2026 Summer Internships in Parallel Computational Science (SIParCS) Project Descriptions

Undergraduate students

Project 1. Natural Language Discovery of NSF NCAR Scientific Data
Project 2. AI Weather and Atmospheric Chemistry Prediction with Physical Constraints
Project 3. FastOSSE: A new tool for optimizing ocean observing networks
Project 4. Data Pathfinders: A GDEX UX/UI Intern Quest
Project 5. Learning to Fight Wildfires: Reinforcement Learning for Fire Suppression in a Forest Fire Model
Project 6. AI Nowcasting Models for Predicting the Evolution of Convective Storms
Project 7. CIRRUS - Developing Workflows for Validating Cloud Native Deployments

Graduate students

Project 7. CIRRUS - Developing Workflows for Validating Cloud Native Deployments
Project 8. Simulating atmospheric chemistry with MUSICA in Julia
Project 9. Improving the code infrastructure of the NCAR air quality sampling drone system
Project 10. Improving Uncertainty Estimates in Earth System Prediction with DART
Project 11. OpenIoTwx Dynamic AI-Mesonet, Edge Computing, and Cyber Infrastructure Integration
Project 12. Generalized framework for the evaluation and comparison of atmospheric chemistry models with observations
Project 13. Developing a Mini-App with the JAX Python Library and/or Rust Programming Language

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

Non-Technical Project

Graduate

CISL Outreach Development Education (CODE) Intern

Project 1. Natural Language Discovery of NSF NCAR Scientific Data

Areas of Interest in order of relevance: Software Engineering

Description: This project is Artificial Intelligence (AI) focused and will use Large Language Models (LLMs) with Agentic AI systems.

The NSF National Center for Atmospheric Research (NSF NCAR) produces a significant amount of scientific data, including climate model output, weather-related data patterns, and data generated by research on atmospheric and climate phenomena. This data is then made available to the public for research purposes, but has been historically difficult to search due to its large volume and diverse datasets.

We have developed an LLM based search tool and would like to enhance the search functionality and provide better responses to end-user queries. We would like to experiment with adding an Agentic AI system to the existing LLM solution to see if it can help answer end-user queries better.

While this project is AI/LLM based, we will be using professional software development practices to build our Agentic AI and metadata workflows. This project will be using the following technologies: Docker, Kubernetes, LLMs (most likely Ollama), Vectorized Data, most likely Java, Python, and Git.

Students: The project is open to undergraduate students.

Skills and Qualifications:

- Ability to interact with mentors and peers in a friendly, professional manner that supports collaboration and inquiry.
- Good problem solving skills.
- Good oral and written communication skills.
- Willingness to learn and use software development tools and programs.
- Curiosity to explore new things.
- Basic understanding of software development programming.
- Basic experience in languages such as Java and Python.
- Basic experience in markup languages such as xml and json.
- Basic understanding of Artificial Intelligence (AI) especially in regards to Large Language Models (LLMs).

Project 2. AI Weather and Atmospheric Chemistry Prediction with Physical Constraints

Areas of Interest in order of relevance: Artificial Intelligence/Machine Learning (AI/ML), Data Science, Numerical Methods

Description: AI weather prediction models have significantly advanced in accuracy in recent years to be competitive with physics-based weather models in terms of overall accuracy. However, the current state-of-the-art AI weather models have limited representation of certain critical physical processes, such as radiation/photolysis, convection, and atmospheric chemistry. NSF NCAR is building new AI weather and air quality model training datasets using our community modeling platforms. These include i) global and regional simulations with coupled physics and chemistry processes, and with hourly outputs of key model variables, ii) as well as datasets for individual processes such as spectrally-resolved actinic fluxes and photolysis rate coefficients calculations using the NSF NCAR TUV (Tropospheric Ultraviolet-Visible) radiative transfer model. TUV was chosen as a prime candidate for AI emulation as the radiative transfer calculation adds 25% to the overall cost of an atmospheric chemistry-climate model. These datasets will be used to train new emulators with NSF NCAR's Community Research Earth Digital Intelligence Twin (CREDIT, miles.ucar.edu/software/credit/) AI weather prediction model platform with inputs, architectures, and physical constraints to improve representation of these processes.

