Universidad de los Andes

Departamento de Física

Métodos Computacionales

Nombre del curso: Métodos Computacionales
Código del curso: FISI 2028 (Magistral) FISI 2029 (Laboratorio)
Unidad académica: Departamento de Física
Periodo académico: 202120
Horario (magistral): Mi y Vi, 17:00 a 18:15.

Nombre profesor magistral: Juan Pablo Mallarino Robayo
Correo electrónico: jp.mallarino50@uniandes.edu.co
Horario de atención: Miercoles 15:00 a 16:00, Microsoft Teams

Nombre profesor Laboratorio: Diego Hernando Useche Reyes
Correo electrónico: dh.useche@uniandes.edu.co
Horario de atención: a disposición del profesor.

I Introducción

Los métodos computacionales son un aspecto inseparable de cualquier área de trabajo en ciencia e ingeniería. Esto se debe a la facilidad de acceso a computadoras programables y al aumento exponencial en su capacidad de procesamiento. Este curso busca guiar a los estudiantes en el desarrollo de una estructura de pensamiento computacional parar obtener o generar y procesar información sobre la realidad que los datos describen. 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: *C/C++* (compilación) y algoritmos (estrategias) de programación, y métodos numéricos para resolver problemas científicos. Ambas partes ilustran el paso entre la formulación matemática de una pregunta a su descripción numérica. El objetivo principal es emplear el lenguaje para formular problemas científicos. La práctica es fundamental y, no menos importante, reproducibilidad.

II Objetivos

Los objetivos principales del curso son:

- Desarrollar en los estudiantes una adecuada actitud computacional, con la capacidad de discernir sobre los métodos adecuados para solucionar cualquier problema y entender sus limitaciones.
- Profundizar sobre dicha actitud computacional que corresponde al conjunto de habilidades para trabajar con computadores en generar y procesar datos que correspondan a sistemas físicos, donde estos datos corresponden a una medición o una simulación
- 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:

- Entender y poder realizar compilación de programas de su propiedad o de cualquier fuente, en el lenguaje preferido.
- Poder analizar una serie de datos con los métodos aprendidos en clase.
- Realizar simulaciones numéricas con aplicación en Física u otras ciencias.

IV Contenido por semanas

Semana 1. Presentación del curso. Repaso Unix.

- Temas: El programa, la consola de comandos (bash scripting y comandos básicos), preliminares de arquitectura de computación, y ¿Compilación? (namespaces y "Hello world!").
- Reto: GodBolt

Semana 2. Introducción a programación en C/C++. Diferencias, lectura de archivos. Iteración

- Temas: Cargando librerías, stdout, stderr, streams, scopes, variables (`typename`), condicionales y bucles.
- Retos: GetOpts, `#define`, `++i` vs `i++`

Semana 3 y 4. Funciones. Recursividad. Manejo de memoria. Punteros.

- Temas: "arrays", stack vs heap, aritmética de punteros.
- Reto: memoization, dynamic programming, LCS.

Semana 5. Estructuras y Sistemas de control de versiones.

- Temas: `struct`, `union`/`enum`, `class`, Git y Github.
- Reto: Crear nuestra librería de física matemática y hacer PR.

Semana 6 y 7. Algebra lineal. Descomposicón LU. Ajustes de mínimos cuadrados. Vectores y valores propios. Análisis de componentes principales (PCA).

- Reto: Gram–Schmidt y Gauss–Jordan

Semana 8. Interpolación.

- Reto: Suavisado de funciones complejas

Semana 9. Análisis de Fourier. Transformada rápida de Fourier.

Semana 10. Integración y derivación numérica. Métodos Monte Carlo para integración. CMake y Makefiles.

- Reto: Cuadraturas Gaussianas

Semana 11 y 12. Ecuaciones diferenciales ordinarias de 1er y 2do orden. Runge–Kutta de 4to orden.

- Reto: Péndulos acoplados, Sol–Tierra–Luna (triángulo de Laplace), y el método de diferencias finitas (FEM).

Semana 13. Ecuaciones diferenciales parciales

- Reto: Ecuación de Laplace y Jacobi, Difusión.

Semana 14. Marchas aleatorias. Movimiento Browniano.

- Reto: Puentes Brownianos y la Información de Fisher. Test de números aleatorios.

Semana 15. Markov Chain Monte Carlo. Ajuste de parámetros.

- Reto: Fluido de Lennard–Jones

V Metodología

Las sesiones serán, sobre todo, una sesión de exploración, práctica y experimentación. Para que esto funcione es necesario que los estudiantes lleguen a clase después de haber leído sobre el tema correspondiente. El programa del curso tiene dos partes bien diferenciadas. La parte de métodos tradicionales de cómputo numérico y la parte de carpintería de software.

Se usarán principalmente los notebooks de IPython complementado con C++. También se podrá usar Python o C si bien lo desean. Se realizarán talleres a lo largo del semestre que evalúen los temas de las clases.

El repositorio de GitHub del curso estará en el siguiente enlace: `https://github.com/jpmallarino/FISI2028-202120`.

VI Criterios de evaluación

La Magistral y el Laboratorio se califican por separado. Las componentes que reciben calificación en la Magistral (en paréntesis su contribución a la nota definitiva) son las siguientes:

- Ejercicios (10 % cada uno). 6 Ejercicios individuales en total para entregar cada dos semanas.
- Trabajo colaborativo (10 %). En grupos de dos personas se encargarán del mantenimiento del repositorio de las librerías de la clase.
- Examen Final (30 %). Será completamente escritoTendrá una componente escrita y otra de programación.

Todos los exámenes y ejercicios son **individuales**. Si en las entregas se detecta que el trabajo no fue individual (esto incluye colaboración con personas no inscritas en el curso, i.e. a través de "monitorías") se llevará el caso a comité disciplinario y la nota del curso queda como Pendiente Disciplinario hasta que el comité tome alguna decisión.

Todas las entregas de talleres y ejercicios se harán a través de GitHub.

VII Bibliografía

Bibliografía principal:

- R. H. Landau, M. J. Páez, C. C. Bordeianu. *A survey of Computational Physics - Enlarged Python Book*, WILEY, 2012.
- Videos del curso Herramientas Computacionales que muestran los fundamentos de Unix y Python `https://www.youtube.com/playlist?list=PLHQtzvthdVM_MGC9dPFKe4hPAwBd_7RJ3`
- Software Carpentry: `http://software-carpentry.org/`
- C++ Tutorial: `https://www.cplusplus.com/doc/tutorial/`
- Bjarne Stroustrup: `https://www.stroustrup.com/`

Bibliografía secundaria:

- W. Krauth. *Statistical Mechanics: Algorithms and Computations*, Vol. 13. OUP Oxford, 2006.
- Guttag, J. V. *Introduction to Computation and Programming Using Python*, The MIT Press, 2013.
- B. Kernighan & D. Ritchie. *The C programming language*. Second Edition, Prentice Hall.