Linearity-preserving flux correction and convergence acceleration for constrained Galerkin schemes

Dmitri Kuzmin

Applied Mathematics III, University Erlangen-Nuremberg
Haberstr. 2, D-91058, Erlangen, Germany

Abstract

This paper is concerned with the development of general-purpose algebraic flux correction schemes for continuous (linear and multilinear) finite elements. In order to enforce the discrete maximum principle (DMP), we modify the standard Galerkin discretization of a scalar transport equation by adding diffusive and antidiffusive fluxes. The result is a nonlinear algebraic system satisfying the DMP constraint. An estimate based on variational gradient recovery leads to a linearity-preserving limiter for the difference between the function values at two neighboring nodes. A fully multidimensional version of this scheme is obtained by taking the sum of local bounds and constraining the total flux. This new approach to algebraic flux correction provides a unified treatment of stationary and time-dependent problems. Moreover, the same algorithm is used to limit convective fluxes, anisotropic diffusion operators, and the antidiffusive part of the consistent mass matrix.

The nonlinear algebraic system associated with the constrained Galerkin scheme is solved using fixed-point defect correction or a nonlinear SSOR method. A dramatic improvement of nonlinear convergence rates is achieved with the technique known as *Anderson acceleration* (or *Anderson mixing*). It blends a number of last iterates in a GMRES fashion, which results in a Broyden-like quasi-Newton update. The numerical behavior of the proposed algorithms is illustrated by a grid convergence study for convection-dominated transport problems and anisotropic diffusion equations in 2D.

Keywords: transport equations, discrete maximum principle, linearity preservation, slope limiting, flux correction, Anderson acceleration

Email address: kuzmin@am.uni-erlangen.de (Dmitri Kuzmin)

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 oscillations (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 nonphysical artifacts in proximity to unresolved small-scale features [30, 33]. 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.

The traditional approach to stabilization of finite element schemes for convection-dominated transport problems involves adding artificial diffusion or using modified test functions in the weak form of the governing equations. We refer to John et al. [20, 21, 22] for a comprehensive survey and a comparative study of such variational stabilization techniques. Their practical utility is undermined by the presence of problem-dependent free parameters. The failure to find a 'right' value of these parameters may result in a violation of the maximum principle or give rise to excessive numerical diffusion.

A fundamentally different way to enforce the discrete maximum principle in CFD codes is the use of flux or slope limiting. High-resolution schemes based on this design philosophy trace their origins to the *flux-corrected transport* (FCT) algorithm [6, 50]. The basic idea is very simple: use a given high-order scheme in smooth regions and a nonoscillatory low-order approximation elsewhere. The work of Harten [15] and Sweby [47] has established a rigorous theoretical framework for the design of *total variation diminishing* (TVD) limiters in 1D. The implementation of FCT and TVD in finite element codes dates back to the late 1980s [2, 35, 43, 44]. The development of edge-based data structures [36, 38, 42, 45] has formed the basis for many straightforward generalizations of 1D limiting techniques to unstructured grids [38].

The principle of *algebraic flux correction* introduced by the author and his collaborators [24, 25, 28] offers a new interpretation of classical high-resolution schemes and a general framework for the design of multidimensional flux limiters. In contrast to the mainstream approach, we add and

remove artificial diffusion at the discrete level. Given a discrete operator resulting from a linear or multilinear Galerkin approximation, we extract its nonoscillatory low-order part. The remainder is an antidiffusive correction which admits a conservative flux decomposition [28]. The discrete maximum principle holds if the antidiffusive part proves *local extremum diminishing* (LED). The purpose of flux limiting is to enforce Jameson's LED constraint [17, 18] by adjusting the magnitudes of antidiffusive fluxes if necessary.

During the last decade, we have experimented with many algebraic flux correction schemes which are based on the same design principles and differ only in the definition of the LED bounds for the sum of limited antidiffusive fluxes. The first representative of such schemes was an implicit version of the FEM-FCT algorithm [23, 26, 28, 31]. The in-depth comparative study by John and Schmeyer [22] indicates that FEM-FCT is far superior to mainstream stabilization techniques for finite elements when it comes to solving strongly time-dependent transport problems with small or vanishing diffusion. However, flux correction of FCT type turns out to be inappropriate for steady-state computations since the results depend on the pseudo-time step, and severe convergence problems may occur. Moreover, the use of large time steps increases the amount of numerical diffusion. Thus, we recommend FCT for truly evolutionary problems which require the use of small time steps.

As an alternative to FCT, we have developed several multidimensional flux limiters which are independent of the time step and produce a TVD scheme in the 1D case [24, 25, 28]. As this methodology has evolved and matured, we realized that the definition of 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 [7, 39]. In the context of algebraic flux correction, it can be enforced using variational gradient recovery to obtain the LED bounds for the slope limiter [30].

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 [24]. 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 paper, we present a new linearity-preserving (LP) algebraic flux correction scheme that can handle both situations equally well. This concludes our quest for the development of a universal flux limiter for general-purpose CFD codes.

The algorithm to be presented is a fully multidimensional counterpart of the slope limiter we developed in [30] 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 LP 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 unidirectional slope limiting. In contrast to [24], the antidiffusive fluxes associated with convective transport, anisotropic diffusion, and mass lumping errors are limited separately. Moreover, there is no need for *ad hoc* 'prelimiting', a trick which is frequently employed in FCT algorithms to ensure that the antidiffusive correction has a steepening effect on the solution profiles.

Another highlight of the present paper is a new iterative solver for the constrained Galerkin approximation. We introduce 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* [3, 49], also known as *Anderson mixing* [9, 10]. As shown by Eyert [9], the accelerated iterative solver belongs to the Broyden family of Jacobian-free quasi-Newton methods. In the case of a linear system, it is essentially equivalent to the (preconditioned) GMRES method [41, 49]. 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.

The paper is structured as follows. In the next section, we discretize a linear convection-diffusion equation using the (continuous) Galerkin method. In Sections 3-5, we formulate sufficient conditions of a discrete maximum principle and introduce the new linearity-preserving limiter for algebraic flux correction schemes. In Section 6, we address the design and acceleration of

iterative solvers for the nonlinear algebraic system. A grid convergence study for three benchmark problems is presented in Section 7. Finally, we summarize the results and outline some promising directions for further research.

2. Galerkin discretization

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

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

which describes the transport of a conserved quantity u in a bounded domain $\Omega \subset \mathbb{R}^n,\ n \in \{1, 2, 3\}$. The velocity $\mathbf{v}$ and the diffusion tensor $\mathcal{D}$ are assumed to be known. The Dirichlet-Neumann boundary conditions are given by

$$\begin{cases} u = g & \text{on } \Gamma_D, \\ \mathbf{n} \cdot \nabla u = 0 & \text{on } \Gamma_N, \end{cases} \tag{2}$$

where $\mathbf{n}$ is the unit outward normal to the boundary $\Gamma = \partial\Omega$. If $\mathcal{D} \neq 0$ then $\Gamma_N \cup \Gamma_D = \Gamma$. In the hyperbolic limit ($\mathcal{D} = 0$) we have $\Gamma_N = \emptyset$ and

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

If the steady-state solution to (1) is of interest, then the problem statement is complete. Otherwise, we prescribe an initial condition of the form

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

The variational form of the above (initial-)boundary value problem reads

$$\int_\Omega w \left(\frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{v}u) \right) \mathrm{d}\mathbf{x} + \int_\Omega \nabla w \cdot (\mathcal{D}\nabla u) \, \mathrm{d}\mathbf{x} = 0 \tag{4}$$

for all admissible test functions w vanishing on the Dirichlet boundary Γ_D.

