On Numerical Integration in Neural Ordinary Differential Equations

Aiqing Zhu [1 2] **Pengzhan Jin** [3] **Beibei Zhu** [4] **Yifa Tang** [1 2]

Abstract

The combination of ordinary differential equations and neural networks, i.e., neural ordinary differential equations (Neural ODE), has been widely studied from various angles. However, deciphering the numerical integration in Neural ODE is still an open challenge, as many researches demonstrated that numerical integration significantly affects the performance of the model. In this paper, we propose the inverse modified differential equations (IMDE) to clarify the influence of numerical integration on training Neural ODE models. IMDE is determined by the learning task and the employed ODE solver. It is shown that training a Neural ODE model actually returns a close approximation of the IMDE, rather than the true ODE. With the help of IMDE, we deduce that (i) the discrepancy between the learned model and the true ODE is bounded by the sum of discretization error and learning loss; (ii) Neural ODE using non-symplectic numerical integration fail to learn conservation laws theoretically. Several experiments are performed to numerically verify our theoretical analysis.

1. Introduction

Recently, Neural Ordinary Differential Equations (Neural ODE) (Chen et al., 2018) were proposed as a continuous model by embedding neural networks into continuous dynamical systems, and became an important option of model architecture. They offered dynamical systems perspectives on deep learning researches and thus have attracted increasing attention. For example, Yan et al. (2020) proposed TisODE to further enhance robustness according to the non-intersecting characteristics of ODE. Botev et al. (2021) experimentally found that Neural ODE are the most effective approach to model latent dynamics from images due to continuous and time-reversible priors.

Despite ODE dynamics benefit models, we have to apply a numerical integration in Neural ODE, which prevents the model from being consistent with the design. Gusak et al. (2020) and Zhuang et al. (2020) observed that changing the numerical solver yields performance degradation. Ott et al. (2021) and Queiruga et al. (2020) studied how the numerical integration influences the Neural ODE model and they proposed to adjust the step size and use high order solver to train Neural ODE, respectively. High-accuracy integration is able to loosen the impact of discretization error, however, quantifying such influences stills remains open. In this work, we also focus on the numerical integration in Neural ODE but we aim to decipher it theoretically and clarify the influence rigorously.

The main ingredient of this work is the formal analysis (Feng, 1991). Historically, modified differential equation is an important tool for understanding the numerical behavior of solving ODE (Eirola, 1993; Feng, 1991; 1993; Sanz-Serna, 1992; Yoshida, 1993). The methodology is to interpret the numerical solution of the original system as the exact solution of a perturbed equation. Herein, this idea is tailored to Neural ODE. We first search for a perturbed differential equation such that its numerical solution matches the exact solution of the true system and then show that training Neural ODE returns a close approximation of this perturbed equation. The perturbed equation is named as inverse modified differential equation (IMDE) in this paper. Several experiments are performed to verify our theoretical analysis. Although the closed-form expression of the IMDE is inaccessible as it is calculated via the true system, we can still clarify the influence of numerical integration on Neural ODE with the help of IMDE. In summary, we list several statements derived via the proposed IMDE that will be documented in detail later:

- The trained Neural ODE model is a close approximation of the IMDE, i.e., the difference between the learned Neural ODE model and the truncation of the

[1] LSEC, ICMSEC, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China [2] School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China [3] School of Mathematical Sciences, Peking University, Beijing 100871, China [4] School of Mathematics and Physics, University of Science and Technology Beijing, Beijing 100083, China. Correspondence to: Yifa Tang <tyf@lsec.cc.ac.cn>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

IMDE is bounded by the sum of the learning loss and a discrepancy which can be made sub exponentially small.

- The difference between the learned Neural ODE model and the true hidden system is bounded by the sum of the discretization error Ch^p and the learning loss, where h is the discrete step and p is the order of the numerical integrator.

- Neural ODE using non-symplectic numerical integration fail to learn conservation laws theoretically.

1.1. Related Works

Continuous models combining ODE and neural networks have a long history. They had already been developed and implemented to learn hidden dynamics decades ago (Anderson et al., 1996; González-García et al., 1998; Rico-Martinez et al., 1994; Rico-Martinez & Kevrekidis, 1993). Recently, these insights have again attracted more and more attention. The connection between dynamical systems and deep neural networks was studied in (E, 2017; E et al., 2019; Li et al., 2017; Sonoda & Murata, 2019). Neural ODE were proposed as a continuous approximation of the ResNets architecture in (Chen et al., 2018). With their successful applications across diverse scientific disciplines (Botev et al., 2021; Rubanova et al., 2019; Yildiz et al., 2019), there have been extensive works studying this learning model in terms of optimization algorithms (Xia et al., 2021; Zhuang et al., 2020), approximation capabilities (Zhang et al., 2020), robustness properties (Yan et al., 2020), augmentation strategies (Dupont et al., 2019; Massaroli et al., 2020) and variant architectures (Jia & Benson, 2019; Norcliffe et al., 2021; Zhang et al., 2019).

This work concentrates on numerical integration in Neural ODE. Pal et al. (2021) proposed a novel regularization for Neural ODE based on the internal cost of the numerical integration. Poli et al. (2020) explored the interplay between Neural ODE and numerical integration, introduced hypersolvers for fast inference. Based on numerical analysis theory, Krishnapriyan et al. (2022) developed a convergence test for selecting solver that makes the Neural ODE learn meaningfully continuous dynamics. As reported in (Gusak et al., 2020; Zhuang et al., 2020), changing the numerical solver yields performance degradation. Ott et al. (2021) and Queiruga et al. (2020) discussed this problem further. They observed that if training using coarse discretization, then testing using another solver of equal or smaller accuracy results in significantly lower performance. Thus, they proposed using a more accurate solver (adjusting the step size and using high order solver, respectively). Due to the discretization error, more accurate solver can only loosen this issue. The aim of our work is to clarify the influence of the numerical integration employed in Neural ODE models mathematically.

Neural ODE can be utilized as a data-driven technique for the discovery of latent dynamics (Botev et al., 2021; Huh et al., 2020; Raissi et al., 2018). For this task, Keller and Du (2021) and Du et al. (2021) provided convergence and stability analysis for Linear Multistep Neural Networks (LMNets) (Raissi et al., 2018). They proved that the grid error of LMNets is bounded by the sum of discretization error and approximation error under auxiliary initial conditions. Different from their works, the IMDE proposed in this paper provides a framework for the error analysis of Neural ODE for learning dynamical systems. As shown in Theorem 3.2, our error bound is similar to theirs but our results can be applied to more general Neural ODE models.

Modified differential equations (MDE) are well-established tools for numerical analysis of solving ODE (Eirola, 1993; Feng, 1993; Sanz-Serna, 1992; Yoshida, 1993). In the deep learning community, Lu et al. (2018) used the concept of MDE to justify the performance boost of the proposed models. França et al. (2021) employed MDE to study the fine behavior of gradient-based optimization. In this paper, the proposed IMDE is inspired by MDE and modified integrator (Chartier et al., 2007) and our arguments rely on fundamental techniques of MDE (Feng, 1991; Hairer & Lubich, 1997; Reich, 1999).

2. Preliminaries

We begin with some notations. Consider autonomous systems of first-order ordinary differential equations

$$\frac{d}{dt}y(t) = f(y(t)), \quad y(0) = x, \tag{1}$$

where $y(t) \in \mathbb{R}^D$, $f : \mathbb{R}^D \to \mathbb{R}^D$ is smooth and x is the initial value. A non-autonomous system $\frac{d}{dt}y(t) = f(y(t), t)$ can be brought into this form by adding the variable $y_{D+1} = t$ to $y(t)$ and appending the equation $\frac{d}{dt}t = 1$. For fixed t, $y(t)$ can be regarded as a function of its initial value x. We denote

$$\phi_t(x) := y(t) = x + \int_0^t f(y(\tau))d\tau,$$

which is known as the time-t flow map of dynamical system (1). In general, we chose a numerical integrator Φ_h that approaches ϕ_h and compose it to obtain the numerical solution. A common choice of the numerical integrator is the Runge-Kutta method:

$$\begin{aligned} v_i &= x + h\sum_{j=1}^{I} a_{ij} f(v_j), \quad i = 1, \cdots, I, \\ \Phi_h(x) &= x + h\sum_{i=1}^{I} b_i f(v_i), \end{aligned} \tag{2}$$

where x is the initial value, h is the discrete step. The coefficients a_{ij}, b_i with $i, j = 1, \cdots, I$ fully characterize the method. In order to emphasize specific differential equation, we will add the subscript f and denote ϕ_t as $\phi_{t,f}$ and Φ_h as $\Phi_{h,f}$.

2.1. Neural ODE

Neural Ordinary Differential Equations (Neural ODE) (Chen et al., 2018) are continuous models by embedding neural networks into continuous dynamical systems. In this work, we consider the empirical risk optimization problem

$$L = \frac{1}{N}\sum_{n=1}^{N} l(\phi_{T,f_\theta}(x_n), z_n),$$

where $\{(x_n, z_n)\}_{n=1}^N$ is the sampled training data, $l(\cdot,\cdot)$ is a loss function that is minimized when its two arguments are equal. ϕ_{T,f_θ} is a Neural ODE model with a trainable neural network f_θ[1]. Depending on the application, input or output layers are employed but we concentrate on the hidden state of the ODE layer in this paper. Exact evaluating ϕ_{T,f_θ} is intractable and we have to use an ODE solver to approximate ϕ_{T,f_θ}. Dividing T in S equally-spaced intervals, the ϕ_{T,f_θ} can be approximated by S compositions of a predetermined one-step numerical integrator Φ_h (e.g. Runge-Kutta method (2)),

$$\begin{aligned}\phi_{T,f_\theta} \approx& \underbrace{\Phi_{h,f_\theta} \circ \cdots \circ \Phi_{h,f_\theta}}_{S \text{ compositions}}(x)\\ =& \left(\Phi_{h,f_\theta}\right)^S(x),\end{aligned}$$

where $h = T/S$ is the discrete step. Therefore, the practical input of loss function is given by the predetermined ODE solver, i.e.,

$$L = \frac{1}{N}\sum_{n=1}^{N} l\left(\left(\Phi_{h,f_\theta}\right)^S(x_n), z_n\right).$$

3. Main Results

Throughout this section we assume that there exists a true (but inaccessible) ODE solution such that $z_n = \phi_{T,f}(x_n)$. If Neural ODE model is employed due to the ODE dynamic benefits (e.g., improving robustness (Yan et al., 2020) or ODE prior (Botev et al., 2021)), it is essential that the assumption holds and the trained model is an approximation of the true ODE. With this assumption, we are able to clarify the influence of the numerical integration on training Neural ODE models by studying the change of approximation target.

3.1. Inverse Modified Differential Equations

We aim to find a perturbed differential equation

$$\begin{aligned}\frac{d}{dt}\tilde{y}(t) =& f_h(\tilde{y}(t))\\ =& f_0(\tilde{y}) + hf_1(\tilde{y}) + h^2 f_2(\tilde{y}) + \cdots,\end{aligned} \tag{3}$$

such that $\Phi_{h,f_h}(x) = \phi_{h,f}(x)$ formally. Here, identity is understood in the sense of the formal power series in h without taking care of convergence issues of Equation (3).

To obtain f_h, we first expand $\phi_{h,f}(x)$ into a Taylor series around $h = 0$,

$$\begin{aligned}\phi_{h,f}(x) =& x + hf(x) + \frac{h^2}{2}f'f(x)\\ &+ \frac{h^3}{6}(f''(f,f)(x) + f'f'f(x)) + \cdots.\end{aligned} \tag{4}$$

Here, the notation $f'(x)$ is a linear map (the Jacobian), the second order derivative $f''(x)$ is a symmetric bilinear map, and similarly for higher order derivatives described as a tensor. A general expansion formula for (4) is given in Appendix A.1.

