Algebraic Flux Correction I
Scalar Conservation Laws

Dmitri Kuzmin

Abstract This chapter is concerned with the design of high-resolution finite element schemes satisfying the discrete maximum principle. The presented *algebraic flux correction* paradigm is a generalization of the flux-corrected transport (FCT) methodology. Given the standard Galerkin discretization of a scalar transport equation, we decompose the antidiffusive part of the discrete operator into numerical fluxes and limit these fluxes in a conservative way. The purpose of this manipulation is to make the antidiffusive term local extremum diminishing. The available limiting techniques include a family of implicit FCT schemes and a new linearity-preserving limiter which provides a unified treatment of stationary and time-dependent problems. The use of Anderson acceleration makes it possible to design a simple and efficient quasi-Newton solver for the constrained Galerkin scheme. We also present a linearized FCT method for computations with small time steps. The numerical behavior of the proposed algorithms is illustrated by a grid convergence study for convection-dominated transport problems and anisotropic diffusion equations.

1 Introduction

A major bottleneck in finite element simulation of transport phenomena is the inability of the standard Galerkin discretization to satisfy the relevant maximum principles and/or maintain positivity on general meshes. This deficiency manifests itself in spurious undershoots and overshoots that pop up in regions of insufficient mesh resolution. Discontinuous weak solutions to hyperbolic conservation laws are particularly difficult to compute using continuous finite elements. The Galerkin "best approximations" to elliptic and parabolic transport equations may also exhibit non-

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

physical artifacts in proximity to unresolved small-scale features [51, 53]. An effective remedy to this problem must be found when it comes to the development of general-purpose finite element codes for Computational Fluid Dynamics.

Many modern high-resolution schemes for the equations of fluid mechanics are based on the flux-corrected transport (FCT) algorithm [6, 81] or use total variation diminishing (TVD) limiters [29, 78] to enforce the discrete maximum principle. The basic idea boils down to using a high-order scheme in smooth regions and a nonoscillatory low-order scheme elsewhere. The implementation of FCT and TVD in explicit finite element codes dates back to the late 1980s [3, 57, 58, 68, 71, 72]. The author and his coworkers developed the first implicit FCT schemes for continuous (linear and multilinear) finite elements [38, 48, 50, 42]. In the first edition of this book, we introduced the *algebraic flux correction* paradigm [46], a general framework for the design of multidimensional flux limiters. This approach leads to many useful generalizations of FCT and TVD-like methods [39, 44, 45].

The recent comparative study by John and Schmeyer [35] indicates that FEM-FCT is superior to mainstream stabilization techniques when it comes to solving unsteady convection problems with linear finite elements. However, flux correction of FCT type is inappropriate for steady-state computations since the results depend on the pseudo-time step, and severe convergence problems may occur. Flux limiters of TVD type [39, 40, 46, 49, 60] are free of these drawbacks but require mass lumping. As an alternative to FCT and TVD, we developed a linearity-preserving flux limiter that can handle stationary and time-dependent problems equally well [45]. In this algorithm, the same strategy is used to constrain the convective term, anisotropic diffusion, and the consistent mass matrix. Furthermore, linearity preservation implies consistency and second-order accuracy for smooth data [10, 62].

The cost of algebraic flux correction depends on the number of iterations required to obtain a converged solution. In our experience, this cost can be significantly reduced using a linearization of the antidiffusive term [42] or convergence acceleration techniques for iterative solvers. In particular, we recommend *Anderson mixing* [1, 18, 19, 80] (also known as *Anderson acceleration*) which combines a number of iterates in a GMRES-like fashion. As shown by Eyert [18], the accelerated solver belongs to the Broyden family of Jacobian-free quasi-Newton methods.

This chapter summarizes our work on algebraic flux correction schemes inspired by Zalesak's FCT algorithm [81]. Due to many recent developments, the presentation of this material differs considerably from the first edition of the book. In Section 2, we briefly review the continuous maximum principles for linear convection-diffusion equations. The discrete maximum principles and sufficient conditions of positivity preservation are formulated in Section 3. In Sections 4 and 5, we analyze the standard Galerkin discretization and explain the philosophy behind algebraic flux correction. The generalized FCT algorithm and the linearity-preserving flux limiter are presented in Sections 6 and 7, respectively. In Section 8, we address the design and acceleration of iterative solvers for the nonlinear system. A grid convergence study for 2D test problems is presented in Section 9. Finally, we summarize the results and outline some promising directions for further research.

2 Analysis of the Continuous Problem

The model problem that will serve as a vehicle for the presentation of our high-resolution finite element schemes is the linear convection-diffusion equation

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

which describes the transport of a conserved scalar quantity $u(\mathbf{x},t)$ in a bounded domain $\Omega \subset \mathbb{R}^d$, $d \in \{1,2,3\}$. The velocity $\mathbf{v}$ and diffusion tensor D are given.

If all terms are present, equation (1) is parabolic Also of interest are steady-state solutions $\left(\frac{\partial u}{\partial t} = 0\right)$ as well as the limiting cases of pure convection ($\mathsf{D} = 0$) and pure diffusion ($\mathbf{v} = 0$). The PDE type for each model is listed in Table 1.

Parabolic type	$\frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{v}u - \mathsf{D}\nabla u) = 0$	$\frac{\partial u}{\partial t} - \nabla \cdot (\mathsf{D}\nabla u) = 0$
Elliptic type	$\nabla \cdot (\mathbf{v}u - \mathsf{D}\nabla u) = 0$	$-\nabla \cdot (\mathsf{D}\nabla u) = 0$
Hyperbolic type	$\frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{v}u) = 0$	$\nabla \cdot (\mathbf{v}u) = 0$

Table 1 Taxonomy of scalar transport equations.

In the case of unsteady transport, we prescribe an initial condition of the form

$$u(\mathbf{x},0) = u_0(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega. \tag{2}$$

The Dirichlet-Neumann boundary conditions for our model problem are given by

$$u = u_D \qquad \text{on } \Gamma_D, \tag{3}$$

$$\mathbf{n} \cdot \nabla u = 0 \qquad \text{on } \Gamma_N, \tag{4}$$

where $\mathbf{n}$ is the unit outward normal to the boundary $\Gamma = \partial\Omega$. In the presence of diffusion, we have $\Gamma_D \cup \Gamma_N = \Gamma$. In the hyperbolic case, we have $\Gamma_N = \emptyset$ and

$$\Gamma_D = \{\mathbf{x} \in \Gamma \,|\, \mathbf{v} \cdot \mathbf{n} < 0\}.$$

Definition 1. Let Σ be the set of points where initial/boundary conditions are prescribed, i.e., $\Sigma := \Gamma_D \cup \Gamma_N$ in the steady case and $\Sigma := \{(\mathbf{x},t) \mid \mathbf{x} \in \Gamma_D \cup \Gamma_N \vee t = 0\}$ in the unsteady case. The (continuous) *maximum principle* holds if

$$\min_{\Sigma} u \le u \le \max_{\Sigma} u. \tag{5}$$

If the initial and boundary conditions are nonnegative, then the maximum principle implies that $u \ge 0$. This yields another useful a priori estimate of u in terms of $u|_\Sigma$.

Definition 2. The solution of a scalar transport equation is *positivity-preserving* if

$$\min_{\Sigma} u \geq 0 \quad \Rightarrow \quad u \geq 0. \tag{6}$$

At the continuous level, the solution of a scalar transport equation without sources or sinks is always positivity-preserving but a proof of the maximum principle is available only for the case of an incompressible velocity field ($\nabla \cdot \mathbf{v} = 0$).

Theorem 1. *The following a priori estimates hold for all PDEs listed in Table 1*

(i) $\nabla \cdot \mathbf{v} = 0 \quad \Rightarrow \quad \min_{\Sigma} u \leq u \leq \max_{\Sigma} u$ *(maximum principle)*

(ii) $u|_{\Sigma} \geq 0 \quad \Rightarrow \quad u \geq 0$ *(positivity preservation)*

A formal proof of this Theorem for each PDE type, its generalization to equations with source terms, and some useful corollaries can be found, e.g., in [44].

The maximum principle and positivity preservation are important for several reasons. On the one hand, the a priori bounds may represent certain physical constraints. For example, concentrations of chemical species are known to lie between 0 and 1. On the other hand, some useful information about the solutions of differential solutions becomes available, although these solutions are generally unknown. Upper/lower bounds, uniqueness proofs, and comparison principles can be obtained using elementary calculus. Last but not least, discrete maximum principles play an important role in the development of numerical methods for transport equations.

3 Analysis of the Discrete Problem

Of course, a good numerical scheme must respect the known properties of exact solutions. In this section, we review algebraic constraints which imply a discrete maximum principle and/or ensure positivity preservation. In the next sections, we will use these sufficient conditions to constrain the Galerkin discretization of the unsteady convection-diffusion equation (1). We tacitly assume that one or two terms in this equation may be missing, so that it represents all models listed in Table 1.

3.1 Semi-Discrete Problem

Any space discretization of (1) produces a system of differential algebraic equations

$$M\frac{\mathrm{d}u}{\mathrm{d}t} = Qu, \tag{7}$$

where $u(t)$ is the vector of time-dependent nodal values, $M = \{m_{ij}\}$ is the so-called *mass matrix*, and $Q = \{q_{ij}\}$ is the discrete transport operator. The properties of M and Q depend on the computational mesh and on the discretization method.

As in the continuous case, the unknown solution values are known to be bounded under certain assumptions. The semi-discrete equation for u_i reads

$$\sum_j \left(m_{ij} \frac{\mathrm{d}u_j}{\mathrm{d}t} \right) = \sum_j q_{ij} u_j. \tag{8}$$

To prevent spurious undershoots/overshoots, we impose the following constraints which imply a semi-discrete maximum principle and positivity preservation.

Theorem 2. *Consider a semi-discrete scheme of the form (8). Suppose that*

$$m_{ii} > 0, \quad m_{ij} = 0, \quad q_{ij} \geq 0, \quad \forall j \neq i. \tag{9}$$

Then the following a priori estimates hold for the solution value u_i

(i) $\quad \sum_j q_{ij} = 0 \quad \wedge \quad u_i \geq u_j, \quad \forall j \neq i \quad \Rightarrow \quad \frac{\mathrm{d}u_i}{\mathrm{d}t} \leq 0$ *(semi-DMP)*

(ii) $\quad u_j(0) \geq 0, \quad \forall j \quad \Rightarrow \quad u_i(t) \geq 0, \quad \forall t > 0$ *(positivity preservation)*

Proof. A comparison of Theorems 1 and 2 reveals that the zero row sum property is a discrete version of the incompressibility constraint. If $\sum_j q_{ij} = 0$, then

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = \frac{1}{m_{ii}} \sum_{j \neq i} q_{ij} (u_j - u_i). \tag{10}$$

Suppose that $u_i = \max_j u_j$. By assumption, we have $m_{ii} > 0$ and $q_{ij}(u_j - u_i) \leq 0$, $\forall j \neq i$. Thus $\frac{\mathrm{d}u_i}{\mathrm{d}t} \leq 0$, i.e., a maximum cannot increase. This proves (i).

To prove (ii), suppose that $u_i(t) = 0$ and $u_j(t) \geq 0$ for all $j \neq i$. It follows that

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = \frac{1}{m_{ii}} \sum_{j \neq i} q_{ij} u_j, \tag{11}$$

where $q_{ij} u_j \geq 0$, $\forall j \neq i$. Thus, the solution value u_i cannot become negative. □

Definition 3. A space discretization of the form (10) with $m_{ii} > 0$ and $q_{ij} \geq 0$ for all i and $j \neq i$ is called *local extremum diminishing* (LED).

The LED criterion was introduced by Jameson [32, 33] in the context of finite volume methods for unstructured grids. It is consistent with the FCT philosophy [6]: *no new extrema can form and existing extrema cannot grow*. The word *local* refers to the fact that the coefficient matrices are sparse, so only the nearest neighbors of node i make a nonzero contribution to (8) and define the bounds for u_i.

It is easy to prove that a LED scheme is total variation diminishing (TVD) in 1D. By the Godunov theorem [24], a linear positivity-preserving / LED discretization of a hyperbolic transport equation can be at most first-order accurate. The order barrier for a linear LED approximation of the diffusive term is 2 (see [31], pp. 118-120). Hence, the conditions of Theorem 2 are very restrictive in the linear case but they turn out to be a handy tool for the design of *nonlinear* high-resolution schemes.

3.2 Fully Discrete Problem

The fully discrete counterpart of problem (7) is a sparse linear system of the form

$$\begin{pmatrix} A_\Omega & A_\Gamma \\ 0 & I \end{pmatrix} \begin{pmatrix} u_\Omega \\ u_\Gamma \end{pmatrix} = \begin{pmatrix} B_\Omega & B_\Gamma \\ 0 & I \end{pmatrix} \begin{pmatrix} g_\Omega \\ g_\Gamma \end{pmatrix}, \tag{12}$$

where I is the identity matrix and $u_\Gamma = g_\Gamma$ is the vector of Dirichlet boundary values.

For a two-level time-stepping scheme, $u_\Omega = u_\Omega^{n+1}$ is the vector of unknowns and $g_\Omega = u_\Omega^n$ is the vector of solution values from the last time step. For stationary problems $B_\Omega = 0$ and $B_\Gamma = 0$. The general form of the $i-$th equation reads

$$a_{ii}u_i = b_{ii}g_i + \sum_{j \in S_i} (b_{ij}g_j - a_{ij}u_j), \tag{13}$$

where $S_i := \{j \neq i \mid a_{ij} \neq 0 \ \vee \ b_{ij} \neq 0\}$ is the set of nearest neighbors of node i.

Definition 4. The solution to (13) satisfies the *local* discrete maximum principle if

$$u_i^{\min} \leq u_i \leq u_i^{\max}, \tag{14}$$

where

$$u_i^{\max} := \max\{\max_{j \in S_i \cup \{i\}} g_j, \max_{k \in S_i} u_k\}, \tag{15}$$

$$u_i^{\min} := \min\{\min_{j \in S_i \cup \{i\}} g_j, \min_{k \in S_i} u_k\} \tag{16}$$

are the largest and smallest solution values that appear in the right-hand side of (13).

The so-defined local DMP implies that u_i should not decrease as result of increasing any other nodal value that contributes to the discretized equation for node i [69]. Conversely, u_i should not increase if another nodal value is decreased, all other things being fixed. If the given solution values are all nonnegative, then so is u_i.

Definition 5. The solution to (13) is said to be *locally* positivity-preserving if

$$u_i^{\min} \geq 0 \quad \Rightarrow \quad u_i \geq 0. \tag{17}$$

The following Theorem presents sufficient conditions of local positivity preservation and an additional constraint which guarantees the validity of (14).

Theorem 3. *Suppose that the coefficients of the discrete problem (13) satisfy*

$$a_{ii} > 0, \quad b_{ii} \geq 0, \quad a_{ij} \leq 0, \quad b_{ij} \geq 0, \qquad \forall j \in S_i. \tag{18}$$

Then the following a priori estimates hold for the solution value u_i

(i) $\sum_j a_{ij} = \sum_j b_{ij} \quad \Rightarrow \quad u_i^{\min} \leq u_i \leq u_i^{\max}$ *(local DMP)*

(ii) $u_i^{\min} \geq 0 \quad \Rightarrow \quad u_i \geq 0.$ *(positivity preservation)*

Proof. To prove (i), we define $w_j := u_j - u_i^{\max}$ and $v_j := g_j - u_i^{\max}$ such that

$$w_j \le 0, \quad \forall j \in S_i, \qquad v_j \le 0, \quad \forall j \in S_i \cup \{i\}. \tag{19}$$

Using the row sum condition $\sum_j a_{ij} = \sum_j b_{ij}$, we can express (13) as follows:

$$a_{ii} w_i = b_{ii} v_i + \sum_{j \in S_i} (b_{ij} v_j - a_{ij} w_j). \tag{20}$$

By (18) and (19), the right-hand side of (20) is nonpositive. Since $a_{ii} > 0$, we have $w_i \le 0$ or, equivalently, $u_i \le u_i^{\max}$. The proof for $u_i \ge u_i^{\min}$ is similar.

To prove (ii), suppose that $u_i^{\min} \ge 0$, i.e., $u_j \ge 0$, $\forall j \in S_i$ and $g_j \ge 0$, $\forall j \in S_i \cup \{i\}$. By (18), the right-hand side of (13) is nonnegative, so $a_{ii} > 0 \;\Rightarrow\; u_i \ge 0$. □

The Theorem implies that u_i is bounded by the solution values in a neighborhood of node i. If the local DMP holds for all nodes, then global maxima and minima must occur on the Dirichlet boundary or at the previous time level. Likewise, local positivity preservation for all nodes implies global positivity preservation.

Definition 6. The solution to (12) satisfies the *global* discrete maximum principle if

$$\min g \le u \le \max g, \tag{21}$$

where u denotes the vector of unknowns and g is the vector of given solution values.

Definition 7. The solution to (12) is said to be *globally* positivity-preserving if

$$g \ge 0 \quad \Rightarrow \quad u \ge 0. \tag{22}$$

A typical proof of (21) and (22) is based on the theory of monotone matrices [79].

