# Algebraic stabilizations for scalar convection-diffusion equations

Erika Gintautas

Master thesis

supervised by

Prof. Dr. Volker John
Dr. Alfonso Caiazzo

submitted to

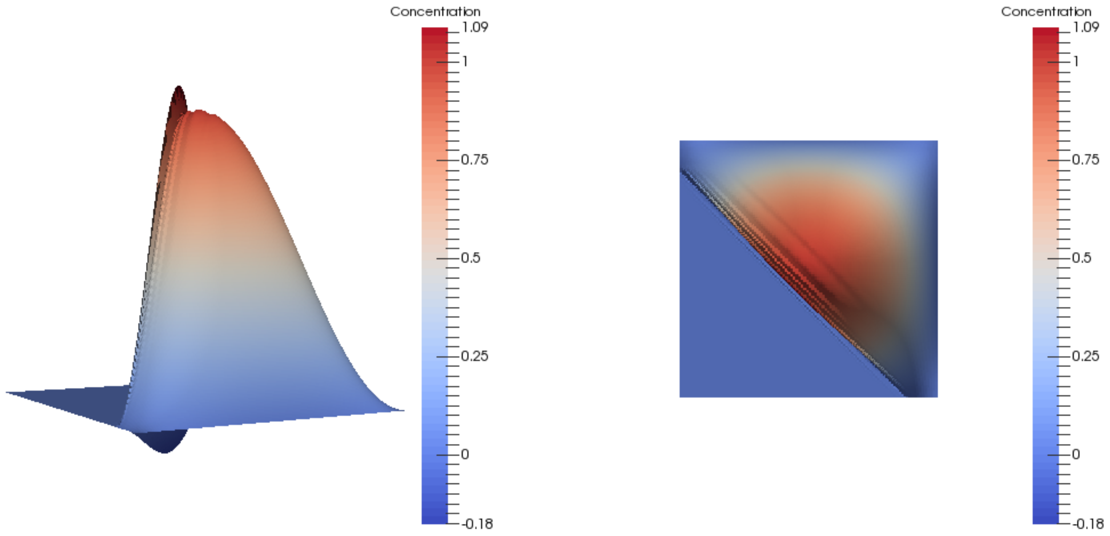
Freie Universität Berlin
Fachbereich für Mathematik und Informatik
Berlin, Germany

on April 13, 2016

# Contents

# 1 Introduction

Scalar convection-diffusion equations are of wide interest in numerous scientific fields as physics, biology and chemistry. Various boundary value problems with incompressible flows are convection-dominated and standard discretizations as SUPG may lead to spurious oscillations in regions of steep gradients. The goal of algebraic flux stabilizations is to suppress these oscillations by adding a specific amount of artificial diffusion in the neighborhood of steep gradients.



(a) $\epsilon = 10^{-8}$

(b) $\epsilon = 10^{-8}$

Figure 1.1: Traveling wave: Oscillations with SUPG

The flux-corrected transport algorithm (FCT) which was first introduced by Boris and Book [BB73] in 1973 was the *“[. . . ] first scheme to ensure positivity/monotonicity even in the limit of pure convection [. . . ] “* [Kuz09, p.2517]. Later in 1979 a new algorithm for multidimensional problems was introduced by Zalesak [Zal79]. Zalesak’s flux limiters will be discussed in Section 2.8.4.

This thesis will provide an overview of advantages and drawbacks of algebraic flux stabilizations described by Dmitri Kuzmin [Kuz09], [Kuz07]. The intention is to create a basic understanding of the design idea of flux correction. We will derive the numerical operators (mass, transport, diffusion, reaction, sources and sinks) with Galerkin discretization and introduce the *Group finite element method* which is a new ansatz to discretize convection [Kuz10, p. 43]. Stationary and instationary equations will be considered separately. The use of the terms algebraic flux correction (AFC) and flux-corrected transport (FCT) is sometimes inconsistent and therefore confusing. AFC is mostly used for stationary problems and FCT for instationary problems. The generic wording will be *algebraic stabilization methods*. Based on [Kuz10] we will describe four main design restrictions, which include *mass lumping*, *mass conservation*, the *zero row-sum property* and *positivity preservation*. They imply that physical properties like density, temperature and concentration will maintain physically meaningful values in the numerical simulations. Afterwards, positivity constraints and maximum principles for the stationary elliptic problem and for the instationary parabolic problem are discussed. In this process the *M-matrix* will be introduced.

After time and space discretization a high-order scheme is obtained, which may be implicit due to the time-discretization. It will be shown that the matrix in the resulting algebraic equation is an M-matrix under certain restrictions for the time step $\Delta t$.

Artificial diffusion is designed such that it enforces the positivity constraints, e.g. no nonphysical undershoots or overshoots are created. The main drawback of artificial diffusion is that it creates a low-order scheme which flattens the solution. Peaks lose a little bit of height in each iteration step. Therefore weighted antidiffusive fluxes must be added. The weights will be called *flux limiters* $\alpha$. These limiters depend on the solution and therefore lead to an implicit scheme. In practice a linearization technique as described in [Kuz07] is applied. The resulting algorithm is similar to a fixed-point iteration.

In Chapter 2.9 main results on error estimation for steady-state linear convection-diffusion-reaction equations from [BJK16] are summarized. The paper is a pioneer on deriving error estimates for these kind of algebraic stabilization methods. Finally four examples will be calculated and discussed particularly with regard to the previous results.

I would like to thank my supervisor Prof. Dr. John, who supported me in numerous meetings and took the time to answer all my questions. I am thankful for the code of algebraic flux correction and flux corrected transport. Moreover, I would like to thank Dr. Caiazzo, who has always been very patient explaining technical issues, as well as Ulrich Wilbrandt, who clarified questions regarding the code and regarding the grid constructions.

# 2 Flux limiting for scalar equations

## 2.1 Model problem for the instationary convection-diffusion equation

In many applications $u$ represents the concentration of a material in a liquid substance, where $b(x)$ represents the direction of the flow and $\epsilon$ defines the amount of diffusion. This thesis will focus on convection-dominated problems, where

$$\begin{aligned} 0 < \epsilon \ll \|b\| \,, & \\ -\frac{1}{2}\nabla \cdot b + c \geq 0 \,, & \qquad c \geq 0 \,, \end{aligned}$$

and $b(x)$ is a convection field. The function $f$ on the right-hand side represents sinks and sources of a concentration in the domain. We will consider the following type of the convection-diffusion equation with standard Dirichlet and homogeneous Neumann boundary conditions

$$\frac{\partial u}{\partial t} - \epsilon\Delta u + b \cdot \nabla u + cu = f \qquad in \quad \Omega \times [0,T] \,, \tag{2.1}$$

$$u(x,t) = g_D(x,t) \qquad \forall (x,t) \in \Gamma_D \times [0,T] \,, \tag{2.2}$$

$$\epsilon \nabla u(x,t) \cdot \eta = 0 \qquad \forall (x,t) \in \Gamma_N \times [0,T] \,, \tag{2.3}$$

$$u(x,0) = u_0(x) \qquad \forall x \in \Omega \times \{0\} \,. \tag{2.4}$$

The boundaries $\Gamma = \Gamma_D \cup \Gamma_N$ are Lipschitz-continuous and the measure of the Dirichlet boundary $(\Gamma_D) > 0$ is positive. The described model problem is a special form of the convection-diffusion equation

$$\frac{\partial u}{dt} + \nabla \cdot (bu - \mathcal{D}\nabla u) + cu = f \quad \text{in} \quad \Omega \times [0,T] \,.$$

If we assume that $b$ is an incompressible velocity field $\nabla \cdot b = 0$, we can transform the convection-diffusion equation into our model problem

$$\begin{aligned} &\nabla \cdot (bu) = b \cdot \nabla u + (\nabla \cdot b)u \,, \\ \Rightarrow &\nabla \cdot (bu) = b \cdot \nabla u \,, \\ \Rightarrow &\frac{\partial u}{\partial t} + b \cdot \nabla u - \nabla \cdot (\mathcal{D}\nabla u) + cu = f \,. \end{aligned}$$

The diffusive part $\mathcal{D}\nabla u$ describes transport of mass or heat by molecular diffusion. This happens for example in fluids, which contain different concentrations of a material or different temperatures in some areas. Molecules start to move to equate the concentration or heat.

Numerical solvers for convection-dominated equations lead to highly nonphysical oscillations if the related discretization is not chosen accordingly. The goal of this thesis is to describe a new type of discretization proposed by Dmitri Kuzmin in [Kuz09], which includes an artificial diffusion in order to minimize those oscillations.

### 2.1.1 Galerkin discretization

After space discretization we will obtain a semidiscrete system of algebraic equations with the following form

$$M_C \frac{\partial u}{\partial t} = -(C + L + R)u + S,$$

where $M_C$ is the *mass matrix*, $C$ is the *discrete transport operator* , $L$ is the *discrete diffusion operator*, $R$ is the *raection matrix* and $S$ represents sources and sinks.
We will assume that $\Omega \subset \mathbb{R}^d$ is a bounded domain with a Lipschitz-continuous boundary $\Gamma$.
In order to find a weak solution, we have to introduce an appropriate solution and test space. For simplification we will assume homogeneous Dirichlet and Neumann conditions

$$\mathcal{L}u = f \qquad \text{in } \Omega \times [0, T], \tag{2.5}$$
$$u = 0 \qquad \forall (x, t) \in \Gamma_D \times [0, T], \tag{2.6}$$
$$A\nabla u \cdot \eta = 0 \qquad \forall (x, t) \in \Gamma_N \times [0, T], \tag{2.7}$$
$$u(x, 0) = u_0(x) \qquad \forall x \in \Omega \times \{0\}\,, \tag{2.8}$$

with $\mathcal{L}u := \frac{\partial u}{\partial t} - \epsilon \Delta u + b \cdot \nabla u + cu$ .

**Definition 2.1** (Weak solution)**.**
*A function $u \in H_0^1(\Omega) := \{v \in H^1(\Omega) : v = 0$ on $\Gamma_D\}$ is called **weak solution** of the mixed boundary value problem* (2.5) - (2.7) *if*

$$\int_\Omega \left( \frac{\partial u}{\partial t} + b \cdot \nabla u + cu \right) v + \epsilon \nabla u \cdot \nabla v \;\, dx - \int_{\Gamma_N \cup \Gamma_D} \epsilon(\nabla u \cdot v) v \;\, dS = \int_\Omega fv \;\, dx$$

*and*

$$u(x, 0) = u_0(x) \qquad \forall x \in \Omega \times \{0\}$$

*holds for all $v \in H_0^1(\Omega \times [0, T])$.*

**Remark 2.1.** *In general we do not have homogeneous Dirichlet boundary conditions $u = 0$ on $\Gamma_D$. For the special case that $\Gamma_N = \emptyset$ we can set $\tilde{u} := u - g_D$. Now it holds that $\tilde{f} := f - \mathcal{L}g_D$, $\mathcal{L}\tilde{u} = \tilde{f}$ in $\Omega$ and $\tilde{u} = 0$ on $\Gamma_D$.*

**Theorem 2.1** (TracesTheorem)**.**
*Assume $\Omega$ is bounded and $\partial\Omega$ is Lipschitz-continuous, then there exists a bounded linear operator $T\colon H^1(\Omega) \to L^2(\partial\Omega)$ such that*

$$Tu = u|_{\partial\Omega} \quad in \quad u \in H^1(\Omega) \cap C(\bar{\Omega}).$$

*Proof.* See L.C. Evans: Partial Differential Equations [Eva98, p. 258] □

**Remark 2.2.** *To guarantee the existence of a solution we have to satisfy some conditions.*

- *We will assume that $f$ is in $L^2(0, T; H^{-1}(\Omega))$.*
- *The coefficients $b$ and $c$ are in $L^\infty(0, T; L^\infty(\Omega)) \quad \forall \quad i, j = 1, \dots, d$.*
- *The coefficient $c$ must be greater or equal than zero.*
- *The solution $\tilde{u} = u - g_D$ will be from $H^1_g(\Omega \times [0, T])$, where $u$ is in $H^1_g(\Omega \times [0, T])$, with:*

$$H^1_g := \left\{ u \in H^1 : Tu = g_{|\Gamma_D} \right\} .$$

- *Be aware that for mixed boundary value problems it is not trivial to show the uniqueness of the solution. The proof will not be part of this thesis.*

*For further analysis we will assume homogeneous Dirichlet conditions, provided that the previous points hold.*

The weak formulation requires a space of test functions. This test space equals the space of the solution $v \in H^1_0(\Omega)$. Multiplying our model problem with a test function $v$, integrating it and applying partial integration we obtain the following form of the weak formulation

$$\int_\Omega \frac{\partial u}{\partial t} v + b \cdot \nabla uv + cuv \ dx - \int_{\Gamma_{N \cup D}} \epsilon \left(\nabla u \cdot \vec{\eta}\right) v \ dS + \int_\Omega \epsilon \nabla u \cdot \nabla v \ dx = \int_\Omega f v \ dx \, . \qquad (2.9)$$

The boundary integral eliminates, because of the homogeneous Neumann condition and the fact that $v$ is zero on the Dirichlet boundary $\Gamma_D$.

Now we can rewrite the variational form into $L^2$-products. Below we will write $(\cdot, \cdot)_{L^2}$ instead of $(\cdot, \cdot)_{L^2(\Omega)}$

$$\left(\frac{\partial u(t)}{\partial t}, v\right)_{L^2} = -\left(b(t) \cdot \nabla u(t), v\right)_{L^2} - \left(\epsilon \nabla u(t), \nabla v\right)_{L^2} + (f, v)_{L^2} - (cu, v)_{L^2} \quad \forall v \in H^1_0\,(\Omega) \ .$$

Let $N$ be the number of degrees of freedom, which corresponds to the number of vertices of the finite element discretization. We will take a finite element subspace $V^N \subset H_0^1$ with its basis $B = \{\phi_1, \phi_2, \dots, \phi_N\}$ to approximate $u_N$

$$\begin{aligned} u_N(x,t) &= \sum_{j=1}^{N} u_j(t)\phi_j(x) \quad u_N, \phi_i \in V^N \ , \\ u_N(x,0) &= u_{N0}(x) \ . \end{aligned}$$

In case of Galerkin discretization the test functions $\phi_n$ are from the same finite element space like the basis functions. Replacing $v$ in the first summand of (2.9) we get the mass matrix $M_C$

$$\begin{aligned} \left(\frac{\partial u_N}{\partial t}(t), \phi_i\right)_{L^2} &= \sum_{j=1}^{N} \left(\frac{\partial u_j(t)}{\partial t}\phi_j, \phi_i\right)_{L^2} \ , \\ &= \sum_{j=1}^{N} \frac{\partial u_j(t)}{\partial t}(\phi_j, \phi_i)_{L^2} \quad \forall \phi_i \in V^N \ , \\ &= \left(M_C \frac{\partial u_N(t)}{\partial t}\right)_i \ , \end{aligned}$$

with $(M_C)_{ij} = (\phi_j, \phi_i)_{L^2} = m_{ij}$. The expression $\left(M_C \frac{\partial u_N(t)}{\partial t}\right)_i$ is the $i$-th row of $M_C$ multiplied with $\frac{\partial u_N(t)}{\partial t}$. The mass matrix $M_C$ is always sparse since only basis functions of neighboring points generate nonzero entries. In order to calculate the *transport operator* $C$ we have to discretize $(b \cdot \nabla u, \omega)_{L^2}$:

$$\begin{aligned} -(b(t) \cdot \nabla u_N(t)\phi_j, \phi_i)_{L^2} &= -\sum_{j=1}^{N} (b(t) \cdot \nabla u_j(t)\phi_j, \phi_i)_{L^2} \ , \\ &= -\sum_{j=1}^{N} (b(t) \cdot \nabla \phi_j, \phi_i)_{L^2} u_j(t) \quad \forall \phi_i \in V^N \ , \\ &= -(C(t)\, u_N(t))_i \ , \end{aligned}$$

with $C_{ij}(t) = (b(t) \cdot \nabla \phi_j, \phi_i)_{L^2}$ .

The discretized *diffusion-operator* $L$ is obtained in the following way

$$\begin{aligned} -\epsilon(\nabla u_N(t), \nabla \phi_i)_{L^2} &= -\epsilon \sum_{j=1}^{N} u_j(t)(\nabla \phi_j, \nabla \phi_i)_{L^2} \ , \\ &= -\epsilon(L u_N(t))_i \ , \end{aligned}$$

with $L_{ij} = \epsilon(\nabla \phi_j, \nabla \phi_i)_{L^2}$.
The *reaction matrix* $R$ will need a special treatment. The straightforward way to calculate $R$ is

to set

$$\begin{aligned}(c(t)u_N(t), \phi_i)_{L^2} &= \sum_j^N (c(t)\phi_j, \phi_i)_{L^2} u_j(t) \,, \\ &= (R(t)u_N)_i \,.\end{aligned}$$

Since $c$ is nonnegative it follows that for $P_1$- elements all entries of R are nonnegative. When it comes to design restrictions for algebraic flux correction, we will see that this may harm a very important restriction which requires that $(C + L + R)_{ij} \leq 0 \quad \forall i \neq j$ (see (2.27)). In order to overcome this, the implementation uses a simple diagonal approximation as described in [BJK16, p. 14]

$$(c(t)u_N(t), \phi_i)_{L^2} \approx (c(t), \phi_i)_{L^2}\, u_i(t) \,. \tag{2.10}$$

The resulting matrix is a diagonal matrix with positive entries. The error generated by this approximation will be discussed in Chapter 2.9.

Sources and sinks are modeled by the right-hand side

$$(f, \phi_i)_{L^2} = S_i.$$

### 2.1.2 Group finite element method

Suppose that $b$ is time-dependent. This implies that the *transport operator* $C_{ij}(t) = \left(b(t) \cdot \nabla\phi_j, \phi_i\right)$ has to be calculated in each time step. The calculation of the *transport operator* requires numerical integration which is expensive. Therefore we need a different approach (see [Kuz10, p. 43]). The *group finite element method's* underlying idea is the following ansatz

$$(bu)_N(x,t) = \sum_{j=1}^N \left(b_j u_j\right)(t)\phi_j(x) \quad b_j \in \mathbb{R}^d \,, \tag{2.11}$$

where $b_j(t)$ is the value of $b$ in point $j$. It is important to be aware that $b$ is from $\mathbb{R}^d$. We can make use of this property. Inserting (2.11) into the $L^2$-product, it can be rewritten in the following way

$$\begin{aligned}
(\nabla \cdot (bu)(t), \phi_i)_{L^2} &= \sum_{j=1}^{N} \left(\nabla \cdot \left((b_j u_j)(t)\phi_j\right), \phi_i\right)_{L^2}, \\
&= \sum_{j=1}^{N} \left(\sum_{k=1}^{d} (b_j u_j)_k(t) \partial_k \phi_j, \phi_i\right)_{L^2}, \quad (b_j u_j)(t)_k \in \mathbb{R}, \\
&= \sum_{k=1}^{d} \left(\sum_{j=1}^{N} (b_j u_j)_k(t) (\partial_k \phi_j, \phi_i)_{L^2}\right), \\
\Rightarrow C_{ij}(t) &= \sum_{k=1}^{d} C^k \cdot b_k(t),
\end{aligned}$$

with $(C^k)_{ij} = \sum_{j=1}^{N} (\partial_k \phi_j, \phi_i)_{L^2}$.

The advantage of this method is that $C^k$ can be assembled once before iterating through the time steps. After that, $d$ matrix-vector multiplications can be executed in each time step.

