Splitting Physics-Informed Neural Networks for Inferring the Dynamics of Integer- and Fractional-Order Neuron Models

Simin Shekarpaz[1], Fanhai Zeng[2] and George Karniadakis[1,*]

[1] *Division of Applied Mathematics, Brown University, Providence, RI 02912, USA.*
[2] *School of Mathematics, Shandong University, Jinan, Shandong 250100, China.*

Abstract. We introduce a new approach for solving forward systems of differential equations using a combination of splitting methods and physics-informed neural networks (PINNs). The proposed method, splitting PINN, effectively addresses the challenge of applying PINNs to forward dynamical systems and demonstrates improved accuracy through its application to neuron models. Specifically, we apply operator splitting to decompose the original neuron model into sub-problems that are then solved using PINNs. Moreover, we develop an L^1 scheme for discretizing fractional derivatives in fractional neuron models, leading to improved accuracy and efficiency. The results of this study highlight the potential of splitting PINNs in solving both integer- and fractional-order neuron models, as well as other similar systems in computational science and engineering.

AMS subject classifications: 92B20, 34C28, 37M05, 34A08

Key words: Operator splitting, neuron models, fractional calculus.

1 Introduction

The human brain is a complex system that involves the interactions of billions of neurons. Mathematical models can be used to simulate the neuronal activity in the brain as a system of differential equations, allowing researchers to better understand how the brain works. Studies related to spiking neurons are performed numerically or biophysically. In numerical studies, the main goal is to solve neural equations and investigate how the dynamic behavior changes for different inputs. Biophysical approaches focus on interpreting the dynamic behavior of spiking neurons according to available experimental observations [2, 61].

Corresponding author. Email addresses: `simin_shekarpaz@brown.edu` (S. Shekarpaz),
`fanhai_zeng@sdu.edu.cn` (F. Zeng), `george_karniadakis@brown.edu` (G. Karniadakis)

Another interesting aspect of spiking neuron models is that they can be formulated as fractional-order equations, which take into account long-term memory. The order of the derivative in these equations can affect the neuron's response [57, 62, 66], making this an important area of research. Recent works in both integer- and fractional-order neuron models are discussed in Section 3.3.

In this work we introduce a new approach for solving neuron models that combines operator splitting methods with physics-informed neural networks (PINNs). Operator splitting methods have been successfully applied in various fields of physics and engineering [7,12,16,19,25,36,54,55], while PINNs provide a powerful tool for approximating the solution of differential equations. A general introduction to the splitting method can be found in [27,46].

PINNs were first introduced by Raissi et al. [53]. In this method, the solution of a differential equation is approximated using a neural network, and the parameters of the network are determined by solving a minimization problem that includes residual functions at collocation points, as well as initial and boundary conditions.

PINNs have been applied successfully to a broad range of ordinary and partial differential equations, including fractional equations [51], integro-differential equations, stochastic partial differential equations [71], and inverse problems [47]. There have also been several extensions to the original PINN, such as Fractional PINN (FPINN) [51], physics-constrained neural networks (PCNN) [38, 72], variable hp-VPINN [32], conservative PINN (CPINN) [31], Bayesian PINN [69], parallel PINN [58], Self-Adaptive PINN [45], and Physics informed Adversarial training (PIAT) [56]. Some other recent works can be found in [6, 24, 42]. Innovations in activation functions, gradient optimization techniques, neural network structures, and loss function structures have driven recent advances in the field. Despite these advances, improvements are still possible, especially concerning unresolved theoretical and practical issues.

Our study makes two important contributions to the field of neural modeling. First, we propose a new method, called the splitting PINN, that employs the operator splitting technique to decompose the original spiking neuron model into sub-problems, which are then solved using PINNs. We demonstrate the effectiveness and accuracy of this method by applying it to integer- and fractional-order neuron models with oscillatory responses, for which vanilla PINN and FPINN formulations fail to predict the solutions. Second, we introduce a novel L^1-scheme for discretizing fractional derivatives in fractional neuron models, which leads to improved accuracy and efficiency in solving these complex models. Our results show that the combination of the splitting PINN method and the L^1-scheme accurately solves fractional neuron models and provides valuable insights into the underlying mechanisms of neural activity.

This paper is organized as follows: Section 2 provides an overview of the proposed method for a given system of differential equations. Section 3 introduces various neuron models and their properties, including the Leaky Integrate-and-Fire (LIF), Izhikevich, Hodgkin-Huxley (HH), and the fractional order Hodgkin-Huxley (FO-HH) models. The efficiency and accuracy of splitting PINNs and FPINNs are demonstrated in Section 4

by applying our algorithm to different neuron models. Finally, we present a discussion of the main results in Section 5. The results and conclusion of this paper will provide valuable insights into the potential of this new approach for solving neuron models and other similar systems in computational science and engineering.

2 Problem setup and solution methodology

Let us consider the general form of a nonlinear system of differential equations as follows,

$$\frac{dx_i}{dt} = f_i(t, x), \quad i = 1, \cdots, n, \quad t \in [0, T]. \tag{2.1}$$

Before introducing the splitting PINN for solving the given systems of differential equations, we briefly review what splitting methods are, in general.

2.1 Splitting method

For solving the given system (2.1), let us rewrite it as follows,

$$\frac{dx}{dt} = f(t, x(t)), \quad x(0) = x_0 \quad x \in \mathbb{R}^n,$$

then by using splitting method, x is decomposed into (x^*, x^{**}), where $x^*, x^{**} \in \mathbb{R}^d$ $(d < n)$. So, we have

$$\frac{dx^*}{dt} = f(t, x^*(t)), \quad x^*(0) = x_0^*, \tag{2.2}$$

$$\frac{dx^{**}}{dt} = f(t, x^{**}(t)), \quad x^{**}(0) = x_0^{**}, \tag{2.3}$$

and x_0^* and x_0^{**} are the given vectors of initial conditions. After solving the above sub-systems by the given practical algorithms, we denote the solutions of (2.2) and (2.3) as

$$\phi_{\Delta t}^{(*)} x_0^* \quad \text{and} \quad \phi_{\Delta t}^{(**)} x_0^{**},$$

where $\phi_{\Delta t}^{(*)}$ and $\phi_{\Delta t}^{(**)}$ are x^*-flow and x^{**}-flow, respectively, and Δt is the step size. Then, by combining the solutions of sub-systems, the approximation operator in x-flow can be written as follows

$$\phi_{\Delta t}^{(*)} \circ \phi_{\Delta t}^{(**)} \quad \text{or} \quad \phi_{\Delta t/2}^{(**)} \circ \phi_{\Delta t}^{(*)} \circ \phi_{\Delta t/2}^{(**)} \tag{2.4}$$

and $\phi_{\Delta t}^{(*)}$ and $\phi_{\Delta t}^{(**)}$ are interchangeable. The first splitting is the Lie-splitting method [65], which is first-order, and the second is the Strang splitting method [59], which is a second-order method.

On interval $[0,T]$, we first split the original system into sub-systems (sub-problems) for each sub-interval $[t^j, t^{j+1}]$ $(j=0,1,\cdots,J-1,\ t^J=T)$. Then the solution of the original system at time t^{j+1} can be approximated as follows

$$x(t^{j+1}) = \phi_{\Delta t}^{(*)} \circ \phi_{\Delta t}^{(**)} x(t^j),$$

where $x(t^j)$ is the accurate solution at $t = t^j$.

2.2 Physics informed neural network

Consider an initial value problem as follows,

$$\begin{aligned} \frac{dx}{dt} &= f(t,x(t)), \quad x \in \Omega \subseteq \mathbb{R}^n, \quad t \in [0,T], \\ x(0) &= x_0, \end{aligned} \tag{2.5}$$

where f is a nonlinear differential operator, and x is the unknown solution with known initial condition.

By using the PINN framework, the solution of the above equation is approximated by a fully connected neural network $\mathcal{N}^L$ with L layers and N neurons, where the output of l-th hidden layer is defined as follows

$$\mathcal{N}^l(t) = W^l \sigma(\mathcal{N}^{l-1}(t)) + b^l, \quad 2 \leq l \leq L, \tag{2.6}$$

$t \in \mathbb{R}$ is the input vector, $\sigma(\cdot)$ is the activation function, $\sigma(\mathcal{N}^{l-1}(t)) \in \mathbb{R}^N$ $\{W^l \in \mathbb{R}^{N \times N}, b^l \in \mathbb{R}^N\}$ are the network parameters, $\mathcal{N}^1(t) = W^1 t + b^1$ and $\mathcal{N}^L(t)$ is the output of the last layer which is used to approximate the solution. The unknown parameters can be learned by solving a minimization problem that consists of the residual error terms as follows

$$\min_{w,b} \quad \frac{1}{N_r} \sum_{i=1}^{N_r} |\frac{dx}{dt}(t^i) - f(t^i, x(t^i))|^2 + |x(0) - x_0|^2, \tag{2.7}$$

where N_r is the number of collocation points. The parameters are randomly initialized and optimized, and then the approximate solution is obtained. The framework uses automatic differentiation to calculate the derivatives of the solution, which eliminates the need for manual calculation or numerical discretization [5]. This capability is available in popular deep-learning frameworks such as TensorFlow and PyTorch.

3 Neuron models

Until now, various neuron models have been presented for the biological simulations of different parts of the brain. Among them, we can mention the leaky integrate-and-fire (LIF) model, the Izhikevich model, the Hodgkin-Huxley (HH) model, and FitzHugh-Nagumo (FHN) model, which model the membrane behavior [2]. These models can
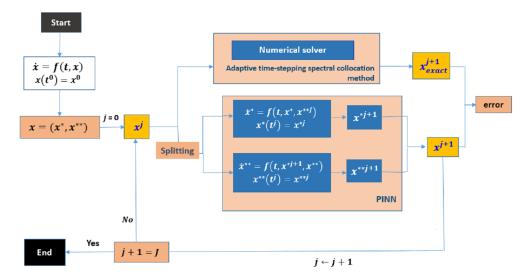
Figure 1: **Overview of the splitting PINN:** We first split the original system into sub-systems (sub-problems) for each sub-interval $[t^j, t^{j+1}]$. For each sub-interval, $x(t^j)$ is known, the sub-systems are solved using PINN and then the solutions are combined to obtain the approximate solution $x(t^{j+1})$. To evaluate the error, we obtain the reference solution, $x_{exact}(t^{j+1})$, by using a high-order numerical solver (for more details, see Appendix B). The algorithm proceeds until arriving at a given accuracy for each sub-interval.

be classified into integer-and fractional-order models. Integer-order models can capture complex phenomena in the neuron system. However, they represent only one type of firing characteristic for constant parameters of the model. On the other hand, fractional-order models can exhibit different dynamic behavior of neurons for constant parameters [61]. This makes fractional-order models more versatile and capable of capturing a wider range of neuron behavior.

