

MÉTODOS COMPUTACIONALES 2

NOMBRE DEL CURSO: Métodos Computacionales 2

CÓDIGO DEL CURSO: FISI-2528

UNIDAD ACADÉMICA: Departamento de Física

CRÉDITOS: 2

PERIODO ACADÉMICO: 202310

HORARIO: Mi y Vi, 12:00 a 13:50

NOMBRE PROFESOR MAGISTRAL: Diego Hernando Useche Reyes

CORREO ELECTRÓNICO: dh.useche@uniandes.edu.co

HORARIO Y LUGAR DE ATENCIÓN: con cita previa

REPOSITORIO DEL CURSO: <https://github.com/diegour1/MetodosComputacionales2>

I Introducción

Los métodos computacionales, en especial, la solución numérica de ecuaciones diferenciales 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 su aumento exponencial en capacidad de procesamiento. Estos recursos para el cómputo sólo se puede aprovechar si las personas interesadas son capaces de utilizar la tecnología de manera eficiente y reconocer los fundamentos matemáticos que los sustenta.

II Objetivos

En el curso se presentan algoritmos y técnicas computacionales básicas para:

- Derivar analíticamente las expresiones usadas en los métodos numéricos.
- Resolver numéricamente problemas que involucren ecuaciones diferenciales ordinarias.
- Resolver numéricamente problemas que involucren ecuaciones diferenciales parciales.
- Resolver problemas numéricos usando los lenguajes de programación **python** y **C++**.
- Diseñar simulaciones computacionales de fenómenos de las ciencias.

III Competencias a desarrollar

Al finalizar el curso, se espera que el estudiante adquiera las siguientes habilidades.

- Describir las expresiones matemáticas que aparecen en los métodos numéricos.
- Implementar computacionalmente modelos de ciencias y otras áreas para solucionar problemas básicos de ecuaciones diferenciales.
- Implementar animaciones que permitan observar el comportamiento de las soluciones.

IV Contenido por semanas

Semana 1

- Temas: Presentación del curso. Uso de **Python** y **C++** en la nube. Introducción a **C++**. Programación orientada a funciones y a objetos en **C++**. Tipos de datos.

Semana 2

- Temas: Apuntadores y Makefiles en **C++**. Simulaciones en **Python**.

Semana 3

- Temas: Series y transformada de Fourier. Señales y filtros (sonido e imágenes). Transformada rápida de Fourier. Aplicaciones.
- Lecturas preparatorias: Capítulo 10 (Fourier Analysis) del libro de Landau.

Semana 4

- Temas: Ecuaciones diferenciales ordinarias de primer orden y sistemas de ecuaciones. Método de Euler. Métodos de un paso Runge-Kutta.
- Lecturas preparatorias: Capítulo 9 (ODEs) del libro de Landau. Capítulo 2 del libro de Hutchinson Landau.

Semana 5

- Temas: Ecuaciones diferenciales ordinarias de segundo orden y sistemas de ecuaciones. Métodos de un paso Runge-Kutta y simpléticos de segundo y cuarto orden. Aplicaciones: Sistema de tres cuerpos, péndulo doble.
- Lecturas preparatorias: Capítulo 9 (ODEs) del libro de Landau. Capítulo 2 del libro de Hutchinson.

Semana 6

- Temas: Ecuaciones diferenciales parciales de primer y segundo orden. Introducción a las diferencias finitas, repaso de los métodos de relajación. Ecuaciones elípticas (Ecuación de Advección, Laplace y Driven cavity flow). Generalización a otras coordenadas convergencia y estabilidad.
- Lecturas preparatorias: Capítulo 17 (PDEs) del libro de Landau. Capítulo 4 del libro de Hutchinson

Semana 7

- Temas: Ecuaciones diferenciales parciales de segundo orden. Ecuaciones parabólicas (Ecuación de difusión en geometría cartesiana y cilíndrica). Ecuaciones hiperbólicas (ecuación de onda). Solución espectral con la transformada rápida de Fourier.
- Lecturas preparatorias: Capítulo 17 (PDEs) del libro de Landau. Capítulo 5 de Hutchinson.

