J. Korean Soc. Ind. Appl. Math. Vol.28, No.1, 33–58, 2024 https://doi.org/10.12941/jksiam.2024.28.033

ERROR ESTIMATES OF PHYSICS-INFORMED NEURAL NETWORKS FOR INITIAL VALUE PROBLEMS

JIHAHM YOO[1], JAYWON KIM[1], MINJUNG GIM [2], AND HAESUNG LEE [3,†]

[1] KOREA SCIENCE ACADEMY OF KAIST, REPUBLIC OF KOREA

[2] NATIONAL INSTITUTE FOR MATHEMATICAL SCIENCES, REPUBLIC OF KOREA

[3] DEPARTMENT OF MATHEMATICS AND BIG DATA SCIENCE, KUMOH NATIONAL INSTITUTE OF TECHNOLOGY, REPUBLIC OF KOREA
Email address: †fthslt@kumoh.ac.kr, fthslt14@gmail.com

ABSTRACT. This paper reviews basic concepts for Physics-Informed Neural Networks (PINN) applied to the initial value problems for ordinary differential equations. In particular, using only basic calculus, we derive the error estimates where the error functions (the differences between the true solution and the approximations expressed by neural networks) are dominated by training loss functions. Numerical experiments are conducted to validate our error estimates, visualizing the relationship between the error and the training loss for various first-order differential equations and a second-order linear equation.

1. INTRODUCTION

Due to the development of computer capabilities, contemporary machine learning technologies have widespread applications in science engineering, and everyday life. Particularly, machine learning approaches are highly beneficial in tasks such as solving differential equations, image recognition, language processing, and statistical inference. Important mathematical ingredients in machine learning are neural networks, which consist of a sequential composition of linear functions and activation functions. Remarkably, neural networks are capable of uniformly approximating a continuous function defined on a compact set, which follows from the result called the universal approximation theorem (see [1, 2]). Aligned with this mathematical principle, neural networks have played a crucial role in approximating functions to describe phenomena that one seeks to analyze. By utilizing big data and formulating a loss function, one can train the parameters of a neural network via gradient descent, aiming to minimize the loss function.

Received March 3 2024; Revised March 24 2024; Accepted in revised form March 25 2024; Published online March 25 2024.

1991 *Mathematics Subject Classification.* 68T07, 34A12, 65L05.

Key words and phrases. neural networks, PINN, error estimates, existence, uniqueness, stability, initial value problems, differential equations.

† Corresponding author.

Recently, in applied mathematics, there has been a trend of using neural networks to approximate solutions to differential equations (see [3, 4, 5, 6]). Particularly if one desires to approximate the solution of a differential equation using neural networks, Physics-Informed Neural Networks (PINN) can be employed. The PINN method involves extracting points at random from the underlying domain to construct the neural network in a manner that forces to satisfy the differential equation (for more details, see [7, 8] and Section).

Let us briefly mention the development history of PINN. In [9], by transforming differential equations into finite difference equations, the authors trained the parameters of neural networks to solve finite difference equations transformed from differential equations. After that, in [10] using neural networks of the form $\phi\Big(W\mathbf{x}+\mathbf{b}\Big)$, the author studied the special type of homogeneous Dirichlet problem for the following Poisson equation:

$$\begin{cases} \Delta u(x,y) = \sin \pi x \sin \pi y & \text{in } [0,1]^2 \\ u = 0 & \text{on } \partial[0,1]^2. \end{cases}$$

The PINN methods for specific equations, such as initial value problems for ordinary differential equations and homogeneous Dirichlet problems for two-dimensional Poisson equations with general forcing terms, were proposed by [7]. In [5], it was confirmed that using TensorFlow for PINN is effectively used to approximate the 1-dimensional Burger's equation and 2-dimensional Navier-Stokes equations.

Utilizing PINN to obtain approximations for a solution to an initial value problem of a first-order differential equation gives significant advantages compared to traditional methods in numerical analysis such as Runge-Kutta methods. Firstly, the Runge-Kutta methods only provide values of approximations at each sampled point on a domain, making it challenging to obtain a closed form of the approximation in the full domain. Additionally, in the case of the Runge-Kutta methods, as the number of sample points increases, the computational cost becomes extensive (see [3, Section 1.5]). In contrast, approximations of a solution to a differential equation through PINN inherently provide a closed form composed of smooth functions. Consequently, there is no need for additional efforts in fitting the data at sample points, so that additional calculations on differentiation for approximations are easily employed. A notable advantage for approximations through PINN lies in the direct confirmation of how well the approximations satisfy the underlying differential equations, which is achieved through the evaluation of the training loss. On the other hand, since the Runge-Kutta methods only provide approximations at each sample point, it is difficult to check how well-fitted approximations satisfy the underlying differential equation.

Finding an approximation of a solution through PINN may have merit in cases where the regularity of the coefficients in differential equations is quite low. Particularly, we can expect error estimates based on loss functions. Consider, for example, the very simple differential equation for finding an anti-derivative function presented below:

$$\begin{cases} y'(t) = f(t), & t \in [0,T] \\ y(0) = 0, \end{cases} \tag{1.1}$$

where f is merely a continuous function on $[0, T]$. A basic approach to finding a solution $y(t)$ to (1.1) is to use the Fundamental Theorem of Calculus, and hence we obtain that

$$y(t) = \int_0^t f(s)dt, \qquad t \in [0, T].$$

However, calculating an approximation of $\int_0^t f(s)ds$ for each point t in $[0, T]$ requires lots of numerical computation effort as the number of t increases. On the other hand, using the Runge-Kutta method requires that f be at least four times differentiable. In that case, PINN II methods (see Section 2.2.1) may be an alternative method to find an approximation of $y(t)$. For a brief explanation, let us define a function N_A on $[0, T]$

$$N_A(t, \theta) = tN(t, \theta), \quad t \in [0, T],$$

where N is a 3-layer neural network defined as in (2.4). Let S be a set of random sample points in $[0, T]$ selected to follow a uniform distribution. Then, we define the loss function L_A by

$$L_A(\theta) := \left(\frac{1}{|S|} \sum_{t_i \in S} (N_A'(t_i, \theta) - f(t_i))^2 \right)^{1/2},$$

where $|S|$ is the number of elements in S. Now we can train the parameter θ to minimize L_A. Remarkably, the training loss function $L_A(\theta)$ and the error function $E_A(\theta)$ defined by

$$E_A(\theta) := \left(\frac{1}{|S|} \sum_{t_i \in S} (N_A(t_i, \theta) - y(t_i))^2 \right)^{1/2}$$

have a deep connection. Indeed, by the Hölder inequality and the Monte Carlo integration, for each $t \in [0, T]$ we obtain that

$$\begin{aligned}
|N_A(t, \theta) - y(t)| &\leq \int_0^T |N_A'(s, \theta) - y'(s)| ds \\
&\leq \sqrt{T} \left(\int_0^T |N_A'(s, \theta) - f(s)|^2 ds \right)^{1/2} \\
&\approx \sqrt{T} \left(\frac{T}{|S|} \sum_{s_i \in S} |N_A'(s_i, \theta) - f(s_i)|^2 \right)^{1/2} \\
&= T \left(\frac{1}{|S|} \sum_{s_i \in S} |N_A'(s_i, \theta) - f(s_i)|^2 \right)^{1/2} = TL_A(\theta).
\end{aligned}$$

Therefore, we can present the following inequality:

$$E_A(\theta) \leq TL_A(\theta), \quad \text{very likely.} \tag{1.2}$$

Though we only give the above mathematical analysis for error estimation in a very simple differential Eq. (1.1), the estimates (1.2) will be extended to more general cases including first-order (non-linear) equations and second-order linear equations, and a rigorous argument will be presented in Section 3 through basic calculus with some probabilistic arguments. In conclusion, we verify in **Error estimates 1-4** that the error functions (the differences between the true solution and the approximations expressed by neural networks) are dominated by training loss functions (see (3.2), (3.4), (3.10) and (3.13)). Then, by visualizing the relationship between the error and the training loss, we present some numerical experiments in Section 4 to validate the error estimates in Section 3 in various initial value problems for ordinary differential equations. In particular, we will distinguish between PINN methods as PINN I and PINN II (see Section 2.2.2). PINN I utilizes conventional real-valued neural networks, while PINN II trains the parameter θ by incorporating appropriate terms into real-valued neural networks, such as N_A and $\overline{N}_A$ in (2.7) and (3.11), respectively, to enforce the satisfaction of initial conditions for the ordinary differential equations. Both approaches (PINN I and II) have been verified to yield error estimates effectively (see Section 4.1).

The main purpose of this paper is to review PINN applied to initial value problems for ordinary differential equations and to derive error estimates through training loss by using basic calculus. Hence, we not only supplement the results of [11] which derived error estimates using the non-trivial stability results in [12], but also present a more accessible proof to readers. This paper is structured as follows: In the next section, we explain the theoretical aspect of the Picard iteration for the existence and uniqueness of solutions to initial value problems for ordinary differential Eqs. (2.1). Then, in Section 2.2, we discuss the basic concepts for Physics-Informed Neural Networks (PINN) applied to the initial value problems for the ordinary differential equations and briefly review in Section 2.3 the existing literature studying the error estimates of PINN for initial value problems for first-order ordinary differential equations. In Section 3, we show that error functions are dominated by training loss functions by using basic calculus and probabilistic arguments. In Section 4, we present some numerical experiments in the cases of an antiderivative, a logistic equation, a separable equation, an exact equation, and a second-order equation with constant coefficients. In the last section, we describe a summary of our numerical experiments and a brief outlook on the error estimates of PINN.