3.1 Neuron models: Integer order

3.1.1 IF and LIF models

The neuron model of integrate-and-fire (IF model) is one of the best models due to the simplicity of calculations and closeness to human biological conditions. This model is a simplified version of the HH model, which is described by an equation and an assumption. Unlike the HH model, the IF model does not automatically generate an action potential. This model can be determined by

$$C_m \frac{dV}{dt} = I(t), \tag{3.1}$$

with the following spike condition: if $V = V_{th}$, a spike at t_{spike} is generated and the membrane potential $V(t)$ is set to V_{rest} for a refractory period τ_{ref} [21, 22]. C_m is the membrane capacitance, V_{th} is the voltage threshold and V_{rest} is the resting membrane potential.

A generalized type of IF model is the leaky integrate-and-fire (LIF) model, which adds a leak to the membrane potential. This model is defined by the following equation,

$$\tau \frac{dV}{dt} = -(V - V_{rest}) + RI(t), \tag{3.2}$$

where $\tau = RC_m$ is the membrane time constant and R is the membrane resistance. Because of the important properties of this model, like its computational simplicity [29], accuracy in terms of the spiking behavior and spike times of neurons, and simulating speed [11,41], this model has become one of the most popular and advantageous neuron models in neuromorphic computing [1, 8, 41, 48]. Also, the characteristic of membrane potential decay over time can be seen in the LIF model.

More complex types of IF models include exponential integrate-and-fire, quadratic integrate-and-fire, and adaptive exponential integrate-and-fire [9].

3.1.2 Izhikevich model

Another neuron model for simulating the membrane behavior is the Izhikevich model. Two important features of this model are computational efficiency and biological plausibility. It reduces the more complex Hodgkin-Huxley model to a 2D system of ordinary differential equations of the form

$$\frac{dv}{dt} = 0.04v^2 + 5v + 140 - u + I(t),$$
$$\frac{du}{dt} = a(bv - u), \tag{3.3}$$

with the auxiliary condition

$$\text{if } v \geq v_{th}, \text{ then } \left\{ \begin{array}{l} v \leftarrow c \\ u \leftarrow u + d \end{array} \right\} \tag{3.4}$$

with u and v being dimensionless variables. The variable v is the membrane potential of the neuron, and u is the membrane recovery variable, which accounts for the activation of the K^+ ion current and the inactivation of the Na^+ ion current and provides negative feedback to the membrane potential. The dimensionless parameters a, b, c, and d regulate the behavior of the neuron.

The auxiliary condition (3.4) triggers a reset of the neuron when the membrane potential surpasses the threshold, simulating a spike. This model is capable of reproducing the spiking and bursting behavior of neurons in real-time, making it a widely used model in simulations of large-scale neural networks.

3.1.3 Hodgkin-Huxley model

From a biophysical perspective, the nerve cell's action potential is generated by the flow of ions through the cell membrane's ion channels. Hodgkin and Huxley described the
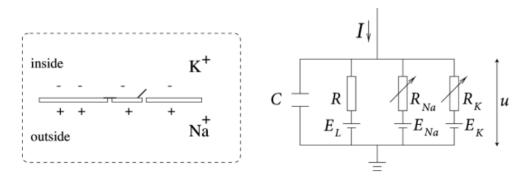
Figure 2: Schematic diagram for the Hodgkin-Huxley model [26]. Left: ion channels on the membrane of the neuron. Right: simulated circuit with I denoting the input current.

dynamics of these membrane currents through a set of coupled differential equations based on their experiments on the giant squid axon [26].

The mechanism of the action potential can be understood with reference to Fig. 2. A capacitor, resistor, and transistor were used to simulate the equivalent circuit. Changes in the action potential were observed by applying the current $I(t)$ and adjusting the capacitance and leakage resistance of the sodium and potassium channels.

The circuit of the Hodgkin-Huxley (HH) model consists of four parallel branches: integrative branch, leaky branch, K^+ channel, and Na^+ channel. A system of four coupled differential equations was used to describe the membrane potential of a giant squid axon as follows

$$\frac{dV_m}{dt} = F_1(t,V_m,n,m,h), \quad \frac{dn}{dt} = F_2(t,V_m,n,m,h),$$
$$\frac{dm}{dt} = F_3(t,V_m,n,m,h), \quad \frac{dh}{dt} = F_4(t,V_m,n,m,h), \tag{3.5}$$

where

$$F_1 = \frac{1}{C_m}(-g_L(V_m - E_L) - g_K n^4 (V_m - E_K) - g_{Na} m^3 h(V_m - E_{Na}) + I(t)),$$
$$F_2 = \alpha_n(V_m(t))(1 - n(t)) - \beta_n(V_m(t))n,$$
$$F_3 = \alpha_m(V_m(t))(1 - m(t)) - \beta_m(V_m(t))m, \tag{3.6}$$
$$F_4 = \alpha_h(V_m(t))(1 - h(t)) - \beta_h(V_m(t))h,$$

and g_{Na}, g_K and g_L are the maximum conductances of the Na^+, K^+ and leak currents. The variables α_x and β_x of the current channel conductances are dependent functions of

$V_m(t)$ as follows:

$$\alpha_n(V_m) = \frac{0.1 - 0.01(V_m - V_0)}{e^{1 - 0.1(V_m - V_0)} - 1}, \quad \beta_n(V_m) = 0.125 e^{-(V_m - V_0)/80},$$

$$\alpha_m(V_m) = \frac{2.5 - 0.1(V_m - V_0)}{e^{2.5 - 0.1(V_m - V_0)} - 1}, \quad \beta_m(V_m) = 4.0 e^{-(V_m - V_0)/80}, \tag{3.7}$$

$$\alpha_h(V_m) = 0.07 e^{-(V_m - V_0)/20}, \quad \beta_h(V_m) = \frac{1}{1 + e^{3 - 0.1(V_m - V_0)}}.$$

3.2 Neuron models: Fractional order

In recent years, fractional differential equations have been developed to improve the modeling of many biological phenomena, including mechanical properties of viscoelastic tissue [43], the tissue-electrode interface [44], pharmacokinetics of drug delivery and absorption [14, 15, 52], and anomalous calcium sub-diffusion in micro-domains [60].

The main characteristic of using fractional derivatives is their non-locality. This means that the next state of the system depends on the current state and all historical states before it. This advantage makes the study of fractional order systems an active area of research.

3.2.1 Fractional derivative definitions

In this Section, we present the definitions of fractional derivatives. There are different methods for defining fractional derivatives, among which we can mention the Grünwald–Letnikov derivative, Riemann–Liouville derivative, and Caputo derivative [35]. The models in this paper are defined using the Caputo fractional derivative.

Definition 3.1. The Caputo fractional derivative of the function $f(t)$ with order $\alpha > 0$ is defined as

$$_cD_{a,t}^{\alpha} f(t) = \frac{1}{\Gamma(n - \alpha)} \int_a^t (t - s)^{n - \alpha - 1} f^{(n)}(s) ds, \tag{3.8}$$

where $n - 1 < \alpha < n$ and n in a non-negative integer.

If $a = 0$, then we can use $\frac{d^\alpha}{dt^\alpha} f(t) = {}_cD_{0,t}^{\alpha} f(t)$.

3.2.2 L^1 scheme to approximate the fractional derivatives

An efficient method for approximating the Caputo derivative of order α $(0 < \alpha < 1)$ is the L^1 scheme, which was introduced by Oldham and Spanier [50]. Using this method, the Caputo derivative can be approximated by using the following formula

$$\frac{d^\alpha}{dt^\alpha} f(t^j) \approx \delta_t^\alpha f^j = \sum_{k=0}^{j-1} b_k^\alpha \left[f(t^{j-k}) - f(t^{j-1-k}) \right], \tag{3.9}$$

where for the uniform time mesh $t^j = j\Delta t$, $j \geq 0$, Δt is the step size. Then, b_k^α is given by $b_k^\alpha = \frac{(\Delta t)^{-\alpha}}{\Gamma(2-\alpha)}[(k+1)^{1-\alpha} - k^{1-\alpha}]$.

In [37], for a smooth f, the error estimate of the above L^1 scheme is

$$\left| \delta_t^\alpha f(t^j) - \frac{d^\alpha}{dt^\alpha} f(t^j) \right| \leq C\Delta t^{2-\alpha}, \tag{3.10}$$

where $C = C(\alpha, f)$. This approximation has been used in many papers discussing fractional-order spiking neurons. The interested readers are referred to [2, 4, 49, 63]. Here, we will develop this method to discretize the fractional derivative, and then FPINN is used.

3.2.3 Fractional Hodgkin-Huxley model

In fractional-order neural models, the neuron's dynamics depends on the order of the derivative, which can create different types of memory-dependent dynamics. The fractional order Hodgkin-Huxley (FO-HH) model is one of the neuron models that has attracted much attention.

The HH model has two basic problems; the first is that the Dielectric losses in the membrane are neglected. The second problem is that membrane capacity is considered ideal. To overcome the above problems, a fractional model is proposed. The idea of fractional capacity is taken from Curie's empirical law [68], which can be written as follows

$$I_c(t) = C_m \frac{d^{q_1} V_m(t)}{dt^{q_1}}, \tag{3.11}$$

where $V_m(t)$ is the excitation voltage, $I_c(t)$ is the current in the capacitor, q_1 is the order of differentiation, and C_m is the fractional capacitance [67]. The fractional order model provides a more accurate description based on long-term memory behavior. Motivated by the above discussion, we propose the following FO-HH model

$$\begin{aligned} \frac{d^{q_1} V_m}{dt^{q_1}} &= F_1(t, V_m, n, m, h), &\quad \frac{d^{q_2} n}{dt^{q_2}} &= F_2(t, V_m, n, m, h), \\ \frac{d^{q_3} m}{dt^{q_3}} &= F_3(t, V_m, n, m, h), &\quad \frac{d^{q_4} h}{dt^{q_4}} &= F_4(t, V_m, n, m, h), \end{aligned} \tag{3.12}$$

where $q = (q_1, q_2, q_3, q_4)$ is the order of differentiation, and the other parameters are as in (3.5).

By defining the fractional model, we apply the dielectric loss in the membrane, and as a result, we will see the change in the refractory period with the same value of the given current in an integer order case. The refractory period is the time when the membrane is hyperpolarized and, hence, requires a stronger stimulus to produce a smaller action potential.