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

1. Improve CREDIT: evaluate new architectures trained on this data in the CREDIT platform for both accuracy and physical consistency. Based on their evaluation, the intern may retrain or fine-tune the models to improve their performance further. The intern will gain hands-on experience with the CREDIT platform and NSF NCAR high performance computing systems and work with a multi-disciplinary team tackling cutting edge challenges spanning both AI and Earth system science.

2. Build a TUV radiative transfer emulator: contribute to the development of an AI emulator for photolysis calculations that can predict spectrally-resolved actinic fluxes and photolysis rates for over 50 tropospheric and stratospheric chemical species from atmospheric state variables (e.g., cloud properties, aerosol loadings). Unlike training on complete model outputs, this project focuses on emulating a single physical process component, presenting unique challenges in maintaining physical consistency and coupling with other model components. Depending on progress made during the summer, the intern may be able to test integration with both the CREDIT AI platform and physics-based Earth system models.

Students: The project is open to undergraduate students.

Skills and Qualifications:

- Classroom or personal experience with the scientific Python stack (e.g, numpy, pandas, matplotlib, Jupyter, xarray) required.
- Familiarity with a machine learning (scikit-learn) or deep learning framework (e.g., PyTorch, Keras, Tensorflow) required.
- Completion of introductory statistics or data science/analysis course preferred.
- Some experience with geospatial data analysis, especially in an Earth science, atmospheric science, or meteorology context, is preferred.

Project 3. FastOSSE: A new tool for optimizing ocean observing networks

Areas of Interest in order of relevance: Data assimilation and observing system design, Numerical optimization, Artificial Intelligence and Machine Learning (AI / ML)

Description: Earth's ocean is notoriously difficult to observe. It is opaque to the electromagnetic spectrum beyond a few meters, and in situ instruments must contend with rough seas, corrosion, biofouling, and piracy. As such, there is great demand for approaches that optimize the placement of costly new data sources. OSSEs (observing system simulation experiments) use numerical ocean models and data assimilation (DA) to identify which observations will best constrain ocean processes. However, the high technical overhead of OSSEs has prevented their widespread adoption. This project aims to make OSSEs practical to use for a range of ocean observing problems. It leverages recent developments in Jupyter-based regional ocean model deployment by the CROCODILE project at NSF NCAR (github.com/CROCODILE-CESM) using the MOM6 ocean model in the Community Earth System Model as well as ocean observations and DA in the Data Assimilation Research Testbed (DART; dart.ucar.edu).

In this project, the student will use Python capabilities in pyDARTdiags (github.com/NCAR/pyDARTdiags) to modify ocean observing datasets (as pandas dataframes) to include hypothetical observations, focusing on upper-ocean temperature. These observations will be assimilated using DART to evaluate how well a proposed observing system constrains ocean temperature evolution on time scales of days to weeks. Next, the student will lead construction of a Jupyter-based GUI interface allowing users to generate new observation dataframes by interactively specifying observation locations and types. We are looking for a motivated candidate who can connect across teams with diverse expertise and who can take advantage of opportunities to shape the directions of the work.

Students: The project is open to undergraduate students.

Skills and Qualifications:

- Python programming experience
- Willingness to collaborate with a team
- Written and verbal communication skills
- Enthusiasm for learning about physical modeling and oceanography

Project 4. Data Pathfinders: A GDEX UX/UI Intern Quest

Areas of Interest in order of relevance: Digital Asset Management, UX/UI (user experience and user interface) Design, Software Engineering

Description: Improving the scientific user journey from landing page to a data collection is essential to ensuring that Earth System Science and interdisciplinary researchers can find, access, and use data effectively. Within the NSF NCAR Geoscience Data Exchange (GDEX) web data portal, this challenge is especially critical due to the wide range of skills, backgrounds, and goals across our incredible user community. A clear, intuitive design is needed to help users succeed in reaching their scientific objectives.