2. THEORETICAL BACKGROUND

2.1. **Existence, uniqueness and stability by the Picard Iteration.**

Let us consider the following initial value problem of the ODE:

$$\begin{cases} y'(t) = f(t, y(t)) \\ y(t_0) = y_0, \end{cases} \tag{2.1}$$

where $(t_0, y_0) \in \mathbb{R}^2$ and f is a Lipschitz continuous function on

$$[t_0 - a_1, t_0 + a_2] \times [y_0 - b_1, y_0 + b_2]$$

satisfying that for some $M_0, K > 0$

$$\begin{aligned}
&|f(t,y)| \le M_0, \quad \text{for all } (t,y) \in [t_0 - a_1, t_0 + a_2] \times [y_0 - b_1, y_0 + b_2].\\
&|f(t,y_1) - f(t,y_2)| \le K|y_1 - y_2|\\
&\text{for all } (t,y_1),(t,y_2) \in [t_0 - a_1, t_0 + a_2] \times [y_0 - b_1, y_0 + b_2]. \qquad (2.2)
\end{aligned}$$

Let $a = \min(a_1, a_2)$ and $b = \min(b_1, b_2)$. Let $J := [t_0 - c, t_0 + c]$ with $c = \min(a, \frac{b}{M_0})$. Then, it is known from Picard iteration (See [13, Chapter 5, Section 4]) that there exists a continuously differentiable function ϕ on J such that ϕ is a unique solution to (2.1) on J, i.e.

$$\phi'(t) = f(t, \phi(t)) \text{ for all } t \in J \text{ and } \phi(t_0) = y_0$$

and if there exists a continuously differentiable function ψ on I satisfying (2.1) on I where I is a compact interval in $[t_0 - a_1, t_0 + a_2]$ with $t_0 \in I$, then $\phi(t) = \psi(t)$ for all $t \in I \cap J$ (see [13, Chapter 5, Section 8]). Indeed, a sequence of continuously differentiable functions on J which converges to u uniformly on J is defined recursively by

$$\phi_0(t) = y_0, \qquad \phi_{k+1} = y_0 + \int_{t_0}^{t} f(s, \phi_k(s))ds, \quad t \in J.$$

Moreover, the following stability estimate is obtained by [13, Chapter 5, Theorem 8]:

$$|\phi(t) - \phi_k(t)| \le \frac{M}{K}\frac{(Kc)^{k+1}}{((k+1)!)}e^{Kc} \quad \text{for all } t \in J. \qquad (2.3)$$

Certainly, the error estimate in (2.3) rapidly decreases to 0 as $k \to \infty$. However, since ϕ_k is composed of k-th iterated integrals, it requires a large computational cost to calculate ϕ_k numerically. Therefore, it leads us to use numerical methods such as Euler and Runge-Kutta methods. Finally, we mention that if (2.2) is replaced by

$$|f(t,y_1) - f(t,y_2)| \le K|y_1 - y_2| \quad \text{for all } (t,y_1),(t,y_2) \in [t_0 - a, t_0 + a] \times \mathbb{R},$$

then J can be replaced by $[t_0 - a_1, t_0 + a_2]$ (see [13, Chapter 5]).

2.2. **Physics-Informed Neural Networks (PINN).**

2.2.1. *Approximations through neural networks.*

In this section, we investigate the main idea for approximating a function through neural networks in machine learning. A neural network is inspired by a human brain, but mathematically a neural network is just defined as a function that consists of a sequential composition of linear functions and activation functions. In particular, the 3-layer neural network is practically and widely used in machine learning. If we apply the 3-layer neural network to approximate a real-valued function defined on a subset of $\mathbb{R}$, then we can consider a function $N : \mathbb{R} \to \mathbb{R}$ defined by

$$N(t,\theta) = A_3\sigma(A_2\sigma(A_1 t + b_1) + b_2) + b_3, \quad t \in \mathbb{R}, \qquad (2.4)$$

where A_1, A_2 and A_3 are $n_1 \times 1$, $n_2 \times n_1$ and $1 \times n_2$ matrices, respectively, with $n_1, n_2 \in \mathbb{N}$, b_1, b_2 and b_3 are $n_1 \times 1$, $n_2 \times 1$ and 1×1 matrices (column vectors), respectively, σ

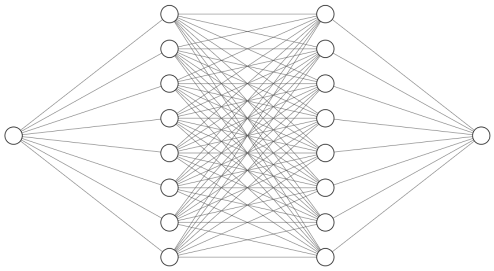

FIGURE 1. A neural network

is a non-linear elementwise activation function and in this case, σ denotes $\tanh$ and $\theta = \theta(n_1, n_2, A_1, A_2, A_3, b_1, b_2, b_3)$ denotes a variable made up of components of A_1, A_2, A_3, b_1, b_2 and b_3. Figure 1 represents a neural network.

The fact that a neural network is expected to be a good approximation of the function we are looking for is based on a mathematical theorem in functional analysis, which is called *"universal approximation theorem"* (see [1, 2]). For instance, let g be an arbitrarily given continuous function on a compact interval I we are looking for. Then, by the universal approximation theorem for 3-layer neural networks ([2, Theorem 12]), given $\varepsilon > 0$ there exist $n_1, n_2 \in \mathbb{N}$, $n_1 \times 1$, $n_2 \times n_1$, $1 \times n_2$ matrices A_1, A_2 and A_3, respectively, and $n_1 \times 1$, $n_2 \times 1$ and 1×1 matrices b_1, b_2 and b_3, respectively, such that

$$|N(t, \theta) - g(t)| < \varepsilon, \qquad \text{for all } t \in I,$$

where $\theta = \theta(A_1, A_2, A_3, b_1, b_2, b_3)$. Therefore, it is theoretically verified by a mathematical theorem that the approximation to the function g we are looking for is represented by a neural network. Therefore, we need to find an algorithm that can find the variable θ that provides an approximation. If we fix n_1 and n_2 large enough, now what we have to find is the components of A_1, A_2, A_3, b_1, b_2, b_3. A standard method to find neural networks to approximate g is to use big data to fit f. For instance, let us assume that for some large $n \in \mathbb{N}$ we have data set $D = \{(t_i, y_i)_{g \le i \le n}\}$ satisfying $y_i = g(t_i)$ for all $1 \le i \le n$. Then we can now train θ so that neural network $N(\cdot, \theta)$ satisfies data set D. Precisely, we define the loss function below:

$$\text{Loss}(\theta) := \left(\frac{1}{n} \sum_{i=1}^{n} \big(N(t_i, \theta) - y_i \big)^2 \right)^{1/2}.$$

Using computer programming such as Python, θ can be trained in the direction of minimizing the loss function through a gradient descent algorithm. This is the whole idea of training a neural network through big data in machine learning to approximate the real function we are looking for.

Now, let us use the neural network (2.4) to solve the initial value problem (2.1). Note that there is no data that the solution fits, but we have physical information expressed as an ordinary differential equation. Therefore, without given data, we can define a loss function in such a way that the neural network satisfies our differential equation. We will discuss this in detail in the next subsection.

2.2.2. *Two ways for approximating the solution to* (2.1) *in* PINN.
Now let us consider the initial value problem (2.1) and approximate the solution to (2.1) through neural networks.

• **PINN I**: First, consider a neural network N as in (2.4) and define loss functions L_{de} and L_{ic} given by

$$L_{de}(\theta) := \left(\frac{1}{n}\sum_{i=1}^{n}(N'(t_i,\theta) - f(t_i, N(t_i,\theta)))^2\right)^{1/2}, \quad L_{ic}(\theta) := |N(t_0,\theta) - y_0|, \tag{2.5}$$

where $S = \{t_1, \dots, t_n\}$ is a set of random sample points of an interval on which the solution is defined and the sample points are selected to follow a uniform distribution. Then our total loss function is

$$L(\theta) := L_{de}(\theta) + L_{ic}(\theta) \tag{2.6}$$

and we train the parameters θ of L for which L is minimized by using the gradient descent. Here, we mention that the loss function does not exactly mean the errors that denote the difference between the true solution and the neural networks. However, in the next section (Section 3), we will mathematically verify the very close relationship between the loss functions and error functions.

• **PINN II**: As in PINN I above, we can see that it is very unlikely that either L_{de} or L_{ic} will be exactly 0. Thus, one can alternatively consider a function N_A defined by

$$N_A(t,\theta) := y_0 + (t - t_0)N(t,\theta), \quad t \in \mathbb{R}, \tag{2.7}$$

where N is a neural network defined as in (2.4). Note that $N_A(t_0,\theta) = y_0$, so that the initial condition of (2.1) is always satisfied. Before defining a new loss function, let us check whether the alternative function N_A above can be a good uniform approximation for the solution to (2.1) from the perspective of the universal approximation theorem. Observe that by the existence and uniqueness theorem, the solution y to (2.1) is continuously differentiable on its domain containing t_0 (see Section 2.1). Now let I be a compact interval in the domain of y with $t_0 \in I$ and define a function z on I

$$\begin{cases} z(t) = \dfrac{y(t) - y_0}{t - t_0} & \text{if } z \in I \setminus \{t_0\} \\ z(t_0) = y'(t_0). & \end{cases}$$

Then, z is continuous on I and

$$y(t) = y_0 + (t - t_0)z(t) \quad \text{for all} \ \ t \in I.$$

By the universal approximation theorem for 3-layer neural networks ([2, Theorem 12]), given $\varepsilon > 0$ there exist $n_1, n_2 \in \mathbb{N}$, $n_1 \times 1$, $n_2 \times n_1$, $1 \times n_2$ matrices A_1, A_2 and A_3, respectively and $n_1 \times 1$, $n_2 \times 1$ and 1×1 matrices b_1, b_2, b_3, respectively, such that if $\theta = \theta(n_1, n_2, A_1, A_2, A_3, b_1, b_2, b_3)$, then

$$|N(t,\theta) - z(t)| < \frac{\varepsilon}{|I|}, \quad \text{for all } t \in I,$$

where $|I|$ is the length of the interval I. Thus, we obtain that

$$|N_A(t,\theta) - y(t)| = \Big|(t - t_0)\Big(N(t,\theta) - z(t)\Big)\Big| \leq |I| \cdot |N(t,\theta) - z(t)| < \varepsilon \quad \text{for all } t \in I.$$

Thus, we can regard $N_A(\cdot, \theta)$ as a good approximation of y in the perspective of the universal approximation theorem. Now we define the alternative loss function L_A in terms of N_A:

$$L_A(\theta) = \left(\frac{1}{n}\sum_{i=1}^{n}\big(N_A'(t_i,\theta) - f(t_i, N_A(t_i,\theta))\big)^2\right)^{1/2}, \tag{2.8}$$

where $S = \{t_1, \ldots, t_n\}$ is a set of random sample points of an interval on which the solution is defined and the sample points are selected to follow a uniform distribution. Then, we train the parameter θ of L_A for which L_A is minimized by using the gradient descent.

