



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS
Licenciatura en Física y Matemáticas
Instituto de Investigación en Ciencias Básicas y
Aplicadas



Ecuaciones Diferenciales Ordinarias

Modelación matemática • Sistemas dinámicos • Métodos analíticos • Computación científica

Syllabus del curso

Otoño 2026

Docente:	Dra. María Luisa Daza Torres
Afiliación:	Instituto de Matemáticas, Unidad Cuernavaca, UNAM
Curso impartido en:	Universidad Autónoma del Estado de Morelos
Semestre:	Tercer semestre
Periodo escolar:	17 de agosto al 5 de diciembre de 2026
Horario de clase:	_____
Salón:	_____
Correo electrónico:	luisa.daza@im.unam.mx
Horario de atención:	Viernes de 9:00 a 13:00 h (o previa cita por correo electrónico).
Oficina:	Instituto de Matemáticas, Unidad Cuernavaca, Oficina N8.

Los materiales del curso, tareas semanales, notas, códigos y cuadernos de Jupyter estarán disponibles en la sección *Lectures* del sitio web:

mdazatorres.github.io



Descripción del curso

Las ecuaciones diferenciales ordinarias constituyen una de las herramientas fundamentales para describir la evolución de fenómenos físicos, biológicos, químicos y económicos. En este curso se estudiarán los principales métodos analíticos para resolver ecuaciones diferenciales de primer orden, de orden superior y sistemas de ecuaciones diferenciales.

El curso enfatizará tres perspectivas complementarias: la resolución analítica, la interpretación geométrica y cualitativa de las soluciones, y la formulación de modelos matemáticos. Como complemento al tratamiento teórico, se utilizará **Python** para visualizar soluciones, realizar experimentos computacionales y resolver numéricamente problemas que no admitan una solución cerrada sencilla.

El contenido se apega al programa oficial de la unidad de aprendizaje *Ecuaciones Diferenciales Ordinarias* de la Licenciatura en Física y Matemáticas de la UAEM.

Propósito formativo

Al finalizar el curso, el estudiantado distinguirá y aplicará los principales métodos de solución de ecuaciones diferenciales ordinarias y sistemas de ecuaciones diferenciales. Asimismo, podrá modelar fenómenos sencillos, interpretar el comportamiento de sus soluciones y utilizar herramientas computacionales para explorar y comunicar resultados.

Resultados de aprendizaje

Al finalizar el semestre, el estudiante será capaz de:

- clasificar ecuaciones diferenciales ordinarias según su orden, linealidad y estructura;
- resolver analíticamente las principales familias de ecuaciones diferenciales de primer orden y de orden superior;
- formular y analizar problemas de valor inicial;
- interpretar geométricamente campos direccionales, líneas de fase y retratos de fase;
- analizar estabilidad y comportamiento cualitativo de sistemas dinámicos sencillos;
- formular modelos matemáticos de fenómenos físicos, biológicos o de ingeniería;
- implementar soluciones numéricas y visualizaciones en Python;
- comunicar procedimientos, resultados y limitaciones de un modelo de manera escrita y oral.

Prerrequisitos

- Cálculo diferencial e integral.
- Técnicas básicas de integración.
- Álgebra lineal elemental.
- No se requiere experiencia previa en programación.

Organización y metodología

El curso se impartirá en dos sesiones semanales de 2.5 horas. Cada semana combinará, según el tema, los siguientes componentes:

1. **Fundamentos teóricos:** conceptos, resultados y derivaciones.
2. **Resolución de problemas:** ejercicios analíticos y discusión de estrategias de solución.
3. **Python de la semana:** introducción gradual de comandos y herramientas computacionales.
4. **Aplicación de la unidad:** análisis de un problema aplicado relacionado con los contenidos de la unidad, cuando resulte pertinente.

