



Sophie Steger, BSc

PHYSICS INFORMED NEURAL NETWORKS

analysis for a dynamical system - the double pendulum

MASTER'S THESIS

to achieve the university degree of

Diplom-Ingenieurin

Master's Degree Program in Electrical Engineering

submitted to

Graz University of Technology

Supervisors

Dipl.-Ing. Dr.techn. Bernhard C. Geiger

Dipl.-Ing. Franz M. Rohrhofer

Signal Processing and Speech Communication Laboratory

Graz, June, 2022

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

date

(signature)

Acknowledgements

First and foremost, I am deeply indebted to my primary supervisor, Bernhard Geiger, for giving me the opportunity to work on this interesting topic. His irreplaceable mentoring, immense knowledge, and continuous support helped me greatly throughout all stages of this work.

Also, I would like to express my deepest gratitude to my co-supervisor Franz Rohrhofer for his valuable knowledge, insightful comments, and patience, which significantly improved the quality of this work. It was the kind help and support of both supervisors that made this thesis possible.

Finally, and most importantly, I want to thank my family and friends for their encouragement, support, and company all through my studies. I would also like to thank my cats, who sat on my lap for many hours while I wrote this thesis.

Abstract (English)

Physics informed neural networks (PINNs) are an emerging class of deep learning methods capable of solving both forward and inverse problems of differential equations. They gained great popularity due to the seamless integration of both observational data and prior information about the underlying physical system in a combined multi-objective cost function. As a result of the additional physics loss term, PINNs can be employed in applications where purely data-driven methods are doomed to failure due to insufficient data quantity and quality. Despite extensive research, PINNs are still difficult to train, especially when little data is available and the optimization relies heavily on the physics loss term. In particular, PINNs suffer from severe convergence problems when simulating dynamical systems with high-frequency components, chaotic or turbulent behavior. In this work, we discuss the question of whether PINNs are a suitable method for predicting chaotic motion by conducting several experiments on the undamped double pendulum. The experimental results demonstrate that the additional information of the physics loss term effectively improves a purely data-driven approach in the presence of noisy, incomplete, or only partially observed data. However, their prediction accuracy degrades immensely in the chaotic regime. In contrast to the behavior of a chaotic system, PINNs do not exhibit any sensitivity to perturbations in the initial condition. Instead, PINNs consistently converge to certain highly attractive solutions that deviate strongly from the reference but display significantly lower values for the physics loss. We find that only a reduced computational domain combined with an appropriate loss weighting scheme allows convergence to the correct solution.

Abstract (German)

Physik-informierte Neuronale Netze (PINNs) sind eine aufstrebende Klasse von Deep-Learning-Methoden, die sowohl Vorwärts- als auch Rückwärtsprobleme von Differentialgleichungen lösen können. Sie gewinnen zunehmend an Popularität, da sie sowohl Messdaten als auch vorhandene Informationen über das zugrundeliegende physikalische System nahtlos in einer kombinierten mehrkriteriellen Kostenfunktion vereinen. Aufgrund des zusätzlichen physikalischen Verlustterms können PINNs in Anwendungen eingesetzt werden, bei denen rein datengesteuerte Methoden aufgrund unzureichender Datenmenge und -qualität zum Scheitern verurteilt sind. Trotz umfangreicher Forschung erweist sich das Training von PINNs aber nach wie vor als schwierig, vor allem, wenn nur begrenzt Daten zur Verfügung stehen und die Optimierung stark vom physikalischen Verlustterm abhängt. Insbesondere leiden PINNs unter gravierenden Konvergenzproblemen bei der Simulation von dynamischen Systemen mit hochfrequenten Komponenten und chaotischem oder turbulentem Verhalten. In dieser Arbeit erörtern wir nun die Frage, ob PINNs eine geeignete Methode zur Vorhersage chaotischer Bewegungen darstellen, indem wir verschiedene Experimente am ungedämpften Doppelpendel durchführen. Die experimentellen Ergebnisse zeigen, dass der zusätzliche physikalische Verlustterm die Genauigkeit eines rein datengesteuerten Modells bei verrauschten, unvollständigen oder nur teilweise beobachteten Daten wirksam verbessern kann. Allerdings sinkt die Zuverlässigkeit von PINNs im chaotischen Bereich drastisch. Im Gegensatz zum Verhalten eines chaotischen Systems zeigen PINNs keinerlei Sensibilität gegenüber Störungen in der Anfangsbedingung. Stattdessen konvergieren sie bevorzugt zu bestimmten dominanten Lösungen, die zwar stark von der Referenz abweichen, aber deutlich geringere Werte für den Physikverlust aufweisen. Wir stellen fest, dass nur ein reduziertes Berechnungsgebiet in Kombination mit einem geeigneten Verlustgewichtungsschema die Konvergenz zur richtigen Lösung gewährleisten kann.

Contents

Statutory Declaration	III
Acknowledgements	V
Abstract (English)	VII
Abstract (German)	IX
1 Introduction	1
2 Dynamical Systems	5
2.1 System Description	5
2.2 Numerical Methods for Differential Equations	6
2.2.1 Euler Method	6
2.2.2 Runge Kutta Methods	7
2.3 Double Pendulum	8
2.3.1 Derivation of Equations of Motion	8
2.3.2 Chaotic Behavior	9
3 Physics Informed Neural Networks	11
3.1 Problem Setup	12
3.1.1 Application: Double Pendulum	13
3.2 Preliminary Experiments	16
3.2.1 Second Order Derivatives	17
3.2.2 Feature Scaling	17
4 Comparison to Purely Data-Driven Neural Networks	19
4.1 Regularization in the Presence of Noisy Data	19
4.2 Reconstruction of Dynamics from Incomplete Data	23
4.3 Reconstruction of Dynamics from Partially Observed Data	27
4.4 Discussion	30
5 Forward Problems of Nonlinear Ordinary Differential Equations	31
5.1 From Periodicity to Chaos	31
5.2 Discussion	37
6 Convergence Issues and Failure Modes	39
6.1 Insufficient Expressivity	41
6.2 Analysis of Physics Residuals	43
6.3 Change of Loss Weighting Scheme	46
6.4 Reduction of Computational Domain	50
6.5 Pareto Front	52
6.5.1 Concave Pareto Front	52
6.5.2 Irregular Pareto Front	53
6.6 Summary	55
7 Conclusion	57

Introduction

Artificial intelligence and machine learning are emerging areas of modern computer science and research. Over the past few decades, machine learning techniques have evolved significantly, spawning a wide range of algorithms. Among those, neural networks (NNs) have gained great popularity as powerful multivariate nonlinear function approximators and are employed in various fields of science and engineering. In particular, this often involves problems concerning modeling and simulating complex physical systems. However, despite their great empirical success in various applications, purely data-driven NNs are highly dependent on the amount and quality of training data. For instance, they exhibit low robustness and generalization capabilities when data is noisy, incomplete, or scarce.

To alleviate those limitations, new branches of deep learning have developed, including *physics-enhanced machine learning* [1]. In addition to observational data, prior knowledge about the underlying physical system is incorporated. By providing the NN model with information about the governing physical equations, the space of admissible solutions is restricted to those that satisfy the physical rules. One recent and popular technique following this type of new informed learning is called *physics informed neural networks (PINNs)* [2]. PINNs integrate physical knowledge into the NN training by incorporating residuals of the governing differential equations via a physics loss function. This physics loss acts as an additional regularization term, penalizing solutions that violate the physical laws of the underlying system. Therefore, PINNs can be trained with little data and are robust against noisy and incomplete data.

The residuals of the physical equations are evaluated on arbitrary collocation points using automatic differentiation (AD) techniques, that allow for an efficient and accurate evaluation of the derivatives. Since collocation points can be sampled randomly at any point in time and space, PINNs provide a continuous-time and mesh-free approach to solving both forward and inverse problems of differential equations. They have proven successful in numerous applications, including multi-physics simulations [3]–[6], system identification [7]–[9], and inference of hidden states [9]–[11]. However, it is still frequently reported that PINNs suffer from severe convergence problems, especially for large computational domains [12]–[16] and solutions with strongly nonlinear, chaotic, or turbulent behavior [2], [17], [18]. These convergence issues have led to the development of several extensions of the original framework to improve accuracy and reliability.

It has been found that the numerous competing loss terms in the multi-objective cost function lead to convergence problems [19], [20]. In particular, the authors of [19] argue that this is a result of the stiffness of the gradient flow. Instead of manually adjusting the loss weights, they proposed an annealing procedure to adaptively balance the different loss terms using the backpropagated gradient statistics. Moreover, in [14] the importance of suitable loss weighting

schemes was shown by visualizing the Pareto front of the multi-objective cost function. In multi-objective optimization, the Pareto front is defined as the set of solutions that can be achieved by the optimization algorithm (Pareto-optimal solutions). In a Pareto optimum, no loss term can be reduced without increasing the other, and vice versa. The authors of [14] showed that the position of the Pareto optimum for PINNs strongly depends on the choice of the loss weighting scheme. Also, the innate shape of the Pareto front has an effect on whether the optimization algorithm is able to reach an accurate solution at all or converges to a suboptimal minimum. Overall, the shape of the Pareto front has been shown to be influenced by many factors, including network architecture, loss weights, or system parameters such as the computational domain.

Several other studies confirmed the influence of the computational domain and suggested different methods to improve the performance of PINNs [12], [13], [16], [21], [22]. From ongoing discussions, it is apparent that several factors contribute to convergence problems caused by a large computational domain. For one, as the size of the computational domain increases, the number of collocation points must be increased accordingly. For sparsely distributed collocation points, the authors of [21] have shown that the predicted solution decays to the trivial solution between collocation points without increasing the resulting physics loss term. In [21] this effect was mitigated by penalizing the gradient of the physics loss. A similar approach was adopted in [22] by proposing an additional loss term using gradient information of the physics residuals. Furthermore, a larger computational domain generally leads to a more complex solution. In order to achieve sufficient expressiveness, the size of the NN must be increased accordingly, resulting in slower convergence times. Besides that, the so-called spectral bias [23] causes NNs to tend to learn functions with low frequencies at a faster rate. A study of [17] has revealed that spectral bias also affects PINNs, which may impede their convergence to an accurate solution for large domains as well as high-frequency components.

A popular approach to mitigate these problems is domain decomposition, where the complex optimization problem is decomposed into smaller problems of lower complexity that are solved individually. For PINNs, the computational domain is partitioned and a separate network is trained for each subdomain. For instance, [16] subdivides the spatio-temporal domain and sequentially trains a separate PINN for each segment, rather than for the entire domain at once. With this approach, the authors were able to achieve higher accuracy and better convergence properties. In order to significantly reduce training times, the subdomains can also be solved in parallel. However, this approach requires cross-boundary communication between adjacent subdomains to ensure continuity between the individual solutions. In [12], the authors propose a domain decomposition approach called *eXtendend PINNs (XPINNs)*. The subdomains are solved in parallel, and continuity between adjacent segments is realized via a soft constraint approach in the form of an additional interface term in the cost function. A similar approach is employed in *Finite Basis PINNs (FBPINNs)* [13], where the individual subdomains partially overlap and the boundary conditions are implemented as hard constraints. The continuity of the interface is enforced by constructing a suitable solution ansatz, thereby avoiding an additional loss term. Overall, reducing the size of the computational domain by appropriate domain decomposition methods has proven successful in a number of applications and mitigates common convergence problems.

In particular, the length of the computational domain plays an important role in solving

forward problems of dynamical systems. For dynamical systems, the computational domain corresponds to the simulation time. For long domains, it frequently occurs that the PINN converges towards the (potentially incorrect and nonphysical) trivial solution of the system. In [15], the authors have aimed to understand and explain these convergence problems in more detail through experiments on a simple dynamical system. In this paper, they have visualized the loss landscape of the dynamical system for gradually increasing domain lengths. On the one hand, an increased complexity of the loss landscape was observed for larger computational domains. This resulted in a more challenging optimization problem. On the other hand, the authors showed that the (incorrect) trivial solution could result in much lower physics residuals compared to the analytical solution. For larger computational domains, this solution can potentially become so economical that it forms the global minimum of the multi-objective optimization problem.

In this work, we investigate whether the current PINNs framework is an appropriate choice for solving forward problems of dynamical systems that exhibit chaotic or turbulent behavior. The purpose of this work is therefore to gain a deeper understanding of PINNs by performing experiments on a nonlinear dynamical system, the undamped double pendulum. We confirm convergence problems of the standard PINN for large computational domains, and solutions with high frequencies or chaotic motion. We characterize two different failure modes for the periodic and the chaotic case and identify the physics loss term as the common root cause. Our work is closely related to the observations of [15] by showing that the correct solution does not necessarily correspond to the global minimum of the cost function. We demonstrate in several experiments that the physics residuals alter in magnitude as a function of the solution trajectory. This favors convergence to certain attractive solutions that produce overall lower loss scores compared to the correct solution. We show that the only robust way of mitigating such problems is to reduce the computational domain using a domain decomposition approach.

The remainder of the thesis is organized as follows.

Chapter 2 All the necessary notations and equations to describe the double pendulum are introduced, including an overview of traditional numerical methods for simulating the dynamical system and their limitations.

Chapter 3 We then present the framework of PINNs which follows the original formulation of [2], as well as extensions such as adaptive loss weighting schemes [19]. In addition, we discuss the influence of preprocessing steps (normalization of input variables) and differences in the formulation of ordinary differential equation (ODE) schemes.

Chapter 4 A comparison between purely data-driven NNs and PINNs is evaluated, highlighting their advantages and difficulties. We show that the physics loss term of the PINN allows accurate prediction of the solution even in the presence of noisy, incomplete, or only partially available data. In the chaotic regime, however, the physics loss term can impede convergence to a correct solution.

Chapter 5 Next, the PINN is used to simulate the undamped double pendulum in the harmonic and chaotic regime. Our results show that PINNs can be used for simulation tasks, but exhibit two different failure modes in the harmonic and chaotic regime of the system.

Chapter 6 The convergence problems for the harmonic and chaotic cases are addressed. We demonstrate the influence of loss weighting schemes and the success of reduced computational domains through several experiments. Finally, the difficulty of training PINNs on long computational domains is discussed, showing that the correct solution does not necessarily correspond to the global minimum.

Chapter 7 We summarize our observations, discuss the implications of our work, and provide an outlook for future research.

Dynamical Systems

Dynamical systems describe the time-evolution of a point in state space governed by some rule. In mathematics, dynamical systems are represented by higher-order differential equations for continuous time systems, and by higher-order difference equations for discrete-time systems. Phenomena of dynamic systems arise in various scientific fields, including heating systems, flow of water, population growth, ecological decay, to name a few. In this work, we study the motion of a double pendulum – a continuous-time dynamical system with rich dynamics that exhibits both harmonic oscillations and chaotic behavior. Therefore, we give a brief overview of continuous-time dynamical systems and derive the state equations of the double pendulum. Finally, we discuss traditional numerical methods for solving differential equations. The content of this following chapter is based on the the books of John C. Butcher [24] and Steven H. Strogatz [25].

2.1 System Description

In continuous-time dynamical systems, we typically consider differential equations of the form

$$y^{(n)} = f(t, y, y', y'', \dots, y^{(n-1)}) . \quad (2.1)$$

Any n^{th} -order differential equation can be converted into a n -dimensional system of coupled first-order nonlinear differential equations of the form

$$\begin{pmatrix} y_1' \\ y_2' \\ \vdots \\ y_n' \end{pmatrix} = \begin{pmatrix} f_1(t, y_1, y_2, \dots, y_n) \\ f_2(t, y_1, y_2, \dots, y_n) \\ \vdots \\ f_n(t, y_1, y_2, \dots, y_n) \end{pmatrix} \quad (2.2)$$

and written more compactly in vector notation

$$\mathbf{y}' = \mathbf{F}(t, \mathbf{y}) \quad \mathbf{y} = [y_1, y_2, \dots, y_n]^T \quad y_i = y^{(i-1)} \quad \text{for } i = 1, \dots, n \quad (2.3)$$

where \mathbf{F} is a vector-valued function denoted as the flow. If $\mathbf{F}(t, \mathbf{y}) = \mathbf{F}(\mathbf{y})$, the system is said to be *autonomous* or *time-independent*.

Starting from any point \mathbf{y}_0 in the state space, we can determine the derivatives that correspond to the direction of the system movement. The set of future points that arise from a given initial condition is called *trajectory*, in continuous-time systems forming a curve in state space.

However, for many dynamical systems a closed-form solution of the trajectory does not exist; in particular most nonlinear systems cannot be solved analytically. In such cases, one has to resort to numerical methods to approximate the solution.

2.2 Numerical Methods for Differential Equations

In this section, we give a brief overview of two important methods to perform numerical integration of an initial value problem $y' = f(y)$ subject to $y(t_0) = y_0$. Evidently, these methods can also be applied to higher-order ODEs by transformation into a system of first-order ODEs.

2.2.1 Euler Method

The Euler method, first published by Euler in his book *Institutiones Calculi Integralis* (1768-1770), is a *finite difference method* for performing numerical integration. Although it is seldom used in practice anymore, its concept is fundamental for more sophisticated integration methods.

The idea is simple. Starting from an arbitrary starting point $y(t_0) = y_0$, we want to determine the velocity v_0 with which this point is moving by evaluating the differential equation $v_0 = y'_0 = f(y_0)$. Following this tangent we move to the next point y_1 . Suppose we choose a time step Δt sufficiently small such that the velocity remains constant during this period. Then we can approximate the next position as follows

$$y_1 = y_0 + (t_1 - t_0)f(y_0) . \tag{2.4}$$

This procedure is iteratively repeated for $t_n = t_{n-1} + \Delta t$

$$y_n = y_{n-1} + (t_n - t_{n-1})f(y_{n-1}) \tag{2.5}$$

to obtain an approximation to the trajectory. If the function $\mathbf{y}' = f(\mathbf{y})$ is vector-valued, we can apply the Euler method to each component.

Although the Euler's method is the simplest numerical integration scheme with many desirable properties, it suffers from several limitations. The error that occurs in a single step (*local truncation error*) is proportional to the squared step size Δt^2 . Moreover, stability problems may arise, especially for stiff equations. This motivated the design of other integration techniques that yield more accurate approximations without needing to drastically decrease the step size, among those higher-order Runge Kutta (RK) methods.

2.2.2 Runge Kutta Methods

RK methods are one of the best known numerical techniques for solving initial value problems governed by differential equations. Originally proposed by Runge (1895), additional contributions were added by Heun (1900) and Kutta (1901). The core idea is the generalization of the Euler method by evaluating a number of derivatives in one step. Since a detailed derivation is beyond the scope of this thesis, we briefly outline the *fourth-order Runge-Kutta method*.

Here, the approximation of the next step y_n in terms of y_{n-1} is given by

$$y_n = y_{n-1} + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (2.6)$$

with

$$k_1 = f(y_{n-1}) \Delta t \quad (2.7)$$

$$k_2 = f\left(y_{n-1} + \frac{1}{2}k_1\right) \Delta t \quad (2.8)$$

$$k_3 = f\left(y_{n-1} + \frac{1}{2}k_2\right) \Delta t \quad (2.9)$$

$$k_4 = f(y_{n-1} + k_3) \Delta t, \quad (2.10)$$

obtained by evaluating the function $f(\cdot)$ in each step four times. Typically referred to as RK45, this scheme is a fourth-order method with a local truncation error proportional to Δt^4 . In general, this method allows for more accurate results compared to a simple Euler method. RK45 is one of the most popular integration schemes for practical purposes, since it offers a good trade-off between accuracy and computational cost.

For the remainder of this thesis, we employ RK45¹ to compute the reference solution for all subsequent simulation problems.

¹ `scipy.integrate.solve_ivp`

2.3 Double Pendulum

The double pendulum is a popular example for a simple nonlinear dynamical system that exhibits chaotic motion. In this work, we consider an undamped double pendulum with point masses m_1 , m_2 and length of each pendulum L_1 , L_2 , illustrated in Fig. 2.1. The angle between each pendulum and the vertical axis is denoted as θ_1 , θ_2 .

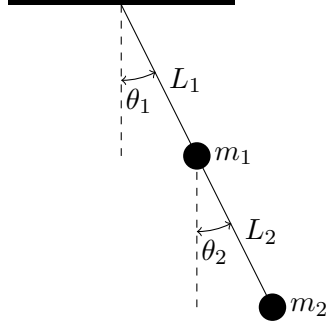


Figure 2.1: Schematic of the double pendulum with point masses m_1 , m_2 , and lengths L_1 , L_2 .

2.3.1 Derivation of Equations of Motion

In this following section we derive the set of nonlinear ODEs that describe the motion of the double pendulum [26].

First, the position of each pendulum is given by

$$\begin{aligned} x_1 &= L_1 \sin \theta_1 \\ y_1 &= -L_1 \cos \theta_1 \\ x_2 &= L_1 \sin \theta_1 + L_2 \sin \theta_2 \\ y_2 &= -L_1 \cos \theta_1 - L_2 \cos \theta_2 . \end{aligned}$$

Then, the potential energy V and the kinetic energy T are

$$\begin{aligned} V &= m_1 g y_1 + m_2 g y_2 \\ &= -(m_1 + m_2) g L_1 \cos \theta_1 - m_2 L_2 g \cos \theta_2 \\ T &= \frac{1}{2} m_1 v_1^2 + \frac{1}{2} m_2 v_2^2 \\ &= \frac{1}{2} m_1 L_1^2 \theta_1'^2 + \frac{1}{2} m_2 \left[L_1^2 \theta_1'^2 + L_2^2 \theta_2'^2 + 2 L_1 L_2 \theta_1' \theta_2' \cos(\theta_1 - \theta_2) \right] \end{aligned}$$

where $g = 9.81 \text{ m s}^{-2}$ is the gravitational acceleration. As the undamped double pendulum is a Hamiltonian system, its total energy $V + T$ is a constant quantity.

We can now write the Lagrangian with some simplifications

$$\begin{aligned} L &\equiv T - V \\ &= \frac{1}{2} (m_1 + m_2) L_1^2 \theta_1'^2 + \frac{1}{2} m_2 L_2^2 \theta_2'^2 + m_2 L_1 L_2 \theta_1' \theta_2' \cos(\theta_1 - \theta_2) + (m_1 + m_2) g L_1 \cos \theta_1 - m_2 L_2 g \cos \theta_2 . \end{aligned}$$

Applying the Euler-Lagrange equations

$$\frac{d}{dt} \frac{\partial L}{\partial \theta'_i} - \frac{\partial L}{\partial \theta_i} = 0 \quad (2.11)$$

and further rearranging yields the second-order ODEs that describe the motion of the pendulum. To simplify notation, we denote $\theta'_1 = \omega_1$, $\theta'_2 = \omega_2$ and $\Delta\theta = \theta_1 - \theta_2$.

$$\begin{pmatrix} \theta''_1 \\ \theta''_2 \end{pmatrix} = \begin{pmatrix} f_1(\theta_1, \theta_2, \omega_1, \omega_2) \\ f_2(\theta_1, \theta_2, \omega_1, \omega_2) \end{pmatrix} \quad (2.12)$$

$$= \begin{pmatrix} \frac{m_2 L_1 \omega_1^2 \sin(2\Delta\theta) + 2m_2 L_2 \omega_2^2 \sin \Delta\theta + 2gm_2 \cos \theta_2 \sin \Delta\theta + 2gm_1 \sin \theta_1}{-2L_1(m_1 + m_2 \sin^2 \Delta\theta)} \\ \frac{m_2 L_2 \omega_2^2 \sin(2\Delta\theta) + 2(m_1 + m_2) L_1 \omega_1^2 \sin \Delta\theta + 2g(m_1 + m_2) \cos \theta_1 \sin \Delta\theta}{2L_2(m_1 + m_2 \sin^2 \Delta\theta)} \end{pmatrix} \quad (2.13)$$

$$\mathbf{y}'' = \mathbf{F}(\mathbf{y}, \mathbf{y}') \quad \text{with} \quad \mathbf{y} = [\theta_1, \theta_2]^T \quad (2.14)$$

Lastly, we can rewrite the second-order ODE to obtain a set of first-order coupled ODEs

$$\begin{pmatrix} \theta'_1 \\ \theta'_2 \\ \omega'_1 \\ \omega'_2 \end{pmatrix} = \begin{pmatrix} \omega_1 \\ \omega_2 \\ f_1(\theta_1, \theta_2, \omega_1, \omega_2) \\ f_2(\theta_1, \theta_2, \omega_1, \omega_2) \end{pmatrix} \quad (2.15)$$

$$\tilde{\mathbf{y}}' = \tilde{\mathbf{F}}(\tilde{\mathbf{y}}) \quad \text{with} \quad \tilde{\mathbf{y}} = [\theta_1, \theta_2, \omega_1, \omega_2]^T. \quad (2.16)$$

As f_1 and f_2 are not a function of the time t , the system is called *autonomous*.

