Nicolson time discretization of (71) was solved using the defect correction method with a diagonal preconditioner. At time steps as small as $\Delta t = 10^{-3}$, one iteration per time step was found to be sufficient. Thus, implicit schemes have the potential of being as efficient as explicit ones even if the problem is nonlinear and the use of extremely small time steps is dictated by accuracy considerations.


(a) scalar dissipation

(b) tensor dissipation

Fig. 6. Shock tube: exact solution (solid line), low-order solution (dotted line), and CN-FCT solution (bullet-marked line) for $h = 10^{-2}$, $\Delta t = 10^{-3}$, $t = 0.231$.

Table 7
Shock tube problem: results for $h = 10^{-2}$, $\Delta t = 10^{-3}$, $t = 0.231$.

|  | $\rho$ |  | $v$ |  | $p$ |  |
|---|---|---|---|---|---|---|
|  | $E_1$ | $E_2$ | $E_1$ | $E_2$ | $E_1$ | $E_2$ |
| SD-low | 2.8971e-2 | 1.4362e-3 | 5.6592e-2 | 1.2112e-2 | 2.6554e-2 | 1.6126e-3 |
| SD-lim | 7.5274e-3 | 2.4005e-4 | 1.4516e-2 | 3.8550e-3 | 6.2166e-3 | 2.0844e-4 |
| TD-low | 2.2876e-2 | 9.8709e-4 | 4.6541e-2 | 1.1114e-2 | 2.1127e-2 | 1.1497e-3 |
| TD-lim | 7.0087e-3 | 2.1452e-4 | 1.4242e-2 | 3.8345e-3 | 6.3035e-3 | 2.1531e-4 |

### *8.4 Blast wave problem*

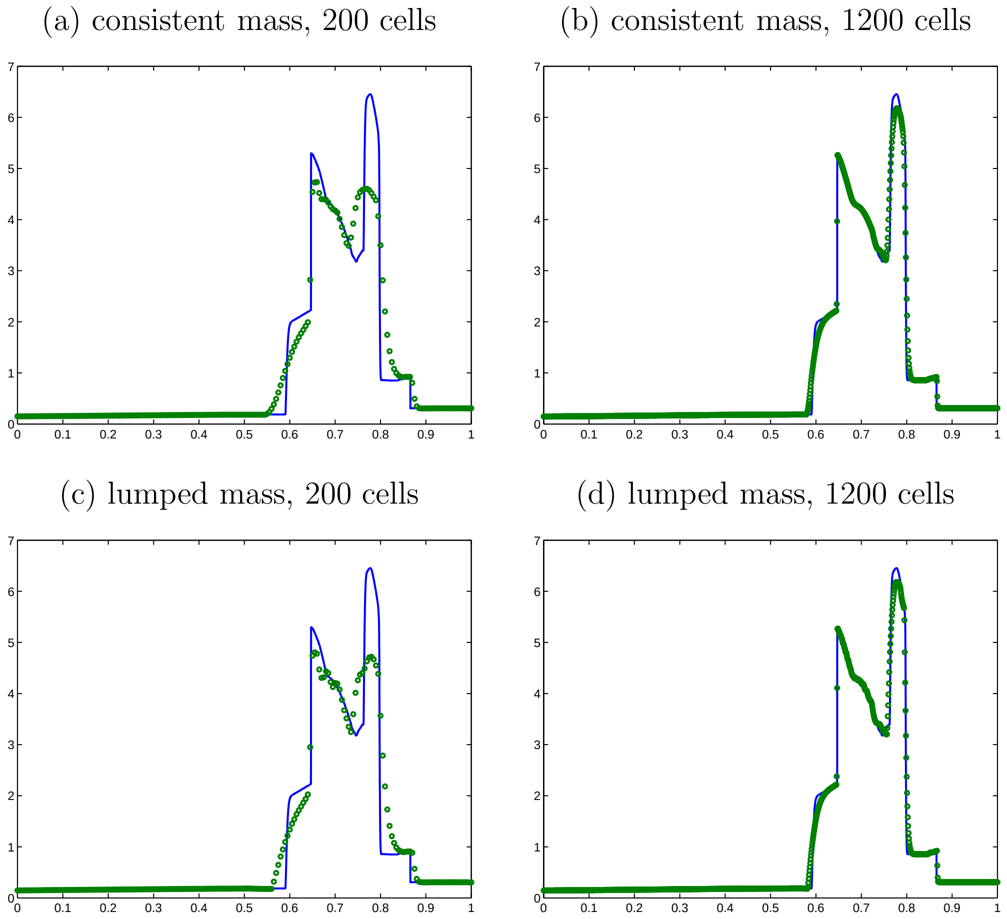
The blast wave problem of Woodward and Colella [35] is a far more challenging test. The flow of a gamma-law gas, with $\gamma = 1.4$, takes place between reflecting walls, and the initial condition consists of the three constant states

