

CSE 548: High Performance Computing II

Spring 2026 Syllabus

Matthew G. Knepley
211A Capen Hall knepley@buffalo.edu

In this course, students use the skills learned in HPC I to build a lasting software artifact, intended to benefit a broader community. We will cover software management, such as version control, hosting, and continuous integration, in addition to algorithmic topics, such as mesh and discretization management, algebraic solvers, and scalable communication.

1 Course Information

Instructor [Matthew G. Knepley](mailto:knepley@buffalo.edu)
Class times 12:30pm to 1:50pm on Tuesday & Thursday
Location Baldy 105

2 Required and Recommended Reading

[Class notes](#) have been prepared for each class, and the lecture will follow the notes. Extensive class notes are also available from the [Computational Science I](#) companion course. Students may read over the class notes prior to attending lecture, but it may deviate from the notes somewhat.

We will use the [PETSc libraries](#) as a foundation for software development. An optional textbook, [PETSc for Partial Differential Equations](#) by Ed Bueler of the University of Alaska Fairbanks, is available from the SIAM Bookstore or Google Play. It is the best resource for learning the advanced features of PETSc.

3 Student Learning Outcomes

- Students will be capable of managing a software repository with multiple developers
- Students will be capable of constructing a build system for a software package
- Students will be capable of continuous integration testing

- Students will be capable of choosing an effective linear solver for a system of algebraic equations
- Students will be capable of modeling parallel performance
- Students will be capable of analyzing parallel scaling

4 Use of Generative AI

All work submitted for this course must be your own. As a result, any use of generative AI (e.g., ChatGPT or Grammarly) is prohibited in this class and will be considered a violation of UB's academic integrity policy. Details of what resources are allowed will be provided for each assignment. If you are unsure if a resource or tool is allowable, be sure to ask.

5 Schedule

5.1 Week 0: Overview 1/22

5.1.1 Lecture 0: Course Outline and Structure

We will go over the course overview and syllabus. We will discuss as a class what topics we will focus on this semester.

5.2 Week 1: Projects 1/27

5.2.1 Lecture 1: Projects

We discuss the course projects. Student proposals for alternative projects should be made on the week projects are assigned. We will also go over the pieces of the Project Report.

5.2.2 Practicum 1: Projects

We will setup source management and hosting, and look at library dependencies.

5.3 Week 2: Software Management 2/3

5.3.1 Lecture 2: Version Control

We review version control.

Online resources

- Really fantastic graphical, interactive [Git Branching Tutorial](#)
- [Online Git Documentation](#)
- [10 Years of Git](#)

- [GitHub](#) and [GitLab](#) are good hosting options

5.3.2 Practicum 2: Version control

We setup project version control, SSH access, and look at source evaluation tools.

Online resources

- [GitHub](#) and [GitLab](#) are good hosting options
- PETSc guide for [contributing](#) code to the repository and [integrating](#) your contribution into the release.

5.4 Week 3: Build Tools 2/10

Students will work on Project I.

5.4.1 Lecture 3: Build Tools

We discuss Configure, Make, Automake, CMake, Meson, and other build tools.

5.4.2 Practicum 3: Build Tools

We assemble a build environment for projects.

Homework 1: Turn in Project Introduction.

5.5 Week 4: Independent Work 2/17

5.5.1 Lecture 4: Project Description

Students will write a one page project description, including its relevance to high performance computing. In addition, they will construct a weekly schedule for project development and testing, and a set of performance milestones.

5.5.2 Practicum 4: Students will install PETSc

Everyone will configure and build PETSc on their machine.

Online resources

- The PETSc [installation](#) instructions
- The PETSc [manual pages](#) and [tutorials](#) are probably the most helpful resources for new users.
- Good [introduction](#) to PETSc

5.6 Week 5: Linear Algebra 2/24

5.6.1 Lecture 5: Linear Algebra

We discuss parallel linear algebra. Project I is assigned.

Online resources

- [Using MPI](#) is a good introduction to parallel programming and has many linear algebra examples.
- [Linear Algebra: Foundations to Frontiers](#) is an excellent multimedia introduction to parallel linear algebra.

5.6.2 Lecture 5b: Implementation

We discuss the PETSc implementation of linear algebra, including sparse matrices, data access and encapsulation, and type management.

5.7 Week 6: Parallel Applications 3/3

5.7.1 Lecture 6a: Guest Lecture

5.7.2 Lecture 6b: Guest lecture

Homework 2: Turn in Project Background.

5.8 Week 7: Performance Analysis 3/10

5.8.1 Paper Discussion 7: LogP Model

We discuss the LogP model paper.

Online resources

- This [paper](#) describes a new model (in 1993) for assessing parallel communication.

5.8.2 Practicum 7: Measuring Communication

We setup our code to measure parallel communication.

5.9 Week 8: Spring Recess 3/17

5.9.1 Lecture 8: No class

5.9.2 Practicum 8: No class

5.10 Week 9: Optimizing Communication 3/24

5.10.1 Paper Discussion 9: Optimizing Collectives

We discuss the paper on modeling and optimizing collective communication.

Online resources

- This [paper](#) describes a model for collective communication that they use to optimize the implementation.

5.10.2 Practicum 9: Optimizing Communication

Students model communication in their project and think about possible optimizations.

5.11 Week 10: Synchronization 3/31

5.11.1 Paper Discussion 10: Synchronization

We discuss the paper on modeling synchronization in Krylov methods.

Online resources

- This [paper](#) describes a model for synchronization in Krylov methods. This [paper](#) has more experiments.

5.11.2 Practicum 10: Synchronization

Students look at the synchronization cost in their projects and think about strategies to mitigate it.

Homework 3: Turn in Project Methodology.

5.12 Week 11: Node-aware Communication 4/7

5.12.1 Paper Discussion 11: Node-aware Communication

We discuss the paper on node-aware communication.

Online resources

- This [paper](#) describes a model for parallel communication that takes into account modern architectures.

5.12.2 Practicum 11: Communication Optimization

Students will benchmark parallel communication in their projects.

5.13 Week 12: Scalable Solvers 4/14

5.13.1 Paper Discussion 12: Communication in Multigrid

We discuss the paper on communication in the multigrid algorithm

Online resources

- This [paper](#) discusses modeling and optimizing communication in multigrid.

5.13.2 Practicum 12: Solver Profiling

Students profile solver performance in their projects and think about possible optimizations.

5.14 Week 13: Communication in AI 4/21

5.14.1 Paper Discussion 13: Network Topology

We discuss the paper on network topology for AI training.

Online resources

- This [paper](#) discusses network topology for AI training.

5.14.2 Practicum 13: Modeling

Students compute the effect of a different topology for communication in their project, using a communication performance model.

Homework 4: Turn in Project Results and Conclusions.

5.15 Week 14: Final Project Presentations 4/28

5.15.1 Lecture 14a: Presentations

Students give a final presentations of their project results.

5.15.2 Lecture 14b: Presentations

Students give a final presentations of their project results.

5.16 Week 15: Course Evaluation 5/5

5.16.1 Lecture 15: Class Discussion

We evaluate the class and discuss improvements.

Homework 5: Turn in final Project Report.