Universidad de Puerto Rico
Recinto de Mayagüez
Facultad de Artes y Ciencias
DEPARTAMENTO DE MATEMATICAS
Programa de Ciencias de Computadoras
Prontuario

Introduction to Modeling and Simulation using Python

Comp4999/4998 (Topics Course)

Horas crédito:	Horas contacto:
3	3 hora de conferencia
Requisitos previos:	**Requisitos concurrentes:**
MATE 3172	N/A

Descripción del curso (español):

Introducción a los principios de modelaje y simulación; introducción progresiva a los principios y destrezas de programación usando Python; aplicación a las destrezas de programación a la solución de diferentes clases de modelos.

Descripción del curso (inglés):

Introduction to the principles of modeling and simulation; progressive introduction of programming principles and skills using Python; application of programming skills to the solution of different classes of models.

Información de la Profesora:

Nombre	Ana C. González-Ríos
Horas de Oficina	Mi: 9:00am-12:00pm and 1:00pm-4:00pm
Oficina	OF316
Ext.	6182
Correo Electrónico	anacarmen.gonzalez@upr.edu
Portal del Curso	online.upr.edu

Objetivos del Curso (Course Goals):

1. Provide a background for more advanced modeling courses.
2. Provide the students with an introduction to modeling and its importance to current practices in different subject domains, like science, social sciences, and engineering.
3. Introduce programming principles and apply them to the solution of different classes of models.
4. Provide an overview of the modeling process and the terminology associated with modeling and simulation.
5. Study the mathematical representation of different classes of models.
6. Introduce techniques for fitting a function to an experimental data set.
7. Provide the opportunity for students to document the development and implementation of a model and presenting it in oral and written from.

Objetivos Instruccionales: (Course Learning Outcomes)

Upon successful completion of this course, students will be able to :

1. Explain the role of modeling in the sciences and engineering
    a. Describe the importance of modeling to science and engineering
    b. Describe the history and need for modeling
    c. Describe the cost effectiveness of modeling
    d. Describe the time-effect of modeling
2. Explain the terms of modeling in the sciences and engineering
    a. Define modeling terms
    b. List questions that would check/validate model results
    c. Describe future trends and issues in science and engineering
    d. Identify specific examples of modeling in science and engineering
3. Create a conceptual model
    a. Utilize the Modeling Process to identify key parameters of a model
    b. Estimate model outcomes
    c. Use Python to implement the mathematical representation of the model
4. Write code in a Programming language:
    a. Understand the concept of syntax in a programming language
    b. Describe the syntax of the programming language constructs
    c. Understand the difference between a compiled and interpreted language
    d. Write and run basic programs in the language of choice
    e. Understand how to de-bug code
    f. Understand the numerical limits of various data types and the implications for numerical accuracy of results
5. Use subprograms in program design:
    a. Describe how logical tasks can be implemented as subprograms
    b. Explain the control flow when a function is called
    c. Explain how function output is used
    d. Understand how languages handle passed data into functions and subprograms, especially one- and two-dimensional arrays
6. Understand and write Pseudo code
    a. List the basic programming elements of Pseudo code
    b. Explain the logic behind an if/then/else statement
    c. Understand the iterative behavior of loops
    d. Describe the difference between several looping constructs
    e. Write Pseudo code to solve basic problems
7. Describe the fundamentals of problem solving
    a. Understand Top-Down thinking and program design. Discuss breaking up a problem into its component tasks. Understand how tasks acquire data
    b. Describe how tasks should be ordered
    c. Represent tasks in a flow-chart style format
8. Use different approaches to data I/O in a program
    a. Explain the advantages and disadvantages of file I/O
    b. Describe the syntax for file I/O in your programming language
    c. Write code using file I/O and keyboard/monitor I/O
9. Understanding and use of fundamental programming Algorithms:
    a. Explain an algorithm as an ordered series of solution steps
    b. Describe an algorithm for a simple programming problem
    c. Describe what a software library is
    d. Construct difference-based computer models
10. Understand discrete and difference-based computer models

a. Write simple Python programs performing numerical calculations as needed for modeling and simulation
b. Implement finite difference modeling equations and create simulations in Python

