Algebraic flux correction for nonconforming finite element discretizations of scalar transport problems

Matthias Möller

Abstract This paper is concerned with the extension of the *algebraic flux-correction* (AFC) approach [13–16, 18–24] to nonconforming finite element methods for the linear transport equation. Accurate nonoscillatory approximations to convection-dominated flows are obtained by stabilizing the continuous Galerkin method by solution-dependent artificial diffusion. Its magnitude is controlled by a flux limiter. This concept dates back to flux-corrected transport (FCT) schemes. The unique feature of *algebraic* flux correction is that all information is extracted from the system matrices which are manipulated to satisfy certain mathematical constraints. AFC schemes have been devised with conforming P_1 and Q_1 finite elements in mind but this is not a prerequisite. Here, we consider their extension to the nonconforming Crouzeix-Raviart element [3] on triangular meshes and its quadrilateral counterpart, the class of rotated bilinear Rannacher-Turek elements [29]. The underlying design principles of AFC schemes are shown to hold for (some variant of) both elements. However, numerical tests for a purely convective flow and a convection-diffusion problem demonstrate that flux-corrected solutions are overdiffusive for the Crouzeix-Raviart element. Good resolution of smooth and discontinuous profiles is attested to Q_1^{nc} approximations on quadrilateral meshes. A synthetic benchmark is used to quantify the artificial diffusion present in conforming and nonconforming high-resolution schemes of AFC-type. Finally, the implementation of efficient sparse matrix-vector multiplications is addressed.

Keywords non-/conforming finite elements, high-resolution schemes, flux-corrected transport, hyperbolic problems, convection-dominated flows

Mathematics Subject Classification (2000) 65M60 · 65Y20

M. Möller
Institute of Applied Mathematics (LS III), TU Dortmund University of Technology, Vogelpothsweg 87, 44227 Dortmung, Germany
Tel.: +49-231-755-3071
Fax: +49-231-755-5933
E-mail: matthias.moeller@math.tu-dortmund.de

1 Introduction

The computation of nonoscillatory accurate solutions to convection-dominated problems is an important issue in numerical flow simulation. An algebraic approach to construct high-resolution schemes was introduced by Kuzmin [13, 22] and refined in a series of publications [14–16, 18–21, 23, 24] (and references therein). It is based on a rigorous analysis of matrix properties that must be fulfilled by numerical methods in order to preserve positivity of physical quantities. In general, the continuous Galerkin finite element method does not satisfy the necessary mathematical constraints but it can be turned into a positivity-preserving scheme by adjusting matrix coefficients *a posteriori*.

This technique is termed algebraic flux correction (AFC) and it has been employed successfully to stabilize linear (P_1) and bi/trilinear (Q_1) finite element approximations on unstructured meshes. In contrast to many stabilization techniques for convection-dominated problems, there is no free parameter that needs to be tuned by the user. The extension of AFC schemes to higher-order finite elements was found to be possible but very challenging [15].

A preliminary numerical evaluation of the algebraic flux correction approach for nonconforming low-order finite element approximations on quadrilaterals was started in [28]. In particular, it was found that the integral mean value based variant of the rotated bi/trilinear Q_1^{nc} finite element introduced by Rannacher and Turek in [29] produces accurate nonoscillatory solutions with crisp resolution of steep fronts for all tested AFC schemes. In contrast, its midpoint-based counterpart gives rise to negative coefficients in the consistent mass matrix which prohibits the successful application of the AFC machinery.

For Q_1^{nc} elements, the degrees of freedom (DOF) are associated with element faces rather than vertices so that each degree of freedom belongs to two elements at most. This property is considered an advantage of nonconforming elements when it comes to reducing communication costs in parallel computation [11, 26, 30]. Since each DOF is connected to a constant number of neighboring DOFs (in 2D: four at interior edges and two at boundary edges) nonconforming approximations give rise to matrices with a small constant number of nonzero entries per matrix row even on unstructured meshes [12]. Another motivation for our interest in high-resolution nonconforming finite element schemes is related to the planned coupling of compressible and incompressible flow solvers [5], whereby the latter relies on the inf-sup stable Q_1^{nc}/Q_0 finite element pair and its extension to higher-order approximations.

In the present paper, we apply the AFC approach to nonconforming P_1^{nc} approximations [3]. This finite element type has been used successfully for the simulation of radiative transport modeled by nonlinear diffusion equations and it is reportedly capable to 'resolve very sharp changes of energies on the heterogeneous domains, similar to finite volume methods with an edge-based flux limiter' (see [12], p. 381). To start with the main result of our investigation, P_1^{nc} approximations satisfy all theoretical prerequisites of algebraic flux-correction but the numerical solutions leave a lot to be desired.

2 Galerkin approximation

Our discussion will be based on the linear convection-diffusion equation

$$\frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{v}u - \epsilon \nabla u) = 0 \qquad \text{in } \Omega \times (t_0, T], \tag{1}$$

where $\Omega \subset \mathbb{R}^{\mathrm{n_{sd}}}$ is a bounded domain and $\mathrm{n_{sd}} = 2, 3$ denotes the number of spatial dimensions. It models the transport of the conserved quantity u by the externally given velocity field $\mathbf{v} = \mathbf{v}(\mathbf{x}, t)$ with ($\epsilon > 0$) or without ($\epsilon = 0$) diffusion. In the latter case, the governing equation is of hyperbolic type so that boundary conditions are imposed on the inflow part of the boundary

$$u = u_D \qquad \text{on } \Gamma_- := \{\mathbf{x} \in \Gamma \,:\, \mathbf{v} \cdot \mathbf{n} < 0\} \times (t_0, T], \tag{2}$$

whereas no boundary conditions are prescribed elsewhere. Here, $\mathbf{n}$ denotes the outward unit normal vector to the boundary. In the presence of diffusion, Dirichlet boundary conditions are prescribed along the whole boundary Γ. Our model problem is completed by prescribing the initial profile at time t_0:

$$u(\mathbf{x}, t_0) = u_0(\mathbf{x}) \qquad \text{in } \Omega. \tag{3}$$

The variational form of problem (1) reads: Find a solution $u \in \mathcal{S}_t$ such that

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

for all admissible test functions $w \in \mathcal{H}^1(\Omega)$ vanishing on Γ_- and Γ, respectively. The time dependency of u carries over to the trial space

$$\mathcal{S}_t := \{u \,:\, u(\cdot, t) \in \mathcal{H}^1(\Omega), t \in [t_0, T] \text{ and } u = u_D \text{ on } \Gamma_{(-)}\}. \tag{5}$$

A common practice in computing approximate solutions $u_h \approx u$ to conservation laws is to adopt Fletcher's group finite element formulation [6]. Let

$$u_j(t) = u(\mathbf{x}_j, t) \qquad \text{and} \qquad \mathbf{v}_j(t) = \mathbf{v}(\mathbf{x}_j, t) \tag{6}$$

denote the time-dependent coefficients of the unknown solution and the values of the given velocity field at the degrees of freedom $\mathbf{x}_j$ (to be defined below), respectively. The same finite set of basis functions $\{\varphi_j\}_{j=1}^{N_{\mathrm{dof}}}$ is used to interpolate the unknown solution, its gradient and the convective fluxes [23]

$$u_h(\mathbf{x}, t) = \sum\nolimits_j \varphi_j(\mathbf{x}) u_j(t), \tag{7}$$

$$\nabla u_h(\mathbf{x}, t) = \sum\nolimits_j \nabla\varphi_j(\mathbf{x}) u_j(t), \tag{8}$$

$$(\mathbf{v}u)_h(\mathbf{x}, t) = \sum\nolimits_j \varphi_j(\mathbf{x}) (\mathbf{v}_j u_j)(t). \tag{9}$$

The semi-discrete problem that is obtained by substituting the above expressions into the weak form (4) and replacing the test function w by φ_i reads [13]

$$M_C \frac{\mathrm{d}u}{\mathrm{d}t} = (K + S)u, \tag{10}$$

where $M_C = \{m_{ij}\}$ denotes the consistent mass matrix. The stiffness matrix $S = \{s_{ij}\}$ vanishes in the absence of diffusion. Their coefficients are given by

$$m_{ij} = \int_\Omega \varphi_i \varphi_j \, \mathrm{d}\mathbf{x}, \qquad s_{ij} = -\epsilon \int_\Omega \nabla\varphi_i \cdot \nabla\varphi_j \, \mathrm{d}\mathbf{x}. \tag{11}$$

The entries of the discrete convective transport operator $K = \{k_{ij}\}$ read

$$k_{ij} = -\mathbf{v}_j \cdot \mathbf{c}_{ij}, \qquad \mathbf{c}_{ij} = \int_\Omega \varphi_i \nabla\varphi_j \, \mathrm{d}\mathbf{x}. \tag{12}$$

For fixed meshes, coefficients m_{ij}, $\mathbf{c}_{ij}$ and s_{ij} do not change and need to be evaluated just once. Hence, matrix K can be assembled and, in case of an unsteady velocity field, updated in each time step in an efficient way from formula (12) without resorting to costly numerical quadrature. This makes the group finite element formulation especially attractive for 'unconventional' finite elements which require summed or higher-order quadrature rules.

