ECE 792: Special Topics in Software and Simulation Techniques for Power Systems

Department of Electrical and Computer Engineering
North Carolina State University

Spring 2026

Course Information

Instructor:	Hantao Cui, Ph.D.
Email:	hcui9@ncsu.edu
Office Hours:	Office meetings and/or Zoom calls by appointment
Lectures:	In-class presentation and short video tutorials
Location:	2121 Fitts-Woolard Hall
Time:	11:45–1:00, Mondays and Wednesdays, Spring 2026
Prerequisites:	ECE 451 Power System Analysis or Equivalent
Credit Hours:	3 Hours

1 Course Description

This course will review the fundamental algorithms for balanced and unbalanced power flow, stability simulation and electromagnetic transient simulation. Open-source software will be used to demonstrate software architecture for structuring complex, engineering-focused simulation tools that can be extendable to various models. Advanced modeling techniques such as dynamic phasor will be discussed. Advanced power system and power electronics models for simulation will also be covered.

Substantial hands-on programming in Python is required throughout the course.

2 Student Learning Outcomes

Upon successful completion of this course, students will be able to:

- Understand the commercial and open-source software landscape in the electric power industry.
- Understand widely used algorithms for power flow calculation, unbalanced power flow calculation, stability analysis, and electromagnetic transient simulation.
- Understand how power system components, such as constant power load, constant impedance load, generators, renewable generators, and controllers, are modeled in existing commercial and open-source software.

- Develop structured computer programs for power system analysis, including power flow calculation and numerical integration for stability simulation.
- Understand and apply software engineering principles, such as test-driven development, documentation, continuous integration, version control, modularity, and maintainability, to power system analysis software development.
- Use software libraries and packages, such as optimized linear algebra libraries and sparse matrix libraries, for scientific computing in power systems. Understand the pros and cons of the libraries.
- Understand and apply design principles to scientific computing. Understand the pros and cons of the design choices.

3 Course Topics and Lecture Schedule

Lecture	**Topic**
Lecture 1	Introduction
Lecture 2	Timescale and Software Structure
Lecture 3	Power Flow Fundamental and Network Modeling
Lecture 4	Power Flow Problem Abstraction and Component-Based Modeling
Lecture 5	Advanced Models in Power Flow
Lecture 6	Power Flow Solution Methods and Jacobian Calculation
Lecture 7	Sparse Matrix Formats
Lecture 9	Newton Method and Variants
Lecture 10	Computational Performance and Solving Sparse Linear Equations
Lecture 11	Revisiting Bus Admittance Matrix
Lecture 12	Numerical Integration Fundamentals
Lecture 13	Partitioned Method and Implicit Integration Methods
Lecture 14	Simultaneous Solution of DAEs
Lecture 15	Causal & Acausal Modeling; Initialization
Lecture 16	Fundamentals of Parallel Computing for Power Systems
Lecture 17	Introduction to GPU Programming
Lecture 18	Accelerating Python for Numerical Studies
Lecture 19	Real Time Simulation

Note: *The exact course schedule is subject to changes without prior notice.*

4 Programming Projects

There will be a series of programming-based assignments to guide students through the development of practical software for power system analysis. Project descriptions, requirements, and code skeletons will be provided.

Project	**Description**
Project 1	Power System Data Pipeline and Plotting *Introduction to Python power systems toolchain through building a data processing and visualization pipeline using MATPOWER's IEEE 14-bus system.*
Project 2	Turning Code Snippets into a Package *Refactor Project 1 codebase into a modular Python package, discuss design considerations, implement an admittance matrix builder function, and introduce unit testing.*
Project 3	Equation Addressing in a General Simulation Framework *Develop a power flow program with focus on addressing issues that form the foundation of a generalized simulation program.*
Project 4	Generalized Addressing and Residual Storage *Continue working on equation residual addressing and generalize it to variable addressing.*
Project 5	Interfacing with scipy.optimize.fsolve *Complete the remaining parts of the component-based modeling framework, implement residual functions in each model, and properly assemble the global residual vector.*
Project 6	Refactor, Refactor, Refactor *Apply refactoring techniques and test the power flow solver on larger systems such as the IEEE 14-bus system.*

Note: *Projects may be extended with additional requirements or tasks.*

5 Final Project

Each student will choose a project for which they will develop an open-source tool. The student can choose a new topic or build upon existing research projects. The expectation is that the software engineering principles will be applied to guide the design, development and documentation of the tool.

