INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2008; **57**:761–792
Published online 13 November 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1654

Implicit FEM-FCT algorithms and discrete Newton methods for transient convection problems

M. Möller[1,*,†], D. Kuzmin[1] and D. Kourounis[2]

[1]*Institute of Applied Mathematics (LS III), University of Dortmund, Vogelpothsweg 87, D-44227 Dortmund, Germany*
[2]*Department of Material Science and Engineering, University of Ioannina, Ioannina, Greece*

SUMMARY

A new generalization of the flux-corrected transport (FCT) methodology to implicit finite element discretizations is proposed. The underlying high-order scheme is supposed to be unconditionally stable and produce time-accurate solutions to evolutionary convection problems. Its nonoscillatory low-order counterpart is constructed by means of mass lumping followed by elimination of negative off-diagonal entries from the discrete transport operator. The raw antidiffusive fluxes, which represent the difference between the high- and low-order schemes, are updated and limited within an outer fixed-point iteration. The upper bound for the magnitude of each antidiffusive flux is evaluated using a single sweep of the multidimensional FCT limiter at the first outer iteration. This semi-implicit limiting strategy makes it possible to enforce the positivity constraint in a very robust and efficient manner. Moreover, the computation of an intermediate low-order solution can be avoided. The nonlinear algebraic systems are solved either by a standard defect correction scheme or by means of a discrete Newton approach, whereby the approximate Jacobian matrix is assembled edge by edge. Numerical examples are presented for two-dimensional benchmark problems discretized by the standard Galerkin finite element method combined with the Crank–Nicolson time stepping. Copyright © 2007 John Wiley & Sons, Ltd.

Received 11 January 2007; Revised 1 October 2007; Accepted 1 October 2007

KEY WORDS: high-resolution schemes; flux-corrected transport; Newton-like solution techniques; sparse Jacobian evaluation; finite elements; implicit time stepping

1. INTRODUCTION

The advent of nonlinear high-resolution schemes for convection-dominated flows traces its origins to the *flux-corrected transport* (FCT) methodology introduced in the early 1970s by Boris and Book [1]. The fully multidimensional generalization proposed by Zalesak [2] has formed a very

*Correspondence to: M. Möller, Institute of Applied Mathematics (LS III), University of Dortmund, Vogelpothsweg 87, D-44227 Dortmund, Germany.
†E-mail: matthias.moeller@math.uni-dortmund.de

general framework for the design of FCT algorithms by representing them as a blend of linear high- and low-order approximations. Unlike other limiting techniques, which are typically based on geometric design criteria, flux correction of FCT type is readily applicable to finite element discretizations on unstructured meshes [3, 4]. A comprehensive summary of the state of the art can be found in [4–7].

The design philosophy behind modern front-capturing methods involves a set of physical or mathematical constraints to be imposed on the discrete solution so as to prevent the formation of spurious undershoots and overshoots in the vicinity of steep gradients. To this end, the following algorithmic components are to be specified [6, 7]:

- a high-order approximation which may fail to possess the desired properties;
- a low-order approximation which does enjoy these properties but is less accurate;
- a way to decompose the difference between the above into a sum of skew-symmetric internodal fluxes which can be manipulated without violating mass conservation; and
- a cost-effective mechanism for adjusting these antidiffusive fluxes in an adaptive fashion so that the imposed constraints are satisfied for a given solution.

Classical FCT algorithms are based on an explicit correction of the low-order solution whose local extrema serve as the upper/lower bounds for the sum of limited antidiffusive fluxes. In the case of an implicit time discretization, which gives rise to a nonlinear algebraic system, the same strategy can be used to secure the positivity of the right-hand side, whereas the left-hand side is required to satisfy the *M-matrix* property. A nonsingular discrete operator A with nonpositive off-diagonal entries $a_{ij} \leqslant 0, \forall j \neq i$ is called an M-matrix if all the coefficients of its inverse are nonnegative. Consequently, for an M-matrix $Ax \geqslant 0$ implies that $x \geqslant 0$.

The rationale for the development of implicit FCT algorithms stems from the fact that the underlying linear discretizations must be stable. In particular, the use of an unstable high-order method may give rise to nonlinear instabilities which manifest themselves in significant distortions of the solution profiles as an aftermath of aggressive flux limiting. In the finite element context, a proper amount of streamline diffusion can be used to stabilize an explicit Galerkin scheme. However, the evaluation of extra terms increases the cost of matrix assembly and the time step must satisfy a restrictive Courant–Friedrichs–Lewy (CFL) condition. On the other hand, unconditionally stable implicit methods can be operated at large time steps (unless iterative solvers fail to converge or the positivity criterion is violated) and there is no need for any extra stabilization. Moreover, the overhead cost is insignificant, since the use of a consistent-mass matrix leads to a sequence of linear systems even in the fully explicit case.

The generalized FEM-FCT methodology introduced in [8, 9] and refined in [10, 11] is applicable to implicit time discretizations but the cost of iterative flux correction is rather high if the sum of limited antidiffusive fluxes and the nodal correction factors need to be updated in each outer iteration. In addition, the nonlinear convergence rates leave a lot to be desired in many cases. The use of 'frozen' correction factors computed at the beginning of the time step by the standard Zalesak limiter alleviates the convergence problems but the linearized scheme can no longer be guaranteed to remain positivity-preserving. The semi-implicit limiting strategy to be described below makes it possible to overcome this problem and enforce the positivity constraint at a cost comparable with that of explicit flux correction. The resulting FEM-FCT algorithm is to be recommended for strongly time-dependent problems discretized in time by the Crank–Nicolson scheme. The design of general-purpose (GP) flux limiters which are more expensive but do not suffer from a loss of accuracy at large time steps is addressed in [12].

DOI: 10.1002/fld

In the present paper, we compare a new semi-implicit FCT scheme with its semi-explicit prototype and focus on the iterative solution of the resulting nonlinear algebraic systems. As an alternative to the fixed-point defect correction scheme which tends to converge rather slowly, a discrete Newton method tailored to the peculiarities of FEM-FCT schemes is developed. The sparse Jacobian matrix is approximated with a second-order accuracy by means of divided differences and assembled edge by edge. The semi-implicit nature of the new FCT limiter makes the Jacobian assembly particularly efficient, since the sparsity pattern of the underlying matrices is preserved. A detailed numerical study illustrates the potential of flux-corrected Galerkin schemes combined with discrete Newton methods for the treatment of nonlinearities.

2. ALGEBRAIC FLUX CORRECTION

In this paper, we adopt an algebraic approach to the design of high-resolution schemes which consists of imposing certain mathematical constraints on discrete operators, so as to achieve some favorable matrix properties. A handy algebraic criterion, which represents a multidimensional generalization of Harten's TVD theorem, was introduced by Jameson [13, 14] who proved that a semi-discrete scheme of the general form

$$\frac{\mathrm{d}u_i}{\mathrm{d}t}=\sum_{j\neq i} c_{ij}(u_j-u_i),\quad c_{ij}\geqslant 0\ \ \forall j\neq i \tag{1}$$

is *local extremum diminishing* (LED). After the discretization in time by a two-level scheme, such methods remain positivity-preserving provided that each solution update $u^n\rightarrow u^{n+1}$ or the converged steady-state solution $u^{n+1}=u^n$ satisfies an algebraic system of the form

$$Au^{n+1}=Bu^n+f \tag{2}$$

where $A=\{a_{ij}\}$ is an M-matrix, whereas $B=\{b_{ij}\}$ and $f=\{f_i\}$ have no negative entries. Under these conditions, the positivity of the old solution carries over to the new one [6, 11]

$$u^n\geqslant 0\ \Rightarrow\ u^{n+1}=A^{-1}[Bu^n+f]\geqslant 0 \tag{3}$$

If the underlying spatial discretization is LED, then the off-diagonal coefficients of both matrices have the right sign, while the positivity condition $b_{ii}\geqslant 0$ for the diagonal entries of B yields a readily computable upper bound for admissible time steps [6]. In what follows, we discretize in time using the standard θ-scheme which yields $A=I-\theta\Delta tC$ and $B=I+(1-\theta)\Delta tC$. The resulting CFL-like condition for the time step Δt reads

$$1+\Delta t(1-\theta)\min_i c_{ii}^n\geqslant 0\quad \text{for } 0\leqslant\theta<1 \tag{4}$$

The discretization is unconditionally positivity-preserving if a fully implicit time-stepping scheme ($\theta=1$) is adopted. Of course, the above algebraic constraints are not the necessary but merely the sufficient conditions for a numerical scheme to be local extremum diminishing and/or positivity-preserving. In the linear case, they turn out to be far too restrictive. According to the well-known Godunov theorem, linear schemes satisfying these criteria are doomed to be (at most) first-order accurate. On the other hand, a high-order discretization which fails to satisfy the imposed constraints unconditionally can be adjusted so that it admits an equivalent representation of form (1) and/or (2), where the matrix entries may depend on the unknown solution. This idea makes it possible to construct a variety of nonlinear high-resolution schemes based on the so-called *algebraic flux*

DOI: 10.1002/fld

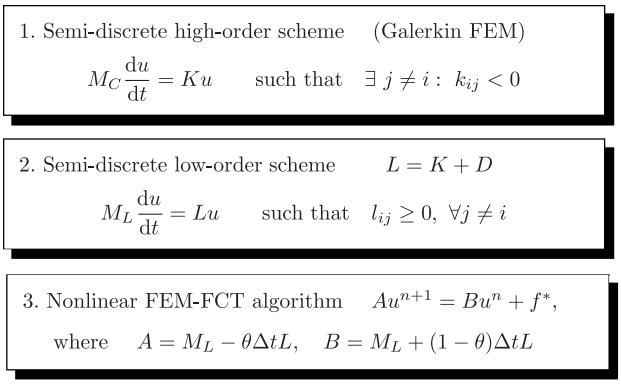


Figure 1. Roadmap of matrix manipulations.

correction (AFC) paradigm. A detailed overview of this methodology is given in the survey article [6]. The design of flux limiters for finite element discretizations with a consistent-mass matrix is addressed in [12].

To keep the presentation self-contained, we will follow the road map displayed in Figure 1 and explain the meaning of all discrete operators in the next three sections. Roughly speaking, a high-order Galerkin discretization is to be represented in the generic form (2), where the matrices A and B do satisfy the above-mentioned positivity constraint. In order to guarantee that the vector f poses no hazard to positivity either, it is to be replaced by its limited counterpart f^* such that the right-hand side remains nonnegative for $u^n \geqslant 0$. This modification is mass conserving provided that both f and f^* can be decomposed into skew-symmetric internodal fluxes as defined below. A family of implicit FEM-FCT schemes based on this algebraic approach was proposed in [8, 9] and combined with an iterative-limiting strategy in [11]. In Section 5.2, we present an alternative generalization of Zalesak's limiter which offers some extra advantages. The new approach to flux correction of FCT type is also based on the positivity constraint (2) but enforces it in another way so that the costly computation of nodal correction factors is performed just once per time step. The positivity of the resulting semi-implicit FCT algorithm will be proven in Section 5.3.

Solution strategies for the nonlinear algebraic system to be solved in each time step are presented in Section 5.4. In particular, a discrete Newton method is proposed as a promising alternative to the standard defect correction approach. A suitable approximation to the Jacobian matrix is constructed using divided differences. In the framework of the semi-implicit FCT algorithm, this can be accomplished in a very efficient way since the correction factors are computed just once per time step in the first outer iteration. In contrast, the use of a semi-explicit FEM-FCT scheme results in an extended sparsity pattern for the Jacobian operator. The same side effect is observed in the context of high-resolution schemes of TVD type [15].

3. SEMI-DISCRETE HIGH-ORDER SCHEME

As a standard model problem, consider the time-dependent continuity equation for a scalar quantity u transported by the velocity field $\mathbf{v}$ which is assumed to be known

$$\frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{v}u) = 0 \tag{5}$$

DOI: 10.1002/fld

Let the discretization in space be performed by an (Galerkin) FEM which yields a DAE system for the vector of time-dependent nodal values

$$M_C \frac{du}{dt} = Ku \tag{6}$$

where $M_C = \{m_{ij}\}$ denotes the consistent-mass matrix and $K = \{k_{ij}\}$ is the discrete transport operator. The latter may contain some streamline diffusion used for stabilization purposes and/or to achieve better phase accuracy, e.g. in the framework of Taylor–Galerkin methods. Its skew-symmetric part $\frac{1}{2}(K - K^T)$ provides a consistent discretization of $\mathbf{v} \cdot \nabla$, whereas the symmetric part $\frac{1}{2}(K + K^T) - \text{diag}\{K\}$ represents a discrete (anti-)diffusion operator.

4. SEMI-DISCRETE LOW-ORDER SCHEME

In the case of linear discretizations, the algebraic constraints (1) and (2) can be readily enforced by means of 'discrete upwinding' as proposed in [8, 9]. For a semi-discrete finite element scheme of the form (6), the required matrix manipulations are as follows:

- replace the consistent-mass matrix M_C by its lumped counterpart $M_L = \text{diag}\{m_i\}$;
- render the operator K local extremum diminishing by adding an artificial diffusion operator $D = \{d_{ij}\}$ so as to eliminate all negative off-diagonal coefficients.