**Remark 2.3.** *Consider the very simple quadrature formula on an element $\tau$ which belongs to a regular family $\mathcal{J}_h$ of triangulations of $\Omega$*

$$\int_\tau f(x)dx \approx \frac{|\tau|}{N} \sum_{l=1}^{N} f(x_l)$$

*and use it to calculate $C_{ij}(t)$ for both presented methods.*

- *Standard method*

$$\begin{aligned}
(b(t) \cdot \nabla \phi_j, \phi_i)_{L^2(\tau)} &= \int_\tau b(x,t) \cdot \nabla \phi_j(x) \phi_i(x) dx, \\
&\approx \frac{|\tau|}{N} \sum_{l=1}^{N} b(x_l, t) \cdot \nabla \phi_j(x_l) \phi_i(x_l), \\
&= \frac{|\tau|}{N} \sum_{l=1}^{N} \sum_{k=1}^{d} b_k(x_l, t) \partial_k \phi_j(x_l) \phi_i(x_l).
\end{aligned}$$

- *Group finite element method*

$$\begin{aligned}
\sum_{k=1}^{d} (\partial_k \phi_j, \phi_i)_{L^2(\tau)} b_k(t) &= \sum_{k=1}^{d} \int_\tau \partial_k \phi_j(x) \phi_i(x) b_k(x,t) dx, \\
&\approx \frac{|\tau|}{N} \sum_{k=1}^{d} \sum_{l=1}^{N} \partial_k \phi_j(x_l) \phi_i(x_l) b_k(x_l, t).
\end{aligned}$$

*It turns out that both methods generate exactly the same solution if the discretization elements are from $P^1$, but the group finite element method is preferable since the calculation costs are cheaper.*

### 2.1.3 Transport operator of nonlinear fluxes

If our problem contains a nonlinear flux,

$$\frac{\partial u}{\partial t} + \nabla \cdot g(u) = f \quad \text{in} \quad \Omega \times [0, T]$$

we have to discretize it in a special way. This section relates to [Kuz10, p. 53]. Instead of discretizing

$$g(u_n(t))$$

which is dependent on the unknown solution, the idea is to interpolate the flux itself with the same basis function as for $u$.

$$g_n(x,t) = \sum_j g_j(t)\phi_j(x), \quad g_j(t) = g(u_j(t)), \quad u_j(t) = u(x_j, t).$$

Thus the variational form of it yields

$$\begin{aligned}(\nabla \cdot g(u_n), \phi_i)_{L^2} &= \sum_{j=1}^{h} \left(\nabla \cdot (g_j(t)\phi_j(x)), \phi_i(x)\right)_{L^2} \ , \\ &= \sum_{j=1}^{n} g_j(t) \left(\nabla \cdot \phi_j, \phi_i\right)_{L^2} ,\end{aligned}$$

where $(C_{ij}) = (\nabla \cdot \phi_j, \phi_i)_{L^2} = (c_{ij})$. Altogether we receive the following algebraic equation:

$$M_C \frac{\partial u}{\partial t} = Cg(u) + S \ .$$

## 2.2 Model problem for the stationary convection-diffusion equation

The stationary convection-diffusion equation is part of our numerical examples and analysis, therefore it will be described separately in this chapter, although the discretization is similar to the instationary case.

For the model problem to be well-defined we have to assume the following conditions

$$0 < \epsilon,$$
$$-\frac{1}{2}\nabla \cdot b + c \geq 0 \ .$$

Consider the following form of the convection-diffusion equation. For simplification reasons we will assume homogeneous Neumann and Dirichlet boundary conditions

$$-\epsilon \Delta u + b \cdot \nabla u + cu = f \qquad in \quad \Omega \subset \mathbb{R}^d \ , \tag{2.12}$$
$$u(x) = 0 \qquad \forall x \in \Gamma_D \ , \tag{2.13}$$
$$\epsilon \nabla u(x) \cdot \eta = 0 \qquad \forall x \in \Gamma_N \ . \tag{2.14}$$

### 2.2.1 Galerkin discretization

After applying Galerkin discretization we will obtain the following algebraic problem

$$(C + L + R)u = S \ ,$$

where $C$ is the *discrete transport operator*, $L$ is the *discrete diffusion operator*, $R$ is the *reaction matrix* and $S$ represents sources and sinks.

The weak formulation requires a space of test functions. This test space equals the space of the solution $v \in H_0^1(\Omega)$. Multiplying our model problem with a test function $v$, integrating it and applying partial integration we obtain the following form of the weak formulation

$$\int_\Omega b \cdot \nabla uv + cuv \ \ dx - \int_{\Gamma_{N \cup D}} \epsilon \left( \nabla u \cdot \vec{\eta} \right) v \ \ dS + \int_\Omega \epsilon \nabla u \cdot \nabla v \ \ dx = \int_\Omega fv \ \ dx. \tag{2.15}$$

The boundary integral eliminates, because of the homogeneous Neumann condition and the fact that $v$ is zero on the Dirichlet boundary $\Gamma_D$.

Now we can rewrite the variational formulation into $L^2$-products

$$(b \cdot \nabla u, v)_{L^2} + (\epsilon \nabla u, \nabla v)_{L^2} + (cu, v)_{L^2} = (f, v)_{L^2} \quad \forall v \in H_g^1(\Omega) \ .$$

We will take a finite element subspace $V^N \subset H_0^1$ with its basis $B = \{\phi_1, \phi_2, \ldots, \phi_N\}$ to approximate $u$

$$u_N(x) = \sum_{j=1}^{N} u_j \phi_j(x) \quad u_N, \phi_i \in V^N \ .$$

The derivation of each discrete operator will be skipped here, since it is analogously to the instationary case. Obviously there is no *mass matrix* for stationary problems.
The discrete operators for the stationary case are listed below

$$\begin{aligned} C_{ij} &= (b \cdot \nabla \phi_j, \phi_i)_{L^2} \ , \\ L_{ij} &= \epsilon (\nabla \phi_j, \nabla \phi_i)_{L^2} \ , \\ R_{ij} &= \begin{cases} (\mathrm{c}, \phi_\mathrm{i})_{\mathrm{L}^2} & \mathrm{i = j} \ , \\ 0 & \mathrm{i \neq j} \ , \end{cases} \\ S_i &= (f, \phi_i)_{L^2} \ . \end{aligned}$$

## 2.3 Design restrictions

Certain restrictions have to be imposed on the semidiscrete algebraic problem

$$M_C \frac{\partial u}{\partial t} = -(C + L + R)u + S, \tag{2.16}$$

in order to maintain physical properties of the original equation. The following subsections will analyze and define those restrictions.

1. “[. . . ] no mass should be created or destroyed inside the domain by the discretized convective and diffusive terms. " [Kuz10, p. 35]

2. “[. . . ] if a continuous operator produces zero when applied to a constant, so should its discrete counterpart." [Kuz10, p. 36]

3. “[. . . ] if convection and diffusion are the only processes to be simulated, the nodal value $u_i^{n+1}$ should not decrease as result of increasing any other nodal value that appears in the discretized equation for node $i$. Conversely, it should not increase if another nodal value is decreased, all other things being fixed [. . . ] ." [Kuz10, p. 36]

4. “[. . . ] if the discretization of convective and diffusive terms is positivity-preserving, inclusion of a reactive part should not destroy this property." [Kuz10, p. 37]

It is a common technique to replace $M_C$ by its lumped counterpart $M_L$.

### 2.3.1 Mass lumping

In many applications it is useful to have a diagonal mass matrix, since this allows an explicit way of solving the semidiscrete algebraic equation. Using *row-sum mass lumping* results in the diagonal matrix $M_L$

$$\begin{aligned} M_L = diag(m_i), \quad m_i = \sum_j^N m_{ij} \quad \forall i \in \{1, 2, \cdots, N\}, \\ m_i \frac{\partial u_i}{\partial t} = -\sum_j^N (c_{ij} + l_{ij} + r_{ij}) u_j . \end{aligned} \tag{2.17}$$

The lumped mass matrix $M_L$ is a good approximation to $M_C$ for low order finite elements. It conserves mass in the sense that

$$\sum_i^N \sum_j^N m_i u_j = \sum_i^N \sum_j^N m_{ij} u_j .$$

### 2.3.2 Mass conservation

**1. “[...] no mass should be created or destroyed inside the domain by the discretized convective and diffusive terms ..." [Kuz10, p. 35]**.

To describe mass conservation we need to understand that the mass $m|_i$, which belongs to node $i$ is given by

$$m|_i = \sum_j^N m_{ij} u_j \, .$$

Mass conservation means that mass does not change in time. Therefore the derivative in time of the global mass must equal zero:

$$\frac{d}{dt} \sum_i^N \sum_j^N m_{ij} u_j \overset{!}{=} 0 \, .$$

Suppose that there are no sinks and sources, but a diffusion operator $L$ in the discrete formulation

$$\frac{d}{dt} \sum_i^N \sum_j^N m_{ij} u_j = - \sum_j^N \sum_i^N (c_{ij} + l_{ij}) u_j \, . \tag{2.18}$$

Setting the column sums of diffusion and convection operator to zero fulfills condition (2.18) for all $u$

$$\sum_i^N l_{ij} = 0 \, , \quad \sum_i^N c_{ij} = 0 \, .$$

**2. “[...] if a continuous operator produces zero when applied to a constant, so should its discrete counterpart." [Kuz10, p. 36]**

We have to assume that $c$ equals zero, else the operator would not produce zero when applied to a constant. In order to guarantee the second rule, the row sums of transport and diffusion operators have to be zero:

$$\sum_j^N c_{ij} = 0, \quad \sum_j^N l_{ij} = 0 \, .$$

**Remark 2.4.** *As Kuzmin mentions in [Kuz10] all assumptions have to be treated with caution. The row sums of C do not have to be zero if we have a compressible (not divergence-free) velocity field. This means that* $\nabla \cdot b \neq 0$ *and the expression*

$$\nabla \cdot (bu) = (\nabla \cdot b) u + b \cdot \nabla u$$

*cannot be reduced. We will only consider incompressible flows, therefore C must fulfill the zero row-sum criterion for the presented examples .*

**3. “[...] if convection and diffusion are the only processes to be simulated, the nodal value $\mathbf{u_i^{n+1}}$ should not decrease as result of increasing any other nodal value that appears in the discretized equation for node i. Conversely, it should not increase if another nodal value is decreased, all other things being fixed [...] ." [Kuz10, p. 36]**

For this rule we have to assume that there are no sources or sinks, which means that $f$ equals zero and therefore $S$ equals zero. Reaction is not part of this process, therefore $c$ is also zero. The rule can be fulfilled by taking a look at the following useful equivalent formulation of (2.16). By applying *mass lumping* and the *zero row-sum property* we receive

$$m_i \frac{\partial u_i}{\partial t} = - \sum_{j \neq i}^{N} (c_{ij} + l_{ij})(u_j - u_i) \ . \tag{2.19}$$

When it comes to time-discretization we obtain an algebraic equation of the form

$$Au^{n+1} = Bu^n \ ,$$

where $u_i^{n+1}$ is dependent on all other nodal values in the corresponding time step and all values of the previous time-step

$$a_{ii} u_i^{n+1} = \sum_{j}^{N} b_{ij} u_j^n - \sum_{j \neq i} a_{ij} u_j^{n+1} \ .$$

Suppose that one of the nodal values $u_j^n$ for $j \in \{1, 2, \ldots, N\}$ or $u_j^{n+1}$ for $j \neq i$ increased. It would be a bad behavior if $u_i^{n+1}$ decreases thereupon. This means that the corresponding coefficient ($b_{ij}$) or ($a_{ij}$ respectively) has to be nonnegative. We get the following conditions for $A$ and $B$

$$\begin{aligned} a_{ii} > 0, \quad & b_{ii} \geq 0, \quad \forall i \ , \\ a_{ij} \leq 0, \quad & b_{ij} \geq 0 \quad \forall j \neq i \ , \end{aligned}$$

which guarantee the third of our required restrictions.

**Remark 2.5** (positivity-preserving). *A numerical scheme is positivity-preserving if*

$$u^n \geq 0 \quad \Rightarrow \quad u^{n+1} \geq 0 \qquad \forall n$$

*holds. If A is a diagonal matrix this property is fulfilled. It will be discussed later that it is sufficient that A is an M-matrix in order to preserve positivity.*

**4. “[...] if the discretization of convective and diffusive terms is positivity-preserving,**

**inclusion of a reactive part should not destroy this property." [Kuz10, p. 37]**
Due to the diagonal approximation from (2.10)

$$\sum_{j}^{N} \Big(c(t)\phi_j, \phi_i\Big)_{L^2} u_j(t) \approx \left( c(t) \underbrace{\sum_{j}^{N} \phi_j}_{=1}, \phi_i \right)_{L^2} u_i(t) = (c(t), \phi_i)_{L^2}\, u_i(t)\ ,$$

$$\Rightarrow R = \operatorname{diag}\left((c(t), \phi_1)_{L^2}, \dots, (c(t), \phi_N)_{L^2}\right)\ ,$$

the *reaction matrix R* has only positive diagonal entries. Therefore it cannot generate positive off-diagonal entries $a_{ij} > 0$ for $j \neq i$ or nonpositive diagonal entries $a_{ii}$.

## 2.4 Positivity Constraints and Maximum Principles

In this chapter we will analyze maximum principles and positivity constraints of the given partial differential equation. The chapter bases on the book *A Guide to Numerical Methods for Transport Equations* [Kuz10]. We will start with the continuous formulation of the *maximum principle* and the *positivity constraint* and formulate them for the discrete elliptic equation and the semidiscrete parabolic equation of our differential equation. All matrix/vector inequalities in this section are meant to hold componentwise. In this chapter the term $cu$ is contained in $f$ ($f := f - cu$). The most general form of our convection-diffusion equation $\mathcal{L}u = f$

$$\frac{\partial u}{\partial t} + \nabla \cdot (bu - \mathcal{D}\nabla u) = f \qquad \text{in } \Omega \times [0, T]\ , \tag{2.20}$$

$$u(x,t) = g_D(x,t) \qquad \forall (x,t) \in \Gamma_D \times [0,T]\ , \tag{2.21}$$

$$u(x,0) = u_0(x) \qquad \forall (x,t) \in \Omega \times \{0\}\ , \tag{2.22}$$

is of parabolic type. The related stationary problem

$$\nabla \cdot (bu - \mathcal{D}\nabla u) = f \tag{2.23}$$

is of elliptic type. Both types fulfill a set of maximum principles, which have to be taken into consideration in order to maintain these properties for the discretized problem.

**Remark 2.6.** *The following theorems assume that* $\Gamma_D = \partial\Omega$ *. This implies that mixed boundary value problems must not satisfy these theorems.*

In our specific case we have a positive definite symmetric matrix $\mathcal{D} = \operatorname{diag}(\epsilon)$ and an incompressible velocity field $\nabla \cdot b = 0$ which reduces the differential equations to

$$\frac{\partial u}{\partial t} + b \cdot \nabla u - \epsilon \Delta u = f$$

and

$$b\nabla \cdot u - \epsilon \Delta u = f\ .$$

### 2.4.1 Maximum principles for stationary elliptic problems

**Theorem 2.2** (Maximum principle for elliptic problems)**.**
*Let the diffusion tensor $\mathcal{D}$ be symmetric positive definite and $\nabla \cdot b = 0$ in $\Omega$ . Then a solution of problem* (2.23) *satisfies the maximum principle*

$$f \leq 0 \Rightarrow \max_{\bar{\Omega}} u = \max_{\Gamma} g_D \, .$$

**Theorem 2.3** (Positivity constraint)**.**
*Let the diffusion tensor $\mathcal{D}$ be symmetric positive definite in $\Omega$ . Then a solution of problem* (2.23) *with arbitrary $b$ satisfies the positivity constraint*

$$f \geq 0, \quad g_D \geq 0 \Rightarrow u \geq 0.$$

*Here and above, inequalities are meant to hold in the whole range of function values.*

*Proof.* For both proofs see [Kuz10, p. 97]. □

### 2.4.2 Maximum principles for instationary parabolic problems

The solution of parabolic unsteady problems is highly dependent on the initial values $u_0(x)$ and $g_D(x)$.

**Theorem 2.4** (Maximum principle for parabolic problems)**.**
*Let the diffusion tensor $\mathcal{D}$ be symmetric positive definite in $\Omega$. Then a solution of problem* (2.20) - (2.22) *with $\nabla \cdot b = 0$ satisfies*

$$f \leq 0 \Rightarrow \max_{\bar{\Omega}} u = \max_{\Gamma \times [0,T]} g_D \quad \textit{or} \quad \max_{\bar{\Omega}} u = \max_{\Omega} u_0 \, .$$

**Theorem 2.5** (Positivity constraint)**.**
*Let the diffusion tensor $\mathcal{D}$ be symmetric positive definite in $\Omega$. Then a solution of problem* (2.20) - (2.22) *with arbitrary $b$ satisfies*

$$f \geq 0, \quad g_D \geq 0, \quad u_0 \geq 0 \Rightarrow u \geq 0 \, .$$

*Proof.* For both proofs see [Kuz10, p.104]. □

### 2.4.3 Positivity constraints for discrete stationary equations

Since it is important to guarantee physically logic density, temperature and concentration, we want the numerical scheme also to preserve positivity. Discretizing may harm this positivity preserving properties which is why we have to define several constraints for the matrix $A$ in the discretized case

$$Au = b \, . \tag{2.24}$$

*"Let the first $N_\Omega$ nodes be associated with the unknown degrees of freedom, and the rest with the Dirichlet boundary values. This numbering convention implies that the discrete operator A and the vector of nodal values u can be partitioned as follows [...] "* [Kuz10, p. 107]:

$$\bar{A} = \begin{bmatrix} A_{\Omega\Omega} & A_{\Omega\Gamma} \\ 0 & \mathbb{1} \end{bmatrix} \in \mathbb{R}^{(N_\Omega + N_\Gamma)\times(N_\Omega + N_\Gamma)}, \quad b = \begin{bmatrix} b_\Omega \\ g_D \end{bmatrix}, \tag{2.25}$$

where $N_\Omega = \{1, \ldots, N\}$ and $N_\Gamma = \{N+1, \ldots \bar{N}\}$.

In order to prove and set up discrete positivity constraints we will need some special matrix properties.

**Definition 2.2** (Monotone matrix).
*A matrix A is called monotone if*

$$A^{-1} \textit{ exists with } (A^{-1} \geq 0)\,.$$

Monotonicity is important, because it inherits positivity in the sense that

$$Au^{n+1} \geq 0 \Rightarrow u^{n+1} \geq 0\,.$$

This will be of interest when it comes to the positivity constraint property for instationary parabolic equations, where we must show that if the initial condition $u_0$ is greater than zero then all following solutions $u^n$ must also be greater than zero.
To guarantee the existence of the inverse of $A$, we consider a subset of monotone matrices with the following properties:

**Definition 2.3** (M-Matrix).
*A matrix $A = (a_{ij}) \in \mathbb{R}^{n\times n}$ is called M-matrix if:*

1. $a_{ij} \leq 0 \quad$ *for* $\quad i \neq j$ .
2. $A^{-1}$ *exists with* $(A^{-1} \geq 0)$ . *(monotone matrix)*

*"The M-matrix property is widely used to prove discrete maximum principles (DMP) for finite element discretizations of elliptic and parabolic problems."* [Kuz08, p. 2520]
In real applications it is too expensive to check if $A$ has an inverse. Especially when it comes to time-dependent problems this check must be done in each time step. Therefore, there exists a subset of $M$-matrices with the following properties.

**Definition 2.4** (Nonnegative-type matrix).
*A matrix $A = (a_{ij})$ is said to be of nonnegative-type if*

$$a_{ii} > 0, \quad \forall i\,, \tag{2.26}$$

$$a_{ij} \leq 0, \quad \forall j \neq i\,, \tag{2.27}$$

$$\sum_j^N a_{ij} \geq 0, \quad \forall i\,. \tag{2.28}$$

**Remark 2.7.** *“A nonnegative-type matrix A is an M-matrix if inequality* (2.28) *is strict or A is irreducible and* (2.28) *is strict for at least one row." [Kuz10, p. 110]*
*The proof is skipped here and can be looked up in [Kuz10, p. 110-111].*

It is interesting that conditions (2.26)-(2.27) are already given by the third rule of our design restrictions in Chapter 2.3. Part three (2.28) is the zero row-sum property from the second design restriction.
We will now define the discrete equivalent of the *maximum principle for elliptic problems* (2.2).