2.3. **The existing literature and our strategy for error estimates.**

The existing literature [12] deals with the error estimates for PINN of first-order differential equations by using the following stability estimates. Here, we restrict the estimates to real-valued functions.

Theorem 2.1 ([12]). *If y is a solution to* (2.1) *on a closed interval $I := [0, c]$ with $t_0 = 0$ and $\widehat{y}$ is an arbitrarily given C^1-function on I, then the following estimate holds:*

$$|y(t) - \widehat{y}(t)| \leq e^{Kt}|y_0 - \widehat{y}(0)| + \int_0^t e^{K(t-s)}\big|\widehat{y}'(s) - f(s, \widehat{y}(s))\big|\, ds, \quad \text{for all } t \in I. \tag{2.9}$$

The proof of Theorem 2.1 is strongly based on the result of [12, Chater I, Variant form of Theorem 10.2], but unfortunately the proof of [12, Chater I, Variant form of Theorem 10.2] is not explicitly presented for readers. Even though the authors of [12] may easily derive it, [12, Chater I, Variant form of Theorem 10.2] is closely related to [12, Chater I, Theorem 10.2] which is also a nontrivial result based on the observation of the mathematicians, Peano and Perron. Therefore, we will derive in the next section the known stability estimates using only basic calculus, which gives an accessible proof of error estimates to readers. Specifically, removing the weight term $e^K(t - s)$ in the right-hand side of (2.9), we will derive in Theorem 3.3

$$|y(t) - \widehat{y}(t)| \leq e^{Kt}\left(|y_0 - \widehat{y}(0)| + \int_0^t \big|\widehat{y}'(s) - f(s, \widehat{y}(s))\big|\, ds\right), \quad \text{for all } t \in I. \tag{2.10}$$

Moreover, in Section 3.2 we also present error estimates for the second-order linear equations. In this paper, the arbitrarily given C^1-function $\widehat{y}$ is replaced by our neural networks $N(\cdot,\theta)$ or alternative functions $N_A(\cdot,\theta)$ defined in Section 2.2.2. This replacement is based on the expectation that the right-hand side of (2.10) where $\widehat{y}$ is replaced by $N(\cdot,\theta)$ or $N_A(\cdot,\theta)$ is sufficiently to be small as θ is trained, and as a result, the error in the left-hand side of (2.10) will also be sufficiently small. Moreover, in Section 3.2 we derive error estimates for initial value problems of second-order linear differential equations by using stability estimates for $\mathbb{R}^2$-valued functions. Indeed, the actual training loss function is expressed in the form of a square mean, and hence we need to convert the loss function expressed as an integral into the actual training loss expressed in the form of a square mean. To do this, we present basic probabilistic arguments based on Monte Carlo integration in Theorems 3.2, 3.4, 3.6 and 3.8.

3. Error estimates through the training loss functions

3.1. **Initial value problems for the first order ODEs.**

Given an interval $J=[a,b]$, we write $|J|=b-a$. Theorems 3.1, 3.2 are mathematical results to derive error estimates (3.2) for PINN II.

Theorem 3.1. *Let* $(t_0,y_0)\in\mathbb{R}^2$ *and* $a_1,a_2,b_1,b_2>0$. *Let* f *be a continuous function on* $[t_0-a_1,t_0+a_2]\times[y_0-b_1,y_0+b_2]$ *such that for some* $M_0,K>0$ (2.2) *holds. Let* N_A *be a function defined as in* (2.7). *Let* y *be a unique solution to* (2.1) *on* $I:=[t_0-c_1,t_0+c_2]$ *for some* $c_1,c_2>0$. *Let*

$$\mathcal{L}_A(\theta):=\int_I |N_A'(s,\theta)-f(s,N_A(s,\theta))|ds, \tag{3.1}$$

where N_A *is defined as in* (2.7). *Then,*

$$|N_A(t,\theta)-y(t)|\le e^{K|t-t_0|}\mathcal{L}_A(\theta)\quad \textit{for all } t\in I.$$

Proof. Let us first consider the case of $t\in[t_0,t_0+c_2]$. Using the fundamental theorem of calculus and the triangle inequality,

$$\begin{aligned}|N_A(t,\theta)-y(t)| &= \left|\int_{t_0}^t N_A'(s,\theta)-y'(s)ds\right| \le \int_{t_0}^t \left|N_A'(s,\theta)-y'(s)\right| ds\\ &\le \int_{t_0}^t |N_A'(s,\theta)-f(s,N_A(s,\theta))|ds+\int_{t_0}^t \Big|f(s,N_A(s,\theta))-f(s,y(s))\Big|ds\\ &\le \mathcal{L}_A(\theta)+K\int_{t_0}^t \big|N_A(s,\theta)-y(s)\big|ds,\qquad \text{for all } t\in[t_0,t_0+c_2].\end{aligned}$$

Let

$$\varphi(t):=\mathcal{L}_A(\theta)+K\int_{t_0}^t \big|N_A(s,\theta)-y(s)\big|ds,\quad t\in[t_0,t_0+c_2].$$

Then, $\frac{1}{K}\varphi'(t) \le \varphi(t)$ for all $t \in [t_0, t_0 + c_2]$, so that

$$\left(e^{-Kt}\varphi(t)\right)' \le 0, \qquad \text{for all } t \in [t_0, t_0 + c_2].$$

Since the map $t \mapsto e^{-Kt}\varphi(t)$ is decreasing on $[t_0, t_0 + c_2]$, $e^{-Kt}\varphi(t) \le e^{-Kt_0}\varphi(t_0)$ for all $t \in [t_0, t_0 + c_2]$. Therefore, we obtain that

$$|N_A(t,\theta) - y(t)| \le \varphi(t) \le e^{K(t-t_0)}\varphi(t_0) = e^{K(t-t_0)}\mathcal{L}_A(\theta) \quad \text{for all } t \in [t_0, t_0 + c_2].$$

Next, consider the case of $t \in [t_0 - c_1, t_0]$. Then, similarly to the above, it holds that

$$|N_A(t,\theta) - y(t)| \le \mathcal{L}_A(\theta) - K\int_{t_0}^{t} \big|N_A(s,\theta) - y(s)\big|ds, \qquad \text{for all } t \in [t_0 - c_1, t_0].$$

Then, defining

$$\psi(t) := \mathcal{L}_A(\theta) - K\int_{t_0}^{t} \big|N_A(s,\theta) - y(s)\big|ds, \quad t \in [t_0 - c_1, t_0],$$

we have $-\frac{1}{K}\psi'(t) \le \psi(t)$ for all $t \in [t_0 - c_1, t_0]$, so that

$$\left(e^{Kt}\psi(t)\right)' \ge 0 \quad \text{for all } t \in [t_0 - c_1, t_0].$$

Thus, we finally have

$$|N_A(t,\theta) - y(t)| \le e^{K(t_0-t)}\psi(t_0) = e^{K(t_0-t)}\mathcal{L}_A(\theta), \qquad \text{for all } t \in [t_0 - c_1, t_0],$$

and hence the assertion follows. □

The following result is derived by replacing the integral term in (3.1) with the Monte Carlo Integration.

Theorem 3.2. *Assume that the conditions of Theorem 3.1 hold. Let $(X_i)_{i\ge1}$ be a sequence of independent and identically distributed random variables on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ that has a continuous uniform distribution on I. Then, for any $t \in I$ the following estimate holds:*

$$|N_A(t,\theta) - y(t)| \le |I|e^{K|t-t_0|}\left(\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\Big(N_A'(X_i,\theta) - f(X_i, N_A(X_i,\theta))\Big)^2\right]^{1/2} + \frac{\sigma}{\sqrt{n}}\right),$$

where N_A is defined as in (2.7), $\sigma := \left(\frac{1}{|I|}\int_I \varphi^2 dt - \left(\frac{1}{|I|}\int_I \varphi dt\right)^2\right)^{1/2} \ge 0$ and

$$\varphi(t) := |N_A'(t,\theta) - f(t, N_A(t,\theta))|, \quad t \in I.$$

Proof. Let $Y_i = \varphi(X_i)$, $i \ge 1$. Then, $(Y_i)_{i\ge1}$ is a sequence of independent and identically distributed random variables satisfying that

$$\mathbb{E}[Y_i] = \frac{1}{|I|}\int_I \varphi(t)dt =: m, \quad \mathbb{V}[Y_i] = \frac{1}{|I|}\int_I \varphi^2 dt - \left(\frac{1}{|I|}\int_I \varphi dt\right)^2 = \sigma^2,$$

where $\mathbb{E}$ and $\mathbb{V}$ are the expectation and variance with respect to $(\Omega, \mathcal{F}, \mathbb{P})$. Let

$$\overline{Y}_n = \frac{1}{n}\sum_{i=1}^{n} Y_i, \quad n \geq 1.$$

Then, $\mathbb{E}[\overline{Y}_n] = m$ and $\mathbb{V}(\overline{Y}_n) = \frac{\sigma^2}{n}$, and hence we have

$$\|\overline{Y}_n - m\|_{L^2(\Omega,\mathbb{P})} = \frac{\sigma}{\sqrt{n}}.$$

Let $\mathcal{L}_A(\theta)$ be a function defined in (3.1). Then,

$$\begin{aligned}\frac{1}{|I|}\mathcal{L}_A(\theta) &= m = \|m\|_{L^2(\Omega,\mathbb{P})} \leq \|m - \overline{Y}_n\|_{L^2(\Omega,\mathbb{P})} + \|\overline{Y}_n\|_{L^2(\Omega,\mathbb{P})} \\ &= \frac{\sigma}{\sqrt{n}} + \mathbb{E}\left[\left(\frac{1}{n}\sum_{i=1}^{n} Y_i\right)^2\right]^{1/2} \underset{\text{by Jensen}}{\leq} \frac{\sigma}{\sqrt{n}} + \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} Y_i^2\right]^{1/2}.\end{aligned}$$

Thus, the assertion follows from Lemma 3.1. □

Assume that the conditions of Theorem 3.1 hold. Then, based on Theorem 3.2, the following error estimates hold:

Error estimates 1 (PINN II)

$$|N_A(t,\theta) - y(t)| \leq |I|e^{K|t-t_0|} \cdot L_A(\theta) + O\left(\frac{1}{\sqrt{n}}\right), \quad \text{for all } t \in I \quad \text{very likely,} \tag{3.2}$$

where N_A is defined as in (2.7).