This straightforward 'postprocessing' transforms (6) into its linear LED counterpart

$$M_L \frac{du}{dt} = Lu, \quad L = K + D \tag{7}$$

where D is supposed to be a symmetric matrix with zero row and column sums. For each pair of nonzero off-diagonal coefficients k_{ij} and k_{ji} of the high-order operator K, the optimal choice of the artificial diffusion coefficient d_{ij} reads [6, 9]

$$d_{ij} = \max\{-k_{ij}, 0, -k_{ji}\} = d_{ji} \tag{8}$$

Alternatively, one can apply discrete upwinding to the skew-symmetric part $\frac{1}{2}(K - K^T)$ of the original transport operator K, which corresponds to

$$d_{ij} = \frac{|k_{ij} - k_{ji}|}{2} - \frac{k_{ij} + k_{ji}}{2} = d_{ji} \tag{9}$$

In either case, the off-diagonal coefficients of the low-order operator $l_{ij} := k_{ij} + d_{ij}$ are nonnegative, as required by the LED criterion (1). Owing to the zero row sum property of the artificial diffusion operator D, the diagonal coefficients of L are given by

$$l_{ii} := k_{ii} - \sum_{j \neq i} d_{ij} \tag{10}$$

The semi-discretized equation for the nodal value $u_i(t)$ can be represented as

$$m_i \frac{du_i}{dt} = \sum_{j \neq i} l_{ij}(u_j - u_i) + u_i \sum_j l_{ij} \tag{11}$$

DOI: 10.1002/fld

where $m_i = \sum_j m_{ij} > 0$ and $l_{ij} \geqslant 0, \ \forall j \neq i$. The last term in the above expression represents a discrete counterpart of $-u\nabla \cdot \mathbf{v}$ which is responsible for a physical growth of local extrema [6]. Recall that the operator D has zero row sums so that $u_i \sum_j l_{ij} = u_i \sum_j k_{ij}$ in Equation (11). In the semi-discrete case, this term is harmless since (cf. [16])

$$u_i(t) = 0, \quad u_j(t) \geqslant 0 \ \forall j \neq i \ \Rightarrow \ \frac{\mathrm{d}u_i}{\mathrm{d}t} \geqslant 0 \tag{12}$$

which proves that the low-order scheme (7) is positivity-preserving. For the fully discrete system to inherit this property, the time step should be chosen in accordance with the CFL-like condition (4) unless the backward Euler time stepping ($\theta = 1$) is employed.

5. NONLINEAR FEM-FCT ALGORITHM

The high-order system (6) discretized in time by a standard two-level θ-scheme

$$[M_C - \theta \Delta t K]u^{n+1} = [M_C + (1-\theta)\Delta t K]u^n \tag{13}$$

admits an equivalent representation in form (2) amenable to flux correction

$$[M_L - \theta \Delta t L]u^{n+1} = [M_L + (1-\theta)\Delta t L]u^n + f(u^{n+1}, u^n) \tag{14}$$

The last term in the right-hand side is assembled from skew-symmetric internodal fluxes f_{ij} which can be associated with the edges of the sparsity graph [6]

$$f_i = \sum_{j \neq i} f_{ij} \quad \text{where } f_{ji} = -f_{ij} \tag{15}$$

Specifically, these *raw antidiffusive fluxes*, which offset the discretization error induced by mass lumping and discrete upwinding, are given by the formula [6, 11]

$$f_{ij} = [m_{ij} + \theta \Delta t d_{ij}^{n+1}](u_i^{n+1} - u_j^{n+1}) - [m_{ij} - (1-\theta)\Delta t d_{ij}^n](u_i^n - u_j^n) \tag{16}$$

Interestingly enough, the contribution of the consistent-mass matrix consists of a truly antidiffusive implicit part and a diffusive explicit part which has a strong damping effect. In fact, explicit mass diffusion of the form $(M_C - M_L)u^n$ has been used to construct the 'monotone' low-order method in the framework of explicit FEM-FCT algorithms [3].

In the case of an implicit time discretization ($0 < \theta \leqslant 1$), the nonlinearities inherent to the governing equation and/or to the employed high-resolution scheme call for the use of an iterative solution strategy. Let successive approximations to the solution u^{n+1} at the new time level $t^{n+1} = t^n + \Delta t$ be computed step by step in the framework of a fixed-point iteration

$$u^{(m+1)} = u^{(m)} + [C^{(m)}]^{-1} r^{(m)}, \quad m = 0, 1, 2, \ldots \tag{17}$$

where $C^{(m)}$ denotes a suitable 'preconditioner' (to be defined below) that should be easy to invert. The corresponding residual vector of the mth outer iteration is given by

$$r^{(m)} = b^{(m)} - Au^{(m)} \tag{18}$$

Here, A represents the 'monotone' evolution operator for the underlying low-order scheme

$$A = M_L - \theta \Delta t L, \quad L = K + D \tag{19}$$

DOI: 10.1002/fld

which enjoys the M-matrix property, since the off-diagonal entries of L are nonnegative by construction. The right-hand side $b^{(m)}$, which needs to be updated in each outer iteration, consists of a low-order part augmented by limited antidiffusion [6]

$$b^{(m)} = Bu^n + f^*(u^{(m)}, u^n), \quad B = M_L + (1-\theta)\Delta t L \tag{20}$$

In order to prevent the formation of nonphysical undershoots and overshoots, the raw antidiffusive fluxes f_{ij} should be multiplied by suitable correction factors so that

$$f_i^* = \sum_{j \neq i} \alpha_{ij} f_{ij} \quad \text{where } 0 \leqslant \alpha_{ij} \leqslant 1 \tag{21}$$

This adjustment transforms (14) into a nonlinear combination of the low-order scheme ($\alpha_{ij} \equiv 0$) and the original high-order one ($\alpha_{ij} \equiv 1$). The task of the flux limiter is to determine an optimal value of each correction factor α_{ij} individually so as to remove as much artificial diffusion as possible without violating the positivity constraint introduced in Section 2.

In a practical implementation, the 'inversion' of the operator $C^{(m)}$ is also performed by a suitable iteration procedure for solving the sequence of linear subproblems

$$C^{(m)} \Delta u^{(m+1)} = r^{(m)}, \quad m = 0, 1, 2, \ldots \tag{22}$$

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

$$u^{(m+1)} = u^{(m)} + \Delta u^{(m+1)}, \quad u^{(0)} = u^n \tag{23}$$

A natural choice for the preconditioner $C^{(m)}$ is the monotone low-order operator (19) so that the iteration procedure (17) yields the standard fixed-point defect correction scheme

$$Au^{(m+1)} = b^{(m)}, \quad m = 0, 1, 2, \ldots \tag{24}$$

Ideally, $C^{(m)}$ should be a good approximation of the Jacobian matrix $J^{(m)}$ with coefficients

$$J_{ij}^{(m)} = -\left.\frac{\partial r_i}{\partial u_j}\right|_{u=u^{(m)}} \tag{25}$$

evaluated at the last iterate $u^{(m)}$. It is well known that the convergence behavior of *Newton's method* which corresponds to (17) with $C^{(m)} = J^{(m)}$ is quite sensitive to the initial guess $u^{(0)}$. Owing to the fact that linear subproblems (22) are solved by an iterative technique, the resulting algorithm is categorized as an inexact Newton method [17]. A simple inexact scheme is based on the following convergence criterion in each linear iteration:

$$\| J^{(m)} \Delta u^{(m+1)} - r^{(m)} \| \leqslant \eta \| r^{(m)} \| \tag{26}$$

whereby the so-called forcing term $\eta \in [0, 1)$ can be chosen adaptively [18]. Furthermore, some globalization strategy may be required to enhance the robustness of Newton's method. For a detailed description of such techniques which are mainly designed to guarantee a sufficient decrease of the nonlinear residual (18), the interested reader is referred to the literature, e.g. [19]. In the case of time-dependent problems, globalization is less critical due to the fact that the solution from the last time step may serve as a good initial guess.

DOI: 10.1002/fld

Linear subproblems (22) can be solved, e.g. using a Krylov subspace method such as BiCGSTAB or GMRES combined with preconditioning of ILU type. Owing to the M-matrix property of the evolution operator (19), its incomplete LU factorization unconditionally exists and is unique [20]. Hence, it is advisable to use A as preconditioner for the Krylov solver even if the Jacobian matrix (25) is adopted in the outer iteration procedure.

5.1. Semi-explicit FCT limiter

The first implicit FCT algorithm for finite element discretizations on unstructured meshes [8, 9] was based on the following limiting strategy which was eventually superseded by further extensions proposed in a series of subsequent publications [10, 11]:

1. Compute the high-order solution to (14) in an iterative way by solving (17) using the total amount of raw antidiffusion ($\alpha_{ij} \equiv 1$) to assemble the term f^*.
2. Evaluate the contribution of the consistent-mass matrix to the raw antidiffusive fluxes (16) using the converged high-order solution as a substitute for u^{n+1}.
3. Solve the explicit subproblem $M_L \tilde{u} = Bu^n$ for the positivity-preserving intermediate solution $\tilde{u}$ which represents an explicit low-order approximation to $u(t^{n+1-\theta})$.
4. Invoke Zalesak's multidimensional FCT limiter to determine the correction factors α_{ij} so as to secure the positivity of the right-hand side as explained below.
5. Compute the final solution by solving the linear system $Au^{n+1} = b$, where

$$b_i = m_i \tilde{u}_i + \sum_{j \neq i} f_{ij}^*, \quad f_{ij}^* = \alpha_{ij} f_{ij} \tag{27}$$

In the fully explicit case ($\theta = 0$), we have $A = M_L$ so that $u^{n+1} = M_L^{-1} b$ can be computed explicitly from (24), and the classical FEM-FCT algorithm of Löhner *et al.* [3] and Löhner and Baum [4] is recovered. The crux of the above generalization lies in the special choice of the operator A which guarantees that the positivity of the right-hand side is preserved, whence

$$\tilde{u} \geqslant 0 \;\Rightarrow\; b \geqslant 0 \;\Rightarrow\; u^{n+1} = A^{-1} b \geqslant 0 \tag{28}$$

The flux correction process starts with an optional 'prelimiting' of the raw antidiffusive fluxes f_{ij}. It consists of cancelling the 'wrong' ones which tend to flatten the intermediate solution and create numerical artifacts. The required adjustment is given by [12]

$$f'_{ij} := \max\{0, p_{ij}\}(\tilde{u}_i - \tilde{u}_j), \quad p_{ij} = f_{ij}/(\tilde{u}_i - \tilde{u}_j) \tag{29}$$

The remaining fluxes are truly antidiffusive and need to be limited. The upper and lower bounds to be imposed on the net antidiffusive flux depend on the local extrema

$$\tilde{u}_i^{\max} = \max_{j \in S_i} \tilde{u}_j, \quad \tilde{u}_i^{\min} = \min_{j \in S_i} \tilde{u}_j \tag{30}$$

where $S_i = \{j \mid m_{ij} \neq 0\}$ denotes the set of nodes which share an element with node i.

In the worst case, all antidiffusive fluxes in node i have the same sign. Hence, it is worthwhile to treat the positive and negative ones separately, as proposed by Zalesak [2].

1. Evaluate the sums of all positive and negative antidiffusive fluxes in node i

$$P_i^+ = \sum_{j \neq i} \max\{0, f'_{ij}\}, \quad P_i^- = \sum_{j \neq i} \min\{0, f'_{ij}\} \tag{31}$$

DOI: 10.1002/fld

2. Compute the distance to a local maximum/minimum of the low-order solution

$$Q_i^+ = \tilde{u}_i^{\max} - \tilde{u}_i, \quad Q_i^- = \tilde{u}_i^{\min} - \tilde{u}_i \tag{32}$$

3. Calculate the nodal correction factors which prevent overshoots/undershoots

$$R_i^+ = \min\{1, m_i Q_i^+ / P_i^+\}, \quad R_i^- = \min\{1, m_i Q_i^- / P_i^-\} \tag{33}$$

4. Check the sign of f'_{ij} and apply $R_i^{\pm}$ or $R_j^{\mp}$, whichever is smaller, so that

$$\alpha_{ij} = \begin{cases} \min\{R_i^+, R_j^-\} & \text{if } f'_{ij} > 0 \\ \min\{R_i^-, R_j^+\} & \text{otherwise} \end{cases} \tag{34}$$

This symmetric-limiting strategy guarantees that the corrected right-hand side (27) satisfies the constraint $\tilde{u}_i^{\min} \leqslant b_i / m_i \leqslant \tilde{u}_i^{\max}$. Owing to the fact that the low-order operator A was designed to be an M-matrix, the resulting scheme proves positivity-preserving [6, 9].

It is worth mentioning that the constituents of the sums $P_i^{\pm}$ vary with Δt, while the corresponding upper/lower bounds $Q_i^{\pm}$ are fixed. Consequently, the correction factors α_{ij} produced by Zalesak's limiter depend on the underlying time step. This peculiarity of FCT methods turns out to be a blessing and a curse at the same time. On the one hand, a larger portion of the raw antidiffusive flux f_{ij} may be retained as the time step is refined. On the other hand, the accuracy of FCT algorithms deteriorates as Δt increases, since the positivity constraint (2) becomes too restrictive. The iterative-limiting strategy proposed in [11] alleviates this problem to some extent by adjusting the correction factors α_{ij} in each outer iteration so as to recycle the rejected antidiffusion step by step. However, the cost of iterative flux correction is rather high and severe convergence problems may occur. Therefore, other limiting techniques such as the general-purpose flux limiter introduced in [12] are to be preferred if the solution is expected to reach a steady state in the long run.