**Theorem 2.6** (Global discrete maximum principle).
*If the matrix $\bar{A}$ is given by* (2.25), *$A_{\Omega\Omega}$ is monotone, $A_{\Omega\Gamma} \leq 0$, and*

$$\sum_{j}^{\bar{N}} a_{ij} = 0 \quad \forall i \in N_\Omega\,, \tag{2.29}$$

*holds, then the solution of* (2.24) *satisfies the global discrete maximum principle*

$$b_\Omega \leq 0 \Rightarrow \max_i u_i = \max_j g_{Dj}\,. \tag{2.30}$$

*Proof.* See [Kuz10, p. 111]. We will show that $w_\Omega \leq 0$ with $w = u - \mu$ and $\mu = \max_j g_{Dj}$. Using the zero row-sum property (2.29) for $\bar{A}$ we obtain

$$\sum_{j=1}^{N} a_{ij}w_j = \sum_{j=1}^{N} a_{ij}u_j - \mu \sum_{j=1}^{N} a_{ij} = b_\Omega \leq 0\,.$$

With the matrix formulation in (2.25) this yields

$$\begin{aligned} & A_{\Omega\Omega}w_\Omega + A_{\Omega\Gamma}w_\Gamma = b_\Omega \leq 0\,, \\ \Rightarrow\quad & w_\Omega = A_{\Omega\Omega}^{-1}[b_\Omega - A_{\Omega\Gamma}w_\Gamma]\,. \end{aligned}$$

Because $(w_\Gamma)_i = g_i - \max_j g_j \leq 0$ it follows that $A_{\Omega\Gamma}w_\Gamma \geq 0$. Because of the monotonicity of $A_{\Omega\Omega}$ this implies that $w_\Omega \leq 0$.

□

As mentioned before it is not practical to calculate $A^{-1}$. Therefore a weaker *discrete maximum principle* for stationary problems can be formulated by using the nonnegativity of $\bar{A}$.

**Theorem 2.7** (Local discrete maximum principle).
*If the matrix $\bar{A}$ is of nonnegative-type and*

$$\sum_{j}^{\bar{N}} a_{ij} = 0, \quad \forall i \in N_\Omega$$

*holds, then the solution of* (2.24) *satisfies the local discrete maximum principle*

$$b_i \leq 0 \quad \Rightarrow \quad u_i \leq \max_{j \in \mathcal{N}_i} u_j \quad \forall i \in N_\Omega\,,$$

*where $\mathcal{N}_i := \{j \neq i | a_{ij} \neq 0\}$ is the set of neighbors that form the stencil of node i.*

*Proof.* The proof will be skipped here. The interested reader is referred to [Kuz10, p. 111]. □

**Remark 2.8.** *If A is obtained by a finite element discretization it is in general possible to prove the global discrete maximum principle by its local counterpart.*

**Theorem 2.8** (Discrete positivity constraint for stationary elliptic problems)**.**
*If the matrix $\bar{A}$ is given by* (2.25)*, where $A_{\Omega\Omega}$ is monotone and $A_{\Omega\Gamma} \leq 0$, then the discretization* (2.24) *is positivity-preserving that is,*

$$b \geq 0 \quad \Rightarrow u \geq 0 \, .$$

*Proof.* The proof can be found in [Kuz10, p. 112]. The inverse of $\bar{A}$ is

$$\bar{A}^{-1} = \begin{bmatrix} A_{\Omega\Omega}^{-1} & -A_{\Omega\Omega}^{-1}A_{\Omega\Gamma} \\ 0 & \mathbb{1} \end{bmatrix} \in \mathbb{R}^{(N_\Omega+N_\Gamma)\times(N_\Omega+N_\Gamma)} \, .$$

All components are nonnegative therefore

$$b \geq 0 \quad \Rightarrow \bar{A}^{-1}b = u \geq 0 \, .$$

□

### 2.4.4 Positivity constraints for semidiscrete instationary equations

Similar to the analysis of the elliptic instationary discrete formulation we want to formulate discrete maximum principles for the semi discrete algebraic equation

$$M_C \frac{\partial u}{\partial t} = -(C + L + R)u + S \, . \tag{2.31}$$

It shall be mentioned here that Kuzmin writes down all following theorems only for problems without the *discrete diffusion operator* $L$ and the reactive term $R$ is hidden in $S$. Since it is part of our parabolic problem, all these statements are expanded by $L$ and adjusted such that $R_i u_i$ is a stand-alone term. For all further observations it is assumed that $M = M_L$ is diagonal and the zero row-sum property is used to rewrite the semidiscrete formulation into

$$m_i \frac{\partial u_i}{\partial t} = -\sum_{j \neq i} (c_{ij} + l_{ij})(u_j - u_i) + S_i - R_i u_i \, . \tag{2.32}$$

**Theorem 2.9** (Local semidiscrete maximum principle for instationary problems)**.**
*If $m_i > 0$ for all i, $c_{ij} + l_{ij} \leq 0$ for all $j \neq i$ then the local semidiscrete maximum principles*

$$\begin{aligned} u_i \geq u_j, \quad \forall j \in \mathcal{N}_i \quad &\Rightarrow \quad \frac{\partial u_i}{\partial t} \leq 0 \quad \textit{for} \quad S_i - R_i u_i \leq 0 \, , \\ u_i \leq u_j, \quad \forall j \in \mathcal{N}_i \quad &\Rightarrow \quad \frac{\partial u_i}{\partial t} \geq 0 \quad \textit{for} \quad S_i - R_i u_i \geq 0 \, , \end{aligned}$$

*hold for* (2.32).

*Proof.* It is

$$\frac{\partial u_i}{\partial t} = -\frac{1}{m_i}\sum_{j\in\mathcal{N}_i}(c_{ij}+l_{ij})\underbrace{(u_j-u_i)}_{\leq 0} + \underbrace{\frac{S_i}{m_i} - \frac{R_i u_i}{m_i}}_{\leq 0},$$

$$\Rightarrow \quad \frac{\partial u_i}{\partial t} \leq 0\,.$$

The proof of the second inequality is performed analogously. [Kuz10, p. 116] □

**Theorem 2.10** (Positivity constraint for semidiscrete equations)**.**
*Given the following properties*

$$m_i > 0, \quad S_i - R_i u_i \geq 0 \quad \forall i, \quad c_{ij} + l_{ij} \leq 0 \quad \forall j \neq i, \tag{2.33}$$

*then a semidiscrete equation is called positivity preserving in the sense that the following estimate holds for the solution vector $u_i$*

$$u_j(0) \geq 0 \quad \forall j \quad \Rightarrow \quad u_i(t) \geq 0 \quad \forall i, \forall t > 0\,. \tag{2.34}$$

*Proof.* See [Kuz10, p.117-118]. The proof is similar to the previous proof. The time derivative of $u_i$ is greater or equal than zero and therefore $u_i$ can only increase in time. □

**Remark 2.9.** *"[...] the numerical solution is not forced to be positive if $\exists\ i \neq j$ such that $u_j(0) < 0$. Positivity preservation means that the numerical scheme cannot produce nonphysical negative values." [Kuz09, p. 2519]*

### 2.4.5 Local extremum diminishing LED

A very useful property is the *Local Extremum Diminishing property LED.*

**Theorem 2.11** (LED)**.**
*Suppose that*

$$m_i > 0 \quad \forall i\,, \quad S_i - R_i u_i = 0 \quad \textit{and} \quad c_{ii} + l_{ii} = -\sum_{j\neq i}(c_{ij}+l_{ij})\,,$$

*then it follows from the local semidiscrete maximum principle that*

$$u_i \geq u_j \quad \forall j \neq i \quad \Rightarrow c_{ij}(u_j - u_i) \leq 0 \quad \Rightarrow \frac{\partial u_i}{\partial t} \leq 0 \quad \textit{and} \tag{2.35}$$

$$u_i \leq u_j \quad \forall j \neq i \quad \Rightarrow c_{ij}(u_j - u_i) \geq 0 \quad \Rightarrow \frac{\partial u_i}{\partial t} \geq 0. \tag{2.36}$$

*Proof.* The proof is analogously to the proof of Theorem 2.9 □

Thus a local maximum (minimum) can only diminish (grow).

**Remark 2.10.**
*If $f$ and $cu$ are nonzero it is very unlikely that the LED property holds. One example (traveling wave) will be discussed in Chapter 3.*

- *"In one space dimension, the LED property guarantees that the total variation of the discrete solution is a nonincreasing function of time. Thus, one-dimensional LED schemes are total variation diminishing (TVD). " [Kuz09, p. 2520]*
- *"By the Godunov theorem [...], a linear positivity-preserving/LED discretization of a hyperbolic transport equation can be at most first-order accurate" [KLT05, p. 149] .*
- *The LED property is only of interest if the continuos solution also does not develop maxima or minima.*

### 2.4.6 Positivity constraints for fully discrete instationary equations

The fully discrete formulation of the parabolic model problem has the following form

$$\begin{bmatrix} A^{n+1}_{\Omega\Omega} & A^{n+1}_{\Omega\Gamma} \\ 0 & \mathbb{1} \end{bmatrix} \begin{bmatrix} u^{n+1}_{\Omega} \\ u^{n+1}_{\Gamma} \end{bmatrix} = \begin{bmatrix} B^{n}_{\Omega\Omega} & B^{n}_{\Omega\Gamma} \\ 0 & \mathbb{1} \end{bmatrix} \begin{bmatrix} u^{n}_{\Omega} \\ u^{n}_{\Gamma} \end{bmatrix} + \begin{bmatrix} S^{n}_{\Omega} \\ S^{n}_{\Gamma} \end{bmatrix} . \tag{2.37}$$

The superscript indicates the time-level and $u_i^0$ equals $u_i(0)$ for all $i$. It is important to have in mind that this algebraic equation does not have to be explicit. We will see in Chapter 2.5 that the matrix $A$ may be dependent on the future time step $u^{n+1}$ if the flux $g(u)$ is of nonlinear type. We will neglect the superscripts of (2.37) in the following analysis if there is no confusion possible.

**Theorem 2.12** (Global maximum principle for fully discrete instationary equations)**.**
*Let the first $N_\Omega$ row-sums of $A$ and $B$ be equal, i.e.,*

$$\sum_{j=1}^{\bar{N}} a_{ij} = \sum_{j=1}^{\bar{N}} b_{ij} \quad \forall i \in N_\Omega ,$$

*and let the block $A_{\Omega\Omega}$ be regular. Then the solution $u^{n+1}_\Omega$ of problem* (2.37) *satisfies the global discrete maximum principle*

$$S_\Omega \leq 0 \quad \Rightarrow \quad u_i^{n+1} \leq \max_j g_j, \quad \forall i \in N_\Omega ,$$

*under the following sign conditions*

$$A^{-1}_{\Omega\Omega} \geq 0, \quad A_{\Omega\Gamma} \leq 0, \quad B_{\Omega\Omega} \geq 0, \quad B_{\Omega\Gamma} \geq 0 .$$

*Proof.* The proof is very similar to the proof of the *global discrete maximum principle for discrete stationary equations* (2.30). It has to be shown that $w = u^{n+1} - \mu$ with $\mu = \max_j g_j$ is nonpositive. It is skipped here and the interested reader is referred to [Kuz10, p.119]. □

**Theorem 2.13** (Local maximum principle for fully discrete instationary equations)**.**
*The solution of* (2.37) *satisfies the local maximum principle for fully discrete instationary equations*

$$S_i \leq 0 \quad \Rightarrow \quad u_i \leq \mu_i \quad \forall i \in N_\Omega \,,$$

*where $\mu_i$ denotes the maximum taken over $\{u_j | a_{ij} \neq 0, j \neq i\}$ and $\{g_j | b_{ij} \neq 0\}$, if the row-sum constraint holds and if the conditions of the third basic rule from Section 2.3, i.e.,*

$$a_{ii} > 0, \quad b_{ii} \geq 0, \quad \forall i \,, \tag{2.38}$$

$$a_{ij} \leq 0, \quad b_{ij} \geq 0, \quad \forall j \neq i \tag{2.39}$$

*hold.*

*Proof.* See [Kuz10, p. 119]. □

**Theorem 2.14** (Positivity constraint for fully discrete instationary equations)**.**
*If the coefficients of* (2.37) *satisfy conditions* (2.38) - (2.39) *and*

$$\sum_{j=1}^{\bar{N}} a_{ij} > 0, \quad \forall i \in N_\Omega \,, \tag{2.40}$$

*then such a discretization is guaranteed to be positivity-preserving that is,*

$$S_\Omega \geq 0, \quad u^n \geq 0 \quad \Rightarrow u_\Omega^{n+1} \geq 0 \,.$$

*Proof.* Because of (2.38)- (2.39) $A_{\Omega\Omega}$ is of nonnegative-type and because of (2.40) it is even an $M$-matrix. Therefore

$$u_\Omega^{n+1} = A_{\Omega\Omega}^{-1}(B_{\Omega\Omega} u_\Omega^n + B_{\Omega\Gamma} u_\Gamma^n - A_{\Omega\Gamma} u_\Gamma^n - A_{\Omega\Gamma} + S_\Omega) \geq 0 \,,$$

holds, see [Kuz10, p. 119]. □

## 2.5 Time discretization

After discretization in time one obtains an algebraic system of the form

$$\begin{bmatrix} A_{\Omega\Omega}^{n+1} & A_{\Omega\Gamma}^{n+1} \\ 0 & \mathbb{1} \end{bmatrix} \begin{bmatrix} u_\Omega^{n+1} \\ u_\Gamma^{n+1} \end{bmatrix} = \begin{bmatrix} B_{\Omega\Omega}^n & B_{\Omega\Gamma}^n \\ 0 & \mathbb{1} \end{bmatrix} \begin{bmatrix} u_\Omega^n \\ u_\Gamma^n \end{bmatrix} + \begin{bmatrix} S_\Omega^n \\ S_\Gamma^n \end{bmatrix} .$$

After the previous analysis we can summarize that the following conditions must be fulfilled in order to preserve positivity:

1. $A_{\Omega\Omega}$ is an $M$-matrix and $A_{\Omega\Gamma} \leq 0$

$$\begin{aligned} a_{ii} &> 0 \quad \forall i \in N_\Omega \,, \\ a_{ij} &\leq 0 \quad \forall j \neq i \,, i \in \bar{N} \,, \\ \sum_j^{\bar{N}} a_{ij} &> 0 \quad \forall i \in N_\Omega \,. \end{aligned}$$

2. $B_{\Omega\Omega}$ and $B_{\Omega\Gamma}$ are nonnegative ($\geq 0$). This ensures the third rule of the design restrictions.

If additionally the row-sums of $A$ and $B$ are equal for all $i$ in $N_\Omega$

$$\sum_{j=1}^{\bar{N}} a_{ij} = \sum_{j=1}^{\bar{N}} b_{ij} \quad \forall i \in N_\Omega \,,$$

even the global maximum principle holds.

**Remark 2.11.** *It shall be noticed that the full discretization can only be positivity preserving if the semidiscrete form is also positivity preserving.*

### 2.5.1 $\theta$ -Scheme

It will be assumed for all following studies that $S$ is time and solution independent. One possible approach is the two-level $\theta$-scheme. With the implicitness parameter $0 \leq \theta \leq 1$ we obtain from

$$M_C \frac{\partial u_i}{\partial t} = -(C + L + R)u + S \,,$$

the following scheme

$$M_C \frac{u^{n+1} - u^n}{\Delta t} = -\left(\theta(C^{n+1} + L^{n+1} + R^{n+1})u^{n+1} + (1-\theta)(C^n + L^n + R^n)u^n\right) + S \,.$$

With $\theta = 0/1$ we receive first order forward/backward Euler-schemes. Choosing $\theta = \frac{1}{2}$ (Crank-Nicolson) the system turns out to be implicit and of second order. The matrices of the algebraic system have the following forms

$$A = M_C + \Delta t\theta(C^{n+1} + L^{n+1} + R^{n+1}) \quad \text{and} \quad B = M_C - \Delta t(1-\theta)(C^n + L^n + R^n) \,.$$

When it comes to time discretization “[...] an upper bound may need to be imposed on the time step $\Delta t = t^{n+1} - t^n$ . This bound can be derived using the concept of monotone matrices.“ [Kuz09, p. 2520]. This will be analyzed in the following theorem. It bases on [Kuz10, p. 121-122].

**Theorem 2.15** (Positivity constraint for the $\theta$-scheme)**.**
*A $\theta$-scheme is positivity preserving if the space discretization fulfills*

$$(c_{ij} + l_{ij}) \leq 0 \quad \forall j \neq i\,, \quad \textit{and} \quad (c_{ii} + l_{ii} + R_i) > 0\,,$$

*and if the time step $\Delta t$ is in the following range*

$$\Delta t \geq -\frac{m_{ij}}{\theta(c_{ij} + l_{ij})} = \Delta t_{\min}$$
$$\Delta t < \min\left\{-\frac{\sum_{j=1}^{\bar{N}} m_{ij}}{\theta \sum_{j=1}^{\bar{N}}(c_{ij} + l_{ij})}, \frac{m_{ii}}{(1-\theta)(c_{ii} + l_{ii} + R_i)}\right\} = \Delta t_{\max}\,.$$

*Proof.*
We will analyze each restriction arising from positivity preservation separatly

$$\begin{aligned}
a_{ij} \leq 0 \quad &\Rightarrow \quad \Delta t \geq \frac{-m_{ij}}{\theta(c_{ij} + l_{ij})} > 0 && \forall i \neq j\,, i \in \bar{N}\,,\\
a_{ii} > 0 \quad &\Rightarrow \quad \Delta t > \frac{-m_{ii}}{\theta(c_{ii} + l_{ii} + R_i)} < 0 && \forall i \in N_\Omega\,,\\
\sum_{j=1}^{\bar{N}} a_{ij} > 0 \quad &\Rightarrow \quad \Delta t < -\frac{\sum_{j=1}^{\bar{N}} m_{ij}}{\theta \sum_{j=1}^{\bar{N}}(c_{ij} + l_{ij}) + \theta R_i} > 0 && \forall i \in N_\Omega\,, \qquad (2.41)\\
b_{ij} \geq 0 \quad &\Rightarrow \quad \Delta t \geq \frac{m_{ij}}{(1-\theta)(c_{ij} + l_{ij})} < 0 && \forall i \neq j\,,\\
b_{ii} \geq 0 \quad &\Rightarrow \quad \Delta t \leq \frac{m_{ii}}{(1-\theta)(c_{ii} + l_{ii} + R_i)} > 0 && \forall i \neq j\,.
\end{aligned}$$

There are no restrictions for $\Delta t$ coming from $a_{ii}$ and $b_{ij}$, because they only require $\Delta t$ to be positive, which is given. In order to satisfy (2.41) we have to assume that $\theta \sum_{j=1}^{\bar{N}}(c_{ij} + l_{ij}) + \theta R_i$ is negative. □

**Remark 2.12.** *If we assume that $M$ is a diagonal matrix, this is in particular the case if $M_L$ is the lumped mass matrix, then there is no lower bound for $\Delta t$.*

## 2.5.2 Runge-Kutta scheme

Another approach for solving

$$M_C \dot{u} = -(C + L + R)u + S$$

is by using an *explicit Runge-Kutta method*, where

$$\begin{aligned}
y'(t) &= f(t, y(t))\\
y_{n+1} &= y_n + \Delta t \sum_{j=1}^{s} b_j k_j\\
k_j &= f\left(t_n + \Delta t c_j, y_n + \Delta t \sum_{l=1}^{s} a_{jl} k_l\right)\,.
\end{aligned}$$

We can rewrite the $k's$ and $y_n$ according to our problem

$$\begin{aligned} k_j &= -(C^{(j)} + L^{(j)} + R^{(j)})u^{(j)} + S \ , && j = 1 \dots s \ , \\ M_C u^{n+1} &= M_C u^n + \Delta t \sum_{l=1}^{s} b_j k_j \ , && j = 1 \dots s \ , \end{aligned}$$

$$\begin{aligned} C^{(j)} &= C(t_n + \Delta t c_j), \quad C^n = C^{(1)}, \quad M_C u^{(j)} = M_C u^n + \Delta t \sum_{l=1}^{j-1} a_{jl} k_j \ , \\ L^{(j)} &= L(t_n + \Delta t c_j), \quad L^n = L^{(1)} \ , \\ R^{(j)} &= R(t_n + \Delta t c_j), \quad R^n = R^{(1)} \ . \end{aligned}$$

The *diffusion operator* is time-independent, therefore it holds that $L^{(j)} = L$.

