CSC 335[1] / F17

Numerical Methods

Prof. Andrew J. Pounds, Ph.D.

Office:	Rm. 105 Willett Science Center, (478) 301-5627 Rm. 203, CSB
e-mail:	pounds_aj@mercer.edu
Alt Phone (SMS):	(478) 227-3444
Office Hours:	T 9:15 a.m. – 10:15 a.m., 3:00 p.m. – 5:00 p.m. (WSC 105) (or by appointment)

Computing is not about numbers, it is about insight. – R.W. Hamming

CSC 335 is a course to introduce students to the foundational principles of numerical methods and numerical analysis. Students in CSC 335 will be exposed to the fundamentals of finite precision arithmetic, its associated error, and the propagation of such errors. Students will also be introduced to methods for solving equations in one and multiple variables by using direct and iterative techniques. Methods of numerical differentiation and integration will also be studied. The numerical solution of ordinary differential equations will be be introduced as well as the use of approximation theory to solve nonlinear systems. The use of public domain libraries and packages for numerical computation will be demonstrated. Commercial symbolic algebra packages like *Mathematica* and *MuPad* will also be used in the class.

Students taking CSC 335 are expected to have a working knowledge of a high level programming language such as C, C++, Fortran, or JAVA. Strategies for high performance code development and project management with these languages will be discussed during the course. Rapid code prototyping utilizing the Python programming language will also be introduced. Students in CSC 335 are expected to read at the college level and also have a working knowledge of differential and integral calculus. Topics from multivariable calculus, linear algebra, and differential equations will be introduced as needed.

Upon completion of this course, a student will demonstrate competence in each of the following areas:

- understanding the errors incurred when one utilizes finite precision arithmetic and applying these concepts on homework and tests,
- the ability to development direct and iterative software algorithms to solve well formed problems from the mathematics literature and applying these concepts on homework and tests,
- the ability to write numerical programs suitable for solving problems from the natural and physical sciences and applying these skills to programming projects.

Class Meeting Times and Locations

Lecture: TR 1:40–2:55 p.m., WSC 310 (sometimes CSC 306)

Course Materials

Required: *Numerical Analysis, 10th ed.*, Burden, Faires, and Burden
Scientific Calculator

Course Structure

Topics from nine chapters of the text will be covered during the semester in the order listed on the class schedule. The lecture time will be used to expound on and augment the text, discuss problem solving strategies, and demonstrate algorithms. Students will also be encouraged to write and test algorithme during the class. Students are responsible for all material covered in class as well as the material from the textual sections listed in the class schedule. Eight assignments and two projects will be submitted for grading. Three 70 minute exams will be given per the schedule. A three hour final exam will be administered at the end of the term.

[1] Cross listed as MAT 335

Grading

Tests (3 @ 100 pts)	300 pts
Programming Projects (2 @ 150 pts)	300 pts
Homework/programming assignments (8 @ 25 pts)	200 pts
Final Exam	200 pts
Total Possible	1000 pts

The following grading scale is assured but *may* be *slightly* lowered based on test results.

A	≥900 pts
B+	≥880 pts
B	≥800 pts
C+	≥780 pts
C	≥700 pts
D	≥600 pts
F	<600 pts

General Information

Honor Code: All students in CSC 335 are expected to adhere to the Mercer University Honor Code. Any suspected violations will be reported to the Honor Council for further investigation.

Many students have difficulty in determining how to apply the Mercer honor code to computer science courses. A few general guidelines should help you in deciding whether you are violating the honor code or not.

1. You are allowed to receive help on your programs from other students, provided the purpose of the help is to help you understand your own program better, not to write your program for you.
2. You are NOT allowed to use copies of programs written by other students, or copies of programs from published sources, even if you plan to modify them extensively.
3. You are NOT allowed to give copies of your programs, or parts of your programs, to other students in any form.
4. YOU MUST WRITE YOUR OWN CODE. DO NOT COPY PROGRAMS OR PARTS OF PROGRAMS FROM ANY SOURCE UNLESS I TELL YOU TO DO SO.

Any violation of the above policies will be treated as academic dishonesty and a violation of the Mercer Honor Code.

Attendance: Except for the first eight class periods of the semester, attendance will not be taken. However, students are still accountable for all material covered in class as well as any announcements made during the lecture period.

Homework: Students are encouraged to work together on homework assignments. Each person, however, must turn in their own assignments in their own words.

Homework Grading Policy: Homework problems will typically be divided into two sections: book problems and special problems from me. A selection of the assigned problems will be graded on a three point scale and the composite score from all problems on a given assignment converted to a 15 point scale. Points will be awarded as follows: 3 (essentially correct), 2 (minor errors or omissions), 1 (major errors or omissions), 0 (no effort). The point breakdown for problems from me will be indicated on the homework handouts and the composite score converted to a 10 point scale. Each homework assignment is worth a total of 25 points. Homework will be considered late if it is not turned in by 5:00 p.m. on the date due. Late homework will be penalized 33% per day.

Missed Exams: Anyone missing an exam for *any* reason (personal illness, death in the immediate family, or other emergency) must notify Dr. Pounds in advance. The absence will be considered unexcused otherwise. Make-up exams will be individually scheduled.

Partial Credit: Partial credit will not be awarded on any exam unless individuals show their work and clearly delineate how they arrived at their answers.

