2027 Summer Internships in Parallel Computational Science (SIParCS) Project Descriptions

Undergraduate students

Project 01. Enhancing the user experience for MELODIES MONET
Project 02. NA-CORDEX - Actionable science for water resource practitioners
Project 03. Scalable High Resolution Data Assimilation
Project 04. Translating Science: Building Community and Use Cases for UXarray
Project 05. What are CREDIT Emulators Learning?

Graduate students

Project 06. A Study of Quantum Applications within Earth System Modeling (ESM)
Project 07. Can Data Assimilation of Spatially and Temporally Dense Surface Pressure Observations Reveal the State of the Entire Troposphere?
Project 08. Decentralized Federated Learning Sensornets for Community Resilience and Actionable Science
Project 09. EcoMAP: Ecological Mapping and AI-Assisted species Profiling
Project 10. Emulators and GPUs for Data Assimilation
Project 11. Ensemble vs. Regional Refinement: Satellite Data Assimilation for Urban Air Quality Forecasting
Project 12. Exploring Agentic Software Engineering and Spec-Driven Development for Scientific Software
Project 13. Exploring AI-Assisted Interfaces for Search and Discovery using Scientific Embeddings
Project 14. Large GPU Ensembles and Rare-Event Sampling for Weather and Climate
Project 15. MPAS-A and Evaluating AI-assisted Coding Methods
Project 16. Using agentic AI to accelerate the modernization of a large scientific codebase

Note: Students may apply to up to two (2) SIParCS projects.

Non-Technical Project

Graduate

CISL Outreach Development Education (CODE) Intern

Project 01. Enhancing the user experience for MELODIES MONET

Areas of Interest in order of relevance: Software Engineering, Data Science, Visualization

Description: MELODIES MONET (Model EvaLuation using Observations, DIagnostics and Experiments Software, coupled with the Model and ObservatioN Evaluation Toolkit), co-developed by NSF NCAR and NOAA, provides a modular framework in Python that integrates different chemistry model outputs and airborne, ground-based, and satellite observations. The overarching objective of the project is to provide a consistent approach for evaluating a wide range of models (both operational, and regulatory) against a large and growing suite of observations. A number of observational datasets and model formats are currently available in MELODIES MONET. Some examples include EPA AirNow and AERONET ground based measurements, TROPOMI and TEMPO satellite data, and NSF NCAR CESM finite volume and variable resolution grids and NOAA UFS-AQM models.

Currently, to configure MELODIES MONET model-measurement comparisons, a yaml file is used. This can be cumbersome to configure, especially for new users. To address this challenge and enable a streamlined user experience we aim to develop an interface to generate yaml files in a more intuitive and human-interpretable manner.

Students: The project is open to undergraduate students.

Skills and Qualifications:

- Intermediate programming skill in Python is required including knowledge of the following topics: Matplotlib, Dask, xarray, yaml
- Understanding of Claude use and best practices
- Familiarity with Jupyter Notebook and Github.
- Interest in atmospheric composition modeling is welcome.
- The ability to work efficiently and collaboratively in a team is strongly preferred.

Project 02. NA-CORDEX - Actionable science for water resource practitioners

Areas of Interest in order of relevance: Data Science, Geostatistics, Visualization

Description: Water resource practitioners make decisions with consequences that last for decades, from sizing reservoirs and stormwater systems to allocating water supplies among cities, farms, and ecosystems. High-resolution regional climate projections like North American Coordinated Regional Downscaling Experiment (NA-CORDEX) can be used to inform these decisions. However, raw model outputs are often difficult for practitioners to use due to the fact that the files are large, the data formats are unique, and the variables rarely map directly onto the metrics that water utilities and planners actually track. Bridging that gap requires translating model output into decision-relevant information, such as return periods for heavy rainfall, changes in snow water equivalent, or shifts in seasonal runoff, and presenting it in ways that reflect how water sector decision-makers think about risk and uncertainty.

Over the summer, the successful applicant will work with researchers in the Computational and Information Systems Laboratory (CISL) and the Research Applications Lab (RAL) to make NA CORDEX data hosted on NSF NCAR Geoscience Data Exchange (GDEX) more accessible and usable. Collaborating with computational, social, and atmospheric scientists, the candidate will design a suite of Jupyter Notebooks to enable climate data retrieval, analytics, and visualization. Prior research in RAL on the water resources community climate information needs will inform the design of the notebooks. The intern will learn about how to make climate data more actionable and usable, develop metrics used in the water resource community such as Intensity-Duration-Frequency (IDF) curves, and Probable Maximum Precipitation (PMP), and explore unique ways of plotting this information through maps and graphs that serve the user community needs.

Students: The project is open to undergraduate students.

Skills and Qualifications:

- Able to troubleshoot problems independently
- Track record of teaching yourself a skill to complete a project
- Creative approach to problem solving as well as a willingness to learn.
- Ability to collaborate effectively with an interdisciplinary team, including other interns, data analysts, engineers, and climate, computational, and social scientists
- Proficiency with Python and Jupyter notebooks; basic UNIX/Linux proficiency.
- Some experience with geospatial data processing and analysis, especially in an Earth science, atmospheric science, or meteorology context, is preferred.
- Some experience or knowledge of scientific data visualization is preferred

Project 03. Scalable High Resolution Data Assimilation

Areas of Interest in order of relevance: Application Optimization/Parallelization, Software Engineering, Numerical Methods

Description: The Data Assimilation Research Testbed (DART) is a widely used community software facility for data assimilation. One application of data assimilation is improving numerical weather prediction. To do this, DART ingests a group of model forecasts, say 80 predictions of weather in the United States, and uses statistics to combine these model forecasts with observations to produce a better estimate of the weather.

