Syllabus for Methods and Practice of Scientific Computing (Fall 2020)
NERS 570/ENGR 570

Instructor:	Prof. Brendan Kochunas	bkochuna@umich.edu
GSIs:	Qicang Shen	qicangsh@umich.edu

Course Description:
This course is designed for graduate students who are developing the methods, and using the tools, of scientific computing in their research. With the increased power and availability of computers to perform massively scaled simulations, computational science and engineering as a whole has become an integral part of research that complements experiment and theory. This course will teach students the necessary skills to be effective computational scientists and how to produce work that adheres to the scientific method. A broad range of topics will be covered including: software engineering best practices, computer architectures, computational performance, common algorithms in engineering, solvers, software libraries for scientific computing, uncertainty quantification, verification and validation, and how to use all the various tools to accomplish these things.

Course Objective
Upon successful completion of the course students shall be able to

- develop and run software in Linux,
- write code in multiple languages,
- use compilers and Makefiles,
- write their own linear solver
- compile and use third party libraries,
- work in software projects with other individuals,
- develop version controlled software,
- implement automated testing in a software project,
- increase the computational performance of their software
- write code that uses MPI and/or OpenMP parallelism,
- perform simulations on high-performance computing resources,
- debug programs more efficiently

Grading (4 credit hours):

Lecture Quizzes	10%
Homework	15%
Lab	40%
Project	35%

Format for Synchronous Participation:

Bi-weekly Lectures on	M/W	2:00 PM - 3:30 PM, 1303 EECS
Weekly Lab on	F	9:00 AM - 11:00 AM, GGBL 2517

Guidelines for Asynchronous Participation

- Deadlines for asynchronous participation will be same as synchronous participation.
- Lecture Recordings will be uploaded to Canvas shortly after the lecture is given.

Office Hours:
Office hours’ time and format will be determined by a class poll. Following the first lecture.

Policies

- Deliverables for assignments must be completed in the appropriate format. Format for assignment deliverables will be given on a case by case basis. Expected formats are:
  - Typed documents
  - Source code
  - LaTeX
  - Program output
  - Repository commits
  - Canvas quizzes
- Late Assignments
  - Late homework assignments are penalized with a 5-day half-life
  - Late lab assignments are penalized with a 7-day half-life
  - Late project deliverables are penalized with a **3-day half-life**
- Honor code: https://elc.engin.umich.edu/honor-council/
- NO POLICY IS PERMANENT
  - If a policy is not working for the class, I'm open to discussing it and changing it.

Inclusion Statement:

It is my intention that students from all backgrounds and perspectives will be well served by this course, and that the diversity that students bring to this class will be viewed as an asset. I welcome individuals of all ages, backgrounds, beliefs, ethnicities, genders, gender identities, gender expressions, national origins, religious affiliations, sexual orientations, socioeconomic background, family education level, ability – and other visible and nonvisible differences. All members of this class are expected to contribute to a respectful, welcoming, and inclusive environment for every other member of the class. Your suggestions are encouraged and appreciated. In addition, I will gladly honor your request to address you by an alternate name or gender pronoun. Please advise me of this preference early in the semester so that I may make appropriate changes to my records. If you have any questions or concerns regarding Diversity, Equity, and Inclusion you may contact the DEI leads in your home department www.engin.umich.edu/about/diversity/faculty/dei-department-leads/

Students Requiring Accommodations:

Students that have documented disabilities and require academic accommodations should make an appointment to discuss their needs with the course instructor. Students must contact the Services for Students with Disabilities, ssd.umich.edu to verify their eligibility for appropriate accommodations.

Student Resources:

If you require additional resources please contact the Office of Student Support and Accountability, ossa.engin.umich.edu or engin-support@umich.edu.
In addition, if you or someone you know is feeling overwhelmed, depressed, and/or in need of support, services are available. You can learn more about the broad range of confidential mental health services available on campus via caps.umich.edu/mitalk

Course Technology Resources

- Canvas (this is the main resource): https://umich.instructure.com/courses/386862
- Zoom for Videoconferencing: https://its.umich.edu/communication/videoconferencing/zoom
    - Remote attendees are not required to have a camera
- VPN for remotely connecting to campus resources: https://its.umich.edu/enterprise/wifi-networks/vpn/getting-started
- CAEN Virtual Linux Desktop for Lab: https://caenfaq.engin.umich.edu/linux-login/how-do-i-connect-to-a-caen-linux-computer-remotely
- Windows SSH Clients:
    - PuTTY - https://www.putty.org/
    - MobaXterm - https://mobaxterm.mobatek.net/
- HPC Resources through ARC-TS Great Lakes: https://arc-ts.umich.edu/greatlakes/
- Instructional Material: https://www.lib.umich.edu/

Lecture Recording

Course lectures will have audio/video recorded and made available to other students in this course. As part of your participation in this course, you may be recorded. If you do not wish to be recorded, please contact Prof. Kochunas the first week of class to discuss alternative arrangements.

Lecture Recording Distribution

Students may not record or distribute any class activity without written permission from the instructor, except as necessary as part of approved accommodations for students with disabilities. Any approved recordings may only be used for the student's own private use.

