



---

<b>Professor</b>	Sebastian Bustamante
<b>E-mail</b>	sbustamante.academic at gmail.com
<b>Classroom</b>	Computer room 6-125
<b>Class Time</b>	( <b>Group 1</b> M-J 8.00-10.00 am), ( <b>Group 2</b> W-F 8.00-10.00 am)
<b>Advisory</b>	–

## Introduction

This course is intended for students of Astronomy at the Universidad de Antioquia and will cover some numerical methods commonly used in science and specially in astronomy. These topics will be addressed from a formal context but also keeping a practical and computational approach, illustrating many useful applications in problems of physics and astronomy.

## Motivation

As we understand more deeply our surrounding world, the involved phenomena exhibit an ever increasing complexity and numerical solutions have become more and more common in physics and astronomy as an alternative (and a complement) to analytic solution. It is thus necessary to well understand the capabilities and limitations of these numerical methods as well as the type of problems where they can be implemented.

## Methodology

Although a formal mathematical approach of the numerical methods is necessary, including deductions and error analysis, an *algorithmic thinking* is more than mandatory when we want to apply these methods to some specific problem. In this context, *algorithmic thinking* may be defined as the ability of abstract thinking in such a way we can tell the computer clearly how to do what we want it to do. Although a computer is not really required when applying numerical methods, the massive number of involved calculations makes more practical computational implementations.

As cycling, programming is a matter of practice, so in order to develop this *algorithmic thinking* there will be a strong practical component. This practical component will be almost entirely developed in *Python* and slightly less in *C* (when computational performance

is required). However, students with knowledge in other programming languages (except privative languages like MatLab, Mathematica) are also aimed to use them.

The course will be addressed in two steps each class: first, a formal introduction of the numerical methods will be presented, including deductions (when appropriated) and error analysis. For this part the students are expected to consult by their own before class the respective topic, so they can go quickly to the practical part. Secondly, the practical part will cover short applications of the presented methods as well as some examples developed by the professor during class.

## **Material**

All the material of the course will be maintained in the next repository:

<https://github.com/sbustamante/ComputationalMethods>

This syllabus along with notes, presentations, solved examples, extra material and homework can be found there. The repository will be updated as the course advances.

## **Evaluation**

The course is divided into three main blocks: the first includes numerical methods for general algebra; the second block is related to numerical methods for calculus and linear algebra; finally the third block is about numerical methods for differential equations and statistics. Each one of these blocks will be evaluated with an exam of 25%. Additionally, weekly homework will cover the last 25%.

For the homework, some bonuses will be taken into account from time to time, for example the faster code, the more accurate, and so on. These bonuses would add 0.5 to the grade of the respective task. The three exams will be held during class time and may involved computational activities as well as analytical work.

# **Program**

## **0. Overview of Python**

Basic commands of Python. Scripting. Scientific Libraries (Numpy, Scipy). Plotting with Matplotlib. IPython notebooks.

## **1. Mathematical preliminaries**

Round-off errors. Computer arithmetic. Algorithms and convergence.

## **2. One variable equations**

Transcendental equations. Bisection method. Fixed-Point iteration. Newton's Method. Error analysis. Zeros of polynomials. Applications.

## **3. Interpolation**

Linear interpolation. Lagrange polynomials. Divided differences. Hermite interpolation. Cubic spline interpolation. Applications.

## **4. Numerical calculus**

Numerical differentiation. Numerical integration. Composite numerical integration. Adaptive Quadrature Methods. Multiple integrals. Improper Integrals. Applications.

## **5. Linear algebra**

Linear systems of equations. Iterative techniques. Inverses and determinants. LU factorization. Cholesky factorization. Pivoting strategies. Applications.

## **6. Differential equations**

First order methods: Euler, leap frog. Second order methods: Runge Kutta methods. Systems of differential equations. Boundary and initial conditions. Applications.

## **7. Statistics**

Data adjust. Least square and non-linear least square. Random numbers. Monte Carlo techniques. Descriptive statistics. Applications.

# Schedule

Class	Date (1/2)	Topics
0	Apr 21/22	Presentation of the course. <b>0. Overview of Python</b>
1	Apr 23/24	Overview. Basic scripting.
2	Apr 28/29	Scientific libraries. Matplotlib.
3	Apr 30/01	IPython notebooks. <b>1. Mathematical Preliminaries</b>
4	May 05/06	Computer arithmetic. Round-off methods.
5	May 07/08	Algorithms and convergence. <b>2. One variable equations</b>
6	May 12/13	Bisection method.
7	May 14/15	Fixed-point iteration.
8	May 19/20	Newton's methods. Error analysis. <b>3. Interpolation</b>
9	May 21/22	Linear interpolation. Lagrange polynomials.
10	May 26/27	Divided differences.
11	May 28/29	Hermite interpolation. Cubic spline interpolation. <b>4. Numerical Calculus</b>
12	Jun 02/03	Numerical differentiation.
<b>E1</b>	Jun 04/05	<b>EXAM 1 (Topics 1, 2 and 3) (Sessions 1-11)</b>
13	Jun 09/10	Numerical integration. Composite numerical integration.
14	Jun 11/12	Adaptive quadrature methods.
15	Jun 16/17	Improper integrals. <b>5. Linear Algebra</b>
15	Jun 18/19	Linear systems of equations. Gaussian elimination.
16	Jul 07/08	Pivoting strategies. Inverses and determinants.
17	Jul 09/10	LU factorization. Cholesky factorization. <b>6. Differential Equations</b>
18	Jul 14/15	First order methods: Euler, Leap frog.
<b>E2</b>	Jul 16/17	<b>EXAM 2 (Topics 4 and 5) (Sessions 13-17)</b>
19	Jul 21/22	Second order methods: Runge Kutta methods.
20	Jul 23/24	Systems of differential equations.
21	Jul 28/29	Boundary initial conditions. <b>7. Statistics</b>
22	Jul 30/31	Data adjust. Least square.
23	Aug 04/05	Random numbers. Descriptive Statistics
24	Aug 06/07	Monte Carlo techniques. Descriptive Statistics
<b>E3</b>	Aug 13/14	<b>EXAM 2 (Topics 6 and 7) (Sessions 18-24)</b>

## Bibliography

- *Numerical Analysis*, Richard L. Burden & J. Douglas Faires. Ninth edition. 2011.
- *Numerical Recipes, the Art of Scientific Computing*, William H. Press, Saul A. Teukolsky, William T. Vetterling & Brian P. Flannery. Third edition. The Cambridge University Press. 2007.
- *Introduction to Computational and Programming Using Python*, Gutttag, J. V. The MIT Press. 2013.