Definition 8. A regular matrix A is called *monotone* if $A^{-1} \ge 0$ or, equivalently, if

$$u \ge 0 \quad \Rightarrow \quad Au \ge 0.$$

Definition 9. A monotone matrix A with $a_{ij} \le 0$, $\forall j \ne i$ is called an *M-matrix*.

Theorem 4. *Consider a fully discrete scheme of the form $Au = Bg$. Suppose that the coefficients of $A = \{a_{ij}\}$ and $B = \{b_{ij}\}$ satisfy conditions (18) for all i.*

If A is strictly or irreducibly diagonally dominant, then A is an M-matrix and

(i) the global DMP holds if $\sum_j a_{ij} = \sum_j b_{ij}$, $\forall i$

(ii) the scheme is globally positivity-preserving

Proof. We refer to Varga [79] for a proof of the M-matrix property. To prove the global DMP, define the vectors $w := u - \max g$ and $v := g - \max g$. Invoking (20), we obtain a linear system of the form $Aw = Bv$, where A is monotone, $B \ge 0$, and $v \le 0$. Hence $w = A^{-1}Bv \le 0$, which implies $u \le \max g$. Similarly, the solution to $Au = Bg$ proves positivity-preserving since $u = A^{-1}Bg \ge 0$ whenever $g \ge 0$. □

4 Galerkin Discretization

Some finite element approximations are known to satisfy the conditions of Theorems 3 and 4 unconditionally or under mild restrictions on the geometric properties of the mesh (no obtuse angles, no thin elements) [13, 21, 36, 37]. However, these sufficient conditions become too restrictive in the case of high-order finite elements, convection-dominated transport equations, and anisotropic diffusion problems. The usual remedy is to add a certain amount of artificial diffusion in order to compensate the contribution of matrix entries that have a wrong sign. Many shock capturing techniques, including algebraic flux correction, are based on this approach.

We will explain the principles of algebraic flux correction in the finite element context. To begin with, let us discretize the generic transport equation (1) using the (continuous) Galerkin approximation which delivers optimal accuracy in smooth regions but tends to produce spurious undershoots and overshoots elsewhere.

The variational form of our Dirichlet-Neumann boundary value problem reads

$$\int_\Omega w\left(\frac{\partial u}{\partial t}+\nabla\cdot(\mathbf{v}u)\right)\mathrm{d}\mathbf{x}+\int_\Omega \nabla w\cdot(\mathsf{D}\nabla u)\,\mathrm{d}\mathbf{x}=0 \tag{23}$$

for all admissible test functions w vanishing on the Dirichlet boundary Γ_D. We assume sufficient regularity without giving a formal definition of Sobolev spaces.

Let $\{\varphi_j\}$ be a finite set of piecewise-linear or multilinear basis functions. The numerical solution $u_h \approx u$ is defined as a linear combination thereof

$$u_h=\sum_j u_j\varphi_j. \tag{24}$$

The unknown degrees of freedom are the coefficients u_j which represent the (possibly time-dependent) values of u_h at the vertices of the mesh.

Instead of differentiating the convective flux, we replace it with the interpolant

$$(\mathbf{v}u)_h=\sum_j(\mathbf{v}_j u_j)\varphi_j, \tag{25}$$

where $\mathbf{v}_j$ denotes the velocity at node j. This approach is known as the *group finite element* formulation [22, 23]. The divergence of (25) is given by

$$\nabla\cdot(\mathbf{v}u)_h=\sum_j u_j(\mathbf{v}_j\cdot\nabla\varphi_j). \tag{26}$$

The contribution of the diffusive flux is evaluated using the consistent gradient

$$\nabla u_h=\sum_j u_j\nabla\varphi_j. \tag{27}$$

To obtain a semi-discrete equation for the solution value u_i, substitute approximations (24), (26), and (27) into (23) with the test function $w_h:=\varphi_i$. This gives

$$\sum_j \left(\int_\Omega \varphi_i \varphi_j \, \mathrm{d}\mathbf{x} \right) \frac{\mathrm{d}u_j}{\mathrm{d}t} = -\sum_j \mathbf{v}_j \cdot \left(\int_\Omega \varphi_i \nabla \varphi_j \, \mathrm{d}\mathbf{x} \right) u_j - \sum_j \left(\int_\Omega \nabla \varphi_i \cdot (\mathsf{D} \nabla \varphi_j) \, \mathrm{d}\mathbf{x} \right) u_j. \tag{28}$$

The resultant semi-discrete problem can be written in the generic matrix form

$$M_C \frac{\mathrm{d}u}{\mathrm{d}t} = (K - L)u, \tag{29}$$

where $M_C = \{m_{ij}\}$ denotes the consistent mass matrix, $K = \{k_{ij}\}$ is the convective part of the discrete transport operator, and $L = \{l_{ij}\}$ is the contribution of the diffusive term. By (28) the coefficients of the three matrices are given by

$$m_{ij} = \int_\Omega \varphi_i \varphi_j \, \mathrm{d}\mathbf{x}, \qquad l_{ij} = \int_\Omega \nabla \varphi_i \cdot (\mathsf{D} \nabla \varphi_j) \, \mathrm{d}\mathbf{x}, \tag{30}$$

$$k_{ij} = -\mathbf{v}_j \cdot \mathbf{c}_{ij}, \qquad \mathbf{c}_{ij} = \int_\Omega \varphi_i \nabla \varphi_j \, \mathrm{d}\mathbf{x}. \tag{31}$$

In the case of an unsteady velocity field, the convective part K must be updated at each time step. If the mesh is fixed, then the coefficients $\mathbf{c}_{ij}$ of the discrete gradient operator do not change and need to be evaluated just once. Hence, the group finite element formulation makes it possible to update K in a very efficient way.

Let $0 = t_0 < t^1 < t^2 < \cdots < t^M = T$ be a sequence of discrete time levels for the time integration of (29). For simplicity, we assume that the time step $\Delta t := t^{n+1} - t^n$ is constant so that $t^n = n\Delta t$. By the Fundamental Theorem of Calculus

$$M_C(u^{n+1} - u^n) = \int_{t^n}^{t^{n+1}} (K - L)u \, \mathrm{d}t.$$

The integral is approximated using a suitable quadrature rule. In particular, we will consider the fully discrete problem for the standard θ-scheme

$$[M_C - \theta \Delta t(K - L)]u^{n+1} = [M_C + (1 - \theta)\Delta t(K - L)]u^n, \tag{32}$$

where $\theta \in [0, 1]$ is the degree of implicitness. The forward Euler ($\theta = 0$) version is unstable for convection-dominated transport problems and gives rise to severe time step restrictions in the case of dominating diffusion. For this reason, we restrict ourselves to the unconditionally stable Crank-Nicolson ($\theta = \frac{1}{2}$) and backward Euler ($\theta = 1$) time stepping. If a fully explicit treatment is desired, we recommend the family of strong stability-preserving Runge-Kutta methods [25, 26] which guarantee the local and global DMP if the underlying space discretization is LED and the time steps are sufficiently small. Other explicit schemes can generate spurious oscillations even if the space discretization satisfies the conditions of Theorem 2.

5 Algebraic Flux Correction

The fully discrete scheme (32) is a linear system of the form $Au^{n+1} = Bu^n$. The diagonal entries of the matrices A and B are positive, at least for sufficiently small time steps Δt. However, a violation of the DMP conditions (18) may be caused by

- positive off-diagonal entries of the consistent mass matrix M_C;
- negative off-diagonal entries of the discrete convection operator K;
- positive off-diagonal entries of the discrete diffusion operator L.

In the process of algebraic flux correction, we constrain the contribution of these entries trying to stay as close as possible to the original Galerkin discretization.

The "good " part of the Galerkin scheme (29) is an ODE system of the form

$$M_L \frac{\mathrm{d}u}{\mathrm{d}t} = (\tilde{K} - \tilde{L})u, \tag{33}$$

where M_L and $\tilde{K} - \tilde{L}$ satisfy the conditions of Theorem 2. We define these matrices in Section 5.1. The "bad" *antidiffusive* part of (29) is given by

$$f(u) = (M_L - M_C)\frac{\mathrm{d}u}{\mathrm{d}t} + (K - \tilde{K})u - (L - \tilde{L})u. \tag{34}$$

To prevent a possible violation of the (semi-)discrete maximum principle, we decompose the antidiffusive term $f(u)$ into numerical fluxes and limit the magnitude of these fluxes in regions where they threaten to create an undershoot or overshoot. To this end, each flux is multiplied by a solution-dependent correction factor. In contrast to mainstream stabilization techniques for finite elements, there are no free parameters. The constrained Galerkin scheme is guaranteed to be positivity-preserving and satisfy the DMP if it holds for the solution of the continuous problem.

In this section, we review the design philosophy behind algebraic flux correction. Some generalizations [42, 48, 50] of the multidimensional FCT algorithm [81] and a new linearity-preserving flux limiter [45, 51] are presented in the next section.

5.1 Artificial Diffusion Operators

The derivation of (33) begins with *row-sum mass lumping*. In explicit finite element codes, the mass matrix M_C is frequently replaced with the diagonal approximation

$$M_L := \mathrm{diag}\{m_i\}, \qquad m_i = \sum_j m_{ij}. \tag{35}$$

For linear finite elements, this conservative modification is equivalent to inexact evaluation of M_C with a low-order Newton-Cotes quadrature rule [28].

The negative off-diagonal entries of the (nonsymmetric) convection operator K are eliminated by adding a suitably designed artificial diffusion operator D.

Definition 10. A symmetric matrix $D = \{d_{ij}\}$ is called a *discrete diffusion operator* if D has zero row and column sums [48]. That is,

$$d_{ij} = d_{ji}, \qquad \sum_j d_{ij} = \sum_i d_{ij} = 0. \tag{36}$$

To make sure that $\tilde{K} := K + D$ has no negative off-diagonal entries, we define

$$d_{ij} := \max\{-k_{ij}, 0, -k_{ji}\}, \qquad \forall j \neq i. \tag{37}$$

Remark 1. Artificial diffusion coefficients that enforce positivity in this way were used to construct low-order schemes for FCT as early as in the mid-1970s [8].

Definition (37) implies $d_{ij} = d_{ji}$. To comply with the zero row sum condition, let

$$d_{ii} := -\sum_{j \neq i} d_{ij}. \tag{38}$$

By symmetry, the column sums are also equal to zero, so D satisfies conditions (36).

In practice, there is no need to assemble the global matrix D. Instead, artificial diffusion can be built into K in a loop over the edges of its sparsity graph. By definition, each edge is a pair of nodes $\{i, j\}$ that corresponds to a pair of nonzero off-diagonal coefficients k_{ij} and k_{ji}. The required update is as follows:

$$\begin{array}{ll} k_{ii} := k_{ii} - d_{ij}, & k_{ij} := k_{ij} + d_{ij}, \\ k_{ji} := k_{ji} + d_{ij}, & k_{jj} := k_{jj} - d_{ij}. \end{array} \tag{39}$$

Without loss of generality, the edges of the sparsity graph are oriented so that

$$k_{ij} \leq k_{ji}. \tag{40}$$

This orientation convention implies that node i is located 'upwind' and corresponds to the row number of the negative off-diagonal entry to be eliminated.

Physical diffusion can be taken into account before or after the assembly of D. If some off-diagonal entries of L are strictly positive, we split it into the antidiffusive part $L^+ = \{l_{ij}^+\}$ and the remainder $\tilde{L} := L - L^+$. The entries of L^+ are given by

$$l_{ii}^+ := -\sum_{j \neq i} l_{ij}^+, \qquad l_{ij}^+ := \max\{0, l_{ij}\}, \quad \forall j \neq i. \tag{41}$$

The conservative elimination of l_{ij}^+ can also be performed edge-by-edge

$$\begin{array}{ll} l_{ii} := l_{ii} + l_{ij}^+, & l_{ij} := l_{ij} - l_{ij}^+, \\ l_{ji} := l_{ji} - l_{ij}^+, & l_{jj} := l_{jj} + l_{ij}^+. \end{array} \tag{42}$$

If all off-diagonal entries of L are nonpositive, then $L^+ = 0$ and $\tilde{L} = L$. However, the standard Galerkin approximation may fail to satisfy the DMP conditions if the mesh and/or the diffusion tensor are highly anisotropic [53]. In this case, $\tilde{L}$ is a

monotone but possibly inconsistent approximation to L. The lack of consistency must be compensated in the course of flux correction (see Section 7).

Example 1. To clarify the implications of (39), consider the 1D convection equation

$$\frac{\partial u}{\partial t}+v\frac{\partial u}{\partial x}=0 \qquad \text{in} \quad \Omega=(0,1), \tag{43}$$

where v is a positive constant. The inflow boundary condition is given by

$$u(0)=g.$$

On a uniform mesh of linear finite elements, the standard Galerkin method yields

$$K=\frac{1}{2}\begin{pmatrix} \cdots & & & & & & \\ & v & 0 & -v & & & \\ & & v & 0 & -v & & \\ & & & v & 0 & -v & \\ & & & & & & \cdots \end{pmatrix}.$$

For any interior node, $m_i=\Delta x$, where Δx is is the constant mesh size. Hence, the lumped-mass version of (29) is equivalent to the central difference scheme

$$\frac{\mathrm{d}u_i}{\mathrm{d}t}+v\,\frac{u_{i+1}-u_{i-1}}{2\,\Delta x}=0.$$

Since $k_{ij}=-\frac{v}{2}$ for $j=i+1$, the artificial diffusion coefficient (37) is $d_{ij}=\frac{v}{2}$ and

$$\tilde{K}=\begin{pmatrix} \cdots & & & & & \\ & v & -v & 0 & & \\ & & v & -v & 0 & \\ & & & v & -v & 0 \\ & & & & & \cdots \end{pmatrix},$$

which corresponds to the first-order accurate upwind difference approximation

$$\frac{\mathrm{d}u_i}{\mathrm{d}t}+v\,\frac{u_i-u_{i-1}}{\Delta x}=0.$$

Thus, the elimination of negative off-diagonal entries from a skew-symmetric operator K can be interpreted as *discrete upwinding* [46]. For any pair of nodes i and $j=i+1$ numbered in accordance with (40), the grid point x_i lies upstream of x_j.

After the discretization in time by the standard $\theta-$scheme, the upwind difference method proves positivity-preserving under the CFL-like condition [48]

$$v\frac{\Delta t}{\Delta x}\le\frac{1}{1-\theta}, \qquad 0\le\theta<1. \tag{44}$$

According to this formula, there is no time step restriction for the backward Euler method ($\theta=1$) which corresponds to first-order 'upwinding' in time.

5.2 Conservative Flux Decomposition

The replacement of the high-order Galerkin scheme (29) by the perturbed system (33) ensures positivity preservation but creates a lot of numerical diffusion. The next ingredient of an algebraic flux correction scheme is a decomposition of (34) into a sum of numerical fluxes. These antidiffusive fluxes enable us to remove artificial diffusion in regions where the Galerkin solution is sufficiently smooth.

The antidiffusive term (34) represents the difference between the residuals of systems (29) and (33). By definition of the matrices M_L, $\tilde{K}$, and $\tilde{L}$, we have

$$f(u) = (M_L - M_C)\frac{\mathrm{d}u}{\mathrm{d}t} - Du - L^+u. \tag{45}$$

By construction, the matrices $M_C - M_L$, D, and L^+ are discrete diffusion operators in the sense of Definition 10. Using the zero row sum property, we obtain

$$(M_Cu - M_Lu)_i = \sum_j m_{ij}u_j - u_i\sum_j m_{ij} = \sum_{j\neq i} m_{ij}(u_j - u_i), \tag{46}$$

$$(Du)_i = \sum_j d_{ij}u_j = \sum_{j\neq i} d_{ij}u_j + d_{ii}u_i = \sum_{j\neq i} d_{ij}(u_j - u_i), \tag{47}$$

$$(L^+u)_i = \sum_j l^+_{ij}u_j = \sum_{j\neq i} l^+_{ij}u_j + l^+_{ii}u_i = \sum_{j\neq i} l^+_{ij}(u_j - u_i). \tag{48}$$

The right-hand sides of (47)–(48) resemble that of a LED scheme. By symmetry, the components of the sums over $j \neq i$ can be interpreted as numerical fluxes that describe a conservative mass exchange between a pair of nodes. Let

$$f_{ij} = \left(m_{ij}\frac{\mathrm{d}}{\mathrm{d}t} + d_{ij} + l^+_{ij}\right)(u_i - u_j), \qquad \forall j \neq i \tag{49}$$

denote the *raw antidiffusive flux* from node j into node i. In the fully discrete version, the time derivative is replaced with a finite difference.

The net antidiffusion received by node i admits the following decomposition

$$f_i = \sum_{j\neq i} f_{ij}, \qquad f_{ji} = -f_{ij}. \tag{50}$$

Since $f_{ij} + f_{ji} = 0$ by definition, the antidiffusive term does not change the total mass of the discrete solution. The mass added to node i is subtracted from its neighbors.