5.2. Semi-implicit FCT limiter

For truly time-dependent problems, the use of moderately small time steps is dictated by accuracy considerations so that flux limiting of FCT type is appropriate. In this case, the underlying time-stepping method should provide (unconditional) stability and be at least second-order accurate in order to capture the evolutionary details. For this reason, we favor an implicit time discretization of Crank–Nicolson type ($\theta = \frac{1}{2}$) and mention the strongly A-stable fractional-step θ-scheme [21] as a promising alternative.

The semi-explicit limiting strategy presented in the previous section can be classified as an algorithm of predictor–corrector type since the implicit part of the raw antidiffusive flux (16) is evaluated using the converged high-order solution in place of u^{n+1}. This handy linearization, which can be traced back to the classical FEM-FCT procedure [3], makes it possible to perform flux correction in a very efficient way, since Zalesak's limiter is invoked just once per time step. However, a lot of CPU time needs to be invested in the iterative solution of the ill-conditioned high-order system and the convergence may even fail if the time step is too large. Moreover, the final solution fails to satisfy the nonlinear algebraic system (17) upon substitution. On the other hand, an update of the auxiliary quantities $P_i^{\pm}$, $Q_i^{\pm}$, and $R_i^{\pm}$ in each outer iteration would trigger the cost of flux limiting and compromise the benefits of implicit time stepping. In order to

DOI: 10.1002/fld

circumvent this problem, let us introduce a semi-implicit FCT algorithm which can be implemented as follows:

- At the first outer iteration ($m=1$), compute a set of antidiffusive fluxes $\tilde{f}_{ij}$ which provide an explicit estimate for the admissible magnitude of $f_{ij}^{*}=\alpha_{ij}f_{ij}$

  1. Initialize all auxiliary arrays by zeros: $P_i^{\pm}\equiv 0,\ Q_i^{\pm}\equiv 0,\ R_i^{\pm}\equiv 0$.
  2. Compute the positivity-preserving intermediate solution of low order:

$$\tilde{u}=u^n+(1-\theta)\Delta t M_{\mathrm{L}}^{-1}Lu^n \tag{35}$$

  3. For each pair of neighboring nodes i and j, evaluate the raw antidiffusive flux

$$f_{ij}^n=\Delta t d_{ij}^n(u_i^n-u_j^n) \tag{36}$$

  and add its contribution to the sums of positive/negative edge contributions

$$P_i^{\pm}:=P_i^{\pm}+\mathop{\max}_{\min}\{0,\ f_{ij}^n\},\quad P_j^{\pm}:=P_j^{\pm}+\mathop{\max}_{\min}\{0,-f_{ij}^n\} \tag{37}$$

  4. Update the maximum/minimum admissible increments for both nodes:

$$Q_i^{\pm}:=\mathop{\max}_{\min}\{Q_i^{\pm},\tilde{u}_j-\tilde{u}_i\},\quad Q_j^{\pm}:=\mathop{\max}_{\min}\{Q_j^{\pm},\tilde{u}_i-\tilde{u}_j\} \tag{38}$$

  5. Relax the constraint $R_i^{\pm}\leqslant 1$ for the nodal correction factors and compute

$$R_i^{\pm}:=m_i Q_i^{\pm}/P_i^{\pm} \tag{39}$$

  6. Multiply the raw antidiffusive fluxes f_{ij}^n by the minimum of $R_i^{\pm}$ and $R_j^{\mp}$:

$$\tilde{f}_{ij}=\begin{cases}\min\{R_i^{+},R_j^{-}\}f_{ij}^n & \text{if } f_{ij}^n>0\\ \min\{R_i^{-},R_j^{+}\}f_{ij}^n & \text{otherwise}\end{cases} \tag{40}$$

- At each outer iteration ($m=1,2,\ldots$), assemble f^{*} and substitute it into (20)

  1. Update the target flux (16) using the solution from the previous iteration:

$$f_{ij}=[m_{ij}+\theta\Delta t d_{ij}^{(m)}](u_i^{(m)}-u_j^{(m)})-[m_{ij}-(1-\theta)\Delta t d_{ij}^n]\ (u_i^n-u_j^n) \tag{41}$$

  2. Constrain each flux f_{ij} so that its magnitude is bounded by that of $\tilde{f}_{ij}$:

$$f_{ij}^{*}=\begin{cases}\min\{f_{ij},\max\{0,\tilde{f}_{ij}\}\} & \text{if } f_{ij}>0\\ \max\{f_{ij},\min\{0,\tilde{f}_{ij}\}\} & \text{otherwise}\end{cases} \tag{42}$$

  3. Insert the limited antidiffusive fluxes f_{ij}^{*} into the right-hand side (20):

$$b_i^{(m)}:=b_i^{(m)}+f_{ij}^{*},\quad b_j^{(m)}:=b_j^{(m)}-f_{ij}^{*} \tag{43}$$

Owing to the fact that f_{ij}^n is not the real target flux but merely an explicit predictor used to estimate the maximum amount of admissible antidiffusion, the multipliers $R_i^{\pm}$ are redefined so that the ratio

DOI: 10.1002/fld

$\tilde{f}_{ij}/f_{ij}^n$ may exceed unity. However, the effective correction factors $\alpha_{ij} := f_{ij}^*/f_{ij}$ are bounded by 0 and 1, as required for consistency.

Instead of computing the optimal upper/lower bounds (32) for a given time step, it is also possible to use some reasonable fixed bounds and adjust the time step if this is necessary to satisfy a CFL-like condition (as in the case of TVD methods). For instance, the auxiliary quantities $Q_i^{\pm}$ can be computed using u^n instead of $\tilde{u}$

$$Q_i^+ = \max_{j\in S_i} u_j^n - u_i^n, \quad Q_i^- = \min_{j\in S_i} u_j^n - u_i^n \tag{44}$$

The corresponding nodal correction factors $R_i^{\pm}$ should be redefined as [12]

$$R_i^{\pm} = (m_i - m_{ii})\, Q_i^{\pm}/P_i^{\pm} \tag{45}$$

where $m_i - m_{ii} = \sum_{j\neq i} m_{ij}$ is the difference between the diagonal entries of the consistent- and lumped-mass matrices. This modification eliminates the need for evaluation of the intermediate solution $\tilde{u}$ in (35) and leads to a single-step FCT algorithm.

For a given time step, multipliers (45) will typically be smaller than those defined by (39). However, in either case the denominator $P_i^{\pm}$ is proportional to Δt. Therefore, the difference between the effective correction factors α_{ij} will shrink and eventually vanish as the time step is refined. As long as Δt is sufficiently small, the accuracy of both FCT techniques depends solely on the choice of the underlying high-order scheme.

5.3. Positivity proof

The positivity proof for the semi-implicit FCT algorithm (35)–(43) follows that for the classical Zalesak limiter, see [6, 9]. In the nontrivial case $f_i^* \neq 0$, the ith component of the right-hand side (20) admits the following representation:

$$b_i^* = m_i\tilde{u}_i + f_i^* = (m_i - \alpha_i)\tilde{u}_i + \alpha_i\tilde{u}_k \tag{46}$$

where the coefficient $\alpha_i = f_i^*/(\tilde{u}_k - \tilde{u}_i)$ is defined in terms of the local extremum

$$\tilde{u}_k = \begin{cases} \tilde{u}_i^{\max} & \text{if } f_i^* > 0 \\ \tilde{u}_i^{\min} & \text{if } f_i^* < 0 \end{cases} \tag{47}$$

This definition implies that $f_i^* = \alpha_i Q_i^{\pm}$, where $\alpha_i > 0$. By virtue of (46), the sign of the intermediate solution $\tilde{u}$ is preserved if the inequality $m_i - \alpha_i \geqslant 0$ holds.

In the case $f_i^* < 0$, the antidiffusive correction to node i is bounded from below by

$$m_i Q_i^- \leqslant R_i^- P_i^- \leqslant \sum_{j\neq i} \min\{0, \tilde{f}_{ij}\} \leqslant f_i^* = \alpha_i Q_i^- \tag{48}$$

Similarly, a strictly positive antidiffusive correction $f_i^* > 0$ is bounded from above by

$$\alpha_i Q_i^+ = f_i^* \leqslant \sum_{j\neq i} \max\{0, \tilde{f}_{ij}\} \leqslant R_i^+ P_i^+ \leqslant m_i Q_i^+ \tag{49}$$

It follows that $0 \leqslant \alpha_i \leqslant m_i$, which proves that $b_i^* \geqslant 0$ provided that $\tilde{u}_i \geqslant 0$ and $\tilde{u}_k \geqslant 0$.

DOI: 10.1002/fld

In light of the above, the semi-implicit FCT limiter is positivity-preserving as long as the diagonal coefficients of the matrix B as defined in (20) are nonnegative. The corresponding CFL-like condition (4) for the maximum admissible time step reads

$$(1-\theta)\Delta t \leqslant \min_i |m_i / l_{ii}| \tag{50}$$

The positivity of the single-step algorithm based on the slack bounds (44)–(45) can be proven in a similar way using the following representation of the right-hand side:

$$b_i^* = (m_i - \alpha_i)u_i^n + \alpha_i u_k^n + (1-\theta)\Delta t \sum_j l_{ij} u_j^n \tag{51}$$

In this case, the limited antidiffusive correction to node i can be estimated as follows:

$$(m_i - m_{ii})Q_i^- \leqslant f_i^* \leqslant (m_i - m_{ii})Q_i^+ \tag{52}$$

so that $m_i - \alpha_i \geqslant m_{ii}$. Thus, the right-hand side given by (51) preserves the sign of u^n if the time step satisfies the positivity constraint for all diagonal coefficients

$$(1-\theta)\Delta t \leqslant \min_i |m_{ii} / l_{ii}| \tag{53}$$

Under the above conditions, the M-matrix property of the low-order operator (19) is sufficient to guarantee that *each* solution update is positivity-preserving if the fixed-point iteration (17) is preconditioned by $C^{(m)} = A, \ \forall m$. On the other hand, only the fully converged solution is certain to remain positive if Newton's method ($C^{(m)} = J^{(m)}$) is employed.

5.4. Approximation of Jacobians

For the practical application of Newton's method, it remains to devise an algorithm for the construction of the Jacobian matrix (25). For simplicity, superscript m will be omitted unless indicated otherwise. In what follows, differentiation is to be performed with respect to u, whereas u^n is regarded as a given constant. The nonlinear residual (18) depends on the 'monotone' operator A and on the right-hand side $b^{(m)}$ which are given by relations (19) and (20), respectively. Hence, it is advisible to split the Jacobian operator $J = \bar{J} + J^*$ into its 'upwind' part $\bar{J} = \{\bar{J}_{ij}\}$ and the contribution of the antidiffusive correction $J^* = \{J_{ij}^*\}$.

Formally, the upwind Jacobian is given by $\bar{J} = A'u + A$ which reduces to the M-matrix $A = M_L - \theta\Delta t L$ for linear model problems of form (5). On the other hand, its derivative A' does not vanish if the original transport operator $K(u)$ and hence its LED counterpart $L(u) = K(u) + D(u)$ depend on the unknown solution. Since the artificial diffusion operator $D(u)$ was derived on the semi-discrete level by means of matrix manipulations, no analytical expression is available for $A'(u)$. As a remedy, the Jacobian matrix is approximated by means of divided differences. To this end, let $f : \mathbb{R}^n \to \mathbb{R}$ denote a generic function for which the central divided difference operator is defined as follows:

$$\mathscr{D}_k[f(u)] := \frac{f(u+\sigma e_k) - f(u-\sigma e_k)}{2\sigma} \tag{54}$$

In the above equation, e_k denotes the kth unit vector. The optimal choice of the perturbation parameter $0 < \sigma \ll 1$ requires some knowledge of the sensitivity of the function f and has been

DOI: 10.1002/fld

addressed by many authors. Following a strategy proposed by Nielsen *et al.* [22] it can be chosen proportional to the square root of the machine precision ε. The formula

$$\sigma = [\varepsilon(1+\|u\|)]^{1/3} \tag{55}$$

suggested in [23] takes into account the norm of the solution and is claimed to reduce the noise arising from the numerical evaluation of the function f. Some alternative choices are given in a survey paper on Jacobian-free Newton–Krylov methods by Knoll and Keyes [24].

The central divided difference operator introduced in (54) yields a second-order accurate approximation for each entry of the upwind Jacobian $\bar{J}$

$$\bar{J}_{ik} = \mathscr{D}_k[(Au)_i] + \mathcal{O}(\sigma^2) \tag{56}$$

Since $A = M_L - \theta\Delta t L$, the so-defined coefficient $\bar{J}_{ik}$ can be cast into the form [15]

$$\bar{J}_{ik} = \delta_{ik} m_i - \theta\Delta t \left(\bar{l}_{ik} + \sum_j \mathscr{D}_k[l_{ij}]u_j \right) \tag{57}$$

where $\delta_{ik} \in \{0, 1\}$ denotes the standard Kronecker delta symbol and the auxiliary quantity $\bar{l}_{ik}$ stands for the average of the perturbed evolution coefficients resulting from discrete upwinding

$$\bar{l}_{ik} = \frac{l_{ik}(u+\sigma e_k) + l_{ik}(u-\sigma e_k)}{2} \tag{58}$$

It is also possible to neglect averaging in Equation (57) and replace the averaged transport operator $\bar{L}$ by the standard low-order term L. If the problem at hand is linear, the upwind Jacobian reduces to $\bar{J} = M_L - \theta\Delta t L$ due to the fact that all divided differences vanish. It is worth mentioning that the decomposition into individual edge contributions yields an efficient assembly procedure for the Jacobian which can be considered as a viable alternative to the element-by-element procedure traditionally employed in the finite element community [25].

