Testing Physics-Informed Neural Networks for the Solution of Hyperbolic Conservation Laws

Leon Jakobi* Simon Krotsch†

July 15, 2024

Abstract. This report investigates the applicability of physics-informed neural networks (PINNs) for the solution of one-dimensional systems of hyperbolic conservation laws. Unlike traditional numerical methods that discretize the governing equations, PINNs integrate the physical laws directly into the neural network training process, utilizing its ability to learn complex patterns from data while respecting the underlying physical model. Using three example problems, we demonstrate the accuracy of PINNs in capturing the motion of different types of waves (both smooth and discontinuous) which are characteristic for hyperbolic systems. Time and accuracy comparisons are made with more traditional finite-difference and finite-volume approaches. Our results give insight into the potential of PINNs as a tool for solving hyperbolic conservation laws, highlighting both their strengths and their limitations.

1 Introduction

The goal of this text is to investigate the applicability of physics-informed neural networks for the solution of a one-dimensional systems of hyperbolic conservation laws

$$q_t + f(q)_x = 0$$

without source term. Here $(x,t) \mapsto q(x,t)$ is the quantity and $q \mapsto f(q)$ the flux function. The indices denote differentiation. In section 2, we give a quick introduction to conservation laws, focusing on their derivation, some theoretical results and three concrete examples of conservation laws. These will be used in section 3 for numerical tests. Whenever exact solutions are not available, we will rely on finite-difference and finite-volume methods to provide sufficiently good approximations to the true solution. This will also allow us to compare the speed and accuracy of PINNs with these more established schemes.

2 Conservation Laws

In this section our goal is to give a quick introduction to an important class of differential equations: conservation laws. Such models arise in many different practical applications, most famously in fluid mechanics, making their solution all the more important. In subsection 2.1 we briefly discuss their origin and define what it means for such an equation to be hyperbolic. Subsection 2.2 introduces the theoretical treatment of conservation laws and gives some insight into their existence theory. For the numerical simulations of section 3 we will study three concrete examples: the simplest

*leon.jakobi@stud-mail.uni-wuerzburg.de
†simon.krotsch@stud-mail.uni-wuerzburg.de

conservation law (linear advection), the simplest nonlinear conservation law (Burgers' equation) and one of the simpler practically-relevant models (the shallow-water equations) which is a nonlinear system. These are described in subsection 2.3.

2.1 Derivation of Conservation Laws

On a surface level subjects like physics, chemistry and biology appear mostly unrelated. On a mathematical level, however, their models often show similarities. One very common class of differential equations that appear frequently are conservation laws. This subsection explains their origin and is based on [13].

When conservation laws appear in practice some sort of balance law is always involved. To this end, consider some fixed region $\Omega \subseteq \mathbb{R}^3$ and in it some quantity with a density or concentration q which is itself a scalar-valued function of space $x = (x, y, z)^\mathrm{T} \in \mathbb{R}^3$ (this notation looks a bit irritating at first, but it is standard in fields like fluid dynamics) and time time $t > 0$. What happens to the quantity inside of Ω as time passes? Typically one has the following relation:

$$\begin{aligned} \text{time rate of change of quantity} = {} & \text{rate at which quantity flows into } \Omega \\ & - \text{rate at which quantity flows out of } \Omega \\ & + \text{rate at which quantity is produced in } \Omega \\ & - \text{rate at which quantity is destroyed in } \Omega. \end{aligned} \tag{2.1}$$

To illustrate this, consider an example from population biology. If the quantity corresponds to the population size of some species (e. g. foxes) in a fixed area Ω (e. g. Germany), then (2.1) can be formulated as

$$\begin{aligned} \text{rate of population change} = {} & \text{immigration rate} - \text{emigration rate} \\ & + \text{birth rate} - \text{death rate}. \end{aligned}$$

The natural mathematical way to phrase (2.1) is via integrals over the region Ω and its boundary $\partial\Omega$. The total quantity inside of Ω is given via the volume integral

$$\int_\Omega q(x,t)\,\mathrm{d}x.$$

Notice that this is a function that varies only with time. Its derivative corresponds to the left side of (2.1).

The inflow and outflow can be combined into one net term, if we introduce a *flux function* $(x,t) \mapsto f(x,t)$. Its components correspond to the amount of the quantity q flowing through the surface $\partial\Omega$ at the point x on the surface and time t per unit area and per unit time in the component's direction. By convention, we will say that a component of the flux function is positive, if the flow is out of the surface, and negative, if it is into the surface. This way the integral

$$-\int_{\partial\Omega} f(x,t)\cdot n(x)\,\mathrm{d}s \tag{2.2}$$

subsumes the first two terms on the right side of (2.1). Here $n(x)$ is the unit normal vector on $x \in \partial\Omega$ pointing out of Ω and the dot $\cdot$ represents the Euclidean inner product in $\mathbb{R}^3$. The minus sign is due to the outward-facing direction of n.

The production and destruction can be handled in somewhat the same way with a scalar-valued *source function* $(q, x, t) \mapsto \psi(q, x, t)$. Notice that ψ can generally vary with the quantity q. If the source function is positive, we speak of a *source*, and if it is negative, we speak of a *sink*. The last two terms on the right side of (2.1) can then be represented as

$$\int_\Omega \psi(q(x,t), x, t)\,\mathrm{d}x.$$

If we use the divergence theorem to transform the surface integral (2.2) into a volume integral, we can state (2.1) as

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\int_\Omega q(x,t)\,\mathrm{d}x\right) = -\int_\Omega \nabla\cdot f(x,t)\,\mathrm{d}x + \int_\Omega \psi(q(x,t), x, t)\,\mathrm{d}x. \tag{2.3}$$

In the case where q is a sufficiently smooth function, the derivative on the left may be pulled into the integral. And since Ω is an arbitrary region, the smoothness of the involved functions would imply the differential equation

$$q_t + \nabla\cdot f(x,t) = \psi(q, x, t).$$

Here and in the following, indices are supposed to signal differentiation, e. g. $()_t := \partial/\partial t$. The unknowns in the equation above are q and f while ψ is given. However, in many applications it is possible to write f as a function of q directly. Notice that this is a special case of the form above since q itself depends on (x, t). In said case the differential equation becomes

$$q_t + \nabla\cdot f(q) = \psi(q, x, t) \tag{2.4}$$

or

$$q_t + f(q)_x + g(q)_y + h(q)_z = \psi(q, x, t) \tag{2.5}$$

by writing out the divergence operator $\nabla\cdot = (\partial/\partial x, \partial/\partial y, \partial/\partial z)^{\mathrm{T}}$ and using the notation $f = (f, g, h)^{\mathrm{T}}$ for the components. The corresponding formulation of (2.3) is

$$\int_\Omega q_t(x,t) + \nabla\cdot f\big(q(x,t)\big)\,\mathrm{d}x = \int_\Omega \psi(q(x,t), x, t)\,\mathrm{d}x. \tag{2.6}$$

Equations (2.6) and (2.4) are the *integral* and *differential* form of a *conservation law*. The unknown function here is q while f and ψ are known. We note that if one is given several conservation laws at the same time, then they can be combined into a vectorized version of (2.5) via

$$q_t + f(q)_x + g(q)_y + h(q)_z = \psi(q, x, t). \tag{2.7}$$

Here q is a vector of conserved quantities; f, g and ψ are defined accordingly, so that each component of (2.7) corresponds to one of the given conservation laws.

Notice that only the case where the source term vanishes is a proper conservation in the colloquial sense. Consequently, equations with a non-trivial ψ are sometimes referred to as *balance laws* instead. Throughout this text, we will only consider models without a source term.

Next up, we want to introduce what it means for the one-dimensional conservation law

$$q_t(x,t) + f\big(q(x,t)\big)_x = \psi(q(x,t),x) \quad \text{for all } (x,t) \in \mathbb{R} \times]0,\infty[\,, \tag{2.8}$$

to be hyperbolic. We will only study the one-dimensional case in this text. However, a suitable generalization for several dimensions exists. We begin with a *linear system* of conservation laws

$$q_t + Aq_x = \psi(q,x) \tag{2.9}$$

for some function $q : \mathbb{R} \times]0,\infty[\to \mathbb{R}^m$ and a constant matrix $A \in \mathbb{R}^{m\times m}$. In this case, we say that (2.9) is hyperbolic, if A is diagonalizable with real eigenvalues. If we were instead given a *quasilinear system* of conservation laws

$$q_t + A(q,x,t)q_x = \psi(q,x), \tag{2.10}$$

then we say that (2.10) is hyperbolic, if the variable matrix $A(q,x,t) \in \mathbb{R}^{m\times m}$ is diagonalizable with real eigenvalues at every point (q,x,t). To define hyperbolicity for (2.8), we carry out the derivative in terms of x with the chain rule to find

$$q_t + f'(q)q_x = \psi(q,x), \tag{2.11}$$

which is valid, if q is smooth. Here

$$f' = \begin{pmatrix} \frac{\partial f_1}{\partial q_1} & \frac{\partial f_1}{\partial q_2} & \cdots & \frac{\partial f_1}{\partial q_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial q_1} & \frac{\partial f_m}{\partial q_2} & \cdots & \frac{\partial f_m}{\partial q_m} \end{pmatrix} \in \mathbb{R}^{m\times m}$$

is the Jacobian matrix of $f = (f_1, f_2, \ldots, f_m)^{\mathrm{T}}$ for $q = (q_1, q_2, \ldots, q_m)^{\mathrm{T}}$. Hence (2.11) is a quasilinear system. We will say that (2.8) is hyperbolic, if (2.11) is hyperbolic for each (physically-relevant) value of q.

2.2 Existence Theory for One-Dimensional Conservation Laws

In this subsection we want to give some insight into the existence theory of one-dimensional systems of conservation laws without a source term. We start by introducing a method to construct solutions for the initial value problem for scalar one-dimensional conservation laws in 2.2.1. After that 2.2.2 introduces an appropriate notion of weak solution. Finally 2.2.3 presents some existence results.

Let $\Omega \subseteq \mathbb{R}^m$ be open ($m \in \mathbb{N}$). Here we assume that the flux function $f : \Omega \to \mathbb{R}^m$ is sufficiently smooth. Then the the one-dimensional system of conservation laws without a source term is

$$q_t + f(q)_x = 0, \tag{2.12}$$

where $x \in \mathbb{R}$, $t > 0$ and $q : \mathbb{R} \times \mathbb{R}_+ \to \Omega$ is a vector-valued function. Here and in the following we set $\mathbb{R}_+ := \{x \in \mathbb{R} \mid x \geqslant 0\}$. Recall that (2.12) is called hyperbolic, if its Jacobian matrix f' has m real eigenvalues for all $q \in \Omega$. If the eigenvalues are also distinct, we call (2.12) *strictly hyperbolic*. In this text we want to study the *initial*

value problem (IVP) for (2.12). It is defined as follows: We want to find a function $q : \mathbb{R} \times \mathbb{R}_+ \to \Omega$ that satisfies (2.12) and the so-called *initial condition*

$$q(x, 0) = q_0(x)$$

for all $x \in \mathbb{R}$, where $q_0 : \mathbb{R} \to \mathbb{R}$ is some given function. These definitions and the next part of this subsection are based on [10].

2.2.1 Method of Characteristics

We start our investigation of the existence of solutions with the initial value problem for the scalar one-dimensional conservation law. In that case Ω is an open subset of $\mathbb{R}$ and the flux function f a smooth function from Ω into $\mathbb{R}$. The initial value problem for the scalar case is to find a function $q : \mathbb{R} \times \mathbb{R}_+ \to \Omega$ that satisfies

$$q_t + f(q)_x = 0 \tag{2.13}$$

for all $x \in \mathbb{R}$, $t > 0$ and

$$q(x, 0) = q_0(x) \tag{2.14}$$

for all $x \in \mathbb{R}$. Once again $q_0 : \mathbb{R} \to \mathbb{R}$ is some given function. We can transform (2.13) to

$$q_t + f'(q)q_x = 0$$

by using the chain rule.

Definition 2.1 *The curves define by $(x(t), t)$, where $x(t)$ is a solution of the ordinary differential equation*

$$\frac{\mathrm{d}}{\mathrm{d}t}x(t) = f'\big(q(x(t), t)\big),$$

are called characteristic curves.

Suppose the solution q of the initial value problem of (2.13) is continuously differentiable. Let $x_0 \in \mathbb{R}$, then the characteristic curve through the point $(x_0, 0)$ is defined by $(x(t), t)$, where $x(t)$ is a solution of the initial value problem

$$\frac{\mathrm{d}}{\mathrm{d}t}x(t) = f'\big(q(x(t), t)\big), \quad x(0) = x_0.$$

From the theory of ordinary differential equations we know the solution of that problem exists at least in some small neighborhood of $t = 0$. By the chain rule we obtain

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}q(x(t), t) &= \frac{\partial}{\partial t}q(x(t), t) + \frac{\partial}{\partial x}q(x(t), t)\frac{\mathrm{d}}{\mathrm{d}t}x(t) \\ &= \frac{\partial}{\partial t}q(x(t), t) + f'(u)\frac{\partial}{\partial x}q(x(t), t) \\ &= 0,\end{aligned}$$

since q satisfies (2.13). Hence q is constant along characteristic curves. Furthermore $x(t)$ is defined by

$$\frac{\mathrm{d}}{\mathrm{d}t}x(t) = f'\big(q(x_0, 0)\big) = f'\big(q_0(x_0)\big),$$

so $x(t)$ is a linear function and the characteristic curve is a straight line in the x-t-plane. With the condition $x(0) = x_0$ we get that the characteristic curve through the point $(x_0, 0)$ is defined by the curve $(x(t), t)$ that obeys

$$x(t) = x_0 + tf'\big(q_0(x_0)\big).$$

We have hereby proven the following result.

Lemma 2.2 *Suppose the solution of the initial value problem for (2.13) is continuously differentiable, then the characteristic curves are straight lines along which the solution is constant. The slope of the characteristic curves depends only on the initial data q_0.*

We can use our findings to construct continuously differentiable solutions of the initial value problem for (2.13). It follows directly that the the value of a solution q at the point $(x, t) \in \mathbb{R} \times \mathbb{R}_+$ is defined by

$$q(x, t) = q_0(x^*),$$

where x^* is the solution of

$$x = x^* + tf'\big(q_0(x*)\big).$$

Since we assumed that q is continuously differentiable, the same thing must hold for the initial data q_0.

This method for constructing continuously differentiable solutions is called the *method of characteristics*. One might think at this point that it allows us to construct solutions without restriction. But that would require that solutions of the initial value problem for (2.13) are continuously differentiable for all points $(x, t) \in \mathbb{R} \times \mathbb{R}_+$. Unfortunately, this is not always the case. Suppose q is a continuously differentiable solution of the initial value problem for (2.13) defined for all points $(x, t) \in \mathbb{R} \times \mathbb{R}_+$. Also assume we have two points $x_1, x_2 \in \mathbb{R}$ with $x_1 < x_2$ and

$$f'\big(q_0(x_1)\big) > f'\big(q_0(x_2)\big). \tag{2.15}$$

Let $C_1 := \{(x^1(t), t) \mid t \geqslant 0\}$ be the characteristic curve through the point $(x_1, 0)$ and $C_2 := \{(x^2(t), t) \mid t \geqslant 0\}$ the characteristic curve trough $(x_2, 0)$, then $x^1(t)$ and $x^2(t)$ are defined by

$$\begin{aligned} x^1(t) &= x_1 + tf'\big(q_0(x_1)\big), \\ x^2(t) &= x_2 + tf'\big(q_0(x_2)\big). \end{aligned}$$

From (2.15) it follows that C_2 and C_1 have to intersect at some point after finite time. At the point where the two meet, the solution q should take the values $q_0(x_1)$ and $q_0(x_2)$. If $q_0(x_1) \neq q_0(x_2)$ this is not possible, if q is supposed to be a single-valued function. So the continuously differentiable solution q cannot be defined at this point. As we can see, the method of characteristics sometimes only allows us to construct continuously differentiable solution of the initial value problem for (2.13) up to some finite time and thereby we are only able show the existence of continuously differentiable solution up to this finite time. The next result shows how this time may be calculated.

Theorem 2.3 *Suppose q_0 and f' are sufficiently smooth and let*

$$t^* := \min_{x \in \mathbb{R}} \frac{\mathrm{d}}{\mathrm{d}x} f'\big(q_0(x)\big).$$

Then there are three cases:

(a) If $t^* \geqslant 0$, i. e., the function $f'\big(q_0(x)\big)$ is monotonically increasing, then a continuously differentiable solution of the initial value problem for (2.13) exists and can be constructed by the method of characteristic for all times $t > 0$.

(b) If $t^* \in \mathbb{R}$ and $t^* < 0$, then a continuously differentiable solution of the initial value problem for (2.13) exists and can be constructed by the method of characteristics up to the time

$$T^* := -\frac{1}{t^*}.$$

(c) If t^* does not exist, then there is no continuously differentiable solution of the initial value problem for (2.13).

If the solution exists, it is implicitly given by

$$q(x,t) = u_0(x^*), \qquad x^* = x - t f'\big(q_0(x^*)\big).$$

Proof: We have already shown that if a continuously differentiable solution of the initial value problem for (2.13) exists up to some time, it can be constructed by the method of characteristics, since the characteristics do not cross. So the solution is given by

$$q(x,t) = u_0(x^*), \qquad x^* = x - t f'\big(q_0(x^*)\big).$$

All that remains to be demonstrated is that in case (a) the characteristic curves do not cross for any time, in case (b) that they cross after some finite time and in case (c) that the characteristic curves cross arbitrarily close to the time $t = 0$.

View $(x_1, 0)$ and $(x_2, 0)$, where $x_1, x_2 \in \mathbb{R}$ are two arbitrary points. Without loss of generality we can assume $x_1 < x_2$. Let $C_1 := \{(x^1(t), t) \mid t \geqslant 0\}$ and $C_2 := \{(x^2(t), t) \mid t \geqslant 0\}$ be the characteristic curves through x_1 and x_2 respectively. The curves C_1 and C_2 cross, if

$$x_1 + t f'\big(q_0(x_1)\big) = x_2 + t f'\big(q_0(x_2)\big).$$

This is equivalent to

$$t\big[f'\big(q_0(x_1)\big) - f'\big(q_0(x_2)\big)\big] = x_2 - x_1. \tag{2.16}$$

In the case (a) where the function $f'\big(q_0(x)\big)$ is monotonically increasing we get

$$f'\big(q_0(x_1)\big) - f'\big(q_0(x_2)\big) \leqslant 0$$

for any $x_1, x_2 \in \mathbb{R}$ with $x_1 < x_2$ and

$$x_2 - x_1 > 0.$$

So there cannot be any $t \geqslant 0$ that solves the equation (2.16). So the characteristics do not cross at any time $t \geqslant 0$. This proves (a).

We can assume that $t > 0$. If $t = 0$, then (2.16) only has a solution if $x_2 = x_1$. So at $t = 0$ the curves C_1 and C_2 do not cross. With $t \neq 0$ equation (2.16) becomes

$$-\frac{1}{t} = \frac{f'\big(q_0(x_2)\big) - f'\big(q_0(x_1)\big)}{x_2 - x_1}. \tag{2.17}$$

Let $t^* \in \mathbb{R}$ and $t^* < 0$ and assume T^* is not the smallest time at which two characteristic curves cross. So there is a $T' < T^*$, which solves (2.16) and so (2.17). This leads to

$$-\frac{1}{T'} = \frac{f'\big(q_0(x_2)\big) - f'\big(q_0(x_1)\big)}{x_2 - x_1}.$$

By the mean value theorem, there is a $\xi \in \mathbb{R}$ such that

$$\frac{f'\big(q_0(x_2)\big) - f'\big(q_0(x_1)\big)}{x_2 - x_1} = \frac{\mathrm{d}}{\mathrm{d}x} f'\big(q_0(\xi)\big).$$

Since we have

$$-\frac{1}{T'} < -\frac{1}{T^*}$$

this implies

$$\frac{\mathrm{d}}{\mathrm{d}x} f'\big(q_0(\xi)\big) < t^* = \min_{x \in \mathbb{R}} \frac{\mathrm{d}}{\mathrm{d}x} f'\big(q_0(\xi)\big).$$

Which is a contradiction to our assumption. So T^* is the smallest time at which two characteristic curves intersect. This mean up to T^* we can construct a continuously differentiable solution of the initial value problem for (2.13) by the method of characteristics and thereby we have proven (b).

