Jack Myers

Computational Scientist · Simulation & GPU-Accelerated Numerical Software

Portland, OR (remote-friendly) • jack@myersjack.dev • linkedin.com/in/j-myers • github.com/jackm97 • myersjack.dev

Computational scientist with a physics foundation and eight years building numerical software end-to-end — from variational formulation to GPU execution. Architected an OpenMDAO-based multiphysics optimization product used for customer design studies; invented a novel meshless lattice Boltzmann method for 3D acoustics; two years of driver-level GPU systems work on Intel's Level Zero runtime.

EXPERIENCE

GPU Software Development Engineer — *Intel Corporation* (Hillsboro, OR) Jun 2024 – Aug 2026

- Root-caused a class of subtle pre-silicon simulation failures (host-buffer handling in TBX mode) that had been deferred for over a year; designed and shipped a page-fault-based host memory manager that retired the entire bug class — announced across three teams (about 50 engineers).
- Became team go-to for the TBX GPU simulation platform: local & CI operation, tracing and log-harvesting workflows (adopted by peers), and Level Zero edge-case guidance for test authors.
- Implemented Level Zero specification features and fixes in the NEO driver: multi-range `zeVirtualMemUnmap`, parameter validation, and GPU memory-telemetry corrections (bandwidth/capacity reporting, missing platform support) for two GPU product lines.
- Built a reproducible pixi/conda development workspace for the NEO driver (build, test, and CI-swap tasks with a validation app) enabling local driver changes to be validated against CI workloads.
- Supported release readiness of an Argonne (Aurora supercomputer) feature via oneDNN debugging.

Software Engineer — *LaunchPoint Electric Propulsion Solutions* (Goleta, CA) Jun 2018 – Apr 2024

- Principal software author of MAAP (Modeling Advanced Aircraft Propulsion) — LaunchPoint's commercial physics-based digital-twin suite for electric/hybrid-electric aircraft propulsion design — from Matlab prototype through full Python/OpenMDAO re-architecture to its 2021 commercial launch; the product remains on the market today.
- Productized electromagnetic machine theory from co-founder Brad Paden (UCSB) in a rapid loop with the customer-facing lead — design-tradeoff turnaround fell from weeks to hours; contract proposal throughput rose 200% in two years.
- Led the Matlab→Python migration (eliminating most licensing costs); introduced CI; authored the team's theoretical model documentation.

Graduate Researcher — *Computer Graphics & Visualization Group, Oregon State University* (concurrent with full-time role) Sep 2020 – Oct 2024

- Invented a meshless lattice Boltzmann method derived from nonlocal vector calculus — the first fluid-simulation approach to immersive 3D acoustics in computer graphics; retains grid-LBM's embarrassing parallelism on unstructured point clouds (RBF/LRPIM interpolation, kNN supports, sparse-matrix streaming, CM-MRT collision, PML boundaries).
- Simulated real 3D scenes (cathedral, 4M nodes; city, 1.9M nodes) with moving sources and observers on a single RTX 4090; validated against analytic solutions, Doppler shift, and double-slit interference. Two SIGGRAPH submissions (2024, Asia 2024).
- Method generalizes beyond LBM: nonlocal gradient/divergence/Laplacian operators apply to arbitrary PDEs — the foundation of current framework work below.

INDEPENDENT RESEARCH & OPEN SOURCE

CospanMDAO — GPU-first differentiable modeling & optimization framework (OpenMDAO-inspired) in modern C++/SYCL (AdaptiveCpp): composable component systems, zero-copy connection aliasing, topological execution, deep-equilibrium solves with hand-written adjoints, and runtime JIT kernel fusion.
Meshless Light-Transport Renderer — kinetic GI on unstructured point clouds (on CospanMDAO): *simulate once, render from any viewpoint* — baked directional radiance rasterizes in real time. SIGGRAPH 2027 submission in preparation.
Mycelium — Neural-field architecture research (Amari/Wilson–Cowan lineage): dynamical, stateful models with persistent inference-time state, trained via equilibrium/adjoint methods on CospanMDAO.
SlabKit & SYCLBuildKit — SYCL USM memory-pool library (N-d arrays, RAII, kernel-safe views); relocatable conda toolchains for AdaptiveCpp (LLVM 20, all GPU backends).

EDUCATION

M.S. Computer Science — *Oregon State University* 2020 – 2024
Thesis: *A Highly-Parallel Meshless LBM Leveraging Nonlocal Calculus for 3D Acoustics in Graphics* · GPA 3.87 · self-funded while employed full-time

B.S. Physics, College of Creative Studies — *UC Santa Barbara* 2015 – 2019
Summa cum laude · GPA 3.92 · Regents Scholarship · CCS research track

PUBLICATIONS

- *Electric Propulsion Component Sizing for Optimal Aircraft Configuration* — VFS 76th Annual Forum & Technology Display, 2020 (co-author).

SKILLS

Languages & GPU: modern C++ (17/20/23), Python, SYCL/AdaptiveCpp, CUDA, Level Zero
Numerical: meshless/particle discretizations, LBM & kinetic methods, RBF interpolation, nonlinear optimization (OpenMDAO), autodiff & adjoints, sparse linear algebra
Systems: Linux, CMake, pixi/conda, CI/CD, GPU profiling & debugging