CS4390 & CS5390
Applied Numerical Computing for Multimedia Applications
Spring 2025
Syllabus

Title:	CS4390: Applied Numerical Computing for Multimedia Applications, CRN 24060 crosslisted as CS5390, CRN 25852
Term:	Spring 2025
Duration:	Tue 01/21/2025 through Thu 05/08/2025
Lecture Time:	Tue, Thu, 1:30PM - 2:50PM
Lecture Location:	GEOL 123
Instructor:	Dr. Christoph Lauter `cqlauter@utep.edu` 915-747-5939
Instructors Office Hours and Location:	Email anytime, Phone during business hours, Office (CCSB 3.0610) Mon-Thur 10:00AM-11:50AM
Class Web page:	`https://www.christoph-lauter.org/utep-numerical/`
Prerequisites:	None. Please bring motivation and joy to class.
Description:	While the first computers in the history of computing were fully dedicated to numerical applications, computer science nowadays mainly focuses on textual information storage and handling. However, in certain fields, the actual numerical compute power of modern machines is still needed and gets leveraged in novel applications such as machine and deep learning. This course gives an overview of the hardware and, in particular, software techniques used for applied numerical computing: how long integers can be stored and manipulated inside a computer, e.g. to allow for strong asymmetric encryption, how real numbers can be represented and manipulate, for example in graphical processing, how signals can be represented and manipulated in software systems, how basic and advanced linear algebra software packages can be designed and then utilized to solve real-life problems such as compressed sensing, dictionary learning or medical imaging, MP3 and JPEG data compression, ADSL transmission, 5G mobile service etc.
Student Learning Outcomes:	See appendix below. Student progress on the learning outcomes will be assessed primarily via programming projects (homework) and exams.
Textbooks:	This is a research-oriented special topics course. There is no specific textbook. Students may want to read the following books and articles:

- Muller et al., *Handbook of Floating-Point Arithmetic*, Springer
- Goldberg, *What every Computer Scientist should know about Floating-Point Arithmetic*

- Overton, *Numerical Computing with IEEE Floating-Point Arithmetic*, SIAM
- Ercegovac et al., *Digital Arithmetic*, Morgan Kaufman
- Oppenheim et al., *Discrete-Time Signal Processing*

Homework Assignments: There will be 2 or 3[1] homework assignments. The homework assignments will be programming based. Students can choose the programming language of their liking[2]

Workload & Attendance: This course requires participation during the in-class lectures and engagement into the homework assignments.

Please form study groups and you are encouraged to discuss the approach and understand the problem. The write up, programming, and actual solutions to the homeworks are team-work but must be the individual work of each student. If you use someone's work for your own, you are committing plagiarism.

Readings: None. See above.

Computers: Students need access to a computer that allows them using a native Linux environment or that allows them `ssh` access to a Linux server, provided by the instructor. For homework assignments that compile in one environment and not the other, the result obtained on the system provided by the instructor will be deemed the canonical result.

Tests and Testing Policy: Three tests will be given: two mid-term tests and a final. All tests are cumulative, with an emphasis on recent material. No make up tests are given but for documented medical emergencies.

Grading: Letter grade:

Points range	Grade
$[90; 100]$	A
$[80; 90)$	B
$[70; 80)$	C
$[60; 70)$	D
$[0; 60)$	F

Grading breakdown:

Homework	60%
Midterm I	7%
Midterm II	8%
Final exam	20%
Attendance	5%

Students taking this course as CS5390 must answer all extra questions marked in all homework assignments. These extra questions will require students read certain scientific papers in order to answer the homework questions.

[1] The exact number will be announced sufficiently in advance.

[2] Reasonable languages only; examples of reasonable languages include C, Python, C++, Java, Fortran.

Make-Ups: Make-ups are not allowed. Make-up work will be given only in the case of a documented emergency. Note that make-up work may be in a different format than the original work, may require more intensive preparation, and may be graded with penalty points. If you miss an assignment and the reason is not considered excusable, you will receive a zero. It is therefore important to reach out to the instructor **before** the respective exam starts, resp. **before** the homework assignment deadline is up, and explain with proper official documentation why you missed a given course requirement. Once a deadline has been established for make-up work, no further extensions or exceptions will be granted.

Approximate Course Schedule: The course will be spread out **approximatively** according the following schedule:

Week 1	Representation of Unsigned & Signed Integers
Week 2	Addition of Integers, Multiplication of Integers
Week 3	Karatsuba, Toom-Cook, Cryptography
Week 4	Introduction to IEEE754 Floating-Point Numbers
Week 5	Issues with IEEE754 Floating-Point Arithmetic
Week 6	Midterm I
Week 7	Introduction to Error Analysis
Week 9	Introduction to Linear Algebra
Week 8	Applications of Linear Algebra, Interpolation, Dictionary Learning
Week 9	Introduction to Signal Processing
Week 10	Midterm II
Week 11	Fast Fourier Transform, ADSL, MP3, JPEG
Week 12	Goertzel Algorithm, FSK Modulation, DTMF
Week 13	Signal Processing Applications, Medical Imaging
Week 14	Revision, Wrap-Up

This course schedule is **approximative**. In the case the instructor thinks that students need more time to better understand a subject (or less time because they are really, really smart), the instructor will deviate from the schedule. This may affect the dates the midterm exams are given.

