CISC4750 SCIENTIFIC COMPUTATION USING MATLAB
SYLLABUS

Semester : Summer Session II 2020
Instructor : Dr. T. Strzemecki
Office : LL 610B
Office hours : M 2:00 PM – 6:00 PM, W 4:00 PM – 6:00 PM or by appointment
Textbook : Hanselman, Littlefield Mastering MATLAB
e-mail : ts@dsm.fordham.edu

I. Course Prerequisites

There are two required course: CISC 1600 Computer Science I ( or its equivalent ) and MATH 1207 Calculus II. Linear Algebra is not required but it is highly recommended.

II. Course Description

This is an advanced class in which MATLAB programming language and its applications to solutions of scientific and engineering problems are discussed. Programming in MATLAB includes input and output of data, algebraic and logical expressions and operators, hierarchy of precedence of operations, all data types including complex numbers and strings, structured and cell arrays, array indexing and array operations, matrices and matrix operations, control structures, loops, script and function m-files, function handles and their use, graphics and plots, MATLAB functions for graphics modifications, interactive figure modifications. Applications of MATLAB to solutions of engineering problems include solutions of systems of linear equations, analysis of electrical circuits, modeling of discrete dynamical systems, Markov chains, population growth, economy input-output models, balancing of chemical reactions. Applications of MATLAB to scientific problems include algorithms for numerical integration and differentiation, polynomial curve fitting, root finding of nonlinear functions, recurrences, Monte Carlo methods, systems of linear differential equations.

III. Course Structure

This is a course with EP3 attribute. Therefore, throughout the semester, students will be required to prepare either two small presentations on two different subjects, or one larger presentation on a larger topic. Written and oral presentations are required. Each of these presentations will require consultations with the class Instructor and possible revisions before the written presentation(s) are ready to provide basis for the subsequent oral presentation. Each revised written presentation, upon the validation by the course Instructor, will be followed by the oral presentation in the classroom. All relevant m-files for the MATLAB project must be included in the written presentation and their use demonstrated in the subsequent oral presentation. Each oral presentation will have to include at least one complete MATLAB-based project pertinent to the subject of the presentation. Each presentation must result in a homework and/or MATLAB project that will be assigned for all students in the class.
**Description of the Written Presentation:** The written presentation will be subjected to the Instructor’s evaluation upon submission. As part of the evaluation process, remarks on the written presentation will be formulated. It is required that a student incorporates these remarks into the revised version of the written presentation so as to make the written presentation a suitable basis for the later oral presentation in the classroom setting. A particular attention in the evaluation process will be paid to two elements. Firstly, examples and programming problems selected for the written presentation will be examined and evaluated carefully. It will be done especially from the point of view of the relevancy of examples to the coverage of the topic, as well as the explanatory value of the selected examples and programming problems. Secondly, the depth and completeness of coverage of the written presentation topic will be carefully considered and evaluated, especially from the point of view of their expected perception by, and usefulness to, other students. After the final form of the written presentation is evaluated the Instructor will assign a grade. This grade will constitute 15% of the final grade. In addition, the students will be required to write a short technical essay after each programming session in the classroom. Each such essay will be treated and graded as additional homework.

**Description of the Oral Presentation:** 25% of the grade assigned to the student’s oral presentation will be decided as follows: students will submit the oral presentation to the Instructor at least two weeks before the actual presentation. The Instructor will provide remarks on the presentation and request final modifications. The finally submitted proposals will be evaluated by the Instructor and will count for 8% (out of 25% available) of the grade for the oral presentation. Students, as part of their oral presentation, will also have to suggest homework assignments based on their presentation. This aspect of student’s oral presentation will be carefully evaluated and will contribute 3% (out of 25% available) of student’s grade for the oral presentation. After these steps are completed the Instructor will give a formal approval for the final form of the oral presentation. The quality of the oral delivery of the presentation will count for 10% (out of 25% available) of the oral presentation grade. Peer evaluation will be used and will account for 4% (out of 25% available) of the grade for the oral presentation. It will be limited to the following criteria: (1) Clarity of the presentation, (2) Logic and organization of the presentation, (3) Effectiveness and explanatory aspects of the presentation, (4) Usefulness of programming problems and examples selected, (5) Overall impression. Peer evaluation will have a form of anonymous written comments and/or open class discussion. In addition, students will participate in the class discussions concerning the programming projects discussed in the class. These discussions will not be graded.

**IV. Grading Policy**

The final grade will consist of the following components: homeworks - 20%, written presentation - 15%, oral presentation - 25%, written final examination - 15%, oral final examination - 25%.

