No. 612 **August 2019**

On the solvability and iterative solution of algebraic flux correction problems for convection-reaction equations

C. Lohmann

ISSN: 2190-1767

On the solvability and iterative solution of algebraic flux correction problems for convection-reaction equations*

Christoph Lohmann†

August 8, 2019

Abstract

New results on the theoretical solvability of nonlinear algebraic flux correction (AFC) problems are presented and a Newton-like solution technique exploiting an efficient computation of the Jacobian is introduced. The AFC methodology is a rather new and unconventional approach to algebraically stabilize finite element discretizations of convection-dominated transport problems in a bound-preserving manner. Besides investigations concerning the theoretical solvability, the development of efficient iterative solvers seems to be one of the most challenging problems. The purpose of this paper is to take the next step to remove such obstacles: For the linear convection-reaction equation, the existence of a unique solution is shown under a mild coercivity condition and some restriction on the limiter. Additionally, the numerical effort for solving such problems is drastically reduced by the use of a highly customized implementation of the Jacobian. The benefit of this approach is illustrated by in-depth numerical studies.

Keywords. finite elements, algebraic flux correction, well-posedness, Newton-like methods, Jacobian

1 Introduction

In the CFD community, it is well-known that the Galerkin discretization of convection-dominated transport problems is highly unstable for continuous finite elements. Particularly in the case of vanishing diffusivity, Galerkin approximations might be polluted by spurious oscillations and unphysical solution values can occur. For that reason, many stabilization techniques have been invented to improve the accuracy of the solution and significantly reduce the amount of overshoots and undershoots. For example, the streamline upwind/Petrov-Galerkin (SUPG) method [BH82] adds an appropriate stabilization term to the Galerkin discretization for penalization of steep gradients in streamline direction. While this approach

*This research was supported by the German Research Association (DFG) under grant KU 1530/23-1.

†Institute of Applied Mathematics (LS III), TU Dortmund University, Vogelpothsweg 87, D-44227 Dortmund, Germany (christoph.lohmann@math.tu-dortmund.de)

might not completely prevent the generation of unphysical function values, the resulting system of equations stays linear and can be solved efficiently.

However, some applications do not allow even small violations of physical bounds and require the use of property-preserving numerical schemes. One representative of this family is the algebraic flux correction (AFC) scheme introduced by Kuzmin [Kuz07]. A nonlinear artificial diffusion operator is incorporated into the residual of the Galerkin scheme to make the validity of discrete maximum principles algebraically provable [Bar+16; Bar+18; Loh19]. Recent years have witnessed an increased interest on this scheme so that several investigations have been conducted for improvements and theoretical analysis of the methodology. For example, more sophisticated limiting techniques were developed to guarantee the linearity preservation property [BB17; Bar+17b; Bar+17a; Kuz+17]. Barrenechea, John, and Knobloch [Bar+16] and Barrenechea et al. [Bar+18] made great progress establishing in theoretical foundations of the AFC methodology for the convection-diffusion-reaction equation. They proved a priori error estimates, showed the validity of local and global discrete maximum principles, and guaranteed the existence of a solution under some mild assumptions. On the other hand, first investigations concerning efficient iterative solution were performed by Möller [Möl07; Möl08] who used finite differences to approximate the Jacobian for Newton-like solution techniques. Badia and Bonilla [BB17] smoothened their non-differentiable scheme to guarantee second order of convergence for their iterative solver. In contrast to this, Jha and John [JJ18; JJ19] analyzed the influence of different preconditioners for solving nonsmooth problems. Even if the total number of iterations can be drastically reduced by using improved preconditioners, more involved iterations may still lead to an increase of the total computational time. For that reason, Jha and John [JJ19] concluded that the simplest Richardson iteration employing a fixed preconditioner might be the most efficient solver for the nonlinear problem.

In this work, the objectives of individual sections are as follows: Section 2 presents the main idea of the AFC methodology for the linear convection-reaction equation and briefly summarizes some statements on its theoretical solvability. In Section 3, we prove the existence of a unique solution for some specific problems. The counterexample presented in Section 4 illustrates why further assumptions might be necessary to extend the result of uniqueness to more practical test cases. After these theoretical investigations, we propose a way to perform efficient assembly of the Jacobian for a specific family of limiters in Section 5. This matrix is well-suited to act as a preconditioner in the iterative solution technique of Section 6. Different limiters are presented in Section 7 and finally analyzed in the numerical studies of Section 8.

2 Definition of AFC scheme

In this work, we present some new results on the algebraic flux correction methodology for hyperbolic equations. The model problem under consideration is given by the steady and linear convection-reaction equation in non-conservative form

$$\mathbf{v} \cdot \operatorname{grad}(u) + cu = f \qquad \text{in } \Omega, \tag{1a}$$

$$u = u_{\text{in}} \qquad \text{on } \Gamma_{\text{in}} \coloneqq \big\{ \mathbf{s} \in \partial\Omega \;\big|\; \mathbf{n}(\mathbf{s}) \cdot \mathbf{v}(\mathbf{s}) < 0 \big\}, \tag{1b}$$

where $u_{\text{in}} : \Gamma_{\text{in}} \to \mathbb{R}$ denotes the inflow boundary data and $\Omega \subset \mathbb{R}^d$, $d \in \{2, 3\}$, is a polyhedral domain with outward normal vector $\mathbf{n} : \partial\Omega \to \mathbb{R}^d$. Furthermore, the velocity field is given by $\mathbf{v} : \bar{\Omega} \to \mathbb{R}^d$ while $c : \Omega \to \mathbb{R}$ and $f : \Omega \to \mathbb{R}$ are the reactivity and the external source term.

For discretization purposes, we assume that $\mathcal{T}_h$ is an admissible triangulation of Ω containing only triangular or quadrilateral elements (tetrahedra or hexahedra in three space dimensions) and a total number of N nodes. The space of (multi-)linear finite elements is denoted by V_h while $\varphi_1, \dots, \varphi_N$ are the commonly used Lagrange basis functions. Based on these definitions, the Galerkin discretization with weakly imposed boundary conditions reads: Find a solution $u_h = \sum_{i=1}^N u_i \varphi_i \in V_h$ s.t.

$$\sum_{j \in \mathcal{N}_i} a_{ij} u_j = g_i \quad \forall i \in \{1, \dots, N\}, \tag{2a}$$

where $\mathcal{N}_i \subseteq \{1, \dots, N\}$ denotes the nodal stencil of index i, i.e.,

$$\mathcal{N}_i \coloneqq \big\{ j \in \{1, \dots, N\} \,\big|\, \operatorname{supp}(\varphi_i) \cap \operatorname{supp}(\varphi_j) \neq \emptyset \big\},$$

while the system matrix $\mathcal{A} = (a_{ij})_{i,j=1}^N$ and the right hand side vector $g = (g_i)_{i=1}^N$ are given by

$$a_{ij} \coloneqq \int_\Omega \varphi_i \mathbf{v} \cdot \operatorname{grad}(\varphi_j) + \int_\Omega c \varphi_i \varphi_j + \int_{\Gamma_{\text{in}}} |\mathbf{v} \cdot \mathbf{n}| \varphi_i \varphi_j \quad \forall i, j \in \{1, \dots, N\}, \tag{2b}$$

$$g_i \coloneqq \int_\Omega f \varphi_i + \int_{\Gamma_{\text{in}}} |\mathbf{v} \cdot \mathbf{n}| \varphi_i u_{\text{in}} \qquad \forall i \in \{1, \dots, N\}. \tag{2c}$$

The finite element approximation corresponding to the solution of (2) is frequently polluted by spurious oscillations. In particular, unphysical solution values might occur when the inflow boundary profile u_{in} is discontinuous. To avoid such drawbacks, the algebraic flux correction methodology was introduced by Kuzmin [Kuz07]. This approach stabilizes the Galerkin method in a way which makes the boundedness of the degrees of freedom provable [Bar+16; Bar+18; Loh19]. The corresponding nonlinear system of equations reads

$$\sum_j a_{ij} u_j - \sum_{j \in \mathcal{N}_i \setminus \{i\}} \big(1 - \alpha_{ij}(u)\big) d_{ij} (u_j - u_i) = g_i \quad \forall i \in \{1, \dots, N\} \tag{3}$$

and possesses the residual $\mathscr{R} : \mathbb{R}^N \to \mathbb{R}^N$

$$\big(\mathscr{R}(v)\big)_i \coloneqq \sum_j a_{ij} v_j - \sum_{j \in \mathcal{N}_i \setminus \{i\}} \big(1 - \alpha_{ij}(v)\big) d_{ij} (v_j - v_i) - g_i \quad \forall i \in \{1, \dots, N\} \quad \forall v \in \mathbb{R}^N. \tag{4}$$

Here, the artificial diffusion operator $\mathcal{D} = (d_{ij})_{i,j=1}^N$ is defined by

$$d_{ij} \coloneqq \begin{cases} \max\{a_{ij}, 0, a_{ji}\} & : j \neq i, \\ -\sum_{k \neq i} d_{ik} & : j = i \end{cases} \quad \forall i, j \in \{1, \dots, N\} \tag{5}$$

and $\alpha_{ij} = \alpha_{ji} : \mathbb{R}^N \to [0, 1]$ are so called correction factors. If these scaling parameters vanish, the scheme coincides with a linear low order method. On the other hand, the Galerkin discretization is recovered when $\alpha_{ij} = 1$ for all $i, j \in \{1, \dots, N\}$, $i \neq j$. Therefore, the correction factors should be chosen as large as possible but at every local extremum they must be equal to zero to guarantee the validity of discrete maximum principles [Loh19]. Examples for the definition of α_{ij} can be found in Section 7 or in the literature mentioned in the introduction.

Based on the results presented by Barrenechea, John, and Knobloch [Bar+16], the existence of a solution to (3) can easily be shown whenever the residual is a continuous function and the following coercivity condition is satisfied [Loh19, Theorem 4.65]:

$$\exists c_0 > 0 : \qquad L^\infty(\Omega) \ni \Lambda \coloneqq c - \tfrac{1}{2}\operatorname{div}(\mathbf{v}) \geqslant c_0 \quad \text{a.e. in } \Omega. \tag{6}$$

Furthermore, Lohmann [Loh19, Theorem 4.68] proved the uniqueness of the solution if the reactive term of (1) is treated in a lumped manner, i.e.,

$$a_{ij} \coloneqq \int_\Omega \varphi_i \mathbf{v}\cdot\operatorname{grad}(\varphi_j) + \delta_{ij}\int_\Omega c\varphi_i + \int_{\Gamma_{\text{in}}} |\mathbf{v}\cdot\mathbf{n}|\varphi_i\varphi_j \quad \forall i,j \in \{1,\dots,N\},$$

and the condition

$$L\max_i\Big(\int_\Omega \big|\operatorname{div}(\mathbf{v}\varphi_i)\big| + \int_{\partial\Omega}\max(0,\mathbf{v}\cdot\mathbf{n})\varphi_i\Big) < \big(\inf_{\mathbf{x}\in\Omega} c(\mathbf{x})\big)\min_i m_i \tag{7}$$

holds for some Lipschitz constant $L > 0$ that satisfies

$$\big|\alpha_{ij}(u)(u_j - u_i) - \alpha_{ij}(\bar u)(\bar u_j - \bar u_i)\big| \leqslant L\|u_k - \bar u\|_\infty \qquad \forall u, \bar u \in \mathbb{R}^N.$$

Therefore, the uniqueness of the solution is guaranteed for a given limiter if the reactivity parameter c is sufficiently large. On the other hand, the result on uniqueness holds true whenever $\inf_{\mathbf{x}\in\Omega} c(\mathbf{x}) > 0$ and the limiter is chosen so that condition (7) is valid. Unfortunately, the positivity of the reactivity parameter is not a consequence of the coercivity condition (6) and results in a further restriction on the coefficients of (1). For that reason, more general statements for the existence of a unique solution are of interest. In the following section, we prove the uniqueness only requiring the validity of the coercivity condition (6), at least for a sufficiently small Lipschitz constant L.