Semana 8

- Temas: Temas Adicionales. Análisis de Algoritmos. Notación Big O. Introducción conceptual a los métodos finitos (oscilaciones en una barra y transferencia de calor).

V Metodología

Los 2 créditos del curso corresponden a 9 horas de dedicación semanal. Cada semana habrá 3,5 horas sincrónicas distribuidas en $2 \times 1,75 = 3,5$ horas de magistral. Las 5,5 horas restantes corresponden a trabajo individual para la preparación de los contenidos de la semana y entrega de trabajo.

- Durante las clases habrá una sesión de 60 – 80 minutos para presentar la parte teórica del tema del día y mostrar ejemplos de implementación, y 30 – 50 para resolver una actividad en clase y resolver las dudas que se tengan sobre el tema del día. En algunas sesiones que lo requieran habrá solamente explicación sin actividad para entregar durante la clase.

Para el desarrollo de la clase se usará el ambiente Binder accesible desde el repositorio del curso, y el sistema de Jupyter en la nube Google Colab.

VI Criterios de evaluación

La evaluación del curso tiene tres componentes: 10 % problemas en horario de la clase, 60 % talleres, 30 % examen final.

- Problemas en clase: Se desarrollaran en algunas sesiones unos problemas para resolver en promedio durante los últimos 30 a 50 minutos de terminar la clase. Estos ejercicios son para entregar de manera individual pero puede haber colaboración entre estudiantes. Se calificará solamente la realización de los ejercicios. Del total del 100 % de ejercicios, se tendrá en cuenta para la nota que el estudiante haya entregado al menos el 60 % de los ejercicios, cada ejercicio tiene la misma ponderación.
- Talleres: Cada semana y media se asignará un taller con 1, 2 o 3 ejercicios para entregar en parejas.
- Examen Final: El último día de clase se realizará un examen final del curso, en principio el examen es acumulativo.

Cada ejercicio de los talleres y el examen final recibe el siguiente porcentaje de calificación:

- Entrega de código fuente que incluye las funciones, algoritmos y procedimientos que se solicitan en el enunciado: 30 %.
- El código corre sin errores dentro del intérprete de Python o C++ de Binder y/o Google Colab y tiene al menos una parte del ejercicio correcta: 30 %.
- El ejercicio es casi correcto, tiene un error mínimo. 20 %.
- El código produce la respuesta correcta: 20 %.

La nota definitiva se reporta con dos cifras decimales. Todas las entregas de talleres y ejercicios se harán a través de la plataforma Bloque-neón. **No se aceptará ninguna tarea por fuera de esa plataforma**, a menos que ocurra una falla en los servidores que afecte a **todos** los estudiantes del curso. Finalmente, no se toman en cuenta entregas realizadas por correo electrónico fuera de los tiempos estipulados. **La nota semanal se puede modificar únicamente dentro de los próximos 8 días calendario.**

Las únicas fuentes autorizadas para reutilización de código son: el repositorio del curso y los libros de la bibliografía principal. En casos de copia, se llevará el caso a comité disciplinario y la nota del curso queda como Pendiente Disciplinario hasta que el comité tome alguna decisión.

VII Bibliografía

- *A survey of Computational Physics - Enlarged Python Book* . R. H. Landau, M. J. Páez, C. C. Bordeianu. WILEY. 2012. <https://psrc.aapt.org/items/detail.cfm?ID=11578>
- *A student's guide to numerical methods*, I. H. Hutchinson. Cambridge University Press. 2015.
- *Elements of Scientific Computing* Tveito A., Langtangen H.P., Nielsen B.F., Cai X. Springer. 2010.
- C++ Tutorial, <https://www.tutorialspoint.com/cplusplus/>