Interestingly enough, the operator $\bar{J}$ exhibits the same sparsity pattern as the finite element matrix, that is, $\bar{J}_{ij} \neq 0$ implies $m_{ij} \neq 0$. As soon as AFC comes into play, this amenable property may be lost. For upwind-biased discretization schemes of TVD type, the phenomenon of matrix fill-in engendered by the flux limiter has been analyzed in Section 4.2 of Reference [15]. Owing to conceptional similarities within the family of AFC schemes, essentially the same analysis remains valid for symmetric flux limiters [6, 12]. As we are about to see, the semi-implicit FCT algorithm presented in Section 5.2 is free of this drawback and hence particularly suitable for the application of Newton's method.

To highlight this advantage of the semi-implicit approach, let us revisit the general procedure of evaluating the antidiffusive contribution J^* to the Jacobian operator. To this end, let the solution vector u be perturbed at some node, say k, and compute the compensating antidiffusive fluxes (21) based on the solution vectors $u+\sigma e_k$ and $u-\sigma e_k$ following the semi-explicit limiting algorithm presented in Section 5.1. By construction, the nodal correction factors $R_i^{\pm}$ defined in (33) may be affected by this perturbation for all i from the set $\mathscr{S}_k$ of nodes which share an element with node k. Consequently, the final correction factors α_{ij} defined in (34) may have different values if at least one of the nodes i and j belongs to the set $\mathscr{S}_k$. To put it in a nutshell, the impact of 'joggling' the solution value at one particular node usually propagates along two edges ij by virtue of the correction factors α_{ij} which are recalculated in each iteration. Figure 2 illustrates the

DOI: 10.1002/fld

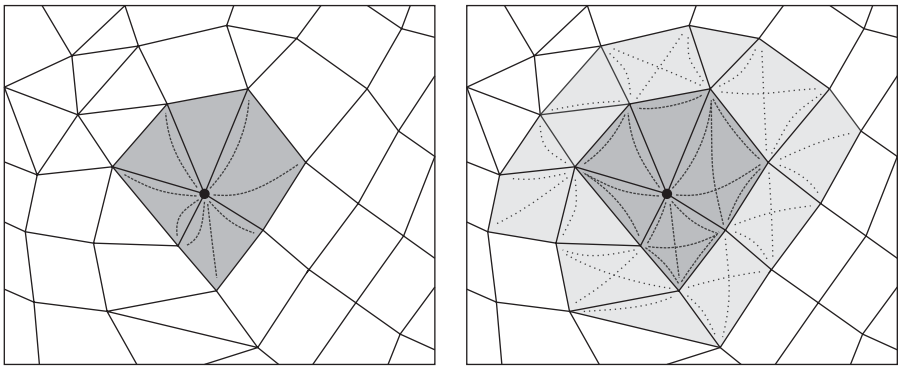


Figure 2. Sparsity pattern: finite element matrix *vs* Jacobian operator.

difference between the sparsity pattern of the finite element matrix and that of the approximate Jacobian operator constructed as explained above.

As demonstrated in [15], the connectivity graph of the Jacobian is known *a priori* and can be directly constructed from that of the stiffness matrix by means of symbolic matrix multiplication. Owing to the matrix fill-in, the amount of memory required to store the much denser Jacobian increases considerably and so does the computational cost of linear algebra operations such as matrix–vector multiplication, construction of an ILU decomposition, etc.

Interestingly enough, it turns out that the antidiffusive Jacobian operator J^* for the semi-implicit FCT algorithm presented in Section 5.2 inherits the sparsity pattern of the finite element matrix. A closer look at Equations (35)–(40) reveals that the initialization of the antidiffusive fluxes $\tilde{f}$ in the first outer iteration ($m=1$) does not rely on the dependent variable $u^{(m)}$. Hence, the explicit estimate for the admissible antidiffusion which is already computed in the residual assembly can also be adopted for the construction of the Jacobian. In essence, the target fluxes (16) which need to be updated in each iteration are constrained such that their magnitude is bounded by that of $\tilde{f}$. Let us emphasize the fact that the correction factors α_{ij} are only defined implicitly and hence will not entail a widening of the matrix stencil. As a result, the Jacobian part J^* can be assembled in a rather efficient way.

As a rule of thumb, the kth column of J^* can be assembled by performing the algorithmic steps (41) and (42) based on the perturbed solution vectors $u+\sigma e_k$ and $u-\sigma e_k$ to evaluate the corresponding fluxes $f^{+*}=f^*(u+\sigma e_k)$ and $f^{-*}=f^*(u-\sigma e_k)$ and scale their difference by 2σ. However, this approach is quite expensive since it does not exploit the sparsity of the Jacobian matrix which is inherited from the global evolution operator A. In order to circumvent this problem, let us introduce an efficient algorithm for assembling the operator J^* for the semi-implicit FCT limiter in an edge-based fashion:

- At each outer iteration ($m=1,2,\ldots$), initialize J^* by zero and rebuild it in a loop over edges ij. This process involves the following steps to be performed for $k=i$ and j

  1. Evaluate the explicit antidiffusive contribution and the solution difference:

$$f_{ij}^n=[m_{ij}-(1-\theta)\Delta t d_{ij}^n](u_i^n-u_j^n), \quad \Delta u_{ij}^{(m)}=u_i^{(m)}-u_j^{(m)} \tag{59}$$

DOI: 10.1002/fld

2. Compute auxiliary coefficients using the perturbed solution $u^{(m)} \pm \sigma e_k$:

$$z^{+}_{ij,k} = m_{ij} + \theta \Delta t d_{ij}(u^{(m)} + \sigma e_k), \quad z^{-}_{ij,k} = m_{ij} + \theta \Delta t d_{ij}(u^{(m)} - \sigma e_k) \tag{60}$$

3. Update the perturbed target fluxes (16) depending on the index k:

$$f^{+}_{ij,k} = \begin{cases} z^{+}_{ij,k}[\Delta u^{(m)}_{ij} + \sigma] + f^{n}_{ij} & \text{if } k = i \\ z^{+}_{ij,k}[\Delta u^{(m)}_{ij} - \sigma] + f^{n}_{ij} & \text{otherwise} \end{cases} \tag{61}$$

$$f^{-}_{ij,k} = \begin{cases} z^{-}_{ij}[\Delta u^{(m)}_{ij} - \sigma] + f^{n}_{ij} & \text{if } k = i \\ z^{-}_{ij}[\Delta u^{(m)}_{ij} + \sigma] + f^{n}_{ij} & \text{otherwise} \end{cases} \tag{62}$$

4. Constrain each flux $f^{\pm}_{ij,k}$ so that its magnitude is bounded by that of $\tilde{f}_{ij}$:

$$f^{+*}_{ij,k} = \begin{cases} \min\{f^{+}_{ij,k}, \max\{0, \tilde{f}_{ij}\}\} & \text{if } f^{+}_{ij} > 0 \\ \max\{f^{+}_{ij,k}, \min\{0, \tilde{f}_{ij}\}\} & \text{otherwise} \end{cases} \tag{63}$$

$$f^{-*}_{ij,k} = \begin{cases} \min\{f^{-}_{ij,k}, \max\{0, \tilde{f}_{ij}\}\} & \text{if } f^{-}_{ij} > 0 \\ \max\{f^{-}_{ij,k}, \min\{0, \tilde{f}_{ij}\}\} & \text{otherwise} \end{cases} \tag{64}$$

5. Compute the divided difference and insert it into the kth column of the Jacobian:

$$f^{*}_{ij,k} = \frac{1}{2\sigma}(f^{+*}_{ij,k} - f^{-*}_{ij,k}), \quad J^{*}_{ik} := J^{*}_{ik} - f^{*}_{ij,k}, \quad J^{*}_{jk} := J^{*}_{jk} + f^{*}_{ij,k} \tag{65}$$

The last three steps call for further explanation. Following expression (41), the implicit contribution of the target fluxes (61)–(62) is multiplied by the perturbed solution difference $(u \pm \sigma e_k)_i - (u \pm \sigma e_k)_j$, whereby k equals i or j. This yields four possible combinations $u_i \pm \sigma \delta_{ik} - u_j \mp \sigma \delta_{jk}$ for $j \neq i$ which need to be multiplied by the coefficients $z^{\pm}_{ij,k}$. The magnitude of the raw antidiffusive fluxes $f^{\pm}_{ij,k}$ is bounded by that of the explicit estimate $\tilde{f}_{ij}$. In the last step, the central difference of the limited fluxes is inserted into the kth column of the Jacobian matrix. Following step (43) of the original algorithm, node j receives the same flux as node i but with opposite sign. Note that the antidiffusive fluxes are now applied to the left-hand side of (65) so that the signs for nodes i and j are reversed.

The above algorithm is applicable to linear and nonlinear transport operators alike. It is worth mentioning that in the linear case the artificial diffusion coefficient d_{ij} does not depend on the solution so that the auxiliary quantity $z_{ij} \equiv m_{ij} + \theta \Delta t d_{ij}$ is not affected by solution variations. Moreover, the perturbed fluxes exhibit the following symmetry property:

$$f^{+}_{ij,i} = z_{ij}[\Delta u^{(m)}_{ij} + \sigma] + f^{n}_{ij} = f^{-}_{ij,j} \tag{66}$$

$$f^{-}_{ij,i} = z_{ij}[\Delta u^{(m)}_{ij} - \sigma] + f^{n}_{ij} = f^{+}_{ij,j} \tag{67}$$

DOI: 10.1002/fld

As a consequence, the final flux which is inserted into the ith column is also applied to column number j but with opposite sign. That is, the following skew symmetry holds:

$$f_{ij,i}^{*}=\frac{1}{2\sigma}(f_{ij,i}^{+*}-f_{ij,i}^{-*})=-f_{ij,j}^{*} \tag{68}$$

Hence, it suffices to compute the divided difference (65) only for one index, say $k=i$, and update the four coefficients of the Jacobian matrix simultaneously according to

$$\begin{aligned} J_{ii}^{*} &:= J_{ii}^{*}-f_{ij,i}^{*}, \quad J_{ji}^{*}:=J_{ji}^{*}+f_{ij,i}^{*} \\ J_{jj}^{*} &:= J_{jj}^{*}-f_{ij,i}^{*}, \quad J_{ij}^{*}:=J_{ij}^{*}+f_{ij,i}^{*} \end{aligned} \tag{69}$$

Roughly speaking, the calculation of the operator J^{*} is approximately twice as expensive as augmenting the right-hand side of (20) by the antidiffusive fluxes making use of the semi-explicit FCT algorithm (41)–(43). As we are about to see, this extra cost clearly pays off in terms of total efficiency when it comes to time-accurate simulation of transient flows. Remarkably, this improvement is already observed if the evolution operator $A=M_{\mathrm{L}}-\theta\Delta t L$ is constant and can be assembled once and for all at the beginning of the simulation so that the standard defect correction approach (24) does not require further matrix evaluations. The benefits of Newton's method become even more significant if the preconditioner (19) needs to be updated in each outer iteration due to a nonlinear governing equation or a linear but time-dependent velocity field $\mathbf{v}=\mathbf{v}(\mathbf{x},t)$ so that the costs for assembling the operators J and J^{*} may be neglected.

5.5. *Convergence behavior*

A remark is in order regarding the convergence behavior of the fixed-point iteration (17). The converged solution u^{n+1} is supposed to satisfy a nonlinear algebraic system of the form

$$A^{*}u^{n+1}=Bu^{n} \tag{70}$$

where A^{*} is the nonlinear FCT operator which includes some built-in antidiffusion

$$A^{*}u^{n+1}:=Au^{n+1}-f^{*} \tag{71}$$

Clearly, the rate of convergence will depend on $\|A^{*}-C\|$, that is, the approximation property of the preconditioner C. On the one hand, the operator A as defined in (19) is linear and easy to 'invert' because it is an M-matrix. On the other hand, it represents a rather poor approximation to the original Galerkin operator $M_{\mathrm{C}}-\theta\Delta t K$ which is recovered in the limit $\alpha_{ij}\rightarrow 1$. As a result, the convergence of a highly accurate FCT algorithm based on the standard defect correction approach is likely to slow down as the high-order solution is approached.

In light of the above, the lumped-mass version, which is obtained by setting $m_{ij}=0$ in the definition of the raw antidiffusive flux, converges much faster than the one based on the consistent target flux (16). However, mass lumping may have a devastating effect on the accuracy of a time-dependent solution, as demonstrated by the numerical study performed in the next section. At the same time, the high phase accuracy provided by the consistent-mass matrix comes at the cost of slower convergence, due to the fact that the 'monotone' preconditioner A is based on M_{L} rather than M_{C}. The original high-order system (13) which corresponds to $\alpha_{ij}\equiv 1$ is particularly difficult to solve, even though it is linear (see below). Moreover, the number of outer iterations tends to increase as the mesh is refined.