$$\begin{bmatrix} \rho_L \\ v_L \\ p_L \end{bmatrix} = \begin{bmatrix} 1.0 \\ 0.0 \\ 1000.0 \end{bmatrix}, \qquad \begin{bmatrix} \rho_M \\ v_M \\ p_M \end{bmatrix} = \begin{bmatrix} 1.0 \\ 0.0 \\ 0.01 \end{bmatrix}, \qquad \begin{bmatrix} \rho_R \\ v_R \\ p_R \end{bmatrix} = \begin{bmatrix} 1.0 \\ 0.0 \\ 100.0 \end{bmatrix} \tag{86}$$

associated with $\Omega_L = (0, 0.1)$, $\Omega_M = (0.1, 0.9)$, and $\Omega_R = (0.9, 1)$, respectively.

The above initial conditions give rise to two strong blast waves which eventually collide. The flow evolution involves complex interactions of shocks, rarefactions, and contact discontinuities in a small region of space. These interactions are particularly difficult to capture using FCT because of its tendency to clip peaks and distort steep fronts. The latter shortcoming is called *terracing* [5]. It can be alleviated by means of prelimiting and/or by increasing the phase accuracy of the high-order scheme [38]. In the FEM-FCT context, this means that the contribution of the consistent mass matrix should be included ($\dot{u}^L \neq 0$).

Figure 7 displays the density distribution at the final time $t = 0.038$. The solid line is the reference solution computed by the characteristic CN-FCT-1 algorithm with minmod prelimiting (50) using $h = 10^{-4}$ and $\Delta t = 10^{-7}$. The grid convergence history for CN-FCT-1 is presented in Table 8. A closer look at the numerical solutions in Fig. 7a and Fig. 7c illustrates the detrimental effect of mass lumping. We invite the reader to compare the results in Fig. 7a–b to those produced by the classical ETBFCT scheme in the paper by Woodward and Colella [35]. The numerical study performed by Zalesak [38] indicates that the accuracy of a characteristic FCT algorithm, as applied to the blast wave problem, increases with the resolving power of the high-order scheme.



(a) consistent mass, 200 cells

(b) consistent mass, 1200 cells

(c) lumped mass, 200 cells

(d) lumped mass, 1200 cells

Fig. 7. Blast wave problem: density distribution at $t = 0.038$. Reference solution (solid line) and CN-FCT solutions on coarser grids (bullet markers).

Table 8
Blast wave problem: results for $\Delta t = 10^{-6}$, $t = 0.038$.

| | $\rho$ | | $v$ | | $p$ | |
|---|---|---|---|---|---|---|
| $h$ | $E_1$ | $E_2$ | $E_1$ | $E_2$ | $E_1$ | $E_2$ |
| 1/200 | 1.8532e-1 | 2.1680e-1 | 4.0423e-1 | 1.6745e00 | 4.7604e00 | 1.7208e+2 |
| 1/400 | 1.1177e-1 | 9.9591e-2 | 2.3738e-1 | 9.8402e-1 | 2.8402e00 | 9.1402e+1 |
| 1/800 | 6.2420e-2 | 3.9024e-2 | 1.2592e-1 | 4.5909e-1 | 1.6313e00 | 4.4893e+1 |
| 1/1600 | 3.4197e-2 | 1.6692e-2 | 6.1241e-2 | 1.9446e-2 | 8.9095e-1 | 2.4748e+1 |

### *8.5 Double Mach reflection*

Another challenging test problem was devised by Woodward and Colella [35] for the two-dimensional Euler equations. The flow pattern involves a Mach 10 shock in air ($\gamma = 1.4$) which initially makes a 60° angle with a reflecting wall.