This internship will focus on translating the results and recommendations from existing usability study outputs into actionable designs. The intern will review the prior ideas, evaluate their potential, and create mock-ups and prototypes that address user needs. Using common UI design and collaboration tools (Adobe XD, Figma, Sketch, Zeplin, Miro, etc.), the intern will design concepts, refine them with the team, and produce draft solutions that can be integrated into the GDEX platform.

The ultimate goal is to deliver practical, user-centered improvements to GDEX, with the opportunity for designs to transition from concept to implementation, allowing for real-world experience to be added to a design portfolio. This role offers valuable experience in applying user research to real-world design challenges, collaborating with a cross-functional team, and making direct contributions to the usability and impact of a foundational data platform.

Students: The project is open to undergraduate students.

Skills and Qualifications:

- Creative approach to problem solving as well as a willingness to learn.
- Knowledge and experience in design, UX, and UI, specifically on web platforms.
- Moderate proficiency with basic web design technologies (HTML, CSS, Bootstrap, JavaScript, etc.)
- Experience using one of the common UI design and collaboration tools (Adobe XD, Figma, Sketch, Zeplin, Miro, etc.)
- Familiarity with best practices in accessibility design and Web Content Accessibility Guidelines (WCAG)
- Ability to collaborate effectively with a team, including other interns, data analysts/engineers/scientists, and researchers.
- Familiarity with usability studies is a plus
- Basic knowledge of various graphic design tools (Adobe Photoshop, Illustrator, etc.) is a plus

Project 5. Learning to Fight Wildfires: Reinforcement Learning for Fire Suppression in a Forest Fire Model

Areas of Interest in order of relevance: Artificial Intelligence/Machine Learning (AI/ML) -- Reinforcement learning, Forest fire management, Data Science

Description: Think of forest fire management like a video game: the goal is to get really good at protecting the health of the forest by stopping damaging fires while still allowing the forest to regrow and thrive over time. This can be accomplished with reinforcement learning (RL), wherein a decision-making algorithm called an “agent” learns through trial and error to balance fire suppression with forest regeneration. In this project, we train this RL agent to play the forest management "game" inside a simulation called the Forest Fire Model (FFM).
The simulation represents a grid of trees that grow, burn, and regrow over time. At each step, the agent observes the current state of the forest and decides where to allocate firefighting resources. It earns points (rewards) based on how well it preserves the long-term health of the forest. The challenge is learning policies that balance immediate fire suppression with forest ecology, sometimes letting small fires burn can actually be better for the ecosystem than suppressing everything aggressively.

Over the summer, the successful applicant will work with researchers in CISL to design and train RL agents using PyTorch and Gym. You'll experiment with different neural network architectures (such as convolutional or transformer-based models), test various reward schemes that encourage both effectiveness and efficiency, and analyze how different policy designs lead to emergent firefighting strategies. You'll gain hands-on experience with the full RL pipeline: environment design, agent training, hyperparameter tuning, and results visualization. The baseline model uses probabilistic fire spread between adjacent cells, and if time allows, the project may explore more realistic challenges, such as incorporating limited firefighting resources (budgets, crew availability), dynamic weather effects (wind direction and speed that make spread directional rather than uniform), or multiple concurrent fires. Students interested in environmental science, AI/ML, or climate applications are especially encouraged to apply.
This project is developed within CISL and is intended for undergraduate students.

Students: The project is open to undergraduate 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 reinforcement learning is a plus but not required; interest in learning RL during the project is important.
- Experience with numerical simulations or agent-based models is helpful but not required.
- Interest in applying AI to environmental or sustainability problems.

Project 6. AI Nowcasting Models for Predicting the Evolution of Convective Storms

Areas of Interest in order of relevance: Artificial Intelligence/Machine Learning (AI/ML), Data Science, Visualization

Description: Improved short-range nowcasting of convective storms, especially their initiation, growth, and decay, can provide critical lead time for severe thunderstorm, tornado, and flash flood warnings. NSF NCAR is collaborating with Penn State University (PSU) on a project to develop an ensemble nowcasting model that incorporates satellite, radar, and numerical weather prediction model output to predict the evolution of convective storms with a focus on improving the uncertainties around convective initiation. The project leverages NSF NCAR's Community Research Earth Digital Intelligence Twin (CREDIT, miles.ucar.edu/software/credit/) platform to scale training, inference, and uncertainty quantification across NSF NCAR's HPC systems.