Next, the numerical solution can be expanded as

$$\Phi_{h,f_h}(x) = x + hd_{1,f_h}(x) + h^2 d_{2,f_h}(x) + \cdots, \tag{5}$$

where the functions d_{j,f_h} are given and typically composed of f_h and its derivatives. Expansion formulas for Runge-Kutta methods are given in Appendix A.2. For consistent integrators[2],

$$d_{1,f_h}(x) = f_h(x) = f_0(x) + hf_1(x) + h^2 f_2(x) + \cdots.$$

In $h^i d_{i,f_h}(x)$, the power of h of the terms containing f_k is at least $k + i$. Thus the coefficient of h^{k+1} in (5) is

$$f_k + \cdots,$$

where the "$\cdots$" indicates residual terms composed of f_j with $j \le k - 1$ and their derivatives. By comparison of the coefficients of equal powers of h in (4) and (5) such that these two series coincide term by term, unique functions f_k in Equation (3) are obtained recursively. In Appendix A.3, we present some examples illustrating the process of the above calculation. In this paper, we name the Equation (3) obtained via the above process as *inverse modified differential equation (IMDE)* since it is inspired by the MDE and learning ODE is an inverse problem.

Furthermore, we obtain that formally

$$\left(\Phi_{h,f_h}\right)^S(x) = \left(\phi_{h,f}\right)^S(x) = \phi_{Sh,f}(x),$$

[1] Under this form, time dependence can be added according to

$$\hat{x}_n = (x_n, 0),\ \hat{z}_n = (z_n, T),\ \hat{f}_\theta = (f_\theta, 1).$$

[2] An integrator is consistent if its order is not less than 1. A Runge-Kutta method (2) is consistent if $\sum_{i=1}^I b_i = 1$.

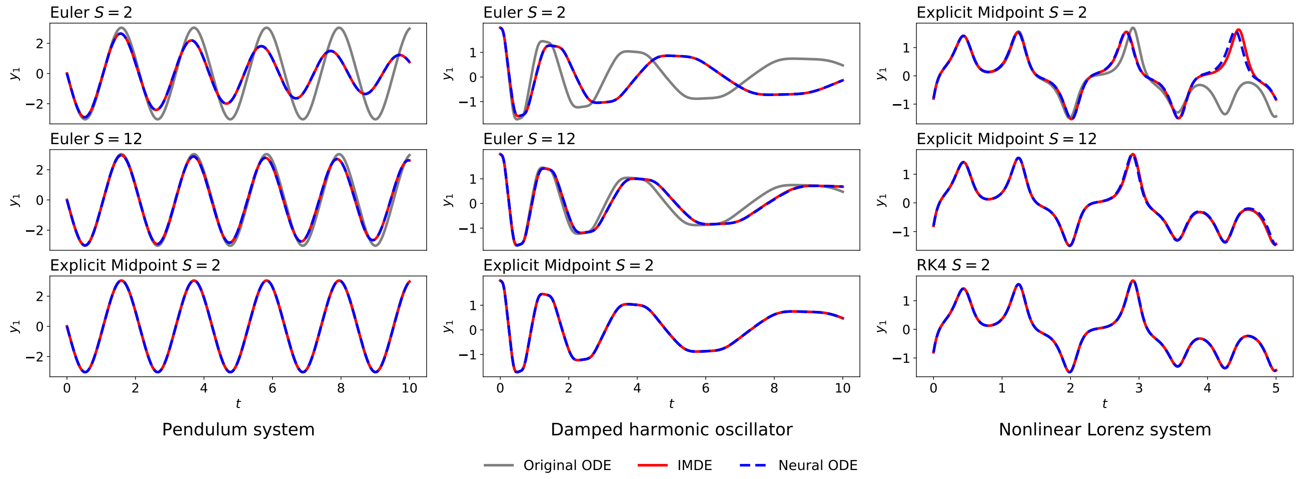

Figure 1. The first component of the trajectories of original, modified and learned equations. The Neural ODE models are trained on three tasks where the used numerical solvers are S compositions of the Euler method (of order 1), the explicit midpoint rule (of order 2) and the RK4 method (of order 4), respectively. The trained Neural ODE capture the evolution of the corresponding IMDE more accurately. With the improvement of the accuracy of the solver used in training, the learned models are closer to the true systems since the discrepancy between the IMDE and the original ODE is reduced. Experimental details are presented in Appendix C.

and the training process of Neural ODE is to minimize the difference between $\left(\Phi_{h,f_h}\right)^S(x)$ and $\left(\Phi_{h,f_\theta}\right)^S(x)$. Thus it is natural to conjecture that the trained f_θ is a close approximation of f_h. In order to substantiate this claim, we use Neural ODE to learn several benchmark problems that are widely investigated for the discovery of hidden dynamics (Du et al., 2021; Greydanus et al., 2019; Keller & Du, 2021; Yu et al., 2021). Here, the training data is generated by a known system, $\mathcal{T} = \{(x_n, \phi_{T,f}(x_n))\}_{n=1}^N$, and we can calculate the corresponding IMDE. We train the Neural ODE model using fixed step solvers with step sizes T/S. After training, we employ a dense numerical integration to compute the trajectories of the learned ODE and the IMDE. As displayed in Figure 1, training Neural ODE returns approximations of the IMDE, which is consistent with the statement. We will rigorously show that this statement is true under reasonable assumptions in Section 3.2.

3.2. Rigorous Analysis

In Section 3.1, we neglected the convergence issue[3]. For rigorous analysis, we truncate the IMDE and denote the truncation of series in (3) as

$$f_h^K(y) = \sum_{k=0}^K h^k f_k(y).$$

We aim to derive an error bound between the trained f_θ and f_h^K in this subsection to circumvent the convergence issue. To begin with, we introduce some notations. For a compact subset $\mathcal{K} \subset \mathbb{C}^D$, let $\mathcal{B}(x,r) \subset \mathbb{C}^D$ be the complex ball of radius $r > 0$ centered at $x \in \mathbb{C}^D$ and let

$$\mathcal{B}(\mathcal{K}, r) = \bigcup_{x \in \mathcal{K}} \mathcal{B}(x, r).$$

We will work with l_∞- norm on $\mathbb{C}^D$ and denote $\|\cdot\| = \|\cdot\|_\infty$ For a analytic vector field g, we define

$$\|g\|_{\mathcal{K}} = \sup_{x \in \mathcal{K}} \|g(x)\|.$$

Now, the main theorem is given as follows.

Theorem 3.1. *For $x \in \mathbb{R}^D$ and $r_1, r_2 > 0$, a given ODE solver that is S compositions of a Runge-Kutta method Φ_h, we denote*

$$\mathcal{L} = \left\| \left(\Phi_{h,f_\theta}\right)^S - \phi_{Sh,f} \right\|_{\mathcal{B}(x,r_1)} / (Sh),$$

and suppose that the target vector field f and the learned vector field f_θ are analytic and bounded by m on $\mathcal{B}(x, r_1 + r_2)$. Then, there exist integer $K = K(h)$ and constants T_0, q, γ, c_1 that depend on r_1/m, r_2/m, S and Φ_h, such that, if $0 < T < T_0$,

$$\left\| f_\theta(x) - f_h^K(x) \right\| \le c_1 m e^{-\gamma / h^{1/q}} + \frac{e}{e-1}\mathcal{L},$$

where e is the base of natural logarithm, $h = T/S$ and f_h^K is the truncated vector field of the IMDE determined by Φ_h and f.

Proof. The proof can be found in Appendix B. □

[3] The series in Equation (3) does not converge in general.

Here, the first term, $c_1 m e^{-\gamma/h^{1/q}}$, is sub exponentially small, i.e., for any $k > 0$, there exists a constant c such that $c_1 m e^{-\gamma/h^{1/q}} < ch^k$. In statistical learning theory, learning error or expected risk typically refers to $\int l(f_{net}(x), z)dP(x,z)$. In this paper, it is of the form $\int \|\Phi_{h,f_\theta}(x) - \phi_{h,f}(x)\|_2^2 dP(x)$, which is the square of L_2-norm of $\Phi_{h,f_\theta} - \phi_{h,f}$. The $\mathcal{L}$ defined here, i.e., the second term, is the L_∞-norm of $\Phi_{h,f_\theta} - \phi_{h,f}$ and thus measures the learning loss in the sense of generalization. If the learning loss converges to zero, the difference between the learned ODE and the truncated IMDE converges to near-zero. Thus we claim that the trained Neural ODE model is a close approximation of the IMDE.

3.3. The Discrepancy between f_θ and f

We have shown that training Neural ODE returns a close approximation of the corresponding IMDE instead of the true ODE. Although the true solution is unknown in practice and the IMDE is also inaccessible, we can quantify the discrepancy between f_θ and f via investigating f_h.

Theorem 3.2. *Suppose that the integrator $\Phi_h(x)$ with discrete step h is of order $p \geq 1$, more precisely,*

$$\Phi_{h,f}(x) = \phi_{h,f}(x) + h^{p+1}\delta_f(x) + \mathcal{O}(h^{p+2}),$$

where $h^{p+1}\delta_f(x)$ is the leading term of the local truncation applied to (1). Then, the IMDE obeys

$$\frac{d}{dt}\tilde{y} = f_h(\tilde{y}) = f(\tilde{y}) + h^p f_p(\tilde{y}) + \cdots,$$

where $f_p(y) = -\delta_f(y)$, i.e., $f_h = f + \mathcal{O}(h^p)$.

Furthermore, under the notations and conditions of Theorem 3.1, there exists a constant c_2 that depends on r_1/m, r_2/m, S and Φ_h, such that,

$$\|f_\theta(x) - f(x)\| \leq c_2 m h^p + \frac{e}{e-1}\mathcal{L}.$$

Proof. The proof can be found in Appendix B. □

For completeness, Theorem 3.2 was experimentally verified in Figure 2, where the error orders are consistent with the theoretical analysis.

As a direct consequence of the well-known Grönwall's Inequality (Howard, 1998), we have the following corollary to provide an error bound between trajectories.

Corollary 3.3. *Under notations and conditions of Theorem 3.1, let $V_t = \{\phi_{\tau,f}(x) | 0 \leq \tau \leq t\}$ and*

$$\mathcal{L}_t = \left\| (\Phi_{h,f_\theta})^S - \phi_{Sh,f} \right\|_{\mathcal{B}(V_t, r_1)} / (Sh).$$

Then, there exist constants C_1, C_2 such that

$$\|\phi_{t,f}(x) - \phi_{t,f_\theta}(x)\| \leq \frac{C_2 e^{C_1 t} - C_2}{C_1}(h^p + \mathcal{L}_t).$$

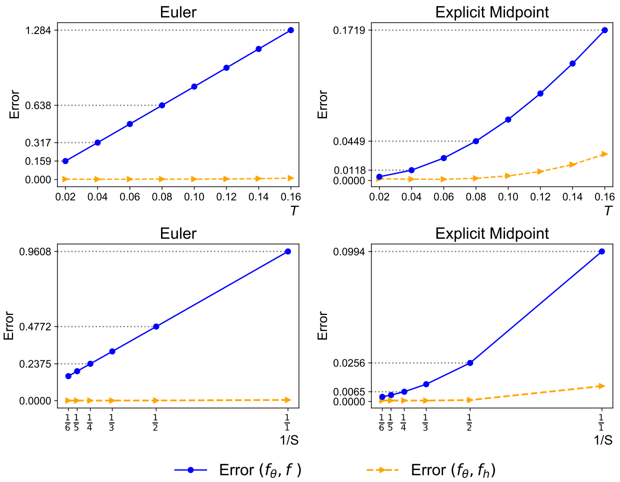

Figure 2. Error versus h for learning the pendulum system. Here, the investigated numerical integrators are the Euler method (of order 1) and the explicit midpoint rule (of order 2). On top row, composition number S is fixed to 1 thus $h = T$. On bottom row, data step T is fixed to 0.12 thus $h = 0.12/S$. The error between f and trained f_θ with respect to h increase linearly for the Euler method and superlinearly for the explicit midpoint rule. More precisely, the error order is 1.01 for the Euler method and 1.97 for the explicit midpoint rule. Experimental details are presented in Appendix C.

