

MÉTODOS COMPUTACIONALES

Programa basado en el desarrollado por: Jaime E. Forero Romero

NOMBRE DEL CURSO: Métodos Computacionales
UNIDAD ACADÉMICA: Departamento de Física
PERIODO ACADÉMICO: 201719 (intersemestral)
HORARIO (MAGISTRAL): Martes a Viernes de 8:00 a 9:50

NOMBRE PROFESORA PRINCIPAL: Verónica Arias Callejas
CORREO ELECTRÓNICO: v.arias@uniandes.edu.co
HORARIO Y LUGAR DE ATENCIÓN: con cita previa por correo

I Introducción

Los métodos computacionales son un aspecto inseparable de cualquier área de trabajo en ciencia e ingeniería. Esto se debe en gran parte a la disminución en costos y complejidad de uso de las computadoras programables, unido al aumento exponencial en su capacidad de procesamiento. Este curso busca guiar a los estudiantes en el desarrollo de sus *habilidades computacionales* en generar y procesar datos para obtener información sobre la realidad que esos datos pretenden describir. Estos datos pueden ser mediciones o simulaciones de sistemas físicos, químicos, geológicos, biológicos, financieros o industriales, entre otros. El programa del curso tiene dos componentes diferenciadas: métodos numéricos y *carpintería* de software. La parte de métodos numéricos busca ilustrar el paso entre la formulación de una pregunta sobre la realidad en términos matemáticos y su descripción numérica/computacional, para mostrar posibles formas de escribir esa pregunta con software. La parte de carpintería busca presentar algunas técnicas y prácticas necesarias (no suficientes) para poder obtener resultados computacionales reproducibles.

II Objetivos

El objetivo principal del curso es presentar algoritmos y técnicas básicas para:

- resolver ecuaciones diferenciales ordinarias, parciales y estocásticas,
- analizar y describir datos con técnicas estadísticas basadas en métodos Monte Carlo,
- desarrollar esquemas reproducibles para el análisis de datos científicos.

III Competencias a desarrollar

Al finalizar el curso, se espera que el estudiante esté en capacidad de:

- manejar dos lenguajes de programación modernos de computación numérica: uno de alto nivel y otro de bajo nivel,
- implementar algoritmos sencillos para la resolución de ecuaciones diferenciales y para el análisis estadístico exploratorio de datos,
- desarrollar un esquema sencillo para obtener resultados computacionales reproducibles.

IV Contenido por semanas

Semana 1

- Temas: Presentación del curso. Unix. Consola. Comandos básicos. Editores de texto. Intérprete de Python. Variables. Aritmética. Listas. Diccionarios.

- Lecturas preparatorias. Videos Introducción a Unix: primera parte, Introducción a Unix: segunda parte, Introducción a Python: primera parte, Python: listas y strings.
- Temas: If/while/break/continue. Ciclos. Funciones. Arreglos (numpy). Lectura y escritura de archivos (numpy). Gráficas y visualización (matplotlib), Objetos.
- Lecturas preparatorias. Videos Intruducción a Python: segunda parte, Programación Orientada a Objetos en Python, Python: Funciones, tipos de variables y recursividad., Numpy con IPython, import matplotlib as plt (los dos últimos videos usan Ipython, no es necesario que uds. lo hagan asi.).

Semana 2

- Temas: Integrales y derivadas.
- Lecturas preparatorias: Capítulos 6 (Integration) y 7.I (Numerical Differentiation) del libro de Landau.
- Temas: Solución de ecuaciones lineales y no lineales.
- Lecturas preparatorias: Capítulos 7.II (Trial-and-Error Searching) y 8 (Matrix Equation Solutions) del libro de Landau.

Semana 3

- Temas: Autovalores y autovectores. Principal Component Analysis.
- Lecturas preparatorias: Secciones 6.3.1 (Principal Component Regression) y 10.2 (Principal Component Analysis) del libro ISL.
- Temas: Transformadas de Fourier
- Lecturas preparatorias: Capítulo 10 (Fourier Analysis) del libro de Landau.
- Talleres: Entrega Taller #1.

Semana 4

- Temas: Ecuaciones diferenciales ordinarias.
- Lecturas preparatorias: Capítulo 9 (ODEs) del libro de Landau.
- Temas: Ecuaciones diferenciales parciales.
- Lecturas preparatorias: Capítulo 17 (PDEs) del libro de Landau.
- Talleres: Entrega Taller#2.

Semana 5

- Temas: C.
- Lecturas preparatorias: Capítulos 1 (Tutorial Introduction), 2 (Types, Operators and Expressions), 3 (Control Flow) y 4 (Functions and Program Structure) de Kernighan&Ritchie
- Temas: C. Git. Github.
- Lecturas preparatorias: Capítulos 5 (Pointers and arrays) y 7 (Output) de Kernighan&Ritchie. Videos de la serie **Version Control with Git** en Software Carpentry.

