MANE 2110 section 01 Syllabus

NUM METH & PROG FOR ENG

Course Information

MANE 2110 01 : NUM METH & PROG FOR ENG

(3 Credits)

2023_Spring Term   Mech, Aerospace and Nucl Engr   School of Engineering

NUMERICAL METHODS AND PROGRAMMING FOR ENGINEERS

2023_Spring Term [202301]
Term Start Date: Monday, 9-Jan-2023 **Term End Date:** Wednesday, 21-Jun-2023

ADD TO CALENDAR

https://lms.rpi.edu/

Prerequisites or Other Requirements:

Prerequisite: ENGR-1100 Introduction to Engineering Analysis
Co-requisite: MATH-2400 Introduction to Differential Equations
Co-requisite: PHYS-1200 or PHYS-1250 Physics II

Please note that the co-requisites may not be enforced – you may have added this course without one or both of them. If you are not currently taking Introduction to Differential Equations, or have not previously successfully completed it, or are currently taking it but subsequently drop it, your success in this course may be jeopardized; and if you remain in this course you implicitly agree to do so at your own risk. Please see your instructor to gauge the effect of not having a pre- or co-requisite.

Additional Information for Course Information Section:

Section	Time	Instructor	Location	In-class TA
1	TF 10:00-11:50AM	Hicken	SAGE 4510	TBD
2	TF 12:00-1:50PM	Pouri	LOW 3039	TBD
3	TF 12:00-1:50PM	Pan	TROY 2012	TBD
4	TF 2:00-3:50PM	Pouri	LOW 3039	TBD
5	TF 2:00-3:50PM	Hicken	SAGE 3713	TBD

Instructor Information

Instructor

Jason Hicken
hickej2@rpi.edu

Additional Instructor Details:

- **Jason Hicken** (hickej2@rpi.edu): see LMS for office hours and contact medium
- **Sama Pouri** (rakhss@rpi.edu): see LMS for office hours and contact medium
- **Shaowu Pan** (pans2@rpi.edu): see LMS for office hours and contact medium

Teaching Assistant(s)

Teaching Assistant Information:

Name	**Email address**	**Office hours and location**
Chad Hess	hessc@rpi.edu	see LMS
Ruixiong “Ray” Hu	hur3@rpi.edu	see LMS
Vignesh Ramakrishnan	ramakv3@rpi.edu	see LMS
Nithin Somasekharan	nithinsomu@gmail.com	see LMS
Jonah Whitt	whittj4@rpi.edu	see LMS
Muiz Adekunle Agbaje	agbajm@rpi.edu	see LMS

Course Description

Additional Course Description :

This studio / learning laboratory course introduces techniques of numerical computing as a problem-solving method. Problems are drawn from the basic sciences (e.g., physics and chemistry) and the engineering sciences (e.g., statics). The numerical methods will be implemented through computer programming as both:

- a way of thinking (algorithms), and
- a language to translate mathematics into computer instructions.

Precision and accuracy, complexity, modularity, dimensionality, and discrete versus stochastic methods are covered.

Course Text(s)

Text Details:

No course text; extensive course materials are provided on LMS.

Supplemental References

These references are not used in this course but are widely available, respected, and may be of help as you progress through this course. Only purchase a text if you intend to use it.

1) *Practical Programming: An Introduction to Computer Science Using Python*   (≈ $32 eBook)
Jennifer Campbell, Paul Gries, Jason Montojo, Greg Wilson. Pragmatic Bookshelf; 2017.

2) *Fundamentals of Engineering Numerical Analysis*   (≈ $41 eBook)
Parviz Moin. Cambridge University Press; 2010.

Course Goals

Goals:

1) Identify and apply appropriate numerical methods to solve engineering problems.
2) Become proficient at computational thinking and writing software that solves engineering problems.
3) Prepare for relevant industry and graduate-school positions that rely on programming and numerical methods.