2.3.2 Chaotic Behavior

A system with chaotic behavior exhibits a strong sensitive dependence on the initial conditions. Meaning that any arbitrary close trajectories will diverge exponentially. Fig. 2.2 shows the RK simulation of the double pendulum for an initial angle $\theta_1^{(t_0)} = \theta_2^{(t_0)} = \theta_0$ and zero initial velocities $\omega_1^{(t_0)} = \omega_2^{(t_0)} = 0$. All system parameters (m_1, m_2, L_1, L_2) are set to one. While a small initial angle leads to periodic oscillation (Fig. 2.2 (a)), larger angles result in complex trajectories with unpredictable motion (Fig. 2.2 (b), (c)).

To characterize initial conditions that lead to harmonic oscillations and chaotic motion, we study the bifurcation diagram for initial angles of $\theta_1^{(t_0)} = \theta_2^{(t_0)} = \theta_0$. Fig. 2.3 shows the qualitative behavior of the double pendulum for increasing initial angles θ_0 . For each initial condition θ_0 , the figure depicts the angular velocity θ'_1 at a certain value of the angular velocity θ'_2 , for this example at the zero crossings $\theta'_2 = 0$. In the case of harmonic oscillation, the zero crossings are at positions that repeat periodically. However, in the chaotic regime, the zero crossings of $\theta'_2 = 0$ can occur at any value of θ'_1 .

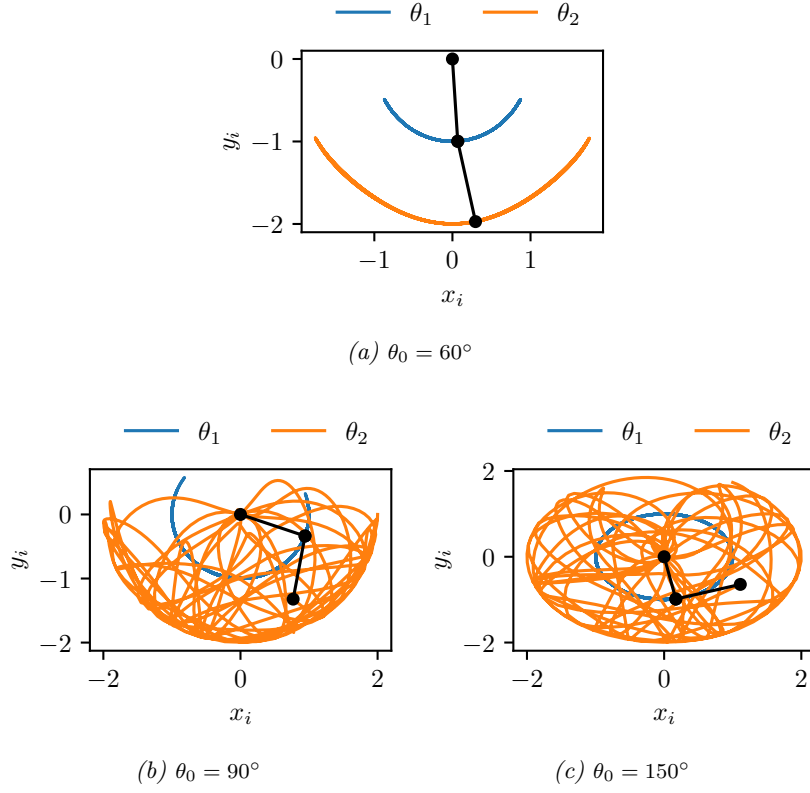


Figure 2.2: Trajectory of the double pendulum for different initial conditions resulting in harmonic oscillation (a) or chaotic motion (b) and (c).

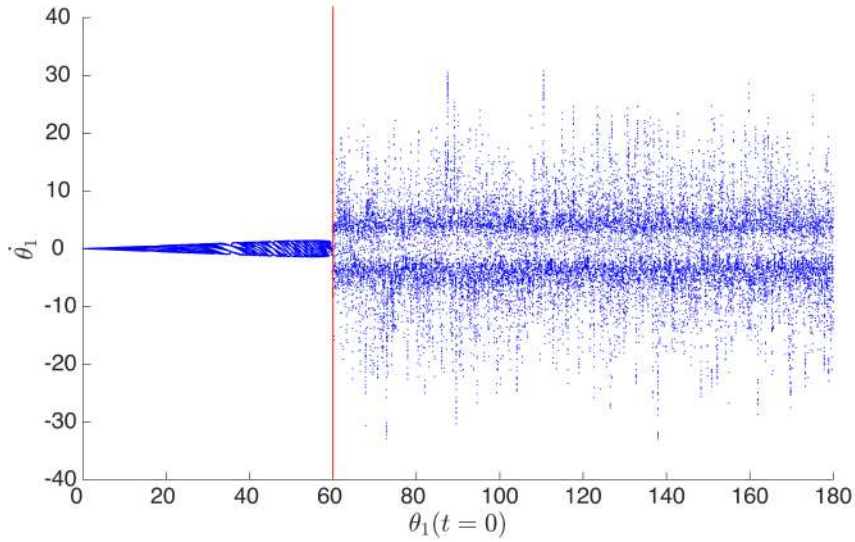


Figure 2.3: Bifurcation diagram for a double pendulum with equal lengths, equal masses, and 9.81 m/s^2 . The initial condition is given by $\mathbf{y}^{(t_0)} = [\theta_0, \theta_0, 0, 0]^T$ (taken from [27]).

Physics Informed Neural Networks

Forward and inverse problem solving of differential equations has a long history. In the past decades, traditional numerical methods have been proven to be successful in solving ordinary differential equations (ODEs) (*Runge Kutta methods*) and partial differential equations (PDEs) (*finite differences, finite element, spectral or mesh-free methods*). In most practical applications, these methods achieve sufficient accuracy and robustness. Despite their tremendous progress and widespread application in practice, there are still some major limitations and drawbacks to classical numerical methods [1]:

- Solving nonlinear multi-scale systems.
- Solving inverse problems for inferring unknown parameters of the governing differential equation.
- Solving ill-posed problems with missing, gappy, or noisy measurement data.

In view of the large amount of available measurement and observation data, there is a desire to utilize this information for solving differential equations. In this regard, physics informed neural networks (PINNs) [1], [2] have emerged as a novel class of machine learning algorithms that leverage the expressive abilities of deep neural networks. In addition, PINNs are able to seamlessly integrate data and physics models into one framework.

A lot of research on PINNs has demonstrated their abilities to effectively solve inverse problems for differential equations as well as ill-posed problems with insufficient data, outperforming traditional numerical methods [3]–[11]. However, when solving forward problems, PINNs still suffer from severe convergence issues and unresolved failure modes. It is therefore of our particular interest to analyze the PINN framework for the double pendulum both in the forward problem and in the presence of measurement data. Thus, we briefly outline the PINN algorithm for deriving the solution of a higher-order nonlinear ODE. For a complete description, we refer the reader to [2].

3.1 Problem Setup

We consider a higher-order ODE of the form

$$y^{(n)}(t) = f\left(t, y(t), y'(t), y''(t), \dots, y^{(n-1)}(t)\right), \quad t \in [0, t_{max}] \quad (3.1)$$

subject to the given initial conditions

$$y(t_0), y'(t_0), \dots, y^{(n-1)}(t_0) \quad (3.2)$$

where $y(t)$ is the unknown solution, f is a known nonlinear function and the variable t specifies the computational domain². For the multivariate case, the variables \mathbf{y} and \mathbf{F} denote vector-valued quantities. In the original framework of [2], the unknown solution $y(t)$ is approximated by a fully-connected deep neural network $y_{\mathbf{w}}(t)$ with \mathbf{w} describing all trainable weights and biases of the network.

The heart of the PINN now consists of combining data- and physics-based constraints into a single loss function via multi-objective optimization. This approach allows a seamless integration of information about the underlying physical system. For forward problems, the PINN is trained by minimizing the following multi-objective loss function.

$$L(\mathbf{w}) = L_{IC}(\mathbf{w}) + L_F(\mathbf{w}) \quad (3.3)$$

with the initial condition loss

$$L_{IC}(\mathbf{w}) = \frac{1}{n} \sum_{i=0}^{n-1} \left(y_{\mathbf{w}}^{(i)}(t_0) - y^{(i)}(t_0) \right)^2 \quad (3.4)$$

and the physics-based loss consisting of the ODE-residuals of the physical system

$$L_F(\mathbf{w}) = \frac{1}{N_{col}} \sum_{i=1}^{N_{col}} \left(y_{\mathbf{w}}^{(n)}(t) \Big|_{t=t_i} - f\left(t, y_{\mathbf{w}}(t), y'_{\mathbf{w}}(t), \dots, y_{\mathbf{w}}^{(n-1)}(t)\right) \Big|_{t=t_i} \right)^2 \quad (3.5)$$

where $\{t_i\}_{i=1}^{N_{col}}$ denotes the collocation points where the physics loss is evaluated. These collocation points can be chosen arbitrarily within the computational domain, either by using fixed locations or by adaptive resampling [3], [28], [29]. In particular, for PDE problems with complex geometries, this alleviates the need for complex mesh generation. To determine the derivatives $y'_{\mathbf{w}}, \dots, y_{\mathbf{w}}^{(n-1)}$ we make use of automatic differentiation (AD) [30]. Open software libraries such as PyTorch [31] or Tensorflow [32] provide efficient implementations.

² In the context of dynamical systems, the size of the computational domain corresponds to the simulation time.

Loss Weighting PINNs incorporate the data and physics residuals as soft constraints in a multi-objective manner. For well-posed problems, the two objective functions should not be contradictory and convergence to the correct solution, in principle, should be possible. However, it has been shown in recent studies [14], [19], [20] that the loss terms can still be conflicting which results in frequently observed convergence issues. Thus, an additional loss weighting parameter α_{data} is introduced to balance the individual losses. This hyperparameter can be chosen manually or tuned adaptively.

$$L(\mathbf{w}) = \alpha_{data} L_{IC}(\mathbf{w}) + (1 - \alpha_{data}) L_F(\mathbf{w}) \quad (3.6)$$

For the adaptive loss weighting scheme, we follow the framework of [19], [20] where the loss weighting parameter is updated dynamically using backpropagated gradient statistics during the network training. The weights \mathbf{w} of the network are trained via gradient descent as

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \eta \nabla_{\mathbf{w}} \left(\alpha_{data} L_{IC}(\mathbf{w}) + (1 - \alpha_{data,n}) L_F(\mathbf{w}_n) \right) \Big|_{\mathbf{w}=\mathbf{w}_n} \quad (3.7)$$

where η denotes the learning rate and n the iteration step. Now, the loss weighting parameter is adapted in each iteration using the mean value of the gradients

$$\hat{\alpha}_{data,n} = \frac{\left| \nabla_{\mathbf{w}} L_F(\mathbf{w}) \Big|_{\mathbf{w}=\mathbf{w}_n} \right|}{\left| \nabla_{\mathbf{w}} L_{IC}(\mathbf{w}) \Big|_{\mathbf{w}=\mathbf{w}_n} \right|} \quad (3.8)$$

$$\alpha_{data,n+1} = (1 - \lambda) \alpha_{data,n+1} + \lambda \hat{\alpha}_{data,n} \quad (3.9)$$

where the hyperparameter λ specifies the speed of the moving average update. For the application of the adaptive loss weighting scheme in later experiments in this work, we choose $\lambda = 0.1$.

3.1.1 Application: Double Pendulum

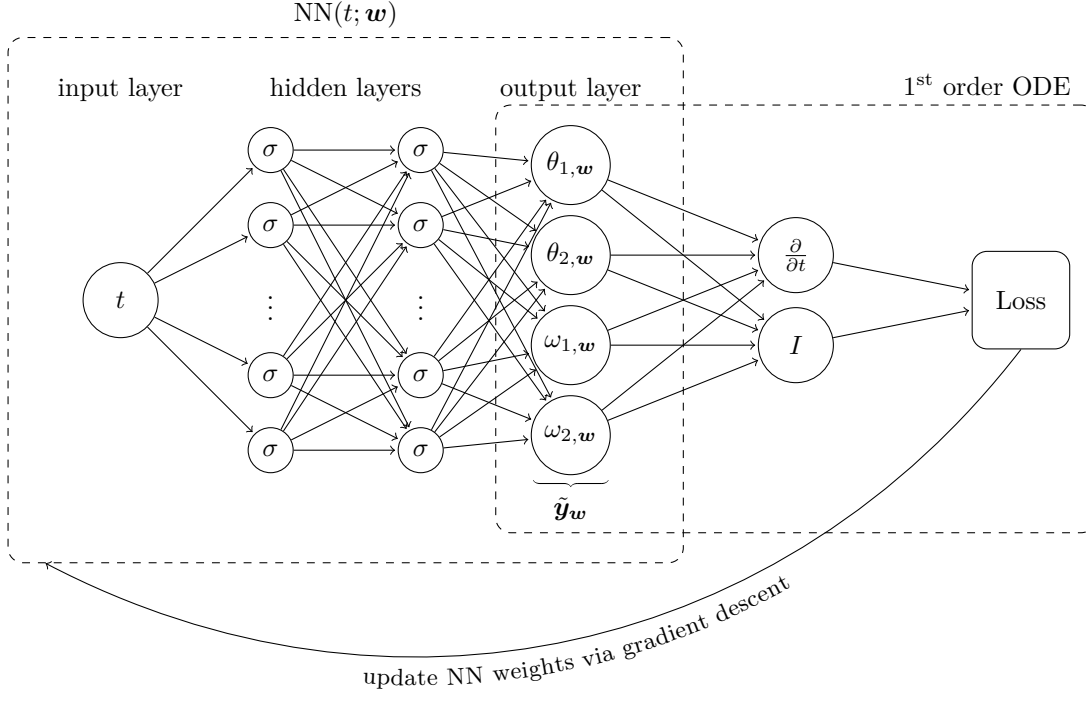
In all the following experiments of this thesis, we employ PINNs to study their behavior for the simulation of a double pendulum. As shown in Section 2.3, the motion of both pendulums is described by an initial-value problem of the form

$$\mathbf{y}'' = \mathbf{F}(\mathbf{y}, \mathbf{y}') \quad \text{with} \quad \mathbf{y} = [\theta_1, \theta_2]^T. \quad (3.10)$$

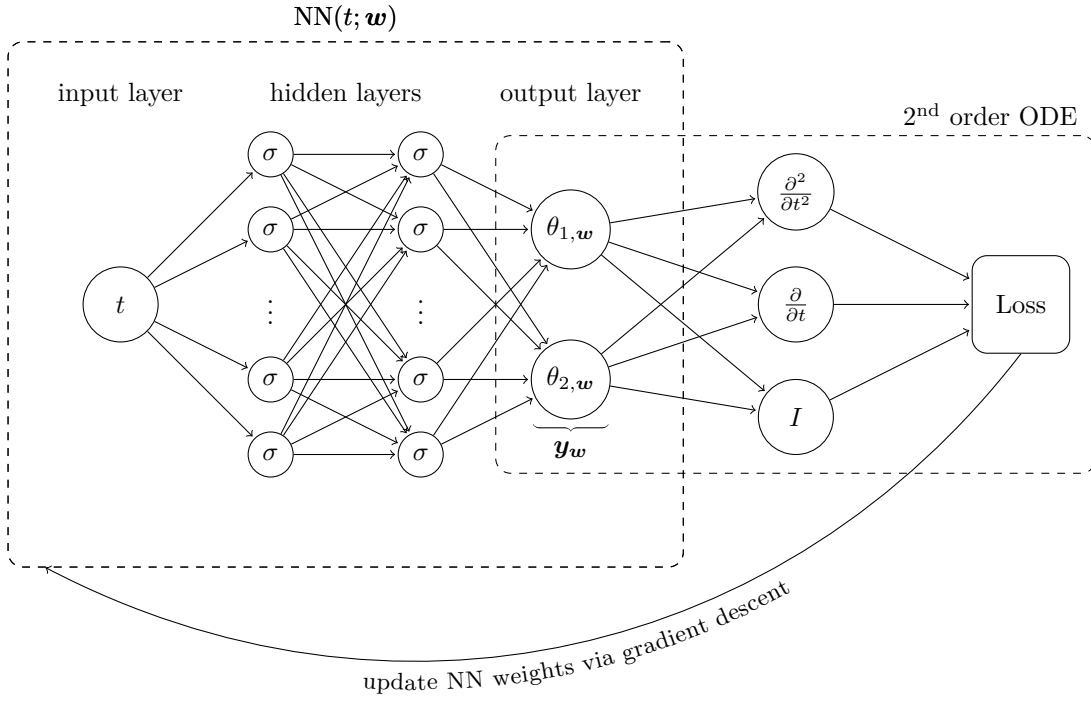
To approximate the solution of this dynamical system with RK, we need to convert the second-order ODEs into a system of first-order ODEs

$$\tilde{\mathbf{y}}' = \tilde{\mathbf{F}}(\tilde{\mathbf{y}}) \quad \text{with} \quad \tilde{\mathbf{y}} = [\theta_1, \theta_2, \omega_1, \omega_2]^T. \quad (3.11)$$

When using a PINN to solve this forward problem, we can use both the first and second-order formulations. While the first-order system (Fig. 3.1 (a)) learns the angular velocities by minimizing $\omega_{\mathbf{w},1} = \theta'_{1,\mathbf{w}}$, $\omega_{\mathbf{w},2} = \theta'_{2,\mathbf{w}}$, the second-order system (Fig. 3.1 (b)) utilizes an additional time derivative to obtain the velocities.



(a) First-order derivatives



(b) Second-order derivatives

Figure 3.1: Network architecture of the PINN for the first-order (a) and second-order (b) ODE description. The NN parameters \mathbf{w} are trained by minimizing the combined loss L consisting of data and ODE loss.

Thus, the following first two physics residuals are implicitly satisfied when using a PINN with the second-order derivatives.

$$L_{F,\theta_1}(\mathbf{w}) = \frac{1}{N_{col}} \sum_{i=1}^{N_{col}} \left| \theta'_{\mathbf{w},1} \left(t_{col}^{(i)} \right) - \omega_{\mathbf{w},1} \left(t_{col}^{(i)} \right) \right|^2 \quad (3.12)$$

$$L_{F,\theta_2}(\mathbf{w}) = \frac{1}{N_{col}} \sum_{i=1}^{N_{col}} \left| \theta'_{\mathbf{w},2} \left(t_{col}^{(i)} \right) - \omega_{\mathbf{w},2} \left(t_{col}^{(i)} \right) \right|^2 \quad (3.13)$$

$$L_{F,\omega_1}(\mathbf{w}) = \frac{1}{N_{col}} \sum_{i=1}^{N_{col}} \left| \omega'_{\mathbf{w},1} \left(t_{col}^{(i)} \right) - f_1(\theta_{\mathbf{w},1}, \theta_{\mathbf{w},2}, \omega_{\mathbf{w},1}, \omega_{\mathbf{w},2}) \right|^2 \quad (3.14)$$

$$L_{F,\omega_2}(\mathbf{w}) = \frac{1}{N_{col}} \sum_{i=1}^{N_{col}} \left| \omega'_{\mathbf{w},2} \left(t_{col}^{(i)} \right) - f_2(\theta_{\mathbf{w},1}, \theta_{\mathbf{w},2}, \omega_{\mathbf{w},1}, \omega_{\mathbf{w},2}) \right|^2 \quad (3.15)$$

$$L_F(\mathbf{w}) = L_{F,\theta_1}(\mathbf{w}) + L_{F,\theta_2}(\mathbf{w}) + L_{F,\omega_1}(\mathbf{w}) + L_{F,\omega_2}(\mathbf{w}) \quad (3.16)$$

The initial condition and data loss are then given as

$$L_{IC}(\mathbf{w}) = \left(\theta_{1,\mathbf{w}}(t_0) - \theta_1(t_0) \right)^2 + \left(\theta_{2,\mathbf{w}}(t_0) - \theta_2(t_0) \right)^2 + \left(\omega_{1,\mathbf{w}}(t_0) - \omega_1(t_0) \right)^2 + \left(\omega_{2,\mathbf{w}}(t_0) - \omega_2(t_0) \right)^2 \quad (3.17)$$

$$L_{data}(\mathbf{w}) = \frac{1}{N_{data}} \sum_{i=1}^{N_{data}} \left[\left(\theta_{1,\mathbf{w}}(t_i) - \theta_1(t_i) \right)^2 + \left(\theta_{2,\mathbf{w}}(t_i) - \theta_2(t_i) \right)^2 + \left(\omega_{1,\mathbf{w}}(t_i) - \omega_1(t_i) \right)^2 + \left(\omega_{2,\mathbf{w}}(t_i) - \omega_2(t_i) \right)^2 \right] \quad (3.18)$$

Theoretically, the two formulations are equivalent and should yield the same results. However, as shown in the following section, this is not the case in practice where implicitly satisfying the physical residuals leads to an improved convergence.

3.2 Preliminary Experiments

In this section, we perform two preliminary experiments to provide evidence for the use of the second-order ODE formulation and input normalization for all subsequent experiments in this work.

Problem Setup

We use an unweighted vanilla PINN according to the specifications given in Table 3.1. In total, we conduct five runs using different seeds for the initialization of the network weights.

For all following experiments in this work, we set the system parameters of the double pendulum (m_1, m_2, L_1, L_2) equal to one. The initial angles of the two pendulums is set equal to θ_0 with zero initial velocities, i.e.,

$$\mathbf{y}(t_0 = 0) = [\theta_0, \theta_0, 0, 0]^T \quad (3.19)$$

Table 3.1: Specifications of the fully connected network architecture for the preliminary experiments.

Description	Value
Hidden layer	6
Neurons per layer	30
Activation function	swish
Weight initialization	Glorot uniform
Optimizer	Adam
Epochs	25 000
# collocation points	1024
α_{data}	0.5

To evaluate the accuracy of the PINN prediction, a fourth-order RK scheme was employed to provide an approximate reference solution.

3.2.1 Second Order Derivatives

In this experiment, we compare the simulation accuracy of the first-order and second-order PINN architecture (see Fig. 3.1) for an initial angle of $\theta_0 = 90^\circ$. Fig. 3.2 summarizes the results of the PINN prediction (*red line*) along with the RK reference solution (*blue line*). It is evident that the network with the second-order derivatives achieves a significantly lower physics loss of up to one order of magnitude and approximates the correct solution more accurately by visual comparison. For this reason, we use the second-order ODE formulation of the double pendulum for all further experiments.