**Example 2.5.1** (Heun-Scheme)**.**
The *Heun-scheme* is a second order explicit Runge-Kutta scheme with the following Butcher tableau

$$\begin{array}{c|cc} 0 & & \\ 1 & 1 & \\ \hline & \frac{1}{2} & \frac{1}{2} \ . \end{array}$$

*“The optimal (in terms of the time step restriction and computational cost) strong stability-preserving Runge-Kutta scheme of second order is the well-known Heun method."* [Kuz10, p.123]

$$\begin{aligned} M_C u^{n+1} &= M_C u^n + \Delta t \frac{1}{2} \left[ -(C^n + L^n + R^n) M_C u^n + S \right] \\ &\quad - \Delta t \frac{1}{2} \left[ (C^{n+1} + L^{n+1} + R^{n+1})(M_C u^n - \Delta t (C^n + L^n + R^n) M_C u^n + \Delta t S) \right] \end{aligned}$$

## 2.6 Algebraic flux correction

*“The basic idea is very simple: if a given discretization fails to satisfy the sufficient conditions of the discrete maximum principle, they can be enforced by adding a discrete diffusion operator that adjusts itself adaptively to the local solution behavior.“* [Kuz10, p. 126]

We will devise a (semi) discrete diffusive equation which is of low (first) order but guarantees all maximum principles and positivity restrictions called *good part*

$$\begin{aligned} (\hat{C} + \hat{L} + R)u &= S && \text{(stationary discrete elliptic equation)} \ , \\ M_L \frac{\partial u}{\partial t} &= -(\hat{C} + \hat{L} + R)u + S && \text{(instationary semidiscrete parabolic equation)} \ , \end{aligned}$$

by adding a so-called artificial diffusion. The difference between the general (semi-) discrete equation and the *good part* is called *bad* or antidiffusive part $f$

$$f = (C - \hat{C} + L - \hat{L})u \qquad \text{(stationary)}\,,$$

$$f = (M_L - M_C)\frac{\partial u}{\partial t} - (C - \hat{C} + L - \hat{L})u \qquad \text{(instationary)}\,.$$

We obtain the original high order equation by adding the antidiffusive fluxes to the *good part*

$$(\hat{C} + \hat{L} + R)u = S + f \qquad \text{(stationary)}\,.$$

$$M_C\frac{\partial u}{\partial t} = -(\hat{C} + \hat{L} + R)u + S + f \qquad \text{(instationary)}\,. \tag{2.42}$$

### 2.6.1 Design idea

The following chapter is based on [KM05]. In contrast to the underlying paper, where the *discrete transport operator* is named $K$, we will stick to our previous notation $C$. Since high order finite element methods of numerical solutions to convection-dominated flow problems result in non-physical oscillations in the vicinity of steep gradients, we want to introduce the flux-corrected transport (FCT) algorithm of Boris and Book and flux-limiters.

- The first approach is to introduce an artificial diffusion operator $D = \{d_{ij}\}$, which suppresses oscillations and enforces the positivity constraint restrictions. The new modeled discrete transport operator is of the form $\hat{C} = \{\hat{c}_{ij}\} = c_{ij} + d_{ij}$.
  *“In addition, we require D to be a symmetric matrix with zero row and column sums. These properties provide consistency and mass conservation."* [Kuz09, p. 2522]
- Physical diffusion $L$ has to be split into a positive part $L^+ = \{l^+_{ij}\}$ and the remainder $\hat{L} := L - L^+$.
- To correct this artificially inserted diffusion we use antidiffusive fluxes $f_i$. These fluxes shall have the following properties:
  1. No new maxima or minima must be generated (see definition LED 2.11).
  2. Existing extrema cannot grow.
- *“Approximate M by its lumped counterpart $M_L$.“* [Kuz09, p. 2522]

*“Roughly speaking, the high-order method is used in regions where the solution is sufficiently smooth and the low-order method elsewhere"* [Kuz08, p. 3]. This means that especially at layers we will need more artificial diffusion to suppress spurious oscillations.

The antidiffusive fluxes shall maintain specific properties of the known analytical solution as positivity, monotonicity and nonincreasing total variation. Therefore certain restraints are imposed on the antidiffusive fluxes. This FCT algorithm is presented by Zalesak [Zal79].
The antidiffusive fluxes will depend on the unknown solution, which involves using an iterative solution scheme.

**Definition 2.5** (Artificial Diffusion operator $D$)**.**
*An artificial diffusion operator D is symmetric and has zero row and column sums*

$$d_{ij} = d_{ji}, \quad \sum_{j=1} d_{ij} = 0, \quad \sum_{i=1} d_{ij} = 0 \; .$$

We will define $D$ in the following way:

$$\begin{aligned} d_{ij} &= -max\{c_{ij}, 0, c_{ji}\} = d_{ji} \quad \forall j \neq i \; , \\ d_{ii} &:= -\sum_{j \neq i} d_{ij} \; . \end{aligned} \tag{2.43}$$

The physical diffusion will be split as follows

$$\begin{aligned} \hat{L} &:= L - L^+ \; , \\ l_{ij}^+ &= \max\{0, l_{ij}\} \qquad\qquad \forall j \neq i \; , \\ l_{ii}^+ &= -\sum_{j \neq i} l_{ij}^+ \; . \end{aligned} \tag{2.44}$$

**Remark 2.13.** *The new operators $\hat{C} = C + D$ and $\hat{L} = L - L^+$ are positivity preserving and fulfill the maximum principles for elliptic and parabolic problems, because*

- $\hat{c}_{ij} := \min\{0, c_{ij}, c_{ij} - c_{ji}\}$ *is nonpositive for all $i \neq j$,*
- *if* $\sum_j c_{ij} = 0$ *it follows that* $\sum_j \hat{c}_{ij} = 0$*,*
- $\hat{l}_{ij} := \min\{l_{ij}, 0\}$ *is nonpositive for all $i \neq j$,*
- *if* $\sum_j l_{ij} = 0$ *it follows that* $\sum_j \hat{l}_{ij} = 0$*.*

The described definition of artificial diffusion can be found in [KM05, p. 150]. In the algorithm used to calculated the examples in Chapter 3, artificial diffusion is not split into a convective and a diffusive part but defined as follows (see [BJK16, p. 4]):

$$\begin{aligned} d_{ij} &= -max\{c_{ij} + l_{ij}, 0, c_{ji} + l_{ji}\} = d_{ji} \quad \forall j \neq i \; , \\ d_{ii} &:= -\sum_{j \neq i} d_{ij} \; . \end{aligned}$$

In order to calculate the new discrete operators $\hat{C} := C + D$ and $\hat{L}$ it is possible to calculate $C$ and $L$ successively instead of assembling $C$, $D$ and $L$, $L^+$ directly by setting:

$$\begin{aligned} &\hat{C} := C \; , \\ &\hat{c}_{ii} := c_{ii} + d_{ij} \qquad \hat{c}_{ij} := c_{ij} + d_{ij} \quad i \neq j \; , \\ &\hat{c}_{ji} := c_{ji} + d_{ji} \qquad \hat{c}_{jj} := c_{jj} + d_{ji} \quad j \neq i \; , \end{aligned}$$

and

$$\begin{aligned}
&\hat{L} := L\,,\\
&\hat{l}_{ii} := l_{ii} + l_{ij}^{+} \qquad \hat{l}_{ij} := l_{ij} - l_{ij}^{+} \quad i \neq j\,,\\
&\hat{l}_{ji} := l_{ji} - l_{ij}^{+} \qquad \hat{l}_{jj} := l_{jj} + l_{ij}^{+} \quad j \neq i\,.
\end{aligned}$$

With this approach we construct two different equations. The first one is the original higher order equation, which causes nonphysical oscillations and contains physical diffusion. The second one replaces the mass matrix $M_C$ by its lumped counterpart $M_L$, adds artificial diffusion $D$ to $C$ and subtracts positive values of $L$.
We define the residuals as follows

$$\begin{aligned}
r_L &= (\hat{C} + \hat{L} + R)u - S\ , && \text{(stationary equation)}\\
r_C &= (C + L + R)u - S\ , &&\\
r_L &= M_L\frac{\partial u}{\partial t} + (\hat{C} + \hat{L} + R)u - S\ , && \text{(instationary equation)}\\
r_C &= M_C\frac{\partial u}{\partial t} + (C + L + R)u - S\ . &&
\end{aligned}$$

Subtracting $r_C$ and $r_L$ we obtain

$$\begin{aligned}
r_L - r_C &= Du - L^{+}u && \text{(stationary equation)}\ ,\\
r_L - r_C &= (M_L - M_C)\frac{\partial u}{\partial t} + Du - L^{+}u && \text{(instationary equation).}
\end{aligned}$$

Due to (2.43), (2.44) and mass lumping we can write

$$\begin{aligned}
[Du]_i &= \sum_{j}^{N} d_{ij}u_j = \sum_{j\neq i}^{n} d_{ij}u_j - \sum_{j\neq i}^{n} d_{ij}u_i = \sum_{j\neq i}^{n} d_{ij}(u_j - u_i)\ ,\\
[L^{+}u]_i &= \sum_{j}^{N} l_{ij}^{+}u_j = \sum_{j\neq i}^{n} l_{ij}^{+}u_j - \sum_{j\neq i}^{n} l_{ij}^{+}u_i = \sum_{j\neq i}^{n} l_{ij}^{+}(u_j - u_i)\ ,\\
[M_Lu - M_Cu]_i &= m_iu_i - \sum_{j}^{N} m_{ij}u_j = \sum_{j\neq i}^{n} m_{ij}(u_i - u_j)\ .
\end{aligned}$$

**Example 2.6.1** (Convection-diffusion equation in 1-D**).**
In [Kuz08, p. 80] one can find an example which shows that artificial diffusion turns a second order discretization into a first order upwind scheme. The $P1$-Galerkin discretization of the following one dimensional convection-diffusion equation

$$v\frac{\partial u}{\partial x} - \epsilon\frac{\partial^2}{\partial^2 x} = 0, \quad v > 0, \quad \epsilon > 0\ ,$$

yields

$$\frac{v}{2\Delta x}\underbrace{\begin{bmatrix} \ddots & & & & \\ -1 & 0 & 1 & & \\ & -1 & 0 & 1 & \\ & & -1 & 0 & 1 \\ & & & & \ddots \end{bmatrix}}_{C} u + \frac{\epsilon}{\Delta x^2}\underbrace{\begin{bmatrix} \ddots & & & & \\ -1 & 2 & -1 & & \\ & -1 & 2 & -1 & \\ & & -1 & 2 & -1 \\ & & & & \ddots \end{bmatrix}}_{L} u = 0\,.$$

Since the physical diffusion operator $L$ has nonpositive off-diagonals and its row-sums are zero it follows that $L$ equals $\hat{L}$. The artificial diffusion $D$ is

$$D = \frac{v}{2\Delta x}\begin{bmatrix} \ddots & & & & \\ -1 & 2 & -1 & & \\ & -1 & 2 & -1 & \\ & & -1 & 2 & -1 \\ & & & & \ddots \end{bmatrix}\,. \tag{2.45}$$

Therefore the new low order equation sums up to

$$\frac{v}{2\Delta x}\underbrace{\begin{bmatrix} \ddots & & & & \\ -2 & 2 & 0 & & \\ & -2 & 2 & 0 & \\ & & -2 & 2 & 0 \\ & & & & \ddots \end{bmatrix}}_{\hat{C}} u + \frac{\epsilon}{\Delta x^2}\underbrace{\begin{bmatrix} \ddots & & & & \\ -1 & 2 & -1 & & \\ & -1 & 2 & -1 & \\ & & -1 & 2 & -1 \\ & & & & \ddots \end{bmatrix}}_{\hat{L}} u = 0\,,$$

which is equivalent to the first order upwind-scheme.

**Definition 2.6** (Algebraic fluxes)**.**
*Under the previously described conditions the correction fluxes $f_i$ are given by*

$$\begin{aligned} f_{ij} &= [d_{ij} - l_{ij}^{+}](u_j - u_i)\,, && \textit{(stationary)} \\ f_{ij} &= [m_{ij}\frac{\partial u}{\partial t} - d_{ij} + l_{ij}^{+}](u_i - u_j)\,, && \textit{(instationary)} \\ f_i &= \sum_{j\neq i}^{n} f_{ij}\,, \quad f_{ji} = -f_{ij}\,. \end{aligned}$$

*They can be interpreted as raw antidiffusive fluxes from node j into node i (see [KM05, p.13]).*

**Remark 2.14.** *Because of the skew-symmetric property of $f_{ij}$ and $f_{ji}$ every contribution of one node is reversed by its negative counterpart. This preserves mass conservation.*

In fact, form (2.42) is just another way to rewrite our original equation, but the explicit appearance of fluxes makes it possible to add a specific amount of diffusion to our equation depending on the solution $u$ and therefore on its vicinities. The idea is to introduce solution-dependent correction factors $\alpha_{ij}(u_h) \in [0, 1]$ such that we can define new fluxes

$$f_{ij}^{\alpha} := \alpha_{ij} f_{ij} ,$$

where for $\alpha = 1$ the equation remains the original one and for $\alpha = 0$ it is the highly diffusive equation. Therefore $\alpha$ must be close to 1 in smooth regions and 0 at steep gradients.

$$(\hat{C} + \hat{L} + R)u = S + f^{\alpha} , \qquad \text{(stationary)} \tag{2.46}$$

$$M_L \frac{\partial u}{\partial t} = -(\hat{C} + \hat{L} + R)u + S + f^{\alpha} . \qquad \text{(instationary)} \tag{2.47}$$

Flux limiters turn an explicit equations into an implicit equation. It implies that iterative schemes have to be used in order to solve the equations.

## 2.7 FEM-FCT schemes for stationary equations

This section will describe common methods for solving (2.46). It is based on [Kuz10, p.141 ff.]. In order to solve (2.46) a *defect-correction-scheme* will be used. This scheme can reduce iterations compared with a fixed-point method

$$u^{(m+1)} = u^{(m)} + \omega^{(m)} \left[\bar{A}^{(m)}\right]^{-1} r^{(m)} ,$$

with $r^{(m)}$ being the residuum or the *defect*

$$r^{(m)} = S - \left(\hat{C}^{(m)} + \hat{L}^{(m)} + R^{(m)}\right) u^{(m)} + f^{\alpha}\left(u^{(m)}\right) ,$$

$[\bar{A}^{(m)}]$ being a suitable preconditioner and $0 < \omega^{(m)} \leq 1$ being a relaxation parameter, which controls the amount of the correction step. Each update needs the inverse of $\bar{A}$. This can be done by solving the following linear system

$$\bar{A}^{(m)} \Delta u^{(m+1)} = r^{(m)} .$$

The old solution is now updated by the correction step $\Delta u^{(m+1)}$

$$u^{(m+1)} = u^{(m)} + \omega \Delta u^{(m+1)} .$$

If we choose $\omega = 1$ and $\bar{A} = \left(\hat{C} + \hat{L} + R^{(m)}\right)$ we receive a fixed-point iteration

$$\left(\hat{C}^{(m)} + \hat{L}^{(m)} + R^{(m)}\right) u^{(m+1)} = S + f^{\alpha}\left(u^{(m)}\right) .$$

**Remark 2.15.**
*The stopping criteria must be defined manually. A tolerance is set and after each correction step the norm of $r^{(m+1)}$ is checked.*

### 2.7.1 Flux limiters for stationary equations

One possible derivation of the flux limiters $\alpha$ for time-independent problems is presented in [Kuz07, p. 2]. Kuzmin claims that the advantage of these limiters is that they converge to a steady state limit if there is one. For each pair of neighboring nodes $i$ and $j$ such that $(\hat{c}+\hat{l})_{ji} \leq (\hat{c}+\hat{l})_{ij} \leq 0$

1. Compute the sums of positive / negative antidiffusive fluxes to be limited

$$P_i^+ = \sum_{\substack{j\neq i\\ a_{ji}\leq a_{ij}}} \max\{0, f_{ij}\}\,, \qquad P_i^- = \sum_{\substack{j\neq i\\ a_{ji}\leq a_{ij}}} \min\{0, f_{ij}\}\,.$$

2. Compute the upper/lower bounds$Q_i^\pm$ to be imposed on the sums $P_i^\pm$

$$\begin{aligned} Q_i^+ &= \sum_{j\neq i} \max\{0, -f_{ij}\}\,, \qquad & Q_j^+ &= \sum_{i\neq j} \max\{0, f_{ij}\}\,,\\ Q_i^- &= \sum_{j\neq i} \min\{0, -f_{ij}\}\,, \qquad & Q_j^- &= \sum_{i\neq j} \min\{0, f_{ij}\}\,. \end{aligned}$$

3. Apply the nodal correction factor $R_i^\pm$ evaluated at the 'upwind' node $i$

$$R_i^\pm = \min\{Q_i^\pm / P_i^\pm\}, \qquad \alpha_{ij} = \begin{cases} R_i^+, & \text{if} \quad f_{ij} > 0\,,\\ 1, & \text{if} \quad f_{ij} = 0\,,\\ R_i^-, & \text{otherwise}\,. \end{cases}$$

## 2.8 Nonlinear FEM-FCT schemes for instationary equations

By applying the $\theta$-scheme to (2.47)

$$M_L \frac{\partial u_i}{\partial t} = -(\hat{C} + \hat{L} + R)u + S + \sum_{j\neq i} m_{ij}(\dot{u}_i - \dot{u}_j) + (-d_{ij} + l_{ij}^+)(u_i - u_j)\,,$$

the fully discrete equation turns into

$$A^{n+1}u^{n+1} = B^n u^n + \Delta t f^\alpha(u^n, u^{n+1}),$$

with

$$A^{n+1} = (M_L + \Delta t\theta(\hat{C}^{n+1} + \hat{L}^{n+1} + R^{n+1})), \quad B^n = (M_L - \Delta t(1-\theta)(\hat{C}^n + \hat{L}^n + R^n))\,,$$

where $f^{\alpha}(u^n, u^{n+1}) = [f^{\alpha}(u^n, u^{n+1})_i]_{i=1\dots n}$ is a vector of the form

$$f_i^{\alpha}(u^n, u^{n+1}) = \sum_{j}^{n} f_{ij}^{\alpha}(u^n, u^{n+1}) \ ,$$

with the following entries

$$\begin{aligned} f_{ij}(u^n, u^{n+1}) =& \frac{1}{\Delta t}[m_{ij}(u_i^{n+1} - u_j^{n+1}) + m_{ij}(u_i^n - u_j^n)] \\ &+\theta(-d_{ij}^{n+1} + l_{ij}^{+^{n+1}})(u_i^{n+1} - u_j^{n+1}) + (1-\theta)(-d_{ij}^n + l_{ij}^{+^n})(u_i^n - u_j^n) \ . \end{aligned}$$

The antidiffusive fluxes $f_{ij}^{\alpha}$ are dependent on the solution $u^{n+1}$. That is why an iterative solver has to be found to solve the nonlinear equation.

The solution $u^{n+1}$ will be approximated by a sequence of intermediate solutions $u^{(m)}$. It is desirable that $\lim\limits_{m\to\infty} u^{(m)} = u^{n+1}$ holds, but there is no proof for this and we will see examples where the algorithm does not converge. The fully discrete equation can be rewritten in matrix form

$$A^{(m)}u^{(m+1)} = B^n u^n + \Delta t f^{\alpha}(u^n, u^{(m)}),$$

where $A$ and $B$ are given by

$$A^{(m)} = (M_L + \Delta t\theta(\hat{C}^{(m)} + \hat{L}^{(m)} + R^{(m)})), \quad B^n = (M_L - \Delta t(1-\theta))(\hat{C}^n + \hat{L}^n + R^n)) \ . \tag{2.48}$$

We will use a three-step solution update method to calculate the auxiliary solution $u^{(m+1)}$. This update ends when the residuum of two successive solutions is small enough.

$$\begin{aligned} &1. && M_L\tilde{u} = B^n u^n && (2.49) \\ &1.1 && \text{Compute correction factors } \alpha \text{ with } u^{(m)}. && \\ &2. && M_L\tilde{u}^{(m+1)} = M_L\tilde{u} + \Delta t f^{\alpha}(u^n, u^{(m)}) && (2.50) \\ &3. && A^{(m)}u^{(m+1)} = M_L\tilde{u}^{(m+1)} && (2.51) \end{aligned}$$

**Remark 2.16.**

- *The intermediate solution $\tilde{u}$ has to be calculated only once per time-step.*
- *It follows from Remark 2.12 that the low order matrices A and B are positivity preserving if*

$$\Delta t \leq \frac{m_i}{(1-\theta)(\hat{c}_{ij}^{(m)} + \hat{l}_{ij}^{(m)})} \ .$$

- *Step 2 is positivity preserving if the prelimiters are designed to be positivity preserving.*

### 2.8.1 Linearization of antidiffusive fluxes

The nonlinear FEM-FCT algorithm as described before might converge very slowly since the antidiffusive fluxes and the correction factors have to be computed after every solution update. The relative changes may be very small and therefore many auxiliary *sweeps* are required.

