Scientific Computing: Special Topic course
CSE 4501/5401-E11

Instructor: Debasis Mitra
Time: TR – TBD-
E-mail: dmitra ‘at’ cs.fit.edu
Class Home Page: http://www.cs.fit.edu/~dmitra/SciComp/
Office Hours: TBA

Prerequisite: Programming skill in a procedural language (not necessarily Matlab). C++ will be the primary vehicle but we may use Python as well. Languages will not be covered in the course but should be picked up by the students on their own. Undergraduate students must have ≥3.4 GPA (or, talk to instructor). The following two math courses provide good background for this course. However, they are not prerequisites.

MTH4311: Introduces numerical methods for solving equations in one variable, polynomial approximation, interpolation, numerical differentiation and integration, initial-value problems for ODE and direct methods for solving linear systems. Prerequisite: CSE 1502 or CSE 1503 or CSE 2050, MTH 2201
MTH5301: Includes Gaussian elimination and solution of linear systems of equations, root finding methods, systems of nonlinear equations, interpolation, numerical integration, initial value problems for ODEs and fast Fourier transform. Prerequisite: CSE 1502 or CSE 1503 or CSE 2050, MTH 2201

Modern science and engineering practices depend on computation. In a spiral fashion, the availability of faster computers is fueling a data explosion in all branches of these disciplines, and in turn demanding better computing resources. Developing efficient algorithms is a part of the equation. This course covers such mathematical algorithms. The algorithms in this area are primarily numerical. However, we go beyond theoretical understanding of numerical analysis, and code some of the algorithms in order to understand their structures and resource usage patterns. From this course the students are expected to become skilled in programming mathematical algorithms that are important for scientists and engineers today.

STUDENTS FROM ANY SCIENCE, ENGINEERING AND MATH DISCIPLINES ARE WELCOME.

We will go over hands-on coding (in C++) from the source provided by this book and do some in-depth realistic projects beyond the scope of the text (e.g. motion compensation in images, topological data analysis, deep learning algorithms, etc.). Tentative topics from the book are provided below.

<table>
<thead>
<tr>
<th>Weeks</th>
<th>Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week 1-3</td>
<td>Intro to computing</td>
</tr>
<tr>
<td></td>
<td>Linear Algebra: ch 2</td>
</tr>
<tr>
<td>Week 4-6</td>
<td>Interpolation: ch 3</td>
</tr>
<tr>
<td>Week 7-9</td>
<td>Optimization: ch 10</td>
</tr>
<tr>
<td>Week 10-12</td>
<td>Eigensystems: ch 11</td>
</tr>
<tr>
<td>Week 12-14</td>
<td>Projects presentation</td>
</tr>
</tbody>
</table>

**Evaluation:**
Class participation, Student presentation, Demo,
Project results and report.
No final exam

**Standard Class Policy:**
Copying, plagiarizing and unauthorized collaboration will be considered as cheating, which may lead to an ‘F’ grade in the class, and other disciplinary actions subject to the Departmental and University policies. Any question about the graded class materials should be raised within two class periods after the graded material is returned to the students. Examinations are announced normally one week prior to the exam date. No make up tests will be given. Students may be asked to attend lectures outside the normal class schedule. Physically challenged students needing any special assistance should consult the instructor. University policy allows a student to be absent from the class on any special religious day for the respective student, provided the instructor is informed at the beginning of the semester.