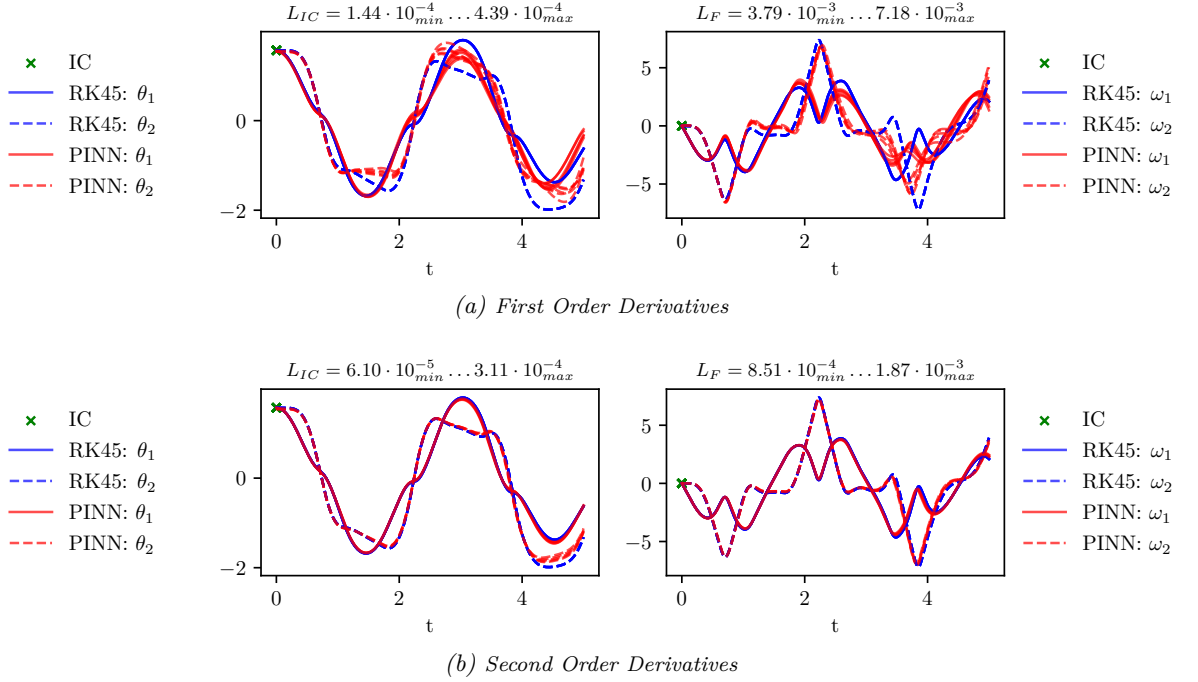


Figure 3.2: Initial condition of $\theta_0 = 90^\circ$ and $\omega_1^{(t_0)} = \omega_2^{(t_0)} = 0$. Prediction of θ_1, θ_2 (left) and ω_1, ω_2 (right) over time t . Comparison between PINN prediction using the set of first-order ODEs and second-order ODEs. In (a) the angular velocities ω_i are trained by minimizing the physics residuals, while in (b) they are obtained by taking the derivative $\omega_i = \theta_i'$.

3.2.2 Feature Scaling

In most practical applications of machine learning algorithms, the input variables are preprocessed in order to achieve more accurate results or speed up computation. For our application of a dynamical system, the PINN takes time t as the only input variable. As the dynamical system is time-invariant, also known as shift-invariant, the output of the neural network should not depend on the absolute time t . Therefore, input normalization to a fixed computational domain also allows reproducible behavior of the network even for time shifts.

Additionally, by normalizing the input, we ensure that the activation function introduces sufficient nonlinearity. The choice of the activation function has a major influence on the performance of the network. In our experiments, the nonlinear activation function was chosen as

the *swish* function depicted in Fig. 3.3 and defined as

$$f(x) = x \cdot \text{sigmoid}(x) = \frac{x}{1 + e^{-x}}. \quad (3.20)$$

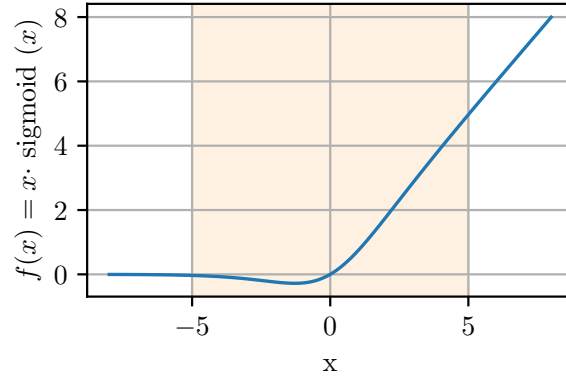


Figure 3.3: Swish activation function. The input variable is normalized to the shaded range $[-5, 5]$.

Fig. 3.4 shows the PINN prediction³ for an initial angle of $\theta_0 = 60^\circ$. While the PINN without input normalization struggles to approximate the high-frequency content of the solution, the PINN with normalization closely corresponds to the reference.

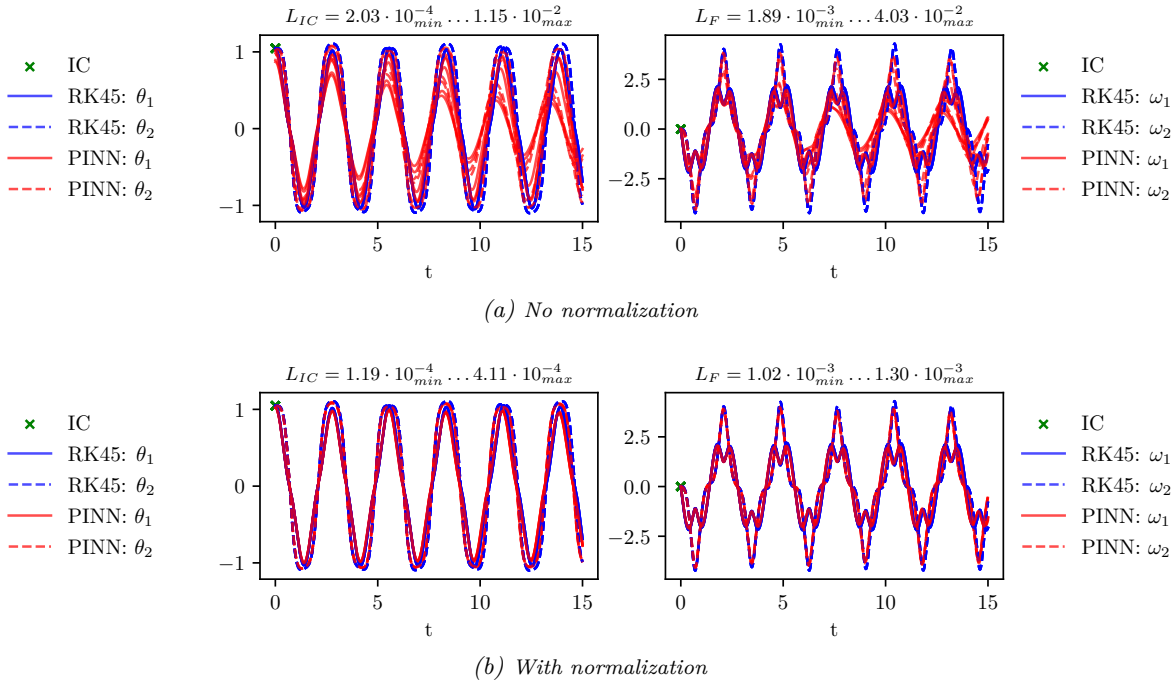


Figure 3.4: $\theta_0 = 60^\circ$. Comparison between PINN prediction without input normalization (a) and with input normalization to $[-5, 5]$ (b).

Thus, unless stated otherwise, the input of the network is normalized to the fixed domain $[-5, 5]$.

³ Using the second-order ODE description.

Comparison to Purely Data-Driven Neural Networks

4.1 Regularization in the Presence of Noisy Data

The performance of purely data-driven NNs is highly dependent on the quality of the training data, as their cost function consists solely of the distance between the predicted trajectory and the data points. If the training data is corrupted by noise, the network will still try to approximate the noisy data points as best as possible, resulting in an overfit model. While the model performs well on training data, it achieves low accuracy when presented with unseen data. A wide-spread method to decrease the level of overfitting is to reduce the network complexity of the model [33]. By reducing the network complexity, its expressive power is reduced as well as the likelihood that the model will overfit noisy data. One way to achieve this is to limit the magnitude of the network parameters, namely its weights and biases. In general, large weights lead to large deviations in the output function given small changes in the input. Thus, by limiting the magnitude of the individual weights, the expressivity of the entire network can be reduced. The value of the weights can be bounded by adding a weight decay rule $-w$ to the update function. Equivalently, we can add a regularization term $\lambda_{L2} \cdot \sum_i w_i^2$ to the cost function [33], known as *L2 regularization*. The L2 penalty term $\sum_i w_i^2$ favors solutions with smaller weight values.

If we recall the cost function of PINNs

$$L(\mathbf{w}) = \alpha_{data} L_{data}(\mathbf{w}) + (1 - \alpha_{data}) L_F(\mathbf{w}) \quad (4.1)$$

we can interpret the physics loss term L_F as a regularization term. Unlike the L2 penalty term, the physics loss carries additional information about the underlying physical system. The physics loss penalizes trajectories that violate the underlying physical constraints. In this way, overfitting of noisy training data points is avoided by a physical consideration.

Problem setup

We consider the double pendulum with all system parameters (m_1, m_2, L_1, L_2) set to one. The initial condition of the pendulum is chosen as follows

$$\mathbf{y}(t_0 = 0) = [\theta_0, \dot{\theta}_0, 0, 0]^T \quad (4.2)$$

without initial angular velocities and with equal deflection of both pendulum arms about the initial angle $\theta_0 = \{30^\circ, 60^\circ, 90^\circ, \dots, 180^\circ\}$. The network parameters are trained with 50 data points corrupted by Gaussian noise with variance $\sigma_n^2 = \{0, 0.1, 0.2, 0.5, 1, 1.5, 2\}$

$$\mathcal{D} = \{\mathbf{y}(t_i) + \eta\}, \quad \eta \sim \mathcal{N}(\mathbf{0}, \sigma_n^2 \mathbb{I}_{4 \times 1}), \quad t_i \in \{0, 4\} \quad (4.3)$$

In this experiment, we compare the performance of a purely data-driven NN, NN with L2 regularization, and a vanilla PINN in terms of robustness to noisy training data. We employ three instances of fully connected feedforward networks with the same specifications given in Table 4.1, but with different seeds for weight initialization.

Table 4.1: Specifications of the fully connected network architecture for training with noisy data.

Description	Value
Hidden layer	6
Neurons per layer	30
Activation function	swish
Weight initialization	Glorot uniform
Optimizer	Adam
Epochs	25 000
# collocation points	1024
λ_{L2}	0.1
α_{data}	0.5

Results

As a first step, we investigate the performance of the data-driven NN, the L2 regularized NN, and the vanilla PINN for an initial angle of 90° and a noise variance of $\sigma_n^2 = 1$. Fig. 4.1 shows the estimated trajectories (*red line*) along with the noisy training data points (*green crosses*) and the noise-free reference trajectory (*blue line*) computed with a RK integration scheme.

The data-driven approach (Fig. 4.1, a) suffers from severe overfitting, especially for the estimation of angular velocities. The predicted curves exhibit sharp transitions between noisy data points. By incorporating an additional L2 penalty term (Fig. 4.1, b) into the cost function, the steep transitions are effectively avoided. However, the predicted trajectories with L2 regularization still show some deviation from the reference solution. In this case, the strength of the physics loss function as a regularization term becomes apparent (Fig. 4.1, c). As a result, the physics loss term penalizes trajectories that do not satisfy the underlying physical constraints. Thus, leading to results that closely agree with the noise-free reference.

To further quantitatively assess the accuracy of the predictions, we consider two measures: First, the mean squared error (MSE) between the estimated trajectory and the reference trajectory. Second, the physics loss of the estimated trajectory, indicating whether the prediction violates the physics of the undamped double pendulum. Fig. 4.2 summarizes these measures for different initial angles on the horizontal axis and different noise variances on the vertical axis.

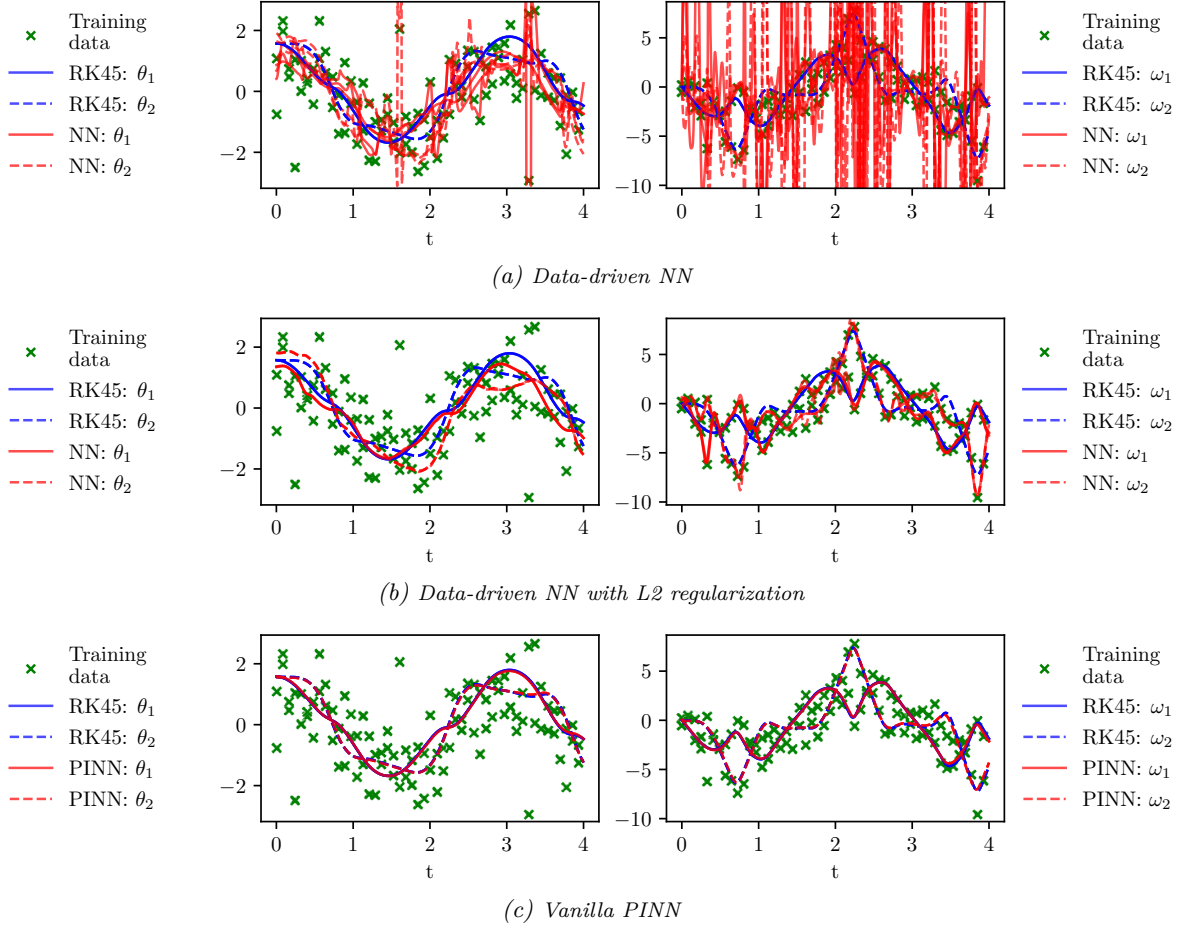
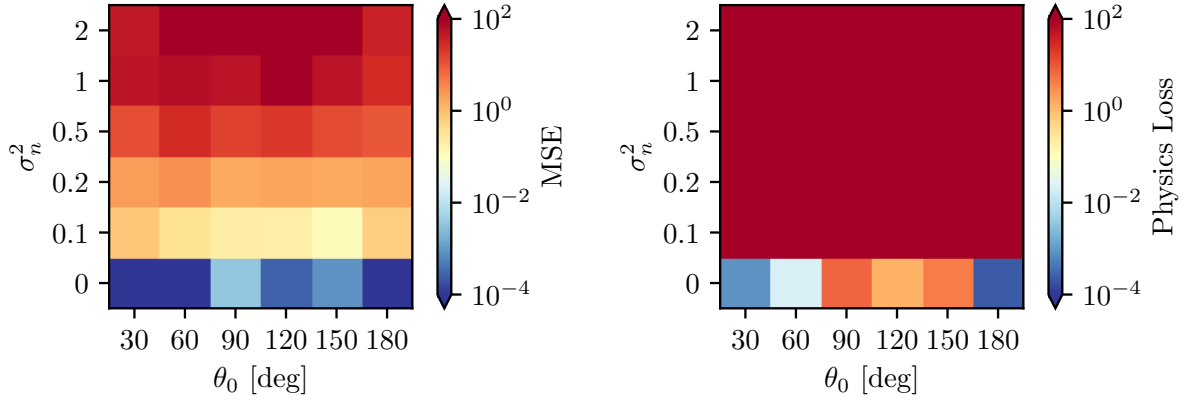


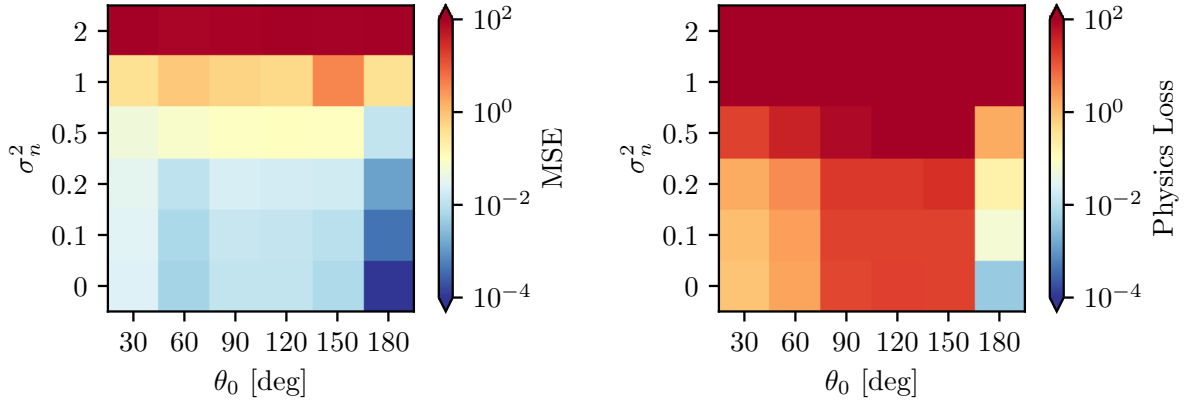
Figure 4.1: Training data corrupted by additive Gaussian noise with $\sigma_n^2 = 1$ for an initial condition of $\theta_0 = 90^\circ$ and $\omega_1^{(t_0)} = \omega_2^{(t_0)} = 0$. Prediction of θ_1, θ_2 (left) and ω_1, ω_2 (right) over time t .

In the noise-free case, the purely data-driven NN indeed achieves a low MSE and a low physics loss. However, any additional noise leads to extreme overfitting associated with sharp transitions of the trajectory which severely violates the physical constraints (Fig. 4.2, a). Limiting the value of the network weights with L2 regularization results in lower prediction accuracy for the noise-free case due to the lack of predictive power. The advantage of this approach becomes immediately apparent when the training data is corrupted by additive noise, considering both MSE and physics loss (Fig. 4.2, b). While the regularization term successfully prevents overfitting, it does not usually guarantee that the physical constraints are satisfied. A remedy to this problem is the use of a physics informed regularization term, given by the PINN framework. Even for extremely noisy training data, the vanilla PINN is typically able converge to a good local minimum where the predictions agree with the physical laws (Fig. 4.2, c).

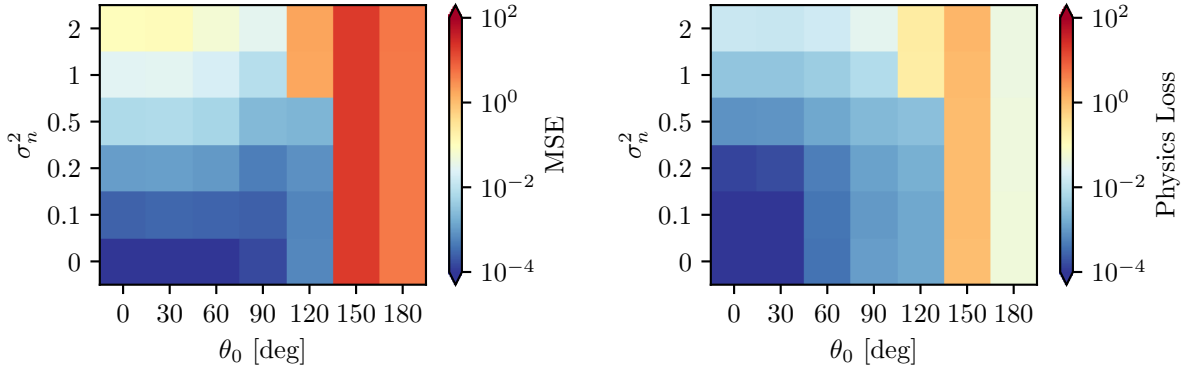
A striking effect is the dependence of the PINN accuracy on the initial angle. For small initial angles, the trajectories are well approximated, leading to small MSE and physics loss values. However, for $\theta_0 = 150^\circ$, the vanilla PINN is not able to converge to a solution close to the reference. Interestingly, even for the noise-free case ($\sigma_n^2 = 0$), the PINN shows significantly higher MSE values compared to the data-driven NN, while the physics loss is lower. Although the PINN is learning a physically plausible solution, it deviates heavily from the reference. This behavior and the strong dependency on initial conditions reappear in subsequent experiments and are discussed in more detail in a separate chapter.



(a) Data-driven NN



(b) Data-driven NN with L2 regularization



(c) Vanilla PINN

Figure 4.2: Performance of NNs and PINNs for varying initial angles θ_0 and noise variance σ_n^2 . The MSE between the predicted and the reference solution is shown in the left column and the physics loss in the right column.

4.2 Reconstruction of Dynamics from Incomplete Data

As shown in the previous experiment, purely data-driven NNs rely on the quality of available training data. Quality, in that sense, does not only refer to the correctness of the data in terms of noise, but also to the number and distribution of data points. Data-driven NNs are trained to fit the given data - if there is no data in a certain domain, the model does not have any restrictions on the trajectory within that interval. Hence, the prediction relies on the network's ability to interpolate within that interval, which for a certain extent becomes highly imprecise. Again, this is a prime use case for physics-based deep learning. A system of first-order continuous ODEs is uniquely determined given the initial (or boundary) conditions. In PINNs, the ability to interpolate between data points can be further supported by means of the physics loss function and a suitable choice of collocation points. With the collocation points also sampled inside the missing interval, PINNs are able to fill the gap and converge to the correct trajectory, even for very sparse training data.

Problem setup

The double pendulum is initialized with its system parameters (m_1, m_2, L_1, L_2) set to one and an initial condition of

$$\mathbf{y}(t_0 = 0) = [\theta_0, \dot{\theta}_0, 0, 0]^T \quad (4.4)$$

with $\theta_0 = \{30^\circ, 60^\circ, 90^\circ, \dots, 180^\circ\}$. The network parameters are trained with 50 data points uniformly distributed over the computational domain but missing in an interval

$$t_{data} \in [0, 8] \setminus [3, t_{miss}]$$

with $t_{miss} = \{3.5, 4, 5, 6, 7\}$. Again, we compare the prediction accuracy of the data-driven NN and PINN with respect to the MSE between the predicted and reference trajectories and the physics loss. We train three PINN instances with the same initial condition and network specifications, but different seeds for the weight initialization. All relevant model specifications for the following experiment are summarized in Table 4.2.

Table 4.2: Specifications of the fully connected network architecture for training with incomplete data.

Description	Value
Hidden layer	6
Neurons per layer	30
Activation function	swish
Weight initialization	Glorot uniform
Optimizer	Adam
Epochs	25 000
# collocation points	1024
α_{data}	0.5, 0.98

Results

First, we examine the results for both the purely data-driven and physically informed approaches for an initial angle of 120° and missing data between $t = 3$ and $t = 6$. The predicted trajectories for the three cases are shown in Fig. 4.3. Evidently, the purely data-driven NN approximates the reference well in domains with data points, but fails completely in the domain without training data. In this region, the predicted trajectory can take arbitrary values without affecting the cost function.

