## Shih Chien University

## STP Program (July 01-Aug 02)

# MATH 310 Computation with MATLAB

## Course Outline

**Course Code: MATH 310**

**Instructor: Professor Vadim Olshevsky**

**Home Institution: University of Connecticut**

**Office Hours: TBA**

**Email: olshevsky@gmail.com**

**Credits: 4**

**Class Hours:**

This course will have 144 class hours, including 50 lecture hours, professor 30 office hours, 20-hour TA discussion sessions, 10-hour review sessions, 34-hour extra classes.

**Course Description:**

This course will provide an introduction to the use of computers in solving problems arising in the physical, biological, and engineering sciences. Various computational approaches commonly used to solve mathematical problems will be presented, including systems of linear equations, FFT, curve fitting, integration, and differential equations. Both the theory and applications of each numerical method will be demonstrated.

MATLAB will be used as the only environment for numerical computation, and it will be the only language to submit homework coding assignments. No previous coding experience is required.

**Course Objectives:**

In this course you will study and implement numerical methods and tools used in
scientific applications. You will learn to:

- Identify common mathematical problems (e.g., linear systems, optimization, curve fitting, differential equations, and principal component analysis) and choose appropriate mathematical methods (e.g., iterative solvers, time-stepping methods, etc.) to solve them.
- Understand the strengths and weaknesses of different numerical algorithms in terms of accuracy, complexity, and speed. Further, you will use the knowledge gained to choose the appropriate technique for the application of interest.
- Write code in MATLAB to implement numerical algorithms.
- Interpret, format, and present results, including visualization of data.

**Required Course Materials:**

- MATLAB. The student version can be purchased online at
  New License for MATLAB Student (mathworks.com)
- "Numerical Mathematics and Computing", by Ward Cheney and David Kincaid. The 7th edition. I recommend renting it for one semester at Amazon.
- "Applied Numerical Methods with MATLAB" by Steven Chapra. The 4th edition. A 6-month access to eBook can be purchased from MvGeaw Hill.

**Homework:**

There will be six or more (time permitting) MATLAB assignments. You will need to write a MATLAB code for each project, and upload it to the free MATLAB GRADER site, where it will be graded automatically. Since the code either works or not, your grade for each assignment will be either 100 or 0. You will have 5 attempts to submit your code.

More information about registration at the MATLAB grader site will be provided when the course starts. You will receive an invitation from your professor to register.

**Grading & Evaluation:**

| | |
|---|---|
| Homework: | 40% |
| Midterm: | 30% |
| Final: | 30% |

**Grading System (1 ~ 100):**

| Quality Points | Grade | Percentage % |
|---|---|---|
| 4 | A | 80-100 |
| 3 | B | 70-79 |
| 2 | C | 60-69 |
| 1 | D | 50-59 |
| 0 | E | 0-49 |

**Course Schedule**

**Week1** *Computer arithmetic. Solving linear systems. Interpolation.*

**Week2** *Roots of polynomials. Iterative methods. FFT.*

**Week3** *Methods for computing eigenvalues. Discrete least squares. Chebyshev polynomials.*

**Week4** *Continuous least squares. Orthogonal polynomials. Numerical integration. Gaussian quadrature.*

**Week5** *ODs..*

**Detailed Course Outline:**

| Week | Chapter | Topic |
|---|---|---|
| | Accuracy | Significant digits of Precision. Errors: absolute and relative. Floating point representation. Single precision. Double precision. Computer errors in number *representation* (not yet in computations). Machine Epsilon (unit roundoff). Computer errors |

| | | |
|---|---|---|
| | | encountered in *computations*. A priory Roundoff Error Analysis. Loss of significant digits. |
| 1 | Solving linear systems | Gaussian elimination. LU factorization. Forward and back substitution. Cost of Gaussian elimination. Partial and complete pivoting. |
| | Interpolation | The existence and uniqueness of the solution to the interpolation problem. Lagrange formula. Newton formula and divided differences. The Neville and the Aitken approaches. |
| | | Matrix formulation of the interpolation problem. The Vandermonde matrix. The Bjork-Pereyra algorithm for interpolation and its remarkable accuracy. |
| | Roots of polynomials | Bisection. Rate of convergence. Regular Falsi. The Newton-Raphson method. Quadratic convergence for simple roots. Newton method for nonlinear equations. |
| 2 | FFT | Discrete Fourier Transform. The DFT matrix and its unitarity. The idea behind divide-and-conquer algorithms. The divide-and conquer formula for the DFT matrix. Implementation details. Bit reversal. Decimation in time and frequency algorithms. |
| | | Fast polynomial multiplication. Convolutions, ordinary and cyclic. Circulant matrices and their DFT-based diagonalization. Embedding. |
| | Iterative methods | The Jacobi, Gauss-Seidel and SOR methods. Matrix interpretations. Norms and convergence analysis. Diagonally dominant and positive definite matrices. |
| | Eigenvalues | The power method. Aitken acceleration. The inverse power method. Shifting strategies. Gershgorin theorem and initial iteration. |
| 3 | Discrete least squares | Why least squares? Orthogonality in R^n, projections. The best approximation theorem. The Gram-Schmidt process and QR factorization. The |

| | | |
|---|---|---|
| | | solution to the least squares method based on normal equations. The solution to the least squares problem based on the QR factorization. |
| | midterm | Midterm |
| | Chebyshev polynomials | Definitions and orthogonality. Roots and extrema. Legendre polynomials and other families of orthogonal polynomials. |
| **4** | Continuous least squares | MATLAB fitting tool. Bases of orthogonal polynomials. Normal equations. Solution to the continuous least squares problem. Modified Gram-Schmidt. |
| | Numerical integration | Trapezoid rule. Error analysis. Simpson's rule. The adaptive Simpson's rule. Composite integration. General framework for the Newton-Cotes formulas. |
| | | Orthogonal polynomials revisited. Gaussian quadrature rules. The Newton-cotes formulas are the "rules". Legendre and Chebyshev polynomials revisited. |
| | ODEs | Initial Value problem. Vector fields. Taylor series and Euler's methods. |
| | | Runge-Kutta methods. Order 2 and 4. Multistep methods.<br>Adams-Bashforth-Moulton formula. |
| **5** | | Methods for the first and higher order systems. Uncoupled and coupled systems. Taylor series method. Runge-Kutta method. Autonomous ODEs. |
| | | Boundary value problems. The shooting method. The finite-difference method. |
| | | Final exam |

**Student responsibilities/expectations**: The main course material will be presented through lectures. A discussion session, to be held every Friday will offer an opportunity for students to discuss course material and assigned problems with a teaching assistant (TA). Students are advised to keep pace with the course material as it is being presented. Consequently, students should endeavor to attend all class meetings and discussion sessions, be early for class, and spend sufficient time working on assigned homework problems. If for any reason a student misses a class, he/she should endeavor to obtain the notes and learn the missed material before the next class meeting. Students should not hesitate to ask questions or seek additional assistance to ensure that they are staying on pace with the class. Students will be expected to come to class prepared and ready to participate actively. Please, turn off your cell phones and put aside any unrelated material before class begins. Students should exhibit a sense of responsibility and respect towards fellow students. Late-coming to class or early departure from class meetings will not be allowed.

**Examinations**: There will be one midterm exam plus one final exam. The exams will contain problems to solve and definitions, brief explanations of concepts, derivation of the algorithms and simple proofs.