Meshfree black hole toy evolution using artificial neural networks

Lyle Kenneth M. Geraldez* and Michael Francis Ian G. Vega II

National Institute of Physics, University of the Philippines Diliman, Quezon City 1101, Philippines
*Corresponding author: lgeraldez@nip.upd.edu.ph

Abstract

We present the use of physics-informed neural networks (PINNs) for handling singularities in the context black hole singularity evolution by offering meshfree alternatives for solving partial differential equations (PDEs). This was done by implementing a spatiotemporal excision surface across the entire spacetime domain rather than iteratively excising grid points. We demonstrate a toy model of black hole evolution using a simplified form of conformal thin sandwich (XTCS) initial data fed into a 1D wave equation as an evolution proxy. While PINNs can learn the asymptotic behavior outside the excised surface, it failed to learn the behavior near it. Although such an implementation has a large room for improvement especially when considering the complexity of NR applications, PINN-based numerical solver piggy-backs on the advancement of neural network architectures which may lay the possibility for feasible hybrid numerical solvers.

Keywords: physics-informed neural networks, singularity handling

1 Introduction

The most popular method to evolve black-hole binary systems is the *moving puncture* approach where the metric is decomposed into an analytic term containing containing the singularity and a correction term that must be obtained numerically [1]. The evolution is carried out by modelling the black hole as a Brill-Lindquist initial data with wormhole topology with a three-metric ansatz of $\psi = 1 + \alpha^{-1} + u$ where the second term containing the singularity is solved analytically and u is found by solving an elliptic equation parametrized by the black hole's physical properties. Feeding this initial data into a BSSN evolver via a moving puncture gauge automatically excises the singularity without the need for artificial numerical excision. However, puncture-based initial data suffer from physical unrealism by producing no physically-realistic gravitational waves [2]. With this, it may be relevant to also study non-puncture evolution schemes by handling singularities artificially.

Recently, the use of artificial neural networks for solving PDEs has been gaining popularity. First popularized by Raissi et al [3], neural networks can be repurposed to encode the governing PDE on its loss function which is emerging as a potentially new subfield under the umbrella of scientific machine learning (SciML). PINNs offer several advantages compared to standard numerical schemes such as finite difference methods by being completely meshfree and highly scalable to systems with higher dimensionality [4]. Moreover, applications are growing in the field of computational fluid dynamics (CFD) in dealing with unknown boundary conditions, among other applications [5]. In this paper, we attempt to exploit the meshfree property of PINNs, among others, and explore applicability in excising singularities which plague numerical relativity simulations.