The computational domain for the double Mach reflection problem is the rectangle $\Omega = (0,4) \times (0,1)$. The following pre-shock and post-shock values of the flow variables are used to define the initial and boundary conditions [2]

$$\begin{bmatrix} \rho_L \\ u_L \\ v_L \\ p_L \end{bmatrix} = \begin{bmatrix} 8.0 \\ 8.25\cos(30^\circ) \\ -8.25\sin(30^\circ) \\ 116.5 \end{bmatrix}, \qquad \begin{bmatrix} \rho_R \\ u_R \\ v_R \\ p_R \end{bmatrix} = \begin{bmatrix} 1.4 \\ 0.0 \\ 0.0 \\ 1.0 \end{bmatrix}. \tag{87}$$

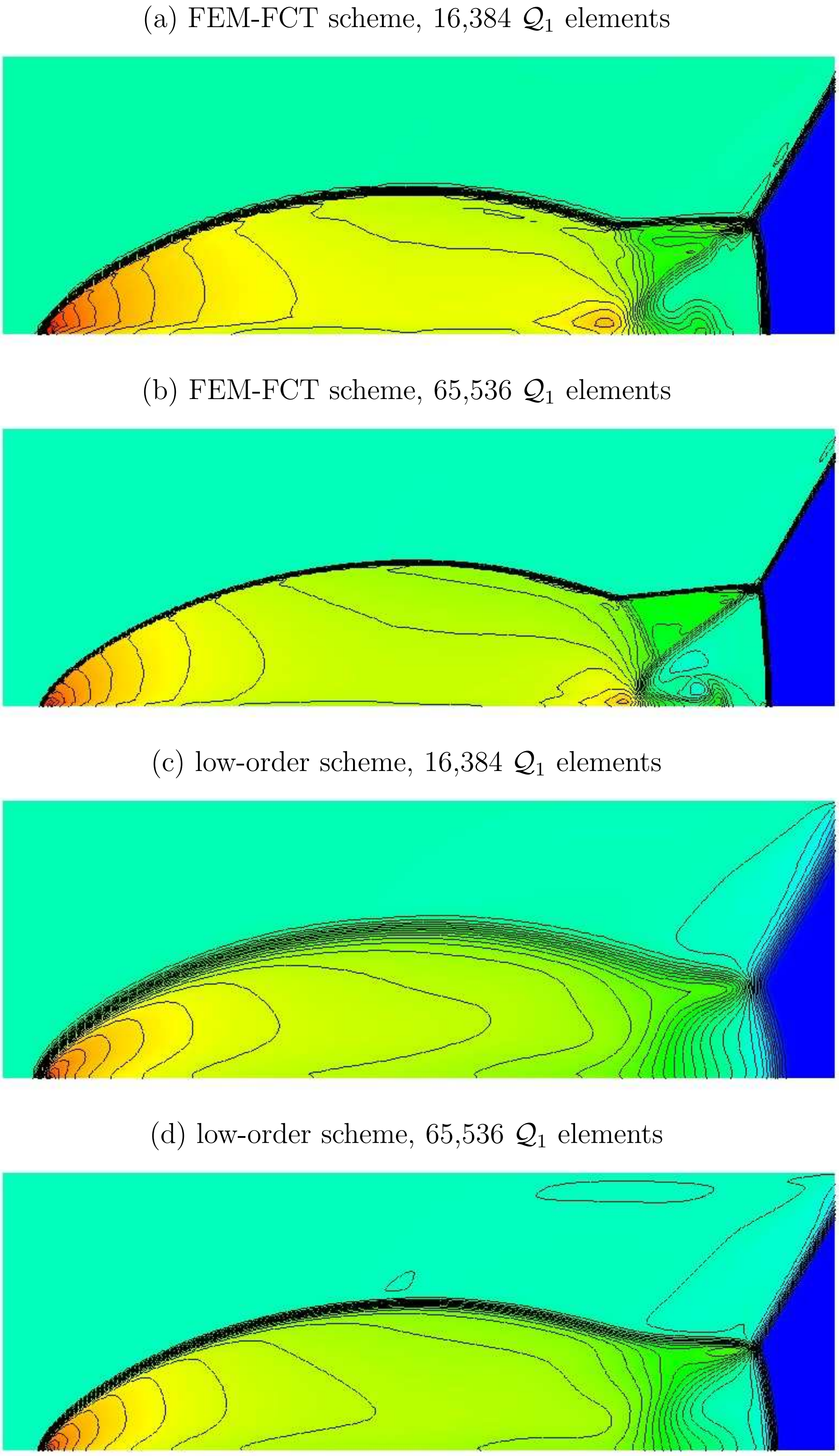
Initially, the post-shock values (subscript $L$) are prescribed in the subdomain $\Omega_L = \{(x,y) \mid x < 1/6 + y/\sqrt{3}\}$ and the pre-shock values (subscript $R$) in $\Omega_R = \Omega \backslash \Omega_L$. The reflecting wall corresponds to $1/6 \leq x \leq 4$ and $y = 0$. No boundary conditions are required along the line $x = 4$. On the rest of the boundary, the post-shock conditions are assigned for $x < 1/6 + (1 + 20t)/\sqrt{3}$ and the pre-shock conditions elsewhere [2]. The so-defined values along the top boundary describe the exact motion of the initial Mach 10 shock.

