
 <p>UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS</p>	<p>UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS</p> <p>Facultad de Ciencias Matemáticas y Naturales</p> <p>Syllabus</p>	
--	---	---

Programa académico de Física

Nombre del espacio académico	Física Computacional 1		
Código del espacio	25126	Número de créditos	2

TIPO DE CURSO:	TEÓRICO	X	PRÁCTICO		TEÓRICO-PRÁCTICO	
----------------	---------	---	----------	--	------------------	--

TIPO DE ESPACIO ACADÉMICO:	SÍ	NO	NÚMERO DE HORAS:	
Obligatorio básico	x		Trabajo directo	2
Obligatorio complementario		x	Trabajo mediano	1
Electivo intrínseco		x	Trabajo autónomo	1
Electivo extrínseco		x		

Ubicación dentro de la malla curricular	SEXTO SEMESTRE
---	----------------

Justificación del espacio académico
<p>La ciencia se ha dividido tradicionalmente en disciplinas experimentales y teóricas, pero durante las últimas décadas la computación ha emergido como una parte muy importante de la ciencia. La computación científica a menudo está estrechamente relacionada con la teoría, pero también tiene muchas características en común con el trabajo experimental. Por lo tanto, a menudo se ve como una nueva (tercera) rama de la ciencia. En la mayoría de los campos de la ciencia, el trabajo computacional es un complemento importante tanto para los experimentos como para la teoría, y hoy en día la mayoría de los trabajos tanto experimentales como teóricos involucran algunos cálculos numéricos, simulaciones o modelos computacionales.</p> <p>En las ciencias experimentales y teóricas, existen códigos de conducta bien establecidos sobre cómo los resultados y los métodos se publican y se ponen a disposición de otros científicos. Por ejemplo, en ciencias teóricas, las derivaciones, las pruebas y otros resultados se publican con todo detalle, o se ponen a disposición a pedido. Del mismo modo que en las ciencias experimentales, los métodos utilizados y los resultados se publican, y todos los datos experimentales deben estar disponibles a pedido. Se considera que no es científico retener detalles cruciales en una prueba teórica o método experimental, que impediría a otros científicos replicar y reproducir los resultados.</p> <p>En ciencias computacionales, todavía no existen pautas bien establecidas sobre cómo se deben manejar el código fuente y los datos generados. Por ejemplo, es relativamente raro que el código fuente utilizado en las simulaciones para los artículos publicados se proporcione a los lectores, en</p>



UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

Facultad de Ciencias
Matemáticas y Naturales

Syllabus



Justificación del espacio académico

contraste con la naturaleza abierta del trabajo experimental y teórico. Y no es raro que el código fuente para el software de simulación se retenga y se considere una ventaja competitiva (o no sea necesario publicarlo).

Sin embargo, este problema ha comenzado recientemente a atraer cada vez más atención, y varios editoriales en revistas de alto perfil han pedido una mayor apertura en las ciencias computacionales. Algunas revistas prestigiosas, incluida Science, incluso han comenzado a exigir a los autores que proporcionen el código fuente para el software de simulación utilizado en las publicaciones a los lectores que lo soliciten. También se están llevando a cabo discusiones sobre cómo facilitar la distribución de software científico, por ejemplo, como material complementario a los documentos científicos.

Prerrequisitos/conocimientos previos: **Ecuaciones Diferenciales 2, Métodos Numéricos.**

Programación del contenido

1. Elementos computacionales para Ciencias Física. Sistemas Operativos: Windows y Linux. Lenguaje Bash, Herramientas de colaboración: Git, GitHub, GitLab y BitBucket. Editores de Texto: Visual Studio Code (VSC), Nano, Gedit, Geany. Herramientas de visualización: Gnuplot, Xmgrace. -Introducción al manejo de software y CAS: Excel, Octave, Libre Office, Scilab, Maxima, Tracker, Computación Simbólica. Depuración de Código. Fuentes, binarios, makefiles, librerías y repositorios.
2. Introducción a lenguajes de programación: C++, Fortran y Python.
3. Implementación. Derivación Numérica: Diferencias Finitas, Diferencias Backward y Forward, Diferencias centradas y Segunda derivada. Integración Numérica: Métodos de Newton-Cotes: Regla del Rectángulo, Regla del Trapecio, Regla de Simpson. Cuadratura Gaussiana. Ecuaciones Diferenciales: Método de Euler. Método Runge-Kutta.
4. Dinámica clásica y no lineal. Movimiento de partícula en una y dos dimensiones: Caída libre, trayectoria parabólica y circular (Problema de dos cuerpos). Osciladores: Osciladores lineales (Sistema de Lorentz, Circuito de Chua). Osciladores no lineales. Modelos de fricción. Resonancia lineal y no lineal. Famosos osciladores no lineales: Ecuación de Van der Pool, Ecuación de Duffing, Péndulo. Doble Péndulo. Solución vía Computación simbólica.
5. Ecuaciones diferenciales acopladas. Osciladores lineales acoplados. Amortiguamiento y forzamiento. Problemas de valores propios.
6. Ecuaciones diferenciales parciales. Serie y transformada de Fourier. Ecuaciones elípticas, parabólicas e hiperbólicas. Diferencias y elementos finitos.
7. Proyectos.



UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Facultad de Ciencias
Matemáticas y Naturales

Syllabus



Estrategias

Metodología pedagógica y didáctica:

Métodos Instructivos: En cada una de las temáticas a desarrollar se hará una presentación magistral, enmarca en la descripción de la teoría de manera rigurosa (desarrollo de pensamiento lógico formal) dentro de las posibilidades de construcción y participación de los estudiantes, haciendo énfasis en aspectos prácticos y en la comprensión de modelos; en lo posible las temáticas se complementarán con sesiones prácticas. Un componente importante de la asignatura serán los talleres, en los cuales se profundizará en el material expuesto por medio de diferentes actividades. Se fomentará una activa participación de los estudiantes en todas las actividades programadas.

Métodos de evaluación:

1. Exámenes parciales.
2. Tareas asignadas.
3. Examen final.

Bibliografía

- [1] Konstantinos Anagnostopoulos. *Computational Physics-A Practical Introduction to Computational Physics and Scientific Computing (using C++)*, Vol. II, volume 2. Lulu. com, 2016.
- [2] Joseph F Boudreau and Eric S Swanson. *Applied computational physics*. Oxford University Press, 2017.
- [3] Richard Fitzpatrick. Computational physics. *Lecture notes, University of Texas at Austin*, 2006.
- [4] Rubin H Landau, Manuel J Páez, and Cristian C Bordeianu. *Computational physics: Problem solving with Python*. John Wiley & Sons, 2015.
- [5] Jos Thijssen. *Computational physics*. Cambridge university press, 2007.
- [6] Darren Walker. *Computational physics*. Stylus Publishing, LLC, 2022.
- [7] David Yevick. *A First Course in Computational Physics and Object-Oriented Programming with C++ Hardback with CD-ROM*. Cambridge University Press, 2005.