GUÍA DOCENTE

INTRODUCCIÓN A LA PROGRAMACIÓN CIENTÍFICA

DOBLE GRADO EN FÍSICA COMPUTACIONAL E INGENIERÍA DEL SOFTWARE

MODALIDAD: PRESENCIAL

CURSO ACADÉMICO: 2026-2027

Denominación de la asignatura:	Introducción a la Programación Científica
Titulación:	DOBLE GRADO EN FÍSICA COMPUTACIONAL E INGENIERÍA DEL SOFTWARE
Facultad o Centro:	Centro Universitario de Tecnología y Arte Digital
Materia:	Métodos Numéricos
Curso:	2
Cuatrimestre:	1
Carácter:	OB
Créditos ECTS:	3
Modalidad/es de enseñanza:	Presencial
Idioma:	Castellano
Profesor/a - email	Francisco Javier García Algarra/javier.algarra@u-tad.com
Página Web:	http://www.u-tad.com/

DESCRIPCIÓN DE LA ASIGNATURA

Descripción de la materia

Esta asignatura se encuentra integrada dentro de la materia de métodos numéricos. La resolución numérica de modelos de sistemas físicos es una herramienta indispensable en el trabajo de ciencia y la ingeniería actuales. El físico computacional recurre a ella como herramienta de investigación y como método de simulación prácticamente en todos los escenarios de trabajo.

Descripción de la asignatura

El objetivo de la asignatura es introducir al alumno en el campo de la computación científica. Partiendo del conocimiento que ha adquirido en asignaturas básicas conocimientos de programación elemental, cálculo y física general, la asignatura se plantea como una toma de contacto con el campo de actuación de la física computacional.

La idea general sobre la que pivota es sobre la de modelo matemático y su importancia dentro del método científico. La resolución numérica de sistemas sencillos (oscilador armónico, campo gravitatorio, circuitos simples) es el hilo conductor de este curso que se plantea en tres fases. En la primera, el alumno aprenderá a simular sistemas básicos con los conocimientos de programación que ha adquirido en la materia básica

(programación en lenguaje C). El objetivo es que sea capaz de resolver un problema físico computacionalmente, sin ligar el proceso a un lenguaje o entorno concreto. Al final de esta fase deben entender que lo importante no es la programación en sí sino la metodología. A continuación, se introduce el lenguaje de programación Python como herramienta profesional en múltiples aplicaciones de la física computacional. El alumno comprobará cómo puede resolver los mismos problemas de una manera mucho más rápida y cómo puede abordar otros más complejos. Con este bagaje se aborda la última parte del curso, en la que se introduce la animación con VPython, como primer contacto con la simulación de sistemas.

Dado que es una asignatura de introducción, se emplearán métodos numéricos muy simples (integración por método del trapecio, método de Euler-Cromer para la resolución de EDOs, etc), pues más adelante cursan una asignatura completa de Análisis Numérico y aun no han visto Métodos Matemáticos para la Física. Eso no impide introducir algún concepto más avanzado como simulación por Monte-Carlo y escenarios que supongan un pequeño reto: ecuación de difusión en una dimensión, oscilador de van der Pol, ecuaciones diferenciales acopladas.

COMPETENCIAS Y RESULTADOS DE APRENDIZAJE DE LA MATERIA

Competencias (genéricas, específicas y transversales)

BÁSICAS Y GENERALES

CG1 - Poseer conocimientos en el área de las Ciencias Físicas y Matemáticas a partir de la base de la educación secundaria general,

a un nivel que, si bien se apoya en libros de texto avanzados, incluye también algunos aspectos que implican conocimientos

procedentes de la vanguardia del estudio de la Física Computacional y la Ciencia de Datos.

CG2 - Aplicar los conocimientos físicos y computacionales de una forma profesional y poseer las competencias que suelen

demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas en el ámbito de la Física

Computacional.

CG7 - Utilizar herramientas de búsqueda de recursos bibliográficos y de Internet.

CB1 - Que los estudiantes hayan demostrado poseer y comprender conocimientos en un área de estudio que parte de la base de la

educación secundaria general, y se suele encontrar a un nivel que, si bien se apoya en libros de texto avanzados, incluye también

algunos aspectos que implican conocimientos procedentes de la vanguardia de su campo de estudio

CB2 - Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las

competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de

su área de estudio

CB3 - Que los estudiantes tengan la capacidad de reunir e interpretar datos relevantes (normalmente dentro de su área de estudio)

para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética

CB4 - Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no

especializado

CB5 - Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores

con un alto grado de autonomía

TRANSVERSALES

CT3 - Conocer los fundamentos hardware y software de los computadores y las redes de comunicación, así como los principios de

almacenamiento y computación en la nube junto con su utilidad y aplicación a los proyectos de desarrollo de la economía digital.

