Dustin Lee Enyeart

Mathematics | Physics | Programming | Technical Writing

https://dustin-enyeart.github.io

dustin.enyeart@pm.me

Summary

Deep knowledge of applied mathematics, scientific computing, theoretical physics and computational physics. Dissertation on neural networks and differential equations. Seeking to solve interesting problems in an innovated environment.

Education

Bachelor's degree in Mathematics, University of New Mexico (3.57 GPA)

Master's degree in Mathematics, Purdue University (4.00 GPA)

Doctorate degree in Mathematics, Purdue University (3.84 GPA) (December 2024)

Technical Skills

Programming Languages: Python, C/C++

Scientific Computing: Finite-Difference Method, Finite-Element Method, Numerical Linear Algebra

Physical Modeling: Computational Electromagnetism, Density Functional Theory

Artificial Intelligence: Scientific Machine Learning, PyTorch, Torch Lightning, Torch Geometric

Data Science: Probability, Statistics, Data Analysis

Parallel Computing: Threads, MPI, CUDA

Software Development: GNU/Linux, Bash, Git, GDB, CMake, Slurm, Hydra

Technical Writing: LaTeX, Markdown, reStructuredText, Sphinx

Relevant Coursework

Besides the mathematics curricula for my degrees, I have taken additional graduate-level courses on a variety of topics, such as physical modeling, computer science, artificial intelligence, material science engineering, nuclear engineering, electrical engineering and semiconductor fabrication. Furthermore, the majority of my studies have been done independently of my courses.

Selected Projects

Operator Learning: Implemented and studied several neural operator architectures to solve differential equations with PyTorch with Lightning. Studied general best practices in operator learning. Compared different loss terms and operator forms for Koopman autoencoders. Introduced novel loss terms for Koopman autoencoders that improved performance. Introduced adversarial additions for Koopman autoencoders and DeepONet that improved performance for small data sets. Trained models on GPU clusters. Scheduled jobs with Slurm. Used Hydra for configurations. Used Neptune AI for logging.

Xray Sources: Simulated several xray sources and studied their spectra. Used C++ and Geant4 to implement an xray tube and a wiggler. Used Python to implemented an undulator and a van-der-Waals source. Used Jefimenko's equations to calculate the electromagnetic fields. The density functional calculations to compute the electronic structure of the van-der-Waals materials were done with ASE and GPAW.