Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Hamburg University

Master Thesis

Two-Component Model for Tracer Simulation

Sophie Externbrink

supervised by

Dr. Sebastian Götschel, Hamburg University of Technology

Prof. Dr. Daniel Ruprecht, Hamburg University of Technology

TUHH
Hamburg University of Technology

December 7, 2022

DOI: https://doi.org/10.15480/882.4765

Statutory Declaration

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit im Masterstudiengang Technomathematik selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel – insbesondere keine im Quellenverzeichnis nicht benannten Internet-Quellen – benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht. Ich versichere weiterhin, dass ich die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe und die eingereichte schriftliche Fassung der auf dem elektronischen Speichermedium entspricht.

Sophie Externbrink

Contents

List of Figures

Introduction

The objective of this thesis is to numerically solve a model, simulating the transportation of a tracer bolus through blood flow in the liver. A good model is important, especially in the field of cancer research, because tumor perfusion and other vascular properties are important parameters of cancer's response to therapy. Good perfusion imaging allows an accurate model of the tumor's vascular state and perfusion. With this model, critical determinants in the tumor's progression and its response to therapy can be derived. A good tracer-kinetic model, combined with dynamic imaging data, can give individual time curves for the concentration of substances, such as chemotherapeutics, measured over whole organs.
We are going to derive a model for the liver similar to the ones in [6] and [18]. The model is going to be a two-component model. A two-component model is used, because a one-component model does not simulate the perfusion accurately [18].

For the implementation, we need a system solver. We will start by deriving a weighted essentially non-oscillatory (WENO) solver and testing it for accuracy, especially for its ability to solve the advection equation with space-dependent velocity. WENO schemes have gained a lot of influence in numerical solutions of hyperbolic problems. WENO schemes are particulary suitable for problems containing both strong discontinuities or shocks and rich smooth region structures. The main advantage of WENO schemes and the reason they are so heavily used is their capability to achieve arbitrarily high-order formal accuracy in smooth regions while still maintaining stable and, most of all, non-oscillatory and sharp discontinuity transitions. The essential idea behind the scheme lies in the stencil choosing procedure. The WENO scheme uses a convex combination of all candidate stencils to gain the locally smoothest stencil for the reconstruction.

The WENO scheme was first developed by Liu, Osher and Chan in 1994 [11]. They developed the procedure by improving the already well-functioning essentially non-oscillatory (ENO) scheme [16],[17]. For a short overview of both procedures, we refer the reader to

[15]. Jiang and Shu then improved and extended the WENO method in 1994 [8] for an easier implementation. An extensive review about this method can be found in [14].

The WENO procedure has been used within finite volume and finite difference schemes: E. g. in applications regarding computational fluid dynamics [3],[13], computational physics [12], engineering [4] and even traffic flow [20].

We hope that this thesis will further improve the knowledge about WENO solvers and helps to increase the knowledge about solving the two-component model for intravascular indicators.

Structure of the thesis. In the first chapter, we will give the reader an introduction into the theory of solving advection equations and especially space-dependent advection equations. We assume that the reader has a general knowledge about partial and ordinary differential equations. Moreover, we will present the model used for the simulation of tracer transportation in the liver in this thesis.
The finite difference WENO scheme will be introduced in detail in the second chapter. We will start with the one-dimensional scalar conservation law in subsection 2.1 and explain the used reconstruction method in subsection 2.2. Then we will move on to the time discretization method and the two-dimensional conservation law in subsections 2.3 and 2.4. We end the chapter by giving an overview about the handling of boundary conditions in subsection 2.5.
In the third chapter, we are going to verificate our solver in subsections 3.1 and 3.2 and finally use the solver for the simulation of the two-component model for an intravascular indicator in subsection 3.3.

1 Theory

1.1 Advection with Space-Dependent Velocity

When looking at the advection equation, we often consider a constant velocity v_0

$$u_t(x,t) + v_0 \cdot u_x(x,t) = 0, \tag{1.1}$$

with partial derivatives $u_x = \frac{\partial u}{\partial x}$ and $u_t = \frac{\partial u}{\partial t}$. But when modelling blood flow, the velocity is not constant. Therefore, we will now take a look at a space-dependent velocity

$$u_t(x,t) + v(x) \cdot u_x(x,t) = 0. \tag{1.2}$$

This is still an advection equation, but the solution differs from the advection equation with constant velocity. When looking at the solution of first order, hyperbolic partial differential equations, it may become helpful to take a look at the characteristics.
Characteristics are areas in the $x-t$-plane, along which the solution of a first order partial differential euqation is constant. Therefore, if we know how the characteristics of an equation are defined, we can easily find the solution at any point (x,t) by simply following the characteristic through this point back to $t = 0$ and find the solution from the initial value. By definition the solution has to be constant along the characteristics $x(t)$

$$\frac{du(x(t),t)}{dt} = u_x(x(t),t) \cdot \frac{dx(t)}{dt} + u_t(x(t),t) \stackrel{!}{=} v(x) \cdot u_x(x(t),t) + u_t(x(t),t) = 0. \tag{1.3}$$

In comparison, we can see that

$$\frac{dx(t)}{dt} = v(x(t),t). \tag{1.4}$$

This means that for the constant velocity case, there are linear functions in t

$$x(t) = x(0) + v_0 \cdot t. \tag{1.5}$$

Now we want to determine the solution u at any point (x,t). At first, we need to identify the characteristic on which the point is located. Then we can follow it back to $t = 0$.

We want to find $x(0)$ such that $x(t) = x$. This would define the characteristic crossing through the point (x, t). We can see that

$$x(0) = x - v_0 \cdot t \tag{1.6}$$

solves our problem. Therefore, the solution $u(x, t)$ is equivalent to the initial value u_0 at $x(0) = x - v_0 \cdot t$

$$u(x, t) = u_0(x - v_0 \cdot t). \tag{1.7}$$

To prove that this is indeed a solution to our advection equation, we can have look at the following calculation

$$\partial_t u_0(x - v_0 \cdot t) + v_0 \cdot \partial_x u_0(x - v_0 * t) = u_0'(x - v_0 \cdot t) \cdot (-v_0) + v_0 \cdot u_0'(x - v_0 \cdot t) = 0. \tag{1.8}$$

Now, the characteristics are not linear for the non-constant case. Instead, they are defined by the following integral

$$x(t) = x(0) + \int_0^t v(x(\tau))d\tau. \tag{1.9}$$

Using the same strategy as for the constant velocity case, we want to find the solution at any point (x, t) in spacetime first. For that, we can integrate the characteristics back in time to $t = 0$

$$x(0) = x + \int_t^0 v(x(\tau))d\tau = x - \int_0^t v(x(\tau))d\tau. \tag{1.10}$$

This gives the solution

$$u(x, t) = u_0\left(x - \int_0^t v(x(\tau))d\tau\right). \tag{1.11}$$

To check if this really is a solution of the advection equation, we calculate

$$\begin{aligned} &\frac{\partial}{\partial t} u_0\left(x - \int_0^t v(x(\tau))d\tau\right) + v(x) \cdot u_0\left(x - \int_0^t v(x(\tau))d\tau\right) \\ &= u_0'\left(x - \int_0^t v(x(\tau))d\tau\right) \cdot (-v(x)) + v(x) \cdot u_0'\left(x - \int_0^t v(x(\tau))d\tau\right) = 0. \end{aligned} \tag{1.12}$$

More generally, we might consider the advection equation with space-dependent velocity as conservation law:

$$u_t(x, t) + (v(x) \cdot u(x, t))_x = 0. \tag{1.13}$$

By applying the product rule, this equation is equivalent to

$$u_t(x,t) + v(x) \cdot u_x(x,t) = -v'(x) \cdot u(x,t). \tag{1.14}$$

We can see that this equation differs slightly from the previous: A solution-dependent source term is added. To solve this equation, we can also use the method of characteristics. We start by taking a look at the general equation

$$a(x,t) \cdot u_x(x,t) + b(x,t) \cdot u_t(x,t) = c(x,t,u). \tag{1.15}$$

We are going to derive the characteristics via a graphical approach similar to the one in [10]. Therefore, we consider the graph of the possible solution $u(x,t)$ defined by

$$S \equiv \{(x,t,u(x,t))\}. \tag{1.16}$$

If $u(x,t)$ is indeed a solution, the following equation has to hold at each point (x,t)

$$(a(x,t), b(x,t), c(x,t,u)) \cdot (u_t(x,t), u_x(x,t), -1)^T = 0. \tag{1.17}$$

In general, the normal vector at a point (x_0,t_0) on a surface $z = f(x,t)$ is given by

$$N = \begin{pmatrix} f_x(x_0,t_0) \\ f_t(x_0,t_0) \\ -1 \end{pmatrix}, \text{ with partial derivatives } f_x = \frac{\partial f}{\partial x} \text{ and } f_y = \frac{\partial f}{\partial y}.$$

This yields the normal vector to the surface S at point (x,t,u)

$$N(x,t) = \begin{pmatrix} u_x(x,t) \\ u_t(x,t) \\ -1 \end{pmatrix}.$$

This shows that the normal vector $(a(x,t), b(x,t), c(x,t,u))$ is actually orthogonal to the normal vector of S and therefore lies in the tangent plane of S.

To find a solution $u(x,t)$, we have to look for a surface S such that at each point (x,t,z) in S the vector $(a(x,t), b(x,t), c(x,t,u))$ lies in the tangent plane. To derive this surface, we have to find a curve $\mathcal{C}$ that lies in S. This curve is given by $\mathcal{C} = \{x(s), t(s), z(s)\}$ and

parametrized by s to ensure that the vector $(a(x(s),t(s)), b(x(s),t(s)), c(x(s),t(s),u))$ is tangent to our curve $\mathcal{C}$. This is guaranteed if the following set of ordinary differential equations (ODE) is fullfilled

$$\begin{aligned} \frac{\partial x}{\partial s} &= a(x(s),t(s)) \\ \frac{\partial t}{\partial s} &= b(x(s),t(s)) \\ \frac{\partial z}{\partial s} &= c(x(s),t(s),u(x(s),t(s))). \end{aligned} \tag{1.18}$$

This set of ODEs is known as the set of characteristic equations of the general partial differential equation (PDE) (1.15). A curve satisfying these ODEs is known as an integral curve [9] for the vector field $(a(x,t), b(x,t), c(x,t,u))$.

When these characteristic curves are calculated, we construct our solution of (1.15) by forming the surface S as a union of characteristic curves. To summarize, we introduce the characteristic equations to reduce our PDE to a system of ODEs, on which we can use the known ODE theory to solve these characteristic equations and build a surface using the derived characteristic curves. This surface will provide us with a solution to our PDE.

To get a unique solution of a PDE, we further need to introduce initial conditions. Assume the initial condition of our equation is given by $u(x,0) = \Phi(x)$.

This means that the curve $\{x, 0, \Phi(x)\}$ has to be contained in the surface S. We can ensure this by introducing inital conditions for the characteristic equations (1.18), but to get general solutions, we have to introduce a new parameter r to parameterize the curve $\Gamma \equiv \{(x,0)\}$ in $\mathbb{R}^2$ with (Γ, Φ) a curve in $\mathbb{R}^3$. When using r, we get $\Gamma = \{(r,0)\}$ and our variables x and t now not only depend on s, but also on r: $x(r,s)$, $t(r,s)$. For a better understanding, we will now take a look at an example.

Consider the equation

$$\begin{aligned} u_t + (f(u))_x = 0 \text{ with } f(u) = x \cdot u \\ u(x,0) = \Phi(x). \end{aligned} \tag{1.19}$$

By filling in the definitions and using the product rule we derive

$$\begin{aligned} u_t + x \cdot u_x &= -u \\ u(x,0) &= \Phi(x). \end{aligned} \tag{1.20}$$

From this, we derive the following set of ODEs with their corresponding initial conditions (ICs)

$$\begin{aligned} \frac{\partial t}{\partial s}(r,s) = 1 \quad &\Rightarrow \quad t(r,0) = 0 \\ \frac{\partial x}{\partial s}(r,s) = x \quad &\Rightarrow \quad s(r,0) = r \\ \frac{\partial z}{\partial s}(r,s) = -z \quad &\Rightarrow \quad z(r,0) = \Phi(r). \end{aligned} \tag{1.21}$$

Solving our system and looking at the initial conditions we get

$$\begin{aligned} t(r,s) = s + c_1(r) \quad &\Rightarrow \quad t(r,0) = 0 = c_1(r) \\ x(r,s) = c_2(r) \cdot e^s \quad &\Rightarrow \quad x(r,0) = r = c_2(r) \\ z(r,s) = c_3(r) \cdot e^{-s} \quad &\Rightarrow \quad z(r,0) = \Phi(r) = c_3(r). \end{aligned} \tag{1.22}$$

Consequently, our solution is uniquely defined by

$$\begin{aligned} t(r,s) &= s \\ x(r,s) &= r \cdot e^s \\ z(r,s) &= \Phi(r) \cdot e^{-s}. \end{aligned} \tag{1.23}$$

Now we want to find the definition of the surface. Therfore, we solve for r, s with regard to x, t

$$\begin{aligned} s(x,t) &= t \\ r(x,t) &= x \cdot e^{-t}. \end{aligned} \tag{1.24}$$

With these equations we can write down our solution $z(r,s)$ and derive our solution $u(x,t)$

$$z(r(x,t), s(x,t)) = u(x,t) = \Phi(x \cdot e^{-t}) \cdot e^{-t}. \tag{1.25}$$

This is indeed a solution of our PDE, as will be proved in the following

$$\begin{aligned}
\frac{\partial u}{\partial x} &= \Phi'(x \cdot e^{-t}) \cdot e^{-2t} \\
\frac{\partial u}{\partial t} &= \Phi'(x \cdot e^{-t}) \cdot (-x \cdot e^{-2t}) - e^{-t} \cdot \Phi(x \cdot e^{-t}) \\
\Rightarrow u_x \cdot x + u_t &= \Phi'(x \cdot e^{-t}) \cdot e^{-2t} \cdot x - x \cdot \Phi'(x \cdot e^{-t}) \cdot e^{-2t} - e^{-t} \cdot \Phi(x \cdot e^{-t}) \\
&= -e^{-t} \cdot \Phi(x \cdot e^{-t}) = -u.
\end{aligned} \tag{1.26}$$

1.2 Two-Component Model

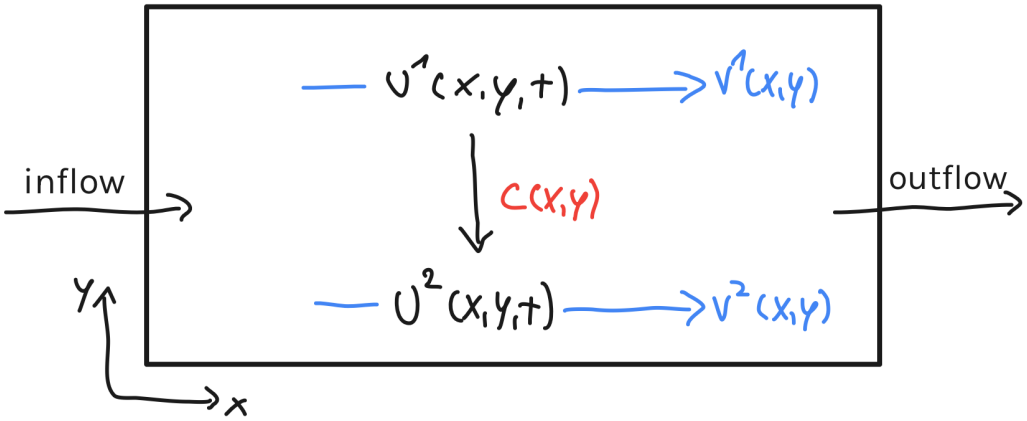


Figure 1: Two-component model we want to simulate

The goal of this thesis is to simulate the transportation of a tracer in the blood flow of the liver. A simplified two-component model of this can be found in Figure 1. It is a simplification of the models used in [18] and [6]. The basis of the model we are using is an advection equation without diffusion. We assume a space-dependent velocity $v(x, y)$. Furthermore, we have two components u^1 and u^2 which represent tracer in the oxygenated and deoxygenated blood respectively. To get a realistic model, we make sure that there is only oxygenated blood at the inflow boundary and only deoxygenated blood at the outflow boundary. The oxygenated blood is converted into deoxygenated blood using a space-dependent rate $c(x, y)$. In addition, both components should be able to move with different velocities v^1 and v^2. v^1 is the velocity of the oxyginated blood which which is fast at the inflow boundary and is reduced to zero at the outflow boundary. v^2 is the velocity of the deoxygenated blood which is slow at the inflow boundary and fast

at the outflow boundary. Altogether, the following system of equations applies

$$\begin{aligned} u_t^1(x,y,t) + f(u^1)_x &= -c(x,y) \cdot u^1(x,y,t) \\ u_t^2(x,y,t) + g(u^2)_x &= c(x,y) \cdot u^1(x,y,t) \\ f(u^1) &= v^1(x,y) \cdot u^1(x,y,t) \\ g(u^2) &= v^2(x,y) \cdot u^2(x,y,t) \end{aligned} \tag{1.27}$$

with $(x,y) \in \Omega$, for given Ω and $t \geq 0$.

2 Finite Difference WENO Scheme

After having presented the problem to be solved in the previous chapter, we now introduce the numerical solver. We will use a finite difference WENO scheme. WENO schemes are based on the method of lines. This means that the PDE we want to solve is discretized in space using the WENO algorithm. After that, the resulting ODE is solved by a time discretization method, in our case a Runge-Kutta scheme.
At the beginning of this chapter, we will give a brief overview of the fifth order finite difference WENO scheme in the framework developed by Chi-Wang Shu [14]. For a better understanding, we consider one-dimensional scalar conservation laws first. In these, the numerical flux is obtained by the WENO scheme. After explaining the WENO reconstruction, we will take a look at the time discretization method. Afterwards, we will explain how to generalize the approximation to multiple space dimensions. Finally, we will discuss the implementation and convergence for different cases of the advection equation.