CT4 - Actualizar el conocimiento adquirido en el manejo de herramientas y tecnologías digitales en función del estado actual del

sector y de las tecnologías empleadas.

ESPECÍFICAS

CE1 - Comprender y utilizar el lenguaje matemático para poder aplicar cálculo, álgebra o estadística utilizando métodos numéricos

para la simulación y experimentación de sistemas físicos.

CE8 - Utilizar aplicaciones informáticas de análisis estadístico, cálculo numérico y simbólico, visualización gráfica, optimización u

otras para simular experimentos y resolver problemas.

CE9 - Diseñar modelos matemáticos detallados de sistemas físicos, biológicos o sociales complejos y desarrollar programas que los

simulen en el entorno computacional adecuado.

CE19 - Comprender los principios de optimización de las aplicaciones informáticas para poder realizar simulaciones en tiempo real

o casi real.

Resultados de aprendizaje

· Comprender los principios de la aritmética computacional en los que se basa la resolución numérica de un problema (errores, condicionamiento y estabilidad).

· Resolver numéricamente sistemas de ecuaciones lineales y calcular de forma aproximada valores y vectores propios, utilizando diversos métodos, dependiendo del tipo de matriz.

· Resolver numéricamente ecuaciones y sistemas de ecuaciones no lineales, usando (entre otros) el método de Newton.

· Calcular aproximaciones de una integral definida mediante métodos basados en las fórmulas de cuadratura de Newton-Cotes.

· Aproximar funciones mediante diferentes técnicas de interpolación numérica.

· Conocer y saber aplicar las fórmulas de Gauss para integración numérica.

· Conocer y saber aplicar métodos numéricos para resolver ecuaciones diferenciales ordinarias y algunas ecuaciones en derivadas parciales sencillas.

· Desarrollar programas que implementen algunos de los algoritmos numéricos estudiados, utilizando un lenguaje estructurado.

· Tener criterios para valorar y comparar distintos métodos en función de los problemas a resolver, el coste operativo y la presencia de errores.

· Evaluar los resultados obtenidos y obtener conclusiones después de un proceso de cálculo.

· Utilizar software matemático para cálculo numérico y optimización.

CONTENIDO

· Introducción al lenguaje Python. Paquetes NumPy, SciPy y Matplotlib. Representación de funciones.

· Álgebra: Interpolación, raíces, resolución de sistemas de ecuaciones lineales. Autovalores y autovectores.

· Cálculo: Diferenciación e integración numérica con NumPy y SciPy. Fórmulas de cuadratura, integrales dobles y triples.

· Resolución numérica de EDOs y sistemas de EDOs con Python

· Matemática simbólica con SciPy.

TEMARIO

Tema 1. Introducción

1.1. El papel de los modelos en el método científico.

1.2. Orígenes y papel de la computación científica.

1.3. Lenguajes de programación científica.

Tema 2. Mínimo teórico

2.1. Integración numérica simple.

2.2. Evaluación del error en problemas numéricos.

2.3. Simulación de Monte-Carlo para el cálculo de áreas simples.

2.4. Resolución de ecuaciones diferenciales ordinarias por el método de Euler. Oscilador Armónico Simple.

Tema 3. Lenguaje Python para la programación científica

3.1. Elementos del lenguaje: operadores, variables, funciones, estructuras de control

3.2. Paquetes numéricos (numpy, scipy)

3.3. Gráficos con Matplotlib.

3.4. Animación básica en Python.

3.5. Simulación de sistemas elementales

Tema 4. Problemas físicos con Python

4.1. Introducción a Jupyter Notebook

4.2. Visualización de gráficas

4.3. Interpolación numérica.

4.4. Integración numérica por cuadratura y Monte-Carlo.

4.5. Resolución de Ecuaciones Diferenciales Ordinarias.

4.6. Resolución de Sistemas de Ecuaciones Diferenciales lineales y no lineales

4.7. Aplicaciones: Momentos de inercia, oscilador en campo gravitatorio, circuito RLC serie forzado, filtrado, generación de figuras de Lissajous, descenso de gradiente, ecuación de calor en una dimensión, modelo de Lotka-Volterra, oscilador van der Pol, modelo de epidemias SIR.