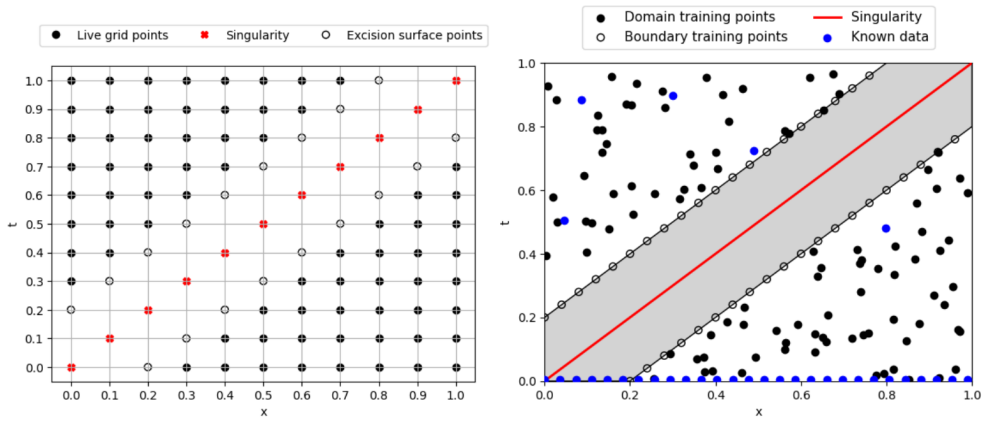
2 PINNs and spatiotemporal excision

In the standard method of lines numerical scheme, spatial discretization converts a PDE into a set of coupled ODE in time $\dot{u} = k(t, u)$ which shall be integrated from $u_0(t_0)$ to $u(t_0 + \Delta t)$ usually via the iterative Crank-Nicholson or RK4 scheme forming an algebraic differential equation. In the context of evolving black hole singularities, one must deal with the infinities along the computational domain by excising its apparent horizon and argue that no relevant data should escape black hole horizons [6]; however, even if excision boundary conditions are not physically required, it is still imperative to re-design a one-sided discretization at the boundaries which, together with handling re-emerging grid points from inside the excised region, constitute a non-trivial problem [7].

Instead of discretization, we recursively define the solution as a compositional function $\mathcal{N}^L \circ \mathcal{N}^l \circ \mathcal{N}^0$ where $\mathcal{N}^0 = \mathbf{x}$ is the initial layer, $\mathcal{N}^l = \sigma(\mathbf{W}^l \mathcal{N}^{l-1}(\mathbf{x}) + \mathbf{b}^l) \in \mathbb{R}^{\mathcal{N}_l}$ is the lth hidden layer where $l = 1, .., L-1$, and $\mathcal{N}^L = \mathbf{W}^L \mathcal{N}^L(\mathbf{x}) + \mathbf{b}^L \in \mathbb{R}^{\mathcal{N}_L}$ is the output layer. The compositional function is

characterized by the set of parameters $\theta := \{\mathbf{W}^L, \mathbf{b}^l\}$, where $\mathbf{W}^l \in \mathbb{R}^{\mathcal{N}_l \times \mathcal{N}_{l-1}}$ and $\mathbf{b}^l \in \mathbb{R}^{\mathcal{N}_l}$. L is the depth, $\dim(\mathbb{R}^{\mathcal{N}_l})$ is the width, and σ is a nonlinear activation function such that $\mathbb{R}^{\mathcal{N}_l} \in [0, 1]$. Throughout the paper, we choose $\sigma(.) = \tanh(.)$.

Solving for the solution $u(\mathbf{x})$ to a PDE $f\left(\mathbf{x}; \partial_{x_1} u, ...\partial_{x_d} u; \partial^2_{x_1 x_1} u, ...\partial^2_{x_1 x_d} u; ...\right) = 0$ on domain $\mathbf{x} \in \Omega$ with boundary condition $\mathcal{B}(u, x) = 0$ on $\partial\Omega$ amounts to finding a set of parameters $\theta = \theta^* := \text{argmin}_\theta \mathcal{L}(\theta; \mathcal{T})$ where $\mathcal{L}$ encodes the deviation from the analytic solution of the system. That is, $\mathcal{L}(\theta; \mathcal{T}) = \omega_f \mathcal{L}_f(\theta; \mathcal{T}_f) + \omega_b \mathcal{L}_b(\theta; \mathcal{T}_b)$ where $\mathcal{L}(\theta; \mathcal{T}_f)$ is the L^2 norm of the DE residual and $\mathcal{L}(\theta; \mathcal{T}_b)$ is the L^2 norm of the boundary residual. Derivative operators are also embedded within $\mathcal{N}^L$ via automatic differentiaton [4]. Here, we choose $\omega_f = \omega_b = 1$. The "learned" solution corresponding to the compositional function with optimal paremeters, $u^*(x; \theta^*) := \mathcal{N}^{*L}(\mathbf{x}, \theta^*)$, is the PINN solution to the PDE.



(a) (b)

Figure 1: **Proposed spatiotemporal excision via PINN**. Discretization-based excision must deal with grid repopulation along the upward t-climb from $t = t_0$ to $t = t_0 + \Delta t$ and stencil-modification along the excision surface (a) while PINNs train data by simultaneously sampling across the entire spatiotemporal domain (b). Excision of the singularity (shown as red markings) can be done without the problem of one-sided boundary discretization and grid repopulation by simply avoiding inclusion of training data on a defined mask region (gray) designed to include the singularity. Imposing initial condition is tantamount to training known data at $t = 0$.