Now we suppose that t^* does not exist, then for all $T^* > 0$ there are always $x_1, x_2 \in \mathbb{R}$, such that the characteristics through $(x_1, 0)$ and $(x_2, 0)$ cross before T^*, since there is a sequence (x_n) such that

$$\frac{\mathrm{d}}{\mathrm{d}x} f'\big(q_0(x_n)\big) \to -\infty$$

as $n \to \infty$. This shows (c). ■

2.2.2 Weak Solutions

We have seen that in general a continuously differentiable solution of the initial value problem for (2.12) does not exists for all $t > 0$. Depending on the equation we can only get a continuously differentiable solution in a very small time interval. So if we want to define solutions in general for all $t > 0$ we need a new notion of solution, which allows for discontinuities. This part is based on [10] and [16].

Definition 2.4 *Let $C_0^1(\mathbb{R} \times \mathbb{R}_+; \mathbb{R}^m)$ be the space of all functions $g \in C^1(\mathbb{R}^2; \mathbb{R}^m)$ with compact support in $\mathbb{R} \times \mathbb{R}_+$.*

To clarify, a function $\phi \in C_0^1(\mathbb{R} \times \mathbb{R}_+; \mathbb{R}^m)$ is the restriction to $\mathbb{R} \times \mathbb{R}_+$ of a function in $C^1(\mathbb{R}^2; \mathbb{R}^m)$ with compact support in an open set containing $\mathbb{R} \times \mathbb{R}_+$. Let $q_0 \in L^\infty_{\text{loc}}(\mathbb{R}; \mathbb{R}^m)$, where $L^\infty_{\text{loc}}(\mathbb{R}; \mathbb{R}^m)$ is the space of locally bounded measurable functions. Now assume q is a continuously differentiable solution of the initial value problem for (2.12). Let $\phi \in C_0^1(\mathbb{R} \times \mathbb{R}_+; \mathbb{R}^m)$. We then multiply (2.12) with ϕ and integrate over $\mathbb{R} \times \mathbb{R}_+$. This gives

$$\int_{\mathbb{R}} \int_0^\infty (q_t + f(q)_x) \cdot \phi \, \mathrm{d}t\mathrm{d}x = 0.$$

The dot $\cdot$ denotes the Euclidean inner product on $\mathbb{R}^m$. We can use integration by parts to obtain

$$\begin{aligned} 0 &= \int_{\mathbb{R}} \int_0^\infty (q_t + f(q)_x) \cdot \phi \, \mathrm{d}t\mathrm{d}x \\ &= -\int_{\mathbb{R}} \int_0^\infty q \cdot \phi_t \, \mathrm{d}t\mathrm{d}x - \int_{\mathbb{R}} \int_0^\infty f(q) \cdot \phi_x \, \mathrm{d}t\mathrm{d}x - \int_{\mathbb{R}} q \cdot \phi \, \mathrm{d}x \Big|_{t=0}. \end{aligned}$$

With the initial condition $q(x, 0) = q_0(x)$ we get

$$\int_{\mathbb{R}} \int_0^\infty q \cdot \phi_t + f(q) \cdot \phi_x \, \mathrm{d}t\mathrm{d}x + \int_{\mathbb{R}} q_0 \cdot \phi(x, 0) \, \mathrm{d}x = 0. \tag{2.18}$$

This equation makes sense even if q is not continuously differentiable. We just need $q \in L^\infty_{\text{loc}}(\mathbb{R} \times \mathbb{R}_+; \mathbb{R}^m)$. This enables us to generalise solutions as follows.

Definition 2.5 *Let $q_0 \in L^\infty_{\text{loc}}(\mathbb{R}; \mathbb{R}^m)$. We call a function $q \in L^\infty_{\text{loc}}(\mathbb{R} \times \mathbb{R}_+; \mathbb{R}^m)$ a* weak solution *of the initial value problem for (2.12) if $q(x, t) \in \Omega$ almost everywhere and q satisfies (2.18) for all $\phi \in C_0^1(\mathbb{R} \times \mathbb{R}_+; \mathbb{R}^m)$.*

The next theorem gives some insight into the connection between weak solutions and classical solutions, i. e. continuously differentiable solutions of the initial value problem for (2.12).

Theorem 2.6 *A continuously differentiable function q is a classical solution of the initial value problem for (2.12) if and only if q is a weak solution.*

Proof: A classical solution of the initial value problem for (2.12) is a weak solution. That follows directly from the construction of (2.18).

Now let $q : \mathbb{R} \times \mathbb{R}_+ \to \Omega$ be a continuously differentiable weak solution of the initial value problem for (2.12). Let $\phi \in C_0^1(\mathbb{R} \times]0, \infty[\,; \mathbb{R}^m)$, then if we do the construction of (2.18) backwards we get

$$\int_{\mathbb{R}} \int_0^\infty \big(q_t + f(q)_x\big) \cdot \phi \, \mathrm{d}t\mathrm{d}x = 0$$

because $\phi(x, 0) = 0$. Since this holds for all $\phi \in C_0^1(\mathbb{R} \times]0, \infty[\,; \mathbb{R}^m)$, we obtain

$$q_t + f(q)_x = 0 \tag{2.19}$$

for all $(x, t) \in \mathbb{R} \times]0, \infty[$. So q satisfies (2.12). We need to show that $q(x, 0) = q_0(x)$ for all $x \in \mathbb{R}$. To do this we can multiply (2.19) by a function $\phi \in C_0^1(\mathbb{R} \times \mathbb{R}_+; \mathbb{R}^m)$ and again proceed like in the construction of weak solution to find

$$\int_{\mathbb{R}} \int_0^\infty q \cdot \phi_t + f(q) \cdot \phi_x \, \mathrm{d}t\mathrm{d}x + \int_{\mathbb{R}} q(x, 0) \cdot \phi(x, 0) \, \mathrm{d}x = 0.$$

Comparing this with (2.18) gives us

$$\int_{-\infty}^{\infty} \big(q(x,0) - q_0(x)\big) \cdot \phi(x,0)\,\mathrm{d}x = 0.$$

Again this holds for all $\phi \in C_0^1(\mathbb{R} \times \mathbb{R}_+; \mathbb{R}^m)$. Thus $q(x,0) = q_0(x)$ for all $x \in \mathbb{R}$, so q is a classical solution of the initial value problem for (2.12). ■

We will end our study of weak solutions here, since we do not want to shift the focus of this text too far away from the numerical side. One can find more results in [16] or [10].

2.2.3 Existence Results

After introducing weak solutions it is natural to ask if there are any existence results for weak solutions of the initial value problem for (2.12). We want to present some results for the initial value problem for one-dimensional systems of conservation laws. First we take a look at the Riemann problem for (2.12). Let $q_\mathrm{l}, q_\mathrm{r} \in \Omega$. The initial value problem for (2.12) with the initial data

$$q_0(x) = \begin{cases} q_\mathrm{l} & \text{if } x < 0 \\ q_\mathrm{r} & \text{if } x > 0 \end{cases} \tag{2.20}$$

is called the *Riemann problem*. From now on we assume that (2.12) is strictly hyperbolic, so the Jacobian matrix $f'(q)$ of f has m real and distinct eigenvalues $\lambda_1(q) < \ldots < \lambda_m(q)$ for all $q \in \Omega$. Corresponding to each eigenvalue we have a right eigenvector $r_k(q) \in \mathbb{R}^m$ defined by

$$f'(q) r_k(q) = \lambda_k(q) r_k(q)$$

and a left eigenvector $\ell_k(q) \in \mathbb{R}^m$ defined by

$$\ell_k(q)^\mathrm{T} f'(q) = \lambda_k(u) \ell_k(u)^\mathrm{T}$$

for all $k \in \{1, \ldots, m\}$.

Definition 2.7 *Let $k \in \{1, \ldots, m\}$. The pair $\big(\lambda_k(q), r_k(q)\big)$ is called the k-th characteristic field.*

Definition 2.8 *The k-th characteristic field $\big(\lambda_k(q), r_k(q)\big)$ is called genuinely nonlinear if*

$$\nabla \lambda_k(q) \cdot r_k(q) \neq 0$$

for all $q \in \Omega$. If

$$\nabla \lambda_k(u) \cdot r_k(u) = 0$$

for all $q \in \Omega$, then we say the k-th characteristic field is linearly degenerated.

These definitions are needed to state the following local existence result for the Riemann problem for (2.12).

Theorem 2.9 *Suppose for all $k \in \{1, \dots, m\}$ the k-th characteristic field is either genuinely nonlinear or linearly degenerated. Let $q_\mathrm{l} \in \Omega$, then there exists a neigborhood $\mathcal{N} \subseteq \Omega$ of q_l, such that if $q_\mathrm{r} \in \mathcal{N}$, then the Riemann problem for (2.12) with initial data*

$$q_0(x) = \begin{cases} q_\mathrm{l} & \text{if } x < 0 \\ q_\mathrm{r} & \text{if } x > 0 \end{cases}$$

has a solution.

The proof of this theorem requires some further investigations into some theoretical aspects surrounding conservation laws. The last part was again based on [10] and [16], in which the reader can find the complete proof of the theorem. In these references the theorem also gives insight into the structure of the solution and states that a solution of this structure is unique. Based on this theorem James Glimm found a way to construct weak solution to an arbitrary initial value problem for (2.12) under some conditions on the initial data.

For the next part we need to define the total variation of a function. It measures the oscillatory behavior of a given function.

Definition 2.10 *Let $\mathcal{V} \subseteq \mathbb{R}^p$ with $p \in \mathbb{N}$ and $g \in L^1(\mathcal{V})$, then*

$$\mathrm{T.V}(g, \mathcal{V}) := \sup_{||\phi|| \leqslant 1} \int_{\mathcal{V}} g(x) \operatorname{div} \phi(x) dx$$

where $\phi \in C_0^1(\mathcal{V}; \mathbb{R}^p)$.

The definition is from [8]. The general idea is that if the L^∞-norm of the initial data is small and it has small total variation then we can approximate the solutions with local Riemann problems. Therefore we define a mesh and approximate the initial data by a piecewise constant function on this mesh. If the constant values of these functions are close enough we can solve Riemann problems in some small time interval. Then we can again approximate the solutions of the Riemann problems at the end of the time interval by a piecewise constant function and do the same thing again. So we can get an approximate solution for all time. If we let our mesh width go to zero, the obtained approximate solution converges to a weak solution of the initial value problem for (2.12). The proof consists of two main parts. First it has to be ensured that we can solve the local Riemann problems for all time. That means the values of the piecewise constant functions need to be near each other. Secondly we need to show that the approximate solution does indeed converge to a weak solution. If this is done, we obtain the following theorem.

Theorem 2.11 *Let (2.12) be strictly hyperbolic and genuinely nonlinear in each characteristic field. There exist constants $C_1, C_2 > 0$ such that if*

(a) $||q_0||_\infty \leqslant C_1$

(b) $\mathrm{T.V}(q_0, \mathbb{R}) \leqslant C_2$,

then a weak solution of the initial value problem for (2.12) exists for all time.

Again we omitted some parts of the theorem, they together with the detailed proof can be found in the original paper of James Glimm [9]. Proofs of the existence theorem in a somewhat simpler form are also contained in [16] and [6]. At the end of this subsection we want to note that there exist stronger results for the special case of scalar one-dimensional conservation laws. For more details we refer to [7] and [10]. But the general statement is that if the flux function f is smooth and uniformly convex, then there exists a weak solution of the initial value problem (2.13). In this case we call a weak solution q an entropy solution if

$$u(x+z,t) - u(x,t) \leqslant C\left(1 + \frac{1}{t}\right) z$$

holds for some constant $C \geqslant 0$ and for almost all $x, z \in \mathbb{R}$ and $t > 0$ with $z > 0$. It can be shown that for the initial value problem for the scalar conservation laws (2.13) a unique entropy solution exists.

2.3 Three Model Conservation Laws

In this subsection we list three very common hyperbolic conservation laws. They will serve as the test cases for the numerical experiments later on. More details and derivations for all of the following differential equations can be found in the book [13] by Logan.

2.3.1 Linear Advection

Just about the simplest possible conservation law is the *linear advection equation*

$$u_t + a u_x = 0 \tag{2.21}$$

where $a \in \mathbb{R}$ is some constant (preferably with $a \neq 0$) and $(x,t) \mapsto u(x,t)$ is the unknown function. In terms of the conservation law framework we have $q := u$, $f(q) := aq$ and $\psi = 0$. So this is a linear system and the coefficient matrix $A := (a) \in \mathbb{R}^{1\times 1}$ is clearly diagonalizable, making the equation hyperbolic. The advection equation is usually combined with an initial condition of the form

$$u(x,0) = u_0(x) \tag{2.22}$$

where $x \mapsto u_0(x)$ is a given function.

The advection equation models the flow of a substance (e. g. a chemical) that is being carried along in the movement of a fluid (e. g. water in a tube). In this sense, u should be thought of as the density of the substance. In practical applications, advection famously shows up in the advection-diffusion equation. Studying just the advection term on its own has some value here to better understand the behavior of the entire advection-diffusion system.

We can use the method of characteristics from the previous subsection to construct a solution to the initial value problem (2.21), (2.22) for the linear advection equation. Since $f'(q) = a$, we have $t^* = 0$ and so $T^* = \infty$. Therefore a solution of the initial value problem for the linear advection equation exists for all time and can be constructed with the method of characteristics, cf. 2.2.1. Its solution is given as

$$u(x,t) = u_0(x^*),$$

where x^* is the solution of

$$x = x^* + ta\big(u_0(x^*)\big) = x^* + ta.$$

So the solution of the initial value problem is

$$u(x,t) = u_0(x - at).$$

This means the initial condition propagates to the right with constant speed a. Keeping in mind the physical model, this solution should not come as a surprise.

2.3.2 Burgers' Equation

As the name already suggests, the advection equation is linear. The simplest nonlinear conservation law is the (inviscid) *Burgers equation*

$$u_t + uu_x = 0 \tag{2.23}$$

where, once again, $(x,t) \mapsto u(x,t)$ is the unknown function. In terms of the conservation law framework we have $q := u$, $f(q) := \frac{1}{2}q^2$ and $\psi = 0$ because (2.3.2) is equivalent to

$$u_t + (\tfrac{1}{2}u^2)_x = 0$$

by the chain rule. This also shows that we are dealing with a quasilinear system with matrix $A(q) = (q) \in \mathbb{R}^{1\times 1}$. This, too, is always diagonalizable, making the equation hyperbolic. Burgers' equation is also typically combined with an initial condition of the form (2.22).

While Burgers' equation does have some practical applications in the field of traffic flow, for example, it is mostly studied because of its theoretical properties. It is

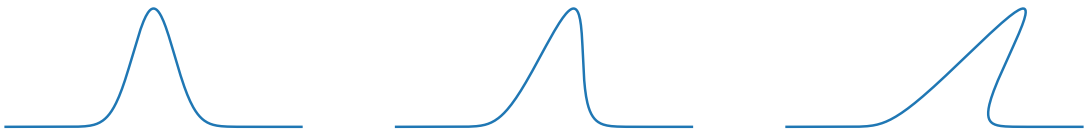

Figure 2.1: Solution of Burgers' equation with Gaussian-shaped initial data. After some time the wave tips over itself and the true solution becomes multi-valued.

the simplest differential equation in which continuous initial conditions can become discontinuous as time passes.

Like before we can use the method of characteristics on the initial value problem for Burgers' equation. Let $u_0 : \mathbb{R} \to \mathbb{R}$ be the initial condition. We have $f'(q) = q$, so

$$t^* = \min_{x\in\mathbb{R}} \frac{\mathrm{d}}{\mathrm{d}x} u_0(x).$$

If u_0 is monotonically increasing, then we get $T^* = \infty$ and the solution of the initial value problem of Burgers' equation is implicitly defined by

$$u(x,t) = u_0(x^*), \qquad x_* = x - tu_0(x^*).$$

If u_0 is not monotonically increasing, a continuously differentiable solution of the initial value problem only exists up to the time $T^* < \infty$. Up to this time the solution is implicitly given like before. This is all we can say at the moment about the solution of the problem because we need to know the function u_0 explicitly to solve the equations above.

2.3.3 The Shallow-Water Equations

A popular nonlinear system of conservation laws are the *two-dimensional shallow-water equations*

$$\begin{aligned} h_t + (hu)_x + (hv)_y &= 0, \\ (hu)_t + (hu^2 + \tfrac{1}{2}gh^2)_x + (huv)_y &= -ghb_x, \\ (hv)_t + (huv)_x + (hv^2 + \tfrac{1}{2}gh^2)_y &= -ghb_y. \end{aligned}$$

They describe the motion of certain types of waves. Here $(x, y, t) \mapsto h(x, y, t)$ is the height of a wave, $(x, y, t) \mapsto u(x, y, t)$ and $(x, y, t) \mapsto v(x, y, t)$ the velocities of the wave in the x and y direction respectively. The function $(x, y) \mapsto b(x, y)$ models the ground underneath the water. Also $g := 9.81$ is the gravitational constant. Figure 2.2 shows a visual representation of these functions in a setting reminiscent of an ocean. For numerical tests, it is common to drop the y direction and instead view the *one-*

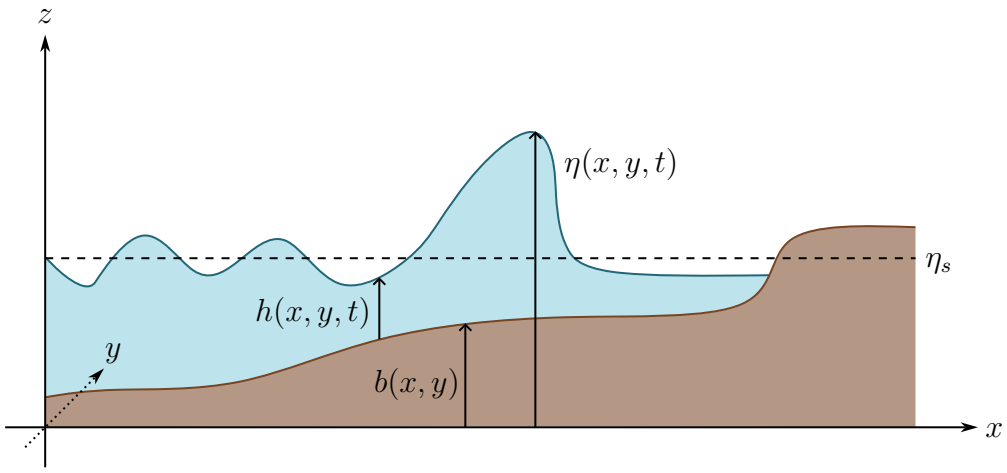

Figure 2.2: Geometry of the shallow-water model for tsunamis.

dimensional shallow-water system

$$h_t + (hu)_x = 0, \tag{2.24}$$

$$(hu)_t + (hu^2 + \tfrac{1}{2}gh^2)_x = -ghb_x. \tag{2.25}$$

Here the unknown functions are $(x, t) \mapsto h(x, t), (x, t) \mapsto u(x, t)$, while $x \mapsto b(x)$ is given. The first equation (2.24) models conservation of mass while the second one (2.25) describes conservation of momentum.

Both versions can be put into conservation law form. We will illustrate this for the one-dimensional system. To this end, define the vector $q := (q_1, q_2)^{\mathrm{T}} := (h, hu)^{\mathrm{T}}$ for the conserved variables and

$$f(q) := \begin{pmatrix} q_2 \\ q_2^2/q_1 + \frac{1}{2}gq_1^2 \end{pmatrix} = \begin{pmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \end{pmatrix}, \quad \psi(q, x) := \begin{pmatrix} 0 \\ -gq_1 b_x \end{pmatrix} = \begin{pmatrix} 0 \\ -ghb_x \end{pmatrix}$$

for the flux and the source term. The Jacobian matrix is given by

$$f'(q) = \begin{pmatrix} 0 & 1 \\ -(q_2/q_1)^2 + gq_1 & 2q_2/q_1 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -u^2 + gh & 2u \end{pmatrix}.$$

Simple linear algebra shows that its eigenvalues are

$$\lambda_1 = u - \sqrt{gh} \quad \text{and} \quad \lambda_2 = u + \sqrt{gh}$$

with corresponding eigenvectors

$$r_1 = \begin{pmatrix} 1 \\ u - \sqrt{gh} \end{pmatrix} \quad \text{and} \quad r_2 = \begin{pmatrix} 1 \\ u + \sqrt{gh} \end{pmatrix}.$$

For physically-relevant depths $h \geqslant 0$ the eigenvalues remain real-valued and as long as $h > 0$ they are actually distinct. In particular, this shows that the shallow-water equations are hyperbolic. Initial conditions need to be provided for both $h(x, 0) := h_0(x)$ and $u(x, 0) := u_0(x)$ (or possibly $(hu)(x, 0) := (hu)_0(x)$, if one is working with the conservation form). As stated before, we only consider the case with vanishing source term in this report. Physically speaking, this means that the function b is constant, i. e., the bottom of the ocean over which the wave passes is assumed to be flat.

