Weierstraß-Institut
für Angewandte Analysis und Stochastik
Leibniz-Institut im Forschungsverbund Berlin e. V.

Preprint ISSN 2198-5855

On basic iteration schemes for nonlinear AFC discretizations

Abhinav Jha[1], Volker John[2]

submitted: August 16, 2018

[1] Berlin Mathematical School (BMS)
and Freie Universität Berlin
Arnimallee 6, 14195 Berlin, Germany
E-Mail: jha@wias-berlin.de

[2] Weierstrass Institute
Mohrenstr. 39, 10117 Berlin, Germany
and Freie Universität Berlin
Dep. of Mathematics and Computer Science
Arnimallee 6, 14195 Berlin, Germany
E-Mail: volker.john@wias-berlin.de

No. 2533
Berlin 2018

2010 *Mathematics Subject Classification.* 65N22, 65N30.

Key words and phrases. Algebraic flux correction schemes, nonlinear discretizations, Kuzmin limiter, BJK limiter, fixed point iterations, formal Newton method.

Edited by
Weierstraß-Institut für Angewandte Analysis und Stochastik (WIAS)
Leibniz-Institut im Forschungsverbund Berlin e. V.
Mohrenstraße 39
10117 Berlin
Germany

Fax: +49 30 20372-303
E-Mail: preprint@wias-berlin.de
World Wide Web: http://www.wias-berlin.de/

On basic iteration schemes for nonlinear AFC discretizations

Abhinav Jha, Volker John

Abstract

Algebraic flux correction (AFC) finite element discretizations of steady-state convection-diffusion-reaction equations lead to a nonlinear problem. This paper presents first steps of a systematic study of solvers for these problems. Two basic fixed point iterations and a formal Newton method are considered. It turns out that the fixed point iterations behave often quite differently. Using a sparse direct solver for the linear problems, one of them exploits the fact that only one matrix factorization is needed to become very efficient in the case of convergence. For the behavior of the formal Newton method, a clear picture is not yet obtained.

1 Introduction

A steady-state convection-diffusion-reaction equation is given by

$$-\varepsilon\Delta u + \boldsymbol{b}\cdot\nabla u + cu = f \quad \text{in } \Omega, \tag{1}$$

where $\Omega \subset \mathbb{R}^d$, $d \in \{2,3\}$, is a bounded domain, $\varepsilon > 0$ is the diffusion coefficient, $\boldsymbol{b}$ the convection field, c describes a reaction, and f sources. Problem (1) has to be equipped with boundary conditions on $\partial\Omega$.

In applications, the convection-dominated regime $\varepsilon \ll \|\boldsymbol{b}\|_{L^\infty(\Omega)}$ is of interest since the transport of the quantity u (temperature, concentration) by the convection field (velocity) is typically much stronger than the transport by molecular diffusion. In this regime, the solution of (1) possesses usually layers whose width is much smaller than the affordable mesh width.

It is well known that the discretization of (1) in the convection-dominated regime requires stabilized schemes. Two obviously desirable requirements are:

- the numerical solution should be accurate, in particular it should exhibit sharp layers,
- the numerical solution must not have spurious oscillations.

The second requirement is particularly important in applications. Mathematically, it is formulated in the form of the discrete maximum principle (DMP). In the numerical analysis, the DMP is usually proved with the sufficient condition that the discretization leads to a system with an M-matrix. However, it is known [17, Chap. 4.4] that a linear discretization of (1) in the limit case $\varepsilon = 0$ leading to an M-matrix cannot have a local discretization error of second order, thus it is only of low accuracy. A similar result for $\varepsilon > 0$ is not known, but experience shows that higher order discretizations lead to numerical solutions with spurious oscillations.

This situation led to the development of many nonlinear discretizations, where the nonlinearity arises from parameters that depend on the numerical solution. However, most of the proposed nonlinear

DOI 10.20347/WIAS.PREPRINT.2533 Berlin 2018

schemes do not satisfy the DMP, see [11]. There is only one notable exception, namely so-called algebraic flux correction (AFC) schemes, proposed the first time for equations of type (1) in the context of finite element methods in [14]. In AFC schemes, one has to compute a so-called limiter, which depends on the finite element solution. The DMP for the Kuzmin limiter from [14] was proved in [3] and for another limiter, the so-called BJK limiter, in [4].

With the nonlinearity, a new issue arises:

- the numerical solution has to be computed efficiently.

So far, there is no discretization for convection-diffusion-reaction equations (1) that satisfies all three requirements. The construction of such a discretization is formulated as important open problem in [13].

This paper studies numerical methods for the solution of the nonlinear problems arising in AFC schemes. There seems to be so far no systematic investigations of this topic in the literature. A few brief studies can be found in [1, 5], which state more or less that the solution of these problems might be problematic. The goal of this paper consists in performing first steps of a systematic study. Two fixed point iterations, which can be derived in a straightforward way, and a formal Newton method are included. Simulations were performed on academic problems in two dimensions. These studies should serve to obtain some insight in the properties of these schemes and, based on that, to develop ideas for improving them or for constructing new schemes.

2 AFC Schemes

This section provides a short presentation of AFC schemes.

Let $A\underline{u} = \underline{f}$, $A = (a_{ij})_{i,j=1}^n \in \mathbb{R}^{n\times n}$, $\underline{u}, \underline{f} \in \mathbb{R}^n$, be a linear system of equations from a conforming Galerkin discretization of (1). Ordering the unknowns such that the $(n-m)$, $m < n$, Dirichlet values are at the end of $\underline{u}$, this system can be written in the form

$$\begin{aligned} \sum_{j=1}^{n} a_{ij}u_j &= f_i, \qquad i = 1, \ldots, m, \\ u_i &= u_i^b, \qquad i = m+1, \ldots, n. \end{aligned} \tag{2}$$

Defining the symmetric artificial diffusion matrix $D = (d_{ij})_{i,j=1}^n$ by

$$d_{ij} = d_{ji} = -\max\{a_{ij}, 0, a_{ji}\} \quad \text{for } i \neq j, \quad d_{ii} = -\sum_{i\neq j} d_{ij} \tag{3}$$

leads to a system that is equivalent to (2)

$$(\hat{A}\underline{u})_i = f_i + (D\underline{u})_i, \quad i = 1, \ldots, n, \tag{4}$$

where $\hat{A} = A + D$. Since the row sums of the matrix D vanish, there is a representation

$$(D\underline{u})_i = \sum_{i\neq j} f_{ij}, \quad i = 1, \ldots, n,$$

with so-called fluxes $f_{ij} = d_{ij}(u_j - u_i) = -f_{ji}$ for all $i, j = 1, \ldots, n$.