Accomodations: The University is committed to providing reasonable accommodations to students with documented disabilities. Students who become pregnant may also request reasonable accommodations, in accordance with state and federal laws and regulations and University policy. Accommodations that constitute undue hardship are not reasonable. To make a request, please register with the UTEP Center for Accommodations and Support Services (CASS). Contact CASS at 915-747-5148, email them at `cass@utep.edu`, or apply for accommodations online via the CASS portal.

Scholastic Integrity:

- Academic dishonesty is prohibited and is considered a violation of the UTEP Handbook of Operating Procedures. It includes, but is not limited to, cheating, plagiarism, and collusion. Cheating may involve copying from or providing information to another student, possessing unauthorized materials during a test, or falsifying research data on laboratory reports. Plagiarism occurs when someone intentionally or knowingly represents the words or ideas of another as ones' own. Collusion involves collaborating with another person to commit any academically dishonest act. Any act of academic dishonesty attempted by a UTEP student is unacceptable and will not be tolerated. All suspected violations of academic integrity at The University of Texas at El Paso must be reported to the Office of Community Standards (`https://www.utep.edu/student-affairs/standards/`) for possible disciplinary action. To learn more, please visit HOOP: Student Conduct and Discipline (`https://www.utep.edu/hoop/section-2/student-conduct-and-discipline.html`).

- Submitted work should be unmistakably your own. You may not transcribe or copy a solution taken from another person, book, or other source (e.g., a web page). Professors are required to report academic dishonesty and any other violation of the Standards of Conduct to the Dean of Students.

- Some AI technologies or automated tools, particularly generative AI such as ChatGPT or DALL-E, can be beneficial during the early brainstorming stages of an activity, and you are welcome to explore them for that purpose. However, keep in mind that AI-generated ideas are not your own and may hinder your ability to think critically and creatively about a problem. It is also important to remember that these technologies often "hallucinate" or produce materials and information that are inaccurate or incomplete—even providing false citations for use. That said, you are not allowed to submit any AI-generated work in this course as your own. If you use any information or materials created by AI technology, you are required to cite it like you would any other source. Consider how this will affect your credibility as a writer and scholar before doing so. Any direct use of AI-generated materials submitted as your own work will be treated as plagiarism and reported to the Office of Community Standards.

- Permitted collaboration: Students may discuss requirements, background information, test sets, solution strategies, and the output of their programs. However, implementations and documentation must be their own creative work. Students are required to document advice received from others and all resources utilized in the preparation of their assignments.
- If academic dishonesty is suspected: The Dean of Students office will be contacted for adjudication. A temporary "incomplete" grade will be issued if their investigation extends beyond the grading period.

Cell phones: Please silence your mobile devices or put them into a vibrate mode for the duration of class - they are disruptive for your fellow students. No mobile devices (cell phones, PDAs, laptops etc.) are allowed during the exams and will result in your expulsion from the test.

Appendix: CS4390 & CS5390 Learning Outcomes

Level 1:

- Explain positional representation of integers.
- Explain negating integers in various representation of signed integers.
- Analyze the complexity of addition algorithms for integers, understand the optimality of traditional addition algorithms.
- Analyze the complexity of schoolbook multiplication, Karatsuba multiplication and Toom-Cook multiplication.
- Understand the requirements for arbitrary length integers in asymmetric cryptography.
- Encode and decode real numbers as IEEE754 Floating-Point Numbers in various formats.
- Understand basic Wilkinson and Higham Error Analysis.
- Understand the need for Linear Algebra Packages in High Performance Computing.
- Explain the use of Linear Algebra for Interpolation, Dictionary Learning and other Multimedia Applications.
- Explain the principles of Digital Signal Processing with Time and Value-Discrete Signals.
- Explain the principles of MP3 audio compression.
- Explain the principles of digital medical imaging, such as MRT.

Level 2:

- Explain positional representation of integers when machine integers are used for radices $\beta = 2^{32}$ or $\beta = 2^{32}$.
- Explain negation of integers in two's complement when machine integers are used for radices $\beta = 2^{32}$ or $\beta = 2^{32}$.
- Design algorithms for addition of integers with machine integers used for radices $\beta = 2^{32}$ or $\beta = 2^{32}$ when no carry generating machine operation is available.
- Analyze the complexity of multiplication through Fast Fourier Transforms in a finite field.
- Understand the effects of rounding on IEEE754 Arithmetic.
- Understand advanced IEEE754 Floating-Point Algorithms like Fast2Sum and Fast2Mul with FMA.
- Explain the function of Linear Algebra Routines like Matrix-Matrix Multiplication or Gaussian Elimination.

- Explain Fast Fourier Transform on Time and Value-Discrete Signals.
- Explain the principles of FSK modulation and demodulation.

Level 3:

- Implement efficient routines for arbitrary length unsigned integers.
- Implement negation of arbitrary length signed integers in two's complement.
- Implement optimal addition of arbitrary length integers.
- Implement schoolbook, Karatsuba and Toom-Cook multiplication for arbitrary length integers.
- Implement routines to decode IEEE754 Floating-Point and encode them, with rounding in various rounding directions.
- Implement basic Linear Algebra routines including Gaussian Elimination/ LU decomposition.
- Implement Fast Fourier Transform.
- Implement basic signal processing routines such as sinusoid generation or frequency detection with the Goertzel algorithm.
- Implement FSK modulation and demodulation.