Proof. The proof can be found in Appendix B.6. □

According to Theorem 3.2, if training using a coarse discretization (i.e., large $h^{p+1}\delta_f(y)$), then testing using another solver will suffer a numerical error of $\mathcal{O}(h^p)$ besides the learning error. And only if training using sufficiently fine discretization, the trained model has the true ODE interpretation. It is worth mentioning that if Neural ODE are used for the discovery of latent dynamics, Theorem 3.2 provides an upper error bound.

3.4. Learning Hamiltonian Systems

Hamiltonian system is an important category in ordinary differential equations and there have been satisfactory works on learning Hamiltonian systems (Bertalan et al., 2019; Chen & Tao, 2021; Greydanus et al., 2019; Jin et al., 2020). Greydanus et al. (2019) observed drifting of the predicted trajectory when learning a Hamiltonian system using Neural ODE. This observation can be illuminated by IMDE.

A Hamiltonian system is formulated as

$$\frac{d}{dt}y = J^{-1}\nabla H(y), \quad J = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix},$$

where I is $D/2$-by-$D/2$ identity matrix. As discussed above, there exists an IMDE such that formally

$$\phi_{h,J^{-1}\nabla H} = \Phi_{h,f_h}.$$

And training Neural ODE returns an approximation of this IMDE. Therefore, learning Hamiltonian systems, or conservation laws, requires the IMDE to be a Hamiltonian system, i.e., Jf_h is a potential field. This is true only when the numerical integrator used in Neural ODE is symplectic.

Lemma 3.4. *Suppose the true system is a Hamiltonian system. If the employed numerical integrator Φ_h is symplectic, then its IMDE is also a Hamiltonian system, i.e., there locally exist smooth functions H_k, $k = 0, 1, 2, \cdots$, such that*

$$f_k(y) = J^{-1}\nabla H_k(y).$$

If the employed numerical integrator Φ_h is not symplectic, then its IMDE is not a Hamiltonian system, i.e., there exists a constant $\hat{k}$ that only depends on Φ_h such that $Jf_{\hat{k}}$ is not a potential field.

Proof. See Appendix B.7 for the basic concepts of symplectic integration and the detailed proof of this lemma. □

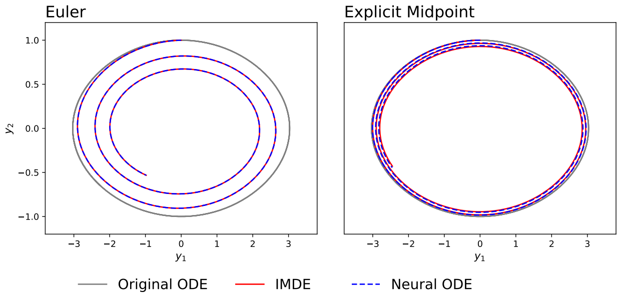

Figure 3. Learning pendulum system using Neural ODE. The dynamics of learned models gradually drift away from the ground truth and accurately match the IMDE. Experimental details are presented in Appendix C.

A non-symplectic numerical integrator cannot guarantee that its IMDE is always a Hamiltonian system. Thus Neural ODE using non-symplectic integration fail to learn conservation laws. Figure 3 validates this statement experimentally. We remark that any explicit Runge-Kutta method[4] can not be symplectic (see e.g., Theorem VI.4.3 of (Hairer et al., 2006)).

3.5. Discussion on Conditions in Theorem 3.1

The generalization requirement, i.e., using $\mathcal{L}$ as an error bound, is in some sense necessary. Otherwise, if the ODE solver is one composition of the implicit Euler method[5], then, the learning model has no information at x. Neural network models have better generalization in practice, thus small $\mathcal{L}$ can be attained and it is reasonable to use $\mathcal{L}$ as a part of the error bound.

The analyticity and boundness requirements are the fundamental assumptions for the estimates in Theorem 3.1, which allow us to use complex analysis techniques to complete the proof. To illustrate their reasonableness, we consider the following two learning tasks.

Example 3.5. Consider learning the differential equation

$$\begin{aligned} &\frac{d}{dt}p = 1,\\ &\frac{d}{dt}q = \sin\left(\frac{2\pi}{h}p + b\right), \end{aligned}$$

with parameter b and initial value $(p(0), q(0)) = (p_0, q_0)$. The exact solution is given as

$$\begin{aligned} &p(t) = p_0 + t,\\ &q(t) = q_0 - \frac{h}{2\pi}\left(\cos\left(\frac{2\pi}{h}(p_0 + t) + b\right) - \cos\left(\frac{2\pi}{h}p_0 + b\right)\right). \end{aligned}$$

Taking $t = h$, we have that

$$\begin{aligned} &p(h) = p_0 + h,\\ &q(h) = q_0. \end{aligned}$$

Thus, same exact solutions are obtained although the parameter b is different, which yields multiple ODE interpretation of training data.

Example 3.6. Consider learning linear equation

$$\frac{d}{dt}p = \lambda p$$

with parameter λ. Applying the explicit Euler method twice yields

$$(\Phi_h)^2(p_0) = (1 + \lambda h)^2 p_0 = (1 + (-2/h - \lambda)h)^2 p_0.$$

Same numerical solutions are obtained for parameters λ and $(-2/h - \lambda)$, indicating that f_θ can approach different targets to minimize loss.

The analyticity and boundness requirements indicate the boundness of derivatives of f and f_θ due to Cauchy's estimate in several variables (see e.g., Section 1.3 of (Scheidemann, 2005)), more precisely,

$$\|D^{\boldsymbol{\alpha}} f\|_{\mathcal{B}(x,r_1)},\ \|D^{\boldsymbol{\alpha}} f_\theta\|_{\mathcal{B}(x,r_1)} \leq \boldsymbol{\alpha}! m r_2^{-|\boldsymbol{\alpha}|}.$$

The requirement for the true ODE excludes counterexamples similar to those in Example 3.5 and implies that our results only hold for low-frequency true ODE interpretation. In addition, the assumption for the learned ODE excludes high-frequency neural network solutions similar to those in Example 3.6. When training Neural ODE, we can enforce

[4] A Runge-Kutta method (2) is explicit if $a_{ij} = 0$ for $i \leq j$ and implicit otherwise. An implicit method has to be calculated iteratively, thus it is not employed in vanilla Neural ODE.

[5]
$$v_1 = x + hf(v_1), \quad \Phi_h(x) = x + hf(v_1).$$

the weights for each layer of f_θ to attain low-frequency assumption of the learned ODE. In classical regression problems, training FNN first captures low-frequency components of the target function and then approximates the high-frequency (Luo et al., 2019; Xu et al., 2019). We conjecture that the implicit regularization is also applied to Neural ODE, and thus the analyticity and boundness assumption of f_θ holds without any explicit regularization.

Experimental results in Figure 1 validate both assumptions, where Neural ODE capture the evolutions of the IMDE without regularization. It is worth mentioning that the assumptions of generalization and analyticity are also required in the global error analysis of GFNN (Chen & Tao, 2021), and their conclusion is also confirmed experimentally.

The requirement of the Runge-Kutta method is not necessary. Theorem 3.1 holds for any numerical integrator satisfying the following assumption.

Assumption 3.7. For analytic g, $\hat{g}$ satisfying $\|g\|_{\mathcal{B}(\mathcal{K},r)} \leq m$, $\|\hat{g}\|_{\mathcal{B}(\mathcal{K},r)} \leq m$, there exist constants b_1, b_2, b_3 that depend only on the Φ_h and S such that

- For $|h| \leq h_0 = b_1 r/m$, $(\Phi_{h,\hat{g}})^S$, $(\Phi_{h,g})^S$ are analytic on $\mathcal{K}$.
- for $|h| \leq h_0$,
$$\left\|(\Phi_{h,\hat{g}})^S - (\Phi_{h,g})^S\right\|_{\mathcal{K}} \leq b_2 h \|\hat{g} - g\|_{\mathcal{B}(\mathcal{K},r)} .$$
- for $|h| < h_1 < h_0$,
$$\begin{aligned}\|\hat{g} - g\|_{\mathcal{K}} \leq & \frac{1}{S|h|} \left\|(\Phi_{h,\hat{g}})^S - (\Phi_{h,g})^S\right\|_{\mathcal{K}} \\ & + \frac{b_2|h|}{h_1 - |h|} \|\hat{g} - g\|_{\mathcal{B}(\mathcal{K},b_3 h_1 m)} .\end{aligned}$$

We will show that Runge-Kutta methods satisfy Assumption 3.7 in Appendix B.2.

4. Conclusion and Limitations

In this paper, we perform numerical analysis on the numerical integration in Neural ODE. The main contribution is that we propose the inverse modified differential equations (IMDE) and prove that training a Neural ODE model actually returns an approximation of the IMDE determined by the learning task and the employed ODE solver, rather than the true ODE. This theorem clarifies the influence of the numerical integration on training Neural ODE models by pointing out the change of approximation target. In addition, we show that the discrepancy between the trained model and the unknown system is bounded by the sum of discretization error Ch^p and learning loss, where h is the discrete step and p is the order of integrator. It provides an upper error bound for the discovery of hidden dynamics using Neural ODE. We also discuss learning the Hamiltonian system. IMDE reveal the potential problems, showing that Neural ODE using non-symplectic integration fail to learn conservation laws. Experimental results support the theoretical analysis.

One limitation of our work is the generalization and boundness requirements on complex space. Quantifying the generalization error and implicit regularization for supervised learning are still open research problems. In addition, the current IMDE is not applicable to adaptive step size selection. We would like to further investigate such problems for Neural ODE in the future.

If the discrete Neural ODE is employed as a discrete model, our assumption that there exists a valid ODE solution does not hold since Neural ODE models are limited in their approximation capabilities (Zhang et al., 2020). In this case, the influence of the numerical integration, together with how to measure this influence, remains unknown.

Like modified differential equations, IMDE only introduce a framework of theoretical analysis of the numerical integration in Neural ODE. Based on the analysis results, improving Neural ODE from the point of view of the ODE solver might be another interesting direction.

Acknowledgements

The authors thank the anonymous reviewers for their valuable comments. This work is supported by the Major Project on New Generation of Artificial Intelligence from MOST of China (Grant No. 2018AAA0101002), and National Natural Science Foundation of China (Grant Nos. 11901564 and 12171466).

References

Anderson, J., Kevrekidis, I., and Rico-Martinez, R. A comparison of recurrent training algorithms for time series analysis and system identification. *Computers & chemical engineering*, 20:S751–S756, 1996.

Arnold, V. I. *Mathematical methods of classical mechanics*, volume 60. Springer Science & Business Media, 2013.

Arnold, V. I., Kozlov, V. V., and Neishtadt, A. I. *Mathematical aspects of classical and celestial mechanics*, volume 3. Springer Science & Business Media, 2007.

Bertalan, T., Dietrich, F., Mezić, I., and Kevrekidis, I. G. On learning hamiltonian systems from data. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(12): 121107, 2019.