5.3 Limited Antidiffusive Correction

Some of the raw antidiffusive fluxes f_{ij} are harmless but others may create an undershoot or overshoot. The contribution of these "bad" fluxes must be limited so as to keep the antidiffusive term local extremum diminishing for a given solution.

The flux-corrected counterpart of (29) is a semi-discrete problem of the form

$$M_L \frac{\mathrm{d}u}{\mathrm{d}t} = (\tilde{K} - \tilde{L})u + \bar{f}(u), \tag{51}$$

where the (nonlinear) term $\bar{f}(u)$ stands for the sum of limited antidiffusive fluxes

$$\bar{f}_i = \sum_{j \neq i} \bar{f}_{ij}, \qquad \bar{f}_{ji} = -\bar{f}_{ij}. \tag{52}$$

A well-designed flux limiter produces $\bar{f}_{ij} = f_{ij}$ in smooth regions and $\bar{f}_{ij} = 0$ in interior or boundary layers. The unconstrained Galerkin scheme (29) and its nonoscillatory part (33) correspond to $\bar{f} = f$ and $\bar{f} = 0$, respectively.

In general, the best definition of $\bar{f}_{ij}$ satisfying the LED constraint is given by the solution of a constrained optimization problem [5]. A nonoptimal but cost-effective alternative is the multiplication by a solution-dependent correction factor

$$\bar{f}_{ij} := \alpha_{ij} f_{ij}, \qquad 0 \leq \alpha_{ij} \leq 1. \tag{53}$$

This kind of flux correction traces its origins to the FCT algorithm and forms the basis for the construction of our algebraic flux correction schemes.

The following criterion guarantees that the antidiffusive term (52) is LED

$$\sum_{j \neq i} q_{ij} \min\{0, u_j - u_i\} \leq \sum_{j \neq i} \alpha_{ij} f_{ij} \leq \sum_{j \neq i} q_{ij} \max\{0, u_j - u_i\} \tag{54}$$

for a given set of bounded nonnegative coefficients q_{ij}. The upper and lower bounds may consist of a single term associated with a local maximum or minimum

$$u_i^{\max} := \max\{u_i, \max_{j \in S_i} u_j\}, \tag{55}$$

$$u_i^{\min} := \min\{u_i, \min_{j \in S_i} u_j\}. \tag{56}$$

Introducing $q_i := \sum_{j \neq i} q_{ij}$, we can replace (54) with the weakened LED constraint

$$q_i(u_i^{\min} - u_i) \leq \sum_{j \neq i} \alpha_{ij} f_{ij} \leq q_i(u_i^{\max} - u_i). \tag{57}$$

If u_i is a local maximum, then (54) and (57) imply the cancellation of all positive fluxes. Similarly, all negative fluxes are cancelled if u_i is a local maximum. Hence, the sum of $\bar{f}_{ij} := \alpha_{ij} f_{ij}$ cannot create an undershoot or overshoot at node i.

The above criteria provide a general framework for the design of algebraic flux correction schemes that differ in the definition of the LED bounds for the sum of limited antidiffusive fluxes. The best choice of q_{ij} and q_i is dictated by accuracy and efficiency considerations. Obviously, increasing the value of these parameters makes the bounds less restrictive. However, this may cause divergence of iterative solvers for the resultant nonlinear system. For accuracy reasons, it is essential to guarantee

that $\alpha_{ij} = 1$ is acceptable whenever the solution varies linearly in a neighborhood of node i. This design principle is called *linearity preservation* [5, 10, 62].

We use a generalization of Zalesak's FCT algorithm [81] to calculate α_{ij} satisfying (57). The same limiting strategy is used to enforce the LED bounds defined by (54) in algebraic flux correction schemes of TVD type [40, 44, 49]. In the following sections, we address the design of multidimensional flux limiters and the iterative solution of nonlinear systems produced by the constrained Galerkin schemes.

6 Generalized FCT Algorithms

FCT was the first nonlinear high-resolution scheme to be equipped with a flux limiter. The classical FCT algorithms of Boris, Book, and Hain [6, 8, 9] belong to the class of *diffusion-antidiffusion* (DAD) methods [14] that involve two steps:

1. Advance the solution in time with an explicit low-order scheme containing 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.

The numerical diffusion of the low-order method makes it possible to maintain positivity and improves the phase accuracy of an explicit appoximation to the convective term. The limited antidiffusive correction reduces the amplitude errors in a LED manner. In contrast to TVD methods [29, 78], the upper and lower bounds for the FCT limiter are defined in terms of the low-order predictor and designed to accept as much antidiffusion as possible without violating the positivity constraint.

Zalesak's fully multidimensional FCT algorithm [81] is based on blending explicit high- and low-order approximations so as to constrain the maximum and minimum increments to each nodal value. The work of Zalesak has formed the basis for the development of all algebraic flux correction schemes to be presented in this Chapter. The combination of FCT with finite elements and unstructured meshes dates back to the explicit algorithms of Parrott and Christie [68] and Löhner et al. [57, 58]. A number of implicit FEM-FCT schemes were published by the author and his coworkers [38, 42, 48, 50]. The rationale for the use of an implicit time discretization stems from the fact that the CFL stability condition becomes too restrictive in the case of nonuniform velocity fields and locally refined meshes.

In this section, we begin with a presentation of predictor-corrector FCT algorithms in which the antidiffusive fluxes are linearized about a provisional low-order solution. The linearized FCT scheme [42] is recommended for evolutionary problems that call for the use of small time steps. We also present the nonlinear version of this scheme which requires iterative flux correction. In particular, we describe an algorithm for 'recycling' the rejected antidiffusion step-by-step [50]. Finally, we summarize the pros and cons of the FCT approach to algebraic flux correction.

6.1 Linearized FCT Scheme

After the discretization in time by the two-level θ-scheme, the constrained Galerkin discretization (51) produces a nonlinear algebraic system of the form

$$Au^{n+1} = Bu^n + \bar{f}, \tag{58}$$

where $\bar{f} = \bar{f}(u^{n+1}, u^n)$ denotes the limited antidiffusive term. The matrices

$$A = \frac{1}{\Delta t} M_L - \theta(\tilde{K} - \tilde{L}) \tag{59}$$

and

$$B = \frac{1}{\Delta t} M_L + (1-\theta)(\tilde{K} - \tilde{L}) \tag{60}$$

represent the nonoscillatory low-order part of the original Galerkin scheme. If the governing equation is nonlinear or the velocity field is time-dependent, then the coefficients of A and B may change as the solution evolves.

If the time step Δt is relatively small, it is worthwhile to linearize (58) using a predictor-corrector strategy. At the first step of the linearized FCT algorithm [42], we disregard the antidiffusive term $\bar{f}$ and solve the linear system

$$Au^L = Bu^n. \tag{61}$$

By construction, the off-diagonal entries of $\tilde{K}$ and $\tilde{L}$ are nonnegative and nonpositive, respectively. The diagonal coefficients of these matrices have the opposite sign (except in the case of a strongly compressible velocity field). By Theorem 4, our low-order scheme (61) is positivity-preserving under the CFL-like condition

$$\Delta t \le \frac{1}{1-\theta} \frac{m_i}{\tilde{l}_{ii} - \tilde{k}_{ii}}, \qquad \forall i. \tag{62}$$

Furthermore, the discrete maximum principle holds if A and B have equal row sums.

Remark 2. Van Slingerland [75, 76] proposed a variable-order θ-scheme in which (62) is used to determine the optimal degree of implicitness $\theta_{ij} \in [0,1]$ individually for each pair of nodes. This approach requires a conservative flux decomposition not only for the antidiffusive term but also for the low-order operator.

Remark 3. The two-level θ-scheme can be replaced with any other time integration scheme, e.g., a strong stability-preserving (TVD) Runge-Kutta method [25, 26]. Clearly, the time step restriction will depend on the time integration method.

The low-order predictor u^L is used to evaluate the raw antidiffusive fluxes

$$f_{ij} = m_{ij}(\dot{u}_i^L - \dot{u}_j^L) + d_{ij}(u_i^L - u_j^L), \qquad j \neq i \tag{63}$$

where $\dot{u}^L$ is an approximation to the vector of time derivatives at the time level t^{n+1}.

For example, the semi-discrete low-order scheme (33) with $u := u^L$ yields

$$\dot{u}^L = M_L^{-1}[(\tilde{K} - \tilde{L})u^L]. \tag{64}$$

The so-defined approximation is smooth but diffusive. Another option is

$$\dot{u}^L = M_C^{-1}[(K - L)u^L]. \tag{65}$$

This formula follows from (29). The well-conditioned mass matrix M_C can be 'inverted' with 3-5 cycles of the preconditioned Richardson iteration [17, 42].

The raw antidiffusive fluxes f_{ij} are passed to the multidimensional FCT limiter (see Section 6.4) which returns a set of correction factors α_{ij}. This gives

$$\bar{f}_i = \sum_{j \neq i} \alpha_{ij} f_{ij}, \qquad 0 \leq \alpha_{ij} \leq 1. \tag{66}$$

After flux limiting, the final solution u^{n+1} is obtained with the explicit correction

$$M_L u^{n+1} = M_L u^L + \Delta t \bar{f}. \tag{67}$$

Remark 4. Due to the linearization about u^L, the unconstrained ($\alpha_{ij} := 1$) version of the above algorithm is no longer equivalent to the original Galerkin scheme.

6.2 Nonlinear FCT Scheme

Linearization errors are avoided if the nonlinear system (58) is solved in an iterative fashion. This approach leads to a FEM-FCT algorithm in which the antidiffusive fluxes f_{ij} and the corresponding correction factors α_{ij} are updated step-by-step until the residuals or relative changes become smaller than a prescribed tolerance.

Let $\{u^{(m)}\}$ be a sequence of successive approximations to the flux-corrected Galerkin solution u^{n+1}. A reasonable initial guess is $u^{(0)} = u^n$ or $u^{(0)} = 2u^n - u^{n-1}$. These settings correspond to the constant and linear extrapolation in time, respectively. Given the current iterate $u^{(m)}$ and the vector of approximate time derivatives

$$\dot{u}^{(m)} := \frac{u^{(m)} - u^n}{\Delta t}, \tag{68}$$

we recalculate the implicit part of the raw antidiffusive fluxes given by

$$\begin{aligned} f_{ij}^{(m)} &= m_{ij}(\dot{u}_i^{(m)} - \dot{u}_j^{(m)}) + \theta(d_{ij} + l_{ij}^+)(u_i^{(m)} - u_j^{(m)}) \\ &\quad + (1-\theta)(d_{ij} + l_{ij}^+)(u_i^n - u_j^n), \qquad j \neq i. \end{aligned} \tag{69}$$

Then we apply the FCT limiter (see Section 6.4) and solve the linear system

$$Au^{(m+1)} = Bu^n + \bar{f}^{(m)}. \tag{70}$$

Each solution update of the form (70) can be split into three steps [38, 42]

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

$$M_L\tilde{u}^{(0)} = Bu^n. \tag{71}$$

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

$$M_L\tilde{u}^{(m+1)} = M_L\tilde{u}^{(0)} + \Delta t\bar{f}^{(m)}. \tag{72}$$

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

$$Au^{(m+1)} = M_L\tilde{u}^{(m+1)}. \tag{73}$$

Note that the auxiliary solution $\tilde{u}^{(0)}$ is independent of the iteration number m, so it needs to be determined just once per time step (for $m = 0$). For its computation to be positivity-preserving, the time step Δt must satisfy (62). The flux limiting procedure presented in Section 6.4 guarantees that $\tilde{u}^{(0)} \geq 0 \;\Rightarrow\; \tilde{u}^{(m+1)} \geq 0$. The last solution update is positivity-preserving by the M-matrix property of A. Thus

$$u^{(0)} \geq 0 \quad \Rightarrow \quad \tilde{u}^{(0)} \geq 0 \quad \Rightarrow \quad \tilde{u}^{(m+1)} \geq 0 \quad \Rightarrow \quad u^{(m+1)} \geq 0 \tag{74}$$

provided that the CFL-like condition (62) holds for the given Δt and $\theta \in (0,1]$.

6.3 Iterative FCT Scheme

The implicit FCT scheme (71)–(73) 'forgets' the history of previous flux correction steps when it comes to the assembly of $\bar{f}^{(m)}$. Hence, it tends to reject more antidiffusion than necessary to enforce the positivity constraint. As shown by Schär and Smolarkiewicz [66], an iterative 'recycling' of the rejected antidiffusive fluxes may significantly improve the accuracy of an FCT algorithm in some cases.

An iterative limiting strategy for maximizing the amount of accepted antidiffusion in implicit FCT schemes was developed in [50]. Replacing (72) with

$$M_L\tilde{u}^{(m+1)} = M_L\tilde{u}^{(m)} + \Delta t\bar{f}^{(m)}, \tag{75}$$

we perform flux limiting in terms of $\tilde{u}^{(m)}$ rather than $\tilde{u}^{(0)}$. The sum of all previous corrections is built into $\tilde{u}^{(m)}$, so only the remainder of $f_{ij}^{(m)}$ needs to be limited

$$f_{ij}^{(m)} := f_{ij}^{(m)} - \sum_{k=0}^{m-1} \alpha_{ij}^{(k)} f_{ij}^{(k)} \tag{76}$$

for all $m > 0$. This simplifies the job of the flux limiter and enables it to accept more antidiffusion. For a detailed description of the algorithm, we refer to [50].

Iterative FCT is more accurate than (71)–(73) but converges very slowly. For this reason, we do not recommend its use unless it is justified by unusually stringent accuracy requirements. For many problems of practical interest, the predictor-corrector approach presented in Section 6.1 offers the best cost/accuracy ratio.

6.4 Zalesak's FCT Limiter

In this section, we present Zalesak's limiter [81] that we use to calculate the correction factors α_{ij} for all FCT schemes. Consider a solution update of the form

$$m_i u_i = m_i \tilde{u}_i + \Delta t \sum_{j \neq i} \alpha_{ij} f_{ij}, \tag{77}$$

where $\tilde{u}$ is a nonoscillatory intermediate solution. Let $u_i^{\max}$ and $u_i^{\min}$ denote the local extrema of $\tilde{u}$. The objective is to find the best value of α_{ij} such that

$$u_i^{\min} \leq u_i \leq u_i^{\max}. \tag{78}$$

This condition implies that (77) satisfies the local discrete maximum principle.

6.4.1 Prelimiting Step

The process of flux correction begins with the optional elimination of fluxes that have the same sign as $\tilde{u}_j - \tilde{u}_i$. Such fluxes flatten the solution profile instead of steepening it. As a consequence, the flux-corrected solution may exhibit spurious ripples within the bounds allowed by the limiter [15]. In the original Boris-Book limiter [6], a wrong sign is reversed, and the magnitude of the antidiffusive flux is limited in the usual way. This fix works well for discontinuities but may distort a smooth profile. A safer remedy is to cancel the "diffusive" fluxes by setting

$$f_{ij} := 0, \quad \text{if} \quad f_{ij}(\tilde{u}_j - \tilde{u}_i) > 0. \tag{79}$$

This optional adjustment is called *prelimiting* because it must be performed before the computation of the correction factors α_{ij} and flux limiting [15, 81].

Zalesak [81] argued that the effect of (79) is marginal and cosmetic in nature since the vast majority of antidiffusive fluxes have the right sign. This remark might have led many readers to disregard equations (14) and (14′) in [81]. Two decades later, the need for prelimiting of the form (79) was emphasized by DeVore [15] who explained its ramifications and demonstrated that it may lead to a marked improvement of accuracy. In our experience, prelimiting is particularly useful in the context of finite element approximations because the contribution of the consistent mass matrix may change the sign of the raw antidiffusive flux and render it diffusive. The cancellation of such outliers is essential for keeping the solution free of ripples.

6.4.2 Limiting Strategy

In accordance with the LED criterion (57), the choice of the correction factors α_{ij} should ensure that positive antidiffusive fluxes cannot create an overshoot, while negative ones cannot create an undershoot. Assuming the worst-case scenario, we enforce condition (78) using Zalesak's multidimensional FCT algorithm [81]:

1. Compute the sums of positive/negative antidiffusive fluxes into node i

$$P_i^+ = \sum_{j \neq i} \max\{0, f_{ij}\}, \qquad P_i^- = \sum_{j \neq i} \min\{0, f_{ij}\}. \tag{80}$$

2. Determine the distance to a local maximum/minimum and the bounds

$$Q_i^+ = \frac{m_i}{\Delta t}(u_i^{\max} - \tilde{u}_i), \qquad Q_i^- = \frac{m_i}{\Delta t}(u_i^{\min} - \tilde{u}_i). \tag{81}$$

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

$$R_i^+ = \min\left\{1, \frac{Q_i^+}{P_i^+}\right\}, \qquad R_i^- = \min\left\{1, \frac{Q_i^-}{P_i^-}\right\}. \tag{82}$$

4. Check the sign of the raw antidiffusive flux f_{ij} and multiply it by

$$\alpha_{ij} = \begin{cases} \min\{R_i^+, R_j^-\}, & \text{if } f_{ij} > 0, \\ \min\{R_i^-, R_j^+\}, & \text{if } f_{ij} < 0. \end{cases} \tag{83}$$

Remark 5. It is worthwhile to set $R_i^{\pm} := 1$ if a Dirichlet boundary condition is imposed at node i and, therefore, the value of u_i does not depend on α_{ij}.