Thus, the physics loss term is included in the cost function to avoid violations of the physical constraints. However, using a vanilla PINN with equal weights for the data and physical loss, we observe a large discrepancy between the predicted and reference trajectories. In fact, the physical loss term with weight $(1 - \alpha_{data}) = 0.5$ hinders convergence to a correct solution, even in the domain with training data. Reducing the weight of the physical loss term to $(1 - \alpha_{data}) = 0.02$ avoids the previous convergence problems. Now, both domains with and without data points are predicted correctly, satisfying the physical constraints and agreeing well with the reference. Although the individual terms of the multi-objective cost function should not be conflicting, this experiment shows the importance of balancing the loss terms with an appropriate choice of α_{data} .

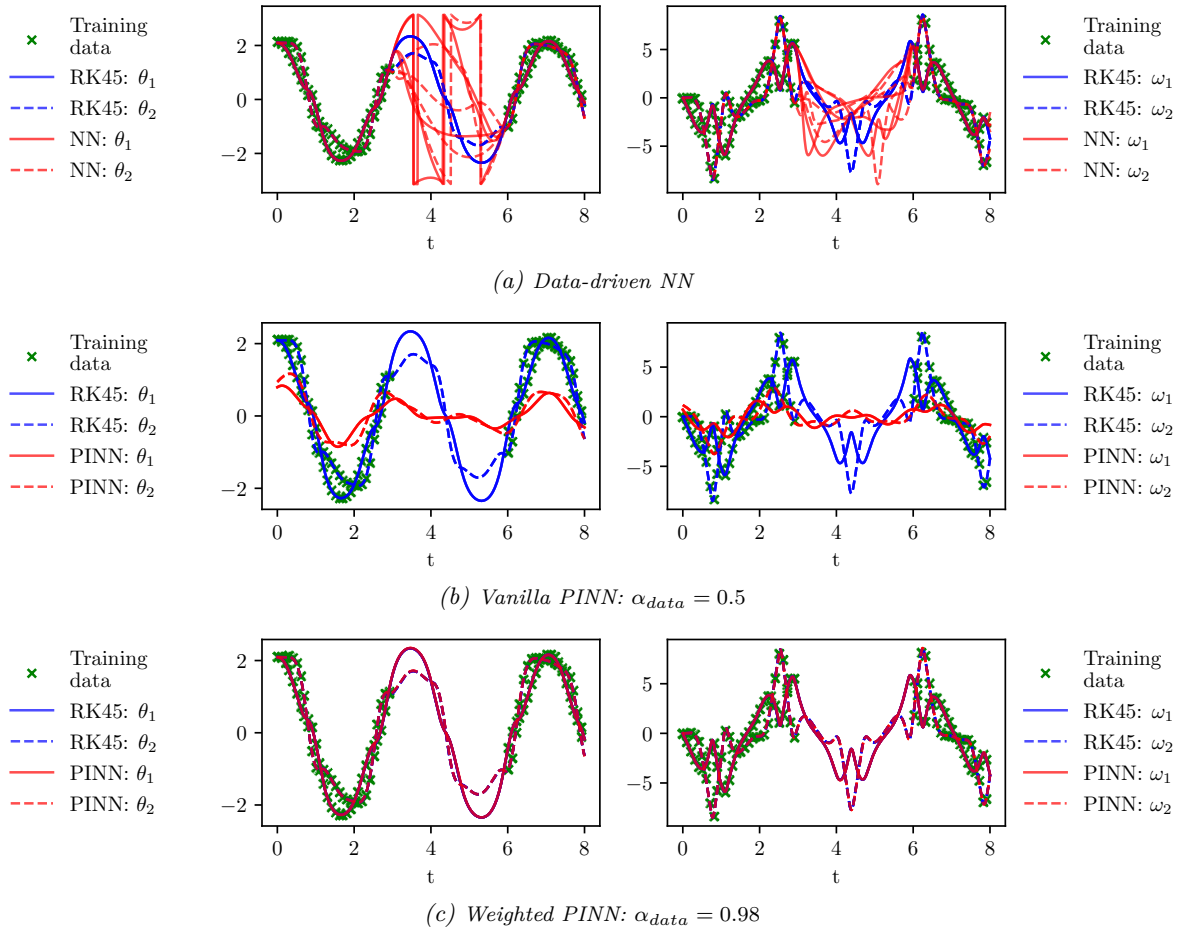


Figure 4.3: Training data is missing in an interval between 3 and 6 for an initial condition of $\theta_0 = 120^\circ$ and $\omega_1^{(t_0)} = \omega_2^{(t_0)} = 0$. Prediction of θ_1, θ_2 (left) and ω_1, ω_2 (right) over time t .

Again, the experiment is repeated for different initial angles and range of the missing data domain. The results with respect to MSE and physics loss are summarized in Fig. 4.4. As expected, the purely data-driven approach is unable to estimate the correct behavior of the double pendulum in the interval of missing data. Both MSE and physics loss increase with an increased missing data domain and larger initial angles. Including the physics loss term immediately improves the overall performance of the model. In particular, for small initial angles, the PINN accurately predicts the trajectory over large domains of missing data. Nevertheless, we also observe in this experiment that the PINN suffers from strong convergence problems, for the same initial conditions as in the previous experiment for noisy data.

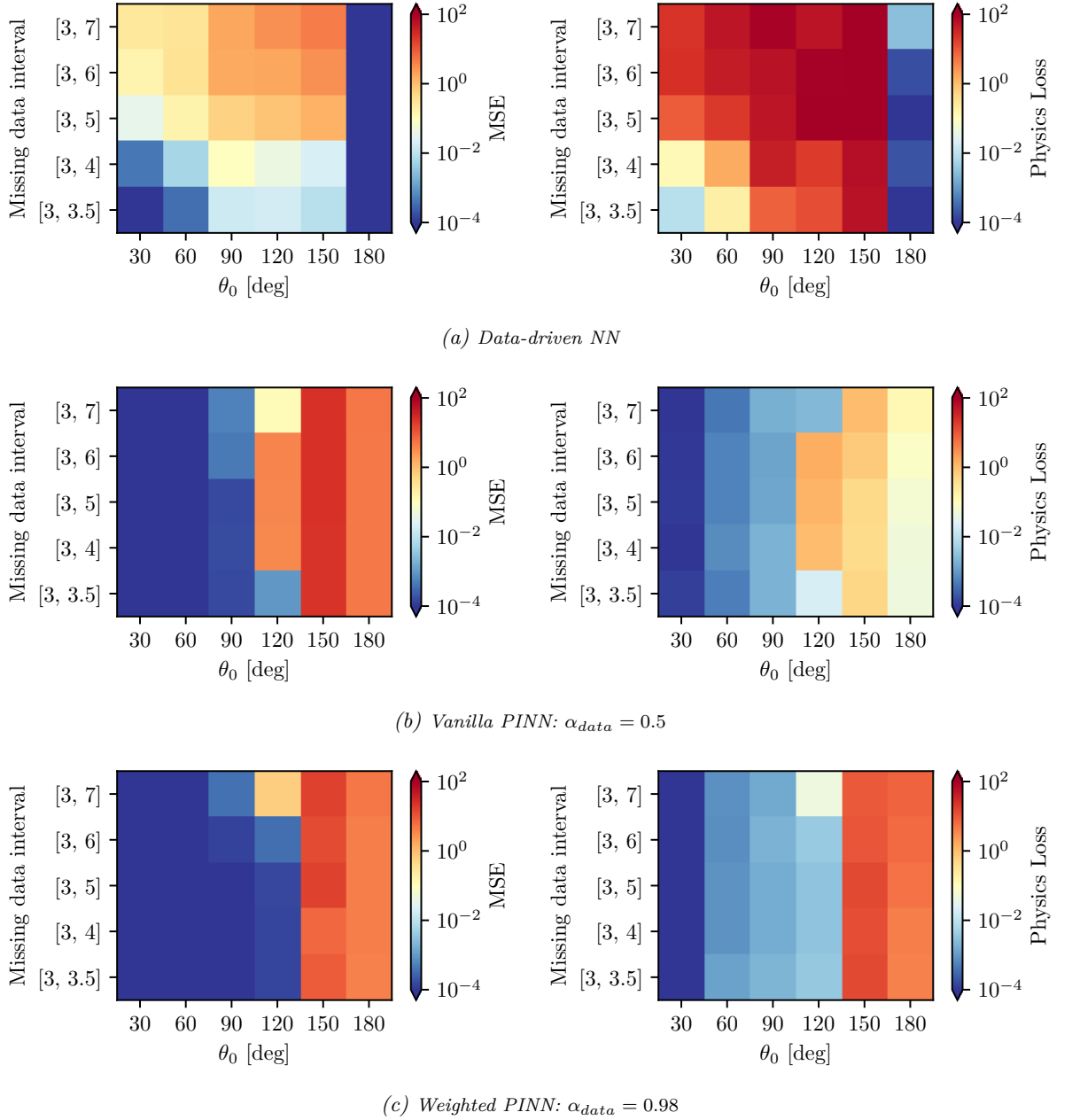


Figure 4.4: Performance of data-driven NNs and PINNs for varying initial angles θ_0 and interval length without training data points. The MSE between the predicted and the reference solution is shown in the left column and the physics loss in the right column.

A notable difference in accuracy is visible for an initial angle of 180° , where the purely data-driven NN achieves highly accurate results compared to the PINN. In this case, the double pendulum is located in the unstable fixed point where both pendulums are in an vertical position exhibiting no movement. There, the NN simply returns a constant value in the missing data domain and achieves good results. The PINN, however, is not able to converge to the correct solution. In some cases, a simple modification of the loss weighting scheme helps to mitigate the convergence problems of the unweighted PINN. However, this technique is not a sufficient solution for all types of convergence problems. The prediction for an initial angle of 150° and 180° deviates significantly from the reference trajectory for both the weighted and unweighted cost function.

4.3 Reconstruction of Dynamics from Partially Observed Data

In many real-world applications, we encounter the problem of partially observed data, as it is often not possible to measure all states of a physical system. Such scenarios frequently occur in engineering and biological systems. In fluid dynamics, for example, experimental measurements in the form of flow visualizations do exist (e.g., transport of paint or smoke), but it is a challenge to infer information about the hidden states (e.g., flow velocity, pressure, temperature) using conventional numerical methods [9]–[11]. In such cases, PINNs can be employed to enforce solutions that satisfy the underlying physical constraints and from that, information about hidden states can be inferred. In the following experiment, we demonstrate the performance of PINNs for reconstructing the hidden dynamics of the double pendulum, where the full system dynamics is inferred from only partially observed data.

Problem setup

The double pendulum is initialized with an initial condition of

$$\mathbf{y}(t_0 = 0) = [\theta_0, \dot{\theta}_0, 0, 0]^T \quad (4.5)$$

with $\theta_0 = \{30^\circ, 60^\circ, 90^\circ, \dots, 180^\circ\}$. Again, the underlying system of ODEs is known, but the training data is incomplete. The angular position and velocity of one pendulum is observable, while the other one is hidden. The aim is to infer the trajectory of the hidden state variables from 50 partial measurement points uniformly distributed over the computational domain $t \in [0, t_{max}]$ with $t_{max} = \{1, 2, 3, 5, 10, 15\}$. The problem is ill-posed as the initial condition is unknown during training. Let the measurement data be (θ_j, ω_j) , with $j = \{1, 2\}$. Thus, the cost function of the PINN is given as

$$L(\mathbf{w}) = \alpha_{data} L_{par}(\mathbf{w}) + (1 - \alpha_{data}) L_F(\mathbf{w}) \quad (4.6)$$

$$L_{par}(\mathbf{w}) = \frac{1}{N_{data}} \sum_{i=1}^{N_{data}} \left[\left(\theta_{j,\mathbf{w}}(t_i) - \theta_j(t_i) \right)^2 + \left(\omega_{j,\mathbf{w}}(t_i) - \omega_j(t_i) \right)^2 \right], \quad j = \{1, 2\}. \quad (4.7)$$

We train three PINN instances with the same initial conditions and network specifications, but different seeds for weight initialization. All relevant model specifications for the following experiment are summarized in Table 4.3.

Table 4.3: Specifications of the fully connected network architecture for training with partially observed data.

Description	Value
Hidden layer	6
Neurons per layer	30
Activation function	swish
Weight initialization	Glorot uniform
Optimizer	Adam
Epochs	25 000
# collocation points	1024
α_{data}	0.99

Results

Fig. 4.5 shows the results of the experiment for an initial angle of 90° , inferring the motion of the first (a) and second pendulum (b). The prediction of the hidden states relies solely on the physics loss term and the measured data of the other pendulum. Even for an ill-posed problem with incomplete training data, the prediction of the inferred hidden states agrees well with the reference. For an increased initial angle of 150° , however, the PINN is not able to converge to an accurate solution. As shown in Fig. 4.6, the prediction of the hidden state deviates considerably from the reference. Moreover, the PINN fails to sufficiently approximate the given observational data as well. From these results it is evident that the prediction accuracy again depends heavily on the initial conditions. Fig. 4.7 summarizes the performance of PINN in inferring the hidden states. The MSE and the physics loss of the PINN prediction are displayed with respect to different initial conditions and length of the computational domain. We can clearly observe that a low physics loss correlates with a low resulting MSE. Overall, the results indicate that it is possible to successfully reconstruct the hidden states for different initial conditions resulting in both periodic and chaotic behavior. Nevertheless, the length of the computational domain must be chosen appropriately to achieve convergence to a solution that minimizes the sum of data and physics loss. Initial angles leading to harmonic oscillations, in this case 30° and 60° , allow an accurate prediction of the observed hidden states over large domains. If, however, the trajectory exhibits chaotic behavior, an accurate prediction is only possible for small domains. Otherwise, the PINN suffers from strong convergence issues and gets stuck in poor local minima with high MSE and physical loss. This observation is consistent with the results of the experiments in [15] for a simple dynamical system with two stable and one unstable solution. In this paper, the authors showed that a larger computational domain leads to a loss landscape with higher complexity, which is generally more difficult to optimize.

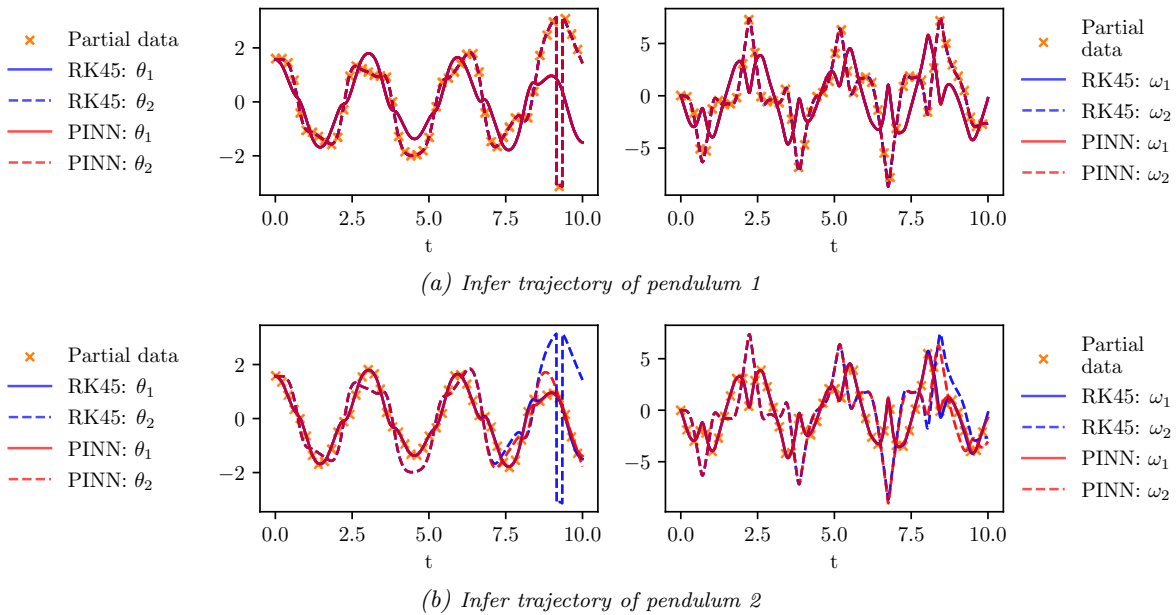


Figure 4.5: Prediction of the motion of the hidden pendulum using partial training data. The problem is ill-posed, the initial angle of $\theta_0 = 90^\circ$ is unknown to the PINN during training. The PINN is able to accurately reconstruct the motion of the hidden pendulum.

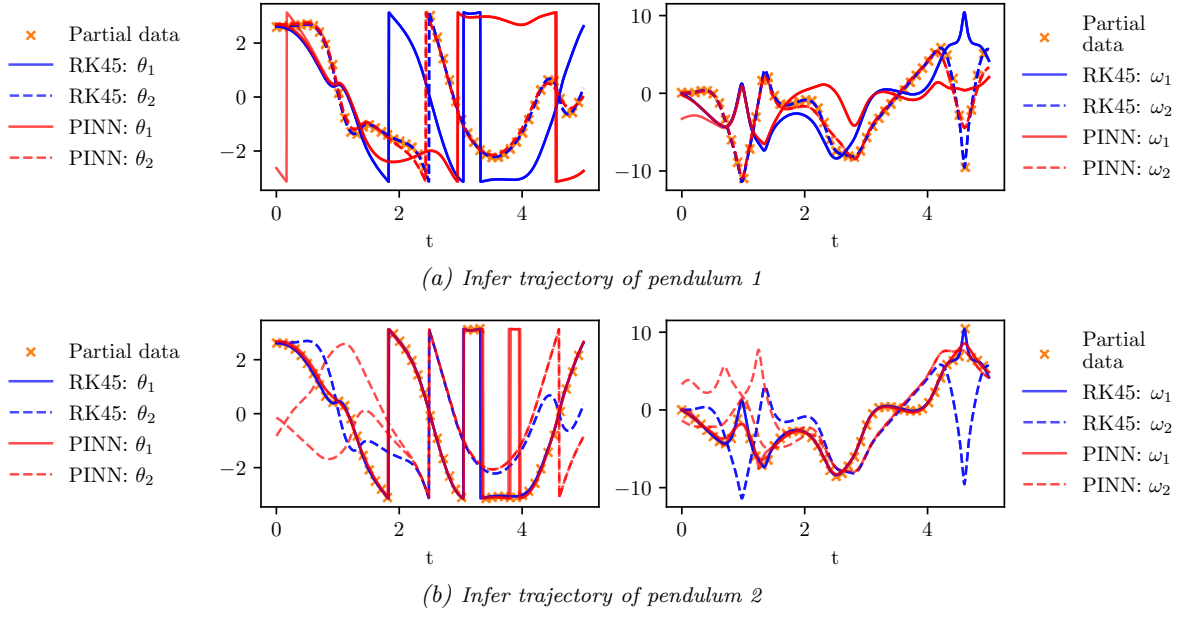


Figure 4.6: Prediction of the motion of the hidden pendulum using partial training data. The problem is ill-posed, the initial angle of $\theta_0 = 150^\circ$ is unknown to the PINN during training. The PINN suffers from strong convergence issues and is not able to predict the observable and the hidden state accurately.

Interestingly, in the chaotic regime of the double pendulum in particular, the prediction of the PINN is more accurate when the second pendulum is observable (Fig. 4.7, a) compared to the opposite case (Fig. 4.7, b). We suspect that this effect is related to the increased complexity of the motion of the second pendulum. While the first pendulum is connected on one end to a fixed point, the second pendulum can swing freely. Thus, the optimization problem becomes more complex when the motion of the second pendulum has to be inferred solely from physics without the help of training data.

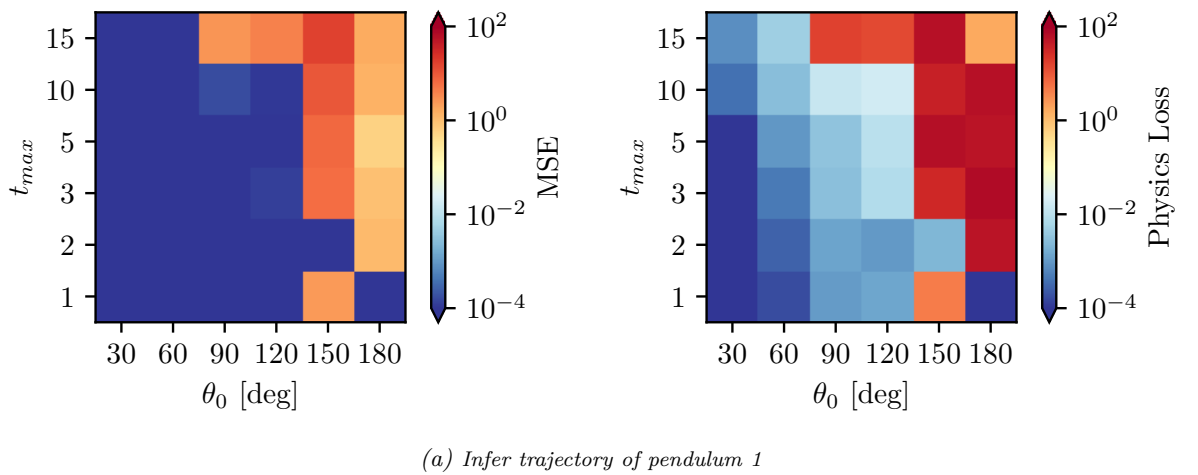
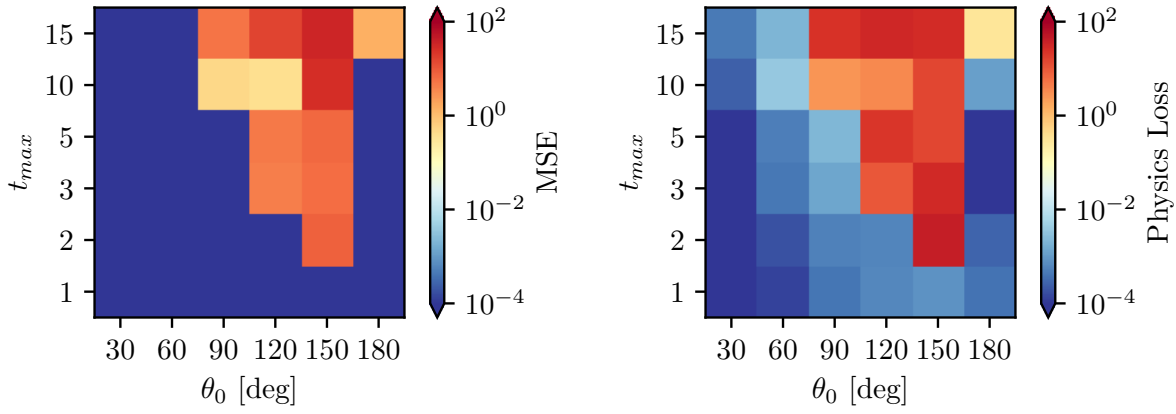


Figure 4.7: Performance of PINNs for varying initial angles θ_0 and lengths of the computational domain given training data about one pendulum and inferring the solution of the other pendulum. The MSE between the predicted and the reference solution is shown in the left column and the physics loss in the right column.



(b) Infer trajectory of pendulum 2

Figure 4.7: Performance of PINNs for varying initial angles θ_0 and lengths of the computational domain given training data about one pendulum and inferring the solution of the other pendulum. The MSE between the predicted and the reference solution is shown in the left column and the physics loss in the right column.

4.4 Discussion

We have shown for several problems involving differential equations with low-quality or incomplete training data that PINNs can immensely improve a purely data-driven deep learning approach.

First, the physics loss acts as an informed regularization term to avoid overfitting noisy data. Additionally, the physics loss penalizes trajectories that do not satisfy the physical constraints in regions with no training data. By providing additional information about the underlying physical system, the prediction accuracy can be significantly increased in both cases. However, it is important to choose a suitable loss weighting scheme when minimizing the multi-objective cost function. In Section 4.2, we showed that the unweighted PINN suffered from convergence problems that could be resolved by manually adjusting the loss weights. In a last experiment, we demonstrated that hidden states can be inferred even for ill-posed problems without knowledge of the initial conditions (or boundary conditions).

