**CS/MATH 4414 Issues in Scientific Computing**
**MW 14:30–15:45 SURGE 109** **CRN 19573/19574**

**Instructor**: L. T. Watson, 2000 Torg, `laynew@acm.org`

**Office Hours**: MW 13:15–14:15, and by appointment.

**Text**: M. T. Heath, *Scientific Computing: An Introductory Survey*, 2nd Ed., McGraw-Hill, New York, 2002.

**Topics Covered**: `Supercomputing hardware` — floating point arithmetic (IEEE 754 Standard), parallel and vector hardware, computational models, network topologies, concurrency/communication tradeoffs; `algorithms` — B-splines, nonlinear equations, quasi-Newton methods, homotopy methods, constrained optimization, sparse iterative methods, approximation theory, genetic and evolutionary methods, cellular automata; `software tools` — ACM TOMS Algorithms, mathematical libraries, programming languages, Fortran 90–2008, OpenMP, MPI-2, Mathematica, TEX, HOMPACK90, VTDIRECT95, QNSTOP, problem solving environments, visualization packages.

**Grading**: `Final grade` will be comprised from (minor and major) homework assignments (60%) and a final presentation and project (40%).

**Midterm Exam**: None.

**Final Exam**: final project due Wednesday, 10:05, May 9, 2018.

**Final Project**: The purpose of the project is to give you a serious taste of scientific computing, by involving as many of the aspects below as are reasonable for the chosen project. Your project need not incorporate all aspects, but the more the better. The aspects are (1) **modelling** (deep understanding of a physical model, reflected in the algorithms and software tools used with the model), (2) **parallel computing** (nontrivial use of parallel hardware), (3) **numerical algorithms** (well researched choice and serious use of state-of-the-art mathematical software—a subroutine from *Numerical Recipes* or a built-in Mathematica or Matlab function is not sufficient here), (4) **presentation of results** (color graphics, animation, live demos).

The project shall be described in a report of length from 8 to 15 pages, including figures and bibliography, but not including source code or demos. The style should be that of a SIAM journal article, and good writing (spelling, grammar, punctuation, organization, clarity) is equally important to good technical content.

### References:

S. G. Akl, *The Design and Analysis of Parallel Algorithms*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

G. Almasi and A. Gottlieb, *Highly Parallel Computing*, Benjamin-Cummings, Redwood City, CA, 1989.

D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

M. Cosnard and D. Trystram, *Parallel Algorithms and Architectures*, International Thomson Computer Press, London, 1995.

R. E. Crandall, *Projects in Scientific Computation*, Springer-Verlag, New York, 1994.

R. E. Crandall, *Topics in Advanced Scientific Computation*, Springer-Verlag, New York, 1996.

J. J. Dongarra, I. S. Duff, D. C. Sorensen, and H. van der Vorst, *Solving Linear Systems on Vector and Shared Memory Computers*, SIAM, Philadelphia, 1990.

L. D. Fosdick, E. R. Jessup, C. J. Schauble, and G. Domik, *An Introduction to High-Performance Scientific Computing*, MIT Press, Cambridge, MA, 1996.

I. Foster, *Designing and Building Parallel Programs*, Addison Wesley, Reading, MA, 1995.

K. Gallivan, M. Heath, E. Ng, B. Peyton, R. Plemmons, J. Ortega, C. Romine, A. Sameh, and R. Voigt, *Parallel Algorithms for Matrix Computations*, SIAM, Philadelphia, 1990.

G. Golub and J. M. Ortega, *Scientific Computing: An Introduction with Parallel Computing*, Academic Press, Boston, 1993.

T. A. Grandine, *The Numerical Methods Programming Projects Book*, Oxford University Press, Oxford, 1990.

M. T. Heath, *Scientific Computing: An Introductory Survey*, McGraw-Hill, New York, 1997.

R. W. Hockney and C. R. Jesshope, *Parallel Computers 2*, Adam Hilger, Bristol, 1988.

E. Isaacson and H. B. Keller, *Analysis of Numerical Methods*, Dover, Mineola, NY, 1994.

J. JáJá, *An Introduction to Parallel Algorithms*, Addison Wesley, Reading, MA, 1992.

H. Jordan and G. Alaghband, *Fundamentals of Parallel Processing*, Prentice Hall, Upper Saddle River, NJ, 2003.

V. Kumar, A. Grama, A. Gupta, and G. Karypis, *Introduction to Parallel Computing*, Benjamin/Cummings, Redwood City, CA, 1994.

F. T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*, Morgan Kaufmann, San Mateo, CA, 1992.

T. G. Lewis and H. El-Rewini, *Introduction to Parallel Computing*, Prentice-Hall, Englewood Cliffs, NJ, 1992.

T. G. Lewis, *Foundations of Parallel Programming*, IEEE Computer Society Press, Los Alamitos, CA, 1993.

L. P. Meissner, *Fortran 90*, PWS, Boston, 1995.

D. I. Moldovan, *Parallel Processing: From Applications to Systems*, Morgan Kaufmann, San Mateo, CA, 1992.

J. M. Ortega, *An Introduction to Fortran 90 for Scientific Computing*, Saunders, Fort Worth, TX, 1994.

M. J. Quinn, *Designing Efficient Algorithms for Parallel Computers*, McGraw-Hill, New York, 1987.

M. J. Quinn, *Parallel Computing: Theory and Practice*, McGraw-Hill, New York, 1994.

G. W. Sabot, *High Performance Computing*, Addison Wesley, Reading, MA, 1995.

L. F. Shampine, R. C. Allen, Jr., and S. Pruess, *Fundamentals of Numerical Computing*, Wiley, New York, 1997.

J. A. Sharp, *An Introduction to Distributed and Parallel Processing*, Blackwell, Oxford, 1987.

M. Snir, S. Otto, S. Huss-Lederman, D. Walker, J. Dongarra, *MPI, the Complete Reference: the MPI Core*, MIT Press, Cambridge, MA, 1998.

M. Snir, S. Otto, S. Huss-Lederman, D. Walker, J. Dongarra, *MPI, the Complete Reference: the MPI-2 Extensions*, MIT Press, Cambridge, MA, 1998.

G. V. Wilson, *Practical Parallel Programming*, MIT Press, Cambridge, MA, 1994.

**Web references:**

NonCS students can get an account on rlogin.cs.vt.edu by going to

`https://admin.cs.vt.edu/create.pl`

Numerical analysis notes, a bibliography, and Fortran material (how to access the Fortran compilers `gfortran` or (Intel) `ifort`, compilation scripts) are at

`http://courses.cs.vt.edu/~cs3414/F17/`

The shared memory multicore machines `fir.rlogin` and `sourwood.rlogin` have 64 cores, and `gfortran` with OpenMP. The machine `fortran.rlogin` has 20 (40 hyperthreaded) cores, with `gfortran`, Intel `ifort`, OpenMP, and MPI.

For scientific problem solving environments see

`http://research.cs.vt.edu/pse/`

Tutorials for MPI with Fortran are

`https://www.cs.kent.ac.uk/people/staff/trh/MPI/mpitutorial.pdf`

`http://www.mcs.anl.gov/research/projects/mpi/tutorial/mpiintro/ppframe.htm`

with local linking instructions at

`http://wordpress.cs.vt.edu/rlogin/how-to-use-mpi/`