Syllabus for Methods and Practice of Scientific Computing (Fall 2020)
NERS 570/ENGR 570

Course Schedule

Date	**Lecture**	**Lab**	**Topic**
08/31	1		Course Overview & Introduction to Linux
09/02	2		Programming Languages: C, C++, Fortran
09/04		*1*	*Introduction to Linux*
09/09	3		Scripting with Bash and Python
09/11		*2*	*Scripting*
09/14	4		Elements of Development: Configuring, Compiling, Linking
09/16	5		Tools of the Trade: Version Control, Dev. Env
09/18		*3*	*Introduction to Great Lakes and Git*
09/21	6		Algorithms for Linear Algebra
09/23	7		Sci. Computing Libs: BLAS, LAPACK, PETSc, Trilinos
09/25		*4*	*Working with Third Party Libraries*
09/28	8		Object-Oriented Programming, Design Patterns, UML
09/30	9		Software Engineering Practices & Development Workflows
10/02		*5*	*Workflows in Practice*
10/05	10		Serial and Parallel Architectures
10/07	11		Performance and Serial Optimization
10/09		*6*	*Micro-Benchmarks and Measuring Performance*
10/12	12		Parallel Programming Models
10/14	13		OpenMP
10/16		*7*	*Parallel Computing: OpenMP*
10/19	14		The Message Passing Interface I
10/21	15		The Message Passing Interface II
10/23		*8*	*Parallel Computing: MPI*
10/26	16		Class Selected Module 1
10/28	17		
10/30		*9*	
11/02	18		Class Selected Module 2
11/04	19		
11/06		*10*	
11/09	20		Class Selected Module 3
11/11	21		
11/13		*11*	
11/16	22		Class Selected Module 4
11/18	23		
11/20		*12*	
11/23	THANKSGIVING BREAK		
11/25			
11/27			
11/30*	24†		Special Topic / Term Project Presentations
12/02*	25		Term Project Presentations
12/04*	OPEN LAB - Work on Term Projects		
12/07*	26		Term Project Presentations

* Remote   † Subject to change

Optional Course Modules

Topic			Description
GPU Parallelism	Lecture 1		Heterogeneous Architectures
	Lecture 2		Programming models for GPUs
		Lab	*Hardware Abstraction with Kokkos*
Testing, Testing, Testing	Lecture 1		Testing, Verification, and Validation
	Lecture 2		How to write a Unit Test
		Lab	*Automated Testing Infrastructure*
Data and Mesh Libraries	Lecture 1		Data Format Libraries: HDF5, NetCDF, SILO
	Lecture 2		Mesh Libraries: Libmesh, Exodus, others
		Lab	*Working with Data Libraries*
Package Management & Containers	Lecture 1		Package and Dependency Management with Spack
	Lecture 2		Containers: Docker and Singularity
		Lab	*Spack and Singularity on Great Lakes*
Python for HPC	Lecture 1		Using Jupyter Notebooks with HPC
	Lecture 2		Packages for Scientific Computing
		Lab	*TBD*
Debugging and Profiling Tools	Lecture 1		Debugging: DDT, GDB, and Valgrind
	Lecture 2		Performance: MAP, HPCToolKit, TAU
		Lab	*Make it work; Make it fast -- Debug and Optimize*
Misc. Topics	Lecture 1		QA, deployment, copyrights, and licensing
	Lecture 2		Visualization Tools and Best Practices
		Lab	*Working with Paraview*

Anticipated Homework Assignments

HW	Description	Supporting Lectures	Due Date
1	LaTeX and Programming in C/C++ and Fortran	1,2	09/18
2	Using scientific software libraries for sparse linear algebra	1,4,6,7	10/12
3	Draft Project Proposal and TBD	8,10,11	10/30
4	Maybe		

Anticipated Lab Assignments

Lab	Description	Supporting Lectures	Due Date
1	Hands on walkthrough of Linux.	1	09/09
2	Bash and Python Scripting	1,3	09/21
3	Hands on walkthrough of Great Lakes	1,4	09/23
4	Working with LAPACK and PETSc	1,4,7	10/05
5	Perform a workflow and setup basic git and CMake	1,5,8,9	10/07
6	Computer performance.	1,4,10,11	10/19
7	OpenMP simulated annealing parallel programming.	1,4,6,10,14	10/26
8	MPI simulated annealing parallel programming.	1,4,6,10,15,16	11/02

Project Assignments

Deliverable	Description	Due Date
Proposal	3-5 page document	10/30
Presentation	10-20 minute presentation	12/02—12/07
Report	10-20 page typed document	12/18

Associated Readings and General References:

Lecture	Topic
1	None
2	Modern Fortran C/C++ programmer's reference
3	Bash Quick Start Guide Learning Python
4	None
5	Pro Git
6	Finite Difference Methods Finite Element Methods The Method of Weighted Residuals and Variational Principles Multigrid Methods Matrix Analysis Vol. 1 and Vol. 2 Iterative Methods for Sparse Linear Systems Model Reduction and Approximation Accuracy and Reliability of Scientific Computing Accuracy and Stability of Numerical Algorithms
7	LAPACK User's Guide ScaLAPACK User's Guide Numerical Recipes in Fortran (Online PDF) Numerical Recipes in C (Online PDF)
8	Scientific Software Design: The Object-Oriented Way The Unified Modeling Language Design Patterns
9	Code Complete Agile Development in the Real World
10-11	Numerical Linear Algebra for High-Performance Computers Performance Optimization of Numerically Intensive Codes
12	Patterns for Parallel Programming Parallel Processing for Scientific Computing
13	Using OpenMP
14	Using MPI
15	Using Advanced MPI