In this paper, we discretize (4) in space using the Galerkin finite element method. Let $\{\varphi_j\}$ denote a finite set of continuous piecewise-linear or multilinear basis functions. The numerical solution u_h is defined as

$$u_h = \sum_j u_j \varphi_j. \tag{5}$$

The unknowns of the semi-discrete problem are the coefficients u_j which represent the time-dependent values of u_h at the vertices of the mesh. The diffusive term is evaluated using the consistent discrete gradient

$$\nabla u_h = \sum_j u_j \nabla \varphi_j. \tag{6}$$

The Galerkin discretization of the convective term is given by the formula

$$\nabla \cdot (\mathbf{v} u_h) = (\nabla \cdot \mathbf{v}) u_h + \mathbf{v} \cdot \nabla u_h. \tag{7}$$

In many cases, it is more convenient to work with Fletcher's [12, 13] *group finite element* interpolant of the convective flux. We define

$$(\mathbf{v}u)_h = \sum_j (\mathbf{v}_j u_j) \varphi_j \tag{8}$$

which implies the following discretization of the convective term

$$\nabla \cdot (\mathbf{v}u)_h = \sum_j u_j (\mathbf{v}_j \cdot \nabla \varphi_j). \tag{9}$$

Using approximations (5), (6), and (9) in the Galerkin weak form (4) with the test function $w_h = \varphi_i$, we obtain the following semi-discrete equation

$$\begin{aligned} \sum_j \left(\int_\Omega \varphi_i \varphi_j \, \mathrm{d}\mathbf{x} \right) \frac{\mathrm{d}u_j}{\mathrm{d}t} = & - \sum_j \mathbf{v}_j \cdot \left(\int_\Omega \varphi_i \nabla \varphi_j \, \mathrm{d}\mathbf{x} \right) u_j \\ & - \sum_j \left(\int_\Omega \nabla \varphi_i \cdot (\mathcal{D} \nabla \varphi_j) \, \mathrm{d}\mathbf{x} \right) u_j. \end{aligned} \tag{10}$$

The system of equations for all unknowns can be written in the generic form

$$M_C \frac{\mathrm{d}u}{\mathrm{d}t} = (K - L) u, \tag{11}$$

where u is the vector of unknowns and $M_C = \{m_{ij}\}$ is the consistent mass matrix. The convective and diffusive part of the discrete transport operator are denoted by $K = \{k_{ij}\}$ and $L = \{l_{ij}\}$, respectively. According to (10)

$$m_{ij} = \int_\Omega \varphi_i \varphi_j \, \mathrm{d}\mathbf{x}, \qquad l_{ij} = \int_\Omega \nabla \varphi_i \cdot (\mathcal{D} \nabla \varphi_j) \, \mathrm{d}\mathbf{x}, \tag{12}$$

and

$$k_{ij} = -\mathbf{v}_j \cdot \mathbf{c}_{ij}, \qquad \mathbf{c}_{ij} = \int_\Omega \varphi_i \nabla \varphi_j \, \mathrm{d}\mathbf{x}. \tag{13}$$

In the case of an unsteady velocity field, the convective part of the discrete transport operator must be updated at each time step. If the mesh is fixed, then the coefficients $\mathbf{c}_{ij}$ of the discrete gradient operator $\mathbf{C} = \{\mathbf{c}_{ij}\}$ 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 $0 = t_0 < t^1 < t^2 < \ldots < t^M = T$ be a sequence of discrete time levels for the time integration of system (11). For simplicity, we assume that the time step $\Delta t = t^{n+1} - t^n$ is constant so that $t^n = n\Delta t$. We have

$$M_C(u^{n+1} - u^n) = \int_{t^n}^{t^{n+1}} (K - L)u \, \mathrm{d}t. \tag{14}$$

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

$$[M_C - \theta\Delta t(K - L)]u^{n+1} = [M_C + (1 - \theta)\Delta t(K - L)]u^n, \tag{15}$$

where $\theta \in [0, 1]$ is the degree of implicitness. We remark that 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.

3. Discrete maximum principles

Under certain assumptions, one can prove that the solution of the continuous problem is bounded by its initial and/or boundary values. A survey of maximum principles for elliptic, hyperbolic, and parabolic transport equations can be found in [27]. Of course, a good numerical scheme must preserve important properties of the exact solution. In this section, we briefly review algebraic constraints which imply a discrete maximum principle and guarantee positivity preservation. In the next section, we will use these sufficient conditions to constrain discrete Galerkin operators in an adaptive way.

Given an approximation of the form (15) or its steady-state counterpart ($u^{n+1} = u^n = u$), the extended linear system can be partitioned as follows

$$\begin{pmatrix} A_{\Omega\Omega} & A_{\Omega\Gamma} \\ 0 & I \end{pmatrix} \begin{pmatrix} u_\Omega \\ u_\Gamma \end{pmatrix} = \begin{pmatrix} B_{\Omega\Omega} & B_{\Omega\Gamma} \\ 0 & I \end{pmatrix} \begin{pmatrix} g_\Omega \\ g_\Gamma \end{pmatrix}, \tag{16}$$

where u_Ω is the vector of unknowns and $u_\Gamma = g_\Gamma$ is the vector of Dirichlet boundary values. In the case of unsteady problems and pseudo-time stepping schemes, we have $u_\Omega = u_\Omega^{n+1}$ and $g_\Omega = u_\Omega^n$. If the (pseudo-)time derivative is omitted, one obtains a system of the form (16) with $B_{\Omega\Omega} = 0$ and $B_{\Omega\Gamma} = 0$.

The algebraic equation for the $i-$th component of u_Ω can be written as

$$a_{ii}u_i = b_{ii}g_i + \sum_{j\in S_i}(b_{ij}g_j - a_{ij}u_j), \tag{17}$$

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

DEFINITION 1. A numerical scheme of the form (17) satisfies the local *discrete maximum principle* (DMP) at node i if [4]

$$u_i \leq u_i^{\max} := \max\{g_i, \max_{j\in S_i} g_j, \max_{k\in S_i} u_k\} \tag{18}$$

In a similar vein, the local *discrete minimum principle* holds at node i if

$$u_i \geq u_i^{\min} := \min\{g_i, \min_{j\in S_i} g_j, \min_{k\in S_i} u_k\} \tag{19}$$

Importantly, this property implies local positivity preservation. That is,

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

THEOREM 1. *Suppose that the coefficients of the i-th equation (18) satisfy*

$$a_{ii} > 0, \quad b_{ii} \geq 0, \quad a_{ij} \leq 0, \quad b_{ij} \geq 0, \qquad \forall j \in S_i. \tag{20}$$

Then (18) is locally positivity-preserving. Moreover, the local discrete maximum and minimum principles hold under the additional condition [11, 27]

$$\sum_j a_{ij} = \sum_j b_{ij}. \tag{21}$$

PROOF. Suppose that $u_i^{\min} \geq 0$, which implies $u_j \geq 0,\ \forall j \in S_i$ and $g_j \geq 0$, $\forall j \in S_i \cup \{i\}$. Due to (20), the right-hand side of (17) is nonnegative and $a_{ii} > 0$, whence $u_i \geq 0$. Thus, the scheme is locally positivity-preserving.

To prove the local DMP property, we introduce $w_j := u_j - u_i^{\max}$ and $v_j := g_j - u_i^{\max}$. By definition, $w_j \leq 0,\ \forall j \in S_i$ and $v_j \leq 0, \forall j \in S_i \cup \{i\}$. Using (17) and the additional row sum condition (21), we obtain

$$a_{ii}w_i = b_{ii}v_i + \sum_{j\in S_i}(b_{ij}v_j - a_{ij}w_j). \tag{22}$$

Due to (20), the right-hand side of this equation is nonpositive and $a_{ii} > 0$. It follows that $w_i \leq 0$, i.e., $u_i \leq u_i^{\max}$. Similarly, we have $u_i \geq u_i^{\min}$. $\square$

The above Theorem implies that the solution value u_i is bounded by the solution values in a neighborhood of node i. If this is the case for all nodes, global maxima and minima must occur on the Dirichlet boundary or at the previous time level, in accordance with the continuous maximum principle.

DEFINITION 2. A discretization of the form (16) satisfies the global discrete maximum/minimum principle for nodal values if

$$\min g \leq u_\Omega \leq \max g, \tag{23}$$

where $g = (g_\Omega, g_\Gamma)^T$ is the vector of initial/boundary data for problem (16).

The corresponding definition of global positivity preservation is as follows

$$g \geq 0 \quad \Rightarrow \quad u \geq 0.$$

Again, a typical proof imposes certain restrictions of the coefficients of the discrete problem. We will need the following set of sufficient conditions:

THEOREM 2. *Consider a discrete problem of the form $Au = Bg$, where all entries of A and B satisfy conditions (20). If A is strictly or irreducibly diagonally dominant, then the discretization is globally positivity-preserving. Moreover, the global DMP holds if the row sums of A and B are equal.*

PROOF. We refer to [27] for a proof based on the concept of monotonicity.

DEFINITION 3. A regular matrix A is called monotone if $A^{-1} \geq 0$ [48]. An equivalent definition of monotonicity is: $Au \geq 0$ for any vector $u \geq 0$.

Obviously, the solution to $Au = Bg$ is positivity-preserving if A is monotone and $B \geq 0$ so that $u^{n+1} = A^{-1}Bu^n \geq 0$ whenever $u^n \geq 0$. If A and B have equal row sums, one can prove the global DMP for nodal values.

4. Algebraic flux correction

If convective effects are too strong or the diffusion tensor is anisotropic, then the standard Galerkin discretization (15) fails to satisfy the conditions of Theorems 1 and 2 even on a uniform mesh of linear or multilinear finite elements. This may result in a violation of the discrete maximum principle and give rise to nonphysical negative values. To suppress undershoots/overshoots

and ensure positivity preservation, we will adjust the coefficients of the Galerkin scheme so as to enforce conditions (20) for an equivalent nonlinear problem. We call this methodology *algebraic flux correction* [24, 28].

