Efficient solution techniques for implicit finite element schemes with flux limiters

M. Möller[*]

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

Abstract

The *algebraic flux correction* (AFC) paradigm is equipped with efficient solution strategies for implicit time-stepping schemes. It is shown, that Newton-like techniques can be applied to the nonlinear systems of equations resulting from the application of high-resolution flux limiting schemes. To this end, the Jacobian matrix is approximated by means of first- or second-order finite differences. The edge-based formulation of algebraic flux correction schemes can be exploited to devise an efficient assembly procedure for the Jacobian. Each matrix entry is constructed from a differential and an average contribution edge-by-edge. The perturbation of solution values affects the nodal correction factors at neighboring vertices so that the stencil for each individual node needs to be extended. Two alternative strategies for constructing the corresponding sparsity pattern of the resulting Jacobian are proposed. For nonlinear governing equations, the contribution to the Newton matrix which is associated with the discrete transport operator is approximated by means of divided differences and assembled edge-by-edge. Numerical examples for both linear and nonlinear benchmark problems are presented to illustrate the superiority of Newton methods as compared to the standard defect correction approach.

Key Words: Newton-like solution techniques; nonlinear solvers;
high-resolution schemes; flux correction; finite elements

1 Introduction

For decades, the development of reliable discretization techniques for convection dominated flows has been one of the primary interests in Computational Fluid Dynamics. A variety of stabilization techniques (streamline diffusion, edge stabilization / interior penalty), and high-resolution schemes based on flux/slope limiting have been presented in the literature to combat the formation of nonphysical oscillations which would be generated otherwise. As a matter of fact, a serious disadvantage of many existing numerical schemes is their lack of generality. The foundations of modern high-resolution schemes were developed in the finite difference framework using essentially one-dimensional concepts and, typically, geometric criteria. As a consequence, most algorithms popular today are only suitable for Cartesian meshes and/or explicit time-stepping.

[*] Correspondence to: matthias.moeller@math.uni-dortmund.de

The origins of modern high-resolution schemes can be traced back to the renowned SHASTA scheme by Boris and Book [5] who set up the *flux corrected transport* (FCT) methodology in the realm of finite differences. The fully multidimensional generalization proposed by Zalesak [46] has put FCT algorithms into a very general framework: The ultimate solution is computed by adaptively blending linear approximations of high and low order so as to prevent the formation of wiggles. This reformulation has paved the way for the generalization of FCT concepts to explicit Galerkin schemes based on linear/bilinear finite element discretizations on unstructured meshes [31, 32]. As an alternative, *total variation diminishing* (TVD) schemes have been introduced in the context of finite differences [16, 17] and extended to explicit finite element/finite volume schemes [3, 33].

Notwithstanding the impressive simulation results produced by the explicit FEM-FCT algorithm by Löhner and his coworkers [31, 32], the early high-resolution methods exhibit some inherent limitations. In the first place, classical FCT schemes are based on an explicit correction of the auxiliary low-order solution whose local extrema serve as upper/lower bounds for the sum of limited antidiffusive fluxes. Due to the explicit nature, the time step must satisfy a restrictive 'CFL' condition which drastically increases the computational costs especially for steady state problems. Furthermore, the use of stable linear discretizations is mandatory for the overall success of the flux limiter. In particular, the use of an unstable high-order method may give rise to nonlinear instabilities which manifest themselves in significant distortions of the 'corrected' solution profiles. It is well known, that the standard Galerkin discretization calls for a suitable stabilization. This extra term not only increases the cost of matrix assembly but also engenders artificial dissipation whose magnitude is controlled by some heuristic parameter.

These severe restrictions have lead to the development of a generalized FEM-FCT methodology introduced by Kuzmin and Turek in [29] and refined by Kuzmin *et al.* in [24, 27, 28]. Flux correction of FCT type is readily applicable to Galerkin schemes with a consistent mass matrix combined with (semi-)implicit time discretizations. The use of a second-order Crank-Nicolson scheme suggests itself for the simulation of strongly time-dependent problems. In comparison to their explicit counterparts, semi-implicit schemes considerably relax the 'CFL' condition and allow for employing moderate time steps. Moreover, unconditionally stable implicit methods can be operated at arbitrarily large time steps (unless iterative solvers fail to converge or the positivity criterion is violated) which makes them a favorable tool for the efficient treatment of steady state problems.

For flux correction of FCT type, the amount of admissible antidiffusion is inversely proportional to the time step, which compromises the advantages of unconditionally stable implicit schemes. On the other hand, flux correction of TVD type is independent of the time step and thus a good candidate for the treatment of stationary problems. Standard TVD limiters can be integrated into unstructured grid codes and applied edge-by-edge [3, 33] or node-by-node [25, 30], so as to control the slope ratio for a local 1D stencil or the net antidiffusion, respectively. Recently, a subtle consolidation of FCT and TVD paradigms has been proposed by Kuzmin [23], who presented a *general purpose* limiter that can be applied to transient and steady state problems alike. In essence, the new algorithm represents the successful marriage of a symmetric flux limiter for the contribution of the consistent mass matrix, which has to be sacrificed in classical TVD schemes, and an upwind-biased one for monitoring the (anti-)diffusive contribution of convective fluxes.

In any case, the price to be paid for the great flexibility provided by (semi-)implicit high-resolution flux correction schemes is a nonlinear algebraic system that has to be solved in each (pseudo-)time step even if the problem at hand is linear. Due to the fact that the nonlinearity originates from the discretization it has no continuous counterpart which could be differentiated analytically and supplied to Newton-type methods. Since flux limiters typically make use of non-differentiable functions, the 'Jacobian' is approximated by means of divided differences. In the present paper, an efficient assembly algorithm for the Newton matrix is devised by exploiting the edge-based formulation of algebraic flux correction schemes. In general, the sparsity pattern of the finite element matrix needs to be extended by one connectivity layer. Moreover, the approximate Jacobian for a nonlinear transport operator can be decomposed into edge contributions so as to allow for an efficient edge-by-edge assembly. Numerical examples are presented for linear and nonlinear two-dimensional stationary benchmark problems to demonstrate the benefits of Newton-type methods as compared to standard defect correction approaches.

2 Algebraic flux correction

In this paper, we adopt the *algebraic flux correction* paradigm [23–30] which consists of imposing mathematical constraints on discrete operators so as to achieve certain matrix properties. A detailed description of this family of high-resolution schemes can be found in the aforementioned publications. As a model problem, consider a stationary conservation law for a scalar quantity u whereby $\mathbf{f}$ is a generic and possibly nonlinear flux function

$$\nabla \cdot \mathbf{f}(u) = 0 \qquad \text{in} \quad \Omega. \tag{1}$$

For the time being, let us assume that $\mathbf{f}$ is composed by convective and diffusive fluxes, i.e., $\mathbf{f}(u) = \mathbf{v}u - d\nabla u$, where $\mathbf{v}$ denotes a nonuniform velocity field and d is the diffusion coefficient. The above problem statement is completed by the prescription of boundary conditions of Dirichlet and/or Neumann type. Let the equation at hand be discretized by a high-order finite element (Galerkin) method and apply algebraic flux correction to turn it into a high-resolution approximation. Even in case the conservation law (1) is linear, this yields a **nonlinear** algebraic equation system for the vector of nodal values

$$K^*(u)u = 0, \tag{2}$$

where the modified transport operator exhibits the following structure [25]

$$K^*(u) = L + F(u) = K + D + F(u). \tag{3}$$

Here, $K = \{k_{ij}\}$ denotes the original transport operator resulting from the Galerkin finite element approximation of convective terms. The artificial diffusion operator $D = \{d_{ij}\}$ is designed to eliminate all negative off-diagonal coefficients from the high-order operator in order to turn K into its *local extremum diminishing* (LED) counterpart $L = K + D$. The error induced by this so-called 'discrete upwinding' [29] is compensated by applying an antidiffusive correction term $F(u)$ which will be addressed below.

Due to the fact that D is a discrete diffusion operator which is defined as a symmetric matrix with zero row/column sums, the term Du can be decomposed into a sum of skew-symmetric internodal fluxes which are associated with the edges of the sparsity graph [25]

$$(Du)_i := -\sum_{j\neq i} f_{ij}, \qquad f_{ij} = d_{ij}(u_i - u_j) = -f_{ji}. \tag{4}$$

A natural choice for the artificial diffusion coefficient for the edge $\vec{ij}$ is [29]

$$d_{ij} = \max\{-k_{ij}, 0, -k_{ji}\} = d_{ji}. \tag{5}$$

As a result, the off-diagonal coefficients of the low-order operator $l_{ij} := k_{ij} + d_{ij} \geq 0$ are nonnegative which is a prerequisite for our scheme to possess the LED property introduced by Jameson in [20, 21]. Due to the fact that the discrete diffusion operator D exhibits zero row sums, the diagonal entries of L are given by

$$l_{ii} := k_{ii} - \sum_{j\neq i} d_{ij}. \tag{6}$$

For our purpose it is expedient to introduce the following convention: Without loss of generality, let the edge $\vec{ij}$ be oriented so that $l_{ij} \leq l_{ji}$, which implies that node i is located 'upwind' and corresponds to the row number of the eliminated negative coefficient [25].

The skew-symmetric antidiffusive flux f_{ij} from node j into its upwind neighbor i which offsets the error induced by our discrete upwinding is defined in (4). In order to prevent the formation of spurious oscillations which would be generated otherwise, it is multiplied by a suitable correction factor $0 \leq \alpha(u)_{ij} \leq 1$ which is determined by means of a multidimensional flux limiter (see below). As a result, the net antidiffusion which is applied to the upwind node i can be expressed as follows

$$(Fu)_i = \sum_{j\neq i} f^*_{ij}, \qquad f^*_{ij} := \alpha_{ij} f_{ij}. \tag{7}$$

By definition the downwind node j receives the same flux $f^*_{ji} := -f^*_{ij}$ which is of the same magnitude but exhibits the opposite sign so that mass conservation is guaranteed.