However, in all three experiments we observed a strong dependence of the prediction accuracy on the initial angle of the pendulum. In particular, for an initial angle of 150° , the PINN exhibits strong convergence problems caused by the additional physics loss term, even in the noise-free case. The prediction of the harmonic oscillations, on the other hand, was successful in all three experiments without significant convergence issues.

Forward Problems of Nonlinear Ordinary Differential Equations

5.1 From Periodicity to Chaos

After successfully showcasing the performance of PINNs for applications with noisy or incomplete data, we now employ them for solving forward ODEs problems. In forward problems, the solution of the ODE system is uniquely determined by the initial or boundary conditions. Thus, by incorporating both the distance from the given initial condition and the physics residuals of the underlying system, PINNs can be used to predict the solution trajectory of the system. However, this setting is known to frequently display strong convergence issues for various systems [13]–[15], [18], [21], [22], [34].

The aim of this chapter is to evaluate the performance of PINNs for simulating a dynamical system that exhibits both harmonic and chaotic behavior. We highlight the difficulty of training and achieving convergence to the correct solution with respect to different initial conditions and computational domains. Convergence issues are analyzed in detail in Chapter 6, where we discuss techniques to improve convergence and possible explanations for the PINN failure modes.

Problem setup

In these experiments we analyze the accuracy of predicted trajectories given the initial condition and precise knowledge of the underlying ODEs, without any additional training data. Specifically, we look at the influence of the initial condition on the convergence of the PINN to the correct solution. Given is the initial condition

$$\mathbf{y}(t_0 = 0) = [\theta_0, \theta_0, 0, 0]^T \quad (5.1)$$

with θ_0 being the initial angle of both pendulums. The solution is estimated over a computational domain $t \in [0, t_{max}]$.

Both initial condition and physics loss are incorporated in the multi-objective cost function in a soft manner and minimized simultaneously. For the following experiments, the two loss terms are weighted equally

$$L(\mathbf{w}) = \alpha_{data} L_{IC}(\mathbf{w}) + (1 - \alpha_{data}) L_F(\mathbf{w}) \quad \alpha_{data} = 0.5 . \quad (5.2)$$

We train five PINN instances with the same initial values and specifications but different seeds for the weight initialization. A summary of the network specifications for the following experiments is given in Table 5.1.

Table 5.1: Specifications of the fully connected network architecture for a forward problem, given the initial condition and ODE.

Description	Value
Hidden layer	6
Neurons per layer	30
Activation function	swish
# Collocation points	1024
Weight initialization	Glorot uniform
Optimizer	Adam
Epochs	25 000
# collocation points	1024
α_{data}	0.5

Results: Harmonic Oscillations

In this first experiment, we consider an initial angle of 60° , resulting in a harmonic oscillation. Fig. 5.1 shows the result of training the network with the physics informed cost function, including the minimum and maximum initial condition loss and physics loss for the runs with different weight initialization. No additional training data, except the initial condition, is used for the optimization of the model weights.

For a computational domain with length $t_{max} = 5$ and $t_{max} = 15$, the network is able to correctly reproduce the reference solution given by RK. When increasing the computational domain further, the PINN fails to converge to the correct minimum and the trajectory slowly fades to the trivial zero solution⁴. As we consider an undamped double pendulum, the total energy of the system should remain constant for all times. The PINN prediction for large domains violates this constraint and results in a physics loss that is one order of magnitude higher compared to the case of $t_{max} = 15$.

Results: Chaotic Motion

Next, we increase the initial angle to values that result in chaotic motion of the two pendulums. The results are summarized in Fig. 5.2. The predicted trajectories for the initial angle of 90° agree well with the reference trajectory computed with RK. However, when further increasing the initial angle, the PINN does not converge to the correct solution. While the initial condition is not fulfilled as accurately anymore, the resulting physics loss is still as low as in the cases where convergence was successful. Especially for an initial angle of 150° , the loss values seem reasonable, but the PINN prediction deviates heavily from the RK reference.

⁴ An increased computational domain corresponds to an increased function frequency of the solution due to performing normalization of the input variable to a fixed range.

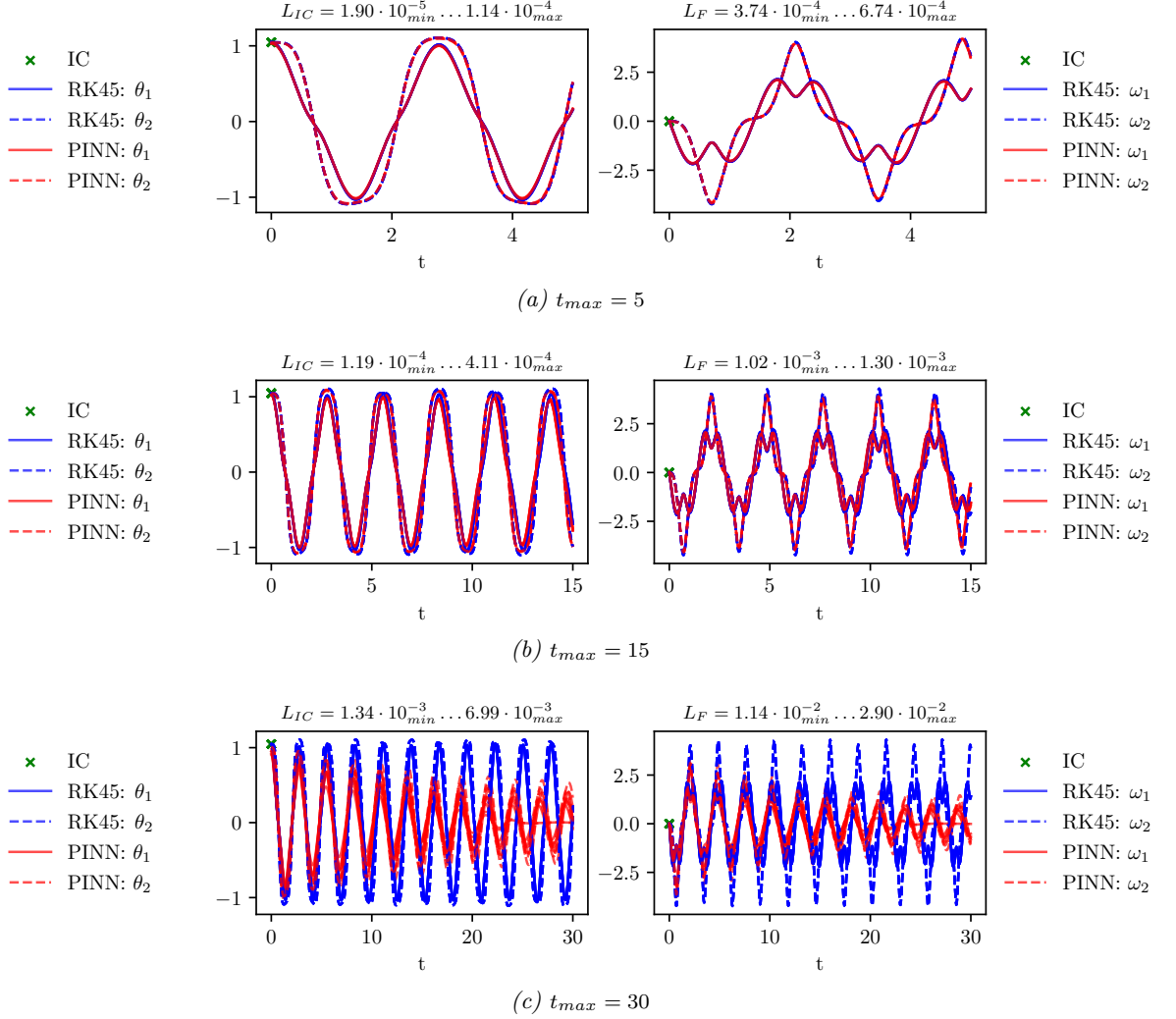


Figure 5.1: Vanilla PINN: Prediction of the solution trajectory for an initial condition of $\theta_0 = 60^\circ$ and precise knowledge about the underlying physics. As the computational domain is increased, the prediction accuracy and the resulting physics loss are observed.

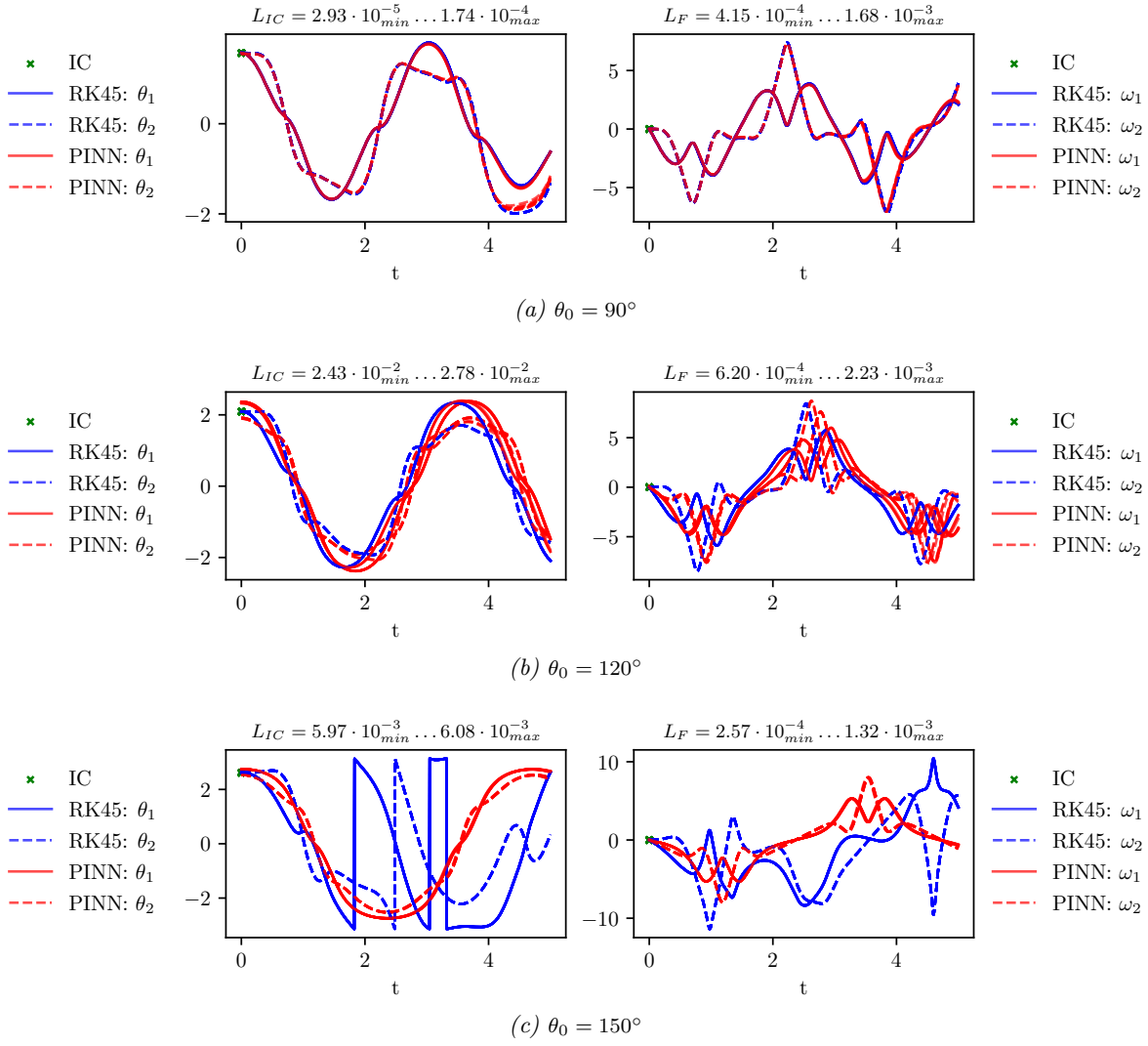


Figure 5.2: Vanilla PINN: Prediction of the solution trajectory given only the initial condition and precise knowledge about the underlying physics. The physics and initial condition loss is weighted equally ($\alpha_{data} = 0.5$). The PINN suffers from convergence issues for an increased initial angle, leading to strong deviations between prediction and reference.

Runge-Kutta Continuation

In order to assess whether the predicted trajectories indeed do correspond to physically correct solutions, we run RK with the corresponding estimation at time step $t_0 = t_{cont}$ for each PINN instance.

Let $\mathbf{y}_w(t)$ denote the prediction of the PINN. We use the PINN prediction at time $\mathbf{y}_w(t = t_{cont})$ as a new initial condition \mathbf{y}_0 for the RK integration scheme. The results are depicted in Fig. 5.3. At the start, the RK solution follows the PINN prediction exactly. However, due to the chaotic nature of the pendulum, the RK trajectories begin to diverge due to small differences in the predicted initial conditions of the PINN. If the RK continuation is started at a later time step, just before the trajectory starts to diverge, there is again good agreement between the two methods.

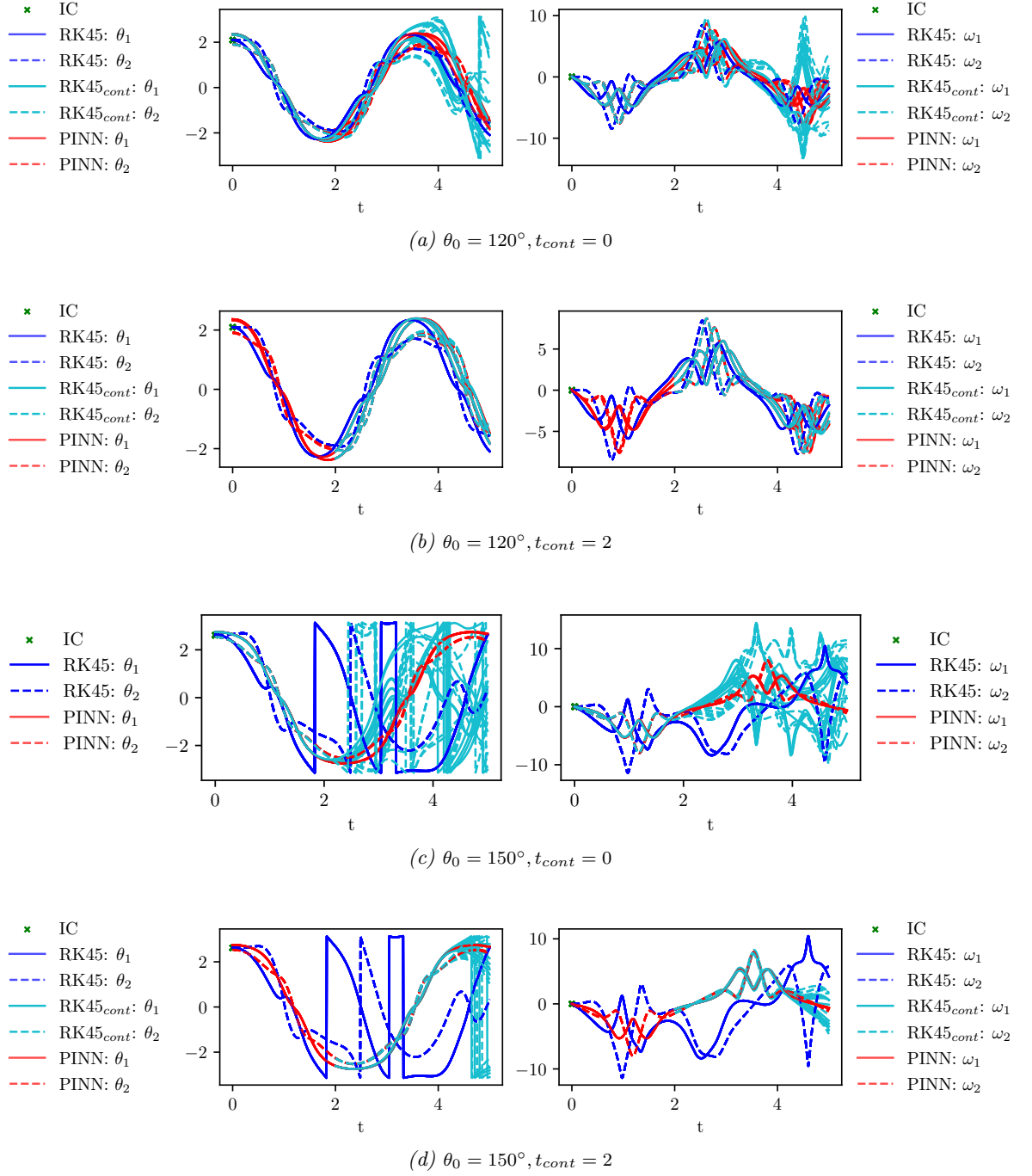


Figure 5.3: The initial condition of the PINN predictions used to estimate the trajectory with RK to verify accuracy with respect to the physical constraints.

Attractive Solution

The chaotic behavior of the double pendulum is characterized by high sensitivity to the initial condition. Minor changes of the initial condition may result in significantly diverging trajectories. In this regime, we have observed major convergence issues, especially for initial angles of 120° and 150° . There, the PINN tends to converge to solutions that do not exactly fit the given initial condition, but still satisfy the underlying physics. Thus, we want to analyze the PINN prediction for small changes in the initial condition and observe whether it converges to the same incorrect solution. We train one initialization of a vanilla PINN with the same seed for the weight initialization for increasing initial angles between $\theta_0 = \{116^\circ, 124^\circ, \dots, 156^\circ\}$ (Fig. 5.4 (a)) and $\theta_0 = \{141^\circ, 142^\circ, \dots, 156^\circ\}$ (Fig. 5.4 (b)).

As expected, the RK solutions are highly sensitivity to minor variations of the initial condition and quickly diverge, especially for 150° . While the RK solutions showcase no correlation after $t = 2$, the vanilla PINN consistently converges to a similar, incorrect trajectory. In both cases, the PINN prediction results in low physics losses, accepting a slight increase in initial condition loss. These results indicate that the PINN solution corresponds to a highly attractive local minimum of the multi-objective cost function, that is not sensitive to minor deviations from the initial condition.

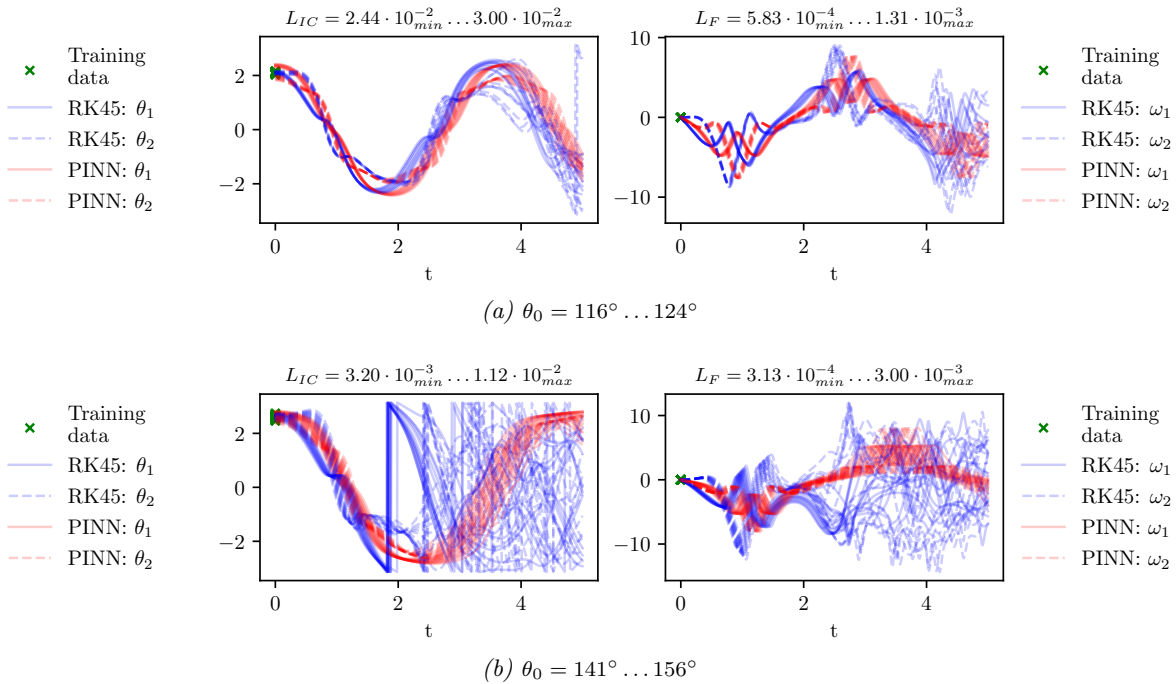


Figure 5.4: The RK reference trajectory shows strong variations due to the chaotic nature of the double pendulum at these initial angles. In contrast, the PINN converges to a similar incorrect solutions for all different initial conditions, indicating a highly attractive minimum in the multi-objective cost function.

5.2 Discussion

In this chapter, we have showcased the ability of the PINN to accurately predict the trajectory of the double pendulum by minimizing both initial condition and physics loss in a soft manner. However, finding a good minimum of this multi-objective cost function via gradient descent is not guaranteed. The loss landscape is highly non-convex and depends both on the initial condition and system parameters (e.g. computational domain) [14], [15]. In our experiments, we have observed two kinds of convergence issues, depending if the correct solution lies in the periodic or chaotic regime of the nonlinear dynamical system.

First, the PINN was used to predict a periodic function with increasing computational domain. For the low frequency cases ($t_{max} = \{5, 15\}$), the PINN was able to quickly converge to the accurate solution. Continuing to increase the computational domain led to trajectories that exhibited damped behavior and could not model the high-frequency elements of the correct solution.

Next, we tested the ability of PINN to predict trajectories with chaotic motion. In this regime, minor changes of the initial condition resulted in significant changes of the resulting trajectories. Here, simultaneous optimization of data and physics losses in a soft manner led to solutions that violated the initial condition slightly but still returned an overall low physics loss. Although the predictions corresponded to physically correct trajectories, they deviated strongly from the reference solution due to the small changes in the initial condition. Additionally, these solutions are shown to be highly attractive local minima that persist even for changes of the initial condition.

Chapter 6 is concerned with analyzing said convergence issues and common methods to mitigate them. As frequently reported in literature [15], [19], [34], [35], we will observe that both loss weighting scheme and length of the computational domain play an important role. Additionally, we will confirm the results of [15] that under certain circumstances the global minimum of the multi-objective cost function does not conform to the correct solution for the underlying problem.

Convergence Issues and Failure Modes

In the previous chapter, we have analyzed the performance of PINNs for simulating a dynamical system in the harmonic and chaotic regime. In summary, we have distinguished two types of convergence problems:

- The first failure mode is observed for the simulation of harmonic oscillations. Convergence to a correct solution is successful for a limited size of the computational domain. Increasing the computational domain, however, results in a solution that slowly fades to the trivial zero solution. The predicted trajectory agrees well with the initial condition, but violates the physical constraints because the total energy of the system does not remain constant, but gradually decreases.
- The second problem occurs for solutions with chaotic motion. Although the PINN is able to converge to a solution that yields both a low initial condition and a low physics loss, the trajectory differs strongly from the reference. In such cases, it seems to be advantageous to not satisfy the initial condition perfectly and thereby provide a solution that is less complex than the reference, but still satisfies the underlying physics.

For a detailed examination of the failure modes, we consider the initial angles of $\theta_0 = 60^\circ$ and $\theta_0 = 150^\circ$ for the first and second failure mode, respectively. Our experiments highlight that the crucial element that determines whether the PINN is able to converge to the correct solution is the physical loss function. We demonstrate that the ODE residuals of the double pendulum are not a suitable minimization objective for the simulation of systems with chaotic behavior. In the chaotic domain, even small perturbations of the initial conditions can lead to strongly diverging solutions. The PINN, on the other hand, does not reflect this extreme sensitivity to initial conditions and instead consistently converges to certain highly attractive solutions. Lastly, we confirm that reducing the size of the computational domain is an appropriate remedy for solving these convergence problems, especially in the chaotic domain.

