# Dustin Lee Enyeart

Mathematics | Physics | Programming

(505) 604-4222 dustin.enyeart@pm.me Gilbert AZ 85297 USA

https://dustin-enyeart.github.io

https://gitlab.com/users/dustin_lee/projects

https://github.com/dustin-enyeart

https://linkedin.com/in/dustin-lee-enyeart-6a15582aa

## 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, Molecular Dynamics

**Artificial Intelligence**: Scientific Machine Learning, PyTorch, Torch Lightning, Torch Script, Torch Geometric, Tensorboard

**Data Science**: Probability, Statistics, Data Analysis

**Parallel Computing**: Threads, MPI, CUDA

**Software Development**: GNU/Linux, Bash, Git, GDB, Make/CMake

**Technical Writing**: LaTeX, Markdown

## 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 schoolwork.

## Selected Projects

**Loss Functions for Koopman Architecture**: Compared common and novel loss functions for the Koopman architecture on seven different differential equations using Python and PyTorch

**Local Embedding for Koopman Architecture**: Developed a novel local embedding for the Koopman architecture to solve PDEs using Python and PyTorch

**Graph Neural Network for Koopman Architecture**: Developed a novel graph-neural-network local embedding for the Koopman architecture to solve PDEs using Python, PyTorch and Torch Geometric

**X-ray Tube Simulation**: Modeled an x-ray tube using C++ and Geant4

**Wiggler Simulation**: Computed photons and their spectrum from electrons in a wiggler using C++ and Geant4

**Undulator Simulation**: Computed electromagnetic waves and their spectrum from an electron in an undulator using Python

**Van-der-Waals Source Simulation**: Computed electromagnetic waves and their spectrum from an electron in a van-der-Waals material using Python where the electronic structure of material was computed using ASE and GPAW

**Bias Detection in Wikipedia**: Implemented natural language processing methods for bias detection using Python and PyTorch

**ODE Schemes Comparison**: Compared finite-difference schemes for time-dependent ODEs using Julia