The intern for this project will have the opportunity to analyze and evaluate the first generation of nowcasting models produced by the project for accuracy and physical consistency with observations. Based on their analysis, the intern may fine-tune or retrain the model. The analysis will involve using ensemble verification statistics, physically-based explainable AI techniques, and innovative AI methods for evaluating the sharpness and perceptual quality of predictions. The intern will work with the NSF NCAR and PSU team.

Students: The project is open to undergraduate students.

Skills and Qualifications:

- Classroom or personal experience with the scientific Python stack (e.g, numpy, pandas, matplotlib, Jupyter, xarray) required.
- Familiarity with a machine learning (scikit-learn) or deep learning framework (e.g., PyTorch, Keras, Tensorflow) preferred.
- Completion of introductory statistics or data science/analysis course preferred.
- Some experience with geospatial data analysis, especially in an Earth science, atmospheric science, or meteorology context, is preferred.

Project 7. CIRRUS - Developing Workflows for Validating Cloud Native Deployments

Areas of Interest in order of relevance: Cloud Native Workflows, GitOps Workflow Optimization, Containers

Description: NSF NCAR has recently deployed internal cloud infrastructure to promote the development and migration to cloud native application development and deployment methodologies. There are many research teams within NSF NCAR that have been migrating to this platform and are bringing with them different development practices and varying levels of experience with cloud native methodologies. This project sits at the intersection of atmospheric science, computer science, and cloud native technologies and will provide insight into how scientific research is evolving with advances in computer science.

This position will look at how to better standardize workflows and testing in a shared Kubernetes environment. This will be completed by utilizing multiple tools such as git, GitHub workflows, Helm, Kustomize, Kyverno, and possibly others. The primary objective will be to introduce pre-commit git hooks that can test & validate kustomizations and helm charts, create reusable GitHub workflows for multiple teams to standardize upon, and utilize kyverno policies to better enforce best practices upon users that are new to cloud native environments. This project will provide meaningful hands-on experience with widely used tools and the opportunity to see how atmospheric research is being supported by cloud native technologies.

Students: The project is open to undergraduate and graduate students.

Skills and Qualifications:

- Basic knowledge of git and Linux environments
- Familiarity with container technologies and tooling
- Basic knowledge of Kubernetes resources and yaml

Project 8. Simulating atmospheric chemistry with MUSICA in Julia

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

Description: The Multiscale Interface for Chemistry and Aerosols (MUSICA, www2.acom.ucar.edu/sections/multi-scale-infrastructure-chemistry-modeling-musica) is a community modeling infrastructure developed by NSF NCAR's Atmospheric Chemistry Observations and Modeling (ACOM) Laboratory with the broader atmospheric chemistry community. It enables interchangeable chemistry components across models, allowing fair, consistent comparisons of mechanisms and solvers. MUSICA includes numerical packages for gas-phase kinetics and photolysis, forming model-independent software for atmospheric chemistry research. It supports Fortran, C, C++, CUDA, Python, and JavaScript.

Caltech, MIT, and NASA's Jet Propulsion Laboratory have developed a next-generation Earth System Model (ESM) that improves on existing climate models by automatically learning from observations and targeted high-resolution simulations, reducing uncertainty, and being more flexible and scalable (CliMA, clima.caltech.edu). In particular, this model is implemented in Julia due to its high performance, high-level syntax, Just-in-time (JIT) compilation, multiple dispatch, GPU support and automatic differentiation. As of 2025, any chemistry and aerosol interactions in CliMA are prescribed, and the scientific developers from that group are interested in bringing in external chemistry modules that can be integrated into CliMA.