Putting it all together, the contribution of the modified transport operator $K^*(u)$ applied to the vector of nodal unknowns u can be expressed for each node i as follows

$$(K^*u)_i = \sum_{j\neq i} k^*_{ij}(u)(u_j - u_i) + u_i \sum_j k^*_{ij}(u). \tag{8}$$

Here, the solution dependent matrix coefficients are given by

$$k^*_{ii}(u) = k_{ii} - \sum_{j\neq i}[1 - \alpha_{ij}(u)]d_{ij}, \qquad k^*_{ij}(u) = k_{ij} + [1 - \alpha_{ij}(u)]d_{ij}, \tag{9}$$

whereby the reactive term in equation (8) represents a discrete counterpart of $-u\nabla \cdot \mathbf{v}$. It vanishes for divergence-free velocity fields and is responsible for a physical growth of local extrema otherwise since it is not affected by discrete upwinding and algebraic flux correction. This can be easily shown be considering the definition of the matrix entries given in (9) and recalling the fact that $\sum_j d_{ij} = 0$, hence

$$\sum_j k_{ij}^*(u) = k_{ii}^*(u) + \sum_{j\neq i} k_{ij}^*(u) = \sum_j k_{ij}. \tag{10}$$

At the end of the day, the resulting transport operator (8) represents a nonlinear combination of the low-order scheme ($\alpha_{ij} \equiv 0$) and the original high-order one ($\alpha_{ij} = 1$). The task of the flux limiter is to determine optimal correction factors α_{ij} so as to remove as much artificial diffusion as possible without generating spurious oscillations.

The idea of node-based flux limiting can be traced back to the multidimensional FCT limiter proposed by Zalesak [46] and has been adopted in the AFC framework [23–30]. In short, antidiffusive fluxes $f_{ij} = p_{ij}(u_j - u_i)$ which are proportional to solution differences multiplied by coefficients $p_{ij} \leq 0$ violate the LED criterion introduced in [20, 21], and hence, need to be limited. On the other hand, edge contributions with positive coefficients resemble diffusive fluxes and are harmless. Some portion of antidiffusion, say, from node j into node i is only admissible if it can be interpreted as a diffusive flux from another node, that is, if there exists a solution-dependent coefficient $q_{ik} \geq 0$ such that $f_{ij} = q_{ik}(u_k - u_i)$.

A general framework for flux correction in multidimensions is presented in [23]. Here, we will only address *upwind-biased* flux limiters which are appropriate for stationary problems of the form (1). If both off-diagonal entries of the high-order operator K are negative the raw flux defined in (4) needs to be 'prelimited' according to

$$f_{ij}' = \min\{d_{ij}, l_{ji}\}(u_i - u_j) \tag{11}$$

which reduces to $f_{ij} = d_{ij}(u_i - u_j)$ otherwise. For each node, the net antidiffusion may consist of both positive and negative edge contributions, but in the worst case, all fluxes have the same sign. Hence, it is worthwhile to treat the positive and negative ones separately, as proposed by Zalesak [46]. The total amount of raw antidiffusion received by node i from its downwind neighbors is given by

$$P_i^{\pm} = \sum_{j\in\mathcal{J}_i} \substack{\max\\ \min}\{0, f_{ij}'\}, \quad \text{where} \quad \mathcal{J}_i = \{j \neq i \,|\, 0 = l_{ij} < l_{ji}\}. \tag{12}$$

The upper/lower bounds to be imposed by the flux limiter can be computed from the off-diagonal coefficients of the low-order operator L which are nonnegative by construction

$$Q_i^{\pm} = \sum_{j\neq i} l_{ij} \substack{\max\\ \min}\{0, u_j - u_i\}, \qquad l_{ij} \geq 0, \quad \forall j \neq i. \tag{13}$$

For each node, the admissible antidiffusion is given by the nodal correction factors

$$R_i^+ = \min\{1, Q_i^+/P_i^+\}, \qquad R_i^- = \min\{1, Q_i^-/P_i^-\}. \tag{14}$$

The final correction factor α_{ij} is taken as the nodal multiplier of the upwind node i

$$\alpha_{ij} = \begin{cases} R_i^+, & \text{if } f_{ij}' > 0, \\ R_i^-, & \text{otherwise} \end{cases} \tag{15}$$

so that the (prelimited) antidiffusive flux (11) can be corrected according to $f_{ij}^* := \alpha_{ij} f_{ij}'$. For a detailed description of multidimensional flux limiting, including rigorous positivity proofs, the interested reader is referred to [23–25] and the references therein.

3 Nonlinear solution strategies

A common practice in the computation of steady state solutions of partial differential equations is to perform pseudo-time stepping for the *method-of-lines* model

$$\frac{\partial u}{\partial \tau} + \nabla \cdot \mathbf{f}(u) = 0. \tag{16}$$

This is the pseudo-transient counterpart of our conservation law (1), where τ denotes the artificial time variable. Let us make use of algebraic flux correction to approximate the spatial derivatives so as to obtain a system of first-order differential algebraic equations (DAE) for the vector of nodal values. For the integration in time, any numerical algorithm that is designed to solve initial value problems may be employed. Since we are only interested in the steady state solution, the fully implicit backward Euler scheme

$$M_L \frac{u^{n+1} - u^n}{\Delta\tau} = K^*(u^{n+1})u^{n+1} \tag{17}$$

is the method of choice which is solved repeatedly until changes in successive solution values are negligibly small. In the above equation $M_L = \{m_i\}$ denotes the lumped counterpart of the consistent mass matrix which comes from the Galerkin discretization of the time derivative. Due to the fact that the backward Euler method is unconditionally stable, it can be operated at arbitrarily large pseudo-time steps $\Delta\tau$ which may even be varied in each iteration [35]. Interestingly enough, algebraic flux correction methods of TVD type [25, 30] are derived on the semi-discrete level, and hence, independent of the employed time step. As a consequence, the converged steady state solution to problem (17) can be computed very efficiently if $\Delta\tau \to \infty$ for $n \to \infty$ without loss of accuracy.

It is noteworthy, that the application of the implicit Euler method to (pseudo-)transient problems leads to algebraic equations which are very similar to those resulting from the use of under-relaxation in the context of steady state flows [14]. If the same time step is adopted for all equations this corresponds to taking a variable under-relaxation factor. Conversely, the use of a constant under-relaxation factor is equivalent to adopting a different time step for the computation of each nodal solution value u_i^{n+1}.

3.1 Iterative defect correction

Note that even if the governing equation is linear, equation (17) exhibits some nonlinearity due to flux correction which calls for an iterative solution strategy. To make the presentation self-contained let us recapitulate nonlinear solution techniques in a more general framework and apply them to the residual form of the algebraic system (17) afterwards

$$\mathcal{R}(u^{n+1}) = [M_L - \Delta\tau K^*(u^{n+1})]u^{n+1} - M_L u^n = 0. \tag{18}$$

Given the global vector of unknowns u which may be either the solution from the last time step ($u = u^n$) or an initial guess ($u = u_0$), the end-of-step solution u^{n+1} can be computed, e. g., by the (possibly relaxed) fixed-point iteration with $u^{(0)} = u$

$$u^{(m+1)} = u^{(m)} + \omega^{(m+1)} \left[C^{(m)}\right]^{-1} r^{(m)}, \qquad m = 0, 1, 2, \dots \tag{19}$$

Here, $r^{(m)} = -\mathcal{R}(u^{(m)})$ denotes the residual of the m-th cycle and $C^{(m)}$ is a suitable 'preconditioner' to be defined below. The iteration process is terminated based on a required drop in the norm of the residual and/or a sufficiently small solution increment

$$\|\mathcal{R}(u^{(m+1)})\| \leq \epsilon_1 \|\mathcal{R}(u^{(0)})\|, \qquad \|\Delta u^{(m+1)}\| \leq \epsilon_2 \|u^{(m)}\|.$$

Here, ϵ_1 and ϵ_2 are user-defined parameters, $\|\cdot\|$ denotes an arbitrary norm and the term $\Delta u^{(m+1)} = u^{(m+1)} - u^{(m)}$ refers to the solution increment to be computed.

Note that, monitoring only the relative changes may be insufficient since small values of the relaxed solution increment may be also caused by strong under-relaxation $|\omega^{(m+1)}| \ll 1$ which leads to extremely slow convergence. Moreover, theoretically motivated stopping criteria can be designed as discussed in [1, 2] making use of the fact, that the iteration procedure results from a finite element approximation of a partial differential equation.

In a practical implementation, the 'inversion' of the preconditioner matrix $C^{(m)}$ is also performed by a suitable iteration procedure over a sequence of **linear** equations

$$C^{(m)} \Delta u^{(m+1)} = r^{(m)}, \qquad m = 0, 1, 2, \dots \tag{20}$$

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

$$u^{(m+1)} = u^{(m)} + \omega^{(m+1)} \Delta u^{(m+1)}, \qquad u^{(0)} = u. \tag{21}$$

It remains to specify a suitable preconditioner. A multivariate Taylor expansion of the residual term $\mathcal{R}(u^{(m+1)})$ about the current state $u^{(m)}$ yields the following approximation

$$\mathcal{R}(u^{(m+1)}) \simeq \mathcal{R}(u^{(m)}) + J^{(m)}(u^{(m+1)} - u^{(m)}) \tag{22}$$

which requires the evaluation of the Jacobian matrix at the last iterate $u^{(m)}$

$$J^{(m)} = \frac{\partial \mathcal{R}(u^{(m)})}{\partial u}. \tag{23}$$

Neglecting terms of higher order curvature in the linearized model (22) and recalling the postulated relation $\mathcal{R}(u^{n+1}) = 0$ one ends up with the well-known *Newton method*

$$u^{(m+1)} = u^{(m)} - \left[J^{(m)}\right]^{-1} \mathcal{R}(u^{(m)}). \tag{24}$$

The formal differentiation of the nonlinear residual (18) with respect to the components of u makes it possible to evaluate the Jacobian (23) and define the preconditioner as follows

$$C^{(m)} = M_L - \Delta\tau \left[\frac{\partial K^*(u^{(m)})}{\partial u} u^{(m)} + K^*(u^{(m)}) \right]. \tag{25}$$

Then, the fixed point iteration (19) yields the consistent Newton method (24) which can be damped ($0 < \omega < 1$) to improve robustness. Another attractive algorithm for the solution of nonlinear equations can be derived from the preconditioner (25) by neglecting the differential term as well as the antidiffusive contribution (see Eq. (3)) such that

$$C^{(m)} = M_L - \Delta\tau L(u^{(m)}). \tag{26}$$

As a consequence, the iteration scheme (19) reduces to the fixed-point defect correction procedure [44] ‘preconditioned’ by the monotone low-order operator which is designed to be an *M-matrix* and hence exhibits amenable matrix properties [28,30].