3 Uniqueness of AFC solution for specific problems

Similarly to the analysis presented in [Loh19, Section 4.5.1.1], we henceforth require that the employed limiter function be Lipschitz continuous and satisfy the condition

$$\big|\alpha_{ij}(u)(u_j - u_i) - \alpha_{ij}(\bar u)(\bar u_j - \bar u_i)\big|^2 \leqslant L^2 \sum_{k\in\mathcal{N}_i\cup\mathcal{N}_j} (u_k - \bar u_k)^2 \qquad \forall u, \bar u \in \mathbb{R}^N \tag{8}$$

for some constant $L > 0$. Furthermore, it is assumed that the system matrix $\mathcal{A}$ is positive definite, i.e.,

$$\exists c_1 > 0 : \qquad \sum_{i,j} u_i a_{ij} u_j \geqslant c_1\|u\|_2^2 \quad \forall u \in \mathbb{R}^N. \tag{9}$$

This requirement is met if the coercivity condition (6) holds [Loh19, Remark 3.7] and, hence, can be interpreted as a discrete version of (6). Let us now suppose that $u_h, \bar u_h \in V_h$ are two distinct finite element approximations whose degrees of freedom solve the AFC system (3).

Then we have

$$
\begin{aligned}
c_1 \sum_i (u_i - \bar{u}_i)^2 &\leqslant \sum_{i,j} (u_i - \bar{u}_i) a_{ij} (u_j - \bar{u}_j) \\
&\leqslant \sum_{i,j} (u_i - \bar{u}_i)(a_{ij} - d_{ij})(u_j - \bar{u}_j) \\
&= \sum_i (u_i - \bar{u}_i) \sum_{j \in \mathcal{N}_i \setminus \{i\}} d_{ij} \big(\alpha_{ij}(\bar{u})(\bar{u}_j - \bar{u}_i) - \alpha_{ij}(u)(u_j - u_i) \big) \\
&\leqslant L \sum_i \sum_{j \in \mathcal{N}_i \setminus \{i\}} |u_i - \bar{u}_i| d_{ij} \Big(\sum_{k \in \mathcal{N}_i \cup \mathcal{N}_j} (u_k - \bar{u}_k)^2 \Big)^{\frac{1}{2}} \\
&\leqslant \frac{L}{2} \sum_i \sum_{j \in \mathcal{N}_i \setminus \{i\}} \Big((u_i - \bar{u}_i)^2 d_{ij}^2 + \sum_{k \in \mathcal{N}_i \cup \mathcal{N}_j} (u_k - \bar{u}_k)^2 \Big) \\
&\leqslant \frac{L}{2} \sum_i \Big((u_i - \bar{u}_i)^2 d_{ii}^2 \\
&\qquad + \sum_{j \in \mathcal{N}_i \setminus \{i\}} \big(\sum_{k \in \mathcal{N}_i} (u_k - \bar{u}_k)^2 + \sum_{k \in \mathcal{N}_j} (u_k - \bar{u}_k)^2 \big) \Big) \\
&\leqslant \frac{L}{2} \sum_i \Big((u_i - \bar{u}_i)^2 \big(d_{ii}^2 + 2 \max_j |\mathcal{N}_j|^2 \big) \Big) \\
&\leqslant \frac{L}{2} \Big(\sum_i (u_i - \bar{u}_i)^2 \Big) \Big(\max_j d_{jj}^2 + 2 \max_j |\mathcal{N}_j|^2 \Big)
\end{aligned}
$$

due to the negative semidefiniteness of $\mathcal{D}$ [Loh19, Lemma 4.36]. Therefore, u_h and $\bar{u}_h$ must coincide if

$$
c_1 > \frac{L}{2} \Big(\max_j d_{jj}^2 + 2 \max_j |\mathcal{N}_j|^2 \Big) \tag{10}
$$

and the solution of (3) is unique if L is sufficiently small. Obviously, this condition might be very restrictive and not meaningful for practical test problems. However, it guarantees the uniqueness of the solution without further requirements on the reactivity parameter c.

4 Example of ill-posedness

In the previous section, the existence of a unique solution is proved if the system matrix is positive definite and the Lipschitz constant of the limiter is sufficiently small. In what follows, we see that this result cannot be readily extended to arbitrarily chosen limiters without further requirements on the problem at hand. For this purpose, let us consider the following model problem in one space dimension:

$$
v u' = f \quad \text{in } (0,1), \qquad u(0) = u_{\text{in}}, \tag{11}
$$

where $v > 0$ denotes a constant velocity field while $f : (0,1) \to \mathbb{R}$ and $u_{\text{in}} \in \mathbb{R}$ are an external source term and the inflow boundary data, respectively. A variational formulation with weakly imposed boundary conditions reads

$$
\int_0^1 v \varphi u' + v \varphi(0) u(0) = \int_0^1 \varphi f + v \varphi(0) u_{\text{in}} \quad \forall \varphi.
$$

Let us now discretize the domain by one element so that V_h is the space of globally linear functions on $[0,1]$ with two degrees of freedom. Then the Galerkin discretization leads to the linear system of equations $\mathcal{A}u = g$, where

$$\mathcal{A} = \frac{1}{2}\begin{pmatrix} v & v \\ -v & v \end{pmatrix} \in \mathbb{R}^{2\times 2}, \qquad g = \begin{pmatrix} g_1 \\ g_2 \end{pmatrix} \in \mathbb{R}^2.$$

Even if the coercivity condition (6) does not hold, the system matrix is positive definite due to

$$u^\top \mathcal{A}u = \tfrac{v}{2}(u_1^2 + u_1u_2 - u_1u_2 + u_2^2) = \tfrac{v}{2}(u_1^2 + u_2^2) = \tfrac{v}{2}\|u\|_2^2 \quad \forall u \in \mathbb{R}^2.$$

The nonlinear AFC problem corresponding to the Galerkin discretization is given by

$$\tfrac{v}{2}u_1 + \tfrac{v}{2}u_2 - (1-\alpha_{12})\tfrac{v}{2}(u_2 - u_1) = g_1, \tag{12a}$$
$$-\tfrac{v}{2}u_1 + \tfrac{v}{2}u_2 - (1-\alpha_{12})\tfrac{v}{2}(u_1 - u_2) = g_2, \tag{12b}$$

where $\alpha_{12} = \alpha_{12}(u_1,u_2) \in [0,1]$ is the only correction factor of this system.

Let us now assume that $g_2 > 0$ and the correction factor is defined in an unconventional way by

$$\alpha_{12} \coloneqq \begin{cases} \min\Big(1, 2\frac{\max\big(0,\max(u_1,u_2)-u_1-|v^{-1}g_2|\big)}{|u_2-u_1|}\Big) & : u_1 \neq u_2, \\ 0 & : u_1 = u_2, \end{cases}$$

then the vector $u = (u_1,u_2)^\top \in \mathbb{R}^2$ with components

$$u_1 \coloneqq v^{-1}(g_1+g_2) - \frac{2g_2}{v(2-\gamma)}, \qquad u_2 \coloneqq v^{-1}(g_1+g_2) \tag{13}$$

solves (12) no matter how $\gamma \in [0,1]$ is chosen. To prove this statement, we first notice that

$$u_2 - u_1 = \frac{2g_2}{v(2-\gamma)} \geqslant \frac{g_2}{v} > 0,$$

which can be exploited to show

$$\begin{aligned} \max(u_1,u_2) - u_1 - |v^{-1}g_2| = u_2 - u_1 - v^{-1}g_2 &= \frac{2g_2}{v(2-\gamma)} - \frac{g_2}{v} \\ &= \frac{2g_2 - g_2(2-\gamma)}{v(2-\gamma)} = \frac{g_2\gamma}{v(2-\gamma)} = \frac{\gamma}{2}(u_2-u_1). \end{aligned}$$

It follows that

$$0 \leqslant \frac{\max(u_1,u_2) - u_1 - |v^{-1}g_2|}{|u_2-u_1|} \leqslant \frac{1}{2} \quad \forall \gamma \in [0,1]$$

and, hence, the correction factor satisfies

$$\alpha_{12} = \gamma \quad \forall \gamma \in [0,1].$$

Exploiting this result, we deduce that

$$\begin{aligned}
\tfrac{v}{2}u_1 + \tfrac{v}{2}u_2 - (1-\alpha_{12})\tfrac{v}{2}(u_2 - u_1) &= \tfrac{v}{2}\big(u_1 + u_2 - (1-\gamma)(u_2 - u_1)\big) \\
&= \tfrac{v}{2}\big(2u_1 + \gamma(u_2 - u_1)\big) \\
&= g_1 + g_2 - \frac{2g_2}{2-\gamma} + \frac{\gamma g_2}{2-\gamma} = g_1, \\
-\tfrac{v}{2}u_1 + \tfrac{v}{2}u_2 - (1-\alpha_{12})\tfrac{v}{2}(u_1 - u_2) &= \tfrac{v}{2}\big(-u_1 + u_2 - (1-\gamma)(u_1 - u_2)\big) \\
&= \tfrac{v}{2}(\gamma - 2)(u_1 - u_2) = \tfrac{v}{2}(\gamma-2)\frac{2g_2}{v(\gamma-2)} = g_2
\end{aligned}$$

and the solution defined by (13) solves (12) no matter how $\gamma \in [0, 1]$ is chosen. In particular, the solution is not unique even if the application of α_{12} is Lipschitz continuous by [Bar+16, Lemma 6] and the system matrix is positive definite.

5 Jacobian of AFC residual

In the following sections, we focus on the numerical solution of the nonlinear system of equations. Unfortunately, the residual of the AFC system (3) is not only highly nonlinear, but might also be non-differentiable so that the standard version of Newton's method may not be applicable. Badia and Bonilla [BB17] tackled this problem by introducing small parameters to smoothen the correction factors in an appropriate manner. The residual of the so-defined regularized AFC system is twice continuously differentiable and a quadratic rate of convergence can be expected for Newton's method. Due to the regularization of the scaling factors, the method is only approximately linearity preserving while discrete maximum principles hold no matter how the adjustable parameters are chosen.

Even if the AFC residual is a twice continuously differentiable function, the application of Newton's method can still be costly: In each iteration, a linear system of equations with the Jacobian has to be solved. By definition of the correction factors, the sparsity pattern of this matrix is more densely occupied than that of the system matrix. This fill-in results in more expensive solution algorithms. Additionally, the entries of the Jacobian depend on the current iterate and have to be recalculated in each iteration of Newton's method. Due to the extended stencil of the Jacobian's connectivity graph, several nested loops might be necessary to compute all nonzero entries.

In this section, we exploit a special structure of frequently used correction factors to derive a convenient decomposition of the Jacobian. In fact, this matrix can be written in terms of auxiliary matrices with the same sparsity pattern as the system matrix. Therefore, the application of the Jacobian is equivalent to a few matrix-vector multiplications and, hence, can be performed very efficiently. The factorized representation can also be used to explicitly compute its entries.