Theorems 3.3, 3.4 are mathematical results to derive Error estimates (3.4) for PINN I.

Theorem 3.3. *Let* $(t_0, y_0) \in \mathbb{R}^2$ *and* $a_1, a_2, b_1, b_2 > 0$. *Let* f *be a continuous function on* $[t_0 - a_1, t_0 + a_2] \times [y_0 - b_1, y_0 + b_2]$ *such that for some* $M_0, K > 0$ (2.2) *holds. Let* y *be a unique solution to* (2.1) *on* $I := [t_0 - c_1, t_0 + c_2]$ *with* $c_1, c_2 > 0$. *Let*

$$\mathcal{L}(\theta) := \int_I |N'(s,\theta) - f(s, N(s,\theta))|ds + |N(t_0,\theta) - y_0|, \tag{3.3}$$

where N *is a neural network defined as in* (2.4). *Then,*

$$|N(t,\theta) - y(t)| \leq e^{K|t-t_0|}\mathcal{L}(\theta) \quad \textit{for all } t \in I.$$

Proof. Let us first consider the case of $t \in [t_0, t_0 + c_2]$. Using the fundamental theorem of calculus and the triangle inequality,

$$
\begin{aligned}
&|N(t,\theta) - y(t)| = \left| \int_{t_0}^{t} N'(s,\theta) - y'(s)ds + (N(t_0,\theta) - y(t_0)) \right| \\
&\leq |(N(t_0,\theta) - y(t_0)| + \int_{t_0}^{t} \left| N'(s,\theta) - y'(s) \right| ds \\
&\leq |(N(t_0,\theta) - y(t_0)| + \int_{t_0}^{t} |N'(s,\theta) - f(s, N(s,\theta))| ds + \int_{t_0}^{t} \Big| f(s, N(s,\theta)) - f(s, y(s)) \Big| ds \\
&\leq \mathcal{L}(\theta) + K \int_{t_0}^{t} \big| N(s,\theta) - y(s) \big| ds, \qquad \text{for all } t \in [t_0, t_0 + c_2].
\end{aligned}
$$

Then, analogously to the proof of Theorem 3.1, the assertion follows. □

As we derive Theorem 3.2 based on Theorem 3.1, we can similarly obtain the following result.

Theorem 3.4. *Assume that the conditions of Theorem 3.3 hold. Let $(X_i)_{i\geq 1}$ be a sequence of independent and identically distributed random variables on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ that has a continuous uniform distribution on I. Then, for any $t \in I$ the following estimate holds:*

$$
\begin{aligned}
&|N(t,\theta) - y(t)| \\
&\leq |I| e^{K|t-t_0|} \left(\mathbb{E}\left[\frac{1}{n} \sum_{i=1}^{n} \Big(N'(X_i,\theta) - f(X_i, N(X_i,\theta)) \Big)^2 \right]^{1/2} + \frac{|N(t_0) - y_0|}{|I|} + \frac{\sigma}{\sqrt{n}} \right),
\end{aligned}
$$

where N is a neural network defined as in (2.4)*, $\sigma := \left(\frac{1}{|I|} \int_I \phi^2 dt - \left(\frac{1}{|I|} \int_I \phi dt \right)^2 \right)^{1/2} \geq 0$ and*

$$
\phi(t) := |N'(t,\theta) - f(t, N(t,\theta))|, \quad t \in I.
$$

Proof. The proof is similar to the one of Theorem 3.2. Let $Y_i = \phi(X_i)$, $i \geq 1$. Then, $(Y_i)_{i\geq 1}$ is a sequence of independent and identically distributed random variables satisfying that

$$
\mathbb{E}[Y_i] = \frac{1}{|I|} \int_I \phi(t) dt =: m, \quad \mathbb{V}[Y_i] = \frac{1}{|I|} \int_I \phi^2 dt - \left(\frac{1}{|I|} \int_I \phi dt \right)^2 = \sigma^2,
$$

where $\mathbb{E}$ and $\mathbb{V}$ are the expectation and variance with respect to $(\Omega, \mathcal{F}, \mathbb{P})$. Let

$$
\overline{Y}_n = \frac{1}{n} \sum_{i=1}^{n} Y_i, \quad n \geq 1.
$$

Then, $\mathbb{E}[\overline{Y}_n] = m$ and $\mathbb{V}(\overline{Y}_n) = \frac{\sigma^2}{n}$, and hence we have

$$
\|\overline{Y}_n - m\|_{L^2(\Omega,\mathbb{P})} = \frac{\sigma}{\sqrt{n}}.
$$

Let $\mathcal{L}(\theta)$ be defined as in (3.3). Then,

$$\frac{1}{|I|}\Big(\mathcal{L}(\theta) - |(N(t_0,\theta) - y_0)|\Big) = m = \|m\|_{L^2(\Omega,\mathbb{P})} \leq \|m - \overline{Y}_n\|_{L^2(\Omega,\mathbb{P})} + \|\overline{Y}_n\|_{L^2(\Omega,\mathbb{P})}$$
$$= \frac{\sigma}{\sqrt{n}} + \mathbb{E}\left[\left(\frac{1}{n}\sum_{i=1}^{n} Y_i\right)^2\right]^{1/2} \underset{\text{by Jensen}}{\leq} \frac{\sigma}{\sqrt{n}} + \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} Y_i^2\right]^{1/2},$$

as desired. □

Assume that the conditions of Theorem 3.3 hold and that $|I| \geq 1$. Then, based on Theorem 3.4, the following error estimates hold:

Error estimates 2 (PINN I)

$$|N(t,\theta) - y(t)| \leq |I|e^{K|t-t_0|} \cdot L(\theta) + O\left(\frac{1}{\sqrt{n}}\right), \quad \text{for all } t \in I \quad \text{very likely}, \tag{3.4}$$

where N is a neural network defined as in (2.4) and L is defined as in (2.6).

3.2. Initial value problems for the second order ODEs.

Now, let us further investigate error estimates for initial value problems for second-order ordinary differential equations of the following form:

$$\begin{cases} y''(t) + p(t)y'(t) + q(t)y(t) = r(t) \\ y(t_0) = y_0, \quad y'(t_0) = y_0' \end{cases} \tag{3.5}$$

where $(t_0, y_0, y_0') \in \mathbb{R}^3$ and p, q and r are continuous functions on a compact interval I with $t_0 \in I$. Then, as a direct consequence of the existence and uniqueness theorem for initial value problems for first-order differential equations, there exists a unique solution $y \in C^2(I)$ to (3.5) on I. Indeed, (3.5) is converted to the following problem:

$$\begin{cases} \begin{pmatrix} y_1'(t) \\ y_2'(t) \end{pmatrix} = \begin{pmatrix} y_2(t) \\ -p(t)y_2(t) - q(t)y_1(t) + r(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -q(t) & -p(t) \end{pmatrix} \begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} + \begin{pmatrix} 0 \\ r(t) \end{pmatrix} \\ (y_1(t_0), y_2(t_0)) = (y_0, y_0') \end{cases}$$

We can rewrite the above equation as follows:

$$\begin{cases} \mathbf{y}'(t) = P(t)\mathbf{y}(t) + \mathbf{g}(t) \\ \mathbf{y}(t_0) = \begin{pmatrix} y_0 \\ y_0' \end{pmatrix}, \end{cases}$$

where $P(t) = \begin{pmatrix} 0 & 1 \\ -q(t) & -p(t) \end{pmatrix}$ and $\mathbf{g}(t) = \begin{pmatrix} 0 \\ r(t) \end{pmatrix}$.

The idea of converting second-order differential equations into first-order differential equations is also crucial for deriving the following error estimates of PINN.

Theorems 3.5, 3.6 are mathematical results to derive Error estimates (3.10) for PINN I.

Theorem 3.5. *Let p, q and r be continuous functions on a compact interval $I := [t_0 - c_1, t_0 + c_2]$ with $c_1, c_2 > 0$. Let y be a unique solution to* (3.5) *on I. Let N be a neural network defined as in* (2.4)*. Let $M = \sqrt{1 + M_1^2 + M_2^2}$, where $M_1 := \max_{t\in I} |p(t)|$, $M_2 := \max_{t\in I} |q(t)|$. Define $\overline{\mathcal{L}}(\theta)$ by*

$$\overline{\mathcal{L}}(\theta) := \int_I |N''(s,\theta)+p(s)N'(s,\theta)+q(s)N(s,\theta)-r(s)|ds+|N(t_0,\theta)-y_0|+|N'(t_0,\theta)-y_0'|,$$

where N is a neural network defined as in (2.4)*. Then,*

$$|N(t,\theta) - y(t)| \le e^{M|t-t_0|}\overline{\mathcal{L}}(\theta), \quad \textit{for all } t \in I.$$

Proof. Let $x(t) = N(t,\theta) - y(t)$ and $h(t) = N''(t,\theta) + p(t)N'(t,\theta) + q(t)N(t,\theta)$, $t \in I$. Then, we get

$$\begin{cases} x''(t) + p(t)x'(t) + q(t)x(t) = h(t) - r(t), & \text{for all } t \in I \\ x(t_0) = N(t_0,\theta) - y_0, \;\; x'(t_0) = N'(t_0,\theta) - y_0'. \end{cases} \tag{3.6}$$