Tema 5. Simulación de sistemas con VPython

5.1. Simulación de dinámica de sólidos sencillos.

5.2. Simulación del sistema Sol, Tierra, Luna.

5.3. Simulación general del problema de los tres cuerpos, soluciones estables.

ACTIVIDADES FORMATIVAS Y METODOLOGÍAS DOCENTES

Actividades formativas

Actividad Formativa	Horas totales	Horas presenciales
Clases teóricas / Expositivas	15	100
Clases Prácticas	12	100
Tutorías	3	50
Estudio independiente y trabajo autónomo del alumno	29	0
Elaboración de trabajos (en grupo o individuales)	14	0
Actividades de Evaluación	2	100

Metodologías docentes

MD1 Clase Teoría

MD2 Prácticas

MD3 Prácticas de Laboratorio

MD4 Tutorías

DESARROLLO TEMPORAL

UNIDADES DIDÁCTICAS / TEMAS PERÍODO TEMPORAL

Introducción Semana 1

Resolución de sistemas con lenguaje C Semanas 2, 3, 4

Python para la programación científica Semanas 5, 6, 7

Problemas físicos con Python Semanas 8, 9, 10, 11

Simulación con VPython Semanas 12, 13, 14

SISTEMA DE EVALUACIÓN

ACTIVIDAD DE EVALUACIÓN	VALORACIÓN MÍNIMA RESPECTO A LA CALIFICACIÓN FINAL (%)	VALORACIÓN MÁXIMA RESPECTO A LA CALIFICACIÓN FINAL (%)
Evaluación de la participación en clase, en prácticas o en proyectos de la asignatura	0	30
Evaluación de trabajos, proyectos, informes, memorias	30	60
Prueba Objetiva	30	60

CRITERIOS ESPECÍFICOS DE EVALUACIÓN

ACTIVIDAD DE EVALUACIÓN	CONVOCATORIA ORDINARIA	CONVOCATORIA EXTRAORDINARIA
Evaluación de la participación en clase, en prácticas o en proyectos de la asignatura	10	0
Evaluación de trabajos, proyectos, informes, memorias	40	40
Prueba Objetiva	50	60

Consideraciones generales acerca de la evaluación

• La evaluación de la participación en clase, en prácticas o en proyectos de la asignatura se realizará a partir de la asistencia y la participación activa en clase y en el resto de las actividades desarrolladas durante el curso. Este aspecto representará el 10% de la calificación final de la asignatura en la convocatoria ordinaria.

• A lo largo del curso se plantearán dos ejercicios, con sus correspondientes memorias que deberán ser entregadas antes de la fecha indicada a través de la plataforma virtual. Este trabajo se evaluará a través de la propia plataforma virtual y supondrá un 40% de la calificación final de la asignatura en la convocatoria ordinaria. La presentación tardía de dichas entregas supondrá una penalización en la nota. No se permitirá la entrega de ninguna actividad con retraso de más de una semana. La media de las dos prácticas debe ser al menos 4.0 para considerar esta parte superada.

• La prueba objetiva consistirá en la defensa ante un panel de profesores del grado del proyecto final. Los alumnos recibirán en tiempo debido la pauta de corrección.

• Para aprobar la asignatura en la convocatoria ordinaria, es imprescindible que la nota final sea al menos 5.0 (sobre 10). Además de ese requisito, es necesario que la prueba objetiva alcance un 5.0 (sobre 10). En caso de no cumplirse alguno de estos requisitos, la asignatura se considerará automáticamente suspensa independientemente del resto de calificaciones.

• En caso de no conseguir el aprobado en la convocatoria de junio, el alumno podrá presentarse a la convocatoria extraordinaria de julio. En esta convocatoria no se tendrá en cuenta el 10% de participación en clase y la nota de la prueba final pondere el 60% del total, siendo imprescindible obtener al menos 5.0 en este apartado. Las notas de las tres prácticas se conservarán si la nota media de las tres superaba el 4.0, de lo contrario el alumno tendrá que resolver nuevas prácticas antes de la prueba final.

• No se conservarán calificaciones de ningún tipo entre distintos cursos académicos.

BIBLIOGRAFÍA / WEBGRAFÍA

Bibliografía Básica:

• Gaël Varoquaux (ed) et al (2023). Scientific Python Lectures. https://lectures.scientific-python.org

• Varó, A. M., Luengo, I. G., & Sevilla, P. G. (2014). Introducción a la programación con Python 3. Universitat Jaume I. Servei de Comunicació i Publicacions

Bibliografía Recomendada:

• Malthe-Sørenssen, A. (2015). Elementary mechanics using Python: A modern course combining analytical and numerical technique. Springer International Publishing Switzerland

• Fitzpatrick, R. (2006). Computational physics. Lecture notes, University of Texas at Austin.

MATERIALES, SOFTWARE Y HERRAMIENTAS NECESARIAS

Tipología del aula
Aula teórica

Equipo de proyección y pizarra

Materiales:
Ordenador personal .

Cuaderno o tablet para tomar apuntes.

Software:
C, Python, VPython