ATOC 4815/5815-001, Spring 2026

Scientific Programming, Data Analysis and Visualization

Laboratory

Instructor: Will Chapman, SEEC N258
Email: wchapman@colorado.edu

Teaching Assistant: Aiden Pape
Email: aiden.pape@colorado.edu

Class: Tuesdays and Thursdays 10:00 am–11:15 am, SEEC N126

Office hours (instructor): Tuesdays/Thursdays 11:15 am–12:15 pm, **Aerospace Café**;
Office hours (TA): Monday / Wednesdday 3:30 am–4:30 pm, **DUAN D319**;

Zoom: `https://cuboulder.zoom.us/my/weccu`

Summary

This course introduces programming and analysis techniques to process, interpret, and visualize data sets that are commonly used in the atmospheric and oceanic sciences. The main programming language is python, which will be taught in the lab component, along with the basics of Linux operating systems. Applications include accessing, reading, and mapping of satellite, ground-based and aircraft Earth observations, as well as correlative/spectral data analysis, functional fitting, and data aggregation from the pixel level to global climatologies. The course will also touch on approaches for merging data sets from sources as different as reanalysis and satellite observations. A final project, involving the independent analysis and visualization of a scientific data set, integrates skills acquired throughout the course.

Prerequisites: ATOC 1050 (or equivalent). Basic programming skills such as CSCI-1300 or equivalent, and some knowledge of calculus and algebra are desired. The course fulfills 3 credits of the ATOC minor.

Reading / Teaching Material

The course draws from multiple textbooks and online material, which is / will be posted on Canvas. Here are the primary resources:

- Adapted version of "How to think like a computer scientist, interactive edition"
  `https://runestone.academy/ns/books/published/ATOC4815_2024/index.html`
- John R. Taylor: *An Introduction to Error Analysis* (on Canvas; fundamentals of statistics, curve fitting, hypothesis testing, spectral analysis, . . .)
- `https://www.datacamp.com/`

- Johnny Wei-Bing Lin: *A Hands-On Introduction to Using Python in the Atmospheric and Oceanic Sciences* (`http://www.johnny-lin.com/pyintro/`)

Programming Language and Environment

The primary programming language taught in this course is python. However, we will very briefly touch on NCL (see also NCL/python cross-reference on Canvas). We will also briefly look at Fortran because they remain indispensable programming languages in the geosciences.

Please bring your personal laptops, and download "anaconda" at `https://www.anaconda.com/download/` with python 3. If you do not have access to a personal laptop, please let me know as early as possible during the semester.

Course Website

Canvas (updated throughout semester).

Class Philosophy and Structure

There are essentially two parts to this course:

1. **"Nuts and bolts" of learning the python language:** As with any language, the best learning approaches can vary from person to person, but they all have in common that mastery comes with practice. For this reason, a significant part of the "memorization" work (learning by doing) should be done at home. Of course, we will introduce and discuss programming principles in class during the lab part (more than 45 minutes) of each class. We will also work on specific science projects that require programming.

2. **"Methods":** In the lecture part of the class (less than 30 minutes), we will get to know basic concepts of scientific data analysis / time series / correlative analysis, functional fitting / hypothesis testing, statistics and error propagation, image processing and mapping / visualization / re-projection (think: NASA WorldView), data aggregation and combination etc. Usually, the lecture part will be followed by hands-on application of the new concepts during the lab part.

Performance Elements and Grading

Homework	25%
Attendance*	10%
In-class tests (on reading / online assignments)	10%
Midterm exams (2)	30%
Project (groups of 2)	25%

*Missing more than 4 lectures [unexcussed] increments this 10% down by 3% each time

Principal Topics Covered

- Programming fundamentals in python for scientific computing (functions, modules, control flow, debugging, and code style)
- Numerical and tabular data handling with numpy and pandas (arrays, masking, time series, resampling, aggregation)
- Multidimensional geoscience data with xarray and common file formats (netCDF, GRIB at a conceptual level, Zarr)
- Reproducible workflows: environments (conda), project structure, simple command line tools, and automation of analyses
- Scientific visualization and figure design for geoscience data using matplotlib (time series, histograms, cross sections, multi-panel figures)
- Maps and geographic projections for atmospheric and oceanic fields using cartopy or similar tools
- Basic statistics and time series concepts for weather and climate (anomalies, climatologies, simple correlations)
- Code quality and collaboration: testing, debugging, profiling, and version control with git and GitHub
- Communicating computational results in notebooks, short reports, and brief presentations
- End-to-end analysis and visualization of a real atmospheric or oceanic data set in a final project