Let $\mathbf{x}(t) = \begin{pmatrix} x(t) \\ x'(t) \end{pmatrix}$, $t \in I$. Then, the Eq. (3.6) is converted to the following:

$$\begin{cases} \mathbf{x}'(t) = P(t)\mathbf{x}(t) + \mathbf{f}(t), & \text{for all } t \in I \\ \mathbf{x}(t_0) = \begin{pmatrix} N(t_0,\theta) - y_0 \\ N'(t_0,\theta) - y_0' \end{pmatrix}, \end{cases}$$

where $P(t) = \begin{pmatrix} 0 & 1 \\ -q(t) & -p(t) \end{pmatrix}$ and $\mathbf{f}(t) = \begin{pmatrix} 0 \\ h(t) - r(t) \end{pmatrix}$. Thus, by the fundamental theorem of calculus,

$$\mathbf{x}(t) = \int_{t_0}^{t} \mathbf{x}'(s)ds + \mathbf{x}(t_0).$$

First, consider the case of $t \in [t_0, t_0 + c_2]$. Then,

$$\begin{aligned} \|\mathbf{x}(t)\| &\le \left\| \int_{t_0}^{t} \mathbf{x}'(s)ds + \mathbf{x}(t_0) \right\| \le \int_{t_0}^{t} \|\mathbf{x}'(s)\|ds + \|\mathbf{x}(t_0)\| \\ &\le \int_{t_0}^{t} \|P(s)\|\|\mathbf{x}(s)\|ds + \int_{0}^{t} \|\mathbf{f}(s)\|ds + \|\mathbf{x}(t_0)\| \\ &\le M \int_{t_0}^{t} \|\mathbf{x}(s)\|ds + \overline{\mathcal{L}}(\theta), \quad \text{for all } t \in [t_0, t_0 + c_2] \end{aligned}$$

Then, using the analogous method in the proof of Theorem 3.1, we get

$$\|\mathbf{x}(t)\| \le e^{M(t-t_0)}\overline{\mathcal{L}}(\theta), \quad \text{for all } t \in [t_0, t_0 + c_2].$$

Next, for the case of $t \in [t_0 - c_1, t_0]$, we have

$$\|\mathbf{x}(t)\| \le -M\int_{t_0}^{t} \|\mathbf{x}(s)\|ds + \overline{\mathcal{L}}(\theta), \quad \text{for all } t \in [t_0 - c_2, t_0].$$

Analogously to the proof of Theorem 3.1, it follows that

$$\|\mathbf{x}(t)\| \le e^{M(t_0-t)}\overline{\mathcal{L}}(\theta), \quad \text{for all } t \in [t_0 - c_1, t_0].$$

Therefore, we finally get

$$|N(t,\theta) - y(t)| = |x(t)| \le \|\mathbf{x}(t)\| \le e^{M|t-t_0|}\overline{\mathcal{L}}(\theta), \quad \text{for all } t \in I.$$

□

In a similar way to the proof of Theorem 3.4 based on Theorem 3.3, we derive the following theorem by using Theorem 3.5.

Theorem 3.6. *Assume that the conditions of Theorem 3.5 hold. Let $(X_i)_{i\ge1}$ be a sequence of independent and identically distributed random variables on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ that has a continuous uniform distribution on I. Then, for any $t \in I$ the following estimate holds:*

$$\begin{aligned}&|N(t,\theta) - y(t)| \\ &\le |I|e^{K|t-t_0|}\left(\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\Big(N''(X_i,\theta) + p(X_i)N'(X_i,\theta) + q(X_i)N(X_i,\theta) - r(X_i)\Big)^2\right]^{1/2}\right. \\ &\qquad\qquad \left. + \frac{|N(t_0,\theta) - y_0| + |N(t_0,\theta) - y_0'|}{|I|} + \frac{\sigma}{\sqrt{n}}\right),\end{aligned}$$

where N is a neural network defined as in (2.4), $\sigma := \left(\frac{1}{|I|}\int_I \phi^2 dt - \left(\frac{1}{|I|}\int_I \phi dt\right)^2\right)^{1/2} \ge 0$ *and*

$$\phi(t) := \Big|N''(t,\theta) + p(t)N'(t,\theta) + q(t)N(t,\theta) - r(t)\Big|, \quad t \in I.$$

Proof. Let $Y_i = \phi(X_i)$, $i \ge 1$. Then, $(Y_i)_{i\ge1}$ is a sequence of independent and identically distributed random variables satisfying that

$$\mathbb{E}[Y_i] = \frac{1}{|I|}\int_I \phi(t)dt =: m, \quad \mathbb{V}[Y_i] = \frac{1}{|I|}\int_I \phi^2 dt - \left(\frac{1}{|I|}\int_I \phi dt\right)^2 = \sigma^2,$$

where $\mathbb{E}$ and $\mathbb{V}$ are the expectation and variance with respect to $(\Omega, \mathcal{F}, \mathbb{P})$. Let

$$\overline{Y}_n = \frac{1}{n}\sum_{i=1}^{n} Y_i, \quad n \ge 1.$$

Then, $\mathbb{E}[\overline{Y}_n] = m$ and $\mathbb{V}(\overline{Y}_n) = \frac{\sigma^2}{n}$, and hence we have

$$\|\overline{Y}_n - m\|_{L^2(\Omega,\mathbb{P})} = \frac{\sigma}{\sqrt{n}}.$$

Let $\overline{\mathcal{L}}(\theta)$ be defined as in (3.3). Then,

$$\begin{aligned}
&\frac{1}{|I|}\Big(\overline{\mathcal{L}}(\theta) - |(N(t_0,\theta) - y_0| - |N'(t_0,\theta) - y_0'|\Big) = m = \|m\|_{L^2(\Omega,\mathbb{P})} \qquad (3.7)\\
&\le \|m - \overline{Y}_n\|_{L^2(\Omega,\mathbb{P})} + \|\overline{Y}_n\|_{L^2(\Omega,\mathbb{P})}\\
&= \frac{\sigma}{\sqrt{n}} + \mathbb{E}\left[\left(\frac{1}{n}\sum_{i=1}^{n} Y_i\right)^2\right]^{1/2} \underset{\text{by Jensen}}{\le} \frac{\sigma}{\sqrt{n}} + \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} Y_i^2\right]^{1/2},
\end{aligned}$$

as desired.

□

Now define a loss function $\overline{L}(\theta)$ corresponding to the problem (3.5) and the neural network N defined as in (2.4):

$$\overline{L}(\theta) := \overline{L}_{de}(\theta) + |N(t_0,\theta) - y_0| + |N'(t_0,\theta) - y_0'|, \qquad (3.8)$$

where

$$\overline{L}_{de}(\theta) := \left[\frac{1}{n}\sum_{i=1}^{n}\Big(N''(t_i,\theta) + p(t_i)N'(t_i,\theta) + q(t_i)N(t_i,\theta) - r(t_i)\Big)^2\right]^{1/2} \qquad (3.9)$$

and $S = \{t_1, \ldots, t_n\}$ is a set of random sample points in I selected to follow a uniform distribution. Assume that all conditions of Theorem 3.5 hold and that $|I| \geq 1$. Then, based on Theorem 3.6, the following error estimates holds:

Error estimates 3 (PINN I)

$$|N(t,\theta) - y(t)| \le |I|e^{M|t-t_0|} \cdot \overline{L}(\theta) + O\left(\frac{1}{\sqrt{n}}\right), \quad \text{for all } t \in I \quad \text{very likely}, \qquad (3.10)$$

where N is defined as in (2.4).

It is obvious that the neural network $N(\cdot,\theta)$ defined as in (2.4) is very unlikely to exactly satisfy the initial condition of (3.5). Similarly to (2.7), we define a new function $\overline{N}_A$ from N so that $\overline{N}_A$ satisfies the initial condition of (3.5).

$$\overline{N}_A(t,\theta) := y_0 + (t - t_0)y_0' + (t - t_0)^2 N(t,\theta). \qquad (3.11)$$

Then, using the argument in PINN II of Section 2.1, the functions of the form $\overline{N}_A(\cdot,\theta)$ are nice approximations for a twice continuously differentiable function on I. Now let us define an alternative loss function $\overline{L}_A$ in terms of $\overline{N}_A$

$$\overline{L}_A(\theta) := \left[\frac{1}{n}\sum_{i=1}^{n}\left(\overline{N}''_A(t_i,\theta) + p(t_i)\overline{N}'_A(t_i,\theta) + q(t_i)\overline{N}_A(t_i,\theta) - r(t_i)\right)^2\right]^{1/2}, \tag{3.12}$$

where $S = \{t_1, \ldots, t_n\}$ is a set of random sample points in I selected to follow a uniform distribution.

Theorems 3.7, 3.8 are mathematical results to derive Error estimates (3.13) for PINN II.

Theorem 3.7. *Let* p, q *and* r *be continuous functions on a compact interval* $I := [t_0 - c_1, t_0 + c_2]$. *Let* y *be a unique solution to* (3.5) *on* I. *Let* $\overline{N}_A$ *be a neural network defined as in* (2.4). *Let* $M = \sqrt{1 + M_1^2 + M_2^2}$, *where* $M_1 := \max_{t\in I}|p(t)|$, $M_2 := \max_{t\in I}|q(t)|$. *Define* $\overline{\mathcal{L}}_A(\theta)$ *by*

$$\overline{\mathcal{L}}_A(\theta) := \int_I |\overline{N}''_A(s,\theta) + p(s)\overline{N}'_A(s,\theta) + q(s)\overline{N}_A(s,\theta) - r(s)|ds,$$

where $\overline{N}_A$ *is a function defined as in* (3.11). *Then,*

$$|\overline{N}_A(t,\theta) - y(t)| \le e^{M|t-t_0|}\overline{\mathcal{L}}_A(\theta), \quad \textit{for all } t \in I.$$

Proof. The proof is the same as the one of Theorem 3.5, if N, $\mathbf{x}(t_0)$ and $\overline{\mathcal{L}}$ are replaced by $\overline{N}_A$, 0 and $\overline{\mathcal{L}}_A$, respectively. □

Theorem 3.8. *Assume that the conditions of Theorem 3.7 hold. Let* $(X_i)_{i\ge1}$ *be a sequence of independent and identically distributed random variables on a probability space* $(\Omega, \mathcal{F}, \mathbb{P})$ *that has a continuous uniform distribution on* I. *Then, for any* $t \in I$ *the following estimate holds:*

$$|\overline{N}_A(t,\theta) - y(t)|$$
$$\le |I|e^{K|t-t_0|}\left(\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\left(\overline{N}''_A(X_i,\theta) + p(X_i)\overline{N}'_A(X_i,\theta) + q(X_i)\overline{N}_A(X_i,\theta) - r(X_i)\right)^2\right]^{1/2}\right),$$

where $\overline{N}_A$ *is a function defined as in* (3.11), $\sigma := \left(\frac{1}{|I|}\int_I \varphi^2 dt - \left(\frac{1}{|I|}\int_I \varphi dt\right)^2\right)^{1/2} \ge 0$ *and*

$$\varphi(t) := \left|\overline{N}''_A(t,\theta) + p(t)\overline{N}'_A(t,\theta) + q(t)\overline{N}_A(t,\theta) - r(t)\right|, \quad t \in I.$$

Proof. Applying Theorem 3.5, the proof is the same as the one of Theorem 3.4 if

$$-|(N(t_0,\theta) - y_0| - |N'(t_0,\theta) - y'_0|$$

in (3.7) is replaced by 0. □

Assume that all conditions of Theorem 3.7 hold. As a direct consequence of Theorem 3.8, we obtain the following error estimates.