Botev, A., Jaegle, A., Wirnsberger, P., Hennes, D., and Higgins, I. Which priors matter? benchmarking models for learning latent dynamics. In *35th Conference on Neural Information Processing Systems (NeurIPS 2021) Track on Datasets and Benchmarks*, 2021.

Burckel, R. B. *An introduction to classical complex analysis*, volume 1. Academic Press, 1980.

Chartier, P., Hairer, E., and Vilmart, G. Numerical integrators based on modified differential equations. *Mathematics of computation*, 76(260):1941–1953, 2007.

Chen, R. and Tao, M. Data-driven prediction of general hamiltonian dynamics via learning exactly-symplectic maps. In *Proceedings of the 38th International Conference on Machine Learning (ICML 2021)*, volume 139, pp. 1717–1727. PMLR, 2021.

Chen, T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. Neural ordinary differential equations. In *32nd Conference on Neural Information Processing Systems (NeurIPS 2018)*, pp. 6572–6583, 2018.

Du, Q., Gu, Y., Yang, H., and Zhou, C. The discovery of dynamics via linear multistep methods and deep learning: Error estimation. *arXiv preprint arXiv:2103.11488*, 2021.

Dupont, E., Doucet, A., and Teh, Y. W. Augmented neural odes. In *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, pp. 3134–3144, 2019.

E, W. A proposal on machine learning via dynamical systems. *Communications in Mathematics and Statistics*, 5 (1):1–11, 2017.

E, W., Han, J., and Li, Q. A mean-field optimal control formulation of deep learning. *Research in the Mathematical Sciences*, 6(1):1–41, 2019.

Eirola, T. Aspects of backward error analysis of numerical odes. *Journal of Computational and Applied Mathematics*, 45(1-2):65–73, 1993.

Feng, K. On difference schemes and symplectic geometry. In *Proceedings of the 5th International Symposium on differential geometry and differential equations, August 1984 Beijing, China*, pp. 42–58. Science Press, Beijing, 1984.

Feng, K. Difference schemes for Hamiltonian formalism and symplectic geometry. *Journal of Computational Mathematics*, 4(3):279–289, 1986.

Feng, K. Formal power series and numerical algorithms for dynamical systems. In *Proceedings of international conference on scientific computation, Hangzhou, China, Series on Appl. Math. Singapore: World Scientific*, volume 1, pp. 28–35, 1991.

Feng, K. Formal dynamical systems and numerical algorithms. *SERIES ON APPLIED MATHEMATICS*, 4:1–10, 1993.

França, G., Jordan, M. I., and Vidal, R. On dissipative symplectic integration with applications to gradient-based optimization. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(4):043402, apr 2021.

González-García, R., Rico-Martìnez, R., and Kevrekidis, I. G. Identification of distributed parameter systems: A neural net based approach. *Computers & chemical engineering*, 22:S965–S968, 1998.

Greydanus, S., Dzamba, M., and Yosinski, J. Hamiltonian neural networks. In *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, pp. 15353–15363, 2019.

Gusak, J., Markeeva, L., Daulbaev, T., Katrutsa, A., Cichocki, A., and Oseledets, I. Towards understanding normalization in neural ODEs. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*, 2020.

Hairer, E. and Lubich, C. The life-span of backward error analysis for numerical integrators. *Numerische Mathematik*, 76(4):441–462, 1997.

Hairer, E., Lubich, C., and Wanner, G. *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*, volume 31. Springer Science & Business Media, 2006.

Howard, R. The gronwall inequality. *lecture notes*, 1998.

Huh, I., Yang, E., Hwang, S. J., and Shin, J. Time-reversal symmetric ODE network. In *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, 2020.

Jia, J. and Benson, A. R. Neural jump stochastic differential equations. In *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, pp. 9843–9854, 2019.

Jin, P., Zhang, Z., Zhu, A., Tang, Y., and Karniadakis, G. E. SympNets: Intrinsic structure-preserving symplectic networks for identifying Hamiltonian systems. *Neural Networks*, 132:166–179, 2020. ISSN 0893-608.

Keller, R. T. and Du, Q. Discovery of dynamics using linear multistep methods. *SIAM Journal on Numerical Analysis*, 59(1):429–455, 2021.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*, 2015.

Krishnapriyan, A. S., Queiruga, A. F., Erichson, N. B., and Mahoney, M. W. Learning continuous models for continuous physics. *arXiv preprint arXiv:2202.08494*, 2022.

Li, Q., Chen, L., Tai, C., and E, W. Maximum principle based algorithms for deep learning. *Journal of Machine Learning Research*, 18(1):5998–6026, 2017.

Lu, Y., Zhong, A., Li, Q., and Dong, B. Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations. In *Proceedings of the 35th International Conference on Machine Learning (ICML 2018)*, volume 80, pp. 3282–3291. PMLR, 2018.

Luo, T., Ma, Z., Xu, Z. J., and Zhang, Y. Theory of the frequency principle for general deep neural networks. *arXiv preprint arXiv:1906.09235*, 2019.

Massaroli, S., Poli, M., Park, J., Yamashita, A., and Asama, H. Dissecting neural odes. In *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, 2020.

Norcliffe, A., Bodnar, C., Day, B., Moss, J., and Liò, P. Neural ODE processes. In *9th International Conference on Learning Representations (ICLR 2021)*, 2021.

Ott, K., Katiyar, P., Hennig, P., and Tiemann, M. Resnet after all: Neural ODEs and their numerical solution. In *9th International Conference on Learning Representations (ICLR 2021)*, 2021.

Pal, A., Ma, Y., Shah, V. B., and Rackauckas, C. V. Opening the blackbox: Accelerating neural differential equations by regularizing internal solver heuristics. In *Proceedings of the 38th International Conference on Machine Learning (ICML 2021)*, volume 139, pp. 8325–8335. PMLR, 2021.

Poli, M., Massaroli, S., Yamashita, A., Asama, H., and Park, J. Hypersolvers: Toward fast continuous-depth models. In *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, 2020.

Queiruga, A. F., Erichson, N. B., Taylor, D., and Mahoney, M. W. Continuous-in-depth neural networks. *arXiv preprint arXiv:2008.02389*, 2020.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. Multistep neural networks for data-driven discovery of nonlinear dynamical systems. *arXiv preprint arXiv:1801.01236*, 2018.

Reich, S. Backward error analysis for numerical integrators. *SIAM Journal on Numerical Analysis*, 36(5):1549–1570, 1999.

Rico-Martinez, R. and Kevrekidis, I. G. Continuous time modeling of nonlinear systems: A neural network-based approach. In *IEEE International Conference on Neural Networks*, pp. 1522–1525. IEEE, 1993.

Rico-Martinez, R., Anderson, J., and Kevrekidis, I. Continuous-time nonlinear signal processing: a neural network based approach for gray box identification. In *Proceedings of IEEE Workshop on Neural Networks for Signal Processing*, pp. 596–605. IEEE, 1994.

Rubanova, Y., Chen, T. Q., and Duvenaud, D. Latent ordinary differential equations for irregularly-sampled time series. In *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, pp. 5321–5331, 2019.

Sanz-Serna, J. M. Symplectic integrators for hamiltonian problems: an overview. *Acta numerica*, 1:243–286, 1992.

Scheidemann, V. *Introduction to complex analysis in several variables*. Springer, 2005.

Sonoda, S. and Murata, N. Transport analysis of infinitely deep neural network. *Journal of Machine Learning Research*, 20(2):1–52, 2019.

Xia, H., Suliafu, V., Ji, H., Nguyen, T. M., Bertozzi, A. L., Osher, S. J., and Wang, B. Heavy ball neural ordinary differential equations. In *35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, 2021.

Xu, Z. J., Zhang, Y., and Xiao, Y. Training behavior of deep neural network in frequency domain. In *Neural Information Processing - 26th International Conference, ICONIP*, pp. 264–274. Springer, 2019.

Yan, H., Du, J., Tan, V., and Feng, J. On robustness of neural ordinary differential equations. In *8th International Conference on Learning Representations, (ICLR,2020)*, 2020.

Yildiz, C., Heinonen, M., and Lähdesmäki, H. ODE2VAE: deep generative second order odes with bayesian neural networks. In *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, pp. 13412–13421, 2019.

Yoshida, H. Recent progress in the theory and application of symplectic integrators. *Qualitative and Quantitative Behaviour of Planetary Systems*, pp. 27–43, 1993.

Yu, H., Tian, X., E, W., and Li, Q. Onsagernet: Learning stable and interpretable dynamics using a generalized onsager principle. *Physical Review Fluids*, 6(11):114402, 2021.

Zhang, H., Gao, X., Unterman, J., and Arodz, T. Approximation capabilities of neural odes and invertible residual networks. In *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*, volume 119, pp. 11086–11095. PMLR, 2020.

Zhang, T., Yao, Z., Gholami, A., Gonzalez, J. E., Keutzer, K., Mahoney, M. W., and Biros, G. ANODEV2: A coupled neural ODE framework. In *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, pp. 5152–5162, 2019.

Zhuang, J., Dvornek, N. C., Li, X., Tatikonda, S., Papademetris, X., and Duncan, J. S. Adaptive checkpoint adjoint method for gradient estimation in neural ODE. In *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*, volume 119, pp. 11639–11649. PMLR, 2020.

A. Calculation of IMDE

A.1. Expanding Exact Solution by Lie Derivatives

Following (Hairer et al., 2006), we briefly introduce Lie derivatives. Given ordinary differential equations

$$\frac{d}{dt}y(t) = f(y(t)),$$

Lie derivative $\mathbf{D}$ is the differential operator defined as:

$$\mathbf{D}g(y) = g'(y)f(y), \quad g : \mathbb{R}^D \to \mathbb{R}^D.$$

According to the chain rule, we have

$$\frac{d}{dt}g(\phi_{t,f}(x)) = (\mathbf{D}g)(\phi_{t,f}(x)),$$

and thus obtain the Taylor series of $g(\phi_{t,f}(x))$ developed at $t = 0$:

$$g(\phi_{t,f}(x)) = \sum_{k=0}^{\infty} \frac{t^k}{k!}(\mathbf{D}^k g)(x).$$

In particular, by setting $t = h$ and $g(y) = I_D(y) = y$, the identity map, it turns to the Taylor series of the exact solution $\phi_{h,f}$ itself, i.e.,

$$\phi_{h,f}(x) = \sum_{k=0}^{\infty} \frac{h^k}{k!}(\mathbf{D}^k I_D)(x) = x + hf(x) + \frac{h^2}{2}f'f(x) + \frac{h^3}{6}(f''(f,f)(x) + f'f'f(x)) + \cdots . \tag{6}$$

Here, the notation $f'(x)$ is a linear map (the Jacobian), the second order derivative $f''(x)$ is a symmetric bilinear map and similarly for higher order derivatives described as tensor, more precisely, for k vector fields $g^1, \cdots, g^k : \mathbb{R}^D \to \mathbb{R}^D$,

$$f^{(k)}(g^1, \cdots, g^k) = \left(\sum_{i_1,\cdots,i_k=1}^{D} \frac{\partial^k f_1}{\partial x_{i_1} \cdots \partial x_{i_k}} g^1_{i_1} \cdots g^k_{i_k}, \ \cdots, \ \sum_{i_1,\cdots,i_k=1}^{D} \frac{\partial^k f_D}{\partial x_{i_1} \cdots \partial x_{i_k}} g^1_{i_1} \cdots g^k_{i_k} \right)^{\top},$$

where the subscript i indicates the i-th component.