5.1 Efficient implementation

Let the correction factors of the stationary AFC system be defined by

$$\alpha_{ij} = \beta_{ij}\beta_{ji} \quad \forall i, j \in \{1, \dots, N\}, j \neq i, \tag{14}$$

where $\beta_{ij} : \mathcal{N}_i \to [0,1]$ are differentiable functions that depend only on the degrees of freedom in the direct neighborhood of node i. Furthermore, we assume that β_{ij} can be either written as

$$\beta_{ij} = \beta_i \quad \forall i, j \in \{1, \dots, N\}, j \neq i \tag{15a}$$

or satisfies the 'upwinding condition' [Loh19; Kno17]

$$\beta_{ij} = \begin{cases} 1 & : a_{ij} \leqslant 0, \\ \beta_i & : a_{ij} > 0 \end{cases} \quad \forall i, j \in \{1, \dots, N\}, j \neq i \tag{15b}$$

for some nodal correction factors $\beta_i : \mathcal{N}_i \to [0,1]$, $i \in \{1, \dots, N\}$. If the auxiliary quantities β_{ij} are defined by (15a), we set

$$\hat{d}_{ij} = d_{ij} \quad \forall i, j \in \{1, \dots, N\}, j \neq i.$$

Otherwise, we introduce

$$\hat{d}_{ij} = \begin{cases} 0 & : a_{ij} < 0, \\ d_{ij} & : a_{ij} \geqslant 0 \end{cases} \quad \forall i, j \in \{1, \dots, N\}, j \neq i.$$

Then the Jacobian $\mathcal{J} \in \mathbb{R}^{N \times N}$ of the residual $\mathscr{R}$ satisfies

$$\begin{aligned}
&\mathcal{J}v - \sum_i \sum_{j \in \mathcal{N}_i} e_i (a_{ij} - d_{ij}) v_j - \sum_i e_i \sum_{j \in \mathcal{N}_i \setminus \{i\}} d_{ij} \alpha_{ij} (v_j - v_i) \\
&\quad = \sum_i e_i \sum_k \partial_{u_k} \Big(\sum_{j \in \mathcal{N}_i \setminus \{i\}} d_{ij} \beta_{ij} \beta_{ji} (u_j - u_i) \Big) v_k \\
&\qquad - \sum_i e_i \sum_{j \in \mathcal{N}_i \setminus \{i\}} d_{ij} \beta_{ij} \beta_{ji} (v_j - v_i) \\
&\quad = \sum_i e_i \sum_k \sum_{j \in \mathcal{N}_i \setminus \{i\}} d_{ij} \big(\beta_{ij} (\partial_{u_k} \beta_{ji}) + (\partial_{u_k} \beta_{ij}) \beta_{ji} \big) (u_j - u_i) v_k \\
&\quad = \sum_i e_i \sum_{j \in \mathcal{N}_i \setminus \{i\}} \sum_{k \in \mathcal{N}_j} d_{ij} \beta_{ij} (\partial_{u_k} \beta_{ji}) (u_j - u_i) v_k \\
&\qquad + \sum_i e_i \sum_{j \in \mathcal{N}_i \setminus \{i\}} \sum_{k \in \mathcal{N}_i} d_{ij} (\partial_{u_k} \beta_{ij}) \beta_{ji} (u_j - u_i) v_k \\
&\quad = \sum_j \sum_{i \in \mathcal{N}_j \setminus \{j\}} \sum_{k \in \mathcal{N}_j} e_i d_{ij} \beta_{ij} (\partial_{u_k} \beta_{ji}) (u_j - u_i) v_k \\
&\qquad + \sum_i \sum_{j \in \mathcal{N}_i \setminus \{i\}} \sum_{k \in \mathcal{N}_i} e_i d_{ij} (\partial_{u_k} \beta_{ij}) \beta_{ji} (u_j - u_i) v_k \\
&\quad = \sum_i \sum_{j \in \mathcal{N}_i \setminus \{i\}} \sum_{k \in \mathcal{N}_i} d_{ij} (\partial_{u_k} \beta_{ij}) \beta_{ji} (u_i - u_j) v_k (e_j - e_i) \\
&\quad = \sum_i \sum_{j \in \mathcal{N}_i \setminus \{i\}} \sum_{k \in \mathcal{N}_i} \hat{d}_{ij} (\partial_{u_k} \beta_i) \beta_{ji} (u_i - u_j) v_k (e_j - e_i) \\
&\quad = \sum_i \Big(\sum_{j \in \mathcal{N}_i \setminus \{i\}} \hat{d}_{ij} \beta_{ji} (u_i - u_j) (e_j - e_i) \Big) \Big(\sum_{k \in \mathcal{N}_i} (\partial_{u_k} \beta_i) v_k \Big) \quad \forall v \in \mathbb{R}^N,
\end{aligned}$$

where $e_i \in \mathbb{R}^N$ is the i-th unit vector, i.e., $(e_i)_j = \delta_{ij}$ for all $j \in \{1, \dots, N\}$. Therefore, the application of the Jacobian is equivalent to a few matrix-vector multiplications, where all the matrices involved have the same sparsity pattern as the system matrix $\mathcal{A}$. In particular, the Jacobian can be written as

$$\mathcal{J} = \mathcal{A} - \mathcal{D} + \tilde{\mathcal{D}} - \mathcal{P}\mathcal{Q}, \tag{16a}$$

where $\tilde{\mathcal{D}} = (\tilde{d}_{ij})_{i,j=1}^N$, $\mathcal{P} = (p_{ij})_{i,j=1}^N$, and $\mathcal{Q} = (q_{ij})_{i,j=1}^N$ are defined by

$$\tilde{d}_{ij} \coloneqq \begin{cases} \alpha_{ij} d_{ij} & : i = j, \\ -\sum_{k \neq i} \tilde{d}_{ik} & : i \neq j \end{cases} \qquad \forall i, j \in \{1, \dots, N\}, \tag{16b}$$

$$p_{ij} \coloneqq \begin{cases} \hat{d}_{ji} \beta_{ij} (u_i - u_j) & : i = j, \\ -\sum_{k \neq i} p_{ki} & : i \neq j \end{cases} \qquad \forall i, j \in \{1, \dots, N\}, \tag{16c}$$

$$q_{ij} \coloneqq \partial_{u_j} \beta_i \qquad \forall i, j \in \{1, \dots, N\}. \tag{16d}$$

Formula (16a) is valid because

$$\begin{aligned} \mathcal{P}\mathcal{Q}v &= \sum_{j,i,k} e_j p_{ji} q_{ik} v_k = \sum_i \sum_{j,k \in \mathcal{N}_i} e_j p_{ji} q_{ik} v_k \\ &= \sum_i \Big(\sum_{j \in \mathcal{N}_i} e_j p_{ji} \Big) \Big(\sum_{k \in \mathcal{N}_i} q_{ik} v_k \Big) \\ &= \sum_i \Big(\sum_{j \in \mathcal{N}_i \setminus \{i\}} (e_j - e_i) p_{ji} \Big) \Big(\sum_{k \in \mathcal{N}_i} q_{ik} v_k \Big) \\ &= -\sum_i \Big(\sum_{j \in \mathcal{N}_i \setminus \{i\}} \hat{d}_{ij} \beta_{ji} (u_i - u_j)(e_j - e_i) \Big) \Big(\sum_{k \in \mathcal{N}_i} (\partial_{u_k} \beta_i) v_k \Big) \quad \forall v \in \mathbb{R}^N \end{aligned}$$

holds due to the fact that $\mathcal{D}$, $\mathcal{P}$, and $\mathcal{Q}$ have the same sparsity pattern as $\mathcal{A}$ and $\mathcal{D}$.

In many cases, the computation of $\partial_{u_i} \beta_i$ is more complicated than that of $\partial_{u_j} \beta_i$ for $j \in \mathcal{N}_i \setminus \{i\}$. However, if the nodal correction factor β_i depends only on $(u_i - u_j)_{j \in \mathcal{N}_i \setminus \{i\}}$, the entry e_{ii} does not need to be calculated from scratch. It can be computed efficiently as follows:

$$q_{ii} = \partial_{u_i} \beta_i = -\sum_{j \in \mathcal{N}_i \setminus \{i\}} \partial_{u_j} \beta_i = -\sum_{j \in \mathcal{N}_i \setminus \{i\}} q_{ij} \quad \forall i \in \{1, \dots, N\}. \tag{17}$$

In this case, the matrices $\tilde{\mathcal{D}}$ and $\mathcal{Q}$ have vanishing row sums while the entries $(p_{ij})_{i=1}^N$ sum up to zero for every column $j \in \{1, \dots, N\}$.

5.2 Numerical effort

Formula (16) yields an efficient way to determine the entries of the Jacobian and can also be used to perform a matrix-vector multiplication without calculating the entries. To analyze the numerical effort of an *explicitly* and *implicitly* stored matrix $\mathcal{J}$, let us consider a structured quadrilateral/hexahedral mesh and ignore degrees of freedom associated with nodes close to the boundary. In this case, each row of the system matrix $\mathcal{A}$ has at most 3^d nonzero entries due to the compact support property of the basis functions (cf. Fig. 1). By the assumption that the nodal correction factors β_i depend only on degrees of freedom u_j in the direct neighborhood

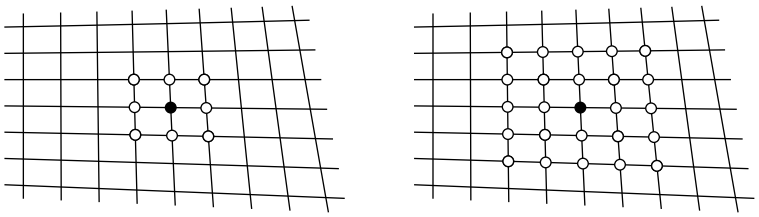


Figure 1: Illustration of connectivity graph corresponding to one row of system matrix $\mathcal{A}$ (left) and Jacobian $\mathcal{J}$ (right).

Table 1: Effort for storing, applying, and recalculating an implicitly and explicitly defined Jacobian for a structured quadrilateral/hexahedral mesh. Note that no column index for $\mathcal{A}-\mathcal{D}+\tilde{\mathcal{D}}$, $\mathcal{P}$, and $\mathcal{Q}$ has to be stored due to same sparsity pattern as $\mathcal{A}$.

	impl.				expl.
	$\mathcal{A}-\mathcal{D}+\tilde{\mathcal{D}}$	$\mathcal{P}$	$\mathcal{Q}$	total	$\mathcal{J}$
storage for nonzero matrix entries	3^dN	3^dN	3^dN	$3\cdot 3^dN$	5^dN
storage for nonzero column indices	—	—	—	—	5^dN
solution-dependent entries	3^dN	3^dN	3^dN	$3\cdot 3^dN$	5^dN
operations per application of $\mathcal{J}$	3^dN	3^dN	3^dN	$3\cdot 3^dN$	5^dN

of node i, i.e., $j\in\mathcal{N}_i$, the connectivity graph of the Jacobian possesses one additional layer. Therefore, each row of $\mathcal{J}$ has up to 5^d nonvanishing entries (cf. Fig. 1). In contrast to this, the matrices $\mathcal{A}-\mathcal{D}+\tilde{\mathcal{D}}$, $\mathcal{P}$, and $\mathcal{Q}$ have the same sparsity pattern as the system matrix $\mathcal{A}$. Hence, column indices of nonzero matrix entries have to be stored only once if the compressed sparse row (CSR) format is used. For this reason, the auxiliary matrices of the implicitly defined Jacobian can be stored just by saving the values of their nonzero matrix entries. Even in one space dimension, this might be less expensive than storing the values and column indices of all nonzero entries for the explicitly defined Jacobian (cf. Table 1). On the other hand, a matrix-vector multiplication employing the implicit definition of $\mathcal{J}$ takes only 8% longer for $d=2$ and should be 54% more efficient in three dimensions. For the explicitly stored Jacobian, recalculation of its matrix entries takes much more effort than determining the entries of $\mathcal{A}-\mathcal{D}+\tilde{\mathcal{D}}$, $\mathcal{P}$, and $\mathcal{Q}$. Indeed, the entries of $\mathcal{J}$ require all information that is stored in the auxiliary matrices of an implicitly defined Jacobian. However, the computation of an explicitly stored matrix $\mathcal{J}$ might be worth the effort when direct solvers are used for corresponding linear systems of equations.