The project will start with profiling of the DART code on NSF NCAR's Derecho supercomputer. The student will use the profiling results, and their own experience to guide the direction of the project. Possible projects include the following, but other directions are welcome depending on the profiling results and the student's interests:

Distributed grids: Each model forecast lives on a grid of points covering the globe or a region. For very high resolution and unstructured grids, storing the full grid on every processor does not scale. This project would investigate how to distribute the grid information along with the model state, and how to find the grid points near an observation when those points live on another processor. This is needed to calculate what the model predicts an observation should be (the forward operator), and to limit an observation's influence to nearby points (localization). The student would use profiling data to quantify the communication cost of different distributions and implement the most promising one for a model interface. Depending on the student's interest, this may include exploring models built on AMReX, a framework for block-structured grids that is used by a growing number of GPU-capable models.

State layout and IO: The ensemble of model states is the largest data structure in DART. It is read from and written to files, and during the assimilation it is reshuffled between two arrangements: each processor holding complete forecasts, or each processor holding every forecast for one part of the state. The student may use the profiling results to focus on one of the following: alternative memory layouts that are faster for the processor to access (for example, layouts that map better to GPUs), a more efficient communication pattern for reshuffling the data, or parallel IO strategies that overlap file reads and writes with computation.

Work that improves how DART stores and moves the model state would help it handle the next generation of high resolution models, and gives the student an opportunity to design data structures that enable efficient runs on tens of thousands of processors.

Students: The project is open to undergraduate students.

Skills and Qualifications:

- Experience with a compiled language (such as Fortran, C or C++, Rust) and a willingness to read and modify a large existing code are needed.
- Familiarity with parallel programming (MPI, OpenMP) and the Linux command line is helpful.
- No prior knowledge of data assimilation or weather modeling is required.

Project 04. Translating Science: Building Community and Use Cases for UXarray

Areas of Interest in order of relevance: Scientific community engagement, Software Engineering

Description: UXarray (https://uxarray.readthedocs.io) is an open-source Python package that extends Xarray to make it easier for scientists to analyze and visualize unstructured-grid data, commonly arising from next generation models of the atmosphere and oceans. Development of UXarray is led by the NSF National Center for Atmospheric Research (NSF NCAR), with contributions from collaborators all around the world. Recently, collaborators at Penn State University and NSF NCAR received an NSF POSE award (https://pure.psu.edu/en/projects/pose-phase-ii-transitioning-uxarray-to-a-sustainable-open-source-) that will help broaden and diversify the package's user and contributor communities. The intern will have the opportunity to help support the goals of the POSE award. Through this opportunity the intern will gain experience with science communications, science translation, and community management of an open source science tool. In their role, they will:

+ Develop technical use cases and domain area examples so researchers can learn to use UXarray.
+ Learn how to set up repositories, and implement.
+ Write blog posts based on the use cases and examples.
+ Write and post social media posts that help people learn about UXarray.
+ Develop other potential communications materials, such as blog posts, visuals, and newsletter content, and help disseminate such materials.
+ Develop website copy for UXarray.
+ Engage and support users and contributors of UXarray.
+ Be involved at the leadership level of UXarray and provide input and ideas on ways to engage our community, which includes users and contributors of UXarray, and organizational partners.

Students: The project is open to undergraduate students.

Skills and Qualifications:

- Working knowledge of Earth Science data and analytical tools.
- Ability to reach out to new people, schedule calls, lead conversations, collect/record accurate information based on those conversations, and synthesize collected information.
- Ability to document technical and scientific workflows into written content that scientists and researchers can understand, follow, and replicate.
- Experience with writing science communications, such as social media posts, blog posts, etc, and be able to post/publish on a schedule.
- Development of visuals appropriate for communicating with scientists and interpreters of science.

Project 05. What are CREDIT Emulators Learning?

Areas of Interest in order of relevance: Machine Learning, Data Science, Numerical Methods

Description: The NSF NCAR CREDIT (Community Research Earth Digital Intelligence Twin) platform is enabling the development of a new generation of machine-learned Earth system emulators that can effectively mix machine learning (ML) and physics components in an end-to-end differentiable pipeline. The MILES team is training a new family of emulators spanning different levels of complexity and physics guidance and is looking for an undergrad SIParCS intern to understand what physical phenomena and concepts these emulators have learned (or not).

The interns for this project will have the opportunity to contribute to the following areas:
1. Analyzing predictions from pretrained models on how well they capture physical phenomena at different space and time scales (weather, S2S, climate) using community diagnostic tools and visualizations.
2. Testing hypotheses for why phenomena are captured in some models but not others with comparative modeling experiments.
3. Fine-tuning lower capacity models to see if they can capture previously missing phenomena.

The undergrad intern will focus on analysis and visualization tasks and will assist in running fine-tuning experiments.