Kuzmin proposes in [Kuz10, p.139-144] three possibilities to reduce the cost of an implicit FEM-FCT scheme. His main idea is to find a good approximation for each $u^{n+1}$ to use it as a starting iterate, such that step 2 has to be calculated only once per time step.

**1. Use the high order solution to calculate the fluxes**
Since the implicitness results from the fluxes it may be useful to first calculate $u^H$, the solution of the high-order system of

$$(M_C + \Delta t\theta(C^{n+1} + L^{n+1} + R^{n+1}))u^H = (M_C - \Delta t(1-\theta)(C^n + L^n + R^n))\, u^n,$$

which is a good approximation of $u^{n+1}$. The amount of computation steps per time decreases since the second auxiliary step

$$2. \quad M_L\tilde{u}^{(m+1)} = M_L\tilde{u}^{(0)} + \Delta t f^{\alpha}(u^n, u^{(m)}) \tag{2.52}$$

has to be computed only once per time step. Therefore only the third step (2.51) has to be calculated iteratively. In case of a linear system where $\hat{C}^{(m)} = \hat{C}$ and $\hat{L}^{(m)} = \hat{L}$, equation (2.51) yields already the new time step solution $u^{n+1}$ after one execution

$$Au^{(m+1)} = M_L\tilde{u}^{(m+1)}.$$

One drawback of this approach is that the calculation of the higher-order system needs disproportionally more time, due to the lack of an $M$-matrix, than solving the third step (2.51).

**2. Use the low order solution to calculate the fluxes**
In contrast to the last approach it is possible to obtain $u^L$ from the low order scheme

$$\left(M_L + \Delta t\theta(\hat{C}^{n+1} + \hat{L}^{n+1} + R^{n+1})\right)u^L = \left(M_L - \Delta t(1-\theta)(\hat{C}^n + \hat{L}^n + R^n)\right)u^n. \tag{2.53}$$

Kuzmin states that the system can be solved efficiently but the flux-corrected solution $u^{n+1}$ turns out to be too diffusive [Kuz10, p. 170].

**3. Predictor-Corrector algorithms**
The third approach is the first one, which is not equivalent to the original iteration (2.48) anymore. It uses for example the low-order solution $u^L$ as an approximation for $u^{n+1}$. This highly diffusive equation is corrected afterwards by adding the fluxes

$$M_L u^{n+1} = M_L u^L + \Delta t\bar{f}(u^n, u^L)\ .$$

The main idea is to use this auxiliary solution (smooth predictor) for the computation of $\dot{u}$ in $f_{ij}$. We will use the non-discretized form of $f_{ij}$

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

and replace $\dot{u}$ by its iterate.

**Definition 2.7** (Richardson iteration)**.**
*Let $x^{(0)} \in \mathbb{R}^n$ be a given initial iterate. The Richardson iteration for computing a sequence of vectors $x^{(k)} \in \mathbb{R}^n, k = 0, 1, 2, \dots,$ has the form*

$$r^{(k)} = b - Ax^{(k)}, \quad x^{(k+1)} = x^{(k)} + \alpha_k r^{(k)} \ .$$

*with appropriately chosen numbers $\alpha_k \in \mathbb{R}$.*

Applying the Richardson iteration to the preconditioned form of the semidiscrete formulation

$$M_L^{-1}[M_C \dot{u}^L] = M_L^{-1}[C^{n+1}u^L],$$

we obtain the following algorithm

$$\dot{u}^{(m+1)} = \dot{u}^{(m)} + M_L^{-1}[C^{n+1}u^L - M_C\dot{u}^{(m)}], \quad m = 0, 1, 2, \dots$$

where the preconditioner is $M_L^{-1}$

$$\alpha_k := M_L^{-1} \quad A = M_C, \quad b = C^{n+1}u^L \ .$$

**Remark 2.17.** *As well as for the previous iteration, there is no proof available to show that the algorithm converges. Let x be the exact solution of $M_C\dot{x} = M_L x$. By subtracting the exact solution from our iteration equation we receive*

$$e^{(k+1)} = e^{(k)} - M_L^{-1}M_C e^{(k)} \ .$$

*We must find a matrix and a vector norm such that*

$$||e^{(k+1)}|| \leq ||\mathbb{1} - M_L^{-1}M_C|| \cdot ||e^{(k)}||$$

*holds. It would be sufficient to show that*

$$||\mathbb{1} - M_L^{-1}M_C|| < 1 \ .$$

**Remark 2.18.** *"The above linearization strategy offers a number of significant advantages. First, the low-order predictor $u^L$ can be calculated by an arbitrary (explicit or implicit) time-stepping method. In the case of an implicit algorithm, iterative solvers are fast due to the M-matrix property of the low-order operator. Second, the leapfrog time discretization of the antidiffusive flux is second-order accurate with respect to the time level $t^{n+1}$ on which $u^L$ and $f_{ij}$ are defined. Third, instead of three different solutions $u^n, u^{n+1}$ and $\tilde{u}$ only the smooth predictor $u^L$ is involved in the computation of $f_{ij}$ and $\alpha_{ij}$ for the correction step [2]." [Kuz10, p. 172]*

**Remark 2.19.** *In the algorithm which is used for the numerical examples a special case of the Predictor-Corrector algorithm is implemented. It can be found in [JN12].*

### 2.8.2 Prelimiter

Kuzmin states that it may be a good approach to prelimit fluxes before calculating the correction factors $\alpha$ especially in case of finite element approximations. If the intermediate solution $\tilde{u}_i$ is a maximum, $f_{ij}$ may be nonpositive but still preserve positivity in (2.50). This implies that $f_{ij} \leq 0$ with $(\tilde{u}_j - \tilde{u}_i) < 0$ and analoguously if $u_i$ is a minimum $f_{ij}$ may be nonnegative and therefore $f_{ij} \geq 0$ with $(u_j - u_i) > 0$ . Such fluxes would flatten the solution instead of steepening it. Therefore they should be set to zero before performing flux limiting

$$f_{ij} := 0, \quad \text{if} \quad f_{ij}(\tilde{u}_j - \tilde{u}_i) > 0 \, ,$$

(see [KM05, p. 17]).

### 2.8.3 Flux correction for time-dependent equations

In order to prevent undershoots or overshoots the limiting fluxes

$$f_i^\alpha = \sum_{j \neq i} f_{ij}^\alpha, \quad f_{ij}^\alpha = -f_{ji}^\alpha, \quad f_{ij}^\alpha = \alpha f_{ij}$$

have to be LED. This is in particular the case if for a given set of positive $q_{ij}$

$$\sum_{j \neq i} q_{ij} min\{0, u_j - u_i\} \leq \sum_{j \neq i} \alpha_{ij} f_{ij} \leq \sum_{j \neq j} q_{ij} max\{0, u_j - u_i\}$$

holds (see [KM05, p. 14]).

### 2.8.4 Multidimensional Zalesak limiter

The limiters are constructed such that the intermediate solution vector $\tilde{u}^{(m)}$ which comes from the second calculation step (2.50) does not exceed any maximum $\tilde{u}_i^{max}$ or minimum $\tilde{u}_i^{min}$. Positive and negative fluxes are limited separately assuming the worst-case scenario. Perform the following steps for each pair of neighboring nodes $i$ and $j$ such that
$(\hat{c} + \hat{l} + r)_{ji} \leq (\hat{c} + \hat{l} + r)_{ij} \leq 0$

(1.) Sum all positive and negative fluxes into node i

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

(2.) Sum all positive and negative distances to a local extremum of the auxiliary solution

$$Q_i^+ = \frac{m_i}{\Delta t} \max\{0, \max_{j \neq i}(\tilde{u}_j - \tilde{u}_i)\}, \quad Q_i^- = \frac{m_i}{\Delta t} \min\{0, \min_{j \neq i}(\tilde{u}_j - \tilde{u}_i)\} \, .$$

(3.) Nodal correction factors for the net increment to node i

$$R_i^+ = \min\left\{1, \frac{\mathrm{Q}_\mathrm{i}^+}{\mathrm{P}_\mathrm{i}^+}\right\}, \quad \mathrm{R}_\mathrm{i}^- = \min\left\{1, \frac{\mathrm{Q}_\mathrm{i}^-}{\mathrm{P}_\mathrm{i}^-}\right\} .$$

(4.) Limiting of the antidiffusive fluxes $f_{ij}$ and $f_{ji}$ in a symmetric way

$$\alpha_{ij} = \begin{cases} \min\{\mathrm{R}_\mathrm{i}^+, \mathrm{R}_\mathrm{j}^-\}, & \text{if} \quad \mathrm{f}_\mathrm{ij} > 0 \\ \min\{\mathrm{R}_\mathrm{i}^-, \mathrm{R}_\mathrm{j}^+\}, & \text{otherwise} . \end{cases}$$

The correction factors $\alpha_i$ are smaller or equal than 1 and greater or equal than 0. This strategy guarantees the positivity constraint because of the following inequality

$$\tilde{u}_i^{min} = \tilde{u}_i + Q_i^- \leq \bar{u}_i \leq \tilde{u}_i + Q_i^+ = \tilde{u}_i^{max}.$$

**Remark 2.20.**

- *One typical drawback of this construction is that if $u_i$ is an extremum $\alpha_i = 0$ is zero. This includes full artificial diffusion and therefore peaks lose a bit of amplitude in each time step (see [KLT05, p. 164]). This can be observed in Example 3.2, where a step loses height while it is transported through a pipe.*

- *The boundaries $Q_i^+$ and $Q_i^-$ depend on $\Delta t$. Increasing the time step results in smaller boundaries and vice versa. "On the one hand, the LED constraints become less restrictive and, consequently, a larger portion of the raw antidiffusive flux $f_{ij}$ is retained as the time step is refined. This makes FCT the method of choice for transient computations. On the other hand, the use of large $\Delta t$ results in a loss of accuracy, and severe convergence problems may occur in the steady state limit." [KM05, p. 20].*

### 2.8.5 Anderson acceleration

Using a *defect-correction-scheme* of the form

$$u^{(m+1)} = u^{(m)} + \omega A^{-1} r^{(m)} ,$$

can be very expensive. In each step a linear system and the correction factors $\alpha$ have to be solved in order to get the update step. In the previously described algorithms only the last calculated correction step $\Delta u^{(m)}$ is taken into account for the calculation of $u^{(m+1)}$. Due to Kuzmin [KM05, p. 34] the convergence can be improved by accelerating a given number of previously damped updates to the current solution. The damping parameters are gained by solving a least squares problem.

**Algorithm 1:** Anderson acceleration

```
for all $m = 0, \dots$ do
    Compute $\tilde{u}^{(m)} := g(u^{(m)})$
    Store $\tilde{u}^{(m)}$ and $\Delta u^{(m)} := \tilde{u}^{(m)} - u^{(m)}$
    Given $k \le m$ iterates, determine the weights
        $\omega^{(m)} = \left(\omega_1^{(m)}, \dots, \omega_k^{(m)}\right)^T$
    by solving the constrained least-squares problem
        $\min_{\omega^{(m)}} \| \sum_{i=1}^k \omega_i^{(m)} \Delta u^{(m-k+i)} \|_2$ s.t. $\sum_{i=1}^k \omega_i^{(m)} = 1$
    Set $u^{(m+1)} := \sum_{i=1}^k \omega_i^{(m)} \tilde{u}^{(m-k+i)}$
end
return $u^{(m+1)}$
```

**Remark 2.21.** *There is a very simple trick to reduce the restricted problem to a simple minimization problem. Therefore we will rewrite the system into a matrix vector multiplication. Instead of finding the vector $(\omega_1, \dots, \omega_k)^T$ we can simply assume the following system*

$$\| \begin{pmatrix} \Delta u^{(m-k+i)} & \Delta u^{(m-k-1+i))} & \dots & \Delta u^{(m-1+i)} \end{pmatrix} \begin{pmatrix} \gamma_1 \\ \gamma_2 - \gamma_1 \\ \vdots \\ \gamma_{k-i+1} - \gamma_{k-i} \\ \vdots \\ \gamma_k - \gamma_{k-1} \end{pmatrix} \| \rightarrow \quad \min ,$$

*whereby $\gamma_1 = \omega_1$ and $\gamma_{k-i+1} - \gamma_{k-i} = \omega_{k-i+1}$.*

Because of the special sorting of the $\gamma$'s, the restriction

$$1 = \sum_{i=1}^{k} \omega_i^{(m)} = \sum_{i=1}^{k} \gamma_i - \sum_{i=1}^{k-1} \gamma_i = \gamma_k$$

reduces to the simple requirement that $\gamma_k = 1$.

## 2.9 Error estimation

The topic of this chapter is to derive an error estimation for the solution vector $u_h$ of the stationary convection-diffusion-reaction equation

$$\begin{aligned} -\epsilon \Delta u + b \cdot \nabla u + cu &= f && \text{in} \quad \Omega , && (2.54) \\ u &= g_D && \text{in} \ \ \partial\Omega , \\ \nabla \cdot b = 0 , \qquad c \ge \sigma_0 &\ge 0 && \text{in} \quad \Omega , \end{aligned}$$

where $\epsilon \in (0, \epsilon_0)$ with $\epsilon_0 < +\infty$ and $\sigma_0$ are constants. The results are based on the paper [BJK16]. For the existence of the solution it will be required that

- $b \in W^{1,\infty}(\Omega)^d$ ,
- $c \in L^\infty(\Omega)$ ,
- $f \in L^2(\Omega)$ ,
- $g_D \in H^{\frac{1}{2}}(\partial\Omega) \cap C(\partial\Omega)$ ,
- $\Omega$ is bounded,
- $\Gamma_\Omega$ is Lipschitz continuous .

We receive following variational problem:
Find $u \in H^1(\Omega)$ such that $u = u_b$ on $\partial\Omega$ and

$$a(u, v) = \langle f, v\rangle \quad \forall v \in H_0^1(\Omega).$$

The weak solution of (2.54) satisfies

$$a(u, v) = \epsilon(\nabla u, \nabla v) + (b \cdot \nabla u, v) + (cu, v) \quad \text{and} \quad \langle f, v\rangle = (f, v) \,, \tag{2.55}$$

where $(\cdot, \cdot)$ denotes the inner scalar product in $L^2(\Omega)$ or $L^2(\Omega)^d$.

In order to solve the variational formulation (2.55) numerically we need to introduce finite element subspaces, which approximate the spaces $H^1(\Omega)$ and $H_0^1(\Omega)$. Therefore we denote $W_h \subset C(\bar{C}) \cap H^1(\Omega)$ and $V_h := W_h \cap H_0^1(\Omega)$. Thus the bilinear form $a : H^1(\Omega) \times H_0^1(\Omega) \to \mathbb{R}$ reduces to $a_h : W_h \times V_h \to \mathbb{R}$. Therefore the numerical form of the variational formulation becomes:
Find $u_h \in W_h$ such that $u_h(x_i) = g_D(x_i)$, $i \in \{N_\Omega + 1, \ldots, N_\Gamma\}$ and

$$a_h(u_h, v_h) = \langle f, v_h\rangle \quad \forall v_h \in V_h \,. \tag{2.56}$$

We have already seen in previous chapters that (2.56) can also be written in a matrix form

$$(\hat{A}u)_i = S_i + \sum_{j \neq i} \alpha_{ij} f_{ij} \,,$$

with

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

This properties are used to rewrite the problem in the following way

$$\begin{aligned} \sum_{j=1}^{N} a_{ij} u_j + \sum_{j=1}^{N} (1 - \alpha_{ij}) d_{ij}(u_j - u_i) = S_i, \quad i = 1, \ldots, N_\Omega, \\ u_i = g_i, \quad i = N_\Omega + 1, \ldots, N_\Gamma \,. \end{aligned}$$

**Remark 2.22.** *This formulation is also used in the implementation of the algorithm.*

We obtain the new reformulated equation, which will be the basis of our error estimations: Find $u_h \in W_h$ such that $u_h(x_i) = g(x_i), \quad i = N_\Omega + 1, \ldots, N$ and

$$a_h(u_h, v_h) + d_h(u_h; u_h, v_h) = \langle f, v_h \rangle \quad \forall v_h \in V_h \, , \tag{2.57}$$

with

$$d_h(w_h; z_h, v_h) = \sum_{i,j=1}^{N} (1 - \alpha_{ij}(w_h)) d_{ij}(z_j - z_i) v_i \quad \forall w_h, z_h, v_h \in W_h \, .$$

Consider a space $W_h \subset H^1(\Omega)$ of continuous piecewise linear functions, i.e.,

$$W_h = \{v_h \in C(\bar{\Omega}); \quad v_h|_\tau \in P_1(\tau) \quad \forall \tau \in \mathcal{J}_h\} \, ,$$

on a regular family of triangulations $\mathcal{J}_h$ of $\Omega$. For further analysis, we will set

$$h := \max \{diam(h_\tau) : h_\tau \in \mathcal{J}_h\} \ .$$

We will denote the norm $|| \cdot ||_a$ with

$$||v||_a^2 = \epsilon |v|_{1,\Omega}^2 + \sigma_0 ||v||_{0,\Omega}^2 \, ,$$

such that $a_h$ is elliptic on the space $V_h$, i.e., there is a constant $C_a > 0$ such that

$$a_h(v_h, v_h) \geq C_a ||v_h||_a^2 \quad \forall \quad v_h \in V_h,$$

where $|| \cdot ||_a$ is a norm on the space $H_0^1(\Omega)$. We will first estimate the error deriving from the diagonal approximation of the *reaction matrix*.

**Lemma 2.1.**
*There is a constant C independent of h such that*

$$\left| (cu_h, v_h) - \sum_{i=1}^{N_\Omega} (c, \phi_i) u_i v_i \right| \leq Ch ||c||_{0,\infty,\Omega} |u_h|_{1,\Omega} ||v_h||_{0,\Omega},$$

*for all $c \in L^\infty(\Omega)$, $u_h \in W_h$, and $v_h \in V_h$*

*Proof.* The proof will be skipped here. It is referred to *Analysis of algebraic flux correction schemes* (p.14). □

The Lagrange interpolator $i_h : C(\Omega) \to W_h$ is defined by

$$i_h v = \sum_{i=1}^{N} v(x_i) \phi_i, \quad v \in C(\bar{\Omega}) \, . \tag{2.58}$$

A norm on $V_h$ is defined according to the left-hand side of the reformulated problem (2.57)

$$\|v_h\|_h := \left(C_a\|v_h\|_a^2 + d_h(u_h; v_h, v_h)\right)^{1/2}, \quad v_h \in V_h \,. \tag{2.59}$$

We will begin with an intermediate result from [BJK16, p. 13] of the error estimation

$$\|u - u_h\|_h \leq C_a^{1/2}\|u - i_h u\|_a + \sup_{v_h \in V_h} \frac{a(u, v_h) - a_h(i_h u, v_h)}{\|v_h\|_h} + (d_h(u_h; i_h u, i_h u))^{1/2} \,. \tag{2.60}$$

Each part on the right-hand side is analyzed and estimated separately. We will begin with the first term on the right-hand side.

Assume that $u \in H^2(\Omega)$ is the solution of (2.54). Then standard interpolation estimates give

$$\|u - i_h u\|_h \leq C(\epsilon + \sigma_0 h^2)^{1/2} h |u|_{2,\Omega} \,.$$

**Lemma 2.2.**
*The second term of the right-hand side of* (2.60) *can be estimated in the following form*

$$\sup_{v_h \in V_h} \frac{a(u, v_h) - a_h(i_h u, v_h)}{\|v_h\|_h} \leq C\left(\epsilon + \sigma_0^{-1}\{\|b\|_{0,\infty,\Omega}^2 + \|c\|_{0,\infty,\Omega}^2\}\right)^{1/2} h\|u\|_{2,\Omega} \,. \tag{2.61}$$

*If* $c = 0$, *then*

$$\sup_{v_h \in V_h} \frac{a(u, v_h) - a_h(i_h u, v_h)}{\|v_h\|_h} \leq C\left(\epsilon + \epsilon^{-1}\|b\|_{0,\infty,\Omega}^2 + h^2\right)^{1/2} h|u|_{2,\Omega} \,. \tag{2.62}$$

*Proof.* See p.16. □

**Remark 2.23.**
*If* $c = 0$ *it follows that* $\sigma_0 = 0$. *According to* (2.62), *if* $\sqrt{\epsilon} < h$ *it implies a bad error estimation. Therefore it is important that*

$$h \lesssim \sqrt{\epsilon}$$

*holds.*
*If* $c > 0$ *and therefore* $\sigma_0 > 0$, *one obtains from* (2.61) *that*

$$\|u - u_h\|_h \leq Ch\|u\|_{2,\Omega} + (d_h(u_h; i_h u, i_h u))^{1/2} \,. \tag{2.63}$$

Finally the third term of estimation (2.60) can be estimated with the following lemma.