It is worth mentioning that intermediate solutions $u^{(m)}$ are not required to be positive so that convergence of the fixed-point iteration (19) is a prerequisite for positivity. The use of an iterative solver applicable to large sparse, non-symmetric systems of linear equations is mandatory. In our experience, standard Krylov methods, e.g., BiCGSTAB or GMRES, combined with an ILU-type preconditioner will do. Moreover, implicit under-relaxation [14,38] can be applied to enhance the diagonal dominance of $C^{(m)}$.

It is well known, that the performance of Newton’s method strongly depends on the quality of the initial guess u. For the solution of the steady Euler equations, Hemker *et al.* [18] suggest a two-step defect correction approach. A provisional first-order solution is computed for the stationary problem directly, i.e., without resorting to pseudo-time stepping. Next, the low-order profile is used as initial guess for a second-order accurate defect correction iteration preconditioned by the first-order operator.

Within a Newton iteration approach, this idea may be adopted as follows. As before, let u denote the initial solution for the current iteration step, i.e., $u = u^n$ for time dependent problems or $u = u_0$ in the stationary case. In order to obtain a usable initial guess for the Newton iteration, perform a small number of ‘presmoothing’ steps. To this end, the low-order operator (26) can be applied either *per se* (without resorting to algebraic flux correction) or as a preconditioner within a high-resolution flux/defect correction scheme. After a few iterations, the preconditioner (25) is used so that the iteration procedure (19) yields Newton’s algorithm which is supposed to converge faster.

3.2 Globalization

Since the linear system (20) is solved iteratively, and hence, the computation of the ‘exact’ solution of the Newton equation is quite costly, the resulting algorithm is categorized as an *inexact Newton method* [11]. Obviously, the accuracy of the (inner) linear solver greatly affects the convergence behavior of the (outer) nonlinear Newton algorithm. If the linear subproblems are solved too inaccurately more Newton steps are required, and hence, the nonlinear convergence rate deteriorates. Conversely, a very small tolerance for the linear solver results in a drastic increase of inner iterations which does not pay off. Moreover, if $u^{(m)}$ is not sufficiently close to the desired root then the linearization by means of the Taylor series expansion (22) may not reflect the behavior of the nonlinear residual $\mathcal{R}$ very well. As a consequence, solving the linear system (20) for the increment $\Delta u^{(m+1)}$ with high accuracy one may obtain a poor Newton update which deteriorates the nonlinear convergence behavior [42,43]. This phenomenon is typically known as *oversolving* [13].

A common practice is to choose the so-called forcing term $\eta^{(m)} \in [0,1)$ *a priori* and require the linear solver to compute the increment $\Delta u^{(m+1)}$ from the Newton equation (20) to a certain accuracy so that the following convergence criterion holds:

$$\|\mathcal{R}(u^{(m)}) + J^{(m)}\Delta u^{(m+1)}\| \le \eta^{(m)}\|\mathcal{R}(u^{(m)})\|. \tag{27}$$

Recall that the left-hand side of the above inequality is both the residual of the linear subproblem and the linearized model of $\mathcal{R}(u^{(m)} + \Delta u^{(m+1)})$ given by the first-order terms of the Taylor series expansion (22). Several strategies for choosing the forcing term in an adaptive fashion are proposed by Eisenstat and Walker in [12,13]. A viable choice is

$$\eta^{(m+1)} = \frac{\left| \|\mathcal{R}(u^{(m+1)})\| - \|\mathcal{R}(u^{(m)}) + J^{(m)}\Delta u^{(m+1)}\| \right|}{\|\mathcal{R}(u^{(m)})\|}, \qquad \text{where} \quad \eta^{(0)} \in [0,1) \tag{28}$$

which directly reflects the agreement between the nonlinear residual $\mathcal{R}$ and its linearized model (22) at the previous step. A *local* convergence theory of inexact Newton methods and a detailed discussion about the impact of forcing terms is given in [11].

It it well known, that this technique is prone to diverge for crude starting values. Due to this lack of convergence robustness, the use of some 'globalization' technique is mandatory. To this end, the computed solution increment needs to be relaxed by the factor $\omega^{(m+1)}$ so that the *sufficient decrease condition* holds on each Newton step [12]

$$\|\mathcal{R}(u^{(m)} + \omega^{(m+1)}\Delta u^{(m+1)})\| \le [1 - \xi(1 - \eta^{(m)})]\|\mathcal{R}(u^{(m)})\| \ , \tag{29}$$

where $\xi \in (0,1)$ represents the prescribed reduction tolerance. In practice, line search and trust region methods are frequently employed to compute an acceptable increment [34,39]. In our implementation we make use of simple backtracking [42] which computes $\omega^{(m+1)}$ as the minimizer of $\|\mathcal{R}(u^{(m)} + \omega^{(m+1)}\Delta u^{(m+1)})\|^2$ by means of quadratic interpolation.

4 Calculation of Jacobians

Now that we have discussed solution strategies for nonlinear problems in a general framework, let us consider the calculation of the Jacobian matrix so as to build up the preconditioner (25). The formal definition of $J^{(m)}$ requires the 'differentiation' of the modified transport operator $K^*(u)$ which consists of diffusive and antidiffusive contributions. Recall the fact that flux limiters frequently makes use of non-differentiable functions [23–25] so that no analytical expression for the Jacobian is available.

The use of an iterative method, such as BiCGSTAB or GMRES, for the solution of the linear system (20) only requires the computation of Jacobian-vector products which may be approximated by means of forward (or backward) differences

$$J^{(m)}v \simeq \pm\frac{1}{h}\left[\mathcal{R}(u^{(m)} \pm hv) - \mathcal{R}(u^{(m)})\right].$$

As an alternative to approximating the Jacobian $J^{(m)}$ times a vector v by means of a first-order Taylor series expansion, a second-order accurate counterpart is given by

$$J^{(m)}v \simeq \frac{1}{2h}\left[\mathcal{R}(u^{(m)} + hv) - \mathcal{R}(u^{(m)} - hv)\right] \tag{30}$$

which requires a double evaluation of the nonlinear residual. In practice, the performance of Newton's method is quite sensitive [19] to the size of the perturbation parameter h which should be sufficiently small. Following a simple strategy proposed by Nielsen *et al.* [36], the step size can be determined using the expression

$$h\|v\| = \sqrt{\epsilon}, \tag{31}$$

where ϵ denotes the machine precision. Some more sophisticated choices for the step size parameter h are given in a survey paper on Jacobian-free Newton-Krylov methods by Knoll and Keyes [22]. Another effective formula proposed by Pernice *et al.* and successfully used in the NITSOL package [40] for determining h is given by

$$h\|v\| = [(1+\|u\|)\epsilon]^{\frac{1}{p+1}}\,, \tag{32}$$

where p denotes the order of the employed finite difference formula. It is claimed to reduce the noise arising from the evaluation of the residual $\mathcal{R}$.

What makes such *matrix-free* approaches most attractive at first glance, is their Newton-like nonlinear convergence behavior without the costs of computing and storing the Jacobian explicitly. However, these advantages are not as overwhelming as one might immediately think. For finite element problems, the Jacobian matrix is very sparse and hence the savings in terms of memory usage are insignificant. The crucial point is, that there are only few preconditioners which can be applied without knowing the system matrix explicitly [10]. As a consequence, the solution process may converge poorly or even fail to converge at all. If approximate preconditioners are employed the use of sophisticated flexible GMRES [41] or GMRES-R Krylov subspace methods [45] is mandatory which call for some extra dense vector storage. Finally, the costs of evaluating the flux limiter in each iteration of the linear solver rapidly grow to an impractical amount.

With these observations in mind, the explicit formation of Jacobians gains more attraction, provided a sufficiently accurate approximation of $J^{(m)}$ can be computed at reasonable costs. Note, that the entries of the Jacobian matrix can be approximated by the formula

$$J_{ik}^{(m)} \simeq \pm\frac{1}{h}\left[\mathcal{R}_i(u^{(m)} \pm he^k) - \mathcal{R}_i(u^{(m)})\right], \tag{33}$$

where e^k denotes the k^{th} unit vector. The error is proportional to step length h which, adopting definition (32), is the same for all columns, i.e., $h = \sqrt{(1+\|u\|)\epsilon}$.

Moreover, a second order accurate approximation can be constructed as follows

$$J_{ik}^{(m)} \simeq \frac{1}{2h}\left[\mathcal{R}_i(u^{(m)} + he^k) - \mathcal{R}_i(u^{(m)} - he^k)\right]. \tag{34}$$

Active development in the field of numerical optimization has revealed a variety of *quasi-Newton methods* which try to recover the ultimate convergence of Newton's approach at a reduced effort. The underlying idea is to replace the costly computation of the Jacobian matrix in each Newton step by a cheaper update of the Newton operator. Such techniques have been mainly developed for optimization problems and, hence, most research has concentrated on symmetric positive definite operators. A promising rank one update for non-symmetric coefficient matrices is due to Broyden [6]. Unfortunately, in its original form, the update is applied to the inverse of the Jacobian which in general is a dense matrix that does not exploit the localized structure of matrices resulting from finite element discretizations. An interesting algorithm which is designed to maintain the sparsity pattern of the FEM matrix is given in the FIDAP Theory manual [15]. It comes at the cost of some extra memory which is required to store two vectors for each nonlinear iteration with dimensions equal to that of the solution vector.

4.1 Discrete transport operators

Algebraic flux correction schemes [23–30] are designed as 'black-box' post-processing tools which extract all required information from the matrix (structure) and make use of the solution values u in order to modify the right-hand side and the residual, respectively. With this observation in mind equations (33) and (34) may be used to devise a straightforward algorithm for assembling the Jacobian. In essence, each column of the matrix J can be constructed by taking the difference between the residuals evaluated at u and/or the 'perturbed' solutions $u \pm he^k$ and scale the result by h or $2h$, respectively. However, this approach is prohibitively expensive since it does not exploit the underlying sparsity pattern of the system matrix. A common practice in finite element methods is to assemble the Jacobian matrix element-by-element [9]. As we are about to see, the edge-based formulation of our AFC techniques can be utilized the construct J edge-by-edge.