Students: The project is open to undergraduate students.

Skills and Qualifications:

- Experience with Python programming and training an ML model in PyTorch or JAX.
- Experience working with geospatial data (point, vector, or gridded/raster data in formats like netCDF, zarr, GRIB, HDF5, parquet, or geoTIFF) with the Python data science stack (libraries like xarray and Jupyter).
- Demonstrated experience with self-directed learning (e.g., teaching oneself new skills outside of a required classroom context, such as through independent study, creative extracurricular activities, or personal hobbies).

Project 06. A Study of Quantum Applications within Earth System Modeling (ESM)

Areas of Interest in order of relevance: Quantum computing, Numerical Methods, Software Engineering

Description: Hardware accelerators have long played a massive role in scientific exploration and making new discoveries possible. Quantum computing has demonstrated new types of performance gains, both real and theoretical, and the need to be prepared for the next generation of hardware accelerators is paramount to scientific computing. NSF NCAR supports the ESM community with large software systems and there is a need to better understand the role that quantum will play in ESM ecosystems and its implications on performance, correctness, and sustainability.

For this project, the student will have the opportunity to explore developments in quantum implementations of classical algorithms. This will be done through several possible avenues, from exploring integration of quantum solutions with classical systems in modeling domains, evaluating the performance and accuracy of simulated results across multiple quantum frameworks, and optimization opportunities of hybrid software approaches. This will provide hands-on experience to both theory and application and help provide a foundation for further quantum explorations within ESM at NSF NCAR.

Students: The project is open to graduate students.

Skills and Qualifications:

- Background in computer science, applied mathematics, or related fields
- Familiarity with programming languages (Python, C++, Fortran, or similar languages)
- Familiarity with at least one quantum programming environment (qiskit, PennyLane, CUDA-Q, etc.) would be beneficial but is not required
- Experience with HPC would also beneficial but is not required
- Strong motivation to learn new skills, investigate open-ended problems, and work collaboratively in a research team is required

Project 07. Can Data Assimilation of Spatially and Temporally Dense Surface Pressure Observations Reveal the State of the Entire Troposphere?

Areas of Interest in order of relevance: Data Assimilation, Numerical Methods, Data Science

Description: Data assimilation is the combination of information from model forecasts with observations of the Earth system to produce a better estimate of the model's state. It is used to generate the initial conditions that are input to weather prediction models that make the forecasts we use every day. Scientists in the NSF NCAR Data Assimilation Research Section develop and use the world's most advanced ensemble data assimilation system, the Data Assimilation Research Testbed (DART). Twenty years ago, DAReS scientists hypothesized that it was possible to use only observations of surface pressure to make initial conditions for long-term global weather forecasts. This hypothesis was confirmed with DART and led to the ability to learn the detailed state of the atmosphere for all past times when surface pressure observations are available, nearly two centuries. This project will explore a related hypothesis: is it possible to predict short-term weather like convection using only surface pressure observations that are dense in space and time? The intern selected for this project will learn about ensemble data assimilation by doing some simple experiments in an idealized model that was developed by Ed Lorenz. They will then run a series of assimilation experiments with a high-resolution weather prediction model and diagnose the results. If the hypothesis is confirmed, this work could lead to a revolution in our understanding of how predictions can be made and will also have significant national security implications since satellite and other remote sensing observations that are central to weather forecasts now can be disrupted by hostile nations.

Students: The project is open to graduate students.

Skills and Qualifications:

- Experience running computer experiment and managing data,
- Knowledge of convective scale weather or data assimilation helpful but not required.

Project 08. Decentralized Federated Learning Sensornets for Community Resilience and Actionable Science

Areas of Interest in order of relevance: Data Science, Software Engineering, Internet-of-Things, AI/ML, and Federated Learning

Description: Low-cost Internet-of-Things (IoT) technologies enable communities to co-design and deploy environmental monitoring sensornets. The NSF NCAR Open IoTwx project has lowered hardware barriers through modular, 3D-printable stations and streamlined assembly. Building on this foundation, the updated OpenIoTwx website includes an interactive wizard that guides users from a local question or concern such as extreme heat, air quality, flood risk, or wildfire preparedness through place, purpose, and resources. The process begins with structured consideration of risk, fragility, preparedness, and resilience, then produces an open, actionable community plan covering station construction, deployment, data collection, analysis, and decision support.

The ultimate goal of this wizard is to seamlessly connect communities to NSF NCAR and UCAR tools such as CREDIT for AI-based Earth system prediction, GDEX for analytics-ready data, MPAS for multi-scale atmospheric modeling, and CIRRUS for cloud-native services without requiring users to become expert modelers first. This pathway from local concern to usable plan and decision tool demonstrates how NSF NCAR can advance actionable science. It also addresses gaps in existing fragility, risk, and resilience frameworks (such as those reviewed by the UNU Centre for Policy Research), which often lack practical, community-accessible decision tools for environmental and disaster contexts.

Federated learning workflows start with federated data. While decentralization, data sovereignty, and accessibility are core to OpenIoTwx, current practice typically routes MQTT data from stations, UAVs, or mesonets to a single hosted instance. This project develops a more fully decentralized infrastructure that supports local data control by default, with optional private, public, or adaptive federated learning across networks modeled after distributed platforms but tailored for environmental sensornets. The system will be AI-ready, support mixed surface and mobile platforms (including UAVs and UGVs), enable resilient data muling during communication disruptions, and power simple decision tools such as dashboards and hazard alerts. These capabilities are particularly relevant for high-impact events like rapidly advancing wildland-urban interface fires, where observation gaps and the need for adaptive sensing are critical.

