High Performance Computing Syllabus (APM 525/MAE 598), Fall 2018

Cross-listing: This course is cross-listed as APM 525 (SLN 92794) and as MAE 598 (SLN 92810).

Professor: Eric Kostelich, WXLR 833, telephone (480) 965-5006, email kostelich@asu.edu.

Class meeting times: Mondays and Wednesdays 3:05–4:20 in Wexler Hall A104.

Office hours: Mondays and Wednesdays 1:45–2:45 and by appointment.

Intended audience: Beginning graduate students who are considering masters and Ph.D. projects that will require substantial simulation in ordinary and partial differential equations, fluid dynamics, and related areas.

Prerequisites: CSE 205, Object-Oriented Programming and Data Structures or comparable second-semester programming course; one semester of linear algebra (MAT 242, 342, 343, or equivalent); and one semester of ordinary differential equations (MAT 274, 275, or equivalent). Some previous exposure to MATLAB, C/C++, Fortran, or other scientific computing language is helpful but not required.

Course topics: The exact schedule will depend on our progress, but the course includes an introduction to each of the following:

- IEEE floating-point arithmetic
- Computer architecture and memory hierarchies
- Arrays and data structures for high-performance computing
- Linear algebra software libraries
- Survey of MATLAB and Julia (the latter will depend on the stability of the implementation on Agave)
- Survey of Fortran 2008 and C99
- Survey of C++ with an emphasis on the Standard Template Library
- Vectorization in MATLAB, Fortran, and C/C++
- Notions of parallelism: shared-memory multiprocessing with OpenMP; distributed memory multiprocessing with the Message Passing Interface (MPI); and coarray Fortran as a standardized partitioned global address space (PGAS) language
- Software development tools: the POSIX shell; make utility; regular expressions and their use in grep, awk, and related command-line tools; source code control with git; job scheduling software.
- Survey of commonly used libraries for scientific computing, including BLAS/LAPACK, NetCDF/HDF, Netlib, and others.

Textbooks: Since so much useful material is available online, there are no required textbooks; see Selected Course Resources, below.

Software and course content: Most programs will be run on the Agave cluster, to which you will receive access. More details will be provided in class.

Course meetings: Attendance at all class meetings in their entirety is expected. If illness or other circumstance prevents you from attending, then I would appreciate an email. *Please do not come to class if you are feeling ill.* I will make appropriate accommodations with documentation from your health care provider.

Course format: The course will be run in a lecture/lab format. Typically, we will have one lecture followed by an in-class computer lab where you can work on related programming exercises with assistance from the instructors and other students. Please bring your laptop to class for the computer lab sessions, which will be announced ahead of time.

Unforeseen circumstances: If unusual circumstances (e.g., fire alarm, flooding, power failure) prevent us from holding class on a particular day, then an announcement will be posted to Blackboard about any rescheduling of course activities.

Grading policy: Grades will be based on a combination of programming exercises and related homework (75%) and a final programming project (25%).

Comments on the prerequisites: The expected mathematical background includes calculus, linear algebra, and differential equations sufficient to understand what is meant by a solution of a linear system, a singular linear system, Euler's method and the Runge-Kutta methods.

The expected programming background is a minimum of a one-year sequence in some procedural programming language, such as C/C++, Java, Fortran, MATLAB, etc. Some familiarity with MATLAB and C/C++ is helpful but not required. We will cover the basics of Fortran, C, and C++ in class, but you should be familiar with basic concepts like subroutines, looping, dummy arguments, etc.

We will cover the basics of the Unix operating system and shells in class. Some familiarity with Unix or the command-line interface of Windows operating systems is helpful but not required.

Collaboration policy: On most assignments, you will be permitted to consult with others on the homework and projects, subject to the following constraints:

1. *You must acknowledge your collaborators* and include a statement describing the portion(s) of the assignment on which you collaborated (and with whom).
2. You must adhere to ASU's Academic Integrity Policy. In part, the latter states that you may not "depend on the aid of others, including other students or tutors, in connection with any Academic Evaluation or assignment to the extent that the work is not representative of [your] abilities." Violations of the Academic Integrity Policy are subject to sanction, including grade penalties, course failure (indicated on the transcript as a grade of E), course failure due to academic dishonesty (indicated on the transcript as a grade of XE), loss of registration privileges, disqualification, and dismissal. Please review the policy carefully.
3. No collaboration is permitted on examinations.
4. If you have incorporated source code written by someone else in your homework or project, then you must document this fact and include a link or reference to the source.

