Program Cover Document -MAT 341: Computational Mathematics

I. Basic Course Information

MAT 341: Computational Mathematics is an upper-level course. It has two 80-minute meeting periods each week. The prerequisites are MAT 200 (Proof Writing through Discrete Mathematics), MAT 205 (Linear Algebra: Theory and Applications) and CSC 220 (Computer Science I: Computational Problem Solving) or CSC 250 (Accelerated Computer Science I, II). The MAT 200 prerequisite can be met with CSC 270 and permission from the chair. It is an option for all majors in the Department of Mathematics and Statistics and is one of two ways students in the Applied Mathematics specialization can fulfill their second computer programming requirement.

Computing is an essential part of modern mathematics. The partnership of applied mathematics, mathematics, and computational mathematics brings the tools of modeling, simulation, and data analysis to bear on real-world problems, producing solutions with the power to predict and explain complex phenomena. Computational methods are used in a wide variety of areas in mathematics, computer science, business, engineering, the natural sciences, and the social sciences. As a result, Computational Mathematics combines the beauty and logic of mathematics with the application of computing to solve mathematically modeled problems.

This course will help students develop the computational skills required to solve real-world problems. Significant work on topics drawn from core courses in mathematics that students have taken will be covered, but from a computer solution point of view. Students completing the course will be well prepared for the following opportunities:

- More advanced undergraduate study of computationally based mathematical topics
- Further training in professional masters or doctoral programs in applied mathematics and mathematics.
- Careers that require the ability to integrate computation and mathematical skills.

The goal of computational mathematics, put simply, is to find or develop algorithms that solve mathematical problems computationally (i.e. using computers). There are multiple programming languages, including but not limited to C++, Java, Python, Mathematica, Matlab, Maple, SAS, R and Excel/VBA, that facilitate solving computational mathematics problems. The choice of the best, or most appropriate, software platform in which to do programming should be completely determined by the applications being studied and the intended student audience. The primary focus of MAT 341 is on the mathematical algorithms and their implementation, and not on learning additional computer languages.

For Undergraduate Bulletin: Computational Mathematics combines the beauty and logic of mathematics with computing. In Computational Mathematics, students will learn how to develop and implement mathematical algorithms that can be utilized to solve real-world problems in many disciplines. Much of the course content will draw on topics from earlier mathematics courses, but these topics will be covered from a computer solution point of view.

II. Learning Goals

The primary learning goals of this course are for students to a) learn how to make a computer either solve a mathematical problem; b) gain insights through simulation into how one might solve a problem; c) use computation to gather data to help formulate and refine a mathematical conjecture; and d) understand how the computational complexity of an algorithm affects the usefulness of a particular computational approach. This course will build on mathematical topics students have been taught in core courses within their major from a theoretic approach and reexamine these same topics from a computational/algorithmic point of view. Specific objectives for the course are:

1. Students should gain an appreciation for the role of computers in mathematics, science, engineering and economics as a complement to analytical and experimental approaches.
2. Students should develop a knowledge of numerical approximation techniques, know how, why, and when these techniques can be expected to work, and have ability to program numerical algorithms
3. Students should have learned what computational mathematics is about: designing algorithms to solve scientific problems that cannot be solved exactly; investigating the robustness and the accuracy of the algorithms and/or how fast the results from the algorithms produce solutions. These items includes a basic understanding of computer arithmetic and round-off errors, how to avoid loss of significance in numerical computations, and computational complexity.
4. Students should be able to use and evaluate alternative approaches and algorithms for the solution to a computational problem, including the use of recursive algorithms, iterative algorithms, and where appropriate, closed form solutions.
5. Students should appreciate and demonstrate skills in oral, written and graphical communication, and know the importance of each.

The specific content goals of the course will be determined by the instructor, but it is expected that many of the following topics will be covered in the course:

- Functional and imperative programming languages.
- Functions and Programming; Differences between programming recursive, iterative, and closed form functions; Subroutines, user-defined Objects
- Computational Complexity
- Lists and Sets
- Recursive and Iterative Algorithms
- Numerical Algorithms and Accuracy. Specifically, Newton's methods, calculation of eigenvalue/vectors, determinants, solving differential equations, Euler's methods,
- Graph and Tree Algorithms. Specifically, Euler and Hamiltonian circuits, Traveling Salesperson problem, Greedy algorithm, Sorted edges methods, Kruskal's method.
- Search and Sorting Algorithms. Insertion, Bubble, Quick, and Heap Sorts; Bisection Algorithm
- Time Series Analysis

III. Student Assessment

To assess student understanding of the mathematical and computing topics covered in the course, feedback will be given to students through any of the following mechanisms: commented and/or graded homework, projects, computer programs, examinations, and in-class work.