Student Learning Outcomes

- Intermediate working knowledge of python, anaconda, Jupyter, GitHub, VScode
- Accessing and working with various kinds of data formats and applications (including satellite imagery, meteorological or geographical data)
- Curve fitting concepts and visualization skills for multi-dimensional data sets
- Simple image and statistical data analysis
- Independent and collaborative work on a project involving a scientific data set

Key Dates (Spring 2026, Tentative)

These dates are subject to change as we progress (but hopefully they will not).

- **Mid-term 1:** Feb 03, 2026

- **Mid-term 2:** March 12, 2026
- **Final Project due:** Monday, April 27 [11:59pm].

Tentative Schedule

The schedule will be adjusted depending on students' needs and progress.

Week	Methods (lecture)	Programming (lab)
1	Course overview; reproducible workflows; python refresher	Environment setup; conda; Jupyter; simple analysis in a first notebook
2	Numpy foundations; numerical thinking with arrays	Array operations, indexing, broadcasting, masking; basic matplotlib plots
3	Tabular data and time series concepts	pandas DataFrames; time handling and resampling; **In-class Test 1** (python, numpy, basic plotting)
4	Multidimensional fields and gridded data	xarray basics: DataArray, Dataset, subsetting, simple climatologies and composites
5	Data formats and reproducible project structure	netCDF I/O; organizing projects (src, data, notebooks, figs); simple command line scripts
6	Map projections and geoscience visualization	Cartopy maps; projections, coastlines, land/sea masks; multi-panel map figures; **Midterm 1** (end of week)
7	Visualization design and colormaps; common pitfalls	Figure critique and redesign; improving colormaps, layout, and labels; **In-class Test 2** (visualization concepts)
8	Code quality: testing, debugging, performance basics	Using debuggers; writing simple tests (pytest); basic profiling and vectorization
9	Version control, collaboration, and open science	git and GitHub workflows; branches, pull requests, code review; final project proposal due
10	Advanced visualization: cross sections, animations, interactivity	Cross sections with xarray; simple animations in matplotlib; brief intro to interactivity (ipywidgets or similar)
11	Integrated workflows and case study	End-to-end pipeline from raw data to figures; **Midterm 2** (integrated analysis and visualization)
12	Communicating results: notebooks, reports, and figure captions	Narrative notebooks or short reports; polishing figures; **In-class Test 3** (interpretation and critique of plots)
13	Project workshopping and peer feedback	Project check-in; peer review of figures and methods; in-class project work
14	Reproducibility and polishing results	Full rerun of analyses; cleaning repositories; refining final figures and documentation
15	Synthesis and reflection	Final project presentations; discussion of lessons learned and next steps

Each week, we will look at a method / package, data set, and / or visualization / analysis technique.

Mandatory Reading

(Details and updates will be provided on Canvas.)

Classroom Behavior

Students and faculty are responsible for maintaining an appropriate learning environment in all instructional settings, whether in person, remote, or online. Failure to adhere to such behavioral standards may be subject to discipline. Professional courtesy and sensitivity are especially important with respect to individuals and topics dealing with race, color, national origin, sex, pregnancy, age, disability, creed, religion, sexual orientation, gender identity, gender expression, veteran status, marital status, political affiliation, or political philosophy.

For more information, see the classroom behavior policy, the Student Code of Conduct, and the Office of Institutional Equity and Compliance.

Use of Large Language Models (LLMs) and AI Tools

In this course, the use of large language models (LLMs) and other generative AI tools (for example ChatGPT, Copilot, Claude, Gemini, etc.) is **not permitted** for any course work unless I explicitly tell you otherwise for a particular assignment or activity. This includes, but is not limited to, using LLMs to:

- write, edit, or debug code for homeworks, labs, or projects,
- generate, outline, or rephrase written answers or reports,
- design figures or data visualizations,
- translate, summarize, or restructure assignment prompts or solutions.

We will eventually use LLMs in this class in a guided and explicit way, once we have built the necessary fundamentals in programming and data analysis. When that time comes, I will clearly state what tools you may use and how you may use them.