The semi-discrete scheme (10) for the single unknown u_i is cast into [24]

$$\sum_j \left[m_{ij} \frac{\mathrm{d}u_j}{\mathrm{d}t} \right] = \sum_{j \neq i} k_{ij}(u_j - u_i) + g_i, \tag{13}$$

exploiting the fact that the 'incompressible' part of the discrete transport operator K can be assembled by summing over all edges ij connecting the i-th degree of freedom with its neighbor j. The remaining contribution vanishes for divergence-free velocity fields and may cause the physical growth of solution values otherwise. Let us introduce the following short hand notation

$$g_i = u_i \sum_j k_{ij} + \sum_j s_{ij} u_j \tag{14}$$

for the 'compressible' part of K and for the diffusive contributions which do not require a special treatment within the flux correction procedure.

3 Low-order approximation

Numerical solutions to equation (1) computed by the Galerkin scheme (13) may suffer from nonphysical undershoots and overshoots if the diffusion coefficient ϵ is small as compared to the velocity or vanishes completely unless some suitable stabilization of the dominating convective term is employed. An algebraic approach to transform an unstable high-order discretization into a positivity-preserving high-resolution scheme was devised by Kuzmin [13, 22].

A key ingredient of algebraic flux correction is the concept of *local extremum diminishing* (LED) schemes suggested by Jameson [10]. It states that local minima/maxima are not enhanced by a semi-discrete scheme of the form

$$m_i \frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_{j \neq i} \sigma_{ij}(u_j - u_i) \qquad \text{for all } i \tag{15}$$

provided that $m_i > 0$, $\forall i$ and all off-diagonal coefficients σ_{ij} are non-negative.

This criterion is typically violated by the high-order discretization (13). In the context of linear and multilinear finite element approximations, it can be enforced *a posteriori* by applying two simply post-processing steps [13, 22]:

(1) Replace the consistent mass matrix M_C by its row-sum lumped counterpart

$$M_L = \{m_i\}, \qquad m_i = \sum_j m_{ij}. \tag{16}$$

Since the basis functions of (multi-)linear finite elements are nonnegative everywhere, there are no negative coefficients in the consistent mass matrix (i.e. $m_{ij} \geq 0$, $\forall i, j$), and therefore, $m_i > 0$ for all i (cf. [17], p. 50).

(2) Eliminate all negative entries from the high-order transport operator K by adding a discrete diffusion operator $D = \{d_{ij}\}$. It is defined as a symmetric matrix with zero row and column sums. Its entries are given by

$$d_{ij} = \max\{-k_{ij}, 0, -k_{ji}\} = d_{ji}, \qquad d_{ii} := -\sum_{j \neq i} d_{ij}. \tag{17}$$

The above manipulations lead to the low-order scheme [22]

$$M_L \frac{\mathrm{d}u}{\mathrm{d}t} = (L + S)u. \tag{18}$$

The off-diagonal entries of the low-order transport operator $K+D =: L = \{l_{ij}\}$ are non-negative by definition as it is required by the LED constraint (15).

However, it should be noted that various finite element approximations give rise to negative matrix coefficients $m_{ij} < 0$, $j \neq i$ so that some entries of the lumped mass matrix M_L may be zero or even negative. In this case, the two-step approach (16)–(17) is *not* applicable to enforce the LED condition. It is therefore mandatory to verify the condition $m_i > 0$, $\forall i$ for nonconforming finite element approximations prior to applying algebraic flux correction.

4 Finite element spaces

Let us postpone the description of the algebraic flux correction approach to Section 6 and discuss the different types of finite elements which were used in our numerical study. To simplify the presentation, consider a two-dimensional computational domain Ω_h which is decomposed into nonoverlapping triangles or convex quadrilaterals and let $\mathcal{T}_h$ denote the ensemble of all cells T.

A common practice to construct parametric finite elements is to define the shape functions for a geometrically simple fixed reference element $\hat{T}$ and transfer coordinates given in the reference frame into the physical space and vice versa using the reference map $\Psi_T : \hat{T} \mapsto T$. The map is unique up to a rotation in the local numbering of vertices in $\hat{T}$ and T, respectively. Let

$$\mathcal{Q}(T) = \{q : T \to \mathbb{R} \,|\, q = \hat{q} \circ \Psi_T^{-1}, \hat{q} \in \hat{\mathcal{Q}}(\hat{T})\} \tag{19}$$

denote the generic finite element space restricted to the physical cell T. The reference triangles and quadrilaterals used in this paper are depicted in Fig. 1.

Conforming P_1 and Q_1 elements. The construction of low-order nodal finite elements for quadrilaterals is typically based on the reference square $\hat{T} = [-1,1]^2$ endowed with the polynomial space $\hat{\mathcal{Q}}(\hat{T}) := \hat{Q}_1(\hat{T}) = \text{span}\langle 1, \hat{x}, \hat{y}, \hat{x}\hat{y}\rangle$. This definition leads to degrees of freedom (DOF) located at the four corners of the reference cell $\hat{T}$. The associated nodal shape functions are given by

$$\hat{\varphi}^{(k)}(\hat{x},\hat{y}) = \frac{1}{4}(1 \pm \hat{x})(1 \pm \hat{y}), \tag{20}$$

where $k = 1, \ldots, 4$ represent the local numbering of DOFs within each element.

Linear P_1 elements on triangles can be constructed in an analog fashion based on the reference triangle $\hat{T} = \{(\hat{x},\hat{y}) \in \mathbb{R}^2_+ : \hat{x} + \hat{y} \leq 1\}$. Without loss of generality the degrees of freedom which are associated with the three vertices are numbered counterclockwise starting at the lower left corner. Hence,

$$\hat{\varphi}^{(1)}(\hat{x},\hat{y}) = 1 - \hat{x} - \hat{y}, \quad \hat{\varphi}^{(2)}(\hat{x},\hat{y}) = \hat{x}, \quad \hat{\varphi}^{(3)}(\hat{x},\hat{y}) = \hat{y}. \tag{21}$$

Nonconforming Q_1^{nc} elements. The global continuity of Q_1 approximations across entire element boundaries can be relaxed by using nonconforming Q_1^{nc} finite elements which require continuity of the solution only in the degrees of freedom. In contrast to discontinuous Galerkin (DG) methods which give up any continuity requirement from the outset, nonconforming *continuous* Galerkin methods do not duplicate the degrees of freedom on cell boundaries so that the total number of unknowns is smaller than for DG methods.

Nonconforming rotated bilinear Q_1^{nc} finite elements can be derived from expression (19) by setting $\hat{\mathcal{Q}}(\hat{T}) := \hat{Q}_1^{\text{nc}}(\hat{T}) = \text{span}\langle 1, \hat{x}, \hat{y}, \hat{x}^2 - \hat{y}^2\rangle$ and associating the degrees of freedom with the four edge midpoints $\hat{\mathbf{m}}_1 = (1,0)$, $\hat{\mathbf{m}}_2 = (0,1)$, $\hat{\mathbf{m}}_3 = (-1,0)$, and $\hat{\mathbf{m}}_4 = (0,-1)$ of the reference cell $\hat{T} = [-1,1]^2$. A unisolvent finite element of this type was proposed by Rannacher and Turek in [29]. It is obtained by constructing the local shape functions $\hat{\varphi}^{(k)}$ such that their integral mean values along the edge $\hat{\gamma}_l$ of the reference cell satisfy

$$|\hat{\gamma}_l|^{-1} \int_{\hat{\gamma}_l} \hat{\varphi}^{(k)}(\hat{x},\hat{y})\, \mathrm{d}\gamma = \delta_{kl} \qquad \text{for} \quad k,l = 1,2,3,4. \tag{22}$$

An explicit expression of the so defined local shape functions is as follows [8]

$$\hat{\varphi}^{(k)}(\hat{x},\hat{y}) = \begin{cases} \frac{1}{4} \pm \frac{1}{2}\hat{x} + \frac{3}{8}(\hat{x}^2 - \hat{y}^2), & k = 1,3, \\ \frac{1}{4} \pm \frac{1}{2}\hat{y} - \frac{3}{8}(\hat{x}^2 - \hat{y}^2), & k = 2,4. \end{cases} \tag{23}$$

An alternative version of rotated bilinear elements [29] is obtained by replacing factor "3/8" by "1/4" in the above expression. The corresponding shape functions satisfy $\hat{\varphi}^{(k)}(\hat{\mathbf{m}}_l) = \delta_{kl}$ for $k,l = 1,2,3,4$ in lieu of property (22).

This midpoint based variant was shown [28] to be of little practical use within the framework of algebraic flux correction, whereas the integral based counterpart (23) leads to accurate numerical approximations. The poor performance of the "1/4"-version can be explained by the presence of negative

coefficients in the consistent mass matrix which contradicts the design principles of AFC schemes to be addressed in Section 6. As an illustration, consider the local mass matrices evaluated on the reference element

$$\hat{M}_{\text{"}3/8\text{"}} = \frac{1}{60}\begin{pmatrix} 41 & 9 & 1 & 9 \\ 9 & 41 & 9 & 1 \\ 1 & 9 & 41 & 9 \\ 9 & 1 & 9 & 41 \end{pmatrix}, \quad \hat{M}_{\text{"}1/4\text{"}} = \frac{1}{180}\begin{pmatrix} 113 & 37 & -7 & 37 \\ 37 & 113 & 37 & -7 \\ -7 & 37 & 113 & 37 \\ 37 & -7 & 37 & 113 \end{pmatrix}. \quad (24)$$

It is clear that mass lumping (16) can be performed without hazard only for the "3/8"-version and is likely to produce incorrect results for the "1/4"-variant.