Therefore, in this project, we are seeking to hire a highly motivated intern who can create a Julia wrapper of the MUSICA package to enable gas-phase chemistry in CliMA. The key deliverables for this project could be 1) a wrapper of MUSICA to expose basic gas-phase chemistry solving options to Julia; 2) the integration of this package into the Julia model for basic gas-phase chemistry; 3) the collection, benchmarking, and analysis of some short global runs of CliMA using the new Julia package with chemistry. Ideal candidates should have a background in computer science, environmental science, or a related field. Programming skill in a scripting language such as Python or Julia is required, and skills in a compiled or JIT compiled language such as Fortran, C, C++, Java, or C# is required. The ability to work efficiently and collaboratively in a team is strongly preferred. Familiarity with Linux is encouraged. Experience with high performance computing will be advantageous but not required.

Students: The project is open to graduate students.

Skills and Qualifications:

- Ideal candidates should have a background in computer science, environmental science, or a related field.
- Programming skill in a scripting language such as Python or Julia is required
- Skills in a compiled or JIT compiled language such as Fortran, C, C++, Java, or C# is required.
- The ability to work efficiently and collaboratively in a team is strongly preferred.
- Familiarity with Linux is encouraged.
- Experience with high performance computing will be advantageous but not required.

Project 9. Improving the code infrastructure of the NSF NCAR air quality sampling drone system

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

Description: The Whole Air Sampling Pilotless Platform (WASPP, www2.acom.ucar.edu/facility/waspp) is an advanced drone-based system developed by NSF NCAR to study air quality and atmospheric chemistry, especially in the lower part of the atmosphere known as the boundary layer. It allows scientists to collect air samples and weather data in areas that are typically hard to reach using traditional methods like towers or crewed aircraft. The WASPP instrument payload weighs 6.0 kg and includes an array of 8–15 100 mL canisters that are mounted on a commercial multirotor drone. During each flight, WASPP can collect 8–15 air samples in pressurized canisters. These samples are later analyzed in the lab using the state-of-the-art TOGA-TOF instrument to detect more than 100 volatile organic compounds (VOCs), ranging from extremely low to moderately high concentrations.

The WASPP measurements and sample collection are controlled and recorded at a high resolution using a Raspberry Pi system and a custom electronic board though Python scripts. However, the WASPP system does not communicate with the ground when deployed. Rather, it is currently designed to begin sampling after a time period (i.e., when a specific altitude is reached), making it challenging for efficient sampling. Therefore, better communication with the ground and utilizing the real-time coordinates system will enhance the scientific capabilities of this WASPP.

This project will provide the student with the opportunity to work on the implementation of this capability using the Python programming packages and cloud-based APIs. Some familiarity with these packages is expected, but the project provides abundant chances to build advanced skills. The student will also gain hands-on experience with operating the WASPP system, and analyzing airborne measurements.

Students: The project is open to graduate students.

Skills and Qualifications:
* Intermediate programming skill in Python is required including knowledge of the following topics: Numpy, Pandas, GPIO, SMBus
* Familiarity with IoT (Internet of Things) systems and sensors
* Familiarity with air quality measurements is encouraged but not required.
* The ability to work efficiently and collaboratively in a team is strongly preferred.

Project 10. Improving Uncertainty Estimates in Earth System Prediction with DART

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

Description: Ensemble data assimilation (DA) systems combine computer models with real-world observations to estimate the state of the Earth system and predict its future evolution. A key challenge is accurately representing uncertainty: models often underestimate their own error. To correct this, DA systems use inflation; a technique that increases the ensemble's spread to better match observations.

The Data Assimilation Research Testbed (DART; dart.ucar.edu) is an NSF NCAR community software facility that supports a wide range of Earth system models. DART includes advanced, adaptive inflation methods that adjust uncertainty estimates over time and space. However, these methods can struggle when observation networks are temporally heterogeneous. For example, land-based sensors often go offline in winter due to snow cover and satellite instruments generally sample different atmospheric layers at different frequencies.