AFC schemes limit those anti-diffusive fluxes f_{ij} that cause spurious oscillations. To this end, system (4) is modified to

$$(\hat{A}\underline{u})_i = f_i + \sum_{j\neq i} \alpha_{ij} f_{ij}, \quad i = 1, \ldots, n, \tag{5}$$

with solution-dependent coefficients $\alpha_{ij} = \alpha_{ij}(\underline{u}) \in [0, 1]$. It is important, for proving the existence of a solution of (5), see [3], and for the conservativity of the method, compare [15], that $\alpha_{ij} = \alpha_{ji}$, $i, j = 1, \ldots, n$. Rewriting (5) yields the following nonlinear system of equations

$$\begin{aligned} \sum_{j=1}^{n} a_{ij} u_j + \sum_{j=1}^{n} (1 - \alpha_{ij}) d_{ij} (u_j - u_i) &= f_i, \qquad i = 1, \ldots, m, \\ u_i &= u_i^b, \qquad i = m+1, \ldots, n. \end{aligned} \tag{6}$$

In the literature, one finds several proposals of limiters for computing α_{ij}. Two of them are briefly presented here. More details, e.g., concerning some issues of the implementation, can be found in [3, 4].

The Kuzmin limiter. This limiter was proposed in [14]. Defining

$$P_i^+ = \sum_{\substack{j=1 \\ a_{ji} \leq a_{ij}}}^{n} f_{ij}^+, \; P_i^- = \sum_{\substack{j=1 \\ a_{ji} \leq a_{ij}}}^{n} f_{ij}^-, \; Q_i^+ = -\sum_{j=1}^{n} f_{ij}^-, \; Q_i^- = -\sum_{j=1}^{n} f_{ij}^+, \tag{7}$$

$i = 1, \ldots, n$, where $f_{ij}^+ = \max\{0, f_{ij}\}$ and $f_{ij}^- = \min\{0, f_{ij}\}$, one computes

$$R_i^+ = \min\left\{1, \frac{Q_i^+}{P_i^+}\right\}, \quad R_i^- = \min\left\{1, \frac{Q_i^-}{P_i^-}\right\}, \quad i = 1, \ldots, m. \tag{8}$$

If P_i^+ or P_i^- is zero, one sets $R_i^+ = 1$ or $R_i^- = 1$, respectively. Also at Dirichlet nodes, one sets

$$R_i^+ = 1, \quad R_i^- = 1, \quad i = m+1, \ldots, n. \tag{9}$$

Finally, for any $i, j \in \{1, \ldots, n\}$ such that $a_{ji} \leq a_{ij}$, the limiter is given by

$$\alpha_{ij} = \begin{cases} R_i^+ & \text{if } f_{ij} > 0 \\ 1 & \text{if } f_{ij} = 0 \\ R_i^- & \text{if } f_{ij} < 0 \end{cases}, \quad \alpha_{ji} = \alpha_{ij}. \tag{10}$$

The Kuzmin limiter can be applied to P_1 and Q_1 finite elements.

The BJK limiter. This limiter was developed in [4] for P_1 finite elements. It was proved that the corresponding AFC method is linearity preserving on arbitrary simplicial grids. First, one sets for $i = 1, \ldots, n$

$$u_i^{\max} = \max_{j \in S_i \cup \{i\}} u_j, \quad u_i^{\min} = \min_{j \in S_i \cup \{i\}} u_j, \quad q_i = \gamma_i \sum_{j \in S_i} d_{ij}, \tag{11}$$

where the index set S_i satisfies

$$\{j \in \{1, \ldots, n\} \setminus \{i\} : a_{ij} \neq 0 \text{ or } a_{ji} > 0\} \subset S_i \subset \{1, \ldots, n\},$$

and γ_i is a positive constant that has to be chosen sufficiently large. Now, one defines for $i = 1, \ldots, m$

$$P_i^+ = \sum_{j \in S_i} f_{ij}^+, \; P_i^- = \sum_{j \in S_i} f_{ij}^-, \; Q_i^+ = q_i(u_i - u_i^{\max}), \; Q_i^- = q_i(u_i - u_i^{\min}), \tag{12}$$

and computes

$$R_i^+ = \min\left\{1, \frac{Q_i^+}{P_i^+}\right\}, \quad R_i^- = \min\left\{1, \frac{Q_i^-}{P_i^-}\right\}, \quad i = 1, \ldots, m.$$

If P_i^+ or P_i^- vanishes, one sets $R_i^+ = 1$ or $R_i^- = 1$, respectively. The final steps consist in setting (9) for the Dirichlet nodes, in computing

$$\bar{\alpha}_{ij} = \begin{cases} R_i^+ & \text{if } f_{ij} > 0 \\ 1 & \text{if } f_{ij} = 0 \\ R_i^- & \text{if } f_{ij} < 0 \end{cases}, \quad i = 1, \ldots, m,\ j = 1, \ldots, n, \tag{13}$$

and in setting

$$\alpha_{ij} = \min\{\bar{\alpha}_{ij}, \bar{\alpha}_{ji}\}, \quad i, j = 1, \ldots, m, \tag{14}$$

$$\alpha_{ij} = \bar{\alpha}_{ij}, \quad i = 1, \ldots, m,\ j = m+1, \ldots, n. \tag{15}$$

3 Methods for Solving the Nonlinear Problem

The methods considered here were already outlined in [5, Sec. 5].

Given an approximation $\underline{u}^{(\nu)}$, $\nu \geq 0$. Then, the next iterate is computed by

$$\underline{u}^{(\nu+1)} = \underline{u}^{(\nu)} + \omega^{(\nu)} \left(\hat{F}\left(\underline{u}^{(\nu)}\right) - \underline{u}^{(\nu)}\right), \tag{16}$$

where $\omega^{(\nu)}$ is a damping parameter and $\hat{F}$ is a map that is determined by the iterative method. The damping parameter is computed with an adaptive strategy which is detailed in [12].

Consider the nonlinear problem (6) in the form $F(\underline{u}) = \underline{0}$ with

$$\begin{aligned} F_i(\underline{u}) &= \sum_{j=1}^{n} a_{ij}u_j + \sum_{j=1}^{n}(1 - \alpha_{ij}(\underline{u}))d_{ij}(u_j - u_i) - f_i = 0, \quad i = 1, \ldots, m, \\ F_i(\underline{u}) &= u_i - u_i^b = 0, \quad i = m+1, \ldots, n. \end{aligned}$$

Then, the damped iteration (16) can be written as

$$\underline{u}^{(\nu+1)} = \underline{u}^{(\nu)} + \omega^{(\nu)} \left(B^{-1}\left[B\underline{u}^{(\nu)} - F\left(\underline{u}^{(\nu)}\right)\right] - \underline{u}^{(\nu)}\right) \tag{17}$$

with a non-singular matrix $B \in \mathbb{R}^{n\times n}$. A vector $\underline{u}$ is a solution of the nonlinear problem (6) if and only if it is a fixed point of (17).