The above definition of α_{ij} guarantees that sum of limited antidiffusive fluxes satisfies (57) with $q_i = \frac{m_i}{\Delta t}$. The LED property (78) follows from the estimate

$$u_i^{\min} = \tilde{u}_i + \frac{\Delta t}{m_i} Q_i^- \leq u_i \leq \tilde{u}_i + \frac{\Delta t}{m_i} Q_i^+ = u_i^{\max}.$$

The presence of the time step Δt in the denominator of $Q_i^{\pm}$ is a blessing or a curse, depending on the purpose of simulation. On the one hand, the LED constraints become less restrictive and, consequently, a larger portion of the raw antidiffusive flux f_{ij} is retained as the time step is refined. This makes FCT the method of choice for transient computations. On the other hand, the use of large Δt results in a loss of accuracy, and severe convergence problems may occur in the steady state limit.

6.4.3 Edge-Based Implementation

The practical implementation of Zalesak's FCT limiter depends on the employed data structures, storage techniques, and software development concepts. In the following pseudo-code (Algorithm 1), we take advantage of the fact that $f_{ji} = -f_{ij}$

and $\alpha_{ji} = \alpha_{ij}$. The flux sums $P_i^{\pm}$ and the corresponding bounds $Q_i^{\pm}$ are assembled in a loop over all neighbors $j \in S_i$ such that $j > i$. The values of $P_j^{\mp}$ and $Q_j^{\mp}$ are updated in the same j-loop. The nodal correction factors $R_i^{\pm}$ are evaluated in the next i-loop. When it comes to the assembly of the antidiffusive term, flux limiting is performed in another loop over $j > i$. The flux $\bar{f}_{ij} := \alpha_{ij} f_{ij}$ is added to $\bar{f}_i$ and subtracted from $\bar{f}_j$. This implementation of FCT calls for the use of edge-based data structures [4, 70, 73] which operate with pairs of nodes, just like finite volume schemes.

Algorithm 1: Edge-based implementation of FCT.

$P^{\pm} := 0,\ Q^{\pm} := 0,\ \bar{f} := 0$

For all i **do**

  For all $j \in S_i,\ j > i$ **do**

    $P_i^{\pm} := P_i^{\pm} + \overset{\max}{\min}\{0, f_{ij}\}$

    $P_j^{\pm} := P_j^{\pm} + \overset{\max}{\min}\{0, -f_{ij}\}$

    $Q_i^{\pm} := \overset{\max}{\min}\left\{Q_i^{\pm}, \frac{m_i}{\Delta t}(u_j - u_i)\right\}$

    $Q_j^{\pm} := \overset{\max}{\min}\left\{Q_j^{\pm}, \frac{m_j}{\Delta t}(u_i - u_j)\right\}$

For all i **do**

  $R_i^{\pm} := \min\left\{1, \frac{Q_i^{\pm}}{P_i^{\pm}}\right\}$

For all i **do**

  For all $j \in S_i,\ j > i$ **do**

    $\alpha_{ij} := \min\{R_i^{\pm}, R_j^{\mp}\}$

    $\bar{f}_{ij} := \alpha_{ij} f_{ij}$

    $\bar{f}_i := \bar{f}_i + \bar{f}_{ij}$

    $\bar{f}_j := \bar{f}_j - \bar{f}_{ij}$

The advantages of edge-based finite element solvers include algorithmic simplicity, low memory requirements, and a major reduction in indirect addressing [55, 56]. Moreover, edge-based data structures are well-suited for large-scale parallel computing [11, 54, 61]. Last but not least, the equivalence between linear finite elements and vertex-centered finite volumes can be exploited to develop a unified framework for edge-based flux/slope limiting on unstructured meshes [60, 74].

Of course, algebraic flux correction schemes can also be implemented in an existing finite element code based on traditional element-by-element matrix assembly. In our code, we use edge-based data structures for limiting purposes only.

6.4.4 Clipping and Terracing

A well-known problem associated with flux correction of FCT type is *clipping* [7, 81]. Since the sum of limited antidiffusive fluxes is forced to be local extremum diminishing, existing peaks lose a little bit of amplitude during each time step. To alleviate peak clipping, Zalesak [81] defined $u_i^{\max}$ and $u_i^{\min}$ as the local extrema of u^n or $\tilde{u}$. This adjustment is consistent with the local discrete maximum principle (Definition 4). However, it may produce an overshoot or undershoot if the transport equation contains source terms that change the definition of the local DMP.

In our experience, a nonclipping flux limiter can be designed using information about the higher-order derivatives. In [43] we developed such a limiter for quadratic finite elements within the framework of a discontinuous Galerkin (DG) method. In the case of linear or multilinear finite element FCT schemes, the second derivatives are not available, which makes it more difficult to distinguish between spurious spikes ('wiggles') and smooth peaks. The use of Hessian recovery techniques produces a smooth approximation which is no longer a reliable shock detector.

Another infamous byproduct of FCT manifests itself in distortions of a smooth profile. This phenomenon is known as *terracing* and represents 'an integrated, nonlinear effect of residual phase errors' [67] or, loosely speaking, 'the ghosts of departed ripples' [7]. A particularly severe form of terracing is caused by the linear instability of the high-order scheme. For this reason, we do not recommend the use of the forward Euler time-stepping ($\theta = 0$) even though the flux-corrected scheme proves positivity-preserving under the CFL-like time step restriction (62).

Terracing can also be caused by the lack of information about the solution behavior in the exterior of Ω. Figure 1 displays a zoom of the FCT solution to the 1D convection equation (43) with $v = 1$ an $u^0 = x$ in $\Omega = (0,1)$. The standard Galerkin method would produce excellent results for a linear profile but the FCT version gives rise to terracing in a neighborhood of the (artificial) open boundary $x = 1$. This happens because the solution value at the last node is treated as a peak, although it is not a peak if we make Ω a little longer [63]. This example indicates that the FCT limiter is not linearity-preserving, which makes it particularly prone to terracing.

6.5 Evaluation of FCT

In our experience, FCT produces excellent results for strongly time-dependent problems. The use of small time steps increases the amount of accepted antidiffusion and justifies the linearization of the antidiffusive flux which leads to a simple and efficient predictor-corrector algorithm. The cost of an implicit FCT scheme 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 the sparse linear system can be solved with 1-2 iterations of the Jacobi or Gauß-Seidel method. Thus, the cost *per time step* approaches that of an explicit finite difference or finite volume scheme. As the time step increases, so does the number of iterations, and advanced linear al-

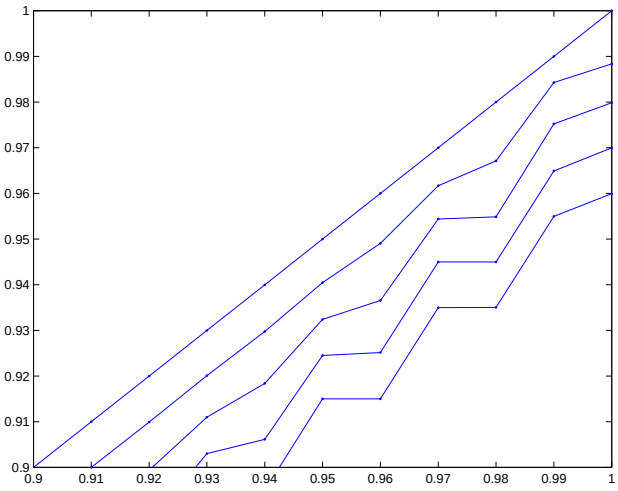

Fig. 1 Terracing in the neighborhood of a hyperbolic outlet.

gebra tools (smoothers, preconditioners, convergence acceleration techniques) may need to be employed. Moreover, the use of large time steps degrades the accuracy of an FCT algorithm since $Q_i^{\pm} \to 0$ as $\Delta t \to \infty$. Other potential drawbacks include clipping, terracing, and the ad hoc nature of the prelimiting procedure.

We conclude that algebraic flux correction of FCT type is the method of choice for evolutionary problems. No other shock capturing technique performs better when it comes to solving an unsteady hyperbolic equation with linear finite elements [35]. For steady-state computations, we recommend the linearity-preserving limiter presented in the next section. It is similar to FCT in many ways but its derivation is based on variational gradient recovery, and all high-order operators (M_C, K, and L) are constrained independently using the same general-purpose limiting strategy.

7 Linearity-Preserving Limiters

As an alternative to FCT, we developed several multidimensional flux limiters which are independent of the time step and produce a TVD scheme in the 1D case [39, 40, 46]. As this methodology evolved and matured, we realized that the definition of the upper and lower bounds for a generalized TVD scheme must guarantee *linearity preservation* on arbitrary meshes. In other words, the constrained approximation must reduce to the underlying Galerkin scheme if the solution is a linear function. This property implies consistency and second-order accuracy for smooth data [10, 62]. In the context of algebraic flux correction, it can be enforced using variational gradient recovery to obtain the LED bounds for the edge-based slope limiter [51].

Another open problem in the design of TVD-like schemes for finite elements was the treatment of the consistent mass matrix which is essential for maintaining the high accuracy of the Galerkin scheme for time-dependent problems. Our multidimensional limiters of TVD type were designed to constrain the entries of the discrete convection operator, and our first attempts to limit the consistent mass matrix independently were rather unsuccessful. This has led us to marry FCT and 'TVD' within the framework of a general-purpose flux limiter [39]. Unfortunately, the resulting scheme inherited not only the advantages but also some drawbacks of the two limiting techniques (dependence on the time step, lack of linearity preservation, artificial coupling between the antidiffusive fluxes associated with different discrete operators). Moreover, the increased complexity of the algorithm has made it too expensive for practical purposes. For some time, we continued using the more efficient special-purpose limiting techniques: FCT for time-dependent problems and lumped-mass 'TVD' for steady-state computations. In this section, we introduce an algebraic flux correction scheme that can handle both situations equally well.

The algorithm to be presented is a fully multidimensional counterpart of the edge-based slope limiter we developed in [51] for anisotropic diffusion problems. In what follows, we extend it to steady and unsteady convective transport. The contribution of the consistent mass matrix is taken into account by applying the limiter to the vector of discretized time derivatives. Furthermore, we constrain the sum of raw antidiffusive fluxes instead of individual fluxes or slopes. This revision results in a marked gain of accuracy as compared to edge-by-edge slope limiting.

Another major improvement is a new iterative solver for the nonlinear algebraic system. We present a nonlinear SSOR scheme which updates the nodal values of the numerical solution and the limited antidiffusive fluxes in a single loop over the nodes of the computational mesh. To speed up convergence, we use *Anderson acceleration* [1, 80], also known as *Anderson mixing* [18, 19]. The efficiency of this approach is confirmed by our numerical study for an anisotropic diffusion equation. On fine meshes, the number of SSOR iterations is reduced by a factor of 60 and more.

7.1 Flux Splitting

So far we have limited all components of the raw antidiffusive flux f_{ij} using a common correction factor $\alpha_{ij} \in [0,1]$. Let us now replace definition (53) with

$$\bar{f}_{ij} := \alpha_{ij}^M f_{ij}^M + \alpha_{ij}^K f_{ij}^K + \alpha_{ij}^L f_{ij}^L, \tag{84}$$

where α_{ij}^M, α_{ij}^K, and α_{ij}^L denote the individual correction factors for the fluxes

$$f_{ij}^M = m_{ij}(\dot{u}_i - \dot{u}_j), \tag{85}$$

$$f_{ij}^K = d_{ij}(u_i - u_j), \tag{86}$$

$$f_{ij}^L = l_{ij}^+(u_i - u_j). \tag{87}$$

The limited antidiffusive term (52) proves local extremum diminishing if

$$q_i^M(\dot{u}_i^{\min} - \dot{u}_i) \le \sum_{j\neq i} \alpha_{ij}^M f_{ij}^M \le q_i^M(\dot{u}_i^{\max} - \dot{u}_i), \tag{88}$$

$$q_i^K(u_i^{\min} - u_i) \le \sum_{j\neq i} \alpha_{ij}^K f_{ij}^K \le q_i^K(u_i^{\max} - u_i), \tag{89}$$

$$q_i^L(u_i^{\min} - u_i) \le \sum_{j\neq i} \alpha_{ij}^L f_{ij}^L \le q_i^L(u_i^{\max} - u_i) \tag{90}$$

for some positive constants q_i^M, q_i^K, and q_i^L independent of u. In this section, we use criterion (88)–(90) to determine the values of α_{ij}^M, α_{ij}^K, and α_{ij}^L.

Without loss of generality, we consider $f_{ij} := f_{ij}^K$ and present the limiting strategy that delivers α_{ij} satisfying (57) for a given $q_i > 0$. The fluxes f_{ij}^L and f_{ij}^M are limited in the same way but the bounds for f_{ij}^M are defined in terms of $\dot{u}$ rather than u.

7.2 Gradient-Based Slope Limiting

To get started, we present the symmetric linearity-preserving (LP) slope limiter we developed in [51] in the context of steady anisotropic diffusion. This algorithm belongs to the family of edge-based stencil reconstruction methods that constrain the jumps of the gradient along the line connecting two nodes [32, 54, 60, 70].

A raw antidiffusive flux of the form $f_{ij} = d_{ij}(u_i - u_j)$ requires limiting if the difference between u_i and u_j is "too large." Introducing the limited slope

$$\bar{s}_{ij} := \alpha_{ij}(u_i - u_j), \qquad 0 \le \alpha_{ij} \le 1, \tag{91}$$

we define

$$\bar{f}_{ij} := d_{ij}\bar{s}_{ij} = \alpha_{ij} f_{ij}. \tag{92}$$

Thus, the multiplication of f_{ij} by α_{ij} is equivalent to replacing $u_i - u_j$ with $\bar{s}_{ij}$. An algorithm that produces $\bar{s}_{ij}$ rather than α_{ij} is called a *slope limiter*.

The definition of $\bar{s}_{ij}$ must guarantee that $\bar{f}_{ij}$ is LED. This will be the case if

$$s_{ij}^{\min} \le \bar{s}_{ij} \le s_{ij}^{\max}, \tag{93}$$

where the upper and lower bounds are given by

$$s_{ij}^{\max} = \gamma_{ij}(u_i^{\max} - u_i), \tag{94}$$

$$s_{ij}^{\min} = \gamma_{ij}(u_i^{\min} - u_i) \tag{95}$$

for some bounded $\gamma_{ij} \ge 0$. That is, each slope is constrained in the same manner as the sum of antidiffusive fluxes in (57). This approach is used in many edge-based extensions of 1D high-resolution schemes to unstructured meshes [32, 54, 60].

To construct the LED bounds $s_{ij}^{\max}$ and $s_{ij}^{\min}$, consider the linear approximation

$$u_i - u_j \approx s_{ij} := (\nabla u)_i \cdot (\mathbf{x}_i - \mathbf{x}_j). \tag{96}$$

The value of $(\nabla u)_i$ is obtained using numerical differentiation. A variety of gradient reconstruction techniques based on averaging or superconvergent patch recovery are available for this purpose. In our method, we use the lumped-mass L^2 projection

$$(\nabla u)_i = \frac{1}{m_i} \sum_k \mathbf{c}_{ik} u_k, \tag{97}$$

where m_i is a diagonal entry of the lumped mass matrix M_L, and $\mathbf{c}_{ik}$ is a vector-valued coefficient of the discrete gradient operator $\mathbf{C}$ given by (31). Since the gradients of Lagrange basis functions sum to zero, we have

$$\mathbf{c}_{ii} = -\sum_{k \neq i} \mathbf{c}_{ik}.$$

Thus

$$(\nabla u)_i = \frac{1}{m_i} \sum_{k \neq i} \mathbf{c}_{ik} (u_k - u_i). \tag{98}$$

We will use this representation to derive the LED bounds for the extrapolated slope s_{ij}, and then we will use these bounds to define γ_{ij} in (94) and (95).

Plugging (98) into the definition of s_{ij}, we obtain the following estimates

$$s_{ij} \leq \frac{1}{m_i} \sum_{k \neq i} |\mathbf{c}_{ik} \cdot (\mathbf{x}_i - \mathbf{x}_j)| (u_i^{\max} - u_i), \tag{99}$$

$$s_{ij} \geq \frac{1}{m_i} \sum_{k \neq i} |\mathbf{c}_{ik} \cdot (\mathbf{x}_i - \mathbf{x}_j)| (u_i^{\min} - u_i). \tag{100}$$

To make the bounds for s_{ij} less restrictive, we multiply them by 2 and define

$$\gamma_{ij} := \frac{2}{m_i} \sum_{k \neq i} |\mathbf{c}_{ik} \cdot (\mathbf{x}_i - \mathbf{x}_j)|. \tag{101}$$

This nonnegative coefficient is used to determine the bounds (94) and (95) for the slope limiter. The localized LED constraint (93) can be enforced by setting

$$\bar{s}_{ij} = \begin{cases} \min\{s_{ij}^{\max}, u_i - u_j\}, & \text{if } u_i > u_j, \\ \max\{s_{ij}^{\min}, u_i - u_j\}, & \text{if } u_i < u_j. \end{cases} \tag{102}$$

The one-sided limiting strategy is sufficient if the slope $\bar{s}_{ji} := -\bar{s}_{ij}$ cannot violate the LED principle for node j. In particular, this is the case if j is a node on the Dirichlet boundary or a downwind neighbor of node i (see Section 7.3). In all other cases, the slope limiter must enforce not only (93) but also $s_{ji}^{\min} \leq \bar{s}_{ji} \leq s_{ji}^{\max}$.

