

Este programa esta en proceso de aprobación por el Consejo de Facultad (última actualización Mon, 11 Jul 2016 12:31:30 -050). Úselo solamente como fuente de información preliminar. Una versión previa del curso puede encontrarse en el enlace:

<http://astronomia-udea.co/principal/Curriculo/planes.php>

Allí se publicará también la versión definitiva de este semestre una vez este aprobado.

Este curso esta en edición y no es una versión distribuible. Esta disponible para edición en:

<http://astronomia-udea.co/principal/Curriculo/links/678336.html>.

FORMATO DE MICROCURRICULO O PLAN DE ASIGNATURA

1. IDENTIFICACIÓN GENERAL			
Facultad	Facultad de Ciencias Exactas y Naturales		
Instituto	Instituto de Física		
Programa(s) Académicos	Astronomía, Física		
Área Académica	Computación		
Ciclo	Fundamentación		
Tipo de Curso	Básico		
Profesores Responsables	Manuel Paez, Nataly Mateus, Mario Sucerquia		
Asistencia	Obligatoria		
2. IDENTIFICACIÓN ESPECÍFICA			
Semestre	2014-2		
Nombre de la Asignatura	Métodos Computacionales		
Código	0302390		
Semestre en el plan	3		
Número de Créditos	3		
Horas Semestrales	HDD:64	HDA:32	TI:48
Semanas	16		
Intensidad Semanal	Teórico: 0	Práctico: 0	Teórico-Práctico: 6
H (Habilitable)	Si		
V (Validable)	Si		
C (Clasificable)	No		
Prerrequisitos	Algebra Lineal (0303208), Fundamentación en Computación (0302150)		
Correquisitos	Ninguno		
Sede en la que se dicta	Ciudad Universitaria Medellín		
3. DATOS DE LOS PROFESORES QUE ELABORAN EL PLAN DE ASIGNATURA			
Nombres y Apellidos	Sebastián Bustamante, Jorge Zuluaga, Juan Carlos Muñoz, Manuel Páez		

4. DESCRIPCIÓN

El curso de Métodos Computacionales tiene propósito fundamental la introducción de métodos numéricos ampliamente usados en Física y Astronomía. Estos temas son abordados desde un contexto formal pero manteniendo una fuerte componente práctica y computacional, ilustrando además algunas aplicaciones en problemas de interés actual.

5. JUSTIFICACIÓN

A medida que entendemos mejor nuestro Universo, los fenómenos involucrados exhiben una creciente complejidad y las soluciones numéricas son cada vez más útiles, necesarias y comunes en física y astronomía. Es por lo tanto de suma importancia entender los métodos numéricos, sus capacidades y límites para la solución de problemas, además de una correcta implementación computacional.

6. OBJETIVOS**Objetivo General:**

Promover la creatividad y la productividad de los estudiantes con el entrenamiento teórico-práctico en métodos computacionales aplicables en la solución, modelación y simulaciones de problemas en Física y Astronomía.

Objetivos Específicos:

Al terminar el semestre el estudiante podrá:

Objetivos Conceptuales:

Solucionar con métodos numéricos problemas en Física y Astronomía que puedan no tener soluciones analíticas.

Utilizar el lenguaje Python en la solución de problemas que exigen pesados cálculos numéricos que hacen imposible su solución de otra manera.

Conocer y desarrollar la derivación formal de los métodos numéricos.

Conocer la convergencia y análisis de error de los métodos numéricos, entendiendo el límite y la fiabilidad de estos en la solución de problemas.

Saber que tipo de métodos numéricos se pueden aplicar a un problema determinado.

Saber implementar computacionalmente (programar) los métodos numéricos vistos.

Aprender a realizar pruebas de los métodos numéricos a través de comparación con soluciones analíticas conocidas, si es posible.

Crear herramientas computacionales para la solución de nuevos problemas.

Conocer paquetes de rutinas que se pueden instalar en Python.

Objetivos Actitudinales:

Capacitar al estudiante para que combine los conocimientos de ciencia con métodos numéricos y las técnicas computacionales.

Adiestrar al estudiante en la observación, interpretación y conclusiones de resultados de cálculos numéricos, y de las gráficas obtenidas.

Objetivos Procedimentales:

Conocer las capacidades del lenguaje Python y de programas que se pueden instalar en Python para aplicaciones específicas como: Matemática Simbólica (SymPy, Sage), Algebra Lineal (Numpy), Programas de graficación (Matplotlib) etc. Hay alrededor de 40 000 aplicaciones que se pueden instalar en el Python.