IV. Learning Activities

In-class learning activities include lectures on mathematical and computer science concepts, discussion, group work, and instruction on programming language syntax and programming techniques. The course will primarily be project based and assignments will be made on each topic covered in the course that involve theory based work, paper and pencil computational work and significant computer programming. Specific activities and work will include the following: 1) Assignments based on each major topic, including written work, programming and in some cases oral presentation; 2) Written and/or oral examinations; 3) As a final assessment tool, either an individual or small group project to be completed, or a formal final cumulative examination will be administered.

(Created August 22, 2023)

MAT 341-01: Computational Mathematics
Fall 2023

Course and Instructor Details

Times:

Section 01:

Tuesday, Friday 2:00-3:20 PM,

Location: Physics & Mathematics P230
Instructor: Dr. John Nardini (I prefer to be called "Dr. Nardini")
Instructor pronouns: he/him/his
Instructor in-person office location: Physics & Mathematics P239
Office hours: Mondays 2-3:30 PM, Wednesdays 10-11:30 AM

> Please note that office hours are YOUR time! You are always welcome to come to my office during these times to further your understanding of class material, work on homework problems, discuss professional development opportunities, talk about a favorite Netflix show, or say hi! If you can't attend these hours, email me and we can set up another time to meet.

Instructor contact: You can contact me by email at NardiniJ@TCNJ.edu. I will respond within 24 hours (except on weekends).

Class Website: We will use the Canvas course management system, which you can access by going to https://tcnj.instructure.com, logging in, and then clicking on "MAT341-01," based on which section you're enrolled in.

Course text and Materials

There is no required course textbook, we will primarily use course lecture notes. The following text books are **optional**:

John T. Guttag. Introduction to Computation and Programming using Python. Third Edition. MIT Press, 2021. Link

Allen B. Downey. Think Python. Second Edition. O' Reilly Media, 2016. Link (includes link to free pdf download)

Gilbert Strang. Computational Science and Engineering. Wellesley Cambridge Press. First edition, 2019. Link

Catalog Description

MAT 341 (Computational Mathematics): Computational Mathematics combines the beauty and logic of mathematics with computing. In Computational Mathematics, students will learn how to develop and implement mathematical algorithms that can be utilized to solve real-world problems across many disciplines. Much of the course content will draw on topics from earlier mathematics courses, but these topics will be covered from a computer solution point of view (1 course unit)

Prerequisite: MAT 200, MAT 205, and CSC 220 or CSC 250 (The MAT 200 prerequisite can be met with CSC 270 and permission from the chair).

Course purpose

The primary learning goals of this course are for students to a) learn how to make a computer either solve a mathematical problem; b) gain insights through simulation into how one might solve a problem; c) use computation to gather data to help formulate and refine a mathematical conjecture; and d) understand how the computational complexity of an algorithm affects the usefulness of a particular computational approach. This course will build on mathematical topics students have been taught in core courses within their major from a theoretic approach and re-examine these same topics from a computational/algorithmic point of view.

Specific objectives for the course are:

1. Students should gain an appreciation for the role of computers in mathematics, science, engineering and economics as a complement to analytical and experimental approaches.
2. Students should develop a knowledge of numerical approximation techniques, know how, why, and when these techniques can be expected to work, and have ability to program numerical algorithms
3. Students should have learned what computational mathematics is about: designing algorithms to solve scientific problems that cannot be solved exactly; investigating the robustness and the accuracy of the algorithms and/or how fast the results from the algorithms produce solutions. These items includes a basic understanding of computer arithmetic and round-off errors, how to avoid loss of significance in numerical computations, and computational complexity.
4. Students should be able to use and evaluate alternative approaches and algorithms for the solution to a computational problem, including the use of recursive algorithms, iterative algorithms, and where appropriate, closed form solutions.
5. Students should appreciate and demonstrate skills in oral, written and graphical communication, and know the importance of each.