Error estimates 4 (PINN II)

$$|\overline{N}_A(t,\theta) - y(t)| \le |I|e^{M|t-t_0|} \cdot \overline{L}_A(\theta) + O\left(\frac{1}{\sqrt{n}}\right), \quad \text{for all } t \in I \quad \text{very likely}, \tag{3.13}$$

where $\overline{N}_A$ is a function defined as in (3.11).

4. Numerical Experiments

4.1. Adjustment for error estimates.

In this section, we validate the **Error estimates 1-4** in (3.2), (3.4), (3.10) and (3.13) by visualizing the relationship between error functions and training loss functions when PINN methods I and II are applied on various differential equations. Precisely, in Examples 4.1–4.4, we will validate the **Error estimates 2** in (3.4), as the following form:

(PINN I):

$$\begin{aligned} E(\theta)^2 &\le \left(|I|e^{K|I|} \cdot L(\theta) + O\left(\frac{1}{\sqrt{n}}\right)\right)^2 \\ &\le 3|I|^2 e^{2K|I|}\left(L_{de}(\theta)^2 + |N(t_0,\theta) - y_0|^2\right) + O\left(\frac{1}{n}\right) \quad \text{very likely}, \end{aligned} \tag{4.1}$$

where N, L_{de} and L are defined as in (2.4), (2.5) and (2.6), respectively,

$$E(\theta) := \left(\frac{1}{n}\sum_{i=1}^{n}(N(t_i,\theta) - y(t_i))^2\right)^{1/2} \tag{4.2}$$

and $S = \{t_1, \dots, t_n\}$ is a set of random sample points of an interval on which the solution is defined and the sample points are selected to follow a uniform distribution. For computational benefit, we will visualize $E(\theta)^2$ and $L_{de}(\theta)^2 + |N(t_0,\theta) - y_0|^2$ as the error and the training loss, respectively.

Likewise, in Examples 4.1–4.4, we will validate the **Error estimates 1** in (3.2), as the following form:

(PINN II):

$$\begin{aligned} E_A(\theta)^2 &\le \left(|I|e^{K|I|} \cdot L_A(\theta) + O\left(\frac{1}{\sqrt{n}}\right)\right)^2 \\ &\le 2|I|^2 e^{2K|I|} L_A(\theta)^2 + O\left(\frac{1}{n}\right) \quad \text{very likely}, \end{aligned} \tag{4.3}$$

where N_A and L_A are defined as in (2.7) and (2.8), respectively,

$$E_A(\theta) := \left(\frac{1}{n} \sum_{i=1}^{n} (N_A(t_i, \theta) - y(t_i))^2 \right)^{1/2}.$$

Then, for computational benefit, we will visualize $E_A(\theta)^2$ and $L_A(\theta)^2$ as the error and the training loss, respectively.

In Example 4.5, we will validate the **Error estimates 3, 4** in (3.10) and (3.13), respectively, similarly to the arguments above. Precisely, **Error estimates 3** in (3.10) is calculated as

(PINN I):

$$\begin{aligned} E(\theta)^2 &\leq \left(|I| e^{M|I|} \cdot \overline{L}(\theta) + O\left(\frac{1}{\sqrt{n}}\right) \right)^2 \\ &\leq 4|I|^2 e^{2M|I|} \left(\overline{L}_{de}(\theta)^2 + |N(t_0, \theta) - y_0|^2 + |N'(t_0, \theta) - y_0'|^2 \right) + O\left(\frac{1}{n}\right) \quad \text{very likely,} \end{aligned} \tag{4.4}$$

where $E(\theta)$, $\overline{L}(\theta)$ and $\overline{L}_{de}$ are defined as in (4.2), (3.8) and (3.9), respectively. For computational benefit, we will visualize $E(\theta)^2$ and $\overline{L}_{de}(\theta)^2 + |N(t_0, \theta) - y_0|^2 + |N'(t_0, \theta) - y_0'|^2$ as the error and the training loss, respectively.

Finally, **Error estimates 4** in (3.13) is calculated as

(PINN II):

$$\begin{aligned} \overline{E}_A(\theta)^2 &\leq \left(|I| e^{M|I|} \cdot \overline{L}_A(\theta) + O\left(\frac{1}{\sqrt{n}}\right) \right)^2 \\ &\leq 2|I|^2 e^{2M|I|} \overline{L}_A(\theta)^2 + O\left(\frac{1}{n}\right) \quad \text{very likely,} \end{aligned} \tag{4.5}$$

where $\overline{N}_A$ and $\overline{L}_A$ are functions defined as in (3.11) and (3.12), respectively,

$$\overline{E}_A(\theta) := \left(\frac{1}{n} \sum_{i=1}^{n} (\overline{N}_A(t_i, \theta) - y(t_i))^2 \right)^{1/2}$$

and $S = \{t_1, \ldots, t_n\}$ is a set of random sample points of an interval on which the solution is defined and the sample points are selected to follow a uniform distribution. For computational benefit, we will visualize $\overline{E}_A(\theta)^2$ and $\overline{L}_A(\theta)^2$ as the error and the training loss, respectively.

4.2. **Various examples.**

Example 4.1. Consider the following initial value problem:

$$\begin{cases} y'(t) = -\,2te^{-t^2}, \quad t \in [0,1] \\ y(0) = 1. \end{cases} \tag{4.6}$$

The unique solution to (4.6) is $y(t) = e^{-t^2}$, $t \in [0,1]$. Since we know explicitly the true solution to (4.6), we can visualize the error and the training loss while training the parameters of approximations through gradient descent as in Fig. 2. As we mentioned in the front of Section 4, the error and the training loss in PINN I are $E(\theta)^2$ and $L_{de}(\theta)^2 + |N(t_0,\theta) - y_0|^2$, respectively. On the other hand, the error and the training loss in PINN II are $E_A(\theta)^2$ and $L_A(\theta)^2$, respectively. As in Fig. 3, the ratio of error to training loss $\left(\frac{\text{error}}{\text{training loss}}\right)$ should be always less than 3 and 2 in PINN I and PINN II by the error estimates (4.1), (4.3), respectively.

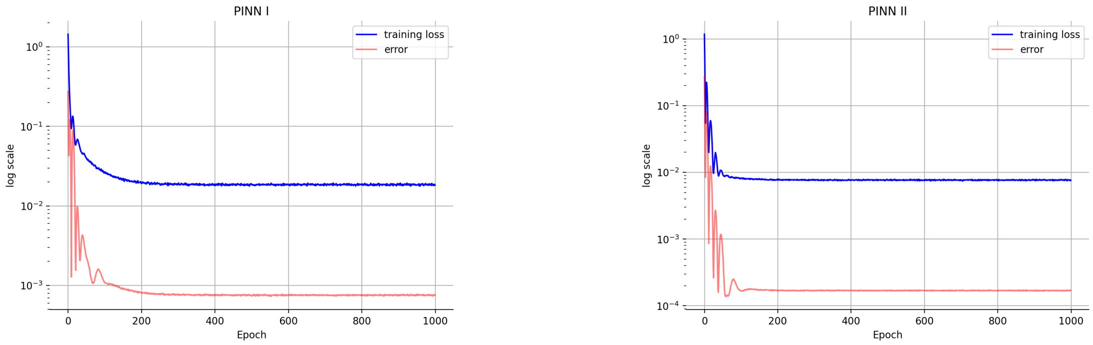

FIGURE 2. Visualizing training loss and error for $y'(t) = -2te^{-t^2}$, $t \in [0,1]$ with $y(0) = 1$

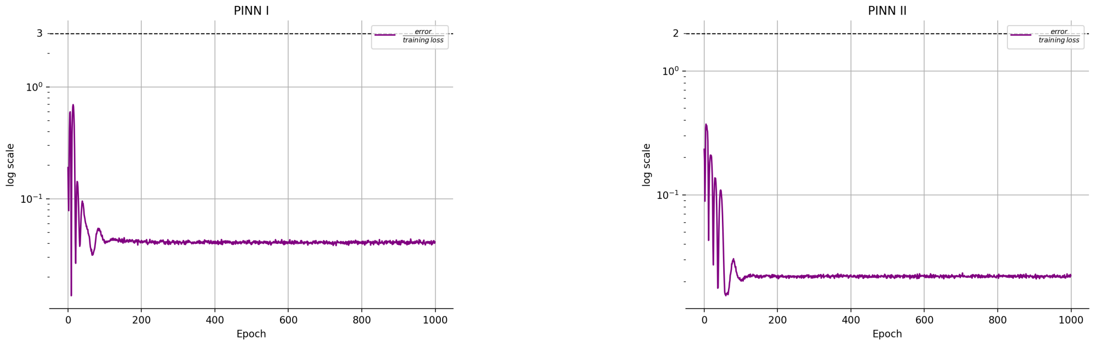

FIGURE 3. Visualizing $\left(\frac{\text{error}}{\text{training loss}}\right)$ for $y'(t) = -2te^{-t^2}$, $t \in [0,1]$ with $y(0) = 1$

Example 4.2. Consider the following initial value problem:

$$\begin{cases} y'(t) = & 1.27y(1-y), \quad t \in [0,1] \\ y(0) = & 0.67. \end{cases} \tag{4.7}$$

The unique solution to (4.7) is $y(t) = \frac{1}{1-3.03e^{-1.27t}}$, $t \in [0,1]$. The error and the training loss in PINN I are $E(\theta)^2$ and $L_{de}(\theta)^2 + |N(t_0,\theta) - y_0|^2$, respectively. As in Fig. 4, we can visualize the error and the training loss. On the other hand, the error and the training loss in PINN II are $E_A(\theta)^2$ and $L_A(\theta)^2$, respectively. Figure 5 describes that the ratio of error to training loss, $\left(\frac{\text{error}}{\text{training loss}}\right)$ is less than 3 and 2 in PINN I and PINN II, respectively, so that we can validate the error estimates, (4.1) and (4.3).