The discrete problem (15) associated with the implicit Galerkin approximation of equation (1) is a linear system of the form $Au^{n+1} = Bu^n$. The diagonal entries of A and B are positive, at least for sufficiently small time steps Δt. However, a violation of conditions (20) 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, the contribution of these entries is constrained by adding a certain amount of discrete diffusion. First, the consistent mass matrix M_C is replaced with its lumped counterpart

$$M_L := \text{diag}\{m_i\}, \qquad m_i = \sum_j m_{ij}. \tag{24}$$

Next, we fix K by adding a discrete diffusion operator $D = \{d_{ij}\}$ with [23, 31]

$$d_{ij} = \max\{-k_{ij}, 0, -k_{ji}\} = d_{ji}, \qquad \forall j \neq i \tag{25}$$

so that $K + D$ has no negative off-diagonal coefficients. The diagonal entries of D are defined so that this symmetric matrix has zero row sums

$$d_{ii} := -\sum_{j \neq i} d_{ij}. \tag{26}$$

Due to symmetry, the column sums are also equal to zero. In the 1D case, the lumped-mass Galerkin approximation on a uniform mesh of linear finite elements is equivalent to the central difference scheme, while the modified operator $K + D$ corresponds to the first-order upwind difference [31].

If the computational mesh and/or the diffusion tensor are anisotropic, some off-diagonal entries of L may be strictly positive. We set them equal to zero and modify the diagonal entries so that the row and column sums remain unchanged. The result is the discrete diffusion operator $L^- := L - L^+$, where $L^+ = \{l_{ij}^+\}$ stands for the antidiffusive part of the stiffness matrix [30]

$$l_{ii}^+ := -\sum_{j \neq i} l_{ij}^+, \qquad l_{ij}^+ := \max\{0, l_{ij}\}, \quad \forall j \neq i. \tag{27}$$

In summary, the semi-discrete Galerkin scheme (11) can be split as follows

$$M_L \frac{\mathrm{d}u}{\mathrm{d}t} = (K + D - L^-)u + f(u), \tag{28}$$

where $f(u)$ is the sum of antidiffusive terms that may destroy positivity

$$f(u) = (M_L - M_C)\frac{\mathrm{d}u}{\mathrm{d}t} - (D + L^+)u. \tag{29}$$

Each component of this vector admits a conservative flux decomposition

$$f_i = \sum_{j \neq i} f_{ij}, \qquad f_{ji} = -f_{ij}. \tag{30}$$

The formula for f_{ij} follows from our definition of M_L, D, and L^+. We have

$$(M_L\dot{u} - M_C\dot{u})_i = m_i\dot{u}_i - \sum_j m_{ij}\dot{u}_j = \sum_{j\neq i} m_{ij}(\dot{u}_i - \dot{u}_j), \tag{31}$$

$$(Du)_i = \sum_j d_{ij}u_j = d_{ii}u_i + \sum_{j\neq i} d_{ij}u_j = \sum_{j\neq i} d_{ij}(u_j - u_i), \tag{32}$$

$$(L^+u)_i = \sum_j l^+_{ij}u_j = l^+_{ii}u_i + \sum_{j\neq i} l^+_{ij}u_j = \sum_{j\neq i} l^+_{ij}(u_j - u_i), \tag{33}$$

where $\dot{u}_i$ stands for $\left(\frac{\mathrm{d}u}{\mathrm{d}t}\right)_i$. Thus, the net antidiffusive flux f_{ij} is given by

$$f_{ij} = f^M_{ij} + f^K_{ij} + f^L_{ij}, \tag{34}$$

$$f^M_{ij} = m_{ij}\left(\dot{u}_i - \dot{u}_j\right), \tag{35}$$

$$f^K_{ij} = d_{ij}(u_i - u_j), \tag{36}$$

$$f^L_{ij} = l^+_{ij}(u_i - u_j). \tag{37}$$

By the symmetry of M_C, D, and L, we have $f_{ji} = -f_{ij}$ for all $j \neq i$. This property implies discrete conservation since the sum of all fluxes is zero.

The above representation of $f(u)$ makes it possible to undo the unnecessary modifications of the Galerkin operators in order to minimize the amount of numerical diffusion. To this end, we replace each flux f_{ij} with

$$\bar{f}_{ij} = \alpha^M_{ij} f^M_{ij} + \alpha^K_{ij} f^K_{ij} + \alpha^L_{ij} f^L_{ij}, \tag{38}$$

where $\alpha_{ij} \in [0,1]$ is a solution-dependent correction factor. The multiplication by α_{ij} is supposed to reduce the magnitude of the antidiffusive flux in regions where undershoots or overshoots would occur otherwise.

The semi-discrete form of the constrained Galerkin discretization reads

$$M_L \frac{\mathrm{d}u}{\mathrm{d}t} = (K + D - L^-)u + \bar{f}(u), \tag{39}$$

where

$$\bar{f}_i = \sum_{j \neq i} \bar{f}_{ij}, \qquad \bar{f}_{ji} = -\bar{f}_{ij}. \tag{40}$$

In accordance with the FCT philosophy [6, 35, 50] and the LED constraint [17, 18], the definition of $\bar{f}_{ij}$ must guarantee that the remaining antidiffusion cannot generate new local extrema or accentuate existing ones. Let

$$u_i^{\max} := \max\{u_i, \max_{j \in S_i} u_j\}, \tag{41}$$

$$u_i^{\min} := \min\{u_i, \min_{j \in S_i} u_j\}. \tag{42}$$

The local maxima and minima of the time derivative vector $\dot{u}$ are defined in the same way. We will denote them by $\dot{u}_i^{\max}$ and $\dot{u}_i^{\min}$, respectively.

The limited antidiffusive term (40) proves local extremum diminishing if

$$q_i^M(\dot{u}_i^{\min} - \dot{u}_i) \leq \sum_{j \neq i} \alpha_{ij}^M f_{ij}^M \leq q_i^M(\dot{u}_i^{\max} - \dot{u}_i), \tag{43}$$

$$q_i^K(u_i^{\min} - u_i) \leq \sum_{j \neq i} \alpha_{ij}^K f_{ij}^K \leq q_i^K(u_i^{\max} - u_i), \tag{44}$$

$$q_i^L(u_i^{\min} - u_i) \leq \sum_{j \neq i} \alpha_{ij}^L f_{ij}^L \leq q_i^L(u_i^{\max} - u_i) \tag{45}$$

for some positive constants q_i^M, q_i^K, and q_i^L independent of u. In the next section, we will use this criterion to determine the correction factors α_{ij}.

5. Linearity-preserving limiters

The purpose of flux limiting is to calculate a set of correction factors such that inequalities (43)–(45) hold for a given solution. In this paper, we use the same algorithm to determine the values of α_{ij}^M, α_{ij}^L, and α_{ij}^L. Without loss of generality, we consider raw antidiffusive fluxes of the form

$$f_{ij} = d_{ij}(u_i - u_j) \tag{46}$$

and present the generic limiting strategy that delivers $\alpha_{ij} = \alpha_{ji}$ satisfying

$$q_i(u_i^{\min} - u_i) \le \sum_{j \neq i} \alpha_{ij} f_{ij} \le q_i(u_i^{\max} - u_i) \tag{47}$$

for a given $q_i > 0$. Under this LED condition, the sum of limited antidiffusive fluxes is nonpositive if u_i is a local maximum and nonnegative if u_i is a local minimum. The trivial solution is $\alpha_{ij} = 0$ but a well-designed limiter should deliver $\alpha_{ij} \approx 1$ in regions where the Galerkin solution is smooth. In particular, the definition of q_i should guarantee that $\alpha_{ij} = 1$ is acceptable whenever the solution varies linearly in a neighborhood of node i. This important design principle is called *linearity preservation* [5, 7, 39]. It keeps the scheme consistent and implies second-order accuracy for smooth data.

5.1. Gradient-based slope limiting

To begin with, we present the symmetric linearity-preserving (LP) slope limiter we developed in [30] in the context of steady anisotropic diffusion. This algorithm belongs to the family of slope-limited positive (SLIP) methods in which the fluxes are constrained individually so as to limit the jumps of the solution gradient along the line connecting two nodes [17, 34, 38, 42].