**Lemma 2.3.**
*Let the matrix D be defined by* $d_{ij} = d_{ji} = -\max\{a_{ij}, 0, a_{ji}\}$, $\forall i \neq j$ *with* $d_{ii} = -\sum_{j \neq i} d_{ij}$. *Then there is a constant C independent of h and the data of problem* (2.54) *such that*

$$d_h(w_h; i_h u, i_h u) \leq C\left(\epsilon + \|b\|_{0,\infty,\Omega} h\right) |i_h u|_{1,\Omega}^2 \quad \forall w_h \in W_h, u \in C(\bar{\Omega}) \,.$$

*Proof.* See p.16. □

If we insert this result into (2.63), the convergence order is reduced.

**Corollary 2.1.**
Let $u \in H^2(\Omega)$ be the solution of (2.60), and $u_h$ be a solution of the discrete problem (2.57) . Then if $\sigma_0 > 0$, there exists a constant $C > 0$, independent of h and the data of (2.60) such that

$$\begin{aligned} \|u - u_h\|_h \le\, & C\left(\epsilon + \sigma_0^{-1}\{\|b\|^2_{0,\infty,\Omega} + \|c\|^2_{0,\infty,\Omega}\} + \sigma_0 h^2\right)^{1/2} h\|u\|_{2,\Omega} \\ & + C\left(\epsilon + \|b\|_{0,\infty,\Omega} h\right)^{1/2} |i_h u|_{1,\Omega} \ . \end{aligned}$$

**Remark 2.24.**
*For the convection-dominated case ($\epsilon < \|b\|_{0,\infty,\Omega} h$) the result of Lemma 2.3 reduces to*

$$d_h(w_h; i_h u, i_h u) \le C(\|b\|_{0,\infty,\Omega} h)|i_h u|^2_{1,\Omega} \quad \forall w_h \in W_h, u \in C(\bar{\Omega}) \ .$$

*Therefore the error estimate* (2.63) *is of order $O(\sqrt{h})$.*

# 3 Numerical examples

## 3.1 Smooth Solution

This example has a polynomial solution. Consider $\Omega = (0, 1)^2$ with $\epsilon = 10^{-6}$, and $b = (10, 2)^T$ , $c = 2$, $g_D = 0$. The right-hand side is chosen such that

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

is the solution of (2.54). The Galerkin discretization is performed with $P^1$-elements. The stopping criterion for the fixed-point iteration (see Remark 2.15) is $10^{-9}$. We store 5 residuals to use Anderson acceleration.

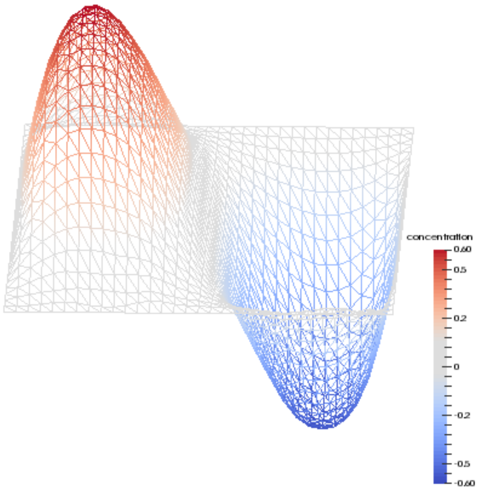


Figure 3.1: Smooth solution on grid 1 mirror, level 4

We will calculate the error $||u - u_h||$ in the $L^2$, $H^1$, $d_h^{1/2}$ -norm and the error in the previously introduced $h$ norm. We want to figure out if the results from the analysis are reflected in the numerical results. According to Remark 2.24 we expect the order of convergence to be 1/2 for $d_h^{1/2}$ and for the $h$-norm. The order of convergence $p$ is calculated by using two succeeding errors

$$\frac{||u_h - u||}{||u_{h'} - u||} = \left(\frac{h}{h'}\right)^p .$$

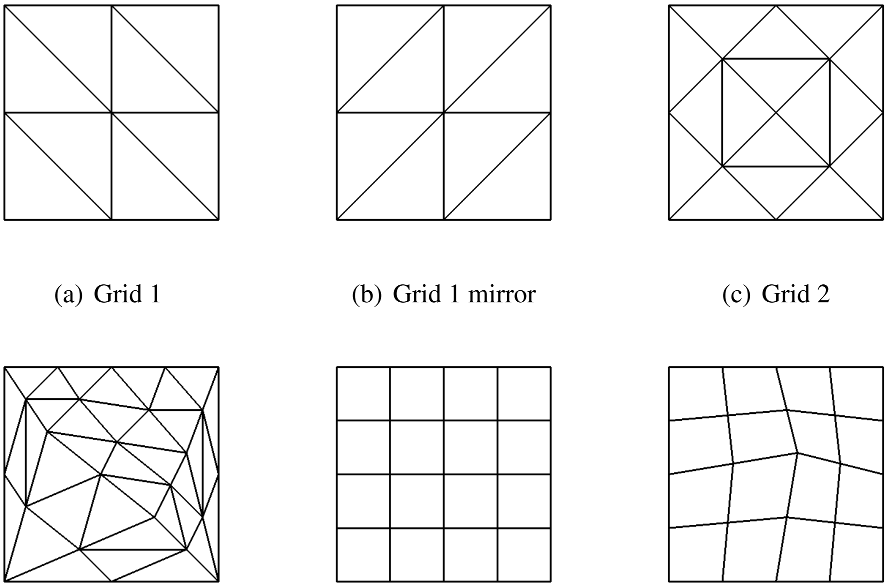
| level | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| $1-\alpha(u_h)$ | 0.429 | 0.324 | 0.150 | 0.877 | 0.051 | 0.0287 | 0.0161 | 0.00881 |

Table 3.1: Arithmetic mean of $\left(1-\alpha_{ij}(u_h)\right)$ on grid 1 mirror

If we take a look at the results in Table 3.2-3.7, it is obvious that the order of convergence for the $h$-norm and $d_h^{1/2}$ is better than $1/2$, i.e. we have an approximate order of 1. This indicates that the estimates in the proof of Lemma 2.3 are too pessimistic. In particular it turns out that the expression $(1-\alpha_{ij})$ from (2.58) is estimated by 1, wherease the correction factors $\alpha_{ij}$ may be close to 1 and therefore lead to a better convergence.

The worst order of convergence is expected on grid 3, since it is an unstructured grid. Nevertheless, it turns out that for the norms $d_h^{1/2}$ and $h$ it does not have significant differences to the other grids. However, for the $L^2$ and $H^1$ semi-norms we obtain worse results. The order of convergence for the $H^1$-norm seems to tend to zero.

Grid 5 evolves from grid 4 by shifting the middle point to (0.6,0.6). As expected, the order of convergence is worse than on grid 4.



(a) Grid 1

(b) Grid 1 mirror

(c) Grid 2

(d) Grid 3

(e) Grid 4

(f) Grid 5

Figure 3.2: Grids for testing in level 0

| level | $L^2$-error | order | $H^1$ -semi | order | $d_h^{1/2}$ | order | $h$-norm | order |
|---|---|---|---|---|---|---|---|---|
| 0 | 2.809e-01 | | 2.046 | | 0 | | 0.3973 | |
| 1 | 2.018e-01 | 0.477 | 1.742 | 0.232 | 1.200 | | 1.234 | -1.64 |
| 2 | 7.421e-02 | 1.44 | 9.823e-01 | 0.827 | 7.184e-01 | 0.741 | 7.260e-01 | 0.765 |
| 3 | 2.119e-02 | 1.81 | 5.329e-01 | 0.882 | 3.169e-01 | 1.18 | 3.183e-01 | 1.19 |
| 4 | 6.383e-03 | 1.73 | 2.823e-01 | 0.917 | 1.252e-01 | 1.34 | 1.255e-01 | 1.34 |
| 5 | 1.835e-03 | 1.80 | 1.437e-01 | 0.975 | 5.067e-02 | 1.30 | 5.073e-02 | 1.31 |
| 6 | 4.748e-04 | 1.95 | 6.600e-02 | 1.12 | 2.229e-02 | 1.18 | 2.230e-02 | 1.19 |
| 7 | 1.192e-04 | 1.99 | 3.084e-02 | 1.10 | 1.022e-02 | 1.13 | 1.022e-02 | 1.13 |
| 8 | 2.922e-05 | 2.03 | 1.459e-02 | 1.08 | 4.893e-03 | 1.06 | 4.893e-03 | 1.06 |
| 9 | 7.189e-06 | 2.02 | 6.985e-03 | 1.06 | 2.390e-03 | 1.03 | 2.390e-03 | 1.03 |

Table 3.2: Grid 1

| level | $L^2$-error | order | $H^1$ -semi | order | $d_h$ | order | $h$-norm | order |
|---|---|---|---|---|---|---|---|---|
| 0 | 2.754e-01 | | 2.022 | | 0 | | 0.3894 | |
| 1 | 1.990e-01 | 0.469 | 1.790 | 0.176 | 1.027 | | 1.065 | -1.45 |
| 2 | 6.259e-02 | 1.67 | 9.576e-01 | 0.902 | 6.857e-01 | 0.582 | 6.914e-01 | 0.623 |
| 3 | 1.568e-02 | 2.00 | 4.846e-01 | 0.983 | 2.963e-01 | 1.21 | 2.971e-01 | 1.22 |
| 4 | 4.822e-03 | 1.70 | 2.395e-01 | 1.02 | 1.153e-01 | 1.36 | 1.155e-01 | 1.36 |
| 5 | 1.272e-03 | 1.92 | 1.164e-01 | 1.04 | 4.906e-02 | 1.23 | 4.910e-02 | 1.23 |
| 6 | 3.179e-04 | 2.00 | 5.876e-02 | 0.987 | 2.201e-02 | 1.16 | 2.202e-02 | 1.16 |
| 7 | 7.850e-05 | 2.02 | 2.976e-02 | 0.982 | 1.028e-02 | 1.10 | 1.028e-02 | 1.10 |
| 8 | 1.939e-05 | 2.02 | 1.469e-02 | 1.02 | 4.911e-03 | 1.07 | 4.911e-03 | 1.07 |
| 9 | 4.768e-06 | 2.02 | 6.856e-03 | 1.10 | 2.383e-03 | 1.04 | 2.383e-03 | 1.04 |

Table 3.3: Grid 1 mirror

| level | $L^2$-error | order | $H^1$ -semi | order | $d_h^{1/2}$ | order | $h$-norm | order |
|---|---|---|---|---|---|---|---|---|
| 0 | 2.254e-01 | | 1.866 | | 0.9514 | | 1.003 | |
| 1 | 8.128e-02 | 1.47 | 9.809e-01 | 0.928 | 7.796e-01 | 0.287 | 7.881e-01 | 0.349 |
| 2 | 2.035e-02 | 2.00 | 5.198e-01 | 0.916 | 3.580e-01 | 1.12 | 3.591e-01 | 1.13 |
| 3 | 6.164e-03 | 1.72 | 2.754e-01 | 0.917 | 1.717e-01 | 1.06 | 1.719e-01 | 1.06 |
| 4 | 1.769e-03 | 1.80 | 1.507e-01 | 0.870 | 7.998e-02 | 1.10 | 8.002e-02 | 1.10 |
| 5 | 4.852e-04 | 1.87 | 8.398e-02 | 0.843 | 3.832e-02 | 1.06 | 3.833e-02 | 1.06 |
| 6 | 1.382e-04 | 1.81 | 5.435e-02 | 0.628 | 1.859e-02 | 1.04 | 1.860e-02 | 1.04 |
| 7 | 4.388e-05 | 1.65 | 4.136e-02 | 0.394 | 9.125e-03 | 1.03 | 9.126e-03 | 1.03 |
| 8 | 1.637e-05 | 1.4225 | 3.595e-02 | 0.20 | 4.514e-03 | 1.02 | 4.514e-03 | 1.02 |

Table 3.4: Grid 2

| level | $L^2$-error | order | $H^1$ -semi | order | $d_h^{1/2}$ | order | $h$-norm | order |
|---|---|---|---|---|---|---|---|---|
| 0 | 1.090e-01 | | 1.290 | | 6.612e-01 | | 6.789e-01 | |
| 1 | 4.482e-02 | 1.28 | 8.735e-01 | 0.563 | 5.332e-01 | 0.310 | 5.370e-01 | 0.338 |
| 2 | 1.616e-02 | 1.47 | 4.644e-01 | 0.912 | 2.775e-01 | 0.942 | 2.785e-01 | 0.947 |
| 3 | 5.643e-03 | 1.52 | 2.879e-01 | 0.690 | 1.372e-01 | 1.02 | 1.374e-01 | 1.02 |
| 4 | 1.735e-03 | 1.70 | 1.670e-01 | 0.786 | 6.870e-02 | 0.998 | 6.875e-02 | 0.999 |
| 5 | 5.156e-04 | 1.75 | 1.028e-01 | 0.700 | 3.387e-02 | 1.02 | 3.388e-02 | 1.02 |
| 6 | 1.544e-04 | 1.74 | 7.194e-02 | 0.515 | 1.655e-02 | 1.03 | 1.655e-02 | 1.03 |
| 7 | 4.901e-05 | 1.66 | 5.699e-02 | 0.336 | 8.095e-03 | 1.03 | 8.096e-03 | 1.03 |

Table 3.5: Grid 3

| level | $L^2$-error | order | $H^1$ -semi | order | $d_h^{1/2}$ | order | $h$-norm | order |
|---|---|---|---|---|---|---|---|---|
| 0 | 1.652e-01 | | 1.533 | | 1.003 | | 1.030 | |
| 1 | 5.297e-02 | 1.64 | 7.750e-01 | 0.985 | 6.124e-01 | 0.712 | 6.170e-01 | 0.740 |
| 2 | 1.555e-02 | 1.77 | 3.909e-01 | 0.987 | 2.817e-01 | 1.12 | 2.826e-01 | 1.13 |
| 3 | 4.972e-03 | 1.64 | 2.011e-01 | 0.959 | 1.177e-01 | 1.26 | 1.179e-01 | 1.26 |
| 4 | 1.398e-03 | 1.83 | 9.855e-02 | 1.03 | 4.988e-02 | 1.24 | 4.991e-02 | 1.24 |
| 5 | 3.582e-04 | 1.96 | 5.009e-02 | 0.976 | 2.209e-02 | 1.17 | 2.210e-02 | 1.18 |
| 6 | 8.861e-05 | 2.02 | 2.507e-02 | 0.999 | 1.015e-02 | 1.12 | 1.015e-02 | 1.12 |
| 7 | 2.158e-05 | 2.04 | 1.238e-02 | 1.02 | 4.808e-03 | 1.08 | 4.808e-03 | 1.08 |

Table 3.6: Grid 4

| level | $L^2$-error | order | $H^1$ -semi | order | $d_h^{1/2}$ | order | $h$-norm | order |
|---|---|---|---|---|---|---|---|---|
| 0 | 1.645e-01 | | 1.522 | | 9.368e-01 | | 9.652e-01 | |
| 1 | 5.239e-02 | 1.65 | 7.758e-01 | 0.972 | 6.018e-01 | 0.638 | 6.063e-01 | 0.671 |
| 2 | 1.672e-02 | 1.65 | 4.163e-01 | 0.898 | 2.777e-01 | 1.12 | 2.787e-01 | 1.12 |
| 3 | 5.596e-03 | 1.58 | 2.178e-01 | 0.935 | 1.190e-01 | 1.22 | 1.192e-01 | 1.23 |
| 4 | 1.591e-03 | 1.81 | 1.036e-01 | 1.07 | 5.178e-02 | 1.20 | 5.183e-02 | 1.20 |
| 5 | 4.161e-04 | 1.93 | 5.152e-02 | 1.01 | 2.326e-02 | 1.15 | 2.326e-02 | 1.16 |
| 6 | 1.054e-04 | 1.98 | 2.572e-02 | 1.00 | 1.077e-02 | 1.11 | 1.078e-02 | 1.11 |
| 7 | 2.633e-05 | 2.00 | 1.289e-02 | 0.997 | 5.127e-03 | 1.07 | 5.127e-03 | 1.07 |
| 8 | 6.574e-06 | 2.00 | 6.358e-03 | 1.02 | 2.474e-03 | 1.05 | 2.474e-03 | 1.05 |

Table 3.7: Grid 5

## 3.2 Transport of a step

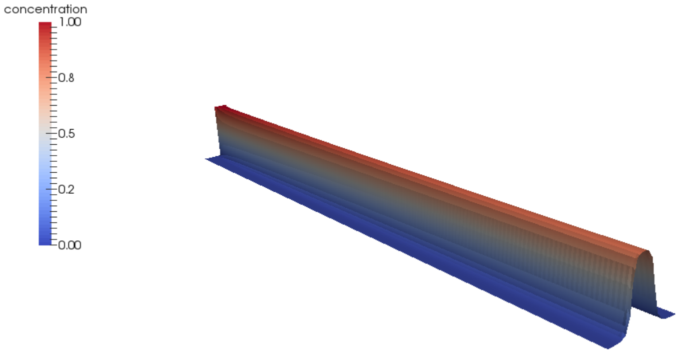


Figure 3.3: AFC: Step solution on a quadrilateral mesh for $\epsilon = 10^{-10}$

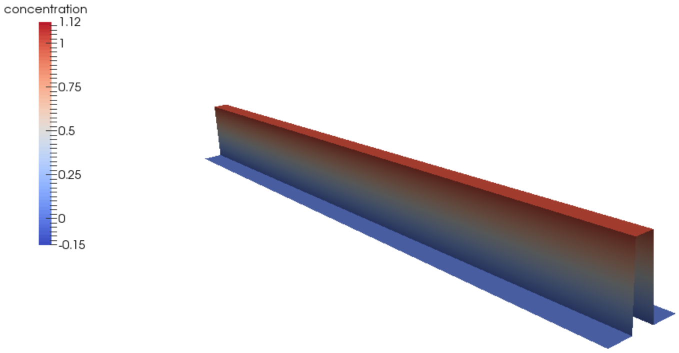


Figure 3.4: SUPG: Step solution on a quadrilateral mesh for $\epsilon = 10^{-10}$

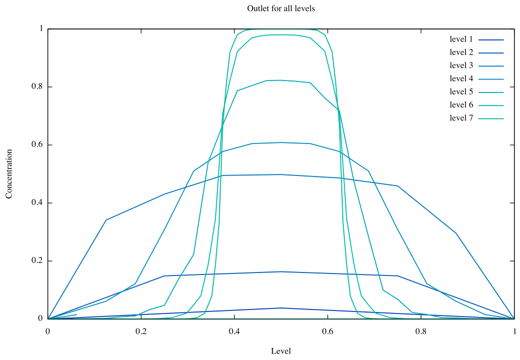
Consider the following problem:

$$\epsilon\Delta u + \begin{pmatrix}1\\0\end{pmatrix} \cdot \nabla u = 0 \quad (x, y) \in (0, 10) \times (0, 1)$$

$$u_b = \begin{cases} 1 & \text{x} = 0, \quad \text{y} \in [0.375, 0.625]\,, \\ 0 & \text{x} = 0, \quad \text{y} \notin [0.375, 0.625]\,, \\ 0, & \text{y} = 0, \quad \text{or} \quad \text{y} = 1\,, \end{cases}$$

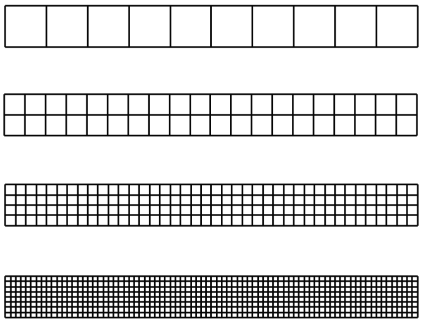
$$\epsilon\nabla u \cdot \eta = 0 \quad \text{on} \quad x = 10, \quad y \in (0, 1)\,.$$

This example shows the impact of artificial diffusion very clearly. While the boundary condition is a step with values 1 or 0, the transportation through a pipe of length 10 yields a smeared outlet. The expected behavior is that for very small $\epsilon$ the solution of the outlet should be very close to the inlet. For better comparison, we will also solve this equation with SUPG, which does not include artificial diffusion and therefore leads to the expected results.