The following definition of $\bar{s}_{ij}$ guarantees the LED property for both nodes [51]

$$\bar{s}_{ij} = \begin{cases} \min\{s_{ij}^{\max}, u_i - u_j, -s_{ji}^{\min}\}, & \text{if } u_i > u_j, \\ \max\{s_{ij}^{\min}, u_i - u_j, -s_{ji}^{\max}\}, & \text{if } u_i < u_j. \end{cases} \tag{103}$$

This symmetric limiting strategy corresponds to a double application of the one-sided slope limiter. In the following Theorem, we prove linearity preservation.

Theorem 5. *If u_h is linear, then the lumped-mass L^2 projection (97) is exact and*

$$s_{ij} = u_i - u_j = \bar{s}_{ij}.$$

Proof. If u_h is a linear, then its gradient is constant and $u_i - u_j = \nabla u_h \cdot (\mathbf{x}_i - \mathbf{x}_j)$. It follows that $s_{ij} = u_i - u_j$ if $(\nabla u)_i = \nabla u_h$. According to (97), we have

$$(\nabla u)_i = \frac{1}{m_i} \int_\Omega \varphi_i \nabla u_h \, \mathrm{d}\mathbf{x} = \nabla u_h \left(\frac{1}{m_i} \int_\Omega \varphi_i \, \mathrm{d}\mathbf{x} \right) = \nabla u_h \tag{104}$$

since the diagonal entry of the lumped mass matrix is given by

$$m_i = \sum_j m_{ij} = \int_\Omega \varphi_i \left(\sum_j \varphi_j \right) \mathrm{d}\mathbf{x} = \int_\Omega \varphi_i \, \mathrm{d}\mathbf{x}.$$

Thus, the L^2 projection is exact and $s_{ij} = u_i - u_j$. By definition of γ_{ij}, the slope $\bar{s}_{ij} = s_{ij}$ satisfies the imposed constraints, whence no limiting is performed. □

Linearity preservation implies that $\bar{f}_{ij} \to f_{ij}$ as $h \to 0$. Therefore, the constrained Galerkin scheme is consistent even if the low-order scheme is inconsistent.

Example 2. To illustrate the relationship of the linearity-preserving slope limiter to classical TVD schemes [29, 78], consider a 1D mesh with uniform spacing Δx. In this case, the coefficients of (97) are given by $m_i = \Delta x$ and $c_{i\pm 1/2} = \pm 1/2$.

The resulting formula for u_i' is equivalent to the second-order central difference

$$u_i' = \frac{1}{2} \left(\frac{u_i - u_{i-1}}{\Delta x} + \frac{u_{i+1} - u_i}{\Delta x} \right) = \frac{u_{i+1} - u_{i-1}}{2\Delta x}.$$

For any interior node, the local maxima and minima of the grid function are

$$u_i^{\max} = \max\{u_{i-1}, u_i, u_{i+1}\}, \qquad u_i^{\min} = \min\{u_{i-1}, u_i, u_{i+1}\}.$$

Furthermore, $\gamma_{ij} = 2$ for $j = i+1$ since estimate (99)–(100) corresponds to

$$u_i^{\min} - u_i \le \Delta x u_i' \le u_i^{\max} - u_i.$$

The one-sided slope limiter (102) can be written as a single-line formula

$$\bar{s}_{ij} = \mathit{minmod}\{2(u_{i-1} - u_i), u_i - u_{i+1}\},$$

and the corresponding formula for the symmetric slope limiter (103) reads

$$\bar{s}_{ij} = minmod\{2(u_{i-1}-u_i), u_i - u_{i+1}, 2(u_{i+1}-u_{i+2})\}.$$

The *minmod* limiter function returns the argument with the smallest magnitude if all arguments have the same sign and zero otherwise. That is,

$$minmod\{a,b,\dots\} = \begin{cases} \min\{a,b,\dots\}, & \text{if} \quad a>0,\ b>0,\dots \\ \max\{a,b,\dots\}, & \text{if} \quad a<0,\ b<0,\dots \\ 0, & \text{otherwise.} \end{cases}$$

It follows that the proposed slope limiter is activated only if two consecutive gradients have opposite signs or their magnitudes differ by a factor of 2 and more.

7.3 Symmetric Flux Limiter

In contrast to the fully multidimensional FCT method, the linearity-preserving (LP) slope limiter presented in Section 7.2 constrains the antidiffusive flux f_{ij} independently of all other fluxes into node i. This is convenient but the results are quite sensitive to the orientation of mesh edges. In this section, we convert the edge-based slope limiter into an FCT-like limiter for the sum of antidiffusive fluxes. The LED constraint (57) can be enforced using the following generalization of (80)–(83)

1. Compute the sums of positive/negative antidiffusive fluxes to be limited

$$P_i^+ = \sum_{j\neq i} \max\{0, f_{ij}\}, \qquad P_i^- = \sum_{j\neq i} \min\{0, f_{ij}\}. \tag{105}$$

2. Define local extremum diminishing upper/lower bounds of the form

$$Q_i^+ = q_i(u_i^{\max} - u_i\}, \qquad Q_i^- = q_i(u_i^{\min} - u_i\}. \tag{106}$$

3. Compute the nodal correction factors for positive/negative fluxes

$$R_i^+ = \min\left\{1, \frac{Q_i^+}{P_i^+}\right\}, \qquad R_i^- = \min\left\{1, \frac{Q_i^-}{P_i^-}\right\}. \tag{107}$$

4. Limit the fluxes f_{ij} and f_{ji} using the common correction factor

$$\alpha_{ij} = \begin{cases} \min\{R_i^+, R_j^-\}, & \text{if } f_{ij}>0, \\ \min\{R_i^-, R_j^+\}, & \text{if } f_{ij}<0. \end{cases} \tag{108}$$

As in the case of FCT, this definition of the correction factor α_{ij} implies that

$$Q_i^- \le R_i^- P_i^- \le \sum_{j\neq i} \alpha_{ij} f_{ij} \le R_i^+ P_i^+ \le Q_i^+. \tag{109}$$

To maintain linearity preservation, we define $Q_i^{\pm}$ as the sum of the LED bounds we imposed on individual slopes/fluxes in Section 7.2. That is, we set

$$q_i := \sum_{j \neq i} \gamma_{ij} d_{ij}. \tag{110}$$

In contrast to FCT, the resulting formula for $Q_i^{\pm}$ is independent of the time step.

7.4 One-Sided Flux Limiter

Algorithm (105)–(108) is ideally suited for constraining a symmetric operator like L or M_C. In the latter case, the antidiffusive fluxes (85) and the bounds $Q_i^{\pm}$ be defined in terms of $\dot{u}$ rather than u. At the fully discrete level, the time derivative is replaced with the finite difference approximation $\dot{u} \approx (u^{n+1} - u^n)/\Delta t$. Note that the same correction factor α_{ij}^M is applied to the explicit and implicit part of f_{ij}^M.

In principle, the discrete convection operator K can also be constrained using (105)–(108). However, it turns out that the LED constraint for node j is satisfied automatically if $k_{ji} > 0$. To take advantage of this fact, we limit the convective part in an upwind-biased fashion [49, 46]. In accordance with our upwind-downwind edge orientation convention (40), we assume that $k_{ij} \leq k_{ji}$. As long as

$$\bar{k}_{ji} := k_{ji} + (1 - \alpha_{ij}) d_{ij}$$

is nonnegative for all $\alpha_{ij} \in [0,1]$, it is enough to make sure that (57) holds for node i.

In the one-sided version of (105)–(108), we begin with the prelimiting step

$$f_{ij} := (d_{ij} + \max\{0, k_{ji}\})(u_i - u_j) \tag{111}$$

which is required to enforce the LED constraint for node j in the unlikely case of $k_{ji} < 0$. After this prelimiting, the correction factors α_{ij} are calculated as follows:

1. Compute the sums of positive/negative antidiffusive fluxes to be limited

$$P_i^+ = \sum_{k_{ij} \leq k_{ji}} \max\{0, f_{ij}\}, \qquad P_i^- = \sum_{k_{ij} \leq k_{ji}} \min\{0, f_{ij}\}. \tag{112}$$

2. Compute q_i and the local extremum diminishing upper/lower bounds

$$Q_i^+ = q_i(u_i^{\max} - u_i\}, \qquad Q_i^- = q_i(u_i^{\min} - u_i\}. \tag{113}$$

3. Compute the nodal correction factors for positive/negative fluxes

$$R_i^+ = \min\left\{1, \frac{Q_i^+}{P_i^+}\right\}, \qquad R_i^- = \min\left\{1, \frac{Q_i^-}{P_i^-}\right\}. \tag{114}$$

4. Multiply f_{ij} and f_{ji} by the nodal correction factor for the *upwind* node i

$$k_{ij} \le k_{ji} \quad \Rightarrow \quad \alpha_{ij} = \begin{cases} R_i^+, & \text{if } f_{ij} \ge 0, \\ R_i^-, & \text{if } f_{ij} < 0, \end{cases} \qquad \alpha_{ji} := \alpha_{ij}. \tag{115}$$

We have used this one-sided limiting strategy to design algebraic flux correction schemes based on a generalization of upwind TVD limiters [39, 40, 46, 49].

8 Solution of Nonlinear Systems

After the discretization in time, the flux-corrected discrete problem can be written in the form (58). Since the antidiffusive term depends on the unknown solution, the nonlinear discrete problem must be solved in an iterative way. In contrast to the nonlinear FCT algorithm presented in Section 6.2, only the fully converged solution is guaranteed to be nonoscillatory. Therefore, it is essential to make sure that iterations converge. Moreover, convergence must be fast enough to keep the cost of algebraic flux correction reasonable. Thus, the robustness and efficiency of the iterative solver for the nonlinear system are just as important as the flux limiting procedure.

8.1 Defect Correction Scheme

As in the case of FCT, the structure of the nonlinear system (58) suggests the use of a fixed-point iteration with a lagged evaluation of the antidiffusive term

$$Au^{(m+1)} = Bu^n + \bar{f}^{(m)}. \tag{116}$$

A more general class of defect correction schemes can be formally written as

$$u^{(m+1)} = u^{(m)} + \omega \tilde{A}^{-1} r^{(m)}, \tag{117}$$

where $\tilde{A}$ is an approximation to the Jacobian of the nonlinear system, $\omega \in [0,1]$ is a relaxation parameter, and $r^{(m)}$ is the residual vector given by

$$r^{(m)} = Bu^n - Au^{(m)} + \bar{f}^{(m)}. \tag{118}$$

In practice, the matrix $\tilde{A}$ is 'inverted' by solving a linear system (see Algorithm 2). The iteration process is typically terminated when certain norms of $u^{(m+1)} - u^{(m)}$ and/or $r^{(m+1)}$ become smaller than a prescribed tolerance. More elaborate stopping criteria based on the finite element theory can be found in [2]. Clearly, the rates of convergence and the overall efficiency of the above defect correction scheme are strongly influenced by the choice of the 'preconditioner' $\tilde{A}$.

Algorithm 2: Defect correction scheme.

Set $u^{(0)} := u^n$

For all $m = 0, 1, \ldots$ **do**

Solve the linear system $\tilde{A}\Delta u^{(m+1)} = r^{(m)}$

Update the solution $u^{(m+1)} := u^{(m)} + \omega\Delta u^{(m+1)}$

Exit if the stopping criteria are satisfied

Set $u^{n+1} := u^{(m+1)}$

The default setting is $\omega := 1$ and $\tilde{A} := A$, which corresponds to (116). By construction, the low-order operator A is an M-matrix. This property results in fast convergence of inner iterations. If the time step Δt is very small, the solution can be updated in a fully explicit fashion using the diagonal preconditioner $\tilde{A} := \mathrm{diag}(A)$. As few as 1-3 outer iterations may suffice if good initial guess is available. Thus, the cost per time step might be comparable to that of an explicit algorithm. On the other hand, such a solver may fail to converge if the time step is too large.

Some advanced preconditioning and underrelaxation techniques are discussed in [44, 51, 64]. In quasi-Newton methods, $\tilde{A}$ must be a good approximation to the Jacobian of (58). Due to the complex structure and nondifferentiability of the limited antidiffusive term, the assembly of such preconditioners is very complicated and expensive. Thus, Jacobian-free solvers are to be preferred. In particular, the convergence acceleration method described in Section 8.3 leads to a Newton-like scheme in which the memory effect is exploited to avoid numerical differentiation.

8.2 Nonlinear SSOR Scheme

A major drawback of fixed-point methods like (116) is the fully explicit treatment of the antidiffusive term. An attempt to build implicit antidiffusion into the preconditioner $\tilde{A}$ aggravates convergence problems if all correction factors are taken from the previous outer iteration. This has led us to update the solution values, the antidiffusive fluxes, and the correction factors simultaneously in a loop over nodes. The resulting algorithm can be classified as a nonlinear Gauß-Seidel / SSOR method.

The i-th equation of the flux-corrected Galerkin scheme (58) can be written as

$$\sum_j a_{ij} u_j = b_i + \bar{f}_i, \tag{119}$$

where the antidiffusive term $\bar{f}_i$ depends on $u = u^{n+1}$, whereas $b_i = \sum_j b_{ij} u_j^n$ is known.

The calculation of $u_i^{(m+1)} \approx u_i^{n+1}$ begins with the assembly of $\bar{f}_i$. In the forward sweep, the new values of u_j are already available for all $j < i$. Thus

$$u_j = \begin{cases} u_j^{(m+1)}, & \text{if } j < i, \\ u_j^{(m)}, & \text{if } j \geq i. \end{cases} \tag{120}$$

In the backward sweep, the solution values are updated in the reverse order, so the i-th step begins with $u_j = u_j^{(m+1)}$ for $j > i$ and $u_j = u_j^{(m)}$ otherwise.

Given the array of current solution values u_i, we recalculate the raw antidiffusive fluxes f_{ij}, apply the flux limiter, and add the result to $\bar{f}_i$ (see Algorithm 3).

Algorithm 3: Assembly of $\bar{f}_i$ (symmetric version).

For all i **do**
- $P_i^{\pm} := 0, \;\; Q_i^{\pm} := 0, \;\; \bar{f}_i := 0$
- **For** all $j \in S_i$ **do**
  - $P_i^{\pm} := P_i^{\pm} + \mathop{\max}_{\min}\{0, f_{ij}\}$
  - $Q_i^{\pm} := \mathop{\max}_{\min}\left\{Q_i^{\pm}, \frac{m_i}{\Delta t}(u_j - u_i)\right\}$
- $R_i^{\pm} := \min\left\{1, Q_i^{\pm}/P_i^{\pm}\right\}$
- **For** all $j \in S_i$ **do**
  - $\alpha_{ij} := \min\{R_i^{\pm}, R_j^{\mp}\}$
  - $\bar{f}_i := \bar{f}_i + \alpha_{ij} f_{ij}$

Since the value of α_{ij} depends not only on $R_i^{\pm}$ but also on $R_j^{\mp}$, we store the updated nodal correction factors, so that they are readily available when it comes to calculating α_{ij}. Due to the lag in evaluation of $\bar{f}_{ij}$ and $\bar{f}_{ji}$, intermediate approximations may be nonconservative but $\bar{f}_{ji} = -\bar{f}_{ij}$ when the algorithm converges.

Given the updated value of $\bar{f}_i$, the old solution value u_i is overwritten by

$$u_i := u_i + \frac{1}{\tilde{a}_{ii}}\left(b_i - \sum_j a_{ij} u_j + \bar{f}_i\right), \tag{121}$$

where $\tilde{a}_{ii} \geq a_{ii}$. Setting $\tilde{a}_{ii} := a_{ii}$, one obtains the symmetric Gauß-Seidel (SGS) method which may fail to converge if the implicit part of $\bar{f}_i$ is too large compared to $\sum_j a_{ij} u_j$. A possible remedy is implicit underrelaxation of the form

$$\tilde{a}_{ii} := \frac{a_{ii}}{\omega}, \qquad 0 < \omega \leq 1.$$

Equivalently, the SSOR scaling factor $\tilde{a}_{ii}$ can be defined by adding a nonnegative number to the diagonal entry. In our numerical experiments, we used

$$\tilde{a}_{ii} := a_{ii} + \theta \sum_{j \neq i} (d_{ij} + l_{ij}^{+}).$$

The flow chart of the nonlinear SSOR method for solving (58) is as follows:

Algorithm 4: Nonlinear SSOR iteration.