5. If you have questions about what is permissible, please see me before starting the assignment.
6. If collaboration is forbidden on a particular assignment, then you must include the statement *I have not received help from any other student on this assignment* in your source code or handwritten work.

If you have any questions as to what is permissible, please consult me before starting the assignment.

Final projects: Final projects will be due at the final exam period, scheduled for Friday, Dec. 7 from 2:30–4:20 p.m. We will meet in the regular classroom and you will demonstrate that your program works.

Copyright notice: Unless otherwise noted, all instructor-prepared materials, including lectures, handouts, homework assignments, and solutions, are subject to copyright and may not be recorded, reproduced, or distributed without written permission. The copyright to other materials is held by their respective authors or as noted. Please do not make copies of these materials for any purpose other than your private use for the course and do not post them elsewhere online.

Classroom courtesy: Please be courteous to your classmates and instructor by avoiding distracting behavior during class. Please silence your cell phone and refrain from talking and texting. Do not use your cell phone, music player, and laptop during class, except for the purpose of taking notes. If you must take an urgent call, please leave the room quickly and quietly.

Disability policy: Students who require disability accommodations for this class are encouraged to notify the instructor at the beginning of the semester during office hours or by appointment. Prior to receiving disability accommodations, verification of eligibility from the Disability Resource Center (DRC) is required. If you feel that you will need accommodations but have not yet registered with the DRC, please contact the DRC as soon as possible by telephone at (480) 965-1234, by email at DRC@asu.edu, or in person on the first floor of the Matthews Center building. Disability information is confidential.

Excused absence policy: If you will be absent from class due to a religious observance or practice, or due to participation in a university sanctioned event or activity, then it is your responsibility to inform the instructor during the first week of class. Your instructor will work with you to make reasonable alternative arrangements for time missed.

Policy against threatening behavior: All incidents and allegations of violent or threatening conduct by an ASU student (whether on or off campus) must be reported to the ASU Police Department (ASU PD) and the Office of the Dean of Students. If either office determines that the behavior poses or has posed a serious threat to personal safety or to the welfare of the campus, the student will not be permitted to return to campus or reside in any ASU residence hall until an appropriate threat assessment has been completed and, if necessary, conditions for return are imposed. ASU PD, the Office of the Dean of Students, and other appropriate offices will coordinate the assessment in light of the relevant circumstances. Students may be withdrawn involuntarily from the course for disruptive classroom behavior at the instructor's discretion.

Title IX statement: Title IX is a federal law that provides that no person may be excluded on the basis of sex from participation in, be denied benefits of, or be subjected to discrimination under any education program or activity. Both Title IX and university policy make clear that sexual violence and harassment based on sex is prohibited. Individuals who believe that they have been subjected to sexual violence or harassed on the basis of sex can seek support, including counseling and academic support, from the university. If you or someone you know has been harassed on the basis of sex or sexually assaulted, you can find information and resources here. The instructor is obligated to report any information of which he becomes aware regarding alleged acts of sexual discrimination, including sexual violence and dating violence. At your option, you may discuss any such concerns confidentially and privately with ASU Counseling Services.

Disclaimer: The information on this syllabus is subject to change. Any changes will be announced in class, and an updated syllabus will be uploaded to Blackboard.

Selected Course Resources

If you are a member of the Society for Industrial and Applied Mathematics (SIAM), then you may purchase SIAM titles at a discount. SIAM titles are available electronically through the ASU Library.

Software for scientific computing

The selection is vast and depends on the application area, but here are a few resources to get you started.

- The Free Software Foundation produces compilers for C, C++, and Fortran, among others. The current release is Version 8.2.
- *The LAPACK User's Guide* and the *LAPACK95 User's Guide*. Print versions are available from SIAM.
- T. A. Driscoll, Learning MATLAB, SIAM, 2009, is a concise introduction and is available electronically through the ASU Library.

References on floating-point arithmetic

Goldberg's article will be posted to the Blackboard site and is available in many places on the Web. Overton's text is available electronically through the ASU Library.

- David Goldberg, "What Every Computer Scientist Should Know about Floating-Point Arithmetic," *ACM Comput. Surv.* **23** (1991), 5–48.
- Michael L. Overton, Numerical Computing with IEEE Floating Point Arithmetic. Philadelphia: Society for Industrial and Applied Mathematics, 2001.
- J.-M. Muller, N. Brisebarre, F. de Dinechnin, *et al.*, *Handbook of Floating-Point Arithmetic*, Boston: Birkhäuser, 2010.