Problem setup

Following the original PINN framework proposed by [2] and used in most research on PINNs, the physics residuals are integrated into the loss function in a soft manner as a regularization term. The introduction of an additional loss weighting parameter α_{data} has been shown to assist convergence by preventing that one of the losses stalls during training [14].

$$L(\mathbf{w}) = \alpha_{data}L_{IC}(\mathbf{w}) + (1 - \alpha_{data})L_F(\mathbf{w}) . \quad (6.1)$$

Ideally, successful convergence to a correct solution would yield zero for both the data and physics loss. Therefore, we propose the unweighted sum of the two objectives as a measure of how accurate the predicted solution is.

$$L(\mathbf{w}) = L_{IC}(\mathbf{w}) + L_F(\mathbf{w}) . \quad (6.2)$$

For all the following experiments, we consider the double pendulum with the initial condition

$$\mathbf{y}(t_0 = 0) = [\theta_0, \dot{\theta}_0, 0, 0]^T \quad (6.3)$$

with an angle of $\theta_0 = \{60^\circ, 150^\circ\}$ resulting in harmonic oscillation and chaotic motion, respectively.

All neural networks of the subsequent experiments are initialized according to the specifications given in Table 6.1.

Table 6.1: Specifications of the fully connected network architecture for the analysis of convergence issues.

Description	Value
Hidden layer	6
Neurons per layer	30
Activation function	swish
Weight initialization	Glorot uniform
# collocation points	1024
Optimizer	Adam
Epochs	25 000

6.1 Insufficient Expressivity

In this first experiment, we aim to determine whether the convergence problems occur due to a lack of expressivity of the neural network. Insufficient capacity of the network to accurately approximate the true solution would lead to violations of the physical constraints. In such cases, it may be beneficial for the PINN to converge to a solution that is less complex and easier to learn.

We consider the two convergence problems of the vanilla PINN for the harmonic ($\theta_0 = 60^\circ$) and chaotic cases ($\theta_0 = 150^\circ$). In addition, we employ a purely data-driven NN with the same network specifications as the PINN (see Table 6.1), trained with 1000 data points generated from the corresponding RK solution.

Results

Fig. 6.1 summarizes the prediction of the vanilla PINN and illustrates its convergence problems for harmonic oscillations ($\theta_0 = 60^\circ$) with large computational domains and for solutions with chaotic behavior ($\theta_0 = 150^\circ$). For comparison, it is not difficult to approximate both solutions with a purely data-driven NN using training data generated from the RK reference solution, see Fig. 6.2.

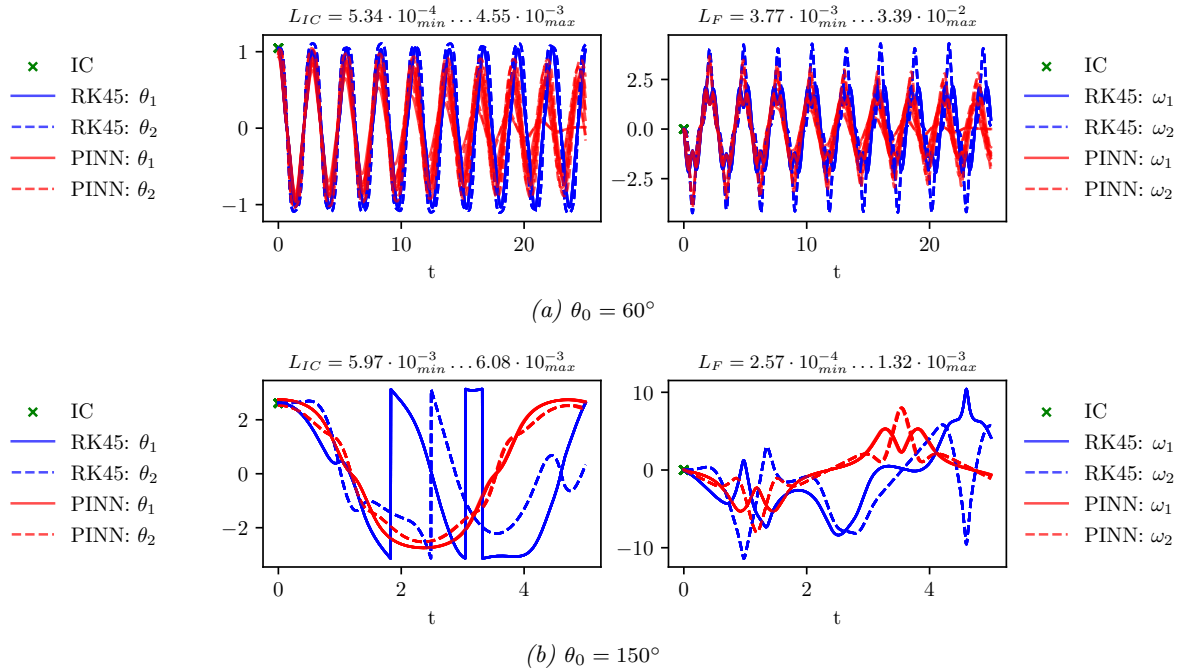


Figure 6.1: Vanilla PINN, $\alpha_{data} = 0.5$. The vanilla PINN exhibits convergence issues for harmonic oscillations over large domains (a) and chaotic motion (b).

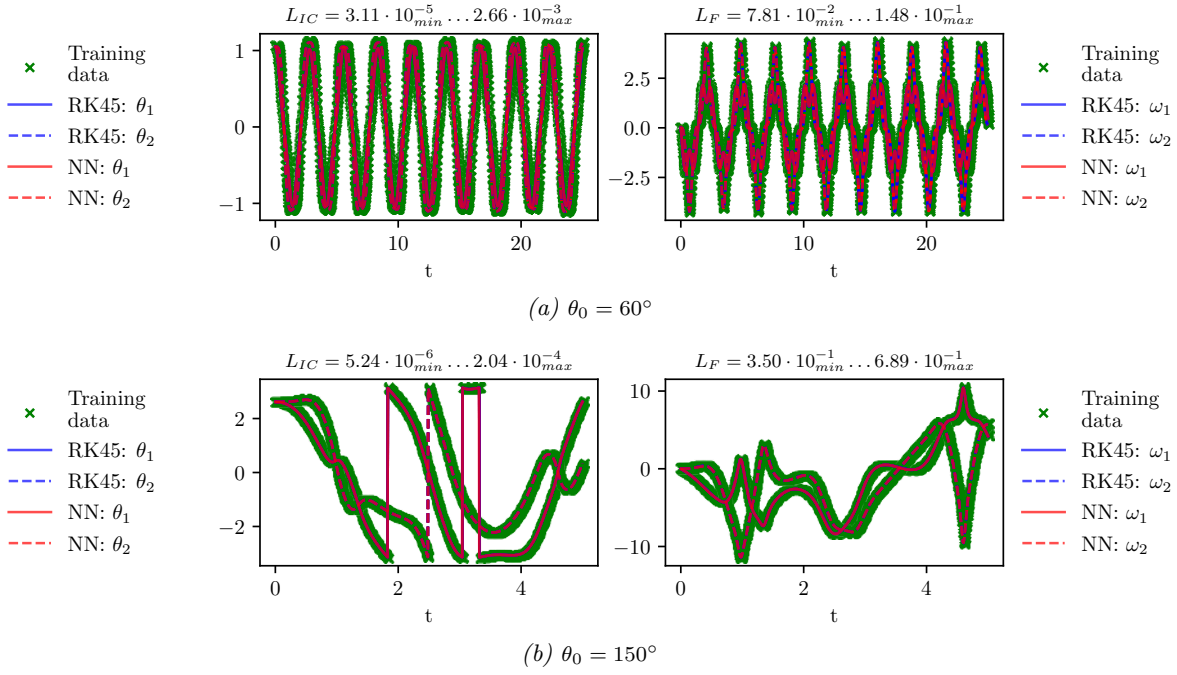


Figure 6.2: Data-driven NN. The purely data-driven NN is approximating RK training data. The network expressivity is sufficient to model high frequency components of the solution.

Discussion

In this experiment, the aim was to rule out insufficient expressivity of the neural network as the cause of the convergence problems shown in Fig. 6.1. Therefore, we trained a purely data-driven NN using training data from the RK reference solution. The results of Fig. 6.2 illustrate that the high-frequency components and sharp transitions of the reference can be well approximated with the specified network architecture.

In Fig. 6.1 (a), the vanilla PINN prediction struggles to approximate the high-frequency features of the function, resulting in minor violations of physics. Although the data-driven NN is in fact able to approximate the high-frequency components of the solution, the resulting physics loss values are about one order of magnitude higher compared to the erroneous PINN prediction. This observation additionally contributes to the convergence issue discussed in the previous chapter. We recall that due to the spectral bias [17], [23], the PINN learns low frequency solutions first, in our experiment the damped oscillation. Subsequently, the PINN is unable to escape this local minimum as the correct trajectory does not result in a lower initial condition or physics loss. Likewise, for the chaotic behavior in Fig. 6.2 (b), it is evident that the data-driven NN is able to accurately approximate the training data of the reference. However, incorporating the physics loss results in a trajectory that does not agree with the reference at all. Although the physics loss is significantly lower compared to the data-driven NN, the PINN converges to an incorrect solution with a shifted initial condition and fewer sharp transitions.

In the following section, we provide a more detailed discussion of the physical residuals for different initial conditions. We will show that the increased physics loss of the chaotic trajectory is not caused by insufficient expressive power of the neural network, being rather a consequence of errors due to the finite precision of the numerical differentiation.

6.2 Analysis of Physics Residuals

In forward problems, PINNs rely solely on the simultaneous optimization of the initial conditions and the physical residuals. However, we will illustrate why the physical residuals of the double pendulum might not be a suitable optimization objective for simulating the system, especially for chaotic motions.

In this experiment, we compute the RK solution for initial conditions $\mathbf{y}(t_0 = 0) = [\theta_0, \dot{\theta}_0, 0, 0]^T$ with increasing initial angles and a computational domain of $t_{max} = 5$. The derivatives of the system state vector are approximated numerically using a finite difference method⁵. When using finite difference methods, the quality of the approximation depends on its step size. Accuracy increases for a smaller step size, which in this experiment was chosen to be $\Delta t = 0.005$. For each initial angle $\theta_0 = \{0^\circ, 5^\circ, 10^\circ, \dots, 180^\circ\}$, we calculate the nonlinear state equations $f_i(\theta_1, \theta_2, \omega_1, \omega_2)$ and the ODE residuals $\text{res}_{F\omega_i}$ as follows.

$$\begin{pmatrix} \theta_1'' \\ \theta_2'' \end{pmatrix} = \begin{pmatrix} f_1(\theta_1, \theta_2, \omega_1, \omega_2) \\ f_2(\theta_1, \theta_2, \omega_1, \omega_2) \end{pmatrix} \quad (6.4)$$

$$= \begin{pmatrix} \frac{m_2 L_1 \omega_1^2 \sin(2\Delta\theta) + 2m_2 L_2 \omega_2^2 \sin \Delta\theta + 2gm_2 \cos \theta_2 \sin \Delta\theta + 2gm_1 \sin \theta_1}{-2L_1(m_1 + m_2 \sin^2 \Delta\theta)} \\ \frac{m_2 L_2 \omega_2^2 \sin(2\Delta\theta) + 2(m_1 + m_2)L_1 \omega_1^2 \sin \Delta\theta + 2g(m_1 + m_2) \cos \theta_1 \sin \Delta\theta}{2L_2(m_1 + m_2 \sin^2 \Delta\theta)} \end{pmatrix} \quad (6.5)$$

$$\text{res}_{F\omega_i} = |\theta_i'' - f_i(\theta_1, \theta_2, \omega_1, \omega_2)|, \quad i = 1, 2. \quad (6.6)$$

Results

The mean value of the physics residuals and the state equations for the different initial angles is displayed in Fig. 6.3. Additionally, the results of the vanilla PINN and the data-driven NN for $\theta_0 = 150^\circ$ are indicated, where the PINN converged to an erroneous solution with lower physics loss.

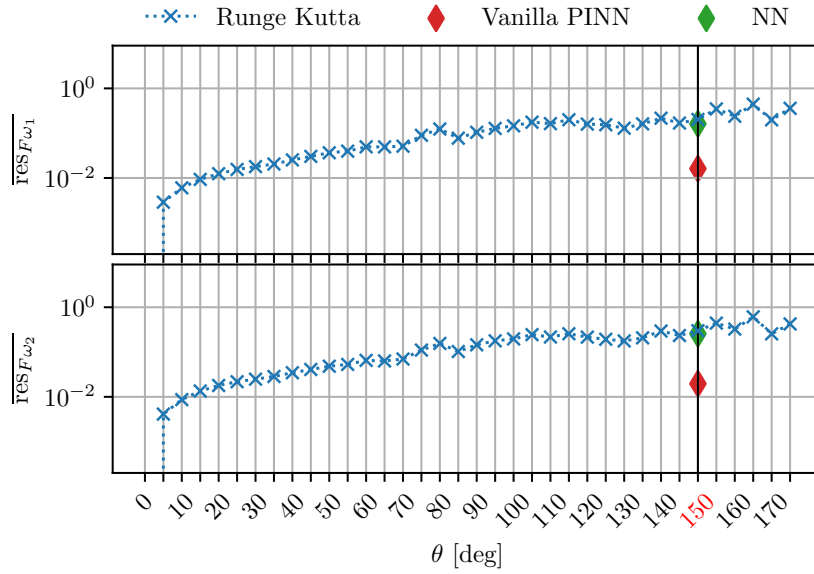
Ideally, the physical residuals of a correct solution should result in exactly zero. But in practice, we usually only achieve a value approaching zero. Aside from possible simulation errors, the derivatives used to calculate the physics residuals are approximated by numerical methods, in this experiment using a finite difference method, resulting in additional numerical errors.

From the results in Fig. 6.3 it is evident that the mean value of the physical residuals depends on the initial condition. With a larger initial angle, the angular velocity of both pendulums changes faster, resulting in larger angular accelerations θ_1'', θ_2'' . This effect is visible in Fig. 6.3 (b), where the mean of the nonlinear state equations is shown. Normalizing the residuals by the mean of the state equations, i.e., considering the relative error instead of the absolute error, yields a fairly constant value over the different initial angles.

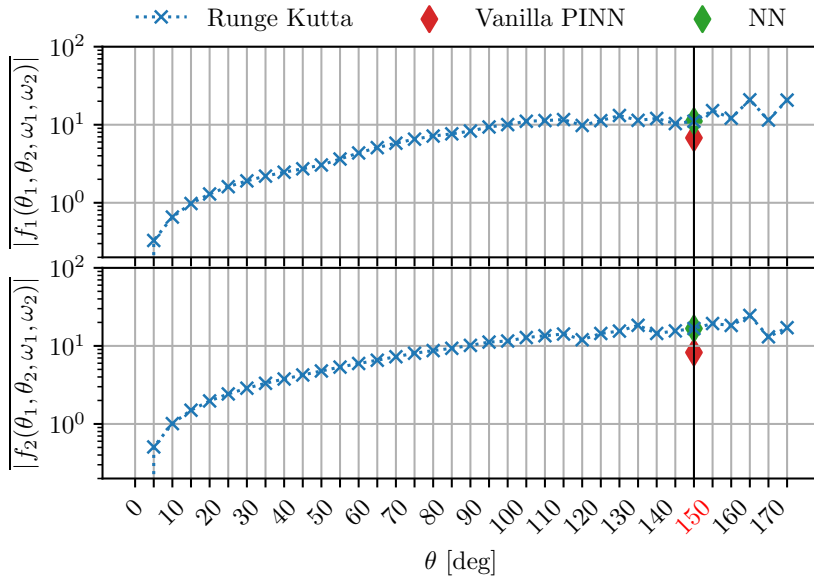
⁵ `numpy.gradient`

Discussion

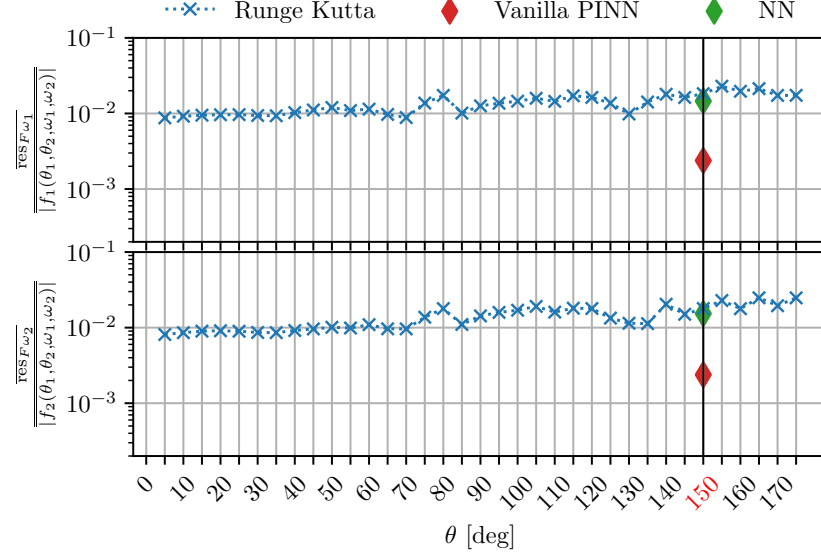
Simulating the double pendulum in the chaotic regime, in particular for $\theta_0 = 150^\circ$, results in the PINN consistently converging to a trajectory that deviates significantly from the reference. The initial condition loss is minor, but the corresponding angular velocities change less rapidly compared to the reference, resulting in lower residuals overall. In this experiment, we explored the question of why the physics loss causes this convergence problem. Therefore, we simulated the double pendulum with RK for initial angles ranging from 0° to 180° to analyze the resulting physics residuals. In Fig. 6.3 (a) and (b), we display the residuals and the state equations of the RK solution for different initial angles to illustrate their dependence on the dynamics of the underlying function. Subsequently, we compute the relative error of the residuals by



(a) Distribution of physics residuals $\text{res}_{F\omega_i} = |\theta''_i - f_i(\theta_1, \theta_2, \omega_1, \omega_2)|$, $i = 1, 2$.



(b) Distribution of nonlinear state equations



(c) Normalized residuals

Figure 6.3: $t_{max} = 5$. Mean value of the physics residuals (a), nonlinear state equations (b), and normalized residuals (c) of trajectories with initial condition $\mathbf{y}(t_0 = 0) = [\theta_0, \theta_0, 0, 0]^T$, solved with RK. The prediction of the vanilla PINN (red dot) for $\theta_0 = 150^\circ$ converges to an erroneous, less dynamical trajectory that results in significantly reduced residuals. The data driven NN (green dot) corresponds to the RK solution, indicating sufficient expressivity to approximate the function.

normalization with the state equations in Fig. 6.3 (c), confirming that functions with high angular velocities indeed lead to increased values of the physical residuals. This observation supports our claim that the physics loss is an inappropriate objective for simulating chaotic systems with PINNs. Not only is the physics loss highly non-convex and difficult to optimize, but due to approximation errors in the numerical computation of the derivatives, there seems to be a bias towards functions with less complexity, meaning smaller angular acceleration.

In this example, it can be beneficial for the optimization of the unweighted loss function to converge to an erroneous solution. Due to the chaotic behavior of the system, the initial condition needs to be violated only slightly in order to obtain a different trajectory that achieved a notably lower physics loss. To mitigate this issue, a straightforward approach is to assign more importance to the initial condition loss by significantly increasing the loss weighting parameter α_{data} . Hence, the following experiment is conducted to analyze and discuss the influence of the loss weighting scheme on PINN convergence problems.

6.3 Change of Loss Weighting Scheme

The initial condition is weakly enforced in the multi-objective cost function, which consists of the combination of data and physics loss, weighted by the hyperparameter α_{data} .

$$L(\mathbf{w}) = \alpha_{data}L_{IC}(\mathbf{w}) + (1 - \alpha_{data})L_F(\mathbf{w}), \quad \alpha_{data} \in (0, 1) \quad (6.7)$$

By increasing the loss weighting parameter $\alpha_{data} \rightarrow 1$, the initial condition loss is given more importance when minimizing the total loss function. While theoretically the two objectives should not conflict, minimization of one objective function may stall and hinder the convergence to the correct solution. In [19], this failure mode was linked to numerical stiffness in the gradient flow dynamics, resulting in an imbalance between the data loss and physics loss term during model training. Although the loss weighting parameter can be adjusted manually, an adaptive learning rate algorithm was introduced to reduce the stiffness of the gradient flow and subsequently balance the gradient of both objective functions, data loss and physics loss [17], [20]. Additionally, the influence of loss weighting schemes was analyzed in [14] by visualizing the Pareto front of the optimization problem.

In our previous experiments, we observed that the magnitude of the physics loss depends on the initial condition and the dynamics of the trajectory. In particular, for initial solutions with chaotic behavior, the unweighted PINN converges to trajectories with significantly lower physics losses compared to the reference solution, accepting a small error in the estimated initial condition. Even with variations in the given initial condition, which in Section 5.1 ranged from 141° to 156° , the PINN consistently converged to a similar attractive local minimum. This behavior contradicts the definition of chaotic behavior, which involves an extremely sensitive dependence on the initial conditions. For the unweighted loss function, these incorrect minima can actually be beneficial, since the small errors in the physics loss compensate for the resulting error in the initial conditions. Therefore, we analyze the influence of different loss evaluation schemes and their ability to penalize solutions that deviate from the given initial condition.

Results

In this experiment, the loss weighting parameter is first set manually to $\alpha_{data} = 0.99$, then the adaptive loss weighting scheme according to [20] is employed. The results for the harmonic case ($\theta_0 = 60^\circ$) and the chaotic case ($\theta_0 = 150^\circ$) are shown in Fig. 6.4 and Fig. 6.5, respectively.

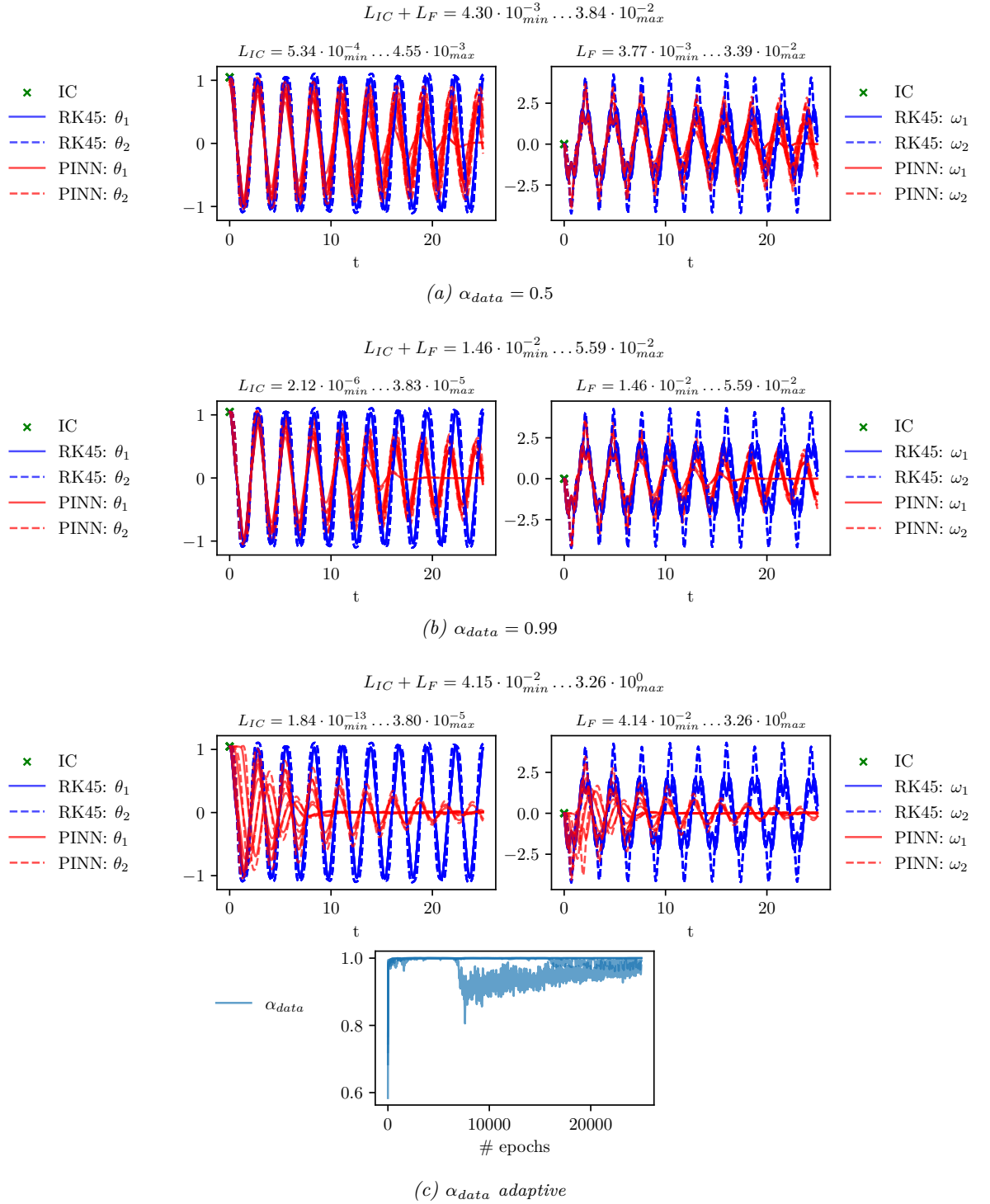


Figure 6.4: $\theta_0 = 60^\circ$: Influence of the loss weighting parameter α_{data} on the convergence to the correct solution. More emphasis on the initial condition (b) decreases the accuracy of the prediction. The PINN converges even faster to the trivial zero solution.