DOI: 10.1002/fld

In general, there is a trade-off between the accuracy of the numerical solution and convergence of the fixed-point iteration (24). Any modification of the flux limiter which makes it possible to accept more antidiffusion has an adverse effect on the nonlinear convergence rates. Conversely, more diffusive schemes converge much better but their accuracy leaves a lot to be desired. To overcome this shortcoming, the use of the discrete Newton method is advisable. The number of outer iterations required to drive the residual to some prescribed tolerance is drastically reduced and becomes largely independent of the grid refinement level. However, one should keep in mind that the Jacobian matrix (25) does not possess the M-matrix property so that intermediate solutions are not necessarily positivity-preserving.

6. NUMERICAL EXAMPLES

In order to evaluate the performance of the new algorithm, we apply it to several time-dependent benchmark problems discretized using the standard Galerkin method and the second-order accurate Crank–Nicolson time stepping. After flux limiting, the order of approximation (in space and time) may vary depending on the local smoothness of the solution. The goal of this numerical study is to examine the accuracy of the resulting high-resolution scheme as well as the convergence behavior of the fixed-point iteration (17) and the implications of mass lumping. To this end, the semi-implicit FCT method (35)–(43) is compared with its semi-explicit prototype (29)–(34) and to the standard Galerkin discretization. Moreover, the standard defect correction scheme (24) and the discrete Newton approach are compared with respect to their nonlinear convergence rates as well as computational efficiency, that is, the total CPU time required to solve the nonlinear algebraic system (14) up to a prescribed tolerance.

6.1. Convection skew to the mesh

In order to study the convergence behavior of the semi-implicit and semi-explicit FEM-FCT algorithms as compared with that of the underlying Galerkin scheme, let us solve Equation (5) with $\mathbf{v}=(1,1)$ so that the initial profile is translated along the diagonal of the computational domain $\Omega=(0,1)\times(0,1)$. The numerical study is to be performed for two different initial configurations centered at the reference point $(x_0, y_0)=(0.3, 0.3)$.

TP1: The first test problem corresponds to the discontinuous initial condition:

$$u(x,y,0)=\begin{cases}1 & \text{if } \max\{|x-x_0|,|y-y_0|\}\leqslant 0.1\\ 0 & \text{otherwise}\end{cases} \tag{72}$$

TP2: The second test problem deals with translation of a smooth function defined as

$$u(x,y,0)=\tfrac{1}{4}[1+\cos(10\pi(x-x_0))][1+\cos(10\pi(y-y_0))] \tag{73}$$

within the circle $\sqrt{(x-x_0)^2+(y-y_0)^2}\leqslant 0.1$ and equal to zero elsewhere.

Figures 3 and 4 display the approximate solutions at time $t=0.5$ computed using $\Delta t=10^{-3}$ on a quadrilateral mesh consisting of 128×128 bilinear elements. The upper diagrams were produced by the consistent-mass semi-implicit FCT algorithm which yields nonoscillatory solutions bounded by 0 and 1. The underlying high-order scheme remains stable but gives rise to nonphysical undershoots and overshoots, as seen in the lower diagrams.

DOI: 10.1002/fld

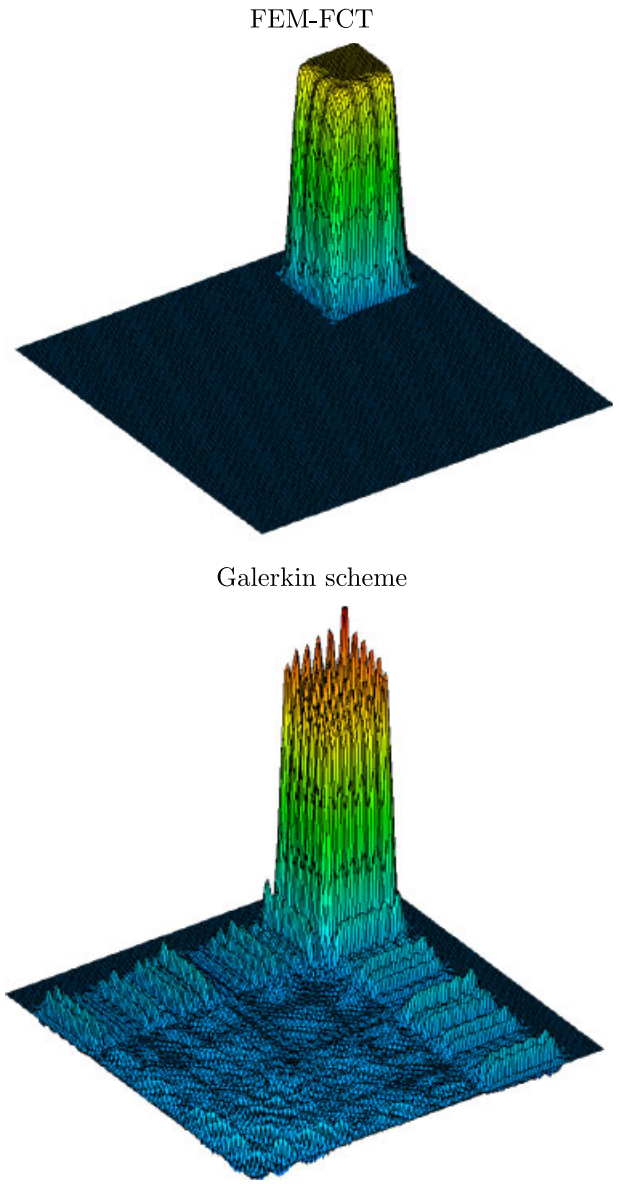


Figure 3. Convection skew to the mesh TP1: 128×128 Q_1-elements, $t = 0.5$.

In either case, the numerical solution was computed in an iterative way using the fixed-point defect correction scheme (17) preconditioned by the low-order operator (19). The stopping criterion was based on the Euclidean norm of the residual vector

$$r = Au^{n+1} - Bu^n - f^*, \quad \|r\| = \sqrt{r^{\mathrm{T}} r} \tag{74}$$

DOI: 10.1002/fld

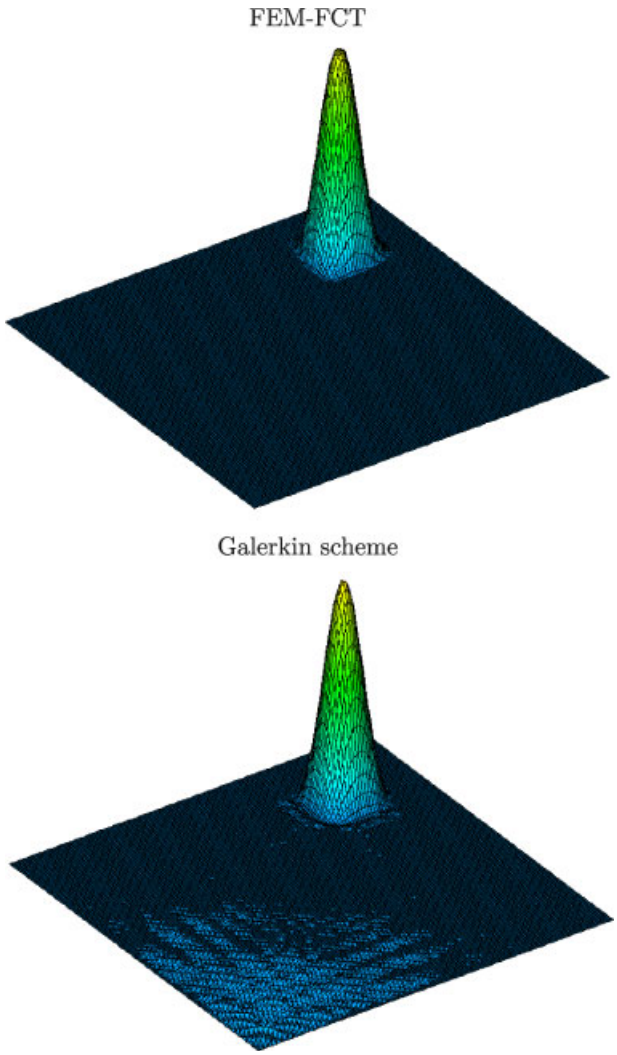


Figure 4. Convection skew to the mesh TP2: 128×128 Q_1-elements, $t=0.5$.

which was required to satisfy the inequality $\|r\|\leqslant10^{-4}$. The difference between the exact solution u and its finite element approximation u_h was measured in the L_1-norm

$$\|u-u_h\|_1=\int_\Omega|u-u_h|\,\mathrm{d}x\approx\sum_i m_i|u(x_i,y_i)-u_i| \tag{75}$$

as well as in the L_2-norm defined by the following formula:

$$\|u-u_h\|_2^2=\int_\Omega|u-u_h|^2\,\mathrm{d}x\approx\sum_i m_i|u(x_i,y_i)-u_i|^2 \tag{76}$$

DOI: 10.1002/fld

Table I. Convection skew to the mesh: convergence behavior for TP1.

Standard defect correction						
NLEV	NVT	NDC	$\|u-u_h\|_1$	$\|u-u_h\|_2$	$u_{\min}$	$u_{\max}$
Semi-implicit FCT/consistent-mass matrix						
6	4225	2500	1.1737e−2	6.2176e−2	0.0	1.0
7	16641	2461	7.3688e−3	4.8577e−2	0.0	1.0
8	66049	2489	4.7039e−3	3.8715e−2	0.0	1.0
Semi-implicit FCT/lumped-mass matrix						
6	4225	751	1.9356e−2	8.4294e−2	0.0	0.9988
7	16641	1000	1.2402e−2	6.5356e−2	0.0	1.0000
8	66049	1014	7.8511e−3	5.1182e−2	0.0	1.0000
Galerkin scheme/consistent-mass matrix						
6	4225	4666	3.6283e−2	7.4952e−2	−0.2557	1.4505
7	16641	7379	2.7340e−2	5.8124e−2	−0.2743	1.3797
8	66049	13852	2.3000e−2	5.2536e−2	−0.4437	1.4080
Galerkin scheme/lumped-mass matrix						
6	4225	1000	6.5181e−2	1.3073e−1	−0.4022	1.5608
7	16641	1423	4.7055e−2	9.8663e−2	−0.4340	1.5732
8	66049	1500	3.5126e−2	8.0298e−2	−0.3713	1.5628
Semi-explicit FCT/consistent-mass matrix						
6	4225	3190	9.3328e−3	5.4115e−2	0.0	1.0
7	16641	3220	5.4794e−3	4.1218e−2	0.0	1.0
8	66049	3590	3.3680e−3	3.2369e−2	0.0	1.0
Semi-explicit FCT/lumped-mass matrix						
6	4225	1500	1.9098e−2	8.3498e−2	0.0	0.9989
7	16641	1501	1.2422e−2	6.5348e−2	0.0	1.0000
8	66049	1540	7.8662e−3	5.1167e−2	0.0	1.0000

where $m_i = \int_\Omega \varphi_i \, \mathrm{d}x$ are the diagonal coefficients of the row-sum lumped-mass matrix. Furthermore, the global minimum $u_{\min} = \min_i u_i$ and maximum $u_{\max} = \max_i u_i$ of the discrete solution u_h were compared with their analytical values 0 and 1.

Tables I and II illustrate the convergence behavior of the iterative flux/defect correction scheme as applied to the test problems TP1 and TP2 on three successively refined meshes. The first three columns in each table display the refinement level NLEV, the number of vertices/nodes NVT, and the total number of outer iterations NDC required to compute the numerical solution at $t = 0.5$. The different performances of the six algorithms under consideration support the arguments presented in Section 5.5. In particular, it can readily be seen that the use of the consistent-mass matrix results in a much better accuracy but the convergence slows down, whereas the lumped-mass version is less accurate but much more efficient. If the difference $\|u^{n+1} - u^n\|$ is large, mass antidiffusion affects the convergence rates even stronger than the convective part of the antidiffusive flux. Since the latter is proportional to Δt, the mass lumping error plays a dominant role at small time steps such that $A \approx M_L$. On the other hand, the linear convergence rates improve since the condition number of A decreases and its diagonal dominance is enhanced as the time step is refined.

Note that the consistent-mass Galerkin scheme faces severe convergence problems and the error may even increase in the course of mesh refinement (see Table II). By contrast, the results

DOI: 10.1002/fld

Table II. Convection skew to the mesh: convergence behavior for TP2.