Materiales del curso

Las notas de clase, tareas semanales, cuadernos de Jupyter, códigos en Python, lecturas complementarias, instrucciones del proyecto y avisos estarán disponibles en:

mdazatorres.github.io

En la sección **Lectures** se publicarán:

- notas y presentaciones de clase;
- tareas semanales;
- cuadernos de Jupyter;
- códigos en Python;
- lecturas y material complementario;
- instrucciones del proyecto final;
- avisos relacionados con el curso.

Se recomienda consultar el sitio web antes de cada clase.

Software

Se utilizarán las siguientes herramientas:

- Python;
- Jupyter Notebook o Google Colab;
- NumPy;
- SciPy;
- Matplotlib;
- SymPy.

Ejes transversales del curso

Python de la semana

No se presupone experiencia previa en programación. En cada unidad se introducirán únicamente los comandos necesarios para resolver el problema correspondiente. El énfasis estará en comprender qué se calcula, cómo se representa y cómo se interpreta el resultado.

Aplicación de la unidad

Las unidades que lo permitan incluirán una aplicación representativa en la que se utilizarán ecuaciones diferenciales para describir un fenómeno real. El objetivo será conectar la teoría con la formulación de modelos, la interpretación de parámetros y el análisis de resultados. En las unidades de carácter más técnico, la actividad podrá sustituirse por un laboratorio computacional.

Contenido del curso

Unidad I. Temas introductorios

- Definiciones y terminología.
- EDO vs ecuaciones diferenciales parciales.
- Orden, linealidad y forma general de una EDO.
- Soluciones explícitas e implícitas.
- Familias de soluciones.
- Problemas de valor inicial y problemas de frontera.
- Introducción al software matemático.

Python de la semana

Introducción a Jupyter Notebook; variables; operaciones; definición de funciones; arreglos con NumPy; `linspace`; gráficas básicas con Matplotlib.

En esta unidad no se incluirá una aplicación independiente. La prioridad será establecer el lenguaje básico de las ecuaciones diferenciales e introducir las herramientas computacionales que se utilizarán durante el semestre.

Unidad II. Ecuaciones diferenciales de primer orden

- Ecuaciones separables.
- Ecuaciones lineales y factor integrante.
- Principio de superposición.
- Ecuaciones exactas y factores integrantes.

- Sustituciones, ecuaciones homogéneas y ecuación de Bernoulli.
- Campos direccionales, isoclinas y curvas solución.
- Líneas de fase y puntos de equilibrio.
- Ecuación logística.
- Introducción a las bifurcaciones.
- Teorema de existencia y unicidad.

Python de la semana

`meshgrid`, `quiver`, `solve_ivp`, visualización de campos direccionales y comparación de curvas solución.

Aplicación de la unidad

Enfriamiento de Newton y crecimiento logístico.

Unidad III. Ecuaciones diferenciales de orden superior

- Teoría preliminar de ecuaciones lineales.
- Reducción de orden.
- Ecuaciones lineales homogéneas con coeficientes constantes.
- Ecuación característica y raíces.
- Ecuaciones lineales no homogéneas.
- Variación de parámetros.
- Conversión entre ecuaciones de orden superior y sistemas.
- Problemas masa–resorte.
- Movimiento libre, amortiguado, forzado y resonancia.

Python de la semana

Conversión de una ecuación de segundo orden en un sistema de primer orden; `solve_ivp`; comparación entre solución analítica y numérica.

Aplicación de la unidad

Oscilador armónico y sistema masa–resorte amortiguado o forzado.

Unidad IV. Soluciones en series de potencias

- Repaso de series de potencias.
- Puntos ordinarios y singulares.

- Soluciones alrededor de un punto ordinario.
- Introducción al método de Frobenius.

Python de la semana

Uso básico de SymPy para desarrollar series y comparar aproximaciones truncadas.

Laboratorio computacional de la unidad. Comparación entre una solución exacta, una aproximación por series truncadas y una solución numérica. En esta unidad no se forzará una aplicación de modelación independiente, pues el énfasis estará en la técnica de aproximación.