Obviously, a raw antidiffusive flux of the form (46) requires limiting if the difference between the nodal values u_i and u_j is too large. In this case, the 'slope' $u_i - u_j$ should be replaced with its limited counterpart

$$\bar{s}_{ij} := \alpha_{ij}(u_i - u_j), \tag{48}$$

where $\alpha_{ij} \in [0, 1]$ is the correction factor for the limited antidiffusive flux

$$\bar{f}_{ij} = \alpha_{ij} d_{ij}(u_i - u_j) = d_{ij}\bar{s}_{ij}. \tag{49}$$

To derive a formula for $\bar{s}_{ij}$, we consider the following linear approximation

$$u_i - u_j \approx s_{ij} := \mathbf{g}_i \cdot (\mathbf{x}_i - \mathbf{x}_j), \tag{50}$$

where $\mathbf{g}_i$ is an approximation to ∇u at node i. In the context of SLIP schemes, $\mathbf{g}_i$ is commonly defined by differentiating the solution in the first element crossed by the line through $\mathbf{x}_i$ and $\mathbf{x}_j$. Alternatively, a continuous approximation $\mathbf{g}_h$ to ∇u can be constructed with superconvergent gradient

recovery techniques which are often used for error estimation purposes. We determine the nodal values of $\mathbf{g}_h$ using the lumped-mass L^2-projection

$$\mathbf{g}_i = \frac{1}{m_i} \sum_k \mathbf{c}_{ik} u_k, \tag{51}$$

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 (13).

Since the gradients of Lagrange basis functions sum to zero, we have $\sum_k \mathbf{c}_{ik} = \mathbf{0}$. This enables us to express the right-hand side of (51) thus:

$$\mathbf{g}_i = \frac{1}{m_i} \sum_{k \neq i} \mathbf{c}_{ik} (u_k - u_i). \tag{52}$$

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 $\bar{s}_{ij}$.

Plugging (52) into the definition of s_{ij}, we obtain the following estimates

$$s_{ij} \leq \frac{1}{m_i} \sum_{k \neq i} |\mathbf{c}_{ik} \cdot (\mathbf{x}_i - \mathbf{x}_j)| (u_i^{\max} - u_i), \tag{53}$$

$$s_{ij} \geq \frac{1}{m_i} \sum_{k \neq i} |\mathbf{c}_{ik} \cdot (\mathbf{x}_i - \mathbf{x}_j)| (u_i^{\min} - u_i). \tag{54}$$

To make the bounds less restrictive, we multiply them by 2 and define

$$\gamma_{ij} := \frac{2}{m_i} \sum_{k \neq i} |\mathbf{c}_{ik} \cdot (\mathbf{x}_i - \mathbf{x}_j)|. \tag{55}$$

Since the coefficient γ_{ij} is nonnegative, the LED constraint (47) holds if the limited slope $\bar{s}_{ij}$ that appears in the definition (49) of $\bar{f}_{ij}$ satisfies

$$\gamma_{ij}(u_i^{\min} - u_i) \leq \bar{s}_{ij} \leq \gamma_{ij}(u_i^{\max} - u_i). \tag{56}$$

If the flux $\bar{f}_{ji}$ does not require limiting, the following definition of $\bar{s}_{ij}$ will do

$$\bar{s}_{ij} = \begin{cases} \min\{\gamma_{ij}(u_i^{\max} - u_i), u_i - u_j\}, & \text{if } u_i > u_j, \\ \max\{\gamma_{ij}(u_i^{\min} - u_i), u_i - u_j\}, & \text{if } u_i < u_j. \end{cases} \tag{57}$$

The one-sided limiting strategy is appropriate if j is a node on the Dirichlet boundary or the original Galerkin operator is skew-symmetric (see Section

5.3). In all other cases, the contribution of $\bar{f}_{ji}$ to node j must also be LED. Hence, the limited slope $\bar{s}_{ij} = -\bar{s}_{ji}$ must satisfy not only (56) but also

$$\gamma_{ji}(u_j - u_j^{\max}) \le \bar{s}_{ij} \le \gamma_{ji}(u_j - u_j^{\min}). \tag{58}$$

A formula that guarantees the LED property for both nodes reads [30]

$$\bar{s}_{ij} = \begin{cases} \min\{\gamma_{ij}(u_i^{\max} - u_i), u_i - u_j, \gamma_{ji}(u_j - u_j^{\min})\}, & \text{if } u_i > u_j, \\ \max\{\gamma_{ij}(u_i^{\min} - u_i), u_i - u_j, \gamma_{ji}(u_j - u_i^{\max})\}, & \text{if } u_i < u_j. \end{cases} \tag{59}$$

This symmetric limiting strategy corresponds to a double application of the one-sided slope limiter (57). As the mesh is refined, the value of $\bar{s}_{ij}$ approaches $u_i - u_j$. Moreover, we can prove linearity preservation.

THEOREM 3. *If the numerical solution u_h is a linear function, then the lumped-mass L^2 projection (51) is exact and $s_{ij} = u_i - u_j = \bar{s}_{ij}$.*

PROOF. Suppose that u_h is a linear. Then the gradient of u_h 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 $\mathbf{g}_i = \nabla u_h$. According to (51), we have

$$\mathbf{g}_i = \frac{1}{m_i} \int_\Omega \varphi_i \nabla u_h \, \mathrm{d}\mathbf{x} = \nabla u_h \left(\frac{1}{m_i} \int_\Omega \varphi_i \, \mathrm{d}\mathbf{x} \right) = \nabla u_h \tag{60}$$

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 (56) and (58), whence no limiting is performed. □

5.2. Relationship to TVD schemes

To illustrate the relationship of the proposed slope limiters to classical TVD schemes, we consider a 1D mesh with uniform spacing Δx. In this case, the coefficients of (51) are given by $m_i = \Delta x$ and $c_{i\pm1/2} = \pm 1/2$. The resulting formula for g_i is equivalent to the second-order central difference

$$g_i = \frac{1}{2} \left[\frac{u_i - u_{i-1}}{\Delta x} + \frac{u_{i+1} - u_i}{\Delta x} \right] = \frac{u_{i+1} - u_{i-1}}{2\Delta x}.$$

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 (53)–(54) corresponds to

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

The one-sided slope limiter (57) can be written as a single-line formula

$$\bar{s}_{ij} = minmod\{2(u_{i-1} - u_i), u_i - u_{i+1}\}, \tag{61}$$

and the corresponding formula for the symmetric slope limiter (59) reads

$$\bar{s}_{ij} = minmod\{2(u_{i-1} - u_i), u_i - u_{i+1}, 2(u_{i+1} - u_{i+2})\}. \tag{62}$$

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

$$minmod\{a, b, \ldots\} = \begin{cases} \min\{a, b, \ldots\}, & \text{if} \quad a > 0,\ b > 0, \ldots \\ \max\{a, b, \ldots\}, & \text{if} \quad a < 0,\ b < 0, \ldots \\ 0, & \text{otherwise.} \end{cases} \tag{63}$$

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

5.3. Multidimensional flux limiting

So far we have limited 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 above stand-alone slope limiter into the format we have used to design fully multidimensional algebraic flux correction schemes of FCT and TVD type [24, 25, 27, 28].

A set of correction factors α_{ij} satisfying the generic LED constraint (47) for the sum of limited antidiffusive fluxes can be calculated as follows:

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

$$P_i^+ = \sum_{j \ne i} \max\{0, f_{ij}\}, \qquad P_i^- = \sum_{j \ne i} \min\{0, f_{ij}\}. \tag{64}$$

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

$$Q_i^+ = q_i(u_i^{\max} - u_i\}, \qquad Q_i^- = q_i(u_i^{\max} - u_i\}. \tag{65}$$

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

$$R_i^+ = \min\left\{1, \frac{Q_i^+}{P_i^+}\right\}, \qquad R_i^- = \min\left\{1, \frac{Q_i^-}{P_i^-}\right\}. \tag{66}$$

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

$$\alpha_{ij} \le \begin{cases} R_i^+, & \text{if } f_{ij} > 0, \\ R_i^-, & \text{if } f_{ij} < 0, \end{cases} \qquad \alpha_{ji} = \alpha_{ij}. \tag{67}$$

This limiting strategy traces its origins to Zalesak's FCT algorithm [50] but does not involve computation of a provisional low-order solution.

The above definition of α_{ij} implies (47). Indeed, it is easy to verify that

$$Q_i^- \le R_i^- P_i^- \le \sum_{j\neq i} \alpha_{ij} f_{ij} \le R_i^+ P_i^+ \le Q_i^+. \tag{68}$$

To maintain linearity preservation, we define $Q_i^\pm$ as the sum of the LED bounds we imposed in (56) on individual slopes/fluxes. That is, we set

$$q_i := \sum_{j\neq i} \gamma_{ij} d_{ij}. \tag{69}$$

It remains to define the correction factor α_{ij} for (67). This definition depends on whether a one-sided or a symmetric limiting strategy is appropriate.