Project Goals
1. AI-ready data workflows: Review, test, and deploy distributed data systems that ingest MQTT feeds from mixed, scalable sensornets of surface stations, UAVs, and UGVs.
2. Federated learning infrastructure: Create an OpenIoTwx community template that manages data locally while enabling private, public, or adaptive federated learning across networks.

3. Mesonet integration with decision tools: Connect sensornet data to NSF AI CREDIT and other NSF NCAR tools (dashboards, hazard alerts) within the wizard-guided resilience framework.

Specific Objectives
1. Design, develop, document, and deploy a distributed infrastructure for OpenIoTwx that includes MQTT data muling, local and shared file systems, and intra-/inter-mesonet workflows suitable for edge devices and CIRRUS.
2. Integrate and document open-source-first software and hardware components for AI readiness and federated learning, supporting multi-rate LoRa/Wi-Fi and mobile platforms.
3. Develop and document a preliminary analysis of best practices in community federated learning for sensornets, including data sovereignty and privacy options.
4. Provide insight and working prototypes showing how sensornet data can feed federated learning and AI workflows, including the NSF NCAR CREDIT model, to improve simple decision tools for risk reduction and resilience planning.

Students: The project is open to graduate students.

Skills and Qualifications:

- Strong Arduino/C programming, Python / Jupyter notebooks, MQTT, and LoRaWAN
- Familiarity with distributed systems and federated learning concepts or use cases
- Experience with Raspberry Pi (optionally Jetson Nano or similar for edge computing)
- Nice to have: Big data analysis (including downscaling and unstructured point clouds), PyTorch, ROS/ROS2, TensorFlow, and HPC systems experience (e.g., CIRRUS or GDEX)

Project 09. EcoMAP: Ecological Mapping and AI-Assisted species Profiling

Areas of Interest in order of relevance: Software Engineering, Data Science, SLM, LLM

Description: Areas of Interest: Software Development, Artificial Intelligence/Machine Learning (AI/ML), Small Language Models (SLMs) with Agentic AI systems, Data Science, Geospatial analysis.

Project Description
Wildfires can significantly change ecosystems, affecting which plant species survive, recover, or become established after a fire. Understanding these changes is important for ecosystem restoration, biodiversity conservation, and land management. However, information about wildfire history, plant observations, and species characteristics is often spread across different datasets, making it difficult for researchers and land managers to explore these connections. This project aims to develop an interactive, map-based application that combines biodiversity observations from iNaturalist, a platform where people share observations of plants and animals, with wildfire history and severity data from Monitoring Trends in Burn Severity (MTBS). The project will also explore the use of artificial intelligence (AI), including small language models (SLMs), to connect observed species with existing ecological information from resources such as the Fire Effects Information System (FEIS). This could help users learn whether species are native or invasive in a particular region, how they respond to fire, and their potential role in ecosystem recovery.

The intern for this project will have the opportunity to contribute to the following areas:

- Integrate biodiversity and wildfire datasets: Use Python and geospatial tools to combine iNaturalist species observations with MTBS wildfire information.
- Develop an interactive mapping application: Create a user-friendly tool that allows users to select a wildfire location, visualize species observations before and after the fire.
- Explore AI-assisted ecological information: Test how small language models can use trusted ecological resources, such as FEIS, to provide information about species characteristics, native or invasive status, and responses to wildfire.
- Test and improve the application: Work with researchers and conservation practitioners to evaluate the tool, identify limitations in the available data.
- The intern will gain hands-on experience in Python programming, geospatial data analysis, interactive application development, and practical applications of AI in environmental science.

The project will involve collaboration between researchers at NSF NCAR and The Nature Conservancy, providing an opportunity to develop tools that connect scientific data with real-world conservation needs.

Students: The project is open to graduate students.

Skills and Qualifications:

- Experience with Python programming, including writing, modifying, testing scripts.
- Experience with High-Performance Computing (HPC) environments, including basic Linux command-line operations.
- Foundational knowledge of software development principles and programming practices.
- Familiarity with Artificial Intelligence (AI), particularly Large Language Models (LLMs) and Small Language Models (SLMs).
- Willingness to learn new tools, develop technical skills, and collaborate with an interdisciplinary team.

Project 10. Emulators and GPUs for Data Assimilation

Areas of Interest in order of relevance: Software Engineering, Machine Learning, Application Optimization/Parallelization

Description: The Data Assimilation Research Testbed (DART) is a widely used community software facility for data assimilation. One application of data assimilation is improving numerical weather prediction. To do this, DART ingests a group of model forecasts, say 80 predictions of weather in the United States, and uses statistics to combine these model forecasts with observations to produce a better estimate of the weather.

Machine learning emulators are models trained to mimic physics-based weather and climate models at a fraction of the cost. They are fast and run naturally on GPUs, which makes them attractive for producing large ensembles. However, DART was designed for CPU supercomputers, with the forecast model run as a separate program that exchanges files with the assimilation. Combining the two raises new questions: the emulator produces its forecasts on the GPU but DART expects them in files on the CPU, and once the forecasts are fast the assimilation itself may become the bottleneck.