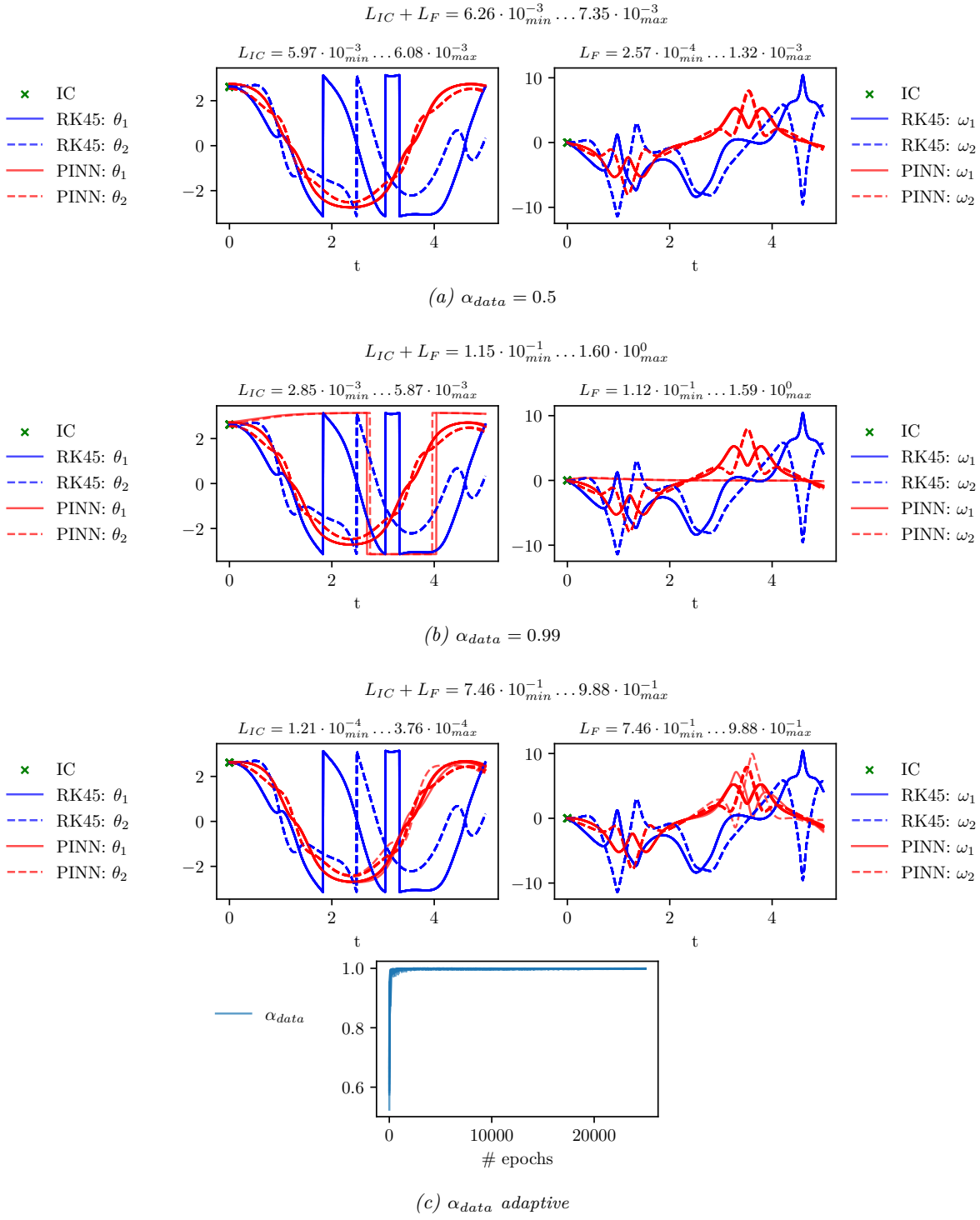


Figure 6.5: $\theta_0 = 150^\circ$: Influence of the loss weighting parameter α_{data} on the convergence to the correct solution. A stronger emphasis on the initial condition (b) even leads to convergence to the unstable fixed point $\theta_1 = \theta_2 = \pi$, $\omega_1 = \omega_2 = 0$. An adaptive loss weighting scheme (c) gives similarly inaccurate results.

Discussion

In this experiment we investigated whether a different loss weighting scheme could mitigate the attraction to incorrect solutions by penalizing deviations from the initial condition more severely. On the one hand, this can be achieved by manually increasing the loss weighting parameter α_{data} . On the other hand, we used an adaptive loss weighting scheme according to the specifications in [20]. The results for both weighting schemes are shown in Fig. 6.4 and Fig. 6.5. In the adaptive approach, the loss weighting parameter is updated with the backpropagated gradient statistics of the initial condition and the physics loss to balance both targets in terms of the stiffness of their gradients. Here, the adaptive loss weighting parameter increases rapidly to $\alpha_{data} \rightarrow 1$, again focusing on minimizing the initial condition. Therefore, it is not surprising that the initial condition loss is lower for both the harmonic and chaotic cases. The physics loss now has a negligible effect on the total cost function and exhibits higher loss values after model training compared to training with the unweighted cost function. Overall, the changed loss weighting parameter does not improve the simulation performance of the PINN by any means.

On the contrary, for the harmonic case an increased loss weighting parameter even leads to a faster convergence to the trivial solution, see Fig. 6.4. Moreover, if we consider the chaotic case in Fig. 6.5 (b), some initializations of the PINN even converge to the unstable equilibrium point at $\mathbf{y} = [\pi, \pi, 0, 0]^T$, where both pendulums are perfectly balanced in a vertical position. It is obvious that convergence to such an unstable equilibrium is physically impossible and should not occur in the simulation of the system. However, the complexity of this trajectory is low, and with initial conditions near the equilibrium point, the physics loss is comparatively small. Basically, the PINN would have to violate physics only in a small domain to reach the equilibrium point, since afterwards the physics loss is fulfilled. In [15], the authors observed similar behavior, with the PINN favoring convergence to the unstable zero solution.

6.4 Reduction of Computational Domain

In Section 5.1, we observed that PINNs can successfully solve the ODE problem in the harmonic domain for low frequencies⁶. As the domain length increases, convergence problems emerge, leading to solutions that slowly fade to the trivial zero solution. Additionally, a second type of convergence problem was presented, occurring in the presence of chaotic motion. There, the PINN prediction yields a nontrivial solution that satisfies the physical constraints but deviates from the given initial condition. For both the harmonic and chaotic convergence problems, changing the loss weighting scheme did not lead to an improved convergence to the correct solution. Instead, it resulted in the same trajectories or solutions that deviated even further from the reference.

Lastly, a promising approach to mitigate these convergence problems is to reduce the computational domain as proposed in several domain decomposition schemes [12], [13], [16], [21], [22]. The authors of [15] have demonstrated that a large computational domain can lead to suboptimal local minima that correspond to nonphysical behavior of the system, such as convergence to unstable solutions. By reducing the size of the computational domain, these suboptimal minima gradually disappear. Furthermore, they have shown that reducing the size of the computational domain leads to a lower complexity of the physical loss landscape, which overall exhibits a smoother landscape with fewer suboptimal minima. For the harmonic case, we have already demonstrated that the PINN provides the correct solution for shorter computational domains. Therefore, we want to investigate in the following experiment whether this is also true for the PINN simulation in the chaotic domain.

Results

Fig. 6.6 shows the results for $\theta_0 = 150^\circ$ for a reduced computational domain and different loss weighting parameters α_{data} .

Discussion

In this experiment, we significantly reduced the computational domain for the simulation of the chaotic solution with $\theta_0 = 150^\circ$. The results in Fig. 6.6 show that convergence is strongly affected by the length of the computational domain and an appropriate choice of the loss weighting parameter α_{data} . The optimization of the unweighted loss function again leads to a solution that deviates from the reference solution. Only after increasing the loss weighting parameter, the PINN is able to converge to the correct solution. While the weighted PINN does approximate the initial condition more precisely, the physics loss of both predictions is still in a similar range.

In theory, the correct trajectory should minimize both initial loss and physics loss perfectly. Thus, we consider the sum of the two objective function as a measure of how well the optimization problem is fulfilled. Interestingly, the overall loss score for both loss weighting parameters is in a similar range, although the predicted trajectories vary strongly. This observation carries some unpleasant consequences with regards to the optimization of the multi-objective cost

⁶ By using input normalization to a fixed range, small computational domains correspond to low frequency functions.

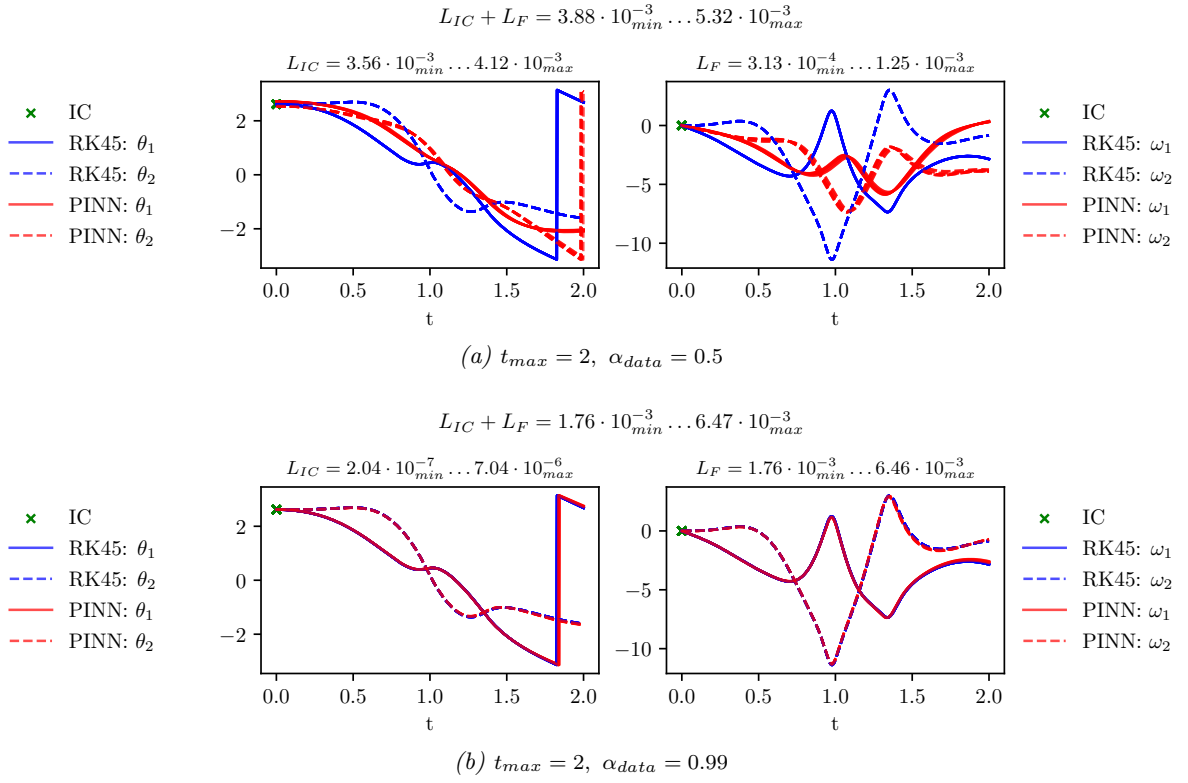


Figure 6.6: Influence of the size of the computational domain for $\theta_0 = 150^\circ$ and $\omega_1^{(t_0)} = \omega_2^{(t_0)} = 0$. Reduction of the computational domain in combination with loss weighting improves convergence. The resulting loss values are in a similar range, although the trajectories vary strongly.

function. Even if it was possible to optimize the loss function without any convergence issues, the optimization relies heavily on the loss weighting scheme. Therefore, it can be challenging to select the loss weighting parameter in a way that the correct solution coincides with the global optimum of the weighted loss function. In the next section, we visualize the Pareto front of the multi-objective optimization problem to further analyze the impact of the loss weighting scheme and the reduction of the computational domain.

6.5 Pareto Front

The Pareto front of a multi-objective optimization problem is the set of Pareto optima, for which no singular objective can be decreased without increasing another one. In this experiment, we follow the procedure described in [14] where the Pareto front is qualitatively scanned by sampling $\alpha_{data} \in (0, 1)$. There, the authors demonstrated that convergence to the correct solutions is determined by the innate shape of the Pareto front, using the diffusion equation and Navier-Stokes equations. The shape of Pareto front can be concave, convex, or a mixture of both, influenced by several factors such as the network architecture, activation function, or systems parameters. If the Pareto front is convex, small changes of the loss weighting parameter lead to closely spaced Pareto optima. In this case, a sweep over α_{data} would allow convergence to a correct solution. For a concave and mixed Pareto front, however, the solutions are highly dependent on the loss weights. The optimal choice is difficult to determine without prior knowledge about the true trajectory.

In the following experiment, we analyze the Pareto front for the double pendulum in the harmonic and chaotic regime. We show that, especially in the chaotic regime, the PINN predominantly converges to some highly attractive local minima that cannot be escaped by simply changing the loss weighting scheme. The correct solution is only achievable for reduced computational domains.

6.5.1 Concave Pareto Front

First, we examine the Pareto front for settings that converge to the correct solution (Fig. 6.7). The corresponding trajectories for the different parameters of the loss weighting are shown in Fig. 6.8 and show that the PINN converges to the correct solution for all $\alpha_{data} \leq 0.5$. If the loss weight is reduced further, the initial condition no longer has any significant influence. Thus, the prediction converges towards trivial zero solution with an extremely low physics loss.

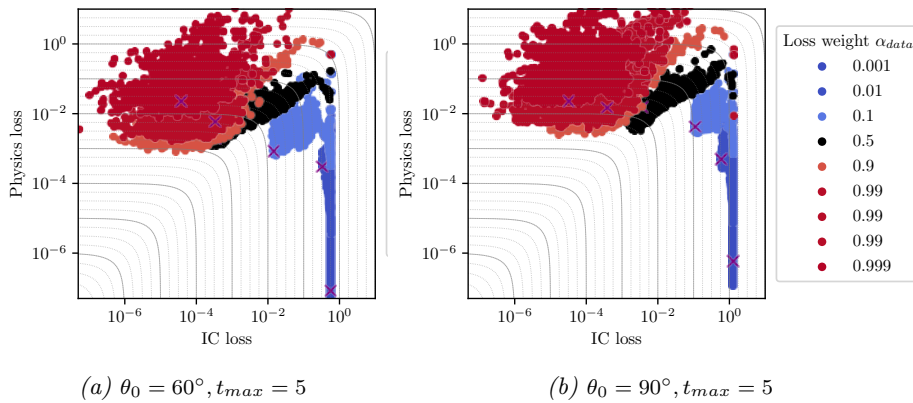


Figure 6.7: Pareto front for $\theta_0 = \{60^\circ, 90^\circ\}$, converging to the correct solution. Initial condition loss L_{IC} (x-axis) versus physics loss L_F (y-axis) over the training epochs for various loss weights α_{data} . Pareto optima for each loss weighting parameter are plotted as crosses. Gray lines indicate regions with constant loss value $L_F + L_{IC}$.

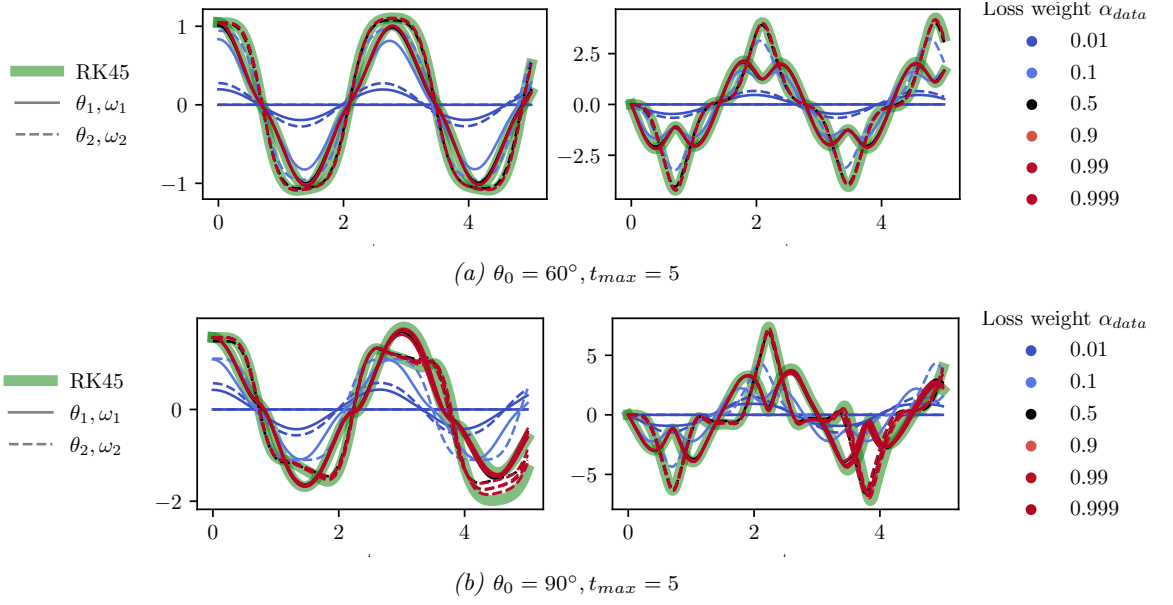


Figure 6.8: PINN prediction for the initial conditions $\theta_0 = 60^\circ$ (a) and $\theta_0 = 90^\circ$ (b). Convergence to the correct solution is possible by increasing the loss weighting parameter α_{data} .

6.5.2 Irregular Pareto Front

Next, the chaotic case for $\theta_0 = 150^\circ$ is considered, where we encountered severe convergence problems. The results for the Pareto front and for the corresponding trajectories are shown in Fig. 6.9 and 6.10. As demonstrated in previous experiments, convergence to the correct solution is only possible after reducing the computational domain and for $\alpha_{data} \rightarrow 1$. For $\alpha_{data} \rightarrow 0$ the prediction corresponds to the physically correct trajectory for a reduced initial angle. Any other choice of α_{data} leads to a local minimum that exhibits a large deviation from the correct trajectory, but achieves comparatively small training losses. While the correct solution is no longer achievable for a larger computational domain, the non-trivial solution is still highly attractive and consistently results in a small optimization loss.

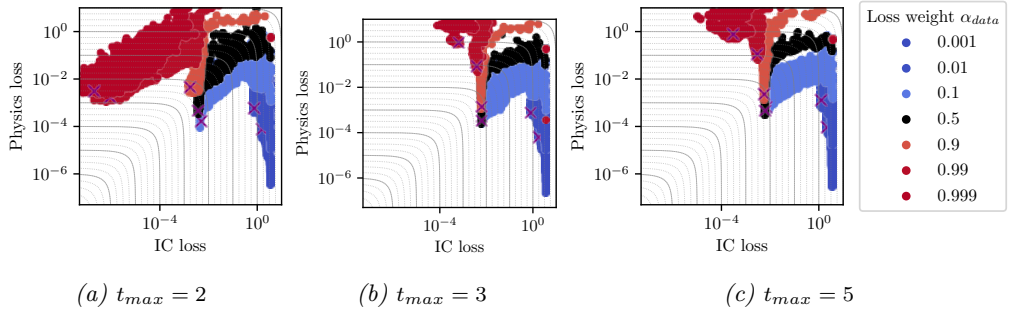


Figure 6.9: Pareto front for $\theta_0 = 150^\circ$, setting with convergence issues. Initial condition loss L_{IC} (x-axis) versus physics loss L_F (y-axis) over the training epochs for various loss weights α_{data} . Pareto optima for each loss weighting parameter are plotted as crosses. Gray lines indicate regions with constant loss value $L_F + L_{IC} = \text{const}$. The Pareto front shape displays a highly attractive local minimum achieving low loss values but strongly deviating from the correct trajectory. Convergence to the correct solution is only achieved for $t_{max} = 2$ (a) and $\alpha_{data} \rightarrow 1$.

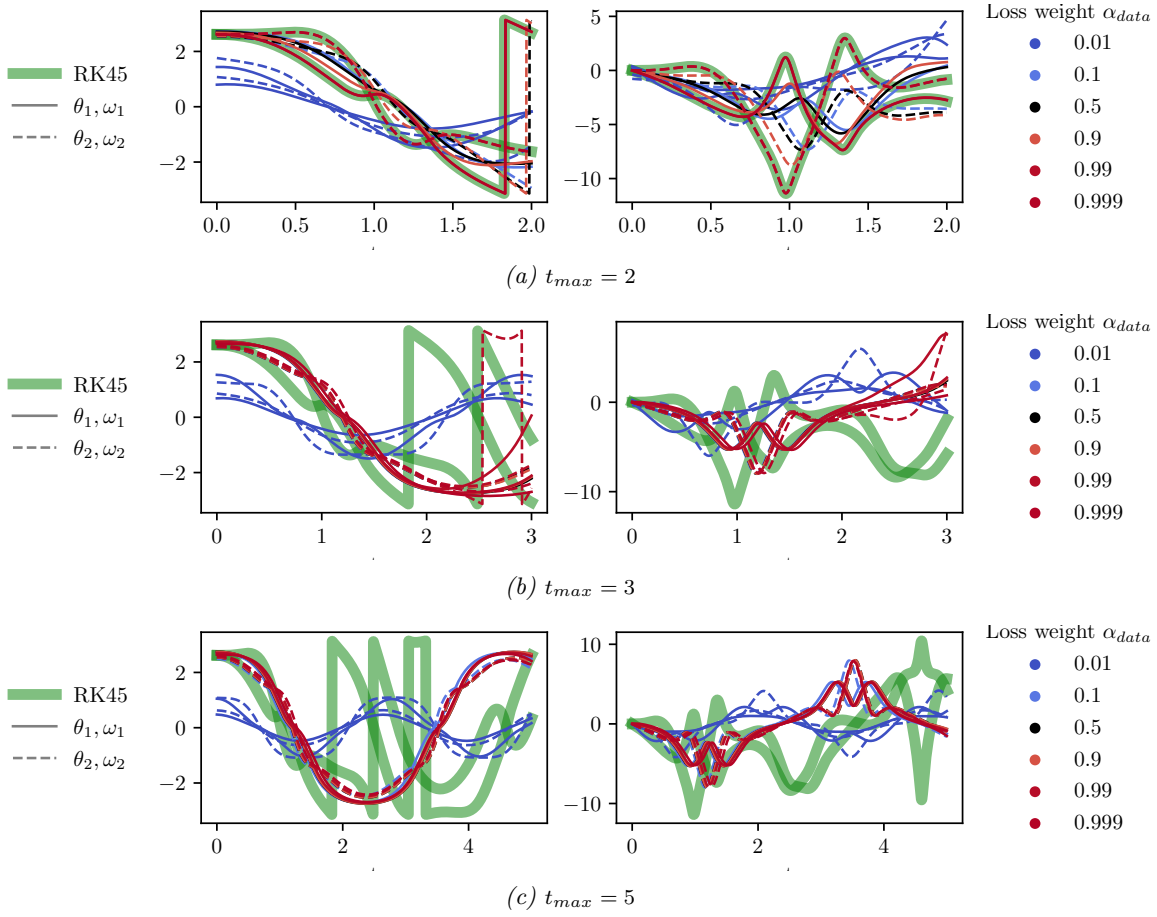


Figure 6.10: PINN prediction for the initial conditions $\theta_0 = 150^\circ$ and different computational domains. Convergence to the correct solution is only achieved for $t_{max} = 2$ (a) and $\alpha_{data} \rightarrow 1$.