For all $i = 1, \ldots, N-1, N$ **do** *(forward sweep)*

For all $i = N, N-1, \ldots, 1$ **do** *(backward sweep)*

Update the antidiffusive term $\bar{f}_i$ using Algorithm 3

Calculate the new solution value u_i using (121)

The forward sweep can be written as $(\tilde{D} + \tilde{L})\Delta u^{*} := r$, where r is the residual, $\tilde{D} = \mathrm{diag}\{\tilde{a}_{ii}\}$ is a diagonal matrix of scaling factors, and $\tilde{L}$ is the strict lower triangular part of A plus limited antidiffusion. Likewise, the backward sweep can be written as $(\tilde{D} + \tilde{U})\Delta u := \Delta u^{*}$, where $\tilde{U}$ is a strict upper triangular matrix. Thus, Algorithm 4 can written in the form (117) with $\omega = 1$ and

$$\tilde{A} = (\tilde{D} + \tilde{L})\tilde{D}^{-1}(\tilde{D} + \tilde{U}).$$

Luo et al. [59] used this sort of defect correction as a preconditioner for a linear GMRES solver. A nonlinear version of this solution strategy is recovered when the method presented in Section 8.3 is employed to accelerate Algorithm 4.

In the iterative solver for steady transport equations, we set $\theta := 1$ and $b_i := 0$. Furthermore, the contribution of the mass matrix is removed, which corresponds to using an infinitely large pseudo-time step Δt. It is also possible to march the solution to the steady state using $\theta := 1$ and local time stepping. In either case, a usable initial guess can be obtained by solving the linear system with $\bar{f}_i = 0$ or $\bar{f}_i = f_i$.

8.3 Anderson Acceleration

Since the cost of recalculating the correction factors for the flux limiter is rather high, slow convergence of an iterative method can make algebraic flux correction very expensive. The fixed-point defect correction scheme (117) and the nonlinear SSOR iteration (121) generate a sequence of successive approximations but only the last iterate $u^{(m)}$ is used when it comes to the computation of $u^{(m+1)}$. It turns out that including information from a number of previous iterates may dramatically im-

prove the convergence behavior. This idea is exploited in many vector extrapolation techniques for vector sequences (see, e.g., [34, 77]). In this work, we employ the convergence acceleration technique known as *Anderson mixing* [1, 18, 19, 80]. As shown in [18], this approach is equivalent to the Broyden scheme for the inverse Jacobian but is easier to implement and explain. On linear problems, the accelerated fixed point iteration is related to the preconditioned GMRES method [80].

Following Walker and Ni [80], we formulate Anderson acceleration as follows:

Algorithm 6: Anderson acceleration.

For all $m = 0, 1, \ldots$ **do**

Compute $\tilde{u}^{(m)} := g(u^{(m)})$ with (117) or (121)

Store $\tilde{u}^{(m)}$ and $\Delta u^{(m)} := \tilde{u}^{(m)} - u^{(m)}$

Given $k \leq m$ iterates, determine the weights

$$\omega^{(m)} = (\omega_1^{(m)}, \ldots, \omega_k^{(m)})^T$$

by solving the constrained least-squares problem

$$\min_{\omega^{(m)}} \left\| \sum_{i=1}^{k} \omega_i^{(m)} \Delta u^{(m-k+i)} \right\|_2 \quad \text{s.t.} \sum_{i=1}^{k} \omega_i^{(m)} = 1$$

Set $u^{(m+1)} := \sum_{i=1}^{k} \omega_i^{(m)} \tilde{u}^{(m-k+i)}$

In practice, it is worthwhile to calculate the weights by solving an equivalent unconstrained least squares problem [80]. Furthermore, Anderson acceleration may need to be restarted if the vectors $\Delta u^{(m)}$ become (almost) linearly dependent, or if the norm of $\Delta u^{(m)}$ is much greater than that of $\Delta u^{(m-1)}$. We refer to [18, 19, 65, 80] for a discussion of various improvements and practical implementation details.

9 Numerical Examples

A properly designed high-resolution scheme should be (i) at least second-order accurate for smooth data and (ii) capable of resolving small-scale features without excessive smearing or steepening. To evaluate the accuracy and efficiency of our linearity-preserving limiting techniques, we apply them to three representative benchmark problems which have already been studied using other algebraic flux correction schemes [42, 44, 46, 51] as well as variational shock capturing [35],

monotone finite volume schemes [52, 53], and slope limiters for discontinuous Galerkin methods [43]. Thus, a quantitative comparison of the results is possible.

Given a reference solution u and a numerical approximation u_h, we define

$$E_1(h) = \sum_i m_i |u(\mathbf{x}_i) - u_i| \approx \|u - u_h\|_1, \tag{122}$$

$$E_2(h) = \sqrt{\sum_i m_i |u(\mathbf{x}_i) - u_i|^2} \approx \|u - u_h\|_2, \tag{123}$$

where $m_i = \int_\Omega \varphi_i \, \mathrm{d}\mathbf{x}$ stands for a diagonal coefficient of the lumped mass matrix M_L.

The objective of the below numerical study is to investigate the dependence of the errors E_1 and E_2 on the mesh size h and on the choice of the limiting strategy. In particular, we will use the numerical solutions computed on the two finest meshes to estimate the expected order of accuracy by the formula [52]

$$p = \log_2 \left(\frac{E_1(2h)}{E_1(h)} \right). \tag{124}$$

In the last two examples, we compare the convergence behavior of the global defect correction scheme to that of nonlinear SSOR with Anderson acceleration.

9.1 Solid Body Rotation

The solid body rotation test [52, 81] is often used to evaluate numerical advection schemes. The problem to be solved is the continuity equation

$$\frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{v} u) = 0 \qquad \text{in } \Omega = (0,1) \times (0,1). \tag{125}$$

The velocity $\mathbf{v}$ describes a counterclockwise rotation about the center of Ω

$$\mathbf{v}(x,y) = (0.5 - y, x - 0.5). \tag{126}$$

After each full revolution, the exact solution u coincides with the given initial data u_0. Hence, the challenge of this test is to preserve the shape of u_0.

Following LeVeque [52], we simulate solid body rotation of the profile displayed in Fig. 2. The geometry of each body is described by a given function $G(x,y)$ defined on a circle of radius $r_0 = 0.15$ centered at some point $(x_0, y_0) \in \Omega$. Let

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

be the normalized distance from the point (x_0, y_0). Then $r(x,y) \leq 1$ inside the circle.

The slotted cylinder is centered at the point $(x_0, y_0) = (0.5, 0.75)$ and

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

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

$$G(x,y) = 1 - r(x,y).$$

The smooth hump is centered at $(x_0, y_0) = (0.25, 0.5)$, and the shape function is

$$G(x,y) = \frac{1 + \cos(\pi r(x,y))}{4}.$$

In the rest of the domain, the solution to (125) is initialized by zero, and homogeneous Dirichlet boundary conditions are prescribed at the inlets.

The snapshots presented in Figs. 2–5 show the shape of the solution at the final time $T = 2\pi$, which corresponds to one full rotation. All computations were performed on a uniform mesh of 128×128 bilinear elements using the Crank-Nicolson time-stepping with the time step $\Delta t = 10^{-3}$. The results obtained with $\alpha_{ij} := 1$ and $\alpha_{ij} := 0$ are displayed in Figs. 3 and 4, respectively. As expected, the unconstrained Galerkin solution exhibits spurious oscillations, while its low-order counterpart is too diffusive. The solution shown in Fig. 5 was computed using linearized FCT (see Section 6.1) with $\dot{u}^L$ given by (65). A detailed numerical study of FCT schemes (explicit vs. implicit, linearized vs. nonlinear) can be found in [42].

In the captions to Figs. 6–9, the abbreviations LPSL and LPFL refer to the limiting techniques described in Sections 7.2 and 7.3, respectively (LP := *Linearity Preserving*, SL:= *Slope Limiting*, FL := *Flux Limiting*). The results shown in Figs. 6 and 7 indicate that LPFL is more accurate than LPSL and almost as accurate as FCT. This is good news since the solid body rotation test belongs to the class of problems that FCT can handle much better than other shock-capturing methods [35].

In contrast to flux limiters of TVD type [39, 40], LPSL and LPFL are applicable to the antidiffusive part of the consistent mass matrix which makes it possible to attain fourth-order accuracy with linear finite elements (see [16], p. 96). To demonstrate the importance of this result, we present the numerical solutions obtained with the lumped mass matrix ($\alpha_{ij}^M := 0$) in Figs. 8 and 9. The diagram in Fig. 10 depicts the E_1 convergence history for the consistent and lumped-mass versions of LPSL and LPFL. The numerical values values of E_1 and E_2 are listed in Tables 2 and 3. The local Courant number $\nu = |\mathbf{v}| \frac{\Delta t}{h}$ equals zero at the center of the square domain and attains its largest value $\nu_{\max} = \frac{1}{\sqrt{2}} \frac{\Delta t}{h}$ at the corners. In the process of mesh refinement, the time step was adjusted to maintain the fixed ratio $\frac{\Delta t}{h} = 0.128$.

The expected order of accuracy p is estimated using (124) with $h = 1/256$. The rates of convergence for the LP algorithms used in Figs. 6–9 are given by $p = 0.96, 0.90, 0.77$, and 0.77, respectively. The consistent-mass LPFL produces smaller errors than LPSL. However, there is hardly any difference if mass lumping

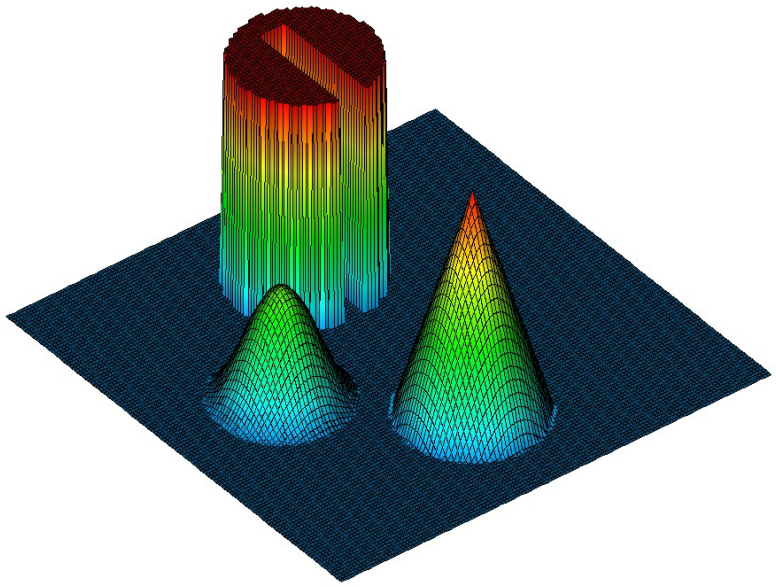

Fig. 2 Solid body rotation: initial data / exact solution at $t = 2\pi$.

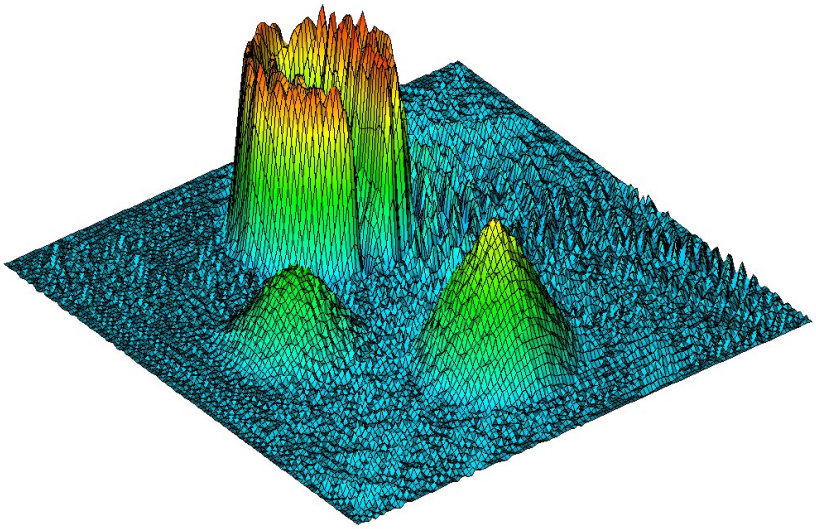

Fig. 3 Solid body rotation: Galerkin solution at $t = 2\pi$.

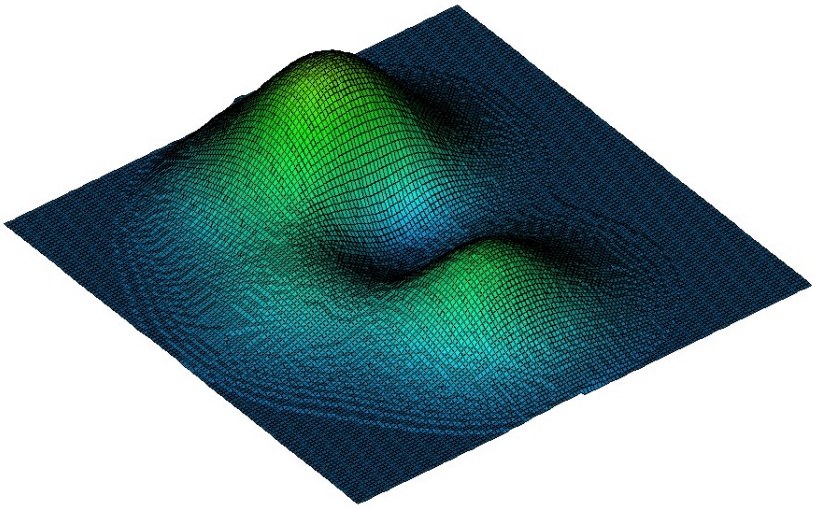

Fig. 4 Solid body rotation: low-order solution at $t = 2\pi$.

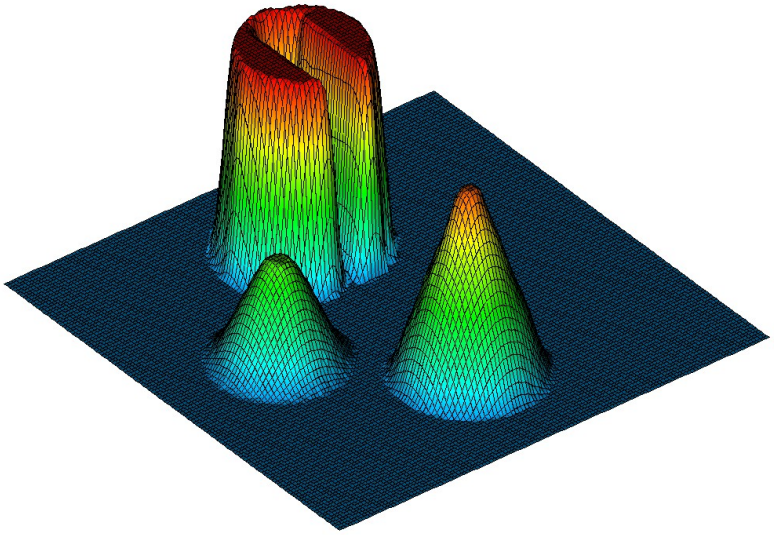

Fig. 5 Solid body rotation: FEM-FCT solution at $t = 2\pi$.

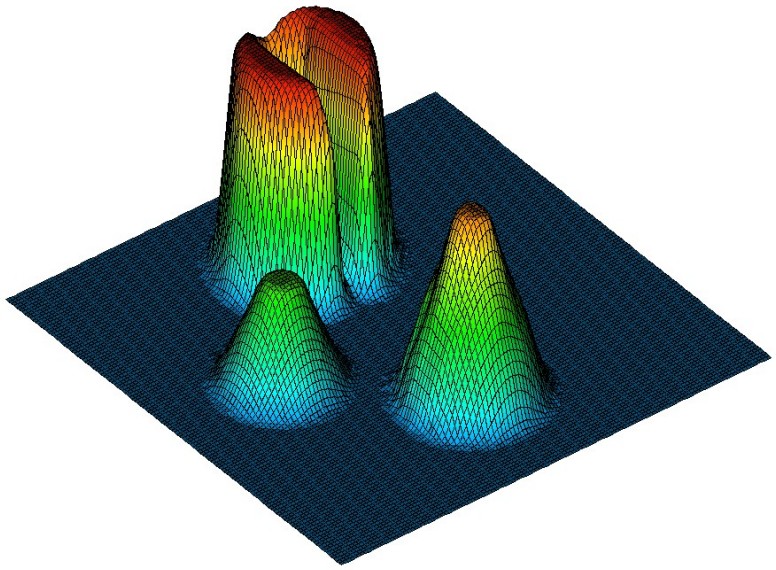

Fig. 6 Solid body rotation: consistent-mass LPSL solution at $t = 2\pi$.

