COSC 594 – 007
3 credit hours

Scientific Computing for Engineers

Web page for the course:
CS 594

Wednesday’s 1:30 – 4:30

- **Scientific Computing for Engineers**
- **Spring 2019 - 3 credits**
  - ➢ Jack Dongarra
  - ➢ with help from:
    » George Bosilca
    » Mark Gates
    » Azzam Haidar
    » Jakub Kurzak
    » Heike Jagode
    » Nuria Losada
    » Piotr Luszczek
    » Stan Tomov

- **Class will meet in Room C-233, Claxton Building**
To Get Hold of Us

- **Email:** dongarra@icl.utk.edu
  - Room: 203, Claxton
  - Phone: 974-8295

- **Office hours:**
  - Wednesday 11:00 - 1:00, or by appointment

- **TA:** Jiali Li: jli111@vols.utk.edu

- **TA’s Office:** Claxton 304
Four Major Aspects Of The Course:

1. **Start with current trends in high-end computing systems and environments, and continue with a practical short description on parallel programming with MPI, OpenMP, and pthreads.**

2. **Deal with numerical linear algebra solvers: both direct dense methods and direct and iterative methods for the solution of sparse problems. Algorithmic and practical implementation aspects will be covered.**

3. **Illustrate the modeling of problems from physics and engineering in terms of partial differential equations (PDEs), and their numerical discretization using finite difference, finite element, and spectral approximation.**

4. **Various software tools will be surveyed and used. This will include PETSc, Sca/LAPACK, MATLAB, and some tools and techniques for scientific debugging and performance analysis.**
Grades Based on:

- 40% on weekly assignments (the lowest grade will be dropped)
- 40% on a written report (15-20 pages) and presentation.
- 20% on a final exam (2 hours) & on class participation.
Homework

- Usually weekly
- Lowest grade will be dropped
- Must be turned in on time (no late assignments accepted)
- Don’t copy someone else’s work.
- Sometimes problems, sometimes programming assignments, sometimes requiring running a program to find the solution.
Homework (continued)

- We expect an analysis and detailed discussion of the results of your efforts.
  - The program itself is not very interesting.
- Programming in C or Fortran.
  - If you don’t know C or Fortran could be a problem.
- Will go over the assignments the week they are due.
- See class web page weekly for details.
Project

- Topic of general interest to the course.
- The idea is to read three or four papers from the literature (references will be provided)
- Implement the application on the cluster you build
- Synthesize them in terms of a report (~15-20 pages)
- Present your report to class (~30 mins)
- New ideas and extensions are welcome, as well as implementation prototype.
- Could be from your dissertation.
Remarks

- Hope for very interactive course
- Willing to accept suggestions for changes in content.
Final Exam

- In class
- Will cover the material presented in the course
- ~2 hours
Material

- For each lecture a set of slides will be made available in pdf or html.
- Other reading material will be made available electronically if possible.
- The web site for the course is:
Important Place for Software

- Netlib - software repository
  ➢ Go to http://www.netlib.org/
What will we be doing?

◆ **Learning about:**
  ➢ High-Performance Computing.
  ➢ Parallel Computing
  ➢ Performance Analysis
  ➢ Computational techniques
  ➢ Tools to aid parallel computing.
  ➢ Developing programs in C or Fortran using MPI and OpenMP.
Outline of the Course

1. January 9th  Class Introduction & Introduction to High Performance Computing
2. January 16th  Parallel programming paradigms and their performances
3. January 23rd  Introduction to MPI
4. January 30th  MPI
5. February 6th  Advanced MPI & OpenSHEM
6. February 13th  Modern Directive Programming with OpenMP and OpenACC
7. February 20th  Accelerators
8. February 27th  Performance Modeling & PAPI
9. March 6th  Dense Linear Algebra
10. March 13th  Dense Linear Algebra
11. March 20th Spring Break
12. March 27th  Machine Learning with Deep Neural Networks
13. April 3rd  Projection and its importance in scientific computing & GPU Computing
14. April 10th  Discretization of PDEs and Parallel Solvers
15. April 17th  Sparse Matrices and Optimized Parallel Implementations
16. April 24th  Iterative Methods in Linear Algebra Part 1
17. May 1st  Final and Reports
What you should get out of the course

In depth understanding of:
- Why parallel computing is useful.
- Understanding of parallel computing hardware options.
- Overview of programming models (software) and tools.
- Some important parallel applications and the algorithms
- Performance analysis and tuning techniques.
Background

- C and/or Fortran programming.
- Good to have an understanding of parallel programming.
- Some background in numerical computing.