A straightforward idea consists in considering linear problems where the currently available approximation of the limiter is used to assemble the matrix:

$$\begin{aligned} \sum_{j=1}^{n} a_{ij}\,\tilde{u}_j^{(\nu+1)} + \sum_{j=1}^{n}\left(1 - \alpha_{ij}^{(\nu)}\right) d_{ij}\,\left(\tilde{u}_j^{(\nu+1)} - \tilde{u}_i^{(\nu+1)}\right) &= f_i, \quad i = 1, \ldots, m\,, \\ \tilde{u}_i^{(\nu+1)} &= u_i^b, \quad i = m+1, \ldots, n\,, \end{aligned} \tag{18}$$

with $\alpha_{ij}^{(\nu)} = \alpha_{ij}\left(u^{(\nu)}\right)$. In this iteration, the matrix B from (17) is given by

$$B\left(\underline{u}^{(\nu)}\right)_{ij} = \begin{cases} a_{ij} + d_{ij} - \alpha_{ij}^{(\nu)} d_{ij} & \text{if } i \neq j, \\ a_{ii} + d_{ii} + \displaystyle\sum_{j=1, j\neq i}^{n} \alpha_{ij}^{(\nu)} d_{ij} & \text{if } i = j, \end{cases}$$

for $i = 1, \ldots, m, j = 1, \ldots, n$. The last $n - m$ rows have just the diagonal entry 1.

Using that the row sums of the matrix D vanish, one can derive a second fixed point iteration where the limiter appears on the right-hand side, see [5] for details,

$$\begin{aligned} \sum_{j=1}^{n}(a_{ij} + d_{ij})\tilde{u}_j^{(\nu+1)} &= g_i + \sum_{j=1}^{n} \alpha_{ij}^{(\nu)} f_{ij}^{(\nu)}, \quad i = 1, \ldots, m, \\ \tilde{u}_i^{(\nu+1)} &= u_i^b, \qquad\qquad\qquad\qquad i = m+1, \ldots, n. \end{aligned} \tag{19}$$

In (19), the matrix in iteration (17) is given by $B = A + D$, i.e., one has the same matrix in each step. Thus, using a (sparse) direct solver, a matrix factorization has to be performed only once.

Also, a formal Newton method can be derived. This method is formal because the limiters are not differentiable. The formal Jacobian is the matrix B of the scheme (17) and it is given by, compare [5],

$$\begin{aligned} & B\left(\underline{u}^{(\nu)}\right)_{ij} \\ &= \begin{cases} a_{ij} + d_{ij} - \alpha_{ij}^{(\nu)} d_{ij} - \displaystyle\sum_{k=1}^{n} \frac{\partial \alpha_{ik}^{(\nu)}}{\partial u_j} d_{ik} \left(u_k^{(\nu)} - u_i^{(\nu)}\right) & \text{if } i \neq j, \\ a_{ii} + d_{ii} + \displaystyle\sum_{\substack{j=1 \\ j\neq i}}^{n} \alpha_{ij}^{(\nu)} d_{ij} - \sum_{k=1}^{n} \frac{\partial \alpha_{ik}^{(\nu)}}{\partial u_i} d_{ik} \left(u_k^{(\nu)} - u_i^{(\nu)}\right) & \text{if } i = j, \end{cases} \end{aligned} \tag{20}$$

for $i = 1, \ldots, m, j = 1, \ldots, n$. Again, The last $n - m$ rows have only the diagonal entry 1.

In the formal Newton method studied here, the non-smooth situations of the limiters are treated as follows. Non-smoothness is given by the maxima and minima in both limiters. In the definition of $f_{ij}^+, f_{ij}^-, R_i^+$, and R_i^-, one argument of the maximum or minimum is constant. Thus, there is a one-sided derivative that vanishes. In our approach, the derivative that appears in the formal Jacobian is set to be zero in these situations. Consider first the Kuzmin limiter and $a_{ki} \leq a_{ik}$. Then, the entry of the Jacobian is set to be zero if $(f_{ik} > 0) \wedge R_i^+ = 1$, $f_{ik} = 0$, or $(f_{ik} < 0) \wedge R_i^- = 1$. For the BJK limiter, this step is performed if $(f_{ik} > 0) \wedge R_i^+ = 1$, $f_{ik} = 0$ or $(f_{ik} < 0) \wedge R_i^- = 1$. In all other cases, the derivative $\partial \alpha_{ik}^{(\nu)} / \partial u_j$ can be computed. For brevity, details are omitted here.

4 Numerical Studies

All examples are defined in the unit square $\Omega = (0, 1)^2$. Various meshes were used in the simulations, see Fig. 1 for the coarsest level. A standard red refinement was performed. The described iterative methods for solving the nonlinear AFC problems were studied with respect to the number of used steps (iterations + rejections, a rejected step is of the same order of costs as an accepted step) and the used computing time. The following abbreviations will be used for the methods:

- *fixed point rhs*: fixed point iteration with changing right-hand side, (19),
- *fixed point matrix*: fixed point iteration with changing matrix, (18),
- *formal Newton*: formal Newton's method, (20).

All schemes were started with the damping parameter $\omega = 1$. The linear systems of equations were solved with the sparse direct solver UMFPACK [7]. Two stopping criteria were applied. Either, the iteration was stopped if the Euclidean norm of the residual vector was below $\sqrt{\#\text{dof}}\, 10^{-10}$, where $\#$dof is the number of degrees of freedom (including Dirichlet nodes). Or, the iteration terminated if

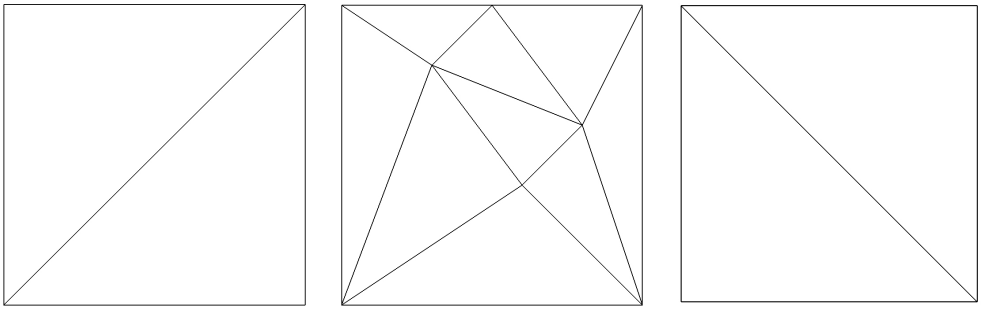

Figure 1: Grid 1, 2 and 3, level 0.

the number of accepted steps reached 25000. In this case, the iteration did not converge. For simplicity of presentation, we do not distinguish between simulations that diverged, giving *nan* or *inf*, and simulations which did not converge. Both cases are indicated with markers at about 25000 iteration steps. All simulations were performed with the code PARMOON [8, 18] at compute servers HP BL460c Gen9 2xXeon, Fourteen-Core 2600MHz.