The student will use performance profiling results, and their own experience to guide the direction of the project. Possible projects include the following, but other directions are welcome depending on the profiling results and the student's interests:

Emulator coupling. The goal is a working cycled assimilation (forecast, assimilate, repeat over many time steps) using an emulator as the forecast model, together with a measured breakdown of where the time goes. The student may explore passing the state in memory instead of through files, running the ensemble of emulators on GPUs as a batch, or enormous numbers of ensemble members with DART. The emulator and a test case will be provided by the mentors.

Assimilation in latent space. Transferring large states can be a bottleneck for high-resolution models. By equipping DART with a learned decoder, can we instead transfer latent-space representations of model states and DA increments?

Fine-tuning Emulators for Assimilation. Depending on progress and student experience, they can explore fine-tuning the emulator response to DA using DART increments. This can then be applied to other emulators of the Earth System.

GPUs in the assimilation. The student may explore offloading parts of the DART assimilation to GPUs, such as the calculation of what the model predicts an observation should be (the forward operator) or the calculation of how much each model value is adjusted (the update increments). This could use a traditional directive-based approach such as OpenACC or OpenMP offload,

modern Fortran constructs, or be a code translation with LLMs to translate DART from Fortran into a GPU-friendly language or framework. Another option is language interoperability, calling GPU code written in another language (for example using FTorch for pytorch calls within Fortran) from DART.

Work that makes DART and emulators run well together could enable much larger ensembles and faster cycling of the assimilation, and gives the student an opportunity to work at the intersection of high performance computing and machine learning.

Students: The project is open to graduate students.

Skills and Qualifications:

- Experience with Python and a compiled language (such as Fortran, C or C++, Rust).
- Familiarity with parallel programming (MPI, OpenMP, or GPU programming), machine learning concepts, and the Linux command line is helpful.

Project 11. Ensemble vs. Regional Refinement: Satellite Data Assimilation for Urban Air Quality Forecasting

Areas of Interest in order of relevance: Data Science, Geostatistics, Supercomputer Systems Operations

Description: Satellites such as TROPOMI and TEMPO now map air pollutants like carbon monoxide (CO), which traces wildfire smoke and long-range transport, and nitrogen dioxide (NO2), which points to traffic and industrial sources. NSF NCAR's Data Assimilation Research Testbed (DART) already blends TROPOMI CO and NO2 with the CESM3 global model by running an ensemble of many model copies side by side. This project will align TEMPO NO_2 with the existing TROPOMI operators in DART. It will then answer a practical question: with a fixed amount of supercomputer time, is it better to run many coarse forecasts or one detailed forecast that zooms in on cities? Both will start from the same assimilated conditions and will be evaluated over Mexico City and Atlanta. These two cities offer very different challenges. Mexico City is a high-altitude megacity in a mountain basin, where pollution builds up under the surrounding terrain. Atlanta's urban pollution mixes with large natural emissions from the surrounding forests. Both lie within TEMPO's field of view, and both are targets of upcoming air quality field campaigns.

The intern for this project will have the opportunity to contribute to the following areas:

- Harmonizing the TEMPO NO2 observation operator with the TROPOMI operators in DART.
- Running ensemble assimilation and forecast experiments with CESM3 on the Derecho supercomputer.
- Running zoomed-in MUSICA forecasts and comparing their cost and accuracy with the ensemble.
- Evaluating CO and NO2 forecasts against observations with Python tools such as MELODIES-MONET.

Students: The project is open to graduate students.

Skills and Qualifications: (next page)

Skills and Qualifications:

- Required:
  - experience programming in Python, including working with numerical arrays (for example NumPy or xarray)
  - comfort working on the Linux command line;
  - familiarity with version control using Git.
- Preferred:
  - experience reading or modifying Fortran code;
  - experience running jobs on a computing cluster or supercomputer (for example using a batch scheduler such as PBS or Slurm);
  - experience working with scientific data formats such as NetCDF or HDF5;
  - coursework or projects in statistics, numerical methods, atmospheric science, environmental science, or remote sensing.
- No prior experience with data assimilation, climate models, or atmospheric chemistry is required.
- We are looking for curiosity about how computers and satellite data can be used to understand air quality, and a willingness to learn a large scientific codebase.

Project 12. Exploring Agentic Software Engineering and Spec-Driven Development for Scientific Software

Areas of Interest in order of relevance: Software Engineering, Application Optimization/Parallelization, Visualization

Description: AI is changing how software is developed and maintained. AI coding agents can help developers understand existing code, write new features, fix bugs, and review changes. However, using these tools effectively requires careful planning, testing, and good software engineering practices.

The Technology Exploration Group (TEG) within the Intelligent Methods, Prediction, Analysis, and Community Tools (IMPACT Section) at NSF NCAR develops and maintains open-source scientific software for data analysis and visualization workflows. In this project, we want to explore how emerging AI-assisted development workflows can help teams maintain and improve scientific software.