2.1 1D Scalar Conservation Law

What we want to solve in general is the following hyperbolic PDE

$$u_t + (f(u))_x = 0, \tag{2.1}$$

where $u(x,t)$ is the concentration of a substance and $f(u)$ describes a flow. As stated earlier, the basic idea is that we want to break the PDE down to an ODE

$$u_t = g(t, y(t)), \tag{2.2}$$

which we can solve with a time discretization method. In order to do so, we have to approximate $(f(u))_x$ via the WENO method with flux splitting.
We discretize the problem with a uniform mesh of the following form

$$a = x_{\frac{1}{2}} < x_{1+\frac{1}{2}} < x_{2+\frac{1}{2}} < \cdots < x_{n+\frac{1}{2}} = b, \tag{2.3}$$

for the given interval $[a, b]$. Furthermore, we define the cells I, the cell-centers x_i and the spacing of the mesh Δx

$$\begin{aligned} I_i &= [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}], \\ x_i &= \frac{1}{2} \cdot (x_{i-\frac{1}{2}} + x_{i+\frac{1}{2}}), \\ \Delta x &= x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}. \end{aligned} \tag{2.4}$$

We want to use a finite difference scheme to derive an approximation for $(f(u))_x$

$$(f(u))_x\Big|_{x=x_i} = \frac{1}{\Delta x}(\hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}}) + \mathcal{O}(\Delta x^5), \tag{2.5}$$

for u smooth on the stencil $(u_{i-s}, \ldots, u_{i+t})$ and with numerical flux $\hat{f}_{i+\frac{1}{2}} = \hat{f}(u_{i-s}, \ldots, u_{i+t})$ consistent with the physical flux $\hat{f}(u, \cdots, u) = f(u)$. To establish a relationship between the finite difference approximation and the reconstruction the following Lemma by Shu and Osher [17] is needed.

Lemma 2.1. *If a function $h(x)$ satisfies*

$$f(u(x)) = \frac{1}{\Delta x} \cdot \int_{x-\frac{\Delta x}{2}}^{x+\frac{\Delta x}{2}} h(\psi) d\psi,$$

then

$$(f(u(x)))_x\Big|_{x=x_i} = \frac{1}{\Delta x}(h(x_{i+\frac{1}{2}}) - h(x_{i-\frac{1}{2}})).$$

Lemma 2.1 shows us that to achieve a fifth order accuracy for the numerical flow, we can set

$$\hat{f}_{i+\frac{1}{2}} = h(x_{i+\frac{1}{2}}) + \mathcal{O}(\Delta x^5). \tag{2.6}$$

Furthermore, we know that the cell average on each cell I_i of a function $h(x)$ is defined as followed

$$\overline{h}_i = \frac{1}{\Delta x} \cdot \int_{x+\frac{\Delta x}{2}}^{x+\frac{\Delta x}{2}} h(\psi) d\psi. \tag{2.7}$$

To derive the numerical flux $\hat{f}_{i+\frac{1}{2}}$, we now need to obtain $h(x_{i+\frac{1}{2}})$ by only using the known cell averages $\overline{h}_i$.

This is a reconstruction problem. The algorithm that solves the reconstruction problem

will be presented in the next section. For the WENO scheme to be stable we need to further add some flux splitting, to avoid upwinding, to the finite difference procedure [14]. The simplest smooth flux splitting is the Lax-Friedrich splitting

$$\begin{aligned} f^+(u) &= \frac{1}{2} \cdot (f(u) + \alpha \cdot u), \\ f^-(u) &= \frac{1}{2} \cdot (f(u) - \alpha \cdot u), \end{aligned} \tag{2.8}$$

with $\alpha = \max_u |f'(u)|$ in the relevant range of u. We can verify that

$$\frac{d}{du} f^+(u) \geq 0 \quad \text{and} \quad \frac{d}{du} f^-(u) \leq 0. \tag{2.9}$$

This shows that we are using a global flux splitting which is usually more robust [14] than the alternative local flux splitting, e. g. Roe splitting which uses the Roe speed [17]. Moreover, local flux splitting can derive entropy violating solutions, which could be avoided by applying local entropy correction [17]. This effort is avoided by choosing global flux splitting in the first place.

2.2 Reconstruction

We have already derived that to compute the numerical flux $\hat{f}_{i+\frac{1}{2}}$, we need to approximate $h(x)$ with accuracy $\mathcal{O}(x^5)$ at the cell border $x_{i+\frac{1}{2}}$ with regard to the given cell average $\overline{h}(x_i)$. This means that for each cell I_i, we need to find a polynomial $p_i(x)$ with a degree of at most $k-1$, such that it is a k-th order accurate approximation to the function $h(x)$ inside I_i

$$p_i(x) = h(x) + \mathcal{O}(\Delta x^k), x \in I_i, i = 1, \ldots, N. \tag{2.10}$$

We will not look into boundary conditions at this moment, therefore we assume that the values $\overline{h}_i$ are also available for $i \leq 0$ and $i > N$ if needed.

Assume we know the location I_i and the order of accuracy k, then we choose our stencil S based on s cells to the left, t cells to the right and I_i itself if $s, t \geq 0$ with $s + t + 1 = k$: $S(i) = \{I_{i-s}, \ldots, I_{i+t}\}$. In this case, we can find a unique polynomial with

a degree of at most $k-1 = s+t$, denoted by $p_i(x)$ whose cell average in each of our stencils $S(i)$ matches that of $h(x)$

$$\frac{1}{\Delta x}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} p(\psi)d\psi = \overline{h}_j, \quad j = i-s,\dots,i+t. \tag{2.11}$$

If the function $h(x)$ is smooth on $S(i)$, the polynomial $p_i(x)$ gives an approximation of order k. Now we want to approximate the values of $h(x)$ at the cell boundaries. We can do this by noticing that the mappings from the known cell averages $\overline{h}_i$ in the stencil $S(i)$ to the values $h^-_{i+\frac{1}{2}}$ and $h^+_{i-\frac{1}{2}}$ are linear. Hence constants c_{rj} and $\tilde{c_{rj}}$ exists, which are dependent on the left shift of the stencil s, the order of accuracy k and on the cell sizes Δx such that

$$v^-_{i+\frac{1}{2}} = \sum_{j=0}^{k-1} c_{rj}\cdot\overline{h}_{i-s+j} \quad \wedge \quad v^+_{i-\frac{1}{2}} = \sum_{j=0}^{k-1} \tilde{c_{rj}}\cdot\overline{h}_{i-s+j}. \tag{2.12}$$

We note that this approximation does not depend on h itself. For further information on how to derive the values c_{rj} and $\tilde{c_{rj}}$ the reader is referred to [14].

In our case, we are interested in $k=3$. We will later combine those three third order approximations with non-linear weights to a fifth order approximation. This gives the approximation

$$\begin{aligned} h^-_{i+\frac{1}{2}} &= p_i(x_{i+\frac{1}{2}}) + \mathcal{O}(\Delta x^3) \\ h^+_{i-\frac{1}{2}} &= p_i(x_{i-\frac{1}{2}}) + \mathcal{O}(\Delta x^3), i = 1,\dots,N. \end{aligned} \tag{2.13}$$

To derive the approximation, we need to choose a stencil $S_i = \{I_{i-2}, I_{i-1}, I_i\}$ first. For this stencil, we can find a unique polynomial of degree at most two which reconstructs the function $h(x)$ on this stencil. We denote this polynomial by $p_1(x)$. For $k=3$ and the three stencils S_1, S_2, S_3, we get the following approximations from [14]

$$\begin{aligned} h^1_{i+/-\frac{1}{2}} &= \frac{1}{3}\cdot\overline{h}_{i-2} - \frac{7}{6}\cdot\overline{h}_{i-1} + \frac{11}{6}\cdot\overline{h}_i, \text{ for } S_1 = \{I_{i-2}, I_{i-1}, I_i\} \\ h^2_{i+/-\frac{1}{2}} &= -\frac{1}{6}\cdot\overline{h}_{i-1} + \frac{5}{6}\cdot\overline{h}_i + \frac{1}{3}\cdot\overline{h}_{i+1}, \text{ for } S_2 = \{I_{i-1}, I_i, I_{i+1}\} \\ h^3_{i+/-\frac{1}{2}} &= \frac{1}{3}\cdot\overline{h}_i + \frac{5}{6}\cdot\overline{h}_{i+1} - \frac{1}{6}\cdot\overline{h}_{i+2}, \text{ for } S_3 = \{I_i, I_{i+1}, I_{i+2}\}. \end{aligned} \tag{2.14}$$

Now, to derive the desired fifth order approximation of the numerical flux $\hat{f}_{i+\frac{1}{2}}$ and $\hat{f}_{i-\frac{1}{2}}$, we can combine the three stencils using a linear convex combination of the three

third order approximations $h^1_{i+\frac{1}{2}}, h^2_{i+\frac{1}{2}}$ and $h^3_{i+\frac{1}{2}}$

$$\begin{aligned} h^-_{i+\frac{1}{2}} &= d_2 \cdot h^1_{i+\frac{1}{2}} + d_1 \cdot h^2_{i+\frac{1}{2}} + d_0 \cdot h^3_{i+\frac{1}{2}} \\ h^+_{i-\frac{1}{2}} &= d_0 \cdot h^1_{i-\frac{1}{2}} + d_1 \cdot h^2_{i-\frac{1}{2}} + d_2 \cdot h^3_{i-\frac{1}{2}}, \end{aligned} \tag{2.15}$$

with $d_0 = \frac{3}{10}$, $d_1 = \frac{3}{5}$ and $d_2 = \frac{1}{10}$. The WENO method uses the same idea, but derives non-linear weights w_j. These non-linear weights are used to achieve a good approximation even if the function $h(x)$ has discontinuities on the stencil $S = \{S_1, S_2, S_3\}$. To guarantee a fifth order approximation for a smooth $h(x)$ and avoid oscillatory behaviour while ensuring at least a third order accuracy for non-smooth functions, these weights have to fulfill the following two conditions:

1. $w_j = d_j + \mathcal{O}(\Delta x^2) \quad \forall j$ if $h(x)$ is smooth on S.

2. $w_j \approx 0$ if $h(x)$ has a discontinuity on S_j, but is smooth in at least one of the other two stencils.

In [14], we can find the following formulas for the non-linear weights, including smoothness parameters β. We start by stating the formulas for the smoothness parameters

$$\begin{aligned} \beta_0 &= \frac{13}{12} \cdot (\overline{h}_i - 2 \cdot \overline{h}_{i+1} + \overline{h}_{i+2})^2 + \frac{1}{4} \cdot (3 \cdot \overline{h}_i - 4 \cdot \overline{h}_{i+1} + \overline{h}_{i+2})^2 \\ \beta_1 &= \frac{13}{12} \cdot (\overline{h}_{i-1} - 2 \cdot \overline{h}_i + \overline{h}_{i+1})^2 + \frac{1}{4} \cdot (\overline{h}_{i-1} - \overline{h}_{i+1})^2 \\ \beta_2 &= \frac{13}{12} \cdot (\overline{h}_{i-2} - 2 \cdot \overline{h}_{i-1} + \overline{h}_i)^2 + \frac{1}{4} \cdot (\overline{h}_{i-2} - 4 \cdot \overline{h}_{i-1} + 3 \cdot \overline{h}_i)^2. \end{aligned} \tag{2.16}$$

Now we can write down the formulas for the weights

$$\begin{aligned} \alpha_0 &= \frac{d_0}{(\epsilon + \beta_0)^2}, \qquad \tilde{\alpha_0} = \frac{d_2}{(\epsilon + \beta_0)^2} \\ \alpha_1 &= \frac{d_1}{(\epsilon + \beta_1)^2}, \qquad \tilde{\alpha_1} = \frac{d_1}{(\epsilon + \beta_1)^2} \\ \alpha_2 &= \frac{d_2}{(\epsilon + \beta_2)^2}, \qquad \tilde{\alpha_2} = \frac{d_0}{(\epsilon + \beta_2)^2}. \end{aligned} \tag{2.17}$$

The last step is to normalize the weights

$$\begin{aligned} w_0 &= \frac{\alpha_0}{\alpha_0+\alpha_1+\alpha_2}, & \tilde{w}_0 &= \frac{\tilde{\alpha_0}}{\tilde{\alpha_0}+\tilde{\alpha_1}+\tilde{\alpha_2}} \\ w_1 &= \frac{\alpha_1}{\alpha_0+\alpha_1+\alpha_2}, & \tilde{w}_1 &= \frac{\tilde{\alpha_1}}{\tilde{\alpha_0}+\tilde{\alpha_1}+\tilde{\alpha_2}} \\ w_2 &= \frac{\alpha_2}{\alpha_0+\alpha_1+\alpha_2}, & \tilde{w}_2 &= \frac{\tilde{\alpha_2}}{\tilde{\alpha_0}+\tilde{\alpha_1}+\tilde{\alpha_2}}. \end{aligned} \tag{2.18}$$

Finally, we get the wanted approximation

$$\begin{aligned} h^-_{i+\frac{1}{2}} &= w_0 \cdot h^3_{i+\frac{1}{2}} + w_1 \cdot h^2_{i+\frac{1}{2}} + w_2 \cdot h^1_{i+\frac{1}{2}} \\ h^+_{i+\frac{1}{2}} = h^+_{j-\frac{1}{2}} &= \tilde{w}_0 \cdot h^3_{j-\frac{1}{2}} + \tilde{w}_1 \cdot h^2_{j-\frac{1}{2}} + \tilde{w}_2 \cdot h^1_{j-\frac{1}{2}}, \text{ for } j = i+1. \end{aligned} \tag{2.19}$$

And therefore

$$\hat{f}_{i+\frac{1}{2}} = \hat{f}^-_{i+\frac{1}{2}} + \hat{f}^+_{i+\frac{1}{2}} = h^-_{i+\frac{1}{2}} + h^+_{i+\frac{1}{2}} \tag{2.20}$$

and in the same way

$$\hat{f}_{i-\frac{1}{2}} = h^-_{i-\frac{1}{2}} + h^+_{i-\frac{1}{2}}. \tag{2.21}$$

When using flux splitting, we derive $\overline{h}_i$ in the following way: $\overline{h}_i = f^+(u_i)$ for all i in S^+. This gives us $h^-_{i+\frac{1}{2}}$ when using the WENO reconstruction procedure. To derive $h^+_{i+\frac{1}{2}}$, we choose $\overline{h}_i = f^-(u_i)$ for all i in S^-. To ensure stability via upwinding, S^- should be biased one stencil to the right. This gives us the stencil $S = \{I_{i-2}, I_{i-1}, I_i, I_{i+1}, I_{i+2}, I_{i+3}\}$.

2.3 Time Discretization Method

We recall that after discretizing the PDE in space via WENO, we derive a semi-discrete problem that can be discretized in time by an ODE solver. Because of its non-oscillatory behaviour, the recommended solver is the third order total variation diminishing (TVD) Runge-Kutta (RK) method. This method works on hyperbolic problems with discontinous solutions and maintains stability in the total variation seminorm of the first order forward Euler method with the same discretization in space.

This means that, for a system of ODEs of the form $u_t = L(u)$, the total variation of the numerical solution

$$TV(u) = \sum_j |u^{j+1} - u^j| \tag{2.22}$$

does not increase:

$$TV(u^{n+1}) \leq TV(u^n), \tag{2.23}$$

for a first order in time Euler forward stepping

$$u^{n+1} = u^n + \Delta t \cdot L(u^n) \tag{2.24}$$

under suitable restriction on Δt

$$\Delta t \leq \Delta t_1, \tag{2.25}$$

for some Δt_1 and where L is the WENO spatial operator. The WENO spatial operator is a fifth order approximation of $\mathcal{L}(u)$ which comes from the abstract form of equation 2.1

$$u_t = \mathcal{L}(u). \tag{2.26}$$

Therefore, the following counts for h the maximum mesh size of the chosen grid and a smooth u

$$L(u) = \mathcal{L}(u) + \mathcal{O}(h^5). \tag{2.27}$$

For the high order in time Runge-Kutta methods we want the same stability result 2.23 under a different restriction on Δt

$$\Delta t \leq c \cdot \Delta t_1, \tag{2.28}$$

where c is the Courant-Friedrichs-Lewy (CFL) coefficient for the high order time discretization. The general explicit Runge-Kutta method is written in the form

$$u^i = \sum_{k=0}^{i-1} (\alpha_{ik} \cdot u^k + \Delta t \cdot \beta_{ik} \cdot L(u^k)) \quad i = 1, \ldots, m, \quad u^0 = u^n, \quad u^m = u^{n+1}. \tag{2.29}$$

If all coefficients are non-negative $\alpha_{ik}, \beta_{ik} \geq 0$, then the equation 2.29 becomes a convex combination of the Euler forwards operator, where Δt is replaced by $\frac{\beta_{ik}}{\alpha_{ik}} \cdot \Delta t$, since by consistency $\sum_{k=0}^{i-1} \alpha_{ik} = 1$. This yields the following lemma for the CFL coefficient.

Lemma 2.2 (from [16]). *The Runge-Kutta method 2.29 is TVD under the CFL coefficient 2.28*

$$c = \min_{i,k} \frac{\alpha_{ik}}{\beta_{ik}}$$

provided that $\alpha_{ik} \geq 0, \beta_{ik} > 0$.