Course Content

Content Details:

1. Numerical Linear Algebra (numerical methods)
2. Root Finding (numerical methods)
3. Interpolation and Discrete Calculus (numerical methods)
4. Numerical Solution of ODEs (numerical methods)
5. Data Types and Variables (programming)
6. Control Flow (programming)
7. Objects, Classes, and Modules (programming)
8. Commenting, Debugging, and Testing (programming)

9. Input, Output, and Visualization (programming)
10. Compiled Languages (programming)

Learning Outcomes

Course (Student) Learning Outcomes (CLOs):

- After completing the "Numerical Linear Algebra" subject area, students will be able to: implement matrix-vector and matrix-matrix multiplication; solve linear systems involving upper- and lower-triangular matrices; implement LU factorization without pivoting; explain pivoting in the context of LU and why it is needed; implement QR factorization using modified Gram-Schmidt; define and identify over- and under-determined systems; use QR to solve fullrank least squares problems; and define and obtain (using NumPy) eigenvalues/eigenvectors of a matrix.

- After completing the "Root Finding" subject area, students will be able to: describe and implement the bisection method; describe and implement the secant method; describe and implement Newton's method for scalar functions; describe and implement Newton's method for nonlinear systems; and use Newton's method to solve implicit schemes applied to nonlinear ODEs.

- After completing the "Interpolation and Discrete Calculus" subject area, students will be able to: define and use Lagrange polynomials to interpolate data; describe the primary disadvantage of Lagrange interpolation; create finite-difference formulas using Taylor tables; state and apply the trapezoidal and midpoint rules for numerical integration; apply Gaussian quadrature given the nodes and weights; and evaluate numerical integration methods in terms of accuracy and cost.

- After completing the "Numerical Solution of ODEs" subject area, students will be able to: explain order of accuracy in the context of time-marching schemes; distinguish between stable, unstable, and conditionally stable schemes; analyze the stability of the forward/backward Euler methods; explain amplitude and phase error of a numerical solution; implement explicit time-marching schemes up to 4th-order accuracy; implement implicit time-marching schemes up to 2nd-order accuracy; describe how to apply implicit schemes to nonlinear ODEs; solve systems of ODEs using both implicit and explicit schemes; explain the concept of stiffness in ODE systems and its resolution; evaluate/compare time marching schemes in terms of efficiency; and solve ODEs for representative engineering problems and analyze the solution.

- After completing the "Data Types and Variables" subject area, students will be able to: explain and compare static and dynamic typing; select the appropriate data type for a given application; describe how floating-point numbers are represented and their limitations; explain machine epsilon, underflow, and overflow; use data types from standard libraries, e.g. lists, NumPy arrays; and explain memory allocation/deallocation and its implications.

- After completing the "Control Flow" subject area, students will be able to: use if/else statements to make branching decisions; explain and use for loops for iterating over integers or list elements; explain and use while loops for repeated operations; use break and continue as needed in loops; and create and use functions to create structured programs.

- After completing the "Objects, Classes, and Modules" subject area, students will be able to: distinguish between data and method attributes in Python; explain namespace and the related scoping rules in Python; define and use new classes in Python for engineering applications; use modules in Python to organize functions, classes; and explain and use inheritance in Python.

- After completing the "Commenting, Debugging, and Testing" subject area, students will be able to: use comments and doc strings to effectively document code; isolate bugs using print statements; use try statements to catch exceptions; and write effective tests to verify functions and classes.

- After completing the "Input, Output, and Visualization" subject area, students will be able to: read and write data to file; format data for ascii output; make use of external libraries, such as numpy and matplotlib; and visualize data by generating clear, concise, and correct professional quality 2-dimensional plots.

- After completing the "Compiled Languages" subject area, students will be able to: compile simple programs written in C or Fortran; and benchmark algorithms written in Python against C/Fortran

Course (Student) Learning Outcomes Assessment Measures:

Assessment	**Assignment Date**	**Learning Outcome Numbers**
Exam 1	Typically in the 6th week of class	1; 5, 6, 8
Exam 2	Typically in the 10th week of class	2, 3; 5, 6, 7
Exam 3	Final Exam Week	4; 5, 6, 7, 8, 9, 10
In-class Exercises	Every studio class	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
Homework	Approximately every 1-1.5 weeks	1, 2, 3, 4, 5, 6, 7, 8, 9, 10

Program Learning Outcomes:

BS-Aeronautical Engineering
Program Outcomes

1 **BS-AERO-01**
An ability to identify, formulate and solve complex engineering problems by applying principles of engineering, science and mathematics.