The project will focus on emerging approaches such as spec-driven development, where clear requirements and expectations guide software implementation, along with agentic coding workflows, automated testing, and AI-assisted code review. The goal is to understand how these approaches can be used responsibly and effectively in real-world software projects. The intern for this project will have the opportunity to contribute to the following areas: Explore modern software engineering tools and practices. Investigate AI coding agents, spec-driven development, automated code review, testing, and other emerging development workflows.

Build and experiment with agentic workflows. Create a prototype development environment to experiment with different approaches. Explore how specifications, coding agents, testing, and code review can work together.

Apply these workflows to scientific software. Work with TEG software engineers to identify a suitable maintenance or improvement task in an open-source software package. Test promising approaches on a real software project.

Evaluate and share findings. Assess the benefits, challenges, and limitations of different workflows. Share demonstrations, documentation, and recommendations with the TEG team. The project will emphasize responsible AI use, software quality, testing, security, and human oversight. The student will have flexibility to explore new ideas and help shape the direction of the project.

Students: The project is open to graduate students.

Skills and Qualifications:

- We are looking for a motivated graduate student interested in modern software engineering, AI-assisted development, and open-source software.
- Graduate student in Computer Science, Software Engineering, or a related field Experience programming in at least one language, preferably Python or C++
- Familiarity with Git, GitHub, and basic software development workflows
- Understanding of software engineering practices such as debugging, testing, and code review
- Interest in AI coding assistants, large language models (LLMs), and emerging developer tools
- Strong problem-solving skills and willingness to explore unfamiliar technologies
- Good communication skills and an interest in collaborating with other developers
- Familiarity with using AI-assisted coding tools or building AI-based applications

Project 13. Exploring AI-Assisted Interfaces for Search and Discovery using Scientific Embeddings

Areas of Interest in order of relevance: Data Science, Software Engineering, Machine Learning and Artificial Intelligence

Description: Scientific datasets are growing rapidly, making it increasingly challenging for researchers to search, explore, and discover meaningful patterns in the data. Embeddings offer a promising solution by representing complex data as numerical vectors. These representations can help researchers find similar events, identify patterns, and explore large collections of weather and climate data. The Technology Exploration Group (TEG) within the Intelligent Methods, Prediction, Analysis, and Community Tools (IMPACT Section) at NSF NCAR has developed a visual analytics workbench for exploring scientific embedding datasets. Building on this work, this project will investigate new ways for researchers to interact with embeddings, formulate searches, and understand their results.

Emerging technologies such as conversational interfaces powered by LLMs, generative user interfaces, and AI-assisted workflows offer exciting possibilities for scientific data exploration. However, researchers need tools that are intuitive, transparent, and grounded in actual scientific data with reproducible workflows.

The goal of this project is to explore how these technologies can support scientific search and discovery while preserving scientific accuracy, reproducibility, and user control.

The intern for this project will have the opportunity to contribute to the following areas:

- Understand research workflows. Collaborate with researchers to learn how they explore datasets, formulate research questions, and evaluate search results. Identify opportunities where new interaction methods could improve their workflows.
- Explore emerging technologies. Investigate approaches such as natural-language search, generative UI, Model Context Protocol (MCP), and interactive visualization. Experiment with different ways to interact with embedding datasets and retrieve scientific information.
- Design and develop a prototype. Translate researchers' needs into a working application using existing embedding datasets and retrieval tools. The student will have flexibility to explore different solutions and help shape the project's technical direction.
- Evaluate and improve the experience. Work with researchers to test the prototype, gather feedback, and assess how effectively it supports scientific exploration. Investigate how users can inspect retrieved results, compare embedding models, and understand the limitations of similarity searches.

The student will work closely with TEG developers and collaborate with researchers across NSF NCAR. The project combines software engineering, human-computer interaction, scientific

visualization, and applied AI research. The project will also provide opportunities to explore software development with coding agents while following responsible engineering practices, including testing, code review, and reproducibility.

Students: The project is open to graduate students.

Skills and Qualifications:

- We are looking for a motivated graduate student interested in software engineering, applications of AI in developing interactive tools for scientific data exploration.
- Graduate student in Computer Science, Software Engineering, Data Science, or a related field.
- Programming experience in at least one language, preferably Python.
- Familiarity with vector databases and related libraries (e.g., LanceDB, FAISS etc.) and their use in storing, indexing, and retrieving embeddings for similarity search and semantic search applications.
- Familiarity with software development practices, including Git and debugging.
- Interest in large language models (LLMs), generative AI, and emerging applications of AI.
- Interest in user interface development, data visualization, or human-computer interaction.
- Strong problem-solving skills, curiosity, and willingness to explore new technologies.
- Good communication skills and interest in collaborating with scientists and software developers.
- Familiarity with embedding models, vector based semantic search.
- Experience developing interactive applications or data visualization tools is a plus.
- Familiarity with LLM APIs, AI agents, or Model Context Protocol (MCP).
- Familiarity with gathering user requirements or evaluating software usability.

Project 14. Large GPU Ensembles and Rare-Event Sampling for Weather and Climate

Areas of Interest in order of relevance: Applications of AI to weather and climate models, Data Science, Application Optimization/Parallelization

Description: Ensembles tell us how much to trust a simulation. They work well for common outcomes, but a rare event may show up in one member out of a thousand, and a typical ensemble of 50 will miss it. Larger ensembles help. With ensembles of about 1000 members, we can study the tails of weather and climate events (e.g., extreme precipitation, rapid intensification, wildfire blowups). Running that many members has been too expensive for most groups.

