GEORGE F. SWITZER

Aerospace Engineer • High Performance Computing Specialist
Newport News, VA 23602 • (757) 291-1556 • 4gswitz@gmail.com

HPC APPLICATIONS PORTFOLIO

Over three decades of experience designing, implementing, and optimizing HPC workflows for large-scale CFD, atmospheric modeling, and data analysis at NASA Langley Research Center. Proven expertise in parallel programming, scientific software development, and user-focused technical support for research teams and external partners.

Technical Skills

CFD analysis and support (FUN3D) • HPC systems and batch scheduling (MPI, GPU acceleration) • Python automation and post-processing (PyTecplot, NumPy, Pandas) • Parallel workflow design and large-scale data processing • Simulation validation and diagnostic analysis • Scientific code maintenance and modernization • Reproducible data products and documentation

Project 1 – CFD Analysis & Automation on Modern Architectures: Boeing X-66 Truss-Braced Wing

Role: HPC Facilitator / CFD Support
Objective: Support aerodynamic CFD analysis and create scalable post-processing workflows on CPU and GPU-accelerated systems.

Technical Environment

FUN3D (unstructured mesh CFD) • MPI on Linux/Unix supercomputers • GPU-accelerated systems including Grace Hopper class architectures • Python, PyTecplot • Shell scripting and batch job orchestration

Problem

Post-processing FUN3D results for the X-66 configuration was largely manual and interactive, which did not scale to large test matrices or rapid design iterations—especially as workloads moved onto modern GPU-accelerated systems.

Contribution

Developed Python/PyTecplot automation to standardize the generation of plots and derived quantities from FUN3D output, independent of underlying CPU or GPU hardware. Structured output directories and scripts so that analysts could process large ensembles of runs consistently, and documented the workflow for reuse across the project team and partners.

Impact

Reduced manual post-processing effort, improved consistency across analysis campaigns, and enabled faster, more reliable comparison of results from CPU and GPU runs for the X-66 project and collaborating organizations.

Project 2 – High Ice Water Content (HIWC) LES & Radar Modeling

Role: Radar Engineer / Principal Investigator (NAS Supercomputing Project)
Objective: Use large-eddy simulations to study cloud microphysics and radar response in high-ice-content environments for aviation safety and sensor development.

Technical Environment
TASS Large Eddy Simulation (LES) model • MPI-based HPC workflows on NASA supercomputers • Python (e.g., NumPy, Pandas) for time-series and spatial data • Custom data formats, HPC file systems, structured directory layouts

Problem
Large 3D, time-dependent simulation outputs needed efficient extraction, aggregation, and visualization to support radar model validation and deliver usable products to NASA and industry partners.

Contribution
Designed and implemented Python workflows to parse LES output, compute key diagnostic quantities, and generate visualizations for technical reports and partner reviews. Organized data products for easier downstream analysis by sensor and modeling teams, and iteratively refined workflows and model setups to use HPC resources more efficiently.

Impact
Delivered validated datasets and analysis products that supported sensor design and evaluation, while improving throughput and scalability of LES-based studies on NASA supercomputers.

Project 3 – Aviation Safety & Wake Turbulence Data Analysis

Role: Senior/Support Scientist and Computer Scientist
Objective: Support turbulence detection, wake prediction, and aviation safety studies through robust data analysis and tool maintenance.

Technical Environment
TASS Fortran atmospheric simulation code for mesoscale and wake-related flows • Python and shell scripting for post-processing and data analysis • Custom empirical and reduced-order models derived from TASS output • Linux/Unix production environments • Data products supporting certification activities and NTSB accident investigations (e.g., American Airlines 587, US Air 1016)

Problem
Radar and flight data, together with large TASS simulation outputs, needed to be transformed into consistent, reproducible data products and simplified models to support certification decisions, safety investigations, and the development of operational guidance.

Contribution
Maintained and modernized tools used to run TASS and post-process its output, extracting salient flow and atmospheric features relevant to wake and turbulence behavior. Developed and automated empirical and reduced-order models based on analyzed TASS results, enabling faster evaluation of scenarios without rerunning full simulations. Ensured that processing workflows and model generation steps were documented and repeatable, and adapted tools to new datasets and investigation needs under time constraints.

Impact

Provided reliable, well-documented data products and derived models that supported aviation safety studies and accident investigations, and helped translate detailed TASS simulations into forms usable by certification and operational decision-makers.

Project 4 – Accelerating CERES Data Processing with Parallel Python Workflows

Role: HPC Facilitator / Implementor
Objective: Dramatically reduce turnaround time for processing large CERES satellite datasets by redesigning the workflow to exploit embarrassingly parallel computation.

Technical Environment

CERES satellite data products • Python for data processing and automation • Embarrassingly parallel workload design (file- or case-based decomposition) • Linux/Unix HPC systems and batch schedulers • Shell scripting for job submission and orchestration

Problem

Existing CERES data processing workflows were largely serial or minimally parallel, leading to long wall-clock times and limiting the ability of researchers to iterate on analyses or process large data volumes efficiently.

Contribution

Redesigned the CERES processing pipeline in Python to decompose the workload into independent units that could be distributed across many cores/nodes with minimal interprocess communication. Automated job creation and submission so large collections of files or cases could be processed concurrently, and ensured that outputs were organized and merged in a reproducible way. Validated the new workflow against legacy results to confirm scientific equivalence.

Impact

Achieved approximately a 97% reduction in total processing time for targeted CERES workflows, transforming multi-day runs into much shorter turnaround. This enabled more frequent analyses, faster iteration on scientific questions, and more effective use of available HPC resources by the research team.
