HARRISBURG UNIVERSITY OF SCIENCE AND TECHNOLOGY

CISC 600
Scientific Computing I

Instructor: Luis Paris, PhD.
Email: lparis@harrisburgu.edu
Office Hours: Moodle Discussion Forum.
Meetings: Please see Moodle page (sessions hosted via Adobe Connect)

COURSE DESCRIPTION

Scientific Computing I and II, is a series of two courses with CISC600 being the first. The purpose of the Scientific Computing series is to open up a few computational black boxes for the student to scrutinize. The idea is to show how to take apart these black boxes using numerical methods, and how to modify them to suit specific needs. The main goal is to enhance the student’s computational skills needed for powerful numerical simulations. Starting from mathematical models (derivation, analysis, classification, etc.), their numerical treatment is analyzed. The course will use Python as a reference language to solve problems. Students can use other languages with adequate support for numerical computations.

Scientific Computing I provides an overview of the numerical methods used in most engineering practices. The course covers: Solution of Linear Algebraic Equations, Interpolation and Extrapolation, Integration and Evaluation of Functions, Random Numbers, Curve Fitting, etc. The course will use Python programming language as a base language to solve problems. Students can use other languages with sufficient numerical support. The course is intended as an introduction to the thriving field of numerical simulation for computer scientists, mathematicians, engineers, or natural scientists with no solid background in numerical methods. This is a core MS CISC course.

COURSE OBJECTIVES

- Apply the fundamental knowledge of mathematics, science & engineering, to analyze and interpret data in various science and engineering disciplines.
- Learn how to write simple codes to solve the given problems.
- Develop computing proficiency by learning and implementing the methods.
- Develop problem-solving skills.
- Demonstrate a basic competence in scientific computing.
- Summarize a research paper on scientific computing.

COURSE PREREQUISITES

A bachelor's degree (BA / BS / BE) in computer science or a related technical field (e.g., electrical and computer engineering, information science, operations research) typically suffices.

Applicants who have majored in other fields are absolutely encouraged to apply provided they have demonstrated knowledge of the following subjects:

- **Introductory level Programming** Topics include program structure and organization, object-oriented programming (classes, objects, types, sub-typing), graphical user interfaces, algorithm analysis, recursion, data structures, simple graph algorithms. Any programming language is fine.

- **Introductory level Linear Algebra** Topics include properties of real numbers, problem-solving using equations and inequalities, algebraic functions, graphing, systems of equations and inequalities, polynomial functions and graphs, exponents and radicals, the binomial theorem, zeros of polynomials, inverse functions, and applications and graphs. Free on-line graphing and calculating utilities are used in lieu of a graphing calculator.
- **Introductory level Calculus** Topics include techniques to evaluate limits and covers continuity, special trigonometric limits, absolute value limits and differentiation of algebraic, trigonometric, logarithmic functions, intermediate value theorem, mean value theorem, and extreme value theorem. Other topics for exploration are application and formal definition of derivative average rate of change versus instantaneous rate of change, velocity, and the introduction of the definite integral and its applications.
- **Discrete Mathematical Structures** Topics include mathematical induction; logical proof; propositional and predicate calculus; combinatorics and discrete mathematics; some basic elements of basic probability theory; basic number theory; sets, functions, and relations; graphs; and finite-state machines.

TEXTBOOKS

1. **"Numerical Methods for Engineers, 7th edition"**, Steven C. Chapra, and Raymond P. Canale. **ISBN:** 007339792x. **Copyright year:** 2015. McGraw-Hill Education Learning Technology Specialist.
2. "**A Primer on Scientific Programming in Python**", Hans Petter Langtangen, 2nd Edition, Springer, 2011. ISSN 1611-0994, ISBN 978-3-642-18365-2
3. "**Scientific Computing in R**" Karline Soetaert, Lecture Notes from Netherlands Institute of Ecology
4. **"Numerical Analysis**", Richard L. Burden and J. Douglas Faires, 2011, 9th Edition, Brooks/Cole, Cengage Learning, ISBN-13: 978-0-538-73351-9, ISBN-10: 0-538-73351-9.
5. "**Numerical Recipes in C the Art of Scientific Computing**", William H. Press, 2nd Edition, Cambridge University Press. ISBN 0 521 43108 5, ISBN 0 521 75037 7
6. There are many **other textbooks** on scientific computing topics and concepts in almost every computational software like MatLab, Maple, etc. You may use any of these for this course.