The reflecting boundary conditions were implemented by invoking a Riemann solver as applied to the set of boundary values and the corresponding ghost state in which the sign of the normal velocity component is reversed [16]. This implementation technique insures that no flow penetrates a solid wall.

The numerical solutions in Fig. 8 depict the density distribution in the rectangle $(0,3) \times (0,1)$ at $t = 0.02$. They were computed using a bilinear finite element approximation and Crank-Nicolson time-stepping. As before, the characteristic FEM-FCT algorithm was equipped with minmod prelimiting. The antidiffusive correction of the low-order solution yields a significant gain of accuracy without generating 'staircase structures' and other peculiar features that were observed by Woodward and Colella [35] in the solutions produced by ETBFCT. Similar results were obtained by Zalesak [38] using characteristic FCT with independent or sequential limiting of the $x-$ and $y-$directed fluxes.

## 9 Conclusions

In the present paper, we addressed the design of implicit and implicit FEM-FCT schemes for transient flow phenomena dominated by convective transport. The main highlights were a new approach to linearization of raw antidiffusive fluxes and its extension to finite element discretizations of the compressible Euler equations. The predictor-corrector strategy of diffusion-antidiffusion type eliminates the need for iterative flux correction and can be combined with an arbitrary time stepping method that preserves the positivity of the under-



(a) FEM-FCT scheme, 16,384 $\mathcal{Q}_1$ elements

(b) FEM-FCT scheme, 65,536 $\mathcal{Q}_1$ elements

(c) low-order scheme, 16,384 $\mathcal{Q}_1$ elements

(d) low-order scheme, 65,536 $\mathcal{Q}_1$ elements

Fig. 8. Double Mach reflection: density distribution at $t = 0.02$.

lying space discretization, at least for sufficiently small time steps. The new FEM-FCT schemes based on the Runge-Kutta, Crank-Nicolson, and backward Euler time-stepping prove more robust and/or efficient than their predecessors proposed in [19–21]. Since flux correction needs to be performed just once per time step, an implicit approach is likely to pay off, for example, in the case of strongly varying velocity fields and locally refined unstructured meshes.

As always, the optimal choice of the time-stepping method, of the iterative solver, and of the limiting strategy is highly problem-dependent. Algebraic flux correction of FCT type is to be recommended for strongly time-dependent problems, whereas multidimensional flux limiters of TVD type are available for steady-state computations [21]. Further research is required to circumvent the first-order accuracy of the unconditionally positivity-preserving backward Euler method and CFL-like conditions that apply to second-order time discretizations. This can be accomplished, e.g., by using a *local $\theta$-scheme* [32] or by blending a backward Euler predictor with a Crank-Nicolson corrector.

