



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE CIENCIAS
CARRERA FÍSICA

**DESARROLLO DE ECUACIONES DIFERENCIALES DE
SEGUNDO ORDEN PARA EL APRENDIZAJE DE FÍSICA
COMPUTACIONAL EN PYTHON 3.10.0**

Trabajo de Titulación

Tipo: Proyecto Integrador

Presentado para optar al grado académico de:

FISICO

AUTOR: DIEGO ALEJANDRO LOGROÑO TORRES

DIRECTOR: Mat. LUIS MARCELO CORTEZ BONILLA

Riobamba – Ecuador

2022

© 2022, **Diego Alejandro Logroño Torres**

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo cita bibliográfica del documento, siempre y cuando se reconozca el Derecho del Autor.

Yo, DIEGO ALEJANDRO LOGROÑO TORRES, declaro que el presente trabajo de titulación es de mi autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor asumo la responsabilidad legal y académica de los contenidos de este trabajo de titulación; el patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 16 de marzo de 2022

A handwritten signature in blue ink, consisting of several overlapping loops and strokes, positioned diagonally on the page.

Diego Alejandro Logroño Torres

060565267-6

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE CIENCIAS
CARRERA FÍSICA

El Tribunal del Trabajo de Titulación certifica que: El trabajo de titulación; tipo: Proyecto Integrador, **DESARROLLO DE ECUACIONES DIFERENCIALES DE SEGUNDO ORDEN PARA EL APRENDIZAJE DE FÍSICA COMPUTACIONAL EN PYTHON 3.10.0**, realizado por el señor: **DIEGO ALEJANDRO LOGROÑO TORRES**, ha sido minuciosamente revisado por los Miembros del Trabajo de Titulación, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal Autoriza su presentación.

FIRMA

FECHA

Dr. Richard Williams Pachacama Choca MSc

RICHARD
WILLIAMS
PACHACAMA
CHOCA



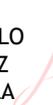
RICHARD WILLIAMS PACHACAMA CHOCA
c-EC, serialNumber=0601921703,
ou=PACHACAMA CHOCA, ou=RICHARD
WILLIAMS PACHACAMA CHOCA,
t.3.6.1.4.1.31462.10.4-0601921703,
email=pachacama@gmail.com,
cn=RIORDIANA, s=CHIMBORAZO,
ou=Certificado de Clase 2 de Persona Fisica
EC (FIRMA)

2022-03-16

PRESIDENTE DEL TRIBUNAL

Mat. Luis Marcelo Cortez Bonilla MSc

LUIS
MARCELO
CORTEZ
BONILLA



Firmado digitalmente por LUIS
MARCELO CORTEZ BONILLA
Nombre de reconocimiento (DN): c=EC,
o=BANCO CENTRAL DEL ECUADOR,
ou=ENTIDAD DE CERTIFICACION DE
INFORMACION EC/CECE, l=QUITO,
serialNumber=0000497075, cn=LUIS
MARCELO CORTEZ BONILLA
Fecha: 2022.06.08 16:34:01 -0500

2022-03-16

**DIRECTOR DE TRABAJO DE
TITULACIÓN**

Biof. María Fernanda Heredia Moyano MSc



Firmado electrónicamente por:
**MARIA FERNANDA
HEREDIA MOYANO**

2022-03-16

MIEMBRO DEL TRIBUNAL

DEDICATORIA

Esta tesis está dedicada a:

Las personas que de manera directa o indirecta me han apoyado en este arduo pero hermoso tiempo de estadía en la universidad empezando por mi madre Silvia que creyó en mí y siempre estuvo a mi lado incondicionalmente, a mi abuelito Miguel que me enseñó a ser un hombre con valores, a mi mascota Benito que me brindó amor y ternura, a mi tía Yadira y su Esposo Ángel quienes me ayudaron en los últimos instantes de mi formación académica, a mi hermano Byron, a mi tía Marcia, a mi tía Patricia a mi tía María quienes fueron importantes en mi desarrollo personal, a mis maestros de la universidad quienes me enseñaron a amar la ciencia, a mis amigos que forme dentro y fuera de la universidad Anahí, Aldo, Dennis, Felipe, que me brindaron momentos de alegría. Y a las personas que día a día compartía un momento mientras me trasladaba a clases, de regreso a casa, a comer, que siempre con una sonrisa y un saludo hacían que el día y el mundo fuera mejor.

Diego

AGRADECIMIENTO

Quiero agradecer a mi madre Silvia quien me inculco que con trabajo duro, humildad y perseverancia todo es posible, a mi tutor Marcelo Cortez quien con paciencia y consejos me supo orientar por el mejor camino, a mis familiares y amigos que me brindaron su apoyo incondicional, a la noble Escuela Superior Politécnica de Chimborazo quien me abrió las puertas para poder forjarme profesionalmente, a mis maestros quienes me ayudaron a escalar cada uno de los peldaños.

Diego

TABLA DE CONTENIDO

ÍNDICE DE TABLAS.....	viii
ÍNDICE DE FIGURAS.....	ix
RESUMEN.....	x

CAPÍTULO I

1.	MARCO TEÓRICO	1
1.1.	Bases teóricas.....	1
1.1.1.	<i>Python</i>	1
1.1.2.	<i>Librerías en Python</i>	1
1.1.3.	<i>Física computacional</i>	2
1.1.4.	<i>Modelos matemáticos</i>	3
1.1.5.	<i>Actualidad en la educación Matemática</i>	3
1.1.6.	<i>Ecuaciones diferenciales</i>	4
1.2.	BASES CONCEPTUALES	10
1.2.1.	<i>Solución General</i>	10
1.2.2.	<i>Solución Particular</i>	10
1.2.3.	<i>Solución Singular</i>	10
1.2.4.	<i>Matrices</i>	10
1.2.5.	<i>Determinantes</i>	10
1.2.6.	<i>Técnicas de Reducción de Orden</i>	10
1.2.7.	<i>Caso General</i>	13