In order to solve a time-dependent PDE with singularity handling, we define a set of unlabelled training data $\mathcal{T} = \{(\mathbf{x}, t)\}$ points (shown as black points) over the spatiotemporal domain $\Omega - \mathcal{X}$ where $\mathcal{X}$ (shown as gray region) is an artificially excised region containing the solution singularity. In order to find θ^*, $\mathcal{T}$ is iteratively fed into the compositional function and Adam optimization algorithm is used to minimize $\mathcal{L}$.

3 PINN toy black hole evolution

As an initial exploration to the proposed meshfree evolution using PINNs, we use a simplified toy-model of black hole evolution. Assuming spherical-symmetry, $\beta^i = 0$, the extended conformal thin sandwich (XTCS) initial data for the conformal factor of the three-metric ψ simplifies to a spherically-symmetric Laplace equation $\partial^i \partial_i \psi = 0$ with conformal flatness and isometry boundary conditions expressed in spherical coordinates as [8]

$$\frac{1}{r^2}\frac{\mathrm{d}}{\mathrm{d}r}\left(r^2 \frac{\mathrm{d}\psi_0(r)}{\mathrm{d}r}\right) = 0, \qquad \lim_{r\to\infty} \psi_0(r) = 1, \qquad \left(\frac{\partial\psi(r)}{\partial r}\right)|_a = -\frac{\psi}{2a} \tag{1}$$

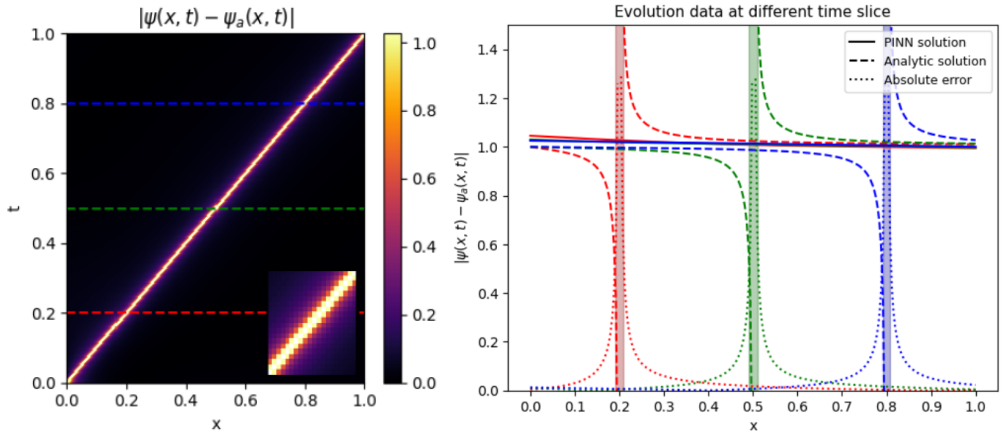
Solution to this can be physically interpreted as an isotropic Schwarzschild black hole at a constant time slice. As a proxy to a full-blown evolution scheme, we feed the x-slice of the initial data ψ_0 to an evolution toy model using a 1D wave equation in Cartesian coordinates

$$\frac{\mathrm{d}^2\psi(x,t)}{\mathrm{d}x^2} - \frac{\mathrm{d}^2\psi(x,t)}{\mathrm{d}t^2} = 0, \qquad \psi(x,0) = \psi_0(x), \qquad \frac{\mathrm{d}\psi(x,t)}{\mathrm{d}t}|_{t=0} = 1 \tag{2}$$