Unidad V. Sistemas de ecuaciones diferenciales ordinarias

- Sistemas lineales de primer orden.
- Valores y vectores propios.
- Interpretación geométrica de vectores propios.
- Estabilidad de sistemas lineales.
- Valores propios reales distintos, repetidos y complejos.
- Sistemas no homogéneos.
- Variación de parámetros.
- Sistemas autónomos y diagramas de fase.
- Modelo depredador–presa de Lotka–Volterra.

Python de la semana

Matrices con NumPy; `numpy.linalg.eig`; `streamplot`; retratos de fase y trayectorias.

Aplicación de la unidad

Modelo depredador–presa y análisis cualitativo de sus trayectorias.

Unidad VI. Transformada de Laplace

- Definición de la transformada de Laplace.
- Transformadas de funciones elementales.
- Transformada inversa y convolución.
- Función de Heaviside.
- Función delta de Dirac.
- Sistemas de ecuaciones mediante transformadas.

Python de la semana

Uso de SymPy para calcular transformadas directas e inversas y verificar soluciones.

Aplicación de la unidad

Respuesta de un circuito RLC o de un sistema mecánico sometido a una entrada discontinua.

Evaluación

Actividad	Porcentaje
Tareas y participación asociada	30 %
Exámenes parciales	30 %
Examen final	20 %
Proyecto final	20 %
Total	100 %

Tareas y participación

Las tareas se asignarán semanalmente y tendrán como propósito consolidar los conceptos y técnicas estudiados. Su evaluación se realizará principalmente mediante participación en clase.

Durante las sesiones, la profesora seleccionará estudiantes para presentar, explicar o discutir uno de los problemas asignados. La evaluación considerará:

- preparación previa;
- comprensión del procedimiento;
- razonamiento matemático;
- claridad de la explicación;
- capacidad para responder preguntas y corregir errores.

No se evaluará únicamente la respuesta final. Se espera que cada estudiante prepare todos los problemas de la tarea, aunque solo se le solicite presentar uno de ellos.

Exámenes parciales

Se realizarán exámenes parciales durante el semestre. Estos evaluarán técnicas analíticas, interpretación cualitativa, formulación de modelos y, cuando corresponda, lectura o interpretación de resultados computacionales.

Examen final

El examen final será acumulativo e integrará los conceptos centrales del curso.

Proyecto final

El proyecto final se realizará en equipos y consistirá en estudiar un fenómeno real mediante una o más ecuaciones diferenciales ordinarias.

El proyecto deberá incluir:

- descripción del fenómeno y pregunta de interés;
- formulación del modelo;
- definición e interpretación de variables y parámetros;
- análisis matemático;
- solución analítica cuando sea posible;
- solución o exploración numérica en Python;
- visualización e interpretación de resultados;
- discusión de supuestos y limitaciones;
- reporte escrito y presentación oral.

Ejemplos de temas posibles:

- crecimiento poblacional;
- caída con resistencia del aire;
- osciladores y péndulos;
- circuitos RLC;
- modelos epidemiológicos;
- sistemas depredador–presa;
- cinética química;
- mezcla de soluciones;
- comparación de métodos numéricos.

Cronograma tentativo

Semana	Fechas	Tema principal	Python / aplicación
1	17–22 ago.	Introducción, clasificación de EDO, soluciones y problemas de valor inicial.	Python: Jupyter, variables y gráficas.
2	24–29 ago.	Ecuaciones separables y modelos elementales.	Crecimiento exponencial.
3	31 ago.–5 sep.	Ecuaciones lineales de primer orden y factor integrante.	Enfriamiento de Newton.
4	7–12 sep.	Ecuaciones exactas, homogéneas y de Bernoulli.	Campos direccionales.

