

MÉTODOS COMPUTACIONALES AVANZADOS

Jaime E. Forero Romero

NOMBRE DEL CURSO: Métodos Computacionales Avanzados

CÓDIGO DEL CURSO: FISI 3028 / FISI 4028

UNIDAD ACADÉMICA: Departamento de Física

PERIODO ACADÉMICO: 201710

HORARIO: Lu 18:00 a 19:50 y Vi 18:00 a 18:50

NOMBRE PROFESOR(A) PRINCIPAL: Jaime E. Forero Romero

CORREO ELECTRÓNICO: je.forero@uniandes.edu.co

NOMBRE PROFESOR(A) COMPLEMENTARIO(A):

CORREO ELECTRÓNICO: @uniandes.edu.co

I Introducción

Los métodos computacionales fundamental el trabajo en todas las áreas técnicas y científicas, ya sean principalmente experimentales o teóricas. Esto se debe en gran parte a que la capacidad de utilizar computadoras de alto rendimiento ha disminuido en costo monetario y en complejidad.

El curso de Métodos Computacionales Avanzados presenta estas posibilidades computacionales a estudiantes de diferentes disciplinas científicas. Para esto se propone profundizar sus conocimientos en tres áreas: resolución de ecuaciones diferenciales, implementación de métodos de machine learning (i.e. algoritmos que aprenden de datos) y utilización de técnicas de cómputo masivamente paralelo.

Se asume que los estudiantes de este curso ya tienen conocimientos básicos en métodos computacionales equivalentes al nivel del curso Métodos Computacionales (FISI-2028).

II Objetivos

Los objetivos principales del curso son:

- Presentar métodos para la resolución de ecuaciones diferenciales ordinarias, parciales y estocásticas.
- Demostrar el uso de métodos y librerías de Machine Learning.
- Mostrar diferentes arquitecturas, paradigmas, lenguajes y librerías para cómputo masivamente paralelo.
- Estudiar diferentes aplicaciones prácticas a problemas científicos y del contexto industrial de las metodologías computacionales modernas.

III Competencias a desarrollar

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

- Manejar lenguajes modernos de computación numérica de bajo nivel (i.e. C/C++) y de alto nivel (i.e. Python/R/Julia).
- Tener un esquema para pre-procesar, analizar y generar reportes científicos y técnicos a partir de diversas fuentes de datos utilizando métodos computacionales.
- Desplegar programas en paralelo en un cluster, al igual que en sistemas de cómputo distribuidos tales como Amazon Web Services.
- Generar conocimiento a partir del modelamiento teórico y computacional de los conceptos vistos en clase.

IV Contenido por semanas

Semana 1. Unix. Repositorios. Makefiles. C. Python. Norma IEEE para aritmética de punto flotante.

Referencia: Capítulos 1, 2, 3, 6 y 9 del libro de ISTC.

Semana 2. Fundamentos de programación en paralelo. Taxonomía de arquitecturas para cómputo en paralelo. Máquinas para cómputo en paralelo en Uniandes. Fundamentos de programación en paralelo: MPI.

Referencia: Capítulos 11 y 12 del libro de ISTC.

Semana 3. Fundamentos de programación en paralelo: MPI.

Referencia: Capítulo 12 del libro de ISTC.

Semana 4. Fundamentos de programación en paralelo: OPENMP.

Referencia: Capítulo 13 del libro de ISTC.

Semana 5. Ecuaciones diferenciales ordinarias. Integradores simplécticos. Leapfrog.

Referencia:

Semana 6. Ecuaciones diferenciales parciales. Diferentes esquemas de solución. Leyes de conservación hiperbólicas. Esquemas de diferencias finitas, elementos finitos y volúmenes finitos.

Referencia:

Semana 7. Generadores de números aleatorios. Ecuaciones diferenciales estocásticas. Convergencia. Estabilidad.

Referencia:

Semana 8. Estimación de parámetros con estadística bayesiana. Algoritmo de Metrópolis-Hastings.

Referencia: Capítulo 29 de ITILA.

Semana 9. Algoritmos Monte Carlo eficientes: Hamiltoniano, Gibbs, annealing.

Referencia: Capítulo 30 de ITILA.

Semana 10. Introducción a Machine Learning.

Referencia: Capítulos 1 y 2 de ISLR.

Semana 11. Regresión lineal y regularización.

Referencia: Capítulo 3 de ISLR.

Semana 12. Árboles de Decisión.

Referencia: Capítulo 8 ISLR.

Semana 13. Support Vector Machines.

Referencia: Capítulo 9 ISLR.

Semana 14. PCA, Clustering, K-means. t-SNE.

Referencia: Capítulo 10 ISLR.

Semana 15. Redes Neuronales.

Referencia:

V Metodología

Cada semana tendremos una corta presentación teórica (45 minutos aproximadamente) para pasar a ejercitar esos conceptos directamente en la computadora/cluster haciendo ejercicios de práctica (90 minutos aproximadamente).

VI Criterios de evaluación

El curso tendrá tres entregas de trabajos, cada una con un valor del 25 % de la nota definitiva. Los temas de las entregas serán los siguientes:

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

El 25 % restante de la nota definitiva se calculará a partir del promedio de 3 entregas de ejercicios cortos hechos en clase.

VII Bibliografía

Bibliografía principal:

- D. J. MacKay., *Information Theory, Inference and Learning Algorithms*, 2003, Cambridge.
<http://www.inference.phy.cam.ac.uk/mackay/itila/>.
- G. James, D. Witten, T. Hastie, R. Tibshirani., *An Introduction to Statistical Learning with Applications in R*, 2015, Springer.
<http://www-bcf.usc.edu/~gareth/ISL/>
- J. VanderPlas., *Python Data Science Handbook*, 2016, O'Reilly.
<https://github.com/jakevdp/PythonDataScienceHandbook>
- A. Tveito, H.P. Langtangem, B.F. Nielsen., *Introduction to Scientific and Technical Computing*, 2017. CRC Press. (Biblioteca General - 502.85 I576)

Bibliografía complementaria:

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