Standard defect correction						
NLEV	NVT	NDC	$\|u-u_h\|_1$	$\|u-u_h\|_2$	$u_{\min}$	$u_{\max}$
Semi-implicit FCT/consistent-mass matrix						
6	4225	2486	1.4799e−3	9.2813e−3	0.0	0.8562
7	16641	1833	4.3436e−4	2.7820e−3	0.0	0.9418
8	66049	2867	1.7887e−4	1.2032e−3	0.0	0.9740
Semi-implicit FCT/lumped-mass matrix						
6	4225	1000	4.2704e−3	2.7827e−2	0.0	0.7308
7	16641	1000	1.7834e−3	1.1294e−2	0.0	0.9218
8	66049	736	7.6982e−4	4.6142e−3	0.0	0.9612
Galerkin scheme/consistent-mass matrix						
6	4225	2500	1.3961e−3	2.6234e−3	−0.0158	0.9890
7	16641	6437	1.8892e−3	3.9001e−3	−0.0480	0.9925
8	66049	13700	2.3237e−3	8.1553e−3	−0.1363	1.0012
Galerkin scheme/lumped-mass matrix						
6	4225	1000	1.0904e−2	4.2409e−2	−0.1911	0.8809
7	16641	1000	3.4837e−3	1.4234e−2	−0.0811	1.0098
8	66049	1000	1.3092e−3	4.3179e−3	−0.0322	1.0046
Semi-explicit FCT/consistent-mass matrix						
6	4225	2651	1.0770e−3	7.6799e−3	0.0	0.8555
7	16641	2328	2.8414e−4	2.1692e−3	0.0	0.9471
8	66049	3434	1.3188e−4	9.8597e−4	0.0	0.9775
Semi-explicit FCT/lumped-mass matrix						
6	4225	1500	4.2671e−3	2.7760e−2	0.0	0.7296
7	16641	1500	1.7751e−3	1.1237e−2	0.0	0.9211
8	66049	1500	6.4767e−4	3.8591e−3	0.0	0.9653

computed by the semi-implicit FCT algorithm exhibit monotone grid convergence as well as some improvement of the convergence rates. Even the consistent-mass version converges slowly but surely to a nonoscillatory time-accurate solution. For large time steps, the single-step implementation based on (44)–(45) would be more diffusive and converge faster. However, for time steps as small as the one employed in this section, it would be just as accurate and converge at the same rate as algorithm (35)–(43). The values of $u_{\max}$ in Table II reveal that flux correction may lead to undesirable 'peak clipping', which is a well-known phenomenon discussed, e.g. in [2, 6]. On the other hand, the associated high-order solution is corrupted by undershoots and overshoots which are particularly large for discontinuous initial data (Table I) and less pronounced for the smooth cosine hill (Table II). These nonphysical oscillations result in a dramatic loss of accuracy and slow/no convergence, so that the results are inferior to those produced by the semi-implicit FCT algorithm using the same settings.

It is not unusual that semi-explicit flux correction (29)–(34) as applied at the end of each time step to the converged high-order predictor requires less outer iterations than the underlying Galerkin scheme (see Tables I and II). However, the residual of the flux-corrected solution can no longer be controlled and the total number of defect correction steps is considerably greater than that for the semi-implicit FCT limiter, whereas the accuracy of the resulting solutions is comparable. Of

DOI: 10.1002/fld

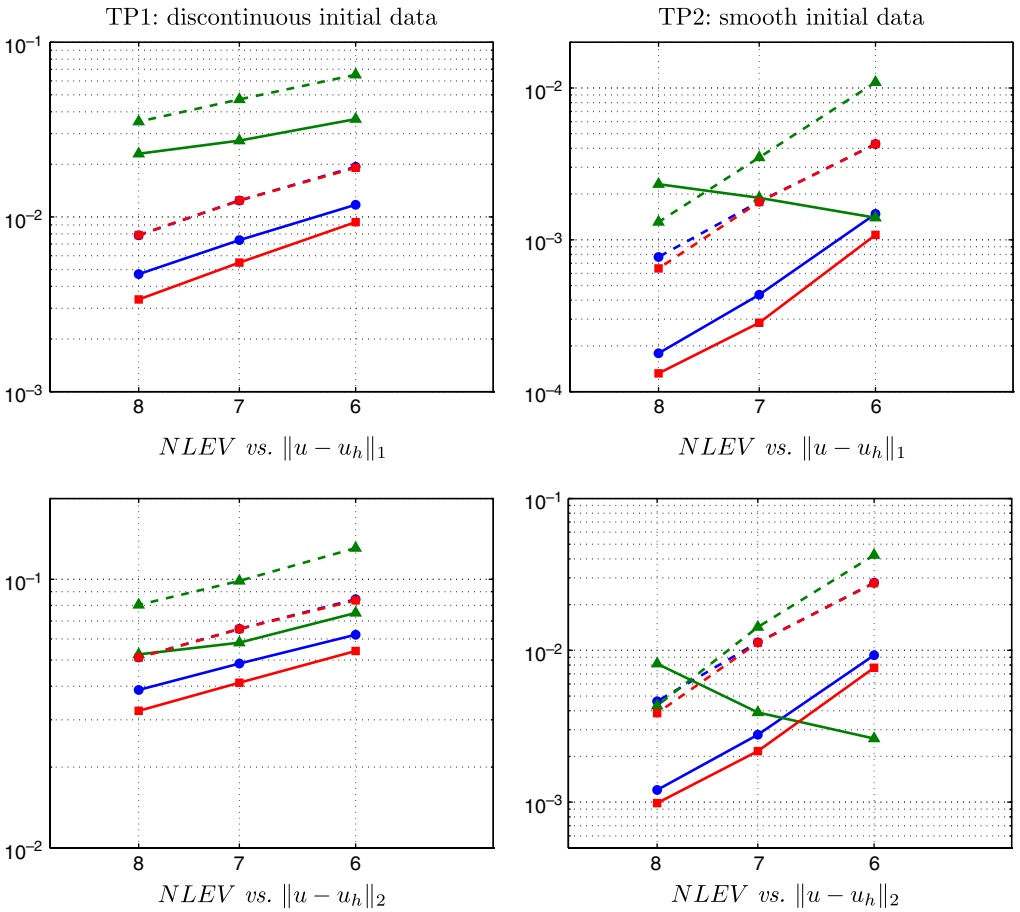


Figure 5. Convection skew to the mesh: error reduction.

course, the linear system (13) could be solved in one step (without resorting to defect correction) but this straightforward approach would inevitably lead to a severe deterioration of the linear convergence rates. Indeed, the high-order operator $M_C - \theta \Delta t K$ is much harder to 'invert' than the preconditioner A which enjoys the M-matrix property. In many cases, the high-order solution may prove prohibitively expensive or even impossible to compute in such a brute-force way, unless a direct solver is employed. Hence, even linear high-order systems of the form (14) call for the use of iterative defect correction.

In order to obtain a better insight into the error reduction rate, Figure 5 displays the L_1-errors (top) and L_2-errors (bottom) of all six methods for both benchmark configurations. For each discretization, the solid line denotes the consistent-mass matrix, whereas the 'lumped' version is indicated by dashed lines. Obviously, the rate of convergence is the same for the implicit (circular markers) and explicit (square markers) FCT algorithm, whereby the norm of the error is slightly smaller for the latter one if the consistent-mass matrix is adopted. Interestingly enough, both FCT algorithms produce nearly the same results if mass lumping is performed. On the other hand,

DOI: 10.1002/fld

Table III. TP3: semi-implicit FCT with consistent-mass matrix, defect correction.

NLEV	NVT	CPU	NN	NN/Δt	NL	NL/NN	$u_{\min}$	$u_{\max}$
$\|r\| \leqslant 10^{-4}$								
5	1089	6	2500	1.0	2500	1.0	0.0	1.0
6	4225	23	2500	1.0	5000	2.0	0.0	1.0
7	16641	95	2500	1.0	5000	2.0	0.0	1.0
8	66049	422	2500	1.0	5000	2.0	0.0	1.0
$\|r\| \leqslant 10^{-8}$								
5	1089	52	46809	18.72	46820	1.0	0.0	1.0
6	4225	200	47046	18.82	49546	1.05	0.0	1.0
7	16641	896	44827	17.93	51858	1.17	0.0	1.0
8	66049	4212	43405	17.36	63425	1.46	0.0	1.0
$\|r\| \leqslant 10^{-12}$								
5	1089	148	142586	57.03	142597	1.0	0.0	1.0
6	4225	676	160254	64.10	162754	1.01	0.0	1.0
7	16641	3311	177602	71.04	184550	1.04	0.0	1.0
8	66049	15875	192895	77.16	212495	1.10	0.0	1.0

the solution produced by the high-order Galerkin scheme denoted by triangular markers is less accurate which manifests itself in greater error norms. Moreover, it suffers from severe convergence problems if the consistent-mass matrix is adopted and fails completely for the second test problem if the mesh is successively refined.

The marginally better accuracy of the semi-explicit FEM-FCT scheme as compared with its semi-implicit counterpart can be attributed to the better phase characteristics of the high-order Galerkin scheme employed at the predictor step. On the other hand, the involved splitting error may become pronounced in other settings, especially as the time step is increased. Moreover, the linear and/or nonlinear convergence rates leave a lot to be desired so that the semi-implicit approach combined with the discrete Newton method is preferable in many cases.

6.2. *Swirling flow*

Let us proceed to another two-dimensional benchmark problem proposed by LeVeque [26]. It deals with a swirling deformation of initial data by the incompressible velocity field given by

$$v_x = \sin^2(\pi x)\sin(2\pi y)g(t), \quad v_y = -\sin^2(\pi y)\sin(2\pi x)g(t)$$

The initial condition to be prescribed is a discontinuous function of the spatial coordinates which equals unity within a circular sector of $\pi/2$ radians and zero elsewhere:

$$u(x, y, 0) = \begin{cases} 1 & \text{if } (x-1)^2+(y-1)^2<0.8 \\ 0 & \text{otherwise} \end{cases}$$

TP3: For the first test problem, let us employ a constant velocity profile which corresponds to

$$g(t) \equiv 1$$

DOI: 10.1002/fld

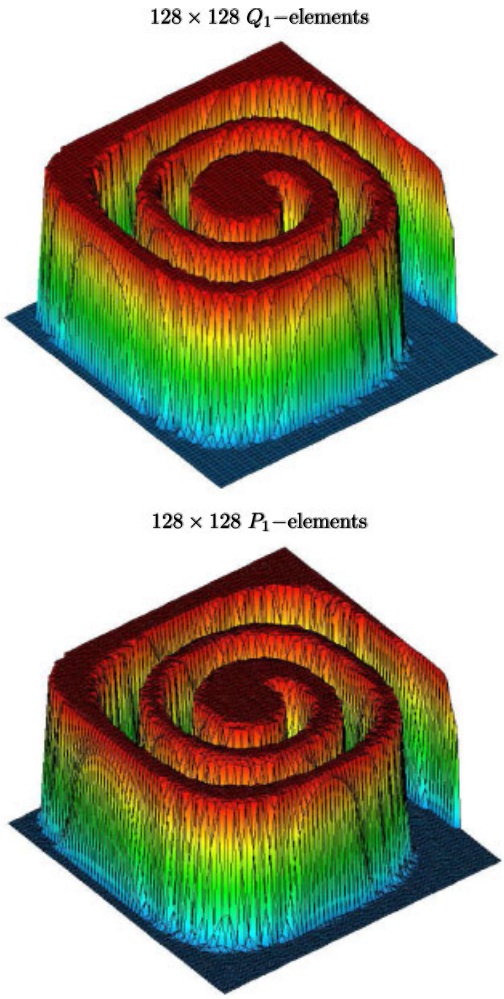


Figure 6. Swirling deformation: semi-implicit FEM-FCT, $t=2.5$.

TP4: For the second test problem, we adopt a more 'agile' velocity field and let

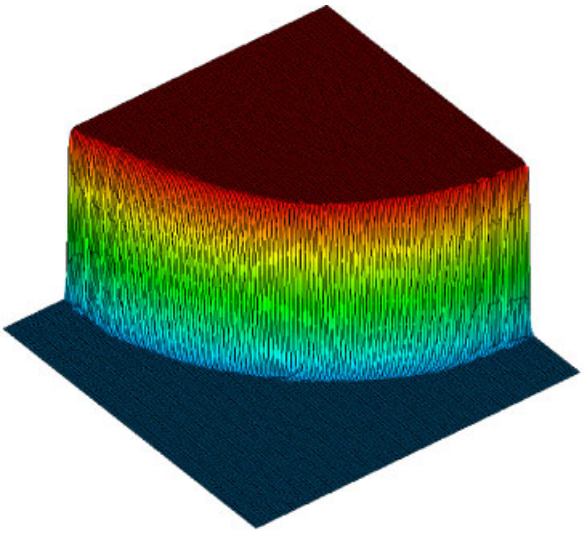
$$g(t)=\cos(\pi t/T), \quad 0\leqslant t\leqslant T$$

For both benchmark configurations, the mass distribution assumes a complex spiral shape in the course of deformation. Figures 6 and 7 display the numerical solutions calculated by the semi-implicit FCT algorithm (35)–(43) with consistent-mass matrix using the time step $\Delta t=10^{-3}$.

Recall that for TP3, the low-order evolution operator A remains constant and can be assembled once and for all at the beginning of the simulation. The numerical results at time $t=2.5$ are computed on a uniform mesh of 128×128 bilinear finite elements and depicted in Figure 6 (top). The use of a piecewise-linear finite element approximation on a triangular mesh with the same number of nodes yields virtually the same solution, see Figure 6 (bottom). For the difference

DOI: 10.1002/fld

initial/exact solution at $t = 1.5$



intermediate solution at $t = 0.75$

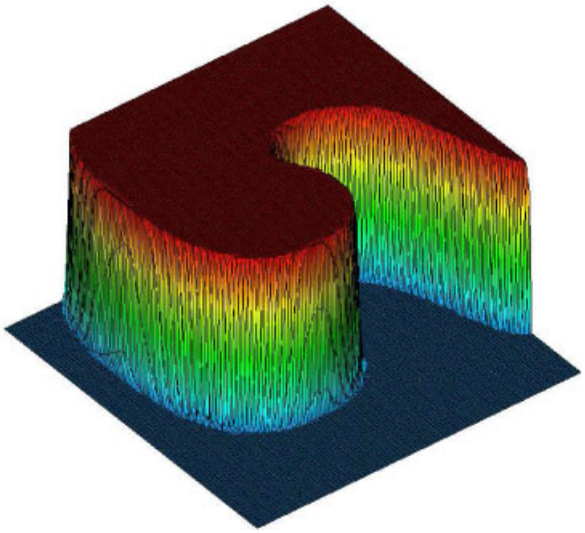


Figure 7. Swirling deformation: semi-implicit FEM-FCT, 128×128 Q_1-elements.

between the underlying triangulations to be visible, both profiles were output on a coarser mesh consisting of 4225 vertices. In either case, the resolution of discontinuities is seen to be remarkably crisp. These results compare well with those presented in [6] using algebraic AFC of TVD type.