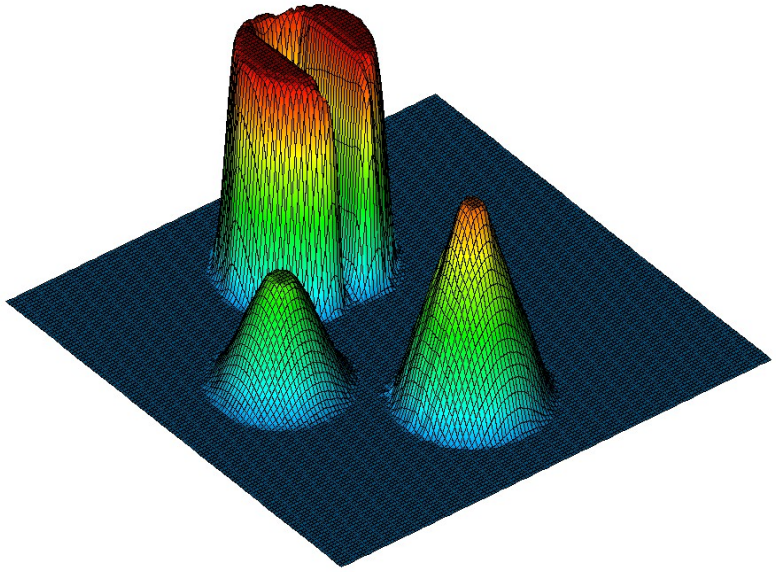

Fig. 7 Solid body rotation: consistent-mass LPFL solution at $t = 2\pi$.

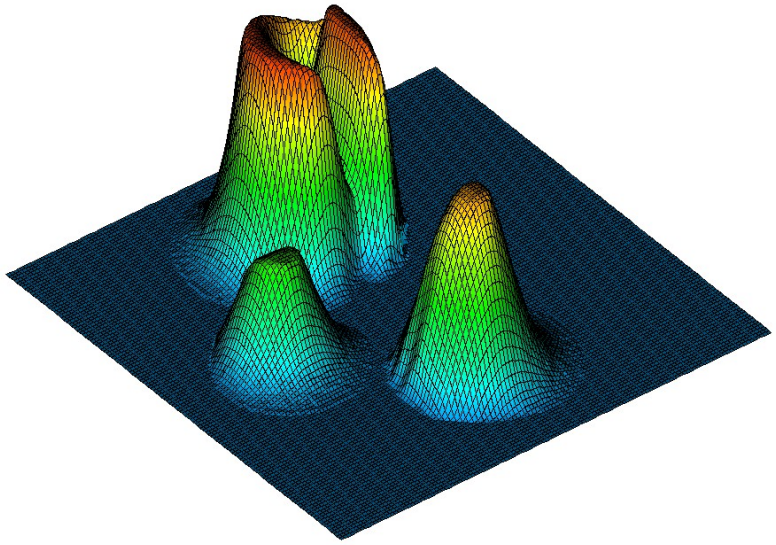

Fig. 8 Solid body rotation: lumped-mass LPSL solution at $t = 2\pi$.

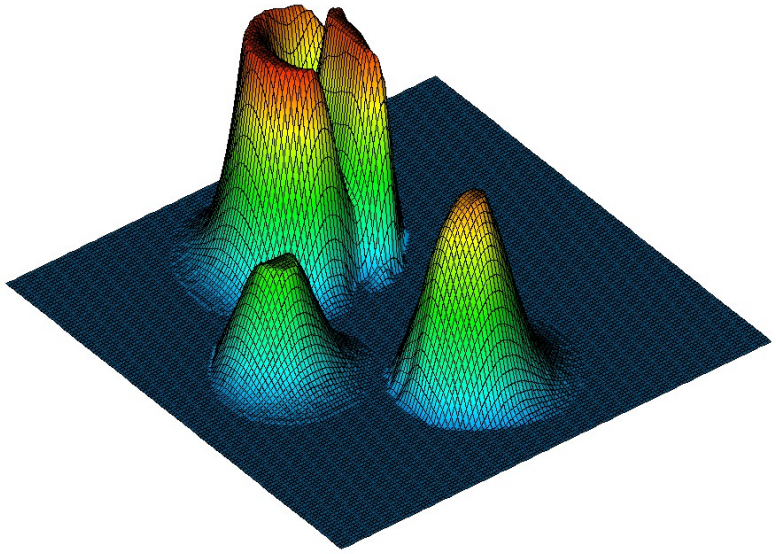

Fig. 9 Solid body rotation: lumped-mass LPFL solution at $t = 2\pi$.

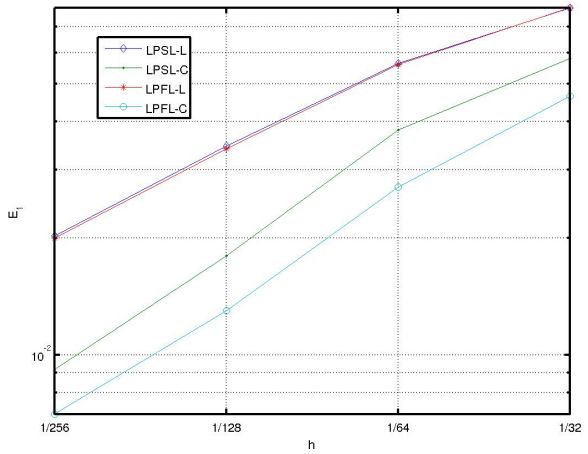

Fig. 10 Solid body rotation, convergence history for LP limiters.

	LPSL, lumped mass		LPSL, consistent mass	
h	E_1	E_2	E_1	E_2
1/32	0.783E-01	0.163E+00	0.582E-01	0.135E+00
1/64	0.564e-01	0.144e+00	0.380E-01	0.111E+00
1/128	0.346e-01	0.109e+00	0.180E-01	0.704E-01
1/256	0.203e-01	0.803e-01	0.919E-02	0.509E-01

Table 2 Solid body rotation: LPSL grid convergence.

	LPFL, lumped mass		LPFL, consistent mass	
h	E_1	E_2	E_1	E_2
1/32	0.785E-01	0.165E+00	0.465E-01	0.125E+00
1/64	0.560E-01	0.147E+00	0.271E-01	0.907E-01
1/128	0.340E-01	0.110E+00	0.130E-01	0.612E-01
1/256	0.200E-01	0.806E-01	0.705E-02	0.459E-01

Table 3 Solid body rotation: LPFL grid convergence.

is performed. In this case, both algorithms converge at the rate $p = 0.77$, which is a typical value for a TVD scheme that delivers $p = 2$ for smooth data. The use of the consistent mass matrix results in a significant gain of accuracy and faster grid convergence. This justifies the additional effort invested in the computation of α_{ij}^M.

9.2 Circular Convection

The second test problem is taken from [30]. Consider the hyperbolic PDE

$$\nabla \cdot (\mathbf{v}u) = 0 \qquad \text{in } \Omega = (-1,1) \times (0,1) \tag{127}$$

which describes steady circular convection if the velocity field is defined as

$$\mathbf{v}(x,y) = (y,-x).$$

The exact solution and inflow boundary conditions for this test are given by

$$u(x,y) = \begin{cases} G(r), & \text{if } 0.35 \le r = \sqrt{x^2+y^2} \le 0.65, \\ 0, & \text{otherwise,} \end{cases}$$

where $G(r)$ is a given function that defines the shape of the solution profile.

To evaluate the performance of LPSL and LPFL for smooth data and discontinuous solutions, we consider the following shape functions

$$G_1(r) = \cos^2\left(5\pi\frac{2r+1}{3}\right), \qquad G_2(r) \equiv 1.$$

As before, computations are performed on a uniform mesh of bilinear finite elements which is successively refined to perform a grid convergence study.

The exact solution to the circular convection problem is constant along the streamlines of the stationary velocity field. Figure 11 displays the results for $G = G_1$ and $G = G_2$ computed using the LPFL algorithm with $h = 1/64$. The convergence history for LPSL and LPFL is presented in Tables 4 and 5, respectively. In the case of the smooth profile G_1, the E_1 errors for LPSL are approximately twice as large as those for LPFL. The expected orders of accuracy are 2.22 and 2.11, respectively. In the case of the discontinuous profile G_2, the convergence rates drop to 0.91 for LPSL and 0.83 for LPFL. The absolute values of the E_1 errors differ by a factor of 1.5. We conclude that the revised limiting strategy leads to a marked improvement not only for transient convection problems but also in steady-state computations.

	smooth data		discontinuous data	
h	E_1	E_2	E_1	E_2
1/32	0.318E-01	0.551E-01	0.821E-01	0.152E+00
1/64	0.104E-01	0.204E-01	0.449E-01	0.108E+00
1/128	0.251E-02	0.595E-02	0.259E-01	0.860E-01
1/256	0.537E-03	0.160E-02	0.138E-01	0.601E-01

Table 4 Circular convection: LPSL grid convergence.

	smooth data		discontinuous data	
h	E_1	E_2	E_1	E_2
1/32	0.146E-01	0.266E-01	0.540-01	0.131E+00
1/64	0.377E-02	0.801E-02	0.295E-01	0.893E-01
1/128	0.944E-03	0.230E-02	0.185E-01	0.757E-01
1/256	0.218E-03	0.632E-03	0.104E-01	0.519E-01

Table 5 Circular convection: LPFL grid convergence.

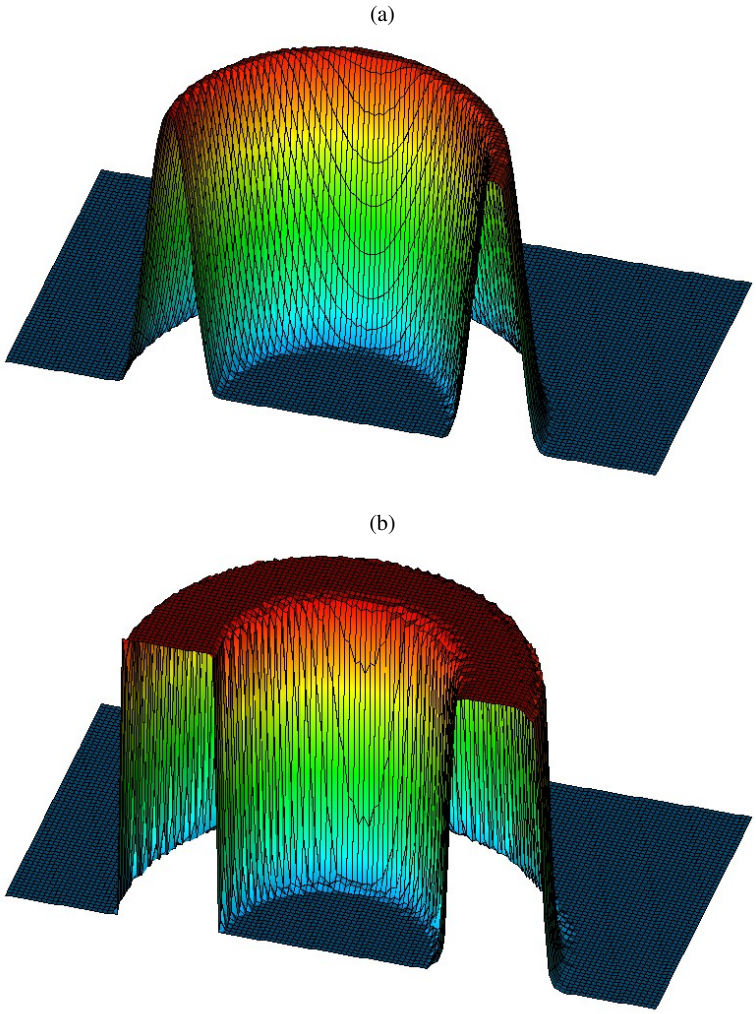

Fig. 11 Circular convection: LPFL results for (a) smooth and (b) discontinuous data.

	defect correction		nonlinear SSOR	
h	$AA(5)$	$AA(10)$	$AA(5)$	$AA(10)$
1/32	576	574	274	287
1/64	—	975	487	501
1/128	—	1866	908	940
1/256	—	—	—	1893

Table 6 Circular convection: number of nonlinear iterations.

The iterative solver was configured to run until the absolute norm of the residual becomes smaller than 10^{-6}. This stopping criterion is more stringent than necessary to obtain an accurate solution. However, it is important to make sure that the residuals go to zero. The methods under investigation are the global defect correction scheme (with $\tilde{a}_{ii} = 2a_{ii}$ and $\tilde{a}_{ij} = a_{ij}$ for $j \neq i$) and the nonlinear SSOR method (with $\tilde{a}_{ii} = \sum_j |a_{ij}|$). The same subroutine was used to evalulate the residuals for both schemes. To prevent division by zero, the LP limiter was implemented using $R_i^{\pm} = \min\left\{1, \frac{Q_i^{\pm}+\varepsilon}{P_i^{\pm}+\varepsilon}\right\}$, where ε is a multiple of the machine precision.

In the circular convection test with the discontinuous profile, the residuals begin to oscillate, and convergence stalls if no Anderson acceleration is performed. The number of nonlinear iterations for the accelerated schemes is presented in Table 6, where AA(k) stands for Anderson acceleration applied to k iterates. The defect correction scheme with $k = 5$ fails to converge in most cases. The total number of iterations for $k = 10$ is twice as large as that for nonlinear SSOR. Moreover, the cost of a defect correction cycle is higher than that of an SSOR iteration.

9.3 Anisotropic Diffusion

In the last example, we consider a steady anisotropic diffusion equation

$$-\nabla \cdot (\mathscr{D}\nabla u) = 0 \quad \text{in } \Omega, \tag{128}$$

where $\Omega = (0,1)^2 \backslash [4/9, 5/9]^2$ is a square domain with a hole in the middle.

The outer and inner boundary of Ω are denoted by Γ_0 and Γ_1, respectively (see Fig. 12a). The following Dirichlet boundary conditions are prescribed

$$u(x,y) = \begin{cases} -1, & \text{if } (x,y) \in \Gamma_0, \\ 1, & \text{if } (x,y) \in \Gamma_1. \end{cases} \tag{129}$$

The diffusion tensor $\mathscr{D}$ is a symmetric positive definite matrix defined as

$$\mathscr{D} = \mathscr{R}(-\theta) \begin{pmatrix} k_1 & 0 \\ 0 & k_2 \end{pmatrix} \mathscr{R}(\theta), \tag{130}$$

h	E_1	E_2	p	$u_{\min}$	$u_{\max}$
1/18	0.826E-01	0.194E+00		-1.06565	1.00000
1/36	0.514E-01	0.136E+00	0.68	-1.05527	1.00000
1/72	0.298E-01	0.904E-01	0.79	-1.03944	1.00000
1/144	0.155E-01	0.544E-01	0.94	-1.01818	1.00000
1/288	0.684E-02	0.278E-01	1.18	-1.00133	1.00000
1/576	0.225E-02	0.103E-01	1.60	-1.00000	1.00000

Table 7 Anisotropic diffusion: Galerkin grid convergence.

h	E_1	E_2	p	NNL-A	NNL
1/18	0.741E-01	0.181E+00		70	258
1/36	0.441E-01	0.128E+00	0.75	293	1,136
1/72	0.257E-01	0.874E-01	0.78	448	4,904
1/144	0.143E-01	0.547E-01	0.85	951	20,375
1/288	0.712E-02	0.292E-01	1.01	1,094	51,763
1/576	0.245E-02	0.111E-01	1.54	1,976	120,213

Table 8 Anisotropic diffusion: LPFL grid convergence.

where k_1 and k_2 are the positive eigenvalues and $\mathscr{R}(\theta)$ is a rotation matrix

$$\mathscr{R}(\theta) = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}. \tag{131}$$

The eigenvalues of $\mathscr{D}$ represent the diffusion coefficients associated with the axes of the Cartesian coordinate system rotated by the angle θ. Let

$$k_1 = 100, \qquad k_2 = 1, \qquad \theta = -\frac{\pi}{6}.$$

By the continuous maximum principle, the exact solution to the above Dirichlet problem is bounded by the prescribed boundary data $u|_\Gamma = \pm 1$. However, the diffusion tensor (130) is highly anisotropic, which may result in a violation of the DMP even if a regular mesh of acute/nonnarrow type is employed.

The above benchmark problem was introduced by Lipnikov et al. [53]. The results obtained with LPSL can be found in [51]. In this section, we discretize the anisotropic diffusion equation (128) using LPFL and linear finite elements on uniform triangular meshes. Since no exact solution is available, the reference solution depicted in Fig. 12b is calculated with the standard Galerkin method on a very fine mesh ($h = 1/1152$). This solution is bounded by the prescribed Dirichlet boundary values, as required by the maximum principle. The unconstrained Galerkin solutions computed on coarser meshes exhibit spurious undershoots shown as the dark blue regions in Figs 12c and 12d. Algebraic flux correction based on the LPFL algorithm makes it possible to enforce the DMP constraint without excessive smearing. The solutions for $h = 1/36$ and $h = 1/288$ are presented in Figs 12e and 12f.