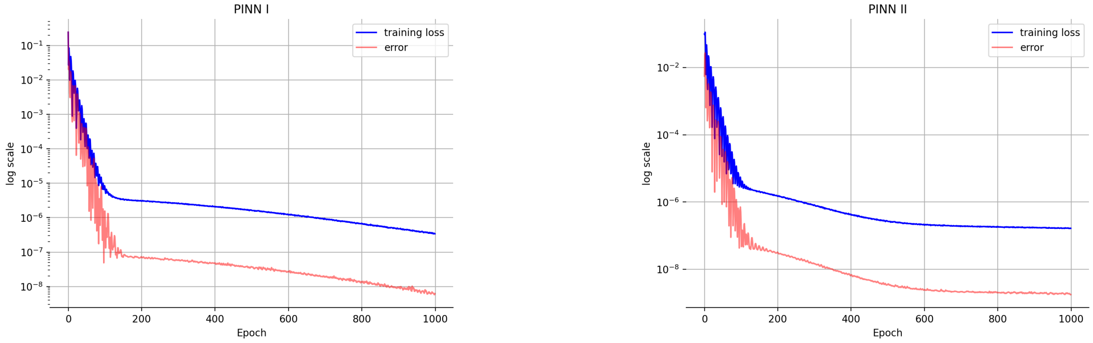

FIGURE 4. Visualizing the training loss and the error for $y'(t) = 1.27y(1-y)$, $t \in [0,1]$ with $y(0) = 0.67$

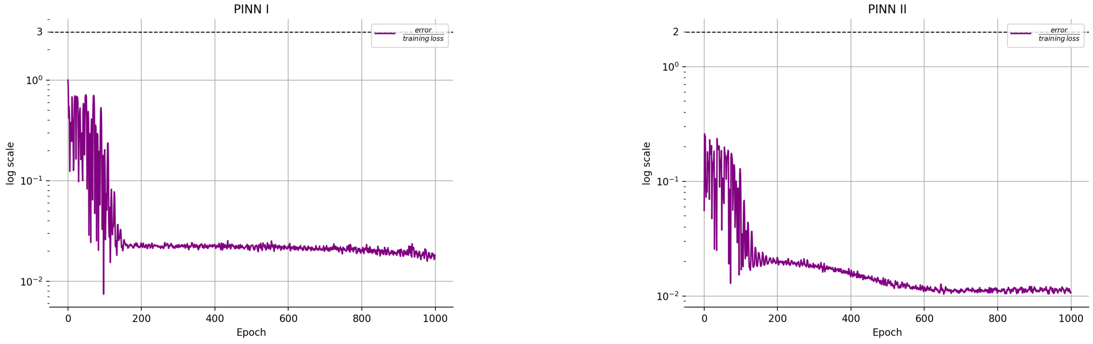

FIGURE 5. Visualizing $\left(\frac{\text{error}}{\text{training loss}}\right)$ for $y'(t) = 1.27y(1-y)$, $t \in [0,1]$ with $y(0) = 0.67$

Example 4.3. Consider the following initial value problem:

$$\begin{cases} y'(t) = & \dfrac{3t^2+4t+2}{2y-2}, \quad t \in [0,1] \\ y(0) = & -1. \end{cases} \tag{4.8}$$

The unique solution to (4.8) is $y(t) = 1 - \sqrt{t^3+2t^2+2t+4}$, $t \in [0,1]$. The error and the training loss in PINN I are $E(\theta)^2$ and $L_{de}(\theta)^2 + |N(t_0,\theta) - y_0|^2$, respectively. On the other hand, the error and the training loss in PINN II are $E_A(\theta)^2$ and $L_A(\theta)^2$, respectively. As in Fig. 6, we can visualize the error and the training loss. Figure 7 describes that the ratio of error to training loss, $\left(\frac{\text{error}}{\text{training loss}}\right)$ is less than 3 and 2 in PINN I and PINN II, respectively, so that we can validate the estimates, (4.1) and (4.3).

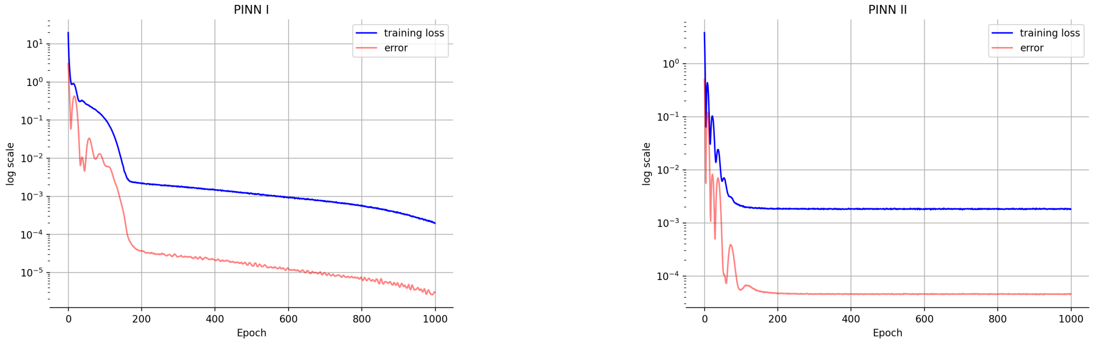

FIGURE 6. Visualizing the training loss and the error for $y'(t) = \frac{3t^2+4t+2}{2y-2}$, $t \in [0,1]$ with $y(0) = -1$

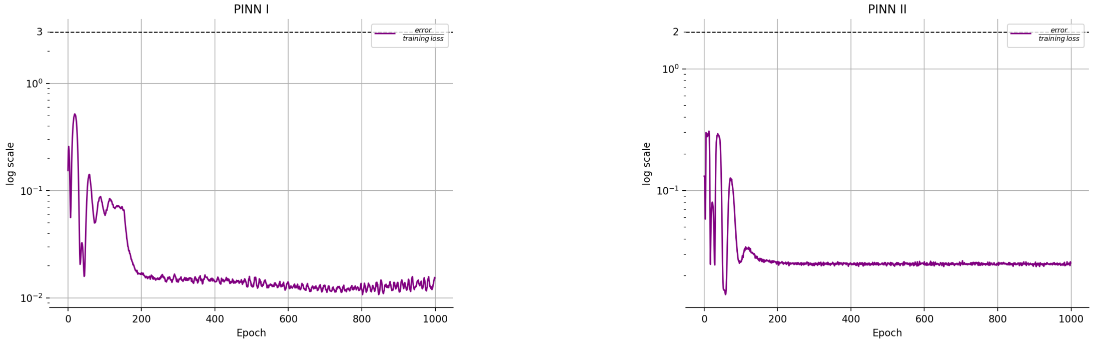

FIGURE 7. Visualizing $\left(\frac{\text{error}}{\text{training loss}}\right)$ for $y'(t) = \frac{3t^2+4t+2}{2y-2}$, $t \in [0,1]$ with $y(0) = -1$

Example 4.4. Consider the following initial value problem:

$$\begin{cases} y'(t) = \dfrac{t^2 + ty + y^2}{t^2}, & t \in [1, 2] \\ y(1) = 0. \end{cases} \tag{4.9}$$

The unique solution to (4.9) is $y(t) = t\tan(\log(t))$, $t \in [1, 2]$. The error and the training loss in PINN I are $E(\theta)^2$ and $L_{de}(\theta)^2 + |N(t_0, \theta) - y_0|^2$, respectively. On the other hand, the error and the training loss in PINN II are $E_A(\theta)^2$ and $L_A(\theta)^2$, respectively. As in Fig. 8, we can visualize the error and the training loss. Figure 9 describes that the ratio of error to training loss, $\left(\frac{\text{error}}{\text{training loss}}\right)$ is less than 3 and 2 in PINN I and PINN II, respectively, so that we can validate the estimates, (4.1) and (4.3).

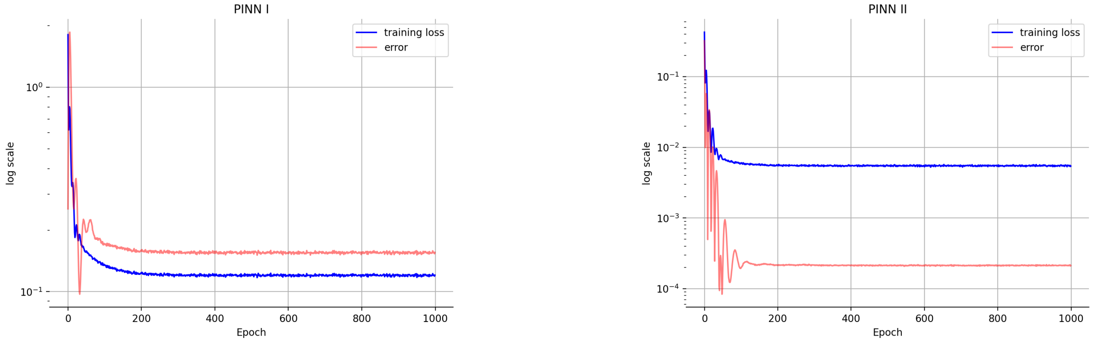

FIGURE 8. Visualizing the training loss and the error for $y'(t) = \frac{t^2+ty+y^2}{t^2}$, $t \in [1, 2]$ with $y(1) = 0$