4.1 Example with a Smooth Solution

In this example, the prescribed solution is

$$u(x) = 100x^2(1-x)y(1-2y)(1-y),$$

the convection field is $\boldsymbol{b} = (3,2)^T$, and the reaction coefficient $c = 1$. Homogeneous Dirichlet boundary conditions are applied on the whole boundary. Results will be presented for two values of the diffusion coefficient: the moderately small value $\varepsilon = 10^{-3}$ and the much smaller value $\varepsilon = 10^{-6}$. This example serves for obtaining first impressions on the behavior of the iterative schemes. Simulations were performed on Grid 1 and Grid 2 from Fig. 1. Note that Grid 2 is not a Delaunay triangulation. For the initial iterate, all values were set to be zero.

In a first study, only the fixed point iterations *fixed point rhs* and *fixed point matrix* were considered. For $\varepsilon = 10^{-3}$, the number of iteration steps is presented in Fig. 2. One can already observe that the behavior of the methods is somewhat different for the different limiters. For the Kuzmin limiter, the method *fixed point rhs* had no difficulties to solve the nonlinear problems and the number of iterations decreased with refinement of the grids. A similar behavior can be observed for *fixed point matrix*, often with a similar number of iterations. For the BJK limiter, in contrast, the method *fixed point matrix* needed consistently much fewer iterations than *fixed point rhs*, apart of the coarsest uniform grid. Altogether, the nonlinear problems in the case of a moderately small value of the diffusion could be solved without real difficulties. We had similar observations also for other examples. For this reason, no further results for moderately small diffusion coefficients will be presented.

Results for $\varepsilon = 10^{-6}$ are shown in Figs. 3 and 4. Figure 3 presents the reduction of the error $\|\nabla(u - u^h)\|_{L^2}$. On the uniform grid, the order of error convergence is similar for both limiters, with the solution of the Kuzmin limiter being somewhat more accurate. For the unstructured grid, it can be observed that the BJK limiter worked well on this grid with an order of convergence of about 1. In contrast, the application of the Kuzmin limiter led to a clear reduction of this order. The behavior of the iterative methods is presented in Fig. 4. Now, there are fundamental differences considering both limiters. For the Kuzmin limiter, *fixed point rhs* worked satisfactory, all problems were solved within the prescribed maximal number of iterations. But even on the uniform grid, *fixed point matrix* failed to converge on

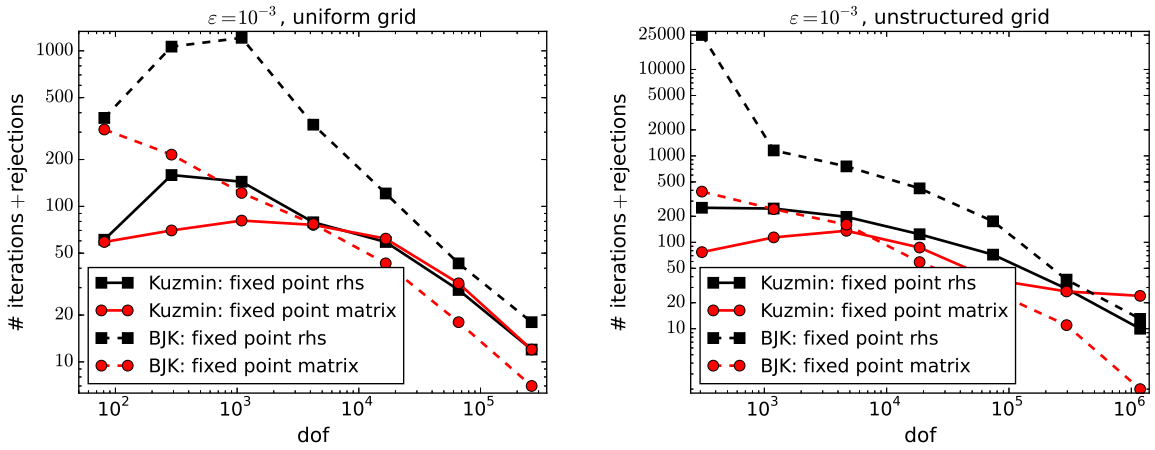

Figure 2: Example 4.1. Number of iterations and rejections for $\varepsilon = 10^{-3}$, left: Grid 1, right: Grid 2.

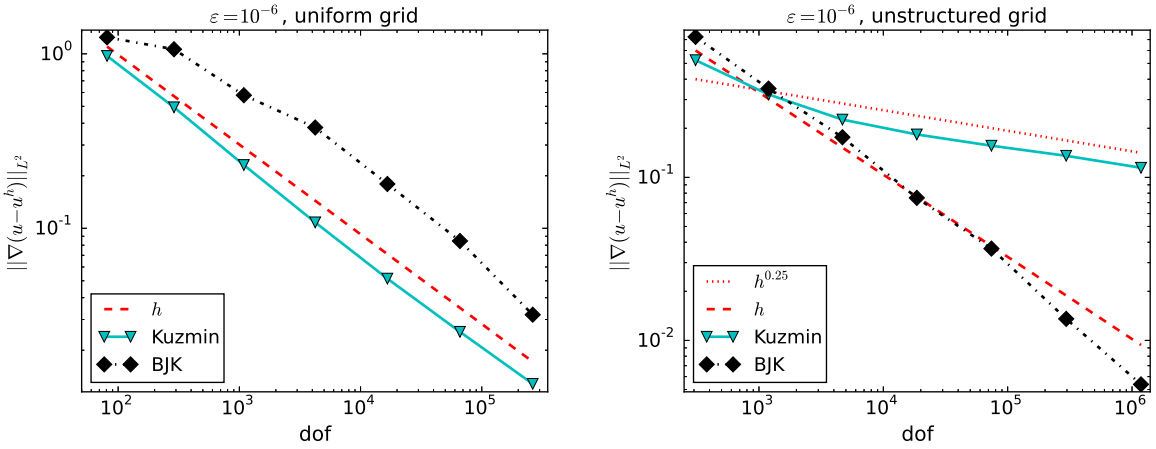

Figure 3: Example 4.1. Errors of the computed solutions, left: Grid 1, right: Grid 2.

fine grids. In case of the BJK limiter, *fixed point rhs* did not converge on many grids, but *fixed point matrix* performed usually quite well.