Nonconforming P_1^{nc} elements. The rotated linear analog on triangles is the well-known Crouzeix-Raviart element [3]. Making use of the reference triangle $\hat{T} = \{(\hat{x}, \hat{y}) \in \mathbb{R}^2_+ : \hat{x} + \hat{y} \leq 1\}$ the local shape functions are defined as

$$\hat{\varphi}^{(1)} = 1 - 2\hat{y}, \quad \hat{\varphi}^{(2)} = 2(\hat{x} + \hat{y}) - 1, \quad \hat{\varphi}^{(3)} = 1 - 2\hat{x}. \quad (25)$$

They are associated with the three edge midpoints, whereby the local numbering starts at the bottom edge and continues in counterclockwise sense. It is easy to verify that the so defined shape functions satisfy both $\hat{\varphi}^{(k)}(\hat{\mathbf{m}}_l) = \delta_{kl}$ as well as the integral property (22) for $k, l = 1, 2, 3$. Moreover, the consistent mass matrix is genuinely diagonal [7] with strictly positive entries. This property makes P_1^{nc} elements particularly attractive for computing nonoscillatory low-order approximations from the semi-discrete scheme (18). In fact, the mass lumping step (16) which deteriorates the phase accuracy of the low-order solution can be omitted since both mass matrices M_C and M_L coincide.

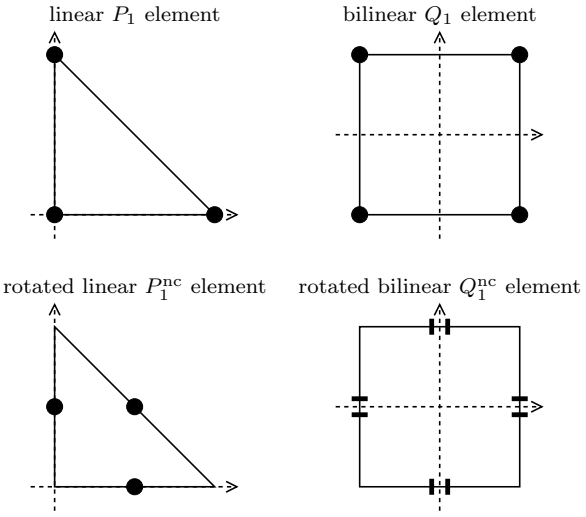


Fig. 1 Conforming and nonconforming low-order finite elements on triangles and quadrilaterals; prescription of function value (•) and integral mean value (‖) at DOFs.

5 Taxonomy of finite elements

A common property of the finite element basis $\{\varphi_i\}_{i=1}^{N_{\text{dof}}}$ resulting from one of the local shape functions (20), (21) and (25) is $\varphi_i(\mathbf{x}_j) = \delta_{ij}$. This means that the i-th basis function vanishes at all degrees of freedom except for vertex $\mathbf{x}_i$ and edge midpoint $\mathbf{m}_i$ for Q_1, P_1 and P_1^{nc} elements, respectively. In contrast, basis functions for Q_1^{nc} elements satisfy an analog of integral condition (22).

Figure 2 illustrates the local support of basis function $\varphi_i(x, y)$ extending over all cells that share the common degree of freedom i. This property implies that the number of edges ij connecting i and j is proportional to the number of surrounding elements. The presence of an edge ij corresponds to a (possibly) non-vanishing off-diagonal coefficient in the sparsity graph of finite element matrices. That is $m_{ij} \neq 0 \Rightarrow \exists ij$ with equivalence for $P_1, Q_1, Q_1^{\text{nc}}$ elements.

For an unstructured mesh, the number of non-zero entries may vary from one row to the other if bi/linear finite elements are adopted. Thus, general-purpose sparse matrix formats such as CSR (compressed sparse row) or COO (coordinate) which admit non-uniform row length are typically used to store system matrices. Due to the lack of a regular sparsity structure, it is difficult to achieve optimal performance in sparse matrix-vector multiplications (SpMV) which is one of the most often used kernels in scientific computing [2, 32–34].

In contrast, nonconforming P_1^{nc} and Q_1^{nc} elements lead to system matrices with a fixed number of non-zero coefficients per row (see Figure 2). In fact, it makes no difference whether the mesh is fully unstructured or exhibits a regular structure. It is therefore advisable to store sparse system matrices in, say, the ELL format [9] which is well suited to vector architectures (SSE/AVX in commodity CPUs, GPUs). Memory is reserved for a constant number of coefficients within each row and shorter rows are padded with zeros. Thus, efficiency is best if the number of non-zero coefficients per row does not differ significantly from the average [2] which is clearly the case for P_1^{nc} and Q_1^{nc} .

Admittedly, one has to solve for a larger number of unknowns N_{dof} when using nonconforming finite elements instead of conforming ones. Euler's polyhedron formula leads to the rough estimate $\mathcal{O}(N_{\text{dof}}^{\text{nc}}) = \mathcal{O}(N_{\text{dof}} + |\mathcal{T}_h|)$ which relates the increase in the problem size to the number of cells [28].

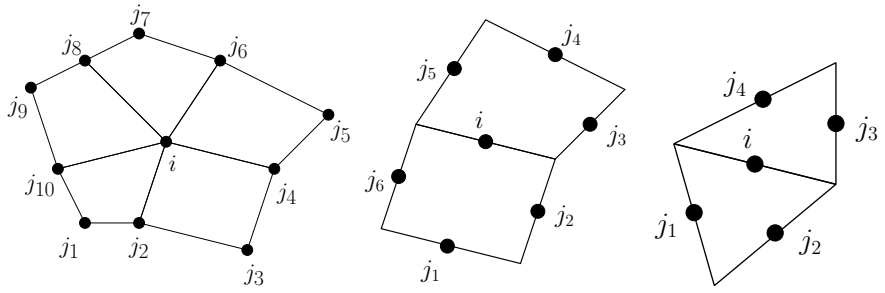


Fig. 2 Connectivity between degrees of freedom and local support of basis function φ_i: conforming Q_1 (left), nonconforming Q_1^{nc} (middle), and nonconforming P_1^{nc} (right) elements.

6 Algebraic flux correction

It goes beyond the intention of this paper to review the AFC framework and its mathematical background in detail. We briefly outline the two flux-corrected transport (FCT) algorithms in the way they were implemented to compute the numerical solutions presented in Section 7 and refer the interested reader to [17, 18] for a comprehensive description of algebraic flux correction.

The basic idea is to blend adaptively between the Galerkin approximation (10) which is accurate for smooth solutions but produces wiggles near steep fronts and the overdiffusive low-order scheme (18) which is optimal to resolve discontinuities. The difference between both approximations reads [21]

$$f = (M_L - M_C)\frac{\mathrm{d}u}{\mathrm{d}t} - Du. \tag{26}$$

It is safe to add a limited amount of this antidiffusion to the low-order scheme

$$M_L \frac{\mathrm{d}u}{\mathrm{d}t} = (L + S)u + \bar{f} \tag{27}$$

in regions where the solution is smooth. This will remove excess artificial diffusion and reduce the error introduced by the lumped mass matrix.

The common base of the limiting strategies [13–16, 18–24] is the decomposition of the antidiffusive term (26) into sums of internodal fluxes

$$f_i = \sum_{j \neq i} f_{ij}, \qquad f_{ij} = \left[m_{ij} \frac{\mathrm{d}}{\mathrm{d}t} + d_{ij} \right] (u_i - u_j), \qquad f_{ji} = -f_{ij}. \tag{28}$$

They are skew-symmetric, and can thus be scaled by symmetric correction factors $0 \leq \alpha_{ij} \leq 1$ without changing the overall mass balance. The limited counterpart of the raw antidiffusive fluxes (28) is then applied to the low-order scheme [13, 22]

$$m_i \frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_{j \neq i} l_{ij}(u_j - u_i) + g_i + \bar{f}_i, \qquad \bar{f}_i = \sum_{j \neq i} \alpha_{ij} f_{ij}. \tag{29}$$

Nonlinear FCT algorithm [13, 22]. Let the semi-discrete equation (27) be discretized in time by the Crank-Nicolson time-stepping scheme

$$Au^{n+1} = b^n + \bar{f}(u^{n+1}, u^n), \tag{30}$$

where the system matrix A and the right-hand side vector b^n are given by

$$A = \frac{1}{\Delta t} M_L - \frac{1}{2}(L + S), \qquad b^n = \frac{1}{\Delta t} M_L u^n + \frac{1}{2}(L + S)u^n. \tag{31}$$

The time derivative in the raw antidiffusive flux (28) is replaced accordingly.

