1 Course Description

Catalog Description

Algorithms and software for fundamental problems in scientific computing. Topics: properties of floating point arithmetic, linear systems of equations, matrix factorizations, stability of algorithms, conditioning of problems, least-squares problems, eigenvalue computations, numerical integration and differentiation, nonlinear equations, iterative solution of linear systems.

1.1 Instructors

Mr. Thomas Kennedy & Dr. Nikos Chrisochoides

1.1.1 Office Hours
Office Hours are listed on my CS homepage, https://www.cs.odu.edu/~tkennedy.

The best way to get help is to come to office hours. If you cannot make office hours, please send an email to set up an appointment. I am available via email, tkennedy@cs.odu.edu, and network conferencing. In general you can expect a response within 24 hours. On holidays or weekends, turn-around-time will be closer to 48 hours.

1.2 Prerequisites

Students are expected to have a strong foundation in mathematics and programming:

1. Math 316 - Good understanding of calculus and linear algebra.
2. CS 250 - Programming skills in a language such as: C/C++, Python, Rust, or Java.

1.3 Meeting Times & Delivery Method

- **Location:** On-campus in Gornto 0217 and online via WebEx
- **Time:** Tuesdays and Thursdays from 01:00pm to 02:15pm

The provision of face-to-face lectures and live WebEx streams are contingent on the setup of the classroom and supporting equipment, and the general level of Covid-19 activity in the Norfolk/Virginia Beach area. If I become uncomfortable with the safety precautions being taken, or feel that the classroom environment and equipment detract from the quality of the course, then I reserve the right to change to a pure web conferencing delivery mode.

2 Basic Course Information

2.1 Reference Texts

Class notes are available on Blackboard and the course site. Supplemental texts are optional:

1. (Optional) Elementary Numerical Analysis: An Algorithmic Approach, by Samuel Daniel Conte, Carl De Boor


3 Course Policies

3.1 Class Attendance

During lectures, we will be covering material from my notes. Lecture will also consist of the exploration of several real world problems not covered in any book. I may assign (or announce through Blackboard) a reading
assignment or thinking assignment at the end of each lecture.

You are responsible for the contents of all lectures. I expect you to attend class and to arrive on time. If you must miss a class, you are responsible for all material (e.g., lecture and assignments). I expect you to watch the recorded lecture within 24 hours.

3.2 Due Dates & Late Submissions

I do not accept late work. Exceptions to this and other grading policies will be made only in situations of unusual and unforeseeable circumstances beyond your control, and such arrangements must be made prior to the due date in any situations where the conflict is foreseeable.

3.3 Classroom Conduct

Please be respectful of your classmates and instructor by minimizing distractions during class. Cell phones must be turned off during class.

3.4 Academic Integrity

By enrolling in CS417 & CS517, you are expected to uphold the standards of academic integrity set by the Old Dominion University (https://www.odu.edu/oscai).

3.5 Submission (Written Exercises)

All written exercises must be submitted through Blackboard in PDF format. Each such submission must take the form of a single multi-page PDF document.

3.6 Submission (Programming Exercises)

All programming exercises (e.g., Machine Assignments) must be submitted as a single zipfile (zip) or tarball (tar.gz, tar, or tgz). Unless explicitly stated in the prompt, other formats will not be accepted. Each submission must include:

1. A brief ReadMe file that specifies how to compile and run your code.
2. Makefiles (or equivalent build files) for any compilable code.
   - For C/C++, I recommend make or Cmake
   - For Java, I recommend Gradle

All code must follow best practices, including:

- Documentation
- Indentation and spacing
- Naming conventions
- Top-down design (i.e., no monolithic functions)
- D.R.Y

3.6.1 Permitted Languages
You may use any combination of C, C++, Java, Python (3.7+), and Rust. All other languages must be discussed with the instructor before assignment/project submission.

Extra consideration will be given to novel solutions (e.g., those written in a functional language).

3.7 Computer Accounts

Students will need an account on the CS Dept. Linux network to participate in this class. This account is unrelated to any University-wide account you may have from the ODU’s Information Technology Services (ITS).

If you have had a CS Unix account in the recent past, you should find it still active with your login name, password, and files unchanged. If you have had an account and it has not been restored, contact the CS Dept systems staff at root@cs.odu.edu requesting that it be restored.

If you do not yet have such an account, go to the CS Dept. home page and look for “Account Creation” under “Online Services”. All students in this course are responsible for making sure they have a working CS Unix account prior to the second week of class.