A.2. Expanding Runge-Kutta methods

The expansion of numerical solutions has been well-developed in the last few decades (Hairer et al., 2006). Here, we briefly introduce the expansion of Runge-Kutta methods. Given real numbers b_i, a_{ij} $(i, j = 1, \cdots, s)$, a Runge-Kutta method for solving $\frac{d}{dt}y(t) = g(y(t))$ with initial value x is defined as

$$\begin{aligned} v_i &= x + h\sum_{j=1}^{I} a_{ij}g(v_j) \quad i = 1, \cdots, I, \\ \Phi_h(x) &= x + h\sum_{i=1}^{I} b_i g(v_i). \end{aligned} \tag{7}$$

The coefficients a_{ij}, b_i with $i, j = 1, \cdots, I$ fully characterize the method and also the expansion.

Lemma A.1. *The derivatives of the solution of a Runge-Kutta method (7) with respect to x, for $h = 0$, are given by*

$$\left.\frac{\partial^k \Phi_h(x)}{\partial h^k}\right|_{h=0} = \sum_{|\tau|=k} \gamma(\tau) \cdot \alpha(\tau) \cdot \phi(\tau) \cdot F(\tau)(x). \tag{8}$$

Here, τ is called trees and $|\tau|$ is the order of τ (the number of vertices). $\gamma(\tau)$, $\phi(\tau)$, $\alpha(\tau)$ are positive integer coefficients, $F(\tau)(y)$ is called elementary differentials and typically composed of $g(y)$ and its derivatives.

Table 1. Trees, coefficients and elementary differentials in Equation (8)

$\lvert\tau\rvert$	τ	$\gamma(\tau)$	$\alpha(\tau)$	$\phi(\tau)$	$F(\tau)$
1	•	1	1	$\sum_i b_i$	g
2	[•]	2	1	$\sum_{ij} b_i a_{ij}$	$g'g$
3	[•, •]	3	1	$\sum_{ijk} b_i a_{ij} a_{ik}$	$g''(g,g)$
3	[[•]]	6	1	$\sum_{ijk} b_i a_{ij} a_{jk}$	$g'g'g$
4	[•, •, •]	4	1	$\sum_{ijkl} b_i a_{ij} a_{ik} a_{il}$	$g'''(g,g,g)$
4	[[•], •]	8	3	$\sum_{ijkl} b_i a_{ij} a_{ik} a_{jl}$	$g''(g'g,g)$
4	[[•, •]]	12	1	$\sum_{ijkl} b_i a_{ij} a_{jk} a_{jl}$	$g'g''(g,g)$
4	[[[•]]]	24	1	$\sum_{ijkl} b_i a_{ij} a_{jk} a_{kl}$	$g'g'g'g$

Proof. Some $\gamma(\tau), \alpha(\tau), \phi(\tau), F(\tau)$ are reported in Table 1, detailed proof and calculations can be found in Section III.1 of (Hairer et al., 2006). □

Due to Lemma A.1, the formal expansion of a Runge-Kutta method is given by

$$\Phi_{h,g}(x) = y + hd_{1,g}(x) + h^2 d_{2,g}(x) + \cdots,$$

where

$$d_{k,g}(x) = \frac{1}{k!}\frac{\partial^k \Phi_h(x)}{\partial h^k}\Big|_{h=0} = \frac{1}{k!}\sum_{|\tau|=k} \gamma(\tau)\cdot\alpha(\tau)\cdot\phi(\tau)\cdot F(\tau)(x).$$

A.3. Two Examples for Calculating IMDE

The next examples illustrate the process of calculation of IMDE.

Example A.2. Consider the explicit Euler method

$$\Phi_{h,f_h}(x) = x + hf_h(x) = x + h\sum_{k=0}^{\infty} h^k f_k. \tag{9}$$

Here, we simply have $d_{1,f_h} = f_h$ and $d_{j,f_h} = 0$ for all $j \geq 2$.

Comparing equal powers of h in the expression (6) and (9), and setting $y := x$ yields recurrence relations for functions f_j, i.e.,

$$\begin{aligned}
f_0(y) =& f(y),\\
f_1(y) =& \frac{1}{2}f'f(y),\\
f_2(y) =& \frac{1}{6}(f''(f,f)(y) + f'f'f(y)),\\
f_3(y) =& \frac{1}{24}(f'''(f,f,f)(y) + 3f''(f'f,f)(y) + f'f''(f,f)(y) + f'f'f'f(y)),\\
&\vdots
\end{aligned}$$

Example A.3. The explicit midpoint rule

$$v_1 = x + \frac{h}{2}f_h(x), \quad \Phi_{h,f_h}(x) = x + hf_h(v_1)$$

can be expanded as

$$\Phi_{h,f_h}(x) = x + hf_h(x) + \frac{h^2}{2}f_h'f_h(x) + \frac{h^3}{8}f_h''(f_h,f_h)(x) + \frac{h^4}{48}f_h'''(f_h,f_h,f_h)(x) + \cdots$$

according to Lemma A.1. Plugging $f_h = \sum_{k=0}^{\infty} h^k f_k$ yields

$$\begin{aligned}\Phi_{h,f_h}(x) = x + hf_0(x) + h^2\big(f_1(x) + \frac{1}{2}f_0'f_0(x)\big) + h^3\big(f_2(x) + \frac{1}{2}f_1'f_0(x) + \frac{1}{2}f_0'f_1(x) + \frac{1}{8}f_0''(f_0, f_0)(x)\big)\\ + h^4\big(f_3(x) + \frac{1}{2}f_1'f_1(x) + \frac{1}{2}f_0'f_2(x) + \frac{1}{2}f_2'f_0(x) + \frac{1}{8}f_1''(f_0, f_0)(x) + \frac{1}{4}f_0''(f_1, f_0)(x) + \frac{1}{48}f_0'''(f_0, f_0, f_0)(x)\big) + \cdots\end{aligned} \tag{10}$$

Comparing equal powers of h in the expression (6) and (10), and setting $y := x$ yields recurrence relations for functions f_j, viz.,

$$\begin{aligned}f_0(y) =& f(y),\\ f_1(y) =& 0,\\ f_2(y) =& \frac{1}{24}f''(f, f)(y) + \frac{1}{6}f'f'f(y),\\ f_3(y) =& -\frac{1}{16}f'f''(f, f)(y) - \frac{1}{8}f'f'f'f(y),\\ &\vdots\end{aligned}$$

We remark that calculating IMDE is one of the steps for constructing modified integrator (Chartier et al., 2007), where an explicit recurrence formula based on B-series is given.

B. Proofs

The proofs rely on the definition of IMDE, the induction idea and some complex analysis techniques such as the maximum principle and Cauchy's estimate (see e.g., (Burckel, 1980)).

B.1. Properties of IMDE

The ODE solver, i.e., S compositions of an integrator Φ_h, can be regarded as a one-step integrator with discrete step Sh and thus has its IMDE. The following lemma indicates that the IMDE of the ODE solver coincides with the IMDE of Φ_h.

Lemma B.1. *For any fixed composition number S, suppose that the vector fields of the IMDE of Φ_h and $(\Phi_h)^S$ are $f_h(y) = \sum_{k=0}^{\infty} h^k f_k(y)$ and $F_{Sh}(y) = \sum_{k=0}^{\infty}(Sh)^k F_k(y)$, respectively. Then, for any integer K, $f_h^K = F_{Sh}^K$.*

Proof. The calculation procedure of f_h uniquely defines the functions f_k and can be rewritten as the following recursion:

$$h^{k+1}f_k = \phi_{h,f} - \Phi_{h,f_h^{k-1}} + \mathcal{O}(h^{k+2}). \tag{11}$$

We first prove

$$Sh^{k+1}f_k = \phi_{Sh,f} - \left(\Phi_{h,f_h^{k-1}}\right)^S + \mathcal{O}(h^{k+2}), \tag{12}$$

by induction on $S \geq 1$. First, the case when $S = 1$ is obvious. Suppose now that the statement holds for $S - 1$. Then, by this inductive hypothesis, we obtain

$$\begin{aligned}\phi_{Sh,f} - \left(\Phi_{h,f_h^{k-1}}\right)^S =& \Phi_{h,f_h^{k-1}} \circ \left(\phi_{(S-1)h,f} - \left(\Phi_{h,f_h^{k-1}}\right)^{S-1}\right) + \left(\phi_{h,f} - \Phi_{h,f_h^{k-1}}\right) \circ \phi_{(S-1)h,f}\\ =& (S-1)h^{k+1}\Phi_{h,f_h^{k-1}} \circ f_k + h^{k+1}f_k \circ \phi_{(S-1)h,f} + \mathcal{O}(h^{k+2})\\ =& Sh^{k+1}f_k + \mathcal{O}(h^{k+2}),\end{aligned}$$

where we have used the fact that

$$\phi_{(S-1)h,f} = I_D + \mathcal{O}\left((S-1)h\right), \ \Phi_{h,f_h^{k-1}} = I_D + \mathcal{O}(h).$$

Hence the induction is completed.

We next prove that $S^k F_k = f_k$ by induction on k. First the case when $k=0$ is obvious since $F_0 = f_0 = f$. Suppose now $S^k F_k = f_k$ holds for $k \leq K-1$. This inductive hypothesis implies that $F_{Sh}^{K-1} = f_h^{K-1}$. Using (11) for F_K we obtain

$$(Sh)^{K+1} F_K = \phi_{Sh,f} - \left(\Phi_{h,F_{Sh}^{K-1}}\right)^S + \mathcal{O}(h^{K+2}) = \phi_{Sh,f} - \left(\Phi_{h,f_h^{K-1}}\right)^S + \mathcal{O}(h^{K+2}).$$

This together with (12) concludes the induction and thus completes the proof. □

Lemma B.2. *Suppose that*

$$\Phi_{h,f}(x) = \phi_{h,f}(x) + h^{p+1}\delta_f(x) + \mathcal{O}(h^{p+2}).$$

Then, the IMDE obeys

$$\frac{d}{dt}\tilde{y} = f_h(\tilde{y}) = f(\tilde{y}) - h^p \delta_f(\tilde{y}) + \cdots .$$

Proof. We prove that $f_k = 0$ for $k \leq p-1$ and $f_k = -\delta_f$ for $k = p$ by induction on k. By Equation (11), we have

$$h^2 f_1 = \phi_{h,f} - \Phi_{h,f} + \mathcal{O}(h^3) = -h^{p+1}\delta_f + \mathcal{O}(h^3).$$

Thus $f_1 = 0$ if $1 \leq p-1$ and $f_1 = -\delta_f$ if $1 = p$. Suppose now the the hypothesis holds for $k \leq K-1 < p$. The function f_K is obtained from

$$h^{K+1} f_K = \phi_{h,f} - \Phi_{h,f_h^{K-1}} + \mathcal{O}(h^{K+2}) = \phi_{h,f} - \Phi_{h,f} + \mathcal{O}(h^{K+2}) = -h^{p+1}\delta_f + \mathcal{O}(h^{K+2}).$$

Thus $f_K = 0$ if $K \leq p-1$ and $f_K = -\delta_f$ if $K = p$. The proof is completed. □

B.2. Properties of Runge-Kutta Methods

We first consider the case $S = 1$ in Assumption 3.7, i.e.,

$$\begin{aligned} &1. \text{ Analytic for } |h| \leq h_0 = b_1 r/m \text{ and } y \in \mathcal{K}, \\ &2. \left\|\Phi_{h,\hat{g}} - \Phi_{h,g}\right\|_{\mathcal{K}} \leq b_2 h \left\|\hat{g} - g\right\|_{\mathcal{B}(\mathcal{K},r)}, \text{ for } |h| \leq h_0, \\ &3. \left\|\hat{g} - g\right\|_{\mathcal{K}} \leq \frac{1}{|h|}\left\|\Phi_{h,\hat{g}} - \Phi_{h,g}\right\|_{\mathcal{K}} + \frac{b_2|h|}{h_1 - |h|}\left\|\hat{g} - g\right\|_{\mathcal{B}(\mathcal{K},b_3 h_1 m)}, \text{ for } |h| < h_1 \leq h_0. \end{aligned} \tag{13}$$

And we prove that the condition (13) is satisfied for Runge-Kutta methods (2).