At the beginning of each time step $t^n \to t^{n+1} := t^n + \Delta t$, an explicit low-order approximation of the intermediate state $u^{n+1/2}$ is computed by 'solving'

$$\frac{1}{\Delta t} M_L \tilde{u} = b^n. \tag{32}$$

Its values at the degrees of freedom serve as local upper and lower bounds to be imposed on the end-of-step solution to the nonlinear system (30); cf. [18]

$$u_i^{\min} := \min\{\tilde{u}_i, \min_{j\neq i} \tilde{u}_j\} \leq u_i^{n+1} \leq \max\{\tilde{u}_i, \max_{j\neq i} \tilde{u}_j\} =: u_i^{\max}. \tag{33}$$

Within an iterative defect correction loop which was terminated after a maximum of 100 steps or if the Euclidean norm of the residual was smaller than 1.0E-12 the flux-corrected solution to (30) was computed as follows:

1. Evaluate the raw antidiffusive fluxes based on the solution u^n from the previous time step and the last iterate $u^{(m)}$ starting with $u^{(0)} = u^n$

$$\begin{aligned} f_{ij}^{(m)} = &\frac{m_{ij}}{\Delta t}(u_i^{(m)} - u_j^{(m)}) + \frac{d_{ij}}{2}(u_i^{(m)} - u_j^{(m)}) \\ &- \frac{m_{ij}}{\Delta t}(u_i^{n} - u_j^{n}) + \frac{d_{ij}}{2}(u_i^{n} - u_j^{n}), \quad j \neq i. \end{aligned} \tag{34}$$

2. 'Prelimit' fluxes which would flatten the solution profile due to wrong sign

$$f_{ij}^{(m)} := 0 \quad \text{if } f_{ij}^{(m)}(\tilde{u}_j - \tilde{u}_i) > 0. \tag{35}$$

3. Compute correction factors α_{ij} (see below) and limit the antidiffusive fluxes

$$\bar{f}_i^{(m)} = \sum_{j\neq i} \alpha_{ij} f_{ij}^{(m)}. \tag{36}$$

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

$$Au^{(m+1)} = b^n + \bar{f}^{(m)}. \tag{37}$$

The repeated application of the flux limiter makes this kind of nonlinear flux correction compute-intensive. Moreover, the nonlinear iteration process may converge very slowly or fail completely. To overcome this drawback, Kuzmin [19] suggests the use of convergence acceleration techniques such as nonlinear SSOR schemes or Anderson mixing [1]. So far, the numerical results in this paper were computed without accelerating nonlinear convergence.

Linearized FCT algorithm [16]. An inexpensive alternative to nonlinear flux correction is to linearize the raw antidiffusive fluxes (28) prior to limiting them.

The following predictor-corrector algorithm is applied in each time step:

1. Compute the provisional solution u^L by the low-order scheme (18) discretized in time by the semi-implicit Crank-Nicolson scheme

$$Au^L = b^n. \tag{38}$$

2. Compute an approximation to the time derivative $\dot{u} \approx \frac{\mathrm{d}u}{\mathrm{d}t}$ from (18)

$$\dot{u}^L = M_L^{-1}(L+S)u^L. \tag{39}$$

3. Linearize the raw antidiffusive fluxes based on the low-order predictor

$$f_{ij}^L = m_{ij}(\dot{u}_i^L - \dot{u}_j^L) + d_{ij}(u_i^L - u_j^L) \tag{40}$$

4. Perform 'prelimiting' of antidiffusive fluxes with wrong sign

$$f_{ij}^L := 0 \quad \text{if } f_{ij}^L(u_j^L - u_i^L) > 0. \tag{41}$$

5. Compute correction factors α_{ij} and update the low-order predictor

$$M_L u^{n+1} = M_L u^L + \Delta t \bar{f}^L, \qquad \bar{f}_i^L = \sum_{j\neq i} \alpha_{ij} f_{ij}^L. \tag{42}$$

For moderately small time steps Δt, the linearization of the antidiffusive flux does not increase the solution error significantly as compared to nonlinear flux correction, whereas computing times may reduced by orders of magnitude.

Remark 1 The consistent mass matrix $M_C = \{m_{ij}\}$ for a P_1^{nc} finite element approximation is diagonal [7], that is, $m_{ij} = 0$ for all $j \neq i$. Thus, the raw antidiffusive fluxes (34) and their linearized counterparts (40) reduce to

$$f_{ij}^{(m)} = \frac{d_{ij}}{2}(u_i^{(m)} - u_j^{(m)}) + \frac{d_{ij}}{2}(u_i^n - u_j^n), \tag{43}$$

$$f_{ij}^L = d_{ij}(u_i^L - u_j^L). \tag{44}$$

In the linearized FCT algorithm, the computation of the approximate time derivative (39) is no longer needed and the prelimiting step becomes obsolete since $d_{ij} \geq 0$, $j \neq i$ implies $f_{ij}^L(u_j^L - u_i^L) \leq 0$, i.e. all fluxes have correct sign.

Zalesak's multidimensional FCT algorithm [35] is used to compute correction factors $0 \leq \alpha_{ij} \leq 1$ for constraining the antidiffusive fluxes. For the sake of completeness, we reproduce the algorithmic description from [18]:

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

$$P_i^+ = \sum_{j\neq i} \max\{0, f_{ij}\}, \qquad P_i^- = \sum_{j\neq i} \min\{0, f_{ij}\}. \tag{45}$$

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

$$Q_i^+ = \frac{m_i}{\Delta t}(u_i^{\max} - \tilde{u}_i), \qquad Q_i^- = \frac{m_i}{\Delta t}(u_i^{\min} - \tilde{u}_i). \tag{46}$$

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

$$R_i^+ = \min\left\{1, \frac{Q_i^+}{P_i^+}\right\}, \qquad R_i^- = \min\left\{1, \frac{Q_i^-}{P_i^-}\right\}. \tag{47}$$

4. Check the sign of the raw antidiffusive flux and multiply it by

$$\alpha_{ij} = \begin{cases} \min\{R_i^+, R_j^-\}, & \text{if } f_{ij} > 0, \\ \min\{R_i^-, R_j^+\}, & \text{if } f_{ij} < 0. \end{cases} \tag{48}$$

7 Numerical examples

In this section, we evaluate the accuracy of nonconforming P_1^{nc} and Q_1^{nc} solutions to the linear transport equation (1) and compare the results to those obtained by conforming P_1 and Q_1 approximations. The exact solutions to the chosen two-dimensional benchmark problems are known so that the errors of numerical approximations can be measured by the following quantities

$$E_1 \approx \int_\Omega |u-u_h|\,\mathrm{d}\mathbf{x}, \qquad E_2 \approx \left[\int_\Omega |u-u_h|^2\,\mathrm{d}\mathbf{x}\right]^{1/2}. \tag{49}$$

These integrals were evaluated in classical finite element manner by summing the contribution of all elements of the triangulation, whereby a high-order numerical quadrature rule was adopted to minimize integration errors.

Rotation of a Gaussian hill. The first test case was suggested by Lapin [25] and used, e.g., in [17, 21] to quantify the amount of artificial diffusion engendered by numerical schemes in the presence of small physical diffusion $\epsilon = 10^{-3}$. A counterclockwise rotation about the origin of the domain $\Omega = (-1,1)^2$ is induced by the steady velocity field $\mathbf{v}(x,y) = (-y,x)$. The exact solution to this synthetic benchmark is given by the rotating Gaussian hill profile

$$u(\mathbf{x},t) = \frac{\exp(-\|\mathbf{x}-\hat{\mathbf{x}}(t)\|^2/(0.004\cdot t))}{0.004\cdot\pi t}. \tag{50}$$

Here, $\hat{\mathbf{x}}(t) = (\hat{x}(t),\hat{y}(t))$ denotes the position of the traveling peak which is smeared by diffusion and undergoes a constant rotation in time

$$\begin{bmatrix}\hat{x}(t)\\ \hat{y}(t)\end{bmatrix} = \begin{bmatrix}\cos(t) & -\sin(t)\\ -\sin(t) & \cos(t)\end{bmatrix}\begin{bmatrix}x_0\\ y_0\end{bmatrix}. \tag{51}$$

Expression (50) is used to evaluate the initial profile at the starting time $t_0 = \pi/2$ with $(x_0,y_0) = (0.0, 0.5)$ and to prescribe time-dependent Dirichlet boundary conditions along the whole boundary. Numerical solutions for a full revolution ($t = 5\pi/2$) were computed by the high-order Galerkin method (10), the low-order scheme (18) and the two FCT algorithms outlined in Section 6. Application of the Galerkin method without stabilization does not produce spurious oscillations in this case since the initial peak is smoothed by diffusion.