To derive the shallow-water system one starts by considering a more general set of differential equations (the Navier-Stokes equations). Then one assumes that the vertical length scale of the wave (i. e. the unit of h) is much smaller than the horizontal length scale(s) — this gives rise to the "shallow" part in the name of the equations. Because of this assumption, it is possible to get rid of the contribution in the z direction. Hence, the shallow-water model is much easier to solve numerically. It seem to strike a good balance between being sufficiently easy to solve to guarantee fast computation and between being sufficiently complex to model the real world accurately enough.

The shallow-water equations are frequently used in real-world applications. They are, for example, the standard model for the type of wave that gets generated by an undersea earthquake: tsunamis. Another popular application is the flow that results when the wall of a large dam collapses. It is worth noting that other (incompressible) fluids like gasses can also be modeled by the shallow-water system. The equations' name is somewhat misleading in this sense.

3 Numerical Tests

In this section we want to test physics-informed neural networks (PINNs) to predict solutions of initial value problems for one-dimensional systems of hyperbolic conservation laws. Let Ω be an open subset of $\mathbb{R}^m$, $m \in \mathbb{N}$ and $f : \Omega \to \mathbb{R}^m$ a sufficiently smooth function. Recall the initial value problem (IVP) was defined as follows: We want to find a function $q : \mathbb{R} \times [0, \infty[\to \Omega$ that satisfies

$$q_t + f(q)_x = 0,$$

for all $x \in \mathbb{R}$, $t > 0$ and

$$q(x, 0) = q_0(x)$$

for all $x \in \mathbb{R}$, where $q_0 : \mathbb{R} \to \mathbb{R}$ is some given function. We will compare the approximate solutions to the analytical solutions of the IVP, given they are available. When this is not possible, we require a different way to obtain the "true" solution to compare with. The classical numerical methods for solving IVPs for conservation laws are the finite-difference and the finite-volume method, see [11]. We do not have the time to elaborate on the inner workings of the methods that were used here. Details can be found in the code and the references. The reader may take these algorithms as more or less a black box with which we are able to produce sufficiently accurate approximations to the exact solution.

As one can see above, the IVP is defined over the whole space $x \in \mathbb{R} \times [0, \infty[$. To solve the problem numerically we need to restrict the IVP to some finite space-time domain $[x_{\min}, x_{\max}] \times [0, T]$ where $x_{\min}, x_{\max} \in \mathbb{R}$ and $T > 0$, i. e., we need to introduce boundary conditions. Numerically, the problem that we want to solve is then defined as follows: We want to find a function $q : [x_{\min}, x_{\max}] \times [0, T] \to \Omega$ that satisfies

$$q_t + f(q)_x = 0, \tag{3.1}$$

for all $x \in]x_{\min}, x_{\max}[$, $t \in]0, T[$ and

$$q(x, 0) = q_0(x)$$

for all $x \in [x_{\min}, x_{\max}]$, where $q_0 : [x_{\min}, x_{\max}] \to \mathbb{R}$ is some given function. Also

$$q(x_{\min}, t) = q_{\mathrm{l}}(t), \qquad q(x_{\max}, t) = q_{\mathrm{r}}(t)$$

for $t \in [0, T]$ for some given functions $q_{\mathrm{r}}, q_{\mathrm{l}} : [0, T] \to \mathbb{R}$. Physics-informed neural networks are similar to standard neural networks with the key difference that we include a term which accounts for the conservation law into the loss function to ensure that the neural network satisfies this term. In our case the loss function consists of three parts: the conservation law L_{CL}, the initial data L_{ID} and the boundary data L_{BD}. Let $N_{\mathrm{CL}}, N_{\mathrm{ID}}, N_{\mathrm{BD}} \in \mathbb{N}$, then we have

$$L_{\mathrm{CL}} = \frac{1}{N_{\mathrm{CL}}} \sum_{i=1}^{N_{\mathrm{CL}}} ||u_t(x_i, t_i) + f(u(x_i, t_i))_x||^2,$$

where u is the current prediction of the PINN and $\{(x_i, t_i) \mid i = 1, \ldots, N_{\mathrm{CL}}\}$ are some given points, called the collocation points. Also we have

$$L_{\mathrm{ID}} = \frac{1}{N_{\mathrm{ID}}} \sum_{i=1}^{N_{\mathrm{ID}}} ||u(x_i, 0) - q_0(x_i)||^2$$

and

$$L_{\mathrm{BD}} = \frac{1}{2N_{\mathrm{BD}}} \sum_{i=1}^{N_{\mathrm{BD}}} ||u(x_{\min}, t_i) - q_{\mathrm{l}}(t_i)||^2 + ||u(x_{\max}, t_i) - q_{\mathrm{r}}(t_i)||^2,$$

where $\{(x_i, 0) \mid i = 1, \dots, N_{\text{ID}}\}$, $\{u(x_{\min}, t_i) \mid i = 1, \dots, N_{\text{BD}}\}$ and $\{u(x_{\max}, t_i) \mid i = 1, \dots, N_{\text{BD}}\}$ are given points at the boundary of $[x_{\min}, x_{\max}] \times [0, T]$. For simplicity, we will call these points the data points. Then the loss function L is given by

$$L = \lambda_1 L_{\text{CL}} + \lambda_2 (L_{\text{ID}} + L_{\text{BD}}),$$

where $\lambda_1, \lambda_2 > 0$ are parameters that are set before the training. More details can be found in [14] and [5].

In the next subsection we want to predict the solutions of initial value problems for the conservation laws that were introduced in the section before, namely the linear advection equation, Burgers' equation and the shallow-water equations, by training PINNs. We will use the Adam optimizer. Also we set $\lambda_1 = \lambda_2 = 1$. For simplicity we only use this setup for our tests in the next subsections, but we note that different setups could yield better results. We will usually choose the collocation points and the data points randomly in each training iteration. It is important to note, as we will see later, that the way we choose these points can have an impact on the accuracy of the predicted solutions. The code we use for the PINNs relies on PyTorch and is based on [5]. We will train the PINNs on a NVIDIA T4 GPU and we compute the solutions of the finite-volume and finite-difference methods on an INTEL XENON CPU. The focus of this text is mostly to test the accuracy of the predicted solutions since it only makes sense to look at the computing time if the predictions are accurate. However, we will still state the times needed to train the networks.

3.1 Linear Advection Equation

We start by testing PINNs to solve two initial value problems for the linear advection equation

$$u_t(x, t) + 0.5 u_x(x, t) = 0.$$

For both IVPs we choose $x_{\min} = -2$, $x_{\max} = 2$, $T = 1$ and the following boundary conditions

$$u(-2, t) = u(2, t) = 0.$$

For the first IVP we use the initial condition

$$u(x, 0) = \exp(-2x^2)$$

and for the second

$$u(x, 0) = \exp\big(-200(x + 0.5)^2\big) + \exp\big(-2000(x - 0.3)^2\big).$$

3.1.1 First Initial Value Problem

We start with the first IVP. Using the method of characteristics we get that the analytical solution of the problem is given by

$$u(x, t) = \exp\big(-2(x - 0,5t)^2\big).$$

We predict the solution of the first IVP by training a PINN with 3 hidden layers. Each of the three hidden layers contains 20 nodes and a hyperbolic tangent activation

function. We train the network 100 times at 650 points which are randomly chosen in each training iteration. These consists of 500 points in $[x_{\min}, x_{\max}] \times [0, T]$, 50 points on the line $t = 0$ and 50 points each on the left and right boundary, i. e. $x = x_{\min}$ and $x = x_{\max}$. In figure 3.1 one can see the prediction of the neural network (blue graph) in comparison with the analytical solution (orange dotted graph). We can see that the solution is already approximated fairly well. We can increase the accuracy if we train the network more often or use more data and collocation points. In figure 3.2 one can see the prediction after training the PINN 100 times at 2600 points, i. e. 2000 collocation points and 200 points on the line $t = 0$ and 200 points each on the left and right boundary. The prediction after training the neural network 1000 times at 2600 points can be seen in figure 3.3. We can see that there is no real difference by using 650 or 2600 points and training the network 100 times, but the prediction after training the network 1000 times is very accurate. Training the network 100 times takes in our tests approximately 1 second. Training the network 1000 times takes circa 7 seconds. We need to note these are only approximations since the training and computing time can vary and depends on the device on which one computes the solutions.

Now we want to predict the solution by training a PINN with 5 hidden layers which contains 50 nodes and a hyperbolic tangent activation function each. The prediction after training the neural network 1000 times at 2600 points can be seen in figure 3.4 and the prediction after training the neural network 3000 times at 11 500, i. e. 10000 collocation points and 500 points on the line $t = 0$ and 500 points each on the left and right boundary can be seen in figure 3.5. Training the network 3000 times takes in our test approximately 34 seconds.

In figure 3.6 one can see the solution computed by the finite-volume method (blue graph) at the given times and at 200 spatial mesh points in comparison with the analytical solution (orange dotted graph). In our test it takes approximately 0.2 seconds to compute the solution with the finite-volume method. Only the prediction after training the PINN with 5 hidden layers 3000 times at 11500 points is as accurate as the solution computed by the finite-volume method.

3.1.2 Second Initial Value Problem

We will now test and compare PINNs and the finite-volume method by solving the second IVP for the linear advection equation stated before. This is more interesting, since here we have a very narrow spike and classical numerical methods sometimes develop problems approximating it. The analytical solution computed with the method of characteristics is

$$u(x,t) = \exp\big(-200(x + 0.5 - 0.5t)^2\big) + \exp\big(-2000(x - 0.3 - 0.5t)^2\big).$$

We begin by predicting the solution of this IVP by training a PINN with 3 hidden layers with 20 nodes and a hyperbolic tangent activation function each 1000 times at 2600 points like before. The results at different times can be seen in figure 3.7, where again the blue graph represents the solution predicted by the neural network and the orange dotted graph the analytical solution. We can see that the prediction is not very accurate, the second narrow spike is not even predicted at all. Reasons for that can be using too few points at which we train the network and training the network too few times. Therefore we will now train the network 2000 times at 11500

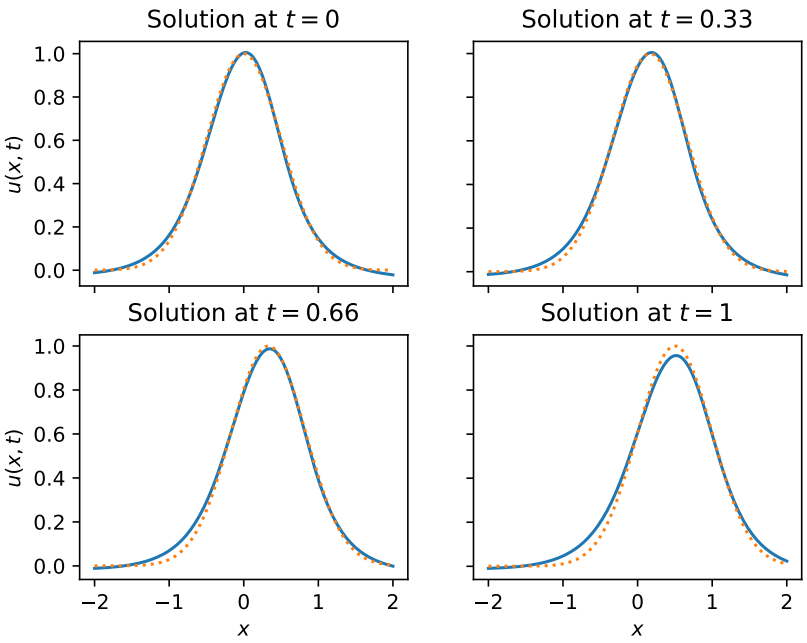

Figure 3.1: The prediction of the PINN after 100 training at 650 points.

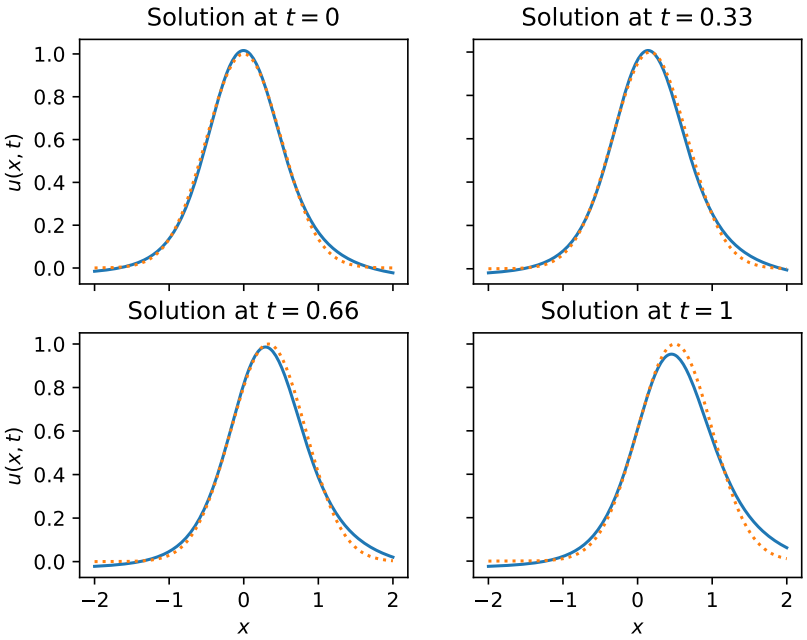

Figure 3.2: The prediction of the PINN after 100 training at 2600 points.

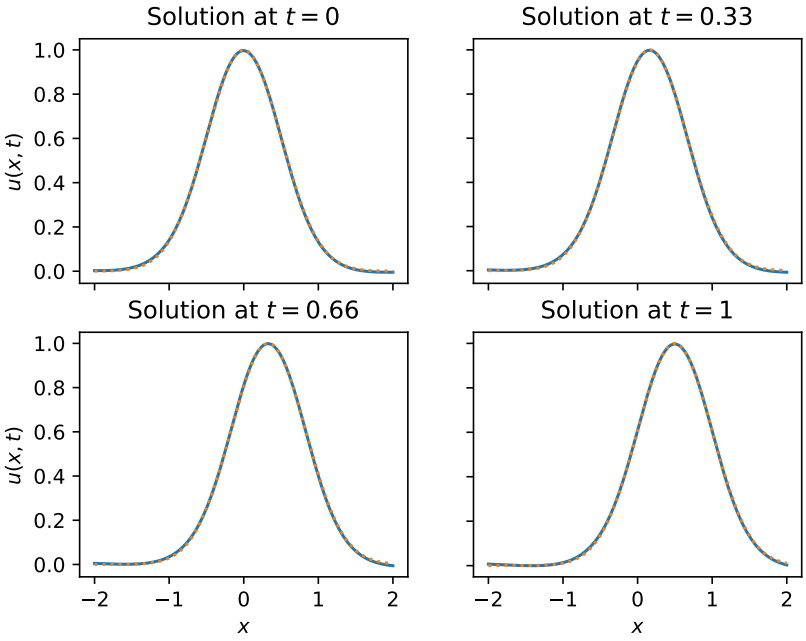

Figure 3.3: The prediction of the PINN after 1000 training at 2600 points.

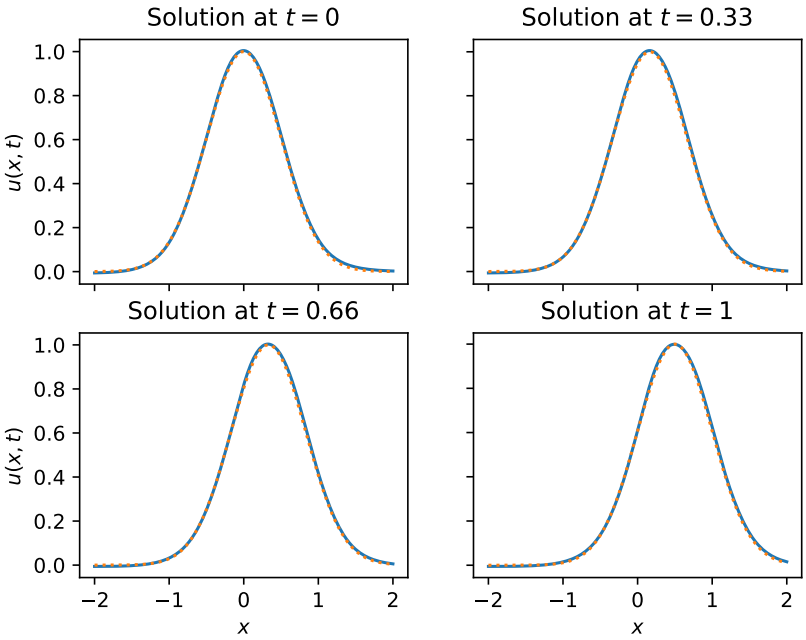

Figure 3.4: The prediction of a PINN with 5 hidden layers after 1000 training at 2600 points.

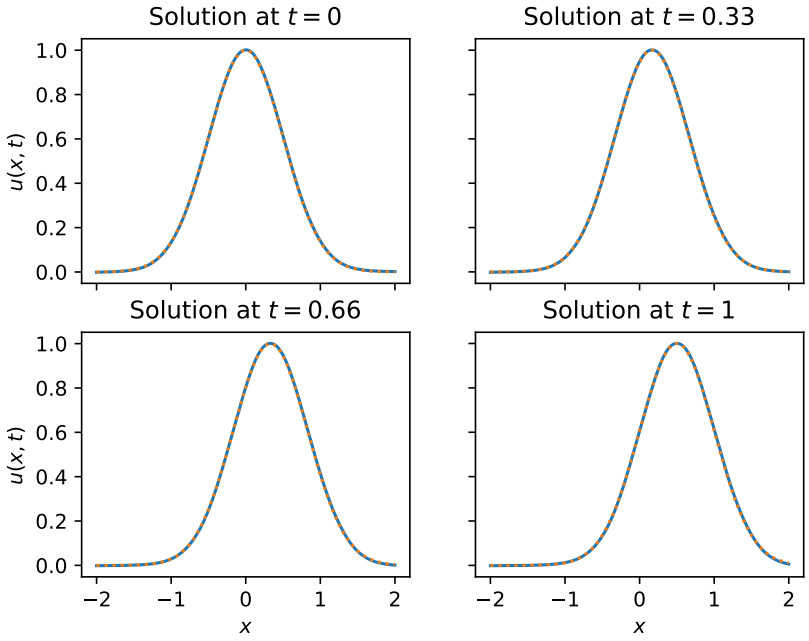

Figure 3.5: The prediction of a PINN with 5 hidden layers after 3000 training at 11500 points.

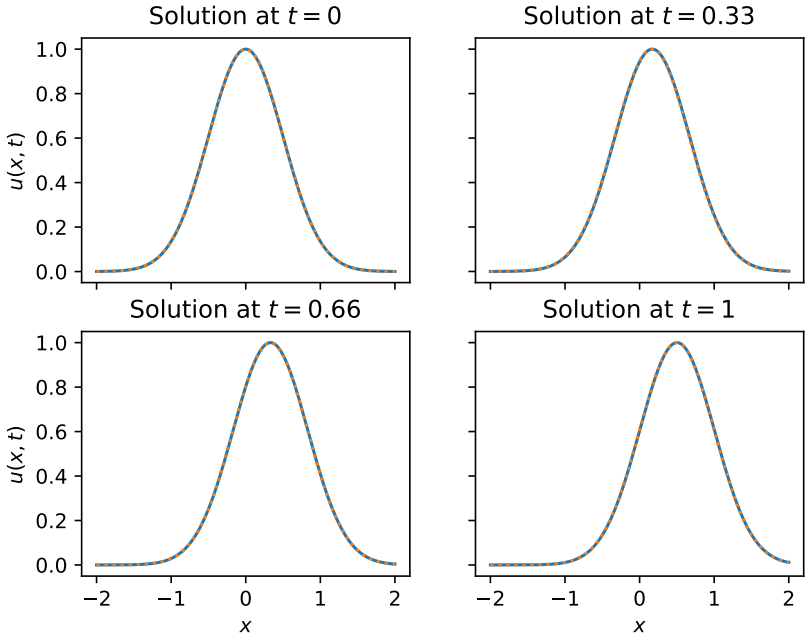

Figure 3.6: The solution computed by the finite-volume method.