Course Conduct

A few ground rules will help us to get the most of our investment in **CISC 600**:

- Moodle is going to be our platform for almost all course activities.
- In addition, we will have 60-minute **lecture sessions** on a weekly or ad-hoc basis.
- **Attending lecture sessions** and actively participate in the discussions is highly recommended as it is where basic concepts of each unit are explained, assignments are discussed and your questions answered.
- **Reading Assignments**. The goal in providing the four textbooks is to help students comprehend the depths of the study material, and prepare them to some sophisticated discussion on the analytical mathematics, algorithmic thinking, actual details of the computations, and the actual working programs on that concept.
- For full mastering of the material; you to do the following:
  - Textbook (1) is our main source, an easy book with clear explanations and examples throughout. The text features a broad array of applications that span all engineering disciplines.
  - Textbooks (2) is a Python books for scientific computing and its libraries.

  - Textbook (3) is same scientific computing concepts in R if you want.
  - Textbook (4) is a reference book, to understand the concept, the derivation of the method, and the algorithmic analysis.
  - Textbook (5) is a C++ programming reference book, you can run the proper codes to see how the computation is done, how the variables in the code are set, and interact, how much they are representing the real world variables, and if the computation is an accurate simulation for the problem.
  - There are many textbooks on the scientific computing topics and its concepts in almost every computational software like MatLab, Maple, etc. You may use any of these during the course of this class.
- **You are responsible for all the readings**, even if the material is not explicitly covered in class. You should read the class materials prior to class and be prepared to discuss and ask questions about the readings and assignments.
- You should also **re-read the material** after class as not every topic will be covered during class time.
- Many passages in the text may need to be read several times to gain clarity. Also, taking notes on the material you are reading and reflecting on the reading and these notes will help you better understand the issues, concepts and techniques that are being presented.
- All work must be completed and turned in on or before the due date. **Late work will be penalized**. Note that a computer's failure is not an excuse (it represents poor planning on your part).
- Your work should be properly referenced and adhere to standards of both academic integrity and proper form. Generally, I prefer the APA style (see http://www.apa.org/).
- All class credit-related electronic mail must be done using Harrisburg's electronic mail service and the student's assigned Harrisburg University ID. By 'credit-related' I mean all work to be evaluated for credit. Any work submitted through a different mail system will **NOT** be accepted.
- All activities will be assigned individually unless mentioned.

COURSE SYLLABUS

W	Topic	Due Dates
1-3	**Modeling, Computers, and Error Analysis** Chapter 1 Math Modeling and Problem Solving Chapter 2 Programming and Software Chapter 3 Approximations and Round-Off Errors Chapter 4 Truncation Errors and the Taylor Series	
4-6	**Roots of Equations** Chapter 5 Bracketing Methods Chapter 6 Open Methods Chapter 7 Roots of Polynomials Chapter 8 Case Studies: Roots of Equations	
7-9	**Linear Algebraic Equations** Chapter 9 Gauss Elimination Chapter 10 LU Decomposition and Matrix Inversion Chapter 11 Special Matrices and Gauss-Seidel Chapter 12 Case Studies: Linear Algebraic Equations	Deliverable 1: Research Paper Initial Proposal
10-13	**Curve Fitting** Chapter 17 Least-Squares Regression Chapter 18 Interpolation Chapter 19 Fourier Approximation Chapter 20 Case Studies: Curve Fitting	Deliverable 2: Dataset, Data Analysis, Mathematical Modeling
14	**Free Topics / Catch-up Week**	Deliverable 3: Research Paper Draft
15	**Finals**	Deliverable 4: Final Research Paper

RESEARCH PROJECT

You are required to search for research articles and related information on one of the applications of scientific computing in a selected domain, then to write a research paper that reviews all related concepts and techniques for the selected application area.