Expected deliverables include:

- Presentations of the design and outcome
- Codebase with adequate tests and documentation

6 References

1. Federico Milano, *Power System Modelling and Scripting*, Springer, 2010

2. EPRI, *OpenDSS Manual*, Available online: `https://sourceforge.net/projects/electricdss/files/OpenDSS/OpenDSSManual.pdf/download`

3. Neville Watson & Jos Arrillaga, *Power Systems Electromagnetic Transients Simulation*, IET Power and Energy Series 39, 2002

7 Grading

Component	**Weight**
Paper-based homework	20%
Programming-based homework	50%
Final project of your choice	30%

8 Homework Policy

Homework assignments and solutions will be posted on the course Moodle site. Late homework will be accepted for 24 hours after the due date/time for 90% credit. No credit will be available after the solutions are posted.

Students should utilize course Moodle forums for online questions regarding homework/exams instead of sending emails directly to the instructors; use personal email for any unusual circumstances that may arise during the semester.

There will be a light paper-based homework assignment for each module to serve as review materials. There will be a series of programming-based assignments to guide students through the development of a practical software for power system analysis. Project descriptions, requirements and code skeleton will be provided.

9 Academic Integrity

Work in this course is to be done under the Academic Integrity Honor Pledge:

> *"I have neither given nor received unauthorized aid on this test or assignment."*

Students must abide by the Code of Student Conduct articulated at: `http://www.ncsu.edu/policies/student_services/student_discipline/POL11.35.1.php`

Evidence of copying, including copying of source code, or any other use of unauthorized aid will be investigated and potentially referred to the University judicial system as a violation of the Code of Student Conduct. The minimum sanction for a violation is a zero on an assignment. Recycling of projects from another resource will be considered an academic integrity violation.

Use of Generative Artificial Intelligence

The use of generative artificial intelligence (AI) tools, including but not limited to large language models such as ChatGPT, Claude, Copilot, or similar technologies, is strictly prohibited for all programming assignments and projects in this course. Specifically:

- Students may not input project descriptions, requirements, specifications, or any course materials into generative AI systems.

- Students may not submit source code, in whole or in part, to generative AI tools for analysis, debugging, completion, or modification.
- Students may not use AI-generated code or AI-assisted code generation tools to complete any portion of the assigned work.

Submitting work that has been generated or substantially assisted by generative AI constitutes a violation of the University's Code of Student Conduct and will be treated as academic dishonesty. Such violations will be subject to the full range of academic sanctions, including but not limited to receiving a zero on the assignment, failure in the course, and referral to the Office of Student Conduct.

The pedagogical objective of this course is to develop your independent problem-solving abilities and programming skills through hands-on practice. The use of generative AI undermines these learning outcomes and deprives you of the educational benefits that come from working through challenges independently.

10 Accommodation

Reasonable accommodations will be made for students with verifiable disabilities. In order to take advantage of available accommodations, students must register with Disability Services for Students at 1900 Student Health Center, Campus Box 7509, (919) 515-7653. For more information on NC State's policy on working with students with disabilities, please see the Academic Accommodations for Students with Disabilities Regulation (REG 02.20.01).

11 N.C. State University Policies, Regulations, and Rules (PRR)

Students are responsible for reviewing the PRRs which pertain to their course rights and responsibilities. These include:

- `http://policies.ncsu.edu/policy/pol-04-25-05` (Equal Opportunity and Non-Discrimination Policy Statement)
- `http://oied.ncsu.edu/oied/policies.php` (Office for Institutional Equity and Diversity)
- `http://policies.ncsu.edu/policy/pol-11-35-01` (Code of Student Conduct)
- `http://policies.ncsu.edu/regulation/reg-02-50-03` (Grades and Grade Point Average)