Robust and Accurate Finite Volume Discretization for Linear Advective Transport

Bachelor Thesis
by

Katharina Charlotte Lina Hoffmann

Matrikelnummer: 369316

born in
Wolfhagen

submitted to

Institut für Mathematik
Prof. Dr. Günter Bärwolff
Technische Universität Berlin

August 2018

Supervisor:
Prof. Dr. Günter Bärwolff
Priv.-Doz. Dr. Alexander Linke

Abstract

The following thesis focuses on two different *finite volume schemes* which are used to solve a partial differential equation, the pure advection equation. Those schemes approximate the flux at the Voronoi faces between two grid points. The two schemes, *simple upwind scheme* and the *complete flux scheme*, vary in their convergence and behavior.

The *simple upwind scheme* has a first order convergence, but maintains the maximum and minimum principle under certain conditions, and one obtains an artificial diffusion. For the *complete flux scheme* one obtains a second order convergence, but the maximum principle is violated due to oscillations.

Zusammenfassung

Die folgende Arbeit beschäftigt sich mit der numerischen Lösung von partiellen Differentialgleichungen, genauer der reinen Advektionsgleichung. Dafür werden zwei verschiedene Finite-Volumen-Schemata betrachtet; das Simple-Upwind-Schema und das Complete-Flux-Schema, welche den Fluss an den Voronoi-Faces approximieren. Diese unterscheiden sich vor allem in ihrer Konvergenz und dem Verhalten gegenüber dem Maximumprinzip.

Das Simple-Upwind-Schema hat nur eine Konvergenz erster Ordnung, erhält aber das Maximumprinzip und eine künstliche Diffusion ist zu sehen. Für das Complete-Flux-Scheme erhält man eine Konvergenzordnung zweiter Ordnung, jedoch wird das Maximumprinzip aufgrund von Oszillationen verletzt.

Contents

List of Figures

1. Introduction

Conservation laws are fundamental in physics, they occur in all kinds of disciplines such as fluid mechanics, gas dynamics, chemical reactions.

Their computer-based simulation is of great importance and the aim is to provide a stable approximation with a good error estimate. The following thesis will focus on the *simple upwind scheme* and the *complete flux scheme* for pure advective transport on an arbitrary 1D-grid.

Consider the pure advection equation

$$c_t + (cv)_x = s, \tag{1.1}$$

where $x \in (a,b)$, $a < b$, where $\varphi := cv$ will be called the flux, s is the source term and v is a velocity. We will consider the initial value problem, where $c(0,x) = c_0(x)$ is prescribed. Additionally, we assume periodic boundary conditions $c(t,a) = c(t,b)$ for all $t \in [0,T]$ with a finite end time $T > 0$. For the purpose of a numerical convergence analysis, we as well make the following assumptions. Let the velocity v be strictly positive with $v \in C^1[a,b]$ with $v(a) = v(b)$, $v'(a) = v'(b)$. For the source term, we assume s be of the regularity $s \in C^2(a,b)$ with $s(a) = s(b)$.

1.1 Notation of the finite volume method

For the space discretization the *Voronoi finite volume method* is introduced. This method is based on applying the conservation law on each of the Voronoi boxes. Herefore, we will define the following grid system:

A set of grid nodes $n \geq 2$ is given as $x_1 = a < x_2 < x_3 \ldots < x_{n-1} < x_n = b$. Then, the Voronoi faces are defined as $W = x_{i+\frac{1}{2}} := \frac{x_i + x_{i+1}}{2}$, $i = 1, \ldots, n-1$.

The control volumes for the *Voronoi finite volume method* are given by the set of Voronoi boxes

$$V = \{V_i := (x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}) : i = 2, \ldots, n-1\} \cup \{V_1 := (a, x_{\frac{3}{2}}) \cup (x_{n-\frac{1}{2}}, b)\}.$$

Figure 1.1 gives an example for such a Voronoi box grid, see also [FFG$^+$ep]. Note that we have only $n-1$ Voronoi boxes due to periodic boundary conditions. In fact, we will always assume that the discrete solution at the grid points x_1 and x_n coincides. Therefore, we need only one control volume for these two grid points.

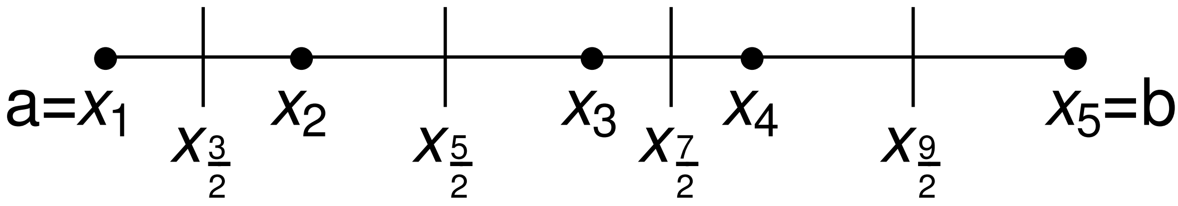

Figure 1.1: A grid for $I = [a, b] = [0, 1]$ with five grid points

Further, we define the local mesh size $h_i := vol(V_i) = x_{i+1} - x_i$, and the global mesh size is defined by $h := \max_{i=1,\dots,n-1} h_i$.

1.2 Approximation of the source term

In the following chapter the source term s will be approximated as shown in Figure 1.2. There are two different approximations of the original source term that will be considered.
First, the piecewise constant source term approximation is given by

$$s_{\text{pc}}(x) = \begin{cases} s(x_i), & \text{for } x_i \leq x < x_{i+\frac{1}{2}}, \quad i = 1, \dots, n-1 \\ s(x_{i+1}), & \text{for } x_{i+\frac{1}{2}} \leq x < x_{i+1}, \quad i = 1, \dots, n-1. \end{cases}$$

Note that it holds $s(x_1) = s(x_n)$ due to the periodic boundary conditions. Therefore, the value of the source term in the Voronoi boxes is constant for all $i = 1, \dots, n-1$.

Second, a continuous, piecewise affine source term approximation is given by

$$s_{\text{affine}}(x)_{|(x_i, x_{i+1})} = s_i + \frac{s_{i+1} - s_i}{x_{i+1} - x_i} (x - x_i), \quad \text{for all } i = 1, \dots, n-1.$$

This approximation is extremely helpful because the integral of s_{pc} is equal to the integral of s_{affine} over the interval (a, b), as shown in the following Lemma 1.
Lemma 1. *For the integral of* s_{pc} *and the integral of* s_{affine} *over the interval* (x_i, x_{i+1}) *it holds*

$$\int_{x_i}^{x_{i+1}} s_{\text{pc}}(x)\, dx = \int_{x_i}^{x_{i+1}} s_{\text{affine}}(x)\, dx.$$

Proof. By integrating s_{pc} from x_i to x_{i+1}, one obtains

$$\int_{x_i}^{x_{i+1}} s_{\text{pc}}(x)\, dx = s_i \frac{x_{i+1} - x_i}{2} + s_{i+1} \frac{x_{i+1} - x_i}{2} = (s_i + s_{i+1}) \frac{x_{i+1} - x_i}{2}.$$

This follows directly by looking at Figure 1.2 or the definiton of s_{pc}.

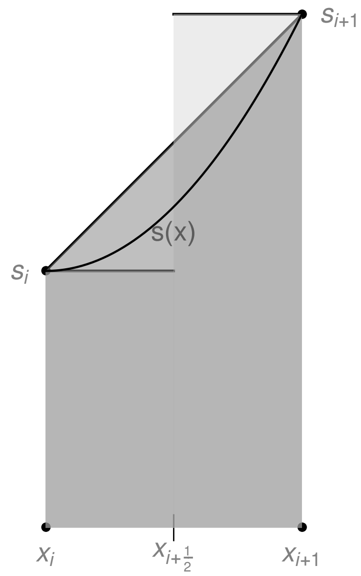

Figure 1.2: Approximation of source term $s(x)$ by piecewise affine and piecewise constant functions

By integrating s_{affine} from x_i to x_{i+1}, one obtains

$$\int_{x_i}^{x_{i+1}} s_{\text{affine}}(x)\,dx = s_i(x_{i+1}-x_i) + \frac{1}{2}(x_{i+1}-x_i)(s_{i+1}-s_i) = (s_i+s_{i+1})\frac{x_{i+1}-x_i}{2}.$$

□

Furthermore, the following approximation can be made and is useful for later error approximation.

Lemma 2. *[FFG$^+$ep] For the source term s(x) and the piecewise affine function s_{affine} it holds*

$$|s(x) - s_{\text{affine}}(x)| \leq \frac{3}{4} h_j^2 \|s''\|_{L^\infty(x_j,x_{j+1})}$$

for all $j = 1, \ldots, n-1$.

Proof. A Taylor expansion of $s(x)$ at $x_{j+\frac{1}{2}}$ and then substituting x_j, respectively x_{j+1}, one obtains

$$s(x_j) = s(x_{j+\frac{1}{2}}) - s'(x_{j+\frac{1}{2}})(x_{j+\frac{1}{2}} - x_j) + \int_{x_j}^{x_{j+\frac{1}{2}}} \int_{\xi}^{x_{j+\frac{1}{2}}} s''(\eta)\,d\eta\,d\xi$$

and

$$s(x_{j+1}) = s(x_{j+\frac{1}{2}}) + s'(x_{j+\frac{1}{2}})(x_{j+1} - x_{j+\frac{1}{2}}) + \int_{x_{j+\frac{1}{2}}}^{x_{j+1}} \int_{x_{j+\frac{1}{2}}}^{\xi} s''(\eta)\,d\eta\,d\xi.$$

By adding those two equations, it holds

$$\left| s(x_{j+\frac{1}{2}}) - \frac{1}{2}\left(s(x_j) + s(x_{j+1})\right) \right| \leq \frac{h_j^2}{4} \|s''\|_{L^\infty(x_j, x_{j+1})},$$

and by subtracting both equations, one gets

$$\left| s'(x_{j+\frac{1}{2}}) - \frac{s(x_{j+1}) - s(x_j)}{x_{j+1} - x_j} \right| \leq \frac{h_j}{2} \|s''\|_{L^\infty(x_j, x_{j+1})}.$$

Another Taylor expansion of $s_{\text{affine}}(x)$ at $x_{j+\frac{1}{2}}$ it holds

$$s_{\text{affine}}(x) = \frac{1}{2}\left(s(x_j) + s(x_{j+1})\right) + \frac{s(x_{j+1}) - s(x_j)}{x_{j+1} - x_j}(x - x_{j+\frac{1}{2}}).$$

Therefore, it holds for all $x \in (x_j, x_{j+1})$ for $j = 1, \ldots, n-1$

$$|s(x) - s_{\text{affine}}(x)| \leq \frac{3}{4} h_j^2 \|s''\|_{L^\infty(x_j, x_{j+1})}.$$

□

2. Steady periodic advection problem

The steady periodic advection problem of (1.1) is given by

$$\begin{aligned} (cv)_x &= s, \qquad x \in (a,b), \\ c(a) &= c(b) \\ \int_a^b c(x)\,dx &= M, \end{aligned} \tag{2.1}$$

where $M > 0$ denotes a prescribed finite mass. Additionally, the compatibility condition $\int_a^b s(x)\,dx = 0$ is required for the source term.
For the discretization of the problem we introduce the following interpolation operator

$$(I_n s)(x) := s_{\mathrm{pc}} + l \tag{2.2}$$

for the source term s, which will preserve the integral $\int_a^b s(x)\,dx$, where the correction term l is defined by

$$l := \frac{1}{b-a}\int_a^b (s(x) - s_{\mathrm{pc}}(x))\,dx. \tag{2.3}$$

Then, it holds

$$\begin{aligned} \int_a^b I_n s(x)\,dx &= \int_a^b (s_{\mathrm{pc}}(x) + l)\,dx = \int_a^b s_{\mathrm{pc}}(x)\,dx + (b-a)l \\ &= \int_a^b s_{\mathrm{pc}}(x)\,dx + \int_a^b (s(x) - s_{\mathrm{pc}}(x))\,dx = \int_a^b s(x)\,dx. \end{aligned}$$

An important goal is to show that the interpolation operator I_n approximates the source term with second order accuracy in some sense.

Lemma 3. *For the correction term l it holds*

$$|l| = |\frac{1}{b-a}\int_a^b (s(x) - s_{\mathrm{pc}}(x))\,dx| \le \frac{3}{4} h_j^2 \|s''\|_{L^\infty(a,b)}.$$

Proof. Using Lemma 1 over the interval $[a, b]$ it follows

$$\begin{aligned}
&\left|\frac{1}{b-a}\int_a^b (s(x) - s_{\text{pc}}(x))\,dx\right| \\
=&\left|\frac{1}{b-a}\int_a^b (s(x) - s_{\text{affine}}(x))\,dx\right| \\
\leq&\frac{1}{b-a}\int_a^b \left|s(x) - s_{\text{affine}}(x)\right|\,dx.
\end{aligned}$$

Thus, applying Lemma 2 to this equation one obtains

$$\begin{aligned}
|l| &\leq \frac{1}{b-a}\int_a^b |s(x) - s_{\text{affine}}(x)|\,dx \\
&= \frac{1}{b-a}\sum_{i=1}^{n-1}\int_{x_i}^{x_{i+1}} |s(x) - s_{\text{affine}}(x)|\,dx \\
&\leq \frac{1}{b-a}\sum_{i=1}^{n-1}\int_{x_i}^{x_{i+1}} \frac{3}{4}h_i^2\|s''\|_{L^\infty(x_i,x_{i+1})}\,dx \\
&\leq \frac{1}{b-a}\sum_{i=1}^{n-1}\int_{x_i}^{x_{i+1}} \frac{3}{4}h^2\|s''\|_{L^\infty(a,b)}\,dx \\
&\leq \frac{1}{b-a}\frac{3}{4}h^2\|s''\|_{L^\infty(a,b)}\sum_{i=1}^{n-1}\int_{x_i}^{x_{i+1}} 1\,dx \\
&\leq \frac{1}{b-a}\frac{3}{4}h^2\|s''\|_{L^\infty(a,b)}(b-a) \\
&\leq \frac{3}{4}h^2\|s''\|_{L^\infty(a,b)}.
\end{aligned}$$

□

For further approximation and to evaluate the continuous solution for both schemes we need a continuous representation for c_a.