The modified HH model can be very effective in biomedical applications such as heart health analysis. For example, in an ECG waveform, the PR interval represents a refractory period. PR interval estimation is very important for cardiac diagnosis [13, 23].

3.3   Prior works in neuron models

In [39], the authors compared the spiking rate patterns of five single neuron models, including LIF, Izhikevich, and Hodgkin-Huxley (HH) models, under different sustained current inputs. Numerical stability and accuracy were also considered. The multi-step methods for neuronal modeling, including the HH model, were proposed in [34]. In [33], the modified Khater (mK) method and B-spline scheme were proposed to find numerical solutions of the FitzHugh-Nagumo (FHN) equation, with a focus on finding different types of soliton wave solutions, studying their stability properties, and using them to obtain numerical solutions of the model.

In [20], the Hybrid Functions (HF) method was proposed as a solution for the HH model. The HF method was compared comprehensively with other algorithms, evaluating computational speed, absolute error, and integral time squared error. The finite difference scheme was used in [70] to solve the stochastic FHN model, including stability analysis and the calculation of explicit optimal a priori estimates for the existence of solutions. In [3], numerical solutions for the FHN and Izhikevich neuron models were obtained using a non-standard finite difference scheme and GL discretization technique. The models were compared, and their behavior was analyzed in different fractional orders.

Fractional order modeling in neural systems is a relatively new area of research. The non-local definitions of fractional calculus used in these models provide a more realistic representation of neural systems and offer a deeper understanding of their behavior. In a study reported in [2], four numerical methods were applied to two fractional-order spiking neuron models, the FO-LIF and FO-HH models. The authors used a finite memory window version of the L^1 approximation for comparison with well-known techniques such as the GL-based method, product integration approximation, and the Z-transform approach. In the four methods, the uniform mesh is used, and low-accurate solutions are obtained due to the singularity of the solution of fractional equations. In this paper, they have analyzed the spiking patterns, inter-spike interval adaptation, and steady-state spiking frequency for each numerical method under varying memory lengths. In a related study reported in [61], the authors have used a L^1 scheme (linear interpolation based on uniform mesh) to discretize the Caputo fractional derivative. The first-order extrapolation is used to derive the linearized scheme, where the global error is $\mathcal{O}(\Delta t^\alpha)$, and the error far from the origin is $\mathcal{O}(\Delta t)$. In this paper, they have investigated the effects of non-Markovian power-law voltage-dependent conductances on the generation of action potentials and spiking patterns in a Hodgkin-Huxley model. They used fractional derivatives to implement the slow-adapting power-law dynamics of the potassium and sodium conductance gating variables. The results showed that, with different input currents and derivative orders, a wide range of spiking patterns can be generated, such as square wave bursting, mixed mode oscillations, and pseudo-plateau potentials. These findings suggest that power-law conductances increase the number of spiking patterns a neuron can produce.

In [10], the dynamics and numerical simulations of a fractional-order coupled FHN neuronal model were discussed, and the stability properties of its equilibrium states were analyzed based on theoretical results. In [64], a non-standard finite difference scheme was used to solve the fractional Izhikevich neuron model, and a general formula for the synchronization of different Izhikevich neurons was proposed.

4 Results

In this Section, we use splitting PINN to solve the neuron models presented in Section 3, using the network architecture and hyperparameters specified in Table 1. The optimization algorithm used is Adam with a learning rate of 0.0001 and a scheduler. IN addition techniques such as adaptive activation function and feature expansion proposed in [30, 40] were also used to improve the computational efficiency of the proposed method.

The relative L^2 norm of errors of the inferred solutions are also computed as follows,

$$\text{relative L}^2 \text{ error} = \frac{\sqrt{\sum_{j=1}^{J}(V_{exact}(t^j) - V_{app}(t^j))^2}}{\sqrt{\sum_{j=1}^{J}(V_{exact}(t^j))^2}}.$$

Further details about the training procedure and hyperparameters can be found in the relevant section.

Table 1: PINN architectures and hyperparameters used for training. The first and second subcolumns under "Depth", "Width", and "Activation" correspond to the first and second sub-problems, respectively.

Problems	Dep.	Wid.	Act.	Opt.	Iter.
LIF model	5	40	tanh	*Adam*	50000
LIF model with	7	60	tanh	*Adam*	10000
threshold voltage					
Izhikevich model	6,6	40,40	tanh,tanh	*Adam, Adamax*	20000
HH model	6,10	20,20	tanh,sin	*Adam, Adamax*	20000
(step current function)					
HH model	6,10	20,20	tanh,sin	*Adam*	20000
(constant current)					
FO-HH model	10,6	100,100	tanh,sin	*Adam*	70000
$(q_i = 0.8)$					
FO-HH model	10,6	100,100	tanh,sin	*Adam*	50000
$(q_i = 0.6)$					
FO-HH model	10,6	100,100	tanh,sin	*Adam*	20000
$(q_i = 0.4)$					

Data validation: In all of the examples, the reference solutions are obtained by using the adaptive time-stepping spectral collocation method, which is presented in Appendix B.

4.1   LIF model

PINN implementation: The approximate solutions of the LIF model can be obtained by using PINNs. For solving this model, we discretize the time domain $[0,T]$ into subdomains $[t^j,t^{j+1}]$ where $j=0,1,\cdots,J-1$ and $t^J=T$ and then PINNs are used to solve the model in each sub-interval. More detailed visual evaluations and the numerical results of this model are provided in Appendix A.

4.2   Izhikevich model

Splitting PINN implementation: Consider (u^j,v^j) to be the numerical solution at $t=t^j=j\Delta t$ with step size, Δt, then by using the splitting method, the following sub-problems on each sub-interval $[t^j,t^{j+1}]$ $(j=0,1,\cdots,J-1,t^J=T)$ are obtained

$$\frac{du}{dt}=a(bv^j-u), \quad u(t^j)=u^j, \tag{4.1}$$

and

$$\frac{dv}{dt}=0.04v^2+5v+140-u^{j+1}+I(t), \quad v(t^j)=v^j. \tag{4.2}$$

This model is solved in the time interval $[0,100\ ms]$, with 2000 sub-intervals and 20 points per sub-interval. For solving sub-problems by using PINN, the fully connected neural networks are assumed to approximate the solutions, with the given architecture described in Table 1, $v_{th}=30mV$,

$$I(t)=\begin{cases} 0 & \text{for } 0<t<2, \\ 15 & \text{for } t\geq 2, \end{cases}$$

and the choice of parameters given in [28], which can be described as follows:

- **a** $\sim 0.02\ \frac{1}{ms}$: Time scale of the recovery variable u. Smaller values result in slower recovery.

- **b** $\sim 0.2\ [dimensionless]$: Sensitivity of the recovery variable u to the sub-threshold fluctuations of the membrane potential v. Greater values couple v and u more strongly resulting in possible sub-threshold oscillations and low-threshold spiking dynamics. The case $b<a$ $(b>a)$ corresponds to saddle-node (Andronov–Hopf) bifurcation of the resting state.

- **c** $\sim -50\ mV$: The after-spike reset value of the membrane potential v caused by the fast high-threshold K^+ conductances.
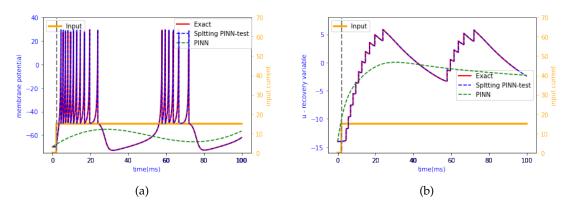
Figure 3: Izhikevich model: Left: membrane potential versus time. Right: recovery variable versus time.

Table 2: Mean relative L^2 norm of errors for the Izhikevich model.

	training error	testing error
membrane potential	0.1006±0.0079	0.1095±0.0046
u	0.0358±0.0030	0.0415±0.0022

Table 3: Relative L^2 norm of errors for the Izhikevich model with different numbers of sub-intervals in $[0,T]$ with $T=100\ ms$. The first and second sub-rows correspond to the training and testing errors, respectively.

	500	1000	2000
membrane potential	0.2576	0.1967	0.0945
	0.2533	0.1944	0.1077
u	0.1442	0.0817	0.0329
	0.1440	0.0805	0.0376

- **d** $\sim 2\ mV$: After-spike reset of the recovery variable u caused by slow high-threshold Na^+ and K^+ conductances.

The relative L^2 norm of errors of approximate solutions have been shown in Table 2. To show the convergency of the method with increasing the numbers of subintervals, we have presented the mean relative L^2 norms of errors for the Izhikevich model in Table 3.

4.3   Hodgkin-Huxley model

Splitting PINN implementation: For solving the HH model by applying Splitting PINN on interval $[0,T]$, we have used the choice of parameters given in Table 4. Consider (V^j, n^j, m^j, h^j) to be the solution at $t^j = j\Delta t$ with step size Δt. Then, the following subsystems on each sub-interval $[t^j, t^{j+1}]$ are obtained using the Lie splitting method. The
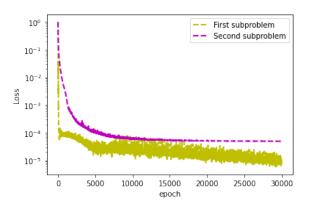
Figure 4: Loss function of the Izhikevich model.

Table 4: Parameter values and their descriptions of the HH neuron model [61].