Since the discrete convection operator K is nonsymmetric, the limiter can take advantage of the fact that k_{ji} and k_{ij} have opposite signs. Without loss of generality, we assume $k_{ij} < 0 \le k_{ji}$. This convention implies that node i is located *upwind* [24, 28]. The constrained entry of row j is given by

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

Thus $\bar{k}_{ij} \ge 0$ for any $\alpha_{ij} \in [0, 1]$, so it is safe to limit f_{ij} using $\alpha_{ij} = R_i^\pm$.

In the unlikely case of $k_{ij} \le k_{ji} < 0$, we redefine the flux f_{ij} as follows

$$f_{ij} := \mathit{minmod}\{f_{ij}, (k_{ji} + d_{ij})(u_i - u_j)\}, \qquad f_{ji} := -f_{ij}. \tag{70}$$

After this 'prelimiting', the value of $\bar{k}_{ji}$ will be nonnegative for any $\alpha_{ij} = R_i^{\pm}$.

In the one-sided version of the generic flux limiter (64)–(67), the contributions of upwind fluxes ($k_{ij} > k_{ji}$) are removed from the sums $P_i^{\pm}$

$$P_i^+ = \sum_{k_{ij} \le k_{ji}} \max\{0, f_{ij}\}, \qquad P_i^- = \sum_{k_{ij} \le k_{ji}} \min\{0, f_{ij}\}. \tag{71}$$

For each pair of off-diagonal coefficients $k_{ij} \le k_{ji}$, upwind-biased flux limiting is performed using the nodal correction factor for the upwind node

$$\alpha_{ij} = \begin{cases} R_i^+, & \text{if } f_{ij} \ge 0, \\ R_i^-, & \text{if } f_{ij} < 0, \end{cases} \qquad \alpha_{ji} := \alpha_{ij}. \tag{72}$$

In the case of a symmetric Galerkin operator like M_C and L, the antidiffusive flux may violate the LED constraint for both nodes. Thus, all fluxes are added to the sums $P_i^{\pm}$ and limited in a symmetric fashion using

$$\alpha_{ij} = \begin{cases} \min\{R_i^+, R_j^-\}, & \text{if } f_{ij} \ge 0, \\ \min\{R_i^-, R_j^+\}, & \text{if } f_{ij} < 0, \end{cases} \qquad \alpha_{ji} := \alpha_{ij}. \tag{73}$$

The bounds $Q_i^{\pm}$ for the antidiffusive mass fluxes f_{ij}^M must 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.

6. Solution of nonlinear systems

After the discretization in time, the nonlinear algebraic system associated with the flux-corrected Galerkin discretization (39) can be written as

$$Au^{n+1} = Bu^n + \bar{f}(u^{n+1}, u^n), \tag{74}$$

where $\bar{f}(u^{n+1}, u^n)$ denotes the sum of limited antidiffusive fluxes and

$$A = \frac{1}{\Delta t} M_L - \theta(K + D - L^-), \tag{75}$$

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

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.

To prove positivity preservation, one can express (74) as $\bar{A}u^{n+1} = \bar{B}u^n$, where $\bar{A}$ and $\bar{B}$ are nonlinear operators satisfying (20). The existence of such an equivalent representation follows from inequalities (43)–(45). The formal proof is straightforward and similar to the proofs presented in [27, 28].

Since the antidiffusive term depends on the unknown solution, the nonlinear discrete problem must be solved in an iterative way. In general, only the fully converged solution is guaranteed to conserve mass and preserve positivity. 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 are just as important as the accuracy of the flux limiting procedure.

6.1. Iterative defect correction

The structure of the nonlinear algebraic system (74) suggests the use of a fixed-point iteration with a lagged treatment of the antidiffusive term

$$Au^{(m)} = Bu^n + \bar{f}(u^{(m-1)}, u^n). \tag{77}$$

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

$$u^{(m)} = u^{(m-1)} + \omega\tilde{A}^{-1}r^{(m-1)}, \tag{78}$$

where ω is a relaxation parameter, $\tilde{A}$ is a 'preconditioner' (see below), and

$$r^{(m-1)} = Bu^n - Au^{(m-1)} + \bar{f}(u^{(m-1)}, u^n), \tag{79}$$

is the residual vector. In practice, we update the new solution as follows:

ALGORITHM 1: Defect correction scheme

---

1. Set $u^{(0)} := u^n$.
2. **For** all $m = 1, 2, \ldots$ **do**
   - Solve the linear system $\tilde{A}\Delta u^{(m)} = r^{(m-1)}$.
   - Update the solution $u^{(m)} := u^{(m-1)} + \omega\Delta u^{(m)}$.
   - Check the stopping criteria, exit if converged.
3. Set $u^{n+1} = u^{(m)}$, go to the next time step.

The iteration process is typically terminated when certain norms of $\Delta u^{(m)}$ and/or $r^{(m)}$ become smaller than a prescribed tolerance. More elaborate stopping criteria based on the FEM theory can be found in [1].

Clearly, the rates of convergence and the overall efficiency of the above defect correction scheme are strongly influenced by the choice of ω and $\tilde{A}$. The default is $\omega := 1$ and $\tilde{A} := A$, which corresponds to (77). By construction, A is monotone. This property results in fast convergence of inner iterations but the convergence of outer iterations may be rather slow.

Some advanced preconditioning and underrelaxation techniques are discussed in [27, 30, 40]. In quasi-Newton methods, $\tilde{A}$ is defined as a suitable approximation to the Jacobian of the nonlinear system. Due to the complex structure and nondiferentiability 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 6.3 leads to a Newton-like scheme in which the memory effect is exploited to avoid numerical differentiation.

6.2. Nonlinear SSOR method

A major drawback of fixed-point methods like (77) is the fully explicit treatment of the nonlinear antidiffusive term. An attempt to build implicit antidiffusion into the preconditioner $\tilde{A}$ for the defect correction scheme aggravates convergence problems if all correction factors are taken from the previous outer iteration. This has led us to update the solution values, the fluxes, and the correction factors simultaneously in a loop over nodes. The resulting algorithm can be classified as a nonlinear SSOR method.

The i-th equation of the limited Galerkin scheme (74) can be written as

$$\sum_j a_{ij} u_j = b_i + \bar{f}_i, \tag{80}$$

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

The calculation of $u_i^{(m)} \approx u_i^{n+1}$ begins with an update of the correction factors α_{ij} for the nonlinear antidiffusive term $\bar{f}_i$. In the forward sweep, the new values $u_j^{(m)}$ are already available for all $j < i$. Thus, we have

$$u_j = \begin{cases} u_j^{(m)}, & \text{if } j < i, \\ u_j^{(m-1)}, & \text{if } j \geq i. \end{cases} \tag{81}$$

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

Given the array of current solution values u_i, we recalculate the raw antidiffusive fluxes f_{ij}, apply the limiter, and add the result to $\bar{f}_i$. For an algebraic flux correction scheme of the form (64)–(67) the algorithm reads:

ALGORITHM 2: Flux limiting procedure

Set $P_i^{\pm} := 0,\ Q^{\pm} := 0.$

For all $j \in S_i$ **do**

- Calculate $f_{ij} = d_{ij}(u_i - u_j)$.
- Set $P_i^{\pm} := P_i^{\pm} + \mathop{\max}_{\min}\{0, f_{ij}\}$.
- Set $Q_i^{\pm} := \mathop{\max}_{\min}\{Q_i^{\pm}, q_i(u_j - u_i)\}$.

Compute $R_i^{\pm} = \min\{1, Q_i^{\pm}/P_i^{\pm}\}$.

For all $j \in S_i$ **do**

- Calculate $\alpha_{ij} = \alpha_{ij}(R_i^{\pm}, R_j^{\mp})$.
- Set $\bar{f}_i := \bar{f}_i + \alpha_{ij} f_{ij}$.

Since the value of α_{ij} depends not only on $R_i^{\pm}$ but also on $R_j^{\mp}$, we store the nodal correction factors in an auxiliary vector, 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 fail to satisfy $\bar{f}_{ji} = -\bar{f}_{ij}$ but the skew-symmetry property is restored when the algorithm converges.

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

$$u_i := u_i + \frac{1}{\tilde{a}_{ii}} \left(b_i - \sum_j a_{ij} u_j + \bar{f}_i \right). \tag{82}$$

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 value of $\tilde{a}_{ii} \geq a_{ii}$ 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 (74) is as follows:

ALGORITHM 3: Nonlinear SSOR iteration

For all $i = 1, \ldots, N_\Omega$ **do**

- Update the antidiffusive term $\bar{f}_i$ using Algorithm 2.
- Calculate the new solution value u_i using (82).