4 Grading

Final grades will be computed using the following weights:

<table>
<thead>
<tr>
<th></th>
<th>CS 417</th>
<th>CS 517</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homework</td>
<td>30%</td>
<td>25%</td>
</tr>
<tr>
<td>Machine Assignments</td>
<td>--</td>
<td>5%</td>
</tr>
<tr>
<td>Exam 1</td>
<td>15%</td>
<td>15%</td>
</tr>
<tr>
<td>Exam 2</td>
<td>15%</td>
<td>15%</td>
</tr>
<tr>
<td>Semester Project</td>
<td>10%</td>
<td>10%</td>
</tr>
<tr>
<td>Final Exam</td>
<td>30%</td>
<td>30%</td>
</tr>
<tr>
<td>Total</td>
<td>100%</td>
<td>100%</td>
</tr>
</tbody>
</table>

Final course grades will be assigned based on the standard 10-point scale:

<table>
<thead>
<tr>
<th>Point Range</th>
<th>Letter Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>90 – 100</td>
<td>A</td>
</tr>
<tr>
<td>80 – 89</td>
<td>B</td>
</tr>
<tr>
<td>70 – 79</td>
<td>C</td>
</tr>
<tr>
<td>60 – 69</td>
<td>D</td>
</tr>
<tr>
<td>59 and Below</td>
<td>F</td>
</tr>
</tbody>
</table>

1. I will apply pluses (+) and minuses (-) to letter grades as appropriate.

2. CS417 students, I encourage you to complete the machine assignments. I will add extra points to your Homework grade.

3. Grade weights may be adjusted. Particular consideration will be given to the Semester Project weight.
5 Exams

Exams will be administered online through Blackboard.

6 Tentative Topics

Part I*

1. MACHINE ARITHMETIC AND RELATED MATTERS
   - Machine Arithmetic and Rounding
   - The Condition of a Problem and Algorithm
2. APPROXIMATION
   - Least Squares Approximation
3. INTERPOLATION
   - Polynomial Interpolation
   - Approximation and Interpolation by Spline Functions
4. NONLINEAR EQUATIONS
   - Examples, Iteration, Convergence, and Efficiency
   - Method of False Position
   - Secant Method
   - Newton’s Method

Part II*

1. NUMERICAL DIFFERENTIATION AND INTEGRATION
   - Numerical Differentiation
   - Numerical Integration

Topics of Interest* (Time Permitting)

1. MATRIX COMPUTATIONS – SYSTEMS OF LINEAR EQUATIONS
   - Gauss Elimination and LU Factorization: Basic Algorithms
   - Iterative Methods: Basic Algorithms

* The topics covered and time spent on each topic may change based on class performance and class pacing.

6.1 Objectives

This course is designed as an introduction to the basics numerical analysis. At the end of this course students will be able to:

1. Examine the impact of finite arithmetic on computation.
2. Formulate numerical approximation problems and investigate their solutions via programming.
3. Investigate the relationship between mathematical models and their matrix representations.
4. Synthesize solutions from simpler parts.
5. Design algorithms and prove their correctness using theory/analysis as building blocks.
6. Examine the spatial complexity, temporal complexity, and computational complexity of numerical solvers.
7. Implement numerical software using existing libraries.
8. Test and verify the correctness of their own codes using model problems and principles from theory.
9. Utilize vocabulary and terminology used by engineers and applied mathematicians.
10. Summarize the literature and utilize basic software packages on numerical approximation.
11. Discuss research issues in numerical computing.

6.2 Expectations

This course covers theory oriented and math oriented topics and problems. You are expected to review required mathematics, programming, and theory as directed. If whilst reviewing you would like direction or clarification contact the instructor (via email or in the discussion board).

We discuss various notations (for math and pseudocode) in this course. Keep in mind “That it is just notation.” Familiarizing oneself with unfamiliar notations is a skill we will endeavor to develop this semester.

If you have questions… Ask the instructor The cliché “If you have a question… at least one of your classmates has the same question.” is true (especially this semester).

7 Academic Accessibility

Old Dominion University is committed to ensuring equal access to all qualified students with disabilities in accordance with the Americans with Disabilities Act. The Office of Educational Accessibility (OEA) is the campus office that works with students who have disabilities to provide and/or arrange reasonable accommodations.

- If you experience a disability which will impact your ability to access any aspect of my class, please present me with an accommodation letter from OEA so that we can work together to ensure that appropriate accommodations are available to you.
- If you feel that you will experience barriers to your ability to learn and/or testing in my class but do not have an accommodation letter, please consider scheduling an appointment with OEA to determine if academic accommodations are necessary.

The Office of Educational Accessibility is located at 1021 Student Success Center and their phone number is (757) 683-4655. Additional information is available at the OEA website (http://www.odu.edu/educationalaccessibility/).