**Time and Place:** MW 3-3:50pm in DSH 333, 4-5:15pm in ESCP 110  
**Professor:** Monika Nitsche, nitsche@math.umn.edu, 277-5039  
**Course Web Site:** http://www.math.umn.edu/~nitsche/math471.html  
**Office Hours:** MWF 10-11am, Humanities 465

**Description:** This is an introductory course on scientific computing fundamentals. The topics included are: parallel programming, performance evaluation (operation counts, timing, memory requirements, parallel efficiency and scalability), discretization and roundoff errors, and convergence and stability of numerical methods (time stepping schemes, finite difference approximations), the Discrete Fourier Transform and, time permitting, fast summation algorithms. These topics will be introduced using applications from celestial mechanics, heat transfer and signal processing.

Note: This course is available for graduate credit. It is also one of the two core requirements for the Graduate Certificate in Computational Science and Engineering.

**Prerequisite:** Vector calculus, elementary linear algebra and good programming skills in Fortran, C or C++. Some Matlab.

**Computer Projects:** Computer projects to be completed in the lab are an integral part of the course. Each project will require a typed report, with an introduction, description of the problem and solution techniques, results and a conclusion. Plots should be used whenever possible as opposed to tables of numbers. We will be working on a 16-node Apple cluster, “hammer”.

**Course outline:**

1. **ODE’s – The N-body problem**  
   - History and equations  
   - Euler and Runge-Kutta methods  
   - Convergence, roundoff error, stability, FLOPS  
   - *Project (in C, C++, or Fortran):* N-body problem

2. **Parallel programming**  
   - MPI: Message passing interface  
   - Point-to-point and collective operations  
   - Latency and bandwidth  
   - Parallel I/O  
   - Performance evaluation  
   - *Project (Parallel):* A simple example. Solve N-body problem in parallel.

3. **Poisson equation**  
   - Finite difference approximations  
   - Direct solvers: LU (memory requirements, sparseness)  
   - Iterative solvers: Jacobi, Gauss-Seidel, Conjugate gradient  
   - *Computer project:* implement iterative methods in parallel

4. **Discrete Fourier Transform**  
   - Basics, FFT  
   - spectral methods: solving Poisson Equation with periodic BCs  
   - solving the Poisson Equation with nonperiod BCs  
   - applications to audio signal processing  
   - *Project:* scalability of parallel FFT, one application

5. **CFL condition**  
   - Scalar advection equation, method of characteristics  
   - Heat equation, finite difference approximations, von Neumann stability analysis

6. **Fast summation algorithm (time permitting)**  
   - Barnes Hut algorithm and improvements