For all $i = N_\Omega, \ldots, 1$ **do**

- Update the antidiffusive term $\bar{f}_i$ using Algorithm 2.
- Calculate the new solution value u_i using (82).

The forward sweep can be written as $(\tilde{D} + \tilde{L})\Delta u^* := r$, where r is the residual, $\tilde{D} = \text{diag}\{\tilde{a}_{ii}\}$ is a diagonal matrix, 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 3 can written in the form (78) with $\omega = 1$ and

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

Luo et al. [37] used such a scheme as a preconditioner for a linear GMRES solver. A nonlinear version of this solution strategy is recovered when the method presented in Section 6.3 is employed to accelerate Algorithm 3.

In an 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. A usable initial guess can be obtained by solving the linear system with $\bar{f}_i = 0$ or $\bar{f}_i = f_i$.

6.3. 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 (78) and the nonlinear SSOR iteration (82) generate a sequence of successive approximations but only the last iterate $u^{(m-1)}$ is used when it comes to the computation of $u^{(m)}$. It turns out that including information from a number of previous iterates may dramatically improve the convergence behavior. This idea is exploited in many vector extrapolation techniques for vector sequences (see, e.g., [19, 46]). In this paper, we employ the convergence acceleration technique known as *Anderson mixing* [3, 9, 10, 49]. As shown in [9], 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 [49].

Following Walker and Ni [49], we formulate Anderson acceleration thus:

ALGORITHM 4: Anderson acceleration

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

For all $m = 1, 2, \ldots$ **do**

- Compute $\tilde{u}^{(m)} := g(u^{(m-1)})$ with (78) or (82).
- Store $\tilde{u}^{(m)}$ and $\Delta u^{(m)} := \tilde{u}^{(m)} - u^{(m-1)}$.
- 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

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

- Set $u^{(m)} := \sum_{i=1}^{k} \omega_i^{(m)} \tilde{u}^{(m-k+i)}$.
- Check the stopping criteria, exit if converged.

Set $u^{n+1} = u^{(m)}$, go to the next time step.

In practice, it is worthwhile to calculate the weights by solving an equivalent unconstrained least squares problem [49]. Furthermore, Anderson acceleration may need to be restarted if the vectors $\Delta u^{(m)}$ become (almost) linearly dependent, or if the norm of $\Delta u^{(m)}$ turns out to be much greater than that of $\Delta u^{(m-1)}$. For a detailed discussion of various improvements and practical implementation details, we refer to the literature [9, 10, 41, 49].

7. Numerical examples

In this section, we apply the proposed algorithms to a suite of 2D test problems which have already been studied using other algebraic flux correction schemes [28, 26, 30, 27]. Given a reference solution u, we use the following norms to assess the accuracy of a numerical approximation u_h

$$E_1(h) = \sum_i m_i |u(\mathbf{x}_i) - u_i| \approx \|u - u_h\|_1, \tag{83}$$

$$E_2(h) = \sqrt{\sum_i m_i |u(\mathbf{x}_i) - u_i|^2} \approx \|u - u_h\|_2, \tag{84}$$

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

The objective of the below numerical study is to investigate the dependence of E_1 and E_2 on the mesh size h and on the choice of the limiting strategy. In particular, we use the solutions computed on the two finest meshes to estimate the expected order of accuracy by the formula [32]

$$p = \log_2 \left(\frac{E_1(2h)}{E_1(h)} \right). \tag{85}$$

In the last example, we also study the convergence behavior of the iterative solver to give a flavor of efficiency gains offered by Anderson acceleration.

7.1. Solid body rotation

The solid body rotation test [32, 50] is often used to evaluate numerical advection schemes. The problem to be solved is the continuity equation

$$\frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{v} u) = 0 \qquad \text{in } \Omega = (0,1) \times (0,1). \tag{86}$$

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

$$\mathbf{v}(x, y) = (0.5 - y, x - 0.5). \tag{87}$$

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 [32], we simulate solid body rotation of a profile that consists of a slotted cylinder, a sharp cone, and a smooth hump (see Fig. 1a).

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). Let

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

be the normalized distance from (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 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 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 (86) is initialized by zero, and homogeneous Dirichlet boundary conditions are prescribed at the inlets.

The snapshots presented in Figs 1 and 2 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 shown in Figs 1b and 1c, respectively. As expected, the unconstrained Galerkin solution exhibits spurious oscillations, while its low-order counterpart is too diffusive. The solution shown in Fig. 1d was computed using the FEM-FCT algorithm developed in [26]. This algebraic flux correction scheme of predictor-corrector type produces excellent results when the time steps are small. However, its accuracy deteriorates at large time steps, and steady-state solutions depend on Δt. In the consistent-mass FEM-FCT scheme, a common correction factor α_{ij} is applied to f_{ij}^M and f_{ij}^K. Due to this coupling, *ad hoc* prelimiting is performed to avoid spurious ripples in situations when the signs of f_{ij}^M and f_{ij}^K differ.

In the caption to Fig. 2, the abbreviations LPSL and LPFL refer to the general-purpose limiting techniques described in Sections 5.1 and 5.3, respectively (LP := *Linearity Preserving*, SL:= *Slope Limiting*, FL := *Flux*

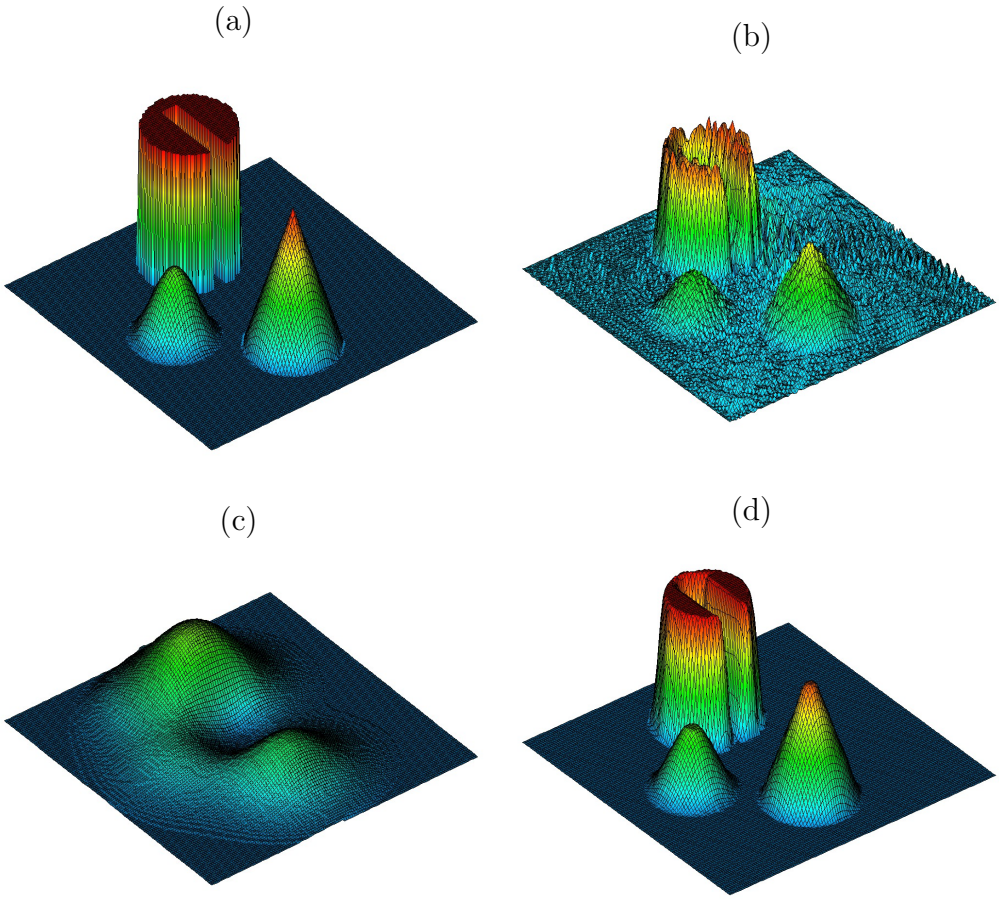

Figure 1: Solid body rotation: (a) initial data / exact solution, (b) standard Galerkin scheme, (c) discrete upwinding, (d) FCT flux correction. Discretization: $\mathcal{Q}_1$ elements ($h = 1/128$), Crank-Nicolson time-stepping ($\Delta t = 10^{-3}$). Simulation time: $T = 2\pi$.

Limiting). The results shown in Figs 2a and 2b 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 for which FCT is far superior to other shock-capturing methods [**?**]. In contrast to flux limiters of TVD type [24, 25], the new methodology is applicable to the antidiffusive part of the consistent mass matrix which makes it possible to attain fourth-order accuracy with linear finite elements (see [8], 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 2c and 2d.