Semana	Fechas	Tema principal	Python / aplicación
5	14–19 sep.	Líneas de fase, equilibrio, modelo logístico y bifurcaciones.	Python: <code>solve_ivp</code> .
6	21–26 sep.	Existencia y unicidad. Repaso de la Unidad II.	Primer examen parcial.
7	28 sep.–3 oct.	Ecuaciones lineales de orden superior y ecuación característica.	Soluciones analíticas y numéricas.
8	5–10 oct.	Ecuaciones no homogéneas y variación de parámetros.	Sistema masa–resorte.
9	12–17 oct.	Movimiento amortiguado, forzado y resonancia.	Python: osciladores.
10	19–24 oct.	Series de potencias y puntos ordinarios.	Laboratorio: aproximaciones truncadas y solución numérica.
11	26–31 oct.	Puntos singulares y método de Frobenius.	Segundo examen parcial.
12	2–7 nov.	Sistemas lineales, valores y vectores propios.	Python: matrices y <code>eig</code> .
13	9–14 nov.	Estabilidad y retratos de fase.	Python: <code>streamplot</code> .
14	16–21 nov.	Sistemas no lineales y Lotka–Volterra.	Taller del proyecto final.
15	23–28 nov.	Transformada de Laplace, Heaviside y delta de Dirac.	Aplicación a circuitos o sistemas mecánicos.
16	30 nov.–5 dic.	Integración del curso, presentaciones de proyectos y evaluación final.	Presentaciones finales.

El cronograma es tentativo y podrá ajustarse de acuerdo con el avance del grupo, los días inhábiles oficiales y las necesidades académicas del curso. Cualquier cambio se anunciará en clase y en el sitio web.

Políticas generales

Asistencia y participación

La asistencia regular es importante porque parte de la calificación de tareas se obtendrá mediante la presentación y discusión de problemas en clase. Las inasistencias justificadas deberán comunicarse oportunamente.

Entregas tardías

Las condiciones para aceptar tareas o avances del proyecto después de la fecha establecida se comunicarán junto con cada actividad. Las extensiones deberán solicitarse antes de la fecha límite, salvo situaciones imprevistas justificadas.

Trabajo colaborativo

Se permite discutir ideas y estrategias con compañeros. Sin embargo, cada estudiante deberá comprender y ser capaz de explicar las soluciones que presente. En evaluaciones individuales no se permite colaboración.

Integridad académica

Todo trabajo entregado deberá respetar los principios de honestidad académica. El plagio, la copia no autorizada o la presentación de trabajo ajeno como propio serán tratados conforme a la normativa institucional.

Bibliografía

Bibliografía básica

- Boyce, W. E. y DiPrima, R. C. *Elementary Differential Equations and Boundary Value Problems*. Wiley.
- Zill, D. G. *Ecuaciones diferenciales con aplicaciones de modelado*. Thomson.
- Birkhoff, G. y Rota, G. C. *Ordinary Differential Equations*. Wiley.
- Rainville, E. D., Bedient, P. E. y Bedient, R. E. *Ecuaciones diferenciales elementales*. Prentice Hall.

Bibliografía complementaria

- Ricardo, H. *Ecuaciones diferenciales: una introducción moderna*. Reverté.
- Borrelli, R. y Coleman, C. S. *Ecuaciones diferenciales: una perspectiva de modelación*. Oxford University Press.
- Lomen, D. y Lovelock, D. *Ecuaciones diferenciales a través de gráficas, modelos y datos*. CECSA.

Python y herramientas computacionales

- James, G., Witten, D., Hastie, T., Tibshirani, R. y Taylor, J. *An Introduction to Statistical Learning with Applications in Python*. Springer.
- VanderPlas, J. *Python Data Science Handbook*. O'Reilly.

Filosofía del curso

Las ecuaciones diferenciales representan uno de los lenguajes fundamentales de la ciencia. Este curso no busca únicamente que el estudiante memorice técnicas de solución, sino que aprenda a formular preguntas, construir modelos, interpretar soluciones y utilizar herramientas computacionales para explorar fenómenos dinámicos.

Al finalizar el semestre, se espera que el estudiantado pueda integrar conocimientos de cálculo, álgebra lineal, modelación y programación para abordar problemas provenientes de la física, las matemáticas y otras áreas científicas.