Here, we have considered only the x-slice by assuming time-independence of the other coordinates throughout the evolution. This propels the black hole singularity towards $\pm x$ directions carving two spatiotemporal singularity fissures at $x(t) = \pm t$; we only consider the computational domain $\Omega = [0, 1] \times [0, 1]$. Using an a priori knowledge of the analytic solution to the toy evolution,

$$\psi_a(x,t) = 1 - \frac{a}{2}\left(\frac{1}{t-x} + \frac{1}{t+x}\right), \tag{3}$$

we define a hexagonal mask $\mathcal{X}$ with vertices $(0,0),(0,a),(1-a,1),(1,1),(1,1-a),(a,0)$ and define uniformly-spaced training sets $\mathcal{T}_f$ on the domain $\Omega - \mathcal{X}$ and $\mathcal{T}_b$ on the boundary $\partial\mathcal{X}$. The mask $\mathcal{X}$ represents the throat of the black hole $x(t) = x_0(t) \pm a$ where the interior region $[x_0(t) - a, x_0(t) + a]$ enclosing the singularity $x_0(t)$ is excised at each time slice.



(a) (b)

Figure 2: **PINN-learned solution to the toy evolution**. PINN-learned solution was found for Eq. (2) excised at $a = 0.01$ using 500 domain training data, 500 boundary training data, using a shallow neural network of a single hidden layer composed by 10 neurons stopped after 10^4 epochs. Density plot was visualized via nearest-neighbour interpolation of 256x256 uniformly spaced points along Ω; central square region was magnified for visualization. The absolute error of the learned solution at the time slices $t = 0.2, 0.5, 0.8$ shown as red, green, and blue on (a) are plotted on (b) together with the excised region $x_0(t) - a < x(t) < x_0(t) + a$ of corresponding color.

For efficient building, compiling, and training of neural networks, we use a tensorflow.compat.v1 backend via the DeepXDE Python package as interface [4] to solve Eq. (1) and (2) with reference to Eq. (3). Observe that PINNs failed to learn the behavior of the solution near the excision surface. Running a hyperparameter study by varying the widths and depths on the neural network, training 10 solutions each for 10^4 epochs, and averaging the corresponding error, it was found that increasing the depth from 1 to 3 decreased the error along $x < 0$ but increased along $x > 0$. It was found out that increasing the domain data to $|\mathcal{T}_f| = 200$ and to $|\mathcal{T}_f| = 500$ from $|\mathcal{T}_f| = 100$ increased the average error. An increase in the boundary data from $|\mathcal{T}_b| = 100$ to $|\mathcal{T}_b| = 200$ increased the average error and error drops as boundary cranks up to $|\mathcal{T}_b| = 500$.

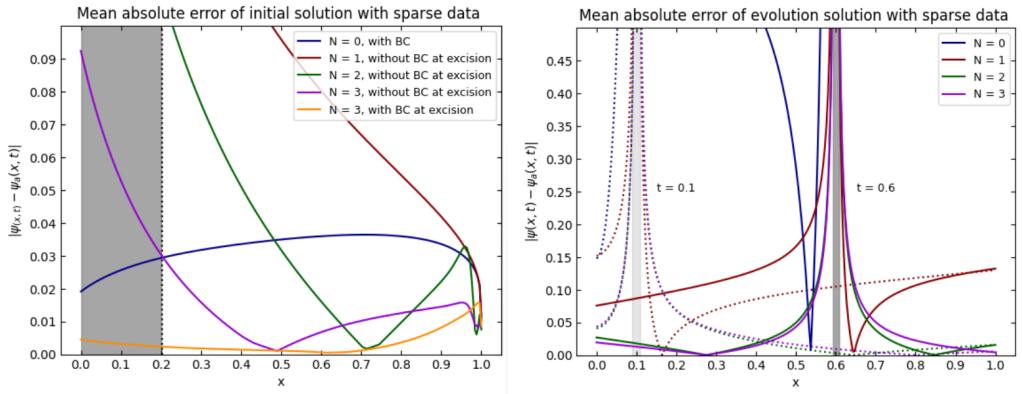


Figure 3: **Sparse data sprinkling**. Figures show the mean absolute error of 10 generated solutions with the inclusion of $N = 0, 1, 2, 3$ regulating sparse data corresponding to the blue points in Fig. 1 for $t \neq 0$. Gray region shows excision surfaces with $a = 0.2$ for the initial data solution of Eq. (1) and $a = 0.01$ for the evolution data Eq. (2).