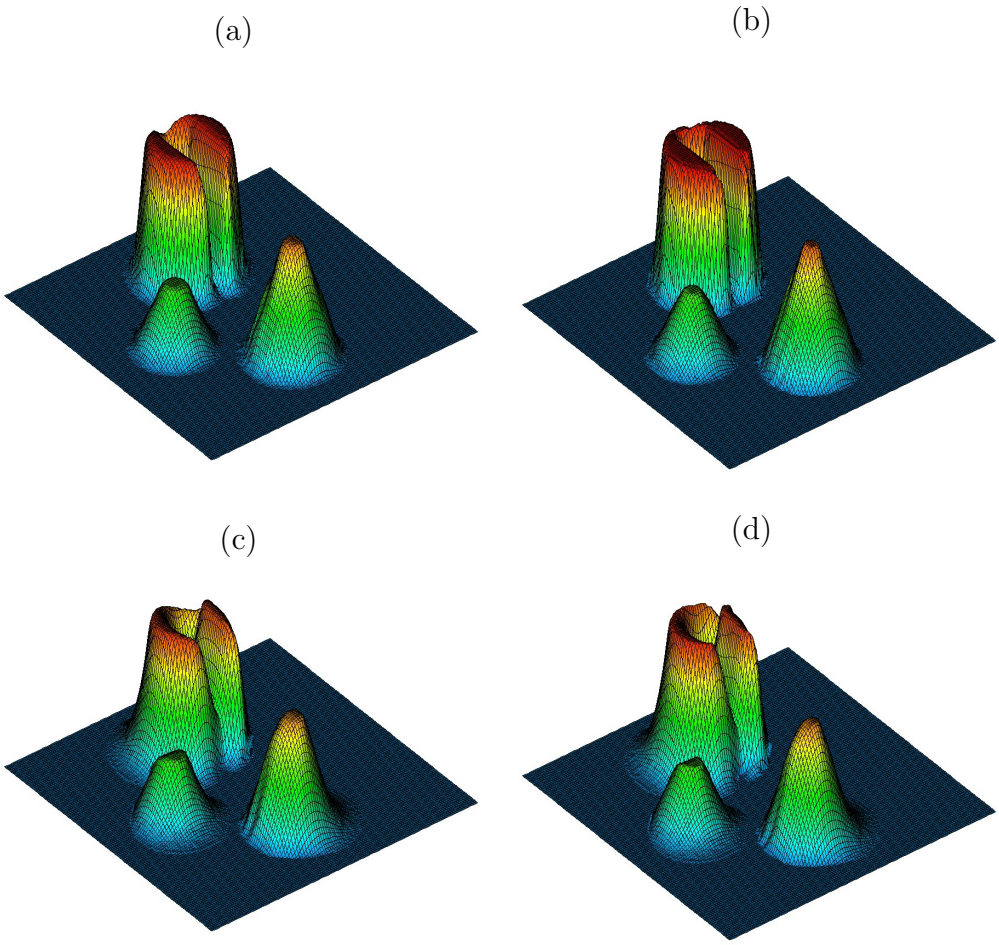

Figure 2: Solid body rotation: (a) LPSL, consistent mass (b) LPFL, consistent mass, (c) LPSL, lumped mass (b) LPFL, lumped mass, Discretization: $\mathcal{Q}_1$ elements ($h = 1/128$), Crank-Nicolson time-stepping ($\Delta t = 10^{-3}$). Simulation time: $T = 2\pi$.

The diagram in Fig. 3 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 1 and 2. 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 (85) with $h = 1/256$. The rates of convergence for the algorithms labeled (a) through (d) in Fig. 2

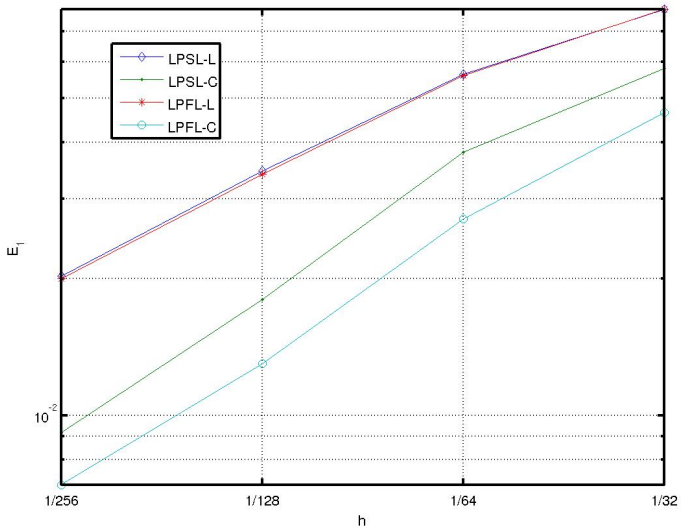

Figure 3: Solid body rotation, convergence history for LP limiters.

	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 1: 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 2: Solid body rotation: LPFL grid convergence.

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 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.

7.2. Circular convection

The second test problem is taken from [16]. Consider the hyperbolic PDE

$$\nabla \cdot (\mathbf{v}u) = 0 \qquad \text{in } \Omega = (-1,1) \times (0,1) \tag{88}$$

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 4 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 3 and 4, 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.

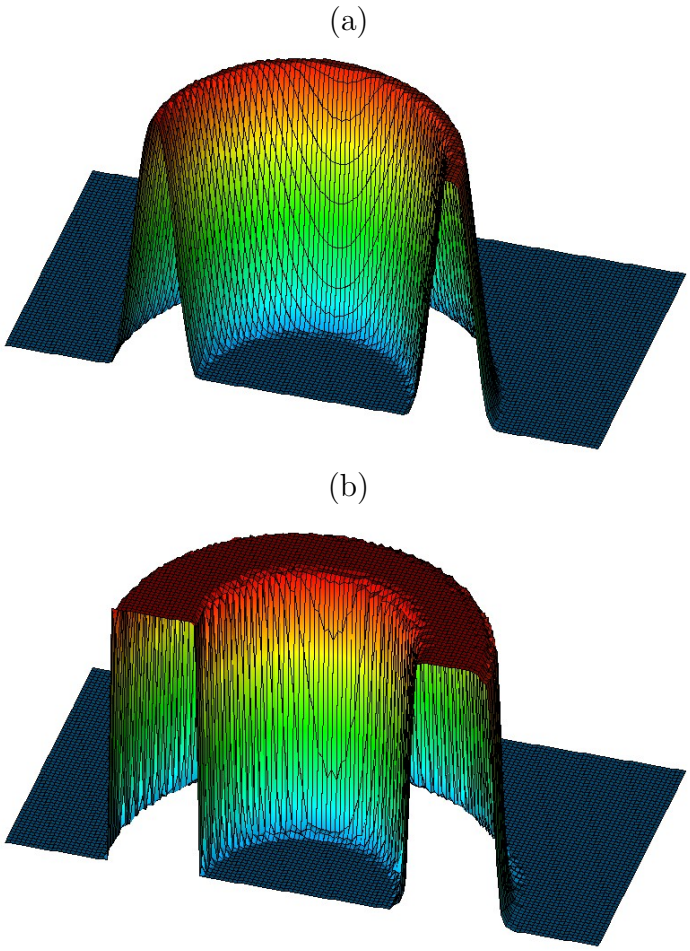

Figure 4: Circular convection: LPFL results for (a) smooth and (b) discontinuous data.

	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 3: 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 4: Circular convection: LPFL grid convergence.