The optimal third order Runge-Kutta method satisfying this Lemma is given in [16]

$$\begin{aligned} u^1 &= u^n + \Delta t \cdot L(u^n) \\ u^2 &= \frac{3}{4} \cdot u^n + \frac{1}{4} \cdot u^1 + \frac{1}{4} \cdot \Delta t \cdot L(u^1) \\ u^{n+1} = u^3 &= \frac{1}{3} \cdot u^n + \frac{2}{3} \cdot u^2 + \frac{2}{3} \cdot \Delta t \cdot L(u^2), \end{aligned} \tag{2.30}$$

with CFL coeffcient $c = 1$.

For the case that a right-hand side term is added to equation 2.1, the Runge-Kutta method has to be adapted. We now want to solve an equation of the form

$$u_t(x,t) + f(u(x,t))_x = g(t). \tag{2.31}$$

We can use our known Runge-Kutta method, but we have to pay attention at which points we evaluate the right-hand side

$$\begin{aligned} u^1 &= u^n + \Delta t \cdot (L(u^n) + g(t)) \\ u^2 &= \frac{3}{4} \cdot u^n + \frac{1}{4} \cdot u^1 + \frac{1}{4} \cdot \Delta t \cdot (L(u^1) + g(t + \Delta t)) \\ u^{n+1} = u^3 &= \frac{1}{3} \cdot u^n + \frac{2}{3} \cdot u^2 + \frac{2}{3} \cdot \Delta t \cdot (L(u^2) + g(t + \frac{1}{2} \cdot \Delta t)). \end{aligned} \tag{2.32}$$

Another equation which we want to solve later on is if the right hand side is a solution dependent source term

$$u_t(x,t) + f(u(x,t))_x = g(u). \tag{2.33}$$

Here, the adaption to the Runge-Kutta method ia a little bit simpler, because we do not have to pay attention to the evaluation points anymore

$$\begin{aligned} u^1 &= u^n + \Delta t \cdot (L(u^n) + g(u^n)) \\ u^2 &= \frac{3}{4} \cdot u^n + \frac{1}{4} \cdot u^1 + \frac{1}{4} \cdot \Delta t \cdot (L(u^1) + g(u^1)) \\ u^{n+1} = u^3 &= \frac{1}{3} \cdot u^n + \frac{2}{3} \cdot u^2 + \frac{2}{3} \cdot \Delta t \cdot (L(u^2) + g(u^2)). \end{aligned} \tag{2.34}$$

2.4 2D Conservation Law

Given

$$u_t(x,y,t) + f_x(u(x,y,t)) + g_y(u(x,y,t)) = 0 \tag{2.35}$$

with suitable initial and boundary conditions. Similar to the one-dimensional case, our goal is to use the method of lines to discretize the PDE in space and use a suiting time discretization on the resulting ODE.

As in the one-dimensional case, we start by discretizing the domain into cells. In the two-dimensional case the domain is a rectangle defined by

$$\begin{gathered} [a,b] \times [c,d] \\ a = x_{\frac{1}{2}} < x_{\frac{3}{2}} < \cdots < x_{N_x-\frac{1}{2}} < x_{N_x+\frac{1}{2}} = b \\ c = y_{\frac{1}{2}} < y_{\frac{3}{2}} < \cdots < y_{N_y-\frac{1}{2}} < y_{N_y+\frac{1}{2}} = d. \end{gathered} \tag{2.36}$$

We build the following cells

$$I_{ij} = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{i-\frac{1}{2}}, y_{i+\frac{1}{2}}] \text{ for } 1 \le i \le N_x, 1 \le j \le N_y \tag{2.37}$$

with N_x and N_y the maximal number of cells in $x-$ and $y-$direction respectively. The centers of the cells are

$$(x_i, y_j) \text{ with } x_i \equiv \frac{1}{2} \cdot (x_{i-\frac{1}{2}} + x_{i+\frac{1}{2}}) \text{ and } y_j \equiv \frac{1}{2} \cdot (y_{j-\frac{1}{2}} + y_{j+\frac{1}{2}}). \tag{2.38}$$

Furthermore, we define the spacing as before

$$\begin{aligned} \Delta x_i &\equiv x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}, \quad i = 1, 2, \ldots, N_x \\ \Delta y_j &\equiv y_{j+\frac{1}{2}} - y_{j-\frac{1}{2}}, \quad i = 1, 2, \ldots, N_y. \end{aligned} \tag{2.39}$$

We will use a uniform mesh. Therefore, $\Delta x = \Delta x_i \quad \forall i$ and $\Delta y = \Delta y_j \quad \forall j$. As in the one-dimensional case, we are using finite differences and a uniform mesh to derive the following discretization

$$\frac{\partial u_{ij}(t)}{\partial t} = -\frac{1}{\Delta x} \cdot (\hat{f}_{i+\frac{1}{2},j} - \hat{f}_{i-\frac{1}{2},j}) - \frac{1}{\Delta y} \cdot (\hat{g}_{i,j+\frac{1}{2}} - \hat{g}_{i,j-\frac{1}{2}}) \tag{2.40}$$

where $u_{ij}(t) = u(x_i, y_j, t)$. With this discretization derived, we can use the one-dimensional WENO approximation procedure to determine the numerical fluxes $\hat{f}_{i+\frac{1}{2},j}$ and $\hat{g}_{i,j+\frac{1}{2}}$. We

do so by fixing the $x-$ or $y-$direction and obtaining the known one-dimensional problem. For $\hat{f}_{i+\frac{1}{2},j}$, we fix j and use the one-dimensional reconstruction on $v(x) = f(u(x, y_j, t))$. Likewise, for $\hat{g}_{i,j+\frac{1}{2}}$ we fix i and use the one-dimensional WENO procedure with $v(y) = g(u(x_i, y, t))$. $\hat{f}_{i-\frac{1}{2},j}$ and $\hat{g}_{i,j-\frac{1}{2}}$ are obtained in a similar way. Using this kind of splitting into one-dimensional problems yields a loss in the approximation order, because the reconstruction methods cannot ensure the previous approximation orders anymore [14].
All the explanations and methods that work in the one-dimensional case, e. g. flux splitting, can be applied in the two-dimensional case as well. They just have to be applied to each dimension seperately. The same applies for higher dimensions.

2.5 Boundary Conditions

To simplify, we have so far assumed that there are no boundaries and that we know the values at all the required points. Depending on the application and the problem, there are different possibilities to set the boundary conditions. First of all, because we are using the fifth order WENO scheme as spatial solver, we have to set three ghost cells on each boundary.
In many application problems, there are physical boundaries which define the numerical boundary conditions. For example, if the boundary describes a solid wall, reflecting boundary conditions have to be applied. In cases where physical boundaries apply, the ghost cells can be set symmetric or anti-symmetric, using values of the corresponding points inside the computational domain.
When setting the boundary conditions, reflecting the physical world is not the only goal that has to be met. The boundary conditions should further ensure stability and accuracy. For applications with in- and outflow, there has been research, especially on the inflow boundary. The outflow boundary can be set by a standard extrapolation of suitable order and accuracy to derive a stable scheme [5]. For the inflow boundary, it is more challenging to find a stable scheme. A procedure called inverse Lax-Wendroff (ILW) has been invented by Hung, Shu and Zang in 2008 [7]. The idea behind this method is to repeatedly use the PDEs to convert the normal spatial derivatives to time and tangential

derivatives of the given boundary conditions. For a detailed description and application, we refer the reader to [19]. Another possibility to set the boundarys and ghost cells is to use the exact solution. Of course, this is only possible when the exact solution is known, for example for convergence tests or when we can measure the values at the ghost cells. This can be the case when we model an experiment and hence are able to ensure the boundary conditions we want. In this thesis, we are using either periodic boundary conditions or the exact solution at the inflow boundary and exact or constant values at the outflow boundary. In two-dimensional cases where there is only unidirectional flow, the boundary values on the other two boundaries without flow are also set constant.

3 Verification and Numerical Experiments

In this section, we will test and analyse our presented PDE solver on different advection equations. We already know that the WENO solver has a fifth order accuracy convergence and is paired with a third order Runge-Kutta solver. Therefore, we expect to see a third order convergence in general. But for accuracy tests, we set $\Delta t = (\Delta x)^{\frac{5}{3}}$ for the third order Runge-Kutta time integration in order to see a fifth order accuracy in time as well as in space. This condition also fulfills the CFL condition of the solver given by $c_{CFL} = 1$. We will use the ℓ_2 and ℓ_∞ norm for the convergence analysis. The ℓ_2 error and ℓ_∞ error are defined as $(\sum_{i,j:(x_i,y_j)\in\mathcal{D}} |e_{i,j}|^2)^{\frac{1}{2}}$ and $\max_{i,j:(x_i,y_j)\in\mathcal{D}} |e_{i,j}|$ respectively. Here, $e_{i,j}$ denotes the error at grid point (x_i, y_j) and $\mathcal{D}$ is the region where the errors are measured. $\mathcal{D} = \Omega$ with Ω not including the ghost cells. If not indicated otherwise, we set Ω to be the interval $[1, 3]$ in the one-dimensional case. Furthermore, if not stated otherwise, the advection velocity is set as $v_0 = 1$.

3.1 Verification: 1D problems

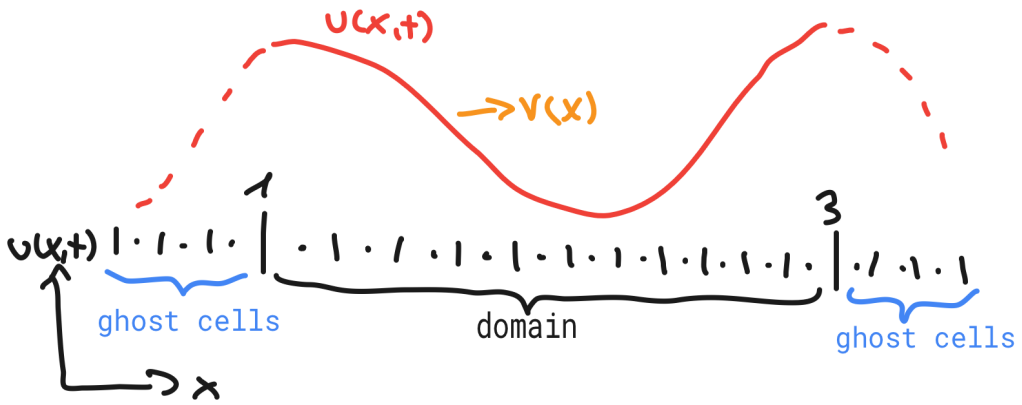


Figure 2: Set-up for the verifications

Figure 2 shows the general set-up we are going to use to validate our code in the

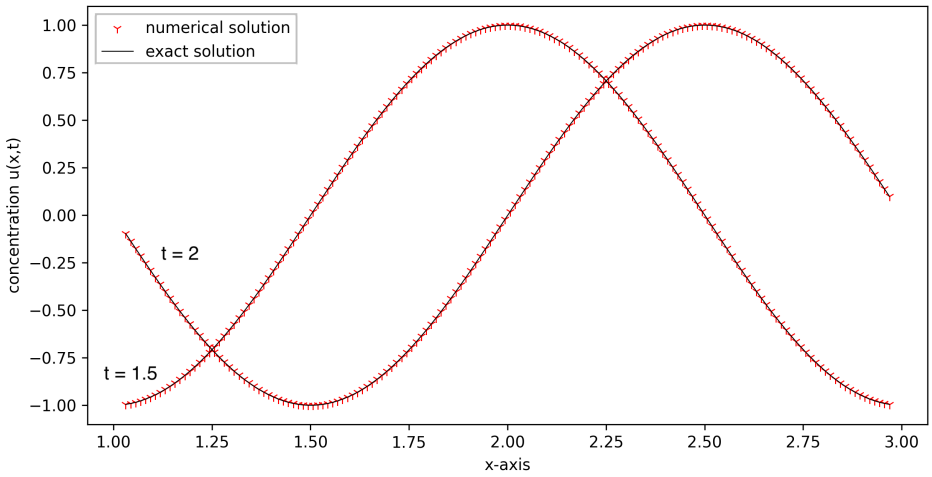
following chapter.

Example 3.1. We start with the simplest form of the advection equation

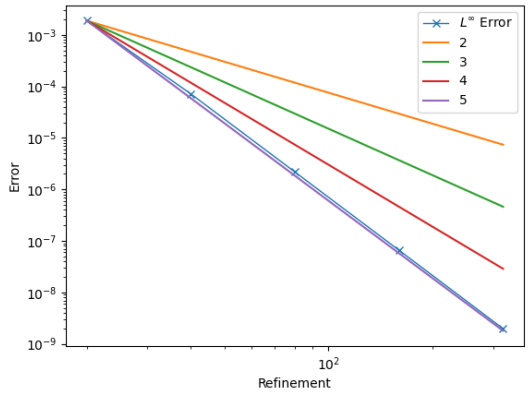
$$\begin{aligned} u_t + v_0 \cdot u_x = 0 \qquad & x \in \Omega, t > 0 \\ u(x,0) = \phi(x) \quad & x \in \Omega \end{aligned} \tag{3.1}$$

The left boundary $x = 1$ is an inflow boundary where a boundary condition is prescribed. The right boundary $x = 3$ is an outflow boundary where no boundary condition is needed. To obtain the expected convergence, we set the ghost cells at the inflow and outflow boundary to be the exact solution. The analytic solution is

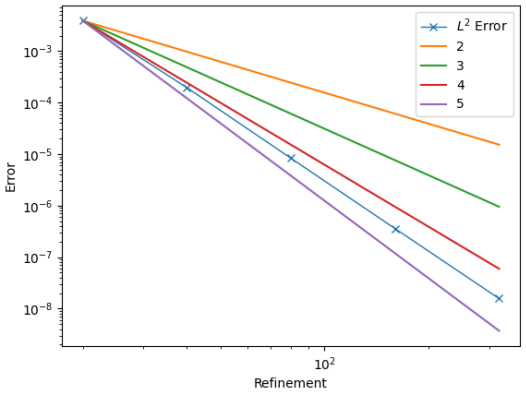
$$u(x,t) = \phi(x - v_0 \cdot t). \tag{3.2}$$



(a) Analytic and numeric solution



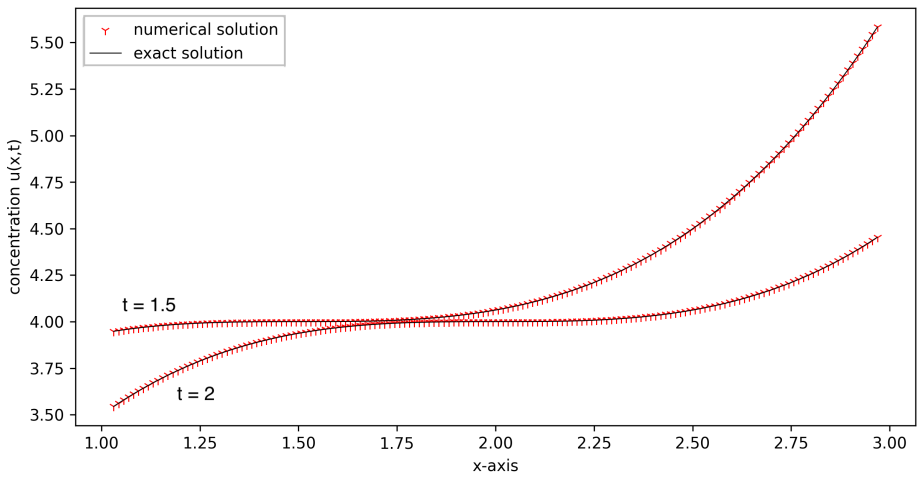
(b) Convergence plot using the ℓ_∞ norm



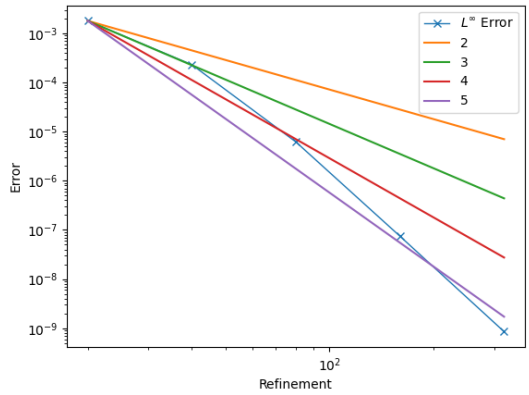
(c) Convergence plot using the ℓ_2 norm

Figure 3: Solution of the advection equation 3.1 with initial condition $\phi(x) = \sin(\pi \cdot x)$ and the convergence plots for the numerical solver

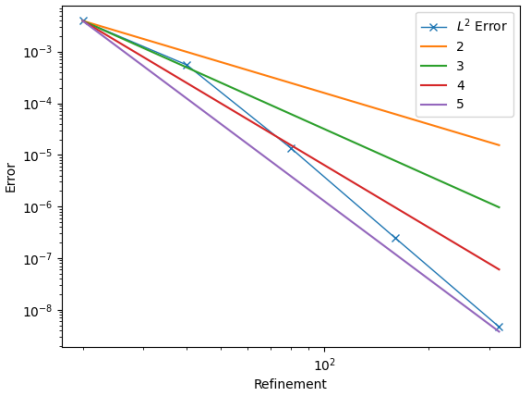
The numerical and analytic solutions for two different initial conditions are plotted in 3a and 4a. The errors and convergence at $t = 2$ can be seen in Figures 3b and 4b. For both initial conditions $\phi_1(x) = \sin(\pi \cdot x)$ and $\phi_2(x) = \frac{1}{2} \cdot x^3 + 4$, we can clearly see the desired fifth order convergence.