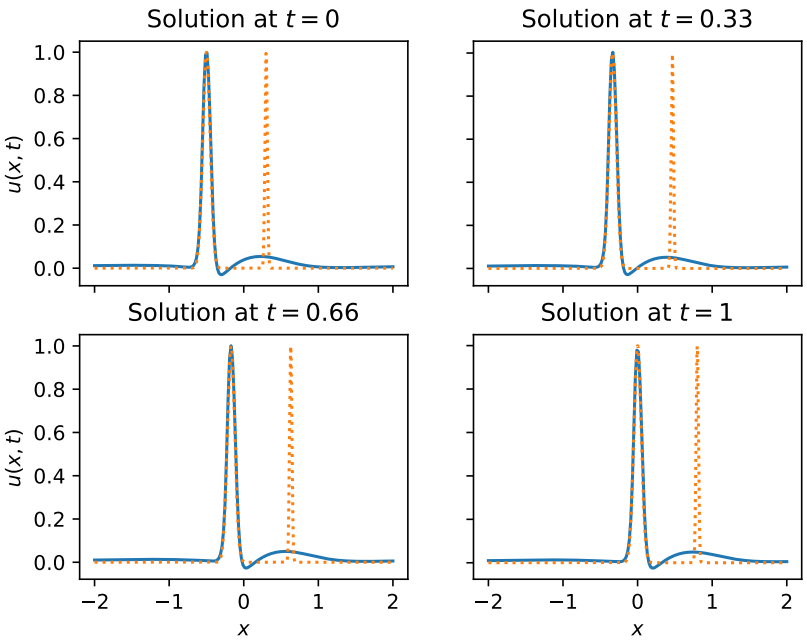

Figure 3.7: The solution predicted after training a PINN 1000 times at 2600 points.

points. The prediction can be seen in figure 3.8. We want to note that the predicted solution can sometimes be better or worse than the prediction shown here. This is a result of the nature of the neural networks. In figure 3.9 one can see the loss of the PINN at the different training times. One can observe that the loss oscillates quite strongly. Therefore we want to present a way to get rid of the oscillations. Instead of randomly chosen points at the boundary we now choose fixed and equidistant points. The prediction of the neural network after 2000 training iterations can be seen in figure 3.10. This prediction has similar accuracy as the prediction before, but the loss does not oscillates as much as before, as seen in figure 3.11. The training time in both cases was approximately 14.5 seconds.

Now we predict the solution by training a PINN with 5 hidden layers. Each of the hidden layers contain 50 nodes and a hyperbolic tangent activation function. The prediction after training the network 5000 times at 29000 points, i. e. 20000 collocation points, 5000 points at the line $t = 0$ and 2000 points each at the boundary, with fixed data points can be seen in figure 3.12. The training of the network takes approximately 60 seconds.

At this point we want to test how the finite-volume method deals with this kind of problem. We approximate the solution at 200 spatial mesh points. The results can be seen in figure 3.13. As we can see the narrow spike is not approximated very well. This effect is called dispersion and it can be fixed by increasing the number of mesh points. In figure figure 3.14 one can see the approximations at 1500 mesh points. At this number of points the dispersion effect is barely noticeable. Around the time $t = 1$ there is still a little bit of dispersion visible. The finite-volume method takes over 15 seconds to approximate the solution in figure 3.14.

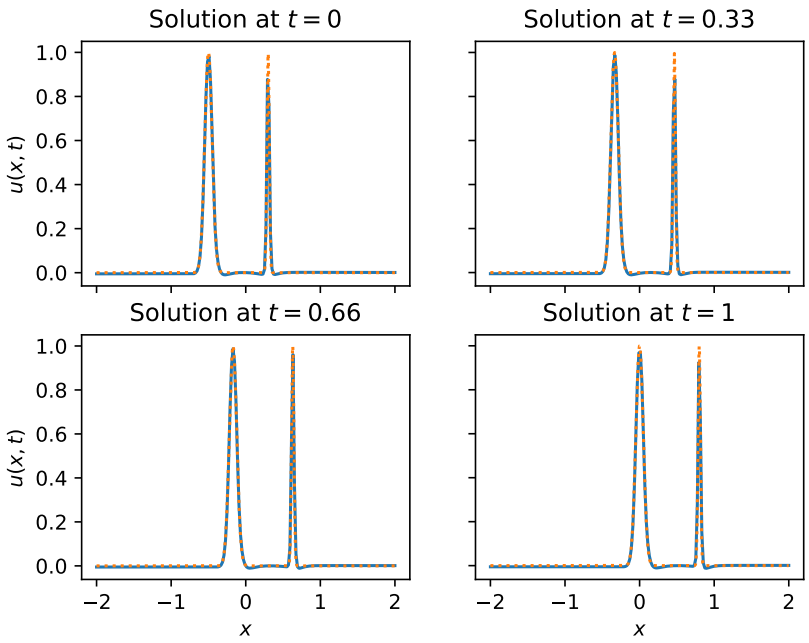

Figure 3.8: The solution predicted after training a PINN 2000 times at 11500 points.

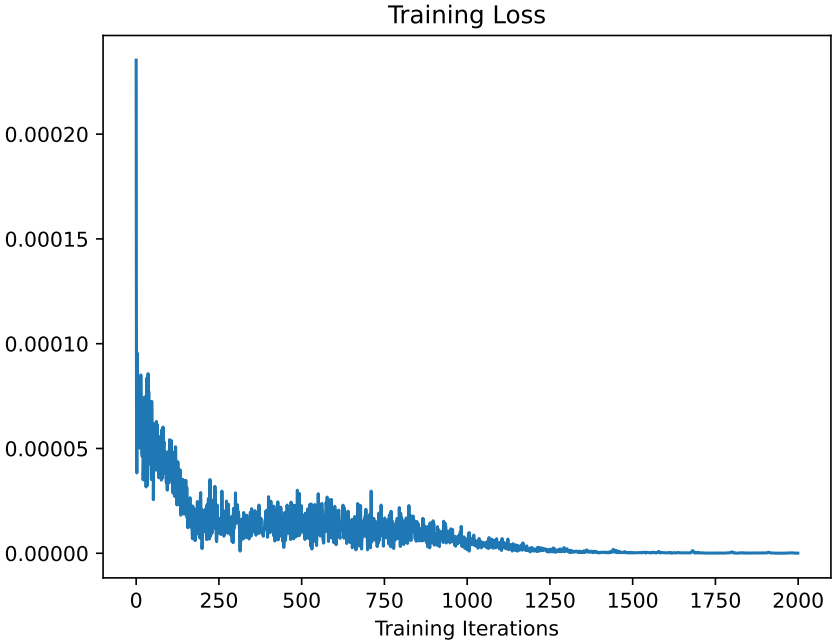

Figure 3.9: The loss at different training iterations.

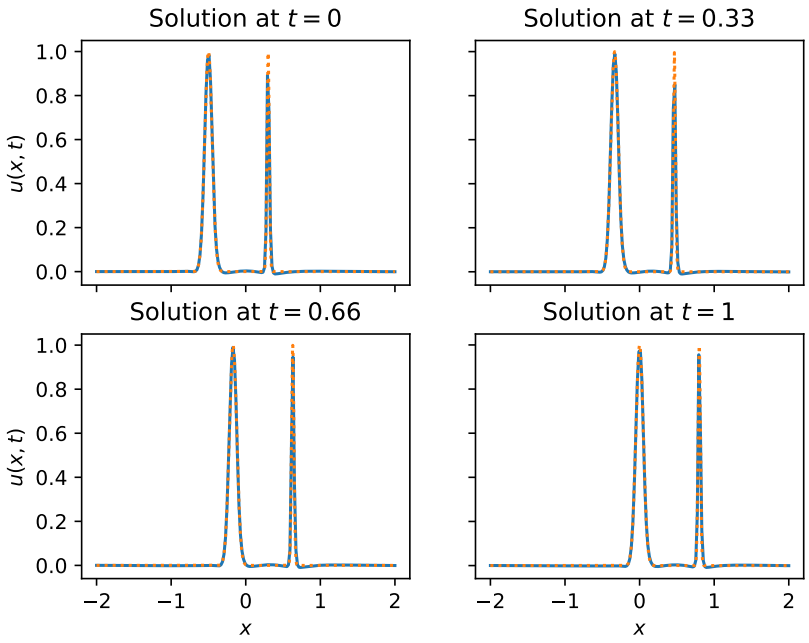

Figure 3.10: The solution predicted by the PINN after training 2000 times at fixed data points.

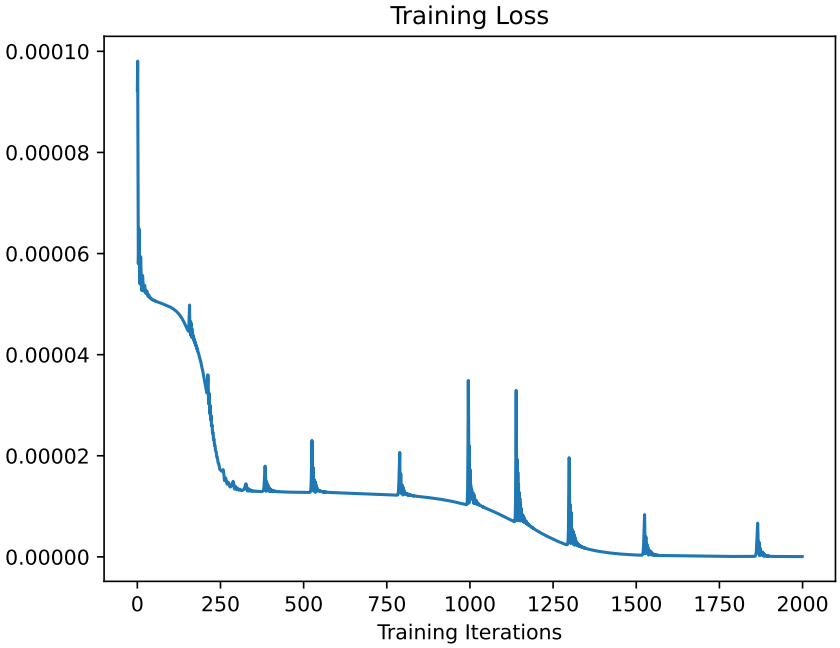

Figure 3.11: The loss at different training iterations.

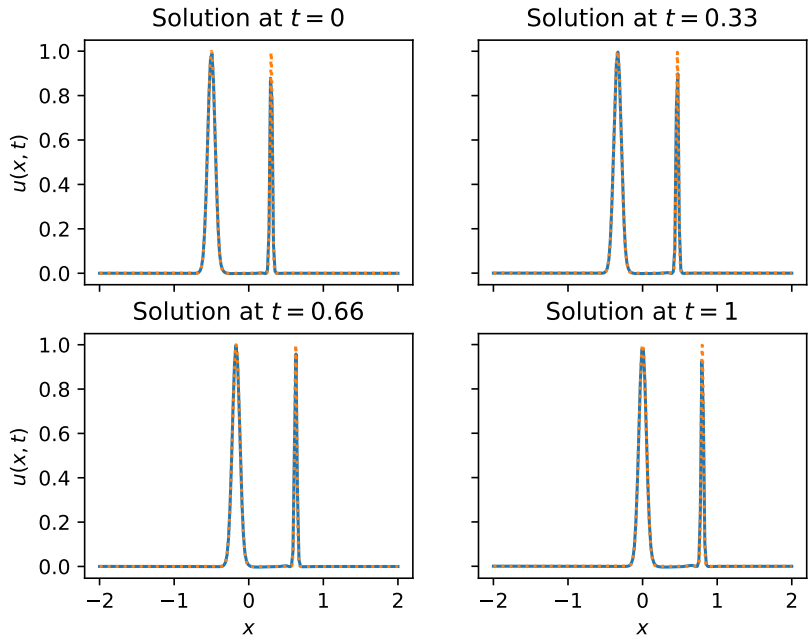

Figure 3.12: The solution predicted by a PINN with 5 hidden layers after training 5000 times at fixed data points.

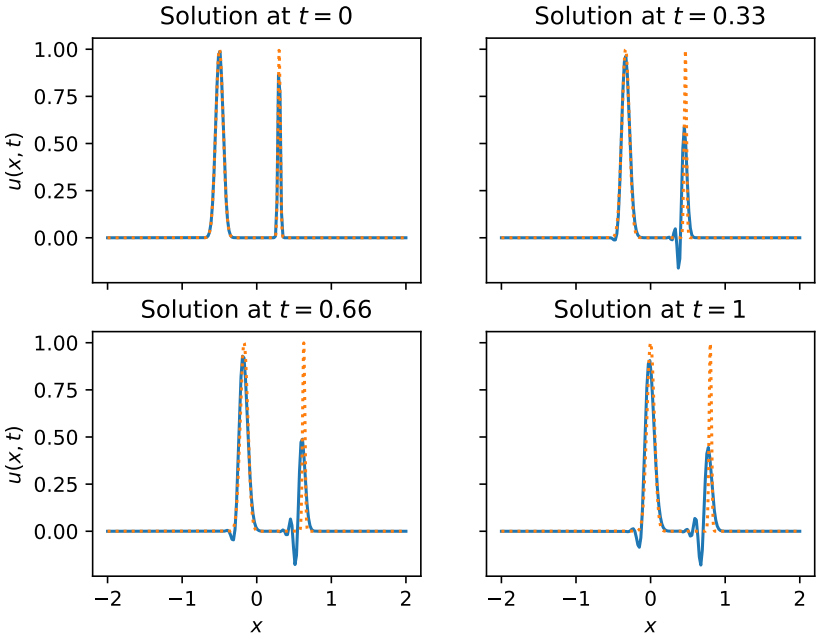

Figure 3.13: The solution computed by the finite-volume method with 200 spatial mesh points.

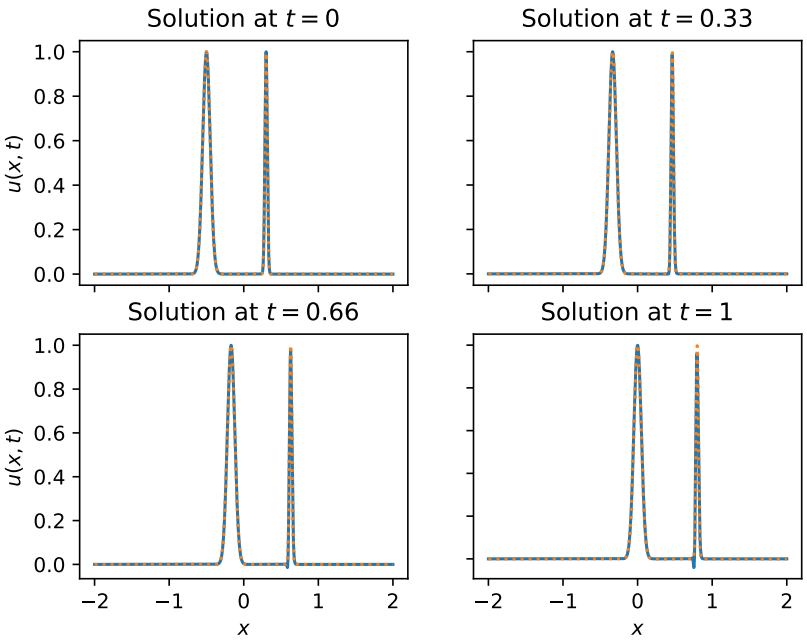

Figure 3.14: The solution computed by the finite-volume method with 1500 spatial mesh points.

3.2 Burgers' Equation

Now we want to take it a step further and test PINNs for Burgers' equation

$$u_t(x,t) + u_x(x,t)u(x,t) = 0.$$

Here we also want to consider two different IVPs. We start with smooth initial data and after that we consider a Riemann problem, i. e. discontinuous initial data. Since there are no easily computable analytical solutions for the IVPs that we are about to study, the finite-volume method will be used to produce approximations. From the discussion above it is clear that we must choose a sufficiently high number of grid points for the approximations to be accurate enough. In this case 200 mesh points were used.

3.2.1 First Initial Value Problem

For the first IVP we choose $x_{\min} = -2$, $x_{\max} = 2$,$T = 0.5$ and the following boundary conditions

$$u(-2,t) = u(2,t) = 0.$$

Also we choose the initial condition

$$u(x,0) = \exp(-2x^2).$$

By choosing $T = 0.5$ we ensure that a continuously differentiable solution exists. We start by training a PINN with 3 hidden layers which contain 20 nodes and a hyperbolic tangent activation function. The results after training the PINN 1000 times at 2600 randomly chosen points can be seen in figure 3.15 where the orange dotted graph is the approximate solution given by the finite-volume method. The training of the PINN takes up to 7 seconds in our tests and the finite-volume method needs just approximately 0.2 seconds to compute the solutions at the given times. As we can see, the predictions by the neural network are quite accurate. With more training at more points or more layers and nodes, the accuracy could be increased further. But we now want to take a look at a more interesting aspect. We want to test how the predictions behave if the solution of the IVP becomes discontinuous. For that we choose $T = 2$. With the method of characteristics one can compute that at this time no continuously differentiable solution of the IVP exists. We train the network again 1000 times at 2600 points. The results can be seen in figure 3.16. The prediction is not very accurate. To improve the accuracy we will train a PINN with 5 hidden layers. Each of the five hidden layers contains 50 nodes and a hyperbolic tangent activation function. We train the network 2000 times at 11500 points. The predictions can be seen in figure 3.17. The prediction is now more accurate but there is still a notable difference between the prediction and the solution computed by the finite-volume method. Now we will train a PINN with 7 hidden layers which contain 100 nodes and a hyperbolic tangent activation function. The predictions after training the network 5000 times at 21500 points, i. e. 20000 collocation points, can be seen in figure 3.18. The predictions of the network are still not very accurate. Training the network 5000 times takes approximately 118 seconds. The finite-volume method only needs under 0.5 second to compute the solution at the spatial mesh points at the given times.

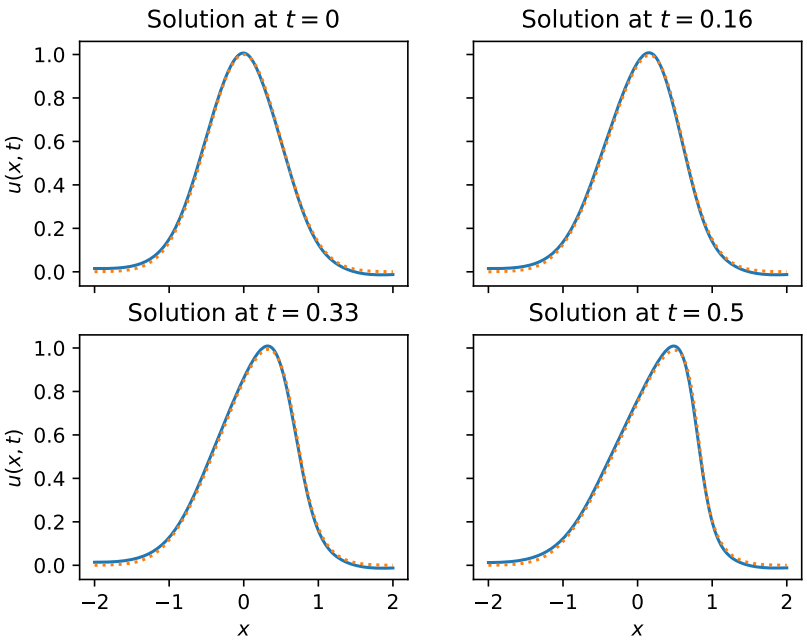

Figure 3.15: The prediction of the PINN after 1000 training.

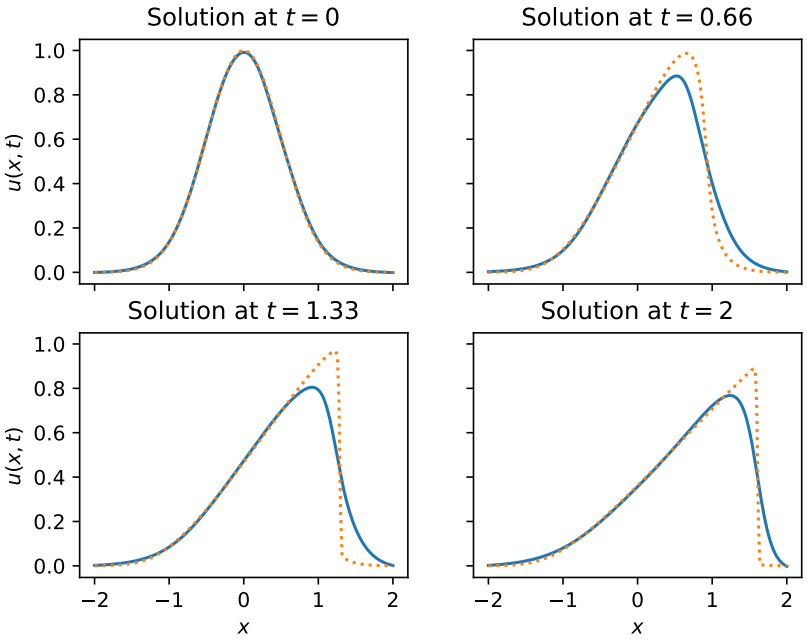

Figure 3.16: The prediction after training the PINN 1000 times.