Recall that the modified evolution operator $K^*(u)$ defined in (3) represents a nonlinear combination of the original high-order operator K and its low-order order counterpart L which can be recovered by varying α_{ij} between zero and unity. Therefore it suffices to devise an efficient algorithm for evaluating the derivative of the convective term

$$\frac{\partial}{\partial u}\left[K^*(u)u\right] = K'(u)u + [D'(u) + F'(u)]u + K^*(u). \tag{35}$$

Let us approximate each entry of the Jacobian matrix by divided differences (34)

$$J_{ik} \simeq \frac{1}{2h}\left\{\left[K^*(u+he^k)(u+he^k)\right]_i - \left[K^*(u-he^k)(u-he^k)\right]_i\right\} \tag{36}$$

and replace both matrix-vector products by the decomposition (8) so as to obtain

$$\begin{aligned} J_{ik} \;=\; & \sum_{j\neq i}\frac{k^*_{ij}(u+he^k)}{2h}(u_j + h\delta_{jk} - u_i - h\delta_{ik}) + (u_i + h\delta_{ik})\sum_j \frac{k^*_{ij}(u+he^k)}{2h} \\ - & \sum_{j\neq i}\frac{k^*_{ij}(u-he^k)}{2h}(u_j - h\delta_{jk} - u_i + h\delta_{ik}) - (u_i - h\delta_{ik})\sum_j \frac{k^*_{ij}(u-he^k)}{2h}, \end{aligned} \tag{37}$$

where δ_{ij} denotes the standard Kronecker delta symbol. Some tedious algebraic manipulations reveal the fact that the above expression can be cast into the form

$$J_{ik} = \sum_{j\neq i}\chi^*_{ij}(u_j - u_i) + u_i\sum_j \chi^*_{ij} + \mu^*_{ik}, \tag{38}$$

whereby the identity $\sum_j \delta_{jk}\mu^*_{ij} = \mu^*_{ik}$ following from the definition of the Kronecker symbol has been used for the third term. In the above equation, the auxiliary quantities

$$\chi^*_{ij} = \frac{k^*_{ij}(u+he^k) - k^*_{ij}(u-he^k)}{2h}, \qquad \mu^*_{ij} = \frac{k^*_{ij}(u+he^k) + k^*_{ij}(u-he^k)}{2} \tag{39}$$

stand for the central difference approximation of the nonlinear coefficient $k^*_{ij}(u)$ and the standard average of its perturbed counterparts, respectively. Interestingly enough, the structure of equation (38) largely resembles that of (8) except for the average term μ^*_{ik}.

In order to investigate the global matrix resulting from (38) in detail, let

$$\chi_{ij}^H = \frac{k_{ij}(u+he^k) - k_{ij}(u-he^k)}{2h}, \qquad \mu_{ij}^H = \frac{k_{ij}(u+he^k) + k_{ij}(u-he^k)}{2} \tag{40}$$

denote the pure Galerkin parts of the discrete Jacobian. For our purpose it makes sense to separate them from the coefficients defined in (39) such that

$$\chi_{ij}^* = \chi_{ij}^H + \chi_{ij}, \qquad \mu_{ij}^* = \mu_{ij}^H + \mu_{ij}. \tag{41}$$

The remaining quantities χ_{ij} and μ_{ij} represent the (anti-)diffusive contributions which are due to the flux limiter. For $j \neq i$ they are given by the following expressions

$$\begin{aligned} \chi_{ij} &= \frac{[1-\alpha_{ij}(u+he^k)]d_{ij}(u+he^k) - [1-\alpha_{ij}(u-he^k)]d_{ij}(u-he^k)}{2h}, \\ \mu_{ij} &= \frac{[1-\alpha_{ij}(u+he^k)]d_{ij}(u+he^k) + [1-\alpha_{ij}(u-he^k)]d_{ij}(u-he^k)}{2}. \end{aligned} \tag{42}$$

Due to the zero row sum property of discrete diffusion operators it follows from (9) and (10) that the corresponding diagonal entries can be computed from the off-diagonal ones

$$\chi_{ii} = -\sum_{j\neq i} \chi_{ij}, \qquad \mu_{ii} = -\sum_{j\neq i} \mu_{ij}. \tag{43}$$

In other words, the Jacobian matrix (38) consists of the standard high-order contribution and artificial (anti-)diffusion which is symmetric and thus cancels out in the reactive term. Interestingly enough, the only difference between the formally derived Jacobian (35) approximated by divided differences $K'(u) = \{\chi_{ij}^H\}$ and $D'(u) + F'(u) = \{\chi_{ij}\}$ and its numerically computed counterpart (38) consists in the treatment of the zeroth-order term. It is also possible to omit the averaging of $K^*(u)$ and use $\mu_{ij}^* := k_{ij}^*(u)$ instead.

Let us consider the special case that all correction factors α_{ij} are equal to unity. As a consequence, the quantities χ_{ij} and μ_{ij} defined in (42) vanish so that expression (38) yields the approximate Jacobian of the original high-order scheme $K(u)u = 0$

$$J_{ik} = \sum_{j\neq i} \chi_{ij}^H (u_j - u_i) + u_i \sum_j \chi_{ij}^H + \mu_{ik}^H. \tag{44}$$

In contrast, the approximate Jacobian for the local extremum diminishing counterpart $L(u)u = 0$ is recovered from (38) if all coefficients α_{ij} are set equal to zero. This is due to the fact that χ_{ij} and μ_{ij} as defined by (42) yield the divided differences of the artificial diffusion coefficient $d_{ij}(u)$ and its standard average, respectively.

We would like to point out, that the decomposition into edge contributions (38)/(44) *per se* provides an efficient alternative to the traditional element-by-element procedure [9] which is commonly used to assemble Jacobians arising from finite element discretizations.

So far, the conservation law (1) is supposed to be nonlinear so that the operator $K^*(u)$ depends on the unknown solution vector u due to both physical/natural and numerical nonlinearities. The latter one results from the application of algebraic flux correction and is still present in case the governing equation is linear. In this special situation, the derivative of the discrete high-order transport operator K vanishes so that $\chi_{ij}^H \equiv 0$ whereas its average yields $\mu_{ij}^H \equiv k_{ij}$. As a consequence, the coefficients (42) which represent the (anti-)diffusive contribution of the operator $K^*(u) = K + D + F(u)$ can be rewritten as

$$\begin{aligned} \chi_{ij} &= -\frac{\alpha_{ij}(u + he^k) - \alpha_{ij}(u - he^k)}{2h} d_{ij}, \\ \mu_{ij} &= \left[1 - \frac{\alpha_{ij}(u + he^k) + \alpha_{ij}(u - he^k)}{2}\right] d_{ij}, \end{aligned} \tag{45}$$

whereby the diffusive term d_{ij} no longer depends on the solution u. With these observations in mind, it is easy to verify that for a linear conservation law (1) the differentiation only affects the antidiffusive correction which is still nonlinear whereas the derivative of the linear transport operator K and that of the discrete diffusion term D are no longer present in the Jacobian matrix (38). As in the case of a nonlinear transport operator, the average of the zeroth-order antidiffusion (45) can also be replaced by $\mu_{ij} = [1 - \alpha_{ij}]d_{ij}$.

In short, both the physical nonlinearity present in the conservation law (1) and the one resulting from the application of algebraic flux correction techniques allow for a handy decomposition of the corresponding Jacobian matrix into edge contributions (38).

Some caution is in order concerning the edge orientation introduced in Section 2 which is indispensable for upwind-biased flux limiting techniques [23,25,30]. Recall that the edge $\vec{ij}$ is oriented so that $0 \leq k_{ij} + d_{ij} = l_{ij} \leq l_{ji}$ which implies that node i is located 'upwind' and corresponds to the row number of the eliminated negative coefficient.

For linear coefficients, the orientation can be determined once and for all at the beginning of the simulation. However, for nonlinear transport operators $K(u)$ the signs of $k_{ij}(u + he^k)$ and $k_{ij}(u - he^k)$ may be different from that of $k_{ij}(u)$ so that edge $\vec{ij}$ cannot be oriented for all three quantities simultaneously. Hence, it is advisable to replace the central difference approximation (39) of the coefficient χ_{ij}^* by its 'upwind-like' counterpart

$$\chi_{ij}^* = s_1 \frac{k_{ij}^*(u + he^k) - k_{ij}^*(u)}{h} + s_2 \frac{k_{ij}^*(u) - k_{ij}^*(u - he^k)}{h} \tag{46}$$

and modify the average term μ_{ij}^* accordingly. In the above equation, the coefficients s_1 and s_2 are used to switch between backward, forward and central differences, thus

$$\begin{aligned} &s_1 = 1 \ \wedge \ s_2 = 0 \quad \text{if} \quad k_{ij}(u + he^k)\, k_{ij}(u) > 0 \ \wedge \ k_{ij}(u - he^k)\, k_{ij}(u) < 0, \\ &s_1 = 0 \ \wedge \ s_2 = 1 \quad \text{if} \quad k_{ij}(u + he^k)\, k_{ij}(u) < 0 \ \wedge \ k_{ij}(u - he^k)\, k_{ij}(u) > 0, \\ &s_1 = \tfrac{1}{2} \ \wedge \ s_2 = \tfrac{1}{2} \quad \text{otherwise.} \end{aligned} \tag{47}$$

The situation that the sign of both perturbed quantities $k_{ij}(u \pm he^k)$ differs from that of $k_{ij}(u)$ for sufficiently small parameters h implies that the discrete transport operator K is discontinuous which is very unlikely to be the case in practice. In all other situations, the upwind difference (46) guarantees that the orientation of edge $\vec{ij}$ can be adopted from the unperturbed coefficients k_{ij} as described in Section 2.

4.2 Sparsity pattern

Let us proceed to algebraic flux correction schemes [23–25,27–30] and consider the limited antidiffusive contributions (42)–(43) which need to be built into the Jacobian matrix (38). It is quite obvious that the perturbation of the nodal solution value u_i affects the diffusion coefficients d_{ij} as well as the corresponding correction factors α_{ij}. As we are about to see, the complex interplay of nonlinear terms requires some minor modifications of the underlying matrix sparsity pattern which will be analyzed step-by-step.

To begin with, consider an unstructured mesh consisting of conforming triangles and/or quadrilaterals as shown in Figure 1. For linear and bilinear finite elements, let the *stencil* of node k be defined as the set of neighboring vertices that share an element with it, i.e., $\mathcal{S}_k = \{l : \exists \vec{kl}\}$. Note that the orientation convention introduced in Section 2 is neglected for the time being. The sparsity pattern of the global matrix can be easily constructed by considering all sets $\mathcal{S}_k$. In Figure 1, the dashed lines starting at node k point to the column numbers of nonzero entries in the k^{th} row of the finite element matrix. Likewise, for each of these neighbors the corresponding rows exhibit a nonzero entry in the k^{th} column due to the symmetry of the undirected connectivity graph.