Semana 6

- Temas: Makefile. Unit Tests.
- Lecturas preparatorias: Videos de la serie **Automation and Make** en Software Carpentry.
- Temas: Conceptos básicos de probabilidad y estadística. Procesos aleatorios.
- Lecturas preparatorias: Capítulo 1 (The Basics) del libro de Sivia&Skilling.
- Talleres: Entrega Taller #3.

Semana 7

- Temas: Métodos de Monte Carlo.
- Lecturas preparatorias: Capítulo 1 (Monte Carlo Methods) del libro de Krauth.
- Temas: Estimación de parámetros en estadística bayesiana.
- Lecturas preparatorias: Capítulos 2 (Parameter Estimation I) y 3 (Parameter Estimation II) del libro de Sivia&Skilling.

Semana 8

- Temas: Ecuaciones diferenciales estocásticas.
- Lecturas preparatorias:
- Talleres: Prueba Final Escrita (requisito para entregar el Taller #4).
- Talleres: Entrega Taller #4.

V Metodología

Las *habilidades computacionales* se desarrollan trabajando activamente. Por esto en las sesiones magistrales, luego de presentar un resumen de los conceptos teóricos, se hará énfasis en la práctica y experimentación. Para que esto funcione es necesario que los estudiantes estudien el tema correspondiente **antes de cada clase** siguiendo las lecturas preparatorias recomendadas. En el *Laboratorio de Métodos Computacionales* habrá más tiempo para practicar lo visto en la clase magistral, hacer ejercicios y aclarar dudas.

La Magistral y el Laboratorio cuentan con repositorios en GitHub:

<https://github.com/ComputoCienciasUniandes/MetodosComputacionales>.

<https://github.com/ComputoCienciasUniandes/MetodosComputacionalesLaboratorio>.

VI Criterios de evaluación

La Magistral y el Laboratorio se califican por separado. Habrá exámenes escritos, talleres y ejercicios.

Al comienzo del semestre se hará un examen (corto, sin nota) para diagnosticar el conocimiento general que ya tienen los estudiantes sobre los temas del curso.

En la Magistral habrá 4 evaluaciones con un valor del 25 % de la nota definitiva. Cada una de estas evaluaciones está compuesta de un taller para resolver individualmente en casa y un quiz que se hace en la clase siguiente a la entrega de cada taller. Si la nota del quiz es menor a 3.0, esa nota será tomada como la nota total de la evaluación. Si la nota del quiz es mayor a 3.0 la nota de la evaluación será un 20 % correspondiente a la nota del quiz y un 80 % correspondiente a la nota de la tarea. También hay un bono de 10 % que consiste en una prueba escrita sobre los temas vistos durante el semestre.

En el Laboratorio habrá cinco ejercicios cortos para desarrollar y entregar en clase. Cada ejercicio corresponde a un 20 % de la nota final del Laboratorio.

De acuerdo a la nota definitiva en el Laboratorio habrá **otro bono** en la nota definitiva de la Magistral. Siendo x la nota de Laboratorio, el bono correspondiente se calcula así: $4,0 < x \leq 4,4 \rightarrow 0,1$, $4,4 < x \leq 4,8 \rightarrow 0,2$, $4,8 < x \leq 5,0 \rightarrow 0,4$.

Todos los exámenes, talleres y ejercicios serán **individuales**. Si en las entregas se detecta que el trabajo no fue individual (esto incluye colaboración con personas no inscritas en el curso), entonces la nota de todos los talleres quedará automáticamente **en cero** y se llevará el caso a comité disciplinario.

Todas las entregas de talleres y ejercicios se harán a través de SICUA. **No se aceptará ninguna tarea por fuera de esa plataforma**, a menos que ocurra una falla en los servidores de SICUA que afecte a **todos** los estudiantes del curso.

VII Bibliografía

Bibliografía principal:

- *A survey of Computational Physics - Enlarged Python Book* . R. H. Landau, M. J. Páez, C. C. Bordeianu. WILEY. 2012.
- *Data Analysis: A Bayesian Tutorial*. D. S. Sivia, J. Skilling. Second Edition, Oxford Science Publications. 2006.
- *Statistical Mechanics: Algorithms and Computations*. W. Krauth, Oxford Univ. Press.
- *An Introduction to Statistical Learning*. G. James, D. Witten, T. Hastie, R. Tibshirani, Springer. <http://www-bcf.usc.edu/~gareth/ISL/>
- *The C programming language*. B. Kernighan & D. Ritchie, Second Edition, Prentice Hall.
- Videos del curso Herramientas Computacionales que muestran los fundamentos de Unix y Python https://www.youtube.com/playlist?list=PLHQztzvthdVM_MGC9dPFKe4hPAwBd_7RJ3
- <http://software-carpentry.org/>

Bibliografia secundaria:

- *Elements of Scientific Computing* Tveito A., Langtangen H.P., Nielsen B.F., Cai X. Springer. 2010.
- *Introduction to Computation and Programming Using Python*, Gutttag, J. V. The MIT Press. 2013.
- <http://xkcd.com/>