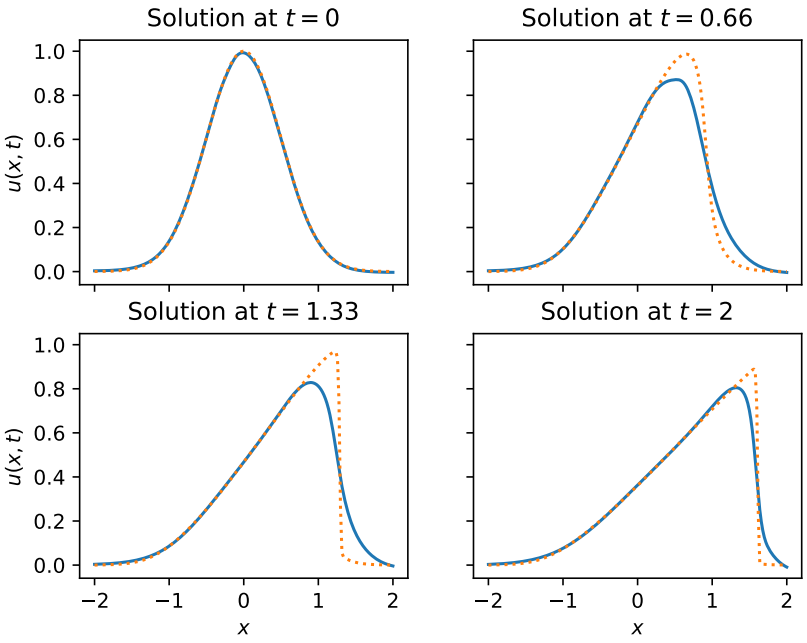

Figure 3.17: The prediction after training the network 2000 times at 11500 points.

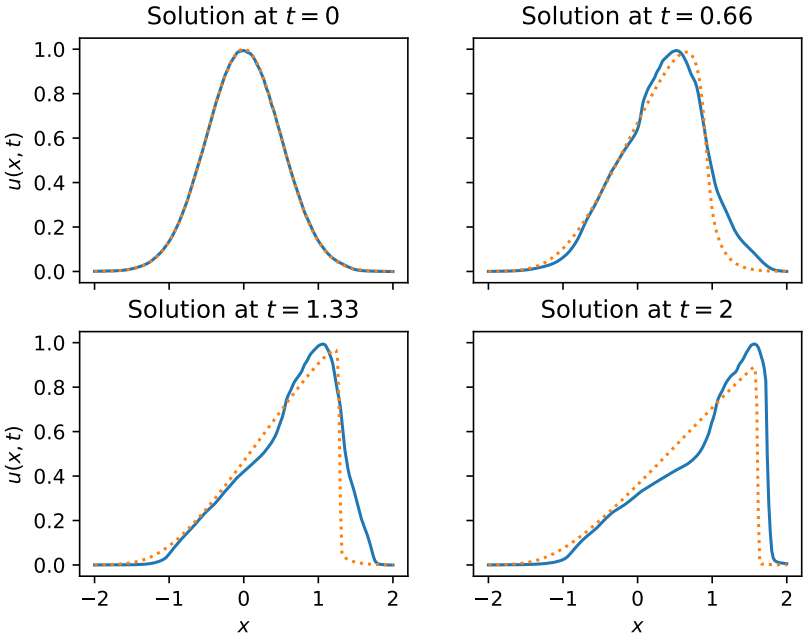

Figure 3.18: The prediction after training the network 5000 times at 21500 points.

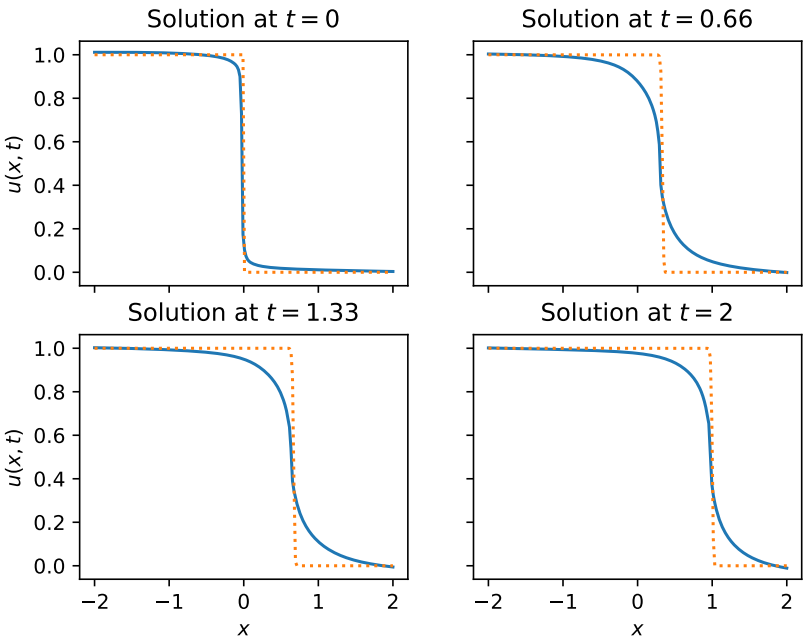

Figure 3.19: The prediction after training a PINN 2000 times at 11500 points.

3.2.2 Second Initial Value Problem

We will now take a look at a Riemann problem for Burgers' equation. Here we have not only a discontinuous solution but this time also discontinuous initial data:

$$u(x, 0) = \begin{cases} 1 & \text{if } x < 0 \\ 0 & \text{if } x > 0. \end{cases}$$

For this IVP we set $x_{\min} = -2$, $x_{\max} = 2$, $T = 2$ and the boundary conditions

$$u(x_{\min}, t) = 1, \qquad u(x_{\max}, t) = 0$$

The predictions after training a PINN with 5 hidden layers, like before, 2000 times at 11500 points, can be seen in figure figure 3.19. In figure 3.20 one can see the predicted solution after training a PINN with 7 hidden layers which contains 100 nodes each and a hyperbolic tangent activation function, 5000 times at 21500 points. Training the PINN 5000 at 21500 times takes in our test circa 120 seconds. The finite-volume method needs roughly 1.5 seconds to compute the solution at the given times. As seen before, the prediction is not very accurate.

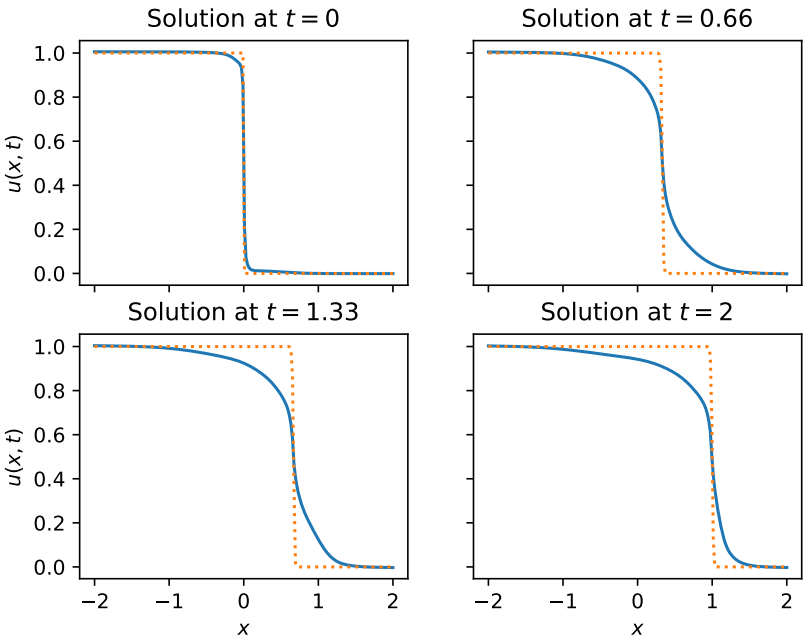

Figure 3.20: The prediction after training a PINN 5000 times at 21500 points.

3.3 Shallow Water Equations

At last we want to compare PINN and the finite-volume method by solving an IVP for a system of two conservation laws, the one-dimensional shallow water equations

$$h_t + (hu)_x = 0, \tag{3.2}$$
$$(hu)_t + (hu^2 + \tfrac{1}{2}gh^2)_x = -ghb_x. \tag{3.3}$$

We set $x_{\min} = 0$, $x_{\max} = 2$ and $T = 0.2$. As initial data for h we will use

$$h_0(x) = \begin{cases} 2 - b(x) & \text{if } x < 0 \\ 1 - b(x) & \text{if } x > 0 \end{cases}$$

for all $x \in [0, 2]$ and for u

$$u_0(x) = 0$$

for all $x \in [0, 2]$. For simplicity we set $b(x) = 0.5$. This IVP is called the dam break problem. Furthermore we set

$$h(x_{\min}, t) = 1.5, \quad h(x_{\max}, t) = 0.5$$

and

$$u(x_{\max}, t) = u(x_{\min}, t) = 0$$

for all $t \in]0, 0.2[$. Details on a finite-volume implementation for the shallow-water equations can be found in [17].

We start by training a PINN with five hidden layers. Each of the hidden layers contain 50 nodes and a hyperbolic tangent activation function. The predicted solutions for h and u (blue graphs) after training the network 2000 times at 11500 points can be seen in figure 3.21 (prediction for h) and figure 3.22 (prediction for u). The predicted solution after training 2000 times at 11500 points with fixed equidistant data points are depicted in figure 3.23 (prediction for h) and figure 3.24 (prediction for u). As before the orange dotted graph is the solution computed with the finite-volume method at the given times at 200 spatial mesh points. In our test the computation took circa 5 second. Training the PINN took in our test 30 seconds. At last, we want to train a PINN with 7 hidden layers. Each of the hidden layers contains 100 nodes and the same activation function as before. The predictions after training the network 5000 times at 21500 points with fixed data points can be seen in figure 3.25 and figure 3.26. The training took approximately 200 seconds.

Two main aspects we partly already encountered in the sections before are visible here. First of all the discontinuities which occur in the solutions are not predicted very accurately. Secondly, we can observe that it can make quite a difference if the points at $t = 0$ and at the boundary are randomly chosen or fixed and distributed equally. As we can see in the figure the difference between the two cases is quite large.

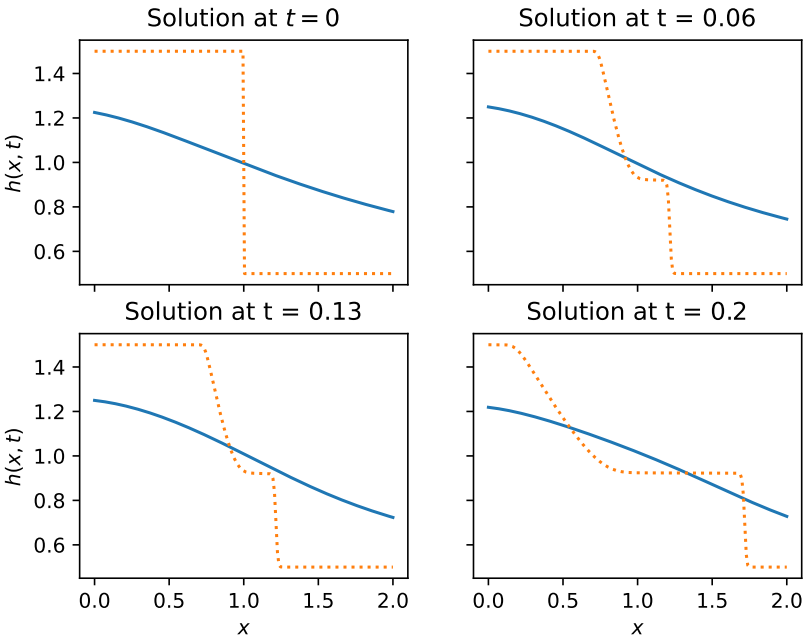

Figure 3.21: The prediction of h after 2000 training iterations with random data points.

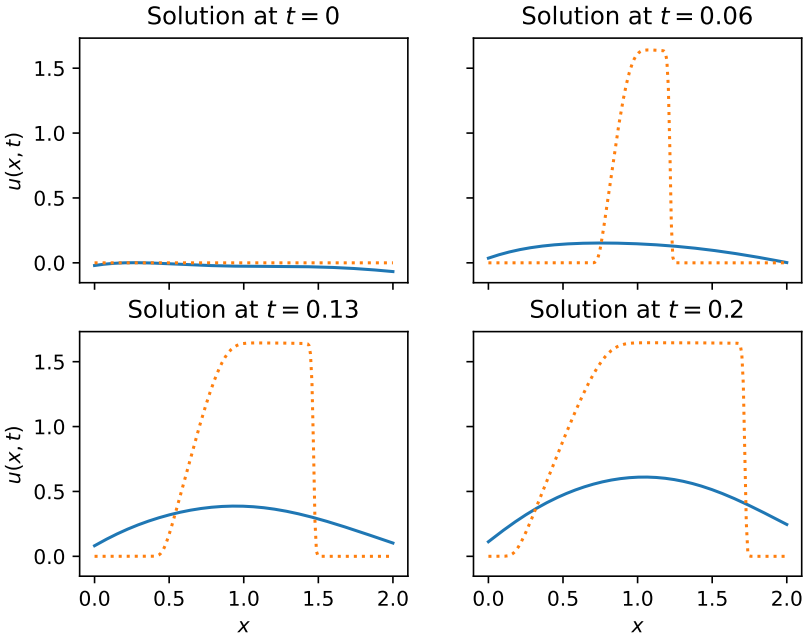

Figure 3.22: The prediction of u after 2000 training iterations with random data points.

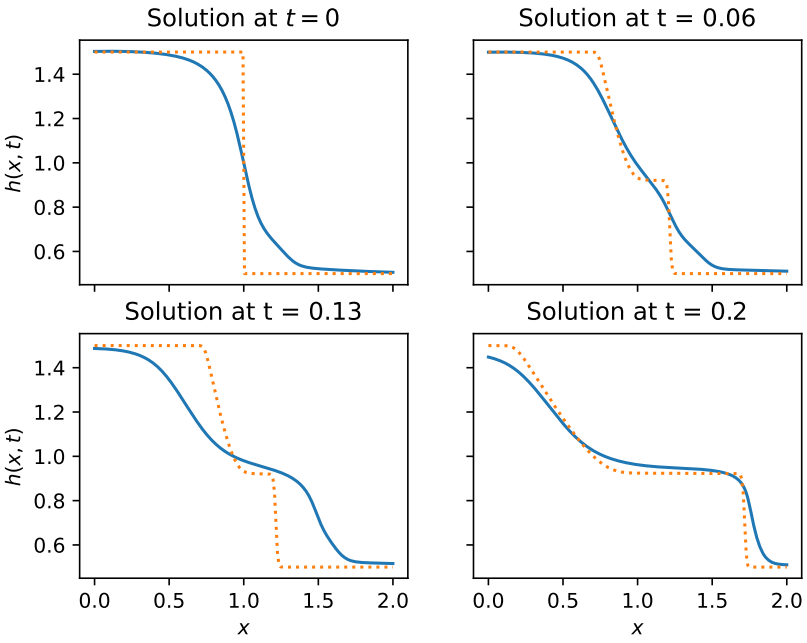

Figure 3.23: The prediction of h after 2000 training iterations with fixed data points.

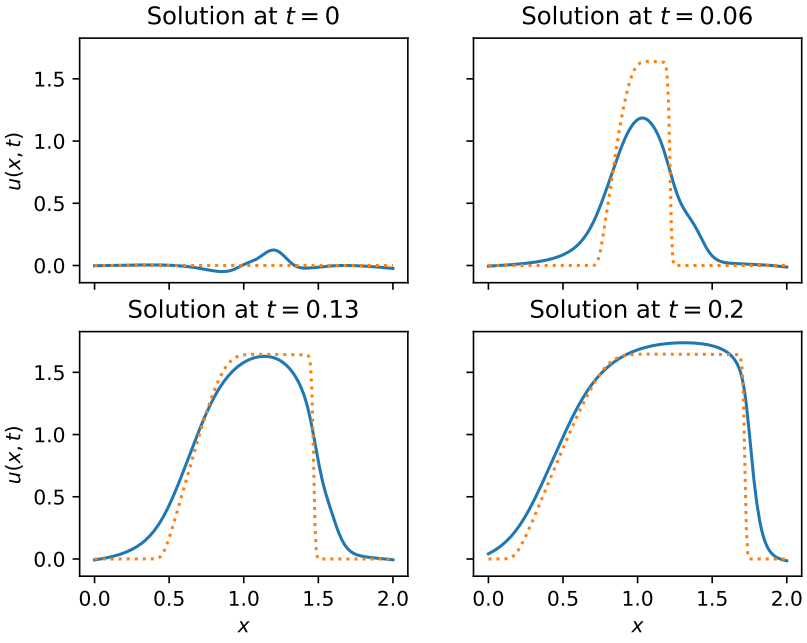

Figure 3.24: The prediction of u after 2000 training iterations with data fixed points.

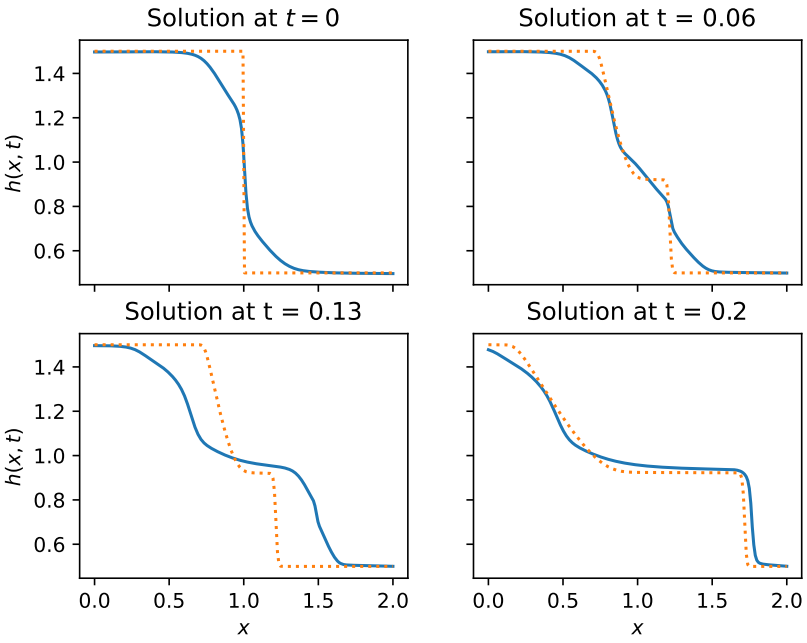

Figure 3.25: The prediction of h after training a PINN 5000 times at 21500 points with data fixed points.

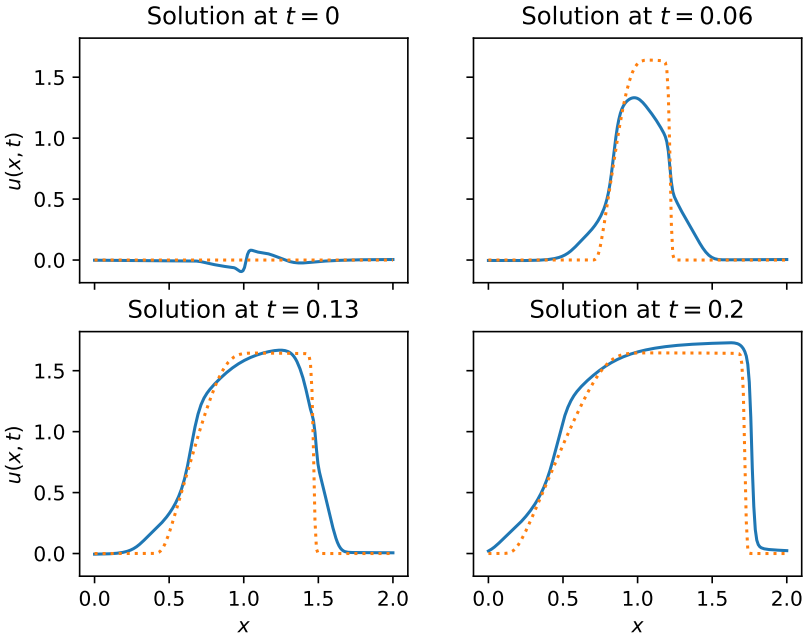

Figure 3.26: The prediction of u after after training a PINN 5000 times at 21500 points with data fixed points.