6 Iterative solution procedure

The AFC system under consideration is highly nonlinear by definition of the correction factors and, hence, calls for the use of an iterative solution technique. This solver must be combined with a suitable stopping criterion to guarantee that the final result satisfies (3) with desired precision an discrete maximum principles hold. For that, the difference between two consecutive

iterates might vanish even though the iteration procedure stagnates. A more meaningful control quantity might be (the norm of) the residual $\mathscr{R}$, which measures the defect of the nonlinear system of equations. Unfortunately, the residual cannot be readily interpreted in the finite element sense and its ℓ_2-norm depends on the number of degrees of freedom. To obtain a physically meaningful quantity, we introduce the vector $r \in \mathbb{R}^N$ solving

$$\mathcal{M}_{\mathrm{L}} r = \mathscr{R}(\cdot),$$

where $\mathcal{M}_{\mathrm{L}} \in \mathbb{R}^{N\times N} = (\delta_{ij} m_i)_{i,j=1}^N$ is the lumped mass matrix with diagonal entries

$$m_i \coloneqq \int_\Omega \varphi_i > 0 \quad \forall i \in \{1,\dots,N\}.$$

Then the solution vector $r = \mathcal{M}_{\mathrm{L}}^{-1}\mathscr{R}(\cdot)$ contains the degrees of freedom of a 'lumped finite element residual' $r_h \in V_h$. Its L^2-norm can be approximated by the following weighted ℓ_2-norm (cf. [RW17, Lemma 4.1]):

$$\|r\|_{\tilde{2}} \coloneqq \Big(\sum_{i=1}^N m_i r_i^2\Big)^{\frac{1}{2}} \approx \Big(\int_\Omega r_h^2\Big)^{\frac{1}{2}} = \|r_h\|_{L^2(\Omega)}. \tag{18}$$

Several authors [BB17; Bar+18; JJ19] directly applied an iterative solver to the nonlinear system of equations (3). They used fixed-point iterations with relaxed Anderson acceleration or (formal) Newton-like methods. However, the steady problem might be very ill-conditioned so that algorithms may stagnate or even diverge due to poor initial guesses. To overcome such drawbacks and stabilize the problem at hand, we introduce a pseudo time stepping technique based on a lumped mass matrix. In each iteration $n \in \mathbb{N}\cup\{0\}$, we approximate $u \in \mathbb{R}^N$ by the solution $u^{n+1} \in \mathbb{R}^N$ of the nonlinear problem

$$\frac{m_i}{\Delta t}u_i^{n+1} + \sum_j a_{ij}u_j^{n+1} - \sum_{j\in\mathcal{N}_i\backslash\{i\}} \big(1-\alpha_{ij}(u^{n+1})\big)d_{ij}(u_j^{n+1}-u_i^{n+1}) = \frac{m_i}{\Delta t}u_i^n + g_i$$
$$\forall i \in \{1,\dots,N\}. \tag{19}$$

Decreasing the pseudo time increment $\Delta t > 0$ improves the stability of the nonlinear system of equations but results in a damped convergence of the time dependent solution to a steady state. For an infinitely large pseudo time increment, i.e., '$\Delta t^{-1} = 0$', problem (19) corresponds to the steady AFC system (3) so that the stationary solution is computed directly.

As before, we define the residual $\bar{\mathscr{R}} : \mathbb{R}^N \to \mathbb{R}^N$ of (19) by

$$\big(\bar{\mathscr{R}}(v)\big)_i \coloneqq \frac{m_i}{\Delta t}(v_i - u_i^n) + \sum_j a_{ij}v_j - \sum_{j\in\mathcal{N}_i\backslash\{i\}} \big(1-\alpha_{ij}(v)\big)d_{ij}(v_j - v_i) - g_i$$
$$\forall i \in \{1,\dots,N\}. \tag{20}$$

Based on this definition, system (19) can be iteratively solved using the damped fixed-point iteration

$$u^{n+1,s+1} = u^{n+1,s} - \omega\tilde{\mathcal{J}}^{-1}\bar{\mathscr{R}}(u^{n+1,s}) \quad s = 0,\dots, \tag{21}$$

where $u^{n+1,0} \in \mathbb{R}^N$ and $\omega \in (0,1]$ are a suitable initial guess and a relaxation parameter, respectively. The preconditioner $\tilde{\mathcal{J}}$ possibly depends on $u^{n+1,s}$ and approximates the exact

Jacobian $\bar{\mathcal{J}}$ of $\bar{\mathscr{R}}(\cdot)$ in an appropriate manner. Choices for $\tilde{\mathcal{J}}$ that are used in our numerical studies below are either the low order system matrix $\Delta t^{-1}\mathcal{M}_{\mathrm{L}} + \mathcal{A} - \mathcal{D}$ or the exact Jacobian $\bar{\mathcal{J}}$. The former definition does not depend on the current iterate $u^{n+1,s}$. Hence, the initial calculation of its $\mathcal{LU}$-decomposition can be exploited to efficiently solve the linear system $\tilde{\mathcal{J}}\Delta u^{n+1,s} = -\bar{\mathscr{R}}(u^{n+1,s})$ in each iteration [JJ19]. In contrast to this, the exact and solution-dependent Jacobian $\bar{\mathcal{J}}$ possesses an extended sparsity pattern compared to $\Delta t^{-1}\mathcal{M}_{\mathrm{L}} + \mathcal{A} - \mathcal{D}$ but its use is likely to improve the convergence behavior of the fixed-point iteration.

Furthermore, the solution procedure of (21) is equipped with a relaxation parameter ω to improve the robustness. Jha and John [JJ18; JJ19] tuned the value of ω iteratively by following [JK07, Fig. 12]. In our case, the damping parameter is chosen adaptively and minimizes the approximate L^2-norm of the finite element residual r_h [BB17], i.e.,

$$\omega = \operatorname*{arg\,min}_{\tilde{\omega}\in(0,1]} \left\| \mathcal{M}_{\mathrm{L}}^{-1}\bar{\mathscr{R}}(u^{n+1,s} + \omega\Delta u^{n+1,s}) \right\|_{\tilde{2}}. \tag{22}$$

Practically, the relaxation parameter ω should be bounded below by a small constant $\omega_0 > 0$ to avoid stagnation of the iterative algorithm. Solution of (22) might be very costly because the objective function of the optimization problem is highly nonlinear. Therefore, we 'loosely' approximate ω by choosing

$$\omega = \operatorname*{arg\,min}_{\tilde{\omega}\in\varpi} \left\| \mathcal{M}_{\mathrm{L}}^{-1}\bar{\mathscr{R}}(u^{n+1,s} + \omega\Delta u^{n+1,s}) \right\|_{\tilde{2}}, \quad \text{where } \varpi \coloneqq \left\{ \omega_0 + \tfrac{k-1}{N_\omega - 1}(1-\omega_0) \;\middle|\; k \in \{1,\dots,N_\omega\} \right\} \tag{23}$$

and $N_\omega \in \mathbb{N}\setminus\{1\}$ denotes the total number of samples. Larger values of N_ω obviously lead to more accurate results for the solution of (22), but increase the computational time and do not necessarily reduce the total number of iterations. In our numerical studies, we use $N_\omega = 10$ and the minimal relaxation parameter $\omega_0 = 10^{-3}$ to obtain a good ratio between accuracy and performance.

The fixed-point iteration (21) is constructed to solve the auxiliary problem (19). However, our main focus is on the solution of the steady state problem (3). Therefore, it suffices to approximately solve (19) by performing only a fixed number of iterations, denoted by $S \in \mathbb{N}$. In the numerical studies below, we use only one iteration per pseudo time step and choose $u^{n+1,0} = u^n = u^{n,1}$.

The whole procedure for solving the stationary AFC problem (3) is summarized in Algorithm 1, where the initial guess is chosen as the low order approximation $u^0 \coloneqq (\mathcal{A} - \mathcal{D})^{-1} g$.

7 Limiters under consideration

In this section, we briefly describe the limiters which will be considered in our numerical examples below. The first definition of the correction factors is largely inspired by the BJK limiter presented by Barrenechea, John, and Knobloch [Bar+17b]. Slight modifications are necessary to design a limiter that fits into the analysis presented in Section 5 without being differentiable. In Section 7.2, we define a similar limiter which exploits a regularization parameter to make the correction factors twice continuously differentiable.

Algorithm 1 Iterative solution of nonlinear system (3) using pseudo time stepping scheme.

calculate the initial guess $u^0 = (\mathcal{A} - \mathcal{D})^{-1} g$
$n \leftarrow 0$
while $\left\| \mathcal{M}_{\mathrm{L}}^{-1} \mathscr{R}(u^n) \right\|_{\tilde{2}} > \mathrm{tol}$ **do**
  $u^{n+1,0} \leftarrow u^n$
  for $s \in \{0, \dots, S-1\}$ **do**
    solve $\tilde{\mathcal{J}}(u^{n+1,s}) \Delta u^{n+1,s} = -\bar{\mathscr{R}}(u^{n+1,s})$
    determine relaxation parameter ω via (23)
    $u^{n+1,s+1} \leftarrow u^{n+1,s} + \omega \Delta u^{n+1,s}$
  end for$u^{n+1} \leftarrow u^{n+1,S}$
  $n \leftarrow n + 1$
end while
$u \leftarrow u^{n+1}$

7.1 BJK-like limiters

The *original BJK limiter* of Barrenechea, John, and Knobloch [Bar+17b] was inspired by [Kuz12a; Kuz12b] and employs the following definition of the correction factors:

$$Q_i^+ \coloneqq q_i |d_{ii}| (u_i^{\max} - u_i), \quad Q_i^- \coloneqq q_i |d_{ii}| (u_i - u_i^{\min}) \qquad \forall i \in \{1, \dots, N\}, \tag{24a}$$

$$P_i^+ \coloneqq \sum_{j \in \mathcal{N}_i} d_{ij} \max(0, u_i - u_j), \quad P_i^- \coloneqq \sum_{j \in \mathcal{N}_i} d_{ij} \max(0, u_j - u_i) \quad \forall i \in \{1, \dots, N\}, \tag{24b}$$

$$R_i^+ \coloneqq \min\Big(1, \frac{Q_i^+}{P_i^+}\Big), \quad R_i^- \coloneqq \min\Big(1, \frac{Q_i^-}{P_i^-}\Big) \qquad \forall i \in \{1, \dots, N\}, \tag{24c}$$

$$\beta_{ij} \coloneqq \begin{cases} R_i^+ & : u_i > u_j, \\ 1 & : u_i = u_j, \\ R_i^- & : u_i < u_j \end{cases} \quad \alpha_{ij} \coloneqq \min(\beta_{ij}, \beta_{ji}) \qquad \forall i, j \in \{1, \dots, N\}, j \neq i, \tag{24d}$$

where the local bounds $u_i^{\max}$ and $u_i^{\min}$ are given by

$$u_i^{\max} \coloneqq \max_{j \in \mathcal{N}_i} u_j, \quad u_i^{\min} \coloneqq \min_{j \in \mathcal{N}_i} u_j \qquad \forall i \in \{1, \dots, N\}$$

and the parameters $q_i > 0$, $i \in \{1, \dots, N\}$, can be used to make the scheme linearity preserving [Bar+17b].