Until you receive that explicit permission, you should assume that use of LLMs and similar AI tools on any graded work is not allowed and will be treated as academic misconduct, in line with the Honor Code and policies described elsewhere in this syllabus.

Consequences for Unauthorized Use of LLMs

If you use an LLM or other generative AI tool on an assignment without prior permission, the following consequences will apply:

1. **First offense:** You will receive a zero on that assignment.
2. **Second offense:** You will receive a negative score on that assignment (i.e., the penalty will reduce your overall course grade, not just give a zero).

3. **Third offense:** I will file an official report of academic misconduct with the university, in addition to the academic penalties described above.

These rules are intended to protect the learning goals of the course. Mastering the core skills yourself is essential before we bring LLMs into the workflow in a controlled and transparent way.

Accommodation for Disabilities, Temporary Medical Conditions, and Medical Isolation

If you qualify for accommodations because of a disability, please submit your accommodation letter from Disability Services to your faculty member in a timely manner so that your needs can be addressed. Disability Services determines accommodations based on documented disabilities in the academic environment. Information on requesting accommodations is located on the Disability Services website. Contact Disability Services at 303-492-8671 or DSinfo@colorado.edu for further assistance. If you have a temporary medical condition, see "Temporary Medical Conditions" on the Disability Services website. If you have a temporary illness, injury or required medical isolation for which you require adjustment, please let me know and we will find a solution. You automatically get three absences from class excused.

Preferred Student Names and Pronouns

CU Boulder recognizes that students' legal information does not always align with how they identify. Students may update their preferred names and pronouns via the student portal; those preferred names and pronouns are listed on instructors' class rosters. In the absence of such updates, the name that appears on the class roster is the student's legal name.

Honor Code

All students enrolled in a University of Colorado Boulder course are responsible for knowing and adhering to the Honor Code. Violations of the Honor Code may include but are not limited to: plagiarism (including use of paper writing services or technology such as essay bots), cheating, fabrication, lying, bribery, threat, unauthorized access to academic materials, clicker fraud, submitting the same or similar work in more than one course without permission from all course instructors involved, and aiding academic dishonesty.

Understanding the course syllabus is a vital part of adhering to the Honor Code. All incidents of academic misconduct will be reported to Student Conduct & Conflict Resolution (StudentConduct@colorado.edu). Students found responsible for violating the Honor Code will be assigned resolution outcomes from Student Conduct & Conflict Resolution as well as be subject to academic sanctions from the faculty member. Visit the Honor Code webpage for more information on the academic integrity policy.

In the spirit of the class, we will philosophically follow the python code of conduct.

Sexual Misconduct, Discrimination, Harassment and/or Related Retaliation

CU Boulder is committed to fostering an inclusive and welcoming learning, working, and living environment. University policy prohibits protected-class discrimination and harassment, sexual misconduct (harassment, exploitation, and assault), intimate partner violence (dating or domestic violence), stalking, and related retaliation by or against members of our community on- and off-campus. These behaviors harm individuals and our community.

The Office of Institutional Equity and Compliance (OIEC) addresses these concerns, and individuals who believe they have been subjected to misconduct can contact OIEC at 303-492-2127 or email cureport@colorado.edu. Information about university policies, reporting options, and support resources can be found on the OIEC website.

Please know that faculty and graduate instructors have a responsibility to inform OIEC when they are made aware of incidents related to these policies regardless of when or where something occurred. This is to ensure that individuals impacted receive an outreach from OIEC about their options for addressing a concern and the support resources available. To learn more about reporting and support resources for a variety of issues, visit the "Don't Ignore It" website.

Religious Accommodations

Campus policy regarding religious observances requires that faculty make every effort to deal reasonably and fairly with all students who, because of religious obligations, have conflicts with scheduled exams, assignments or required attendance. In this class, please let the instructor know at the beginning of the course if you foresee any absences or conflicts, so that an accommodation can be found right away. See the campus policy regarding religious observances for full details.

Mental Health and Wellness

The University of Colorado Boulder is committed to the well-being of all students. If you are struggling with personal stressors, mental health or substance use concerns that are impacting academic or daily life, please contact Counseling and Psychiatric Services (CAPS) located in C4C or call (303) 492-2277, 24/7. Free and unlimited telehealth is also available through Academic Live Care. The Academic Live Care site also provides information about additional wellness services on campus that are available to students.