Lemma 4. *For the boundary value $c_a^{\text{con}}(s)$ for a continuous periodic problem, with the right hand side s, it holds*

$$c_a^{\text{con}}(s) = \frac{1}{\int_a^b \frac{v(a)}{v(x)}\,dx}\left(M - \int_a^b \frac{1}{v(x)}\int_a^x s(z)\,dz\,dx\right).$$

Proof. Applying the fundamental theorem of calculus to (2.1) one obtains

$$\begin{aligned} &\int_a^x (vc)_z \, dz = \int_a^x s(z) \, dz \\ \Leftrightarrow \quad & v(x)c(x) - v(a)c_a^{\mathrm{con}}(s) = \int_a^x s(z) \, dz \\ \Leftrightarrow \quad & c(x) = \frac{1}{v(x)} \left(v(a)c_a^{\mathrm{con}}(s) + S(x) \right) \end{aligned}$$

with $S(x) = \int_a^x s(z) \, dz$.
By integrating $c(x)$ from a to b, it follows

$$\int_a^b c(x) \, dx = M = c_a^{\mathrm{con}}(s) \int_a^b \frac{v(a)}{v(x)} \, dx + \int_a^b \frac{1}{v(x)} \int_a^x s(z) \, dz \, dx.$$

Then $c_a^{\mathrm{con}}(s)$ is given as

$$\begin{aligned} c_a^{\mathrm{con}}(s) =& \frac{1}{\int_a^b \frac{v(a)}{v(x)} \, dx} \left(M - \int_a^b \frac{1}{v(x)} \int_a^x s(z) \, dz \, dx \right) \\ =& \frac{1}{v(a)\tau} \left(M - \int_a^b \frac{1}{v(x)} \int_a^x s(z) \, dz \, dx \right) \end{aligned}$$

with

$$\tau := \int_a^b \frac{1}{v(x)} \, dx. \tag{2.4}$$

□

Therefore, the exact solution of (2.1) is given by

$$c(x) = \frac{1}{v(x)} \left(\int_a^x s(z) \, dz + v(a) c_a^{\mathrm{con}}(s) \right). \tag{2.5}$$

For the error approximation of the *finite volume scheme* it is helpful to define an auxiliary function $\bar{c}$, which solves exactly the problem where $s(x)$ is approximated by s_{pc} and the velocity $v(x)$ is approximated by v_{affine}

$$\begin{aligned} (\bar{c} v_{\mathrm{affine}})_x &= I_n s, \qquad x \in (a, b), \\ \bar{c}(a) &= \bar{c}(b) \\ \int_a^b \bar{c}(x) \, dx &= M, \end{aligned} \tag{2.6}$$

with

$$v_{\mathrm{affine}}(x)_{|(x_i, x_{i+1})} = v(x_i) + \frac{v(x_{i+1}) - v(x_i)}{x_{i+1} - x_i} (x - x_i), \quad \text{for all } i = 1, \ldots, n-1.$$

Thus, the exact solution of (2.6) is given by

$$\bar{c}(x) = \frac{1}{v(x)} \left(\int_a^x (I_n s)(z) \, dz + v(a) c_a^{\mathrm{con}}(I_n s) \right). \tag{2.7}$$

The following error estimation for c and $\bar{c}$ can then be made.

Lemma 5. *For the solutions c and $\bar{c}$ it holds*

$$\|c - \bar{c}\|_{l_\infty} \leq \frac{3}{2} \left(1 + \frac{v_{max}}{v_{min}} \right) \frac{(b-a)}{v_{min}} h^2 \|s''\|_{L^\infty(a,b)}.$$

Proof. For all x_i with $i \in \{1, \ldots, n\}$ it holds

$$\begin{aligned} |c(x_i) - \bar{c}(x_i)| &= \left| \left(\frac{1}{v(x_i)} \int_a^{x_i} (s(\xi) - (I_n s)(\xi)) \, d\xi + v(a) c_a^{\mathrm{con}}(s) - v(a) c_a^{\mathrm{con}}(I_n s) \right) \right| \\ &\leq \frac{1}{v(x_i)} \left(\left| \int_a^{x_i} s(\xi) - (I_n s)(\xi) d\xi \right| + |v(a) c_a^{\mathrm{con}}(s) - v(a) c_a^{\mathrm{con}}(I_n s)| \right). \end{aligned}$$

We estimate the first term on the right hand side by

$$\begin{aligned} &\frac{1}{v(x_i)} \left| \int_a^{x_i} (s(\xi) - (I_n s)(\xi)) \, d\xi \right| \\ =&\frac{1}{v(x_i)} \left| \int_a^{x_i} (s(\xi) - s_{\mathrm{pc}}(\xi) - l) \, d\xi \right| \\ =&\frac{1}{v(x_i)} \left| \int_a^{x_i} (s(\xi) - s_{\mathrm{affine}}(\xi) - l) \, d\xi \right| \\ \leq&\frac{1}{v_{min}} \int_a^b |(s(\xi) - s_{\mathrm{affine}}(\xi) - l)| \, d\xi \\ \leq&\frac{1}{v_{min}} \left(\int_a^b |s(\xi) - s_{\mathrm{affine}}(\xi)| \, d\xi + (b-a)|l| \right) \\ \leq&\frac{1}{v_{min}} \left(\int_a^b |s(\xi) - s_{\mathrm{affine}}(\xi)| \, d\xi + (b-a) \left| \frac{1}{b-a} \int_a^b (s(x) - s_{\mathrm{pc}}(x)) dx \right| \right) \\ \leq&\frac{1}{v_{min}} \left(\int_a^b |s(\xi) - s_{\mathrm{affine}}(\xi)| \, d\xi + (b-a) \left| \frac{1}{b-a} \int_a^b (s(x) - s_{\mathrm{affine}}(x)) dx \right| \right) \\ \leq&\frac{2}{v_{min}} \int_a^b |s(\xi) - s_{\mathrm{affine}}(\xi)| \, d\xi \\ \leq&\frac{2}{v_{min}} \frac{3}{4} h^2 \|s''\|_{L^\infty(a,b)} (b-a) \\ \leq&\frac{3(b-a)}{2 v_{min}} h^2 \|s''\|_{L^\infty(a,b)}. \end{aligned}$$

Here we used Lemma 1 and the same estimation as in Lemma 3.

We estimate the second term on the right hand side by

$$
\begin{aligned}
&\frac{1}{v(x_i)}|(v(a)c_a^{\text{con}}(s) - v(a)c_a^{\text{con}}(I_n s))| \\
=&\frac{1}{v(x_i)}\left|\frac{v(a)}{v(a)\tau}\left(\int_a^b \frac{1}{v(x)}\int_a^x (s(z) - (I_n s)(z))\,dz\,dx\right)\right| \\
\leq&\frac{1}{v(x_i)}\frac{1}{\tau}\frac{b-a}{v_{min}}\left|\int_a^x (s(z) - (I_n s)(z))\,dz\right|
\end{aligned}
$$

for some $x \in [a, b]$.
Similar to the argument above we estimate $\frac{1}{v(x_i)}\left|\int_a^x (s(z) - (I_n s)(z))\,dz\right|$ by $\frac{3(b-a)}{2v_{min}}h^2\|s''\|_{L^\infty(a,b)}$. Due to the definition of τ in (2.4), it holds, where v_{max} is the maximum of the continuous function v in the interval $[a, b]$

$$
\begin{aligned}
\tau &\geq \frac{b-a}{v_{max}} \\
\Rightarrow \frac{1}{\tau} &\leq \frac{v_{max}}{b-a}.
\end{aligned}
$$

Therefore, we get

$$
\begin{aligned}
&\frac{1}{\tau}\frac{b-a}{v_{min}}\frac{3(b-a)}{2v_{min}}h^2\|s''\|_{L^\infty(a,b)} \\
\leq&\frac{v_{max}}{v_{min}}\frac{3(b-a)}{2v_{min}}h^2\|s''\|_{L^\infty(a,b)}.
\end{aligned}
$$

This concludes the proof. □

2.1 Abstract finite volume scheme

If a flux approximation $\varphi_{i+\frac{1}{2}} \approx \varphi(x_{i+\frac{1}{2}})$ is given, one can construct an abstract *finite volume scheme* for the steady advection problem with periodic boundary conditions. One gets $n-1$ conditions by integrating over the Voronoi boxes V_i for $i = 1, \ldots, n-1$

$$
\begin{aligned}
\varphi_{\frac{3}{2}} - \varphi_{n-\frac{1}{2}} &= (s_1 + l)\left(x_{\frac{3}{2}} - a\right) - (s_n + l)\left(b - x_{n-\frac{1}{2}}\right), \\
\varphi_{i+\frac{1}{2}} - \varphi_{i-\frac{1}{2}} &= (s_i + l)\left(x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}\right), \qquad i = 2, \ldots, n-1,
\end{aligned} \tag{2.8}
$$

where l is the correction term (2.3) that we need for the discrete compatibility condition and $s_i = s(x_i)$ for all $i = 1, \ldots, n$.

For the steady advection problem, we additionally have the condition

$$\sum_{i=1}^{n-1} m_i c_i = M, \tag{2.9}$$

with $m_i := |V_i| = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$.

Remark 1. *In this abstract framework, nothing is said about the solvability of* (2.9) *and* (2.8) *with respect to* c_i *for* $i = 1, \ldots, n-1$.

Lemma 6. *For the flux approximation* $\varphi(x_{i+\frac{1}{2}})$ *and the discrete solution of* (2.6) *it holds*

$$\varphi(x_{i+\frac{1}{2}}) - \bar{c}(x_{i+\frac{1}{2}}) v_{\text{affine}}(x_{i+\frac{1}{2}}) = \varphi(x_{\frac{3}{2}}) - \bar{c}(x_{\frac{3}{2}}) v_{\text{affine}}(x_{\frac{3}{2}}),$$

for $i = 2, \ldots, n-1$.

Proof. It holds

$$\begin{aligned} \varphi_{i+\frac{1}{2}} - \varphi_{i-\frac{1}{2}} &= (s_i + l)\left(x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}\right) \\ \Rightarrow \varphi_{i+\frac{1}{2}} &= \varphi_{\frac{3}{2}} + \int_{x_{\frac{3}{2}}}^{x_{i-\frac{1}{2}}} (I_n s)(z)\, dz \end{aligned}$$

for $i = 2, \ldots, n-1$, see (2.8).
Using (2.6), one obtains

$$\begin{aligned} \varphi_{i+\frac{1}{2}} &= \varphi_{\frac{3}{2}} + \bar{c}(x_{i+\frac{1}{2}}) v_{\text{affine}}(x_{i+\frac{1}{2}}) - \bar{c}(x_{\frac{3}{2}}) v_{\text{affine}}(x_{\frac{3}{2}}) \\ \Rightarrow \varphi(x_{i+\frac{1}{2}}) - \bar{c}(x_{i+\frac{1}{2}}) v_{\text{affine}}(x_{i+\frac{1}{2}}) &= \varphi(x_{\frac{3}{2}}) - \bar{c}(x_{\frac{3}{2}}) v_{\text{affine}}(x_{\frac{3}{2}}) \end{aligned}$$

for $i = 2, \ldots, n-1$. □

In this thesis we will only examine the error estimation of the two different schemes for the steady advection problem with an initial value, see Lemma 8 and Lemma 9.
The error estimation of the analytical solution and the numerical solution, which one obtains by solving the advection problem with periodic boundary conditions, will be further discussed in [FFG$^+$ep]. This being said, the knowledge we obtain for the convergence order of the *simple upwind scheme*, respectivley the *complete flux scheme*, can be applied to the periodic problem as well.
The steady advection problem with an inital value is given by

$$\begin{aligned} (cv)_x &= s, \qquad x \in (a, b), \\ c(a) &= c_a, \end{aligned} \tag{2.10}$$

with the velocity v being strictly positive with $v \in C^1[a,b]$ and the source term s being of the regularity $s \in C^2(a,b)$.
For this problem we have a slightly different setting with for the Voronoi boxes. A set of grid nodes $n \geq 2$ is given as $N = x_1 = a < x_2 < x_3 \ldots < x_{n-1} < x_n = b$. Then the Voronoi faces are defined as $W = x_{i+\frac{1}{2}} := \frac{x_i + x_{i+1}}{2} : i = 1, \ldots, n-1$. The control volumes for the Voronoi finite volume method are given by the set of Voronoi boxes

$$V = \{V_i := (x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}) : i = 2, \ldots, n-1\} \cup \{V_1 := (a, x_{\frac{3}{2}}), V_n := (x_{n-\frac{1}{2}}, b)\}.$$

Note that we now have n Voronoi boxes. By applying the fundamental theorem of calculus the exact solution of (2.10) is given by

$$\begin{aligned} \int_a^x (vc)_z \, dz &= \int_a^x s(z) \, dz \\ \Leftrightarrow \quad v(x)c(x) - v(a)c_a &= \int_a^x s(z) \, dz \\ \Leftrightarrow \quad c(x) &= \frac{1}{v(x)} \left(\int_a^x s(z) \, dz + v(a)c_a \right). \end{aligned}$$

2.2 Simple upwind scheme

The *simple upwind discretization*, which approximates the flux $\varphi(x_{i+\frac{1}{2}})$ between the nodes x_i and x_{i+1}, is defined as

$$\varphi^{\mathrm{SU}}_{i+\frac{1}{2}} := c_i \frac{v(x_{i+1}) + v(x_i)}{2} \tag{2.11}$$

for all $i = 1, \ldots, n-1$. By integrating over the Voronoi boxes V_i for $i = 2, \ldots, n-1$, one gets

$$\varphi_{i+\frac{1}{2}} - \varphi_{i-\frac{1}{2}} = c_i \frac{v(x_{i+1}) + v(x_i)}{2} - c_{i-1} \frac{v(x_i) + v(x_{i-1})}{2}.$$

Lemma 7. *The discrete value $c_a^{disc}(s) := c_1$ is given as*

$$c_a^{disc}(s) = \frac{1}{v_{\frac{3}{2}} \tau^{\mathrm{SU}}} \left(M - \sum_{i=1}^{n-1} m_i - \frac{1}{v_{i+\frac{1}{2}}} \int_{x_{\frac{3}{2}}}^{x_{i+\frac{1}{2}}} (I_n s)(z) \, dz \right)$$

with $\tau^{\mathrm{SU}} := \sum_{i=1}^{n-1} m_i \frac{1}{v_{i+\frac{1}{2}}}$ and $v_{i+\frac{1}{2}} = \frac{v(x_{i+1}) + v(x_i)}{2}$, for $i = 1, \ldots, n-1$.

Proof. Using Lemma (6), one obtains

$$\begin{aligned}
\varphi_{i+\frac{1}{2}} =& \varphi_{\frac{3}{2}} + \int_{x_{\frac{3}{2}}}^{x_{i-\frac{1}{2}}} (I_n s)(z)\, dz \\
\Rightarrow c_i v_{i+\frac{1}{2}} =& c_1 v_{\frac{3}{2}} + \int_{x_{\frac{3}{2}}}^{x_{i-\frac{1}{2}}} (I_n s)(z)\, dz \\
\Rightarrow c_i =& \frac{1}{v_{i+\frac{1}{2}}} \left(c_1 v_{\frac{3}{2}} + \int_{x_{\frac{3}{2}}}^{x_{i-\frac{1}{2}}} (I_n s)(z)\, dz \right) \\
\Rightarrow M = \sum_{i=1}^{n-1} m_i c_i =& c_1 v_{\frac{3}{2}} \tau^{\mathrm{SU}} + \sum_{i=1}^{n-1} m_i \frac{1}{v_{i+\frac{1}{2}}} \int_{x_{\frac{3}{2}}}^{x_{i-\frac{1}{2}}} (I_n s)(z)\, dz \\
\Rightarrow c_a^{disc}(I_n s) = c_1 =& \frac{1}{v_{\frac{3}{2}} \tau^{\mathrm{SU}}} \left(M - \sum_{i=1}^{n-1} m_i - \frac{1}{v_{i+\frac{1}{2}}} \int_{x_{\frac{3}{2}}}^{x_{i+\frac{1}{2}}} (I_n s)(z)\, dz \right).
\end{aligned}$$

□

Lemma 8. *Let c_i be the solution of* (2.10)*, which one obtains by using the finite volume discretization with the simple upwind flux. Then the error estimate*

$$\begin{aligned}
|c(x_i) - c_i| \leq& h \left(\frac{\|s\|_{L^\infty}}{2v_{\min}} + \frac{\|v'\|_{L^\infty} \left(|c_a v_a| + (b-a)\|s\|_{L^\infty} \right)}{2v_{\min}^2} \right) + \frac{3(b-a)}{4v_{\min}} h^2 \|s''\|_{L^\infty(a,b)} \\
\leq& O(h) \qquad (2.12)
\end{aligned}$$

holds for all $i = 1, \ldots, n$.