(a) Analytic and numeric solution



(b) Convergence plot using the ℓ_∞ norm



(c) Convergence plot using the ℓ_2 norm

Figure 4: Solution of the advection equation 3.1 with initial condition $\phi(x) = \frac{1}{2}x^3 + 4$ and the convergence plots for the numerical solver

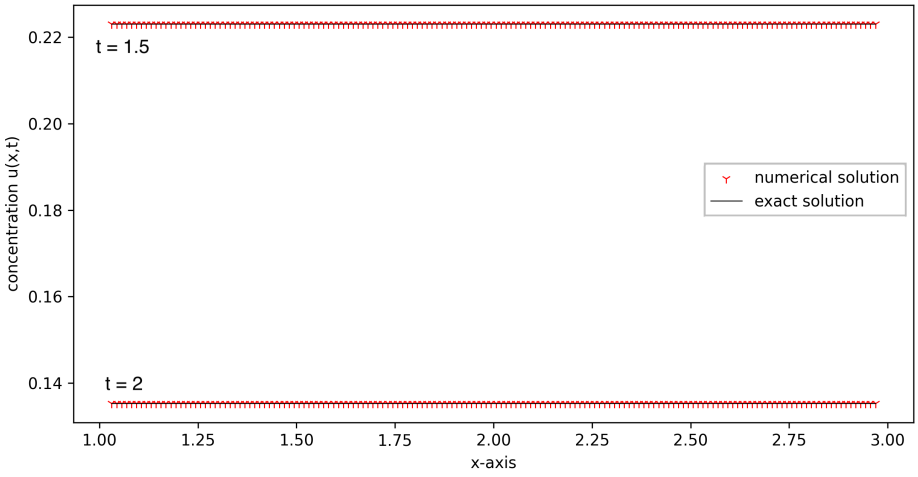
Example 3.2. The next step towards our wanted model 1.27 is to introduce a solution-dependent source term

$$\begin{aligned} u_t + v_0 \cdot u_x &= -c \cdot u \qquad x \in \Omega, t > 0 \\ u(x,0) &= \phi(x) \qquad x \in \Omega. \end{aligned} \tag{3.3}$$

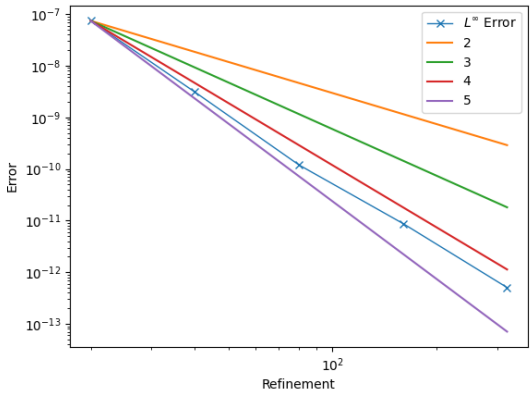
If not stated otherwise, the left boundary remains the inflow boundary and the right boundary remains the outflow boundary. Furthermore, we set the ghost cells on the

inflow and outflow boundary to be the exact solution. The exact solution is given by

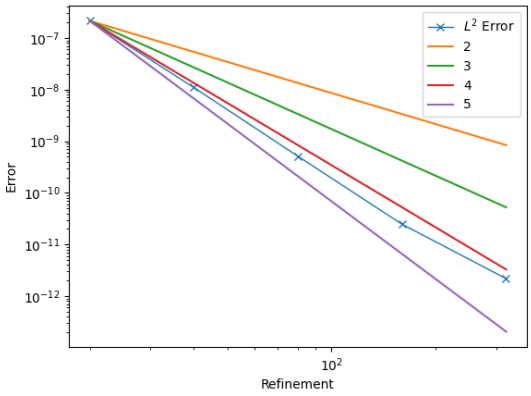
$$u(x,t) = e^{-c \cdot t} \cdot \phi(x - v_0 \cdot t). \tag{3.4}$$



(a) Analytic and numeric solution



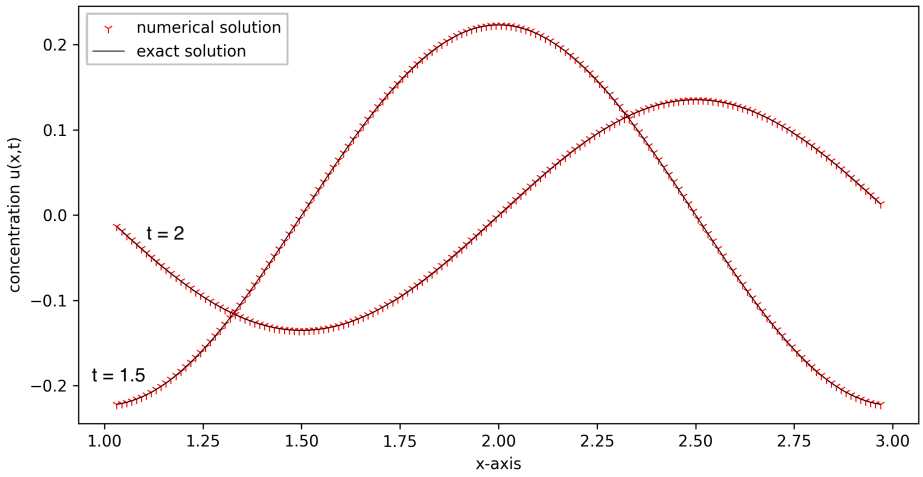
(b) Convergence plot using the ℓ_∞ norm



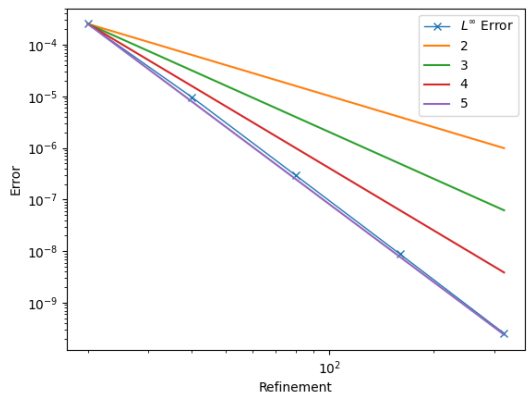
(c) Convergence plot using the ℓ_2 norm

Figure 5: Solution of the advection equation 3.3 with initial condition $\phi(x) = 1$ and the convergence plots for the numerical solver

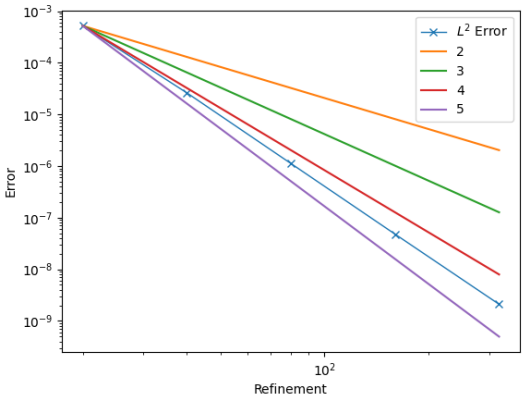
The solution and convergence at $t = 2$ for three different initial conditions can be seen in Figures 5, 6 and 7. For the initial conditions $\phi_2(x) = \sin(\pi \cdot x)$ and $\phi_3(x) = \frac{1}{2} \cdot x^3 + 4$, we can clearly see the desired fifth order convergence. For the constant initial condition



(a) Analytic and numeric solution



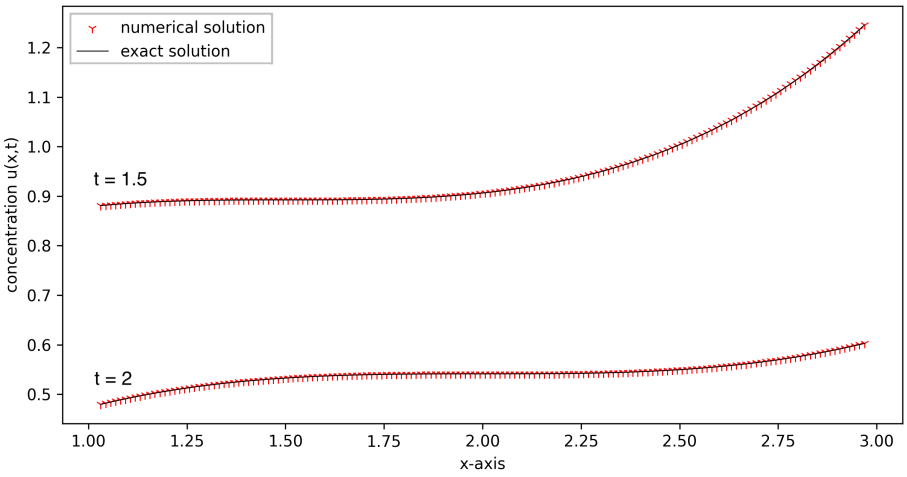
(b) Convergence plot using the ℓ_∞ norm



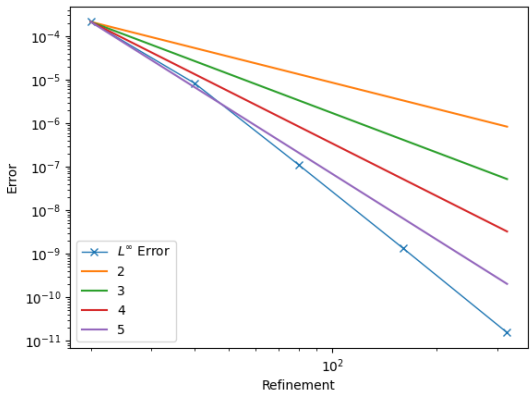
(c) Convergence plot using the ℓ_2 norm

Figure 6: Solution of the advection equation 3.3 with initial condition $\phi(x) = \sin(\pi \cdot x)$ and the convergence plots for the numerical solver

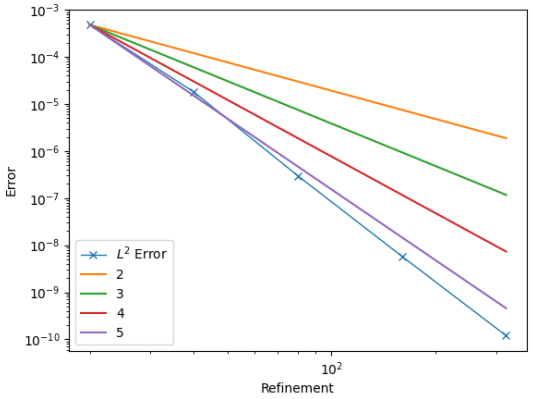
$\phi_1(x) = 1$ on the other hand, the convergence order is between three and four. This results from only having to solve an ODE when we have a constant initial condition. There is no change in space, therefore we can only see the convergence of our time solver, which is three. This shows that the expected convergence rate is also obtained for constant initial conditions.



(a) Analytic and numeric solution



(b) Convergence plot using the ℓ_∞ norm



(c) Convergence plot using the ℓ_2 norm

Figure 7: Solution of the advection equation 3.3 with initial condition $\phi(x) = \frac{1}{2}x^3 + 4$ and the convergence plots for the numerical solver

Example 3.3. The next adaptation to our model is introducing a right-hand side which is not solution-dependent. Therefore, we are introducing a source term which is t dependent.

$$\begin{aligned} u_t + v_0 \cdot u_x &= c \cdot e^{-(c+1)\cdot t} \qquad x \in \Omega, t > 0 \\ u(x,0) &= \phi(x) \qquad x \in \Omega. \end{aligned} \tag{3.5}$$

We use a right-hand side which is only dependent on t and not on x, because we can

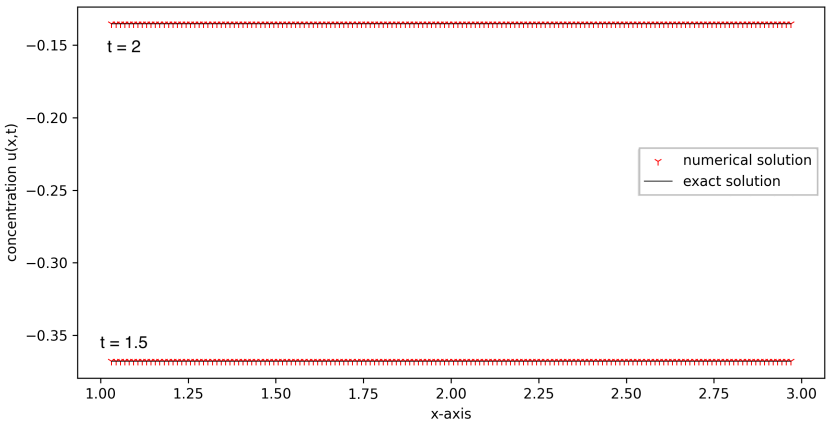
derive an analytic solution for this case. The analytic solution is given by

$$u(x,t) = -e^{-c \cdot t}, \tag{3.6}$$

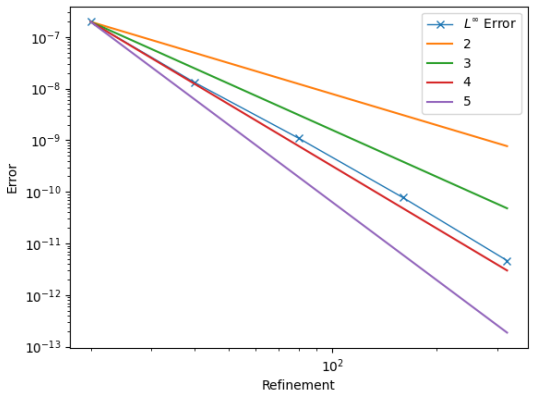
for the initial condition $\phi_1(x) = -1$ and by

$$u(x,t) = -e^{-c \cdot t} + \phi(x - v_0 \cdot t), \tag{3.7}$$

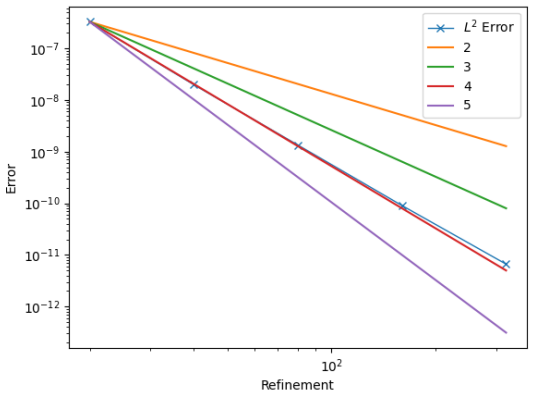
for the initial condition $\phi_2(x) = \sin(\pi \cdot x) - 1$.



(a) Analytic and numeric solution

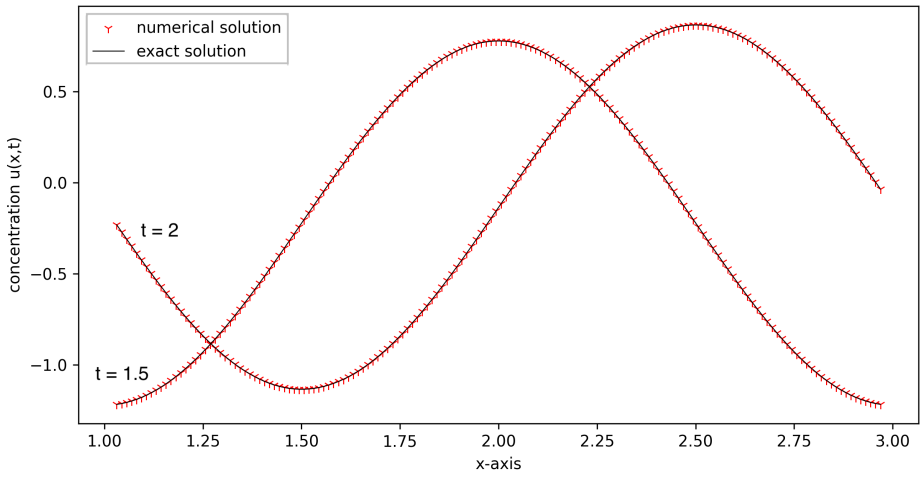


(b) Convergence plot using the ℓ_∞ norm

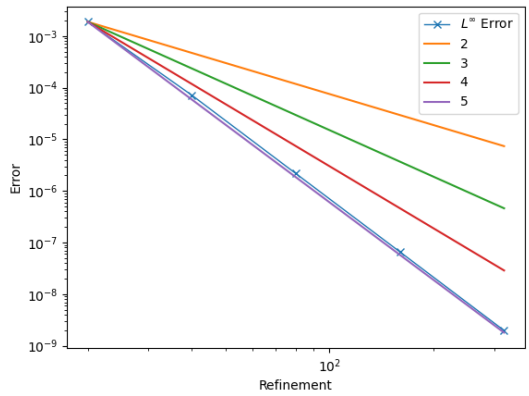


(c) Convergence plot using the ℓ_2 norm

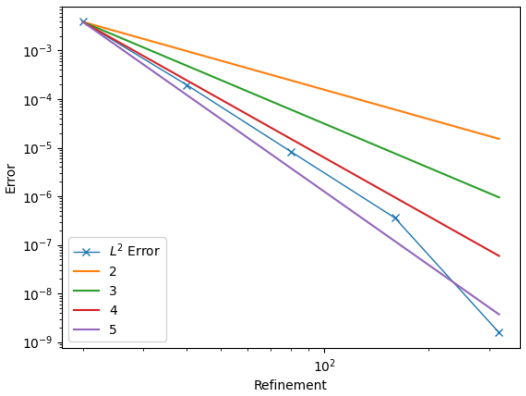
Figure 8: Solution of the advection equation 3.5 with initial condition $\phi(x) = -1$ and the convergence plots for the numerical solver



(a) Analytic and numeric solution



(b) Convergence plot using the ℓ_∞ norm



(c) Convergence plot using the ℓ_2 norm

Figure 9: Solution of the advection equation 3.5 with initial condition $\phi(x) = \sin(\pi \cdot x) - 1$ and the convergence plots for the numerical solver

The convergence rate and errors at $t = 2$ can be seen in Figures 8 and 9. As explained above, the constant initial condition $\phi_1(x) = -1$ does not present itself in fifth order convergence but in third. The second initial condition, $\phi_2(x) = \sin(\pi \cdot x) - 1$ shows the expected fifth order convergence. This example concludes the cases with a constant velocity.

