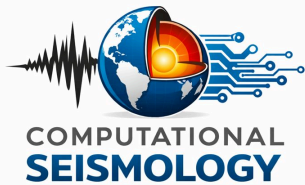


# GLY 6739.017S26 – Computational Seismology

**Credit Hours:** 3

**Semester:** Spring 2026



---

## Course Information

**Class Meetings:** Tuesday & Thursday, 11:00–12:15

**Location:** SCA 534

**Lab:** SCA 210

**Instructor:** Glenn Thompson

**Office:** SCA 226a

**Office Hours:** Friday, 2:00–3:30 pm

**Email:** [thompsonsg@usf.edu](mailto:thompsonsg@usf.edu) (<mailto:thompsonsg@usf.edu>)

---

## Welcome

Welcome to *Computational Seismology*. This course explores how modern seismology is actually practiced: with computers embedded in instruments, data pipelines, and monitoring systems. While the term “computational seismology” is often used to describe numerical wavefield modeling and inversion, this course instead emphasizes computational methods for working with **real digital seismic and infrasound data**, from acquisition through interpretation.

The course is intentionally hands-on, exploratory, and adaptive, reflecting how computational seismology is done in research and observatory environments.

---

## University Course Description

Computational Seismology introduces graduate students to modern computational methods used in observational seismology, with an emphasis on digital seismic and infrasound data, reproducible workflows, and practical analysis techniques. The course focuses on how seismic data are acquired, processed, analyzed, and interpreted using contemporary computational tools, rather than on numerical wavefield modeling or synthetic seismogram generation.

Students will develop foundational fluency in Python and commonly used scientific libraries (NumPy, pandas, ObsPy) and will work extensively with Jupyter notebooks and version control systems. Topics include seismic instrumentation and digitization, signal processing, event detection and phase picking, earthquake location, event classification and quantification, continuous seismic data analysis, ambient noise methods, velocity change detection, and introductory machine learning approaches.

Assessment is based on weekly laboratories, quizzes, participation, and a final project rather than high-pressure exams.

---

## Prerequisites

Graduate standing in Geosciences or permission of the instructor. No prior experience with Python or Linux is required, though familiarity will be helpful.

---

## Course Purpose

This course introduces computational methods in **observational seismology**, emphasizing reproducibility, automation, and real-world workflows. Students will learn how seismic data are acquired, processed, analyzed, and interpreted using modern computational tools.

Unlike theory- or modeling-focused seismology courses, this course centers on:

- real seismic instruments,
- real datasets,
- operational constraints, and
- computational pipelines used in research and monitoring.

The course supports students pursuing research, observatory, or applied geophysics careers.

---

## Course Format

The course combines:

- short lectures,
- live demonstrations,
- discussion, and
- asynchronous, notebook-based laboratories.

Most technical learning occurs through weekly Jupyter notebook assignments completed on shared Linux workstations in the USF Seismic Lab.

---

## Student Learning Outcomes

By the end of the course, students will be able to:

- Understand modern seismic data formats and standards (MiniSEED, StationXML, QuakeML)
  - Use ObsPy Stream, Inventory, and Catalog objects in data-processing pipelines
  - Process, visualize, and interpret waveform and catalog data in Python
  - Apply event detection, association, location, and quantification methods
  - Perform basic digital signal processing and spectral analysis
  - Work with both event-based and continuous seismic data
  - Build reproducible workflows using GitHub and Jupyter notebooks
  - Critically evaluate computational methods used in seismological research
- 

## Course Objectives

During the semester, students will:

- Gain foundational fluency in Python, NumPy, pandas, and ObsPy
  - Work directly with real seismic and infrasound data streams
  - Understand how instrumentation and digitization constrain analysis
  - Explore both earthquake-based and noise-based seismology
  - Develop workflows that can contribute to active research projects
-

# Course Materials

There are no required textbooks. All materials are free and open-source:

- Python (Anaconda or Miniconda)
  - Jupyter Notebook / JupyterLab
  - ObsPy
  - NumPy, pandas, matplotlib
  - GitHub
  - Visual Studio Code
  - Linux & bash
- 

## How to Succeed in This Course

- Expect to learn by doing, not memorizing
  - Start labs early—computational work benefits from iteration
  - Use version control consistently
  - Treat Jupyter notebooks as scientific documents
  - Ask questions early when concepts feel unclear
- 

## Grading

### Grade Scale

Percentage	Grade
90–100	A
80–89	B
70–79	C
60–69	D
0–59	F

### Grade Components

Component	Weight
Weekly Labs	40%
Weekly Quizzes	15%
Participation & Engagement	10%
Midterm Project (workflow reproduction)	15%
Final Project (GitHub repo + presentation)	20%

This course emphasizes frequent, low-stakes assessment rather than high-pressure exams.

## Course Schedule (Tentative)

*The course schedule represents a planned progression and may be adjusted based on pacing and interests. There is no written final exam; the final project replaces the exam.*

Computational Seismology is a huge topic that could easily occupy several 3-credit courses, so the following schedule is just a suggestion. Material will be developed as we go, giving us flexibility to decide which topics interest us most, and substitute accordingly. Some material I can teach you, but much of it we will be exploring together (though I will try to stay one step ahead!).