Since the application of the Kuzmin limiter on the unstructured grid led to quite inaccurate numerical solutions, this limiter should not be used on this grid. This combination will not be considered in the further studies.

Next, the *formal Newton* method will be included in the studies. It is well known that Newton-type methods possess generally a smaller domain of convergence than simpler fixed point iterations. We could observe this behavior also here: applying *formal Newton* from the first step of the iteration led usually to unsatisfactory results concerning the number of steps. For brevity, those results are not presented here.

The first approach for involving the *formal Newton* method was quite simple. In the first part of the iteration, a fixed point method was applied until the Euclidean norm of the residual vector was below a switching tolerance tol_{sw}. Then, *formal Newton* was performed without any possibility of switching back. The current damping parameter ω was used in the first step of the *formal Newton* method. For the first part, we applied *fixed point rhs* as well as *fixed point matrix*. From the results obtained with these methods, Fig. 4, it can be expected that *fixed point rhs* is a better choice for the Kuzmin limiter

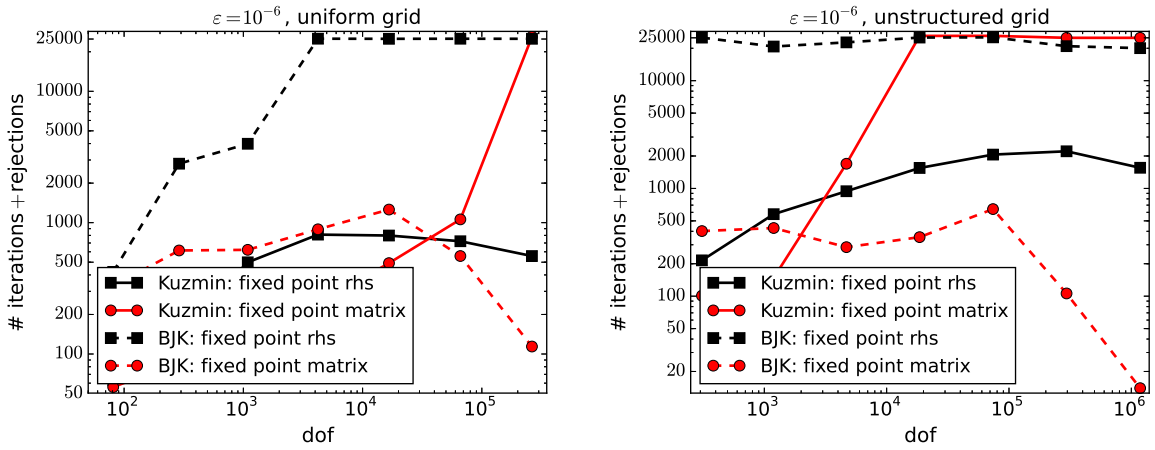

Figure 4: Example 4.1. Number of iterations and rejections for $\varepsilon = 10^{-6}$, left: Grid 1, right: Grid 2.

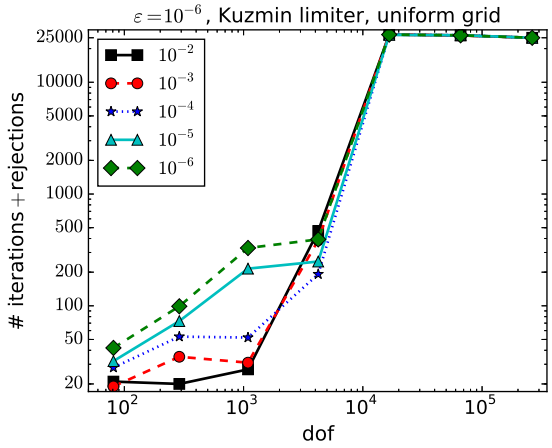

Figure 5: Example 4.1. Number of iterations and rejections for $\varepsilon = 10^{-6}$, Kuzmin limiter and *formal Newton* method with *fixed point rhs* in the first part, different values for the parameter tol_{sw}, Grid 1.

and *fixed point matrix* for the BJK limiter. In fact, the numerical results confirmed these expectations. Thus, for brevity, only the corresponding results are presented in Figs. 5 and 6.

For the Kuzmin limiter, Fig. 5, it can be seen that *formal Newton* worked well only on coarse grids. On finer grids, it did not converge even for small switching tolerances tol_{sw}. The observations for the BJK limiter are different. On some levels, *formal Newton* worked well, at least for sufficiently small tol_{sw}, but on other levels, this method failed to converge.

Examining the non-convergent simulations more closely, we found that often the Euclidean norm of the residual increased within a few steps after having switched to the *formal Newton* method, sometimes it increased considerably. A straightforward idea to mitigate this behavior consists in switching back to the fixed point iteration that was used in the first part after the norm of the residual exceeds a certain limit. This approach was implemented in the form that the back switch to the method from the first part took place always if the Euclidean norm of the residual became larger than $100 \cdot \text{tol}_{\text{sw}}$. While switching between the methods, the current damping parameter ω was not changed. However, the behavior of the *formal Newton* method generally did not improve. The only exception is presented in Fig. 7, where it can be seen that the choice $\text{tol}_{\text{sw}} = 10^{-5}$ led to a convergent method for the BJK limiter on all levels

