

MÉTODOS COMPUTACIONALES

Jaime E. Forero Romero

NOMBRE DEL CURSO: Métodos Computacionales
CÓDIGO DEL CURSO: FISI 2028 (Magistral) FISI 2809 (Laboratorio)
UNIDAD ACADÉMICA: Departamento de Física
PERIODO ACADÉMICO: 201620
HORARIO (MAGISTRAL): Mi 12:30 a 14:00 y Vi, 15:30 a 17:00
HORARIO (LABORATORIO): Ju

NOMBRE PROFESOR(A) PRINCIPAL: Jaime E. Forero Romero
CORREO ELECTRÓNICO: je.forero@uniandes.edu.co
HORARIO Y LUGAR DE ATENCIÓN: Mi 14:00 a 16:00, Oficina Ip208

NOMBRE PROFESOR(A) COMPLEMENTARIO(A):
CORREO ELECTRÓNICO: @uniandes.edu.co

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,
- 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 (Python) y otro de bajo nivel (C),
- 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: Unix. Consola. Editores de texto. Comandos básicos. Intérprete de Python. Variables. Aritmética. Listas. Diccionarios.
- Lecturas preparatorias:
- Talleres: Examen diagnóstico (sin nota).

Semana 2

- Temas: If/while/break/continue. Ciclos. Funciones. Arreglos (numpy). Lectura y escritura de archivos (numpy). Gráficas y visualización (matplotlib).
- Lecturas preparatorias:
- Talleres:

Semana X

- Temas:
- Lecturas preparatorias:
- Talleres:

Semana X

- Temas:
- Lecturas preparatorias:
- Talleres:

Semana X

- Temas:
- Lecturas preparatorias:
- Talleres:

Semana X

- Temas:
- Lecturas preparatorias:
- Talleres:

Semana X

- Temas:
- Lecturas preparatorias:
- Talleres:

Semana X

- Temas:
- Lecturas preparatorias:
- Talleres:

Semana X

- Temas:
- Lecturas preparatorias:
- Talleres:

Semana X

- Temas:
- Lecturas preparatorias:
- Talleres:

Semana X

- Temas:
- Lecturas preparatorias:
- Talleres:

Semana X

- Temas:
- Lecturas preparatorias:
- Talleres:

Semana 3. Integración de ecuaciones diferenciales ordinarias. Método leapfrog, integradores simplécticos. Ecuaciones diferenciales parciales. Diferentes esquemas de solución. Mecánica de fluidos. Leyes de conservación hiperbólicas. Riemann Solvers. Método de Godunov.

Semana 4. Generadores de números aleatorios. Integración. Cadenas de Markov.

Semana 5. Ecuaciones diferenciales estocásticas. Convergencia. Estabilidad.

Semana 6. Termodinámica de sistemas simples. Estimación de parámetros. **Entrega Taller 1.**

Semana 7. Aprendizaje Supervisado. Árboles de Decisión, Clasificación, Ranking Regresión, Redes Neuronales.

Semana 8. Clustering: k-means, Jerárquico y Maximización del valor esperado.

Semana 9. Reducción de Dimensionalidad y Predicción Estructurada: Principal Component Analysis (PCA), Multi-dimensional Scaling (MDS), Linear Discriminant Analysis (LDA); Modelos Gráficos, Redes Complejas y Análisis de datos Topológico.

Semana 10. Receso.

Semana 11. Procesamiento Natural de Lenguaje (Naïve Bayes) y Aprendizaje Profundo (Deep Learning).

Semana 12. Fundamentos de programación en paralelo. Taxonomía de arquitecturas para cómputo en paralelo. Máquinas para cómputo en paralelo en Uniandes. **Entrega Taller 2.**

Semana 13. Fundamentos de programación en paralelo: OpenMP

Semana 14. Fundamentos de programación en paralelo: MPI

Semana 15. Fundamentos de programación en paralelo: MPI

Semana 16. Cómo desplegar máquinas de cómputo en La Nube. Ejemplos de aplicación en Amazon Web Services y Docker. **Entrega Taller 3 al final de la primera semana de finales.**

V Metodología

Cada clase tendrá una corta presentación teórica (30 minutos aproximadamente) para pasar a practicar todos los conceptos directamente en la computadora/cluster a través de ejercicios de práctica (50 minutos aproximadamente).

VI Criterios de evaluación

El curso tendrá tres entregas de trabajos, cada una con un valor del $(100/3)\%$ de la nota definitiva. Los temas de las entregas serán los siguientes:

1. Ecuaciones diferenciales (ordinarias, parciales, estocásticas).
2. Machine Learning.
3. Cómputo en paralelo (OpenMP, MPI).

VII Bibliografía

Bibliografía principal:

- R. L. Burden, J. D. Faires. *Numerical analysis*, 2011. (Biblioteca General - 519.4 B862 2011)
- A. Tveito, H.P. Langtangem, B.F. Nielsen., *Elements of Scientific Computing*, 2010. (Biblioteca General, Recurso Electrónico 510.)

- O. Maimon and L. Rokach, *The Data Mining and Knowledge Discovery Handbook*, 2010. (Biblioteca General, Recurso Electrónico 006.312)
- M. Snir, *MPI : the complete reference*, 1996. (Biblioteca General, 004.35 M637)
- J. Sanders, E. Kandrot. *CUDA by example: an introduction to general-purpose GPU programming*, 2010. (Biblioteca General - 005.275 S152)

Bibliografía complementaria:

- D. Conway and J. M. White. *Machine learning for hackers*, 2012.
- S.Bird. *Natural Language Processing with Python*, 2009.
- Theano Development. *Deep Learning Tutorial* <http://deeplearning.net/tutorial/>