Figure 3 displays the errors of numerical approximations computed on a sequence of uniform regular grids with equidistant mesh spacing h. For (rotated) linear finite elements, each quadrilateral was subdivided into two triangles by connecting the upper-right corner with the lower-left one. The total number of degrees of freedom for the approximations under consideration are as follows:

	$h=1/16$	$h=1/32$	$h=1/64$	$h=1/128$	$h=1/256$
P_1 and Q_1	1,089	4,225	16,641	66,049	263,169
Q_1^{nc}	2,112	8,320	33,024	131,584	525,312
P_1^{nc}	3,136	12,416	49,408	197,120	787,456

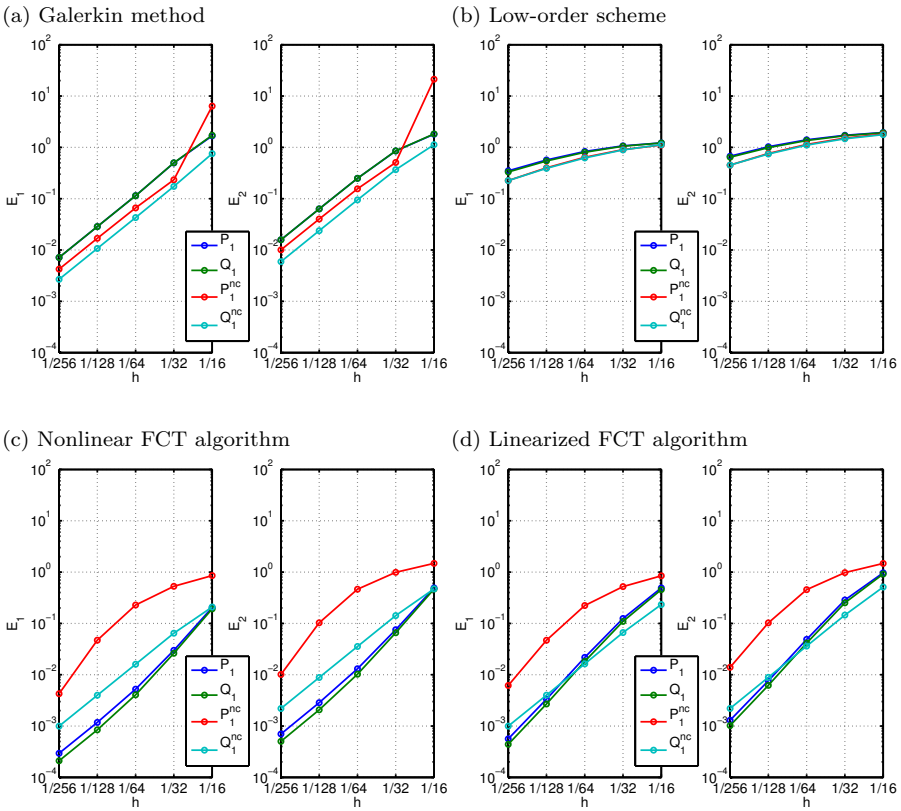


Fig. 3 Rotation of a Gaussian hill: convergence history of L_1- and L_2-errors.

The ratio $\Delta t/h = 0.128$ was kept fixed in all simulations which corresponds to the configuration reported in [17], pp. 182–183. In addition to the error plots, Table 1 shows the values of the estimated order of accuracy $p_q = \log_2(E_q(h)/E_q(h/2))$ computed from solutions on the two finest meshes.

The Galerkin method converges at optimal rate $p = 2$ for all elements under consideration. Application of artificial diffusion deteriorates the accuracy of the low-order scheme below first order which is to be expected. The nonlinear FCT algorithm succeeds in restoring second-order convergence rates nearly perfectly except for the Crouzeix-Raviart element. The error of the flux-corrected P_1^{nc}

	P_1		Q_1		P_1^{nc}		Q_1^{nc}	
	p_1	p_2	p_1	p_2	p_1	p_2	p_1	p_2
Galerkin method	2.00	2.00	2.00	2.00	1.99	1.99	2.00	2.00
Low-order method	0.70	0.61	0.71	0.62	0.83	0.75	0.79	0.72
Nonlinear FCT algorithm	2.00	2.03	2.00	2.04	3.47	3.34	2.00	2.01
Linearized FCT algorithm	2.57	2.55	2.61	2.60	2.92	2.88	2.00	2.01

Table 1 Rotation of a Gaussian hill: estimated order of accuracy on two finest meshes.

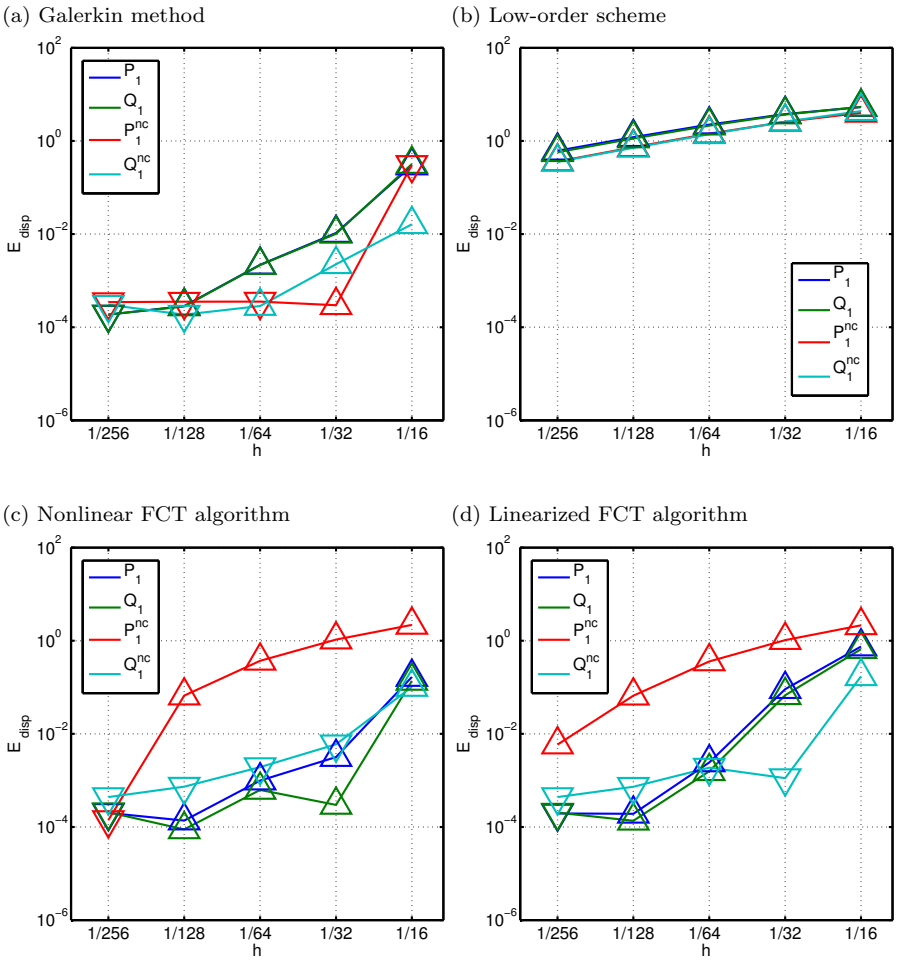


Fig. 4 Rotation of a Gaussian hill: relative dispersion errors E_{disp}.

approximation is nearly one order larger in magnitude but it converges at rate higher than 3. A similar behavior is also observed for the linearized FCT algorithm being applied to rotated linear elements on triangles which may indicate that the asymptotic range has not been reached yet for this particular finite element. The convergence rate for Q_1 approximations ($p_1 = 2.61$) is in very good agreement with value $p = 2.7$ reported in [17], p. 183, where the linearized FCT algorithm is termed CN-FCT-4.

Given an approximate solution u_h, the position $\hat{\mathbf{x}}_h(t) = (\hat{x}_h(t), \hat{y}_h(t))$ of the discrete peak can be estimated as follows [17, 21]

$$\hat{x}_h(t) = \int_\Omega x u_h(\mathbf{x}, t)\,\mathrm{d}\mathbf{x}, \qquad \hat{y}_h(t) = \int_\Omega y u_h(\mathbf{x}, t)\,\mathrm{d}\mathbf{x}. \tag{52}$$

These values may deviate from the exact coordinates (51) due to phase errors. The difference between variance $\sigma^2(t) = 0.004 \cdot t$ of the exact solution (50) and

$$\sigma_h^2(t) = \int_\Omega \|\mathbf{x} - \hat{\mathbf{x}}_h(t)\|^2 u_h(\mathbf{x}, t)\, \mathrm{d}\mathbf{x} \tag{53}$$

can be used to measure the amount of numerical (anti-)diffusion [17, 21]

$$E_{\mathrm{disp}}(t) = \frac{\sigma_h^2(t) - \sigma^2(t)}{\sigma^2(t)} \tag{54}$$

inherent to the underlying finite element scheme. A positive value indicates that the approximate solution contains extra numerical diffusion, whereas negative values of the dispersion error state that some portion of the physical diffusion $\epsilon = 10^{-3}$ is absorbed by numerical antidiffusion.

The absolute values of the relative dispersion errors for the numerical approximations are displayed in Figure 4 using logarithmic scaling for both axes. The actual sign of E_{disp} is indicated by markers '$\triangle$' and '$\triangledown$' for positive and negative values, respectively. Let us briefly comment on the curves for the Crouzeix-Raviart element colored in red. In contrast to the other approximations under consideration, a significant amount of artificial diffusion outlasts the flux-correction step and leads to overdiffusive solution profiles. This correlates with the large error of P_1^{nc} approximations display in Figure 3(c)–(d).

Solid body rotation. This two-dimensional benchmark [27] has a long tradition in testing the performance of numerical schemes for transport problems. The solution is initialized by the geometric shapes depicted in Figure 5(a) which are rotated about the center of domain $\Omega = (0, 1)^2$ by the velocity field

$$\mathbf{v}(\mathbf{x}) = (0.5 - y, x - 0.5). \tag{55}$$

It is incompressible, so that $u_i \sum_j k_{ij}$ vanishes in definition (14). Moreover, absence of physical diffusion is assumed ($\epsilon = 0$) so that the exact solution matches the initial data u_0 after each complete revolution ($t = 2\pi n$, $n \in \mathbb{N}$).