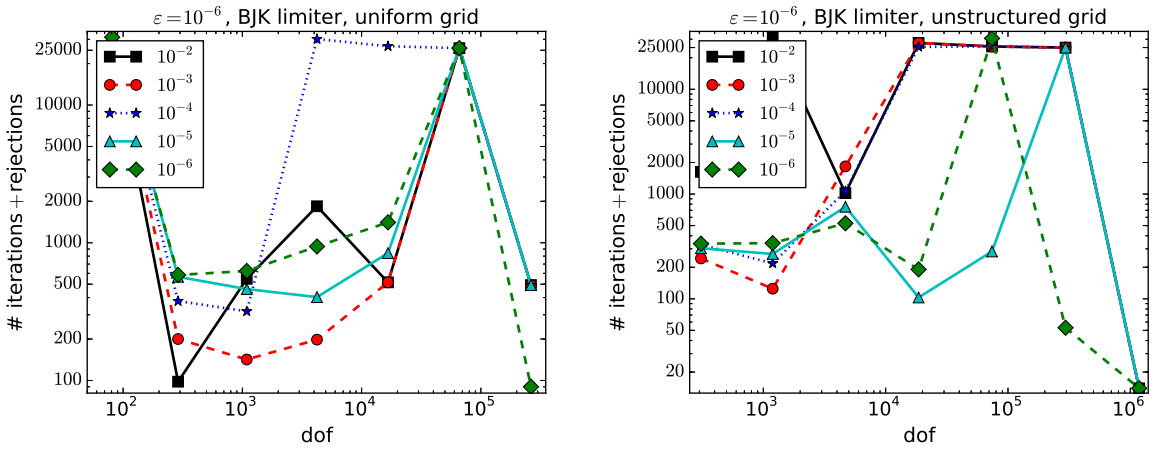

Figure 6: Example 4.1. Number of iterations and rejections for $\varepsilon = 10^{-6}$, BJK limiter and *formal Newton* method with *fixed point matrix* in the first part, different values for the parameter $\mathrm{tol}_{\mathrm{sw}}$, left: Grid 1, right: Grid 2.

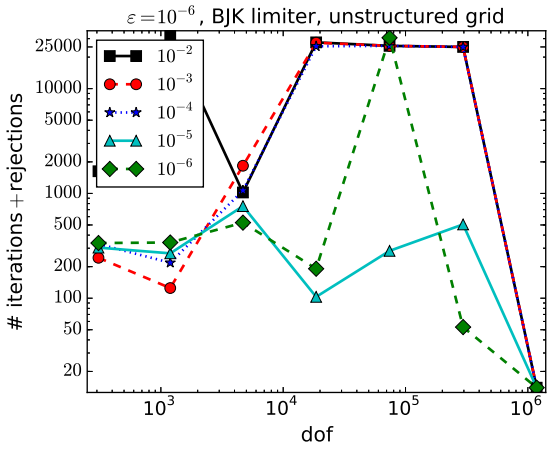

Figure 7: Example 4.1. Number of iterations and rejections for $\varepsilon = 10^{-6}$, BJK limiter and *formal Newton* method with *fixed point matrix* in the first part and switching back to *fixed point matrix* if the norm of the residual became too large, different values for the parameter $\mathrm{tol}_{\mathrm{sw}}$, Grid 2.

of the unstructured grid.

The last investigation for this example studies computing times. On Grid 1, the methods *fixed point rhs* for the Kuzmin limiter and *fixed point matrix* applied to the BJK limiter converged without difficulties, compare Fig. 4. The times for calculating the limiters were very similar. Thus, differences in computing times are mainly due to differences of the needed time for examining the iterations. All setups were simulated five times, the slowest and the fastest computing times were removed, and the averages of the other three times are presented in Fig. 8. One can observe that *fixed point rhs* is generally about one order of magnitude faster than *fixed point matrix*, although the number of iterations is similar for many levels and *fixed point matrix* needed even far less iterations on the finest grid, see Fig. 4. Hence, the possibility to use in *fixed point rhs* just one factorization of the matrix for the complete iteration has a strong positive impact on the computing times for this method.

In further studies, we could observe that one step with the *formal Newton* method is even more ex-

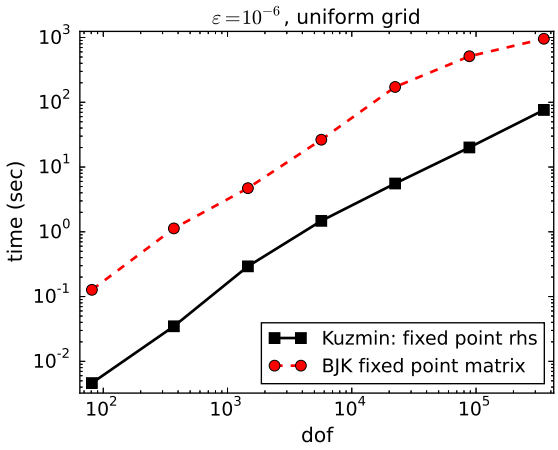

Figure 8: Example 4.1. Simulation times for $\varepsilon = 10^{-6}$, Grid 1.

pensive than one step with *fixed point matrix*, because of the time needed for computing the entries of the formal Jacobian. For brevity, these results are not presented here.

Already for an example with a smooth solution, there were only few of the considered methods that converged in the convection-dominated case on every refinement level. On the uniform grid, for the Kuzmin limiter only *fixed point rhs* worked well and for the BJK limiter only *fixed point matrix*. There were two satisfactory performing approaches for the BJK limiter on the unstructured grid: *fixed point matrix* and *formal Newton* with $\text{tol}_{\text{sw}} = 10^{-5}$, where *fixed point matrix* was used as starting method and it was switched back to *fixed point matrix* if the norm of the residual became too large. With respect to computing times, *fixed point rhs*, in the case of convergence, outperformed all other methods.

4.2 Example with Interior and Boundary Layers

This example, proposed in [10], is a standard academic example for numerical studies of steady-state convection-diffusion equations. It is given in $\Omega = (0,1)^2$ with $\boldsymbol{b} = (\cos(-\pi/3), \sin(-\pi/3))$, $c = f = 0$ and the Dirichlet boundary condition

$$u = \begin{cases} 1 & (y = 1 \wedge x > 0) \text{ or } (x = 0 \wedge y > 0.7), \\ 0 & \text{else.} \end{cases}$$

Again, the strongly convection-dominated case $\varepsilon = 10^{-6}$ is considered. Then, the solution exhibits an internal layer in the direction of the convection starting from the jump of the boundary condition at the left boundary and two exponential layers at the right and the lower boundary, see Fig. 9.

In this example, a study of the impact on choosing the initial iterate in different ways will be presented. For the initial iterate, we considered the following options:

- setting all non-Dirichlet degrees of freedom to zero (zero),
- using the solution of the upwind finite element method from [16] (upwind),
- using the solution of the SUPG method from [6, 9] (SUPG),
- using the solution of the Galerkin method (Galerkin).