Re-grading Policy: If a student suspects that an error was made in the grading of a submitted work, they may return the paper for re-grading with the understanding that the entire work will be re-graded and not only the portion in question.

Posting of Grades: Grades will not be posted. If you are curious about your cumulative grade, see Dr. Pounds.

E-mail Listserve: I maintain an e-mail listserve which I use use copiously to send information to the class and which you can use to communicate with each other. To sign up for the listserve and to learn how to send information to it, please go to: `http://theochem.mercer.edu/mailman/listinfo/csc335`.

Departmental Computing Facilities: While students may work on their own computer systems to develop code for the class, students are encouraged to work on the departmental server named *cobra* (`cobra.cs.mercer.edu`). This server is guaranteed to have all of the requisite compilers and software libraries required to build code for this course. All programs will be tested on *cobra* – SO MAKE SURE YOUR CODE RUNS ON *cobra*. Students who need to use *Mathematica* can access it from the computer labs in the Computer Science Building or the Chemistry Department. If you need information on how to transfer code to *cobra*, just ask me.

Platforms and Editors: I get asked this question every year – so I am going to go ahead and answer it here. I don't care what editors you use to develop your code. You can use *Eclipse*, *CodeBlocks*, *Visual Studio*, *Emacs*, *vim*, *joe*, *gedit*, *jed*, etc. Just know that all of those are not available on all systems. You are GUARANTEED to find *vi* on any Unix box. Please remember that the FINAL version of your program will be tested by me on *cobra*. It therefore must be in a format that can be read and compiled in a normal manner. If you code ONLY runs inside *eclipse* then it does not meet these guidelines. In addition, you must provide directions, either in your code or in a README file that states how the code should be compiled and run on *cobra*.

Code Projects and GitLab: The department has a *Gitlab* server named `anvil.cs.mercer.edu` that students are STONGLY encouraged to use for the storage and development of their programs. For the two programming projects students are REQUIRED to submit thier code via `anvil`. Students are therefore encouraged to learn how to use the code versioning/management system as soon as possible. Please include a *README.md* file in your repository that describes how to compile and run your code.

American Disability Act: "Students requiring accommodations for a disability should inform the instructor at the close of the first class meeting or as soon as possible. The instructor will refer you to the ACCESS and Accommodation Office to document your disability, determine eligibility for accommodations under the ADAAA/Section 504 and to request a Faculty Accommodation Form. Disability accommodations or status will not be indicated on academic transcripts. In order to receive accommodations in a class, students with sensory, learning, psychological, physical or medical disabilities must provide their instructor with a Faculty Accommodation Form to sign. Students must return the signed form to the ACCESS Coordinator. A new form must be requested each semester. Students with a history of a disability, perceived as having a disability or with a current disability who do not wish to use academic accommodations are also strongly encouraged to register with the ACCESS and Accommodation Office and request a Faculty Accommodation Form each semester. For further information, please contact Katie Johnson, Director and ADA/504 Coordinator, at 301-2778 or visit the ACCESS and Accommodation Office website at http://www.mercer.edu/disabilityservices"

Electronic Submission of Materials: "Students bear sole responsibility for ensuring that papers or assignments submitted electronically to a professor are received in a timely manner and in the electronic format(s) specified by the professor. Students are therefore obliged to have their e-mail client issue a receipt verifying that the document has been received. Students are also strongly advised to retain a copy of the dated submission on a separate disk. Faculty members are encouraged, but not required, to acknowledge receipt of the assignment."

Tentative Class Schedule[2]

Week Starting	Chapter	Lecture Topics
August 20th	Chapter 1	Introduction Math Preliminaries Roundoff Errors
August 27th	Chapter 2	Algorithms and Convergence Bisection and Fixed Point Methods Newton's Method Error Analysis for Iterative Methods
September 3rd		**Holiday – Labor Day – 9/4/17** Convergence Acceleration Zeros of Polynomials
September 10th	Chapter 3	Interpolation and the Lagrange Polynomial Divided Differences Hermite Interpolation
September 17th		Cubic Splines **EXAM #1, 9/19/17**
September 24th	Chapter 4	Numerical Differentiation Elementary Numerical Integration
October 1st		Romberg Integration Adaptive and Gaussian Quadrature Multiple Integrals **FALL BREAK – 10/5/17-10/6/17**
October 8th	Chapter 5	Elementary theory of Ordinary Differential Equations Initial Value Problems Euler's Methods
October 15th		Taylor Polynomial Methods Runge-Kutta Methods **EXAM #2, 10/17/17**
October 22nd	Chapter 6	Direct Solutions for Linear Systems of Equations Matrix Operations Gaussian Elimination and Pivoting
October 29th		Matrix Factorization Special Matrices
November 5th	Chapter 7	Iterative Techniques for Matrix Algebra Solving Linear Systems
November 12th		Error Bounds for Iterative Matrix Techniques **EXAM #3, 11/16/17**
November 19th	Chapter 8	Least Squares Approximation Orthogonal Polynomials **THANKSGIVING BREAK – 11/22/17-11/24/17**
November 26nd	Chapter 10	Direct Fourier Transforms Newton's Method for Nonlinear Systems
December 3rd		Steepest Descent Techniques **REVIEW**
December 10th		**FINAL EXAM, 12/14/17, 7 p.m.** (Thursday)

[2] I reserve the right to modify this schedule as situations warrant.