GPUs may change this. We now have PyTorch implementations of WRF, MPAS, and components of CESM, and their batched design lets many members advance together on one device. In this project, the student will work with CISL researchers to build a GPU workflow for ensembles of up to 1000 members, and to ask where those ensembles still fall short.

First, the student will choose a focused application and one model. Next, they will profile and improve the ensemble workflow. The questions include how to batch members on one GPU, how to spread them across devices, and how memory and output limit throughput. Every change must preserve the numerical results to the accuracy the application requires.
Finally, the student will test what happens when 1000 members are not enough. For events that are rarer still, we will try forward flux sampling (FFS), a rare-event method that splits a rare transition into stages and multiplies the stage probabilities. We have shown that FFS can recover rates from far fewer trajectories than direct sampling in a neural weather emulator (arXiv:2606.30920). The student will compare FFS to a large ensemble of equal cost in a physics-based model, and will document where each approach works.

Deliverables include a demonstrated large-ensemble experiment, measured speed and memory results, a comparison with FFS on one rare event, and a reproducible workflow for other researchers.

This project is intended for a graduate student with experience in Python and PyTorch. Background in statistics, atmospheric science, or HPC is helpful but not required. The central work is designing the experiments and establishing when each method can be trusted.

Students: The project is open to graduate students.

Skills and Qualifications:

- Proficiency with Python and Jupyter notebooks; basic UNIX/Linux proficiency.
- Prior experience with machine learning or PyTorch is desired.

- Exposure to WRF, MPAS, or CESM is a plus but not required.
- Experience with numerical simulations or agent-based models is helpful but not required.
- Interest in applying AI to environmental or sustainability problems.

Project 15. MPAS-A and Evaluating AI-assisted Coding Methods

Areas of Interest in order of relevance: Application Optimization/Parallelization, Software Engineering

Description: MPAS-Atmosphere (MPAS-A) is a global, variable-resolution atmosphere model written in Fortran and used for weather and climate research. Its dynamical core has been ported to GPUs with OpenACC directives, keeping a single source code for CPUs and GPUs. This project looks at further optimization and porting, including porting using AI tools, and covers kernel performance and possibly data management and helping evaluate alternative APIs, such as Kokkos.

The intern for this project will have the opportunity to contribute to the following areas:

1. Helping benchmark performance of the code at different resolutions and on different hardware
2. Evaluate NVIDIA tools and compare porting methods.
3. Verify correctness using the MPAS-A automated testing workflows
4. Share the results. Present benchmarks and findings in a final talk and report and, where suitable, contribute changes to MPAS-Dev/MPAS-Model.

Students: The project is open to graduate students.

Skills and Qualifications:

- Programming experience, especially in Fortran or C/C++, and the ability to navigate a large existing code base
- Experience or coursework in parallel programming; GPU experience is a plus
- Familiarity with Linux, Git
- Interest in AI coding assistants, with a careful approach to checking their output
- No atmospheric science background needed; mentors will provide the model context.

Project 16. Using agentic AI to accelerate the modernization of a large scientific codebase

Areas of Interest in order of relevance: Agentic AI, Software Engineering, Application Optimization/Parallelization

Description: Modernizing large-scale scientific applications presents significant challenges, particularly when transforming legacy code while maintaining correctness and performance. Recent advances in agentic AI offer new opportunities to automate and assist with complex software engineering tasks. Efforts are underway to use agentic AI to convert existing Fortran codebases into a C++-based productivity framework. This project offers the opportunity to explore how agentic AI can support the development and modernization of scientific software. The goal of this summer internship is to investigate and develop practical approaches for incorporating agentic AI into a code transformation workflow. Potential areas of work include evaluating AI agents for different transformation tasks, exploring the use of formal reasoning tools such as Lean theorem proving, improving AI-assisted code review, identifying performance issues, and transforming selected portions of the code. While the specific focus will depend on the state of the project at the time of the internship and available AI technologies, the overall objective of the internship will be to identify effective approaches for using agentic AI to support scientific software development.

Students: The project is open to graduate students.

Skills and Qualifications:

- Familiarity with Python, C/C++, and the Linux environment is desirable.
- Experience with Fortran, artificial intelligence, or high-performance computing is beneficial.
- Strong motivation to learn new skills, investigate open-ended problems, and work collaboratively in a research team is required.

Non-Technical Project:
CISL Outreach, Development, and Education (CODE) Intern

Areas of Interest in order of relevance: Higher Education Administration / Student Affairs, Science Education, Education Policy, Social Work, or related program.

Description: *This is a paid student intern position.* As the CODE Intern, you will provide administrative support to the SIParCS Program office and affiliated programs. You will also assist with planning and preparation for education and outreach programs to occur during the 2027 - 2028 school year. This is a full-time (40 hours/week) student intern position that runs from **May 10 - July 30, 2027**. End date may be flexible.