Each solid body has its own region given by a circle of radius $r_0 = 0.15$ which is centered about the point (x_0, y_0). The initial profiles within the three separated regions can be characterized by the following expressions [27]:

a) slotted cylinder centered at $(x_0, y_0) = (0.5, 0.75)$

$$u_0(x, y) = \begin{cases} 1, \text{ if } |x - x_0| \geq 0.025 \text{ or } y \geq 0.85, \\ 0, \text{ otherwise.} \end{cases} \tag{56}$$

b) cone centered at $(x_0, y_0) = (0.5, 0.25)$

$$u_0(x, y) = 1 - r(x, y). \tag{57}$$

c) hump centered at $(x_0, y_0) = (0.25, 0.5)$

$$u_0(x, y) = 0.25[1 + \cos(\pi \min\{1, r(x, y)\})]. \tag{58}$$

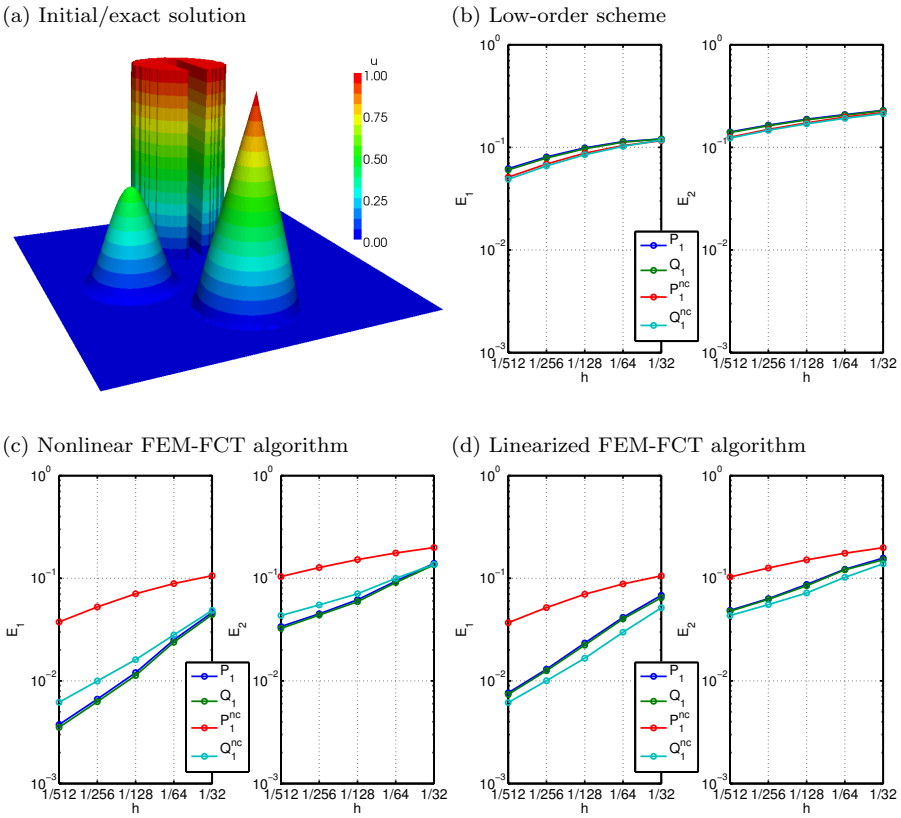


Fig. 5 Solid body rotation: convergence history of L_1- and L_2-errors.

In the rest of the domain where the normalized distance function

$$r(x,y) = \frac{1}{r_0}\sqrt{(x-x_0)^2 + (y-y_0^2)} \tag{59}$$

is larger than one, the solution is initialized by zero which also serves as Dirichlet value to be imposed along the inflow part of the boundary.

Figures 6–8 show approximations to the exact solution at time $t = 2\pi$. They were computed on a regular grid with $h = 1/128$ by the Crank-Nicolson scheme with $\Delta t = 10^{-3}$. Conversion from rectangles to triangles was performed as described before. The raw coefficients u_j of the finite element solution are displayed directly to preclude the generation of artifacts and the addition of extra numerical diffusion caused by projecting u_h into some conforming space.

The overly diffusive profiles produced by the low-order scheme are unacceptable for practical use. Those for Q_1 and Q_1^{nc} elements are not shown since they are visually indistinguishable from Figures 6 (a) and (b), respectively.

Application of the FCT algorithm ensures that the numerical approximation remains bounded between zero and one which means that the algebraic design criteria also apply to the nonconfoming elements under consideration.

(a) P_1 solution (b) P_1^{nc} solution

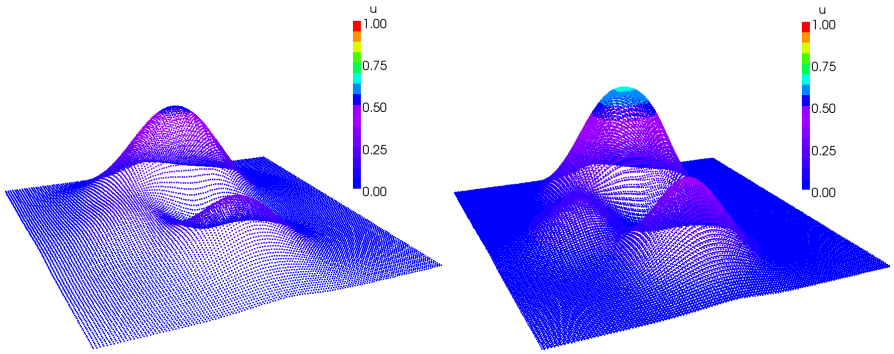


Fig. 6 Solid body rotation: solutions at $t = 2\pi$ computed by the low-order method.

	P_1	Q_1	P_1^{nc}	Q_1^{nc}
Low-ord.	0.53 / 0.29 / 0.25	0.55 / 0.31 / 0.26	0.64 / 0.38 / 0.25	0.65 / 0.41 / 0.26
NL-FCT	1.00 / 0.90 / 0.48	1.00 / 0.90 / 0.48	0.69 / 0.51 / 0.27	1.00 / 0.92 / 0.49
Lin-FCT	1.00 / 0.85 / 0.48	1.00 / 0.86 / 0.48	0.70 / 0.51 / 0.27	1.00 / 0.91 / 0.49

Table 2 Solid body rotation: maximum solution values computed on the $h = 1/128$ grid; numbers represent global maximum value / local max of peak / local max of hump.

However, the flux-corrected P_1^{nc} solutions depicted in Figures 7(c) and 8(c) is nearly as inaccurate as its low-order counterpart. The flux limiter seems to be too restrictive in this case since antidiffusive fluxes are canceled nearly completely. In contrast, the resolution of the solid bodies approximated by P_1, Q_1 and Q_1^{nc} elements is very crisp and free of oscillations. Let us remark that the peak of the cone and the top of the smooth hump are clipped least in the Q_1^{nc} solution as stated by the (local) maximum solution values given in Table 2.

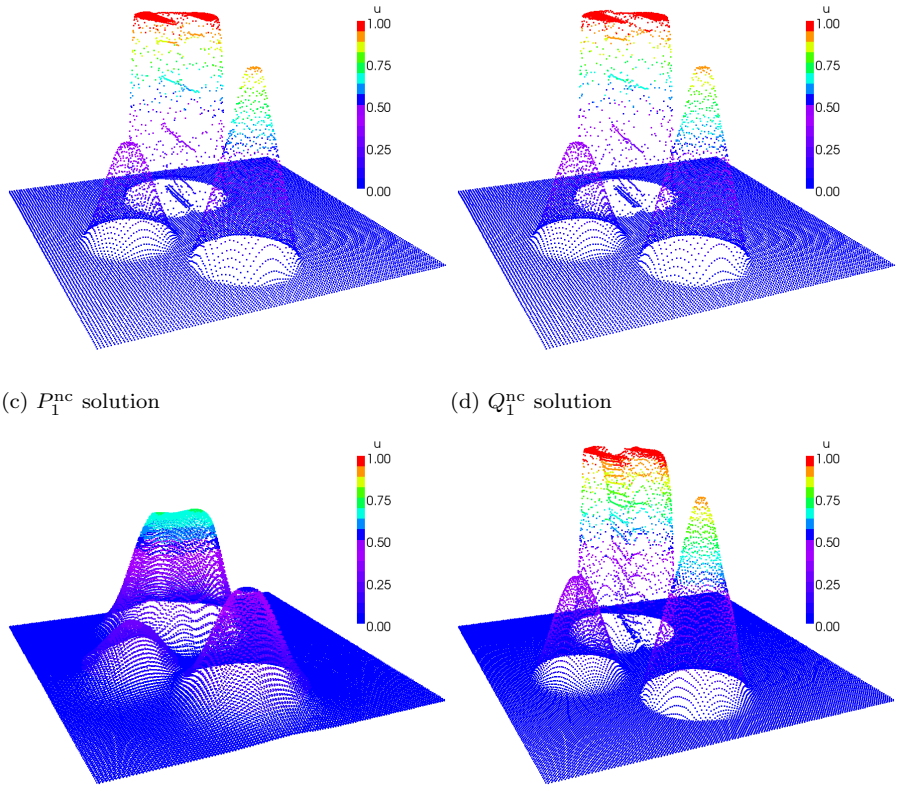
The result of a full convergence study is presented in Figures 5(b)–(d). The poor resolution of flux-corrected P_1^{nc} approximations correlates with considerably larger E_1-/E_2-errors. The expected orders of accuracy given in Table 3 were estimated by computing $p_q = \log_2(E_q(h)/E_q(h/2))$ for $h = 1/512$.