Parameter	Value	Description
g_{Na}	$120\ mS/cm^2$	*Maximum Na^+ current conductance*
g_K	$36\ mS/cm^2$	*Maximum K^+ current conductance*
g_L	$0.3\ mS/cm^2$	*Maximum leak current conductance*
E_{Na}	$50\ mV$	*Na^+ current reversal potential*
E_K	$-77\ mV$	*K^+ current reversal potential*
E_L	$-54\ mV$	*Leak current reversal potential*
C_m	$1\ \mu F \cdot s^{q_1-1}/cm^2$	*Membrane Capacitance*
V_0	-65	*Initial membrane potential*
m_0	0.0529	*Initial Na^+ current activation*
n_0	0.3177	*Initial Na^+ current inactivation*
h_0	0.5960	*Initial K^+ current activation*

first part is,

$$\begin{aligned}
\frac{dn}{dt} &= F_2(t, V_m^j, n, m, h), & n(t^j) &= n^j, & t &\in (t^j, t^{j+1}], \\
\frac{dm}{dt} &= F_3(t, V_m^j, n, m, h), & m(t^j) &= m^j, & t &\in (t^j, t^{j+1}], \\
\frac{dh}{dt} &= F_4(t, V_m^j, n, m, h), & h(t^j) &= h^j & t &\in (t^j, t^{j+1}].
\end{aligned} \tag{4.3}$$

where V_m^j is the known solution at time $t = t^j$. The second part is given by

$$\frac{dV_m}{dt} = F_1(t, V_m, n^{j+1}, m^{j+1}, h^{j+1}), \quad V_m(t^j) = V_m^j, \quad t \in (t^j, t^{j+1}], \tag{4.4}$$
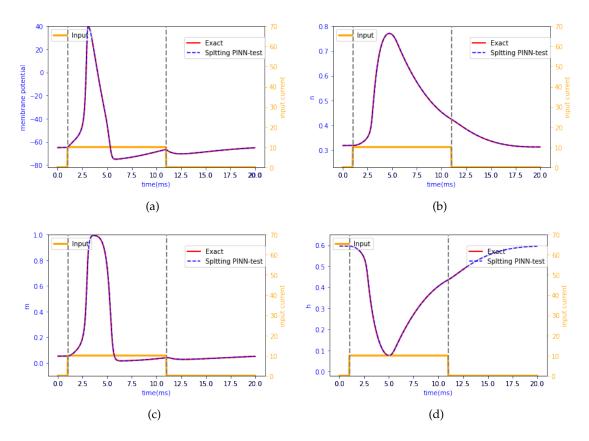
Figure 5: HH model: Comparison of splitting PINN results with the reference solution. (a) membrane potential; (b) activation variable of potassium channel; (c) activation variable of the sodium channel; (d) deactivation variable of the sodium channel. The inset plot shows the step function input current.

and $(n^{j+1}, m^{j+1}, h^{j+1})$ is the solution obtained from the first part. The splitting approximation of the solution at $t = t^{j+1}$ is $(V_m^{j+1}, n^{j+1}, m^{j+1}, h^{j+1})$, and the solution for the whole domain is obtained by repeating this process.

Different kinds of input currents, including step current function and constant current, are considered for solving the HH model.

Step current function: For solving the HH model with the given step current at the time interval $[0, 20ms]$, we have used 800 sub-intervals in the splitting procedure with 40 training points for solving each sub-problem. The hyperparameters and architecture of PINNs for solving the sub-problems can be found in Table 1.

The relative L^2 norm of errors of approximate solutions have been presented in Table 5.

Constant current: This system is solved at time interval $[0, 100ms]$, with 3000 sub-intervals and 30 points at each sub-interval. With the given hyperparameters and architectures in

Table 5: Mean relative L^2 norm of errors for HH model with current step function.

	training error	testing error
membrane potential	0.0012 ± 0.0001	0.0009 ± 0.0001
n	$0.0027\pm5.3\times10^{-5}$	$0.0027\pm5.3\times10^{-5}$
m	0.0143 ± 0.0004	0.0140 ± 0.0005
h	$0.0027\pm7.3\times10^{-5}$	$0.0027\pm7.2\times10^{-5}$,

(a)

(b)

(c)

(d)

Figure 6: HH model: Absolute error of the normalized solutions using splitting PINN with a current step function. (a) membrane potential; (b) activation variable of potassium channel; (c) activation variable of the sodium channel; (d) deactivation variable of the sodium channel.

Table 1, PINN is used for solving each part, and the solutions are combined to obtain the approximate solutions.

The relative L^2 norm of errors of approximate solutions are shown in Table 6. Moreover, the mean relative L^2 norms of errors for different numbers of subintervals have been presented in Table 7.
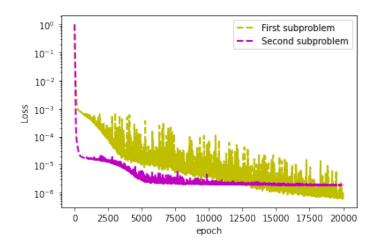
Figure 7: Loss function of the HH model with a current step function.

Table 6: Mean relative L^2 norm of errors for HH model for constant current.

	training error	testing error
membrane potential	0.0099±0.0020	0.0094±0.0019
n	0.0052±0.0007	0.0052±0.0007
m	0.0307±0.0039	0.0303±0.0039
h	0.0071±0.0009	0.0071±0.0009

Table 7: Relative L^2 norm of errors for the HH model with different numbers of sub-intervals in $[0,T]$ with $T=100\ ms$. The first and second sub-rows correspond to the training and testing errors, respectively.

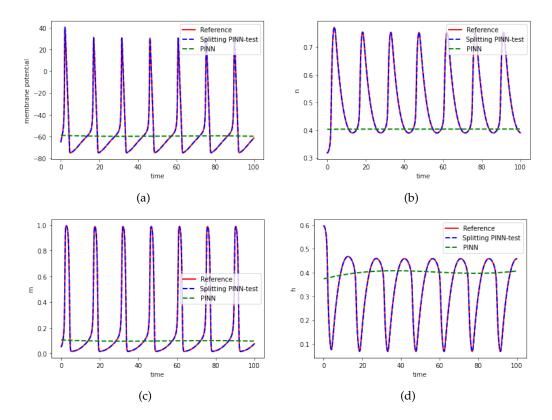|	500	1500	3000	5000
membrane potential	0.3678	0.0585	0.0106	0.0033
	0.3581	0.0562	0.0102	0.0034
n	0.1553	0.0217	0.0053	0.0013
	0.1547	0.0217	0.0053	0.0013
m	0.7677	0.1269	0.0312	0.0074
	0.7597	0.1255	0.0309	0.0073
h	0.2203	0.0293	0.0072	0.0017
	0.2204	0.0293	0.0072	0.0017

Figure 8: HH model: Comparison of splitting PINN results with the reference solution for constant current input ($I = 10 \ nA/cm^2$). (a) membrane potential; (b) activation variable of potassium channel; (c) activation variable of the sodium channel; (d) deactivation variable of the sodium channel. Note that PINN fails to predict the solution.

4.4 Fractional Hodgkin-Huxley model

Splitting FPINN implementation: This problem is solved using the splitting fractional physics-informed neural network (FPINN) approach [51]. In this approach, the memory of the fractional derivative at each sub-interval encompasses the entire domain. So for each sub-interval $[t^j, t^{j+1}]$ $(0 \leq j \leq N_1 - 1)$, by using the developed L^1 - scheme, we have the following sub-problems:

$$\delta_t^{q_1} V_m^{N_2 j + l + 1} = F_1 \big(t^{N_2 j + l + 1}, V_m^{N_2 j + l + 1}, n, m, h \big), \qquad (4.5)$$

where

$$\delta_t^{q_1} V_m^{N_2 j + l + 1} = \sum_{k=0}^{N_2 j + l} b_k^{q_1} \Big[V_m \big(t^{N_2 j + l + 1 - k} \big) - V_m \big(t^{N_2 j + l - k} \big) \Big], \qquad (4.6)$$
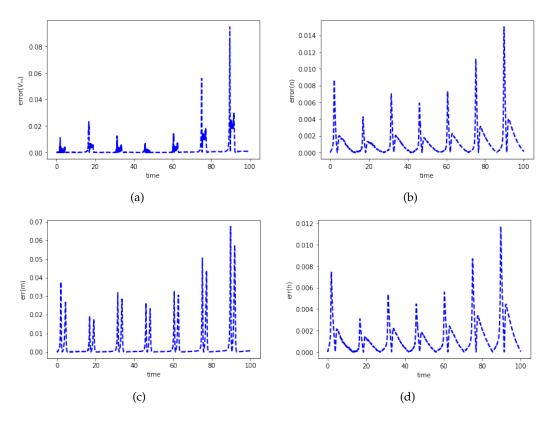
(a)   (b)

(c)   (d)

Figure 9: HH model: Absolute error of the normalized solutions using splitting PINN with a constant current ($I=10 \ nA/cm^2$). (a) membrane potential; (b) activation variable of potassium channel; (c) activation variable of the sodium channel; (d) deactivation variable of the sodium channel.
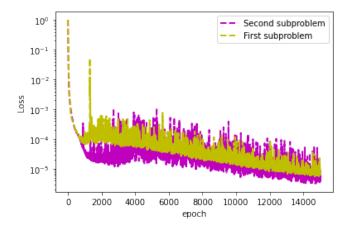

Figure 10: Loss function of the HH model for constant current.

Table 8: Mean relative L^2 norm of errors for FO-HH model for different orders of fractional derivatives. The first and second sub-rows correspond to the training and testing errors, respectively.

	$q_i = 0.8$	$q_i = 0.6$	$q_i = 0.4$
membrane	0.0843 ± 0.0111	0.0981 ± 0.0277	0.0432 ± 0.0015
potential	0.0845 ± 0.0126	0.0969 ± 0.0281	0.0432 ± 0.0038
n	0.0244 ± 0.0055	0.0239 ± 0.0084	$0.0019 \pm 8.8 \times 10^{-5}$
	0.0244 ± 0.0055	0.0238 ± 0.0084	0.0019 ± 0.0002
m	0.1297 ± 0.0350	0.0939 ± 0.0318	$0.0060 \pm 9.4 \times 10^{-5}$
	0.1291 ± 0.0359	0.0936 ± 0.0324	0.0061 ± 0.0002
h	0.0363 ± 0.0090	0.0409 ± 0.0140	0.0075 ± 0.0004
	0.0363 ± 0.0091	0.0409 ± 0.0140	0.0076 ± 0.0006

and n, m and h are the known solutions at $t = t^{N_2 j + l}$. The second subproblem is

$$\delta_t^{q_2} n^{N_2 j + l + 1} = F_2(t^{N_2 j + l + 1}, V_m, n^{N_2 j + l + 1}, m, h),$$
$$\delta_t^{q_3} m^{N_2 j + l + 1} = F_3(t^{N_2 j + l + 1}, V_m, n, m^{N_2 j + l + 1}, h), \qquad (4.7)$$
$$\delta_t^{q_4} h^{N_2 j + l + 1} = F_4(t^{N_2 j + l + 1}, V_m, n, m, h^{N_2 j + l + 1}),$$

where V_m is the solution of the first sub-problem at $t = t^{N_2 j + l + 1}$, $l = 0, 1, \cdots, N_2 - 1$, and N_2 is the number of residual points for each sub-interval.

Then, the solutions of sub-systems are combined to obtain the solution of the original system.