Example 3.4. Now we introduce a space-dependent velocity. We have already taken a look at the analytical solution in Chapter 1. We use a space-dependent velocity that fulfills the conservation law because our solver needs to know the flow of the advection equation. The general advection equation fulfilling the conservation law is written down in the following equation

$$u_t + (f(u))_x = 0. \tag{3.8}$$

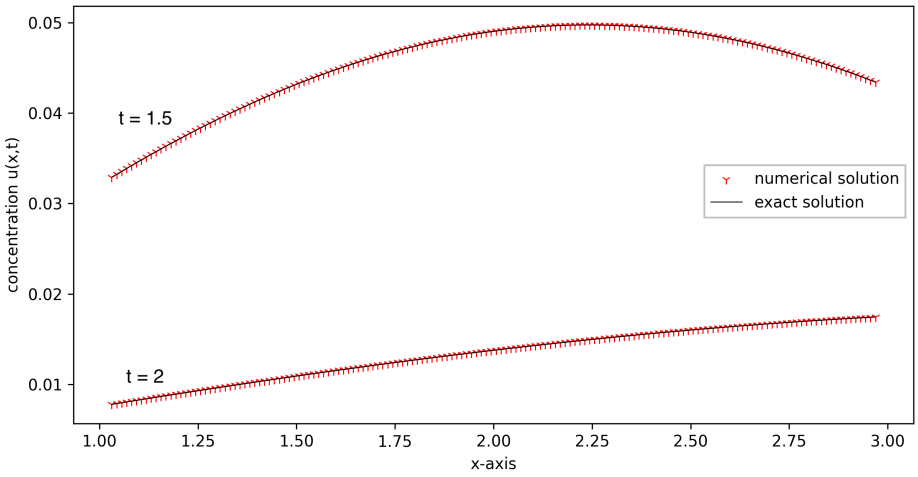
Now we set $f(u) = x \cdot u$ and introduce a solution-dependent source term again. This gives us the following problem

$$\begin{aligned} u_t + x \cdot u_x &= -(c+1) \cdot u \qquad x \in \Omega, t > 0 \\ u(x,0) &= \phi(x) \qquad x \in \Omega. \end{aligned} \tag{3.9}$$

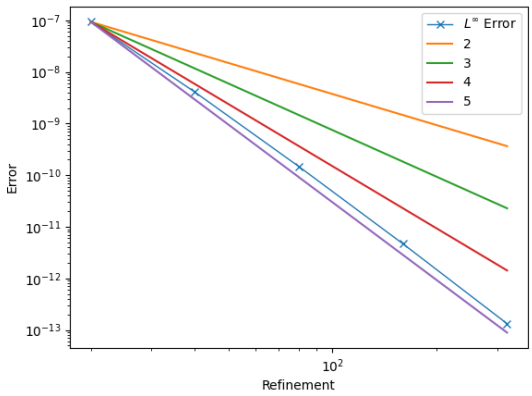
The analytic solution is given by

$$u(x,t) = \phi(x \cdot e^{-t}) \cdot e^{-(c+1)\cdot t}. \tag{3.10}$$

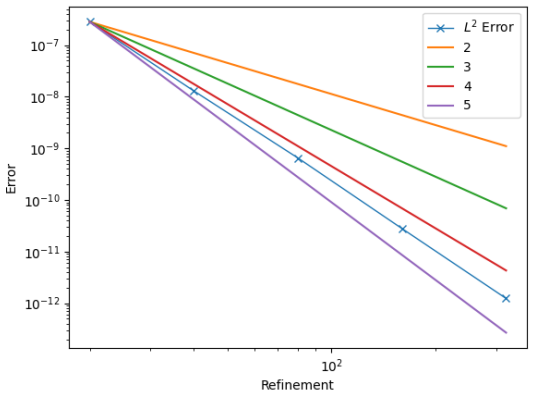
As in the previous examples, the solutions, convergence rates and errors at $t = 2$ can be found in Figures 10, 11 and 12. As expected, all three problems show a fifth order convergence.



(a) Analytic and numeric solution

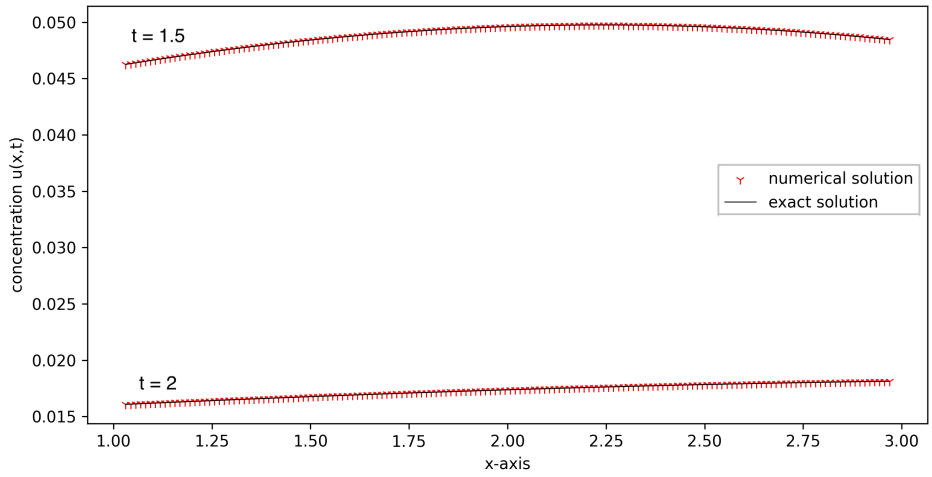


(b) Convergence plot using the ℓ_∞ norm

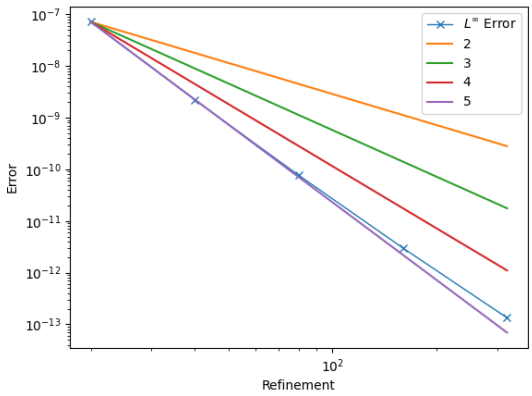


(c) Convergence plot withusing the ℓ_2 norm

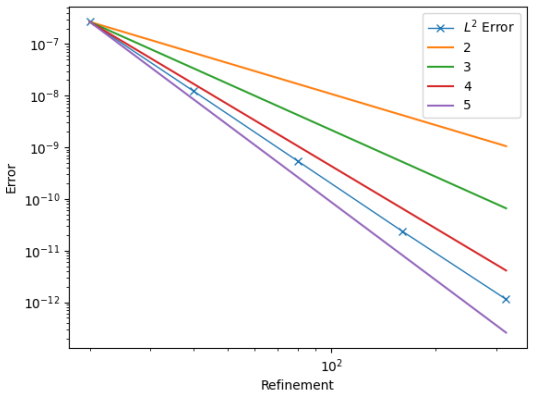
Figure 10: Solution of the advection equation 3.9 with initial condition $\phi(x) = \sin(\pi \cdot x)$ and the convergence plots for the numerical solver



(a) Analytic and numeric solution



(b) Convergence plot using the ℓ_∞ norm

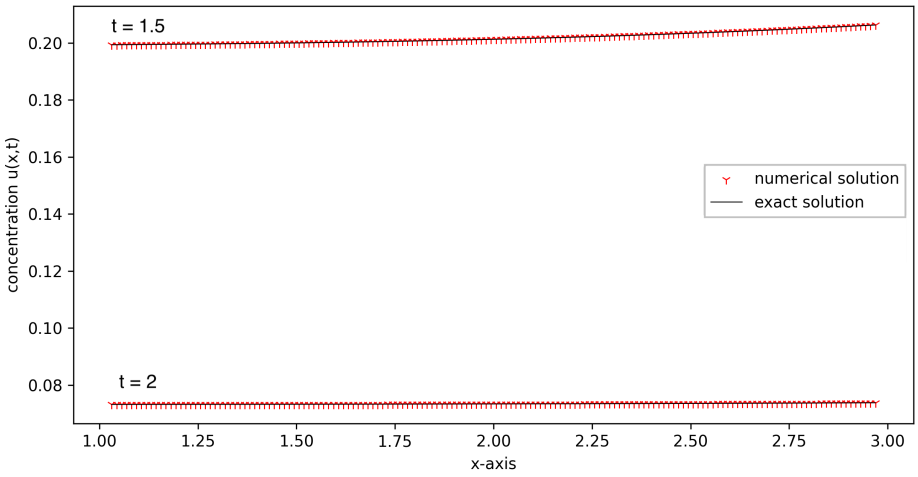


(c) Convergence plot using the ℓ_2 norm

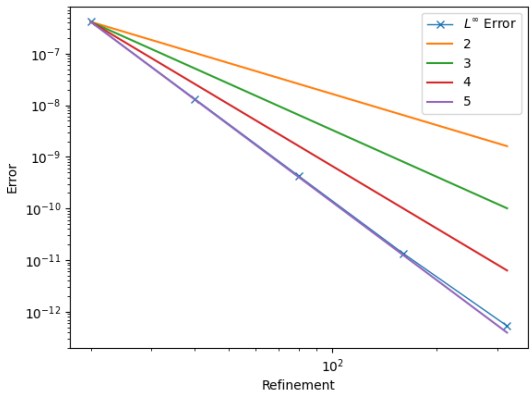
Figure 11: Solution of the advection equation 3.9 with initial condition $\phi(x) = e^{-(x-\frac{1}{2})^2}$ and the convergence plots for the numerical solver

Example 3.5. Before starting with system problems, we take a look at the space-dependent velocity advection equation with a t dependent source term. The problem is characterized as follows

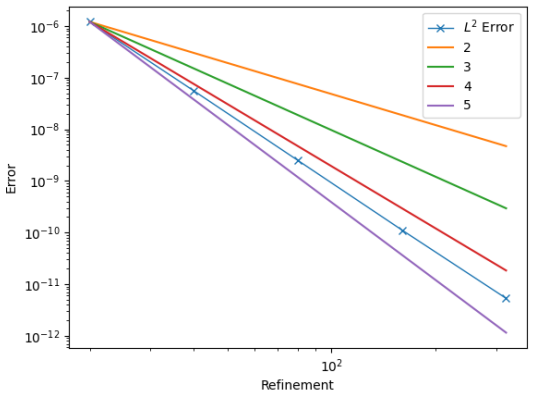
$$\begin{aligned} u_t + x \cdot u_x &= -u + c \cdot e^{-(c+1)\cdot t} \qquad x \in \Omega, t > 0 \\ u(x,0) &= \phi(x) \qquad x \in \Omega. \end{aligned} \tag{3.11}$$



(a) Analytic and numeric solution



(b) Convergence plot using the ℓ_∞ norm



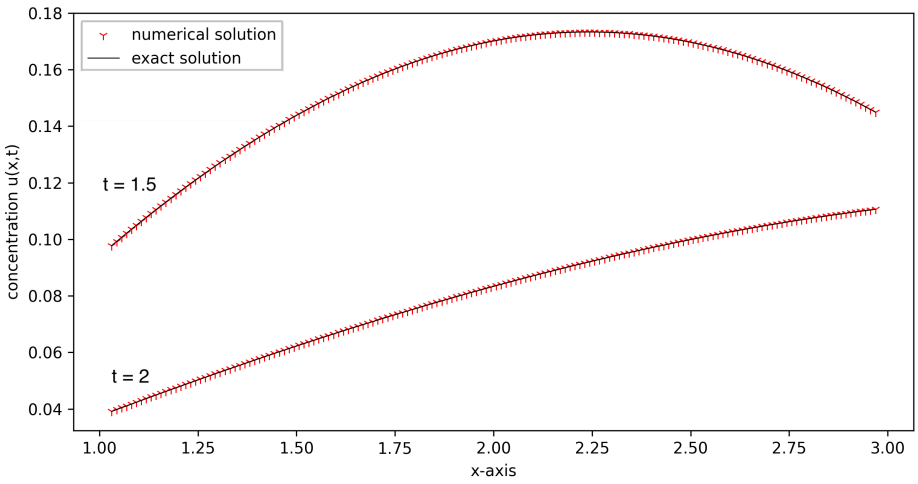
(c) Convergence plot using the ℓ_2 norm

Figure 12: Solution of the advection equation 3.9 with initial condition $\phi(x) = \frac{1}{2}x^3 + 4$ and the convergence plots for the numerical solver

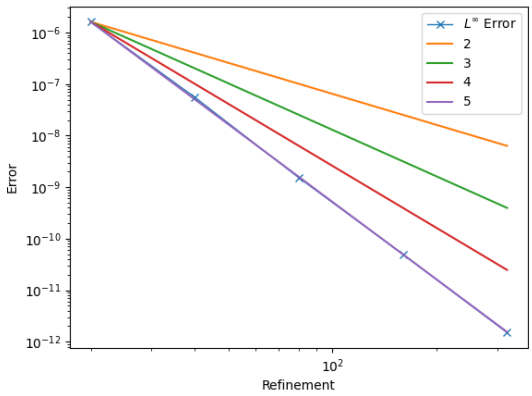
The analytic solution is given by

$$u(x,t) = -e^{-(c+1)\cdot t} + e^{-t} \cdot \sin(\pi \cdot (x \cdot e^{-t})) \tag{3.12}$$

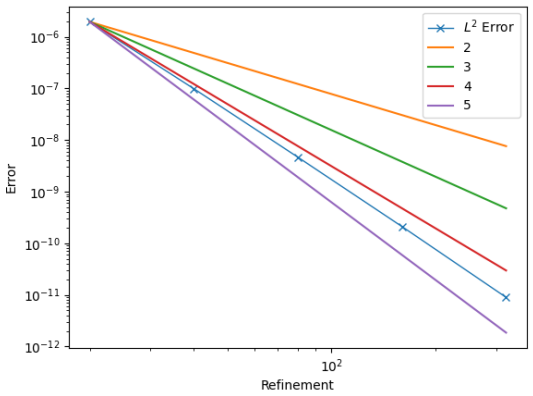
for the initial condition $\phi(x) = \sin(\pi \cdot x) - 1$.



(a) Analytic and numeric solution



(b) Convergence plot using the ℓ_∞ norm



(c) Convergence plot using the ℓ_2 norm

Figure 13: Solution of the advection equation 3.11 with initial condition $\phi(x) = \sin(\pi \cdot x) - 1$ and the convergence plots for the numerical solver

The solution, convergence rate and errors at $t = 2$ can be found in Figure 13. We can clearly see the fifth order convergence.

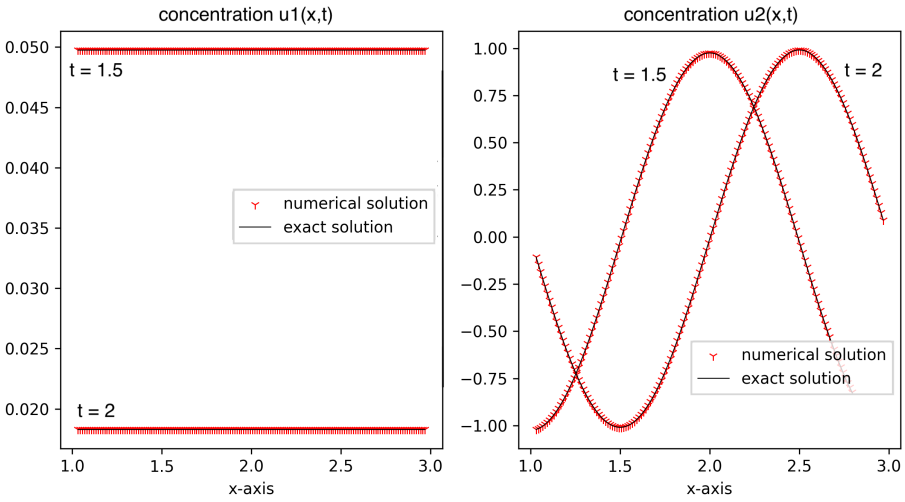
Example 3.6. Now we can have a look at a system of problems similar to the one we want to model. Here, we have two substances u^1 and u^2 flowing through a medium. u_1 flows at a space-dependent velocity and is converted into u^2 which flows at a constant velocity. The variables $u^1(x,t)$ and $u^2(x,t)$ represent the concentration of the substance at time t and space x. The problem is defined by the following equation

$$\begin{aligned} u_t^1 + x \cdot u_x^1 &= -(c+1) \cdot u^1 \qquad x \in \Omega, t > 0 \\ u_t^2 + v_0 \cdot u_x^2 &= c \cdot u^1 \qquad x \in \Omega, t > 0 \\ u^1(x,0) &= \phi(x) \\ u^2(x,0) &= \psi(x) \end{aligned} \tag{3.13}$$

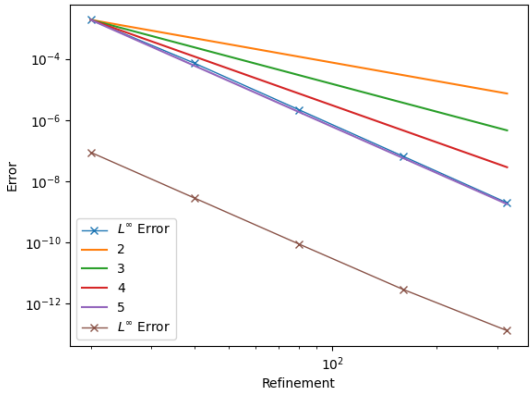
The analytic solution is given by

$$\begin{aligned} u^1(x,t) &= e^{-(c+1)\cdot t} \\ u^2(x,t) &= -\frac{c}{c+1} \cdot e^{-(c+1)\cdot t} + \sin(\pi \cdot (x - v_0 \cdot t)) \end{aligned} \tag{3.14}$$

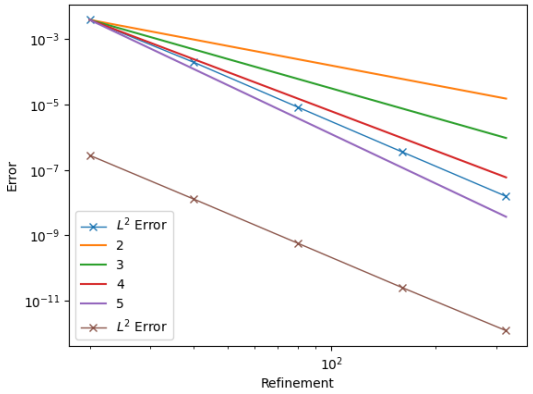
for initial conditions $\phi(x) = 1$ and $\psi(x) = -\frac{c}{c+1} + \sin(\pi \cdot x)$.