Lemma B.3. *For a consistent Runge-Kutta method (2) denoted as Φ_h, let*

$$\mu = \sum_{i=1}^{I} |b_i|, \quad \kappa = \max_{1 \leq i \leq s} \sum_{j=1}^{I} |a_{ij}|.$$

Consider analytic g, $\hat{g}$ satisfying $\|g\|_{\mathcal{B}(\mathcal{K},r)} \leq m$, $\|\hat{g}\|_{\mathcal{B}(\mathcal{K},r)} \leq m$, if $\kappa \neq 0$, then $\Phi_{h,g}$, $\Phi_{h,\hat{g}}$ are analytic for $|h| \leq h_0 = r/(4\kappa m)$ and

$$\left\|\Phi_{h,\hat{g}} - \Phi_{h,g}\right\|_{\mathcal{K}} \leq 2\mu|h| \left\|\hat{g} - g\right\|_{\mathcal{B}(\mathcal{K},|h|\kappa m)}.$$

Furthermore, for $|h| < h_1 \leq h_0$,

$$\left\|\hat{g} - g\right\|_{\mathcal{K}} \leq \frac{\left\|\Phi_{h,\hat{g}} - \Phi_{h,g}\right\|_{\mathcal{K}}}{|h|} + \frac{2\mu|h| \left\|\hat{g} - g\right\|_{\mathcal{B}(\mathcal{K},h_1\kappa m)}}{h_1 - |h|}.$$

Proof. For $y \in \mathcal{B}(\mathcal{K}, r/2)$ and $\|\Delta y\| \leq 1$, the function $\alpha(z) = g(y + z\Delta y)$ is analytic for $|z| \leq r/2$ and bounded by m. By Cauchy's estimate, we obtain

$$\|g'(y)\Delta y\| = \|\alpha'(0)\| \leq 2m/r,$$

and $\|g'(y)\| \leq 2m/r$ for $y \in \mathcal{B}(\mathcal{K}, r/2)$ in the operator norm.

For a Runge-Kutta method (2) with initial point $x \in \mathcal{K}$, the solution can be obtained by the nonlinear systems

$$u_i = x + h\sum_{j=1}^{I} a_{ij}\hat{g}(u_j) \quad i = 1, \cdots, I, \quad \Phi_{h,\hat{g}}(x) = x + h\sum_{i=1}^{I} b_i\hat{g}(u_i),$$
$$v_i = x + h\sum_{j=1}^{I} a_{ij}g(v_j) \quad i = 1, \cdots, I, \quad \Phi_{h,g}(x) = x + h\sum_{i=1}^{I} b_i g(v_i).$$

Due to the Implicit Function Theorem (Scheidemann, 2005), u_i, v_i possess unique solutions on the closed set $\mathcal{B}(\mathcal{K}, |h|\kappa m)$ if $2|h|\kappa m/r \leq \gamma < 1$ and the method is analytic for $|h| \leq \gamma r/2\kappa m$.

In addition,

$$\begin{aligned}\|u_i - v_i\| \leq& |h|\sum_{j=1}^{I}|a_{ij}|(\|\hat{g}(u_j) - \hat{g}(v_j)\| + \|\hat{g}(v_j) - g(v_j)\|)\\ \leq& |h|\kappa\frac{2m}{r}\max_{1\leq j\leq I}\|u_j - v_j\| + |h|\kappa\left\|\hat{g} - g\right\|_{\mathcal{B}(\mathcal{K},|h|\kappa m)}.\end{aligned}$$

Thus we obtain

$$\max_{1\leq i\leq I}\|u_i - v_i\| \leq \frac{\kappa}{1 - |h|\kappa\frac{2m}{r}}|h|\left\|\hat{g} - g\right\|_{\mathcal{B}(\mathcal{K},|h|\kappa m)}.$$

Next, we have

$$\begin{aligned}\|\Phi_{h,\hat{g}}(x) - \Phi_{h,g}(x)\| \leq& |h|\sum_{i=1}^{I}|b_i|\left\|\hat{g}(u_i) - \hat{g}(v_i)\right\| + |h|\sum_{i=1}^{I}|b_i|\left\|\hat{g}(v_i) - g(v_i)\right\|\\ \leq& |h|\mu\frac{2m}{r}\max_{1\leq j\leq s}\|u_j - v_j\| + |h|\mu\left\|\hat{g} - g\right\|_{\mathcal{B}(\mathcal{K},|h|\kappa m)}\\ \leq& \left(|h|\mu\frac{2m}{r}\frac{\kappa}{1 - |h|\kappa\frac{2m}{r}} + \mu\right)|h|\left\|\hat{g} - g\right\|_{\mathcal{B}(\mathcal{K},|h|\kappa m)}.\end{aligned}$$

Taking $\gamma = 1/2$, together with the arbitrariness of x, yields

$$\left\|\Phi_{h,\hat{g}} - \Phi_{h,g}\right\|_{\mathcal{K}} \leq 2\mu|h|\left\|\hat{g} - g\right\|_{\mathcal{B}(\mathcal{K},|h|\kappa m)}.$$

These complete the first part of the proof.

Finally, using Cauchy's estimate, we deduce that for $h_1 \leq h_0$,

$$\left\|\frac{d^i}{dh^i}\left(\Phi_{h,\hat{g}}(x) - \Phi_{h,g}(x)\right)\Big|_{h=0}\right\| \leq \frac{i!2\mu\left\|\hat{g} - g\right\|_{\mathcal{B}(\mathcal{K},h_1\kappa m)}}{h_1^{i-1}}.$$

By the analyticity and triangle inequality, we obtain for $|h| < h_1$,

$$\begin{aligned}\|\Phi_{h,\hat{g}}(x) - \Phi_{h,g}(x)\| \geq& |h|\left\|\hat{g}(x) - g(x)\right\| - \sum_{i=2}^{\infty}\left\|\frac{h^i}{i!}\frac{d^j}{dh^j}\left(\Phi_{h,\hat{g}}(x) - \Phi_{h,g}(x)\right)\Big|_{h=0}\right\|\\ \geq& |h|\left\|\hat{g}(x) - g(x)\right\| - 2\mu|h|\left\|\hat{g} - g\right\|_{\mathcal{B}(\mathcal{K},h_1\kappa m)}\sum_{i=2}^{\infty}\left(\frac{|h|}{h_1}\right)^{i-1}.\end{aligned}$$

Therefore,

$$\left\|\hat{g} - g\right\|_{\mathcal{K}} \leq \frac{\left\|\Phi_{h,\hat{g}} - \Phi_{h,g}\right\|_{\mathcal{K}}}{|h|} + \frac{2\mu|h|\left\|\hat{g} - g\right\|_{\mathcal{B}(\mathcal{K},h_1\kappa m)}}{h_1 - |h|},$$

which concludes the proof. □

We could easily check that for the case $\kappa = 0$, i.e., the Euler method, condition (13) also holds.

If we apply a Runge-Kutta method Φ_h with coefficients a_{ij}, b_i, step h and initial value x, then apply another Runge-Kutta method $\hat{\Phi}_{\hat{h}}$ with coefficients $\hat{a}_{ij}, \hat{b}_i$, step $\hat{h}$ and initial value $\Phi_h(x)$, this composition of two methods can be regarded as a single Runge-Kutta method with discrete step $h+\hat{h}$:

$$
\begin{aligned}
&v_i = x + (h+\hat{h}) \sum_{j=1}^{I} \frac{a_{ij}h}{h+\hat{h}} g(v_j) \quad i = 1, \cdots, I,\\
&v_i = x + (h+\hat{h}) \sum_{j=1}^{I} \frac{b_j h}{h+\hat{h}} g(v_j) + (h+\hat{h}) \sum_{j=I+1}^{I+\hat{I}} \frac{\hat{a}_{ij}\hat{h}}{h+\hat{h}} g(v_j) \quad i = I+1, \cdots, I+\hat{I},\\
&\hat{\Phi}_{\hat{h}} \circ \Phi_h(x) = x + (h+\hat{h}) \sum_{j=1}^{I} \frac{b_j h}{h+\hat{h}} g(v_j) + (h+\hat{h}) \sum_{j=I+1}^{I+\hat{I}} \frac{\hat{b}_j\hat{h}}{h+\hat{h}} g(v_j).
\end{aligned}
$$

According to this fact, we obtain that, under the notations and conditions of Lemma B.3, $(\Phi_{h,g})^S$, $(\Phi_{h,\hat{g}})^S$ are analytic for $S|h| \le h_0' = r/(4\frac{\kappa+(S-1)\mu}{S}m)$ and

$$
\left\|(\Phi_{h,\hat{g}})^S - (\Phi_{h,g})^S\right\|_{\mathcal{K}} \le 2S\mu|h| \left\|\hat{g}-g\right\|_{\mathcal{B}(\mathcal{K},|h|(\kappa+(S-1)\mu)m)}.
$$

Furthermore, for $S|h| < h_1 \le h_0'$,

$$
\left\|\hat{g}-g\right\|_{\mathcal{K}} \le \frac{\left\|(\Phi_{h,\hat{g}})^S - (\Phi_{h,g})^S\right\|_{\mathcal{K}}}{S|h|} + \frac{2S\mu|h| \left\|\hat{g}-g\right\|_{\mathcal{B}(\mathcal{K},h_1\frac{\kappa+(S-1)\mu}{S}m)}}{h_1 - S|h|}.
$$

Tanking $h_0 := h_0'/S$ implies that Runge-Kutta methods satisfy Assumption 3.7 with $b_1 = \frac{1}{4(\kappa+(S-1)\mu)}$, $b_2 = 2S\mu$, $b_3 = \kappa + (S-1)\mu$.

B.3. Estimation of the Truncation

The series in (3) does not converge in general and needs to be truncated. Inspired by the induction idea for conventional modified equations in (Reich, 1999), we prove the truncation estimation for the IMDE scenario below.

Lemma B.4. *Let $f(y)$ be analytic in $\mathcal{B}(\mathcal{K},r)$ and satisfies $\|f\|_{\mathcal{B}(\mathcal{K},r)} \le m$. Suppose the pth-order numerical integrator Φ_h satisfies condition (13). Take $\eta = \max\{6, \frac{b_2+1}{29}+1\}$, $\zeta = 10(\eta-1)$, $q = -\ln(2b_2)/\ln 0.912$ and let K be the largest integer satisfying*

$$
\frac{\zeta(K-p+2)^q|h|m}{\eta r} \le e^{-q}.
$$

If $|h|$ is small enough such that $K \ge p$, then the truncated IMDE satisfies

$$
\begin{aligned}
&\left\|\Phi_{h,f_h^K} - \phi_{h,f}\right\|_{\mathcal{K}} \le b_2\eta m e^{2q-qp}|h|e^{-\gamma/|h|^{1/q}},\\
&\left\|f_h^K - f\right\|_{\mathcal{K}} \le b_2\eta m \left(\frac{\zeta m}{b_1 r}\right)^p (1+1.38^q d_p)|h|^p,\\
&\left\|f_h^K\right\|_{\mathcal{K}} \le (\eta-1)m,
\end{aligned}
$$

where $\gamma = \frac{q}{e}\left(\frac{b_1 r}{\zeta m}\right)^{1/q}$, $d_p = p^{qp}e^{-q(p-1)}$.