Proof. It holds

$$|c(x_i) - c_i| \leq |c(x_i) - \bar{c}(x_i)| + |\bar{c}(x_i) - c_i| . \qquad (2.13)$$

According to Lemma 5, the first part can be estimated by

$$|c(x_i) - \bar{c}(x_i)| \leq O(h^2).$$

For the second part it holds

$$\bar{c}(x_i) - c_i = \left(\frac{1}{v(x_i)} - \frac{1}{v(x_{i+\frac{1}{2}})} \right) \left(c_a v(a) + \int_a^{x_i} s_{\mathrm{pc}}(\xi)\, \mathrm{d}\xi \right) - \frac{s_i}{v(x_{i+\frac{1}{2}})} \left(x_{i+\frac{1}{2}} - x_i \right).$$

The second term on the right can be estimated by

$$\left| - \frac{s_i}{v(x_{i+\frac{1}{2}})} \left(x_{i+\frac{1}{2}} - x_i \right) \right| \leq \frac{1}{2v_{\min}} h \|s\|_{L^\infty}.$$

Furthermore, one obtains

$$\left|\frac{1}{v(x_i)} - \frac{1}{v(x_{i+\frac{1}{2}})}\right| = \left|\frac{v(x_{i+\frac{1}{2}}) - v(x_i)}{v(x_i)v(x_{i+\frac{1}{2}})}\right| \leq \frac{h_i|v'(\xi_i)|}{2v(x_i)v(x_{i+\frac{1}{2}})}$$

with $\xi_i \in (x_i, x_{i+\frac{1}{2}})$.
Eventually comparing those estimations with (2.13) , one obtains

$$|\bar{c}(x_i) - c_i| \leq h\left(\frac{\|s\|_{L^\infty}}{2v_{\min}} + \frac{\|v'\|_{L^\infty}\left(|c_a v_a| + (b-a)\|s\|_{L^\infty}\right)}{2v_{\min}^2}\right) \leq O(h).$$

□

Even though the *simple upwind scheme* is stable because of its first-order convergence, the error can get really big depending on the mesh size.

2.3 Complete flux scheme

Our goal is to define a second-order convergence scheme, see [tTBA11]. Herefore, we assume that one knows the velocity $v(x)$ and its derivative $v'(x)$ at the Voronoi faces $x_{i+\frac{1}{2}}$ for $i = 1, \ldots, n-1$.
Approximating the velocity $v(x)$ with discontinous, piecewise affine functions, see Figure 2.1 (similar to a Taylor expansion of $v(x)$ at $x(i+\frac{1}{2})$) and with the flux definintion $\varphi = cv$ in mind, it holds

$$\varphi(x_i) = c_i\left(v(x_{i+\frac{1}{2}}) - v'(x_{i+\frac{1}{2}})(x_{i+\frac{1}{2}} - x_i)\right).$$

Because of the assumption that $v(x) > 0$ for all $x \in [a, b]$ it holds
$v'(x_{i+\frac{1}{2}})(x_{i+\frac{1}{2}} - x) > 0$ for all $x \in [a, b]$ (see Figure 2.1).

For the discrete problem it holds

$$\begin{aligned} v(x_{i+\frac{1}{2}}) &= \frac{v(x_{i+1}) + v(x_i)}{2} \\ v'(x_{i+\frac{1}{2}}) &= \frac{v(x_{i+1}) - v(x_i)}{x_{i+1} - x_i}(x_{i+\frac{1}{2}} - x_i). \end{aligned}$$

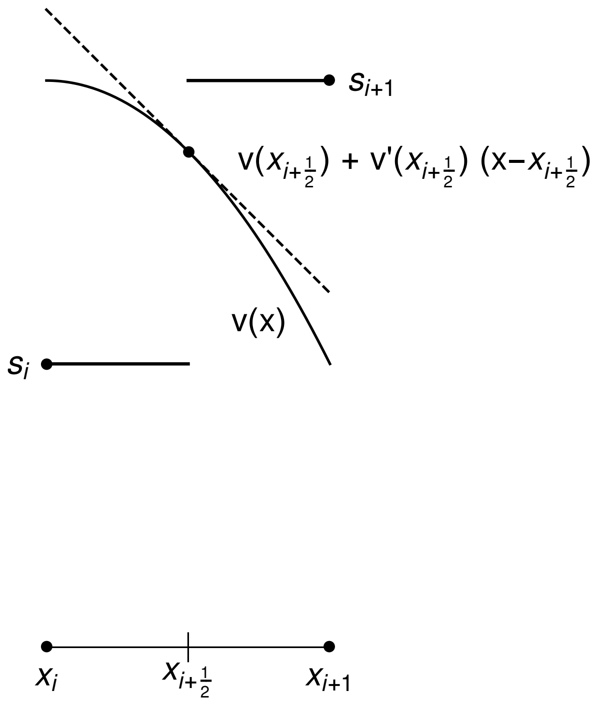

Figure 2.1: Second order complete flux approximation for steady linear advection

Therefore, the *complete flux scheme* is defined as

$$
\begin{aligned}
\varphi^{\mathrm{CFS}}_{i+\frac{1}{2}} :=& c_i \left(\frac{v(x_{i+1}) + v(x_i)}{2} - \frac{v(x_{i+1}) - v(x_i)}{x_{i+1} - x_i} (x_{i+\frac{1}{2}} - x_i) \right) + s_i \left(\frac{x_{i+1} - x_i}{2} \right) \\
=& c_i \left(\frac{v(x_{i+1}) + v(x_i)}{2} - \frac{v(x_{i+1}) - v(x_i)}{x_{i+1} - x_i} \left(\frac{x_{i+1} + x_i}{2} - x_i \right) \right) + s_i \left(\frac{x_{i+1} - x_i}{2} \right) \\
=& c_i \left(\frac{1}{2} (v(x_{i+1}) + v(x_i)) - \frac{v(x_{i+1}) - v(x_i)}{x_{i+1} - x_i} \left(\frac{x_{i+1} - x_i}{2} \right) \right) + \left(\frac{x_{i+1} - x_i}{2} \right) \\
=& c_i v(x_i) + s_i \left(\frac{x_{i+1} - x_i}{2} \right).
\end{aligned}
\tag{2.14}
$$

Lemma 9. *Let c_i be the solution of* (2.10)*, which one obtains by using the finite volume discretization with the complete flux scheme. Then the error estimate*

$$|c(x_i) - c_i| \leq O(h^2)$$

holds for all $i = 1, \dots, n$.

Proof. It holds

$$|c(x_i) - c_i| \leq |c(x_i) - \bar{c}(x_i)| + |\bar{c}(x_i) - c_i| . \tag{2.15}$$

According to Lemma 5, the first part can be estimated by

$$|c(x_i) - \bar{c}(x_i)| \le O(h^2),$$

and for the second part it holds for all $i = 1, \dots, n$

$$\bar{c}(x_i) - c_i = \left(\frac{1}{v(x_i)} - \frac{1}{v(x_{i+\frac{1}{2}}) + v'(x_{i+\frac{1}{2}}) \left(x_i - x_{i+\frac{1}{2}} \right)} \right) \left(c_a v(a) + \int_a^{x_i} s_{\mathrm{pc}}(\xi)\, d\xi \right).$$

The leading term can be estimated by

$$\begin{aligned} \left| \frac{1}{v(x_i)} - \frac{1}{v(x_{i+\frac{1}{2}}) + v'(x_{i+\frac{1}{2}}) \left(x_i - x_{i+\frac{1}{2}} \right)} \right| &= \left| \frac{v(x_{i+\frac{1}{2}}) + v'(x_{i+\frac{1}{2}}) \left(x_i - x_{i+\frac{1}{2}} \right) - v(x_i)}{v(x_i) \left(v(x_{i+\frac{1}{2}}) + v'(x_{i+\frac{1}{2}}) \left(x_i - x_{i+\frac{1}{2}} \right) \right)} \right| \\ &\le \left| \frac{h_i^2}{8} \frac{\|v''\|_{L^\infty(x_i, x_{i+\frac{1}{2}})}}{v(x_i) \left(v(x_{i+\frac{1}{2}}) + v'(x_{i+\frac{1}{2}}) \left(x_i - x_{i+\frac{1}{2}} \right) \right)} \right| \\ &\le O(h^2). \end{aligned}$$

Therefore, it holds for all $x_i \in [a, b]$

$$\begin{aligned} |\bar{c}(x_i) - c_i| &\le O(h^2) \\ \Rightarrow |c(x_i) - c_i| &\le O(h^2). \end{aligned}$$

□

2.4 Eigenvalue problem for the steady problem

For the steady advection problem (2.1), one obtains the following eigenvalue problem

$$(cv)' = \lambda c \quad \text{with} \quad \varphi = cv,$$

where additionally $c(a) = c(b)$ is required.
Assuming an eigenvalue λ is known, the eigenvalue v can be determined as followed.

For φ it holds

$$
\begin{aligned}
&\varphi' = \lambda \frac{1}{v} \varphi \\
\Rightarrow &\frac{\varphi'}{\varphi} = \lambda \frac{1}{v} \\
\Rightarrow &(\log|\varphi|)' = \lambda \frac{1}{v} \\
\Rightarrow &\log|\varphi| = \lambda \int_a^x \frac{1}{v(s)} \, ds \\
\Rightarrow &\varphi = \varphi_a \exp\left(\lambda \int_a^x \frac{1}{v(s)} \, ds\right) \\
\Rightarrow &c = \frac{1}{v} \varphi_a \exp\left(\lambda \int_a^x \frac{1}{v(s)} \, ds\right).
\end{aligned}
$$

The admissible eigenvalues λ can be determined by using the periodic boundary condition

$$
\begin{aligned}
&c(a) = \frac{1}{v} \varphi_a \exp(\lambda 0) = c(b) = \frac{1}{v} \varphi_a \exp\left(\lambda \int_a^b \frac{1}{v(s)} \, ds\right) \\
\Leftrightarrow &\exp(\lambda\tau) = 1,
\end{aligned}
$$

where τ is defined in (2.4).
Since the exponential function is $2\pi i$-periodic, λ is an admissible eigenvalue if and only if

$$
\begin{aligned}
&\lambda_k \tau = 2\pi k i \\
\Leftrightarrow &\lambda_k = \frac{2\pi k i}{\tau}, \quad k \in \mathbb{Z}.
\end{aligned}
$$

3. Time-dependent advection problem

The time-dependent periodic advection problem of (1.1) is given by

$$\begin{aligned} &c_t + (cv)_x = s, \quad x \in (a,b), \\ &c(t,a) = c(t,b) \\ &v(t,a) = v(t,b), \quad \forall t > 0. \end{aligned} \tag{3.1}$$

Additionally, an initial value is prescribed by $c(0,x) = c_0(x)$.

3.1 Mass conservation of the periodic advection problem

First, we make sure that this equation is mass conserving.
If $\int_a^b s(x)\,dx = 0$, it holds

$$\begin{aligned} &\frac{d}{dt}\left[\int_a^b c(x)\,dx\right] \\ =&\left[\int_a^b c_t(x)\,dx\right] \\ =&\int_a^b (-(cv)_x + s(x))\,dx \\ =&-cv(b) + cv(a) \\ =&0. \end{aligned}$$

Hence, the periodic advection problem is mass conserving.

3.2 Three different physical regimes

The time-dependent pure advection equation (1.1) can be rewritten as

$$\frac{1}{v}\varphi_t + \varphi_x = s, \tag{3.2}$$

where $\varphi = cv$ and the same assumptions for s and v are made as for the problem in (1.1).

We can split the problem into three different physical regimes:

$$1. \qquad \frac{1}{v}\varphi_t + \varphi_x = 0 \tag{3.3}$$

$$2. \qquad \varphi_x = s \tag{3.4}$$

$$3. \qquad \frac{1}{v}\varphi_t = \hat{s}. \tag{3.5}$$

If we are able to make statements about e.g. the stability, convergence, solvability for those three equations, we can also make statements regarding those properties for (3.2). Equation (3.4) is a steady advection problem and therefore solvable as shown previously. For equation (3.5) it follows, if $\hat{s} = \frac{\hat{c}}{v}$, with $\hat{c}$ being a constant, that the problem can be rewritten as

$$\begin{aligned} \frac{1}{v}\varphi_t &= \frac{\hat{c}}{v} \\ \varphi_t &= \hat{c}, \end{aligned}$$

and therefore solvable with

$$\varphi = \hat{c}t \Leftrightarrow c^{con} := \frac{\hat{c}t}{v}, \tag{3.6}$$

with c^{con} being the continuous solution to the equation $\frac{1}{v}(cv)_t = \hat{s}$.
The time-dependent advection problem of (3.3) is given by

$$\varphi_t + v\varphi_x = 0 \tag{3.7}$$

$$\varphi(0, x) = \varphi_0(x). \tag{3.8}$$

Using the *characteristic method* (see [HR15]) we can solve the equation (3.7) exactly with the solution

$$\begin{aligned} &\varphi(x,t) = \varphi_0(\xi(x,t)), \\ &\text{with} \int_x^{\xi} \frac{1}{v(s)} ds = -t. \end{aligned} \tag{3.9}$$

To prove that this is indeed the solution, we take a closer look at (3.9).
Because v is strictly positive and therefore well defined there exists an V with $V'(s) = \frac{1}{v(s)}$ and V is monotonically increasing and thus invertible. Hence, one obtains

$$\begin{aligned} &V(\xi) - V(x) = -t \\ \Rightarrow &V(\xi) = V(x) - t \\ \Rightarrow &\xi(x,t) = V^{-1}(V(x) - t). \end{aligned}$$

For the derivatives of ξ it follows

$$\xi_t = \left(V^{-1}(V(x)-t)\right)'(-1)$$
$$\xi_x = \left(V^{-1}(V(x)-t)\right)'\frac{1}{v}.$$

For the derivatives of φ it follows

$$\begin{aligned}\varphi_t &= \varphi_0'(\xi(t,x))\xi_t \\ &= \varphi_0'(\xi(t,x))\left(V^{-1}(V(x)-t)\right)'(-1) \\ \varphi_x &= \varphi_0'(\xi(t,x))\xi_x \\ &= \varphi_0'(\xi(t,x))\left(V^{-1}(V(x)-t)\right)'\frac{1}{v}.\end{aligned}$$

Therefore, for (3.3) one gets

$$\varphi_t + v\varphi_x = \varphi_0'(\xi(t,x))\left(V^{-1}(V(x)-t)\right)'(-1) + v\left(\varphi_0'(\xi(t,x))\left(V^{-1}(V(x)-t)\right)'\frac{1}{v}\right) = 0.$$

3.3 Free energy

In the following, we show that for several choices of so-called free energy functions the free energy of the exact solution does not change over time. The notion of a free energy $\mathbb{F}(c)$ will be used in the publication [FFG$^+$ep] to define new discrete schemes for the pure advection problem. The change of the discrete free energy over time in a discrete scheme contains information about the quality of the numerical solution. For example, if the free energy falls over time, we know, that the discrete solution is smeared. If the discrete free energy rises over time, one knows, that the maximum principle can be violated.