Shells and scripting languages

There are many; here are references to a few.

- E. Quigley, *Unix Shells by Example*, 4th Ed. New York: Prentice-Hall, 2005.
- Bash Reference Manual, which summarizes the POSIX shell most commonly found on Linux systems.
- R. L. Schwartz, B. D. Foy, and T. Phoenix, *Learning Perl*, 6th Ed. Sebastopol, CA: O'Reilly Media, Inc., 2011. Tutorials and additional resources on Perl may be found here.
- The Python Software Foundation provides tutorials and online resources for the eponymous language.
- Online manual for Julia. The software may be downloaded here. Additional resources for learning Julia may be found here.
- Avoid the C and Tcsh shells: Csh Programming Considered Harmful; Top Ten Reasons Not to Use the C Shell.

References on parallel programming

The first three references address multicore programming (symmetric multiprocessing); the rest describe the MPI library.

- C. Breshears, *The Art of Concurrency*. Sebastopol, CA: O'Reilly Media, Inc., 2009.
- B. Chapman, G. Jost, and R. van der Pas, *Using OpenMP*. Cambridge, MA: The MIT Press, 2008.
- Additional resources may be found on the OpenMP website.
- W. Gropp, E. Lusk, and A. Skjuellum, *Using MPI: Portable Parallel Programming with the Message-Passing Interface*, 2nd ed. Cambridge, MA: The MIT Press, 1999.
- M. Snir, S. Otto, S. Huss-Lederman, D. Walker, and J. Dongarra, *MPI: The Complete Reference*, Vols. 1 and 2. Cambridge, MA: The MIT Press, 1998. Volume 1 is available online.

References on the Fortran language

The first two are very introductory. The last two texts are primarily reference works but are useful once you know the basics of Fortran.

- W. S. Brainerd, *Guide to Fortran 2008 Programming*. London: Springer-Verlag, 2015.
- I. Chivers and J. Sleightholme, *Introduction to Programming with Fortran*, 3rd ed. London: Springer-Verlag, 2015. A fourth edition will appear soon.
- R. W. Numrich and J. K. Reid, *CoArrays: Parallel Programming in Fortran*. Chapman and Hall, 2012.

- M. Metcalf, J. Reid, and M. Cohen, *Modern Fortran Explained*, 4th ed. Oxford: Oxford University Press, 2011; corrected printing, 2013.
- J. C. Adamas, W. S. Brainerd, *et al.*, *The Fortran 2003 Handbook: The Complete Syntax, Features, and Procedures.* London: Springer-Verlag, 2009.

References on the C language

Both texts cover the first (1989) C language standard. There were two revisions in 1999 and 2011, but the core language is still the same. Plauger's text provides portable C implementations of every function in the C89 standard library, and he discusses many of the issues in the design of good library software. A Google search will return dozens of sources for learning C.

- B. W. Kernighan and D. M. Ritchie, *The C Programming Language*, 2nd ed. Englewood Cliffs, NJ: Prentice Hall, 1988.
- P. J. Plauger, *The Standard C Library.* Englewood Cliffs, NJ: Prentice Hall, 1992.

References on the C++ language

The first two books are introductory. The third is a comprehensive reference that covers the 2011 standard. Meyers' books are more advanced and illustrate many of the difficulties in writing robust C++ code.

- A. Koenig and B. E. Moo, *Accelerated C++*. Reading, MA: Addison-Wesley, 2000.
- B. Stroustrup, *Programming: Principles and Practice Using C++*, 2nd ed. Reading, MA: Addison-Wesley, 2014.
- B. Stroustrup, *The C++ Programming Language*, 4th ed. Reading, MA: Addison-Wesley, 2013.
- S. Meyers, *Effective C++*, 3rd ed. Boston: Pearson Education, Inc., 2008.
- S. Meyers, *Effective Modern C++*. Sebastopol, CA: O'Reilly Media, 2014.
- S. Meyers, *Effective STL*. Boston: Pearson Education, Inc., 2008.

Other parallel languages

- Chapel is a parallel computing language under development at Cray Inc.
- X10 is an open-source effort by IBM Research to implement an asynchronous partitioned global address space (APGAS) language.
- Unified Parallel C (UPC) is an extension of the C programming language for high-performance parallel computing, developed by the University of California, Berkeley and the Lawrence Berkeley National Laboratory.

Software development

- S. Oliveira and D. Stewart, *Writing Scientific Software: A Guide to Good Style*. Cambridge University Press, 2006.
- Resources for git, a popular version control system.