This model is solved with the proposed method for $q_i = 0.8, 0.6, 0.4$ at time interval $[0, 100ms]$, with 2000 sub-intervals and 40 residual points in each sub-interval. Each part is solved via FPINN, with the given hyperparameters and network architectures in Table 1. The relative L^2 norm of errors of approximate solutions have been shown in Table 8. To show the convergency of the method with increasing the numbers of subintervals, we have presented the mean relative L^2 norms of errors for different numbers of subintervals for $q_i = 0.4$ in Table 9.

5   Discussion

We have presented a deep learning approach for solving nonlinear systems of differential equations. The performance and accuracy of splitting Physics-Informed Neural Networks (PINNs) are studied in the context of solving neuron models. In Section 4.2, the proposed method was used to solve the Izhikevich model. We showed the reference and learned solutions in Fig. 3 for equi-distant training and random test datasets. It can be seen that the nonlinear solutions learned by the nonlinear splitting PINN match well with the reference solutions. The relative L^2 norm of errors for the solutions are presented
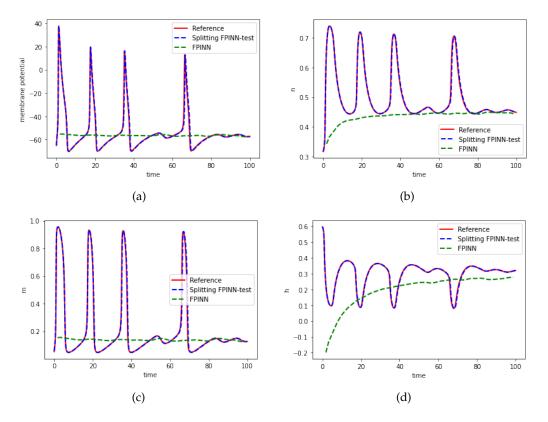
Figure 11: FO-HH model: Comparison of splitting FPINN results with the reference solution for constant current input ($I=20\,nA/cm^2$) and fractional order $q_1=q_2=q_3=q_4=0.8$. (a) membrane potential; (b) activation variable of potassium channel; (c) activation variable of the sodium channel; (d) deactivation variable of the sodium channel. Note that FPINN fails to predict the solution.
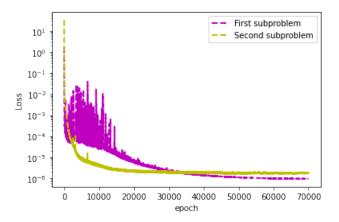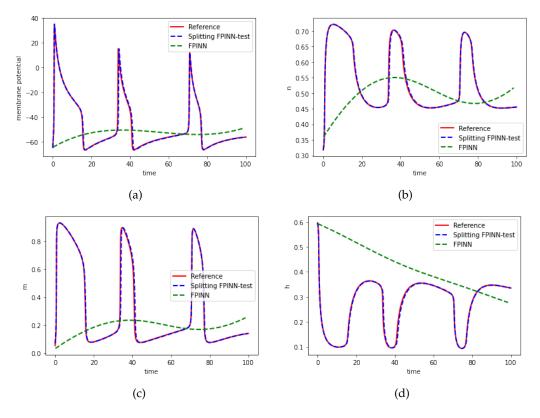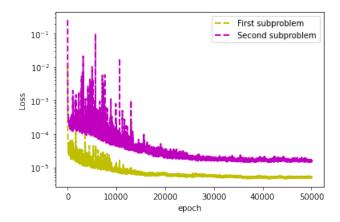

Figure 12: Loss function of the FO-HH model for fractional order $q_1=q_2=q_3=q_4=0.8$.

Figure 13: FO-HH model: Comparison of splitting FPINN results with the reference solution for constant current input ($I=20\,nA/cm^2$) and fractional order $q_1=q_2=q_3=q_4=0.6$. (a) membrane potential; (b) activation variable of potassium channel; (c) activation variable of the sodium channel; (d) deactivation variable of the sodium channel. Note that FPINN fails to predict the solution.

Figure 14: Loss function of the FO-HH model for fractional order $q_1=q_2=q_3=q_4=0.6$.
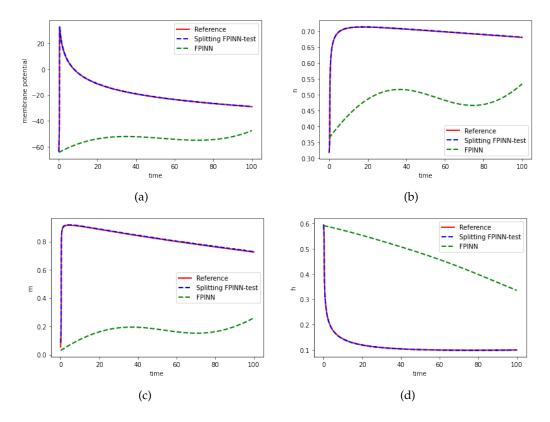
Figure 15: FO-HH model: Comparison of splitting FPINN results with the reference solution for constant current input ($I=20\,nA/cm^2$) and fractional order $q_1=q_2=q_3=q_4=0.4$. (a) membrane potential; (b) activation variable of potassium channel; (c) activation variable of the sodium channel; (d) deactivation variable of the sodium channel. Note that FPINN fails to predict the solution.
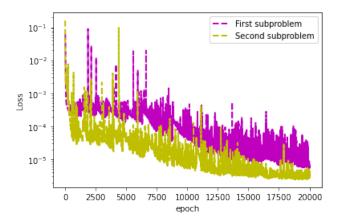

Figure 16: Loss function of the FO-HH model for fractional order $q_1=q_2=q_3=q_4=0.4$.

Table 9: Relative L^2 norm of errors for the FHH model ($q_i = 0.4$) with different numbers of sub-intervals in $[0, T]$ with $T = 100 \ ms$. The first and second sub-rows correspond to the training and testing errors, respectively.

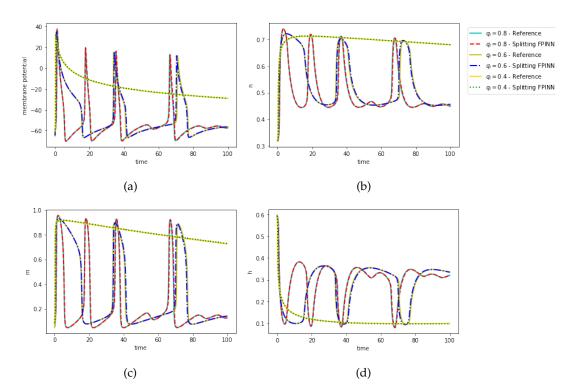|	500	1000	2000
membrane potential	0.2308	0.1462	0.0413
	0.2241	0.1492	0.0393
n	0.0185	0.0099	0.0018
	0.0180	0.0100	0.0017
m	0.0434	0.0249	0.0059
	0.0418	0.0253	0.0058
h	0.0773	0.0378	0.0081
	0.0745	0.0374	0.0080

Figure 17: Plots of the membrane potential and the corresponding activation and deactivation variables of the FO-HH model with constant current $I = 20 \ nA/cm^2$ for different orders of fractional derivatives, obtained using the splitting FPINN.

in Table 2, which shows the method's capability for this model. The convergence of the loss functions is shown in Fig. 4. It can be seen that this function rapidly tends to values lower than 10^{-5} and 10^{-4} for the first and second sub-problems.

After solving the Izhikevich model, we focus on using the Splitting PINN method for the Hodgkin-Huxley model. In this model, applying input current and changes in voltage resulting from the opening and closing of ion channels lead to the generation of spikes. The amount of input current needed to generate a spike is at least $2.7nA/cm^2$, and with increasing input current, the number of spikes increases. In Section 4.3, the voltage is obtained for two states of input current: step function current and constant input current. In the first case, when the current is applied over a given time interval, the voltage increases and produces an action potential (positive peak). After the spike, the potassium channel opens, and the sodium channel closes, causing the voltage to decrease and the neuron to enter a refractory period where the potential is below the resting potential of $V_{rest} = -65mV$. The voltage then slowly returns to $-65mV$. The splitting PINN method was used to calculate approximate solutions and compare them with reference solutions. The results are shown in Fig. 5, and the absolute errors are given in Fig. 6, which demonstrate the accuracy of the method. The relative L^2 norm of errors for the current step function is presented in Table 5. The convergence of the loss functions is also shown in Fig. 7 and tends to smaller values of 10^{-6} and 10^{-5} for the first and second sub-problems, respectively.

In the case of a constant input current, the approximate and reference solutions are shown in Fig. 8. The proposed method's effectiveness is demonstrated by comparing the approximate results with the reference and numerical solutions obtained using PINN. The absolute errors of the solutions are shown in Fig. 9, and the plots of the loss functions are displayed in Fig. 10, which converge to values of 10^{-6} for sub-problems. The relative L^2 norm of errors of the solutions is presented in Table 6. These results demonstrate the capability and accuracy of the proposed algorithm in solving this model.

The fractional-order Hodgkin-Huxley model was also studied in Section 4.4. The system was modeled using Caputo's fractional derivative and simulated using the proposed method. The effect of memory, associated with fractional order, on firing activity was also examined. To address this, the L^1 scheme was improved for solving the fractional-order model by proposing using full-domain memory, rather than sub-domain memory, to calculate the fractional derivative. The numerical results obtained using the improved method for $q_i = 0.8, 0.6, 0.4$ are shown in Figs. 11, 13 and 15, where the approximate results are compared with the reference solutions. The plots of the loss functions are also shown in Figs. 12, 14, and 16, demonstrating the efficiency and convergence of the current method. The relative L^2 norm of errors of the solutions is presented in Table 8, and as shown in the Section, the results are accurate and have converged to the reference solutions.

The voltage responses for the different orders of fractional derivatives under constant input current $I = 20$ are also displayed in Fig. 17. The figure demonstrates that various spike patterns can be produced. As q_i approaches 1, the firing frequency decreases and results in an increase in the number of spikes in the same time period. Furthermore, the first spike occurs at a later time. Following the cessation of the injected current, the memory-dependent spiking activity can also be observed by applying a step function

current. Additionally, the regularity of solutions is observed to depend on the order of fractional derivatives in Fig. 17. As q_i decreases, the regularity increases. Table 8 shows that, unregular solutions are more sensitive to parameter initialization, leading to a greater standard deviation. To address this issue, the network architecture can be improved.

The results in this section demonstrate high accuracy, and the solutions have converged to the reference values.

6 Conclusion

In this study, we introduced a novel method for solving neuron models represented as systems of differential equations. The Splitting PINN algorithm was demonstrated to be effective and accurate through comparisons with reference solutions and the mean L^2 relative norm of errors. In addition, the results of the fractional-order Hodgkin-Huxley model highlighted the effect of memory on firing activity and voltage responses, showing that as q_i approaches 1, the firing frequency decreases while the number of spikes in the same time period increases.