3.4 Conclusion

At this point we want to draw some conclusions. Recall if we trained the PINNs to predicts continuously differentiable solutions of initial value problems for conservation laws the predicted solutions were quite accurate, as seen by predicting the solutions of the first IVP for the linear advection equation and the first IVP with $T = 0.5$ for Burgers' equation. One could increase the accuracy of the predictions by training them more times at more points or one could choose more layers and nodes and different activation functions to increase the accuracy even more. However we saw that for these two initial value problems the finite-volume method computed a perfectly accurate solution in very little time. So it is questionable if it is useful to predict the solutions by training PINNs instead of using the finite-volume method.

By predicting the solution of the second IVP for the linear advection equation we found that even after training a PINN with five hidden layers, the second spike was not perfectly predicted. The accuracy could possibly be increased even more by training more times and/or using different amounts of layers and nodes and different activation functions and optimizers. We also saw that the finite-volume method took quite long to compute a suitable solution. Maybe in these special cases, in which the commonly used numerical methods have problems, the predictions of solutions by training PINNs could deliver some promising results in less time. However in our tests the finite-volume method computed a more accurate solution faster than the neural network.

By training PINNs to predict discontinuous solutions of IVPs for conservation laws, i. e. the initial value problems for Burgers' equation, we saw that the predictions were not very accurate. We now want to try to explain the problem of PINNs with discontinuous solutions. There are two aspects. Firstly we used the hyperbolic tangent as our activation function for the neural network. Hence our network is a smooth function and therefore it is quite complicated for the network to approximate discontinuous functions properly. Secondly by including the conservation law in differential form into the loss, we assumed that the network is continuously differentiable. If the network would predict a discontinuous solution perfectly, we get a contradiction, because that would mean that the loss is zero and the network satisfies the conservation law. This would require the network to be continuously differentiable. It is not impossible that PINNs can approximate discontinuous solutions quite well, but it may take a huge amount of layers and nodes or different activation functions and optimizers and probably much more training, which may result in high training time and even high memory usage. As we saw in our test even after training a PINN with seven hidden layers many times at many points, the predicted solutions were still quite inaccurate. In comparison the finite volume method computed very accurate solutions in little time.

At last we tried predicting the solution of an IVP for the shallow-water equations by training PINNs. Here we encountered the same problems with discontinuous solutions like before, since even after training many times, the predictions were still not very accurate. Also we found out that the way we choose the points at which the network is trained can have an impact on the accuracy. The reason for the different accuracy after training at fixed and randomly chosen points could be a result of the initial data. Since it is constant left and right of a discontinuity we do not always have the same

amount of points on the left or right side of the discontinuity and so the value of the loss L_{ID} can vary quite strongly with each iterations.

At this point we need to note, there is very much room for optimizing the predictions of the neural networks. And by finding the right parameters and values one might be able to get accurate predictions quicker, but it would require immense testing to get the optimal values and parameters for which the PINN gives a suitable approximation after training acceptable times. One advantage of predicting solutions with PINNs is that we get predictions for all points in the domain. The finite-volume or finite-difference method only computes the solution at certain mesh points.

To conclude if the IVP has a continuously differentiable solution, the PINN gives a quite accurate prediction of the solution, sometimes even after very few training iterations. But the predictions of discontinuous solutions are not very accurate. This is a crucial issue, since discontinuities occur often in the solutions of IVPs for conservation laws. The finite-volume method has no problems with discontinuous solutions and generally computes very accurate solutions in little time. So to sum up until the problem with discontinuous solution is not solved, the finite-volume method is at the moment still the better choice for solving initial value problems for conservation laws, since it is generally quite fast and more importantly computed more accurately.

3.5 Outlook

In this subsection we want to present some aspects and ideas, which were not covered in the text so far or which might be interesting to investigate in the future. We want to show that with PINNs we can predict the solution of IVPs outside of the domain in which we trained the network. Also in this text we only considered one-dimensional conservation laws, so we want to present the predictions of the initial value problem for the two-dimensional advection equation by training a PINN. We want to show that with PINNs we can predict the solution of IVPs, which are defined for all points in $x \in \mathbb{R} \times [0, \infty[$. At last we want to reference some improvements one cloud make to increase the accuracy of the predictions for discontinuous solutions.

3.5.1 Predictions Outside of the Training domain

In this text we only considered initial value problems that are defined over some bounded domain. The reason for that was that the numerical method needs such a bounded domain. However, we can train a neural network on some bounded domain and it will give predictions of the solution for all points in $x \in \mathbb{R} \times [0, \infty[$. Here we want to train a PINN with 5 hidden layers to solve the first IVP for the linear advection equation, from the section before. Each of the five hidden layers contain 50 nodes and a hyperbolic tangent activation function. We will train the network 5000 times with no boundary conditions at 10500 points in the domain $[-2, 2] \times [0, 1]$. From the 10500 points are 10000 collocation points and 500 are at the line $t = 0$. The prediction of the network at $t = 0$, $t = 1$, $t = 5$ and $t = 10$ in comparison with the analytical solution can be seen in figure 3.27. We can see the predictions are quite accurate and even at $t = 5$ and $t = 10$. This could be an important advantage of PINNs in comparison to the finite-volume method, since even if we only have data in some small domain available it allows us to get predictions quite accurate outside of that domain.

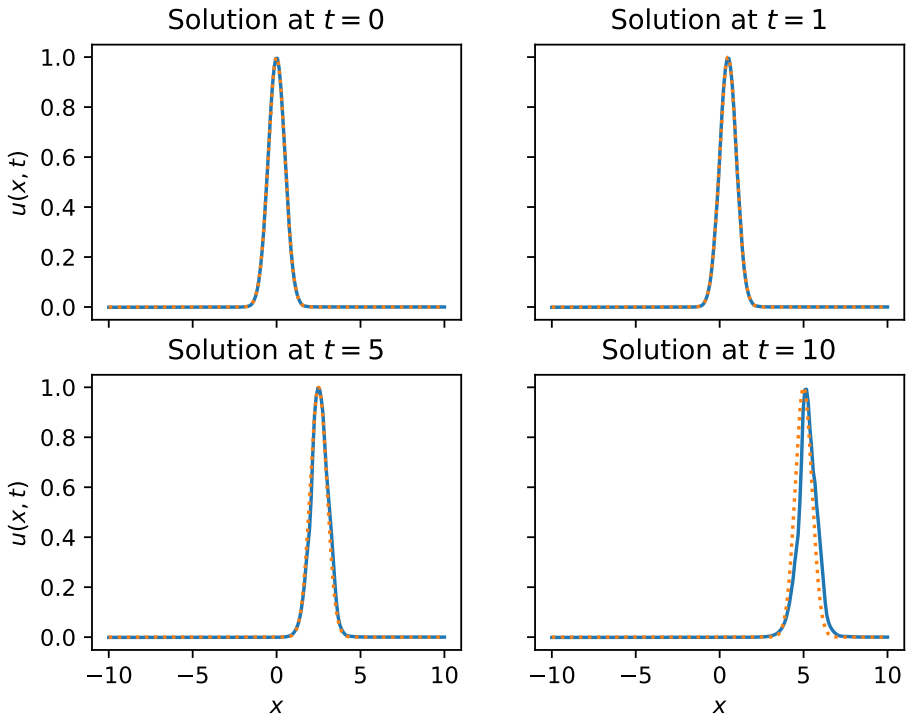

Figure 3.27: The prediction of the solution in comparison with the analytical solution.

3.5.2 Two-Dimensional Advection Equation

We want to predict the solutions of an initial value problem for the two-dimensional linear advection equation

$$u_t(x, y, t) + 0.5u_x(x, y, t) + 0.5u_y(x, y, t) = 0. \tag{3.4}$$

The IVP we want to solve is defined as the following: We want to find a function $u : [-2, 2] \times [-2, 2] \times [0, 1] \to \Omega$ that satisfies (3.4) for all $x \in]-2, 2[$, $y \in]-2, 2[$ and $t \in]0, 1[$ and

$$u(x, y, 0) = \exp(-5(x^2 + y^2))$$

for all $x \in]-2, 2[$ and $y \in]-2, 2[$. Also u needs to satisfy

$$u(-2, -2, t) = u(-2, 2, t) = u(2, -2, t) = u(2, 2, t) = 0$$

for all $t \in [0, 1]$. We want to predict the solution by training by training a PINN with 3 hidden layers. Each of the three hidden layers contains 20 nodes and a hyperbolic tangent activation function. We will use the Adam optimizer. The loss functions for the two-dimensional case are similar to the one-dimensional case, we just need to keep in mind that we have four instead of two boundary conditions. The prediction after training the PINN 400 times at 750 points, i. e. 500 collocation points, 50 points at $t = 0$ and 50 points each at the four boundaries, in comparison with the analytical solution at different times can be seen in figure 3.28. We can see that the prediction by the PINN is quite accurate. The reason for that is probably that the solution is smooth.

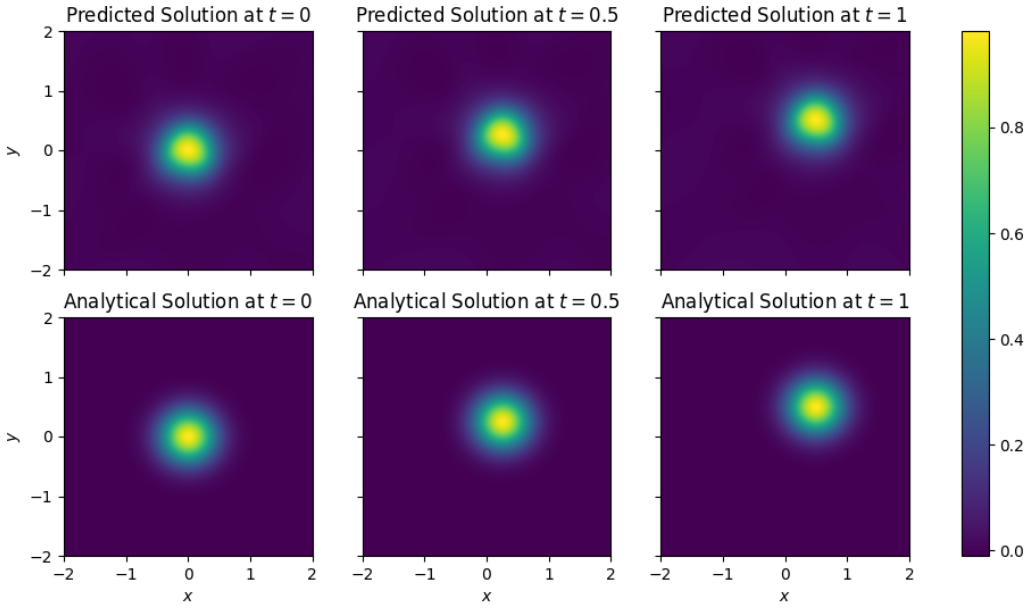

Figure 3.28: The prediction of the solution in comparison with the analytical solution.

3.5.3 Improvements for Higher Accuracy

The main issue we encountered by using PINNs to predict the solution of IVPs for hyperbolic one-dimensional systems of conservation laws is the fact that they struggle with discontinuities. As said before, one could try different activation functions that are not smooth to predict the discontinuities better. Another option is to modify the loss function and not include the conservation law in the loss function. In [15] the author presents a new approach to predict the entropy solution scalar conservation laws, the so-called *weak PINNs*. Instead of minimizing the conservation laws the weak PINNs minimize an optimization problem which is connected to some entropy conditions. Another approach is the so-called *deep finite-volume method*, which is presented in [4]. The author states it is designed according to the weak form of the partial differential equation and so may achieve better accuracy than PINNs when the solution is insufficiently smooth. Approximating the discontinuous solutions of IVPs for systems of conservation laws is an interesting topic for future research.

4 A Theoretical Result: Smooth Approximation in Sobolev Spaces

In this section we provide the details of an important theoretical result on Sobolev spaces which the lecture could not cover. We say that a subset U of some (function) space $(V, ||\cdot||)$ is *dense* in said space, if any element in V can be approximated arbitrarily well by elements from U. Formally, this means that for every $v \in V$ there exists a sequence $(u_k)_{k\in\mathbb{N}}$ in U such that $||u_k - v|| \to 0$ as $k \to \infty$. In our case, U will correspond to functions in C^∞ or C_c^∞ and V will be some type of L^p or Sobolev space

H^m with the corresponding norm.

The way in which approximation results tend to show up in in the context of Sobolev spaces is roughly as follows: First, one proves a claim (e. g. an inequality) for all "nice" functions which are dense in the space, then one establishes that all operators that appear are continuous which, in turn, implies the same result for the entire function space via density, upon passing to the limit.

This section is divided into three subsections. We begin in subsection 4.1 by introducing the idea of the convolution of two functions. The result is a new function which can be thought of as their average. By averaging arbitrary functions with special types of smooth functions (so-called mollifiers) we can obtain smooth approximations to the original functions. This will be explained in subsection 4.2. Finally, a similar approach generalizes to weak derivatives and thus to Sobolev spaces. Subsection 4.3 derives the Meyers-Serrin theorem which is the most famous density result for Sobolev spaces.

4.1 The Convolution

In this subsection we present the fundamental tool that is used to find smooth approximations to complicated functions. The idea is to average the initial function f with a special kind of weight function w which is so smooth that it forces the weighted average (later denoted $w * f$) to be smooth itself. By a clever choice of weights one can then generate a sequence of smooth functions which converge to the original function. For example, in applications f might be a distorted signal and the averaging procedure that we are about to describe would then remove some of the noise.

Let us begin by finding an appropriate notion of averaging two functions together. Say we are given some (e. g. continuous) map $f : \mathbb{R} \to \mathbb{R}$. Let us pick a concrete value $x \in \mathbb{R}$ at which we want to average f. Doing so for all x later on will yield a new function which can be seen as an averaged version of f. A standard result of real analysis is that $\frac{1}{b-a}\int_a^b f(y)\,\mathrm{d}y$ measures the average value of f over the interval $[a, b] \subseteq \mathbb{R}$. Indeed, the fundamental theorem of calculus tells us that $F'(x) = f(x)$ for $F(x) := \int_a^x f(y)\,\mathrm{d}y$ which implies that for sufficiently small $h > 0$ we should have

$$f(x) = F'(x) \approx \frac{F(x+h) - F(x-h)}{2h} = \int_{x-h}^{x+h} \frac{1}{2h} f(y)\,\mathrm{d}y,$$

employing a central difference approximation to the derivative. So a suitable average could be found by picking a fixed $h > 0$ which is (i) as small as needed to still be close enough to the original function's value and (ii) as large as needed to yield a proper averaging. Let us assume for the sake of discussion that $h := 1$ is a suitable choice for our function f. Then the averaged value of $f(x)$ is given by a new function

$$f_{\text{average}}(x) := \int_{x-1}^{x+1} \frac{1}{2} f(y)\,\mathrm{d}y. \tag{4.1}$$

We can rewrite f_{average} somewhat by introducing a weight

$$w_{\text{uniform}}(z) := \begin{cases} \frac{1}{2}, & \text{for } z \in [-1, 1] \\ 0, & \text{for } z \in \mathbb{R}\backslash[-1, 1]. \end{cases}$$

With it (4.1) can be recast into an integral over the entire domain:

$$f_{\text{average}}(x) = \int_{\mathbb{R}} w_{\text{uniform}}(y-x) f(y)\,\mathrm{d}y. \tag{4.2}$$

Notice that the weight function w_{uniform} is chosen such that all function values in an interval of length 2 around the given point x are equally "important" to the averaging process. This is comparable to the arithmetic mean. Just like the arithmetic mean can be generalized to a weighted arithmetic mean, so, too, can the weight function w_{uniform} be generalized to some other kinds of weights $w : \mathbb{R} \to \mathbb{R}$. For example, the weight

$$w_{\text{quadratic}}(z) := \begin{cases} \frac{3}{4}(1-z^2), & \text{for } z \in [-1,1] \\ 0, & \text{for } z \in \mathbb{R}\backslash[-1,1] \end{cases}$$

prioritizes values closer to x. Notice, however, that any true averaging function, often called a *kernel function* or *filter function* in applications, should always satisfy $\int_{\mathbb{R}} w(z)\,\mathrm{d}z = 1$ and $w(z) \geqslant 0$ for all $z \in \mathbb{R}$.

Obviously, (4.2) can also be worked out for every point and with much more general functions. This leads (more or less) to the following definition.

Definition 4.1 (Convolution)
Given two functions $f, g : \mathbb{R}^d \to \mathbb{R}$. We define a new function $f * g : \mathbb{R}^d \to \mathbb{R}$ called the convolution of f and g *via*

$$(f * g)(x) := \int_{\mathbb{R}^d} f(x-y)g(y)\,\mathrm{d}y, \quad x \in \mathbb{R}^d.$$

Whenever f and g are such that the integral exists for almost all $x \in \mathbb{R}^d$, we say that f and g are convolvable. *If any of the functions is only defined on a subset $\Omega \subseteq \mathbb{R}^d$, then we can extend it by zero outside Ω and still compute the integral above.*

We follow the typical conventions here where the input of f is "the wrong way around", i. e., it uses $x - y$ instead of $y - x$ as in the derivation. The reader should not worry about this too much. It is mostly just a convention.

We note several important results related to convolutions of L^p with L^1 functions. It is worth keeping in mind, however, that in our applications the L^1 function will actually be a C_c^∞ function. We recall the notation $A + B := \{a + b \mid a \in A, b \in B\}$ for two sets A and B. In addition, we will say that a subset $\Omega \subseteq \mathbb{R}^d$ is a *domain*, if it is open and connected. The connectedness assumption is usually not needed for the results that follow. But domains are the typical setting that one needs for the study of partial differential equations. The *support* of a continous function $f : \Omega \to \mathbb{R}$ is defined as $\operatorname{supp} f := \overline{\{x \in \Omega \mid f(x) \neq 0\}}$ where the closure is taken with respect to the Euclidean norm $|\cdot|$ in $\mathbb{R}^d$. This means that $\operatorname{supp} f \subseteq \overline{\Omega}$, but not $\operatorname{supp} f \subseteq \Omega$ in general. If f was an equivalence class of functions in the L^p sense, then a generalized version of the support (the *essential support*) could be considered instead, cf. [12, Section 1.5].

Lemma 4.2 (Properties of the convolution)
Let $\Omega \subseteq \mathbb{R}^d$ be a domain, $f \in L^p(\Omega)$ for some $1 \leqslant p < \infty$ and $g \in L^1(\Omega)$. Then the following properties hold:

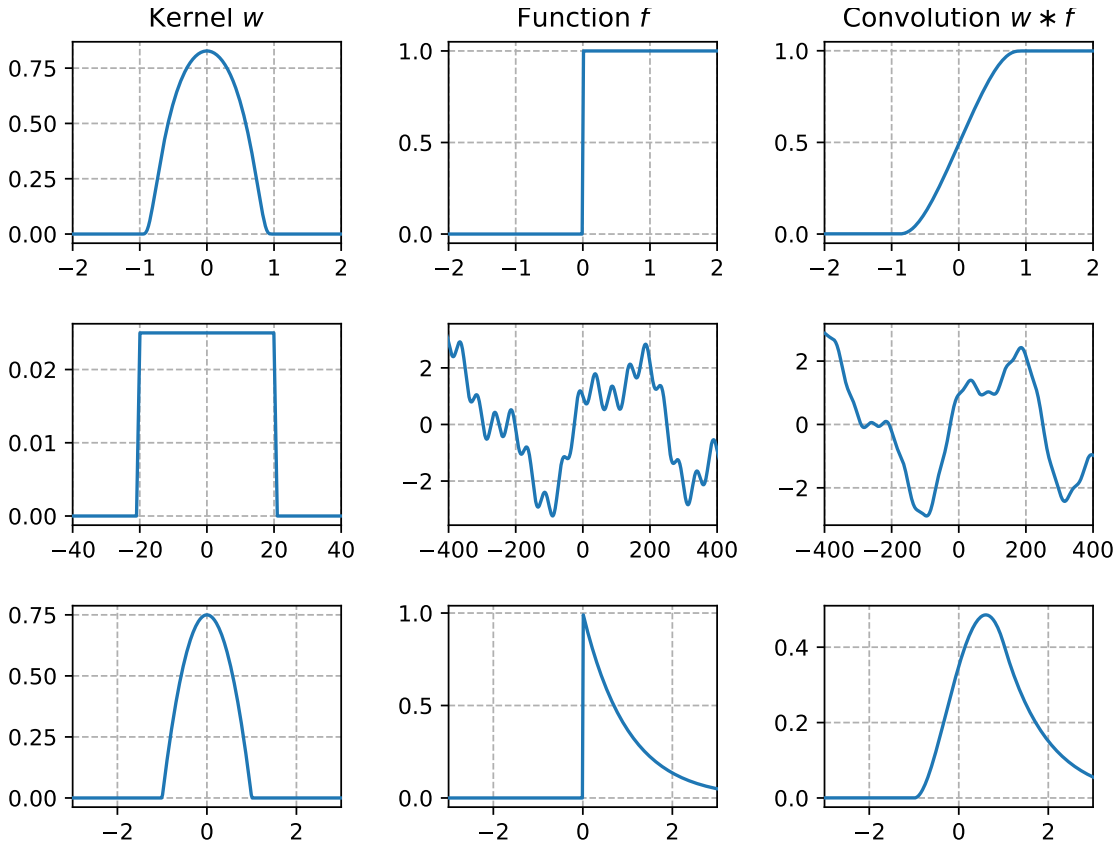

Figure 4.1: Examples of convolutions $w * f$ (third column) of functions f (second column) with kernels w (first column).