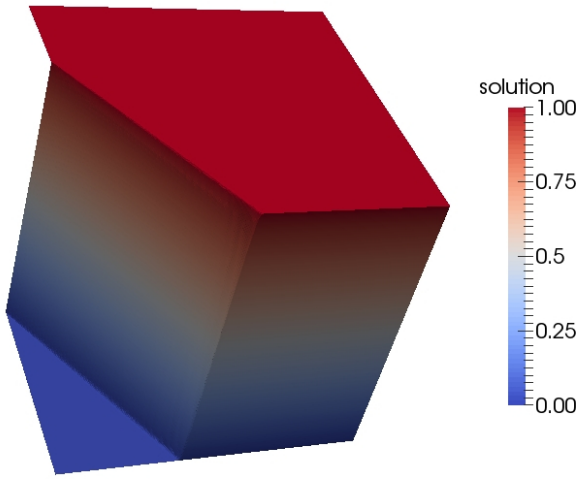

Figure 9: Example 4.2. Solution (computed with the BJK limiter, Grid 3, level 9).

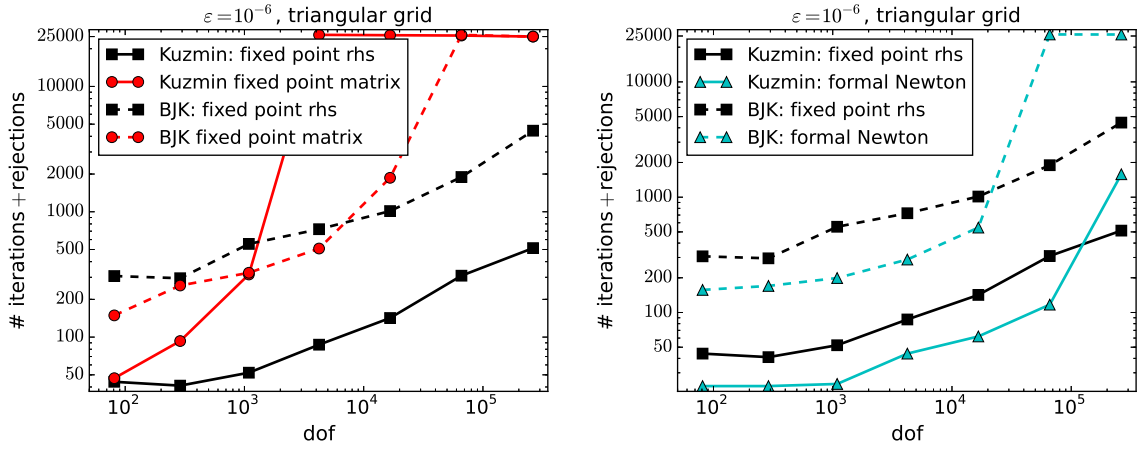

Figure 10: Example 4.2. Number of iterations and rejections, Grid 3.

Starting with the zero initial iterate is a usual approach if no information about the expected solution are available. With the upwind method as initial iterate, the positions of the layers are known from the beginning, but the layers are strongly smeared. The positions of the layers are also known with the SUPG method, the layers are sharp, but there are considerable spurious oscillations in a vicinity of the layers. The incorporation of the Galerkin finite element method in this study is just for completeness.

First, again the behavior of the fixed point iterations was studied, see Fig. 10, left picture. All simulations presented in this figure were started with the SUPG solution as initial iterate. In this example, *fixed point rhs* converged for both limiters on all levels, whereas *fixed point matrix* did not converge for both limiters on fine levels. For the Kuzmin limiter, the *fixed point rhs* method needed considerably less iterations. Representative results for the *formal Newton* method, with *fixed point rhs* as scheme that was used if the norm of the residual was too large and $\mathrm{tol}_{\mathrm{sw}} = 10^{-5}$, are displayed in Fig. 10, right picture. On coarser levels, this approach needed less iterations than *fixed point rhs*, but on finer levels, it even failed in two cases.

The dependency of the number of iterations and rejections on the initial iterate is illustrated in Fig. 11. Generally, there are only minor differences between the four initial iterates. Often, using the SUPG solution proved to be a good choice.

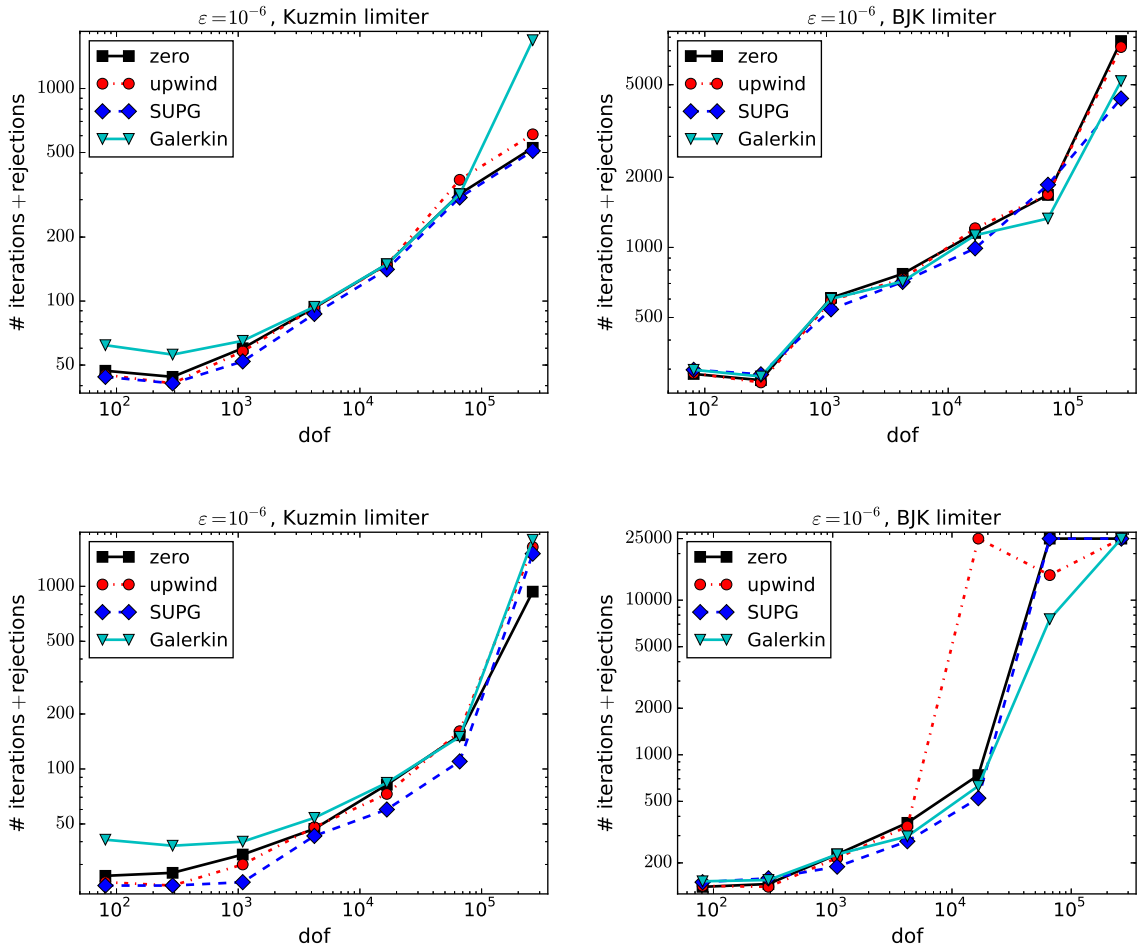

Figure 11: Example 4.2. Number of iterations and rejections depending on the initial iterate, top: *fixed point rhs*, bottom: *formal Newton*, Grid 3.