(a) Analytic and numeric solution



(b) Convergence plot using the ℓ_∞ norm



(c) Convergence plot using the ℓ_2 norm

Figure 14: Solution of the advection equation 3.13 with initial condition $\phi_1(x) = 1$ and $\phi_2(x) = -\frac{c}{c+1} + \sin(\pi \cdot x)$ and the convergence plots for the numerical solver

The errors and convergence rate at $t = 2$ can be seen in Figure 14. We can clearly see the fifth order convergence in both variables. We also see that the errors in u^2 are higher than those in u^1. This is according to expectations, because the calculations for u^2 use the calculations already made for u^1 and therefore their errors as well.

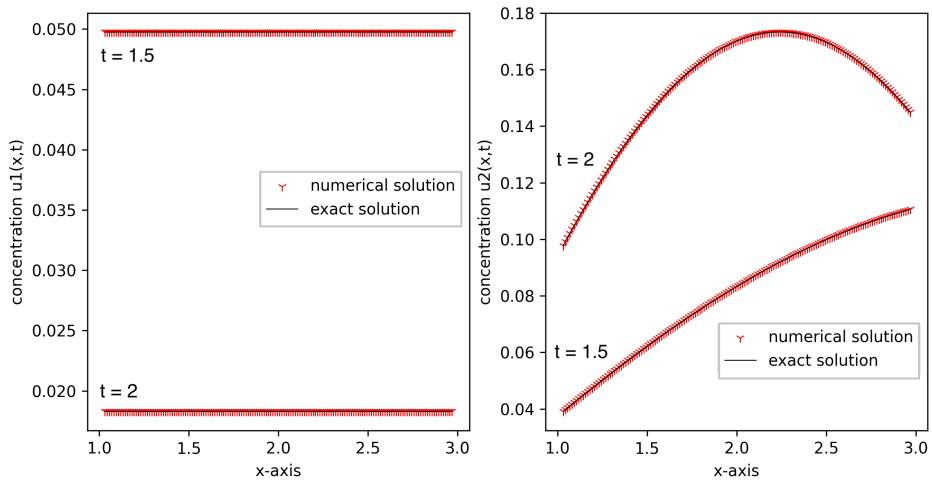
Example 3.7. For the last accuracy test with a given analytic solution, we take a look at a similar system as in Example 3.6. As before, we have two substances u^1 and u^2 flowing through a medium. u_1 flows at a space-dependent velocity and is converted into u^2. The difference is that u^2 now flows at a space-dependent velocity as well. The variables $u^1(x,t)$ and $u^2(x,t)$ represent the concentration of the substance at time t and space x. The problem is defined by the following equation

$$\begin{aligned} u^1_t + x \cdot u^1_x &= -(c+1) \cdot u^1 \qquad x \in \Omega, t > 0 \\ u^2_t + x \cdot u^2_x &= c \cdot u^1 - u^2 \qquad x \in \Omega, t > 0 \\ u^1(x,0) &= \phi(x) \\ u^2(x,0) &= \psi(x) \end{aligned} \tag{3.15}$$

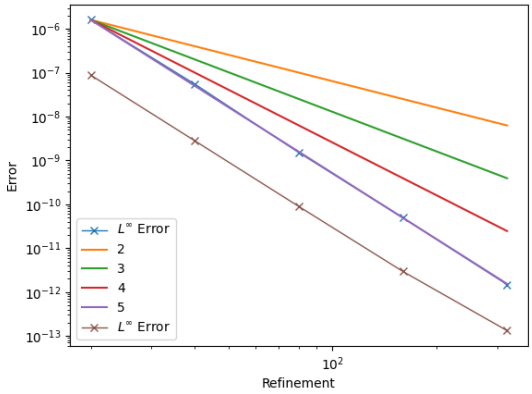
The analytic solution is given by

$$\begin{aligned} u^1(x,t) &= e^{-(c+1)\cdot t} \\ u^2(x,t) &= -e^{-(c+1)\cdot t} + e^{-t} \cdot \sin(\pi \cdot x \cdot e^{-t})) \end{aligned} \tag{3.16}$$

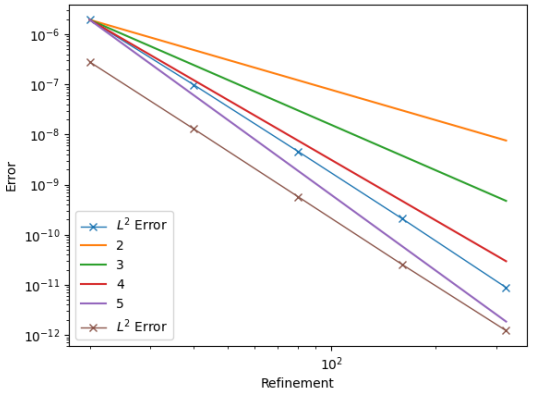
for initial conditions $\phi(x) = 1$ and $\psi(x) = \sin(\pi \cdot x) - 1$.



(a) Analytic and numeric solution



(b) Convergence plot using the ℓ_∞ norm



(c) Convergence plot using the ℓ_2 norm

Figure 15: Solution of the advection equation 3.15 with initial condition $\phi_1(x) = 1$ and $\phi_2(x) = -1 + \sin(\pi \cdot x)$ and the convergence plots for the numerical solver

The errors and convergence rate at $t = 2$ can be seen in Figure 15. We can clearly see the fifth order convergence in both variables.

3.2 Verification: 2D problems

For completeness, we also need to take a look at two-dimensional problems to check if our solver works as well as in the one-dimensional case. We set Ω to be $[1,3] \times [1,3]$, the rest of the parameters remain the same as in the one-dimensional examples. We use the same settings as in the one-dimensional case. Therfore, we are going to have an inflow boundary at $x = 1$ and an outflow boundary at $x = 3$. We set the flow in y-direction to be zero. This way, we can reuse the analytic solutions already derived for the one-dimensional case.

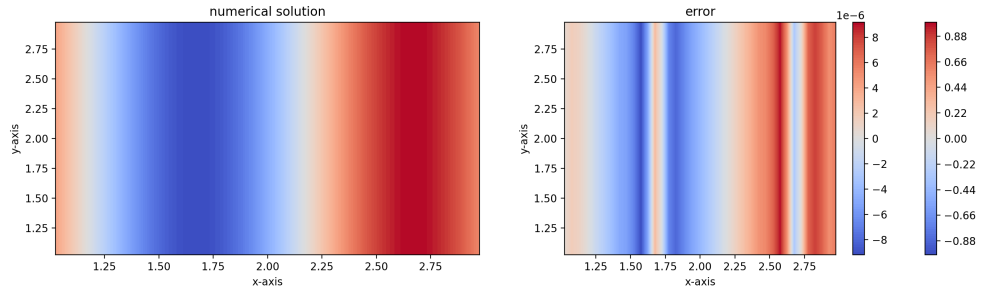
Example 3.8. We begin with a sine curve as in Problem 3.1. The curve should move forward with velocity $v_0 = 1$. The equation remains the same as in 3.1

$$\begin{aligned} u_t + \begin{pmatrix} v_0 \\ 0 \end{pmatrix} \circ u_{\vec{x}} = 0, \qquad \vec{x} = \begin{pmatrix} x \\ y \end{pmatrix} \in \Omega, t > 0 \\ u(\vec{x}, 0) = \phi(x, y), \qquad (x, y) \in \Omega. \end{aligned} \tag{3.17}$$

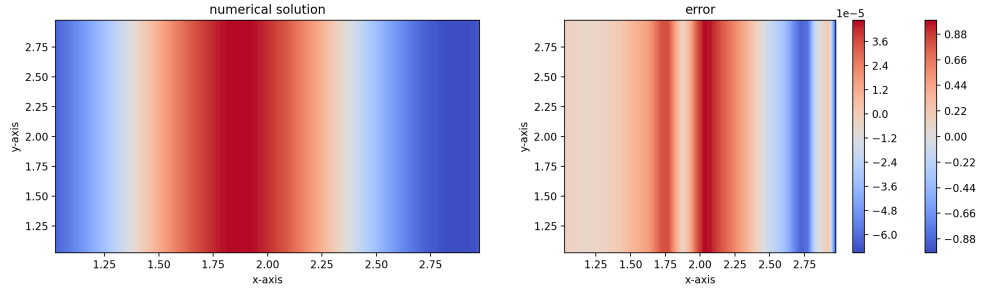
Here $\circ$ represents the Hadamard product also known as the element-wise product. The analytic solution is given by

$$u(x, y, t) = \phi(x - v_0 \cdot t, y). \tag{3.18}$$

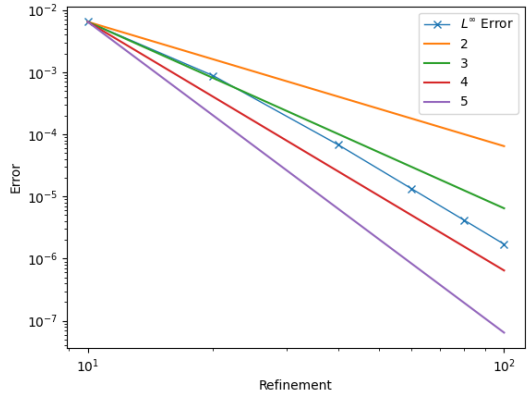
The solution, error and convergence rate can be seen in Figure 16 for the initial condition $\phi_1(x, y) = sin(\pi \cdot x) \quad \forall y$ and in Figure 17 for the initial condition $\phi_2(x, y) = e^{-((x-0.5)^2+(y-2)^2)}$. We can clearly see a fourth order convergence and small errors. The loss of one order in convergence from the one-dimensional to the two-dimensional case can be explained by the used flux splitting which does not maintain convergence orders.



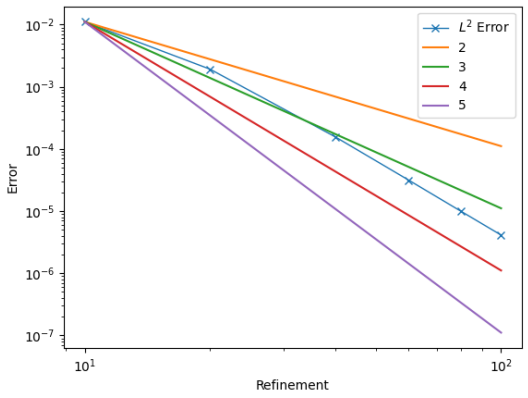
(a) Numeric solution and error at $t = 0.3$



(b) Numeric solution and error at $t = 1.3$



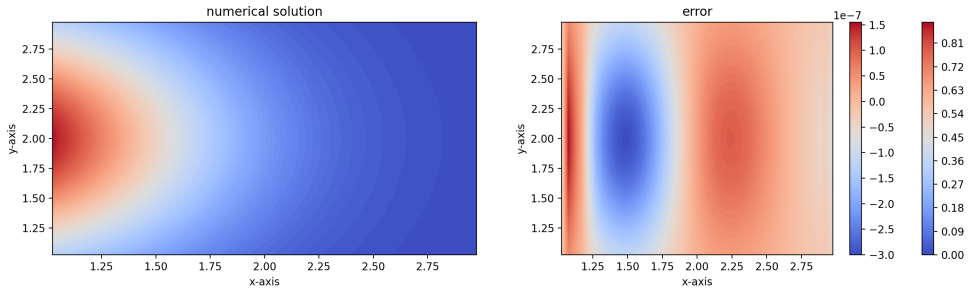
(c) Convergence plot using the ℓ_∞ norm



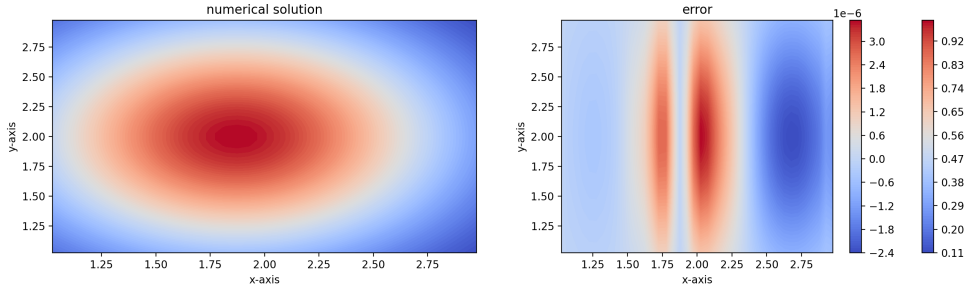
(d) Convergence plot using the ℓ_2 norm

Figure 16: Solution of the advection equation 3.17 with initial condition $\phi_1(x, y) = \sin(\pi \cdot x) \quad \forall y$ and the convergence plots for the numerical solver

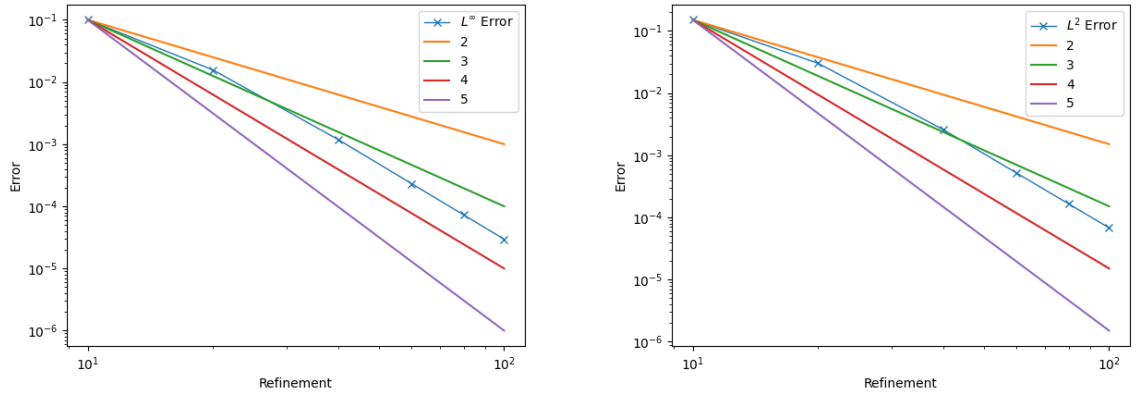
Example 3.9. We conclude the analysis of our solver by applying the solver to the two-dimensional advection equation with space-dependent flux and solution dependent source



(a) Numeric solution and error at $t = 0.3$



(b) Numeric solution and error at $t = 1.3$



(c) Convergence plot using the ℓ_∞ norm

(d) Convergence plot using the ℓ_2 norm

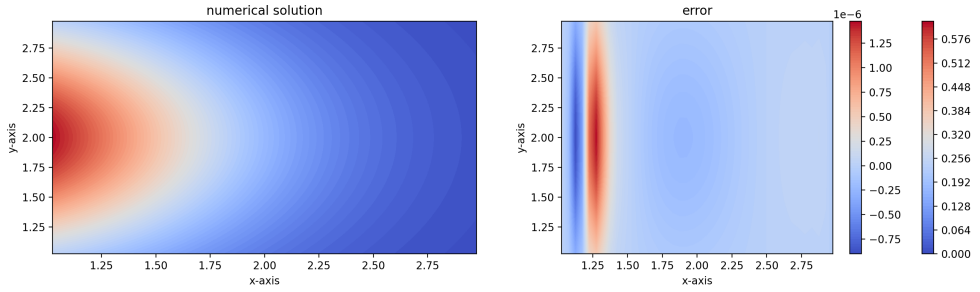
Figure 17: Solution of the advection equation 3.17 with initial condition $\phi_2(x, y) = e^{-((x-0.5)^2+(y-2)^2)}$ and the convergence plots for the numerical solver

term. This yields a similar problem as in Example 3.9

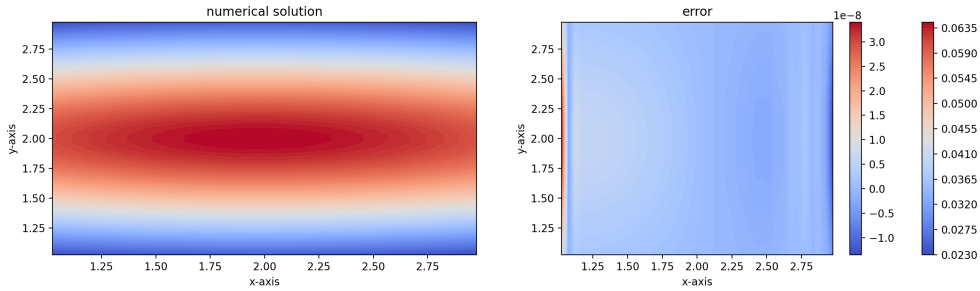
$$u_t + \begin{pmatrix} x \\ 0 \end{pmatrix} \circ u_{\vec{x}} = -(c+1) \cdot u, \qquad \vec{x} = \begin{pmatrix} x \\ y \end{pmatrix} \in \Omega, t > 0$$
$$u(\vec{x}, 0) = \phi(x, y), \qquad (x, y) \in \Omega. \tag{3.19}$$