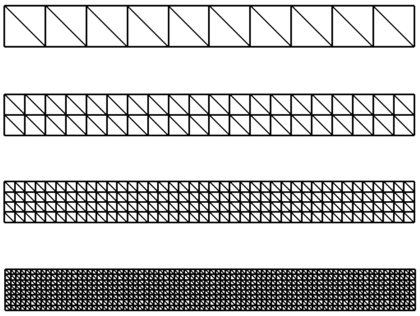


(a) Smeared outlets for all levels in point $x = 10$

Figure 3.5: $\epsilon = 10^{-6}$



(a) Quadrilateral grids for levels 0,1,2 and 3



(b) Triangular grids for levels 0,1,2 and 3

We set $\epsilon = 10^{-6}, 10^{-8}, 10^{-10}$ and we store 5 solutions for the Anderson acceleration. The solutions in level 0 are calculated on a quadrilateral grid consisting of 10 squares and on a triangular grid consisting of 20 triangles. The grids are refined uniformly (see figure 3.2).
We will take a look at the following error measuring factors

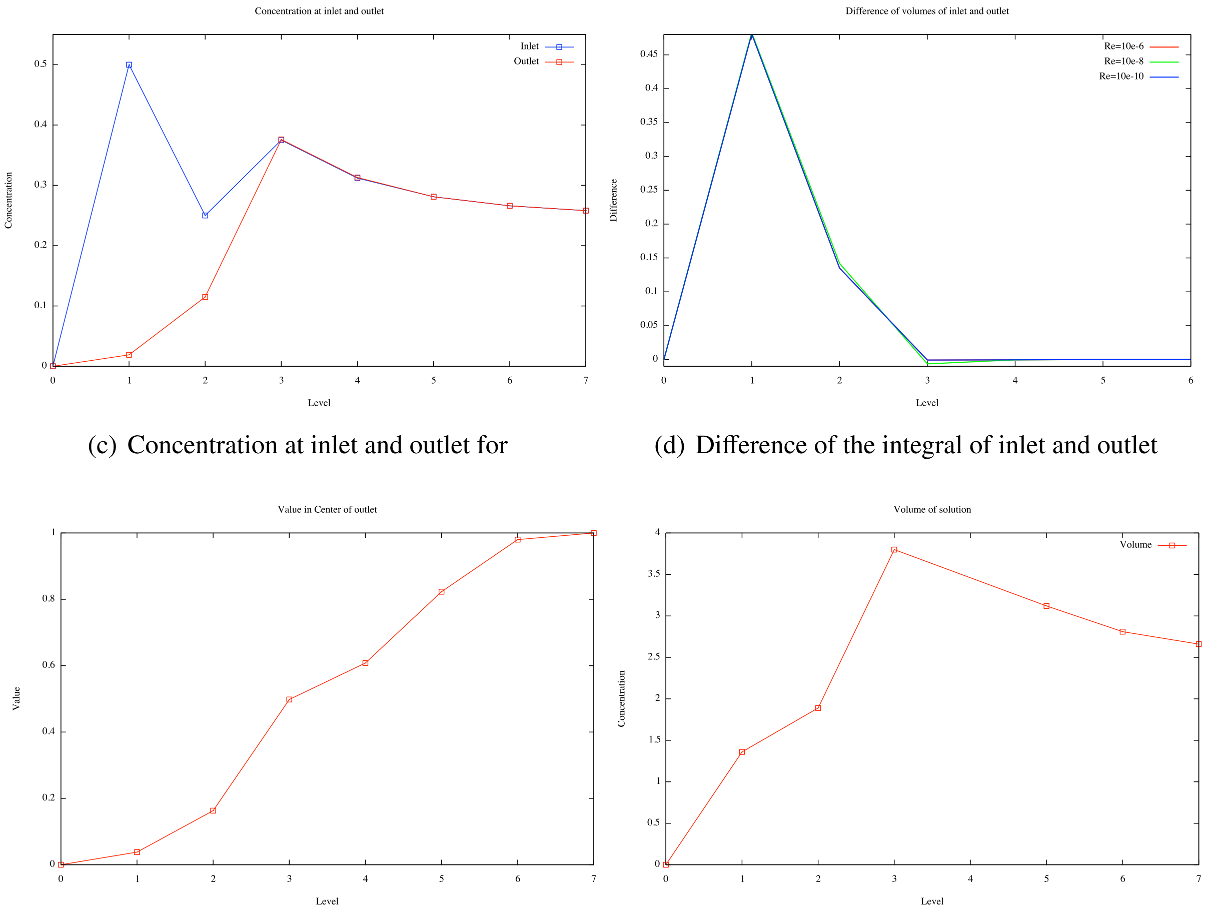
- The integral of the inlet and outlet. The exact solution at the inlet is 0.25.
- The value at the center of the outlet. It should be close to 1. The smaller it is, the more diffusive is the solution method.
- The volume of the solution. If there is no diffusion, the inlet must be transported through the pipe without loss of height. Therefore the solution should be close to 2.5

| methode/$\epsilon$ | $10^{-6}$ | $10^{-8}$ | $10^{-10}$ |
|---|---|---|---|
| AFC: quadrilateral | 0.98 | 0.98 | 0.98 |
| AFC: triangular | 0.959 | 0.959 | 0.959 |
| SUPG: quadrilateral | not converged | | 1.0 |

Table 3.8: Value at center of outlet for different $\epsilon$ in level 6.

- The difference of the volumes of the inlet and the outlet.

The values of the first three measuring factors look exactly the same for all $\epsilon$, that is why we will show only $\epsilon = 10^{-6}$. Nevertheless there are variances visible for the difference of the volume of the inlet and outlet for different $\epsilon$. It is clear that the solutions for level 0 and level 1 are not representative. But taking a deeper look at higher levels, we see that the solutions are very close to the expected values.
We can resume that $\epsilon$ does not have a strong impact on the solution. Table 3.2 indicates that very small $\epsilon$ are intercepted by artificial diffusion.



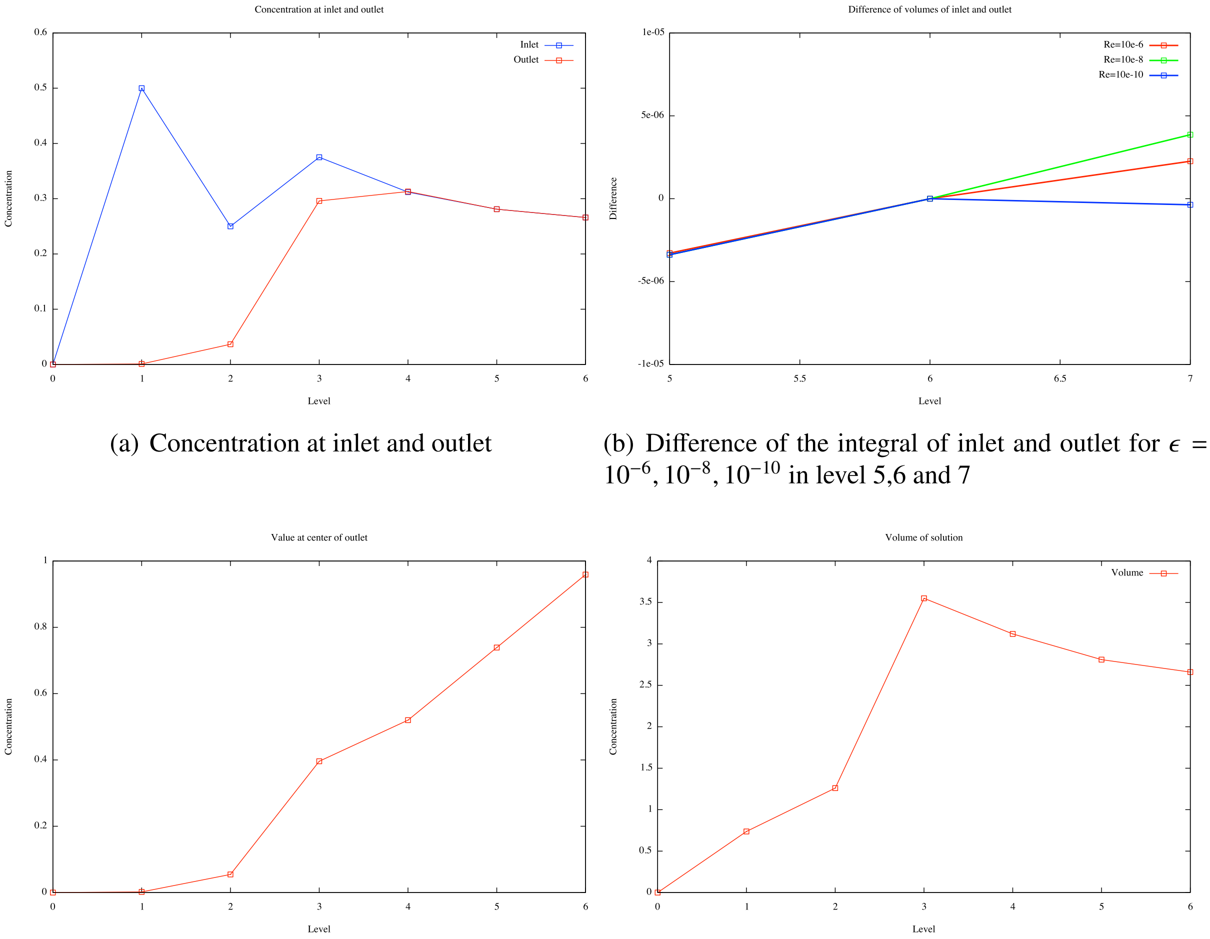
(c) Concentration at inlet and outlet for

(d) Difference of the integral of inlet and outlet

(e) Value at center of outlet

(f) Volume of solution for $\epsilon = 10^{-6}$

Figure 3.6: AFC: Error measuring for $\epsilon = 10^{-6}$ on a quadrilateral grid.



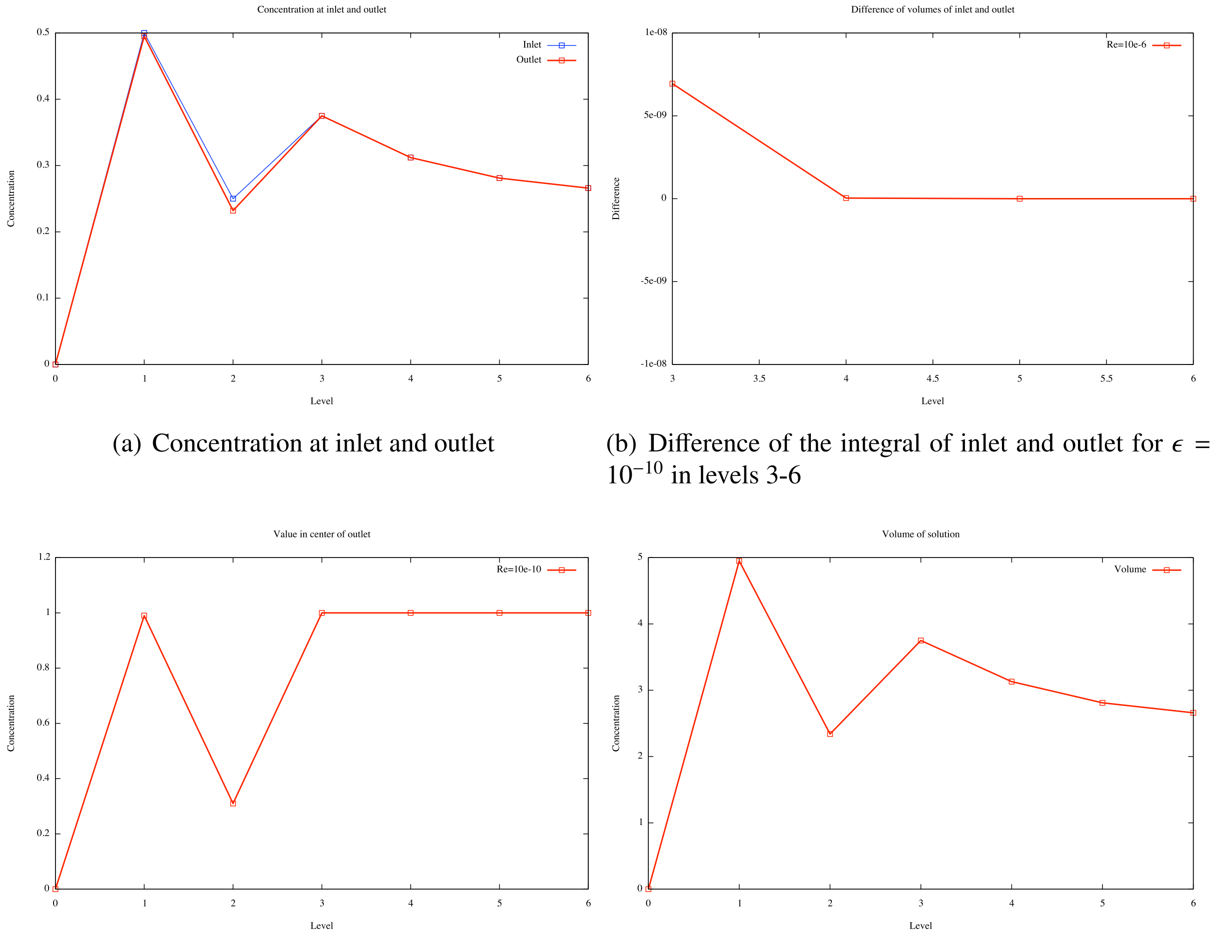
(a) Concentration at inlet and outlet

(b) Difference of the integral of inlet and outlet for $\epsilon = 10^{-6}, 10^{-8}, 10^{-10}$ in level 5,6 and 7

(c) Value at center of outlet

(d) Volume of solution for $\epsilon = 10^{-6}$

Figure 3.7: AFC: Error measuring for $\epsilon = 10^{-6}$ on a triangular grid.



(a) Concentration at inlet and outlet

(b) Difference of the integral of inlet and outlet for $\epsilon = 10^{-10}$ in levels 3-6

(c) Value at center of outlet

(d) Volume of solution for $\epsilon = 10^{-10}$

Figure 3.8: SUPG: Error measuring for $\epsilon = 10^{-10}$ on a quadrilateral grid.

## 3.3 Traveling wave

The following instationary example from [GIJW15] will show that algebraic flux correction may still generate oscillations. It is given in the domain $\Omega = (0,1)^2$ and $(0,T) = (0,1)$

$$\partial_t u - \epsilon \Delta u + \begin{pmatrix} cos(\pi/3) \\ sin(\pi/3) \end{pmatrix} \cdot \nabla u + u = f \qquad \text{in} \quad \Omega \times [0,T]\,, \tag{3.1}$$

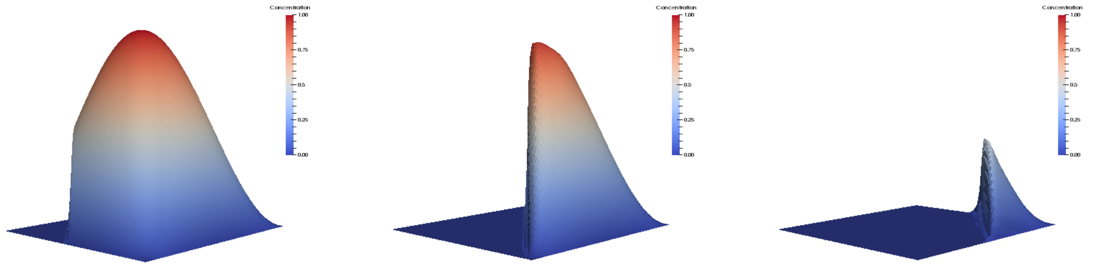
$$u(x,0) = u_0(x) \qquad \forall x \in \Omega\,, \tag{3.2}$$

$$u(x,t) = 0 \qquad \forall x \in \partial\Omega\,. \tag{3.3}$$

The solution is defined by

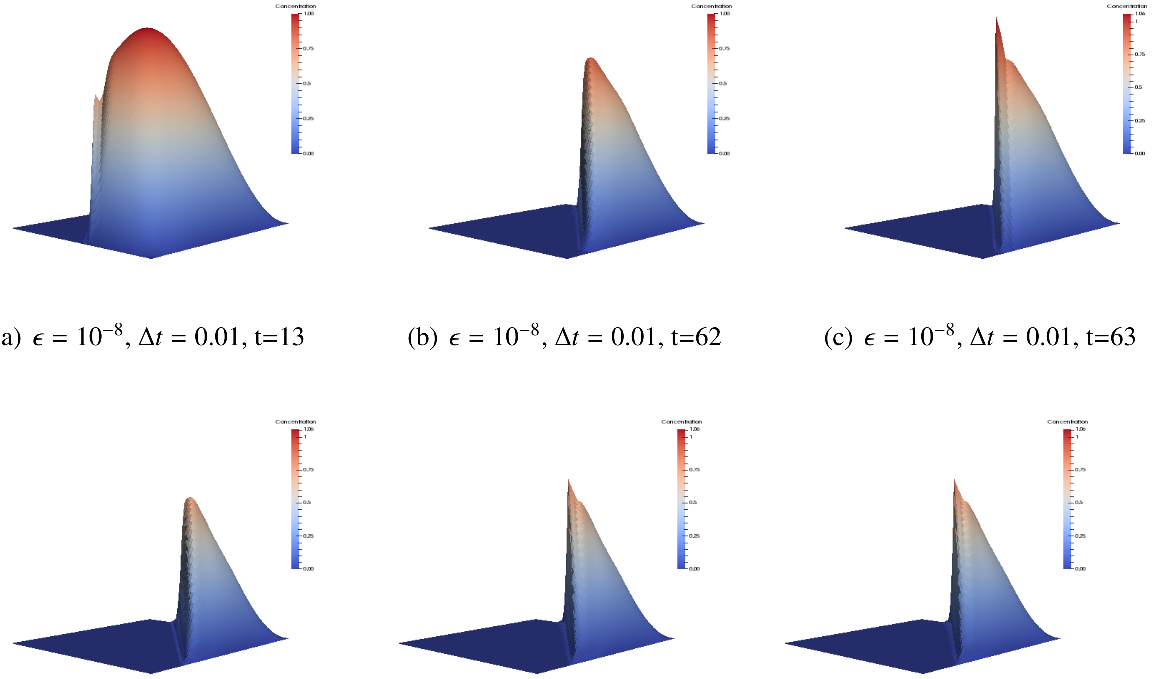
$$u(t,x,y) = 0,5 sin(\pi x) sin(\pi y) \left[ tanh\left( \frac{x+y-t-0.5}{\sqrt{\epsilon}} \right) + 1 \right] .$$

It has a moving layer. The right-hand side $f$ and the initial condition $u_0$ are chosen such that the solution satisfies the boundary value problem (3.1). The example will be analyzed on a simple triangular grid which is refined to level 7. Because the results on a simple quadrilateral grid are similar, they will be skipped here. The solution is calculated for $\epsilon = 10^{-6}, 10^{-8}$ and $10^{-12}$. It turns out that the solutions tend to generate oscillations the smaller $\epsilon$ becomes. For $\epsilon = 10^{-8}$ the first peak occurs in time step 13. After time step 50, that means when the wave begins to get smaller, the peaks appear more frequently. The worst results are obtained for $\epsilon = 10^{-12}$. At some point $u_h(x, t)$ reaches a concentration of 65.
It is not clear why these peaks develop. The first assumption was that $\Delta t$ needs to be smaller. Analyzing $\Delta t = 0.001$ and $\Delta t = 0.0001$ reveals that the highest peak decreases to approximately 15 and 26, but these results are still not satisfactory.