BS-Mechanical Engineering
Program Outcomes

2 **BS-MECH-01**
An ability to identify, formulate and solve complex engineering problems by applying principles of engineering, science and mathematics.

3 **BS-MECH-03**
An ability to communicate effectively with a range of audiences.

BS-Nuclear Engineering
Program Outcomes

4 **BS-NUCL-01**
An ability to identify, formulate and solve complex engineering problems by applying principles of engineering, science and mathematics.

5 **BS-NUCL-02**
An ability to apply the engineering design process to produce solutions that meet specified needs with consideration for public health and safety, and global, cultural, social, environmental, economic factors.

6 **BS-NUCL-04**
An ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts.

7 **BS-NUCL-06**
An ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions.

8 **BS-NUCL-07**
An ability to acquire and apply new knowledge as needed, using appropriate learning strategies.

Grading Criteria

Criteria Details:

In-class exercises: 20%

Homework assignments: 50%

Exams: 30%

The lowest four in-class exercise scores will be dropped from the calculation of the in-class exercise total. However: the homework assignments and subsequent exercises often build on in-class exercises so it may be detrimental to skip an exercise.

The grade thresholds are:

		A	≥ 94	A-	≥ 90
B+	≥ 87	B	≥ 84	B-	≥ 80
C+	≥ 77	C	≥ 74	C-	≥ 70
D+	≥ 67	D	≥ 64	F	< 64

This system establishes a common base grade scheme for the entire class. Based on previous semesters this is typically how course grades are assigned. However, the course team reserves the right to "curve up" at the end of the course by applying one or more alternative grading schemes to determine each student's final course grade:

1. As long as each scheme is applied equally to every student in the class; and
2. Each student receives the highest grade determined by all schemes so that no student's grade will be lower than the base grade scheme, only the same or higher.

Grade disputes: grade disputes for exams must be made in writing to your course instructor within 14 days of the grade being posted. For each problem being disputed please identify the question and why you believe the grade should be different. For disputes of assignment and exercise grades please first meet with the grader identified by Submitty to see if it can easily be resolved between you.

Course Calendar

Calendar Details:

Please see the tentative schedule posted on the LMS

Policies

Attendance Policy:

Attendance Policy

In-person attendance is expected given the studio format of the lessons and the numerous in-class exercises that you are expected to try to finish in class. Instructors may take attendance at their discretion and may use 0% to 4% of the 20% of the course grade allocated to in-class exercises and attendance to reward attendance.

In case of a switch to remote delivery:

During periods of remote delivery regular "attendance" may be problematic for some of you. Whether you attend live via WebEx during your scheduled class time, asynchronously through recordings (when available), and/or as deliberate and careful study with your fellow students of the in-class content and exercises, maintaining a regular schedule to learn the material and complete the in-class exercises on time can be vital to success in this course. Please practice good time management skills for this and all of your courses.

Other Course Policies:

Other Course Policies

In-Class Exercises

The approximately 41 In-Class Exercises (ICEs) must be successfully submitted to Submitty by 9:00pm the day they are assigned in class. Most ICEs prepare you for the assignments, so it is important to complete them in a timely manner. ICEs submitted later or not submitted will receive a zero. As noted previously, your lowest four in-class exercises will be dropped from your grade calculation.

Homework Assignments

The ten assignments are typically due one week after they are posted. Solutions may be reviewed in the class following the due date, so it is important that you finish them on time. However, we understand that work sometimes piles up such that even a short delay may be very helpful. Thus, you are assigned five late days that you may apply to any of the ten assignments, with the limit that you may only apply up to two late days to any one assignment. For example, you could be two days late on two assignments and one day late on a third, or one day late on five assignments, or some other combination not to exceed two days on any one assignment and five days overall.

Extensions

Extensions for assignments and in-class exercises will generally not be given without a valid excused absence from a coach, medical staff, Class Dean, or a representative of an Associate Dean (e.g., for Engineering Ambassadors).

