

MÉTODOS COMPUTACIONALES 1

NOMBRE DEL CURSO: Métodos Computacionales 1

CÓDIGO DEL CURSO: FISI-2526

UNIDAD ACADÉMICA: Departamento de Física

PRERREQUISITOS: Introducción a la Programación (ISIS-1221). Álgebra Lineal I (MATE-1105).

CRÉDITOS: 3

PERIODO ACADÉMICO: 202220

HORARIO: Miércoles y Viernes, 11:00 am a 12:20 pm. Salón: ML-513

NOMBRE PROFESOR MAGISTRAL: Manuel Alejandro Segura Delgado

CORREO ELECTRÓNICO: ma.segura10@uniandes.edu.co

HORARIO Y LUGAR DE ATENCIÓN: con cita previa

NOMBRE PROFESOR COMPLEMENTARIA: Diego Hernando Useche Reyes

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

HORARIO DE ATENCIÓN: con cita previa.

REPOSITORIO DEL CURSO: <https://github.com/asegura4488/MetodosComputacionales20222>

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 su aumento exponencial en capacidad de procesamiento. Estos recursos para el cómputo sólo se puede aprovechar si las personas interesadas son capaces de utilizarlos de manera eficiente. De manera complementaria, la obtención y comprensión de los resultados obtenidos con estos métodos computacionales requieren una comprensión básica de probabilidad y estadística.

II Objetivos

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

- resolver numéricamente problemas que involucren derivadas, integrales y sistemas de ecuaciones algebraicas.
- analizar datos y modelos con conceptos probabilísticos y métodos estadísticos.

III Competencias a desarrollar

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

- Usar herramientas de **Python** para análisis estadístico de datos, para la solución de problemas en ciencias y otras áreas.
 - Implementar computacionalmente modelos físico-matemáticos para el análisis de datos utilizando **clases** y **funciones**.
 - Utilizar de la biblioteca **pandas** para el manejo eficiente de los datos teniendo en cuenta los siguientes procedimientos: preprocesamiento, re-formato de columnas y normalización.
 - Utilizar la biblioteca **numpy** en la implementación de los algoritmos matemáticos que representen los diferentes sistemas en estudio.
 - Utilizar la biblioteca **matplotlib** para visualizar datos.

- Solucionar numéricamente problemas sencillos de cálculo diferencial, cálculo integral y álgebra lineal: cálculo de integrales, cálculo de derivadas, sistemas de ecuaciones lineales, integración de funciones y problemas de valores propios.
 - Calcula numéricamente la primera y segunda derivada de funciones de una y más variables.
 - Calcula numéricamente integrales definidas con los siguientes métodos: trapecio, Simpson y cuadratura gaussiana.
 - Encuentra numéricamente la solución a sistemas lineales y no-lineales.
 - Utiliza bibliotecas de álgebra lineal para calcular valores y vectores propios.
 - Interpreta los vectores y valores propios de una matriz de covarianza como herramientas para la reducción de dimensionalidad de un conjunto de datos.
- Analizar datos usando apropiadamente conceptos básicos de probabilidad y estadística en la solución de problemas en ciencias y otras áreas: variables aleatorias, valores esperados, métodos de MonteCarlo y pruebas de hipótesis:
 - Diferenciar distribuciones de probabilidad comúnmente encontradas en ciencias: normal, poisson, log-normal, etc.
 - Hacer histogramas para identificar la distribución de los datos como función de una variable específica.
 - Estimar la moda, media, mediana, desviación estándar, curtosis y otros momentos de series de datos.
 - Realizar pruebas cuantitativas para ver si dos series de datos corresponden a la misma distribución de probabilidad.
 - Utilizar datos para proponer modelos paramétricos y acotar los sus términos libres.
 - Resolver problemas de mínimos cuadrados matriciales.
 - Usar métodos de remuestreo para encontrar la incertidumbre asociada al problema de mínimos cuadrados y para la estimación de parámetros.
 - Realizar reducción de dimensionalidad de datos a través del método de componentes principales.

IV Contenido por semanas

Semana 1

- Temas: Presentación del curso. Lenguaje de programación **Python**. Comandos y programación básica de Unix. Repaso de **Python**: variables, listas, diccionarios, bucles y condicionales, lectura y escritura de archivos.

Semana 2

- Temas: Repaso de **Python**: funciones, programación orientada a objetos, numpy y matplotlib.

Semana 3

- Temas: Derivada finita y cálculo del gradiente. Raíces de ecuaciones: método de Newton-Raphson y Bisección.

Semana 4

- Temas: Interpolación de Lagrange y de newton. Concepto de base cardinal y aplicaciones.

Semana 5

- Temas: Métodos de integración de tipo Newton-Côtes y cuadratura gaussiana. Polinomios ortogonales y su relación con la integración. Método de MonteCarlo. **Parcial**

Semana 6

- Temas: Sistemas de ecuaciones lineales: eliminación gaussiana, factorización LU y relajación sucesiva. Método de descenso multivariado (Machine Learning application).