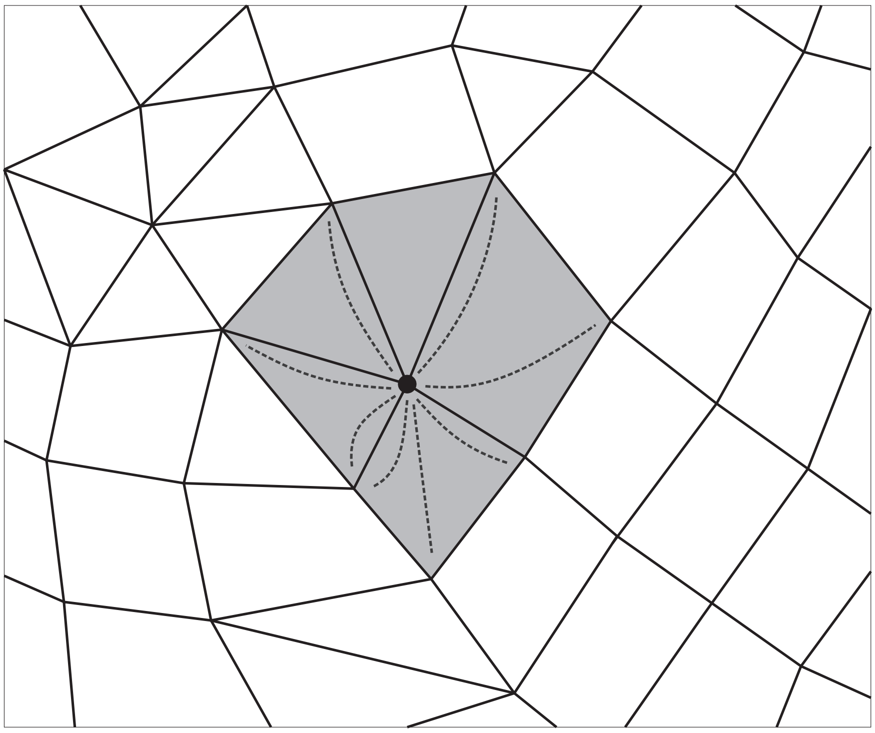


Figure 1: Influence of the perturbation of the k^{th} nodal value.

From the definition of the stencil $\mathcal{S}_k$ it follows, that the perturbation of the solution vector u at vertex k only affects the quantities $P_l^\pm$ and $Q_l^\pm$ defined in (12)/(13) if $l = k$ or $l \in \mathcal{S}_k$. As a consequence, the multipliers $R_l^\pm$ need to be recomputed from formula (14). For all other nodes $i \neq k$ which are not comprised in the set $\mathcal{S}_k$, the nodal quantities $P_i^\pm$, $Q_i^\pm$ and $R_i^\pm$ coincide with their unperturbed counterparts which are already known.

Recall that the correction factor α_{ij} is taken as a combination of $R_i^{\pm}$ and $R_j^{\mp}$ depending on the sign of the antidiffusive flux (15). Obviously, local perturbations of the nodal solution value u_k are quite likely to affect the magnitude of α_{ij} if *at least* one endpoint of the edge $\vec{ij}$ belongs to $\mathcal{S}_k$, i.e., the set of neighbors directly connected to the perturbed vertex k. In other words, the impact of 'joggling' the solution value u_k may propagate along paths until some node $i \notin \mathcal{S}_k$ is reached for which there exists an edge $\vec{ij}$ such that $j \in \mathcal{S}_k$. This observation suggests the definition of an extended list of neighboring nodes

$$\tilde{\mathcal{S}}_k = \bigcup_{l \in \mathcal{S}_k} \mathcal{S}_l \tag{48}$$

so as to reflect the new connectivity pattern of the resulting matrix. The structure of nonzero entries in the k^{th} column of the Jacobian is illustrated in Figure 2. Those edges $\vec{ij}$ for which both the diffusion coefficient d_{ij} and the limiting factor α_{ij} may exhibit a change in magnitude due to the perturbation of the solution at node k are marked by dashed paths. Moreover, dotted ones are used to indicate edges which indirectly depend on the perturbed solution value u_k. Even though the corresponding diffusion coefficients d_{ij} coincide with their unperturbed counterparts, the net antidiffusive contributions (42) may persist due to different magnitudes of the multipliers $\alpha_{ij} = \alpha_{ij}(u \pm he^k)$.

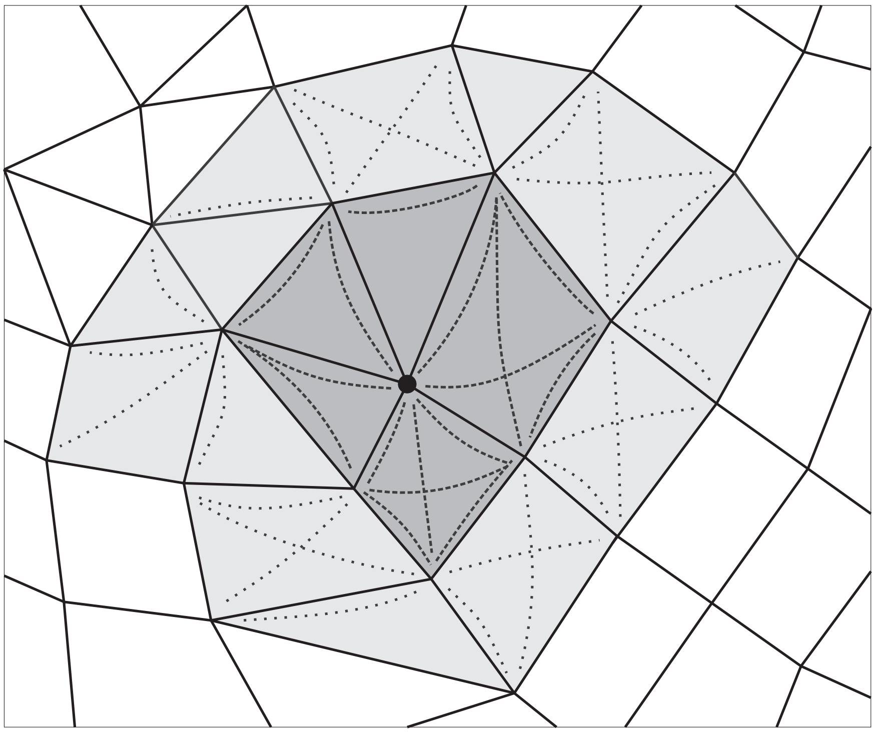


Figure 2: Structure of the k^{th} column of the Jacobian.

Interestingly enough, the extended sparsity pattern of the approximate Jacobian resembles that of the edge-oriented FEM stabilization technique [7, 8]. A detailed description of the edge-oriented data structure for non-conforming bilinear finite elements is given in [37]. The presented storage algorithm can be carried over to conforming (bi-)linear finite elements but care must be taken to avoid duplicate entries in the rows of the resulting finite element matrix. Let us briefly investigate some ideas from classical graph theory and devise an alternative storage algorithm directly tailored to conforming elements.

Let $A = \{a_{ij}\}$ denote the adjacency matrix which represents the undirected connectivity graph of the finite element matrix. The matrix coefficients are given by

$$a_{ij} \in \{0,1\} \quad : \quad a_{ij} = 1 \Leftrightarrow \exists\, \vec{ij}, \tag{49}$$

that is, a_{ij} does not vanish if nodes i and j share the same element. In other words, there exists a path of length one connecting vertices i and j. Furthermore, let $Z = A^2$ be the product of A multiplied by itself. Then $z_{ij} > 0$ if and only if there exists a path of length two connecting nodes i and j. This can be easily verified by recalling that

$$z_{ij} = \sum_k a_{ik}\, a_{kj} > 0 \quad \Leftrightarrow \quad \exists k \,:\, a_{ik} = 1 \wedge a_{kj} = 1, \tag{50}$$

that is, vertex j can be reached from i and vice versa via node k passing two edges. As a result, a standard algorithm [4] for sparse matrix multiplications can be employed to assemble Z which can be used to construct the sparsity pattern of the Jacobian matrix.

5 Numerical examples

In order to demonstrate the ideas presented in this paper, we apply our discrete algebraic Newton strategy to a number of two-dimensional benchmark problems and compare the nonlinear convergence behavior to that of the classical defect correction procedure.

To this end, consider the model problem (1) and define the flux function **f** so that

$$\nabla \cdot \mathbf{f}(u) := \frac{\partial f(u)}{\partial x} + \frac{\partial u}{\partial t} = 0 \tag{51}$$

which is solved in the space-time domain $\Omega = I \times (0,T)$. Interestingly enough, the above equation can be interpreted as a one-dimensional time-dependent conservation law defined in the spatial interval I for which the final solution is obtained at time T. Note that the reformulation as a two-dimensional steady state problem corresponds to computing the solution for all time levels simultaneously instead of doing it step-by-step. The stationary conservation law (51) is complemented by suitable boundary conditions which need to be prescribed at the 'inlet' of the space-time domain Ω. Moreover, the initial data can be chosen arbitrarily since they do not affect the converged steady state solution.

It is noteworthy that upwind-biased flux limiting (11)–(15) can be directly applied to problems of the form (51) so as to end up with the nonlinear algebraic equation system (2). As an alternative, pseudo-time stepping (17) can be adopted in order to march the solution into the stationary limit. As pointed out in Section 2 the upper/lower bounds (13) used by the flux limiter are independent of the employed pseudo-time step. Thus we can benefit from the unconditional stability of the backward Euler method and use large values for the 'relaxation parameter' $\Delta\tau$ to reduce the computational costs.

5.1 Convection in space-time

As a first test case, set $f(u) = vu$ in equation (51) so that the problem reads

$$v\frac{\partial u}{\partial x} + \frac{\partial u}{\partial t} = 0. \tag{52}$$

Moreover, let us prescribe discontinuous Dirichlet boundary conditions at the 'inlet' of the space-time domain $\Omega = (0,1) \times (0,0.5)$ which are given by the following relations

$$u(x,t) = \begin{cases} 1 & \text{if} \quad |x-0.2| \leq 0.1 \quad \wedge \quad t=0, \\ 0 & \text{if} \quad |x-0.2| > 0.1 \quad \wedge \quad t=0 \\ & \text{or} \qquad\qquad x=0 \quad \wedge \quad 0 \leq t \leq 0.5. \end{cases} \tag{53}$$

This two-dimensional steady state problem is discretized by 100 × 50 bilinear finite elements on a uniform mesh with spacing $\Delta x = \Delta t = 10^{-2}$. For a constant velocity $v = 1$, this corresponds to the Courant number $\nu = 1$ in the associated transient convection equation in one space dimensions. The implicit backward Euler method is used to march the solution to the steady state which is reached once the relative changes of the global profile drops below 10^{-6}. The numerical result obtained using algebraic flux correction (12)–(15) is presented in Figure 3. The discontinuous data at the inflow boundary ($t = 0$) is shown in the background while the solution at the outflow appears in the front.