**V. List of Presentations**

(See pages 3-6)
PRESENTATIONS


MATLAB

Remark: Students must read Chapters 1, 2, and 3 from the Hanselman book before participating in the sequence of presentations outlined below.

1. Before Starting Calculations
   - Presentation by the Instructor

2. Scripts and Arrays
   - Script Files (H4.1, H4.2)
   - One-Dimensional Arrays (H5.1 - H5.12)
     - Sections 5.10 and 5.11 should be covered when time allows
   - Multi-Dimensional Arrays (H6.1 - H6.3)

3. Data Types and Cell Arrays
   - Numerical Data Types (H7.1 - H7.3)
   - Cell Arrays (H8.1 - H8.6)
     - Sections H8.7, H8.8, H8.11 should be covered when time allows
   - String Types (Chapter H9.1 - H9.5)
     - Section H9.6 should be covered when time allows
4. Flow Control and Matrix Algebra
   - Relational and Logical Operations (H10.1 - H10.5)
   - Control Flow (H11.1 - H11.4)
   - Matrix Algebra (H16.1 - H16.3)

5. Functions and Vectorization
   - Presentation by the Instructor

6. Two-Dimensional Graphics
   - Basics (H25.1 - H25.8, H25.12)
     - Sections H27.1 - H27.4 should be covered when time allows

7. Monte-Carlo Methods
   - Presentation by the Instructor

ENGINEERING

1. Linear Systems
   - Solving Methods and Notation (L1.1.1 - L1.1.4)
   - Matrix representation of Linear Systems (L1.1.5)
     - Augmented Matrix
     - Elementary Row operations
     - Solving Linear Systems Using Matrices
   - Echelon Forms (L1.2.1 - L1.2.6)
     - Echelon Forms, Rank, Consistent Systems
     - Homogeneous Linear Systems
   - Applications of Linear Systems
     - Linear Models, Networks (L1.3.1 - L1.3.2)
       - Solving Electrical Networks (1.3.3)
       - Balancing chemical reactions (1.3.4)
2. Matrices
   - Matrix Algebra (L2.1)
   - Inverses (L2.2.1, L2.2.3, L2.2.4)
     - (skip elementary matrices)
   - LU Factorization (2.3)
   - Applications of Matrices
     - Graphs (L2.4.1)
     - Discrete Dynamical Systems (L2.4.2)
   - Determinants
     - Definition and Computation (L5.1)
     - (Include Theorems L5.6 and Theorem L5.8 on p.261)
   - Inverses and Products (L5.2 - L5.7)

3. Eigenvalues and Eigenvectors
   - Eigenvalues and Eigenvectors (L6.1)
     - omit eigenspaces
   - Diagonalization (L6.2)
   - Applied Eigenvalue Problems (L6.3)
     - Predator-Prey Model (Illustration L6.3 on p.293)
     - Car Rental (Illustration L6.4 on p.298)
     - Linear or second-order recurrence relations
       - Markov Chains (6.4) (optional)
     - Systems of Linear Differential Equations (L6.5)
       (optional)

4. Complex Numbers
   - Algebraic Theory (L8.1)
   - Geometric Theory (L8.2 up to complex functions on p.378)
   - Polar Form (L8.3)
1. Polynomials and Interpolation
   - Polynomials ( H19.1 - H19.9 )
   - Curve-Fitting ( H20.1 - H20.2 )
   - MATLAB Commands for Polynomials ( N4.1 )
   - Linear Interpolation ( N4.2 )
   - Polynomial Interpolation With Power Series ( N4.3 )

2. Numerical Integration
   - Trapezoidal Rule ( N5.1 )
   - Simpson’s Rules ( N5.2 )
   - m-files FM 5-1 -- FM5-3 ( N5.7 ) ( optional )
   - Interpolation H20.4
   - One Dimensional Interpolation ( H23.1 - pp.398-401 )

3. Numerical Differentiation
   - Derivatives of Interpolation Polynomials ( N6.1 )
   - Difference Approximations ( N6.2 )
   - Taylor Expansion Method ( N6.3 ) ( if possible )
   - Differentiation ( H20.5 )
   - Differentiation ( H23.2 )
     - Skip MATLAB gradient function on pp.408-409

4. Optimization
   - Zero-Finding ( H22.1 )
   - Roots of Nonlinear Equations
     - Graphical Method ( N7.1 )
     - Bisection Method ( N7.2 )
     - Newton Iteration Method ( N7.3 )
     - Secant Method ( N7.4 )
     - m-files FM 7-1, FM 7-2, FM 7-3, FM 7-4 ( N7.7 )
   - Nonlinear Curve Fitting with a Power Function N8.2