5 Summary and Outlook

This paper presented first steps of a systematic study of schemes for solving the nonlinear problems arising in AFC finite element discretizations of steady-state convection-diffusion-reaction equations. Two basic fixed point iterations and a formal Newton method where included in these studies. The studies were performed for two limiters.

Consider only the results for the strongly convection-dominated situations.

- It could be observed that both fixed point iterations behaved rather differently, which we did not expected before the studies. Whereas *fixed point rhs* always converged for the Kuzmin limiter and in Example 4.2 also for the BJK limiter, *fixed point matrix* often failed to converge on fine grids.

- For the *formal Newton* method, there is no clear picture. Its behavior depended on the choice of tol_{sw}, sometimes it needed considerably less iterations than the fixed point methods, however, rather often it did not converge.

- For all methods, the choice of the initial iterate did generally not possess a big impact on the

number of iterations. Usually, using the SUPG solution was an appropriate choice.

- From the point of view of efficiency, *fixed point rhs* exploited the fact that a sparse direct solver was used and this method requires only one matrix factorization for the whole iteration. In case of convergence, it was by far the most efficient approach.

The findings collected for the basic schemes will serve in our future work as basis for the development of schemes that behave hopefully better. We want to pursue the approaches of constructing more sophisticated transitions between the schemes, of appropriate combinations of schemes, of utilizing regularizations in Newton-type methods, and of using better damping strategies. In examples where it can be applied, a projection step to a space of admissible functions, as proposed in [2], should be utilized. Furthermore, more complex examples in two and three dimensions will be studied.

References

[1] Matthias Augustin, Alfonso Caiazzo, André Fiebach, Jürgen Fuhrmann, Volker John, Alexander Linke, and Rudolf Umla. An assessment of discretizations for convection-dominated convection-diffusion equations. *Comput. Methods Appl. Mech. Engrg.*, 200(47-48):3395–3409, 2011.

[2] Santiago Badia and Jesús Bonilla. Monotonicity-preserving finite element schemes based on differentiable nonlinear stabilization. *Comput. Methods Appl. Mech. Engrg.*, 313:133–158, 2017.

[3] Gabriel R. Barrenechea, Volker John, and Petr Knobloch. Analysis of algebraic flux correction schemes. *SIAM J. Numer. Anal.*, 54(4):2427–2451, 2016.

[4] Gabriel R. Barrenechea, Volker John, and Petr Knobloch. An algebraic flux correction scheme satisfying the discrete maximum principle and linearity preservation on general meshes. *Math. Models Methods Appl. Sci.*, 27(3):525–548, 2017.

[5] Gabriel R. Barrenechea, Volker John, Petr Knobloch, and Richard Rankin. A unified analysis of algebraic flux correction schemes for convection-diffusion equations. *SeMA*, 2018. in press.

[6] Alexander N. Brooks and Thomas J. R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 32(1-3):199–259, 1982. FENOMECH '81, Part I (Stuttgart, 1981).

[7] Timothy A. Davis. Algorithm 832: UMFPACK V4.3—an unsymmetric-pattern multifrontal method. *ACM Trans. Math. Software*, 30(2):196–199, 2004.

[8] S. Ganesan, V. John, G. Matthies, R. Meesala, S. Abdus, and U. Wilbrandt. An object oriented parallel finite element scheme for computing pdes: Design and implementation. In *IEEE 23rd International Conference on High Performance Computing Workshops (HiPCW) Hyderabad*, pages 106–115. IEEE, 2016.

[9] T. J. R. Hughes and A. Brooks. A multidimensional upwind scheme with no crosswind diffusion. In *Finite element methods for convection dominated flows (Papers, Winter Ann. Meeting Amer. Soc. Mech. Engrs., New York, 1979)*, volume 34 of *AMD*, pages 19–35. Amer. Soc. Mech. Engrs. (ASME), New York, 1979.

[10] Thomas J. R. Hughes, Michel Mallet, and Akira Mizukami. A new finite element formulation for computational fluid dynamics. II. Beyond SUPG. *Comput. Methods Appl. Mech. Engrg.*, 54(3):341–355, 1986.

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

[12] Volker John and Petr Knobloch. On spurious oscillations at layers diminishing (SOLD) methods for convection-diffusion equations. II. Analysis for P_1 and Q_1 finite elements. *Comput. Methods Appl. Mech. Engrg.*, 197(21-24):1997–2014, 2008.

[13] Volker John, Petr Knobloch, and Julia Novo. Finite elements for scalar convection-dominated equations and incompressible flow problems: a never ending story? *Comput. Visual Sci.*, 2018. in press.

[14] Dmitri Kuzmin. Algebraic flux correction for finite element discretizations of coupled systems. In M. Papadrakakis, E. Oñate, and B. Schrefler, editors, *Proceedings of the Int. Conf. on Computational Methods for Coupled Problems in Science and Engineering*, pages 1–5. CIMNE, Barcelona, 2007.

[15] Dmitri Kuzmin and Matthias Möller. Algebraic flux correction. I. Scalar conservation laws. In *Flux-corrected transport*, Sci. Comput., pages 155–206. Springer, Berlin, 2005.

[16] Masahisa Tabata. A finite element approximation corresponding to the upwind finite differencing. *Mem. Numer. Math.*, 1(4):47–63, 1977.

[17] Pieter Wesseling. *Principles of computational fluid dynamics*, volume 29 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2001.

[18] Ulrich Wilbrandt, Clemens Bartsch, Naveed Ahmed, Najib Alia, Felix Anker, Laura Blank, Alfonso Caiazzo, Sashikumaar Ganesan, Swetlana Giere, Gunar Matthies, Raviteja Meesala, Abdus Shamim, Jagannath Venkatesan, and Volker John. ParMooN—A modernized program package based on mapped finite elements. *Comput. Math. Appl.*, 74(1):74–88, 2017.