## Acknowledgments

The author would like to thank the anonymous reviewers for many valuable and insightful remarks regarding the original version of the manuscript.

## References

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

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

[3] T. Barth, M. Ohlberger, Finite volume methods: foundation and analysis, in: E. Stein, R. de Borst, T. J. R. Hughes (Eds.), Encyclopedia of Computational Mechanics, John Wiley & Sons, 2004, pp. 439–474.

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

[5] D. L. Book, The conception, gestation, birth, and infancy of FCT, in: D. Kuzmin, R. Löhner, S. Turek (Eds.), Flux-Corrected Transport: Principles, Algorithms, and Applications, Springer, Berlin, 2005, pp. 5–28.

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

[7] J. Donea, S. Giuliani, H. Laval, L. Quartapelle, Time-accurate solution of advection-diffusion equations by finite elements, Comp. Meth. Appl. Mech. Eng. 193 (1984) 123–145.

[8] G. S. Dietachmayer, A comparison and evaluation of some positive definite advection schemes, in: J. Noyle, R. May (Eds.), Computational Techniques and Applications, Elsevier, 1986, pp. 217–232.

[9] I. Farago, R. Horvath, S. Korotov, Discrete maximum principle for linear parabolic problems solved on hybrid meshes, Appl. Numer. Math. 53 (2-4) (2005) 249–264.

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

[11] S. K. Godunov, Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics, Mat. Sb. 47 (1959) 271–306.

[12] S. Gottlieb, C. W. Shu, Total Variation Diminishing Runge-Kutta schemes, Math. Comp. 67 (1998) 73–85.

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

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

[15] J. Karatson, S. Korotov, M. Křížek, On discrete maximum principles for nonlinear elliptic problems, Math. Comp. Simulation 76 (2007) 99–108.

[16] L. Krivodonova, M. Berger, High-order accurate implementation of solid wall boundary conditions in curved geometries. J. Comput. Phys. 211 (2006) 492–512.

[17] D. Kuzmin, Positive finite element schemes based on the flux-corrected transport procedure, in: K. J. Bathe (Ed.), Computational Fluid and Solid Mechanics, Elsevier, 2001, pp. 887-888.

[18] D. Kuzmin, On the design of general-purpose flux limiters for implicit FEM with a consistent mass matrix. I. Scalar convection, J. Comput. Phys. 219 (2006) 513–531.

[19] D. Kuzmin, S. Turek, Flux correction tools for finite elements, J. Comput. Phys. 175 (2002) 525–558.

[20] D. Kuzmin, M. Möller, S. Turek, High-resolution FEM-FCT schemes for multidimensional conservation laws, Comp. Meth. Appl. Mech. Eng. 193 (2004) 4915–4946.

[21] D. Kuzmin, M. Möller, Algebraic flux correction I. Scalar conservation laws, in: D. Kuzmin, R. Löhner, S. Turek (Eds.), Flux-Corrected Transport: Principles, Algorithms, and Applications, Springer, Berlin, 2005, pp. 155–206.

[22] D. Kuzmin, M. Möller, Algebraic flux correction II. Compressible Euler equations, in: D. Kuzmin, R. Löhner, S. Turek (Eds.), Flux-Corrected Transport: Principles, Algorithms, and Applications, Springer, Berlin, 2005, pp. 207–250.

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

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

[25] R. Löhner, K. Morgan, J. Peraire, 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.

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

[27] R. Löhner, J. D. Baum, 30 Years of FCT: Status and Directions, in: D. Kuzmin, R. Löhner, S. Turek (Eds.), Flux-Corrected Transport: Principles, Algorithms, and Applications, Springer, Berlin, 2005, pp. 131–154.

[28] M. Möller, D. Kuzmin, Adaptive mesh refinement for high-resolution finite element schemes, Int. J. Numer. Meth. Fluids, 52 (2006) 545–569.

[29] M. Möller, D. Kuzmin, D. Kourounis, Implicit FEM-FCT algorithms and discrete Newton methods for transient convection problems, Int. J. Numer. Meth. Fluids, in press.