7.3. Anisotropic diffusion

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

$$-\nabla \cdot (\mathcal{D}\nabla u) = 0 \quad \text{in } \Omega, \tag{89}$$

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. 5a). The following Dirichlet boundary conditions are prescribed

$$u(x,y) = \begin{cases} -1, & \text{if } (x,y) \in \Gamma_0, \\ 1, & \text{if } (x,y) \in \Gamma_1. \end{cases} \tag{90}$$

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

$$\mathcal{D} = \mathcal{R}(-\theta) \begin{pmatrix} k_1 & 0 \\ 0 & k_2 \end{pmatrix} \mathcal{R}(\theta), \tag{91}$$

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

$$\mathcal{R}(\theta) = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}. \tag{92}$$

The eigenvalues of $\mathcal{D}$ 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,

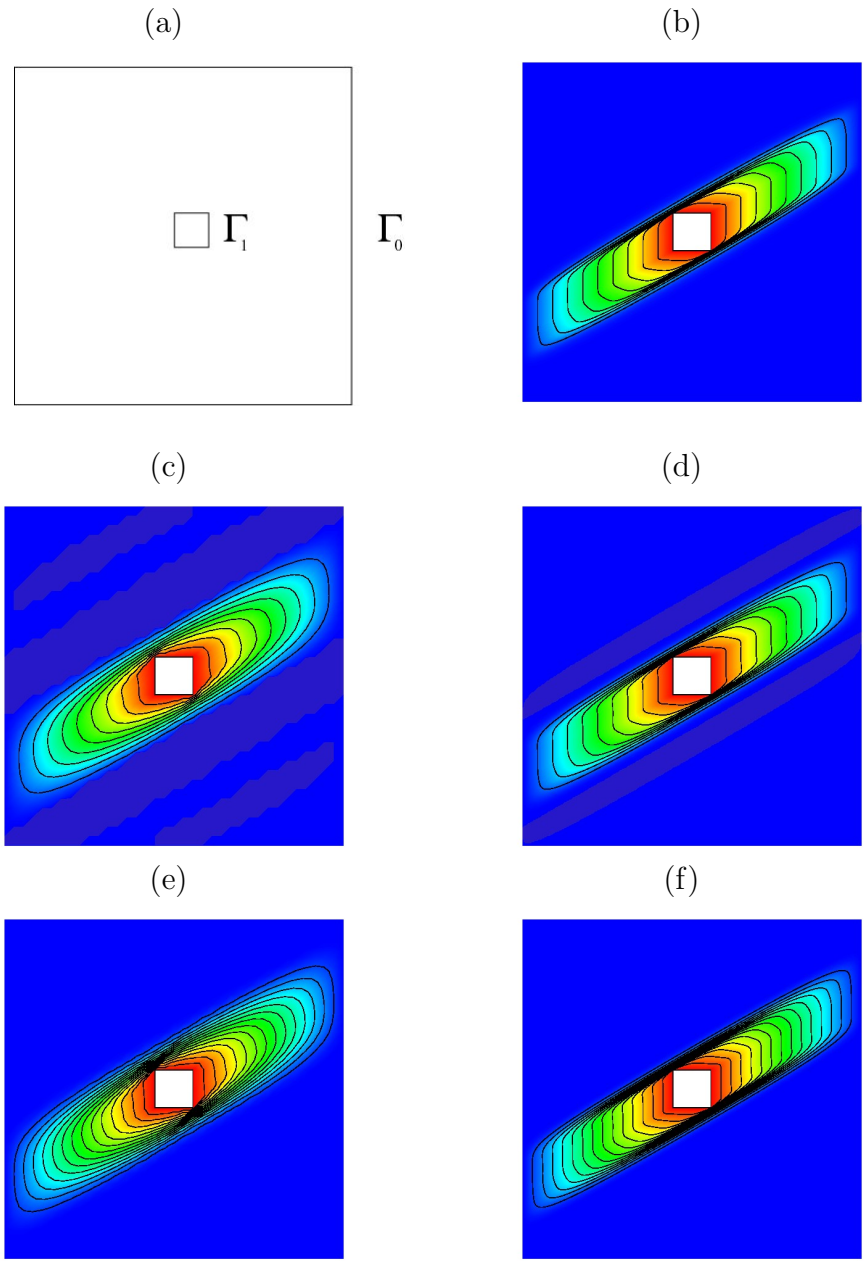

Figure 5: Anisotropic diffusion: (a) computational domain, (b) reference solution, (c) Galerkin, $h = 1/36$, (d) Galerkin, $h = 1/288$, (e) LPFL, $h = 1/36$, (f) LPFL, $h = 1/288$.

the diffusion tensor (91) 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. [33]. The results obtained with the LPSL scheme can be found in [30]. In this section, we discretize the anisotropic diffusion equation (89) using LPFL and linear finite elements on uniform triangular meshes. Since no exact solution is available, the reference solution depicted in Fig. 5b 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 5c and 5d. 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 5e and 5f.

The results of a grid convergence study are summarized in Tables 5 and 6. On coarse meshes, the LPFL algorithm produces smaller errors than the un-

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 5: 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 6: Anisotropic diffusion: LPFL grid convergence.

derlying 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 (85) 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 [30]. The last two columns in Table 6 list the total number of nonlinear SSOR iterations required to make the maximum norm of the residual smaller than $\epsilon = 10^{-6}$. 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 [1]. The numbers in the column labeled NNL-A were obtained with Anderson acceleration, as described in Section 6.3. If it is switched off, a dramatic increase in the number of nonlinear iterations NNL is observed (see the last column in Table 6). 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 [9, 10, 41, 49] are likely to result in a further gain of efficiency.

8. Conclusions

This paper has significantly advanced the state of the art in the design of efficient general-purpose flux limiters for implicit finite element discretizations. We extended the linearity-preserving slope limiter based on gradient reconstruction to unsteady convection-diffusion problems, converted it into a fully multidimensional algebraic flux correction scheme, designed a nonlinear SSOR method for solving the nonlinear algebraic system, and explored the potential of the Anderson acceleration technique in this context. The proposed methodology is closely related to the flux-corrected transport (FCT) algorithm but is readily applicable to stationary problems, as well as to unsteady transport processes which converge to a steady state equilibrium.

If the problem at hand is strongly time-dependent and the time steps are rather small, the proposed scheme can be linearized in much the same way as the FCT algorithm presented in [26]. This idea leads to a simple predictor-corrector algorithm in which the antidiffusive fluxes are evaluated using a provisional solution calculated without taking the antidiffusive term

into account. In this linearized version, all fluxes must be limited in a symmetric fashion since the convective flux into the downwind node is no longer balanced by the nonoscillatory part of the Galerkin transport operator.

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 powerful 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).

In summary, the efficiency of algebraic flux correction schemes can be enhanced by using the predictor-corrector strategy for small time steps and convergence acceleration techniques otherwise. In the case of hyperbolic systems, the new linearity-preserving flux limiter can be applied to the primitive or characteristic variables following the methodology developed in [14, 29].

Acknowledgments

The author would like to thank Dr. Mikhail Shashkov (Los Alamos National Laboratory), Dr. John N. Shadid (Sandia National Laboratories), and Prof. Peter Bastian (IWR Heidelberg) for discussions that motivated and inspired the work presented in this paper. This research was supported by the German Research Association (DFG) under grant KU 1530/3-2.

References

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

[2] M. Arioli, D. Loghin, 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] T. Barth, M. Ohlberger: Finite volume methods: foundation and analysis. In: E. Stein, R. de Borst, T.J.R. Hughes (eds), *Encyclopedia of*

Computational Mechanics, Volume 1: Fundamentals. John Wiley & Sons, 2004, 439–474.

[5] P. Bochev, D. Ridzal, G. Scovazzi, M. Shashkov: Formulation, analysis and numerical study of an optimization-based conservative interpolation (remap) of scalar fields for arbitrary Lagrangian-Eulerian methods. Submitted to *J. Comput. Phys.*

[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] J.-C. Carette, H. Deconinck, H. Paillère, P.L. Roe: Multidimensional upwinding: Its relation to finite elements. *Int. J. Numer. Methods Fluids* **20**:8-9 (1995) 935–955.

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

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

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

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

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

[13] 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.

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

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

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

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

[18] 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.

[19] A. Jemcov, 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.

[20] V. John, P. Knobloch: On spurious oscillations at layers diminishing (SOLD) methods for convection-diffusion equations: Part I - A review. *Comput. Methods Appl. Mech. Engrg.* **196** (2007) 17–20 2197–2215.

[21] V. John, P. Knobloch: On spurious oscillations at layers diminishing (SOLD) methods for convection-diffusion equations: Part II - Analysis for P_1 and Q_1 finite elements. *Comput. Methods Appl. Mech. Engrg.* **197** (2008) 1997–2014.

[22] V. John, 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.

[23] 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.

[24] 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.

[25] 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.

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

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

[28] D. Kuzmin, M. Möller: Algebraic flux correction I. Scalar conservation laws. In: D. Kuzmin, R. Löhner, S. Turek (eds) *Flux-Corrected Transport: Principles, Algorithms, and Applications*. Springer, 2005, 155–206.

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

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

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

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

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

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

[35] R. Löhner, K. Morgan, J. Peraire, M. Vahdati: Finite element flux-corrected transport (FEM-FCT) for the Euler and Navier-Stokes equations. *Int. J. Numer. Meth. Fluids* **7** (1987) 1093–1109.

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

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

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

[39] 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.

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

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

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

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

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

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

[46] D.A. Smith, W.F. Ford, A. Sidi: Extrapolation methods for vector sequences. *SIAM Review* **29** (1987) 199–233.

[47] P.K. Sweby, High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM J. Numer. Anal.* **21** (1984) 995–1011.

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

[49] H.W. Walker, P. Ni: Anderson acceleration for fixed-point iterations. *WPI Math. Sci. Dept. Report* MS-9-21-45, September 2009. Submitted to *SIAM J. Numer. Anal.*

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