The analytic solution is given by

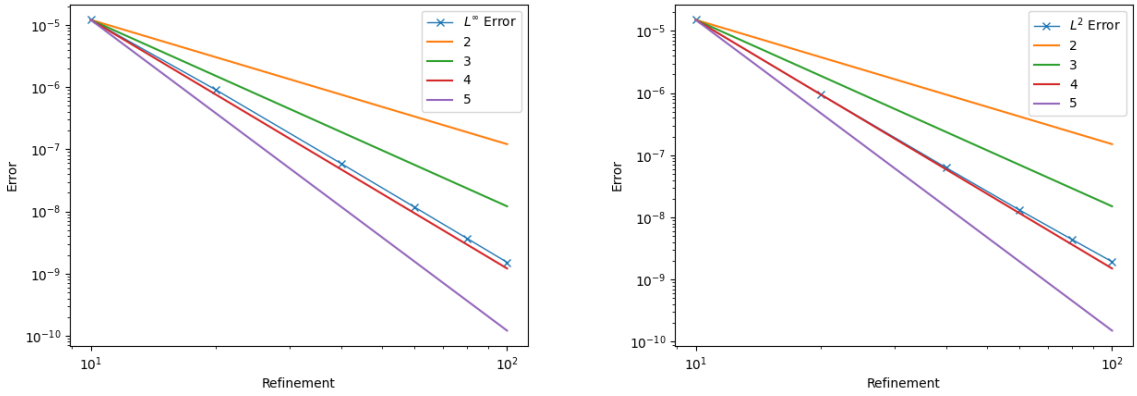
$$u(x, y, t) = \phi(x \cdot e^{-t}, y) \cdot e^{-(c+1)\cdot t}. \tag{3.20}$$



(a) Numeric solution and error at $t = 0.3$



(b) Numeric solution and error at $t = 1.3$



(c) Convergence plot using the ℓ_∞ norm

(d) Convergence plot using the ℓ_2 norm

Figure 18: Solution of the advection equation 3.19 with initial condition $\phi(x, y) = e^{-((x-0.5)^2+(y-2)^2)}$ and the convergence plots for the numerical solver

The solution, error and convergence rate can be seen in Figure 17 for the initial

condition $\phi(x, y) = e^{-((x-0.5)^2+(y-2)^2)}$. As in the example before, we can see the fourth order convergence and even smaller errors.

This concludes the analytic analysis of our solver. We have tested several cases and always got the expected convergence rate and small errors. Furthermore, we also changed the given parameters c,v,x and t to ensure that the solver works in different spaces and for different values of c as well as for examplary negative velocities. We did not include these results in the thesis because they gave, as expected, the same convergence rates and small errors as the shown examples. After all these tests, we are confident that we can use the solver for the simulation of model 1.27.

3.3 Simulations of Bolus Dispersion

Now we are using our solver to simulate the model 1.27. In the model, we have a space-dependent velocity and a space-dependent source term.

Example 3.10. First of all, we have a look at the model with a space-dependent velocity and a constant source term $c = 2$. We are going to split the area $\Omega = [1,3] \times [1,3]$ into

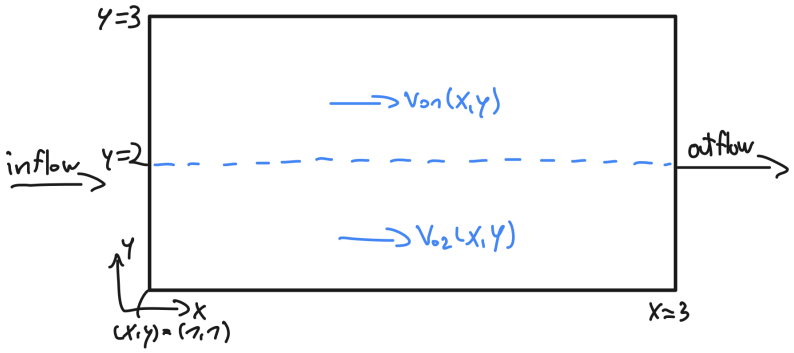


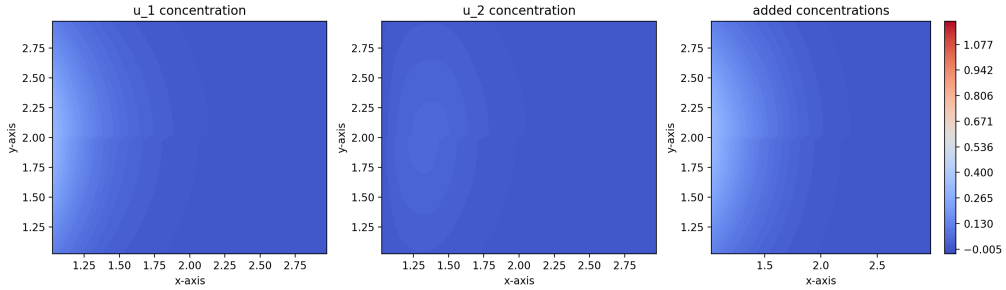
Figure 19: Experimental setting for the first two simulations

two parts, an upper and a lower one. The upper one is in the area from $[1,2] \times [3,3]$ with a velocity in x-direction given by $v_{01} = 2$ and the lower one is in the area $[1,1] \times [3,2]$ with a velocity in x-direction given by $v_{02} = 1.7$, see Figure 19. This gives us the following modeling equation

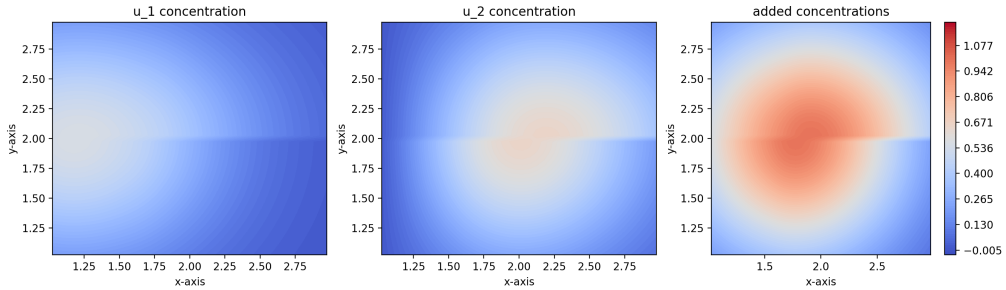
$$\begin{aligned} u_t^a(\vec{x},t) + \begin{pmatrix} 2 \\ 0 \end{pmatrix} \circ u_{\vec{x}}^a(\vec{x},t) &= -2 \cdot u^a(\vec{x},t) \\ u_t^v(\vec{x},t) + \begin{pmatrix} 1.7 \\ 0 \end{pmatrix} \circ u_{\vec{x}}^v(\vec{x},t) &= 2 \cdot u^a(\vec{x},t), \quad \vec{x} = \begin{pmatrix} x \\ y \end{pmatrix} \in \Omega, t > 0 \\ u^a(\vec{x},0) &= e^{-((x+0.5)^2+(y-2)^2)} \\ u^v(\vec{x},0) &= 0. \end{aligned} \tag{3.21}$$

The velocity in y-direction is set to zero for both concentrations u^a and u^v. Here, u^a denotes the concentration of the tracer in the arterial blood and u^v denotes the concentration of the tracer in the venous blood. We start the simulation with both concentrations

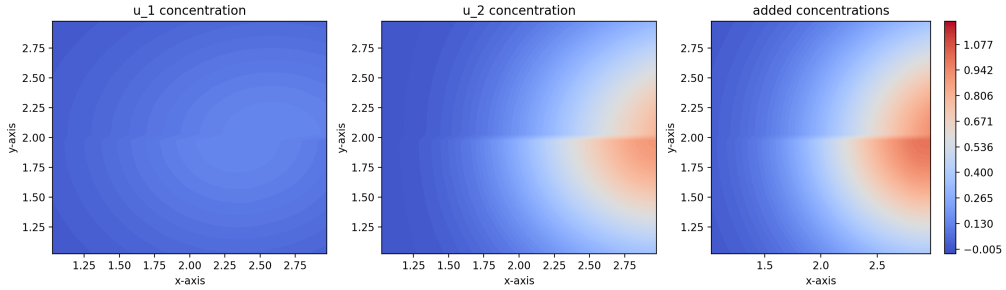
set to zero. Then, there is an inflow of u^a at the left boundary. Next, the substance gets transported with the described velocity field. We expect the substance to move faster in the upper half than in the lower. Furthermore, the concentration of u^a should decrease while the concentration of u^v increases.



(a) Simulation at time $t = 0.25$



(b) Simulation at time $t = 1.3$



(c) Simulation at time $t = 2.0$

Figure 20: Simulation of the advection equation 3.21

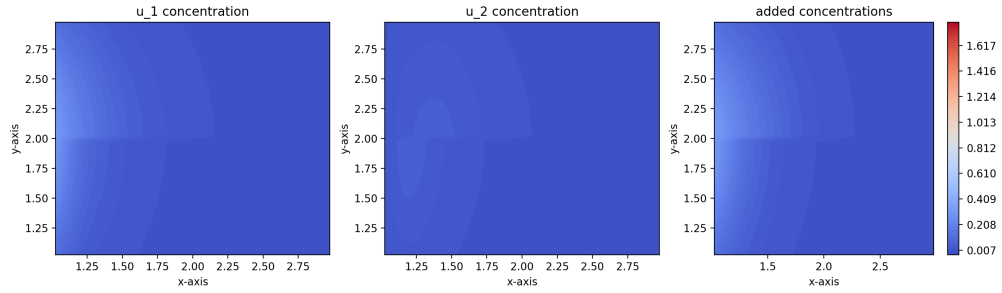
The plots in Figure 20 show the results of the simulation. We can clearly see the expected behaviour.

Example 3.11. For the next simulation, we keep the setting of Example 3.21. The only change is that the velocity in x-direction is set to v_{02}. Instead of having a constant term, we add a sine term and obtain the velocity $v_{02} = \begin{pmatrix} 1 + 0.25 \cdot \sin(\pi \cdot x) \\ 0 \end{pmatrix}$. This yields the following problem

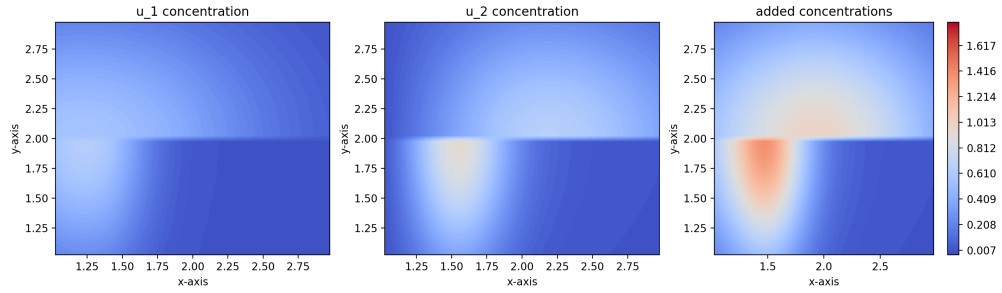
$$
\begin{aligned}
u_t^a(\vec{x},t) + \begin{pmatrix} 2 \\ 0 \end{pmatrix} \circ u_{\vec{x}}^a(\vec{x},t) &= -2 \cdot u^a(\vec{x},t) \\
u_t^v(\vec{x},t) + \begin{pmatrix} 1 + 0.25 \cdot \sin(\pi \cdot x) \\ 0 \end{pmatrix} \circ u_{\vec{x}}^v(\vec{x},t) &= 2 \cdot u^a(\vec{x},t), \quad \vec{x} = \begin{pmatrix} x \\ y \end{pmatrix} \in \Omega, t > 0 \\
u^a(\vec{x},0) &= e^{-((x+0.5)^2+(y-2)^2)} \\
u^v(\vec{x},0) &= 0.
\end{aligned} \tag{3.22}
$$

Again, we start the simulation with both concentrations set to zero. Then, there is an inflow of u^a at the left boundary. Next, the substance is transported with the described velocity field. We expect the substances to move faster in the upper half than in the lower. Furthermore, in the lower area the concentration should get compressed and stretched because of the sine part in the velocity. As before, the concentration of u^a should decrease while the concentration of u^v increases.

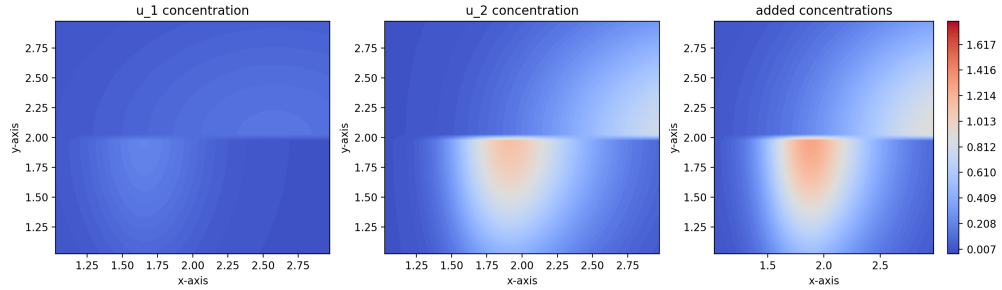
The plots in Figure 21 show the results of the simulation. Here, we can also see the expected behaviour: the sinus part leads to a compression of the concentration in the lower part of the field.



(a) Simulation at time $t = 0.25$



(b) Simulation at time $t = 1.3$



(c) Simulation at time $t = 2.0$

Figure 21: Simulation of the advection equation 3.22

Example 3.12. We try to derive a realistic model for the simulation of blood flow through the liver now. This means that we need to use a space-dependent source term as well as space-dependent velocities. The parameter c which determines how fast u^a gets transformed into u^v is zero at the outer parts of the area. This simulates the change from arterial to venous blood which only takes place in the organ itself. Furthermore, there should only be arterial blood at the inflow boundary and exclusively venous blood at the outflow boundary. Moreover, the velocity of the arterial blood is high at the inflow

boundary and decreases linearly to zero around the middle of the organ. The velocity of the venous blood, on the other side, is zero at the inflow boundary and increases linearly until the outflow boundary. In addition, the maximal velocity of arterial blood is higher than the maximal velocity of venous blood. For a better understanding, a graphic

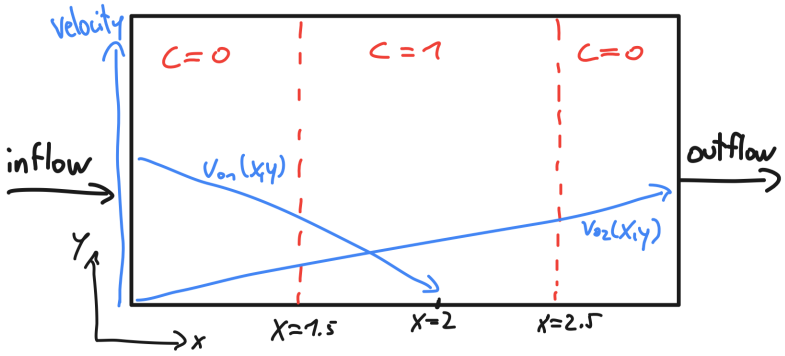


Figure 22: Experimental setting for the last two simulations

description of the setting can be found in Figure 22. This behaviour can be described by the following equations

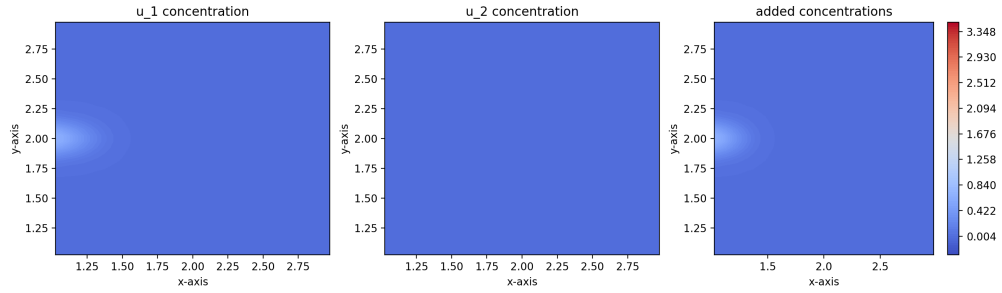
$$
\begin{aligned}
c(x,y) &= \begin{cases} 3, & 1.5 \leq x \leq 2.5 \\ 0, & \text{else} \end{cases}, \\
v^a(x,y) &= \begin{cases} 6-3x, & 1 \leq x \leq 2 \\ 0, & \text{else} \end{cases}, \\
v^v(x,y) &= \frac{5}{4} \cdot x - \frac{5}{4}, \\
v^a_{max} &> v^v_{max}.
\end{aligned}
\tag{3.23}
$$

Now we can include this information into our known advection equation and derive the following system

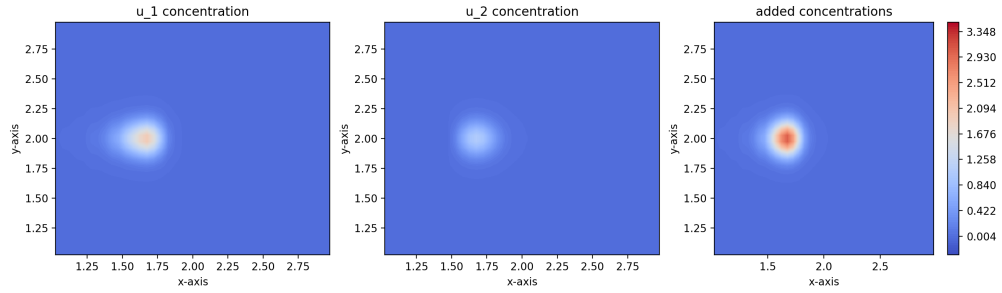
$$
\begin{aligned}
u_t^a(\vec{x},t) + v^a(\vec{x}) \circ u_{\vec{x}}^a(\vec{x},t) &= -c(\vec{x}) \cdot u^a(\vec{x},t) \\
u_t^v(\vec{x},t) + v^v(\vec{x}) \circ u_{\vec{x}}^v(\vec{x},t) &= c(\vec{x}) \cdot u^a(\vec{x},t), \quad \vec{x} = \begin{pmatrix} x \\ y \end{pmatrix} \in \Omega, t > 0 \\
u^a(\vec{x},0) &= e^{-\frac{(x+0.5)^2+(y-2)^2}{2\cdot\sigma^2}}, \quad \sigma = 0.2 \\
u^v(\vec{x},0) &= 0.
\end{aligned}
\tag{3.24}
$$

We start the simulation with both concentrations set to zero. Then, we have an inflow in the form of a Gauss peak of of u^a at the left boundary. This leads to the transportation of u^a with a decreasing velocity. As soon as we pass $x = 1.5$, the change from u^a into u^v starts and u^v is transported to the outflow boundary with an increasing velocity.