This research provides valuable insights into the behavior of neuron membranes and the various spike patterns that can be generated. The performance of the proposed method demonstrates its superiority over the vanilla PINN algorithm in solving complex neuron models represented as systems of differential equations. This study contributes to the development of tools for investigating the behavior of neurons and the underlying mechanisms of neural activity.
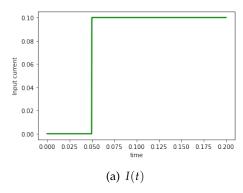
In conclusion, the proposed splitting PINN algorithm is a promising method for solving systems of differential equations and provides valuable insights into the behavior of neurons and their underlying mechanisms.

Acknowledgments

A PINN implementation for LIF model

In this Section, PINN has been used to solve the LIF model with the given parameters in Table 10. For different input currents in time interval $[0, 0.5s]$, the network is trained with
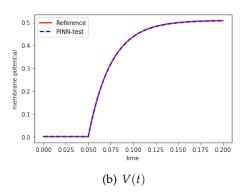
(a) $I(t)$ (b) $V(t)$

Figure 18: LIF model: Left: Input current versus time. Right: Membrane potential versus time (comparison of PINN results with the reference solutions).

Table 10: Parameter values and their descriptions of LIF model [17, 18].

Parameter	Value	Description
C_m	$5 \times 10^{-3} F$	*Membrane Capacity*
R	$5.1\ \Omega$	*Membrane Resistance*
τ	$RC_m\ s$	*Membrane time constant*
V_{rest}	$0\ V$	*Resting membrane potential*
V_{th}	$1\ V$	*Threshold membrane potential*

1000 residual points. Initially, assuming that there is no input current, then if the current $I(t) = 0.1A$ is injected into the neuron, the membrane potential tends to $R \cdot I(t)$, which can be seen in Fig. 18.

The membrane potentials for different forms of step current function have been displayed in Figs. 19-21. By applying an input current in a time step, the membrane potential increases, and then after cutting off the input current, the potential decreases with the time constant τ.

If the width of the injected current increases, the membrane potential increases at a slower rate. As the input current pulse amplitude decreases, the membrane potential jumps directly up in a short time. The membrane potential decreases over time in the absence of an input current.

So far, we have seen the neuron's behavior against different input currents (spikes). We need to apply a voltage threshold condition to the model to see the process of generating output spikes by a neuron. According to this condition, if the membrane potential exceeds this threshold, the neuron produces a spike, and after firing the spike, the membrane potential returns to its initial state (resting state).

If the injected current increases, the membrane potential approaches the threshold faster, and the firing rate increases. Similar behavior in firing frequency can be produced by lowering the threshold. These experiments hold for the constant current application,

(a) $I_1(t)$

(b) $I_2(t)$

(c) $I_3(t)$

(d) $V_1(t)$

(e) $V_2(t)$

(f) $V_3(t)$

(g) absolute error of $V_1(t)$

(h) absolute error of $V_2(t)$
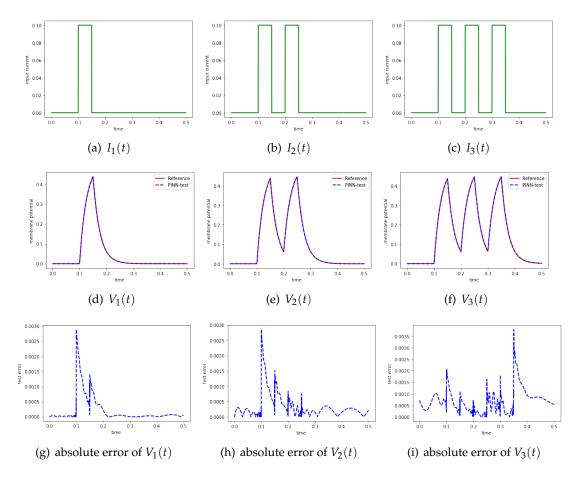
(i) absolute error of $V_3(t)$

Figure 19: LIF model; (a)-(c): Input step function currents; (d)-(f): Comparison of the PINN results with the reference solutions; and (g)-(i): The error results of PINN.

but neurons may be more likely to receive spikes in other cases.

The approximate solutions of this model, when the threshold voltage is applied, have been shown in Fig. 22. In this case, for solving the problem in $[0, 0.2s]$, we have used 1000 sub-intervals and 20 points in each sub-interval. Furthermore, for solving the problem in each sub-interval, PINN has been used with the given architecture in Table 1.

B   Reference solutions

We present a highly efficient and accurate spectral collocation method for obtaining the reference solutions of the fractional differential equations studied in the paper. Consider the following system of FODEs

$$_C D_{0,t}^{\alpha_k} U_k(t) = f_k(t, U), \quad 0 < t \le T, \quad k = 1, 2, \cdots, d, \quad d \in \mathbb{N}, \tag{B.1}$$

(a) $I_1(t)$ (b) $I_2(t)$ (c) $I_3(t)$

(d) $V_1(t)$ (e) $V_2(t)$ (f) $V_3(t)$

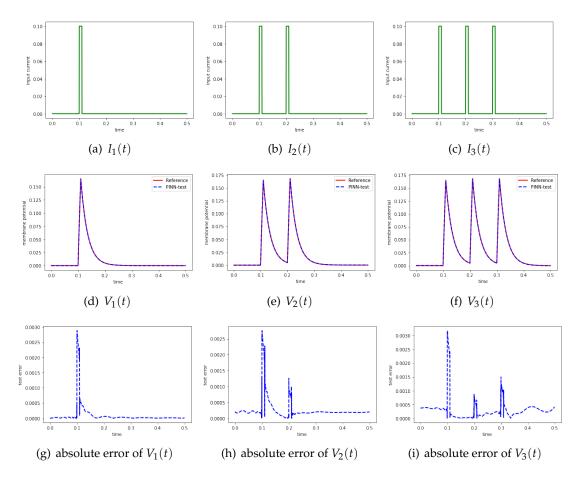(g) absolute error of $V_1(t)$ (h) absolute error of $V_2(t)$ (i) absolute error of $V_3(t)$

Figure 20: LIF model; (a)-(c): Input spikes; (d)-(f): Comparison of the PINN results with the reference solutions; and (g)-(i): The error results of PINN.

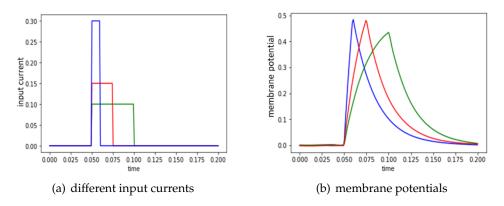(a) different input currents   (b) membrane potentials

Figure 21: LIF model: (a) Input currents with different lengths and heights; (b) Corresponding membrane potentials by using PINN.
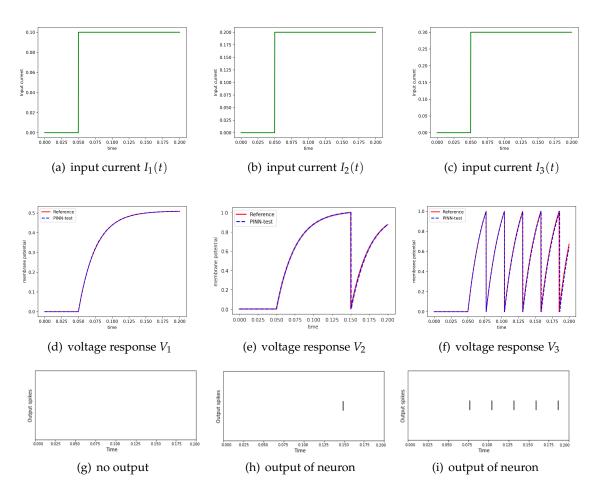
Figure 22: LIF model; (a)-(c): Input currents, (d)-(f): Comparison of the PINN results with the reference solutions, and (g)-(i): Output spikes.

subject to the initial condition $U_k(0) = \phi_k, f_k(t,U), U = (U_1, U_2, \cdots, U_d)$.

We apply the fast time-stepping spectral collocation method to solve (B.1). Divide the interval into subintervals $[t^{k-1}, t^k]$ such that $[0,T] = [t^0, t^1] \cup [t^1, t^2] \cup \cdot \cup [t^{N-1}, t^N]$ and $t^N = T$, N is a positive number. Let p be a positive number and $\{x_j\}_{j=0}^p$ be the Legendre–Gauss–Lobatto (LGL) quadrature points on the standard interval $[-1, 1]$. Then, the LGL quadrature points $\{t_j^k\}_{j=0}^p$ in the interval $[t^{k-1}, t^k]$ can be written as

$$t_j^k = \frac{1}{2}\left[(t^k - t^{k-1})x_j + t^{k-1} + t^k\right].$$

Denote Π^p the interpolation operator as

$$\Pi^p u(t)\big|_{t \in [t^{k-1}, t^k]} = \sum_{j=0}^p u(t_j^k)\ell_j^k(t), \quad \ell_j^k(t) = \prod_{q=0, q \neq j}^p \frac{t - t_q^k}{t_j^k - t_q^k}.$$

Next, we show the fast calculation of $_{C}D_{0,t}^{\alpha}u(t)$, the algorithm is illustrated by the following **Step 1)**–**Step 4)**.