The free energy of the one-dimensional pure advection equation with no source term is time-independent. The free energy for (1.1) is defined to be

$$\mathbb{F}(c) = \int_a^b f(c)\,dx. \tag{3.10}$$

The relative free energy of the one-dimensional pure advection equation with no source term is time-independent.

Lemma 10. *If c solves* (1.1) *with $s = 0$, one obtains*

$$\frac{\partial}{\partial t}\mathbb{F}(c) = 0.$$

Proof. Using the partial differential equation (1.1) without the source term, it follows

$$\frac{\partial}{\partial t}\mathbb{F}(c) = \int_a^b \frac{d}{dt} f(c)\, dx = \int_a^b f'(c) c_t\, dx = -\int_a^b f'(c)(cv)_x\, dx,$$

because $c_t = -(cv)_x$ for the time-dependent pure advection equation without the source term. With the substitution

$$\sigma = cv \Leftrightarrow c = \frac{\sigma}{v} \quad \text{which implies} \quad \frac{d\sigma}{dx} = (cv)_x \Leftrightarrow d\sigma = (cv)_x\, dx,$$

we obtain

$$\frac{\partial}{\partial t}\mathbb{F}(c) = -\int_{(cv)(a)}^{(cv)(b)} f'\left(\frac{\sigma}{v}\right) d\sigma = \int_{(cv)(b)}^{(cv)(a)} f'\left(\frac{\sigma}{v}\right) d\sigma.$$

Since the velocity v and the density c are periodic, it follows $(cv)(a) = (cv)(b)$, which in turn implies that the integral vanishes. □

4. Discretization of the time-dependent problem

In this chapter we will take a closer look at the resulting linear systems of the two mentioned schemes, *simple upwind scheme* and *complete flux scheme* for the time-dependent advection problem, when using the discretization that is explained at the beginning. Therefore, we will try to solve the three different regimes that one obtained in the chapter before.
For (1.1) we can solve the following linear system numerically

$$Mc_t + Ac = s \tag{4.1}$$

with a time integration method e.g. Implicit or Explicit Euler. For the Implicit Euler M, A are $n-1 \times n-1$ matrices with n being the number of nodes given. In this case M, A are sparse matrices, where the entries depend on the scheme one is using. M, A also depend on the boundary conditions, whether they are periodic or not.
To solve (1.1), we define an abstract time-dependent scheme for the time-dependent periodic problem for a given $\varphi_{i+\frac{1}{2}} \approx \varphi(x_{i+\frac{1}{2}})$

$$\begin{aligned}
\dot{c}_i(x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}) + \varphi_{i+\frac{1}{2}} - \varphi_{i-\frac{1}{2}} &= s_i\left(x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}\right), \\
\dot{c}_1\left((x_{\frac{3}{2}} - x_1) + (x_n - x_{n-\frac{1}{2}})\right) + \varphi_{\frac{3}{2}} - \varphi_{n-\frac{1}{2}} &= s_1\left(x_{\frac{3}{2}} - x_1\right) + s_n\left(x_n - x_{n-\frac{1}{2}}\right).
\end{aligned}$$

Additionally, we prescribe an initial value $c_i(0) = c_0(x_i)$ for $i = 1, \ldots, n-1$.

4.1 Simple upwind system

For the *finite volume discretization* of (2.6) with a *simple upwind flux* for a time-dependent problem, it holds

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} c_t + (vc)_x \, dx = \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} s_{\text{pc}}(x) \, dx, \qquad i = 1, \ldots n,$$

with $x_{1-\frac{1}{2}} = x_{n-\frac{1}{2}}$, respectively $x_{n+\frac{1}{2}} = x_{1+\frac{1}{2}}$. Therefore, one obtains
$i = 2, \ldots, n-1$.

The continuous balance on the control volume V_i is approximated by the discrete balance

$$
\begin{aligned}
&\dot{c}_i(x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}) + \varphi^{\mathrm{SU}}_{i+\frac{1}{2}} - \varphi^{\mathrm{SU}}_{i-\frac{1}{2}} = s(x_i)(x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}) \\
\Leftrightarrow &\dot{c}_i(x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}) + c_i\frac{v(x_{i+1}) + v(x_i)}{2} - c_{i-1}\frac{v(x_i) + v(x_{i-1})}{2} = s(x_i)(x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}) \\
\Leftrightarrow &\dot{c}_i(x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}) + c_i\frac{v(x_{i+1}) + v(x_i)}{2} - c_{i-1}\frac{v(x_i) + v(x_{i-1})}{2} = s_i(x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}),
\end{aligned}
\tag{4.2}
$$

and, because the problem is periodic, one obtains

$$
\begin{aligned}
&\dot{c}_1\left((x_{\frac{3}{2}} - x_1) + (x_n - x_{n-\frac{1}{2}})\right) + c_1\frac{v(x_2) + v(x_1)}{2} - c_{n-1}\frac{v(x_n) + v(x_{n-1})}{2} \\
=&s(x_1)\left(x_{\frac{3}{2}} - x_1\right) + s(x_n)\left(x_n - x_{n-\frac{1}{2}}\right).
\end{aligned}
$$

Consequently, for the linear system it holds

$$
\begin{bmatrix}
x_{1,1} & & & \\
& \ddots & & \\
& & x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}} & \\
& & & \ddots
\end{bmatrix}
\begin{pmatrix} \vdots \\ \dot{c}_i \\ \vdots \end{pmatrix}
+
\begin{bmatrix}
v_{2,1} & & & & & & -v_{n,n-1} \\
-v_{2,1} & v_{3,2} & & & & & \\
& \ddots & \ddots & & & & \\
& & -v_{i,i-1} & v_{i+1,i} & & & \\
& & & \ddots & \ddots & & \\
& & & & & -v_{n-1,n-2} & v_{n,n-1}
\end{bmatrix}
\begin{pmatrix} c_1 \\ \vdots \\ c_{i-1} \\ c_i \\ \vdots \\ c_{n-1} \end{pmatrix}
$$

$$
= \begin{pmatrix}
s_1\left(x_{\frac{3}{2}} - x_1\right) + s_n\left(x_n - x_{n-\frac{1}{2}}\right) \\
\vdots \\
s_i(x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}) \\
\vdots
\end{pmatrix},
\tag{4.3}
$$

with $v_{i,i-1} = \frac{v(x_i)+v(x_{i-1})}{2}$ and $x_{1,1} = (x_{\frac{3}{2}} - x_1) + (x_n - x_{n-\frac{1}{2}})$.

4.1.1 Mass conservation

We now show that the *simple upwind scheme* that is used to solve te homogeneous equation $c_t + (cv)_x = 0, \quad x \in (a, b)$ is mass conserving. We therefore take a closer look at the following equation:

$$(1,\dots,1)_n\left[M\dot{c}+Ac\right]=0$$

$$(1,\dots,1)_n\begin{pmatrix} x_{1,1}\dot{c}_1 \\ \vdots \\ \left(x_{i+\frac{1}{2}}-x_{i-\frac{1}{2}}\right)\dot{c}_i \\ \vdots \\ \left(x_{n-1+\frac{1}{2}}-x_{n-1-\frac{1}{2}}\right)\dot{c}_{n-1}\end{pmatrix}+(1,\dots,1)_n\begin{pmatrix} v_{2,1}c_1-v_{n,n-1}c_{n-1} \\ -v_{2,1}c_1+v_{3,2}c_2 \\ \vdots \\ -v_{i,i-1}c_{i-1}+v_{i+1,i}c_i \\ \vdots \\ -v_{n-1,n-2}c_{n-2}+v_{n,n-1}c_{n-1}\end{pmatrix}=0.$$

With

$$\begin{aligned} &v_{2,1}c_1-v_{n,n-1}c_{n-1}-v_{2,1}c_1+v_{3,2}c_2+\ldots+ \\ &-v_{n-2,n-3}c_{n-3}+v_{n-1,n-2}c_{n-2}-v_{n-1,n-2}c_{n-2}+v_{n,n-1}c_{n-1}=0, \end{aligned}$$

we obtain

$$x_{1,1}\dot{c}_1+\sum_{i=2}^{n-1}\left(x_{i+\frac{1}{2}}-x_{i-\frac{1}{2}}\right)\dot{c}_i=0.$$

Thus, the mass does not change over time which is equal to mass conservation.

4.1.2 Solution of the three different regimes

The regime (3.4) is equivalent to the steady periodic advection (2.1) and is solvable with the *simple upwind scheme* (2.11).
For the third regime (3.5) we take a closer look at (4.2),

$$\dot{c}_i(x_{i+\frac{1}{2}}-x_{i-\frac{1}{2}})+c_i\frac{v(x_{i+1})+v(x_i)}{2}-c_{i-1}\frac{v(x_i)+v(x_{i-1})}{2}=s_i(x_{i+\frac{1}{2}}-x_{i-\frac{1}{2}}).$$

For a constant $\hat{c}$ and

$$\begin{aligned} c_i &:= \frac{\hat{c}t}{\frac{v(x_{i+1})+v(x_i)}{2}}=\frac{2\hat{c}t}{v(x_{i+1})+v(x_i)} \Leftrightarrow \dot{c}_i=\frac{\hat{c}}{\frac{v(x_{i+1})+v(x_i)}{2}}=\frac{2\hat{c}}{v(x_{i+1})+v(x_i)} \quad \text{and} \\ s_i &:= \frac{\hat{c}}{\frac{v(x_{i+1})+v(x_i)}{2}}=\frac{2\hat{c}}{v(x_{i+1})+v(x_i)}, \end{aligned} \tag{4.4}$$

it holds

$$
\begin{aligned}
&\frac{2\hat{c}}{v(x_{i+1})+v(x_i)}\left(x_{i+\frac{1}{2}}-x_{i-\frac{1}{2}}\right)+\frac{2\hat{c}t}{v(x_{i+1})+v(x_i)}\frac{v(x_{i+1})+v(x_i)}{2}-\frac{2\hat{c}t}{v(x_i)+v(x_{i-1})}\frac{v(x_i)+v(x_{i-1})}{2}\\
=&\frac{2\hat{c}}{v(x_{i+1})+v(x_i)}\left(x_{i+\frac{1}{2}}-x_{i-\frac{1}{2}}\right)\\
\Leftrightarrow&\frac{2\hat{c}}{v(x_{i+1})+v(x_i)}\left(x_{i+\frac{1}{2}}-x_{i-\frac{1}{2}}\right)+\frac{2\hat{c}t}{2}-\frac{2\hat{c}t}{2}=\frac{2\hat{c}}{v(x_{i+1})+v(x_i)}\left(x_{i+\frac{1}{2}}-x_{i-\frac{1}{2}}\right)\\
\Leftrightarrow&\frac{2\hat{c}}{v(x_{i+1})+v(x_i)}\left(x_{i+\frac{1}{2}}-x_{i-\frac{1}{2}}\right)=\frac{2\hat{c}}{v(x_{i+1})+v(x_i)}\left(x_{i+\frac{1}{2}}-x_{i-\frac{1}{2}}\right).
\end{aligned}
$$

Therefore, (4.4) indeed solves the third regime numerically with the *simple upwind scheme*. Furthermore, the discrete solution c_i converges to the continuous solution (3.6) with first order convergence; $c_i = \frac{\hat{c}t}{\frac{v(x_{i+1})+v(x_i)}{2}} \to c^{con}(x_i) = \frac{\hat{c}t}{v(x_i)}$.
The discretization of the first regime is equivalent to the homogeneous linear system that belongs to (4.1).

4.1.3 Homogeneous linear system

We can rewrite the system $M\dot{c} + Ac = 0$.
Using the substitution $\varrho = Mc \Leftrightarrow c = M^{-1}\varrho$ one obtains

$$
\begin{aligned}
&\dot{\varrho} + AM^{-1}\varrho = 0\\
\Rightarrow\ &\dot{\varrho} + SVM^{-1}\varrho = 0,
\end{aligned}
$$

where

$$
S = \begin{bmatrix} 1 & 0 & \cdots & 0 & -1 \\ -1 & 1 & 0 & & 0 \\ 0 & -1 & 1 & 0 & \vdots \\ & & \ddots & \ddots & 0 \\ & & & -1 & 1 \end{bmatrix}
$$

and

$$
V = \begin{bmatrix} \ddots & & \\ & v_{i,i-1} & \\ & & \ddots \end{bmatrix}.
$$

One obtains

$$
\dot{\varrho} + K\varrho = 0 \tag{4.5}
$$

with $K = SVM^{-1}$. To solve this linear system we need to calculate the eigenvalues and eigenvectors of K.

To get a better understanding of this eigenvalue problem we take a closer look at the problem where $\frac{v(x_i)+v(x_{i-1})}{2} = \hat{v} = constant$, with a constant $v \in \mathbb{C}$, and a structured grid with $\frac{x_i - x_{i-1}}{2} = \hat{m}$, for all $i = 1, \ldots, n-1$. Thus one obtains

$$K = \begin{bmatrix} \frac{\hat{v}}{\hat{m}} & 0 & \cdots & 0 & -\frac{\hat{v}}{\hat{m}} \\ -\frac{\hat{v}}{\hat{m}} & \frac{\hat{v}}{\hat{m}} & 0 & \vdots & \\ & \ddots & \ddots & & 0 \\ & & & -\frac{\hat{v}}{\hat{m}} & \frac{\hat{v}}{\hat{m}} \end{bmatrix}.$$

Lemma 11. *For a matrix of the form*

$$K = \begin{bmatrix} \frac{\hat{v}}{\hat{m}} & 0 & \cdots & 0 & -\frac{\hat{v}}{\hat{m}} \\ -\frac{\hat{v}}{\hat{m}} & \frac{\hat{v}}{\hat{m}} & 0 & \vdots & \\ & \ddots & \ddots & & 0 \\ & & & -\frac{\hat{v}}{\hat{m}} & \frac{\hat{v}}{\hat{m}} \end{bmatrix}_{n\times n}$$

the characteristic polynomial is given as $\chi_K(\lambda) = \left(\frac{\hat{v}}{\hat{m}} - \lambda\right)^n - \left(\frac{\hat{v}}{\hat{m}}\right)^n$.