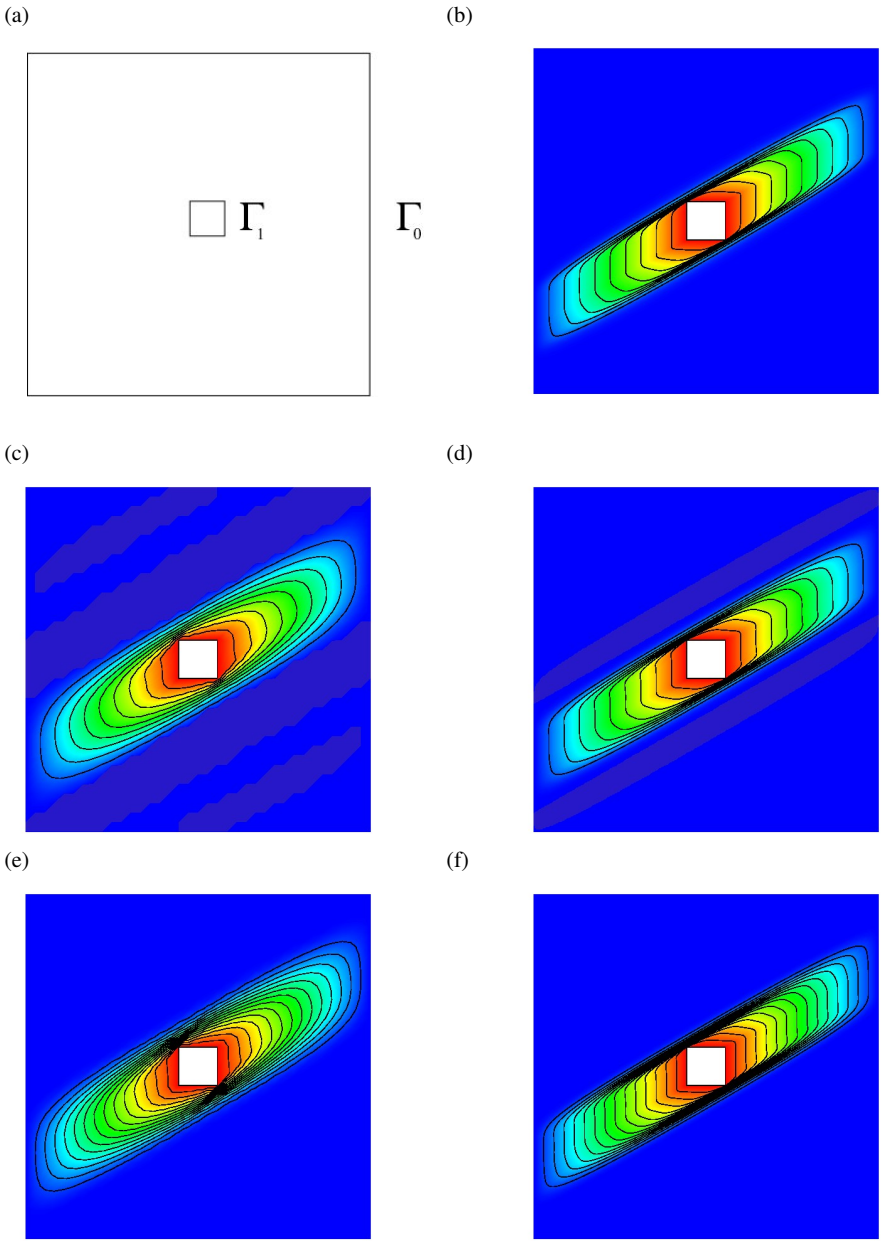

Fig. 12 Anisotropic diffusion: (a) domain geometry, (b) reference solution, (c) Galerkin, $h = 1/36$, (d) Galerkin, $h = 1/288$, (e) LPFL, $h = 1/36$, (f) LPFL, $h = 1/288$.

The results of the grid convergence study are summarized in Tables 7 and 8. On coarse meshes, the LPFL algorithm produces smaller errors than the underlying Galerkin scheme. As the mesh is refined, the undershoots produced by the latter method become smaller and eventually disappear. In the fourth column, we list the rate of convergence (124) for each pair of meshes. Note that the value of p increases monotonically as the mesh size h goes to zero.

The nonlinearity of the algebraic system associated with the flux-corrected Galerkin discretization of the anisotropic diffusion equation is more severe than in the case of pure convection. This phenomenon was first discovered in [51]. The last two columns in Table 8 list the total number of nonlinear SSOR iterations required to make the maximum norm of the residual smaller than $\varepsilon = 10^{-6}$. It is worth mentioning that the values of E_1 and E_2 converged at early stages of the iteration process. Hence, a better choice of stopping criteria would make the iterative solver more efficient [2]. The numbers in the column labeled NNL-A were obtained with Anderson acceleration, as described in Section 8.3. If it is switched off, a dramatic increase in the number of nonlinear iterations NNL is observed (see the last column in Table 8). The accelerated version is 60 times faster on the finest mesh.

In the current implementation of Anderson acceleration, we always mix $k = 5$ iterates and calculate the corresponding weights using the LAPACK subroutine DGELS to solve the (unconstrained) least squares problem. The improvements proposed in [18, 19, 65, 80] are likely to result in a further gain of efficiency.

10 Summary and Outlook

The algebraic flux correction paradigm presented in this chapter provides a set of general rules, concepts, and tools for enforcing the discrete maximum principle and positivity preservation in the context of low-order finite element approximations on arbitrary meshes. The presented methodology is based on a generalization of FCT. In particular, we addressed the design of implicit FCT schemes, developed a linearity-preserving slope limiter and converted it into a fully multidimensional format. In contrast to FCT, the new approach to flux correction is well-suited not only for time-dependent problems but also for steady transport equations.

The use of flux limiting gives rise to a nonlinear system which must be linearized or solved in an iterative way. The former approach has led us to an efficient predictor-corrector algorithm for computations with small time steps. In the case of stationary transport equations or large time steps, the linearization of antidiffusive fluxes about a low-order predictor would degrade the accuracy of the algebraic flux correction scheme and inhibit convergence. Hence, there is no way to replace the iterative solution of a nonlinear system with a single postprocessing step. Our results for the anisotropic diffusion equation indicate that Anderson acceleration is a very useful tool for the design of efficient quasi-Newton iterative solvers. The nonlinear SSOR method presented in this paper can also be used as a smoother within the framework of a full multigrid / full approximation scheme (FMG-FAS).

The generality of algebraic flux correction makes it very powerful. The same limiter routine can be employed to enforce positivity constraints in 2D and 3D, on structured and unstructured meshes. The origin of discrete operators makes no difference as far as the M-matrix property is concerned. However, the flux limiter must be designed to keep the perturbation of the discrete problem as small as possible. The demand for high resolution is particularly difficult to meet in the case of higher-order finite elements because the fluxes may depend on solution values at more than two nodes, and even the construction of an optimal low-order scheme becomes a nontrivial task [41]. This has led us to believe that higher-order Galerkin schemes must be constrained within the framework of hp-adaptivity. In regions where the derivatives of order $p \geq 1$ are smooth, no limiting is required. Otherwise, the polynomial degree p must be reduced until a smooth derivative is found [43] or a (multi-) linear approximation ($p = 1$) is recovered in a given element. In the latter case, flux limiting can be performed using the methodology presented in this chapter.

The unavoidable loss of accuracy around internal and boundary layers can be compensated using h-adaptation, i.e., local mesh refinement. The Galerkin orthogonality error produced by the flux limiter is computable and easy to localize. Thus, it provides valuable feedback for goal-oriented mesh adaptation [47].

In the next two chapters, we extend algebraic flux correction to systems of conservation laws including the compressible Euler and incompressible Navier-Stokes equations. The topics to be addressed include the construction of artificial viscosity operators, flux limiting in terms of nonconservative variables, synchronization of the correction factors, and failsafe control of the solution behavior. We also discuss the treatment of source/sink terms in the context of the $k-\varepsilon$ turbulence model.

References

[1] D.G. Anderson, Iterative procedures for nonlinear integral equations. *J. Assoc. Comput. Machinery* **12** (1965) 547–560.

[2] M. Arioli, D. Loghin, and A. J. Wathen, Stopping criteria for iterations in finite element methods. *Numer. Math.* **99** (2006) 381–410.

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

[4] J.D. Baum and R. Löhner, Numerical simulation of pilot/seat ejection from an F-16. *AIAA Paper*, 93-0783, 1993.

[5] P. Bochev, D. Ridzal, G. Scovazzi, and M. Shashkov, *Constrained-Optimization Based Data Transfer: A New Perspective on Flux Correction*. Chapter 10 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] D. L. Book, *The Conception, Gestation, Birth, and Infancy of FCT.* Chapter 1 in this volume.

[8] D.L. Book, J.P. Boris, and K. Hain, Flux-Corrected Transport: II. Generalizations of the Method. *J. Comput. Phys.* **18** (1975) 248–283.

[9] J.P. Boris and D.L. Book, Flux-Corrected Transport: III. Minimal-error FCT algorithms. *J. Comput. Phys.* **20** (1976) 397–431.

[10] J.-C. Carette, H. Deconinck, H. Paillère, and P.L. Roe, Multidimensional upwinding: Its relation to finite elements. *Int. J. Numer. Methods Fluids* **20** (1995) 935–955.

[11] L. Catabriga and A.L.G.A. Coutinho, Implicit SUPG solution of Euler equations using edge-based data structures. *Comput. Methods Appl. Mech. Engrg.* **191** (2002) 3477–3490.

[12] P.G. Ciarlet, Discrete maximum principle for finite-difference operators. *Aequationes Math.* **4** (1970) 338–352.

[13] P.G. Ciarlet and P.-A. Raviart, Maximum principle and convergence for the finite element method. *Comput. Methods Appl. Mech. Engrg.* **2** (1973) 17–31.

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

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

[16] J. Donea and A. Huerta, *Finite Element Methods for Flow Problems*. John Wiley & Sons, Chichester, 2003.

[17] J. Donea, S. Giuliani, H. Laval, and L. Quartapelle, Time-accurate solution of advection-diffusion equations by finite elements. *Comput. Methods Appl. Mech. Engrg.* **193** (1984) 123–145.

[18] V. Eyert, A comparative study on methods for convergence acceleration of iterative vector sequences. *J. Comput. Phys.* **124** (1996) 271–285.

[19] H. Fang and Y. Saad, Two classes of multisecant methods for nonlinear acceleration. *Numer. Linear Algebra Appl.* **16** (2009) 197–221.

[20] I. Faragó and R. Horváth, Continuous and discrete parabolic operators and their qualitative properties. *IMA J. Numer. Anal.* **29** (2009) 606–631.

[21] I. Faragó, R. Horváth, and S. Korotov, Discrete maximum principle for linear parabolic problems solved on hybrid meshes. *Appl. Numer. Math.* **53** (2005) 249–264.

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

[23] C.A.J. Fletcher, A comparison of finite element and finite difference solutions of the one- and two-dimensional Burgers' equations. *J. Comput. Phys.* **51** (1983) 159–188.

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

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

[26] S. Gottlieb, C.-W. Shu, and E. Tadmor, Strong stability-preserving high-order time discretization methods. *SIAM Review* **43** (2001) 89–112.

[27] M. Gurris, D. Kuzmin, and S. Turek, Implicit finite element schemes for the stationary compressible Euler equations. *Int. J. Numer. Methods Fluids*. In press, DOI: 10.1002/fld.2532.

[28] P. Hansbo, Aspects of conservation in finite element flow computations. *Comput. Methods Appl. Mech. Engrg.* **117** (1994) 423–437.

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

[30] M.E. Hubbard, Non-oscillatory third order fluctuation splitting schemes for steady scalar conservation laws. *J. Comput. Phys.* **222** (2007) 740–768.

[31] W. Hundsdorfer and J.G. Verwer, *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*. Springer, 2003.

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

[33] 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.

[34] A. Jemcov and J.P. Maruszewski, Algorithm stabilization and acceleration in computational fluid dynamics: exploiting recursive properties of fixed point algorithms. In: R.S. Amano and B. Sundén (eds) *Computational Fluid Dynamics and Heat Transfer*, WIT Press, 2010.

[35] V. John and E. Schmeyer, On finite element methods for 3D time-dependent convection-diffusion-reaction equations with small diffusion. *Comput. Meth. Appl. Mech. Engrg.* **198** (2008) 475-494.

[36] J. Karátson and S. Korotov, Discrete maximum principles for finite element solutions of nonlinear elliptic problems with mixed boundary conditions. *Numer. Math.* **99** (2005) 669–698.

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

[38] 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, 887–888.

[39] 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.

[40] D. Kuzmin, Algebraic flux correction for finite element discretizations of coupled systems. In: E. Oñate, M. Papadrakakis, B. Schrefler (eds) *Computational Methods for Coupled Problems in Science and Engineering II*, CIMNE, Barcelona, 2007, 653–656.

[41] D. Kuzmin, On the design of algebraic flux correction schemes for quadratic finite elements. *J. Comput. Appl. Math.* **218**:1 (2008) 79–87.

[42] D. Kuzmin, Explicit and implicit FEM-FCT algorithms with flux linearization. *J. Comput. Phys.* **228** (2009) 2517-2534.

[43] D. Kuzmin, A vertex-based hierarchical slope limiter for p-adaptive discontinuous Galerkin methods. *J. Comput. Appl. Math.* **233** (2010) 3077–3085.

[44] D. Kuzmin, *A Guide to Numerical Methods for Transport Equations*, University Erlangen-Nuremberg, 2010, http://www.mathematik.uni-dortmund.de/~kuzmin/Transport.pdf.

[45] D. Kuzmin, Linearity-preserving flux correction and convergence acceleration for constrained Galerkin schemes. To appear in *J. Comput. Appl. Math.* (2012).

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

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

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

[49] D. Kuzmin and S. Turek, High-resolution FEM-TVD schemes based on a fully multidimensional flux limiter. *J. Comput. Phys.* **198** (2004) 131-158.

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

[51] D. Kuzmin, M.J. Shashkov, and D. Svyatskiy, A constrained finite element method satisfying the discrete maximum principle for anisotropic diffusion problems. *J. Comput. Phys.* **228** (2009) 3448-3463.

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

[53] K. Lipnikov, M. Shashkov, D. Svyatskiy, and Yu. Vassilevski, Monotone finite volume schemes for diffusion equations on unstructured triangular and shape-regular polygonal meshes. *J. Comput. Phys.* **227** (2007) 492–512.

[54] R. Löhner, *Applied CFD Techniques: An Introduction Based on Finite Element Methods* (2nd edition). John Wiley & Sons, Chichester, 2008.

[55] R. Löhner, Edges, stars, superedges and chains. *Comput. Methods Appl. Mech. Engrg.* **111** (1994) 255–263.

[56] R. Löhner and M. Galle, Minimization of indirect addressing for edge-based field solvers. *Commun. Numer. Methods Eng.* **18**:5 (2002) 335–343.

[57] 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.

[58] 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.

[59] H. Luo, J.D. Baum, and R. Löhner, A Fast, Matrix-free Implicit Method for Compressible Flows on Unstructured Grids. *J. Comput. Phys.* **146** (1998) 664–690.

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

[61] P.R.M. Lyra, R.B. Willmersdorf, M.A.D. Martins, and A.L.G.A. Coutinho, Parallel implementation of edge-based finite element schemes for compress-

ible flow on unstructured grids, Proceedings of the 3rd International Meeting on Vector and Parallel Processing, Faculdade de Engenharia da Universidade do Porto, Porto, Portugal, 21.-23. Juni 1998.

[62] K. Mer, Variational analysis of a mixed element/volume scheme with fourth-order viscosity on general triangulations. *Comput. Methods Appl. Mech. Engrg.* **153** (1998) 45–62.

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

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

[65] P. Ni, *Anderson Acceleration of Fixed-point Iteration with Applications to Electronic Structure Computations*. PhD thesis, Worcester Polytechnic Institute, 2009.

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

[67] E.S. Oran and J.P. Boris, *Numerical Simulation of Reactive Flow* (2nd edition). Cambridge University Press, 2001.

[68] A. K. Parrott and 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.

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

[70] 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, 1993, 221-239.

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

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

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

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

[75] P. van Slingerland, *An Accurate and Robust Finite Volume Method for the Advection Diffusion Equation*. M.Sc. thesis, Delft University of Technology, June 2007.

[76] P. van Slingerland, M. Borsboom, and C. Vuik, A local theta scheme for advection problems with strongly varying meshes and velocity profiles. Report **08-17**, Department of Applied Mathematical Analysis, Delft University of Technology, June 2008.

[77] D.A. Smith, W.F. Ford, and A. Sidi, Extrapolation methods for vector sequences. *SIAM Review* **29** (1987) 199–233.
[78] P.K. Sweby, High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM J. Numer. Anal.* **21** (1984) 995–1011.
[79] R. S. Varga, *Matrix Iterative Analysis*. Prentice-Hall, Englewood Cliffs, 1962.
[80] H.W. Walker and P. Ni, Anderson acceleration for fixed-point iterations. *SIAM J. Numer. Anal.* **49** (2011) 1715–1735.
[81] S.T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids. *J. Comput. Phys.* **31** (1979) 335–362.