Proof. For $0 \le \alpha < 1$ and $|h| \le h_0 = b_1(1-\alpha)r/m$, the condition (13), together with the fact that $\mathcal{B}(\mathcal{B}(\mathcal{K},\alpha r),(1-\alpha)r) = \mathcal{B}(\mathcal{K},r)$ implies

$$
\begin{aligned}
\left\|\Phi_{h,f} - \phi_{h,f}\right\|_{\mathcal{B}(\mathcal{K},\alpha r)} \le& \left\|\Phi_{h,f} - I_D\right\|_{\mathcal{B}(\mathcal{K},\alpha r)} + \left\|\phi_{h,f} - I_D\right\|_{\mathcal{B}(\mathcal{K},\alpha r)}\\
\le&(b_2+1)|h|m \le b_1(b_2+1)(1-\alpha)r.
\end{aligned}
$$

Here, the map $\Phi_{h,f} - \phi_{h,f}$ contains the factor h^{p+1} since $\Phi_{h,f}$ is of order p. By the maximum principle for analytic functions, we obtain that

$$
\left\|\frac{\Phi_{h,f} - \phi_{h,f}}{h^{p+1}}\right\|_{\mathcal{B}(\mathcal{K},\alpha r)} \le \frac{b_1(b_2+1)(1-\alpha)r}{h_0^{p+1}}.
$$

The calculation procedure of f_k can be rewritten as the following recursion:

$$f_k = \lim_{h\to 0} \frac{\phi_{h,f} - \Phi_{h,f_h^{k-1}}}{h^{k+1}}. \tag{14}$$

Therefore, we deduce that

$$\|f_p\|_{\mathcal{B}(\mathcal{K},\alpha r)} \leq \frac{b_1(b_2+1)(1-\alpha)r}{h_0^{p+1}} = (b_2+1)m\left(\frac{m}{b_1(1-\alpha)r}\right)^p. \tag{15}$$

Below we proceed to prove that for $\alpha \in [0,1)$, if

$$|h| \leq h_k := \frac{b_1(1-\alpha)r}{\zeta(k-p+1)^q m},$$

then

$$\|f_k\|_{\mathcal{B}(\mathcal{K},\alpha r)} \leq b_2\eta m\left(\frac{\zeta(k-p+1)^q m}{b_1(1-\alpha)r}\right)^k \tag{16}$$

for $k \geq p$ by induction, where $\eta = \max\{6, \frac{b_2+1}{29}+1\}$, $\zeta = 10(\eta-1)$ and $q = -\ln(2b_2)/\ln 0.912$. First, the case when $k = p$ is obvious since inequality (15). Suppose now (16) holds for $k \leq K$. If $|h| \leq h_{K+1}$, taking

$$\delta_{K+1} := \frac{\eta-1}{(K-p+2)^q\zeta},\ \beta_K := (1-\delta_{K+1})(K-p+2)^q$$

yields that for $p \leq k \leq K$,

$$|h| \leq \frac{b_1(1-\alpha)r}{\zeta(K-p+2)^q m} \leq \frac{b_1(1-\alpha-\delta_{K+1}(1-\alpha))r}{\zeta(k-p+1)^q m}.$$

Therefore, by inductive hypothesis and replacing α by $\alpha+\delta_{K+1}(1-\alpha) \in [\delta_{K+1},1)$ in (16), we obtain

$$\|f_k\|_{\mathcal{B}(\mathcal{K},(\alpha+\delta_{K+1}(1-\alpha))r)} \leq b_2\eta m\left(\frac{\zeta(k-p+1)^q m}{b_1(1-\delta_{K+1})(1-\alpha)r}\right)^k.$$

This indicates that

$$\left\|f_h^K\right\|_{\mathcal{B}(\mathcal{K},(\alpha+\delta_{K+1}(1-\alpha))r)} \leq m\left[1+(b_2+1)\left(\frac{1}{\zeta\beta_K}\right)^p + b_2\eta\sum_{k=p+1}^{K}\left(\frac{(k-p+1)^q}{\beta_K}\right)^k\right].$$

Since

$$\sum_{k=p+1}^{K}\left(\frac{k-p+1}{K-p+1.9}\right)^k \leq 0.912,$$

which is maximal for $K = 6$ and $p = 1$, and

$$\sum_{k=p+1}^{K}\left(\frac{(k-p+1)^q}{\beta_K}\right)^k \leq \left[\sum_{k=p+1}^{K}\left(\frac{k-p+1}{K-p+1.9}\right)^k\right]^q,$$

we deduce that

$$\left\|f_h^K\right\|_{\mathcal{B}(\mathcal{K},(\alpha+\delta_{K+1}(1-\alpha))r)} \leq (\eta-1)m. \tag{17}$$

Here, we have used the definition of η, ζ and q. Subsequently, by this estimate and condition (13), we obtain

$$\left\|\Phi_{h,f_h^K} - I_D\right\|_{\mathcal{B}(\mathcal{K},\alpha r)} \leq |h|b_2\left\|f_h^K\right\|_{\mathcal{B}(\mathcal{K},(\alpha+\delta_{K+1}(1-\alpha))r)} \leq |h|b_2(\eta-1)m,$$

where

$$|h| \leq h_K = \frac{b_1(1-\alpha)r}{\zeta(K-p+2)^q m} = \frac{b_1\delta_{K+1}(1-\alpha)r}{(\eta-1)m}.$$

And then using the triangle inequality yields that

$$\left\|\Phi_{h,f_h^K}-\phi_{h,f}\right\|_{\mathcal{B}(\mathcal{K},\alpha r)} \leq \left\|\Phi_{h,f_h^K}-I_D\right\|_{\mathcal{B}(\mathcal{K},\alpha r)} + \left\|\phi_{h,f}-I_D\right\|_{\mathcal{B}(\mathcal{K},\alpha r)} \leq h_K b_2 \eta m.$$

Again by the maximum principle for analytic functions, together with the fact that $\Phi_{h,f_h^K}-\phi_{h,f}$ contains the factor h^{K+2}, we deduce that

$$\left\|\frac{\Phi_{h,f_h^K}-\phi_{h,f}}{h^{K+2}}\right\|_{\mathcal{B}(\mathcal{K},\alpha r)} \leq \frac{h_K b_2 \eta m}{h_K^{K+2}}. \tag{18}$$

Again by (14), we conclude that

$$\|f_{K+1}\|_{\mathcal{B}(\mathcal{K},\alpha r)} \leq b_2\eta m\left(\frac{\zeta(K-p+2)^q m}{b_1(1-\alpha)r}\right)^{K+1},$$

which completes the induction.

The above induction also shows that (18) holds if $|h| \leq h_{K+1}$. Taking $\alpha = 0$ we have

$$\left\|\Phi_{h,f_h^K}-\phi_{h,f}\right\|_{\mathcal{K}} \leq b_2\eta|h|m\left(\frac{\zeta(K-p+2)^q|h|m}{b_1 r}\right)^{K+1}.$$

We set K^* to be the largest integer satisfying

$$\frac{\zeta(K^*-p+2)^q|h|m}{b_1 r} \leq e^{-q}.$$

Clearly, $|h| \leq h_{K^*+1}$ with $\alpha = 0$. Therefore,

$$\begin{aligned}\left\|\Phi_{h,f_h^{K^*}}-\phi_{h,f}\right\|_{\mathcal{K}} \leq & b_2\eta|h|m\left(\frac{\zeta(K^*-p+2)^q|h|m}{b_1 r}\right)^{K^*+1} \\ \leq & b_2\eta m e^{2q-qp}|h|e^{-\gamma/|h|^{1/q}},\end{aligned}$$

where $\gamma = \frac{q}{e}\left(\frac{b_1 r}{\zeta m}\right)^{1/q}$. The first part of the lemma has been completed.

Next, according to (16) we obtain

$$\begin{aligned}\left\|f_h^{K^*}-f\right\|_{\mathcal{K}} \leq & b_2\eta m\left(\frac{\zeta|h|m}{b_1 r}\right)^p\Big[1+\sum_{k=p+1}^{K^*}\frac{(k-p+1)^{qp}}{e^{q(k-p)}}\left(\frac{k-p+1}{K^*-p+2}\right)^{q(k-p)}\Big] \\ \leq & b_2\eta m\left(\frac{\zeta m}{b_1 r}\right)^p(1+1.38^q d_p)|h|^p,\end{aligned}$$

where $d_p = p^{qp}e^{-q(p-1)}$ satisfies $d_p \geq (k-p+1)^{qp}e^{-q(k-p)}$ for any $k \geq p+1$.

Finally, we immediately derive the bound of f_h^K due to (17). The proof has been completed. □

B.4. Proof of Theorem 3.1

Proof. Since the ODE solver satisfy Assumption 3.7, regarding the ODE solver as a one-step integrator and applying the first inequality of Lemma B.4, we have that if $T \leq T_0 := \eta r_2/((2e)^q\zeta m)$,

$$\left\|\left(\Phi_{h,F_{Sh}^K}\right)^S-\phi_{Sh,f}\right\|_{\mathcal{B}(x,r_1)} \leq b_2\eta m e^q She^{-\gamma'/(Sh)^{1/q}},$$

where $h > 0$, b_2 is the coefficient defined in Assumption 3.7 and η, q, γ are given by Lemma B.4

By Lemma B.1, $F_{Sh}^K = f_h^K$. And thus we obtain that

$$\delta := \frac{1}{Sh}\left\|\left(\Phi_{h,f_\theta}\right)^S - \left(\Phi_{h,f_h^K}\right)^S\right\|_{\mathcal{B}(x,r_1)} \le \mathcal{L} + cme^{-\gamma/h^{1/q}}, \tag{19}$$

where $\gamma = \gamma'/S^{1/q} = \frac{q}{e}\left(\frac{b_1 r_2}{S\zeta m}\right)^{1/q}$, $c = b_2\eta e^q$. Next, by the third inequality of Lemma B.4, $\left\|f_h^K\right\|_{\mathcal{B}(x,r_1)} < (\eta-1)m$. Let

$$h_1 = (eb_2+1)h,\ \lambda = \frac{b_2 h}{h_1 - h} = e^{-1},\ M = (\eta-1)m.$$

Using the third item of Assumption 3.7, we deduce that for $0 \le j \le r_1/h_1 b_3 M$,

$$\left\|f_\theta - f_h^K\right\|_{\mathcal{B}(x,jh_1b_3M)} \le \delta + \lambda\left\|f_\theta - f_h^K\right\|_{\mathcal{B}(x,(j+1)h_1b_3M)}.$$

This yields

$$\left\|f_\theta - f_h^K\right\|_{\mathcal{B}(x,jh_1b_3M)} - \frac{\delta}{1-\lambda} \le \lambda\left(\left\|f_\theta - f_h^K\right\|_{\mathcal{B}(x,(j+1)h_1b_3M)} - \frac{\delta}{1-\lambda}\right).$$

Using this estimate iteratively, we deduce that

$$\left\|f_\theta(x) - f_h^K(x)\right\| \le e^{-\hat{\gamma}/h}\left\|f_\theta - f_h^K\right\|_{\mathcal{B}(x,r_1)} + \frac{\delta}{1-\lambda},$$

where $\hat{\gamma} = \frac{r_1}{(eb_2+1)b_3M}$. By this estimation and (19), we conclude that

$$\left\|f_\theta(x) - f_h^K(x)\right\| \le c_1 me^{-\gamma/h^{1/q}} + C\mathcal{L},$$

where $C = e/(e-1)$ and c_1 is a constant satisfying $c_1 \ge C\cdot c + \eta e^{\gamma/h^{1/q} - \hat{\gamma}/h}$. □

B.5. Proof of Theorem 3.2

Proof. The first part has been proved in Lemma B.2, and the second part is a direct consequence of Theorem 3.1, the second inequality of Lemma B.4 and Triangle Inequality. □

B.6. Proof of Corollary 3.3.