CAPÍTULO II

2.	MARCO METODOLÓGICO.....	15
2.1.	Tipo de Investigación	15
2.2.	Diseño de la Investigación.....	15
2.2.1.3.	<i>Matriz de consistencia</i>	16
2.2.1.4.	<i>Operacionalización de las variables</i>	17
2.2.1.5.	<i>Localización del Estudio</i>	17
2.2.1.6.	<i>Población de estudio</i>	17
2.2.1.7.	<i>Tamaño de la muestra</i>	18
2.2.1.8.	<i>Método de muestreo</i>	18

2.2.1.9.	<i>Análisis Estadístico Inferencial</i>	18
2.3.	TÉCNICAS DE INVESTIGACIÓN	18
2.3.1.	<i>Método de Runge-Kutta para Ecuaciones Diferenciales</i>	18

CAPÍTULO III

3.	MARCO DE ANÁLISIS E INTERPRETACION DE RESULTADOS	22
3.1.	ANÁLISIS DE DATOS	22
3.1.1.	<i>Ejemplo 1</i>	22
3.1.2.	<i>Ejemplo 2</i>	26
3.1.3.	<i>Ejemplo 3</i>	27
3.1.4.	<i>Ejemplo 4</i>	30
	CONCLUSIONES	34
	RECOMENDACIONES	35

BIBLIOGRAFIA

ÍNDICE DE TABLAS

Tabla 1-1: Matriz de consistencia	16
Tabla 2-1: Operacionalización de las variables.....	17

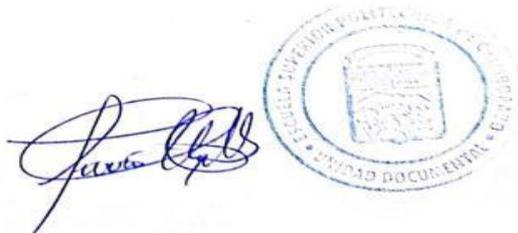
ÍNDICE DE FIGURAS

Figura 2-1: Interfaz gráfica Anaconda.....	20
Figura 2-2: Interfaz gráfica Jupyter	21
Figura 3-1: Llamado a librerías “numpy” y “matplotlib”	22
Figura 3-1: Definición de funciones	22
Figura 3-2: Cálculos	23
Figura 3-3: Graficación.....	23
Figura 3-4: Ejecución	28
Figura 3-5: Resultados	28
Figura 3-6: Ilustración del sistema masa – resorte – amortiguador	30
Figura 3-7: Ejecución	31

RESUMEN

Durante el tiempo de emergencia sanitaria los docentes y estudiantes de la carrera de Física de la Escuela Superior Politécnica de Chimborazo no han tenido acceso a los laboratorios y han implementado el uso de laboratorios virtuales por lo cual, se presentó esta propuesta de trabajo integrador en el cual se pondrá en funcionamiento las diferentes prácticas que se puede realizar con el programa Python. En el presente trabajo se determinó la importancia de desarrollar ecuaciones diferenciales de segundo orden para el aprendizaje de física computacional en Python 3.10.0; para llevar a cabo, se trabajó en un tipo de investigación de revisión bibliográfica ya que puede ser aplicada a cualquier tema de investigación y determinar la relevancia e importancia de este, y asegurar la originalidad de una investigación. Así como también es de carácter experimental pues, se ha procedido a realizar ejercicios en el software Python. Los resultados demuestran que la utilización del método Runge- Kutta de cuarto orden, es ideal para resolver ecuaciones diferenciales de segundo orden y orden N, pues solo se precisa conocer los métodos de reducción de ecuaciones y así poder aplicar dicho algoritmo, también proporciona un pequeño margen de error con respecto a la solución real del problema y es fácilmente programable en Python para realizar las iteraciones necesarias. Se concluye que, al desarrollar los ejercicios de ecuaciones diferenciales de segundo orden, Python es una herramienta de física computacional muy poderosa mediante sus librerías “numpy” y “matplotlib” ya que “mumpy” realiza EDO de primer orden la cual permitió realizar los cálculos de las EDO al aplicar el método de Runge-Kutta, y “matplotlib” permite graficar de forma idónea los resultados obtenidos. Finalmente se recomienda utilizar las variables descritas en el presente trabajo y a su vez elaborar una guía metodológica del mismo.

Palabras clave: <SIMULACIÓN>, <PYTHON (PROGRAMA)>, < ECUACIONES DIFERENCIALES>, <ECUACIONES DIFERENCIALES DE SEGUNDO ORDEN>.

The image shows a handwritten signature in blue ink on the left and a circular blue stamp on the right. The stamp contains the text 'ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO' around the top edge and 'UNIDAD DOCU-ENTRENA' around the bottom edge. In the center of the stamp is a small emblem.

0989-DBRA-UTP-2022

ABSTRACT

During the health emergency, the teachers, and students of the Physics course at the Escuela Superior Politécnica de Chimborazo have not had access to the laboratories and have implemented the use of virtual laboratories. For this reason, this integrative work proposal was presented in which the different practices that can be carried out with the Python program will be put into operation. In this work, the importance of developing second order differential equations for learning computational physics in Python 3.10.0 was determined; to carry it out, we worked on a type of literature review research as it can be applied to any research topic and determine the relevance and importance of this and ensure the originality of a research. It is also experimental in nature, as exercises have been carried out in Python software. The results show that the use of the fourth order Runge-Kutta method is ideal for solving second-order and N-order differential equations, since it is only necessary to know the methods of equation reduction and thus be able to apply this algorithm, it also provides a small margin of error with respect to the real solution of the problem, and it is easily programmable in Python to carry out the necessary iterations. It is concluded that, when developing the second order differential equations exercises, Python is a very powerful computational physics tool through its "numpy" and "matplotlib" libraries, since "numpy" performs first order ODEs, which allowed performing the ODE calculations by applying the RungeKutta method, and "matplotlib" allows the results obtained to be plotted in a suitable way. Finally, it is recommended to use the variables described in this work and at the same time to elaborate a methodological guide of the same.

