CS 3513: Numerical Methods for Digital Computers

Required Course: Required
Course Number: CS 3513
Course Name: Numerical Methods for Digital Computers
Credit Hours: 3
Lecture Hours: 3
Lab Hours: 0
Instructors: Dr. Thanh Thieu

Book Title(s): Elements of Numerical Computation
Book Author(s): J.P. Chandler
Book Year(s): 2010

Book Title(s): Numerical Mathematics and Computing, 7th edition
Book Author(s): W.Cheney and D.Kincaid
Book Year(s): 2013

Book Title(s): Numerical Methods, 4th edition
Book Author(s): J D Faires and R L Burden
Book Year(s): 2012

Course Description: Errors, floating point numbers and operations, interpolation and approximation, solution of nonlinear equations and linear systems, condition and stability, acceleration methods, numerical differentiation and integration.

Course Prerequisites: MATH 2153 (Calculus II(A)); MATH 3013 (Linear Algebra (A)) or concurrent enrollment; or MATH 3263(Linear Algebra and Differential Equations) and knowledge of programming.

Course Goals:
- Understand different types of generated errors in computation
- Recognize machine representation of floating-point numbers and associated properties
- Apply Taylor expansion and approximate function derivatives
- Program iterative algorithm to solve non-linear equations
- Apply acceleration methods to increase convergence of iterative algorithms
- Analyze condition of a problem and stability of an algorithm
- Implement Gaussian elimination to solve systems of linear equations
- Understanding of interpolation methods and difficulty in extrapolation
- Calculate definite integral using numerical approximation
Student Outcomes:

<table>
<thead>
<tr>
<th>Student Outcome</th>
<th>Course Outcome</th>
</tr>
</thead>
</table>
| 1               | • Identify segments of the input domain that ill-condition the problem or destabilize the algorithm  
|                 | • Transform the original problem formulation into a solvable form, including non-linear, linear, or Taylor approximation |
| 2               | • Implement algorithms that solve non-linear equations, system of linear equations, and convergence acceleration  
|                 | • Evaluate accuracy of the solution given the platform-specific numerical representation |
| 3               | • Follow programming guidelines to write code that communicates its purpose clearly and effectively, using comments and self-documenting variables, methods and organization |
| 4               | • Understand how generated errors in unstable algorithms can have dangerous or fatal consequences, and use transformation strategies to redesign the algorithm into a stable version |
| 6               | • Practice code reuse and encapsulation in implementation of root finding methods  
|                 | • Analyze the trade-offs between speed, resource use, and accuracy of the computed solution |

Course Topics:

Knowledge areas that contain topics and learning outcomes covered in the course

<table>
<thead>
<tr>
<th>Knowledge Area</th>
<th>Total Hours of Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithms and Complexity (AL)</td>
<td>2</td>
</tr>
<tr>
<td>Architecture and Organization (AR)</td>
<td>3</td>
</tr>
<tr>
<td>Computational Science (CN)</td>
<td>27</td>
</tr>
<tr>
<td>Parallel and Distributed Computing (PD)</td>
<td>1.5</td>
</tr>
<tr>
<td>Programming Languages (PL)</td>
<td>1</td>
</tr>
<tr>
<td>Software Engineering (SE)</td>
<td>2</td>
</tr>
<tr>
<td>Social Issues and Professional Practice (SP)</td>
<td>0.5</td>
</tr>
</tbody>
</table>
### Body of Knowledge coverage

<table>
<thead>
<tr>
<th>Knowledge Area</th>
<th>Knowledge Unit</th>
<th>Topics</th>
<th>Hours of coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>AL</td>
<td>Basic Analysis</td>
<td>Empirical measurements of performance, Time and space trade-offs in algorithms</td>
<td>2</td>
</tr>
<tr>
<td>AR</td>
<td>Machine Level Representation of Data</td>
<td>Fixed- and floating-point systems</td>
<td>3</td>
</tr>
<tr>
<td>CN</td>
<td>Processing</td>
<td>Numerical methods</td>
<td>10</td>
</tr>
<tr>
<td>CN</td>
<td>Numerical Analysis</td>
<td>Error, stability, convergence, function approximation including Taylors series, interpolation, extrapolation, and regression, Numerical differentiation and integration, Differential equations</td>
<td>17</td>
</tr>
<tr>
<td>PD</td>
<td>Parallel Decomposition</td>
<td>Independence and partitioning, Data-parallel decomposition</td>
<td>0.5</td>
</tr>
<tr>
<td>PD</td>
<td>Parallel Algorithms, Analysis, and programming</td>
<td>Parallel matrix computations</td>
<td>1</td>
</tr>
<tr>
<td>PL</td>
<td>Advanced programming constructs</td>
<td>Lazy evaluation, language support for checking assertions, invariants, and pre/post-conditions</td>
<td>1</td>
</tr>
<tr>
<td>SDF</td>
<td>Algorithms and Design</td>
<td>Problem-solving strategies, iterative and recursive numerical algorithms, divide and conquer strategies</td>
<td>2</td>
</tr>
<tr>
<td>SDF</td>
<td>Development methods</td>
<td>Testing fundamentals and test case generation</td>
<td>1</td>
</tr>
<tr>
<td>SE</td>
<td>Tools and Environment</td>
<td>Version control systems</td>
<td>1</td>
</tr>
<tr>
<td>SP</td>
<td>History</td>
<td>History of numerical computation algorithms</td>
<td>0.5</td>
</tr>
</tbody>
</table>