Manejar diferentes rutinas numéricas en las que se debe conocer los datos de entrada, lo que la rutina produce y los posibles errores involucrados.

Solucionar problemas semejantes a los expuestos en clase con los mismos métodos numéricos.

7. CONTENIDOS

Contenido Resumido

- 1-Preliminares computacionales
- 2-Ecuaciones de una variable
- 3-Métodos de interpolación
- 4-Cálculo numérico
- 5-Álgebra lineal numérica
- 6-Ecuaciones diferenciales
- 7-Métodos de Monte Carlo

Unidades Detalladas

Unidad 1. Preliminares computacionales (3 semanas)

Contenidos conceptuales:

Repaso breve del Python

Estructura básica de un programa en Python.

Numpy y rutinas de graficación.

Operaciones aritméticas con reales y complejos.

Números aleatorios, método del congruente lineal y problemas.

Errores: redondeo, truncamiento, sustracción cancelativa, precisión de la máquina.

Algoritmos.

Contenidos procedimentales:

Programas ilustrativos de cada uno de los temas de esta unidad.

Contenidos actitudinales:

Observar como el computador produce errores en los cálculos.

El computador no maneja infinitas cifras decimales.

Unidad 2. Ecuaciones de una variable (2 semanas)

Contenidos conceptuales:

Algoritmo de la bisección.

Algoritmo de la regla falsa.

Algoritmo de la secante.

Algoritmo de Newton-Raphson.

Ecuaciones trascendentales.

Análisis de errores.

Contenidos procedimentales:

Se utiliza el gráfico de una función $f(x)$ para estimar el rango de valores de x en el cual la función $f(x)=0$, para usarlo en la solución numérica.

Se explican gráficamente cada uno de los métodos para hallar raíces.

Contenidos actitudinales:

No todos los métodos convergen; si no se estiman apropiadamente las condiciones iniciales, el algoritmo puede diverger.

Cada algoritmo tiene un método para terminar la búsqueda de la raíz. Se debe comprender cada criterio de terminación.

Unidad 3. Métodos de interpolación (2 semanas)

Contenidos conceptuales:

Interpolación lineal.

Polinomio de Lagrange, problemas que conlleva, y soluciones.

Diferencias divididas.

Interpolación de Hermite.

Interpolación con splines cúbicos.

Contenidos procedimentales:

Se ilustran ejemplos de los métodos de interpolación. El polinomio de Lagrange de orden superior exhibe desviaciones en algunos

puntos que no sirven para la interpolación. Si se usan polinomios de Lagrange de segundo orden para interpolación de tres en tres puntos, mejora la interpolación.

Con los splines cúbicos mejora la interpolación.

Contenidos actitudinales:

Comparación de los diferentes métodos para observar la manera como cada uno realiza la interpolación.

Observar las derivadas en los puntos dados para la interpolación. En algunos dichas derivadas no existen.

Unidad 4. Cálculo numérico (2 semanas)

Contenidos conceptuales:

Diferenciación numérica, método de la diferencia hacia adelante y de la diferencia central, segundas derivadas numéricas, errores.

Integración numérica, método trapezoidal y de Simpson. Métodos de la cuadratura gaussiana.

Método de von Neumann para integrales (Monte Carlo).

Integrales impropias.

Integrales múltiples.

Contenidos procedimentales:

Se ilustran los métodos para integración, se comparan con resultados analíticos.

Se muestra cómo las integrales elípticas se pueden hallar numéricamente con facilidad mientras que las soluciones analíticas son demasiado complicadas.

Para las derivadas numéricas se utilizan las series de Taylor para estimar los errores dependiendo del método.

Métodos de Monte Carlo para integrales que no tienen solución analítica.

Contenidos actitudinales:

Observar lo que ocurre cuando se utilizan un número diferente de intervalos de integración, (muy pocos o demasiados).

Observar los errores que se pueden cometer en derivación numérica cuando se escogen puntos demasiado cercanos o demasiado alejados.

Unidad 5. Álgebra lineal numérica (2 semanas)

Contenidos conceptuales:

Solución de Sistemas de ecuaciones lineales.

Matrices transpuesta, conjugada, hermítica, inversa.

Determinantes, trazas.

Operaciones matriciales.

Autovalores y autofunciones.

Contenidos procedimentales:

Explotar las facilidades que trae el paquete de Numpy de Algebra Lineal.

Contenidos actitudinales:

Observar la facilidad con que se manejan los problemas que involucran matrices utilizando el paquete de Algebra Lineal del Numpy.