11. Understand the use of empirical data
    a. Visualize empirical data and the fitting function using Python
    b. *Use Data Science techniques to illustrate data relevant to social change issues and interpret the results*

12. Understand the modeling process
    a. Identify different types of models and simulations
    b. Describe iterative development of a model
    c. Explain use of models and simulation for hypothesis testing
13. Verification and Validation
    a. Discuss methods for reviewing models their verification and validation
    b. Describe the differences between predictions of model, actual results and relevance of these differences to the problem
    c. Suitability/limits of models
14. Technical communication
    a. Document the development and implementation of a model and present it in oral and written form
15. Demonstrate computational programming
    a. Describe the computational programming system environment
    b. Define elementary representations, functions, matrices and arrays, script files,
    c. Explain relational operations, logical operations, condition statements, loops, debugging programs
    d. Create tabular and visual outputs
    e. Translate the conceptual models to run with this system and assess the model results

Representative Textbook:

Introduction to Modeling and Simulation with Matlab and Python , Steven I. Gordon and Brian Guilfoos, Chapman & Hall/CRC , ISBN: 13:978-1-4987-7387-4

Topics and time outline base on textbook: Introduction to Modeling and Simulation with Matlab and Python , Steven I. Gordon and Brian Guilfoos, Chapman & Hall/CRC , ISBN: 13:978-1-4987-7387-4

There might be slight modifications to this outline, students will be responsible to keep up to date.

Course Topics time, assessment, Learning Outcome mapping

Week	Topic	Assessment Activity/Evaluation/Homework/midterm/project	Learning Outcomes
1	Introduction to the course	Survey, syllabus	
2	Introduction to modeling; modeling concepts and definitions		1a,b,c,d 2a,b,c,d 13c 12c
3	Introduction to the Programming Environment		4a,b,c,d,e 9c
4	Deterministic Linear Models		3a,b,c 12a,b 4f
5	Array Mathematics and Python		15a,b 5d
6	Plotting in Python		11a,b 15d
7	Problem solving		7a,b,c 9a,b 6a,e
8	Conditional Statements		6b,15c
9	Iteration and Loops		4b,6c,d,e
10	Nonlinear and Dynamic Models		10a,b 12a,b 9d
11	No classes		
12	MidTerm (approximately 22 oct.)		
12-13	Estimating Models from Empirical Data		3a,b,c 8a,b,c 11a,b 15d,e
13	Proposal for final project(approximately 29 oct.)		
14	Stochastic Models		12a,b
15	Functions in Python		5a,b,c,d
16	Verification, Validation, and Errors		13a,b,c 12c
17-18	Project implementation, Students are given in class time to work is groups in their final project presentations (Thanks giving and last day of class)		14a

Class Format:

(Instructional Strategies)

Lectures will be based on video presentations, video demonstrations using the Python programming environment, walkthroughs.

All video presentations and demonstrations will be posted before the lectures.

The students will be asked to watch the videos before class, at times the video clips will be viewed during the class period, this will provide the bases for in-class discussions.

To effectively introduce programming skills and the modeling process, in-class exercises will be given. During the class period, the students will analyze problems and will be asked to develop algorithms and/or to implement the solutions. The students will work individually or in groups. This environment allows for the interaction of students among themselves and with the instructor. The instructor will have the opportunity to help and give feedback to the students about the design and implementation phases of the problem-solving process as well as in the modeling process.

At the end of each topic (group of lectures) there will be an assessment. For the programming and implementation aspects of the course the students will be required to use Python, and the assessment will be a written report. Templates for the report will be provided. Submitting a written report will prepare the students in documenting the development and implementation of a model which will be require as part of the final project of the course.