Proof. The characteristic polynomial is calculated by $\det(K - \lambda E_n)$, where E_n is the identity matrix with the dimension n. Calculating this, using the Laplace expansion for the determinant, one obtains

$$\det\left(\begin{bmatrix} (\frac{\hat{v}}{\hat{m}} - \lambda) & 0 & \cdots & 0 & -\frac{\hat{v}}{\hat{m}} \\ -\frac{\hat{v}}{\hat{m}} & (\frac{\hat{v}}{\hat{m}} - \lambda) & & & 0 \\ 0 & -1 & (1-\lambda) & 0 & \vdots \\ & & \ddots & \ddots & 0 \\ & & & -\frac{\hat{v}}{\hat{m}} & (\frac{\hat{v}}{\hat{m}} - \lambda) \end{bmatrix}_{n\times n}\right)$$

$$= (\frac{\hat{v}}{\hat{m}} - \lambda)(-1)^2 \det \left(\begin{bmatrix} (\frac{\hat{v}}{\hat{m}} - \lambda) & 0 & \cdots & & 0 \\ -\frac{\hat{v}}{\hat{m}} & (\frac{\hat{v}}{\hat{m}} - \lambda) & & & 0 \\ 0 & -\frac{\hat{v}}{\hat{m}} & (\frac{\hat{v}}{\hat{m}} - \lambda) & 0 & \vdots \\ & & \ddots & \ddots & 0 \\ & & & -\frac{\hat{v}}{\hat{m}} & (\frac{\hat{v}}{\hat{m}} - \lambda) \end{bmatrix}_{n-1 \times n-1} \right)$$

$$+ (-\frac{\hat{v}}{\hat{m}})(-1)^{n+1} \det \left(\begin{bmatrix} -\frac{\hat{v}}{\hat{m}} & (\frac{\hat{v}}{\hat{m}} - \lambda) & 0 & \cdots & 0 \\ 0 & -\frac{\hat{v}}{\hat{m}} & (\frac{\hat{v}}{\hat{m}} - \lambda) & 0 & \vdots \\ & & \ddots & \ddots & 0 \\ & & & -\frac{\hat{v}}{\hat{m}} & (\frac{\hat{v}}{\hat{m}} - \lambda) \\ & & & & -\frac{\hat{v}}{\hat{m}} \end{bmatrix}_{n-1 \times n-1} \right).$$

On the right hand side both determinants can be calculated because both matrices are in the form of an upper respectively lower triangular matrix. Therefore, it holds for the characteristic polynomial

$$\chi_K(\lambda) = \left(\frac{\hat{v}}{\hat{m}} - \lambda\right)\left(\frac{\hat{v}}{\hat{m}} - \lambda\right)^{n-1} + \left(-\frac{\hat{v}}{\hat{m}}\right)(-1)^{n+1}\left(-\frac{\hat{v}}{\hat{m}}\right)^{n-1} = \left(\frac{\hat{v}}{\hat{m}} - \lambda\right)^{n} + \left(-\frac{\hat{v}}{\hat{m}}\right)^{n}(-1)^{n+1}.$$

Since $v(x) > 0, \quad \forall x \in [a, b]$, it follows $\hat{v} > 0$ and since the nodes are in order, one also gets $\hat{m} > 0$. For the characteristic polynomial, one then gets

$$\chi_K(\lambda) = \left(\frac{\hat{v}}{\hat{m}} - \lambda\right)^{n} - \left(\frac{\hat{v}}{\hat{m}}\right)^{n}.$$

□

Lemma 12. *Let the matrix K be of the form*

$$K = \begin{bmatrix} \frac{\hat{v}}{\hat{m}} & 0 & \cdots & 0 & -\frac{\hat{v}}{\hat{m}} \\ -\frac{\hat{v}}{\hat{m}} & \frac{\hat{v}}{\hat{m}} & 0 & \vdots & \\ & \ddots & \ddots & & 0 \\ & & & -\frac{\hat{v}}{\hat{m}} & \frac{\hat{v}}{\hat{m}} \end{bmatrix}_{n \times n}.$$

Then K has n eigenvalues λ_j and n orthogonal eigenvectors v_j

$$\lambda_j = \frac{\hat{v}}{\hat{m}}\left(1 - e^{i\frac{2\pi j}{n}}\right), \quad j = 0, 1, 2, \cdots, n-1$$

$$v_j = \begin{bmatrix} \rho_j^{n-1} \\ \rho_j^{n-2} \\ \vdots \\ \rho_j^2 \\ \rho_j \\ 1 \end{bmatrix}, \qquad \textit{with} \quad \rho = e^{i2\pi j/n}, \qquad j = 0, 1, 2, \cdots, n-1.$$

Proof. The eigenvalues of K correspond to the roots of the characteristic polynomial $\chi_K(\lambda) = \left(\frac{\hat{v}}{\hat{m}} - \lambda\right)^n - \left(\frac{\hat{v}}{\hat{m}}\right)^n$,

$$\begin{aligned} 0 &= \left(\frac{\hat{v}}{\hat{m}} - \lambda\right)^n - \left(\frac{\hat{v}}{\hat{m}}\right)^n \\ \Rightarrow \left(1 - \frac{\lambda}{\frac{\hat{v}}{\hat{m}}}\right)^n &= 1. \end{aligned}$$

We then use the substitution $z = \left(1 - \frac{\lambda}{\frac{\hat{v}}{\hat{m}}}\right)$ and obtain

$z^n = 1$ and because the nth roots of 1 are $\rho_j = e^{i\frac{2\pi j}{n}}, \quad j = 0, 1, 2, \cdots, n-1$, see [Tee07],

it follows

$$\begin{aligned} z_j &= \left(1 - \frac{\lambda_j}{\frac{\hat{v}}{\hat{m}}}\right) = e^{i\frac{2\pi j}{n}} \\ \Rightarrow \left(1 - e^{i\frac{2\pi j}{n}}\right) &= \frac{\lambda_j}{\frac{\hat{v}}{\hat{m}}} \\ \Rightarrow \lambda_j &= \frac{\hat{v}}{\hat{m}}\left(1 - e^{i\frac{2\pi j}{n}}\right) \\ \Rightarrow \lambda_j &= \frac{\hat{v}}{\hat{m}}\left(1 - \rho_j\right), \quad j = 0, 1, 2, \cdots, n-1. \end{aligned}$$

For the eigenvectors v_j, it must hold

$$Kv_j = \lambda_j v_j$$

$$\begin{bmatrix} \frac{\hat{v}}{\hat{m}} & 0 & \cdots & 0 & -\frac{\hat{v}}{\hat{m}} \\ -\frac{\hat{v}}{\hat{m}} & \frac{\hat{v}}{\hat{m}} & 0 & \vdots & \\ & \ddots & \ddots & & 0 \\ & & & -\frac{\hat{v}}{\hat{m}} & \frac{\hat{v}}{\hat{m}} \end{bmatrix} \begin{bmatrix} \rho_j^{n-1} \\ \rho_j^{n-2} \\ \vdots \\ \rho_j^2 \\ \rho_j \\ 1 \end{bmatrix} = \frac{\hat{v}}{\hat{m}}(1-\rho_j) \begin{bmatrix} \rho_j^{n-1} \\ \rho_j^{n-2} \\ \vdots \\ \rho_j^2 \\ \rho_j \\ 1 \end{bmatrix}$$

$$\Leftrightarrow \begin{bmatrix} \frac{\hat{v}}{\hat{m}}\rho_j^{n-1} - \frac{\hat{v}}{\hat{m}} \\ \frac{\hat{v}}{\hat{m}}\rho_j^{n-2} - \frac{\hat{v}}{\hat{m}}\rho_j^{n-1} \\ \vdots \\ \frac{\hat{v}}{\hat{m}}\rho_j^1 - \frac{\hat{v}}{\hat{m}}\rho_j^2 \\ \frac{\hat{v}}{\hat{m}} - \frac{\hat{v}}{\hat{m}}\rho_j \end{bmatrix} = \begin{bmatrix} \frac{\hat{v}}{\hat{m}}\rho_j^{n-1} - \frac{\hat{v}}{\hat{m}}\rho_j^n \\ \frac{\hat{v}}{\hat{m}}\rho_j^{n-2} - \frac{\hat{v}}{\hat{m}}\rho_j^{n-1} \\ \vdots \\ \frac{\hat{v}}{\hat{m}}\rho_j^1 - \frac{\hat{v}}{\hat{m}}\rho_j^2 \\ \frac{\hat{v}}{\hat{m}} - \frac{\hat{v}}{\hat{m}}\rho_j \end{bmatrix}.$$

□

For the eigenvectors and eigenvalues of K one obtains the following equation: $Kv_j = \lambda_j v_j$.
To solve (4.5), we now use the approach $\varrho_i(t) = e^{-\lambda_i t} v_i, \quad i = 1, \cdots, n-1$.
It follows

$$\dot{\varrho}_i = -\lambda_i e^{-\lambda_i t} v_i = -\lambda_i \varrho_i$$

and

$$K\varrho_i = K(e^{-\lambda_i t} v_i) = (e^{-\lambda_i t})Kv_i = e^{-\lambda_i t}\lambda_i v_i = \lambda_i \varrho_i.$$

Therefore, one gets

$$\dot{\varrho}_i + K\rho_i = -\lambda_i \varrho_i + \lambda_i \varrho_i = 0.$$

For the whole system one obtains

$$\varrho(t) = \sum_1^{n-1} \alpha_i e^{-\lambda_i t} v_i,$$

with $\rho(0) = \sum_1^{n-1} \alpha_i v_i$.
For the time derivative of ρ it follows

$$\dot{\varrho}(t) = \sum_1^{n-1} \alpha_i (-\lambda_i) e^{\lambda_i t} v_i,$$

since the eigenvector space is a basis of the vector space.

As a result, one gets

$$K\varrho = K(\sum_1^{n-1} \alpha_i e^{-\lambda_i t} v_i) = \sum_1^{n-1} \alpha_i e^{-\lambda_i t} K v_i = \sum_1^{n-1} \alpha_i (\lambda_i) e^{\lambda_i t} v_i$$
$$\Rightarrow \dot{\varrho} + K\varrho = 0.$$

4.1.4 Positive stability

Using the *implicit Euler method* to solve the homogeneous PDE $M\dot{c}+Ac = 0$ one obtains

$$\left(\frac{1}{\Delta t} M + A\right) c^{n+1} = c^n, \tag{4.6}$$

with Δt being the time step. Let E be $E = \left(\frac{1}{\Delta t} M + A\right)$.

Definition 1. *A matrix $A \in \mathbb{R}_{n\times n}$ is called positive stable, if it has only eigenvalues with positive real part. That is equivalent to $v^T Av \geq 0 \quad v \in, \mathbb{R}_n$.*

Remark 2. *The matrix $A \in \mathbb{R}_{n\times n}$ is positive stable, if and only if $A + A^T$ is positive definite, see [HJ94].*

From this remark we recognize at once that the matrix E from above, built from the *simple upwind scheme*, is positive stable.

Definition 2. *We define the set $Z_n \in \mathbb{R}_{n\times n}$ by*

$$Z_n := \{A = [a_{i,j}] \in \mathbb{R}_{n\times n} : a_{i,j} \leq 0 \quad \textit{if} \quad i \neq j, \quad i,j = 1, \ldots n\}.$$

Definition 3. *A matrix is called M-matrix if $A \in Z_n$ and A is positive stable.*

Note that the matrix K is therefore a M-matrix.
We now want to derive some important, advantageous properties of M-matrices.

Remark 3. *The simple sign pattern of matrices in Z_n has many striking consequences. First, it is easy to see that $A = [a_{i,j}] \in Z_n$ if and only if $A \in \mathbb{R}_{n\times n}$ and $A = \alpha I - P$ for some $P \in \mathbb{R}_{n\times n}$ with $P \geq 0$ (every matrix entry of P is non-negative) and some $\alpha \in \mathbb{R}$ (even $\alpha > 0$). One can take any $\alpha \geq max\{a_{i,i} : i = 1, \ldots, n\}$. This representation is important since we will soon use the Perron-Frobenius theory of non-negative matrices.*

The following Lemma is very useful in the study of M-matrices:

Lemma 13. *Let $A = [a_{i,j}] \in Z_n$ and suppose that $A = \alpha I - P$ with $\alpha \in \mathbb{R}$ and $P \geq 0$. Then, $\alpha - \rho(P)$ is an eigenvalue of A, every eigenvalue of A lies in the disc$\{z \in \mathbb{C} : |z - \alpha| \leq \rho(P)\}$, and hence every eigenvalue of A satisfies $Re(\lambda) \geq \alpha - \rho(P)$. In particular A is an M-matrix if and only if $\alpha > \rho(P)$. If A is indeed a M-matrix, one may always write $A = \gamma I - P$ with $\gamma max\{a_{i,i} : i = 1, \ldots, n\}, P = \gamma I - A \geq 0$; necessarily, we have $\gamma > \rho(P)$.*

Proof. For the eigenvalues of A holds

$$0 = det(\lambda_A I - A) = det(\lambda_A I - \alpha I + P) = (-1)^{n-1} det((\alpha - \lambda_A)I - P).$$

Therefore, it holds $\lambda_P = \alpha - \lambda_A$, respectively $\lambda_A = \alpha - \lambda_P$, where λ_P is an eigenvalue of P. Thus, every eigenvalue of A lies in a disc on the complex plane with radius $\rho(P)$, which is centered at $z = \alpha$. Especially we have $Re(\lambda_A) \geq \alpha - \rho(P)$. Since $\rho(P)$ is an eigenvalue of P due to the Perron-Frobenius theorem, $\alpha - \rho(P)$ is indeed a real eigenvalue of A. The Perron-Frobenius theorem can be applied to P because one can choose α such that $\alpha < \max_{i=1,\dots,n-1} a_{i,i}$ and therefore P is at least a matrix with a full diagonal. If A is a M-matrix it is positive stable, hence $\alpha - \rho(P) > 0$. Conversely, if $\alpha > \rho(P)$ then the disc $\{z \in \mathbb{C} : |z - \alpha| \leq \rho(P)\}$ lies in the right half plane, so A is positive stable. □

Theorem 4.1.1 (Perron-Frobenius theorem)**.** *[BTa92] Let A be a $n \times n$ non-negative and irreducible matrix. Then, there exists a simple positive eigenvalue λ of A which has an associated positive eigenvector (i.e, all of whose coordinates are positive), and which has the highest value among the moduli of the other eigenvalues of A.*

The most important property of M-matrices is that their inverses are non-negative.

Lemma 14. *For a M-matrix A it holds that*

$$A^{-1} \geq 0.$$

Proof. We assume the representation

$$A = \gamma I - P$$

for A with $\gamma > \rho(P)$.
Then, the inverse of A is

$$\begin{aligned} A^{-1} &= \frac{1}{\gamma}\left(I - \frac{1}{\gamma}P\right)^{-1} \\ &= \frac{1}{\gamma}\left(I + \frac{1}{\gamma}P + \frac{1}{\gamma^2}P^2 + \dots\right). \end{aligned}$$

Here one used the Neumann series, see also [Wer00].
Since $P \geq 0$, the inverse of A is also non-negative. □

As a result, it follows that the inverse of E exists and is non-negative.