Due to the definition of α_{ij} as the minimum of two auxiliary quantities that depend on the sign of $u_i - u_j$, this limiter cannot be written in the form considered in Section 5. For that reason, we slightly modify the calculation in two steps: First, we define the *multiplicative BJK limiter* in which α_{ij} is the product of β_{ij} and β_{ji}. Second, we introduce the *modified BJK limiter* in which the auxiliary quantities β_i are set to

$$\beta_i \coloneqq R_i^+ R_i^- \quad \forall i \in \{1, \dots, N\} \tag{25}$$

and β_{ij} is determined by employing either (15a) or (15b).

As we will see below, the distinction between $u_i > u_j$ and $u_i < u_j$ has only a small positive impact on the accuracy of the scheme and, hence, might not be worth the effort. Even though

we did not discuss such a limiter in Section 5, the Jacobian of the corresponding residual can be written in a form similar to that of (16) but possesses distinct terms $\mathcal{P}^+\mathcal{Q}^+$ and $\mathcal{P}^-\mathcal{Q}^-$ for $u_i > u_j$ and $u_i < u_j$, respectively.

7.2 Regularized limiter

The correction factors of the modified BJK limiter are not (continuously) differentiable and, strictly speaking, the Jacobian does not exist. Badia and Bonilla [BB17] modified the AFC system by using smoothed correction factors that depend on two regularizations of the absolute value. These approximations are defined in terms of a regularization parameter $\varepsilon > 0$ by

$$\sqrt{x^2+\varepsilon} \gtrapprox |x|, \quad \frac{x^2}{\sqrt{x^2+\varepsilon}} \lessapprox |x| \qquad \forall x \in \mathbb{R}$$

and can also be exploited to smooth the maximum and minimum of two quantities: For example, Badia and Bonilla [BB17] and Jha and John [JJ19] used

$$\max\nolimits_\varepsilon(x,y) \coloneqq \frac{1}{2}\Big((x+y)+\sqrt{(x-y)^2+\varepsilon}\Big) \gtrapprox \max(x,y) \quad \forall x,y \in \mathbb{R}$$

in their numerical studies. To the best knowledge of the author, this technique cannot be easily extended to uniquely regularize

$$Q_i^\pm = q_i|d_{ii}||u_i^{\max\,/\,\min} - u_i| = q_i|d_{ii}| \max_{j\in\mathcal{N}_i\backslash\{i\}} \max\big(0, \pm(u_j-u_i)\big) \quad \forall i \in \{1,\dots,N\}$$

if $|\mathcal{N}_i| > 3$. For that reason, we modify the definition of $Q_i^\pm$ by setting

$$Q_i^\pm \coloneqq q_i \sum_{j\in\mathcal{N}_i} d_{ij}\big|\pm(u_j-u_i)\big|_{+,\varepsilon} \quad \forall i \in \{1,\dots,N\}, \tag{26a}$$

where $|\cdot|_{+,\varepsilon}$ denotes a smooth approximation of $\max(0,\cdot)$. Additionally, the auxiliary value P_i is introduced as an approximation of $P_i^+ + P_i^-$, i.e.,

$$P_i \coloneqq \sum_{j\in\mathcal{N}_i} d_{ij}|u_j-u_i|_\varepsilon \quad \forall i \in \{1,\dots,N\}, \tag{26b}$$

where $|\cdot|_\varepsilon \approx |\cdot|$ is a differentiable and nonnegative function. Then the nodal correction factors β_i of the *regularized limiter* are defined by

$$\beta_i \coloneqq 1 - \max\Big(0, 1 - \frac{Q_i^+Q_i^-}{(P_i+\varepsilon)^2}\Big)^{p+1} \quad \forall i \in \{1,\dots,N\}, \tag{26c}$$

where $p \in \mathbb{N}$ must not be smaller than 2 to guarantee that β_i is twice continuously differentiable. The quantities β_{ij} and α_{ij} are computed by (15a) or (15b) and (14), respectively.

This limiter guarantees the validity of discrete maximum principles if β_i satisfies [Loh19]

$$\beta_i \in [0,1], \qquad \beta_i = 0 \quad \text{if } u_i = u_i^{\max} \text{ or } u_i = u_i^{\min}, \tag{27}$$

which is the case whenever

$$|x|_{+,\varepsilon} = 0 \quad \text{if } x \leqslant 0, \qquad |x|_{+,\varepsilon} \geqslant 0 \quad \text{if } x > 0 \tag{28}$$

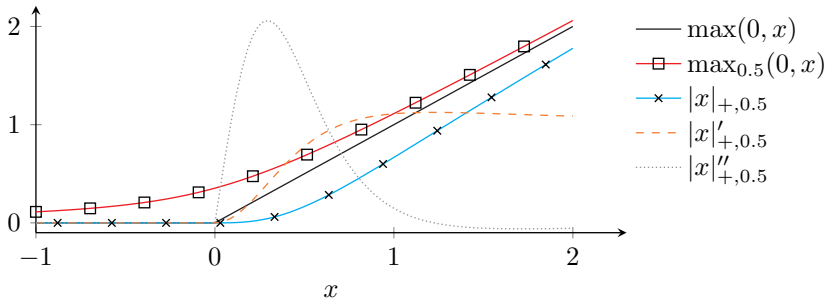


Figure 2: Illustration of different regularizations for $\max(0,\cdot)$.

holds. Condition (28) is violated for the regularization presented in (7.2) due to the fact that $\max_\varepsilon(0,x) > 0$ for every $x < 0$ (cf. Fig. 2). For that reason, we consider the following approximations:

$$|x|_\varepsilon := \sqrt{x^2+\varepsilon}, \quad |x|_{+,\varepsilon} := \frac{\max(0,x)^{p+1}}{|x|^p+\varepsilon} \qquad \forall x \in \mathbb{R},$$

where $p \in \mathbb{N}$ controls the regularity of $|\cdot|_{+,\varepsilon}$ for $\varepsilon > 0$. In the numerical examples presented below, we choose $p = 2$ to guarantee that $|\cdot|_{+,\varepsilon}$ is twice continuously differentiable.

To study the influence of the regularization in the limit $\varepsilon \to 0$, we also introduce the nonsmooth counterpart of the correction factors using

$$|x|_0 := |x|, \quad |x|_{+,0} := \max(0,x) \qquad \forall x \in \mathbb{R}.$$

If $P_i = 0$ holds, we have $|u_j - u_i| = 0$ for all $j \in \mathcal{N}_i$. In this case, the application of the correction factors can be continuously extended by the use of

$$\beta_i = 0 \quad \forall i \in \{1,\dots,N\} \text{ s.t. } P_i = 0.$$

8 Numerical examples

In what follows, we discuss the performance of the iterative solver described in Algorithm 1 by considering the problem

$$\mathbf{v}\cdot\operatorname{grad}(u) = 0 \qquad \text{in } \Omega = (0,1)^2, \tag{29a}$$

$$u = u_{\text{in}} \qquad \text{on } \Gamma_{\text{in}} \tag{29b}$$

for two different velocity fields and inflow boundary conditions. In the *discontinuous translation* test case, the inflow boundary profile u_{in} is convected along the streamlines of the constant velocity field $\mathbf{v} = (\frac{1}{2}, -\sin\frac{\pi}{3})^\top$ while the exact solution is given by

$$u(\mathbf{x}) = \begin{cases} 1 & : x_2 > 0.7 - 2x_1 \sin\frac{\pi}{3}, \\ 0 & : \text{otherwise.} \end{cases} \tag{30}$$

The *smooth circular convection* benchmark corresponds to the velocity field $\mathbf{v} = (x_2, -x_1)^\top$ while the exact solution reads

$$u(\mathbf{x}) = \begin{cases} 1 - \cos\big(5\pi(\|\mathbf{x}\|_2 - 0.4)\big) & : 0.4 < \|\mathbf{x}\|_2 < 0.8, \\ 0 & : \text{otherwise.} \end{cases} \tag{31}$$

Both problems are discretized using a uniform quadrilateral grid, where the mesh size is chosen to be $h = \frac{1}{48}$ if not mentioned otherwise. We approximate the L^2-error of the bound-preserving finite element solution u_h by

$$E_2 \coloneqq \|u_h - I_h u\|_{L^2} = \Big(\int_\Omega (u - I_h u)^2 \Big)^{-\frac{1}{2}} \approx \|u_h - u\|_{L^2},$$

where $I_h u \in V_h$ denotes the finite element interpolant of the exact solution u and is uniquely determined by the definition of the nodal function values, i.e.,

$$(I_h u)(\mathbf{x}_i) = u(\mathbf{x}_i) \quad \forall i \in \{1, \dots, N\}.$$

To measure the rate of convergence when refining the mesh, the *experimental order of convergence* (EOC) can be determined using the formula [LeV96]

$$\mathrm{EOC} \coloneqq \log\big(\frac{E_2(h_2)}{E_2(h_1)}\big) \log\Big(\frac{h_2}{h_1}\Big)^{-1}, \tag{32}$$

where $E_2(h_2)$ and $E_2(h_1)$ are the approximate L^2-errors corresponding to the mesh sizes h_2 and h_1.

Our MATLAB implementation of Algorithm 1 was executed on a system with two XEON E5-2670 CPUs and 64 GB main memory. We measured the elapsed time till $\|r\|_{\tilde{2}}$ became smaller than $\mathrm{tol} = 10^{-10}$ and aborted the simulation after a maximum of 10 000 iterations. If the low order system matrix was used as the preconditioner, i.e., $\tilde{\mathcal{J}} = \Delta t^{-1} \mathcal{M}_\mathrm{L} + \mathcal{A} - \mathcal{D}$, we initially computed the $\mathcal{LU}$-decomposition to efficiently solve the corresponding linear system of equations in each iteration. In all other cases, the backslash-operator was employed using explicitly determined matrices.