[30] A. K. Parrott, M. A. Christie, FCT applied to the 2-D finite element solution of tracer transport by single phase flow in a porous medium, in: Numerical Methods for Fluid Dynamics, Oxford Univ. Press, London, 1986, pp. 609-619.

[31] A. Quarteroni, A. Valli, Numerical Approximation of Partial Differential Equations, Springer, Berlin, 1994.

[32] P. van Slingerland, An accurate and robust finite volume method for the advection-diffusion equation, M. Sc. Thesis, Delft University of Technology, June 2007.

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

[34] R. S. Varga, Matrix Iterative Analysis. Prentice-Hall, Englewood Cliffs, 1962.

[35] P. R. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, J. Comput. Phys. 54 (1984) 115–173.

[36] D. Young, Iterative Solution of Large Linear Systems, Academic Press, New York, 1971.

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

[38] S. T. Zalesak, The design of Flux-Corrected Transport (FCT) algorithms for structured grids, in: D. Kuzmin, R. Löhner, S. Turek (Eds.), Flux-Corrected Transport: Principles, Algorithms, and Applications, Springer, Berlin, 2005, pp. 29–78.

## A Finite element discretization

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

$$\int_\Omega w \left[ \frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{v}u) \right] \mathrm{d}\mathbf{x} = 0, \qquad \forall w. \tag{1}$$

The finite element approach to discretization of this variational problem is based on the following representation of the approximate solution

$$u \approx \sum_j u_j \varphi_j, \tag{2}$$

where $\varphi_j$ denotes a piecewise-polynomial basis function with compact support.

It is convenient to approximate the flux function $\mathbf{f}(u) = \mathbf{v}u$ in the same way

$$\mathbf{f}(u) \approx \sum_j \mathbf{f}_j \, \varphi_j. \tag{3}$$

This approach to approximation of (possibly nonlinear) flux functions was proposed by Fletcher [10] who called it the *group finite element formulation.*

Substitution of (2)–(3) into (1) with $w = \varphi_i$ yields a system of equations

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

which can be written in matrix form as follows

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

The coefficients of the matrices $M_C = \{m_{ij}\}$ and $K = \{k_{ij}\}$ are given by

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

where the coefficients $\mathbf{c}_{ij}$ correspond to the discretized space derivatives

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

If the mesh is fixed, then $m_{ij}$ and $\mathbf{c}_{ij}$ need to be evaluated just once during the initialization step. The coefficients $k_{ij}$ depend on the velocity field which may be time-dependent. In this case, the matrix $K$ can be updated in an efficient way using the above definition of $k_{ij}$ instead of costly numerical integration.

The coefficients $\mathbf{c}_{ij}$ can also be used to approximate derived quantities, e.g.,

$$[\nabla \cdot \mathbf{v}]_i \approx \frac{1}{m_i} \sum_j \mathbf{v}_j \cdot \mathbf{c}_{ij} = -\frac{1}{m_i} \sum_j k_{ij}, \tag{8}$$

where $m_i = \sum_j m_{ij}$ is a diagonal entry of the lumped mass matrix. This approximation technique corresponds to a lumped-mass $L_2$-projection [28].

Since the row sums of $K$ are related to the discrete divergence of the velocity field, it is instructive to consider the following decomposition of $Ku$

$$\sum_j k_{ij} u_j = \sum_{j \neq i} k_{ij}(u_j - u_i) + u_i \sum_j k_{ij}. \tag{9}$$

Due to (8), the approximations associated with each term are as follows

$$\sum_j k_{ij} u_j \approx -m_i [\nabla \cdot (\mathbf{v}u)]_i, \tag{10}$$

$$\sum_{j \neq i} k_{ij}(u_j - u_i) \approx -m_i [\mathbf{v} \cdot \nabla u]_i, \tag{11}$$

$$u_i \sum_j k_{ij} \approx -m_i u_i [\nabla \cdot \mathbf{v}]_i. \tag{12}$$

If the matrix $K$ has zero row sums, then the ‘compressible’ part (12) vanishes and $Ku = K(u + c)$ for any additive constant $c$. Artificial diffusion does not affect this property since discrete diffusion operators have zero row sums.