This project will explore and improve DART's inflation capabilities in such real-world, time-varying observation networks. The student will investigate how current inflation strategies behave under changing data availability, develop and test targeted inflation techniques that adapt locally to data-rich regions while avoiding unnecessary spread elsewhere, and analyze the results in both Bayesian and heuristic (non-Bayesian) frameworks. Work will begin with simple low-order models and, if time permits, scale to larger atmospheric or land-surface models.

Through this work, the student will gain hands-on experience with uncertainty quantification and ensemble DA, learn to run and modify scientific software (Fortran/Python), visualize and interpret large datasets, and work in a collaborative, high-performance computing (HPC) environment. This project is well suited for a graduate student interested in data assimilation, numerical modeling, and scientific software development.

Students: The project is open to graduate students.

Skills and Qualifications:

- Background in computer science, applied mathematics, geoscience, or related fields.
- Familiarity with coding (e.g., Fortran, Python, or similar languages).
- Curiosity about Earth system modeling and prediction.
- Good oral and written communication skills.
- Ability to work in a collaborative, diverse environment

Project 11. OpenIoTwx Dynamic AI-Mesonet, Edge Computing, and Cyber Infrastructure Integration

Areas of Interest in order of relevance: Software Engineering, Internet-of-Things, Data Science

Description: Low cost Internet-of-things (IoT) technologies have the potential to increase accessibility of communities to co-design and deploy sensornets of environmental monitoring equipment. The open IoTwx project uses low cost IoT based electronics and 3D printed parts to create accessible open source instrumentation. openIoTwx can operate in both “small (low bandwidth) data” and “big data” configurations. Small and low bandwidth data configurations include standard measurement nodes such as digital rain, wind, and air (T,P,RH, air quality, etc).Big data modes may include high density, high frequency data such as those produced by a LIDAR (Laser Imaging, Detection, and Ranging) instrument, which could generate over 100,000 cloud point data per second or data from ultra high-resolution video or images from super-HD cameras.

The goal of this project is to expand on prior progress to IoT edge computing and cyber-infrastructure integration within CISL’s cloud computing platform. Thus far, a small atmospheric sensornets has been deployed in Boulder and the next goals we have are aimed at direct improvements and enhancements to that. Our goals are 1) to explore how to harden and improve onboarding, deployment, maintenance and scalability of the current open source containers for new network operators, 2) improve existing automation workflow and pipelines to accommodate the current toolchains (e.g. AI and agentic automation tools) to support user assignment, permissions and dashboards development, and 3) expand the sensornet architecture to include integration of short term, rapid deployed and mobile instrumentation such as UAVs, drones, and backpack systems.

Students: The project is open to graduate students.

Skills and Qualifications:
- Strong Arduino, Python, MQTT, LoRa/LoRaWAN, RS485 protocols, IPFS, Jupyter Notebooks experience
- Containerization (Docker) of microservices architectures, deployment on Kubernetes clusters (e.g., CIRRUS), and integration of AI models/LLMs (e.g., NSF NCAR’s MILES (Community Research Earth Digital Intelligence Twin (CREDIT))
- Real-time data pipelines with MQTT brokers, PostgreSQL databases, and visualizations using Leaflet.js and Metabase dashboards
- Raspberry Pi (or similar SBC platforms) or Robotic/UAV platforms for edge computing algorithms would be helpful
- Big data analysis including downscaling, unstructured point cloud data streams, filtering and data transforms, image analysis, tensorflow, and experience with HPC systems.

Project 12. Generalized framework for the evaluation and comparison of atmospheric chemistry models with observations

Areas of Interest in order of relevance: Data Science, Software Engineering, 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 models 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 against a large and growing suite of observations. Specifically, MELODIES MONET aims to provide a framework for evaluating research, operational, and regulatory models against a variety of observations within a common framework. A number of observational datasets and model formats are currently available in MELODIES MONET. Some examples include EPA AirNow and AERONET ground based measurements, MODIS and TEMPO satellite data, and NCAR CESM and NOAA UFS-AQM models.