As already mentioned above, the nodal correction factors β_i of the modified BJK limiter and the regularized limiter with $\varepsilon = 0$ are not differentiable due to the occurrence of maxima and minima. To determine a 'Jacobian' even for these limiters, we generalize the idea presented in [JJ18; JJ19] and define the following 'derivatives':

$$\partial_x \max_i f_i(x) \coloneqq \min_{i \in \{j | f_j(x) = \max_l f_l(x)\}} \mathrm{mod}\, f_i'(x) \qquad \forall x \in \mathbb{R}, \tag{33a}$$

$$\partial_x \min_i f_i(x) \coloneqq \min_{i \in \{j | f_j(x) = \min_l f_l(x)\}} \mathrm{mod}\, f_i'(x) \qquad \forall x \in \mathbb{R}, \tag{33b}$$

where $f_i : \mathbb{R} \to \mathbb{R}$ are differentiable functions and the min mod-operator reads

$$\min_i \mathrm{mod}\, v_i = \begin{cases} \min_i v_i & : \min_i v_i > 0, \\ \max_i v_i & : \max_i v_i < 0, \\ 0 & : \text{otherwise.} \end{cases}$$

Due to the definition of the correction factors, the sparsity pattern of the Jacobian is more densely than that of the system matrix $\mathcal{A}$. Jha and John [JJ18] reduced the computational costs for determining the entries of the Jacobian and solving corresponding linear systems by neglecting all matrix entries that do not fit into the sparsity pattern of $\mathcal{A}$. To analyze

the influence of this approximation and simplify our notation, we introduce the restriction operator $\mathscr{S} : \mathbb{R}^{N\times N} \to \mathbb{R}^{N\times N}$ as follows:

$$\big(\mathscr{S}(\mathcal{V})\big)_{ij} \coloneqq \begin{cases} v_{ij} & : j \in \mathcal{N}_i, \\ 0 & : j \notin \mathcal{N}_i \end{cases} \quad \forall i,j \in \{1,\dots,N\} \qquad \forall \mathcal{V} \in \mathbb{R}^{N\times N}. \tag{34}$$

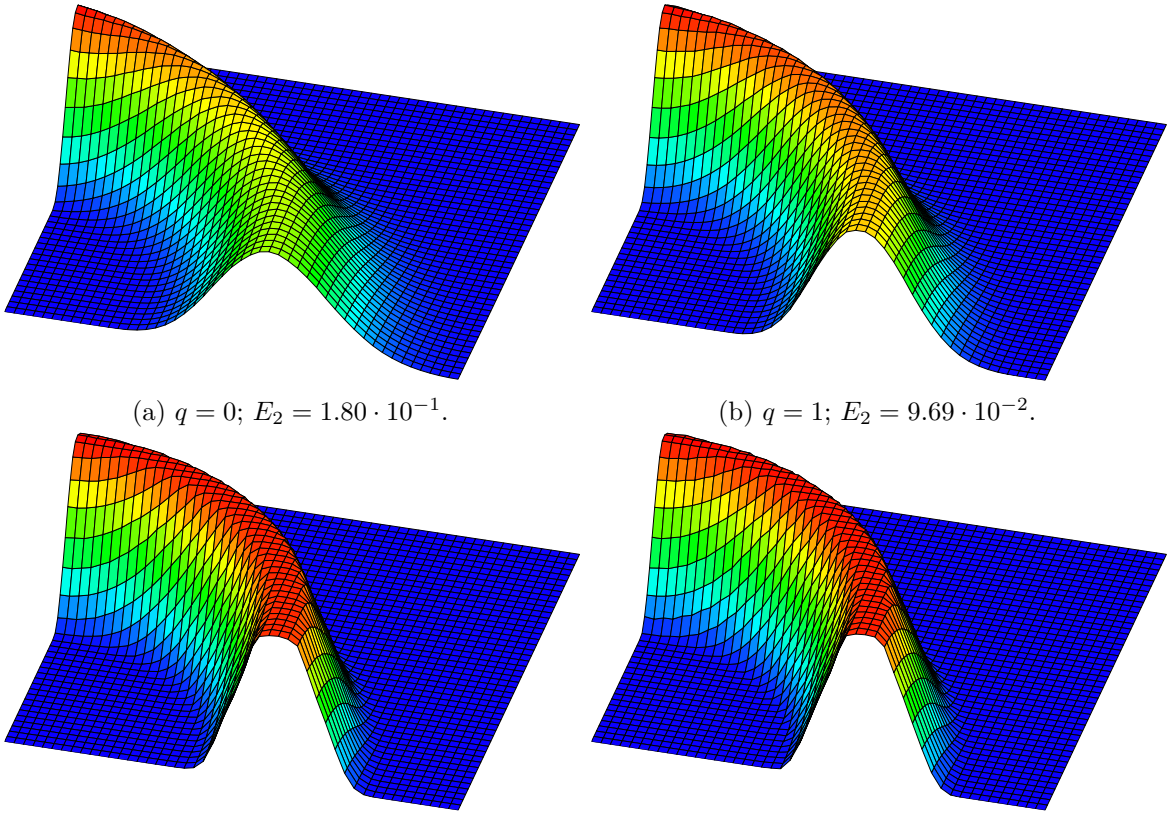
If not mentioned otherwise, formula (15b) is employed to determine the correction factors in an upwind based manner. Corresponding AFC solutions tend to be more accurate than those based on the symmetric definition of β_{ij} and the nonlinearity of the system at hand is reduced. As we are just considering uniform meshes, our analysis is further restricted to the case of a constant parameter $q > 0$, i.e., $q_i = q > 0$ for all $i \in \{1,\dots,N\}$.

Let us first consider the regularized limiter with $\varepsilon = 10^{-6}$. Solutions for both benchmarks and different values of q are presented in Figs. 3 and 4. If the parameter q vanishes, all correction factors are equal to zero and the method coincides with the linear low order scheme (cf. Figs. 3a and 4a). In this case, the peak of the smooth solution decreases and the discontinuity is smeared as the solution profile is convected away from the inflow boundary. By increasing q, this diffusive behavior becomes less pronounced and the accuracy of the method improves. For $q = 3$, the AFC approximation to solution (30) exhibits an artificial plateau at the bottom of the discontinuity. This so called *terracing effect* is due to an unstable high order target scheme and might vanish if the Galerkin method is augmented by some high order stabilization.

While an increase of q leads to more accurate finite element approximations, iterative solution of system (3) becomes more difficult. Therefore, fixed-point iterations should employ a suitable preconditioner to improve the iterative solution behavior. Furthermore, a pseudo time stepping approach can be used to stabilize the problem at hand. We solved the smooth circular convection benchmark using different preconditioners $\tilde{\mathcal{J}}$ and pseudo time increments Δt. In Table 2, we summarize the total number of iterations and the computational time which were required to solve the nonlinear AFC system. Obviously, the notation $\Delta t^{-1} = 0$ corresponds to an iterative solver without the use of pseudo time stepping.

For small time increments and a moderate choice of q (like $\Delta t^{-1} = 100$ and $q = 1$), the fixed-point iteration preconditioned by the low order system matrix is the most efficient solver among the methods under investigation (cf. Table 2). However, if the problem under consideration becomes more involved due to greater values of Δt or q, the use of matrices $\tilde{\mathcal{J}}$ that must be recalculated in each iteration may pay off. In particular, the exact Jacobian $\bar{\mathcal{J}}$ seems to result in a significant reduction of the numerical effort for both benchmarks. The projection of this matrix to the sparsity pattern of $\mathcal{A}$ reduces the number of nonzero matrix entries and might result in more efficient solvers for each linear system, but degrades the quality of the descent direction Δu. In fact, the scheme does not even converge for the smooth circular convection benchmark if the parameters $q = 3$ and $\Delta t^{-1} = 0.01$ or $\Delta t^{-1} = 0$ are used. However, this preconditioner drastically reduces the number of iterations when the pseudo time increment is $\Delta t = 0.1$ and $q \in \{2, 3\}$ is used. For the exact Jacobian $\tilde{\mathcal{J}} = \bar{\mathcal{J}}$, Algorithm 1 converges faster as the pseudo time increment Δt increases. Thus, the use of the pseudo time stepping approach has no positive effect on the computational time.

Similarly to the influence of q, the choice of the relaxation parameter ε has a great impact on the accuracy of the solution: The reduction of ε generally improves the accuracy of u_h but requires more iterations to solve the problem at hand (cf. Table 4). Therefore, appropriate



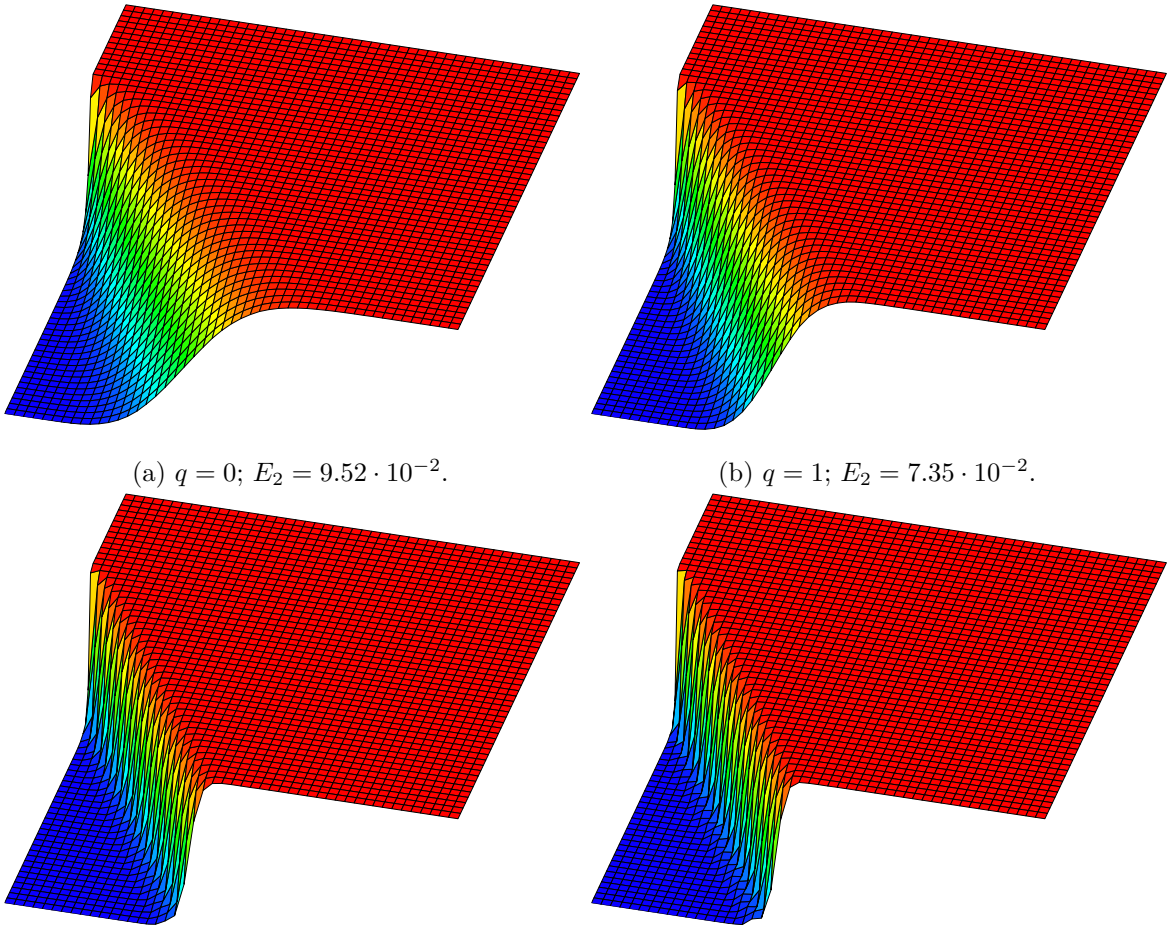
(a) $q = 0$; $E_2 = 1.80 \cdot 10^{-1}$.

(b) $q = 1$; $E_2 = 9.69 \cdot 10^{-2}$.

(c) $q = 2$; $E_2 = 1.43 \cdot 10^{-2}$.

(d) $q = 3$; $E_2 = 1.08 \cdot 10^{-2}$.

Figure 3: Smooth circular convection: AFC approximations obtained using regularized limiter with $\varepsilon = 10^{-6}$ and different values of q.



(a) $q = 0$; $E_2 = 9.52 \cdot 10^{-2}$.

(b) $q = 1$; $E_2 = 7.35 \cdot 10^{-2}$.

(c) $q = 2$; $E_2 = 3.32 \cdot 10^{-2}$.

(d) $q = 3$; $E_2 = 3.05 \cdot 10^{-2}$.

Figure 4: Discontinuous translation: AFC approximations obtained using regularized limiter with $\varepsilon = 10^{-6}$ and different values of q.

Table 2: Number of iterations for regularized limiter with $\varepsilon = 10^{-6}$ using different preconditioners $\tilde{\mathcal{J}}$.

