SC250 Scientific Computing Toolbox
Tue/Thu 1:10pm-2:25pm
Featheringill Hall 211

Instructors

Dr. Robert Bodenheimer
Department of Computer Science
368 Jacobs Hall
322-3555
bobby.bodenheimer@vanderbilt.edu
Office Hours: Mon 1:00-2:00, Tues 2:30-3:30, or by appointment

Dr. Thomas Palmeri
Department of Psychology
507 Wilson Hall
343-7900
thomas.j.palmeri@vanderbilt.edu
Office Hours: Tue 12:00-1:00 or by appointment

Dr. Greg Walker
Department of Mechanical Engineering
335 Olin Hall
343-6959
greg.walker@vanderbilt.edu
Office Hours: Tue and Thu 3:00-4:00 or by appointment

Course Overview

An astronomer studying the formation of massive black holes, an economist studying complex financial markets, a neuroscientist studying brain networks for human memory, a chemist studying the structure of large proteins, and an engineer designing new nanostructured materials could not appear be more different. Their research involves vastly different forces that govern physical, biological, or social interactions, for structures with spatial scales ranging from subatomic to extragalactic and timescales ranging from picoseconds to gigayears. Yet, from a computational standpoint, the astronomer, economist, neuroscientist, chemist, and engineer face similar challenges in working to understand the behavior of complex systems.

This course introduces some of the scientific computing tools used by scientists and engineers to understand complex physical, biological, and social systems. Students may be introduced to numerical and computational methods for simulating models of complex systems, techniques for optimizing and evaluating models, scientific visualization and data mining techniques for detecting structure in large multidimensional data sets, and high performance computing techniques for simulating models and analyzing data.

This is a multidisciplinary team taught course. The three core instructors come from Computer Science, Psychology and Neuroscience, and Mechanical Engineering. Guest lecturers may come from other disciplines, including Astronomy, Biology, Biomedical Engineering, Chemistry, Chemical Engineering, and Physics. The course illustrates scientific computing tools within the context of a particular scientific or engineering domain reflecting the expertise of the faculty.

Prerequisites

Introductory computer programming (CS101 or CS103 or equivalent) and one semester of calculus (MATH150A or equivalent) are required. Students should have taken at least one college-level science or engineering courses before enrolling in this course. We do not assume background in any particular science or engineering discipline apart from what would have been learned prior to coming to college.

Course Requirements and Grading

Homework assignments (90%) handed out once or twice a week will be used throughout the course to allow students the opportunity to put the scientific computing tools into practice. There will be no exams. Attendance and class participation (10%) are also expected. Final letter grades will based on percentages as follows:

		A	92.5 – 100%	A-	90.0 – 92.5%
B+	87.5 – 90.0%	B	82.5 – 87.5%	B-	80.0 – 82.5%
C+	77.5 – 80.0%	C	72.5 – 77.5%	C-	70.0 – 72.5%
D+	67.5 – 70.0%	D	62.5 – 67.5%	D-	60.0 – 62.5%
		F	0.0 – 60.0%		

All homework assignments must be completed individually. Unexcused late assignments will be penalized 10% for every 24 hours late, starting from the time class ends, for a maximum of two days, after which they will earn a 0. Individual grades from each of the three modules (taught respectively by Profs. Bodenheimer, Palmeri and Walker) will be averaged to calculate a final grade.

Any student auditing the course is expected to attend every class.

Python

The Python programming language will be used for all assignments in this course. We assume no prior knowledge of Python and will provide an introduction to Python programming in the course.

Python is a high-level computer programming language particularly well suited to Scientific Computing applications. It is free, open software that runs on multiple platforms (Windows, Mac, and Linus). It is highly extensible with thousands of libraries and modules written and shared by scientists and engineers from around the world. It allows easy interface with programs written in languages like C, Fortran, Java, or Matlab.

Details for installing Eclipse, Python, PyDev, and required Python modules are provided on OAK and will be discussed in class.

We will only scratch the surface of Python in this course. A far more extensive introduction to Python is given in *CS204 Program Design and Data Structures for Scientific Computing*.

OAK

We will use OAK (www.vanderbilt.edu/oak). It will contain the detailed up-to-date schedule of topics and assignments that you should consult regularly. Below we provide only a very rough outline of the topics and their instructors over the course of the semester.

There is no textbook for this course. Copies of all readings, homework assignments, and handouts will be on OAK. Powerpoint slides and code from Python demonstrations will be on OAK sometime soon after each class.

Course Schedule

The following course schedule is subject to change. The most up-to-date schedule will be posted on OAK.

<u>Week 1</u>

Thu, Aug 22 Introduction to the Course

Introduction to Python

<u>Week 2</u>

Tue, Aug 27 Introduction to Python: Syntax, Control Statements, Lists, Strings

Thu, Aug 29 Introduction to Python: Functions, Methods, NumPy, SciPy, Matplotlib, graphing

<u>Week 3</u>

Tue Sep 3 Introduction to Python: Classes, Reading and writing files

Thu, Sep 5 Introduction to Visualization: Graphs and Data Representation

Scientific Visualization

<u>Week 4</u>

Tue, Sep 10 Introduction to Scientific Visualization: Continuous and Discrete Data

Thu, Sep 12 Introduction to Scientific Visualization: Sampling

Week 5

Tue Sep 17 Introduction to Scientific Visualization: Reconstruction

Thu Sep 19 Guest Lecture: Medical Engineering (Prof. Michael Miga)

Simulations: How the Brain Makes Decisions

Week 6

Tue Sep 24 How the Brain Makes Decisions

Thu Sep 26 The Neuron

Week 7

Tue Oct 1 Calculus Review

Thu Oct 3 Modeling the Neuron

Week 8

Tue Oct 8 Simple Differential Equations

Thu Oct 10 FALL BREAK

Week 9

Tue Oct 15 Random Numbers

Thu Oct 17 Introduction to Monte Carlo Simulation

Week 10

Tue Oct 22 Guest Lecture: Computing Protein Structures (Prof. Jens Meiler)

Thu Oct 24 Guest Lecture: Multi-scale modeling (Prof. Caglar Oskay)

N-Body Simulation: Electrical Properties of Nanostructures

Week 11

Tue Oct 29 Projectiles and laws of motion

Thu Oct 31 Gas in a box (molecular dynamics)

Week 12

Thu Nov 5 Coupled motion (molecular dynamics)

Tue Nov 7 Integrators (molecular dynamics)

Week 13

Tue Nov 12 Guest Lecture: N-Body Simulation in Astronomy
(Prof. Andreas Berlind)

Thu Nov 14 Guest Lecture: Biomolecular modeling (Prof. Peter Cumming)

Week 14

Tue Nov 19 Electrons and multiple scattering mechanisms

Thu Nov 21 Phonons (Boltzmann transport)

Thanksgiving Break

Week 15

Tue Dec 3 Light and collecting statistics

Thu Dec 5 Guest Lecture: Scientific Computing with Massive Data
(Prof. Paul Sheldon)

Vanderbilt's Honor Code Governs All Work in this Course