On the other hand, the velocity vector is strongly time dependent for benchmark TP4. After the startup, the flow gradually slows down and reverses at $t = T/2$ such that the initial profile

DOI: 10.1002/fld

Table IV. TP3: semi-implicit FCT with consistent-mass matrix, Newton's method, $\eta = 10^{-4}$.

NLEV	NVT	CPU	NN	NN/Δt	NL	NL/NN	$u_{\min}$	$u_{\max}$
$\|r\| \leqslant 10^{-4}$								
5	1089	7	2500	1.0	2500	1.0	−1.789e−02	1.026
6	4225	24	2500	1.0	2500	1.0	−9.153e−03	1.054
7	16641	102	2500	1.0	2500	1.0	−3.906e−02	1.121
8	66049	442	2500	1.0	2500	1.0	−5.087e−02	1.202
$\|r\| \leqslant 10^{-8}$								
5	1089	32	9891	3.96	44547	4.50	−1.319e−11	1.0
6	4225	138	9917	3.97	48379	4.88	−1.430e−09	1.0
7	16641	611	9294	3.72	49464	5.32	−5.427e−12	1.0
8	66049	2736	8974	3.59	50991	5.68	−8.505e−09	1.0
$\|r\| \leqslant 10^{-12}$								
5	1089	84	25640	10.26	123412	4.81	0.0	1.0
6	4225	369	26538	10.62	141645	5.34	0.0	1.0
7	16641	1674	25061	10.02	146212	5.83	0.0	1.0
8	66049	7113	22287	8.91	139868	6.28	0.0	1.0

Table V. TP3: semi-implicit FCT with consistent-mass matrix, Newton's method, $\eta = 10^{-1}$.

NLEV	NVT	CPU	NN	NN/Δt	NL	NL/NN	$u_{\min}$	$u_{\max}$
$\|r\| \leqslant 10^{-4}$								
5	1089	7	2500	1.0	2500	1.0	−1.789e−02	1.026
6	4225	24	2500	1.0	2500	1.0	−9.153e−03	1.054
7	16641	104	2500	1.0	2500	1.0	−3.906e−02	1.121
8	66049	443	2500	1.0	2500	1.0	−5.085e−02	1.202
$\|r\| \leqslant 10^{-8}$								
5	1089	22	10008	4.0	17436	1.74	−4.087e−10	1.0
6	4225	85	9997	4.0	17574	1.76	−1.165e−09	1.0
7	16641	370	9938	3.98	17944	1.81	−9.854e−09	1.0
8	66049	1597	9472	3.79	18005	1.90	−1.797e−07	1.0
$\|r\| \leqslant 10^{-12}$								
5	1089	56	26448	10.75	45770	1.70	0.0	1.0
6	4225	223	27697	11.08	48512	1.75	0.0	1.0
7	16641	939	25922	10.37	49030	1.89	0.0	1.0
8	66049	3787	22540	9.02	47634	2.11	0.0	1.0

is recovered as exact solution at the final time $t = T$, that is, $u(x, y, T) = u(x, y, 0)$. The value $T = 1.5$ is used, which corresponds to performing 1500 time steps of size $\Delta t = 10^{-3}$. The numerical solutions at $t = 0.75$ and 1.5, which are displayed in Figure 7, were calculated on a mesh of 128×128 bilinear finite elements by the semi-implicit FCT algorithm with the consistent-mass matrix. The solution profiles resulting from the application of the lumped-mass matrix are slightly more diffusive but 'look' quite similar.

DOI: 10.1002/fld

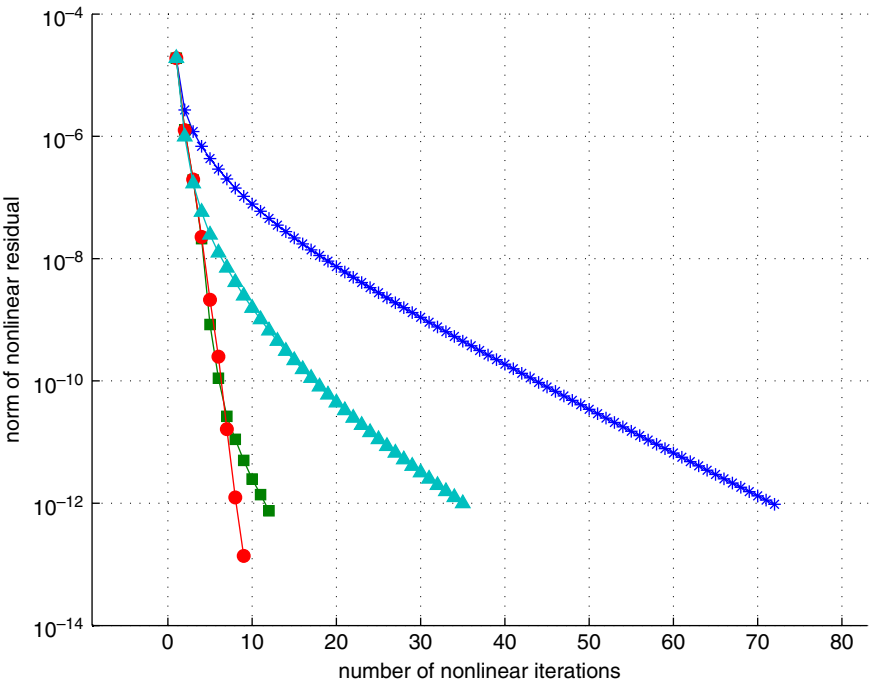


Figure 8. TP3: influence of perturbation parameter σ, $t \in [1.0, 1.0+\Delta t]$.

For these two benchmark configurations, we performed an in-depth convergence study on four successively refined quadrilateral meshes. A detailed comparison between the standard defect correction method and the discrete Newton approach is presented in Tables III–V.

As before, the first two columns display the refinement level NLEV and the number of vertices/nodes NVT. All tests were performed on an Intel Core Duo T2400 (1.83 GHz, FSB 667 MHz) processor with 1024 MB (553 MHz) of system memory. The code was compiled with the Intel Fortran 9.1 Compiler for Linux making use of the `-fast` switch which yields the best results for this setup. The total CPU time (in seconds) required to reduce the norm of the nonlinear residual to the prescribed tolerance in each time step is given in the third column. In the next four columns, the total number of nonlinear iterations (NN), the number of nonlinear iterations per time step (NL/Δt), the total number of linear iterations (NL) and the number of linear iterations per nonlinear iteration (NL/NN) are displayed in successive order. Owing to the lack of an exact solution for this benchmark configuration only the global minimum and maximum of the discrete solution u_h are compared with their analytical values 0 and 1.

It can be seen from Table III that the convergence behavior of the standard defect correction scheme deteriorates significantly if the tolerance for the residual norm is reduced from 10^{-8} to 10^{-12}. Moreover, for the latter, the number of outer iterations increases if the mesh is successively refined. On the other hand, the minimal and maximal solution values perfectly match their analytical bounds 0 and 1 due to the M-matrix property of A.

The convergence behavior of the discrete Newton approach making use of a constant forcing term $\eta = 10^{-4}$ as suggested in [27] is displayed in Table IV. This choice is quite restrictive and requires

DOI: 10.1002/fld

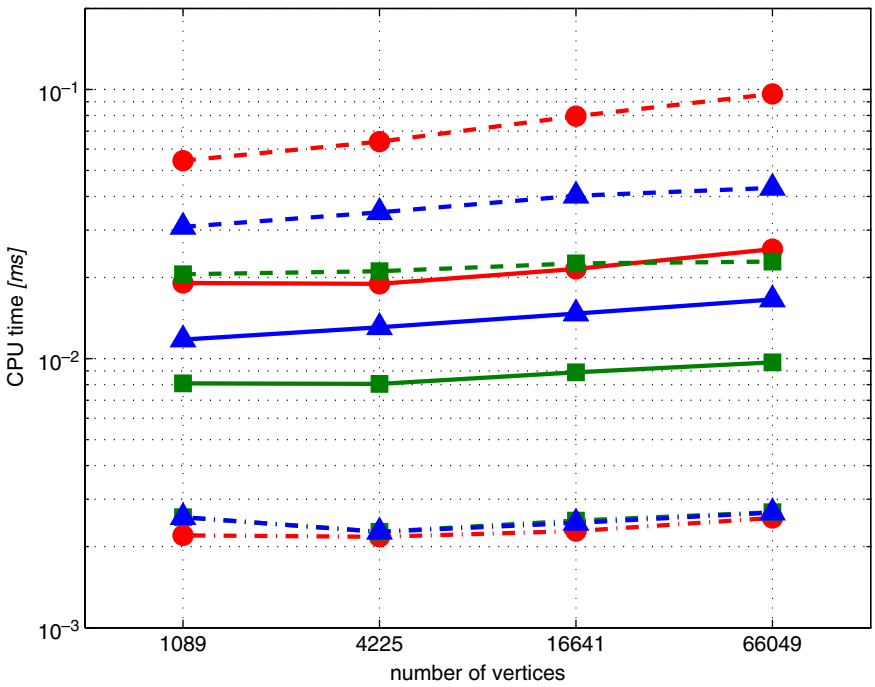


Figure 9. TP3: influence of perturbation parameter σ, $t \in [1.0, 1.0+\Delta t]$.

uniformly close approximations of Newton steps in each nonlinear iteration. It reportedly yields local q-linear convergence in some special norm [18]. As compared with the defect correction approach, the number of outer iterations is drastically reduced for all prescribed tolerances and, in addition, it does not increase for finer grids. On the basis of the moderate number of linear sub-iterations, we believe that the ILU decomposition of the monotone evolution operator A constitutes an appropriate preconditioner for the employed BiCGSTAB algorithm. Importantly, convergence of the fixed-point iteration is a prerequisite for the Newton method to produce a positivity-preserving solution. This is best illustrated by the unsatisfactory minimal and maximal solution values for the loose residual tolerance 10^{-4}.

Let us briefly address the phenomenon of *oversolving* [18] the linear subproblems. To this end, we relax the forcing term $\eta = 10^{-1}$ and leave all other parameters unchanged. The results computed by the discrete Newton method are shown in Table V. The nonlinear convergence behavior is quite similar to that observed for the more restrictive choice $\eta = 10^{-4}$. However, the number of inner iterations is reduced by a factor of 2.5–3, which results in a significant reduction of the overall CPU time. Our experiments with different strategies for choosing the forcing term η adaptively [18] and even solving the linear subproblems directly [28] revealed the fact that the simplest choice $\eta = 10^{-1}$ yields the most competitive results in terms of overall performance for this class of time-dependent flows. On the one hand, the time step $\Delta t = 10^{-3}$ was chosen moderately small to resolve the temporal evolution with high precision. On the other hand, the amount of antidiffusion accepted by the FCT flux limiter is inversely proportional to Δt so that the computed solution profiles become more diffusive if larger time steps are employed. Consequently, the convergence

DOI: 10.1002/fld

Table VI. TP4: semi-implicit FCT, $\|r\| \leqslant 10^{-8}$, $\eta = 10^{-1}$.

NLEV	NVT	CPU	NN	NN/Δt	NL	NL/NN	$\|u-u_h\|_1$	$\|u-u_h\|_2$
Defect correction/consistent-mass matrix								
5	1089	89	24136	16.09	24136	1.0	2.7748e−2	8.8019e−2
6	4225	333	22694	15.13	23488	1.03	1.5630e−2	6.7038e−2
7	16641	1293	19883	13.26	21954	1.10	8.8456e−3	5.0641e−2
8	66049	5203	17959	11.97	22812	1.27	5.1680e−3	3.8747e−2
Defect correction/lumped-mass matrix								
5	1089	17	2720	1.81	2720	1.0	4.5446e−2	1.1689e−1
6	4225	57	2738	1.83	3481	1.27	2.9877e−2	9.4992e−2
7	16641	259	2804	1.87	3818	1.36	1.9192e−2	7.5658e−2
8	66049	1186	2953	1.97	4777	1.62	1.2250e−2	5.9934e−2
Newton's method/consistent-mass matrix								
5	1089	28	5506	3.67	9501	1.73	2.7743e−2	8.8007e−2
6	4225	106	5241	3.94	9074	1.73	1.5624e−2	6.7021e−2
7	16641	442	4831	3.22	8342	1.73	8.8374e−3	5.0609e−2
8	66049	1844	4506	3.0	7802	1.73	5.1604e−3	3.8719e−2
Newton's method/lumped-mass matrix								
5	1089	12	1510	1.01	1510	1.0	4.5441e−2	1.16882e−1
6	4225	42	1513	1.01	1513	1.0	2.9868e−2	9.4975e−2
7	16641	188	1572	1.05	1572	1.0	1.9175e−2	7.5623e−2
8	66049	910	1803	1.20	1803	1.0	1.2231e−2	5.9887e−2

rates of the defect correction method and of the discrete Newton algorithm improve but the solution is smeared by numerical diffusion.