Keywords: <SIMULATION>, <PYTHON (PROGRAM) >, <DIFFERENTIAL EQUATIONS>, <SECOND ORDER DIFFERENTIAL EQUATIONS>.



Dra. Carmita Eulalia Rojas Castro. Mgs.

0602890469

0989-DBRA-UTP-2022

CAPÍTULO I

1. MARCO TEÓRICO

1.1. Bases teóricas

1.1.1. *Python*

Es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código 2. Se trata de un lenguaje de programación multiparadigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma. Python, el lenguaje de programación seleccionado, facilita portabilidad y el acceso a las librerías necesarias para la representación de partículas. Las principales librerías de Python usadas en este ejercicio son: matplotlib, numpy, PyQt4, scipy, Tkinter y VPython (Arias , Jiménez , & Porras , 2016).

Según Johansson (2015), este lenguaje permite escribir código en varios estilos de programación -imperativa y orientación a objetos- con una sintaxis fácil de leer y actualizar. De acuerdo con Solano (2011), entre las ventajas más importantes del uso del lenguaje se encuentran su sintaxis elegante y legible; el entorno de ejecución que coadyuva en la detección de errores de codificación; la posibilidad de escribir código orientado a objetos; su vasta estructura de datos y la expresividad que permite crear programas cortos y funcionales. Por otra parte, Toomey (2018) destaca la importancia de las plataformas de trabajo diseñadas para el análisis de datos en tiempo real con Python y el beneficio añadido de mezclar código de otros lenguajes de programación para potenciar sus capacidades analíticas.

1.1.2. *Librerías en Python*

1.1.2.1. *Matplotlib*

Es una librería gráfica de python estándar y por ello es la más conocida a la hora de realizar ejercicios matemáticos de cualquier índole además de ser utilizada para realizar gráficos de calidad necesaria para publicarlos tanto en papel como digitalmente entre ellos se encuentran series temporales, histogramas, diagramas de barras, diagramas de errores, espectros de potencia, etc. (Zamorano , 2019).

1.1.2.2. NumPy

Nos proporciona una estructura de dato universal que agiliza el análisis de datos y el intercambio de datos entre distintos algoritmos su forma de trabajar es mediante vectores multidimensionales y matrices con capacidad para gran cantidad de datos, además de ser una de las librerías más poderosas en proporcionar funciones matemáticas de alto nivel que operan en estas estructuras de datos (Blanco , 2021).

1.1.2.3. SciPy

Al igual que NumPy nos proporciona rutinas numéricas eficientes fáciles de usar y opera en las mismas estructuras de datos que nos proporciona NumPy así como por ejemplo se puede realizar integración numérica, optimización, interpolación, transformadas de Fourier, algebra lineal, estadística, etc. (Zamorano , 2019).

1.1.3. Física computacional

Es una rama de la física que se centra en la elaboración de modelos por ordenador de sistemas con muchos grados de libertad para los cuales ya existe una teoría computacional. En general, se efectúan modelos microscópicos en los cuales las "partículas" obedecen a una dinámica simplificada, y se estudia el que puedan reproducirse las propiedades macroscópicas a partir de este modelo muy simple de las partes constituyentes. Las simulaciones se hacen resolviendo ecuaciones que gobiernan el sistema. Por lo general, son grandes sistemas de ecuaciones diferenciales ordinarias, ecuaciones diferenciales a derivadas parciales y ecuaciones diferenciales estocásticas, que no pueden ser resueltos explícitamente de manera analítica (Arias , Jiménez , & Porras , 2016)

Arce, *et al* (2021). Señalan que en Física la mayoría de los problemas que involucran la resolución de ecuaciones diferenciales (EDOs) aparecen en casi todas las asignaturas de la especialidad, las cuales en muchos casos no cuentan con soluciones matemáticas exactas y son restringidas a través de condiciones de contorno y problemas de valores iniciales. Por tanto, debemos enfocar el problema en cuestión desde otro punto de vista. Es ahí donde aparecen los metodos numéricos como una potente herramienta capaz de resolver las EDOs y EDPs, donde nos permite encontrar el valor aproximado de la función incógnita para un conjunto de valores de las variables independientes discretas. Que pueden ser implementados en diferentes lenguajes y técnicas de programación que un ordenador puede interpretar (Segundo, 2010).

Actualmente, la comunidad científica internacional cuenta con una variedad extensa de software especializado para el tratamiento y análisis estadístico de datos; entre los más representativos se encuentran SAS, SPSS, STATA, Statistica; Minitab, EViews, Statgraphics, entre otros. Todos incluyen herramientas sofisticadas de análisis de Interfaces Gráficas de Usuario (GUI, por sus siglas en inglés) que permite su uso en entornos con necesidades diversas, como ocurre en los ámbitos científico, industrial y académico (Acosta, Solís, & Contreras, 2019).

Este tipo de software especializado es licenciado por las empresas que los desarrollan y para su uso legal es necesario pagar derechos que ascienden desde cientos hasta decenas de miles de dólares (USD). Por consiguiente, con regularidad quedan fuera del alcance del usuario medio y, únicamente, algunas organizaciones, universidades y centros de investigación pueden sufragar los costos muy entusiastas que, regularmente, no obtienen retribución económica alguna (Acosta, Solís, & Contreras, 2019).