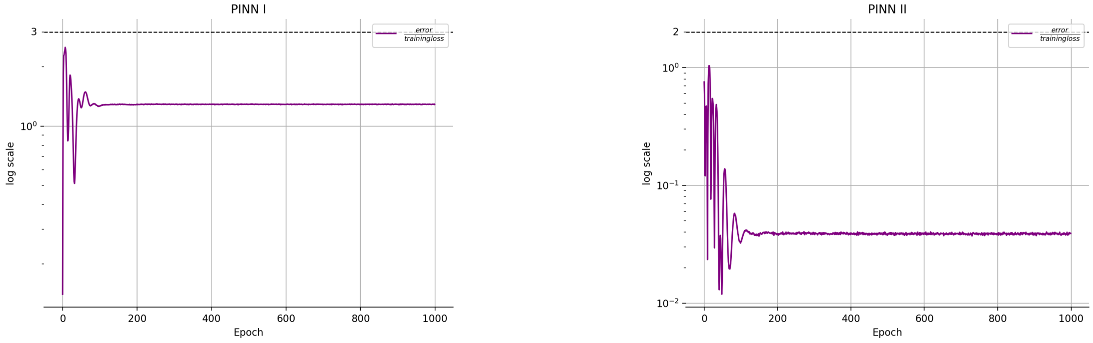

FIGURE 9. Visualizing $\left(\frac{\text{error}}{\text{training loss}}\right)$ for $y'(t) = \frac{t^2+ty+y^2}{t^2}$, $t \in [1, 2]$ with $y(1) = 0$

Example 4.5. Consider the following initial value problem:

$$\begin{cases} y''(t) + 2y'(t) + 10y(t) = 0, \quad t \in [0,1] \\ \qquad\qquad y(0) = 0.75, \quad y'(0) = 0 \end{cases} \tag{4.10}$$

The unique solution to (4.10) is $y(t) = e^{-t}\left(\frac{3}{4}\cos(3t) + \frac{1}{4}\sin(3t)\right)$, $t \in [0,1]$. Since we know explicitly the true solution to (4.10), we can visualize the error and the training loss while training the parameters of approximations through gradient descent. As we mentioned in the front of Section 4, the error and the training loss in PINN I are $E(\theta)^2$ and $\overline{L}_{de}(\theta)^2 + |N(t_0,\theta) - y_0|^2 + |N'(t_0,\theta) - y_0'|^2$, respectively. As in Fig. 10, we can visualize the error and the training loss. On the other hand, the error and the training loss in PINN II are $\overline{E}_A(\theta)^2$ and $\overline{L}_A(\theta)^2$, respectively. Figure 11 describes that the ratio of error to training loss, $\left(\frac{\text{error}}{\text{training loss}}\right)$ is less than 4 and 3 in PINN I and PINN II, respectively, so that we can validate the estimates, (4.4) and (4.5).

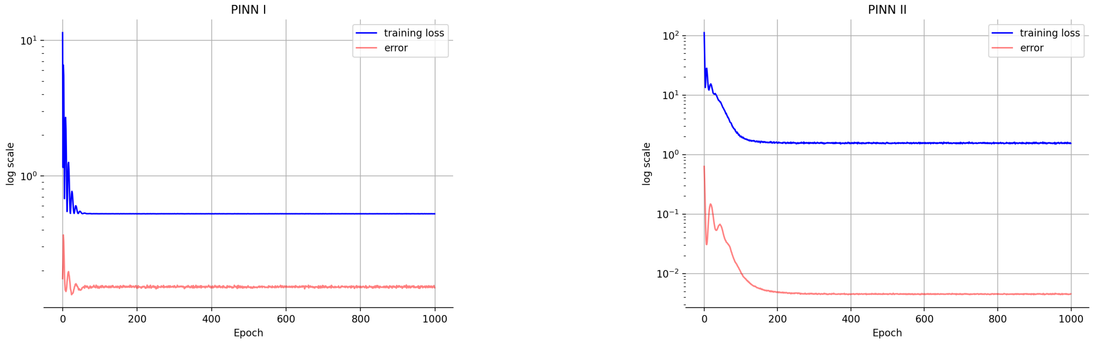

FIGURE 10. Visualizing the training loss and the error for $y''(t)+2y'(t)+10y(t) = 0$, $t \in [0,1]$ with $y(0) = 0.75$, $y'(0) = 0$

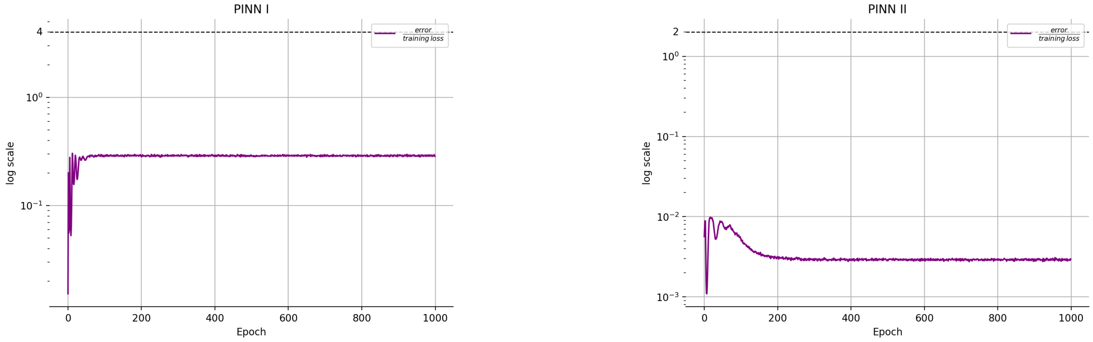

FIGURE 11. Visualizing $\left(\frac{\text{error}}{\text{training loss}}\right)$ for $y''(t) + 2y'(t) + 10y(t) = 0$, $t \in [0,1]$ with $y(0) = 0.75$, $y'(0) = 0$

4.3. Summary and Outlook.

For the numerical experiments in Section 4.2, the PINN model of $(1, 16, 32, 1)$ was used with an activation function $\tanh$. Each experiment contained 1000 epochs of training with 10000 sample size and a learning rate of 0.01. Adam optimizer and a scheduler were used additionally. For further details, the code used for numerical experiments is available at https://github.com/hahmYoo/Error-Estimates-in-PINN.

In summary, by numerical experiments, we verified the **Error estimates 1–4** in (3.2), (3.4), (3.10) and (3.13) hold well. Furthermore, a significant relationship exists between the trajectory of the error and the training loss. In other words, the trajectories of error and training loss show similar behavior, and we expect that there is a strong tendency beyond the error estimates we derived. Our error estimates are valid even if the neural network is replaced with an arbitrary smooth function, which means that inherent properties of neural networks were not used to derive our error estimates. Although Fig. 9 presents fairly sharp upper bounds for $\frac{\text{error}}{\text{training loss}}$ in the case when the epoch was very small, we can see in many examples that the upper bounds for $\frac{\text{error}}{\text{training loss}}$ indicated by the dotted line are quite relaxed. If error estimates are derived based on the inherent structure of a neural network, such as the universal approximation theorem, it is expected that more accurate error estimates can be derived in PINN.

Throughout this paper, we verify that error is strongly related to training loss functions by using basic calculus with probabilistic arguments. By expanding our discussion further, we expect that PINN methods can be efficiently applied to find approximations of solutions to various elliptic and parabolic partial differential equations. We also expect that mathematical analysis for error estimates via training loss functions can be studied in general partial differential equations as in [14, 15, 16].

Acknowledgments

This research was supported by Kumoh National Institute of Technology(2023 $\sim$ 2024)

References

1. K. Hornik, M. Stinchcombe, H. White, *Multilayer feedforward networks are universal approximators*, Neural Networks, **2** (1989), 359-366.
2. E.K. Ryu, *Infinitely Large Neural Networks*, Lecture Notes in Mathematics, Research Institute of Mathematics, Number 58 (2023).
3. N. Yadav, A. Yadav, M. Kumar, *An introduction to neural network methods for differential equations*, SpringerBriefs Appl. Sci. Technol., Springer, Dordrecht, 2015.
4. R.T.Q. Chen, Y. Rubanova, J. Bettencourt, D.K. Duvenaud, *Neural Ordinary Differential Equations*, Proceedings of 32nd Conference on Neural Information Processing Systems(NeurIPS2018), Montréal, Canada 2018.
5. M. Raissi, P. Perdikaris, 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, , **378** (2019), 686-707.
6. G.E. Karniadakis, I.G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, *Physics-informed machine learning*, Nature Reviews Physics, **3** (2021), 422–440.

7. I.E. Lagaris, A Likas, D.I. Fotiadis, *Artificial Neural Networks for Solving Ordinary and Partial Differential Equations*, IEEE Transactions on Neural Networks, **9** (1998), 987-1000.
8. A. Malek, R.S. Beidokhti, *Numerical solution for high order differential equations using a hybrid neural network—optimization method*, Appl. Math. Comput., **183** (2006), 260–271.
9. H. Lee, I. Kang, *Neural Algorithm for Solving Differential Equations*, Journal of Computational Physics, **91** (1990), 110-131 .
10. M. Dissanayake, N. Phan-Thien, *Neural-Network-Based Approximations for Solving Partial Differential equations*, Communications in Numerical Methods in Engineering, **10** (1994), 195-201.
11. B. Hillebrecht, B. Unger, *Certified machine learning: A posteriori error estimation for physics-informed neural networks*, Proceedings of 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy, 2022.
12. E. Hairer, S.P. Nørsett, G. Wanner, *Solving Ordinary Differential Equations*, I, 3rd ed. Berlin, Heidelberg: Springer, 2008.
13. E.A. Coddington, *An introduction to ordinary differential equations*, Prentice-Hall Mathematics Series Prentice-Hall, Inc., Englewood Cliffs, NJ, 1961.
14. S. Mishra, R. Molinaro, *Estimates on the generalization error of physics-informed neural networks for approximating a class of inverse problems for PDEs*, IMA J. Numer. Anal., **42** (2022), 981–1022.
15. S. Mishra, R. Molinaro, *Estimates on the generalization error of physics-informed neural networks for approximating PDEs*, IMA J. Numer. Anal., **43** (2023), 1–43.
16. T. De Ryck, A.D. Jagtap, S. Mishra, *Error estimates for physics-informed neural networks approximating the Navier–Stokes equations*, IMA J. Numer. Anal., **44** (2024), 83–119.