- **Step 1)** Find a sum-of-exponentials to approximate the kernel function $k_{-\alpha}(t)$ as

$$k_{-\alpha}(t) = \frac{t^{-\alpha-1}}{\Gamma(-\alpha)} = \sum_{j=1}^{Q} w_j e^{-\lambda_j t} + \mathcal{O}(\varepsilon t^{-\alpha-1}), \quad t \in [t_1, T], \tag{B.2}$$

where ε is the relative error, $Q > 1$ is a positive integer, and

$$\lambda_j = e^{y_j}, \quad w_j = -\frac{\sin(\alpha\pi)}{\pi}\Delta y e^{(1+\alpha)y_j}.$$

Here, $y_j = y_{\min} + j\Delta y$, $\Delta y = (y_{\max} - y_{\min})/(Q-1)$, and

$$y_{\min} = (1+\alpha)^{-1}\ln(\varepsilon_0) - \ln(T), \quad \varepsilon_0 = 10^{-16},$$
$$y_{\max} = \log\left(\frac{-\ln(\varepsilon_0) + (1+\alpha)\ln(t_1)}{0.5 t_1}\right), \quad \varepsilon_0 = 10^{-16}.$$

- **Step 2)** For $t > t^{n-1}$, divide

$$_{C}D_{0,t}^{\alpha}u(t) = P.V. \int_0^t k_{-\alpha}(t-s)(u(s)-u(0))\mathrm{d}s$$

into two parts as

$$_{C}D_{0,t}^{\alpha}u(t) = \underbrace{P.V. \int_{t^{n-1}}^t k_{-\alpha}(t-s)u(s)\mathrm{d}s - \frac{t^{-\alpha}u(0)}{\Gamma(1-\alpha)}}_{L^{\alpha}u} + \underbrace{\int_0^{t^{n-1}} k_{-\alpha}(t-s)u(s)\mathrm{d}s}_{H^{\alpha}u},$$

where $P.V.$ stands for the principal value with

$$P.V. \int_a^t k_{-\alpha}(t-s)u(s)\mathrm{d}s = \frac{1}{\Gamma(1-\alpha)}\frac{\mathrm{d}}{\mathrm{d}t}\int_a^t (t-s)^{-\alpha}u(s)\mathrm{d}s.$$

For $j = 1, 2, \cdots, N$, the local part $Lu(t_j^n)$ is calculated by

$$L^{\alpha}u(t_j^n) \approx L^{\alpha}u_j^n = P.V. \int_{t^{n-1}}^{t_j^n} k_{-\alpha}(t_j^n-s)(\Pi^p u)(s)\mathrm{d}s - \frac{u(0)}{\Gamma(1-\alpha)}(t_j^n)^{-\alpha}.$$

- **Step 3)** By (B.2), the history part $H^\alpha u(t_j^n), j = 1, 2, \cdots, p$, is approximated by

$$
\begin{aligned}
H^\alpha u(t_j^n) &\approx \int_0^{t^{n-1}} k_{-\alpha}(t_j^n - s) \Pi^p u(s) \mathrm{d}s \\
&\approx \sum_{k=1}^Q w_k \int_0^{t^{n-1}} e^{-\lambda_k(t_j^n - s)} \Pi^p u(s) \mathrm{d}s \\
&= \sum_{k=1}^Q w_k e^{-\lambda_k(t_j^n - t^{n-1})} \underbrace{\int_0^{t^{n-1}} e^{-\lambda_k(t^{n-1} - s)} \Pi^p u(s) \mathrm{d}s}_{Y_k(t^{n-1})} \\
&= \sum_{k=1}^Q w_k e^{-\lambda_k(t_j^n - t^{n-1})} Y_k(t^{n-1}) = {}_F H^\alpha u_j^n,
\end{aligned}
\tag{B.3}
$$

where $Y_k(t)$ satisfies the following linear ODE

$$
Y_k'(t) = -\lambda_k Y_k(t) + \Pi^p u(t), \quad Y_k(0) = 0,
$$

which can be exactly solved by the following recurrence formula

$$
Y_k(t^{n-1}) = e^{-\lambda_k(t^{n-1} - t^{n-2})} Y_j(t^{n-2}) + \int_{t^{n-2}}^{t^{n-1}} e^{-\lambda_k(t^{n-1} - t)} \Pi^p u(t) \mathrm{d}t, \quad Y_j(0) = 0.
$$

- **Step 4)** Output ${}_F \delta_t^\alpha u_j^n = L^\alpha u_j^n + {}_F H^\alpha u_j^n$, $j = 1, 2, \cdots, p$.

We are now in a position to develop the fast time-stepping collocation method for solving (B.1).

The fast time-stepping collocation method for (B.1) is: Given $U_{\ell,j}^k$, $k \leq n-1$, $0 \leq j \leq p$, $n \geq 1$, find $U_{\ell,j}^n$, such that

$$
{}_F \delta_t^\alpha U_j^n = f(t_j^n, U_{\ell,j}^n), \quad 1 \leq j \leq p, \quad 1 \leq \ell \leq d.
\tag{B.4}
$$

The above equation can be rewritten as

$$
\begin{aligned}
&P.V. \int_{t^{n-1}}^{t_j^n} k_{-\alpha_\ell}(t_j^n - s)(\Pi^p U_\ell)(s) \mathrm{d}s - f_\ell(t^n, U_{\ell,j}^n) \\
&= -{}_F H^{\alpha_\ell} U_{\ell,j}^n + \frac{(t_j^n)^{-\alpha_\ell}}{\Gamma(1-\alpha_\ell)} \phi_\ell, \quad 1 \leq j \leq N, \quad 1 \leq \ell \leq d.
\end{aligned}
\tag{B.5}
$$

The unknowns $u_j^n (1 \leq j \leq N)$ in the above nonlinear system can be derived by the Newton iteration method.

We can rewrite the FOHH model (4.3)-(4.4) can be solved by (B.5).

Some computational issues are addressed below.

(i) Generally speaking, the solution of the FODE, e.g., (B.1), has a singularity at the origin ($t = 0$ for (B.1)), we need to refine the mesh near the origin such that high accurate numerical solutions are obtained. For the numerical method (B.5), for $t \in [0, \Delta T]$, the mesh t^n can be defined by $t^n = \Delta T (n/M_1)^r$, $r \geq 1$. For $t \in [\Delta T, T]$, the uniform mesh is used which can be given by $t^n = t^{M_1} + (n - M_1)\Delta t$, $\Delta t = (T - \Delta T)/M_2$, $M_2 = \lfloor (T - \Delta T)/(t^{M_1} - t^{M_1 - 1}) \rfloor$.

(ii) For the fractional LIF model with the current I being piecewise constant, the solution also has a singularity at the points of the discontinuity of $I(t)$. We can use the strategy of (i) to refine the mesh near the discontinuity points when we use the method (B.5) to solve the fractional LIF.

References

[1] S. A. Aamir, Y. Stradmann, P. Müller, C. Pehle, A. Hartel, A. Grübl, J. Schemmel, and K. Meier, *An accelerated LIF neuronal network array for a large-scale mixed-signal neuromorphic architecture*, IEEE Transactions on Circuits and Systems I: Regular Papers **65** (2018), no. 12, 4299–4312.

[2] A. M. AbdelAty, M. E. Fouda, and A. M. Eltawil, *On numerical approximations of fractional-order spiking neuron models*, Communications in Nonlinear Science and Numerical Simulation **105** (2022), 106078.

[3] M. Armanyos and A. G. Radwan, *Fractional-order Fitzhugh-Nagumo and Izhikevich neuron models*, 2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2016, pp. 1–5.

[4] G. Ascione and E. Pirozzi, *On fractional stochastic modeling of neuronal activity including memory effects*, Computer Aided Systems Theory – EUROCAST 2017 (Cham) (R. Moreno-Díaz, F. Pichler, and A. Quesada-Arencibia, eds.), Springer International Publishing, 2018, 3–11.

[5] A. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, *Automatic differentiation in machine learning: A survey*, Journal of Machine Learning Research **18** (2017), no. 1, 5595–5637.

[6] S. Basir, *Investigating and mitigating failure modes in physics-informed neural networks (PINNs)*, Communications in Computational Physics **33** (2023), 1240–1269.

[7] C. Beck, S. Becker, P. Cheridito, A. Jentzen, and A. Neufeld, *Deep splitting method for parabolic PDEs*, SIAM Journal on Scientific Computing **43** (2021), no. 5, A3135–A3154.

[8] B. Benjamin, P. Gao, E. McQuinn, S. Choudhary, A. Chandrasekaran, J. M. Bussat, R. Alvarez-Icaza, J. Arthur, P. Merolla, and K. Boahen, *Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations*, Proceedings of the IEEE **102** (2014), 1–18.

[9] A. Borst and F. E. Theunissen, *Information theory and neural coding*, Nature Neuroscience **2** (1999), no. 11, 947–957.

[10] O. Brandibur and E. Kaslik, *Stability analysis for a fractional-order coupled Fitzhugh-Nagumo-type neuronal model*, Fractal and Fractional **6** (2022), no. 5, 257.

[11] R. Brette, M. Lilith, T. Carnevale, M. Hines, D. Beeman, J. Bower, M. Diesmann, A. Morrison, P. Goodman, F. Harris, M. Zirpe, T. Natschläger, D. Pecevski, B. Ermentrout, M. Djurfeldt, A. Lansner, O. Rochel, T. Viéville, E. Muller, and A. Destexhe, *Simulation of networks of spiking neurons: A review of tools and strategies*, Journal of Computational Neuroscience **23** (2008), 349–398.

[12] C. Chen, Z. Wang, and Y. Yang, *A new operator splitting method for american options under fractional Black-Scholes models*, Computers and Mathematics with Applications **77** (2019), no. 8, 2130–2144.

[13] J. M. Chignon, J. P. Lepine, and J. Ades, *Panic disorder in cardiac outpatients*, American Journal of Psychiatry **150** (1993), no. 5, 780–785.

[14] A. Dokoumetzidis and P. Macheras, *Fractional kinetics in drug absorption and disposition processes*, Journal of Pharmacokinetics And Pharmacodynamics **36** (2009), no. 2, 165–178.

[15] A. Dokoumetzidis, R. Magin, and P. Macheras, *Fractional kinetics in multi-compartmental systems*, Journal of Pharmacokinetics and Pharmacodynamics **37** (2010), 507–24.

[16] L. Einkemmer and A. Ostermann, *Overcoming order reduction in diffusion-reaction splitting. Part 1: Dirichlet boundary conditions*, SIAM Journal on Scientific Computing **37** (2015), no. 3, A1577–A1592.

[17] J. K. Eshraghian, *snnTorch. online: https://github.com/jeshraghian/snntorch*, 2021.

[18] J. K. Eshraghian, M. Ward, E. Neftci, X. Wang, G. Lenz, G. Dwivedi, M. Bennamoun, D. S. Jeong, and W. D. Lu, *Training spiking neural networks using lessons from deep learning*, arXiv preprint: 2109.12894, 2021.

[19] E. Faou, A. Ostermann, and K. Schratz, *Analysis of exponential splitting methods for inhomogeneous parabolic equations*, IMA Journal of Numerical Analysis **35** (2014), no. 1, 161–178.

[20] A. Ganguly, M. Saha, A. Ghosh, U. Maity, J. P. Kucera, and S. Raha, *The method of hybrid functions for the numerical solution of the Hodgkin-Huxley model*, IFAC-PapersOnLine **55** (2022), no. 1, 623–630, 7th International Conference on Advances in Control and Optimization of Dynamical Systems ACODS 2022.

[21] W. Gerstner and W. M. Kistler, *Spiking Neuron Models: Single Neurons, Populations, Plasticity*, Cambridge University Press, 2002.

