CAS MA 539 — Methods of Scientific Computing

Name: Emmy Blumenthal | **BU ID:** U87312711

Email: emmyb320@bu.edu

Final Project Proposal

Due Date: Nov 11, 2022

Physics-Informed Neural Networks

In this project, I will implement physics-informed neural networks (PINNs) in *Julia* using packages that implement neural networks (i.e., Flux.jl), automatic differentiation (e.g., Zygote.jl, AutoDiff.jl, etc.), and optimization (i.e., Optimization,jl). PINNs use neural networks to parameterize solutions by minimizing the the difference between the left-hand and right-hand side of a PDE as a least-squares problem. Specifically, let $u(\vec{x}, t)$ such that $\partial_t u + \mathcal{N}[u] = 0$ for $\vec{x} \in \Omega$ and $t \in [0, t_1]$ with boundary conditions specified as a collection of n points $((\vec{x}_i, t_i), u_i)_{i=1}^n$ with $\vec{x}_i \in \partial\Omega$. By introducing a parametrized function $u(\vec{x}, t|\vec{\lambda})$ with $\vec{\lambda} \in \mathbb{R}^p$ being parameters, the PINN is fit using the optimization problem:

$$\min_{\vec{\lambda} \in \mathbb{R}^p} \quad \int_0^{t_1} \int_\Omega \left(|\partial_t u(\vec{x}, t|\vec{\lambda}) + \mathcal{N}[u(\vec{x}, t|\vec{\lambda})]|^2 \right) d\vec{x}dt + \alpha \sum_{i=1}^{n} |u(\vec{x}_i, t_i|\vec{\lambda}) - u_i|^2. \tag{1}$$

To solve the problem on the computer, all derivatives and integrals will be discretized. Here, the first term enforces that $u(\vec{x}, t|\vec{\lambda})$ obeys the PDE and the second term enforces boundary conditions; $\alpha > 0$ is a hyper-parameter. Additionally, the operator $\mathcal{N}_{\vec{\mu}}$ can be parameterized by additional parameters $\vec{\mu} \in \mathbb{R}^q$ which are optimized in addition to the parameters $\vec{\lambda}$ with the same objective function except the provided data belongs to the region Ω instead of as boundary conditions. In an non-ideal case, this may constitute real-world or experimental data and the objective of the PINN is to learn the operator $\mathcal{N}$. In this project, I will implement this method in four cases:

1. Solving the 1D time-dependent Schrödinger equation
2. Discover parameters in the potential function used in the 1D time-dependent Schrödinger equation after generating data using the Crank-Nicolson method and/or using the PINN method from the previous step; the potential may be parameterized in three ways:
   (a) The potential will have a closed analytical form with few parameters that will be discovered
   (b) The potential will have a closed analytical form but optimized parameters will represent coefficients of a Fourier series representing the potential
   (c) The potential (maybe the entire Hamiltonian) will be parameterized entirely by a neural network
3. Solve the Liouville equation from Hamiltonian and statistical mechanics for one particle
4. Solve the Liouville equation from Hamiltonian and statistical mechanics for multiple particles

These four goals may be too ambitious, so I will proceed through them in order, completing as least item 1 and item 2(a).

References:

- https://arxiv.org/abs/1711.10561
- https://arxiv.org/abs/1711.10566
- https://en.wikipedia.org/wiki/Physics-informed_neural_networks
- https://en.wikipedia.org/wiki/Schrodinger_equation
- https://en.wikipedia.org/wiki/Liouville's_theorem_(Hamiltonian)
- https://fluxml.ai
- https://docs.sciml.ai/Optimization/stable/