(a) $f * g \in L^p(\Omega)$ *with* $||f * g||_{L^p(\Omega)} \leqslant ||f||_{L^p(\Omega)} ||g||_{L^1(\Omega)}$. *In particular,* f *and* g *are convolvable.*

(b) $f * g = g * f$.

(c) $\operatorname{supp} f * g \subseteq \overline{\operatorname{supp} f + \operatorname{supp} g}$.

(d) If $\phi \in C_c^m(\Omega)$, *then* $\phi * f \in C^m(\Omega)$ *and for every multi-index* $\alpha \in \mathbb{N}_0^d$ *with* $|\alpha| \leqslant m$ *we have*

$$D^\alpha(\phi * f) = f * D^\alpha \phi.$$

Proof sketch: We only give a sketch of the proof and refer to section X.7 of the book [2] by Amann and Escher for details. First, one must show that the convolution integral is well-defined for L^p functions. This comes down to proving that the value of the integral does not depend on the concrete representative of the equivalence class.

The proof of (a) is by Hölder's inequality and Fubini's theorem. It also relies on the translational invariance of the Lebesgue measure.

The proof of (b) is due to the change of variables formula for integrals (German: Transformationssatz).

The proof of (c) goes as follows. Choose concrete representatives $f \in \mathcal{L}^p(\Omega)$ and $g \in \mathcal{L}^1(\Omega)$ from the respective equivalence classes. We may assume $f * g \neq 0$ without loss of generality. For any $x \in \{x \in \mathbb{R}^d \mid (f * g)(x) \neq 0\}$ there exists a $y \in \mathbb{R}^d$ such that

$f(x-y)g(y) \neq 0$. Hence $y \in \operatorname{supp} g$ and $x \in y + \operatorname{supp} f$, so $x \in \operatorname{supp} f + \operatorname{supp} g$. This yields

$$\{x \in \mathbb{R}^d \mid (f * g)(x) \neq 0\} \subseteq \operatorname{supp} f + \operatorname{supp} g$$

and taking the closure over both sides gives the desired inclusion.

We will prove (d) only for the first partial derivative; the more general case then follows by induction. Let $e_1, \dots, e_d \in \mathbb{R}^d$ denote the standard unit vectors and fix $i \in \{1, \dots, d\}$. Then we have

$$\frac{(\phi * f)(x + he_i) - (\phi * f)(x)}{h} = \int_\Omega \frac{\phi(x + he_i - y) - \phi(x - y)}{h} f(y) \, \mathrm{d}y$$

for all $h \neq 0$ where $|h|$ is sufficiently small such that the convolution is still defined at $x + he_i$. So if the difference quotient converges uniformly to $\frac{\partial \phi}{\partial x_i}(x - y)$, we could exchange limits and integration to obtain the desired assertion. Let us see why this is indeed the case. A straightforward argument can be used to show that $\frac{\partial \phi}{\partial x_i}$ is even uniformly continuous on $\mathbb{R}^d$ because of its compact support (continuous functions are uniformly continuous on compact sets). Hence, for every $\varepsilon > 0$ there exists a $\delta > 0$ such that

$$|x - y| < \delta \implies \left| \frac{\partial \phi}{\partial x_i}(x) - \frac{\partial \phi}{\partial x_i}(y) \right| < \varepsilon$$

for all $x, y \in \mathbb{R}^d$. So if we choose $y := x + the_i$ with $t \in [0, 1]$ and $|h| < \delta$, then $|x - y| = t|h| < \delta$, so

$$\begin{aligned} \left| \frac{\partial \phi}{\partial x_i}(x) - \frac{\phi(x + he_i) - \phi(x)}{h} \right| &= \left| \frac{\partial \phi}{\partial x_i}(x) - \int_0^1 \frac{\partial \phi}{\partial x_i}(x + the_i) \, \mathrm{d}t \right| \\ &\leqslant \int_0^1 \left| \frac{\partial \phi}{\partial x_i}(x) - \frac{\partial \phi}{\partial x_i}(x + the_i) \right| \mathrm{d}t \\ &\leqslant \varepsilon \end{aligned}$$

by the mean value theorem. This establishes the uniform convergence by taking the supremum of this inequality over all x. ■

In the previous theorem it is essential that f and g are both defined on Ω and not on a bigger set. For example, notice that (the restriction of) a function from $\mathbb{R}^d$ to $\mathbb{R}$ could lie in $L^p(\Omega)$ even if it does not vanish outside Ω. And, indeed, the theorem is not true in this case, see [12, Theorem 2.16], because the "extension by zero outside of Ω" argument no longer works.

4.2 Mollification

This subsection is dedicated to the approximation of "bad" functions with "good" functions. More concretely, we wish to approximate arbitrary L^p functions with C_c^∞ functions. The approximation process will actually be constructive. Its basic idea is to take a general function from L^p and to "average it" with a special kind of smooth function from C_c^∞. The result is then C^∞ and can be used to approximate the initial function. Averaging two functions will be done using the convolution as discussed in the previous subsection.

We begin by introducing the functions which, upon being convolved with the original function, will yield the smooth approximations. Such functions are called *mollifiers* ("to mollify" means "besänftigen" in German). Our discussion begins with the so-called *standard mollifier*

$$J(x) := \begin{Bmatrix} C_d \exp\left(\frac{1}{|x|^2-1}\right), & \text{if } |x| < 1 \\ 0, & \text{if } |x| \geqslant 1 \end{Bmatrix}, \quad x \in \mathbb{R}^d. \tag{4.3}$$

Here the constant $C_d > 0$ depends on the dimension of the space d and is choosen such that $\int_{\mathbb{R}^d} J(x)\,\mathrm{d}x = 1$. Notice that J defines a kernel function in the sense of the

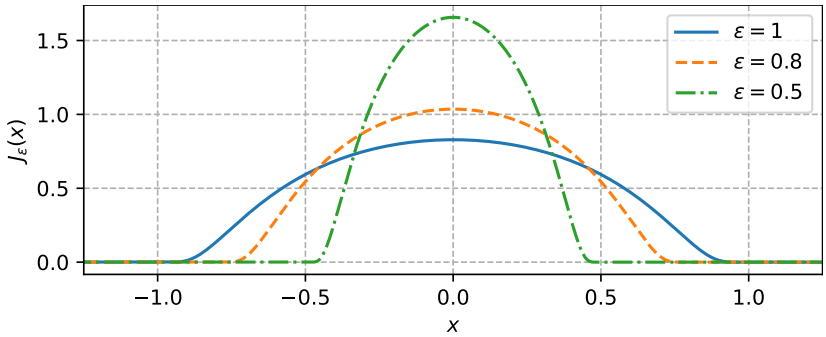

Figure 4.2: Some mollifiers (4.4) in $\mathbb{R}^1$.

previous subsection. For each $\varepsilon > 0$ we also study the rescaled versions

$$J_\varepsilon(x) := \frac{1}{\varepsilon^d} J\left(\frac{x}{\varepsilon}\right), \quad x \in \mathbb{R}^d, \tag{4.4}$$

see figure 4.2. These functions are our mollifiers. They have the following properties.

Lemma 4.3 (Properties of mollifiers)
For $\varepsilon > 0$ the mollifier $J_\varepsilon : \mathbb{R}^d \to \mathbb{R}$ from (4.4) has the following properties:

(a) $J_\varepsilon \in C_c^\infty(\mathbb{R}^d)$;

(b) $\operatorname{supp} J_\varepsilon = \overline{B}_\varepsilon(0) = \left\{x \in \mathbb{R}^d \;\middle|\; |x| \leqslant \varepsilon\right\}$;

(c) $\int_{\mathbb{R}^d} J_\varepsilon(x)\,\mathrm{d}x = 1$;

(d) $J_\varepsilon(x) \geqslant 0$ *for all* $x \in \mathbb{R}^d$.

The proof of everything except the fact that $J_\varepsilon \in C^\infty(\mathbb{R}^d)$ is essentially an immediate consequence of the construction of J_ε. The differentiability requires a somewhat involved analysis argument that we do not want to give here. Essentially, one proceeds by induction and shows that the derivative of the function $t \mapsto \exp(-1/t)$ at $t = 0$ can be expressed as the product of two functions with certain properties. This can then be used to prove the existence of the differential quotient by some standard limit estimates.

We now describe how mollifiers are used to "smooth out" functions. To this end, let $\Omega \subseteq \mathbb{R}^d$ be a domain. We will write $K \subset\subset \Omega$ and say that K is *compactly contained*

in Ω when $\overline{K}$ is compact and still contained in Ω, i. e. $\overline{K} \subseteq \Omega$. With this notation, let us define the sets

$$L^p_{\mathrm{loc}}(\Omega) := \left\{ u \in L^p(\Omega) \;\middle|\; u|_K \in L^p(K) \text{ for any } K \subset\subset \Omega \right\}, \quad 1 \leqslant p < \infty$$

to denote the so-called *locally p-integrable functions*. To calculate the values of the restriction $u|_K$, we can use the characteristic function of the set χ_K which outputs 1 when $x \in K$ and 0 else and write $\chi_K u$. We note that $L^p_{\mathrm{loc}}(\Omega) \subseteq L^1_{\mathrm{loc}}(\Omega)$ by a well-known result about L^p spaces (see [1, Theorem 2.14]) and hence $L^p(\Omega) \subseteq L^p_{\mathrm{loc}}(\Omega) \subseteq L^1_{\mathrm{loc}}(\Omega)$. For any function $u \in L^1_{\mathrm{loc}}(\Omega)$ we define its *mollification* by

$$u_\varepsilon := J_\varepsilon * u. \tag{4.5}$$

Notice that since u is defined on Ω and J_ε has support in $\overline{B}_\varepsilon(0)$ (by lemma 4.3 (b)), this operator is well-defined at all points in the subset

$$\Omega_\varepsilon := \{x \in \Omega \mid \operatorname{dist}(x, \partial\Omega) > \varepsilon\}. \tag{4.6}$$

Indeed, we have

$$u_\varepsilon(x) = \int_{B_\varepsilon(x)} J_\varepsilon(x-y) f(y)\,\mathrm{d}y = \int_{B_\varepsilon(0)} J_\varepsilon(z) f(x-z)\,\mathrm{d}z.$$

Since the *distance function* $x \mapsto \operatorname{dist}(x, \partial\Omega) := \inf\left\{|x-y| \;\middle|\; y \in \partial\Omega\right\}$ is continuous

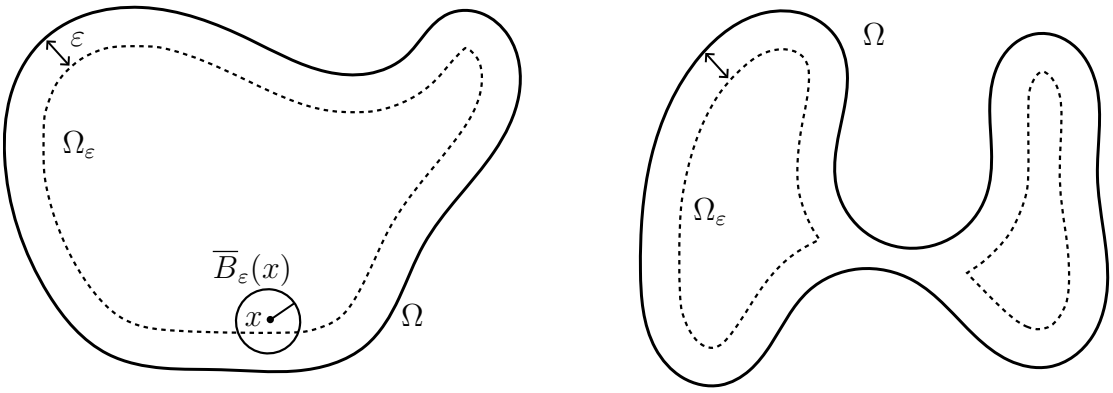

Figure 4.3: The set $\Omega_\varepsilon \subseteq \Omega$ in which every point has distance of at least ε to the boundary $\partial\Omega$. The figure on the right shows that while Ω_ε is always an open set, it may no longer be connected.

(clear?), the set Ω_ε is open (as the preimage of an open set); although it may no longer be a domain, cf. figure 4.3.

Perhaps the effect of a mollification is best illustrated by an example.

Example 4.4 Let us view the Heaviside step function

$$u(x) = \begin{Bmatrix} 1, & \text{if } x > 0 \\ 0, & \text{if } x \leqslant 0 \end{Bmatrix}, \quad x \in \mathbb{R}.$$

Its mollification is

$$u_\varepsilon(x) = (J_\varepsilon * u)(x) = \int_{-\varepsilon}^{x} J_\varepsilon(y)\,\mathrm{d}y.$$

We see immediately that $u_\varepsilon(x) = 0$ for $x \leqslant -\varepsilon$ and $u_\varepsilon(x) = 1$ for $x \geqslant \varepsilon$. In $[-\varepsilon, \varepsilon]$ the function must grow monotonously. The case $\varepsilon := 1$ can be seen in the first row of figure 4.1. ◆

The mollification of a function has many wonderful properties. The following result is theorem A.16 in the book [3] by BRESSAN.

Theorem 4.5 (Properties of mollification)
Let $\Omega \subseteq \mathbb{R}^d$ be a domain. Then the following hold:

(a) If $u \in L^1_{\text{loc}}(\Omega)$, then for every $\varepsilon > 0$ one has $u_\varepsilon \in C^\infty(\Omega_\varepsilon)$.

(b) If $u \in L^1_{\text{loc}}(\Omega)$, then, as $\varepsilon \downarrow 0$, $u_\varepsilon(x) \to u(x)$ pointwise for almost all $x \in \Omega$.

(c) If $u \in C(\Omega)$, then, as $\varepsilon \downarrow 0$, $u_\varepsilon \to u$ uniformly on compact subsets of Ω (i. e. $||u_\varepsilon - u||_{C(K)} \to 0$ for all $K \subset\subset \Omega$).

(d) If $1 \leqslant p < \infty$ and $u \in L^p_{\text{loc}}(\Omega)$, then, as $\varepsilon \downarrow 0$, $u_\varepsilon \to u$ in $L^p_{\text{loc}}(\Omega)$ (i. e. $||u_\varepsilon - u||_{L^p(K)} \to 0$ for all $K \subset\subset \Omega$).

Proof: (a) This is an immediate consequence of lemma 4.2 (d).

(b) We do not prove (b) because it will not play a role in the following discussion. We refer the interested reader to BRESSAN's book.

(c) Let $u \in C(\Omega)$ and pick sets $K \subset\subset U \subset\subset \Omega$. The continuous function u is even uniformly continuous on the compact set U by a well-known result from analysis. So for every $\varepsilon > 0$ there exists a $\delta > 0$ such that

$$|x - y| \leqslant \delta \implies |u(x) - u(y)| \leqslant \varepsilon$$

for $x, y \in U$. Now make δ small enough such that the support of the mollification $u_\delta : \Omega_\delta \to \mathbb{R}$ lies inside of U, e. g. $\delta < \operatorname{dist}(K, \partial U)$. Then

$$|u_\delta(x) - u(x)| = \left| \int_{B_\delta(x)} J_\delta(x - y)\big(u(y) - u(x)\big)\,\mathrm{d}y \right| \leqslant \varepsilon \int_{B_\delta(x)} J(x - y)\,\mathrm{d}y \leqslant \varepsilon$$

for all $x \in K$ by lemma 4.3 (c). Taking the supremum of this inequality over all $x \in K$ proves the uniform convergence.

(d) Let $u \in L^p_{\text{loc}}(\Omega)$ and pick $K \subset\subset U \subset\subset \Omega$. Note that $u_\varepsilon \in L^p(K)$ by lemma 4.2 (a) with

$$||u_\varepsilon||_{L^p(K)} \leqslant ||u||_{L^p(K)}\, ||J_\varepsilon||_{L^1(K)}.$$

Also, by lemma 4.3 (c), J_ε is normalized, so $||J_\varepsilon||_{L^1(K)} \leqslant 1$. Since $K \subseteq U$ we have $||u||_{L^p(K)} \leqslant ||u||_{L^p(U)}$. All in all this yields the estimate

$$||u_\varepsilon||_{L^p(K)} \leqslant ||u||_{L^p(U)}. \tag{4.7}$$

Now fix a $\delta > 0$. Since continuous functions $C(U)$ are dense in $L^p(U)$ (cf. theorem 2.19 in [1]), we can find $g \in C(U)$ with $||u - g||_{L^p(U)} < \delta$. Then

$$\begin{aligned} ||u_\varepsilon - u||_{L^p(K)} &\leqslant ||u_\varepsilon - g_\varepsilon||_{L^p(K)} + ||g_\varepsilon - g||_{L^p(K)} + ||g - u||_{L^p(K)} \\ &\overset{(4.7)}{\leqslant} ||u - g||_{L^p(U)} + ||g_\varepsilon - g||_{L^p(K)} + ||g - u||_{L^p(U)} \\ &\leqslant \delta + ||g_\varepsilon - g||_{L^p(K)} + \delta. \end{aligned}$$

Since g is continuous $g_\varepsilon \to g$ uniformly on the compact set K by (c). Hence, $\limsup_{\varepsilon \downarrow 0} ||u_\varepsilon - u||_{L^p(K)} \leqslant 2\delta$ and since $\delta > 0$ was arbitrary, this proves $||u_\varepsilon - u||_{L^p(K)} \to 0$ for $\varepsilon \downarrow 0$. ■

We can see from (a) that mollification has the desired smoothing property. The results (b)–(d) give insight into the convergence of u_ε to u (in different norms). Indeed, as we explain next, these findings are not that surprising.

Remark 4.6 (Distributions)
The intuition behind why u_ε should converge to u at least pointwise is actually related to the motivation for the convolution from the previous subsection: We can think of the mollification as an averaging with a smoothed out step function. As the size of the step shrinks, the average becomes closer and closer to the actual function value. At the same time the height of the step must grow larger and larger to keep the area of the weight at unity. In the limit, then, we would expect a weight function like

$$\delta(x) := \begin{cases} \infty, & \text{if } x = 0 \\ 0, & \text{if } x \neq 0. \end{cases}$$

This obviously isn't really the case (since $\int_{\mathbb{R}^d} \delta(x - y)u(y)\,\mathrm{d}y = 0$), but it is still a nice visual to keep in mind. The "function" δ is called the δ *distribution*. It is the entry point to the theory of distributions. We refer the interested reader to the book [2, Section X.7] for an introductory overview. ◆

On the basis of the results from theorem 4.5 we are able to prove the following density theorem for L^p spaces.