Sparse-regulated PINNs have also been found to improve convergence compared to vanilla PINNs by morphing the loss landscape topology making it easily traversable for the optimization algorithm [9]. For both solution to the initial data Eq.(1) and evolution data Eq. (2), sprinkling N random known points gathered from the analytic solution (3) reduces the mean absolute error of the 10 generated best solutions after 10^4 epochs. Moreover, while the PINN solution got stuck at a trivial solution upon the removal of the boundary condition at the throat when solving Eq. (1), inclusion of sparse data helps the PINN escape this. Sparse regulation of PINNs provides a bridge for constructing new forms of hybrid numerical solvers using known data from analytical, numerical, or even empirical sources.

4 Conclusion

We demonstrate PINN capabilities to solve PDEs using vanilla neural networks in the context of excising singularities plaguing numerical relativity simulation. Specifically, we present a spatiotemporal excision method by designing an excision mask $\mathcal{X}$ along the spacetime geometry rather than excising grid points at each time iteration commonly used in finite-differencing schemes. We have shown that while PINNs do learn the asymptotic behavior of ψ outside the excised surface, they failed to learn the behavior near it which may or may not be attributed to optimization error due to the lack of extensive hyperparameter optimization. Although PINNs are still at its infancy, it is interesting to explore its applications on numerical relativity such as constructing appropriate excision masks using horizon-finding algorithms, utilizing its high-dimensional scalability to tackle the full Einstein field equations, or constructing hybrid numerical solver by exploiting sparse-data feeding as bridge towards other data sources (e.g. PN-approximations, analytical data, empirical data, existing numerical solvers), which may lead to a new type of a PINN-based NR code to be tested with standard numerical relativity Apples with Apples [10] testbeds. Constructing a PINN-based numerical solver piggy-backs its advancements on the development of more advanced neural network architectures.

References

[1] T. W. Baumgarte and S. L. Shapiro, *Numerical relativity: solving Einstein's equations on the computer* (Cambridge University Press, 2010).

[2] G. Reifenberger and W. Tichy, Alternatives to standard puncture initial data for binary black hole evolution, *Physical Review D* **86**, 064003 (2012).

[3] M. Raissi, P. Perdikaris, and G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *Journal of Computational physics* **378**, 686 (2019).

[4] L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis, DeepXDE: A deep learning library for solving differential equations, *SIAM Review* **63**, 208 (2021).

[5] J. F. Du Toit and R. Laubscher, Evaluation of physics-informed neural network solution accuracy and efficiency for modeling aortic transvalvular blood flow, *Mathematical and Computational Applications* **28**, 62 (2023).

[6] M. Alcubierre and B. Brügmann, Simple excision of a black hole in 3+ 1 numerical relativity, *Physical Review D* **63**, 104006 (2001).

[7] D. Shoemaker, K. Smith, U. Sperhake, P. Laguna, E. Schnetter, and D. Fiske, Moving black holes via singularity excision, *Classical and Quantum Gravity* **20**, 3729 (2003).

[8] E. Gourgoulhon, Construction of initial data for 3+ 1 numerical relativity, in *Journal of Physics: Conference Series* (IOP Publishing, 2007), vol. 91, 012001.

[9] V. Gopakumar, S. Pamela, and D. Samaddar, Loss landscape engineering via data regulation on pinns, *Machine Learning with Applications* **12**, 100464 (2023).

[10] M. C. Babiuc, S. Husa, D. Alic, I. Hinder, C. Lechner, E. Schnetter, B. Szilagyi, Y. Zlochower, N. Dorband, D. Pollney, et al., Implementation of standard testbeds for numerical relativity, *Classical and Quantum Gravity* **25**, 125012 (2008).