Robust Physics-Informed Neural Networks for Modeling Reaction-Diffusion Dynamics under Adversarial Perturbations

Keywords: Physics-informed neural networks, Non-linear PDEs, Deep learning, Scientific machine learning, Neural network solvers, Computational efficiency, Burgers' equation

1. Proposal

Partial differential equations (PDEs) are fundamental to modeling complex physical phenomena across various scientific and engineering domains. Non-linear PDEs, in particular, present significant computational challenges due to their inherent complexity and the difficulty in obtaining analytical solutions. Traditional numerical methods such as finite difference, finite element, and spectral methods have been widely employed to solve these equations, but they often encounter limitations related to computational efficiency, accuracy, and adaptability to complex geometries. The reaction-diffusion equation is a type of partial differential equation (PDE) that models the combined effects of local interactions (reactions) and spatial movement (diffusion). In its standard form, it looks like this:

$$\frac{\partial u}{\partial t} = D\nabla^2 u + f(u),$$

$u(x,t)$ represents the quantity of interest at location x and time t for example, the density of infected individuals. D is the diffusion coefficient, governing the rate at which individuals move and spread out in space. $\nabla^2 u$ is the Laplacian operator representing spatial diffusion. $f(u)$ is a nonlinear reaction term that

encapsulates local growth, death, or other dynamic processes (such as the rate of new infections and recoveries).

n an epidemic scenario, the reaction term $f(u)$ models the rate at which the infection grows or decays due to local interactions. For instance, a common choice is a logistic growth model:

$$f(u) = ru(1 - \frac{u}{k})$$

where r represents the growth rate (how quickly the disease spreads due to contact between individuals) and K is the carrying capacity (the maximum sustainable density of infections). This captures the idea that, initially, the infection can grow exponentially, but as more people become infected (or immune), the growth slows down. The diffusion term $D\nabla^2 u$ represents the physical movement of infected individuals from one location to another. This could be due to daily commuting, travel, or random movement. The higher the diffusion coefficient D, the faster the infection spreads geographically.

When you combine these two effects, the reaction-diffusion equation provides a framework to predict how an outbreak evolves both in time and across different spatial regions. For example, in modeling COVID-19:

- The reaction part might capture how quickly a local outbreak can grow due to interpersonal contacts.
- The diffusion part captures how the disease disperses as individuals move from one area to another.
- The overall equation can produce traveling wave solutions, where an initially localized outbreak spreads outwards over time, mimicking the observed patterns in many epidemic scenarios.

This mathematical framework is valuable not only for predicting the outbreak but also for informing control strategies, such as quarantines or travel restrictions, that effectively alter either the reaction (e.g., reducing contact rates) or diffusion (e.g., limiting movement) components.

1.1. Computational Complexity

- **Discretization Complexity:** The reaction-diffusion equation requires discretizing both space and time. For a 1D domain with N spatial grid points and M time steps, each evaluation of the spatial differential operator (using, say, a finite-difference scheme) typically takes $O(N)$ operations per time step. In higher dimensions, the number of grid points grows exponentially (e.g., $O(N^d)$ in d dimensions), leading to significant increases in computational cost.

- **Time-Stepping Complexity:**

  - **Explicit Methods:** Each time step is $O(N)$ (in 1D) since the update at a grid point depends on a fixed number of neighboring points. However, explicit methods must satisfy stability conditions (e.g., the CFL condition) which often force you to take very small time steps.

  - **Implicit Methods:** For stability, implicit methods (like backward Euler or Crank-Nicolson) require solving a linear (or nonlinear) system at each time step. Naively, solving a dense $N \times N$ system would be $O(N^3)$, but in practice the systems are sparse. Efficient sparse solvers or iterative methods (e.g., conjugate gradient) can often reduce this to nearly $O(N)$ per time step in 1D and $O(N^2)$ or better in 2D.

- **Overall Cost:** The total cost is the product of the per-time-step cost and the number of time steps required. If you refine the mesh to capture steep gradients or reaction zones (typical in reaction-diffusion problems), both N and M can become large, making the problem computationally intensive especially in higher dimensions.

1.2. Traditional Methods

1. **Finite Difference Methods (FDM):**

- **Space Discretization:** Replace spatial derivatives by finite-difference approximations. For example, the Laplacian in 1D is approximated as

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2}. \tag{1}$$

- **Time Integration:** Time stepping can be performed explicitly (e.g., using Euler forward) or implicitly (e.g., using Crank-Nicolson). Explicit schemes are simple but require small Δt for stability, while implicit schemes are more stable at the expense of solving a system at each step.

2. **Finite Element Methods (FEM):**
   - **Mesh Generation:** The domain is divided into elements (which can be irregularly shaped), making FEM suitable for complex geometries.
   - **Weak Formulation:** The PDE is reformulated into its weak form, which is then discretized using basis functions.
   - **System Assembly and Solution:** This typically leads to a sparse linear (or nonlinear) system. FEM offers flexibility and is especially advantageous when dealing with complex boundary conditions or non-uniform domains.
3. **Finite Volume Methods (FVM):**
   - A third traditional approach, particularly popular in computational fluid dynamics, which conserves fluxes across control volumes.

1.3. Physics-Informed Neural Networks for Reaction-Diffusion Systems

Physics-Informed Neural Networks (PINNs) offer a fundamentally different approach to solving reaction-diffusion equations compared to traditional numerical methods. This subsection outlines the key aspects of the PINN methodology as applied to our problem context.