Make-up Exams

Make-up exams will be provided with a valid excused absence only.

The course team may choose to extend deadlines uniformly for all sections of the course (e.g., for a snow day), but this should never be relied on.

Academic Integrity

Student-teacher relationships are built on trust. For example, students must trust that teachers have made appropriate decisions about the structure and content of the courses they teach, and teachers must trust that the assignments that students turn in are their own. Acts that violate this trust undermine the educational process. The Rensselaer Handbook of Student Rights and Responsibilities and The Graduate Student Supplement define various forms of Academic Dishonesty and you should make yourself familiar with these. In this class, all assignments that are turned in for a grade must represent the student's own work. In cases where help was received, or teamwork was allowed, a notation on the assignment should indicate your collaboration.

Violations of academic integrity may also be reported to the appropriate Dean (Dean of Students for undergraduate students or the Dean of Graduate Education for graduate students, respectively).

If you have any question concerning this policy before submitting an assignment, please ask for clarification. In addition, you can visit the following site for more information on our Academic Integrity Policy: Students Rights, Responsibilities, and Judicial Affairs.

Disability Services

Rensselaer Polytechnic Institute strives to make all learning experiences as accessible as possible. If you anticipate or experience academic barriers based on a disability, please let me know immediately so that we can discuss your options. To establish reasonable accommodations, please register with The Office of Disability Services for Students. After registration, make arrangements with the Director of Disability Services as soon as possible to discuss your accommodations so that they may be implemented in a timely fashion. DSS contact information: dss@rpi.edu; +1-518-276-8197; 4226 Academy Hall.

Disability Services for Students

Support Services

RPInfo - contains various resource links for students, academic resources, support services, and safety & emergency preparedness.

Rensselaer IT Services and Support Center

Additional Academic Integrity Course Policy and Penalty Information:

Academic Integrity

Student-teacher relationships are built on trust. For example, students must trust that teachers have made appropriate decisions about the structure and content of the courses they teach, and teachers must trust that the assignments that students turn in are their own. Acts that violate this trust undermine the educational process. The Rensselaer Handbook of Student Rights and Responsibilities and The Graduate Student Supplement define various forms of Academic Dishonesty and you should make yourself familiar with these.

In this class, all in-class exercises and homework assignments (that are turned in for a grade) must be significantly representative of your own work. No “group” hand-ins are permitted: as an individual you will submit your own code with your own substantial and significant individual content and style.

As noted previously, we do suggest the effective and equitable use of pair programming as a learning and achievement method. However, at the end of the process you must each have your own copy of your assignment in which you identify (e.g., in comments clearly marked toward the top of the main code file) your collaboration. Prior to submission you should further review and revise your code to match your unique style and approach to organizing the code, associated comments, and input and output format.

Specific to this course, academic dishonesty can take the following forms:

1) Taking an exam with any assistance from another person, regardless of who they are, unless the instructor specifically states that it is a group activity (e.g., a group oral exam).

2) Using any materials (such as a crib sheet or a calculator) that you did not bring for you and you alone to use on an exam, unless given explicit permission by the proctor (e.g., typically for extraordinary circumstances such as the failure of your calculator during an exam, etc., though such assistance should not be expected).

3) Using any materials (such as a text, crib sheets, a laptop, a cell phone, any electronic device – wireless or otherwise, etc.) that are not explicitly permitted during an exam.

4) Interaction with a wireless device (e.g., a cell phone or tablet) during an exam (except when explicitly allowed by the proctor) will be considered an illicit data exchange and will result in a zero for the entire exam.

5) Exiting the exam room without permission, or not returning in a prompt and timely manner when given permission. “Prompt and timely” is at the discretion of the instructional team.

6) Submission of an assignment that, except for short phrases or properly referenced material, consists of any content not created by and original to you.

7) Indicating someone as “present” (e.g., on an attendance sheet or submitting an in-class exercise) who is not present in class at the time the indication is made.

8) Collaboration is encouraged. However, assisting someone else with work where they intend to turn in the work as if it was done on their own will itself be considered a serious violation of academic honesty. Give credit for any assistance received, and have sufficient content of your own making as is expected for the assignment.