Table 1 presents the average number of iterations for the different solution strategies, i.e., the number of nonlinear / linear steps required to process one pseudo-time step / nonlinear step, respectively. The nonlinear solver is supposed to gain at least four digits per outer iteration. For the solution of the linear problems, the standard BiCGSTAB method is employed whereby the tolerance is determined from the forcing term (28). In case the discrete Newton method is applied, the monotone low-order operator (26) serves as a preconditioner. For the defect correction procedure no preconditioning is performed and a constant value of 10^{-12} is adopted for the tolerance of the linear solver.

Due to the fact that the problem at hand is linear, both solution strategies exhibit the same convergence behavior for discrete upwinding. As soon as some nonlinearity is engendered by the application of algebraic flux correction the number of nonlinear steps increases significantly for the standard defect correction approach. In contrast, a moderate number of $4-5$ nonlinear iterations suffices for the discrete Newton strategy.

Let us increase the grid size of the temporal dimension, i.e., $\Delta t = 2 \cdot 10^{-2}$, which corresponds to doubling the Courant number $\nu = 2$ of the associated transient problem. Interestingly enough, the number of Newton iterations is nearly the same as before whereas the nonlinear convergence behavior of the fixed-point defect correction approach deteriorates noticeably. In addition, the linear subproblems are less well-behaved so that more BiCGSTAB steps need to be performed in each nonlinear iteration.

Our second test problem is a slightly modified version [23] of the one used in [47]. The linear convection equation (52) is solved for the following continuous boundary conditions

$$u(x,t) = \begin{cases} \sqrt{1-\left(\frac{x-0.2}{0.15}\right)^2} & \text{if} \quad |x-0.2| \leq 0.15 \quad \wedge \quad t=0, \\ 0 & \text{if} \quad |x-0.2| > 0.15 \quad \wedge \quad t=0 \\ & \text{or} \qquad\qquad x=0 \quad \wedge \quad 0 \leq t \leq 0.5. \end{cases} \tag{54}$$

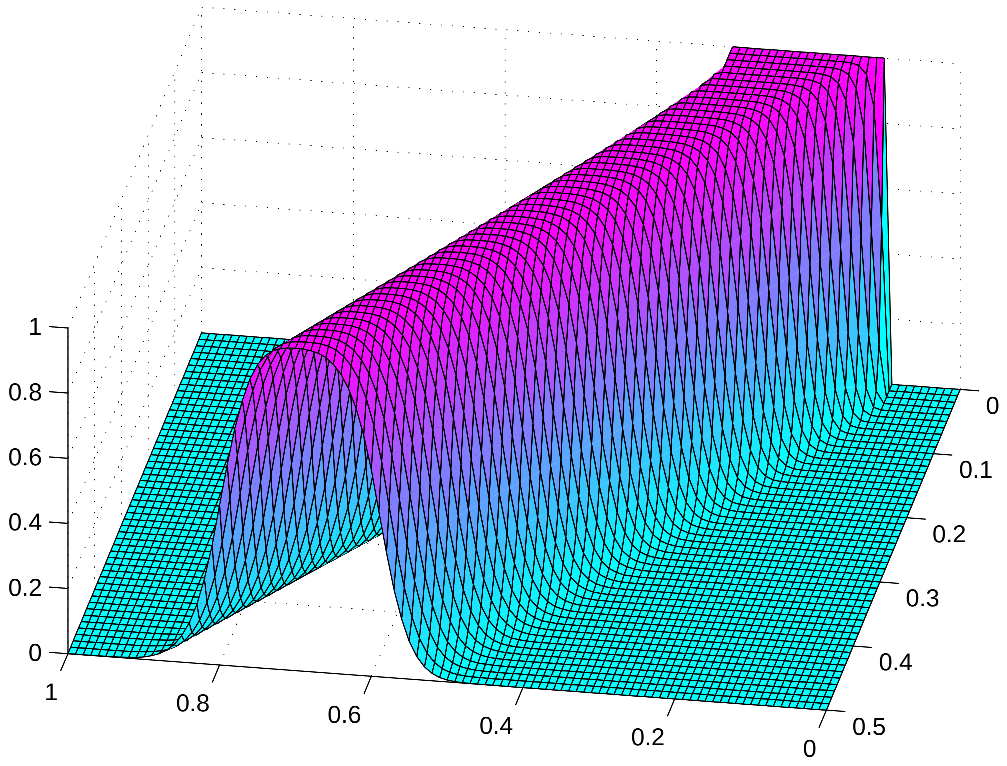


Figure 3: Convection in space-time: square wave.

	Newton's method		defect correction	
	nonlinear iter.	linear iter.	nonlinear iter.	linear iter.
	$\nu = 1, \quad \Delta x = 10^{-2}, \quad \Delta t = 10^{-2}$			
low-order	1.00	15.37	1.00	15.37
spatial AFC	4.49	14.55	13.48	19.00
space-time AFC	3.59	7.26	14.69	20.39
	$\nu = 2, \quad \Delta x = 10^{-2}, \quad \Delta t = 2 \cdot 10^{-2}$			
low-order	1.00	30.63	1.00	30.63
spatial AFC	4.91	23.97	20.85	30.97
space-time AFC	3.60	8.66	20.11	32.35

Table 1: Convection in space-time: square wave.

whereby all other parameters are adopted from the previous example. Figure 4 depicts the converged steady state solution computed on a uniform mesh with $\Delta x = \Delta t = 10^{-2}$ making use of the fully implicit backward Euler pseudo-time stepping approach.

The average number of iterations for the different solvers are shown in Table 2. As in our previous example, the newly proposed discrete Newton strategy is very well capable of treating the nonlinearity stemming from the application of the flux limiter at moderate costs. In contrast, the standard defect correction approach suffers from a drastic increase of iteration steps once the discrete algebraic system becomes nonlinear. If the number of grid points utilized in the $t-$direction is halved a significant increase of nonlinear iterations can be observed for the defect correction scheme. In contrast, an average of $4-5$ nonlinear steps per outer iteration are sufficient for the algebraic Newton approach.

5.2 Burgers' equation

Let us proceed to *nonlinear* problems and consider the solution-dependent flux function $f(u) = \frac{1}{2}u^2$ so that the conservation law (51) yields the inviscid Burgers' equation

$$u\frac{\partial u}{\partial x} + \frac{\partial u}{\partial t} = 0. \tag{55}$$

As a first benchmark, let $T = 0.5$ be the right endpoint of the time domain and prescribe the following boundary conditions at the 'inlet' of $\Omega = (0,1) \times (0,0.5)$

$$u(x,t) = \begin{cases} 1 & \text{if} \quad 0 \leq x < 0.4 \quad \wedge \quad t = 0, \\ \frac{1}{2} & \text{if} \quad 0.4 \leq x \leq 0.8 \quad \wedge \quad t = 0, \\ 0 & \text{if} \quad 0.8 < x \leq 1 \quad \wedge \quad t = 0 \\ & \text{or} \qquad x = 0 \quad \wedge \quad 0 \leq t \leq 0.5. \end{cases} \tag{56}$$

For the numerical results shown in Figure 5 a uniform mesh with spacing $\Delta x = \Delta t = 10^{-2}$ is employed. This corresponds to the unit Courant number for the associated transient Burgers' equation in one space dimension. Figure 5 (a) displays the approximate solution computed by the low-order method. It is free of nonphysical oscillations but suffers from smearing due to excessive artificial diffusion. Neglecting the temporal antidiffusion and applying algebraic flux correction only in space yields the solution profile depicted in Figure 5 (b). The resolution of discontinuities is improved but the rarefaction wave gets overly flattened. In contrast, the numerical solution presented in Figure 5 (c) results from the application of algebraic flux correction both in time and space. As a result, the two shock waves are resolved with high accuracy and the smooth transition along the rarefaction fan is captured precisely. The one-dimensional profiles sampled along the cutline $t = 0.5$ are shown in Figure 5 (d). The beneficial impact of compensating antidiffusion applied to the overly diffusive low-order solution is clearly seen.

For the discrete Newton method, an average of $2-4$ nonlinear steps per outer iteration suffices for all three discretizations (c.f. Table 3). This remains valid even if the mesh width in the time direction is doubled. Interestingly enough, the monotone low-order operator (26) constitutes a viable preconditioner for the linear subproblems which need to be solved in each Newton step. For $\nu = 1$, the number of linear iterations which