So one can solve the equation (4.6):

$$\begin{aligned} &\left(\frac{1}{\Delta t}M + A\right)c^{n+1} = c^n \\ \Leftrightarrow &Ec^{n+1} = c^n \\ \Leftrightarrow &c^{n+1} = E^{-1}c^n. \end{aligned}$$

Remark 4. *Let c be the numerically solution, which one obtains when using the simple upwind discretization. Then, it holds $c(t) \geq 0$ for all times t, if the start value $c(0) \geq 0$.*

4.1.5 Artifical diffusion

When using the *simple upwind scheme* to solve (3.1) it shows that, even though we solve a pure advection equation, the problem diffuses over time. Hence we have an artificial diffusion. We take a closer look at (4.2). With the grid being structured and v being constant, we can bracket the following equation:

$$\begin{aligned} c_i - c_{i-1} &= c_i - c_{i-1} + \left(\frac{1}{2}c_{i+1} - \frac{1}{2}c_{i-1}\right) - \left(\frac{1}{2}c_{i+1} - \frac{1}{2}c_{i-1}\right) \\ &= \frac{1}{2}c_{i+1} - \frac{1}{2}c_{i-1} + \left(c_i - \frac{1}{2}c_{i+1} - \frac{1}{2}c_{i-1}\right). \end{aligned}$$

The first part of this equation $\frac{1}{2}c_{i+1} - \frac{1}{2}c_{i-1}$ is the central difference approximation, which is another form to approximate the flux at a certain point, see also [Kum17].
The second part is a diffusion term. Thus, the *simple upwind scheme* can be rewritten as the central difference approximation and a diffusion term (see (4.9)).
The pure diffusion equation $-c_{xx} = 0$ can be written as

$$\Phi_x = 0 \quad \text{with} \quad \Phi = -c_x. \tag{4.7}$$

Solving this with a *finite volume discretization*, one obtains

$$\begin{aligned} &\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \Phi_x \, dx \\ =&\Phi_{i+\frac{1}{2}} - \Phi_{i+\frac{1}{2}} \\ =&-c_x(x_{i+\frac{1}{2}}) - \left(-c_x(x_{i-\frac{1}{2}})\right) \\ \approx&\left(-\frac{c_{i+1} - c_i}{h}\right) - \left(-\frac{c_i - c_{i-1}}{h}\right) \qquad (4.8) \\ =&\frac{1}{h}\left(2c_i - c_{i+1} - c_{i-1}\right), \qquad (4.9) \end{aligned}$$

with h being the mesh size. The approximation in (4.8) comes from an approximation of the derivative.
Thus, one obtains an artifical diffusion when using the *simple upwind scheme*.

4.2 Simple upwind and discrete free energy

This chapter discusses the change of the free energy over time for the *simple upwind system*. For further information about the free energy, see [FFG⁺ep].

4.2.1 Simple upwind system with $v = 1$ and $s = 0$

The discrete *simple upwind system* for $v = 1$ and $s = 0$ is given by

$$M\dot{c} + Ac = 0. \tag{4.10}$$

It is assumed, that the grid has $n + 1$ nodes. Due to the periodic boundary conditions, we then have n degrees of freedom.
Hence, the matrix A is given by

$$A = \begin{bmatrix} 1 & 0 & \cdots & 0 & -1 \\ -1 & 1 & 0 & & 0 \\ 0 & -1 & 1 & 0 & \vdots \\ & & \ddots & \ddots & 0 \\ & & & -1 & 1 \end{bmatrix}. \tag{4.11}$$

For an arbitrary convex function $f : \mathbb{R} \to \mathbb{R}$, which is assumed to model some kind of physically given *free energy function*, the *total discrete free energy* is given for all $t \in [0, T]$ during the discrete time evolution by

$$F(t) = \sum_{i=1}^{n} m_i f(c_i(t)),$$

with m_i being the i-th diagonal entry of M. It will be shown that the total discrete free energy decays monotonously in time, if f is a convex function. In order to prove this, one computes

$$\begin{aligned}\frac{d}{dt}F(t) &= \sum_{i=1}^{n} m_i f'(c_i(t))\dot{c}_i(t)\\ &= -(f'(c_1(t)), f'(c_2(t)), \dots, f'(c_n(t)))\begin{pmatrix} c_1 - c_n \\ c_2 - c_1 \\ \vdots \\ c_n - c_{n-1}\end{pmatrix}\\ &= \sum_{i=1}^{n} -f'(c_i(t))(c_i(t) - c_{i-}(t))\end{aligned}$$

where we substituted the expression $m_i\dot{c}_i$ using (4.10) and (4.11), and where we define $c_0(t) := c_n(t)$.
Now we make a Taylor expansion for all $i = 1, \dots, n$

$$f(c_{i-1}) = f(c_i) + f'(c_i)(c_{i-1} - c_i) + \frac{1}{2}f''(\bar{c}_i)(c_{i-1} - c_i)^2$$

with $\bar{c}_i \in]c_i, c_{i-1}[$.
Therefore, one derives for all $i = 1, \dots, n$,

$$-f'(c_i)(c_{i-1} - c_i) = f(c_{i-1}) - f(c_i) - \frac{1}{2}f''(\bar{c}_i)(c_{i-1} - c_i)^2.$$

Using this, one gets

$$\begin{aligned}\frac{d}{dt}F(t) &= \sum_{i=1}^{n} -f'(c_i(t))(c_i(t) - c_{i-}(t))\\ &= \sum_{i=1}^{n} f(c_{i-1}(t)) - f(c_i(t)) - \frac{1}{2}f''(\bar{c}_i(t))(c_{i-1}(t) - c_i(t))^2.\end{aligned}$$

The first sum vanishes due to $c_0 = c_n$ and the second sum is element-wise non-positive, since $f''(\bar{c}_i) > 0$ due to convexity of the free energy function f.
Thus, the total discrete free energy decays monotonously in time.

4.2.2 Simple upwind with $v \neq \text{const}$ and $s = 0$

In the general case (with $v > 0$) we have to slightly change the argument. Now the discrete time evolution is given by

$$M\dot{c} + AVc = 0,$$

where V is a diagonal matrix with diagonal entries $\frac{v(x_i)+v(x_{i-1})}{2}$ due to the *simple upwind flux* in the general case.

We introduce $\Phi := Vc$ and get the new time evolution for the discrete flux Φ:

$$MV^{-1}\Phi + A\Phi = 0.$$

The total discrete free energy in the new discrete flux variables is then given by

$$F(t) = \sum_{i=1}^{n} m_i f\left(\frac{\Phi_i}{v_i}\right).$$

Taking the time derivative as above, one gets

$$\begin{aligned}\frac{d}{dt}F(t) &= \sum_{i=1}^{n} \frac{m_i}{v_i} f'\left(\frac{\Phi_i}{v_i}\right)\dot{\Phi}_i \\ &= \sum_{i=1}^{n} -f'\left(\frac{\Phi_i}{v_i}\right)(\Phi_i - \Phi_{i-1}).\end{aligned}$$

Now the argument from above can be repeated. Then, the total discrete free energy function will decay monotonously in time, if the new function $g(\Phi) := f(\Phi/v)$ is convex. More precisely, one has to assume convexity of the function $f(\Phi/v_h)$, where v_h is the continuous, piecewise linear interpolation at the Voronoi faces $x_{i+\frac{1}{2}}$.

4.3 Complete flux system

For the time-dependent problem the *complete flux scheme* is defined as

$$\varphi_{i+\frac{1}{2}}^{\mathrm{CFS_t}} := c_i v(x_i) + s_i(x_{i+\frac{1}{2}} - x_i) - \dot{c}_i(x_{i+\frac{1}{2}} - x_i). \tag{4.12}$$

For the *finite volume discretization* of (2.6) with the *complete flux scheme* (4.12) it holds

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} c_t + (vc)_x \, dx = \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} s_{\mathrm{pc}}(x) \, dx, \qquad i = 1, \ldots n,$$

with $x_{1-\frac{1}{2}} = x_{n-\frac{1}{2}}$, respectively $x_{n+\frac{1}{2}} = x_{1+\frac{1}{2}}$.
Therefore, one obtains $i = 2, \ldots, n-1$.

The continuous balance on the control volume V_i is approximated by the discrete balance

$$
\begin{aligned}
&\dot{c}_i(x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}) + \varphi^{\mathrm{CFS_t}}_{i+\frac{1}{2}} - \varphi^{\mathrm{CFS_t}}_{i-\frac{1}{2}} = s(x_i)(x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}) \\
&\Leftrightarrow \dot{c}_i(x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}) + c_i v(x_i) + s_i(x_{i+\frac{1}{2}} - x_i) - \dot{c}_i(x_{i+\frac{1}{2}} - x_i) \\
&- (c_{i-1} v(x_{i-1}) + s_{i-1}(x_{i-\frac{1}{2}} - x_{i-1}) - \dot{c}_{i-1}(x_{i-\frac{1}{2}} - x_{i-1})) \\
&= s_i(x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}) \\
&\Leftrightarrow \dot{c}_i \left(\frac{x_i - x_{i-1}}{2}\right) + \dot{c}_{i-1} \left(\frac{x_i - x_{i-1}}{2}\right) + c_i v(x_i) - c_{i-1} v(x_{i-1}) \\
&= s_i \left(\frac{x_i - x_{i-1}}{2}\right) + s_{i-1} \left(\frac{x_i - x_{i-1}}{2}\right), \qquad (4.13)
\end{aligned}
$$

and because the problem is periodic one also gets

$$
\begin{aligned}
&\dot{c}_1 \left(\frac{x_n - x_{n-1}}{2}\right) + \dot{c}_{n-1} \left(\frac{x_n - x_{n-1}}{2}\right) + c_1 v(x_1) - c_{n-1} v(x_{n-1}) \\
&= s_n \left(\frac{x_n - x_{n-1}}{2}\right) + s_{n-1} \left(\frac{x_n - x_{n-1}}{2}\right). \qquad (4.14)
\end{aligned}
$$

Consequently, for the linear system it holds

$$
M \begin{pmatrix} \vdots \\ \dot{c}_i \\ \vdots \end{pmatrix} + A * \begin{pmatrix} \vdots \\ c_i \\ \vdots \end{pmatrix} = \tilde{s}, \qquad (4.15)
$$

with $\tilde{s} = \begin{pmatrix} \vdots \\ \tilde{s}_i \\ \vdots \end{pmatrix}$

and

$$
\tilde{s}_i = s_i \left(\frac{x_i - x_{i-1}}{2}\right) + s_{i-1} \left(\frac{x_i - x_{i-1}}{2}\right).
$$

For A it holds

$$
A = \begin{bmatrix}
a_{1,1} & & & a_{1,n-1} \\
a_{2,1} & a_{2,2} & & \\
& \ddots & \ddots & \\
& & a_{n-1,n-2} & a_{n-1,n-1}
\end{bmatrix}
$$

with

$$
a_{i,i-1} = -v(x_{i-1}),
$$

$$a_{i,i} = v(x_i),$$

and for M it holds

$$M = \begin{bmatrix} m_{1,1} & 0 & \dots & 0 & m_{n-1,n-1} \\ & \ddots & & & \\ & m_{i,i} & m_{i,i} & & \\ & & & \ddots & \\ & & & & \end{bmatrix}$$

with

$$m_{i,i} = \left(\frac{x_i - x_{i-1}}{2}\right)$$

for $i = 2, \dots, n-1$ and for $i = 1$ one gets (4.14).

For further discussions about the solvability of the system we will examine an example.

Example 1. *Let $I = (0, 1)$ be the intervall with $n = 5$ nodes given. The uniform grid is then given by $x_1 = 0 < \frac{1}{4} < \frac{2}{4} < \frac{3}{4} < 1 = x_n$.*
For the mass matrix M one obtains

$$M = \frac{1}{8} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

For the vector $v_l = \begin{pmatrix} -1 \\ 1 \\ -1 \\ 1 \end{pmatrix}$ and $\lambda_l = 0$, it holds

$$Mv_l = \lambda_l v_l = 0.$$

Therefore, λ_l is an eigenvalue of M and v_l the corresponding eigenvector.

Remark 5. *Let the source term s be zero. If the number of grid nodes n is odd, we obtain the mass matrix $M \in \mathbb{R}^{n-1 \times n-1}$. For this mass matrix, we get an eigenvalue $\lambda = 0$. Therefore, for certain start values c_0 it holds*

$$Mc_0 + Ac_0 = 0$$
$$Ac_0 = 0.$$

So, this start value is no solution of the problem and the system with start value c_0 is not

numerically solvable.
Hence, for an odd number of grid nodes, the start value has to be in the vector space, which is spanned by all the eigenvectors of M without the eigenvector v_l. Here v_l is the eigenvector, which corresponds to the eigenvalue $\lambda_l = 0$.

4.3.1 Mass conservation

We now show that the *complete flux scheme* that is used to solve the equation $c_t + (cv)_x = 0, \quad x \in (a,b)$ is mass conserving. We therefore take a closer look at the following equation:

$$(1,\dots,1)_n\,[M\dot{c} + Ac] = 0$$

$$(1,\dots,1)_n \begin{pmatrix} m_{1,1}\dot{c}_1 + m_{n-1,n-1}\dot{c}_{n-1} \\ \vdots \\ m_{i,i}\dot{c}_{i-1} + m_{i,i}\dot{c}_i \\ \vdots \\ m_{n-1,n-1}\dot{c}_{n-2} + m_{n-1,n-1}\dot{c}_{n-1} \end{pmatrix} + (1,\dots,1)_n \begin{pmatrix} v(x_1)c_1 - v(x_{n-1})c_{n-1} \\ -v(x_1)c_1 + v(x_2)c_2 \\ \vdots \\ -v(x_{i-1})c_{i-1} + v(x_i)c_i \\ \vdots \\ -v(x_{n-2})c_{n-2} + v(x_{n-1})c_{n-1} \end{pmatrix} = 0.$$

With

$$\begin{aligned} &v(x_1)c_1 - v(x_{n-1})c_{n-1} - v(x_1)c_1 + v(x_2)c_2 + \ldots + \\ &- v(x_{n-3})c_{n-3} + v(x_{n-2})c_{n-2} - v(x_{n-2})c_{n-2} + v(x_{n-1})c_{n-1} = 0, \end{aligned}$$

we obtain

$$m_{1,1}\dot{c}_1 + m_{n-1,n-1}\dot{c}_{n-1} \sum_{i=2}^{n-1} m_{i,i}\left(\dot{c}_{i-1} + \dot{c}_i\right) = 0.$$

Thus, the mass does not change over time which is equal to mass conservation.

4.3.2 Solution for the three different regimes

The regime (3.4) is equivalent to the steady periodic advection (2.1) and hence is solvable with the *complete flux scheme* (2.14).
For the third regime (3.5) we take a closer look at (4.13),

$$\dot{c}_i\left(\frac{x_i - x_{i-1}}{2}\right) + \dot{c}_{i-1}\left(\frac{x_i - x_{i-1}}{2}\right) + c_i v(x_i) - c_{i-1}v(x_{i-1}) = s_i\left(\frac{x_i - x_{i-1}}{2}\right) + s_{i-1}\left(\frac{x_i - x_{i-1}}{2}\right).$$

For a constant $\hat{c}$ and

$$\begin{aligned} c_i &:= \frac{\hat{c}t}{v(x_i)} \Leftrightarrow \dot{c}_i = \frac{\hat{c}}{v(x_i)} \quad \text{and} \\ s_i &:= \frac{\hat{c}}{v(x_i)}, \end{aligned} \tag{4.16}$$

it holds

$$\begin{aligned} &\frac{\hat{c}}{v(x_i)}\left(\frac{x_i - x_{i-1}}{2}\right) + \frac{\hat{c}}{v(x_{i-1})}\left(\frac{x_i - x_{i-1}}{2}\right) + \frac{\hat{c}t}{v(x_i)}v(x_i) - \frac{\hat{c}t}{v(x_{i-1})}v(x_{i-1}) \\ =&\frac{\hat{c}}{v(x_i)}\left(\frac{x_i - x_{i-1}}{2}\right) + \frac{\hat{c}}{v(x_{i-1})}\left(\frac{x_i - x_{i-1}}{2}\right) \\ \Leftrightarrow&\frac{\hat{c}}{v(x_i)}\left(\frac{x_i - x_{i-1}}{2}\right) + \frac{\hat{c}}{v(x_{i-1})}\left(\frac{x_i - x_{i-1}}{2}\right) + \hat{c}t - \hat{c}t = \frac{\hat{c}}{v(x_i)}\left(\frac{x_i - x_{i-1}}{2}\right) + \frac{\hat{c}}{v(x_{i-1})}\left(\frac{x_i - x_{i-1}}{2}\right) \\ \Leftrightarrow&\frac{\hat{c}}{v(x_i)}\left(\frac{x_i - x_{i-1}}{2}\right) + \frac{\hat{c}}{v(x_{i-1})}\left(\frac{x_i - x_{i-1}}{2}\right) = \frac{\hat{c}}{v(x_i)}\left(\frac{x_i - x_{i-1}}{2}\right) + \frac{\hat{c}}{v(x_{i-1})}\left(\frac{x_i - x_{i-1}}{2}\right). \end{aligned}$$

Therefore, (4.16) indeed solves the third regime numerically with the *simple upwind scheme*. Furthermore, the discrete solution c_i approximates the continuous solution (3.6) exactly at the nodes; $c_i = \frac{\hat{c}t}{v(x_i)} = c^{con}(x_i) = \frac{\hat{c}t}{v(x_i)}$.
To solve the first regime, one needs to solve the homogeneous linear system that belongs to (4.1). This will be investigated in [FFG$^+$ep].