(a) Discontinuous translation.

q	Δt^{-1}	$\tilde{\mathcal{J}} = \Delta t^{-1}\mathcal{M}_{\mathrm{L}} + \mathcal{A} - \mathcal{D}$		$\tilde{\mathcal{J}} = \mathscr{S}(\bar{\mathcal{J}})$		$\tilde{\mathcal{J}} = \bar{\mathcal{J}}$	
		iter.	time (s)	iter.	time (s)	iter.	time (s)
1	100	185	16.9	188	22.5	188	26.2
1	10	49	4.4	31	3.7	32	4.5
1	1	43	3.6	14	1.6	11	1.5
1	0.10	42	3.8	13	1.5	6	0.8
1	0	42	3.9	13	1.5	6	0.8
2	100	867	78.8	499	60.1	501	63.0
2	10	622	56.9	60	7.3	54	7.1
2	1	764	69.1	887	106.3	19	2.4
2	0.01	1071	98.3	1513	179.7	15	2.0
2	0	1055	95.8	1281	153.4	15	1.9
3	100	2186	191.2	929	109.4	881	108.4
3	10	1482	132.0	100	11.9	82	10.2
3	1	2622	234.4	3056	364.5	44	5.8
3	0.01	3198	276.6	3988	471.6	39	5.2
3	0	3193	265.4	3390	406.5	43	5.5

(b) Smooth circular convection.

q	Δt^{-1}	$\tilde{\mathcal{J}} = \Delta t^{-1}\mathcal{M}_{\mathrm{L}} + \mathcal{A} - \mathcal{D}$		$\tilde{\mathcal{J}} = \mathscr{S}(\bar{\mathcal{J}})$		$\tilde{\mathcal{J}} = \bar{\mathcal{J}}$	
		iter.	time (s)	iter.	time (s)	iter.	time (s)
1	100	353	32.7	350	42.3	350	50.9
1	10	59	5.6	49	5.6	47	8.5
1	1	36	3.4	575	69.8	14	2.5
1	0.01	35	3.2	862	103.7	5	0.8
1	0	35	3.4	1039	118.7	5	0.9
2	100	2595	229.2	1917	225.9	1921	238.6
2	10	986	91.4	186	22.4	184	24.8
2	1	1021	95.3	2491	301.1	29	4.1
2	0.01	1039	97.9	6350	766.3	13	1.9
2	0	1039	97.8	7618	936.8	12	1.6
3	100	4840	427.4	3432	411.7	3427	391.2
3	10	2428	223.2	350	42.0	345	42.8
3	1	2347	214.5	7380	884.4	53	6.5
3	0.01	2501	236.8	—	—	47	6.3
3	0	2617	247.3	—	—	28	3.7

Table 4: Smooth circular convection: Number of iterations and approximate L^2-error E_2 for regularized limiter using preconditioner $\tilde{\mathcal{J}} = \bar{\mathcal{J}}$.

(a) Number of iterations.

q	Δt^{-1}	$\varepsilon = 10^{-1}$	$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-4}$	$\varepsilon = 10^{-8}$	$\varepsilon = 0$
1	100	266	332	370	325	317
1	10	35	45	50	47	47
1	1	9	12	14	14	14
1	0.01	2	4	5	5	5
1	0	1	2	5	5	4
2	100	282	347	1974	1903	1898
2	10	38	47	194	183	185
2	1	10	13	30	30	38
2	0.01	3	4	11	15	28
2	0	1	3	10	14	27
3	100	291	354	3986	3692	7005
3	10	39	49	410	366	431
3	1	10	13	59	58	171
3	0.01	3	4	22	36	94
3	0	1	3	18	32	72

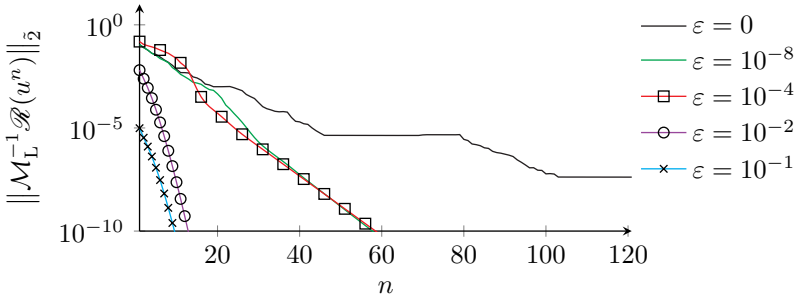
(b) Approximate L^2-error E_2.

q	$\varepsilon = 10^{-1}$	$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-4}$	$\varepsilon = 10^{-8}$	$\varepsilon = 0$
1	0.180 30	0.179 50	0.096 34	0.097 82	0.097 92
2	0.180 30	0.176 80	0.029 86	0.014 26	0.014 38
3	0.180 30	0.172 00	0.023 96	0.010 13	0.010 17

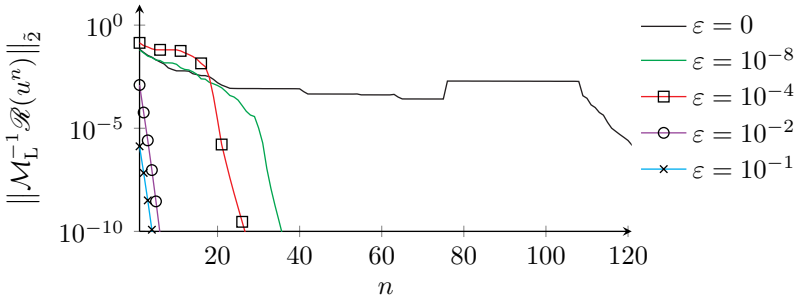
choices of q and ε might be necessary to obtain the best ratio between accuracy and performance. The AFC system with $\varepsilon = 0$ seems to be also solvable with a moderate numerical effort by Algorithm 1 employing $\tilde{\mathcal{J}} = \bar{\mathcal{J}}$. However, a small regularization parameter may still be exploited to reduce the number of iterations without deteriorating the accuracy.

The reason for the suboptimal rate of convergence for $\varepsilon = 0$ is illustrated in Fig. 5: While the norm $\|r\|_{\tilde{2}}$ decreases quadratically for any $\varepsilon > 0$ after a few initial iterations, this behavior cannot be observed when the relaxation parameter vanishes. A quadratic order of convergence can only be expected for $\varepsilon > 0$ and $\Delta t^{-1} = 0$. If a pseudo time stepping approach is used and S is greater than 1, the inner loop of Algorithm 1 would converge quadratically. However, we can only observe linear convergence with a rate proportional to Δt^{-1} for the norm of the steady state residual.

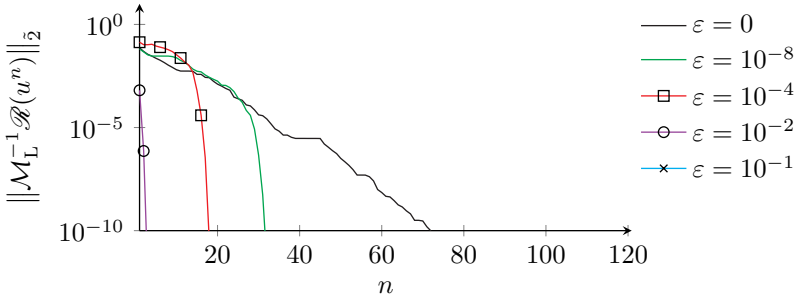
Formulas (16) and (33) can also be used to precondition the fixed-point iteration for the modified BJK limiter by $\bar{\mathcal{J}}$. In Table 6, this limiter is compared to the regularized limiter using $\varepsilon = 0$ in terms of accuracy of the solution and total number of iterations which were required to reach the stopping tolerance. Different values of q and h are considered without using pseudo time steps, i.e., $\Delta t^{-1} = 0$. While the modified BJK limiter converges already for



(a) $\Delta t^{-1} = 1$.



(b) $\Delta t^{-1} = 10^{-1}$.



(c) $\Delta t^{-1} = 0$.

Figure 5: Smooth circular convection: Evolution of $\|r\|_{\tilde{2}}$ during nonlinear iterations for regularized limiter with $q = 3$ using preconditioner $\tilde{\mathcal{J}} = \bar{\mathcal{J}}$.

Table 6: Smooth circular convection: Comparison of nonsmooth limiters using preconditioner $\tilde{\mathcal{J}} = \bar{\mathcal{J}}$ and pseudo time increment $\Delta t^{-1} = 0$.

q	h^{-1}	mod. BJK limiter			regularized limiter ($\varepsilon = 0$)		
		E_2	EOC	iter.	E_2	EOC	iter.
1	24	$5.70 \cdot 10^{-2}$		29	$1.66 \cdot 10^{-1}$		4
1	48	$1.96 \cdot 10^{-2}$	1.54	34	$9.79 \cdot 10^{-2}$	0.76	4
1	96	$5.51 \cdot 10^{-3}$	1.83	38	$5.48 \cdot 10^{-2}$	0.84	5
1	192	$1.43 \cdot 10^{-3}$	1.94	39	$2.95 \cdot 10^{-2}$	0.90	4
2	24	$4.49 \cdot 10^{-2}$		77	$4.77 \cdot 10^{-2}$		25
2	48	$1.27 \cdot 10^{-2}$	1.82	70	$1.44 \cdot 10^{-2}$	1.73	27
2	96	$3.46 \cdot 10^{-3}$	1.87	133	$3.97 \cdot 10^{-3}$	1.86	30
2	192	$8.85 \cdot 10^{-4}$	1.97	911	$1.04 \cdot 10^{-3}$	1.93	34
3	24	$3.97 \cdot 10^{-2}$		258	$3.83 \cdot 10^{-2}$		144
3	48	$1.06 \cdot 10^{-2}$	1.90	—	$1.02 \cdot 10^{-2}$	1.91	72
3	96	$2.95 \cdot 10^{-3}$	1.85	—	$2.93 \cdot 10^{-3}$	1.80	136
3	192	$7.69 \cdot 10^{-4}$	1.94	—	$7.78 \cdot 10^{-4}$	1.91	380

$q = 1$ with the optimal order 2, the regularized limiter requires $q \geqslant 2$ to achieve nearly the same order of convergence. For $q = 3$ and $h \geqslant \frac{1}{96}$, the latter limiter is more accurate than the modified BJK limiter. However, we were not able to compute the solution for mesh sizes $h \leqslant \frac{1}{48}$ and the modified BJK limiter using Algorithm 1 without pseudo time steps.

For both benchmarks, the upwind version of the modified BJK limiter is more accurate than the original BJK limiter (cf. Table 7). This improvement is due to the fact that the sign of the system matrix entries is taken into account in the definition of α_{ij}: For the hyperbolic problem (29) and a solenoidal velocity field, we have $a_{ij} = -a_{ji}$ if $\mathbf{x}_i \notin \partial\Omega$ or $\mathbf{x}_j \notin \partial\Omega$. Therefore, the correction factors employing (15b) depend either on β_i *or* on β_j and, hence, are greater than those of the symmetric version. If we just replace the definition of α_{ij} for the original BJK limiter by (14) and use the symmetric definition of β_{ij} instead of (24d), the accuracy of the solution deteriorates slightly.

In Table 9, the solution behavior of the fixed-point iteration using $\bar{\mathcal{J}}$ is compared to the algorithm employing $\Delta t^{-1}\mathcal{M}_{\mathrm{L}} + \mathcal{A} - \mathcal{D}$ for the discontinuous translation test case and the modified BJK limiter. As already observed in Table 2, the low order system matrix acting as a preconditioner results in more efficient solvers only for small values of q and Δt. As the problems become more involved, the recalculation of $\bar{\mathcal{J}}$ in each iteration pays off and reduces the computational time up to 10 times. In contrast to the regularized limiter with $\varepsilon = 10^{-6}$ (cf. Table 3a), the use of pseudo time steps may reduce the number of iterations for the modified BJK limiter even if the 'exact Jacobian' $\bar{\mathcal{J}}$ is used as a preconditioner. However, the optimal choice of Δt exhibits strong dependence on the current problem and cannot be determined a priori.

