Jeremy M. Myers, Ph.D.

Berlin, DE jeremy.moulton.myers@gmail.com +17576517679 linkedin github website

Summary

I am a research software engineer and computational scientist with expertise in numerical methods for machine learning and artificial intelligence and experience developing performant code. I am seeking a full-time position blending robust software engineering and research where high-performance computing, software architecture, and AI/ML converge.

Education

Ph.D., Computer Science, College of William & Mary, Williamsburg, VA, USA, May 2024
M.S., Mathematical Sciences, Virginia Commonwealth University, Richmond, VA, USA, May 2017
B.S., Mathematical Sciences, Virginia Commonwealth University, Richmond, VA, USA, Dec 2014
B.A., International Affairs, James Madison University, Harrisonburg, VA, USA, May 2009

Experience

R&D Science & Engineering, Computer Science, Sandia National Labs, CA, USA Apr 2024–Aug 2025

- Designed and implemented a research-focused, statistical software component for supervised machine learning and tensor regression (NumPy, statsmodels). It is scheduled for integration as a sub-package within the Python Tensor Toolbox (pyttb) in an upcoming release.
- Authored and open-sourced a federated-learning algorithm (C++, MPI, CUDA, OpenMP) as part of a collaborative contribution to Genten, an unsupervised machine learning project for tensors and multi-way arrays.
- Developed algorithm-based data error detection and recovery mechanisms for HPC systems (C++, MPI) inside a production-grade conjugate gradient solver with distributed parallelism.
- Applied advanced NLP techniques, including Transformer-based model training, prompt engineering, and fine-tuning (PyTorch, Hugging Face Transformers, CUDA/cuDNN for GPU acceleration), to enable an LLM to model complex rule-based logic and generate prototype Python implementations for experimental wargames, tracked with Weights & Biases for experiment reproducibility.
- College ofborated across institutions to evaluate and compare methodological approaches to an anomaly detection application of cyber traffic data, ensuring alignment and knowledge transfer between teams for customer-facing deliverables.

R&D Graduate Intern, Sandia National Labs, CA, USA Jun 2019–Apr 2024

- Created two tensor algorithms, executed large-scale computational simulations on HPC clusters (SLURM, MPI, OpenMP, CUDA, MATLAB) for model tuning (PyTorch, Dakota), analyzed data (pandas, Jupyter) conducted large-scale experiments on HPC clusters, and published the results in a peer-reviewed journal.
- Developed a software-based load-balancing strategy for sparse tensors on CPUs (OpenMP) and GPUs (CUDA), resulting in up to 10× speedup on NVidia Volta GPUs.
- Presented novel research at numerous academic conferences, workshops, and symposia as talks and at poster sessions as well as to customers and other stakeholders.

Graduate Research Assistant, College of William & Mary, VA, USA Jan 2019–Apr 2024

- Architected and built a library for matrix sketching and streaming data called Skema (C, C++, CMake), leveraging several parallel backends (OpenMP, CUDA) for high performance.
- Derived convergence criteria for kernel learning solvers applied to streaming data-driven stock forecasting.
- Co-authored a winning National Science Foundation (NSF) grant proposal ($680K) for machine learning in asset price prediction.

Graduate Teaching Assistant, College of William & Mary, VA, USA Aug 2017–Jan 2019

- Applied expertise in Python, C/C++, programming languages theory, and discrete mathematics to support instruction, guiding students through code implementation and optimization, clarify complex algorithmic concepts, and strengthening analytical reasoning and software development skills.

Graduate Teaching Assistant, Virginia Commonwealth University, VA, USA Aug 2015–Aug 2017

- Served as instructor of record for two undergraduate mathematics courses, responsible for curriculum development, classroom instruction, student evaluation, in addition to standard teaching assistant duties.

Skills

Programming Languages: Python, C, C++, CUDA, Rust, MATLAB, R
Frameworks/Libraries: PyTorch, NumPy, pandas, Hugging Face Transformers, Apache Spark
Software/Tools: Git, Linux, Jupyter Notebook, VS Code