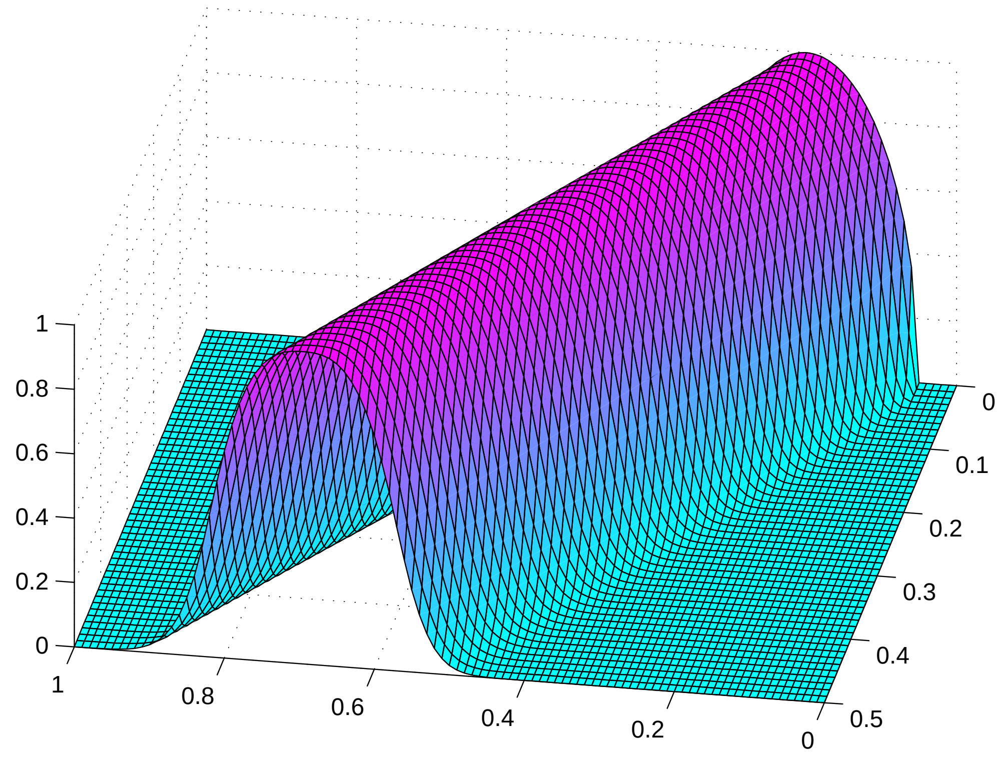
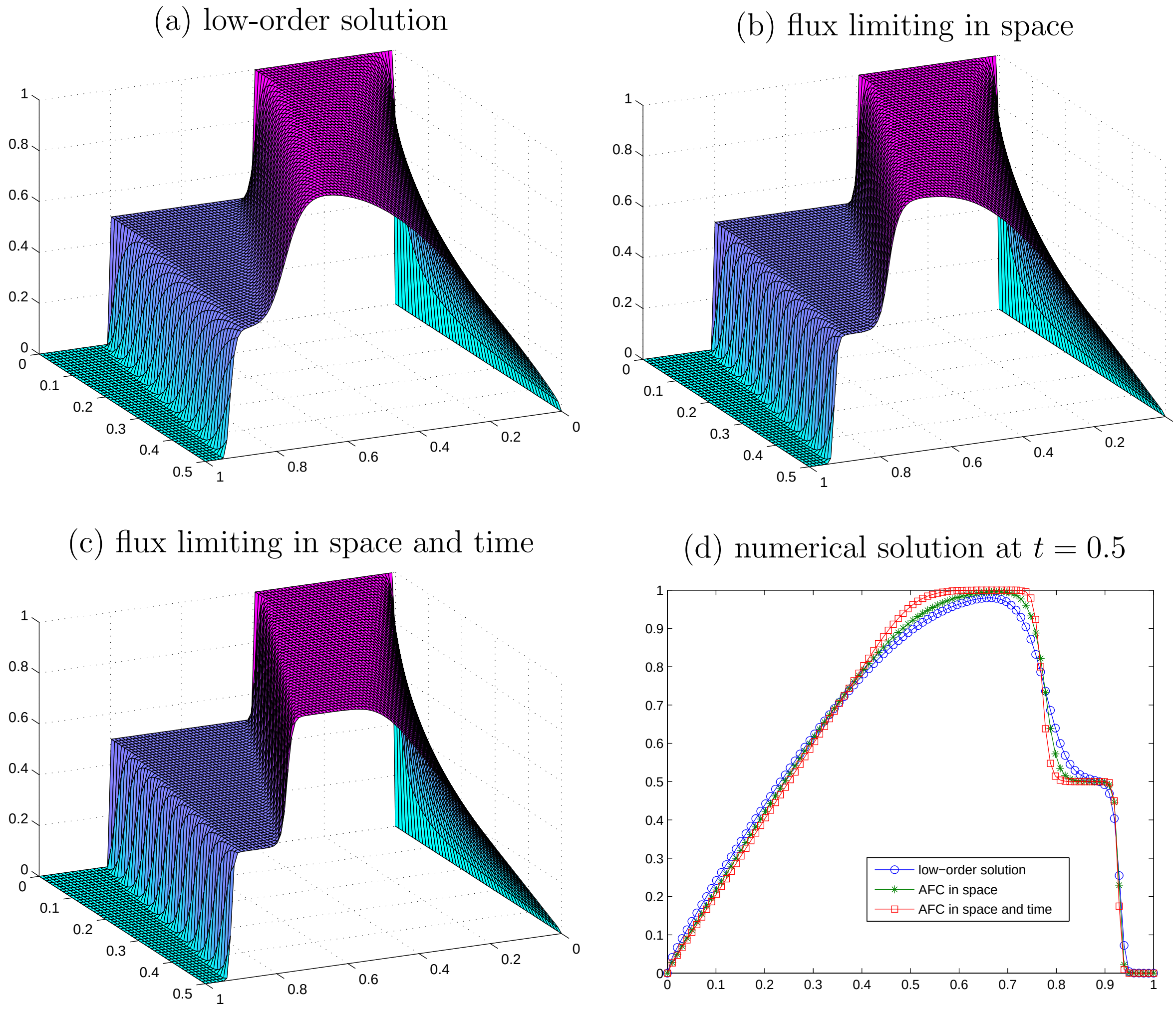


Figure 4: Convection in space-time: semi-ellipse.

	Newton's method		defect correction	
	nonlinear iter.	linear iter.	nonlinear iter.	linear iter.
	$\nu = 1, \quad \Delta x = 10^{-2}, \quad \Delta t = 10^{-2}$			
low-order	1.00	16.72	1.00	16.72
spatial AFC	4.21	14.38	14.27	19.47
space-time AFC	3.61	7.61	14.90	20.54
	$\nu = 2, \quad \Delta x = 10^{-2}, \quad \Delta t = 2 \cdot 10^{-2}$			
low-order	1.00	29.42	1.00	29.42
spatial AFC	4.68	24.03	21.02	30.79
space-time AFC	3.50	9.13	20.34	31.88

Table 2: Convection in space-time: semi-ellipse.


(a) low-order solution

(b) flux limiting in space

(c) flux limiting in space and time

(d) numerical solution at $t = 0.5$

Figure 5: Burgers' equation: numerical solutions.

are required to solve the linear system (20) if no preconditioner is employed are given in brackets. Note that without preconditioning, the BiCGSTAB solver is unable to solve the linear subproblems for the Courant number $\nu = 2$. Concerning the standard defect correction procedure the nonlinear convergence behavior deteriorates noticeably if the nonlinearity becomes stronger due to additional antidiffusive terms. The same applies if the mesh size used to discretize the time direction is increased.

As the final benchmark consider the inviscid Burgers' equation (55) in the space-time domain $\Omega = (0, 1)^2$ and prescribe the following Dirichlet boundary conditions

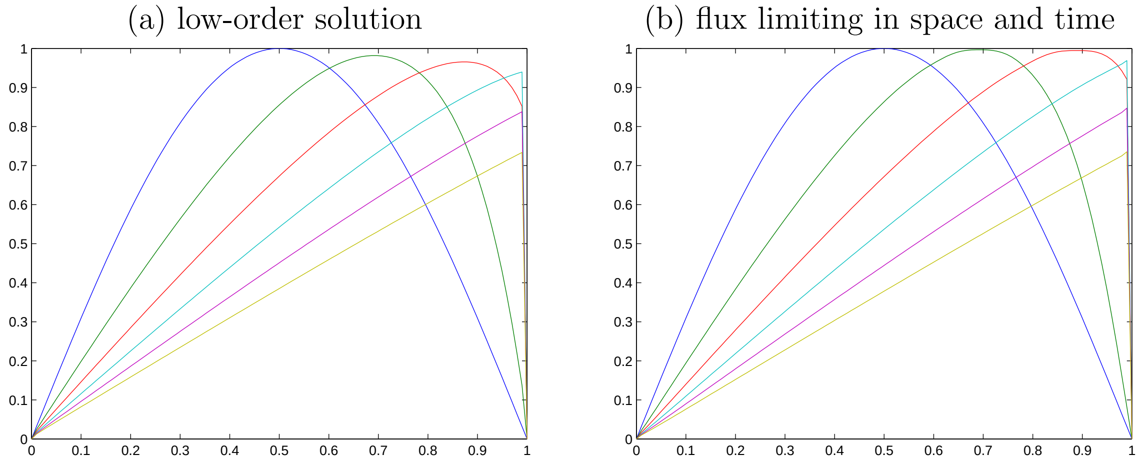
$$u(x,t) = \begin{cases} \sin(\pi x) & \text{if} \quad 0 \leq x \leq 1 \qquad\quad \wedge \quad t = 0, \\ 0 & \text{if} \quad (x = 0 \vee x = 1) \quad \wedge \quad 0 \leq t \leq 1 \end{cases} \tag{57}$$

which give rise to the formation of a boundary layer at $x = 1$. To begin with, a uniform mesh of 101×101 grid points is employed such that $\Delta x = \Delta t = 10^{-2}$. This corresponds to $\nu = 1$ for the associated time-dependent Burgers' equation in one space dimension.

	Newton's method		defect correction	
	nonlinear iter.	linear iter.	nonlinear iter.	linear iter.
	$\nu = 1, \quad \Delta x = 10^{-2}, \quad \Delta t = 10^{-2}$			
low-order	1.67	2.05 (5.54)	4.71	15.01
spatial AFC	2.77	8.97 (13.63)	10.73	18.27
space-time AFC	3.18	6.55 (9.70)	13.83	13.24
	$\nu = 2, \quad \Delta x = 10^{-2}, \quad \Delta t = 2 \cdot 10^{-2}$			
low-order	1.69	2.35 (n.a.)	5.85	24.19
spatial AFC	4.04	16.07 (n.a.)	20.01	30.11
space-time AFC	3.02	8.48 (n.a.)	61.85	23.58

Table 3: Burgers' equation: nonlinear solution behavior.

Figure 6 shows the approximate solutions at different 'time instants', that is, the two-dimensional solution profile is sampled along the cutlines $t \in \{0, 0.2, 0.4, 0.6, 0.8, 1.0\}$. Due to the excessive artificial diffusion engendered by the low-order scheme, the solution profile gets smeared which results in the early flattening of the prescribed sine wave. In contrast, the correct amplitude of the solution profile is recovered by our algebraic flux correction scheme. Moreover, a crisp resolution of the peak near the boundary layer at $x = 1$ is obtained which is completely free of nonphysical over- and undershoots.


(a) low-order solution

(b) flux limiting in space and time

Figure 6: Burgers' equation: numerical solutions at $t = 0$, 0.2, 0.4, 0.6, 0.8, 1.0.