Date	Class	Topics
<b>Tue Jan 13</b> (away)	1A	<b>01 History of Computing: Part 1</b> (Reading) <ul style="list-style-type: none"> <li>Ask a senior faculty (Jan 27, 5 pts)</li> </ul>
<b>Thu Jan 15</b> (away)	1B	<b>02 Set up a modern scientific computing stack</b> (DIY, 5 pts)
	Lab 1	<b>03 Run C, Fortran, and Python programs on a server</b> (5 pts)
<b>Tue Jan 20</b>	2A	<b>Course introduction &amp; overview:</b> Review of week 1 materials, shell script. Overview of today's materials. <b>04 History of Computing: Part 2</b> (Reading) <b>05 History of Seismic Instrumentation</b> (Reading) Demo: Helicorder plots from Pinatubo, Soufriere Hills

Date	Class	Topics
Thu Jan 22	2B	<b>Analog vs Digital:</b> ADC, sampling, quantization. <b>Installing &amp; Servicing a seismic station &amp; data QC:</b> Nanometrics, Raspberry Shake. Logon.
	Lab 2	Synthetic analog 64 bit Stromboli VLP EQ, 5000 Hz? With a Mark L4 and with a CMG 3T, and a Trillium 120s Reconstitution. Effects of clipping, single component. Also, play with sampling & quantization.
Tue Jan 27	3A	Python Crash Course: Part 1
Thu Jan 29	3B	Python Crash Course: Part 2
	Lab 3	Python Lab
Tue Feb 3	4A	Python Crash Course: Part 3
Thu Feb 5	4B	ObsPy Crash Course: Part 1
	Lab 4	ObsPy Lab
Tue Feb 10	5A	ObsPy Crash Course: Part 2
Thu Feb 12	5B	ObsPy Crash Course: Part 3
	Lab 5	TBD
Tue Feb 17	6A	ObsPy III: Data processing essentials
Thu Feb 19	6B	Alternatives to ObsPy: SAC, Seisan, Antelope
	Lab 6	TBD
Tue Feb 24	7A	Continuous amplitude & frequency analysis

Date	Class	Topics
Thu Feb 26	7B	Event Detection: STA/LTA triggering
	Lab 7	TBD
Tue Mar 3	8A	Event Classification: Basic
Thu Mar 5	8B	Phase Picking & Association: Basic
	Lab 8	TBD
Tue Mar 10	9A	1-D Velocity Model Estimation
Thu Mar 12	9B	Event Location & Quantification
	Lab 9	TBD
Mar 16–22		<b>SPRING BREAK – No Classes</b>
Tue Mar 24	10A	Catalog analysis & seismicity patterns
Thu Mar 26	10B	Swarms, multiplet analysis, and double difference
	Lab 10	TBD
Tue Mar 31	11A	Ambient Noise Cross-Correlation
Thu Apr 2	11B	Green's Function Estimation
	Lab 11	TBD
Tue Apr 7	12A	Velocity Change Detection ( $dv/v$ , CWI)
Thu Apr 9	12B	Moment tensor estimation (MTUQ)

Date	Class	Topics
	Lab 12	TBD
<b>Tue Apr 14</b>	13A	Machine Learning 1: Phase Picking (& Association)
<b>Thu Apr 16</b>	13B	Machine Learning 2: Event Classification
	Lab 13	TBD
<b>Tue Apr 21</b>	14A	Locating & Quantifying Moving Sources
<b>Thu Apr 23</b>	14B	Rocket Seismo-Acoustics
	Lab 14	TBD
<b>Tue Apr 28</b>	15A	Course Summary
<b>Thu Apr 30</b>	15B	<b>Final Project Presentations</b>
<b>May 2-7</b>	<b>Final Project Due</b>	<b>FINAL EXAM WEEK</b> No written exam – Final projects submitted.

## Course Policies

### Late Work

Assignments submitted one day late receive a half-letter-grade penalty; assignments 2–7 days late receive a full-letter-grade penalty. Work more than seven days late will not be accepted.

### Attendance & Professionalism

Students are expected to attend class, arrive on time, and participate respectfully. Repeated disruptions or habitual lateness may result in a reduction of the final course grade.



## Generative AI Use

The ethical use of generative AI tools (e.g., ChatGPT) is **encouraged**. AI may be used to explore ideas, draft code, and support learning, provided that:

- work is grounded in real, credible sources,
- AI use is disclosed and cited,
- interactions with AI tools are documented and available upon request.

AI may not be used as a substitute for original analysis or understanding.

## Academic Integrity

Students must comply with USF's Academic Integrity Policy. Contract-cheating services (e.g., Chegg, CourseHero) may not be used in any capacity that violates this policy.

---

## Academic Continuity

In the event of campus disruption (e.g., hurricane), the course will continue via Canvas using recorded lectures, asynchronous labs, and remote datasets.

---

## University Policies and Student Resources

This course follows all **USF Core Syllabus Policies**, including policies on accessibility, academic integrity, religious observances, free expression, Title IX, and student wellbeing. Students are responsible for reviewing these policies on the USF Provost website.

Information on academic accommodations, counseling services, tutoring, the Writing Studio, and technical support is available through USF Student Success and Canvas Help.

---

## Final Note

Computational seismology is a rapidly evolving field. This course emphasizes thinking computationally, working ethically, and developing skills that will remain valuable as tools and methods change.