1.1.4. Modelos matemáticos

Es una representación simplificada, a través de ecuaciones, funciones o fórmulas matemáticas, de un fenómeno o de la relación entre dos o más variables. La rama de las matemáticas que se encarga de estudiar las cualidades y estructura de los modelos es la llamada “teoría de los modelos” (Aravena, Caamano, & Gimenez, 2008).

Recalde, & Henao (2018). Afirman que la modelación matemática de algunos problemas físicos se hizo con base en el naciente cálculo en términos de ecuaciones diferenciales ordinarias. A finales del siglo XVII se publicaron algunos trabajos en los cuales se exponían métodos para resolver ecuaciones diferenciales ordinarias, entre ellos separación de variables, solución de ecuaciones homogéneas por sustitución, solución por series y resolución de ecuaciones exactas. Aunque muchos de éstos no fueron generalizados sino hasta Euler, es posible encontrar casos particulares de estos métodos antes del siglo XVIII. Si bien a partir de estos problemas se encontraron diversas técnicas de solución, hubo un estancamiento en esta disciplina, pues, en su conjunto, todas estas técnicas resolvían casos particulares, pero no dieron lugar a algoritmos generales de resolución de las EDO. Como consecuencia de ello, en el siglo XVIII no se produjeron resultados teóricos, como el Teorema de Existencia y Unicidad.

1.1.5. Actualidad en la educación Matemática

Es muy importante tomar en cuenta que debido a la pandemia se han tenido cambios sociales que han acelerado desarrollo de la TICs de una manera vertiginosa lo cual lleva a buscar nuevas estrategias innovadoras dentro de pedagogía, así como la necesidad de implementar el uso de las

TICs en el ámbito docente (Santillán, 2017a) en este sentido Zetterqvist (2017) cuestiona el considerar la mera existencia de datos y tecnología computacional como eje central en el desarrollo de aprendizajes, es decir, el uso de paquetes especializados o lenguajes de programación per se no garantiza su logro. Sin embargo, no utilizarlos puede impedir el desarrollo de habilidades fundamentales en el ejercicio futuro de una profesión.

Por su parte Balanskat, Blamire y Kefala (2006), señalan que, “a pesar del incremento de la disponibilidad de recursos tecnológicos en las escuelas . . . la práctica pedagógica de los docentes en el aula no supone necesariamente una alteración sustantiva del modelo de enseñanza tradicional” (citado en Area, 2008). Reafirmando la idea que la educación no solo necesita la implementación de las TICs sino su correcta aplicación por parte de docentes y estudiantes como herramienta de aprendizaje (Santillán, 2017)

Almeida (2020), en su tesis doctoral afirma que el desarrollo de aplicaciones requiere de tiempo, estudio, y colaboración, principalmente en este tipo de aplicaciones, donde se requiere de conocimientos claros de varios temas de estudio. Así también afirma que de acuerdo con la encuesta que realizó el investigador las aplicaciones resultan de gran apoyo tanto para estudiantes como para docentes, de tipo didáctico que coadyuva en el aprendizaje del estudiante.

1.1.6. Ecuaciones diferenciales

Una ecuación que contiene derivadas de una o más variables respecto a una o más variables independientes, se dice que es una ecuación diferencial (ED). Una ecuación diferencial es una ecuación que involucra derivadas de una función desconocida de una o más variables. Si la función desconocida depende sólo de una variable (de tal modo que las derivadas son derivadas ordinarias) la ecuación se llama una ecuación diferencial ordinaria. Sin embargo, si la función desconocida depende de más de una variable (de tal modo que las derivadas son derivadas parciales) la ecuación se llama una ecuación diferencial parcial (Zamora Ramos , 2011).

Una ecuación diferencial ordinaria (EDO) es una ecuación que relaciona una variable independiente x , una función incógnita $y = y(x)$ y sus derivadas $y', y'', \dots, y^{(n)}$, es decir, una ecuación de la forma:

$$G(x, y, y', y'', \dots, y^{(n)}) = 0 \quad \text{(forma implícita) (1)}$$

ó

$$y^{(n)} = F(x, y, y', y'', \dots, y^{(n-1)}) \quad \text{(forma explícita) (2)}$$

Ejemplos 1, 2, 3: Ecuaciones diferenciales:

$$y' + 3xy = 0 \quad (3)$$

$$xy''' - 4y' + x^3y = \cos x \quad (4)$$

$$\frac{dy}{dx} + x^2y + x = 0 \quad (5)$$

La ecuación diferencial se llama *ordinaria* porque la función incógnita depende de una sola variable independiente.

Si la función incógnita depende de dos o más variables independientes, la ecuación diferencial contendría derivadas parciales por lo que a estas ecuaciones se las llama ecuaciones diferenciales en derivadas parciales.

El orden de una ecuación diferencial es de la derivada de mayor orden que figura en la ecuación.

Ejemplo 4 y 5: ecuaciones diferenciales:

$$y'' + 2y' = x \quad \text{EDO de segundo orden (6)}$$

$$4xy''' - 3y = \sin x \quad \text{EDO de tercer orden (7)}$$

Una solución de la ecuación diferencial (1) en un intervalo $I \subseteq \mathbf{R}$ es una función $\phi(x)$ que admite derivadas sucesivas hasta el orden n inclusive en I , tal que al sustituir y por $\phi(x)$ en la ecuación diferencial la convierte en una identidad, es decir

$$\phi(x) \text{ es solución de la ecuación (1)} \Leftrightarrow G(x, \phi(x), \phi'(x), \dots, \phi^{(n)}(x)) = 0 \forall x \in I$$

Ejemplo 6: resolución de ecuación diferencial:

la función

$$y = e^x(1+x) \quad (8)$$

es una solución de la ecuación diferencial

$$y'' - 2y' + y = 0 \quad (9)$$

En efecto, si derivamos dos veces la función dada tenemos:

$$y' = e^x(1+x) + e^x = e^x(2+x)$$

$$y'' = e^x(2 + x) + e^x = e^x(3 + x)$$

sustituyendo y , y' y y'' en la ecuación diferencial dada resulta la identidad

$$e^x(3+x) - 2e^x(2+x) + e^x(1+x) = e^x(3+x-4-2x+1+x) = 0$$

Resolver o integrar una ecuación diferencial ordinaria es hallar todas sus soluciones.

