

Overview (30% of Final Grade)

The final project is an opportunity to explore a mathematical, scientific, or technical question related to or inspired by this course. Topics may involve real-world modelling, engineering, or abstract theory, including areas such as differential equations, control, signal processing, or deep learning.

Projects must be done in groups of 2–4 (with workload scaling by group size). Individual work is discouraged but may be approved by instructors. Interdisciplinary and applied topics are welcome, as long as the project has a clear mathematical core.

Project Timeline and Components

- **Proposal (5%).** A concise 2-page proposal outlining your motivation, objectives, mathematical background, and expected outcomes. **Due: October 3, 11:59pm.**
- **Weekly Check-ins (5%).** Regular progress updates during your lab sessions with lab TA.
- **Final Report (10%).** A structured academic-style paper, due during the last week of classes. **Due: TBD.**
- **Code/Computational Deliverables (10%).** Submit well-documented and reproducible code. **Due: TBD.**

Suggested Project Types

Projects may include (but are not limited to):

- Modeling a real-world system using ODEs and analyzing stability, control, or behavior.
- Developing or evaluating a numerical method for solving ODEs or PDEs.
- Implementing a machine learning technique and analyzing training dynamics through the lens of differential equations.
- Investigating physical phenomena (e.g., mechanical vibrations, electrical circuits, fluid flow) with mathematical modeling.
- Exploring a mathematical concept not covered in class (e.g., bifurcation theory, chaos, neural ODEs).

Proposal Rubric (5%)

Criteria	Weight	Description
Motivation and Relevance	30%	The problem is clearly stated, interesting, and tied to a real-world or mathematical context. A clear, self-contained project goal statement should be included somewhere in the document.
Scope and Feasibility	30%	The proposed work is realistic given the timeline; objectives are clearly defined. A list of project milestones and estimated time of completion should be included.
Technical Background	20%	Some mathematical or computational context is provided; <u>at least 5</u> academic references are included.
Clarity and Writing Style	20%	The proposal is clearly written, well-structured, and formatted in L ^A T _E X (see L ^A T _E X starter document).

Code Rubric (10%)

Criteria	Weight	Description
Functionality	40%	Code runs without errors and produces expected, reproducible results.
Documentation	25%	Code includes helpful comments, clear function headers, and usage instructions.
Reproducibility	20%	Code generates the results shown in the report without modification.
Style and Organization	15%	Code is modular, readable, and well-organized.

Final Report Rubric (10%)

Criteria	Weight	Description
Structure and Organization	15%	Follows the structure of an academic paper (Abstract, Intro, etc.) with logical flow.
Mathematical Rigor	25%	Mathematical content is accurate, well-justified, and clearly presented.
Originality and Insight	20%	Demonstrates critical thinking, creativity, or depth of analysis.
Writing and Presentation	20%	Clear, grammatically correct writing using professional tone and L ^A T _E X formatting.
Use of Visuals	10%	Figures, tables, and graphs are well-integrated and support the findings.
Citations and References	10%	Sources are properly cited in a consistent bibliographic style.

The final report should include the following sections:

1. **Abstract** — A brief summary of the objectives, methods, and key findings.
2. **Introduction** — Describe the problem domain and motivation. Include relevant background.
3. **Mathematical Model or Theoretical Foundation** — Present core equations, derivations, or theoretical tools.
4. **Methodology** — Detail any computational/numerical methods or experimental setup.
5. **Results** — Present and interpret key results using figures, graphs, or tables.
6. **Discussion** — Interpret your findings, identify limitations, and suggest further work.
7. **Conclusion** — Summarize your contribution.
8. **References** — Include a properly formatted bibliography.
9. **Appendix (Optional)** — Include code snippets, proofs, or extended results.

Example Project Ideas

- *Transfer Learning in Time Series Forecasting* — Explore how pretrained neural networks can accelerate training on ODE-governed data.
- *Modeling Epidemics with SIR Models* — Implement numerical integration and perform parameter sensitivity analysis.
- *Stability of Inverted Pendulums* — Investigate equilibrium points and apply linearization techniques.
- *Physics-Informed Neural Networks (PINNs)* — Train a PINN to solve boundary value problems.
- *Damped Harmonic Oscillator with Forcing* — Compare analytical and numerical solutions under external driving forces.
- *Chebyshev Polynomials* — Study approximation properties and applications in spectral methods.
- *Bifurcations in Higher Dimensions* — Survey homoclinic, hysteresis, and other bifurcation phenomena.
- *The Poincaré–Bendixson Theorem* — Explore its role in characterizing planar dynamical systems.
- *The Lorenz Attractor* — Analyze chaotic dynamics through simulation and visualization.
- *Discrete Dynamical Systems and Fractals* — Investigate iteration maps and fractal structures.
- *Classical Limits of Quantum Systems* — Examine the transition from quantum to classical mechanics.
- *Fast Fourier Transform (FFT)* — Implement the FFT and study applications in signal analysis.
- *Boundary Value Problems and the Relaxation Method* — Apply relaxation techniques, e.g., to adiabatic cooling models.

Note that some projects will be simpler than others to complete. If you present something truly novel, low expectations will be set on the quality of the output. If you pick a simpler project, your group will be held to much stricter standards. Discuss with someone on the teaching team to gauge the difficulty of your project.

Additional Details

Submission Instructions: All materials (PDF report, proposal, code files, and any data) will be submitted via Quercus in a designated folder. A zip file with all code + README is expected. Use `.zip` format and include your name(s) in the filename.

Academic Integrity: You may discuss high-level ideas with classmates, but all submitted work must be your own. Any external sources must be clearly cited. Plagiarism will not be tolerated.

Questions? Contact someone from the teaching team early and often. We are here to support you.