Theorem 4.7 (C_c^∞ is dense in L^p)
Let $\Omega \subseteq \mathbb{R}^d$ be a domain. Then $C_c^\infty(\Omega)$ is dense in $L^p(\Omega)$ for every $1 \leqslant p < \infty$, i. e., for every $u \in L^p(\Omega)$ there exists a sequence $(u_k)_{k\in\mathbb{N}}$ in $C_c^\infty(\Omega)$ such that

$$||u_k - u||_{L^p(\Omega)} \to 0 \quad \text{as} \quad k \to \infty.$$

Proof: Given $u \in L^p(\Omega)$ and $\delta > 0$. A simple application of the dominated convergence theorem yields a (possibly very large) compact set $K \subseteq \Omega$ such that

$$||u||_{L^p(\Omega\setminus K)} \leqslant \tfrac{\delta}{2}.$$

Now if we choose $\varepsilon < \operatorname{dist}(K, \partial\Omega)$, then the convolution $v_\varepsilon := J_\varepsilon * (u\chi_K)$ is well-defined. By theorem 4.5 (d) we can make ε small enough such that

$$||u\chi_K - v_\varepsilon||_{L^p(\mathbb{R}^d)} = ||u - v_\varepsilon||_{L^p(K)} \leqslant \tfrac{\delta}{2}.$$

Therefore

$$\begin{aligned} ||u - v_\varepsilon||_{L^p(\Omega)} &= ||u\chi_K + u\chi_{\Omega\setminus K} - v_\varepsilon||_{L^p(\Omega)} \\ &\leqslant ||u\chi_K - v_\varepsilon||_{L^p(\Omega)} + ||u\chi_{\Omega\setminus K}||_{L^p(\Omega)} \\ &= ||u\chi_K - v_\varepsilon||_{L^p(\mathbb{R}^d)} + ||u||_{L^p(\Omega\setminus K)} \\ &\leqslant \delta. \end{aligned}$$

Hence, setting $u_k := v_{1/k}$ for $k \in \mathbb{N}$ yields the desired approximating sequence. ■

We note that the previous result was used for a crucial theorem on Sobolev spaces in the course (namely what is often called the *fundamental lemma of the calculus of variations* in chapter II, section I) which was needed to establish the uniqueness of the weak derivative.

We end our discussion on mollifiers with a remark.

Remark 4.8 (Friedrichs mollifiers)
At the beginning of this subsection we defined the standard mollifier (4.3). The resulting family of mollifiers are sometimes called Friedrichs mollifiers. They are by far the most popular family of mollifiers. However, in principle it is possible to start with a different function J that has all the same properties as our standard mollifier. The definition (4.4) still works and produces another family of mollifiers that also obey lemma 4.3. For example, one could adapt the construction from example 4.4 to create smoothed versions of step functions. It is worth pointing out that the the ideas from remark 4.6 do not rely on the special shape of the mollifiers. Each of them converges to the δ distribution pointwise. ◆

4.3 The Meyers-Serrin Theorem

In this subsection we will prove the Sobolev space analog of the previous subsection's approximation theorem. It seems reasonable to hope that a similar mollifier ansatz might work here, too, because Sobolev spaces are closely related to L^p spaces. And this is indeed so. To this end, recall that any Sobolev function $u \in H^m(\Omega)$ lies in $L^2(\Omega)$, so we can simply define u_ε as in (4.5) on Ω_ε as in (4.6).

We begin with the following approximation result which is akin to to theorem 4.5 (d). It shows that every Sobolev function can locally be approximated by a smooth Sobolev function.

Lemma 4.9 (Mollification in Sobolev spaces) [1, Lemma 3.16]
Let $\Omega \subseteq \mathbb{R}^d$ be a domain and $U \subset\subset \Omega$ a subdomain. Then for any $m \in \mathbb{N}$ and $u \in H^m(\Omega)$ we have

$$u_\varepsilon := (J_\varepsilon * u) \in H^m(U) \cap C^\infty(U)$$

for all sufficiently small $\varepsilon > 0$ and, as $\varepsilon \downarrow 0$, $u_\varepsilon \to u$ in $H^m(U)$, i. e. $||u_\varepsilon - u||_{H^m(U)} \to 0$.

Proof: Let $\alpha \in \mathbb{N}_0^d$ be a multi-index with $|\alpha| \leqslant m$. We show that

$$D^\alpha u_\varepsilon = J_\varepsilon * (D^\alpha u) \quad \text{in } U \tag{4.8}$$

where $\varepsilon > 0$ is chosen sufficiently small such that the convolution is well-defined, e. g. $\varepsilon < \operatorname{dist}(U, \partial\Omega)$. Hence, $U \subseteq \Omega_\varepsilon$ for such ε and $u_\varepsilon \in C^\infty(U)$ follows immediately by

theorem 4.5 (a). Once (4.8) is established, we can use theorem 4.5 (d) to conclude that $||D^\alpha u_\varepsilon - D^\alpha u||_{L^2(U)} \to 0$ as $\varepsilon \downarrow 0$ which, in turn, implies the desired approximation

$$||u_\varepsilon - u||^2_{H^m(U)} = \sum_{|\alpha|\leqslant m} ||D^\alpha u_\varepsilon - D^\alpha u||^2_{L^2(U)} \to 0.$$

It also establishes that $u_\varepsilon - u \in H^m(U)$, so $u_\varepsilon = (u_\varepsilon - u) + u \in H^m(\Omega)$, too. This is all that was to prove.

So let us show (4.8). To work with the definition of the weak derivative, we require a test function $\phi \in C^\infty_c(U)$. One obtains

$$\begin{aligned}
&\int_U u_\varepsilon(x)(D^\alpha\phi)(x)\,\mathrm{d}x \\
&\quad= \int_U (J_\varepsilon * u)(x)(D^\alpha\phi)(x)\,\mathrm{d}x && \text{(by (4.5))} \\
&\quad= \int_U \left[\int_{\mathbb{R}^d} J_\varepsilon(y)u(x-y)\,\mathrm{d}y\right](D^\alpha\phi)(x)\,\mathrm{d}x && \text{(by lemma 4.2 (b))} \\
&\quad= \int_{\mathbb{R}^d}\int_U J_\varepsilon(y)u(x-y)(D^\alpha\phi)(x)\,\mathrm{d}x\mathrm{d}y && \text{(by Fubini's theorem)} \\
&\quad= \int_{\mathbb{R}^d} J_\varepsilon(y)\left[\int_U u(x-y)(D^\alpha\phi)(x)\,\mathrm{d}x\right]\mathrm{d}y && \text{(by Fubini's theorem)} \\
&\quad= \int_{\mathbb{R}^d} J_\varepsilon(y)\left[\int_U (-1)^\alpha(D^\alpha u)(x-y)\phi(x)\,\mathrm{d}x\right]\mathrm{d}y && \text{(by definition of the weak derivative)} \\
&\quad= (-1)^\alpha\int_U \left[\int_{\mathbb{R}^d} J_\varepsilon(y)(D^\alpha u)(x-y)\mathrm{d}y\right]\phi(x)\,\mathrm{d}x && \text{(by Fubini's theorem)} \\
&\quad= (-1)^\alpha\int_U (J_\varepsilon * D^\alpha u)(x)\phi(x)\,\mathrm{d}x. && \text{(by lemma 4.2 (b))}
\end{aligned}$$

Since ϕ was arbitrary, this establishes (4.8) which ends the proof. ■

Proving the titular density result will require a product rule for weak derivatives. For its formulation we remind the reader that two multi-indices $\alpha = (\alpha_1, \ldots, \alpha_d), \beta = (\beta_1, \ldots, \beta_d) \in \mathbb{N}_0^d$ are compared with

$$\alpha \leqslant \beta :\iff \alpha_i \leqslant \beta_i \text{ for all } i = 1, \ldots, d.$$

We will also set $\alpha! := \alpha_1! \cdot \ldots \cdot \alpha_d!$ for a multi-index' factorial. This allows us to define a binomial coefficient for multi-indices via

$$\binom{\alpha}{\beta} := \binom{\alpha_1}{\beta_1} \cdot \ldots \cdot \binom{\alpha_d}{\beta_d} = \frac{\alpha_1!}{\beta_1!(\alpha_1-\beta_1)!} \cdot \ldots \cdot \frac{\alpha_d!}{\beta_d!(\alpha_d-\beta_d)!} = \frac{\alpha!}{\beta!(\alpha-\beta)!}.$$

With this notation, we can generalize a well-known result for regular derivatives.

Lemma 4.10 (Leibniz rule)
Let $\Omega \subseteq \mathbb{R}^d$ be a domain, $u \in H^m(\Omega)$ for some $m \in \mathbb{N}$ and $\psi \in C^\infty_c(\Omega)$. Then $\psi u \in H^m(\Omega)$, too, and for a multi-index $\alpha \in \mathbb{N}_0^d$ with $|\alpha| \leqslant m$ we have

$$D^\alpha(\psi u) = \sum_{\beta\leqslant\alpha} \binom{\alpha}{\beta} D^\beta\psi D^{\alpha-\beta}u.$$

Proof: The proof is by induction on $|\alpha|$. Suppose first that $|\alpha| = 1$. Pick $\phi \in C_c^\infty(\Omega)$. Then $\phi\psi \in C_c^\infty(\Omega)$, too. Hence,

$$\int_\Omega \frac{\partial u}{\partial x_i}\phi\psi \,\mathrm{d}x = -\int_\Omega u\frac{\partial}{\partial x_i}(\phi\psi)\,\mathrm{d}x = -\int_\Omega u\left(\frac{\partial\phi}{\partial x_i}\psi + \phi\frac{\partial\psi}{\partial x_i}\right)\mathrm{d}x.$$

for all $i = 1, \ldots, d$, using the definition of the weak derivative and the standard chain rule for differentiable functions. This is equivalent to

$$-\int_\Omega (\psi u)\frac{\partial\phi}{\partial x_i}\,\mathrm{d}x = \int_\Omega \left(\frac{\partial\psi}{\partial x_i}u + \psi\frac{\partial u}{\partial x_i}\right)\phi\,\mathrm{d}x \quad \text{for all } i = 1, \ldots, d,$$

so we have deduced

$$\frac{\partial}{\partial x_i}(\psi u) = \frac{\partial\psi}{\partial x_i}u + \psi\frac{\partial u}{\partial x_i} \in L^2(\Omega) \quad \text{for all } i = 1, \ldots, d$$

in the weak sense. This establishes the base case of the induction. For details on the induction step, see the proof of theorem 1 in subsection 5.2.3 in [7]. ■

We can now give the desired approximation theorem for Sobolev spaces. It shows that any Sobolev function can be approximated arbitrarily well by a smooth Sobolev function. Particularly noteworthy about this result is that it does not require the domain's boundary to have any regularity.

Theorem 4.11 (Meyers-Serrin theorem) [1, Theorem 3.17]
Let $\Omega \subseteq \mathbb{R}^d$ be a domain and $m \in \mathbb{N}$. Then $C^\infty(\Omega) \cap H^m(\Omega)$ is dense in $H^m(\Omega)$, i. e., for every $u \in H^m(\Omega)$ there exists a sequence $(u_k)_{k\in\mathbb{N}}$ in $C^\infty(\Omega) \cap H^m(\Omega)$ such that

$$||u_k - u||_{H^m(\Omega)} \to 0 \quad \text{as} \quad k \to \infty.$$

Proof: The sets

$$O_k := \left\{x \in \Omega \mid \operatorname{dist}(x, \partial\Omega) > \tfrac{1}{k} \text{ and } |x| < k\right\}, \quad k \in \mathbb{N}$$

are open as they are the intersection of two open sets, namely $O_k = \Omega_{1/k} \cap B_k(0)$ (cf. (4.6)), and bounded, so $O_k \subset\subset \Omega$ for all $k \in \mathbb{N}$. Also, define $O_0 := \varnothing$ and $O_{-1} := \varnothing$. Then the layers (see figure 4.4)

$$U_k := O_{k+1}\backslash\overline{O_{k-1}}, \quad k \in \mathbb{N}_0$$

are also open as the finite intersection of two open sets, namely $U_k = O_{k+1} \cap (\mathbb{R}^d\backslash\overline{O_{k-1}})$. The reason why we cannot work with $O_{k+1}\backslash\overline{O_k}$ directly is that the boundary ∂O_k would be missing.

Notice that the O_k form an open cover of Ω, i. e. $\Omega \subseteq \bigcup_{k=-1}^\infty O_k$, because any point in Ω is eventually (i. e. for sufficiently large k) contained in O_k. Hence, the U_k also form an open cover of Ω, i. e. $\Omega \subseteq \bigcup_{k=0}^\infty U_k$. Now choose a corresponding partition of unity $(\psi_k)_{k\in\mathbb{N}_0}$ subordinate to the cover $U_0, U_1, \ldots$ (cf. chapter I, section I of the lecture or theorem 3.15 in [1]). Recall that this means that $\psi_0, \psi_1, \ldots \in C_c^\infty(\mathbb{R}^d)$ with

(i) $\operatorname{supp}\psi_k \subseteq U_k$, $k \in \mathbb{N}_0$;

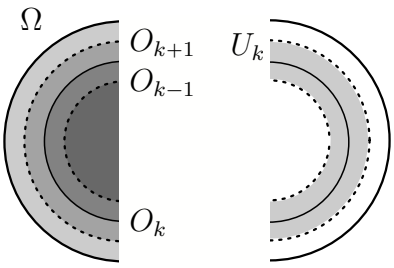

Figure 4.4: The sets in the proof of the Meyers-Serrin theorem.

(ii) $0 \leqslant \psi_k(x) \leqslant 1$ for all $x \in U_k$, $k \in \mathbb{N}_0$;

(iii) if $K \subset\subset \Omega$, then $\psi_k(x) = 0$ in $x \in K$ for all but finitely many $k \in \mathbb{N}_0$;

(iv) $\sum_{k=0}^{\infty} \psi_k(x) = 1$ for all $x \in \Omega$.

With this setup, we are now able to prove the approximation result. To this end, pick $u \in H^m(\Omega)$ and let $\varepsilon > 0$ be given. For fixed $k \in \mathbb{N}_0$ we have $\psi_k u \in H^m(\Omega)$ by the Leibniz rule (lemma 4.10) and $\operatorname{supp} \psi_k u \subseteq U_k$ by (i). In fact, $\operatorname{supp} \psi_k u$ is compact (because U_k is bounded), so $\operatorname{dist}(\operatorname{supp} \psi_k u, \partial U_k) > 0$ because U_k is open. We can thus use lemma 4.3 (b) and lemma 4.9 to choose a sufficiently small $\varepsilon_k > 0$ such that

$$\operatorname{supp}\big(J_{\varepsilon_k} * (\psi_k u)\big) \subseteq U_k \tag{4.9}$$

and

$$||J_{\varepsilon_k} * (\psi_k u) - \psi_k u||_{H^m(U_k)} < \frac{\varepsilon}{2^k}. \tag{4.10}$$

Now set

$$v_\varepsilon := \sum_{k=0}^{\infty} J_{\varepsilon_k} * (\psi_k u).$$

Notice that $v_\varepsilon \in C^\infty(\Omega)$ because for any point $x \in \Omega$ we can choose a neighborhood $K \subset\subset \Omega$ and by (iii) the series then sums over only finitely many nonzero terms, each of which is C^∞ in a neighborhood of x itself (cf. theorem 4.5 (a)). Since $u = (\sum_{k=0}^{\infty} \psi_k)u = \sum_{k=0}^{\infty} \psi_k u$ by (iv), we find

$$\begin{aligned}
||v_\varepsilon - u||_{H^m(\Omega)} &= \left|\left| \sum_{k=0}^{\infty} J_{\varepsilon_k} * (\psi_k u) - \sum_{k=0}^{\infty} \psi_k u \right|\right|_{H^m(\Omega)} \\
&\leqslant \sum_{k=0}^{\infty} ||J_{\varepsilon_k} * (\psi_k u) - \psi_k u||_{H^m(\Omega)} \\
&\overset{(4.9)}{=} \sum_{k=0}^{\infty} ||J_{\varepsilon_k} * (\psi_k u) - \psi_k u||_{H^m(U_k)} \\
&\overset{(4.10)}{<} \sum_{k=0}^{\infty} \frac{\varepsilon}{2^k} \\
&= \varepsilon.
\end{aligned}$$

In particular, this proves $v_\varepsilon - u \in H^m(\Omega)$, so $v_\varepsilon = (v_\varepsilon - u) + u \in H^m(\Omega)$, too. Hence, setting $u_k := v_{1/k}$ for $k \in \mathbb{N}$ yields the desired approximating sequence. ∎

It is worth noting that the Meyers-Serrin theorem actually allows an alternative definition of $H^m(\Omega)$, namely

$$H^m(\Omega) := \overline{C^\infty(\Omega) \cap H^m(\Omega)}$$

where the closure is taken with respect to the $||\cdot||_{H^m(\Omega)}$-norm. This is somewhat akin to how we defined the spaces $H_0^m(\Omega)$ in the lecture.

Notice that the approximating smooth functions from the Meyers-Serrin theorem are only smooth on Ω in general. This means that they could still "blow up" towards the boundary $\partial\Omega$ (like $x \mapsto \frac{1}{x}$ on $\Omega :=]0,1[$ for example). As it turns out, preventing this is only possible when the boundary of the domain is not too irregular (see example 3.20 in [1]). For example, it is possible to show the following density theorem.

Theorem 4.12 *Let $\Omega \subseteq \mathbb{R}^d$ be a domain with C^1 boundary and $m \in \mathbb{N}$. Then the set of restrictions of the functions from $C_c^\infty(\mathbb{R}^d)$ to Ω is dense in $H^m(\Omega)$, i. e., for every $u \in H^m(\Omega)$ there exists a sequence $(u_k)_{k\in\mathbb{N}}$ in $C_c^\infty(\mathbb{R}^d)$ such that*

$$||u_k - u||_{H^m(\Omega)} \to 0 \quad \text{as} \quad k \to \infty.$$

A proof of this important result can be found in Adam's and Fournier's book [1, theorem 3.22], see also section 4.11 there.

Acknowledgments

This text was written in the context of the module *Research in Groups - Numerical Mathematics and Applied Analysis* at the University of Würzburg, supervised by Dr. Eloi Martinet in the summer term of 2024. The course was titled "Finite Element Methods and Physics Informed Neural Networks". We would like to thank Philipp Graser who proofread parts of this document.

References

[1] R. A. Adams and J. J. F. Fournier. *Sobolev spaces*. 2nd edition. New York: Academic Press, 2003. DOI: 10.1016/S0079-8169(13)62897-4.

[2] H. Amann and J. Escher. *Analysis III. Translation from the German by* Silvio Levy *and* Matthew Cargo. Basel: Birkhäuser, 2009. DOI: 10.1007/978-3-7643-7480-8.

[3] A. Bressan. *Lecture notes on functional analysis. With applications to linear partial differential equations*. Providence: American Mathematical Society, 2013. DOI: 10.1090/gsm/143.

[4] J. Cen and Q. Zou. *Deep Finite Volume Method for High-Dimensional Partial Differential Equations*. 2024. arXiv: 2305.06863 [math.NA].

[5] N. S. Chauhan. *The Role of Physics-Informed Neural Networks in Deep Learning Evolution*. Link accessed on June 30, 2024. Jan. 6, 2023. URL: https://web.archive.org/web/20240712074246/https://www.theaidream.com/post/theroleof-physics-informed-neuralnetworks-in-deeplearning-evolution.

[6] C. M. Dafermos. *Hyperbolic Conservation Laws in Continuum Physics*. 4th edition. Berlin: Springer, 2016. DOI: 10.1007/978-3-662-49451-6.
[7] L. C. Evans. *Partial differential equations*. 2nd edition. Providence: American Mathematical Society, 2010. DOI: 10.1090/gsm/019.
[8] L. C. Evans and R. F. Gariepy. *Measure theory and fine properties of functions*. 2nd revised edition. Boca Raton: CRC Press, 2015. DOI: 10.1201/b18333.
[9] J. Glimm. "Solutions in the large for nonlinear hyperbolic systems of equations". In: *Communications on Pure and Applied Mathematics* 18 (1965), pp. 697–715. DOI: 10.1002/cpa.3160180408.
[10] E. Godlewski and P.-A. Raviart. *Numerical Approximations of Hyperbolic Systems of Conservation Laws*. 2nd edition. New York: Springer, 2021. DOI: 10.1007/978-1-0716-1344-3.
[11] R. J. LeVeque. *Numerical methods for conservation laws*. 2nd edition. Basel: Birkhäuser, 1992. DOI: 10.1007/978-3-0348-8629-1.
[12] E. H. Lieb and M. Loss. *Analysis*. 2nd edition. Providence: American Mathematical Society, 2001. DOI: 10.1090/gsm/014.
[13] J. D. Logan. *An introduction to nonlinear partial differential equations*. 2nd edition. New York, NY: John Wiley & Sons, 2008. DOI: 10.1002/9780470287095.
[14] M. Raissi, P. Perdikaris, and G. E. Karniadakis. *Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations*. 2017. arXiv: 1711.10561 [cs.AI].
[15] T. D. Ryck, S. Mishra, and R. Molinaro. *wPINNs: Weak Physics informed neural networks for approximating entropy solutions of hyperbolic conservation laws*. 2022. arXiv: 2207.08483 [math.NA].
[16] J. Smoller. *Shock Waves and Reaction-Diffusion Equations*. 2nd edition. New York: Springer, 1994. DOI: 10.1007/978-1-4612-0873-0.
[17] E. F. Toro. *Shock-capturing methods for free-surface shallow flows*. Chichester: Wiley, 2001.