[22] W. Gerstner, W. M. Kistler, R. Naud, and L. Paninski, *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition*, Cambridge University Press, 2014.

[23] A. H. Glassman and J. T. Bigger Jr, *Cardiovascular effects of therapeutic doses of tricyclic antidepressants: A review*, Archives of General Psychiatry **38** (1981), no. 7, 815–820.

[24] L. Guo, H. Wu, X. Yu and T. Zhou, *Monte Carlo FPINNs: deep learning approach for forward and inverse problems involving high dimensional fractional PDEs*, Computer Methods in Applied Mechanics and Engineering **400** (2022), 115523.

[25] E. Haghighat, D. Amini, and R. Juanes, *Physics-informed neural network simulation of multiphase poroelasticity using stress-split sequential training*, Computer Methods in Applied Mechanics and Engineering **397** (2022), 115–141.

[26] A. L. Hodgkin and A. F. Huxley, *A quantitative description of membrane current and its application to conduction and excitation in nerve*, The Journal of Physiology **117** (1952), no. 4, 500–544.

[27] H. Holden, K. H. Karlsen, and K. A. Lie, *Splitting methods for partial differential equations with rough solutions: Analysis and MATLAB programs*, vol. 11, European Mathematical Society, 2010.

[28] E. M. Izhikevich, *Simple model of spiking neurons*, IEEE Transactions on Neural Networks **14** (2003), no. 6, 1569–1572.

[29] E. M. Izhikevich, *Which model to use for cortical spiking neurons?*, IEEE Transactions on Neural Network **15** (2004), 1063–1070.

[30] A. D. Jagtap, K. Kawaguchi, and G. Karniadakis, *Adaptive activation functions accelerate convergence in deep and physics-informed neural networks*, Journal of Computational Physics **404** (2020), 109136.

[31] A. D. Jagtap, E. Kharazmi, and G. Karniadakis, *Conservative physics-informed neural networks*

on discrete domains for conservation laws: Applications to forward and inverse problems, Computer Methods in Applied Mechanics and Engineering **365** (2020), 113028.

[32] E. Kharazmi, Z. Zhang, and G. Karniadakis, *hp-VPINNs: Variational physics-informed neural networks with domain decomposition*, Computer Methods in Applied Mechanics and Engineering **374** (2021), 113547.

[33] M. M. A. Khater, R. A. M. Attia, A. H. Abdel-Aty, S. A. Khalek, Y. Al-Hadeethi, and D. Lu, *On the computational and numerical solutions of the transmission of nerve impulses of an excitable system (the neuron system)*, Journal of Intelligent and Fuzzy Systems **38** (2020), 2603–2610.

[34] C. Koch and I. Segev, *Methods in Neuronal Modeling: From Ions to Networks*, second ed., MIT Press, Cambridge, MA, USA, 1998.

[35] C. Li and F. Zeng, *Finite difference methods for fractional differential equations*, International Journal of Bifurcation and Chaos **22** (2012), no. 04, 1230014.

[36] D. Li, C. Quan, and J. Xu, *Stability and convergence of Strang splitting. Part I: Scalar Allen-Cahn equation*, Journal of Computational Physics **458** (2022), 111087.

[37] Y. Lin and C. Xu, *Finite difference/spectral approximations for the time-fractional diffusion equation*, Journal of Computational Physics **225** (2007), no. 2, 1533–1552.

[38] D. Liu and Y. Wang, *A dual-dimer method for training physics-constrained neural networks with minimax architecture*, Neural Networks **136** (2021), 112–125.

[39] L Long and G. Fang, *A review of biologically plausible neuron models for spiking neural networks*, AIAA Infotech at Aerospace 2010, AIAA Infotech at Aerospace 2010, American Institute of Aeronautics and Astronautics Inc., 2010.

[40] L. Lu, X. Meng, S. Cai, Z. Mao, S. Goswami, Z. Zhang, and G. Karniadakis, *A comprehensive and fair comparison of two neural operators (with practical extensions) based on fair data*, Computer Methods in Applied Mechanics and Engineering **393** (2022), 114778.

[41] W. Maass, *Networks of spiking neurons: The third generation of neural network models*, Neural Networks **10** (1997), no. 9, 1659–1671.

[42] H. Ma, Y. Zhang, N. Thuerey, X. Hu and O. J. Haidn, *Physics-driven learning of the steady Navier-Stokes equations using deep convolutional neural networks*, Communications in Computational Physics **32** (2022), 715-736.

[43] R. L. Magin, *Fractional calculus models of complex dynamics in biological tissues*, Computers and Mathematics with Applications **59** (2010), no. 5, 1586–1593.

[44] R. L. Magin and M. Ovadia, *Modeling the cardiac tissue electrode interface using fractional calculus*, Journal of Vibration and Control **14** (2008), no. 9-10, 1431–1442.

[45] L. D. McClenny and U. M. Braga-Neto, *Self-adaptive physics-informed neural networks using a soft attention mechanism*, arXiv preprint: 2009.04544, 2020.

[46] R. I. McLachlan, G. Quispel, and W. Reinout, *Splitting methods*, Acta Numerica **11** (2002), 341–434.

[47] Xuhui Meng and George Em Karniadakis, *A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse pde problems*, Journal of Computational Physics **401** (2020), 109020.

[48] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura, B. Brezzo, I. Vo, S. K. Esser, R. Appuswamy, B. Taba, A. Amir, M. D. Flickner, W. P. Risk, R. Manohar, and D. S. Modha, *Artificial brains. A million spiking-neuron integrated circuit with a scalable communication network and interface*, Science (New York, N.Y.) **345** (2014), no. 6197, 668–673.

[49] A. Mondal, S. K. Sharma, R. K. Upadhyay, and A. Mondal, *Firing activities of a fractional-order Fitzhugh-Rinzel bursting neuron model and its coupled dynamics*, Scientific Reports **9** (2019),

15721.

[50] K. B. Oldham and J. Spanier (eds.), *The fractional calculus: Theory and applications of differentiation and integration to arbitrary order*, Mathematics in Science and Engineering, vol. 111, Elsevier, 1974.

[51] G. Pang, L. Lu, and G. Karniadakis, *FPINNs: Fractional physics-informed neural networks*, SIAM Journal on Scientific Computing **41** (2019), no. 4, A2603–A2626.

[52] I. Petráš and R. L. Magin, *Simulation of drug uptake in a two compartmental fractional model for a biological system.*, Communications in Nonlinear Science & Numerical Simulation **16 12** (2011), 4588–4595.

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

[54] S. Shekarpaz and H. Azari, *An inverse problem of identifying two unknown parameters in parabolic differential equations*, Iranian Journal of Science and Technology, Transactions A: Science **42** (2018), 2045–2052.

[55] S. Shekarpaz and H. Azari, *Splitting method for an inverse source problem in parabolic differential equations: Error analysis and applications*, Numerical Methods for Partial Differential Equations **36** (2020), no. 3, 654–679.

[56] S. Shekarpaz, M. Azizmalayeri, and M. H. Rohban, *PIAT: Physics informed adversarial training for solving partial differential equations*, arXiv preprint: 2207.06647, 2022.

[57] H. H. Sherief, A. M. A. El-Sayed, S. H. Behiry, and W. E. Raslan, *Using fractional derivatives to generalize the Hodgkin–Huxley model*, Fractional Dynamics and Control (D. Baleanu, J. A. T. Machado, and A. C. J. Luo, eds.), Springer New York, New York, NY, 2012, pp. 275–282.

[58] K. Shukla, A. D. Jagtap, and G. Karniadakis, *Parallel physics-informed neural networks via domain decomposition*, arXiv preprint: 2104.10013, 2021.

[59] G. Strang, *On the construction and comparison of difference schemes*, SIAM Journal on Numerical Analysis **5** (1968), 506–517.

[60] W. Tan, C. Fu, C. Fu, W. Xie, and H. Cheng, *An anomalous subdiffusion model for calcium spark in cardiac myocytes*, Applied Physics Letters **91** (2007), no. 18, 183901.

[61] W. Teka, D. Stockton, and F. Santamaria, *Power-law dynamics of membrane conductances increase spiking diversity in a Hodgkin-Huxley model*, PLOS Computational Biology **12** (2016), no. 3, 1–23.

[62] W. W. Teka, R. K. Upadhyay, and A. Mondal, *Fractional-order leaky integrate-and-fire model with long-term memory and power law dynamics*, Neural Networks **93** (2017), 110–125.

[63] W. W. Teka, R. K. Upadhyay, and A. Mondal, *Spiking and bursting patterns of fractional-order Izhikevich model*, Communications in Nonlinear Science and Numerical Simulation **56** (2018), 161–176.

[64] M. F. Tolba, A. H. Elsafty, M. Armanyos, L. A. Said, A. H. Madian, and A. G. Radwan, *Synchronization and FPGA realization of fractional-order Izhikevich neuron model*, Microelectronics Journal **89** (2019), 56–69.

[65] H. F. Trotter, *On the product of semi-groups of operators*, Proceedings of the American Mathematical Society **10** (1959), no. 4, 545–551.

[66] S. H. Weinberg, *Membrane capacitive memory alters spiking in neurons described by the fractional-order Hodgkin-Huxley model*, PLOS ONE **10** (2015), no. 5, 1–27.

[67] S. H. Weinberg, *Membrane capacitive memory alters spiking in neurons described by the fractional-order Hodgkin-Huxley model*, PLOS ONE **10** (2015), no. 5, 1–27.

[68] S. Westerlund and L. Ekstam, *Capacitor theory*, IEEE Transactions on Dielectrics and Electrical Insulation **1** (1994), no. 5, 826–839.

[69] L. Yang, X. Meng, and G. Karniadakis, *B-PINNs: Bayesian physics-informed neural networks for forward and inverse pde problems with noisy data*, Journal of Computational Physics **425** (2021), 109913.

[70] M. W. Yasin, M. S. Iqbal, N. Ahmed, A. Akgül, A. Raza, M. Rafiq, and M. B. Riaz, *Numerical scheme and stability analysis of stochastic Fitzhugh-Nagumo model*, Results in Physics **32** (2022), 105023.

[71] D. Zhang, L. Guo, and G. Karniadakis, *Learning in modal space: Solving time-dependent stochastic pdes using physics-informed neural networks*, SIAM Journal on Scientific Computing **42** (2020), no. 2, A639–A665.

[72] Y. Zhu, N. Zabaras, P. S. Koutsourelakis, and P. Perdikaris, *Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data*, Journal of Computational Physics **394** (2019), 56–81.