Sparse matrix vector products. This final example briefly addresses the performance of sparse matrix-vector multiplication (SpMV) routines which exploit the special sparsity pattern of finite element matrices. This research area has gained great impetus by the advent of multicore CPUs and throughput-

	P_1		Q_1		P_1^{nc}		Q_1^{nc}	
	p_1	p_2	p_1	p_2	p_1	p_2	p_1	p_2
Low-order method	0.38	0.22	0.38	0.22	0.42	0.25	0.43	0.25
Nonlinear FCT algorithm	0.82	0.41	0.83	0.43	0.48	0.29	0.69	0.34
Linearized FCT algorithm	0.76	0.38	0.76	0.38	0.49	0.29	0.71	0.35

Table 3 Solid body rotation: estimated order of accuracy on two finest meshes.

(a) P_1 solution (b) Q_1 solution



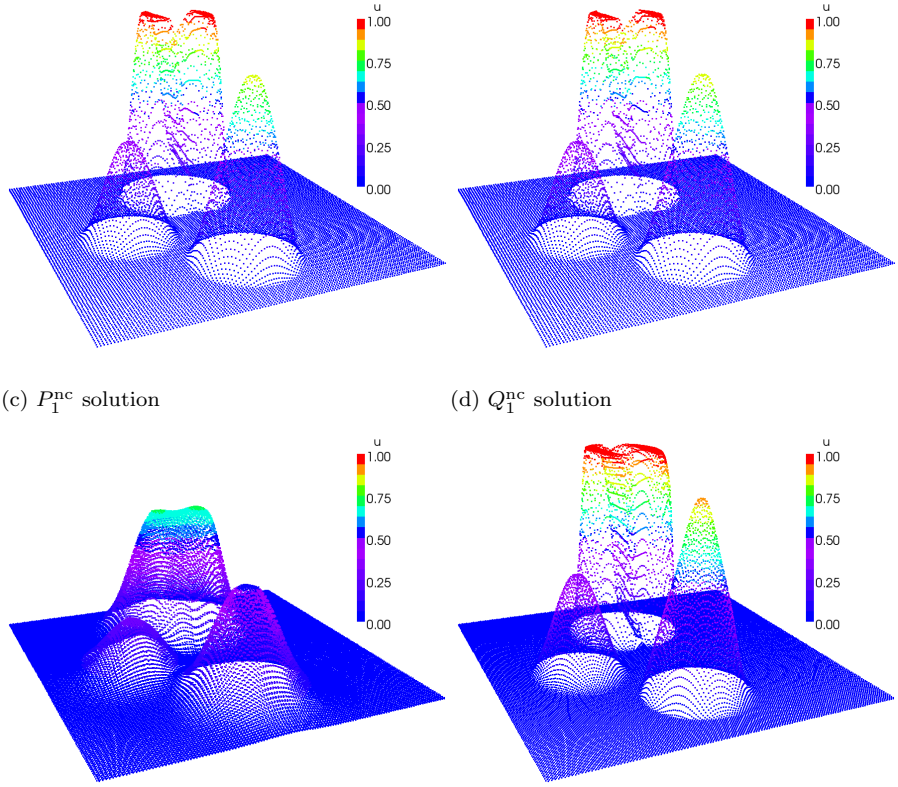
(c) P_1^{nc} solution (d) Q_1^{nc} solution

Fig. 7 Solid body rotation: solutions at $t = 2\pi$ computed by the nonlinear FCT algorithm.

oriented manycore processors such as Graphics Processing Units (GPUs) which offer massive parallelism at reasonably cheap costs. For an overview of this topic we refer to [32–34]. The task is to compute $v \leftarrow Su + v$, where u and v are column vectors and $S = \{s_{ij}\}$ is the stiffness matrix with coefficients (11) assembled on an unstructured mesh with not more than five cells meeting at a common vertex. Thus, the length of matrix rows for Q_1 elements varies between 4 and 11, whereas either 4 or 7 entries per row exist for Q_1^{nc} elements. Independent of the matrix dimension, the number of floating point operations (FLOP) is twice the number of potentially nonzero coefficients in S. A widely accepted performance measure is GFLOP/s which is used to compare the performance of different double-precision SpMV kernels on two target platforms. A measure of practical interest is the normalized time per degree of freedom which allows to decide how much more expensive a particular combination of a SpMV kernel and a finite element is compared to the cheapest combination.

The results obtained on a dual-socket system with Intel Xeon X5550 quad-core CPUs are depicted in Figure 9 (a). 'Standard' SpMV kernels [5] have been

(a) P_1 solution (b) Q_1 solution



(c) P_1^{nc} solution (d) Q_1^{nc} solution

Fig. 8 Solid body rotation: solutions at $t = 2\pi$ computed by the linearized FCT algorithm.

implemented in Fortran 95 using OpenMP for parallelization. The code was compiled by the Intel Fortran Compiler 12.1.4 with options `-xSSE4.2 -ipo -openmp -O3 -unroll-aggressive -ip -fp-model precise -no-prec-div -pad -opt-malloc-options=3` and run using 8 threads for matrix dimensions:

FE	Format	N_{dof}	Nonzeros	Padding
Q_1	CSR	2.133.248	19.179.264	–
Q_1	ELL	2.133.248	23.465.728	22.3%
Q_1^{nc}	CSR	1.066.624	7.456.384	–
Q_1^{nc}	ELL	1.066.624	7.466.368	0.1%

The overall performance strongly depends on the ordering of the degrees of freedom. Stochastic ordering yields the worst performance since the entries of vectors x and y are accessed pseudo randomly, and therefore, cache misses are most likely to occur. The cache miss rate is significantly reduced if the degrees of freedom are ordered either with respect to their geometric position based on their u- and v-coordinates or by hierarchically grouping degrees of freedom

by element macros [31]. Neglecting the randomly ordered pathological case, the time per DOF for the Q_1 finite element matrix stored in CSR format and its Q_1^{nc} counterpart stored in ELL format differs by factor 2 at most.

A slightly different behavior is observed for the SpMV kernels implemented in the open-source software HONEI [4]. The optimized CUDA implementation targets NVIDIA GPUs as current representatives of modern throughput-oriented manycore systems. For achieving optimal performance on such hardware it is crucial to chose a 'good' data layout. From the benchmark results given in Figure 9 (b) one can conclude that the CSR format is not well-suited for this type of hardware. To explore its full potential the GPU-friendly ELLR-T variant [32] of the original ELLPACK format has been used which is currently one of the most efficient formats for SpMV on GPUs. The significant reduction of the time per DOF justifies sacrificing some memory space.

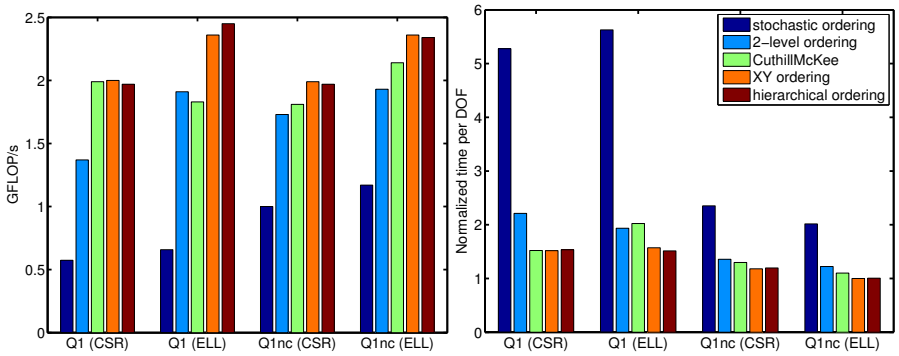
The performance of SpMV kernels is very sensitive to the particular matrix pattern so that a large spectrum of test matrices from 'practical' applications is typically included in publications [2,32–34]. To the author's best knowledge, performance studies for Q_1^{nc} approximations have not been published before which led us to include the above benchmark results into the present paper.

8 Conclusions

This paper hopefully sheds some light on the extension of algebraic flux correction schemes to nonconforming finite elements. Theoretical analysis of the Crouzeix-Raviart and Rannacher-Turek element shows that both nonconforming approximations satisfy the necessary prerequisites for applying algebraic flux correction. However, practical computations demonstrate that nonconforming P_1^{nc} elements fail completely within AFC-type methods yielding over-diffusive approximate solutions. Our preliminary explanation for this behavior is as follows: The local upper and lower bounds (33) imposed on the converged end-of-step solution are computed using the values of the low-order approximations (32) or (38) at surrounding degrees of freedom. For P_1^{nc} discretizations, the 'stencil' of this process comprises only five points. The resulting bounds may be too pessimistic so that the flux limiter rejects the antidiffusive correction to a large extent. Two-dimensional benchmark computations for flux-corrected Q_1^{nc} approximations show that this type of element does provide an accurate resolution of steep fronts without generating oscillations if the integral mean value based variant is adopted. It should be mentioned that the use of its three-dimensional counterpart is not recommended in [30] (see Remark 2.6, p.20 and Section 4.8, pp. 69-71). In short, basis functions on the reference element depend on the particular curved face, and therefore, they need to be recomputed for each cell unless an affine reference map is used. There are also some theoretical aspects which are not fully solved in 3D.