4.3.3 Dispersion

When solving the advection problem numerically with the *complete flux scheme*, one obtains oscillations. Hence, we can find a modified equation, that the *complete flux scheme* is practically solving. According to [Kum17], this modified equation is given as

$$\begin{aligned} &c_t + (vc)_x + \epsilon c_{xxx} = 0 \\ &c(0,x) = c_0(x) \end{aligned} \tag{4.17}$$

with $\epsilon = vO(h^2)$.
To solve this equation we use the following approach

$$c(x,t) = c_0 e^{i(kx - \omega t)}, \qquad k \in \mathbb{R},$$

where k is the wave number and ω is the frequency.
The relationship between k and ω with

$$\omega = \omega(k)$$

is called dispersion relation, see also [Tre82].
One obtains

$$\begin{aligned} c_t &= -i\omega c_0 e^{i(kx-\omega t)} \\ c_x &= ihc_0 e^{i(kx-\omega t)} \\ \epsilon c_{xxx} &= \epsilon(ik)^3 c_0 e^{i(kx-\omega t)} = -\epsilon i k^3 c_0 e^{i(kx-\omega t)}. \end{aligned}$$

Substituting these derivatives in (4.17), we get

$$\begin{aligned} &(-iw + vik - \epsilon ik^3)c_0 e^{i(kx-\omega t)} = 0 \\ \Rightarrow &- iw + vik - \epsilon ik^3 = 0 \\ \Rightarrow &i\omega = vik + \epsilon ik^3 \\ \Rightarrow &\omega = vk + \epsilon k^3. \end{aligned}$$

The phase velocity corresponding to the dispersion relation is then given by

$$v_{phase} := \frac{\omega}{k} = v + \epsilon k^2.$$

With a *Fourier transformation*, we can decompose the start value $c_0(x)$ in different waves c_k; $c_0 = \sum \alpha_k c_k$. The different waves have different phase velocities; respectively phase speeds. This causes the numerical solution to propagate faster for a continuous start value, respectively, the numerical solution obtains oscillations for non-continuous start values over time, see also [Kum17].

5. Example

In this chapter we will solve the periodic advection problem

$$
\begin{aligned}
c_t + (vc)_x = 0, \qquad x \in (0,1),\\
c(t,0) = c(t,1) \quad \forall t > 0\\
c(0,x) = c_0(x).
\end{aligned}
$$

To solve the equation we use both schemes, the *simple upwind scheme* and the *complete flux scheme* with different start value, a continuous and a non-continous one. Additionally, for the velocity it holds $v(t,x) = 1, \forall t > 0,\ x \in (0,1)$.

5.1 Continuous start value

For the continuous start value, one used 26 grid nodes.
The continuous start value is $c_0(x) = 2 + \sin(2\pi x)$, see Figure 5.1.

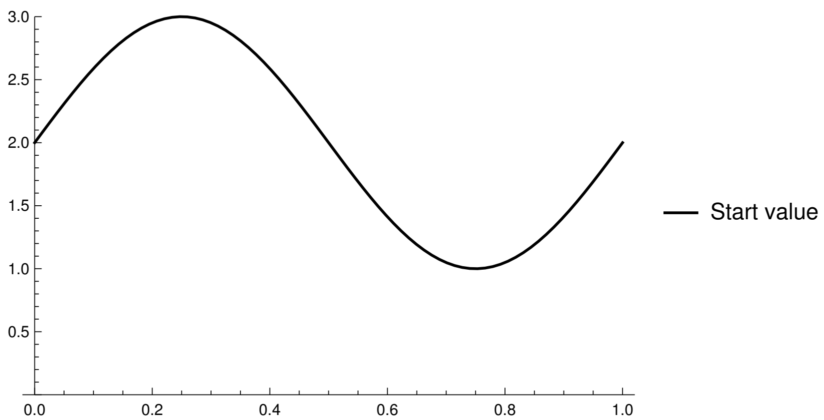

Figure 5.1: Continuous start value

With the *simple upwind discretization* we obtain the following numerical solutions at different times $t = 0.2,\ 0.5,\ 1,\ 2,\ 3,\ 5$.

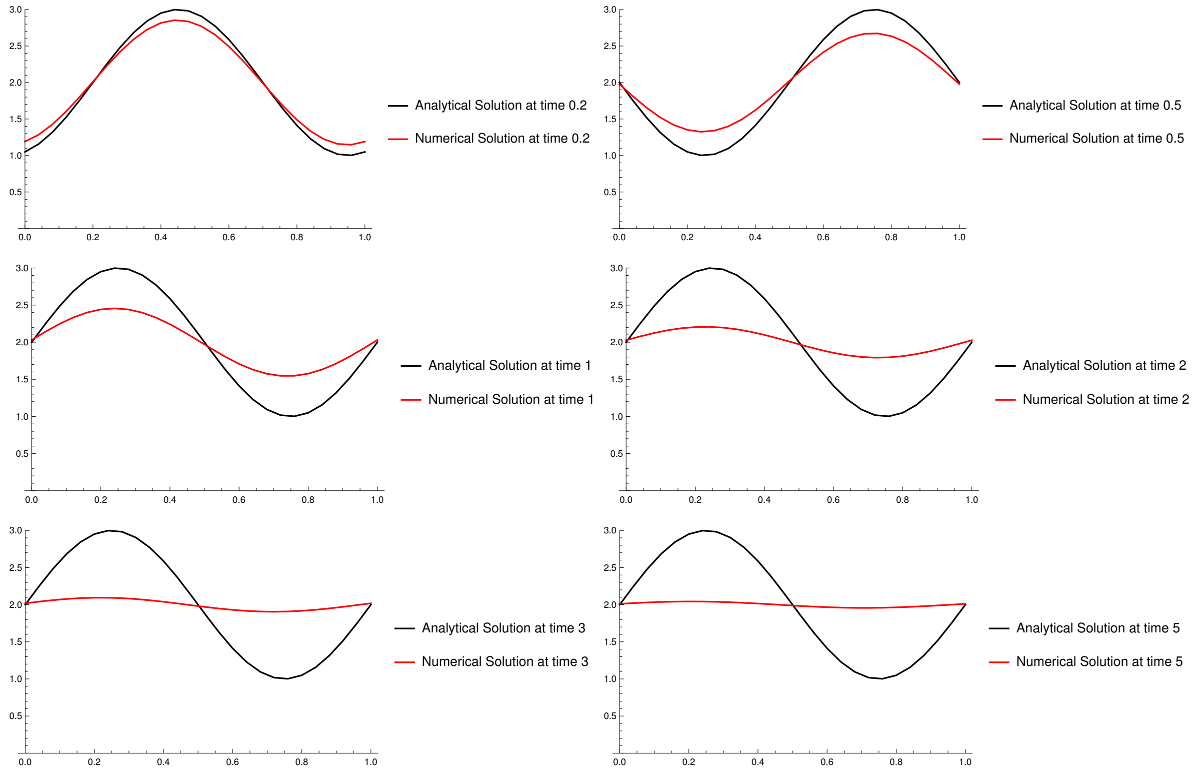

Figure 5.2: Analytical and numerical solutions $c_{\text{num}}^{\text{SU}}$ at different times for a continuous start value

In Figure 5.2 the artificial diffusion can be seen. Over time the numerical solution is smeared until it holds $c_n^{SU} = 2$, with c_n^{SU} being the numerical solution for the *simple upwind discretization*.
The mass is conserved with $M = \int_0^1 2 + \sin(2\pi x)dx = 2$.

With the *complete flux discretization* we obtain the following numerical solution at different times $t = 0.2,\ 0.5,\ 1,\ 2,\ 3,\ 5$.

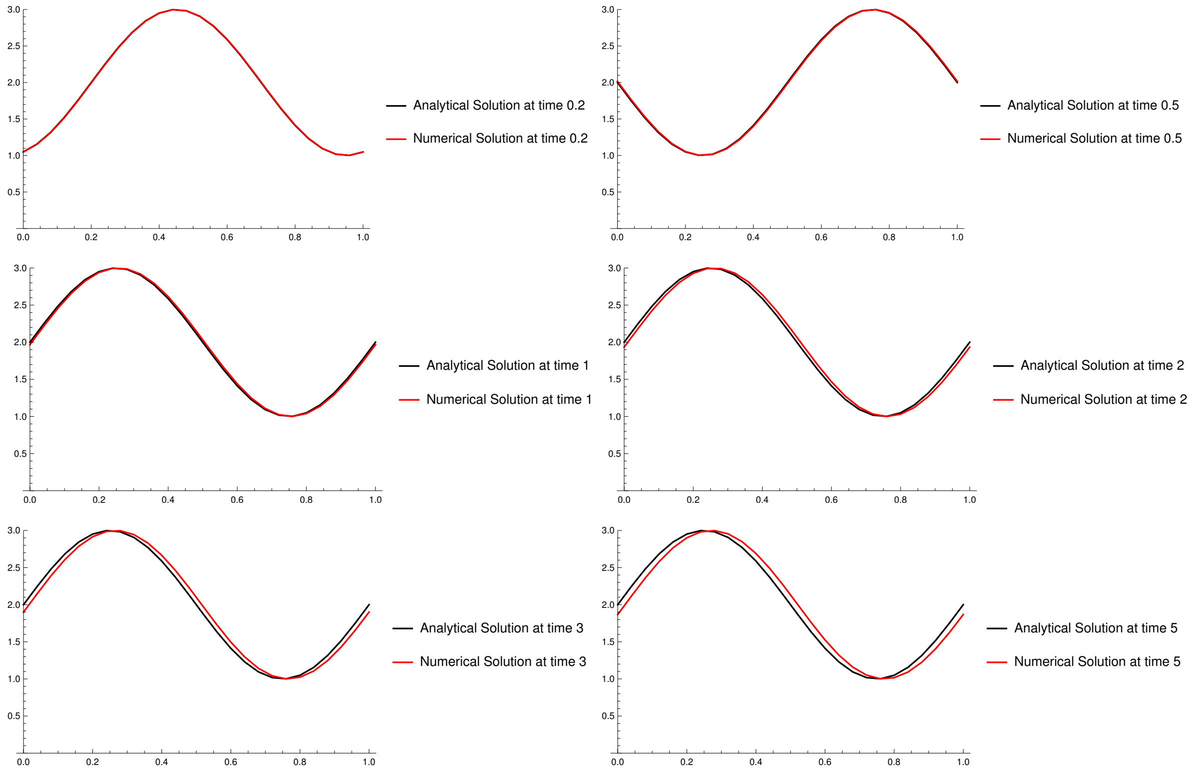

Figure 5.3: Analytical and numerical solutions $c_{\text{num}}^{\text{CFS}}$ at different times for a continuous start value

In Figure 5.3 it can been seen that the numerical solution propagates faster, due to the dispersion. But the mass is still conserved.

5.2 Non-continuous start value

For the non-continuous start value, one used 78 grid nodes.
The non-continuous start value, see Figure 5.4, is given as

$$c_0(x) = \begin{cases} 0, & \text{for} \quad 0 \leq x < \frac{1}{4},\, \frac{3}{4} < x \leq 1 \\ 1, & \text{for} \quad \frac{1}{4} \leq x \leq \frac{3}{4}. \end{cases}$$

One can easily see that this function is not continuous.

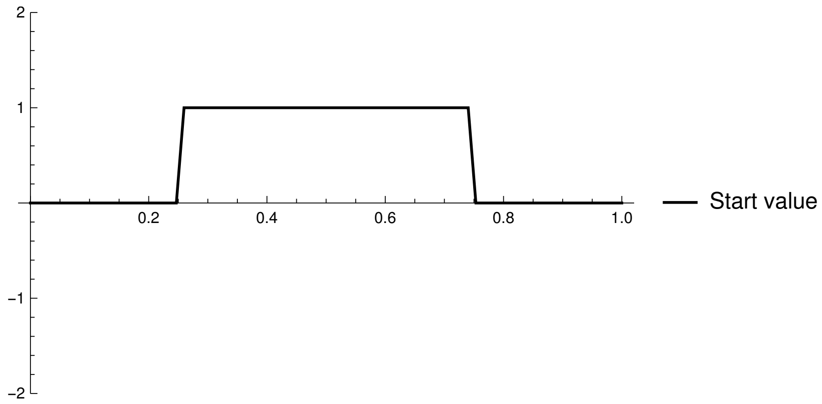

Figure 5.4: Non-continuous start value

With the *simple upwind discretization*, we obtain the following numerical solutions at different times $t = 0.01$, 0.05, 0.1, 0.5, 1, 2.

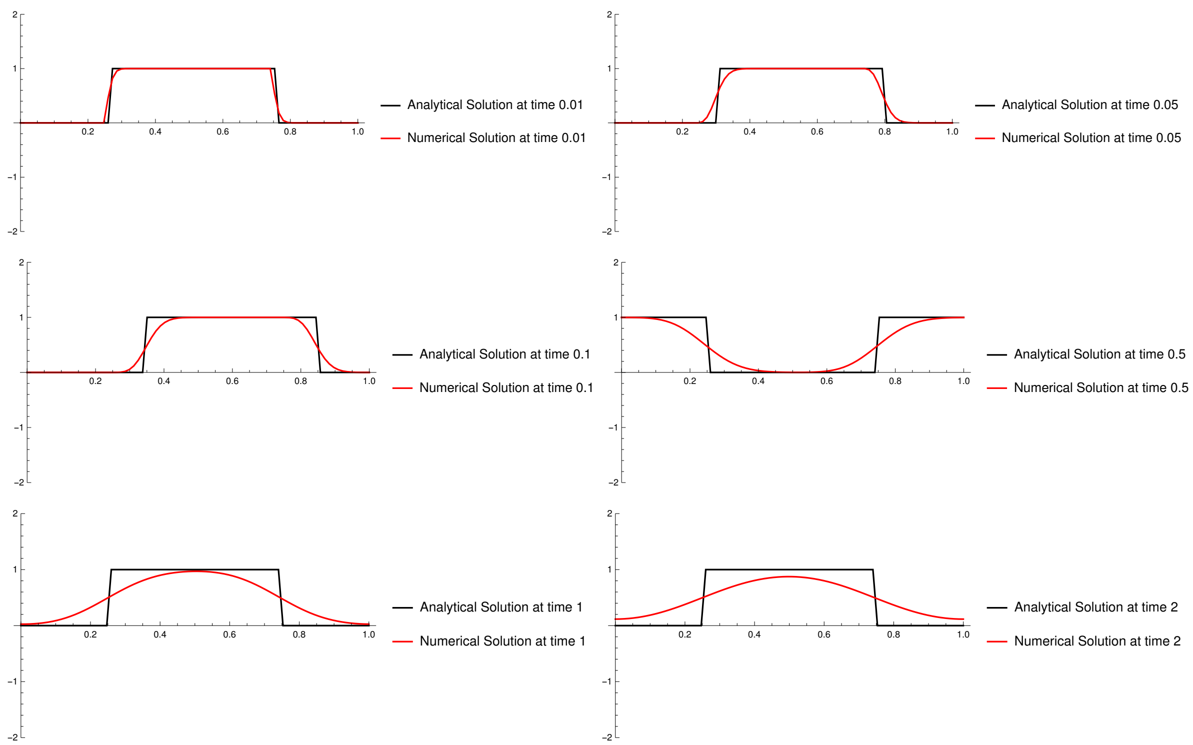

Figure 5.5: Analytical and numerical solutions $c_{\text{num}}^{\text{SU}}$ at different times for a non-continuous start value

In Figure 5.2 the artificial diffusion can be seen. Over time, the numerical solution is smeared

until the numerical solution converges towards $c_{\text{num}}^{\text{SU}} = 0.5$, with $c_{\text{num}}^{\text{SU}}$ being the numerical solution for the *simple upwind discretization*.
The mass is conserved with $M = \int_0^1 c_0(x)\,dx = 0.5$.