Assignments:

After the discussion of each topic is completed, the students will be required to complete a quiz, the assessment will be online and could include multiple choice questions as well as open-ended questions.

For the practical part of the course, the students will complete a laboratory report. A template will be provided for the report. As part of the report the students will have to submit code and describe the solution to the problem , the development of a model ,and explain the modeling process.

For the different activities there will be an in class announcement of whether or not it will be counted toward your final grade.

Nota Importante:

- Durante los períodos de clase, se espera uso de cortesía para tratar a sus compañeros y a la profesora.

- Se les pide a los estudiantes que lleguen a tiempo a la clase para así no interrumpir y que eviten (a menos que sea una emergencia) el estar entrando y saliendo del salón, además es importante que no esté constantemente hablando con sus vecinos ya que esto distrae tanto a sus compañeros como a la profesora.

- No se permite el uso del celular, durante la clase ni durante las evaluaciones

- Si usted no puede cumplir con la fecha pautada para una evaluación, sea ésta, un examen, tarea o prueba corta, deberá presentar una carta a la profesora donde explica sus razones, la carta debe estar acompañada de certificado médico y cualquier otra documentación que corresponda. La profesora evaluará el caso y tomará una decisión. De ser necesario un examen de reposición se ofrecerá el 4 de diciembre de 2019.
- Los exámenes serán comenzados y terminados durante el periodo indicado por la profesora

- Para obtener crédito por los trabajos éstos deben ser entregados en la hora y fecha acordada

Estrategias de evaluación:

Course Grading

One written examination that should be completed during the class period. Approximately 10 quizzes/activities and approximately 10 laboratories/activities reports. Each will have a value of 10 points. One final project, as part of the final project the students will have to present a written proposal for the work they will be developing and present the references that they plan to use. This proposal will be submitted by the middle of the semester and will be part of the project grade. Also, as part of the final project the students will have to give an oral presentation and submit a written report. Rubrics will be provided.

Description	Points	Percentage
Written examination	100	30
Quizzes(cumulative)	100	20
Laboratory report(cumulative)	100	20
Final Project (including proposal)	100	30

Rererences:

1. Python for Informatics, Exploring Information , Author: Charles Severance Available on ibook ,Published: August 13,2014

2. IPython Interacttive Computing: (http://ipython.org)

3. http://matplotlib.org/users/pyplot_t utorial.html

Topics and subtopics

- Introduction to modeling; modeling concepts and definitions
  - ✓ Brief review of the history or computational modeling and its contributions to the advancement of science
  - ✓ Overview of the modeling process and the terminology associated with modeling and simulation
- Introduction to the Python Programming Environment
  - ✓ Introduction
  - ✓ Installation
  - ✓ Spyder Interface
  - ✓ Basic syntax, variables and operators, Python Mathematical Operators, Order of Execution for Mathematical Operations in Python, Keywords
  - ✓ List and Arrays
  - ✓ Loading Libraries
- Deterministic Linear Models
  - ✓ Selecting a mathematical representation for a model
  - ✓ Linear models and linear equations
  - ✓ Linear interpolation
  - ✓ System of linear equations
  - ✓ Limitation of linear models
- Plotting in Python
  - ✓ Matplotlib library
  - ✓ Ipython graphics backend setting to Automatic
  - ✓ Examples of a 2D plot in Python
- Problem solving
  - ✓ Engineering method to approach solving problems
  - ✓ Bottle filling examples
  - ✓ Tools for program development:
    - Pseudocode
    - Top-Down design
    - Flowcharts
  - ✓ Bottle filling example continued
- Conditional Statements
  - ✓ How to select one of several possible program paths, distinct blocks of code to execute base on a conditional
- An evaluation of a variable against some test state
  - ✓ An evaluation of a variable against some test state
  - ✓ Relational operators
  - ✓ Logical operators
  - ✓ Conditional statements
  - ✓ Python
- Iterations and Loops
  - ✓ Use to calculate each time step of a simulation, or to step though the elements of an array or matrix to do element-wise calculations
  - ✓ Python loops
  - ✓ Python while loops
  - ✓ Controls statements

