

FISI 2028 Métodos Computacionales Semestre 2014 - 20

Martes y Miércoles 15:30 - 16:50

Salon LL-204

Profesor: Jaime Forero, email: je.forero, Oficina Ip208

Monitores:

(Magistral & Laboratorio) Nicolás Garavito, email: jn.garavito57

(Laboratorio) Luis Alberto Gutiérrez, email: la.gutierrez1280

(Laboratorio) Christian Poveda, email: cn.poveda542

(Magistral) María Camila Remolina, email: mc.remolina197

## Objetivo

El curso tiene como objetivo principal *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.*

En esta clase daré énfasis a esa *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.

## Metodología

Esa *actitud computacional* se desarrolla trabajando. Las sesiones de física computacional 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. La primera es probable que le sea útil a una fracción de los asistentes al curso en su vida profesional. La segunda parte le será útil a *todos*.

## Software

Se usarán principalmente los notebooks de IPython complementado con C. También se aceptan tareas en los siguientes lenguajes de programación: FORTRAN 90/95, C++ y Python. No se aceptarán tareas en Matlab, Mathematica o cualquier otro lenguaje de programación que no esté en la lista mencionada antes.

## Evaluación

Hay 8 talleres para entregar. Los primeros 4 valen el 10 % cada uno, los siguientes 4 valen el 15 % cada uno. No habrá parciales ni examen final. Los primeros cuatro talleres serán individuales. Los últimos cuatro talleres serán en parejas. Si en las entregas individuales es claro que hubo trabajo en grupos entonces la nota de todos los talleres individuales quedará automáticamente en cero (0.0).

Las entregas para los últimos 4 talleres se harán en dos tiempos: una primera entrega donde se muestre explícitamente un borrador del código con comentarios, luego la entrega definitiva con el código completo. La primera entrega es una condición necesaria para aceptar la segunda. Solamente la segunda entrega recibe una nota.

Esta materia se ve al mismo tiempo que el Laboratorio de Métodos Computacionales. El objetivo del Laboratorio es tener más tiempo para practicar todo lo visto en clase. De acuerdo a la nota definitiva en Laboratorio habrá un bono en la nota definitiva de este curso. Siendo  $x$  la nota

de Laboratorio, el bono correspondiente se calcula así:  $4,0 < x \leq 4,4 \rightarrow 0,2$ ,  $4,4 < x \leq 4,8 \rightarrow 0,3$ ,  $4,8 < x \leq 5,0 \rightarrow 0,5$ .

El curso cuenta con un repositorio en github: <https://github.com/forero/ComputationalMethods>. El material se encuentra distribuido en las siguientes carpetas.

- `hands_on/`: ejemplos para hacer en clase.
- `homework/`: enunciados y calificaciones de las tareas.
- `notas/`: notas de clase.
- `syllabus/`: programa del curso.

## Programa

Semana	Teoría	Carpintería	Taller
1 (28.7)	Algoritmos	Consola/Emacs/Gnuplot	#1 (shell scripts)
2 (4.8)		Consola/Emacs - C	
3 (11.8)		C	#2 (C)
4 (18.8)		Python	
5 (25.8)		IPython Notebook (gráficas) / Github Individual	#3 (Python)
6 (1.9)	Solución de sistemas lineales de ecuaciones		#4 (Matrices, Git, C)
7 (8.9)	Interpolación	Diseño de programas / Github Colaborando	
8 (15.9)	Análisis de Fourier - FFT	Ipython / Numpy	#5 (Fourier, Git, Python)
9 (22.9)	<b>Semana de trabajo individual</b>		
10 (29.9)	Integración y derivación numérica	Ipython / Numpy	
11 (6.10)	Ecuaciones diferenciales ordinarias (1er orden)		#6 (EDO1, Derivadas, C, Python, Git)
12 (13.10)	Ecuaciones diferenciales ordinarias (2do orden)	Makefile	
13 (20.10)	Ecuaciones diferenciales parciales		#7 (EDO2, EDP, C, Python, Git)
14 (27.10)	Métodos Monte Carlo		
15 (3.11)	MCMC para hacer fits		#8 (MCMC, Python)
16 (10.11)			

## Bibliografía

- *A survey of Computational Physics*. R. H. Landau, M. J. Páez, C. C. Bordeianu. Princeton Univ. Press. 2006
- *Statistical Mechanics: Algorithms and Computations*. W. Krauth, Oxford Univ. Press.
- *Introduction to Computational and Programming Using Python*, Gutttag, J. V. The MIT Press. 2013.
- *The C programming language*. B. Kernighan & D. Ritchie, Second Edition, Prentice Hall.
- <http://software-carpentry.org/>
- <http://xkcd.com/>