Jha and John [JJ19] observed that the use of the regularized limiter only for construction of the preconditioner does not improve the rate of convergence when the residual is evaluated without smoothing the limiter. In fact, the regularized Jacobian acting as a preconditioner behaves as well as the matrix $\bar{\mathcal{J}}$ exploiting (16) for large values of $\|r\|_{\tilde{2}}$ (cf. Fig. 6). As the norm of the residual decreases, the accuracy of the regularized Jacobian deteriorates and damps

Table 7: Approximate L^2-errors E_2 for modifications of BJK limiter.

(a) Discontinuous translation.

Limiter	$q = 1$		$q = 2$	
	$h^{-1} = 48$	$h^{-1} = 96$	$h^{-1} = 48$	$h^{-1} = 96$
ori. BJK	0.037 35	0.028 65	0.032 80	0.025 76
mult. BJK	0.037 40	0.028 65	0.032 82	0.025 76
mod. BJK (sym.)	0.037 79	0.028 80	0.032 82	0.025 77
mod. BJK (upw.)	0.036 38	0.027 93	0.031 96	0.025 02

(b) Smooth circular convection.

Limiter	$q = 1$		$q = 2$	
	$h^{-1} = 48$	$h^{-1} = 96$	$h^{-1} = 48$	$h^{-1} = 96$
ori. BJK	0.022 15	0.006 16	0.013 49	0.003 69
mult. BJK	0.022 15	0.006 16	0.013 49	0.003 69
mod. BJK (sym.)	0.023 49	0.006 60	0.013 97	0.003 83
mod. BJK (upw.)	0.019 59	0.005 51	0.012 67	0.003 46

Table 9: Discontinuous translation: Time savings for modified BJK limiter using different preconditioners $\tilde{\mathcal{J}}$.

q	Δt^{-1}	$\tilde{\mathcal{J}} = \Delta t^{-1}\mathcal{M}_{\mathrm{L}} + \mathcal{A} - \mathcal{D}$		$\tilde{\mathcal{J}} = \bar{\mathcal{J}}$		$\frac{\mathrm{time}(\bar{\mathcal{J}})}{\mathrm{time}(\Delta t^{-1}\mathcal{M}_{\mathrm{L}}+\mathcal{A}-\mathcal{D})}$
		iter.	time (s)	iter.	time (s)	
1	100	1114	82.0	696	79.6	0.97
1	10	492	37.9	79	9.1	0.24
1	1	515	39.0	31	3.8	0.10
1	0.01	523	40.3	33	3.9	0.10
1	0	525	40.0	32	3.7	0.09
2	100	1459	110.5	861	96.4	0.87
2	10	676	52.1	117	11.2	0.21
2	1	684	43.0	82	9.4	0.22
2	0.01	694	54.1	64	7.6	0.14
2	0	701	53.3	95	11.3	0.21
3	100	1444	108.8	850	95.1	0.87
3	10	904	69.4	179	20.3	0.29
3	1	1330	100.9	363	42.2	0.42
3	0.01	1582	122.2	227	26.2	0.21
3	0	1580	120.9	154	18.1	0.15

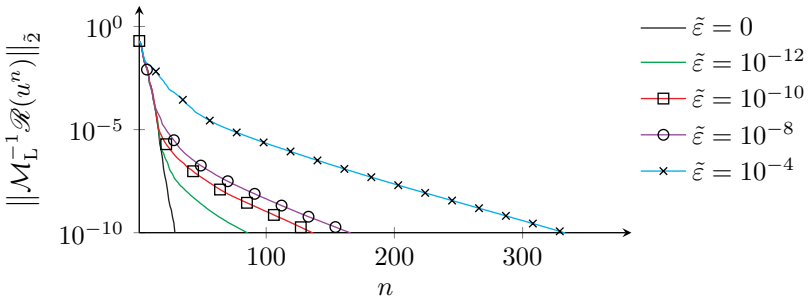


Figure 6: Discontinuous translation: Evolution of $\|r\|_{\tilde{2}}$ during nonlinear iterations for regularized limiter with $\varepsilon = 0$, $q = 2$, and $\Delta t^{-1} = 0$ when preconditioner $\tilde{\mathcal{J}} = \bar{\mathcal{J}}$ is defined in terms of regularized limiter with $\tilde{\varepsilon}$.

the rate of convergence. Therefore, smoothing only the Jacobian with a fixed regularization parameter ε seems to have no positive impact on the numerical performance.

9 Summary

In this work, we presented new results on the solvability of algebraic flux correction schemes for the convection-reaction equation. We proved the existence of a unique solution assuming just the validity of the coercivity condition (6) for limiters with sufficiently small Lipschitz constants. Without further assumptions on the discrete problem, this analysis does not seem to carry over to arbitrary limiters. Indeed, the counterexample of Section 4 illustrates that the requirements of a positive definite system matrix and Lipschitz continuous limiter functions do not suffice to guarantee the uniqueness of the solution.

The main part of this work was focused on the construction and validation of an efficient solver for the nonlinear AFC problem. For a specific family of limiters, we presented an efficient way to perform a matrix-vector multiplication with the Jacobian or determine its entries. This representation provides new possibilities to develop fast solvers for nonlinear AFC problems due to the efficient computation of accurate preconditioners. The solver presented in Algorithm 1 employs a pseudo time stepping approach and a line search algorithm to make the scheme more robust. The use of pseudo time steps may improve the convergence behavior of the fixed-point iteration but the optimal choice of the time increment Δt depends strongly on the problem at hand. Therefore, an improved convergence behavior could be achieve using an adaptively determined Δt depending on the behavior of the residual or relaxation parameter [Val+05]. The definition of the relaxation parameter ω as an approximate solution to an optimization problem makes it possible to reduce the number of iterations. However, the calculation of ω significantly contributes to the computational time of each iteration. Therefore, more efficient calculations of the damping parameter as proposed in [JK07; EW94] or the use of Anderson acceleration [WN11] might further reduce the total effort to solve the AFC system.

As shown in our numerical studies, the limiter under consideration has a great impact on the solution behavior of Algorithm 1: Even for $\varepsilon = 0$, the AFC system employing the regularized limiter required less iterations to compute a more accurate solution than that using the modified BJK limiter. Furthermore, slight regularizations of the correction factors may

result in a significant reduction of the number of iterations without worsening the accuracy of the solution. For limiters with twice continuously differentiable correction factors, second order of convergence was observed for Algorithm 1 without the use of pseudo time steps.

References

[BB17] S. Badia and J. Bonilla. "Monotonicity-preserving finite element schemes based on differentiable nonlinear stabilization". In: *Computer Methods in Applied Mechanics and Engineering* 313 (2017), pp. 133–158.

[Bar+17a] G. R. Barrenechea, E. Burman, and F. Karakatsani. "Edge-based nonlinear diffusion for finite element approximations of convection–diffusion equations and its relation to algebraic flux-correction schemes". In: *Numerische Mathematik* 135.2 (02/2017), pp. 521–545.

[Bar+16] G. R. Barrenechea, V. John, and P. Knobloch. "Analysis of Algebraic Flux Correction Schemes". In: *SIAM Journal on Numerical Analysis* 54.4 (2016), pp. 2427–2451.

[Bar+17b] G. R. Barrenechea, V. John, and P. Knobloch. "An algebraic flux correction scheme satisfying the discrete maximum principle and linearity preservation on general meshes". In: *Mathematical Models and Methods in Applied Sciences* 27.03 (2017), pp. 525–548.

[Bar+18] G. R. Barrenechea, V. John, P. Knobloch, and R. Rankin. "A unified analysis of algebraic flux correction schemes for convection–diffusion equations". In: *SeMA Journal* (05/2018).

[BH82] A. N. Brooks and T. J. Hughes. "Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations". In: *Computer Methods in Applied Mechanics and Engineering* 32.1 (1982), pp. 199–259.

[EW94] S. Eisenstat and H. Walker. "Globally Convergent Inexact Newton Methods". In: *SIAM Journal on Optimization* 4.2 (1994), pp. 393–422.

[JJ18] A. Jha and V. John. "On basic iteration schemes for nonlinear AFC discretizations". In: *WIAS Preprint 2533* (2018). Weierstrass Institute for Applied Analysis and Stochastics.

[JJ19] A. Jha and V. John. "A study of solvers for nonlinear AFC discretizations of convectiondiffusion equations". In: *Computers & Mathematics with Applications* (2019).

[JK07] V. John and P. Knobloch. "On spurious oscillations at layers diminishing (SOLD) methods for convectiondiffusion equations: Part I A review". In: *Computer Methods in Applied Mechanics and Engineering* 196.17 (2007), pp. 2197–2215.

[Kno17] P. Knobloch. "On the Discrete Maximum Principle for Algebraic Flux Correction Schemes with Limiters of Upwind Type". In: *Boundary and Interior Layers, Computational and Asymptotic Methods BAIL 2016.* Ed. by Z. Huang, M. Stynes, and Z. Zhang. Springer International Publishing, 2017, pp. 129–139.

[Kuz12a] D. Kuzmin. “Algebraic Flux Correction I. Scalar Conservation Laws”. In: *Flux-Corrected Transport.* Ed. by D. Kuzmin, R. Löhner, and S. Turek. Scientific Computation. Springer Netherlands, 2012, pp. 145–192.

[Kuz07] D. Kuzmin. “Algebraic flux correction for finite element discretizations of coupled systems”. In: *Computational Methods for Coupled Problems in Science and Engineering II, CIMNE, Barcelona* (2007), pp. 653–656.

[Kuz12b] D. Kuzmin. “Linearity-preserving flux correction and convergence acceleration for constrained Galerkin schemes”. In: *Journal of Computational and Applied Mathematics* 236.9 (2012), pp. 2317–2337.

[Kuz+17] D. Kuzmin, S. Basting, and J. N. Shadid. “Linearity-preserving monotone local projection stabilization schemes for continuous finite elements”. In: *Computer Methods in Applied Mechanics and Engineering* 322 (2017), pp. 23–41.

[LeV96] R. J. LeVeque. “High-resolution conservative algorithms for advection in incompressible flow”. In: *SIAM Journal on Numerical Analysis* 33.2 (1996), pp. 627–665.

[Loh19] C. Lohmann. “Physics-compatible finite element methods for scalar and tensorial advection problems”. PhD thesis. TU Dortmund University (in press), 2019.

[Möl07] M. Möller. “Efficient solution techniques for implicit finite element schemes with flux limiters”. In: *International Journal for Numerical Methods in Fluids* 55.7 (2007), pp. 611–635.

[Möl08] M. Möller. “Adaptive high-resolution finite element schemes”. PhD thesis. PhD thesis, Dortmund University of Technology, 2008.

[RW17] A. Rösch and G. Wachsmuth. “Mass Lumping for the Optimal Control of Elliptic Partial Differential Equations”. In: *SIAM Journal on Numerical Analysis* 55.3 (2017), pp. 1412–1436.

[Val+05] A. M. P. Valli, G. F. Carey, and A. L. G. A. Coutinho. “Control strategies for timestep selection in finite element simulation of incompressible flows and coupled reactionconvectiondiffusion processes”. In: *International Journal for Numerical Methods in Fluids* 47.3 (2005), pp. 201–231.

[WN11] H. Walker and P. Ni. “Anderson Acceleration for Fixed-Point Iterations”. In: *SIAM Journal on Numerical Analysis* 49.4 (2011), pp. 1715–1735.