Semana 7

- Temas: Determinantes $n \times n$, introducción a autovalores y autovectores y sistemas no lineales (Método de Newton).

Semana 8

- Temas: Muestreo aleatorio simple, estratificado y sistemático de una población. Definición de variable aleatoria, interpretación frecuentista de probabilidad y función de densidad de variable discreta, uso de histogramas.

Semana 9

- Temas: Estadística descriptiva: definición de los momentos de una distribución (media y varianza), distribución acumulada, probabilidad condicional y total. Teorema de Bayes. Aplicaciones.

Semana 10

- Temas: Distribuciones de probabilidad en variables variables. Distribuciones marginales (cross-tabs), independencia y cálculo de covarianza. Análisis de componentes principal (**PCA**). **Parcial**

Semana 11

- Temas: Distribuciones estadística del muestreo. Método de bootstrapping, teorema central del límite y distribución χ^2 .

Semana 12

- Temas: Regresión lineal, mínimos cuadrados y mínimos cuadrados matriciales.

Semana 13

- Temas: Estimación de parámetros. Método de re-muestreo (bootstrapping), función de máxima verosimilitud y efectos sistemáticos. Estimación puntual y por intervalos de confianza.

Semana 14

- Temas: Estimación de parámetros. Cadenas de Markov, método de Metropolis-Hastings, estimación puntual y por intervalos de confianza.

Semana 15

- Temas: Prueba de hipótesis. Definición de una hipótesis y del proceso de inferencia, error tipo-I y error tipo-II. Significancia estadística de una observación, definición del p-valor (p-value) y su interpretación.

Semana 16

- Temas: Pruebas de hipótesis para la diferencias de medias, proporción, diferencia de proporciones y estimación de varianza poblacional. Cálculo de valores críticos para modelos de conteo de partículas en altas energías (Poisson process). Lema de Neyman-Pearson. **Examen final**

V Metodología

Los 3 créditos del curso corresponden a 4 horas de dedicación semanal. Cada semana habrá 3 horas magistrales y 1 hora dedicadas a la complementaria. **La clase magistral tiene un porcentaje de 75 % y la complementaria de 25 %.**

- En la clase magistral, se dedicará la primera mitad a la explicación de los conceptos, desarrollos matemáticos y algorítmicos y en la siguiente mitad, los estudiantes realizarán un serie de problemas asociados al tema de estudio.
- La entrega de talleres y parciales se hará a través del repositorio de cada grupo de trabajo.
- Se conformarán equipos de trabajo para la entrega de las actividades de máximo 2 personas.

Para el desarrollo de la clase se usará el ambiente Binder o un entorno local accesible desde el repositorio del curso.

VI Criterios de evaluación

Cada taller o parcial recibe el siguiente porcentaje de calificación:

- El código corre sin errores dentro del intérprete de Python: 20 %.

Actividad	Porcentaje	Cantidad
Talleres	60	1 por semana
Parciales	40	1 por mes

Cuadro 1: Esquema de actividades

- Entrega de código fuente que incluye las funciones, algoritmos y procedimientos que se solicitan en el enunciado: 30 %.
- El código produce la respuesta correcta: 50 %.

La nota de cada actividad se reporta en la plataforma bloque-neón y la nota definitiva se reporta con dos cifras decimales. Los ejercicios deben ser entregados en las fechas límite establecidas en bloque-neón. Finalmente, los parciales deben ser entregados durante la clase.

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

Bibliografía principal:

- *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>
- *An introduction to statistics with Python, with applications in the life sciences*. Haslwanter Thomas, Springer. 2015.
- *Introduction to Probability and Statistics for Engineers and Scientists*. Sheldon M. Ross. Third Edition. Elsevier Academic Press. 2004.

Bibliografía secundaria:

- *Introduction to Computation and Programming Using Python*, Gutttag, J. V. The MIT Press. 2013.
- *A student's guide to numerical methods*, I. H. Hutchinson. Cambridge University Press. 2015.
- *Introduction to statistics with python*. Thomas Haslwanter. Springer. 2016.
- *Pattern recognition and Machine Learning*. Christopher Bishop. Springer. 2011.
- *Data Analysis: A Bayesian Tutorial*. D. S. Sivia, J. Skilling. Second Edition, Oxford Science Publications. 2006.
- *Statistical Mechanics: Algorithms and Computations*. W. Krauth, Oxford Univ. Press.
- *An Introduction to Statistical Learning*. G. James, D. Witten, T. Hastie, R. Tibshirani, Springer. <http://www-bcf.usc.edu/~gareth/ISL/>
- *Elements of Scientific Computing* Tveito A., Langtangen H.P., Nielsen B.F., Cai X. Springer. 2010.