It is well known that choosing an appropriate perturbation parameter σ is a delicate task. In our simulations we employed $\sigma = [(1+\|u\|)\varepsilon]^{1/3}$, where $\varepsilon_{\text{mach}}$ denotes the machine precision, as proposed by Pernice *et al.* and successfully used in the NITSOL package [23]. In order to investigate the influence of this 'free' parameter we repeated the simulation on mesh level 7 for fixed parameter values $\sigma = \varepsilon$ and 0.01, respectively. Figure 8 displays the nonlinear convergence behavior for the different solution strategies. The curve for the defect correction method is marked by stars, whereas circles stand for the rapidly converging Newton method ($\eta = 10^{-1}$, $\sigma = \varepsilon$). Using machine precision as perturbation parameter works for this test case, but it is likely to diverge in other situations due to round-off errors and, hence, not to be recommended in general. The strategy proposed by Pernice *et al.* (square markers) requires slightly more nonlinear steps but turns out to be more robust. Furthermore, the devastating effect of choosing the perturbation parameter too large, e.g. $\sigma = 0.01$, is illustrated by the fourth curve (triangles). If σ is increased even further, the convergence of Newton's method slows down until it resembles that of the defect correction approach.

Another quantity of interest is the computing time per time step spent for each vertex, which is illustrated in Figure 9. Here, the circles correspond to the standard defect correction approach, whereas triangles and squares stand for Newton's method making use of the forcing term $\eta = 10^{-4}$ and 10^{-1}, respectively. The three curves plotted for each method denote the different tolerances for the nonlinear residual. It is worth mentioning that for the least restrictive choice $\eta = 10^{-1}$, the nodal CPU time remains nearly constant if the number of vertices is increased, whereas a systematic growth is observed for both other methods.

DOI: 10.1002/fld

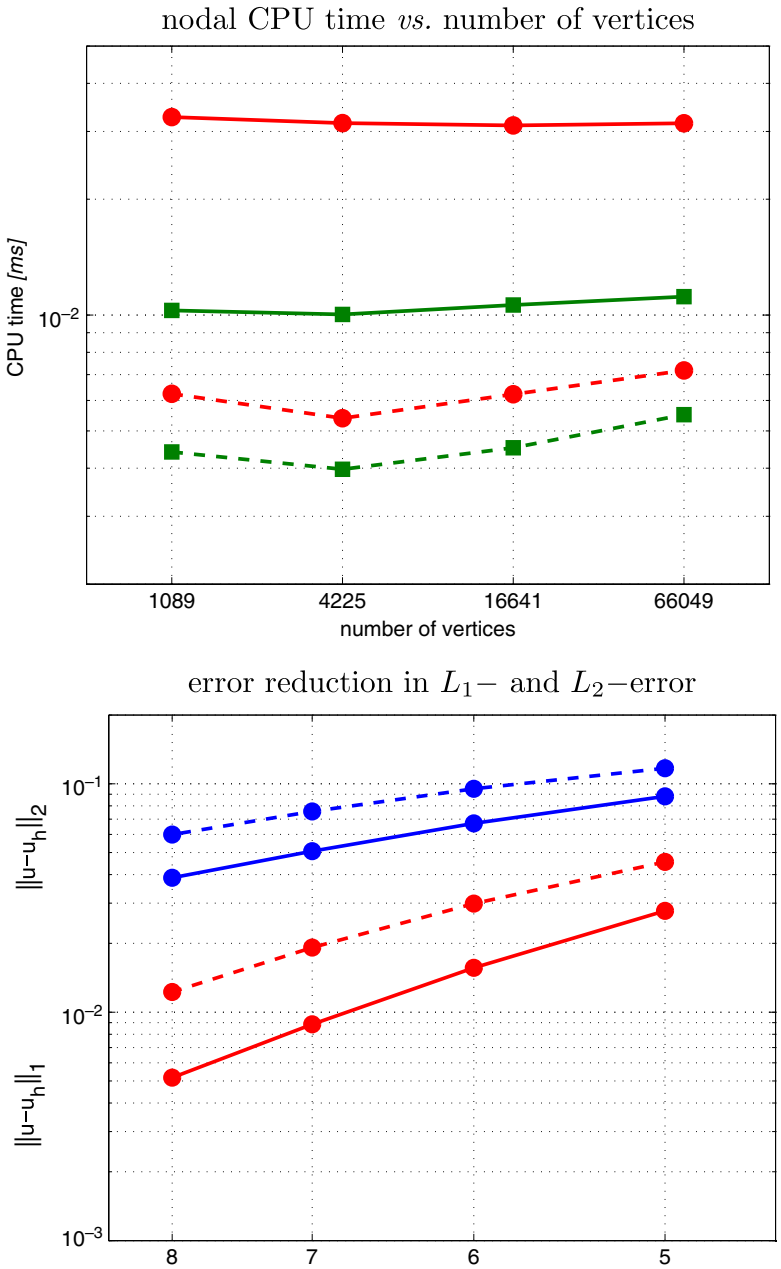


Figure 10. TP4: nodal CPU time/error reduction.

Table VI illustrates the convergence behavior of the different solution methods and the errors of the finite element approximation u_h at time $t=1.5$ for our benchmark configuration TP4. For all computations, a moderate stopping criterion $\|r\|\leqslant 10^{-8}$ was used and the approved forcing strategy

DOI: 10.1002/fld

$\eta = 10^{-1}$ was adopted for Newton's method. Moreover, the perturbation parameter σ was computed as proposed by Pernice and Walker [23] and utilized for the divided difference approximation. All other parameter settings, e.g. the configuration of the linear solver, remain unchanged. It can be readily seen that the discrete Newton approach outperforms the standard defect correction scheme in all situations.

Figure 10 (top) illustrates the CPU time spent per node in each time step which remains nearly constant for all mesh levels. As before, the circular markers correspond to the standard defect correction method, whereas squares are used for the discrete Newton approach. Here, the dashed lines represent the lumped-mass versions of the two algorithms. The significant overhead costs of the slowly converging defect correction method are clearly visible. The solution errors, which are virtually the same for both nonlinear solution strategies, exhibit a monotone reduction on sufficiently fine meshes as illustrated in Figure 10 (bottom).

7. CONCLUSIONS

The semi-implicit approach to flux correction of FCT type leads to a robust and efficient special-purpose algorithm for time-dependent problems discretized in space by the FEM. The accuracy of the resulting scheme improves as the time step is refined and the consistent-mass matrix can be included in a positivity-preserving fashion. The new limiting strategy makes it possible to avoid a repeated computation of the nodal correction factors at each outer iteration. Therefore, the use of an implicit time-stepping method pays off in spite of the CFL-like condition to be satisfied by the time step in the case $\theta < 1$. For sufficiently small time steps, the new algorithm is more accurate and/or efficient than the AFC schemes proposed in [9, 11]. On the other hand, it is not to be recommended for steady-state computations which call for the use of large time steps. In this case, both the limiting strategy and the underlying constraints need to be redefined as explained in [12].

In order to solve the nonlinear algebraic systems, a discrete Newton approach was devised making use of the fact that the underlying sparsity pattern is known *a priori*. The Jacobian matrix was assembled edge by edge using numerical differentiation as applied to the low-order operator and to the vector of limited antidiffusive fluxes. The use of a new semi-implicit limiting strategy makes it possible to assemble the Jacobian matrix in a particularly efficient way, which results in a significant reduction (by a factor of 2.5–3.5) of the total CPU time as compared with standard defect correction. The semi-explicit FCT algorithm was found to provide a slightly better accuracy for the test cases considered in the present paper. However, the high-order system to be solved at the predictor step is extremely ill conditioned, which requires the use of a slowly converging defect correction scheme preconditioned by the low-order operator.

REFERENCES

1. Boris JP, Book DL. Flux-corrected transport. I. SHASTA, a fluid transport algorithm that works. *Journal of Computational Physics* 1973; **11**:38–69.
2. Zalesak ST. Fully multidimensional flux-corrected transport algorithms for fluids. *Journal of Computational Physics* 1979; **31**:335–362.
3. Löhner R, Morgan K, Peraire J, Vahdati M. Finite element flux-corrected transport (FEM-FCT) for the Euler and Navier–Stokes equations. *International Journal for Numerical Methods in Fluids* 1987; **7**:1093–1109.

DOI: 10.1002/fld

4. Löhner R, Baum JD. 30 Years of FCT: status and directions. In *Flux-corrected Transport*: *Principles*, *Algorithms*, *and Applications*, Kuzmin D, Löhner R, Turek S (eds). Springer: Berlin, 2005; 251–296.
5. Book DL. The conception, gestation, birth, and infancy of FCT. In *Flux-corrected Transport*: *Principles*, *Algorithms*, *and Applications*, Kuzmin D, Löhner R, Turek S (eds). Springer: Berlin, 2005; 5–28.
6. Kuzmin D, Möller M. Algebraic flux correction I. Scalar conservation laws. In *Flux-corrected Transport*: *Principles*, *Algorithms*, *and Applications*, Kuzmin D, Löhner R, Turek S (eds). Springer: Berlin, 2005; 155–206.
7. Zalesak ST. The design of flux-corrected transport (FCT) algorithms for structured grids. In *Flux-corrected Transport*: *Principles*, *Algorithms*, *and Applications*, Kuzmin D, Löhner R, Turek S (eds). Springer: Berlin, 2005; 29–78.
8. Kuzmin D. Positive finite element schemes based on the flux-corrected transport procedure. In *Computational Fluid and Solid Mechanics*, Bathe KJ (ed.). Elsevier: Amsterdam, 2001; 887–888.
9. Kuzmin D, Turek S. Flux correction tools for finite elements. *Journal of Computational Physics* 2002; **175**: 525–558.
10. Kuzmin D, Möller M, Turek S. Multidimensional FEM-FCT schemes for arbitrary time-stepping. *International Journal for Numerical Methods in Fluids* 2003; **42**:265–295.
11. Kuzmin D, Möller M, Turek S. High-resolution FEM-FCT schemes for multidimensional conservation laws. *Computer Methods in Applied Mechanics and Engineering* 2004; **193**:4915–4946.
12. Kuzmin D. On the design of general-purpose flux limiters for implicit FEM with a consistent mass matrix. *Journal of Computational Physics* 2006; **219**:513–531.
13. Jameson A. Computational algorithms for aerodynamic analysis and design. *Applied Numerical Mathematics* 1993; **13**:383–422.
14. Jameson A. Positive schemes and shock modelling for compressible flows. *International Journal for Numerical Methods in Fluids* 1995; **20**:743–776.
15. Möller M. Efficient solution techniques for implicit finite element schemes with flux limiters. *International Journal for Numerical Methods in Fluids* 2007; **55**(7):611–635.
16. Jongen T, Marx YP. Design of an unconditionally stable, positive scheme for the K–ε and two-layer turbulence models. *Computers and Fluids* 1997; **26**(5):469–487.
17. Dembo RS, Eisenstat SC, Steihaug T. Inexact Newton methods. *SIAM Journal on Numerical Analysis* 1982; **19**:400–408.
18. Eisenstat SC, Walker HF. Choosing the forcing term in an inexact Newton method. *SIAM Journal on Scientific Computing* 1996; **17**:16–32.
19. Kelley CT. *Iterative Methods for Linear and Nonlinear Equations*. SIAM: Philadelphia, PA, 1995.
20. Meijerink JA, van der Vorst HA. Iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. *Mathematics of Computation* 1977; **31**:148–162.
21. Turek S. *Efficient Solvers for Incompressible Flow Problems*: *An Algorithmic and Computational Approach*. Springer: Berlin, 1999.
22. Nielsen EJ, Anderson WK, Walters RW, Keyes DE. Application of Newton–Krylov methodology to a three-dimensional unstructured Euler code. *Proceedings of the 12th AIAA CFD Conference*, San Diego, CA, June 1995; AIAA Paper 95-1733-CP.
23. Pernice M, Walker HF. NITSOL: a Newton iterative solver for nonlinear systems. *SIAM Journal on Scientific Computing* 1998; **19**:302–318.
24. Knoll DA, Keyes DA. Jacobian-free Newton–Krylov methods: a survey of approaches and applications. *Journal of Computational Physics* 2004; **193**:357–397.
25. Capon PJ, Jimack PK. An inexact Newton method for systems arising from the finite element method. *Applied Mathematics Letters* 1997; **10**(3):9–12.
26. LeVeque RJ. High-resolution conservative algorithms for advection in incompressible flow. *SIAM Journal on Numerical Analysis* 1996; **33**:627–665.
27. Cai XC, Gropp WD, Keyes DE, Tidriri MD. Newton–Krylov–Schwarz methods in CFD. In *Proceedings of the International Workshop on the Navier–Stokes Equations*, *Notes in Numerical Fluid Mechanics*, Rannacher R (ed.). Vieweg: Braunschweig, 1994.
28. Davis TA. Algorithm 832: UMFPACK V4.3—an unsymmetric-pattern multifrontal method. *ACM Transactions on Mathematical Software* 2004; **30**:196–199.

DOI: 10.1002/fld