1.1.6.1. Ecuaciones Diferenciales Lineales de primer orden

Son ecuaciones diferenciales ordinarias de primer orden que pueden expresarse en la forma:

$$y' + p(x)y = q(x) \quad (10)$$

donde p y q son funciones continuas en un intervalo I .

Si $q(x)$ no es idénticamente nula en I , la ecuación (10) se llama *lineal no homogénea*.

Si $q(x) \equiv 0$ en I se dice que la ecuación (10) es *lineal homogénea* y resulta una ecuación a variables separables.

Una de las formas de resolver la ecuación (10) es mediante la sustitución

$$y(x) = u(x) \cdot v(x) \quad (11)$$

Para ello calculamos $y' = u' \cdot v + u \cdot v'$, reemplazamos y y y' en (10) y obtenemos

$$\begin{aligned} u' \cdot v + u \cdot v' + p \cdot u \cdot v &= q \\ (u' + p \cdot u)v + u \cdot v' &= q \end{aligned}$$

Hallamos una función u tal que $u' + p \cdot u = 0$ (esta es una ecuación diferencial a variables separables con función incógnita $u = u(x)$). Luego calculamos la función v resolviendo la ecuación $u \cdot v' = q$, ecuación diferencial a variables separables con función incógnita $v = v(x)$.

Finalmente $y(x) = u(x) \cdot v(x)$ es la solución general de la ecuación diferencial lineal no homogénea.

Ejemplo 7: resolución ecuación diferencial de primer orden: Resolver la siguiente ecuación diferencial:

$$xy' - y = x^3 \quad (12)$$

Esta ecuación puede escribirse como

$$y' - \frac{1}{x}y = x^2 \text{ con } x > 0 \quad (13)$$

Realizamos la sustitución $y = u \cdot v$ de donde $y' = u' \cdot v + u \cdot v'$ y reemplazando en la ecuación diferencial dada tenemos:

$$u' \cdot v + u \cdot v' - \frac{1}{x}u \cdot v = x^2 \rightarrow (u' - \frac{1}{x}u) v + uv' = x^2$$

Hallamos una función u tal que $u' - \frac{1}{x}u = 0$. Esta es una ecuación diferencial a variables separables, resolviéndola obtenemos $u = Cx$, $C \in \mathbf{R}$, como interesa una función u elegimos $C = 1$, entonces $u = x$, con esta función hallamos v de manera que $xv' = x^2$, se trata también de una ecuación diferencial a variables separables, cuya solución general es $v = \frac{1}{2}x^2 + C$ con $C \in \mathbf{R}$.

Luego la solución general de la ecuación diferencial dada es

$$y = u \cdot v = x \cdot (\frac{1}{2}x^2 + C) \rightarrow y = \frac{1}{2}x^3 + Cx \text{ con } C \in \mathbf{R} \quad (14)$$

1.1.6.2. Ecuaciones diferenciales ordinarias de segundo orden

Este tipo de ecuaciones cumplen con la propiedad de poder ser consideradas como operadores lineales, de aquí surge el concepto para poder encontrar sus soluciones (Zamora Ramos, 2011) Son de la forma

$$G(x, y, y', y'') = 0 \quad (\text{forma implícita}) \quad (15)$$

ó

$$y'' = F(x, y, y') \quad (\text{forma explícita}) \quad (16)$$

Un problema con valores iniciales o problema de Cauchy es de la forma

$$\Leftrightarrow PVI: \begin{cases} y'' = F(x, y, y') \\ y(x_0) = y_0, y'(x_0) = y_1 \end{cases} \quad (17)$$

La *solución general* de una ecuación diferencial de segundo orden es una función que depende de dos constantes arbitrarias y que satisface la ecuación para cualquier valor de dichas constantes, es decir, la solución general será

$$g(x, y, C_1, C_2) = 0 \text{ (forma implícita)} \quad (18)$$

ó

$$y = \phi(x, C_1, C_2) \text{ (forma explícita)} \quad (19)$$

La *solución particular* es la que se obtiene de la solución general por valores determinados a las constantes arbitrarias C_1 y C_2 .

Ejemplo 8: Resolución de ejercicios de segundo orden. Verificar que la función

$$y = C_1 + C_2 e^{2x} \quad (20)$$

es la solución general de la ecuación diferencial

$$y'' - 2y = 0 \quad (21)$$

y hallar la solución particular que satisface las condiciones iniciales $y(0) = 1$, $y'(0) = 2$.

Dentro de esta clasificación también encontramos a las ***Ecuaciones diferenciales lineales de segundo orden***, son ecuaciones diferenciales de la forma

$$a_0(x)y'' + a_1(x)y' + a_2(x)y = f(x) \quad (22)$$

donde $a_0(x)$, $a_1(x)$, $a_2(x)$ y $f(x)$ son funciones continuas en un intervalo I y $a_0(x)$ no es idénticamente nula en I .

Si $f(x) \equiv 0$ en I , la ecuación (22) se llama *lineal homogénea*. Si $f(x)$ no es idénticamente nula en I , se dice que la ecuación (22) es *lineal no homogénea*.

Si las funciones $a_0(x)$, $a_1(x)$, $a_2(x)$ son constantes la ecuación diferencial lineal es a *coeficientes constantes*, si son variables se tiene una ecuación diferencial a *coeficientes variables*.