With the *complete flux discretization*, we obtain the following numerical solution at different times $t = 0.0001, 0.0005, 0.001, 0.01, 0.1, 0.5$.

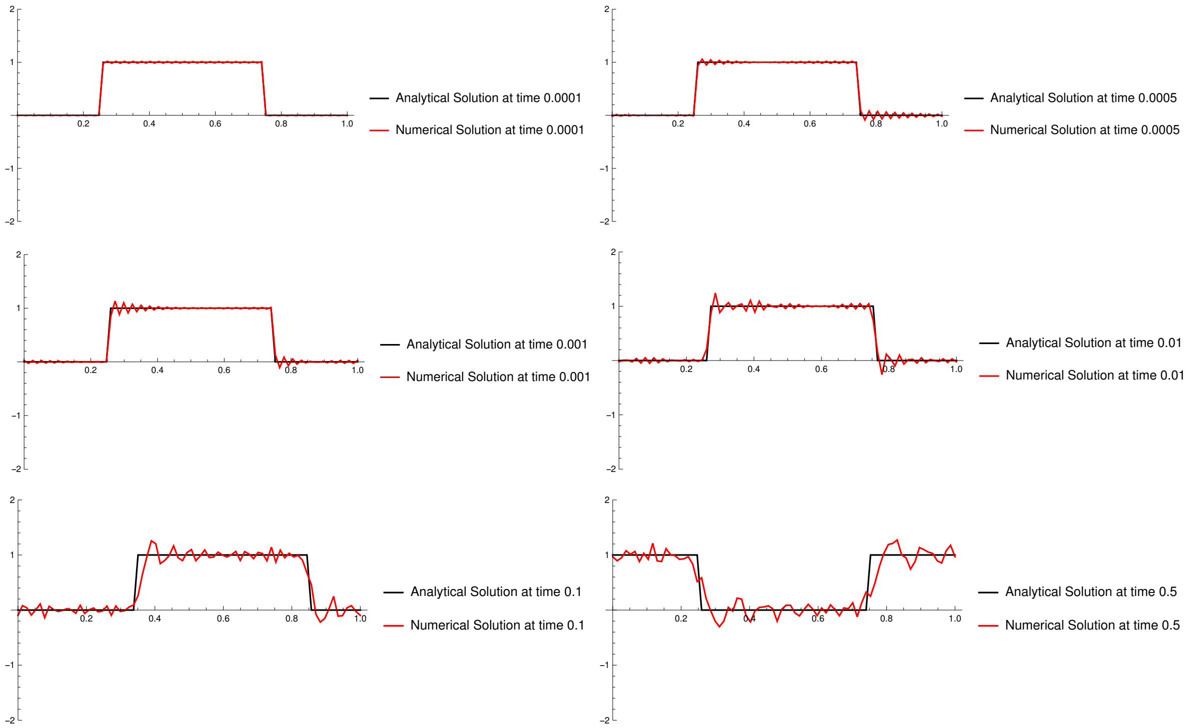

Figure 5.6: Analytical and numerical solutions $c_{\text{num}}^{\text{CFS}}$ at different times for a non-continuous start value

In Figure 5.6 the oscillations of the numerical solution $c_{\text{num}}^{\text{CFS}}$ are visible. As a result of the dispersion, these oscillations are obtained. They grow and spread, so the error rises over time. Due to the oscillations, the maximum principle is violated, but the mass is still conserved. When using more grid nodes the error of the *complete flux scheme* decays, i.e. the amplitude of the oscillations is smaller than before.

Bibliography

[BTa92] Alberto Borobia and Ujué R. Trí as. A geometric proof of the Perron-Frobenius theorem. *Rev. Mat. Univ. Complut. Madrid*, 5(1):57–63, 1992.

[FFG+ep] Patricio Farrell, Juergen Fuhrmann, Annegret Glitzky, Katharina Hoffmann, Alexander Linke, and J. H. M ten Thije Boonkkamp. On a nonlinear dissipative interpolation of a simple upwind and a second order complete flux scheme for linear advective transport, in prep.

[HJ94] Roger A. Horn and Charles R. Johnson. *Topics in matrix analysis.* Cambridge University Press, Cambridge, 1994. Corrected reprint of the 1991 original.

[HR15] Helge Holden and Nils Henrik Risebro. *Front tracking for hyperbolic conservation laws*, volume 152 of *Applied Mathematical Sciences.* Springer, Heidelberg, second edition, 2015.

[Kum17] Nikhil Kumar. *Flux Approximation Schemes for Flow Problems using Local Boundary Value Problems.* PhD thesis, Technische Universiteit Eindhoven, 2017.

[Tee07] Garry J. Tee. Eigenvectors of block circulant and alternating circulant matrices. *New Zealand J. Math.*, 36:195–211, 2007.

[Tre82] Lloyd N. Trefethen. Group velocity in finite difference schemes. *SIAM Rev.*, 24(2):113–136, 1982.

[tTBA11] J. H. M. ten Thije Boonkkamp and M. J. H. Anthonissen. The finite volume-complete flux scheme for advection-diffusion-reaction equations. *J. Sci. Comput.*, 46(1):47–70, 2011.

[Wer00] Dirk Werner. *Funktionalanalysis.* Springer-Verlag, Berlin, extended edition, 2000.

Appendix: Mathematica Code

```
In[∘]:= getUniformGrid[n_] := Module[{i, grid},
   grid = Table[i 1/n, {i, 0, n}]; (* get uniform grid with n nodes *)
   Return[grid];
  ];
```

```
In[∘]:= getRandomGrid[n_] := Module[{i, grid},
   grid = Table[Random[], {i, 0, n}]; (* get random grid with n nodes *)
   grid[[1]] = 0;
   grid[[n + 1]] = 1;
   Return[Sort[grid]];
  ];
```

```
In[∘]:= getSlightlyRandomGrid[n_, p_] := (* get slightly random grid with n nodes
   and p being the percentage that the node can be off the uniform node*)
  Module[{i, grid},
   grid = Table[i 1/n - 1/n p (Random[] - 1/2), {i, 0, n}];
   grid[[1]] = 0;
   grid[[n + 1]] = 1;
   Return[Sort[grid]];
  ];
```

```
In[∘]:= getVoroFaces[grid_] :=
  Module[{n, d, i, dvec}, n = Length[grid] - 1; (* get the nodes of the voronoi box*)
   dvec = Table[(grid[[i + 1]] + grid[[i]]) / 2, {i, 1, n}];
   Return[dvec];
  ];
```

```
In[∘]:= getSUmassmatrix[grid_, dvec_] := Module[{n, d, i, Dmat},
   n = Length[dvec]; (* get the mass matrix of the Simple Upwind scheme*)
   Dmat = IdentityMatrix[n];
   Dmat[[1]][[1]] = grid[[n + 1]] - dvec[[n]] + dvec[[1]] - grid[[1]];
   For[i = 2, i ≤ n, i++, Dmat[[i]][[i]] = dvec[[i]] - dvec[[i - 1]]];
   Return[Dmat];
  ];
```

```
In[∘]:= getCFSmassmatrix[grid_, dvec_] := Module[{n, d, i, Dmat},
   n = Length[grid]; (* get the mass matrix of the Complete Flux scheme*)
   Dmat = IdentityMatrix[n - 1];
   Dmat[[1]][[1]] = (grid[[n]] - grid[[n - 1]]) / 2;
   Dmat[[1]][[n - 1]] = (grid[[n]] - grid[[n - 1]]) / 2;

   For[i = 2, i ≤ n - 1, i++,
    Dmat[[i]][[i]] = (grid[[i]] - grid[[i - 1]]) / 2;
    Dmat[[i]][[i - 1]] = (grid[[i]] - grid[[i - 1]]) / 2;
   ];
   Return[Dmat];
  ];
```

```
In[•]:= getSUAmatrix[grid_, dvec_, vfunc_] := Module[{n, d, i, Mmat},
    n = Length[grid]; (* get the flux matrix of the Simple Upwind scheme*)
   Mmat = IdentityMatrix[n - 1];
   Mmat[[1]][[1]] = (vfunc[grid[[2]]] + vfunc[grid[[1]]]) / 2;
    Mmat[[1]][[n - 1]] = -(vfunc[grid[[n - 1]]] + vfunc[grid[[n]]]) / 2;
   For[i = 2, i ≤ n - 1, i++,
     Mmat[[i]][[i - 1]] = -(vfunc[grid[[i - 1]]] + vfunc[grid[[i]]]) / 2];
    For[i = 2, i ≤ n - 1, i++, Mmat[[i]][[i]] =
      (vfunc[grid[[i + 1]]] + vfunc[grid[[i]]]) / 2];
   Return[Mmat];
   ];

In[•]:= getCFSAmatrix[grid_, dvec_, vfunc_] := Module[{n, d, i, Mmat},
    n = Length[grid]; (* get the flux matrix of the Complete Flux scheme*)
   Mmat = IdentityMatrix[n - 1];
   Mmat[[1]][[1]] = vfunc[grid[[1]]];
    Mmat[[1]][[n - 1]] = -vfunc[grid[[n - 1]]];
  For[i = 2, i ≤ n - 1, i++, Mmat[[i]][[i - 1]] = -vfunc[grid[[i - 1]]]];
    For[i = 2, i ≤ n - 1, i++, Mmat[[i]][[i]] = vfunc[grid[[i]]]];
   Return[Mmat];
   ];

In[•]:= LinearSystemSU[n_, grid_, time_] :=
    Module[{A, M, voro, eqns, sol}, (* solve the SU linear system*)
   voro = getVoroFaces[grid];
  M = getSUmassmatrix[grid, voro];
    A = getSUAmatrix[grid, voro, vfunc];
 eqns = Join[{M.Table[c[i]'[t], {i, 1, n}] +
        A.Table[c[i][t], {i, 1, n}] == Table[0, {i, 1, n}]},
      {Table[c[i][0], {i, 1, n}] == startvaluefunc[n, grid]}];
    sol = NDSolve[eqns, Table[c[i], {i, n}],
      {t, time}, Method -> {"EquationSimplification" -> "Residual"}];
    Return[sol];
   ];

In[•]:= LinearSystemCFS[n_, grid_, time_] :=
    Module[{A, M, voro, eqns, sol}, (* solve the CFS linear system*)
   voro = getVoroFaces[grid];
  M = getCFSmassmatrix[grid, voro];
    A = getCFSAmatrix[grid, voro, vfunc];
 eqns = Join[{M.Table[c[i]'[t], {i, 1, n}] +
        A.Table[c[i][t], {i, 1, n}] == Table[0, {i, 1, n}]},
      {Table[c[i][0], {i, 1, n}] == startvaluefunc[n, grid]}];
    sol = NDSolve[eqns, Table[c[i], {i, n}],
      {t, time}, Method -> {"EquationSimplification" -> "Residual"}];
    Return[sol];
   ];

In[•]:= vfunc[x_] := 1(*2+Sin[Pi x] *) (*velocity*)

In[•]:= startfunc[x_] := If[Mod[Round[2 x - 1], 2] == 0, 1, 0]
 (* 2+Sin[2Pi x]*)(* startvalue*)
```

```
In[ ]:= startvaluefunc[n_, grid_] := Module[{startvalue},
           startvalue = Table[startfunc[grid[[i]]], {i, 1, n}];
           Return[startvalue];
          ];

In[ ]:= numsol[sol_, t_] :=
         Module[{vecplot}, (* create vector with  numerical solution at time t*)
        vecplot = Table[(c[i][t] /. sol)[[1]], {i, 1, n}];
          AppendTo[vecplot, (c[1][t] /. sol)[[1]]];
          Return[vecplot];
        ];

       mass[grid_, sol_, n_] := Module[{mass, voro}, (* numerates the mass of the solution*)
            voro = getVoroFaces[grid];
           mass = 0;
           For[i = 1, i ≤ n - 1, i++, mass += (voro[[i + 1]] - voro[[i]]) * sol[[i + 1]]];
           mass += ((1 - voro[[n]]) + (voro[[1]] - 0)) * sol[[1]];
           Return[mass];
          ];

In[ ]:=
       analyticalsol[t_, grid_]  := Module[{solution},
           solution = Table[startfunc[grid[[i]] - vfunc[ grid[[i]]] * t], {i, 1, n}];
           (* create vector with  analytical solution at time t with v constant*)
           Return[AppendTo[solution, solution[[1]]]];
         ];


In[ ]:= massprinciple[grid_, sol_, time_, n_] :=
         Module[{massvec, timevec}, (*creates vector of certain masses until time t*)
          massvec = Table[mass[grid, numsol[sol, i], n], {i, 0, time, time / (time * 10)}];
          timevec = Table[i, {i, 0, time, time / (time * 10)}];
          Return[{massvec , timevec}];
          ];

In[ ]:= MaxError[sol_, grid_, t_] := Module[{solution, err}, (* Max Error for v = const*)
           err = {};
            For[i = 0, i ≤ t, i += t / 20,
             AppendTo[err, Norm[numsol[sol, i] - analyticalsol[i, grid], Infinity]]];
           Return[Max[err]];
         ];

In[ ]:= n = 11

Out[ ]= 11

In[ ]:= time = 1

Out[ ]= 1

In[ ]:= grid = getUniformGrid[n]
```

Out[ ]= $\left\{0, \frac{1}{11}, \frac{2}{11}, \frac{3}{11}, \frac{4}{11}, \frac{5}{11}, \frac{6}{11}, \frac{7}{11}, \frac{8}{11}, \frac{9}{11}, \frac{10}{11}, 1\right\}$

```
In[ ]:= sol = LinearSystemCFS[n, grid, time]
```

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und eigenhändig sowie ohne unerlaubte fremde Hilfe und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.

Die selbständige und eigenhändige Anfertigung versichert an Eides statt:

Berlin, den 21.August 2018

Katharina Charlotte Lina Hoffmann