Unidad 6. Ecuaciones diferenciales (3 semanas)

Contenidos conceptuales:

Problemas con condiciones iniciales.

Métodos de primer orden: método de Euler.

Métodos de más alto orden: métodos de Runge-Kutta, Leap-Frog (Salto del sapo).

Sistemas de ecuaciones diferenciales.

Problemas con condiciones frontera.

Contenidos procedimentales:

Se solucionan problemas como el movimiento de un planeta o la caída libre con fricción del aire utilizando el método de Euler. Se explican sus limitaciones.

Se utiliza el método de Runge Kutta y otros métodos para solución de ecuaciones diferenciales de segundo orden.

Contenidos actitudinales:

El estudiante debe poder distinguir entre el método de Euler y sus limitaciones y los otros métodos vistos.

Unidad 7. Métodos de Monte Carlo (2 semanas)

Contenidos conceptuales:

Generación de números aleatorios con distribuciones normal,

gaussiana y exponencial decreciente.

Técnicas de montecarlo, caminos aleatorios, caminos aleatorios autoevitables.
Estadística descriptiva.

Contenidos procedimentales:

Se ilustran métodos para generar números aleatorios que estén distribuidos de acuerdo a una distribución gaussiana o una exponencial decreciente, y se comprueba con gráficos que realmente satisfacen estas distribuciones.

Contenidos actitudinales:

Generalmente las rutinas generadoras de números aleatorios, generan realmente números pseudoaleatorios.

Uso de Semillas para mejorar la aleatoriedad.

8. ESTRATEGIAS METODOLÓGICAS

Se explica el fundamento teórico de un método numérico, se detalla el programa en Python, se discute la programación y su fundamento matemático y científico, se analizan los resultados, se estudian los errores que el método conlleva.

Se proponen tareas complementarias para ayudar al estudiante a aplicar el método a situaciones nuevas.

Se utilizan aplicaciones en la nube para compartir información del contenido y las evaluaciones con los estudiantes.

El fraude será severamente castigado.

9. EVALUACIÓN

La evaluación busca que el estudiante:

Permanentemente comprenda los métodos numéricos vistos y los pueda aplicar a nuevas situaciones.

Se de cuenta de los conceptos que no ha entendido correctamente, o no los sabe aplicar, para que repase y logre una buena comprensión.

Esté actualizado y tenga las bases para los temas que se tratarán en clases posteriores.

10. BIBLIOGRAFÍA

Bibliografía básica:

Numerical Analysis, Richard L. Burden & Douglas Faires, Annette M. Burden, 10 E, Cengage Center, Boston, 2015.

Numerical Recipes, the Art of Scientific Computing, William H. Press, Saul A. Teukolsky, William T. Vetterling & Brian P. Flannery. Third Edition. The Cambridge University Press, 2007.

Computational Physics 3er Ed., Landau, Páez, Bordeianu, Wiley VCH, Weinheim, Germany, 2015.

Libro de Landau, Paez y Bordeianu en línea, A survey of computational physics de:
<http://www.compadre.org/psrc/items/detail.cfm?ID=11578>

Introduction to Numerical Programming, Titus Beu, CRC Press, Taylor & Francis Group,
Boca Raton , Fl., 2014.

Bibliografía complementaria:

Python:

Traducción española del manual de Python escrita por su creador Guido
van Rossum: <http://pyspanishdoc.sourceforge.net/tut/tut.html>

Manual escrito por Swaroop, A Byte of Phyton: (en inglés)
<http://www.swaroopch.org/notes/Python>

Libro “Python para todos”, del mejicano Raúl González
Duque: <http://www.s3.com.mx/tutoriales/pythonparatodos.pdf>

Elementary Numerical Analysis, 3rd Ed.,Kendall Atkinson, Weimin Han, Wiley, New York,
2003.

Introduction to Computational and Programming Using Python, J. V. Guttag, The MIT Press,
Cambridge, Mass., 2013

Numerical Methods in Engineering with Python.
Jaan Kiusalass., Cambridge University Press, Cambridge, 2013.

Sitios web del curso: el curso cuenta con su propia página web:
<https://github.com/sbustamante/ComputationalMethods>

Última actualización: Fri, 05 Aug 2016 09:24:22 -0500

Versión legal: La versión legal de este documento reposa en la Biblioteca de la Universidad de Antioquia y esta firmada por el Decano y el Director de Instituto.

Firma Autorizada Facultad Versión Electrónica: (No autorizado. Este documento es solo un borrador.)