The penalty for submission of an exam that is in violation of this policy is failure of the course and referral to the Dean of Students.

Submission of an assignment that is in violation of this policy will result in a grade of zero on the assignment for the first offense. If there is a subsequent infraction the student will receive a grade of F for the course.

If you have any question concerning this policy before submitting an assignment, please ask for clarification.

Peer teaching is encouraged for the in-class exercises. However, you must write your own code. Verbal explanations are acceptable, but any exchange of files is not unless as part of an exercise in peer programming or peer code review involving the LMS discussion boards.

Other Course-Specific Information

Additional Course Information:

Meet the Instructional Team

Chad Hess, Teaching Assistant

I'm in my second semester as a PhD student in aeronautical engineering. I am working for the MOVE VTOL lab under Dr. Gandhi. I am also working on the aero team for Rensselaer Motorsport. Outside of class, I am a tournament chess player, speak French, and enjoy Formula 1 and hockey.

Jason Hicken, Instructor

I am interested in simulation-based design optimization, a field at the intersection of numerical analysis and optimization. Broadly speaking, my goal is to help engineers design complex fluid systems more effectively and efficiently using numerical simulations. To this end, my research expertise is in computational fluid dynamics, aerodynamic shape optimization, and multidisciplinary design optimization of aerospace systems. When I have time, and the wind is right, I like to go windsurfing.

Muiz Adekunle Agbaje

Muiz started his PhD in mechanical engineering at RPI in the Fall of 2022. Had his bachelor's and master's degrees from Nigeria and Turkey, respectively. His research focus is on the thermal-hydraulics of high-temperature novel energy conversion systems through component and system-level performance enhancement. Outside of class, he enjoys soccer, travelling, and trying out new things.

Jonah Whitt, Teaching Assistant

Jonah is in his first semester as a PhD student in aeronautical engineering working out of the MOVE laboratory. Jonah's career so far has focused on modeling and simulation for VTOL aircraft, and in his current work he's applying VTOL models to study vehicle configurations. Outside if school, Jonah plays for RPI's club hockey team, and enjoys playing MMO's.

Nithin Somasekharan, Teaching Assistant

Nithin is a first year PhD student starting spring 2023 in Aeronautical engineering. The focus of his research will be to explore ways to combine CFD and machine learning for computational analysis. Outside of class, he enjoys football (soccer), cooking, animes and swimming.

Ruixiong Hu, Teaching Assistant

Ruixiong Hu, 5th year PhD student in Mechanical Engineering, 4th time TA of NumPE class, working on writing code for numerical simulation of 3D printing technique. Expecting to graduate with CS degree as well so I'll be happy to answer questions on underlying programming logic.

Sama Pouri, Instructor

I have a Ph.D. and a master's degree in Nuclear Engineering from VCU, and a master's degree in Chemical Engineering from IUST. My dissertation was related to the comparison between Cyclic Voltammetry (CV) modeling within an Electrorefiner (ER) in Pyroprocessing technology using MATLAB software and the implementation of Artificial Neural Intelligence (ANI) on a massive CV experimental data set. I enjoy coding in Python, C++, and Matlab. If I find time, I watch movies, do puzzles/Legos, and read books.

Shaowu Pan, Instructor

Professor Shaowu Pan received his Bachelors in Aerospace Engineering and Applied Mathematics from Beihang University, China in 2013. After that, he received M.S. and Ph.D. in Aerospace Engineering and Scientific Computing from the University of Michigan, Ann Arbor in April 2021. Then he started as a Postdoctoral Scholar in the AI Institute in Dynamic Systems at the University of Washington, Seattle. His research interests lie in the intersection of computational fluid dynamics, data-driven modeling of complex systems, scientific machine learning, dynamical systems.

Vignesh Ramakrishnan, Teaching Assistant

I'm a second year Ph.D. student in aeronautical engineering. I currently work on design optimization of systems under chaotic dynamics. I'm currently working on the application of this to turbulence and CFD. Outside of work, I enjoy singing, playing soccer, painting and playing board games.