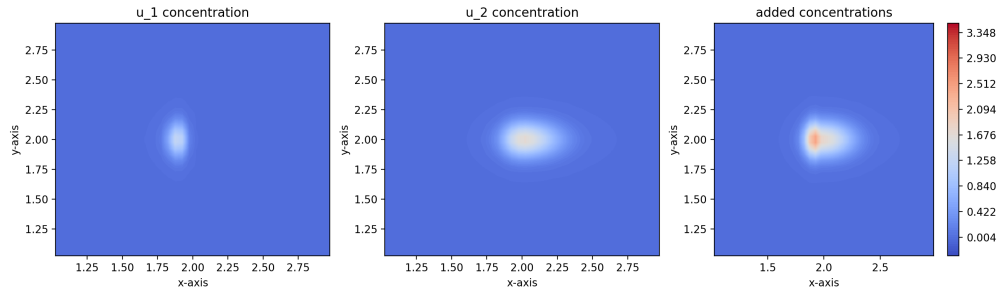
The plots of the simulation are shown in Figure 23; the expected behaviour can be observed.



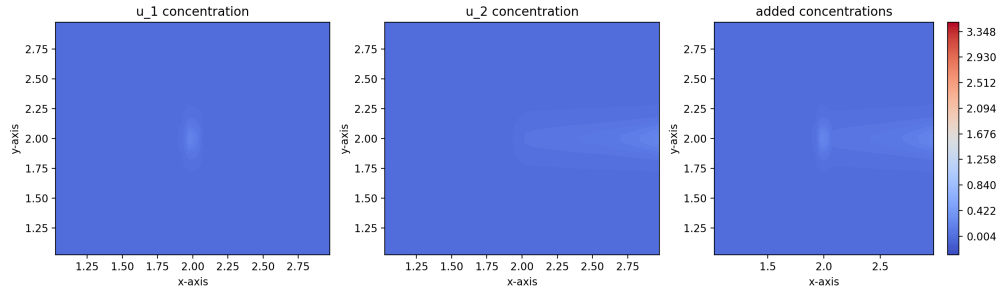
(a) Simulation at time $t = 0.25$



(b) Simulation at time $t = 1.0$



(c) Simulation at time $t = 1.5$



(d) Simulation at time $t = 2.0$

Figure 23: Simulation of the advection equation 3.26

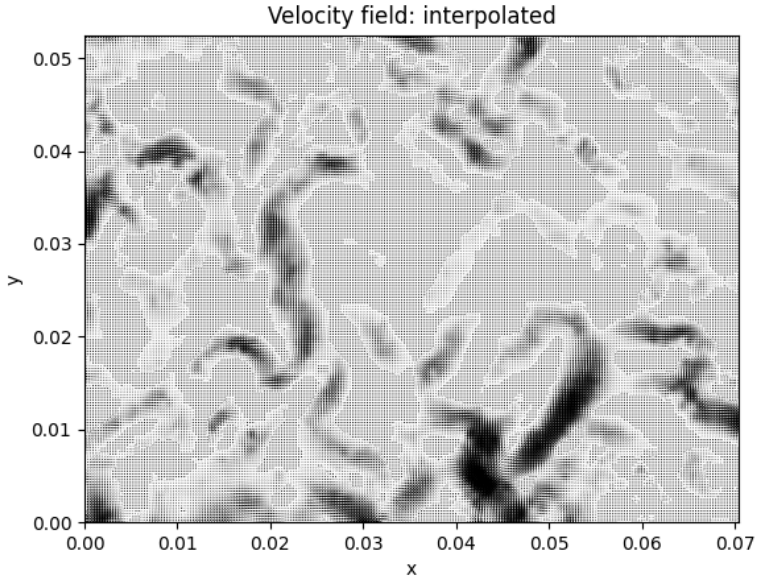


Figure 24: Interpolation of the measured velocity field [1][2]

Example 3.13. For the last experiment, we use the same setting as in the example before, but we add a disturbance to the velocity now. The disturbance is given by an experimental velocity field from [1] and [2] (see Figure 24). The experiment measured Faraday waves. These are waves occurring in a fluid that is subject to vertical shaking. They obtained those temporally and spatially well-resolved velocity fields by particle image velocimetry measurements at planes that where parallel and perpendicular to the water surface. In this manner, they gained a space- and time-dependent velocity field. For our purposes, we only extract the space-dependent velocity field at one moment of time. Because the space in the experiment is different from the area we used so far, we have to adapt the settings in order to be able to add the disturbance to our known model from Example 3.12. First of all, we set $\Omega = [0, 0.07] \times [0, 0.05]$. Secondly, we adapt the space-dependent source term and velocities

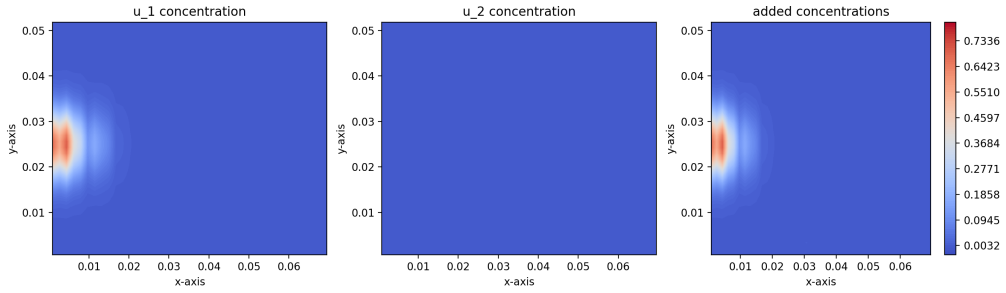
$$
\begin{aligned}
c(x,y) &= \begin{cases} 3, & 0.02 \le x \le 0.05 \\ 0, & \text{else} \end{cases}, \\
v^a(x,y) &= \begin{cases} 0.12 - 3.43x + \text{dist}(x,y), & 0 \le x \le 0.035 \\ 0, & \text{else} \end{cases}, \\
v^v(x,y) &= 1.43 \cdot x + \text{dist}(x,y).
\end{aligned} \tag{3.25}
$$

This gives us the maximal velocitys $v^a_{max} = 0.12$ and $v^v_{max} = 0.1001$ in Ω without disturbance. This still fulfills the condition $v^a_{max} > v^v_{max}$. Now we can include this information into our known advection equation and derive the following system

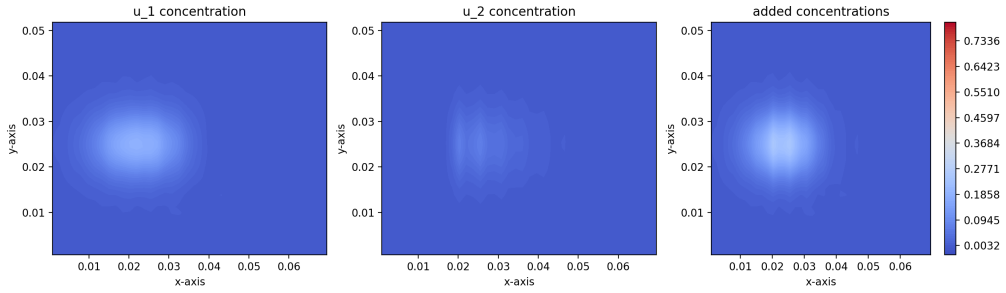
$$
\begin{aligned}
u^a_t(\vec{x},t) + v^a(\vec{x}) \circ u^a_{\vec{x}}(\vec{x},t) &= -c(\vec{x}) \cdot u^a(\vec{x},t) \\
u^v_t(\vec{x},t) + v^v(\vec{x}) \circ u^v_{\vec{x}}(\vec{x},t) &= c(\vec{x}) \cdot u^a(\vec{x},t), \quad \vec{x} = \begin{pmatrix} x \\ y \end{pmatrix} \in \Omega, t > 0 \\
u^a(\vec{x},0) &= 3e^{-\frac{(x+0.02)^2+(y-0.025)^2}{2\cdot\sigma^2}}, \quad \sigma = 0.005 \\
u^v(\vec{x},0) &= 0.
\end{aligned} \tag{3.26}
$$

We start the simulation with both concentrations set to zero. Then, we have an inflow in the form of a Gauss peak of u^a at the left boundary. This leads to the transport of u^a with a decreasing velocity and a disturbance from the velocity field. As soon as we pass $x = 0.02$, the change from u^a into u^v starts and u^v is transported to the outflow boundary with an increasing velocity. We expect to see a similar behaviour as in Example 3.12 regarding the transportation of the tracer but the disturbance of the velocity field should lead to frayed boundaries of the tracer concentration.

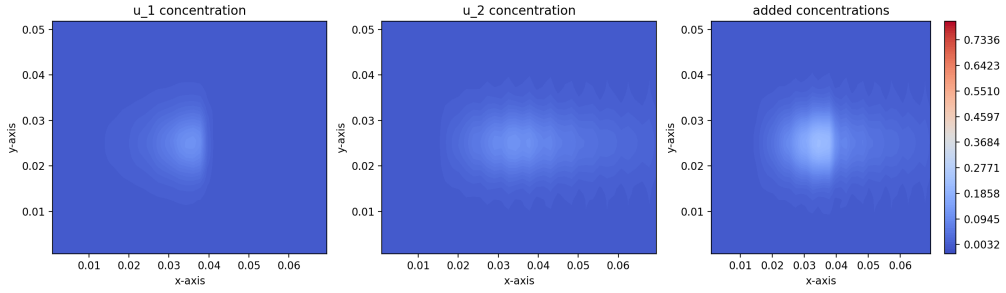
The plots of the simulation are shown in Figure 25; the expected behaviour can be observed.



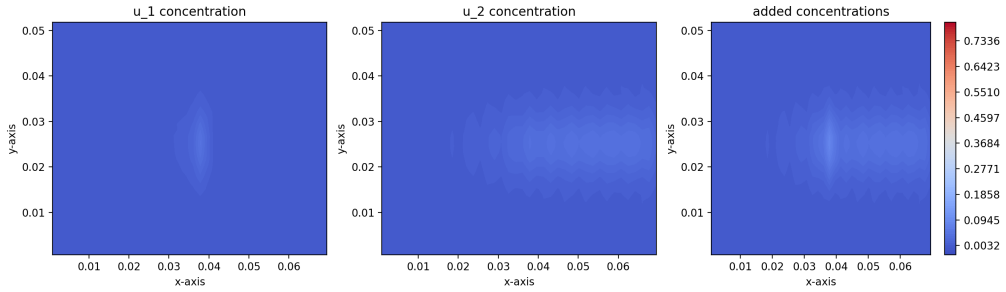
(a) Simulation at time $t = 0.25$



(b) Simulation at time $t = 1.0$



(c) Simulation at time $t = 1.5$



(d) Simulation at time $t = 2.0$

Figure 25: Simulation of the advection equation 3.26

4 Conclusion and Open Questions

This thesis discusses the numerical solution of a promising model for tracer transportation in blood flow. A finite difference WENO scheme was succesfully implemented and carefully verified. Furthermore, the solver was used on a two-component model for perfusion and the created simulations accurately represent the expected behaviour.

Open questions. In order to further improve the models for tumor perfusion and two-component models for intravascular indicators in general, a model that represents the relation between tissue parameters and spatial concentration gradients more explicitly is needed. This thesis gives a promising basis for further developments in this area. Furthermore, it could be interesting to see if the developed WENO scheme can be extended to solve other hyperbolic conservation laws with spatially varying fluxes and to extend the solver for not just spatially but also time-varying fluxes.

References

[1] Raffaele Colombi, Niclas Rohde, Michael Schlüter, and Alexandra von Kameke. "Coexistence of Inverse and Direct Energy Cascades in Faraday Waves". In: *Fluids* 7 (2022), p. 148.

[2] Raffaele Colombi, Michael Schlüter, and Alexandra von Kameke. "Three dimensional flows beneath a thin layer of 2D turbulence induced by Faraday waves". In: *Experiments in Fluids* 8 (2021), p. 62.

[3] A. Dolezal and S. Wong. "Relativistic hydrodynamics and essentially non-oscillatory shock capturing schemes". In: *Journal of Computational Physics* v120 (1995), pp. 266–277.

[4] E. Fatemi, J. Jerome, and S. Osher. "Solution of the hydrodynamic device model using high order non-oscillatory shock capturing algorithms". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* v10 (1991), pp. 232–244.

[5] Moshe Goldberg and Eitan Tadmore. "Scheme-independent stability criteria for dif- ference approximations of hyperbolic initial-boundary value problems, I." In: *Mathematics of Computation* 32 (1978), pp. 1097–1107.

[6] Dimitre Hristov, Lauri Mustonen, Rie von Eyben, Sebastian Götschel, Michael Minion, and Ahmed El Kaffas. "Dynamic Contrast-Enhanced Ultrasound Modeling of an Analog to Pseudo-Diffusivity in Intravoxel Incoherent Motion Magnetic Resonance Imaging". In: *IEEE Transactions On Medical Imaging* 10 (2022).

[7] Ling Huang, Chi-Wang Shu, and Mengping Zhang. "Numerical Boundary Conditions for the Fast Sweeping High Order WENO Methods for Solving the Eikonal Equation". In: *Journal of Computational Mathematics* 26.3 (2008), pp. 336–346.

[8] Guang-Shan Jiang and Chi-Wang Shu. "Efficient Implementation of Weighted ENO Shemes". In: *Journal of Computational Physics* 126 (1996), pp. 202–228.

[9] Serge Lang. *Differential manifolds*. Springer-Verlag, 1972.

[10] Julie Levandosky. "Partial Differential Equations of Applied Mathematics, Math 220A". lecturenotes. Stanford University, 2002.

[11] Xu-Dong Lu, Stanley Osher, and Tony Chan. "Weighted Esentially Non-oscillatory Schemes". In: *Journal of Computational Physics* 115 (1994), pp. 200–212.

[12] C.W. Shu and W. Cai. "Uniform high-order spectral methods for one- and two-dimensional Euler equations". In: *Journal of Computational Physics* v104 (1993), pp. 427–443.

[13] C.W. Shu, T.A. Zang, G. Erlebacher, D. Whitaker, and S. Osher. "High order ENO schemes applied to two- and three- dimensional compressible flow". In: *Applied Numerical Mathematics* v9 (1992), pp. 45–71.

[14] Chi-Wang Shu. "essentially Non-oscillatory and Weighted Essentially Non-oscillatory Schemes for Hyperbolic Conservation Laws". In: *ICASE* 97–65 (1997).

[15] Chi-Wang Shu. "Essentially non-oscillatory and weighted essentially non-oscillatory schemes". In: *Cambridge University Press* (2020), pp. 1–63.

[16] Chi-Wang Shu and Stanley Osher. "Efficient Implementation of Essentially Non-oscillatory Shock-Capturing Schemes". In: *Journal of Computational Physics* 77 (1988), pp. 439–471.

[17] Chi-Wang Shu and Stanley Osher. "Efficient Implementation of Essentially Non-oscillatory Shock-Capturing Schemes, II". In: *Journal of Computational Physics* 83 (1989), pp. 32–78.

[18] Steven Sourbron. "A Tracer-Kinetic Field Theory for Medical Imaging". In: *IEEE Transactions On Medical Imaging* 33.4 (2014).

[19] Sirui Tan. "Boundary conditions and applications of WENO finite difference schemes for hyperbolic problems". PhD thesis. Applied Mathematics at Brown University, 2012.

[20] Peng Zhang, S.C. Wong, and Chi-Wang Shu. “A weighted essentially non-oscillatory numerical scheme for a multi-class traffic flow model on an inhomogeneous highway”. In: *Journal of Computational Physics* 2.212 (2006), pp. 739–756.