We first state a version of the well-known Grönwall's Inequality (Howard, 1998).

Proposition B.5. *Let $U \subset \mathbb{R}^D$ be an open set, Let $g_1, g_2 : U \to \mathbb{R}^D$ be continuous functions and let $y, \tilde{y} : [t_0, t_1] \to U$ satisfy*

$$\frac{d}{dt}y(t) = f(y(t)),\ y(0) = x, \quad \textit{and} \quad \frac{d}{dt}\tilde{y}(t) = \tilde{f}(\tilde{y}(t)),\ \tilde{y}(0) = \tilde{x}.$$

Assume $\tilde{f}$ is Lipschitz with Lipschitz constant C_1 and $\left\|f(y(t)) - \tilde{f}(y(t))\right\| \le \varphi(t)$ for continuous function φ. Then, for $t \in [t_0, t_1]$,

$$\|y(t) - \tilde{y}(t)\| \le e^{C_1(t-t_0)}\|x - \tilde{x}\| + e^{C_1(t-t_0)}\int_{t_0}^t e^{-C_1(\tau - t_0)}\varphi(\tau)d\tau.$$

Proof of Corollary 3.3. Consider the following two equations

$$\frac{d}{dt}y(t) = f(y(t)),\ y(0) = x, \quad \text{and} \quad \frac{d}{dt}\tilde{y}(t) = f_\theta(\tilde{y}(t)),\ \tilde{y}(0) = x.$$

We denote the set of the points on exact trajectory as $V_t = \{\phi_{\tau,f}(x) | 0 \le \tau \le t\}$. By Theorem 3.2, there exist constant C_2 such that

$$\|f(y(t)) - f_\theta(y(t))\| \le C_2(h^p + \mathcal{L}_t), \quad \text{where } \mathcal{L}_t = \left\|\left(\Phi_{h,f_\theta}\right)^S - \phi_{Sh,f}\right\|_{\mathcal{B}(V_t,r_1)}/(Sh).$$

Therefore, by Proposition B.5,

$$\|\phi_{t,f}(x) - \phi_{t,f_\theta}(x)\| \le e^{C_1 t}\int_0^t e^{-C_1\tau}C_2(h^p + \mathcal{L}_\tau)d\tau \le C_2(h^p + \mathcal{L}_t)\cdot e^{C_1 t}\int_0^t e^{-C_1\tau}d\tau \le \frac{C_2 e^{C_1 t} - C_2}{C_1}(h^p + \mathcal{L}_t),$$

which concludes the proof. □

B.7. Proof of Lemma 3.4

For even dimension D, denote the $D/2$-by-$D/2$ identity matrix by I, and let

$$J = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}.$$

Definition B.6. A differentiable map $g : U \to \mathbb{R}^D$ (where D is even and $U \subseteq \mathbb{R}^D$ is an open set) is called symplectic if

$$g'(x)^T J g'(x) = J,$$

where $g'(x)$ is the Jacobian of $g(x)$.

A Hamiltonian system is given by

$$\frac{d}{dt} y = J^{-1} \nabla H(y), \quad y(0) = x, \tag{20}$$

where $y \in \mathbb{R}^D$ and H is the Hamiltonian function typically representing the energy of (20) (Arnold, 2013; Arnold et al., 2007). A remarkable property of Hamiltonian system is the symplecticity of the phase flow, which was proved by Poincaré in 1899 (Arnold, 2013), i.e.,

$$\phi_t'(x)^T J \phi_t'(x) = J,$$

where $\phi_t'(x) = \frac{\partial \phi_t(x)}{\partial x}$ is the Jacobian of ϕ_t. Due to the intrinsic symplecticity, it is natural to search for numerical methods that preserve symplecticity, i.e., make Φ_h be a symplectic map. Such numerical methods are called symplectic methods, see e.g., (Feng, 1984; 1986; Hairer et al., 2006).

Proof of Lemma 3.4. For a Hamiltonian system (20), the target function f obeys $f(y) = J^{-1}\nabla H(y)$, which yields $f_0 = J^{-1}\nabla H(y)$. If the employed numerical integrator Φ_h is symplectic, suppose $f_k(y) = J^{-1}\nabla H_k(y)$ for $k = 1, 2, \cdots, K$, we need to prove the existence of $H_{K+1}(y)$ satisfying

$$f_{K+1}(y) = J^{-1}\nabla H_{K+1}(y).$$

By induction, the truncated IMDE

$$\frac{d}{dt}\tilde{y} = f_h^K(\tilde{y}) = f(\tilde{y}) + hf_1(\tilde{y}) + h^2 f_2(\tilde{y}) + \cdots + h^K f_K(\tilde{y})$$

has the Hamiltonian $H(\tilde{y}) + hH_1(\tilde{y}) + \cdots + h^K H_K(\tilde{y})$. For arbitrary initial value x, the numerical solution $\Phi_{h,f_h^K}(x)$ satisfies

$$\phi_{h,f}(x) = \Phi_{h,f_h^K}(x) + h^{K+2} f_{K+1}(x) + \mathcal{O}(h^{K+3}).$$

And thus

$$\phi_{h,f}'(x) = \Phi_{h,f_h^K}'(x) + h^{K+2} f_{K+1}'(x) + \mathcal{O}(h^{K+3}).$$

According to the facts that $\phi_{h,f}$ and Φ_{h,f_h^K} are symplectic maps, and $\Phi_{h,f_h^K}'(x) = I + \mathcal{O}(h)$, we have

$$J = \phi_{h,f}'(x)^T J \phi_{h,f}'(x) = J + h^{K+2}(f_{K+1}'(x)^T J + J f_{K+1}'(x)) + \mathcal{O}(h^{K+3}).$$

Consequently, $f_{K+1}'(x)^T J + J f_{K+1}'(x) = 0$, i.e., $J f_{K+1}'(x)$ is symmetric. According to the Poincaré Lemma (see e.g., Lemma VI.2.7 of (Hairer et al., 2006)), for any x, there exists a neighbourhood and a smooth function H_{K+1} obeying

$$J f_{K+1} = \nabla H_{K+1}$$

on this neighbourhood. Hence the induction holds and the first part of the proof is completed.

If the employed numerical integrator Φ_h is not symplectic, we suppose Φ_h preserves symplectic form of order $\hat{k}$, i.e.,

$$\Phi_h'(x)^T J \Phi_h'(x) = J + \mathcal{O}(h^{\hat{k}+1})$$

when the method is applied to Hamiltonian systems. By repeating the above induction, we can prove that $f_k(y) = J^{-1}\nabla H_k(y)$ for $k = 1, 2, \cdots, \hat{k} - 1$. Subsequently, since $\phi'_{h,f}(x) = \Phi'_{h, f_h^{\hat{k}-1}}(x) + h^{\hat{k}+1} f'_{\hat{k}}(x) + \mathcal{O}(h^{\hat{k}+2})$ and $\phi'_{h,f}(x)^T J \phi'_{h,f}(x) = J$, we have that

$$J = \phi'_{h,f}(x)^T J \phi'_{h,f}(x) = J + \mathcal{O}(h^{\hat{k}+1}) + h^{\hat{k}+1}(f'_{\hat{k}}(x)^T J + J f'_{\hat{k}}(x)) + \mathcal{O}(h^{\hat{k}+2}).$$

Consequently, $f'_{\hat{k}}(x)^T J + J f'_{\hat{k}}(x) \neq 0$, i.e., $J f'_{\hat{k}}(x)$ is not symmetric. This fact yields that $J f_{\hat{k}}$ is not a potential field since the Jacobian of a potential field must be symmetric. The proof is completed. □

C. Experimental Details

Since both true f and the IMDE f_h are inaccessible in practice, we consider several benchmark problems that are widely investigated for the discovery of hidden dynamics (Du et al., 2021; Greydanus et al., 2019; Keller & Du, 2021; Yu et al., 2021). Here, the true system is known and we can calculate the corresponding IMDE. We use solvers with different levels of accuracy to train Neural ODE and we use a Runge-Kutta method of order 4, denoted as RK4, as the test solver. The code accompanying this paper are publicly available at `https://github.com/Aiqing-Zhu/IMDE`.

The benchmark problems are the pendulum system, the damped harmonic oscillator and the nonlinear Lorenz system, which are respectively formulated as

$$\begin{cases} \dfrac{d}{dt}y_1 = -10\sin y_2, \\ \dfrac{d}{dt}y_2 = y_1, \end{cases} \qquad \begin{cases} \dfrac{d}{dt}y_1 = -0.1y_1^3 + 2.0y_2^3, \\ \dfrac{d}{dt}y_2 = -2.0y_1^3 - 0.1y_2^3, \end{cases} \qquad \begin{cases} \dfrac{d}{dt}y_1 = 10(y_2 - y_1), \\ \dfrac{d}{dt}y_2 = y_1(28 - 10y_3) - y_2, \\ \dfrac{d}{dt}y_3 = 10y_1y_2 - \dfrac{8}{3}y_3. \end{cases}$$

The training dataset consists of grouped pairs of points with shared data step T, i.e., $\mathcal{T} = \{(x_n, \phi_T(x_n))\}_{n=1}^N$. On all experiments, the neural networks employed in Neural ODE are all fully connected networks with two hidden layers, each layer having 128 hidden units. The activation function is chosen to be tanh. We optimize the mean-squared-error loss

$$\frac{1}{N}\sum_{n=1}^{N} \|(\Phi_{\frac{T}{S}, f_\theta})^S(x_n) - \phi_T(x_n)\|^2$$

for 3×10^5 epochs with Adam optimization (Kingma & Ba, 2015) where the learning rate is set to decay exponentially with linearly decreasing powers from 10^{-2} to 10^{-5}.

For the first two benchmarks, we take $N = 10000$ and randomly sample x_n from $[-3.8, 3.8] \times [-1.2, 1.2]$ and $[-2.2, 2.2] \times [2.2, 2.2]$, where T is chosen to be 0.04 and 0.02, respectively. For the Lorenz system, the training dataset consists of $N = 251$ data points on a single trajectory starting from $(-0.8, 0.7, 2.6)$ with shared data step of $T = 0.04$, i.e., $x_1, \cdots, x_{N+1}$ where $x_n = \phi_{nT}(x_0)$. These data points are grouped into pairs before training, and denoted as $\mathcal{T} = \{(x_n, x_{n+1})\}_{n=1}^N$. After training, we plot the trajectories of the benchmark problems starting at $(0, 1)$, $(2, 0)$, $(-0.8, 0.7, 2.6)$, respectively. For comparison, the first components of the trajectories are presented in Figure 1.

To investigate errors versus h for the first benchmark problem, we take $N = 10000$ and randomly sample x_n from $[-3.8, 3.8] \times [-1.2, 1.2]$. We take multiple T to generate corresponding training data, and use the Euler method and the explicit midpoint rules to train the model on these data, where the composition numbers are set to be $S = 1, \cdots, 6$. After training, we calculate the mean absolute error between f_θ and f via

$$\frac{1}{2000}\sum_{x} \|f_\theta(x) - f(x)\|_\infty,$$

where x is randomly sampled from $[-3.8, 3.8] \times [-1.2, 1.2]$. The mean error based on 5 independent experiments are recorded in Figure 2. We calculate the order of $Error$ with respect to discrete step h by $\log_2(\frac{Error(2h)}{Error(h)})$.

The first benchmark problem is a Hamiltonian system. To investigate the behavior of learning Hamiltonian system using Neural ODE, we test the model trained for investigating errors. Here, we take $T = 0.12$. As for the ODE solver, we take the Euler method with $S = 6$ and the explicit midpoint rule with $S = 1$. We select the trained models with the above parameters and depict the orbits starting from $(0, 1)$ in Figure 3.