About the Role:
The Summer Internships in Parallel Computational Science (SIParCS) program at the NSF National Center for Atmospheric Research (NCAR) offers graduate students and undergraduates who have completed their sophomore year significant hands-on R&D opportunities in high performance computing (HPC) and related fields that use HPC for scientific discovery and modeling. This program embeds students as summer interns in the Computational and Information Systems Laboratory (CISL), an organization within NSF NCAR charged with provisioning supercomputing and data systems to the geosciences research community, as well as conducting research and development in computational science, data analysis, scientific visualization and numerical modeling. These twin roles of service and research in CISL support NSF NCAR's broad scientific mission of discovery in the atmospheric and related sciences.

The goal of the Summer Internships in Parallel Computational Sciences (SIParCS pronounced "sigh-parks") program is to make a long-term, positive impact on the quality and development of the people who will use, operate, and support 21st century supercomputers. SIParCS students serve as paid interns at NSF NCAR's Computational & Information Systems Lab (CISL) that provides supercomputing and data services for scientists studying the Sun-Earth System. You will work with a small, flexible group of committed people and enjoy working on a wide range of projects within a supportive workplace.

As the CODE Intern, you will provide administrative support to the SIParCS Program office and affiliated programs. You will also assist with planning and preparation for education and outreach programs to occur during the 2027-2028 school year.

The 2027 CODE internship is 12 weeks in duration (one week longer than the technical SIParCS summer internship). Applicants must be available to work 40 hours per week from May 10, 2027 through July 30, 2027 for participation in the program. NSF NCAR has 3 unpaid

holidays for interns during the summer internship (Memorial Day, Juneteenth, Independence Day).

Program requirements, beyond working on projects, include attending appropriate technical seminars, attending skills-enhancing workshops, preparing a poster, and giving an oral presentation of results at the end of the summer.

For Summer 2027, we are planning for an in-person internship program where interns come on-site to our Boulder, Colorado campuses and meetings may be held in person or hybridly over Zoom or Google Meet.

*Please note: the project qualifications describe the ideal skill set we look for in candidates. We encourage you to apply even if you do not possess all of the listed qualifications.

Responsibilities:

Student intern support:

- Be an active participant on the SIParCS team to provide support and mentoring for students.
- Live in the suite-style apartments with the interns, and plan and participate in after-hours team building activities.
- Keep program leadership informed of any issues that arise.
- May assist students/participants with special needs.
- May travel to assist with intern recruitment during fall months.
- Attend the Rocky Mountain Advanced Computing Consortium (RMACC) with the SIParCS program.

Summer program logistics:

- Assist with summer program support, including planning and running events. Events include orientation, professional development workshops, field trips, and other learning opportunities for interns.
- Assist with apartment move-in and move-out logistics, distributing and collecting student supplies, and coordinating with apartment administration.
- Help write and edit the SIParCS Annual Report.
- Update SIParCS program alumni tracking documents for program assessment and evaluation purposes.

General administrative support:

- Maintain program databases and update web pages.
- Edit and compile information for documents, posters, brochures, and newsletters.
- Provide multimedia support at program events (photos, video) and keep inventory of program pictures.
- Provide meeting support and prepare meeting materials.

School-year program support:

- Update presentation slides and other educational materials for school-year programs.
- May write and/or edit educational materials such as text updates for Meteo-AR.
- May test and evaluate new educational materials.
- May provide input to the new exhibit space at NWSC

The above statements describe the primary work being performed. Additional tasks and professional development opportunities may be assigned according to the intern's skills and interests.

Required Education and Experience:

- Currently enrolled in a Master's-level program in Higher Education Administration/Student Affairs, Science Education, Education Policy, Social Work, or a related program.
- Experience working with undergraduate and graduate students.

Other Relevant Requirements:

- Must have attained a bachelor's degree at the time of application.
- Must currently be enrolled in a U.S. university.
- Must be a graduate or doctoral student.
- Must be authorized to work in the U.S. to be eligible for the SIParCS program.
    - For F visa students accepted to the SIParCS program, CPT work authorization issued through a U.S. Designated School Official (DSO) or OPT issued through both a DSO and the U.S. Citizenship and Immigration Services (USCIS) will be required.
- On-site participation in Boulder, Colorado.

Knowledge, Skills, and Abilities:

- Ability to work 40 hours/week from May 10 - July 30, 2027 (dates to be confirmed). End date may be flexible.
    - NSF NCAR has 3 unpaid holidays for interns during the summer internship (Memorial Day, Juneteenth, 4th of July).
- Experience with word processing, database, and spreadsheet applications (Microsoft Word, PowerPoint, Excel, Access, Google Meet, Zoom) as well as a Google Apps environment (Sheets, Docs, etc).
- Proven ability to organize, prioritize, and follow through on multiple tasks, with close attention to detail.
- Good written and verbal communication skills with the ability to convey information to interns, visitors, and partners in a welcoming and professional manner.
- Interest and sensitivity in working in settings that include interns from a broad spectrum of identities and communities, including those from rural and urban areas.
- Proven ability to work in a team and individually. A strong work ethic.

- Ability to work with a broad group of peers.
- Ability to interact with mentors and peers in a manner that supports collaboration and inquiry.
- Discretion in handling confidential information.
- Patience and adaptability.
- Ability and willingness to learn and use scientific computing tools and programs.
- Basic problem-solving skills.
- Ability to analyze data and draw conclusions.
- Ability and willingness to work within guidelines and policies of the organization and assigned work groups.

More information: cisl.ucar.edu/outreach/internships/code