It follows from Table 4 that an average of $2-3$ Newton-type subiterations are sufficient to resolve the nonlinearity in each pseudo-time step. Moreover, the nonlinear convergence behavior is hardly affected by the flux limiter. In contrast, a significant slow-down of the fixed-point iteration can be observed for the defect correction approach if the nonlinearity becomes stronger. The drawbacks of the latter solution algorithm as compared to the algebraic Newton strategy become even more obvious if the Courant number is increased.

	Newton's method		defect correction	
	nonlinear iter.	linear iter.	nonlinear iter.	linear iter.
	$\nu = 1, \quad \Delta x = 10^{-2}, \quad \Delta t = 10^{-2}$			
low-order	1.42	1.68	2.56	9.48
spatial AFC	1.65	5.59	8.81	15.66
space-time AFC	2.50	5.68	10.85	15.64
	$\nu = 2, \quad \Delta x = 10^{-2}, \quad \Delta t = 2 \cdot 10^{-2}$			
low-order	1.43	2.16	3.33	14.61
spatial AFC	2.06	7.11	14.90	25.84
space-time AFC	2.80	7.42	15.81	22.29

Table 4: Burgers' equation: nonlinear solution behavior.

6 Conclusions

In this paper, we demonstrated that implicit high-resolution discretization schemes and efficient solution strategies can be successfully combined. The concept of the algebraic flux correction paradigm [23–25, 30] was revisited. The application of flux limiters inevitably led to discrete algebraic systems of equations which called for strong nonlinear solution strategies. Due to the lack of a continuous counterpart of the discrete problem at hand Newton-type schemes were derived on the fully discrete level. The Jacobian matrix was approximated by means of divided differences. Each entry of the Newton matrix was rewritten in terms of edge contributions so that an efficient assembly of the Jacobian could be performed edge-by-edge. This idea turned out to be an interesting alternative to the elementwise evaluation of the Jacobian which is traditionally employed in finite element context [9]. We identified the individual sources of nonlinearities, i.e., the physical one present in the conservation law and the numerical one engendered by the flux limiter, and analyzed their contributions to the global Jacobian. The construction of the Newton matrix for upwind-biased flux limiting schemes was addressed in detail. The indirect coupling of solution values at non-neighboring vertices via the nodal correction factors made it necessary to extend the sparsity pattern of the underlying finite element matrix. Due to the similarities to edge-oriented stabilization techniques existing storage algorithms could be adopted with slight modifications. An alternative strategy for generating the connectivity pattern was derived based on classical graph theory.

High-resolution schemes based on *algebraic* flux correction techniques can be equipped with efficient nonlinear solution strategies of Newton-type whereby the Jacobian matrix is constructed by applying divided differences edge-by-edge. This concept can be extended to flux limiters especially designed for transient flows such as the semi-implicit FEM-FCT limiter proposed in [24]. Even in case the time step needs to be moderately small due to physical reasons the use of (semi-)implicit time-stepping schemes may be favorable due to less severe stability restrictions. The benefits of implicit discretization schemes become even more obvious if local grid refinement is employed which would require impractically small time steps for explicit methods. A promising direction for further research is the application of the discrete Newton method to the Euler and Navier-Stokes equations for which algebraic flux correction can be performed as explained in [26].

References

[1] M. Arioli, D. Loghin, and A.J. Wathen. Stopping criteria for iterations in finite element methods. *Numer. Math.*, 99(3):381–410, 2005.

[2] M. Arioli, E. Noulard, and A. Russo. Vector stopping criteria for iterative methods: applications to PDE's. *Calcolo*, 38:97–112, 2001.

[3] P. Arminjon and A. Dervieux. Construction of TVD-like artificial viscosities on two-dimensional arbitrary FEM grids. *J. Comput. Phys.*, 106(1):176–198, 1993.

[4] E.B. Bank and C.C. Douglas. SMMP: Sparse matrix multiplication package. *Adv. Comput. Math.*, 1:127–137, 1993.

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

[6] C.G. Broyden. A class of methods for solving nonlinear simultaneous equations. *Math. Comp.*, 19:577–593, 1965.

[7] E. Burman and A. Ern. Stabilized galerkin approximation of convection-diffusion-reaction equations: discrete maximum principle and convergence. *Math. Comp.*, 74(252):1637–1652, 2005.

[8] E. Burman and P. Hansbo. Edge stabilization for galerkin approximations of convection-diffusion-reaction problems. *Comput. Methods Appl. Mech. Engrg.*, 193(15–16):1437–1453, 2005.

[9] P.J. Capon and P.K. Jimack. An inexact Newton method for systems arising from the finite element method. *Appl. Math. Lett.*, 10(3):9–12, 1997.

[10] R. Choquet. A matrix-free preconditioner applied to CFD. Technical Report 2605, INRIA, 1995.

[11] R.S. Dembo, S.C. Eisenstat, and T. Steihaug. Inexact Newton methods. *SIAM J. Numer. Anal.*, 19(2):400–408, 1982.

[12] S.C. Eisenstat and H.F. Walker. Globally convergent inexact Newton methods. *SIAM J. Optim.*, 4(2):393–422, 1994.

[13] S.C. Eisenstat and H.F. Walker. Choosing the forcing term in an inexact Newton method. *SIAM J. Sci. Comput.*, 17:16–32, 1996.

[14] J.H. Ferziger and M. Perić. *Computational Methods for Fluid Dynamics.* Springer, Berlin, 1996.

[15] *FIDAP 8: Theory Manual.* http://www.fluent.com, 1998.

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

[17] A. Harten. On a classs of high resolution total-variation-stable finite-difference-schemes. *SIAM J. Numer. Anal.*, 21:1–23, 1984.

[18] P.W. Hemker and B. Koren. Defect correction and nonlinear multigrid for steady Euler equations. Technical report, Centre for Mathematics and Computer Science, Amsterdam (Netherlands), 1988.

[19] J. Hron, A. Ouazzi, and S. Turek. A computational comparison of two FEM solvers for nonlinear incompressible flow. Technical Report 228, University of Dortmund, 2003.

[20] A. Jameson. Computational algorithms for aerodynamic analysis and design. *Appl. Numer. Math.*, 13(5):383–422, 1993.

[21] 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. J. Comput. Fluid Dyn.*, 4:171–218, 1995.

[22] D.A. Knoll and D.E. Keyes. Jacobian-free Newton-Krylov methods: a survey of approaches and applications. *J. Comput. Phys.*, 193:357–397, 2004.

[23] D. Kuzmin. On the design of general-purpose flux limiters for finite element schemes. I. Scalar convection. Technical Report 299, University of Dortmund, 2005.

[24] D. Kuzmin and D. Kourounis. A semi-implicit FEM-FCT algorithm for efficient treatment of time-dependent problems. Technical Report 302, University of Dortmund, 2005.

[25] D. Kuzmin and M. Möller. Algebraic flux correction I. Scalar conservation laws. In D. Kuzmin, R. Löhner, and S. Turek, editors, *Flux-Corrected Transport, Principles, Algorithms, and Applications*, pages 155–206. Springer, Germany, 2005.

[26] D. Kuzmin and M. Möller. Algebraic flux correction II. Compressible Euler equations. In D. Kuzmin, R. Löhner, and S. Turek, editors, *Flux-Corrected Transport, Principles, Algorithms, and Applications*, pages 207–250. Springer, Germany, 2005.

[27] D. Kuzmin, M. Möller, and S. Turek. Multidimensional FEM-FCT schemes for arbitrary time-stepping. *Internat. J. Numer. Methods Fluids*, 42(3):265–295, 2003.

[28] D. Kuzmin, M. Möller, and S. Turek. High-resolution FEM-FCT schemes for multidimensional conservation laws. *Comput. Methods Appl. Mech. Engrg.*, 193(45–47):4915–4946, 2004.

[29] D. Kuzmin and S. Turek. Flux correction tools for finite elements. *J. Comput. Phys.*, 175(2):525–558, 2002.

[30] D. Kuzmin and S. Turek. High-resolution FEM-TVD schemes based on a fully multidimensional flux limiter. *J. Comput. Phys.*, 198(1):131–158, 2004.

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

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

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

[34] J.J. Moré and D.J. Thuente. Line search algorithms with guaranteed sufficient decrease. *ACM Trans. Math. Software*, 20:286–307, 1984.

[35] W. Mulder and B.V. Leer. Experiments with implicit upwind methods for the Euler equations. *Internat. J. Numer. Methods Engrg.*, 59:232–246, 1985.

[36] E.J. Nielsen, W.K. Anderson, R.W. Walters, and D.E. Keyes. Application of Newton-Krylov methodology to a three-dimensional unstructured Euler code. *AIAA 95-0221*, 1995.

[37] A. Ouazzi and S. Turek. Unified edge-oriented stabilization of nonconforming finite element methods for incompressible flow problems. Technical Report 284, University of Dortmund, 2005.

[38] S.V. Patankar. *Numerical Heat Transfer and Fluid Flow.* McGrawHill, 1980.

[39] R. P. Pawlowski, J.N. Shadid, J.P Simonis, and H.F. Walker. Globalization techniques for Newton-Krylov methods and applications to the fully-coupled solution of the Navier-Stokes equations. Report SAND2004-1777, Sandia National Laboratories, 2004.

[40] M. Pernice and H.F. Walker. NITSOL: a Newton iterative solver for nonlinear systems. *SIAM J. Sci. Comput.*, 19:302–318, 1998.

[41] Y. Saad. A flexible inner-outer preconditioned GMRES algorithm. *SIAM J. Sci. Comput.*, 14:461–469, 1993.

[42] J.N. Shadid, R.S. Tuminaro, and H.F. Walker. An inexact Newton method for fully coupled solution of the Navier-Stokes equations with heat and mass transport. *J. Comput. Phys.*, 137:155–185, 1997.

[43] R.S. Tuminaro, H.F. Walker, and J.N. Shadid. On backtracking failure in Newton-GMRES methods with a demonstration for the Navier-Stokes equations. *J. Comput. Phys.*, 180:549–558, 2002.

[44] S. Turek. *Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approach.* Number 6 in Lecture Notes in Computational Science and Engineering. Springer, 1999.

[45] H.A. van der Vorst and C. Vuik. A comparison of some GMRES-like methods. *Linear Algebra Appl.*, 160:131–162, 1992.

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

[47] S.T. Zalesak. The design of flux-corrected transport (FCT) algorithms for structured grids. In D. Kuzmin, R. Löhner, and S. Turek, editors, *Flux-Corrected Transport: Principles, Algorithms, and Applications*, pages 29–78. Springer, 2005.