- Nonlinear and Dynamic Models
  - ✓ Examine the components of dynamic systems, extending our approach to creating a conceptual model to reflect that dynamism
  - ✓ Review several examples of the mathematical representations of nonlinear systems

- ✓ Illustrate several mathematical approaches to modeling nonlinear dynamic systems.
- ✓ Examples of dynamic systems and the computer algorithms that can used to model those systems

- Estimating Models from Empirical Data
  - ✓ Using data to build forecasting models
  - ✓ Limitations of Empirical Models
  - ✓ Fitting a mathematical function to data
  - ✓ Fitting a linear model
  - ✓ Linear models with multiple predictors
  - ✓ Nonlinear model estimation
  - ✓ Nonlinear fitting and regression
  - ✓ Segmentation
- Stochastic Models
  - ✓ Introduction
  - ✓ Creating a stochastic model
  - ✓ Random number generators in Python
  - ✓ Code example
  - ✓ Examples of larger scale stochastic models
- Functions
  - ✓ Python functions
  - ✓ Python Modules
- Verification, Validation , and Errors
  - ✓ Introduction
  - ✓ Errors
  - ✓ Truncation codes
  - ✓ Euler's method
  - ✓ Runge-Kutta method
  - ✓ Verification and validation
  - ✓ History and Definitions
  - ✓ Verification Guidelines
  - ✓ Validation guidelines
  - ✓ Quantitative and Statistical validation measures
  - ✓ Graphical methods
- Capstone Projects

Ley 51: Ley de Servicios Educativos Integrales para Personas con Impedimentos: Después de identificarse con el profesor y la institución, los estudiantes con impedimento recibirán acomodo razonable en sus cursos y evaluaciones. Para más información comuníquese con *Servicios a Estudiantes con Impedimentos* en la Oficina del Decano de Estudiantes, 787-265-3862 ó 787-832-4040 x 3250 ó 3258.

Ley 51: Ley de Servicios Educativos Integrales para Personas con Impedimentos: 5 Después de identificarse con el profesor y la institución, los estudiantes con impedimento recibirán acomodo razonable en sus cursos y evaluaciones. Para más información comuníquese con Servicios a Estudiantes con Impedimentos en la Oficina del Decano de Estudiantes, 787-265-3862 ó 787-832-4040 x 3250 ó 3258.

Asistencia:
Asistencia a clases: La asistencia a clases es compulsoria. La Universidad de Puerto Rico, Recinto de Mayagüez se reserva el derecho de tratar en cualquier momento con casos individuales de falta de asistencia. Se espera que los profesores tomen nota de la asistencia de sus estudiantes. Las ausencias frecuentes pueden afectar la calificación final e incluso puede resultar en la pérdida total de créditos. Es responsabilidad del estudiante el hacer arreglos para reponer trabajos atrasados por razón de alguna ausencia legítima a clases. (Boletín Informativo de Estudios Subgraduados, página 39, 1995-96)
En cumplimiento con las directrices establecidas por la oficina del Registrador, se tomará la asistencia diariamente

Ética: Cualquier fraude académico está sujeto a sanciones disciplinarias descritas en los artículos 14 y 16 del Reglamento General de Estudiantes, según revisado y contenido en la Certificación 018-1997-98 de la Junta de Síndicos. El profesor seguirá las normas establecidas en los artículos 1-5 del Reglamento
 Se considera comportamiento inaceptable cualquier actividad que interfiera con el aprendizaje y la enseñaza: hablar en clase, abrir páginas que no sean de la clase, presentar material de otro como si fuera su trabajo, usar celular, oír música.