(a) $\epsilon = 10^{-6}$, $\Delta t = 0.01$, t=0 (b) $\epsilon = 10^{-6}$, $\Delta t = 0.01$, t=50 (c) $\epsilon = 10^{-6}$, $\Delta t = 0.01$, t=100

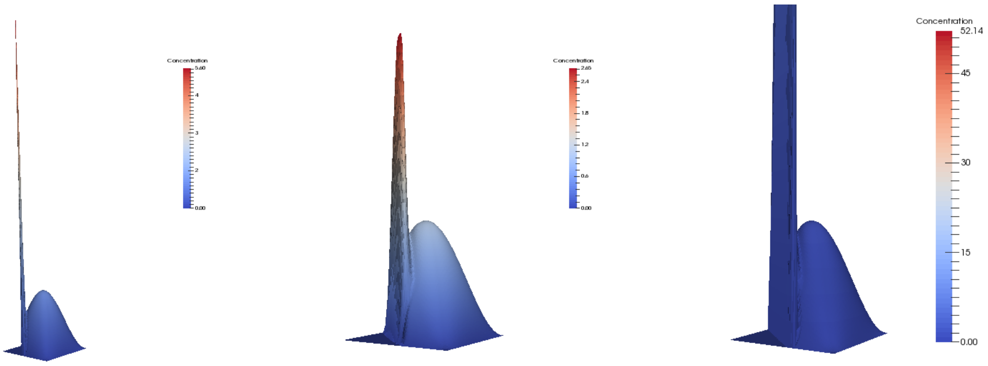
Figure 3.9: Some results for $\epsilon = 10^{-6}$, $\Delta t = 0.01$



(a) $\epsilon = 10^{-8}$, $\Delta t = 0.01$, t=13 (b) $\epsilon = 10^{-8}$, $\Delta t = 0.01$, t=62 (c) $\epsilon = 10^{-8}$, $\Delta t = 0.01$, t=63

(d) $\epsilon = 10^{-8}$, $\Delta t = 0.01$, t=74 (e) $\epsilon = 10^{-8}$, $\Delta t = 0.01$, t=76 (f) $\epsilon = 10^{-8}$, $\Delta t = 0.01$, t=76

Figure 3.10: Some results for $\epsilon = 10^{-8}$, $\Delta t = 0.01$



(a) $\epsilon = 10^{-12}$, $\Delta t = 0.01$, t=1 (b) $\epsilon = 10^{-12}$, $\Delta t = 0.01$, t=12 (c) $\epsilon = 10^{-12}$, $\Delta t = 0.01$, t=13

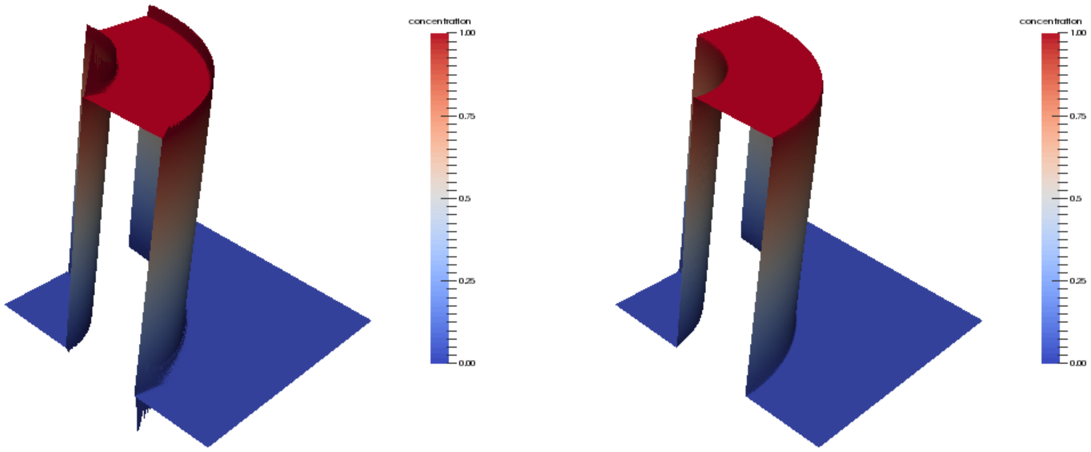
Figure 3.11: Some results for $\epsilon = 10^{-12}$, $\Delta t = 0.01$

## 3.4 Two interior layers

This stationary example is a standard example from [Kuz10, p. 201]. It generates strong under- and overshoots when using SUPG. We will calculate the solution on the previously introduced grids in Figure 3.1 and take a look at undershoots and overshoots.
Let the domain be $\Omega = (0,1)^2$. The solution has to fulfill the following differential equation:

$$\begin{aligned} 10^{-6}\Delta u + \begin{pmatrix} -y \\ x \end{pmatrix} \nabla u =& 0, \quad x \in \Omega\,, \\ g_D(x,y) =& \begin{cases} 1 & \text{if} \quad \text{(x, y)} \in (1/3, 2/3) \times \{0\}\,, \\ 0 & \text{on} \quad \partial\Omega_{\mathrm{D}}\,, \end{cases} \\ \epsilon \nabla u(x,y) \cdot \eta =& 0 \quad \forall (x,y) \in \{0\} \times (0,1)\,. \end{aligned} \tag{3.4}$$

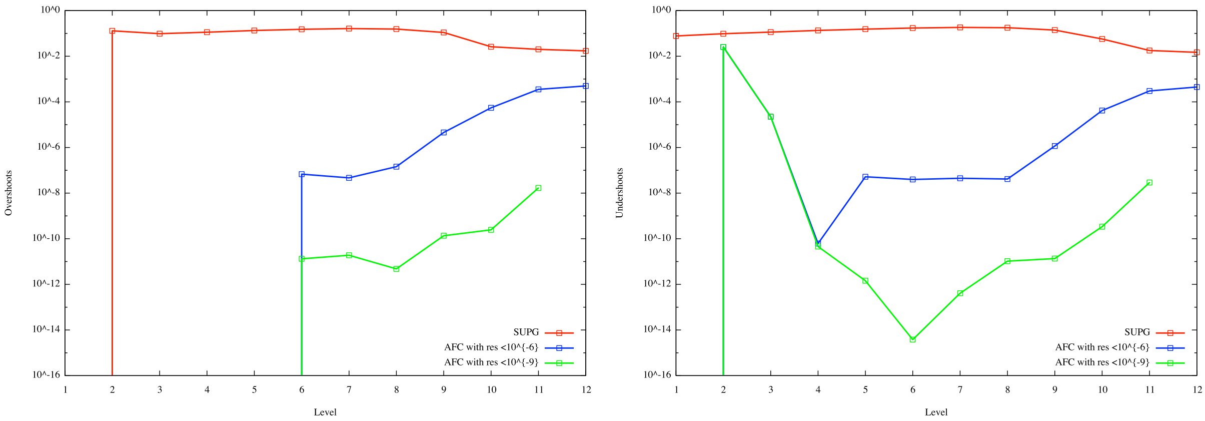
Figure 3.4 shows the solution of (3.4) on the eighth refinement of Grid 4 from Figure 3.1. The left Subfigure (a) depicts the SUPG solution. At the top of both layers a border of overshoots is visible as well as a peak of undershoots at the bottom. It seems that Subfigure (b) which shows the solution of the linearized AFC algorithm does not contain any overshoots and undershoots or at least has very small peaks. Subfigures (a)-(j) of Figure 3.13 display the summed up over- and undershoots for grids 1-5 on a logarithmic scale. The results are plotted for two stopping criteria from Remark 2.15: $r^{(m+1)} < 10^{-9}$ and $r^{(m+1)} < 10^{-6}$. The levels 1-12 are plotted on the X axis. Some cases did not converge and therefore these plots do not contain values for those levels.
The first obvious result to point out is that SUPG is always worse compared to AFC. The values of SUPG vary between 0.1 and 0.01, although there is a slight improvement for higher grid levels. On Grid 3 and Grid 5 we could not obtain values for SUPG on level 12 , because the stopping criterion has not been reached.
It is striking that the stopping criterion in AFC was mostly reached only for grid levels 1-8. Calculations for higher grid levels needed at least 3 days or did not converge. Subfigures (a) and (b) indicate that AFC tends to generate higher overshoots and undershoots for higher grid levels.
Weakening the stopping criterion allows to obtain more solutions on higher grids on the one hand, on the other hand it leads to higher calculation inaccuracies.



(a) SUPG, $\epsilon = 10^{-6}$, Grid 4, level 8

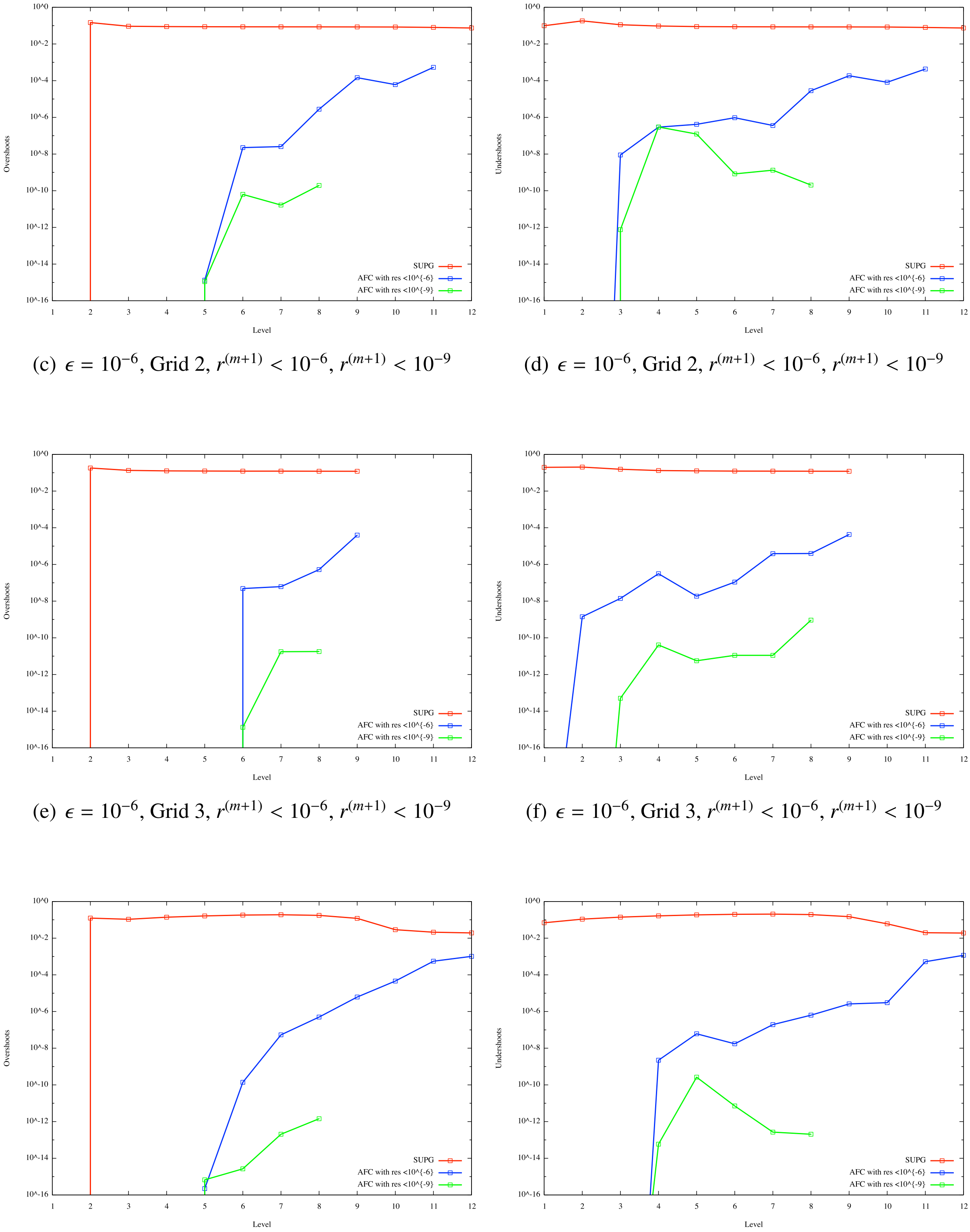
(b) AFC, $\epsilon = 10^{-6}$, Grid 4, level 8

Figure 3.12: Two solutions for SUPG and AFC.



(a) $\epsilon = 10^{-6}$, Grid 1, $r^{(m+1)} < 10^{-6}$ ,$r^{(m+1)} < 10^{-9}$

(b) $\epsilon = 10^{-6}$, Grid 1,$r^{(m+1)} < 10^{-6}$, $r^{(m+1)} < 10^{-9}$



(c) $\epsilon = 10^{-6}$, Grid 2, $r^{(m+1)} < 10^{-6}$, $r^{(m+1)} < 10^{-9}$

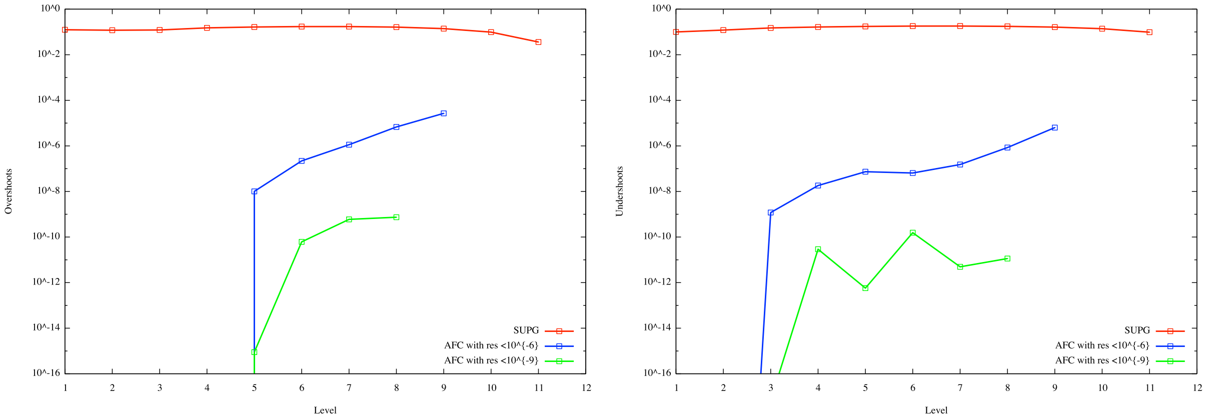
(d) $\epsilon = 10^{-6}$, Grid 2, $r^{(m+1)} < 10^{-6}$, $r^{(m+1)} < 10^{-9}$

(e) $\epsilon = 10^{-6}$, Grid 3, $r^{(m+1)} < 10^{-6}$, $r^{(m+1)} < 10^{-9}$

(f) $\epsilon = 10^{-6}$, Grid 3, $r^{(m+1)} < 10^{-6}$, $r^{(m+1)} < 10^{-9}$

(g) $\epsilon = 10^{-6}$, Grid 4, $r^{(m+1)} < 10^{-6}$, $r^{(m+1)} < 10^{-9}$

(h) $\epsilon = 10^{-6}$, Grid 4, $r^{(m+1)} < 10^{-6}$, $r^{(m+1)} < 10^{-9}$



(i) $\epsilon = 10^{-6}$, Grid 5, $r^{(m+1)} < 10^{-9}$

(j) $\epsilon = 10^{-6}$, Grid 5, $r^{(m+1)} < 10^{-9}$

Figure 3.13: Overshoots and undershoots for AFC and SUPG with stopping criterion $r^{(m+1)} < 10^{-6}$ and $r^{(m+1)} < 10^{-9}$.

# 4 Summary

The main advantages of algebraic flux stabilizations are their ability to satisfy discrete maximum principles due to their positivity-preserving property. Nevertheless, they impose certain restraints on the velocity field $b$. It has to be incompressible in order to guarantee the zero row-sum property which is a fundamental condition of this algorithm. In addition, the algorithm can only be performed with $P^1$ and $Q^1$ finite elements. Time-dependent problems solved with the $\theta$-scheme lead to constraints on the time step $\Delta t$ which are not trivial, if a reactive term is part of the equation (see Theorem 2.15).

Our numerical examples showed that algebraic flux stabilizations have main advantages as well as some drawbacks. On the one hand oscillations are reduced by adding artificial diffusion. On the other hand the solution often turns out to be too diffusive. This is particularly visible in Example 3.2 for $\epsilon = 10^{-10}$, where the outlet for AFC is too diffusive.

Another drawback is the dependency between the grid structure and the order of convergence as shown in Tables 3.2-3.7. The $H^1$-semi norm for Grids 2 and 3 is twice as bad as for the other grids. In contrast to Grid 3, Grid 2 is a structured mesh, therefore this behavior can not only stem from unstructured grids. An approach to improve this, is to use adaptive grid refinement as presented in [Kuz10, p. 197ff.].

Example 3.4 revealed three important properties of the AFC algorithm.

(1.) It reduces overshoots and undershoots compared to SUPG, even though they tend to increase for higher levels.

(2.) It does not converge for all levels or at least it needs far more time to reach the stopping criterion.

(3.) Reducing the stopping criterion 2.15 for AFC improves convergence but worsens the amount of over- and undershoots.

The time-dependent Example 3.3 made clear that algebraic stabilizations can also have strong oscillations. Neither smaller time steps, nor a finer grid could resolve this problem.

It remains a task of further analysis to find the cause of the oscillations occurring in the traveling wave example. Altogether, algebraic flux stabilization is a good approach to solve elliptic and parabolic problems which still requires a deeper analysis regarding convergence, dependency on grid structures and error estimations.

# Bibliography

[BB73] Boris, Jay P and David L Book: *Flux-corrected transport. i. shasta, a fluid transport algorithm that works*. Journal of computational physics, 11(1):38–69, 1973.

[BJK16] Barrenechea, G, Volker John, and Petr Knobloch: *Analysis of algebraic flux correction schemes*. SIAM J. Numer. Anal., 2016. in press.

[Eva98] Evans, L.C.: *Partial Differential Equations*. Graduate studies in mathematics. American Mathematical Society, 1998, ISBN 9780821807729.

[GIJW15] Giere, Swetlana, Traian Iliescu, Volker John, and David Wells: *Supg reduced order models for convection-dominated convection–diffusion–reaction equations*. Computer Methods in Applied Mechanics and Engineering, 289:454–474, 2015.

[JN12] John, Volker and Julia Novo: *On (essentially) non-oscillatory discretizations of evolutionary convection-diffusion equations*. J. Comput. Phys., 231:1570–1586, 2012, ISSN 0021-9991.

[KLT05] Kuzmin, D, R Löhner, and S Turek: *Flux-Corrected Transport: Principles, Algorithms, and Applications. Scientific Computation*. Springer, 2005.

[KM05] Kuzmin, Dmitri and Matthias Möller: *Algebraic flux correction. I. Scalar conservation laws*. In *Flux-corrected transport*, 155206. Springer, Berlin, 2005.

[Kuz07] Kuzmin, DMITRI: *Algebraic flux correction for finite element discretizations of coupled systems*. Computational Methods for Coupled Problems in Science and Engineering II, CIMNE, Barcelona, pages 653–656, 2007.

[Kuz08] Kuzmin, Dmitri: *On the design of algebraic flux correction schemes for quadratic finite elements*. J. Comput. Appl. Math., 218(1):79–87, 2008, ISSN 0377-0427.

[Kuz09] Kuzmin, Dmitri: *Explicit and implicit FEM-FCT algorithms with flux linearizetion*. J. Comput. Phys., 228(7):2517–2534, 2009, ISSN 0021-9991.

[Kuz10] Kuzmin, Dmitri: *A guide to numerical methods for transport equations*. 2010.

[Zal79] Zalesak, Steven T: *Fully multidimensional flux-corrected transport algorithms for fluids*. Journal of computational physics, 31(3):335–362, 1979.

Freie Universität Berlin

# Selbstständigkeitserklärung

| | |
|---|---|
| Name: | (Nur Block- oder Maschinenschrift verwenden.) |
| Vorname: | |
| geb.am: | |
| Matr.Nr.: | |

Ich erkläre gegenüber der Freien Universität Berlin, dass ich die vorliegende ____________________________ selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe.

Die vorliegende Arbeit ist frei von Plagiaten. Alle Ausführungen, die wörtlich oder inhaltlich aus anderen Schriften entnommen sind, habe ich als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch bei keiner anderen Universität als Prüfungsleistung eingereicht und ist auch noch nicht veröffentlicht.

Datum: _________________ Unterschrift: ___________________________

(_______________________)