1.3.1. Continuous Function Approximation

Unlike grid-based traditional methods, PINNs represent the solution $u(x,t)$ as a neural network $u_\theta(x,t)$ with parameters θ. This formulation provides several advantages:

- The solution is represented as a continuous and differentiable function across the entire domain.
- The approach eliminates the need for explicit mesh generation, particularly advantageous for complex geometries.
- The solution can be evaluated at arbitrary points in the domain without interpolation.

1.3.2. Physics-Constrained Optimization

The core innovation of PINNs lies in embedding the governing physics directly into the training process:

- Automatic differentiation is employed to compute spatial and temporal derivatives of $u_\theta(x,t)$ with respect to inputs x and t.
- The neural network is trained to minimize a composite loss function $\mathcal{L}$ that typically includes:

$$\mathcal{L} = \mathcal{L}_{PDE} + \lambda_{BC}\mathcal{L}_{BC} + \lambda_{IC}\mathcal{L}_{IC} \tag{2}$$

where $\mathcal{L}_{PDE}$ represents the PDE residual:

$$\mathcal{L}_{PDE} = \frac{1}{N_f}\sum_{i=1}^{N_f}\left|\frac{\partial u_\theta}{\partial t}(x_i,t_i) - D\nabla^2 u_\theta(x_i,t_i) - f(u_\theta(x_i,t_i))\right|^2 \tag{3}$$

and $\mathcal{L}_{BC}$ and $\mathcal{L}_{IC}$ enforce boundary and initial conditions, respectively.

1.3.3. Computational Advantages for Complex Systems

For reaction-diffusion systems, PINNs offer specific advantages:

- **Adaptability to Irregular Domains:** The mesh-free nature of PINNs makes them particularly suitable for modeling biological systems or chemical reactions in complex geometries.

- **Data-Physics Integration:** PINNs naturally accommodate the incorporation of measurement data alongside physical laws, enabling the solution of inverse problems and parameter identification within the same framework.

- **Multi-scale Phenomena:** Reaction-diffusion systems often exhibit behavior across multiple scales (e.g., sharp reaction fronts). PINNs can adaptively allocate their representational capacity to capture these features without requiring adaptive mesh refinement.

1.3.4. Implementation Considerations

While promising, effective implementation of PINNs for reaction-diffusion systems requires careful consideration of:

- **Network Architecture:** Deep networks with sufficient capacity are needed to capture complex solution behaviors.

- **Sampling Strategy:** Strategic placement of collocation points is crucial, particularly in regions with steep gradients or reaction zones.

- **Training Dynamics:** Multi-objective optimization involving different loss components requires careful balancing of weights and potentially curriculum learning strategies.

2. Project Objectives and Scope

This research aims to advance the application of Physics-Informed Neural Networks (PINNs) for solving reaction-diffusion equations with a specific focus on modeling pandemic spread dynamics such as COVID-19. Our work extends beyond traditional solution methods by incorporating adversarial robustness

testing to evaluate reliability under uncertainty. The project encompasses three primary objectives:

2.1. PINN Framework Development

We propose to develop a comprehensive PINN framework specifically tailored for reaction-diffusion systems that model epidemic spread. Our approach:

- Constructs a deep neural network architecture that provides a continuous approximation to the solution field across the spatiotemporal domain.
- Implements a physics-constrained loss function that enforces:

$$\mathcal{L}_{total} = \mathcal{L}_{PDE} + \lambda_{IC}\mathcal{L}_{IC} + \lambda_{BC}\mathcal{L}_{BC} \tag{4}$$

  where the PDE loss encodes the reaction-diffusion dynamics governing infection spread, while initial and boundary condition losses ensure adherence to epidemiological constraints.

- Optimizes training strategies through adaptive sampling of collocation points, focusing computational resources on regions with steep infection gradients or active transmission zones.

2.2. Adversarial Robustness Analysis

To address the critical issue of model reliability in high-stakes applications like pandemic modeling, we systematically evaluate PINN robustness through:

- Generation of targeted adversarial perturbations to input data, including:
  - Perturbations to initial condition profiles (representing uncertainty in early-stage infection data)
  - Noise injection at collocation points (simulating measurement errors or data inconsistencies)
  - Boundary condition modifications (modeling uncertainty in containment measures)

- Quantification of solution sensitivity to various perturbation classes, identifying vulnerable aspects of the PINN framework.
- Development of robustness metrics specific to epidemiological modeling to guide reliability assessments in practical deployments.

2.3. Comparative Performance Evaluation

We establish a rigorous benchmarking framework to compare our PINN approach against traditional numerical methods:

- Implementation of finite difference and finite element methods as baseline comparisons, using identical reaction-diffusion formulations.
- Evaluation across multiple performance dimensions:
  - Solution accuracy (measured against analytical solutions where available or highly-resolved numerical solutions)
  - Computational efficiency (training time vs. solution time)
  - Adaptability to changing parameters (representing evolving transmission dynamics)
  - Robustness to data perturbations (comparative vulnerability assessment)
- Analysis of trade-offs between traditional and PINN-based approaches in scenarios with varying data availability, domain complexity, and computational constraints.

The significance of this work lies in its potential to enhance computational epidemiology with more resilient and adaptive modeling tools. By explicitly addressing the robustness challenges in PINN implementations, we aim to bridge the gap between theoretical capabilities of deep learning methods and their practical deployment in critical public health applications where reliability and interpretability are paramount.