Course policies

Policies may change over time: All course policies are subject to change based on official TCNJ policies and public health concerns. Here are the current policies. If anything needs to be changed, I will announce the change during class time, send an updated policy to your TCNJ email, and update the syllabus posted on our course's Canvas website.

Class Recording: I may record some our regularly-scheduled courses over Zoom. These recordings will be posted to the course Canvas page, which you can use to enhance your studies. These recordings are only intended as a study resource and are not a substitute for synchronous in-person attendance of lectures.

Suggestion box: If you would like to comment on any weaknesses of the course that could be improved (or on strengths that you'd like to be continued), you can leave an anonymous suggestion at https://forms.gle/u2aQNGJkVh44LukAA. You must be logged into your TCNJ google account to submit a suggestion.

Class goals: Our classroom is intended to be an equitable and inclusive learning environment. If something about how the course is running is not working for you, you can reach out to me so we can fix that. If you prefer to leave anonymous feedback, you can do so in the suggestion box (see above).

Diversity in this classroom is a source of strength and should be respected and appreciated. We must all recognize the value that each individual brings to the class. Any speech or actions that do not serve this sense of community in our classroom will not be tolerated.

Class structure: This class will be a mix of interactive lecture and group work. It is highly encouraged that you actively participate in class and to treat it as a comfortable environment in which to ask questions, answer questions, and make mistakes.

> Asking questions during class is one of the best ways for you to solidify your understanding of material! I thoroughly appreciate when students ask questions because it helps me tailor our discussion to your learning style and helps all students. You can ask questions at any time, but I will also stop and ask for questions several times each lecture.

Distractions: Any behavior that can be deemed distracting to others should be minimized. Please make every attempt to arrive on time for class. If you must arrive late or leave early, do so without disrupting the lesson.

Attendance Policy: Homework, quizzes, and exams are based on material presented in class, so attendance during class is integral to learning the course material. Details on TCNJ's College Attendance Policy can be found at the following link: https://policies.tcnj.edu/?p=77.

Flexible deadlines policy: We are living through very uncertain times. While I encourage you to try and stay up-to-date on the course work, I will grant extensions if your life situation does not permit you to complete an assignment or quiz on time. When this happens, you can ask for the extension in-person during office hours or after class, or you can send me an email. The extension will be granted with no questions asked. If possible, please come to me before an assignment is due, but I understand this may not always be the case so you may ask for extensions after the deadline. Late exams are not allowed, except for emergencies. *The instructor reserves the right to change or modify this policy as needed.*

Academic integrity: You are expected to know the college's policy on academic integrity, which can be found at the following link: https://academicintegrity.tcnj.edu/. While I encourage you to work with your classmates on assignments, each write-up must represent your own work. While quizzes and exams will be open notes, it is cheating to collaborate with a peer, to use online mathematics tools, or to work with a tutor during a quiz or exam.

Assessment: The following formula will be used to calculate your final grade. In addition, if your average ends up on the border of two letter grades, very good/poor class participation can affect your final grade by one-third a letter grade (up/down). *The instructor reserves the right to change or modify this formula as needed.*

Participation	10%
Written Homework Assignments	25%
Written Groupworks Assignments	25%
Quizzes	20%
Final project	20%

According to TCNJ policies, we will use the following grading system:

	93-100 A	90-92.99 A-
87-89.99 B+	83-86.99 B	80-82.99 B-
77-79.99 C+	73-76.99 C	70-72.99 C-
67-69.99 D+	63-66.99 D	60-62.99 D-
0-60.99 F		

Participation 10% There are many possible ways for you to earn credit participating in the class. Your final participation grade will be computed out of a possible 100 participation points. The following table outline the activities you can perform to obtain participation points (with a maximum of 100). I am open to discussing other options for how you may earn participation points if you have an idea.

You are responsible for tracking the number of activities you complete over the course of the semester. At the end of a semester, you will complete a form detailing the activities you completed to receive your final participation score. I will double check all completed forms for accuracy.