PROJECT OBJECTIVES

A project aims to train students to:

- Search for the precise information independently.
- Write a research paper based on the standard format.
- Obtain more specific ideas about scientific computing applications in real-world problems.
- Be able to provide new ideas based on information being reviewed

INSTRUCTIONS

- You are required to produce a **research paper** that critically discuss, review the existing works and propose your analysis regarding your selected application of scientific computing techniques in solving real-world problems. Here is a good source about the content and the structure of a generic research paper (https://explorable.com/research-paper-outline)
- You have to find **at least 4** related journal/conference articles from e.g. IEEE, ACM or any referred journal or conference publication related to your selected domain.
- You should **follow the due dates** determined in the course schedule of your course syllabus, and on Moodle for your deliverables.
- By completion of your research paper, you are highly encouraged to submit your paper to any local or international scientific computing related conference.
- You are free to select any application area as long as you are implementing at least one of main of the concepts of the course.

FORMAT

- Your paper must follow IEEE standard (.doc word template is given).
- Total pages should not exceed 6 pages
- Font times roman 10, single spacing.

EVALUATION RUBRIC				
Excellent (20%)	**Very Good (18%)**	**Good (15%)**	**Moderate (10%)**	**Weak (5%)**
• Fulfill all the requirements. • Number of papers reviewed is more than 4. • Paper content matches the title. • Do not exceed required pages. • Formatting is according to given template. • Comments/reviews/analysis written in the paper are **excellent** reflecting the references made.	• Fulfill all the requirements. • Number of papers reviewed is more than 4. • Paper content matches the title. • Do not exceed required pages. • Formatting is according to given template. • Comments/reviews/analysis written in the paper are good reflecting the references made.	• Fulfill all requirements • Number of journals reviewed is 4. • Paper content matches the title. • Do not exceed maximum pages. • Comments/reviews/analysis written in the paper are acceptable.	• Do not fulfill all requirements. • Number of journals reviewed is less than 4. • Paper content moderately matches the title. • Do not exceed required pages. • Comments/ reviews/analysis written in the paper are moderate.	• Do not fulfill all requirements • Number of journals reviewed is less than 3. • Paper content do not match the title • Do not meet required pages. • Comments/reviews/analysis given is weak, not based on facts and too simple.

STATEMENT ON ACADEMIC INTEGRITY

According to the University's Student Handbook: Academic integrity is the pursuit of scholarly activity free from fraud and deception, and is the educational objective of this institution. Academic dishonesty includes, but is not limited to cheating, plagiarism, fabrication of information or citations, facilitating acts of academic dishonesty by others, unauthorized possession of examinations, submitting work of another person, or work previously used without informing the instructor, or tampering with the academic work of other students. Any violation of academic integrity will be thoroughly investigated, and where warranted, punitive action will be taken. Students should be aware that standards for documentation and intellectual contribution may depend on the course content and method of teaching, and should consult the instructor for guidance in this area.

Honor Code - We as members of Harrisburg University community pledge not to cheat, plagiarize, steal, or lie in matters related to academic work. As a Community of Learners, we honor and uphold the ***HU Honor Code***.

CLASS PARTICIPATION

Without practical application and serious discussions, students often fail to comprehend the depths of the study material. The following guidelines express the focus of interactive learning styles:

- **Brainstorming –** Typically performed in group sessions. The process is useful for generating creative thoughts and ideas. Brainstorming helps students learn to pull together.
- **Think, Pair and Share –** Establish a problem or a question. Pair the students. Give each pair sufficient time to form a conclusion. Permit each participant to define the conclusion in his or her personal voice. Student are also would explain a concept while the other student evaluates what is being learned. Apply different variations of the process.
- **Buzz Session –** Participants come together in session groups that focus on a single topic. Within each group, every student contributes thoughts and ideas. Encourage discussion and collaboration among the students within each group. Everyone should learn from one another's input and experiences.
- **Incident Process –** This teaching style involves a case study format (mostly programming project), but the process is not as rigid as a full case study training session. The focus is on learning how to solve real problems that involve real people. Small groups of participants are given details from actual incidents and asked to develop a workable solution.

PROBLEMS ARISE

Problems happen to people when they are least expected. If any problems arise that you expect could impact your work in this class -- **PLEASE CONTACT ME AS SOON AS POSSIBLE! I want to see every student succeed -- but I can only help if you let me know as soon as possible!**

GRADING

Hands-on Assignments – At least six (8) assignments, worth a total of **60** points (3/5th of the final grade).

Research Paper – 40 points (2/5th of the final grade) **–** See research paper instructions.

Exams - There will be no exams in this course.