MELODIES MONET framework provides the configuration file and driver to perform the desired model-observation comparison by calling MONET I/O for each model and observation, and then running MONET to perform the pairing, followed by plotting and statistics calculations. Currently, the MONET can only work with regular meshes such as WRF-chem and CESM finite volume dynamical core and thus cannot handle the unstructured mesh such as the one provided by FV3, CESM-SE-RR, CESM-MPAS and MPAS models. The MONET I/O software heavily uses XARRAY and regridding tools for interpolation and conservative regridding for comparison of models with observations, especially satellite data.

The goal of this project is to generalize the model readers so that any grid can be ingested without prior processing, with an initial focus on popular community models such as CESM, MPAS of NSF NCAR but also NOAA FV3. The main task will be to modify the XARRAY commands to UXARRAY.

The ACOM mentor will provide test case examples and expertise on the various models so that the student does not need to have specific knowledge about chemistry. A good knowledge of python and related tools (jupyterhub, github) is required for this project to succeed. We also work in close collaboration with NOAA CSL in Boulder.

Students: The project is open to graduate students.

Skills and Qualifications:

- Intermediate programming skill in Python is required including knowledge of the following topics: Xarray, UXarray, Matplotlib, Dask
- 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 13. Developing a Mini-App with the JAX Python Library and/or Rust Programming Language

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

Description: Recent trends in the architecture of high-performance computing systems have made effectively utilizing graphics processing units (GPUs) crucial for the applications that drive scientific discovery today. In response, a plethora of software packages and programming models have been developed to enable writing scientific applications that are performant on both CPUs and GPUs. NCAR has developed a benchmark mini-application that solves a simplified set of atmospheric modeling equations. This mini-app has been used to assess the advantages and disadvantages of several popular programming languages and scientific computing libraries, including Python/NumPy, Julia, C/C++, and Fortran.

This internship will focus on developing new versions of the mini-app, written using either the JAX Python library or Rust. The JAX Python library enables the creation of array computations designed for high-performance numerical computing and large-scale machine learning. Rust is a newer systems programming language that enables safer scientific computing practices without compromising computational performance. Rust combines memory safety and efficiency with Python-like packaging, making it a promising language for scientific applications.

In both cases, students will explore the accuracy and computational performance of the newly added mini-app versions. This part of the project will provide students with the opportunity to explore how performance varies between programming languages as well as between hardware types (ex., CPU vs GPU). This project will offer hands-on experience in scientific programming, numerical modeling, and working with advanced scientific programming tools like the JAX Python package and Rust.

Students: The project is open to graduate students.

Skills and Qualifications:

- Familiarity with Python, and Python-based scientific computing libraries (like NumPy, SiPy, or JAX) or familiarity with Rust and Rust packages in a Linux environment.
- Strong motivation to learn new skills and resolve issues in a team is required.
- Experience with computer architecture and high-performance computing is highly desirable.
- Currently enrolled in a program in Applied Mathematics, Physics, Computer Science, or Engineering.

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 2026 - 2027 school year. This is a full-time (40 hours/week) student intern position that runs from **May 11 - July 31, 2026**. End date may be flexible.

Responsibilities
Student intern support:

- Be an active participant on the SIParCS team to provide support and mentoring for students.
- Live-in at 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 specific requests or concerns.
- 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 SIParCS Annual Report.
- Update SIParCS program alumni tracking documents for program assessment and evaluation purposes.

General administrative support:

- Maintain program databases and updating 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
- May test and evaluate new educational materials.
- The above statements describe the primary work being performed. Additional tasks and professional development opportunities may be assigned according to the intern's skill set and interests.

Students: The project is open to graduate students in Higher Education Administration / Student Affairs, Science Education, Education Policy, Social Work, or related program.

Skills and Qualifications:
What you need:

- Must have attained a bachelor's degree at the time of application.
- Must be enrolled in a graduate level college, university or accredited professional program during the normal school year.
- Ability to work 40 hours/week from **May 11 - July 31, 2026** (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) 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.
- Patience and adaptability.
- Discretion in handling confidential information.
- On-site participation in Boulder.

Preferred:

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

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