Supongamos que $a_0(x) \neq 0 \forall x \in J \subseteq I \subseteq \mathbf{R}$ ó $I = J$, entonces dividimos la ecuación (22) por la función $a_0(x)$

$$y'' + \frac{a_1(x)}{a_0(x)} y' + \frac{a_2(x)}{a_0(x)} y = \frac{f(x)}{a_0(x)} \quad (23)$$

llamando

$$p_1(x) = \frac{a_1(x)}{a_0(x)}, p_2(x) = \frac{a_2(x)}{a_0(x)}, q(x) = \frac{f(x)}{a_0(x)}$$

se tiene

$$y'' + p_1(x)y' + p_2(x)y = q(x) \forall x \in J \quad (24)$$

donde p_1, p_2 y q son funciones continuas en el intervalo J .

La ecuación (24) se llama *ecuación lineal normalizada* y al intervalo J , *intervalo de normalidad*.

Dada la ecuación (24), se llama *ecuación lineal homogénea asociada* a (24) a la ecuación

$$y'' + p_1(x)y' + p_2(x)y = 0 \forall x \in J \quad (25)$$

Definición: Llamaremos *operador diferencial lineal* a:

$$L_2[y] = y'' + p_1(x)y' + p_2(x)y \quad (26)$$

Este operador posee las siguientes propiedades (propiedades de linealidad)

- i) $L_2[ky] = kL_2[y]$ k : constante
- ii) $L_2[y_1 + y_2] = L_2[y_1] + L_2[y_2]$, y_1, y_2 : funciones de x

Como consecuencia de estas propiedades resulta:

$$L_2[k_1y_1 + k_2y_2] = k_1L_2[y_1] + k_2L_2[y_2], y_1, y_2: \text{funciones de } x, k_1, k_2: \text{constantes}$$

Utilizando el operador diferencial lineal L_2 la ecuación diferencial lineal no homogénea puede escribirse $L_2[y] = q(x)$ y la ecuación diferencial lineal homogénea como $L_2[y] = 0$.

1.1.6.3. Lenguaje de programación

Un lenguaje de programación es un conjunto de símbolos y códigos usados para orientar la programación de estructuras en el desarrollo web.

1.2. Bases conceptuales

1.2.1. Solución General

Es una familia biparametrica de funciones $\phi(x,y,C1,C2)=0$ en donde si cada una de la familia satisface la ecuación y si dados unos valores y_0 e y_1 , existen C1 y C2 tales que la correspondiente función cumpla que $y(x_0) = y_0$ y que $y'(x_0) = y_1$ (Costa , Torroba , & Devece , 2013).

1.2.2. Solución Particular:

Vienen hacer cada una de las funciones que se obtienen de nuestra función general al dar ciertos valores a los parámetros (Costa , Torroba , & Devece , 2013).

1.2.3. Solución Singular:

Está compuesta por una solución que no puede extraerse de la solución general (Costa , Torroba , & Devece , 2013).

1.2.4. Matrices

En python las matrices no son una estructura propia de Python, simplemente es una lista de listas que nosotros interpretamos desde el punto de vista matemático así que se puede pasarlo a una cadena de caracteres para así poder representar una matriz (Santorum , 2021).

1.2.5. Determinantes

Se puede calcular el determinante de una matriz cuadrada de dos maneras. La primera es tratar de transformar la matriz en otra en la que solo si hay ceros debajo de la diagonal principal. La matriz transformada tiene el mismo determinante que la matriz original (salvo tal vez el signo). Otra posible solución, seria calcular menores de la matriz original y calcular el determinante a partir del cálculo de determinantes de matices más pequeñas (Santorum , 2021).

1.2.6. Técnicas de Reducción de Orden

La solución general de una ecuación diferencial lineal homogénea de segundo orden

$$a_2(x)y'' + a_1(x)y' + a_0(x)y = 0 \quad (27)$$

es una combinación lineal

$$y = c_1 y_1 + c_2 y_2,$$

donde y_1 y y_2 son soluciones que constituyen un conjunto linealmente independiente en cierto intervalo I . Al comienzo de la siguiente sección se analiza un método para determinar estas soluciones cuando los coeficientes de la ED en (27) son constantes. Este método, que es un ejercicio directo en álgebra, falla en algunos casos y sólo produce una solución simple y_1 de la ED. En estos casos se puede construir una segunda solución y_2 de una ecuación homogénea (27) (aun cuando los coeficientes en (27) son variables) siempre que se conozca una solución no trivial y_1 de la ED.

La idea básica que se describe en esta sección es que la ecuación (27) se puede reducir a una ED lineal de primer orden por medio de una sustitución en la que interviene la solución conocida y_1 . Una segunda solución y_2 de (27) es evidente después de resolver la ED de primer orden.

Suponga que y_1 denota una solución no trivial de (27) y que y_1 se define en un intervalo I . Se busca una segunda solución y_2 tal que y_1 y y_2 sean un conjunto linealmente independiente en I . Recuerde qué si y_1 y y_2 son linealmente independientes, entonces su cociente y_2/y_1 no es constante en I , es decir,

$$y_2(x)/y_1(x) = u(x) \quad (28)$$

o

$$y_2(x) = u(x)y_1(x) \quad (29)$$

La función $u(x)$ se determina al sustituir

$$y_2(x) = u(x)y_1(x) \quad (30)$$

en la ecuación diferencial dada.

Este método se llama reducción de orden porque debemos resolver una ecuación diferencial lineal de primer orden para encontrar a u .

Ejemplo 9. Reducción de orden

Dado que

$$y_1 = e^x \quad (31)$$

es una solución de

$$y'' - y = 0 \text{ en el intervalo } (-\infty, \infty), \quad (32)$$

use reducción de orden para determinar una segunda solución y_2 .