Discussion

As a first step in this experiment, we considered the settings where we did not experience any convergence issues, i.e. initial angles of $\theta_0 = 60^\circ$ and 90° , and a computational domain of $t \in [0, 5]$. The results for the Pareto front and the predicted trajectories corresponding to the Pareto optima are shown in Fig. 6.7 and 6.8. In both cases, the Pareto front is concave and the PINN converges to a correct solution for $\alpha_{data} \leq 0.5$.

The more interesting case occurs in the prediction of chaotic behavior for an initial angle of $\theta_0 = 150^\circ$. Here, convergence to the correct solution is possible only after a substantial reduction of the computational domain from $t_{max} = 5$ to $t_{max} = 2$, and for a loss weighting parameter of $\alpha_{data} \rightarrow 1$, see Fig. 6.9 (a). Besides the correct solution, two other distinct dips of the Pareto front can be seen. The first occurs for loss weights $\alpha_{data} \rightarrow 0$, where the resulting trajectory strongly violates the initial condition but precisely satisfies the physical conditions. For all other values of α_{data} , the PINN converges to the second, dominating Pareto optimum, where both the initial condition and the physics loss are low. Previously, we observed in Section 5.1 that the PINN displays no sensitivity with respect to perturbations of the given initial condition, preferring to converge to this erroneous solution. In this experiment, we show that this local minimum is also highly robust against changes in the loss weighting parameter and changes in

the computational domain. While the local minimum corresponding to the correct trajectory can only be obtained for $t_{max} = 2$ and $\alpha_{data} \rightarrow 1$, the incorrect local minima persist in all computational domains. This highlights the difficulty of training PINNs to simulate a system with chaotic behavior.

6.6 Summary

In the previous experiments, we have demonstrated that the ODE residuals of the undamped double pendulum are not a suitable loss function for simulating dynamical systems using PINNs, especially in the chaotic regime. Due to the numerical approximation of the derivatives, the physics loss is typically a nonzero quantity. In our case, the physics loss is increased for solutions that exhibit rapidly changing motion, i.e., high values of angular acceleration θ_i'' . Based on this observation, we can distinguish two different convergence problems for the harmonic and chaotic cases, respectively.

The simulation of harmonic motion is successful for time intervals covering up to five periods of oscillation. However, convergence problems occur when the computational domain is further increased. Then, the PINN solution displays a damped behavior and slowly fades to the trivial zero solution. Compared to the data-driven NN solution, which is able to model the high-frequency features, the resulting physics loss is still in the same range as the erroneous PINN solution. Due to the spectral bias [17], [23], the PINN exhibits a learning bias towards lower frequency solutions. As a result, the PINN may not be able to escape the local minimum corresponding to the damped solution as it cannot minimize the multi-objective loss function further.

In the chaotic regime, the PINN suffers severely from convergence issues, even for computational domains where the harmonic oscillation could be predicted accurately. Although the expressivity of the network would be sufficient to model the correct solution, the PINN still converges to a solution that completely differs from the reference, exhibiting less dynamics and rapidly changing trajectories. The PINN prediction does violate the given initial condition slightly, but the physical constraints are fulfilled accurately and result in notably lower loss values compared to the reference solution. As we incorporate the initial condition and physics loss as soft constraints, it can be beneficial to accept a small residual error for the initial condition, that results in a simpler trajectory for which the physics loss is low. As a consequence, PINNs do not reflect a core property of chaotic systems - the sensitivity to perturbations of the initial conditions. On the contrary, the PINN consistently converges to a similar incorrect solution for small variations of the initial angle, while the corresponding RK reference solutions diverge strongly.

Intuitively, we assumed that a different loss weighting scheme might improve convergence, by assigning more importance to the initial condition. However, even if the given initial condition is fulfilled accurately, the PINN is not guaranteed to converge to the correct solution. Still, it may be advantageous to violate the physical constraints in a small domain and converge to a trajectory of low complexity and a final low physics loss, compared to the reference. The larger the size of the computational domain and the lower the weight on the physics loss, the more beneficial this trade-off seems to become.

These attractive local minima were also evident in the visualization of the Pareto front of the optimization problem. It was demonstrated that convergence to the correct solution is only possible for reduced computational domains and for particular loss weighting parameters. While the correct solution is no longer achieved for larger computational domains, the incorrect local minimum persists and forms the Pareto optimum for a wide range of loss weighting parameters.

In summary, PINNs have been shown to provide unreliable results for the simulation of chaotic behavior, unless the computational domain is reduced significantly and the loss weighting scheme chosen appropriately.

Conclusion

The aim of this thesis was to study the performance of physics informed neural networks (PINNs) for the simulation of a dynamical system. By performing several experiments on the undamped double pendulum, we analyzed failure modes in the harmonic and the chaotic regime.

In initial experiments, we demonstrated the performance of PINNs in the presence of noisy, incomplete, or only partially available training data. Compared to a purely data-driven approach, the PINN was able to achieve noticeably better predictions. However, we also observed a strong dependence of the prediction accuracy on the initial condition. Particularly in the chaotic regime, the simultaneous optimization of the data and physics loss hindered convergence to the correct solution even for noise-free training data. In further experiments, we focused on solving forward problems without the aid of training data. For the simulation of harmonic oscillations, convergence failed for large computational domains. The PINN prediction faded to the trivial zero solution, although the expressivity of the network was sufficient to approximate the correct solution. Despite the fact that this behavior violated the physical constraints of the undamped pendulum, the physics loss of the PINN prediction was in a similar range to that of the reference solution. In the chaotic regime, the PINN prediction heavily deviated from the correct solution and did not exhibit any sensitivity to changes in initial conditions. Instead, the PINN consistently converged to incorrect but dominating local minima that result in a solution with lower complexity and a significantly reduced physics loss compared to the reference. Convergence to the correct solution was shown to be limited to short computational domains along with an appropriate choice of loss weights. This defeats the advantageous property of PINNs of predicting the solution for the entire computational domain at once. Again, a small discretization of the domain is required to achieve convergence to the correct solution, making the PINN more akin to a physically informed integrator.

In this work, we have provided some further insight into the problem of training PINNs for the simulation of chaotic behavior. However, there are still some open questions that need further research. We would like to investigate the influence of different activation functions in more detail and address whether the PINN would show the same behavior in the chaotic regime and converge to a similar attractive solution. Moreover, it could be informative to analyze the effects of encoding the initial conditions as hard conditions to avoid small violations of the initial loss. In addition, it remains open to what extent the convergence problems of PINNs in our case are related to gradient descent optimization problems and whether hybrid models with, e.g., particle swarm optimization could be an appropriate tool.

In summary, PINNs are an interesting research area with a lot of potential and benefits over traditional numerical integration methods and purely data-driven neural networks. However, up to now, they have not been successful in simulating dynamical systems for large computational

domains and in the chaotic regime. To overcome those problems, we also need to gain a deeper understanding of the underlying physics to assess which applications can be solved reliably using PINNs.

List of Figures

2.1	Schematic of the double pendulum with point masses m_1 , m_2 , and lengths L_1 , L_2 .	8
2.2	Trajectory of the double pendulum for different initial conditions resulting in harmonic oscillation (a) or chaotic motion (b) and (c).	10
2.3	Bifurcation diagram for a double pendulum with equal lengths, equal masses, and 9.81 m/s^2 . The initial condition is given by $\mathbf{y}^{(t_0)} = [\theta_0, \dot{\theta}_0, 0, 0]^T$ (taken from [27]).	10
3.1	Network architecture of the PINN for the first-order (a) and second-order (b) ODE description. The NN parameters \mathbf{w} are trained by minimizing the combined loss L consisting of data and ODE loss.	14
3.2	Initial condition of $\theta_0 = 90^\circ$ and $\omega_1^{(t_0)} = \omega_2^{(t_0)} = 0$. Prediction of θ_1, θ_2 (left) and ω_1, ω_2 (right) over time t . Comparison between PINN prediction using the set of first-order ODEs and second-order ODEs. In (a) the angular velocities ω_i are trained by minimizing the physics residuals, while in (b) they are obtained by taking the derivative $\omega_i = \theta'_i$.	17
3.3	Swish activation function. The input variable is normalized to the shaded range $[-5, 5]$.	18
3.4	$\theta_0 = 60^\circ$. Comparison between PINN prediction without input normalization (a) and with input normalization to $[-5, 5]$ (b).	18
4.1	Predicted trajectories of the data driven and physics informed approach for an initial angle of $\theta_0 = 90^\circ$ and noisy training data.	21
4.2	Performance of NNs and PINNs for varying initial angles θ_0 and noise variance σ_n^2 . The MSE between the predicted and the reference solution is shown in the left column and the physics loss in the right column.	22
4.3	Predicted trajectories of the data-driven and physics informed approach for an initial angle of 120° and noise-free training data that is missing in an interval.	24
4.4	Performance of data-driven NNs and PINNs for varying initial angles θ_0 and interval length without training data points. The MSE between the predicted and the reference solution is shown in the left column and the physics loss in the right column.	25
4.5	Prediction of the motion of the hidden pendulum using partial training data. The problem is ill-posed, the initial angle of $\theta_0 = 90^\circ$ is unknown to the PINN during training. The PINN is able to accurately reconstruct the motion of the hidden pendulum.	28
4.6	Prediction of the motion of the hidden pendulum using partial training data. The problem is ill-posed, the initial angle of $\theta_0 = 150^\circ$ is unknown to the PINN during training. The PINN suffers from strong convergence issues and is not able to predict the observable and the hidden state accurately.	29

4.7	Performance of PINNs for varying initial angles θ_0 and lengths of the computational domain given training data about one pendulum and inferring the solution of the other pendulum. The MSE between the predicted and the reference solution is shown in the left column and the physics loss in the right column.	30
5.2	Predicted trajectories for different initial angles with no additional data. Strong convergence issues for initial angles of 120° and 150° , leading to different trajectories.	34
5.3	The initial condition of the PINN predictions used to estimate the trajectory with RK to verify accuracy with respect to the physical constraints.	35
5.4	Convergence of the PINN to an attractive local minimum for several different initial conditions in the chaotic regime.	36
6.1	Vanilla PINN , $\alpha_{data} = 0.5$. The vanilla PINN exhibits convergence issues for harmonic oscillations over large domains (a) and chaotic motion (b).	41
6.2	Data-driven NN. The purely data-driven NN is approximating RK training data. The network expressivity is sufficient to model high frequency components of the solution.	42
6.3	$t_{max} = 5$. Mean value of the physics residuals (a), nonlinear state equations (b), and normalized residuals (c) of trajectories with initial condition $\mathbf{y}(t_0 = 0) = [\theta_0, \dot{\theta}_0, 0, 0]^T$, solved with RK. The prediction of the vanilla PINN (red dot) for $\theta_0 = 150^\circ$ converges to an erroneous, less dynamical trajectory that results in significantly reduced residuals. The data driven NN (green dot) corresponds to the RK solution, indicating sufficient expressivity to approximate the function. .	45
6.4	$\theta_0 = 60^\circ$: Influence of the loss weighting parameter α_{data} on the convergence to the correct solution. More emphasis on the initial condition (b) decreases the accuracy of the prediction. The PINN converges even faster to the trivial zero solution.	47
6.5	$\theta_0 = 150^\circ$: Influence of the loss weighting parameter α_{data} on the convergence to the correct solution. A stronger emphasis on the initial condition (b) even leads to convergence to the unstable fixed point $\theta_1 = \theta_2 = \pi$, $\omega_1 = \omega_2 = 0$. An adaptive loss weighting scheme (c) gives similarly inaccurate results.	48
6.6	Influence of the size of the computational domain for $\theta_0 = 150^\circ$ and $\omega_1^{(t_0)} = \omega_2^{(t_0)} = 0$. Reduction of the computational domain in combination with loss weighting improves convergence. The resulting loss values are in a similar range, although the trajectories vary strongly.	51
6.7	Pareto front for $\theta_0 = \{60^\circ, 90^\circ\}$, convergence to correct solution	52
6.8	PINN prediction for the initial conditions $\theta_0 = 60^\circ$ (a) and $\theta_0 = 90^\circ$ (b). Convergence to the correct solution is possible by increasing the loss weighting parameter α_{data}	53
6.9	Pareto front for $\theta_0 = 150^\circ$, convergence issues	53

6.10 PINN prediction for the initial conditions $\theta_0 = 150^\circ$ and different computational domains. Convergence to the correct solution is only achieved for $t_{max} = 2$ (a) and $\alpha_{data} \rightarrow 1$	54
--	----

List of Tables

3.1	Specifications of the fully connected network architecture for the preliminary experiments.	16
4.1	Specifications of the fully connected network architecture for training with noisy data.	20
4.2	Specifications of the fully connected network architecture for training with incomplete data.	23
4.3	Specifications of the fully connected network architecture for training with partially observed data.	27
5.1	Specifications of the fully connected network architecture for a forward problem, given the initial condition and ODE.	32
6.1	Specifications of the fully connected network architecture for the analysis of convergence issues.	40

Abbreviations

AD automatic differentiation.

FBPINN Finite Basis PINN.

MSE mean squared error.

NN neural network.

ODE ordinary differential equation.

PDE partial differential equation.

PINN physics informed neural network.

RK Runge Kutta.

XPINN eXtendend PINN.

Bibliography

- [1] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, “Physics-informed machine learning,” *Nature Reviews Physics*, DOI: 10.1038/s42254-021-00314-5. [Online]. Available: www.nature.com/natrephys.
- [2] M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” *Journal of Computational Physics*, vol. 378, pp. 686–707, Feb. 2019, ISSN: 0021-9991. DOI: 10.1016/J.JCP.2018.10.045.
- [3] C. L. Wight and J. Zhao, “Solving allen-cahn and cahn-hilliard equations using the adaptive physics informed neural networks,” *Communications in Computational Physics*, vol. 29, pp. 930–954, 3 Jul. 2020, ISSN: 19917120. DOI: 10.48550/arxiv.2007.04542. [Online]. Available: <https://arxiv.org/abs/2007.04542v1>.
- [4] F. S. Costabal, Y. Yang, P. Perdikaris, D. E. Hurtado, and E. Kuhl, “Physics-informed neural networks for cardiac activation mapping,” *Frontiers in Physics*, vol. 8, p. 42, Feb. 2020, ISSN: 2296424X. DOI: 10.3389/FPHY.2020.00042/BIBTEX.
- [5] K. Shukla, P. C. D. Leoni, J. Blackshire, D. Sparkman, and G. E. Karniadakis, “Physics-informed neural network for ultrasound nondestructive quantification of surface breaking cracks,” *Journal of Nondestructive Evaluation*, vol. 39, pp. 1–20, 3 Sep. 2020, ISSN: 15734862. DOI: 10.1007/S10921-020-00705-1/FIGURES/16. [Online]. Available: <https://link.springer.com/article/10.1007/s10921-020-00705-1>.
- [6] S. Wang and P. Perdikaris, “Deep learning of free boundary and stefan problems,” *Journal of Computational Physics*, vol. 428, p. 109914, Mar. 2021, ISSN: 0021-9991. DOI: 10.1016/J.JCP.2020.109914.
- [7] E. Kharazmi, M. Cai, X. Zheng, Z. Zhang, G. Lin, and G. E. Karniadakis, “Identifiability and predictability of integer- and fractional-order epidemiological models using physics-informed neural networks,” *Nature Computational Science 2021 1:11*, vol. 1, pp. 744–753, 11 Nov. 2021, ISSN: 2662-8457. DOI: 10.1038/s43588-021-00158-0. [Online]. Available: <https://www.nature.com/articles/s43588-021-00158-0>.
- [8] G. Pang, L. U. Lu, and G. E. Karniadakis, “Fpinns: Fractional physics-informed neural networks,” <https://doi.org/10.1137/18M1229845>, vol. 41, A2603–A2626, 4 Aug. 2019, ISSN: 10957197. DOI: 10.1137/18M1229845. [Online]. Available: <https://epubs.siam.org/doi/abs/10.1137/18M1229845>.
- [9] M. Yin, X. Zheng, J. D. Humphrey, and G. E. Karniadakis, “Non-invasive inference of thrombus material properties with physics-informed neural networks,” *Computer Methods in Applied Mechanics and Engineering*, vol. 375, p. 113603, Mar. 2021, ISSN: 0045-7825. DOI: 10.1016/J.CMA.2020.113603.
- [10] M. Raissi, A. Yazdani, and G. E. Karniadakis, “Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations,” *Science*, vol. 367, pp. 1026–1030, 6481 Feb. 2020, ISSN: 10959203. DOI: 10.1126/science.aaw4741.

- [11] S. Cai, Z. Wang, F. Fuest, Y. J. Jeon, C. Gray, and G. E. Karniadakis, “Flow over an espresso cup: Inferring 3d velocity and pressure fields from tomographic background oriented schlieren videos via physics-informed neural networks,” 2021.
- [12] A. D. Jagtap and G. E. Karniadakis, “Extended physics-informed neural networks (xpinns) : A generalized space-time domain decomposition based deep learning framework for non-linear partial differential equations,” [Online]. Available: <https://github.com/AmeyaJagtap/XPINNs>.
- [13] B. Moseley, A. Markham, and T. Nissen-Meyer, “Finite basis physics-informed neural networks (fbpinns): A scalable domain decomposition approach for solving differential equations a preprint,” 2021.
- [14] F. M. Rohrhofer, S. Posch, and B. C. Geiger, “On the pareto front of physics-informed neural networks,” May 2021. [Online]. Available: <http://arxiv.org/abs/2105.00862>.
- [15] F. M. Rohrhofer, S. Posch, C. Gössnitzer, and B. C. Geiger, “Understanding the difficulty of training physics-informed neural networks on dynamical systems,” 2021.
- [16] A. S. Krishnapriyan, A. Gholami, S. Zhe, R. M. Kirby, and M. W. Mahoney, “Characterizing possible failure modes in physics-informed neural networks,”
- [17] S. Wang, H. Wang, and P. Perdikaris, “On the eigenvector bias of fourier feature networks: From regression to solving multi-scale pdes with physics-informed neural networks,” 2020. [Online]. Available: <https://github.com/>.
- [18] S. Wang, S. Sankaran, and P. Perdikaris, “Respecting causality is all you need for training physics-informed neural networks,” 2022.
- [19] S. Wang, Y. Teng, and P. Perdikaris, “Understanding and mitigating gradient pathologies in physics-informed neural networks a preprint,” 2020. [Online]. Available: <https://github.com/PredictiveIntelligenceLab/GradientPathologiesPINNs>.
- [20] X. Jin, S. Cai, H. Li, and G. E. Karniadakis, “Nsfnets (navier-stokes flow nets): Physics-informed neural networks for the incompressible navier-stokes equations,” *Journal of Computational Physics*, vol. 426, Mar. 2020. DOI: 10.1016/j.jcp.2020.109951. [Online]. Available: <https://arxiv.org/abs/2003.06496v1>.
- [21] R. Leiteritz and D. P. Uger, “How to avoid trivial solutions in physics-informed neural networks,” 2021.
- [22] J. Yu, L. Lu, X. Meng, and G. E. Karniadakis, “Gradient-enhanced physics-informed neural networks for forward and inverse pde problems,” 2021.
- [23] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville, “On the spectral bias of neural networks,” in *Proceedings of the 36th International Conference on Machine Learning*, K. Chaudhuri and R. Salakhutdinov, Eds., ser. Proceedings of Machine Learning Research, vol. 97, PMLR, Jun. 2019, pp. 5301–5310.
- [24] J. C. Butcher, *Numerical Methods for Ordinary Differential Equations*. John Wiley & Sons, Ltd, Jul. 2016. DOI: 10.1002/9781119121534. [Online]. Available: <https://doi.org/10.1002/9781119121534>.

-
- [25] S. H. Strogatz, *Nonlinear Dynamics and Chaos*. CRC Press, May 2018. DOI: 10.1201/9780429492563. [Online]. Available: <https://doi.org/10.1201/9780429492563>.
 - [26] T. Stachowiak and T. Okada, “A numerical analysis of chaos in the double pendulum,” *Chaos, Solitons & Fractals*, vol. 29, no. 2, pp. 417–422, 2006, ISSN: 0960-0779. DOI: <https://doi.org/10.1016/j.chaos.2005.08.032>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0960077905006703>.
 - [27] Fridtj of Nansen. “Bifurkationsdiagram Doppelpendel.” This file is licensed under the Creative Commons Attribution-Share Alike 4.0 International license. (2016), [Online]. Available: https://commons.wikimedia.org/wiki/File:Bifurkationsdiagram_Doppelpendel.png.
 - [28] M. A. Nabian, R. J. Gladstone, and H. Meidani, “Efficient training of physics-informed neural networks via importance sampling,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 36, no. 8, pp. 962–977, Apr. 2021. DOI: 10.1111/mice.12685. [Online]. Available: <https://doi.org/10.1111/mice.12685>.
 - [29] C. J. Arthurs and A. P. King, “Active training of physics-informed neural networks to aggregate and interpolate parametric solutions to the navier-stokes equations,” *Journal of Computational Physics*, vol. 438, p. 110364, Aug. 2021. DOI: 10.1016/j.jcp.2021.110364. [Online]. Available: <https://doi.org/10.1016/j.jcp.2021.110364>.
 - [30] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, “Automatic differentiation in machine learning: A survey,” 2015. DOI: 10.48550/ARXIV.1502.05767. [Online]. Available: <https://arxiv.org/abs/1502.05767>.
 - [31] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic differentiation in pytorch,” in *NIPS 2017 Workshop on Autodiff*, Long Beach, California, USA, 2017. [Online]. Available: <https://openreview.net/forum?id=BJJsrmfCZ>.
 - [32] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, *Tensorflow: Large-scale machine learning on heterogeneous distributed systems*, 2016. DOI: 10.48550/ARXIV.1603.04467. [Online]. Available: <https://arxiv.org/abs/1603.04467>.
 - [33] R. D. Reed and R. J. Marks, *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*. MIT Press, 1999.
 - [34] S. Maddu, D. Sturm, C. L. Müller, and I. F. Sbalzarini, “Inverse dirichlet weighting enables reliable training of physics informed neural networks,” 2021. DOI: 10.1088/2632. [Online]. Available: <https://creativecommons.org/licenses/by/3.0>.
 - [35] X. Jin, S. Cai, H. Li, and G. E. Karniadakis, “Nsfnets (navier-stokes flow nets): Physics-informed neural networks for the incompressible navier-stokes equations,” *Journal of Computational Physics*, vol. 426, Mar. 2020. DOI: 10.1016/j.jcp.2020.109951. [Online]. Available: <https://arxiv.org/abs/2003.06496v1>.
-