Activity	**Points per engagement**	**Max # engagements (for points)**	**Possible points**
Post an introduction on padlet	5	1	5
Ask a question on Canvas	5	7	35
Respond to a question on Canvas	5	7	35
Post a meme involving class content on Canvas	5	2	10
Post a 1-3 page overview of a course topic on Canvas	15	2	30
Complete a 2-5 page research document expanding on class content	25	1	25
Write a 2-5 page report on a research article using a method from class	25	1	25
Create a visual representation of a class topic	15	2	30
Attend office hours (at least 10 minutes)	10	3	30

Written homework Assignments 25% We will have written homeworks due roughly every two weeks. You are expected to type up your response to each question on the assignment, which will include any visual graphs requested in the prompt. At the end of each assignment, you should include all python code that you used to compute your analyses and graphics. This code should be executable, meaning that if I were to run it from scratch, it would not produce any errors.

You are welcome and encouraged to complete the homework assignments with your peers, however, your turned in work, code, and graphics must be your own work.

Written groupwork Assignments 25% We will have several written groupwork assignments due throughout the semester. Your group is expected to turn in one report for all group members. Please type up your response to each question on the assignment, which will include any visual graphs requested in the prompt. At the end of each report, you should include all python code that you used to compute your analyses and graphics.

We will work on many group activities this semester, but ***not all group activities will require a written report.*** I will clearly communicate with the class which activities are to be turned in with a report. You are expected to participate fully in all groupwork activities.

Quizzes 20% We will have a 20 minute quiz every two weeks. The quiz will be open notes and you may use a non-graphing calculator, but not a tablet, laptop, or phone. Quiz topics will be announced at least 5 days in advance. Your two lowest quiz scores will be dropped at the end of the semester.

Final Project 20% We will have a final project in lieu of a final examination. Final project details will be provided at a later date.

Corrections to Grading If you believe an error has been made in grading on a test write a statement making your case and bring it to your instructor. I will give partial credit where appropriate. You have 1 week after a test is returned to request re-grading. Do not alter the original work!

Students in need of accommodations: Students who experience barriers in this course are encouraged to contact the instructor as early in the semester as possible. The Accessibility Resource Center (ARC) is available to facilitate the removal of barriers and to ensure reasonable accommodations. For more information about ARC, please visit: https://arc.tcnj.edu/. All documented accommodations will be respected, as specified by the Americans with Disabilities Act Policy (https://policies.tcnj.edu/?p=145).

Resources for success: While I encourage you to come to my office hours and contact me via email to work through any difficulty you have with the material, I realize that sometimes students' want a fresh perspective on the material. I highly recommend you study with your classmates and talk about material with one another.

TCNJ's Division of Student Affairs (https://sa.tcnj.edu) is designed to provide resources for students who face problems involving financial challenges, health/wellness, social concerns, or basic needs such as housing, food insecurity, or technology. If you do come across any of these this semester, I encourage you to share with me if you are comfortable. This is not necessary, but will help me be accommodating of your life's situation.

For Absence Due to Religious Observance: Students are expected to notify their instructors of anticipated absence for religious observance in advance of the date on which any absence will occur. As with other substantiated reasons for absence, and in consideration of the needs of our diverse

campus community, it is recommended that the instructor provide a fair and reasonable opportunity for work to be made up by the student, whenever possible.

Schedule (Tentative)

Day	Class Date	Topic		Day	Class Date	Topic
T	29-Aug	Python introduction		T	24-Oct	Finite difference methods
F	1-Sep	Python introduction		F	27-Oct	Math modeling: graphs
T	5-Sep	**No class: Labor day**		T	31-Oct	Math modeling: electric networks
F	8-Sep	Eigenvalues		F	3-Nov	Linear least squares
T	12-Sep	The power method		T	7-Nov	Linear least squares
F	15-Sep	The power method		F	10-Nov	Nelder-Mead Algorithm
T	19-Sep	Operation counts		T	14-Nov	Newton's method
F	22-Sep	Linear algebra code		F	17-Nov	Gradient descent
T	26-Sep	Random events		T	21-Nov	Nonlinear least squares
F	29-Sep	Random events		F	24-Nov	**No class: Thanksgiving**
T	3-Oct	Histograms & distributions		T	28-Nov	Nonlinear least squares
F	6-Oct	Monte Carlo methods		F	1-Dec	Neural networks
T	10-Oct	**No class: Fall break**		T	5-Dec	Neural networks
F	13-Oct	Monte Carlo methods		F	8-Dec	Parting words
T	17-Oct	Monte Carlo methods				
F	20-Oct	Math modeling: springs				