SOLUCIÓN

Si

$$y = u(x)y_1(x) = u(x)e^x,$$

entonces aplicando la regla del producto se obtiene

$$\begin{aligned}y' &= ue^x + e^x u', \\y'' &= ue^x + 2e^x u' + e^x u'',\end{aligned}$$

por tanto

$$y'' - y = e^x(u'' + 2u') = 0$$

Puesto que $e^x \neq 0$, la última ecuación requiere que $(u'' + 2u') = 0$.

Si se hace la sustitución $w = u'$, esta ecuación lineal de segundo orden en u se convierte en

$$w' + 2w = 0$$

que es una ecuación lineal de primer orden en w .

Si se usa el factor integrante e^{2x} , se puede escribir

$$d/dx[e^{2x}w] = 0$$

Después de integrar, se obtiene

$$w = c_1 e^{-2x} \text{ o } u' = c_1 e^{-2x}$$

Al integrar de nuevo se obtiene

$$u = -\frac{1}{2} c_1 e^{-2x} + c_2.$$

$$y = u(x)e^x = -\frac{c_1}{2} e^{-x} + c_2 e^x.$$

Haciendo $c_2 = 0$ y $c_1 = -2$, se obtiene la segunda solución deseada,

$$y_2 = e^{-x}.$$

Puesto que $W(e^x, e^{-x}) \neq 0$ para toda x , las soluciones son linealmente independientes en $(-\infty, \infty)$. Puesto que se ha demostrado que $y_1 = e^x$, $y_2 = e^{-x}$ son soluciones linealmente independientes de una ecuación lineal de segundo orden, la expresión en (2) es en realidad la solución general de $y'' - y = 0$ en $(-\infty, \infty)$.

1.2.7. Caso General

Suponga que se divide entre $a_2(x)$ para escribir la ecuación (27) en la forma estándar

$$y'' + P(x)y' + Q(x)y = 0, \quad (33)$$

donde $P(x)$ y $Q(x)$ son continuas en algún intervalo I . Supongamos además que $y_1(x)$ es una solución conocida de (33) en I y que $y_1(x) \neq 0$ para toda x en el intervalo. Si se define $y = u(x)y_1(x)$, se tiene que

$$\begin{aligned} y' &= uy_1' + y_1u', & y'' &= uy_1'' + 2y_1'u' + y_1u'' \\ y'' + Py' + Qy &= u \underbrace{[y_1'' + Py_1' + Qy_1]}_{\text{cero}} + y_1u'' + (2y_1' + Py_1)u' = 0. \end{aligned} \quad (34)$$

Esto implica que se debe tener

$$y_1u'' + (2y_1' + Py_1)u' = 0 \quad \text{o} \quad y_1w' + (2y_1' + Py_1)w = 0,$$

donde hacemos que $w = u'$. Observe que la última ecuación en (34) es tanto lineal como separable. Separando las variables e integrando, se obtiene

$$\begin{aligned} \frac{dw}{w} + 2 \frac{y_1'}{y_1} dx + P dx &= 0 \\ \ln|wy_1^2| &= - \int P dx + c \quad \text{o} \quad wy_1^2 = c_1 e^{-\int P dx}. \end{aligned}$$

Despejamos a w de la última ecuación, usamos $w = u'$ e integrando nuevamente:

$$u = c_1 \int \frac{e^{-\int P dx}}{y_1^2} dx + c_2.$$

Elijiendo $c_1=1$ y $c_2=0$, se encuentra de $y = u(x)y_1(x)$ que una segunda solución de la ecuación (3) es

$$(35) \quad y_2 = y_1(x) \int \frac{e^{-\int P(x) dx}}{y_1^2(x)} dx.$$

Un buen ejercicio de derivación es comprobar que la función $y_2(x)$ que se define en (35) satisface la ecuación (33) y que y_1 y y_2 son linealmente independientes en algún intervalo en el que $y_1(x)$ no es cero.

CAPÍTULO II

2. MARCO METODOLÓGICO

2.1. Tipo de Investigación

Este trabajo de investigación de tipo experimental corresponde al método mixto, con una connotación teórica aplicada que buscará mediante la toma de datos corroborar los datos teóricamente y corroborarlos en python para así demostrar la eficiencia del programa y su aplicabilidad a la física experimental como una herramienta de apoyo hacia los estudiantes. Exploratorio al tratarse de un sistema de apoyo hacia los estudiantes de la institución, así también desde la perspectiva descriptiva al medir y definir variables para el análisis de resultados.

2.2. Diseño de la Investigación

2.2.1. *Diseño Experimental / Cuasi Experimental / Pre Experimental*

Diseño Experimental: Este proyecto es experimental ya que mediante el empleo de Python 3.10.0 se comprobarán y se cuantificara la solución a ecuaciones diferenciales de segundo orden de física demostrando así su eficiencia, rentabilidad y proyectada al estudio de la Física con una visión practica y experimental.

2.2.1.1. *Identificaciones variables*

Variables dependientes

- Modelos matemáticos
- Ecuaciones diferenciales

Variables independientes

- Programa Python

2.2.1.2. *Planteamiento de la hipótesis*

Existe la posibilidad de que se desconozca la facilidad con la que el programa Python trabaja para la resolución de ejercicios de ecuaciones diferenciales de segundo orden, lo que conlleva a que se realice este estudio para brindar una ayuda a los estudiantes de física y así mejorar su aprendizaje.