In summary, it seems that AFC-type methods based on low-order nonconforming approximations are not to be recommended for practical applications.

(a) Dual-socket Intel Xeon quad-core X5550



(b) Tesla C2070 GPU (ECC off)

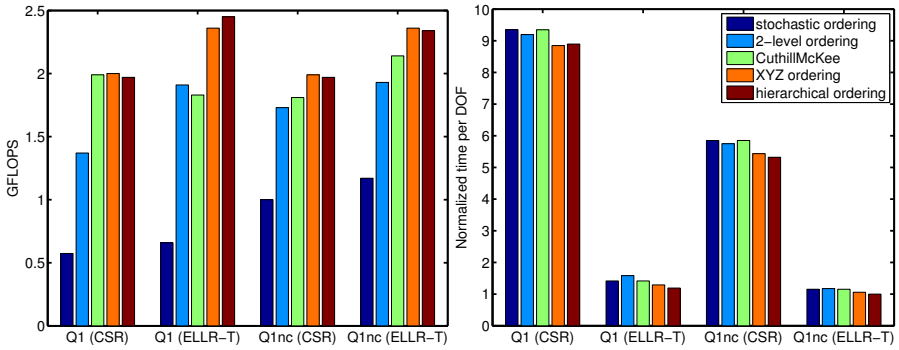


Fig. 9 Comparing the performance of SpMV for finite element matrices stored in CSR and ELL format on CPUs and CSR and ELLR-T format on GPUs.

Acknowledgements The author wishes to thank professor Friedhelm Schieweck from Otto-von-Guericke Universität Magdeburg for fruitful discussions on nonconforming finite element approximations. He is also very thankful to the anonymous reviewers for giving helpful comments and for sharing their opinion on the restricted usefulness of nonconforming AFC-type methods. Finally, the author expresses gratitude to his colleagues Dipl-Infs. Markus Geveler and Dirk Ribbrock for providing benchmark results for SpMV implementations on GPUs. This work was supported by the German Research Foundation (DFG) under grant SFB 708 "3D-Surface Engineering of Tools for Sheet Metal Forming – Manufacturing, Modelling, Machining".

References

1. D.G. Anderson: Iterative procedure for nonlinear integral equations. *J. Assoc. Comput. Mach.* 12 (1965) 547–560.
2. N. Bell, M. Garland: Implementing sparse matrix-vector multiplication on throughput-oriented processors. In: SC '09: Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis, (2009) 1–11.
3. M. Crouzeix. P.-A. Raviart: Conforming and nonconforming fnite element methods for solving the stationary Stokes equations I. Revue française d'automatique, informatique, recherche opérationnelle. Mathématique, 7(3) (1973), 33–75.

4. D. van Dyk, M. Geveler, S. Mallach, D. Ribbrock, D- Gddeke, C. Gutwenger: HONEI: A collection of libraries for numerical computations targeting multiple processor architectures, *Comput. Phys. Commun.*, 180(12) (2009) 2534–2543
5. FeatFlow: `http://www.featflow.de`, access date: Dec 4, 2012.
6. C.A.J. Fletcher: The group finite element formulation. *Comput. Methods Appl. Mech. Engrg.* 37 (1983) 225–243.
7. T. Gallouët, L. Gastaldo, R. Herbin, J.-C. Latché: An unconditionally stable pressure correction scheme for the compressible barotropic Navier-Stokes equations. *ESAIM: Mathematical Modelling and Numerical Analysis* 42 (2008) 303–331.
8. I. Georgiev, J. Kraus, S. Margenov: Multilevel preconditioning of rotated bilinear nonconforming FEM problems. *Comput. Math. Appl.* 55 (2008) 2280–2294.
9. R. Grimes, D. Kincaid, D. Young: ITPACK 2.0 Users Guide. Technical Report CNA-150, Center for Numerical Analysis, University of Texas, August 1979. `http://rene.ma.utexas.edu/CNA/ITPACK/`
10. A. Jameson: Computational algorithms for aerodynamic analysis and design. *Appl. Numer. Math.* 13 (1993) 33–422.
11. V. John: Parallele Lösung der inkompressiblen Navier-Stokes Gleichungen auf adaptiv verfeinerten Gittern. Otto-von-Guericke Universität Magdeburg, 1997. PhD thesis.
12. K.S. Kang: P_1 nonconforming finite element methods for the solution of radiation transport problems. *SIAM J. Sci. Comput.* 25(2) (2003) 369–384.
13. 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.
14. D. Kuzmin: On the design of general-purpose flux limiters for implicit FEM with a consistent mass matrix. I. Scalar convection. *J. Comp. Phys.* 219 (2006) 513–531.
15. D. Kuzmin: On the design of algebraic flux correction schemes for quadratic finite elements. *Comput. Appl. Math.*, 218 (2008) 79–87.
16. D. Kuzmin: Explicit and implicit FEM-FCT algorithms with flux linearization. *J. Comput. Phys.* 228 (2009) 2517–2534.
17. D. Kuzmin: *A Guide to Numerical Methods for Transport Equations*. University Erlangen-Nuremberg, Erlangen (2010). `http://www.mathematik.uni-dortmund.de/~kuzmin/Transport.pdf`
18. D. Kuzmin: Algebraic flux correction I. Scalar conservation laws. In: D. Kuzmin, R. Löhner, S. Turek (eds): *Flux-Corrected Transport: Principles, Algorithms, and Applications*. Springer, 2nd edition 2012, 145–192.
19. D. Kuzmin: Linearity-preserving flux correction and convergence acceleration for constrained Galerkin schemes. *J. Comput. Appl. Math.* 236 (2012) 2317–2337.
20. D. Kuzmin, M. Möller, S. Turek: High-resolution FEM-FCT schemes for multidimensional conservation laws. *Comput. Methods Appl. Mech. Engrg.* 193 (2004) 4915–4946.
21. D. Kuzmin, M. Möller: Algebraic flux correction I. Scalar conservation laws. In: D. Kuzmin, R. Löhner, S. Turek (eds): *Flux-Corrected Transport: Principles, Algorithms, and Applications*. Springer, 2005.
22. D. Kuzmin, S. Turek: Flux correction tools for finite elements. *J. Comp. Phys.* 175 (2002) 525–558.
23. D. Kuzmin, M. Möller, S. Turek: Multidimensional FEM-FCT schemes for arbitrary time-stepping. *Int. J. Numer. Meth. Fluids* 42 (2003) 265–295.
24. D. Kuzmin, S. Turek: High-resolution FEM-TVD schemes based on a fully multidimensional flux limiter. *J. Comp. Phys.* 198 (2004) 131–158.
25. A. Lapin. University of Stuttgart. Private communication (2001).
26. W.J. Layton, J.M. Maubach, P.J. Rabier: Robustness of an elementwise parallel finite element method for convection-diffusion problems. *SIAM J. Sci. Comput.* 19(6) (1998) 1870–1891.
27. R.J. LeVeque: High-resolution conservative algorithms for advection in incompressible flow. *SIAM J. Numer. Anal.* 33 (1996) 627–665.
28. M. Möller: On the design of non-conforming high-resolution finite element schemes. In: J. Eberhardsteiner et al. (eds): Proceedings of the 6th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2012), Vienna.
29. R. Rannacher, S. Turek: A simple nonconforming quadrilateral Stokes element. *Numer. Meth. PDEs*, 8 (1992) 97–111.

30. F. Schieweck: *Parallele Lösung der stationären inkompressiblen Navier-Stokes Gleichungen*. Otto-von-Guericke Universität Magdeburg, Fakultät für Mathematik, 1997. Habilitation thesis.
31. S. Turek: On Ordering Strategies in a Multigrid Algorithm. In: Proceedings of 8th GAMM-Seminar, Volume 41 of Notes on Numerical Fluid Mechanics, 1992.
32. F. Vázquez, J.J. Fernández, E.M. Garzón: Automatic tuning of the sparse matrix vector product on GPUs based on the ELLR-T approach. *Parallel Computing* 38 (2012) 408–420.
33. S. Williams, L. Oliker, R. Vuduc, J. Shalf, K. Yelick, J. Demmel: Optimization of sparse matrix-vector multiplication on emerging multicore platforms. In: Proceedings of the 2007 ACM/IEEE conference on Supercomputing. ACM (2007) 38:1–38:12.
34. S. Williams, N. Bell, J.W. Choi, M. Garland, L. Oliker, R. Vuduc: Sparse Matrix-Vector Multiplication on Multicore and Accelerators. In: J. Kurzak, J.J. Dongarra, D.A. Bader (eds): *Scientific Computing with Multicore and Accelerators*. CRC Press (2011) 83–109.
35. S.T. Zalesak: Fully multidimensional flux-corrected transport algorithms for fluids. *J. Comput. Phys.* 31 (1979) 335–362.