2.2.1.3. *Matriz de consistencia*

Tabla 1-1: Matriz de consistencia

PROBLEMA	OBJETIVOS	HIPÓTESIS	VARIABLES
<p>Problema general: ¿Cuál es el nivel de conocimiento de los estudiantes de física en la utilización del programa Python?</p> <p>Problemas específicos: No se dispone de una guía que facilite la resolución de los ejercicios de física en Python.</p>	<p>Objetivo general: Desarrollo de ecuaciones diferenciales de segundo orden para el aprendizaje de física computacional en Python 3.10.0.</p> <p>Objetivos específicos: Elaborar las guías de uso de estas ecuaciones para su óptimo empleo en la institución y dejarlas disponibles a los estudiantes de la carrera.</p>	<p>Hipótesis general: Existe la posibilidad de que se desconozca la facilidad con la que el programa Python trabaja para la resolución de ejercicios de ecuaciones diferenciales de segundo orden</p> <p>Hipótesis específicas: Realizar este estudio para brindar una ayuda a los estudiantes de física y así mejorar su aprendizaje.</p>	<p>Variable independiente: Programa Python</p> <p>Variable dependiente: Modelos Matemáticos Ecuaciones Diferenciales.</p>

Realizado por: Logroño, D., 2022.

2.2.1.4. Operacionalización de las variables

Tabla 1-2: Operacionalización de las variables

VARIABLE	CONCEPTO	INDICADOR	INSTRUMENTO
Variable independiente:			
Programa Python	Es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código.	Función, código.	Estudio funcionamiento
Variable dependiente:			
Modelos Matemáticos	Es una representación simplificada, a través de ecuaciones, funciones o fórmulas matemáticas	Con sus expresiones se puede leer y transcribir los datos.	Números
Ecuaciones Diferenciales	Es una ecuación que involucra derivadas de una función desconocida de una o más variables.	Variables	Números

Realizado por: Logroño, D., 2022.

2.2.1.5. Localización del Estudio

El proyecto se encuentra localizado en la ciudad de Riobamba en la ESCUELA SUPERIOR POLITECNICA DE CHIMBORAZO.

2.2.1.6. Población de estudio

Para el desarrollo de la presente investigación será tomado en cuenta como población de estudio los estudiantes de la carrera de Física de la ESCUELA SUPERIOR POLITECNICA DE CHIMBORAZO

2.2.1.7. *Tamaño de la muestra*

Para la recolección de datos tomaremos en cuenta a los estudiantes del tercer semestre de física de la ESCUELA SUPERIOR POLITECNICA DE CHIMBORAZO.

2.2.1.8. *Método de muestreo*

Existen varios métodos de muestreo, entre los principales se encuentra los probabilísticos el cual será nuestras técnicas de recolección de datos.

Se utilizará libros de matemáticas y física que contengan ejercicios resueltos de ecuaciones diferenciales de segundo orden. Juntamente con el programa de Python.

2.2.1.9. *Análisis Estadístico Inferencial*

En el siguiente trabajo se utilizará estadística de tipo descriptiva con análisis de parámetros basados en los datos de la muestra.

2.3. **Técnicas de investigación**

2.3.1. *Método de Runge-Kutta para Ecuaciones Diferenciales*

El método de Runge-Kutta de orden cuatro (RK4) es uno de los métodos más utilizados en la práctica. Su deducción sigue las ideas de los métodos de Runge-Kutta orden dos y como los anteriores métodos, es un método recursivo y su cálculo no implica utilizar información adicional como ocurre en los métodos de Taylor (Arévalo Ovalle, Bernal Yermanos, & Posada Restrepo, 2021).

Uno de los métodos más utilizados para resolver numéricamente problemas de ecuaciones diferenciales ordinarias con condiciones iniciales es el método de Runge- Kutta de cuarto orden, el cual proporciona un pequeño margen de error con respecto a la solución real del problema y es fácilmente programable en un software para realizar las iteraciones necesarias. El método de Runge-Kutta se utiliza para resolver ecuaciones diferenciales de la forma:

$$\frac{dy}{dt} = f(t, y), y(t_0) = y_0 \quad (36)$$

Y es sumamente útil para casos en los que la solución no puede hallarse por los métodos convencionales (como separación de variables). Hay variaciones en el método de Runge-Kutta

de cuarto orden, pero el más utilizado es el método en el cual se elige un tamaño de paso h y un número máximo de iteraciones n tal que

$$\begin{aligned}
 y(t_0) &= y_0 \\
 k_1 &= h \cdot f(t_i, y_i) \\
 k_2 &= h \cdot f\left(t_i + \frac{h}{2}, y_i + \frac{k_1}{2}\right) \\
 k_3 &= h \cdot f\left(t_i + \frac{h}{2}, y_i + \frac{k_2}{2}\right) \\
 k_4 &= h \cdot f(t_i + h, y_i + k_3)
 \end{aligned}$$

Y se realiza la iteración

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

Para $i = 0, \dots, n - 1$. La solución se da a lo largo del intervalo $(t_0, t_0 + h_n)$

El algoritmo para el método de Runge-Kutta de cuarto orden en pseudo código es:

INICIO

INPUT: Número de iteraciones n (o tamaño de paso h), punto inicial del intervalo a (punto final del intervalo b), condición inicial $y(t_0) = y_0$. el siguiente:

$$n = (b - a)/h$$

$$t = t_0$$

$$y = y_0$$

OUTPUT (t, y)

PARA $i=1, \dots, n$

$$k_1 = h \cdot f(t_i, y_i)$$

$$k_2 = h \cdot f\left(t + \frac{h}{2}, y + \frac{k_1}{2}\right)$$

$$k_3 = h \cdot f\left(t + \frac{h}{2}, y + \frac{k_2}{2}\right)$$

$$k_4 = h \cdot f(t + h, y + k_3)$$

$$y = y + (k_1 + 2k_2 + 2k_3 + k_4)$$

$$t = t + h$$

OUTPUT (t, y)

FIN PARA

FIN

2.3.2. Pasos para realizar los ejercicios

1. Ingresar a la interfaz gráfica Anaconda de Python

Para poder utilizar este algoritmo con precisión aprovechamos la potencia que nos ofrece Python, y para utilizar con mayor facilidad dicho software utilizamos la interfaz anaconda. En la figura 1 encontraremos la interfaz gráfica.

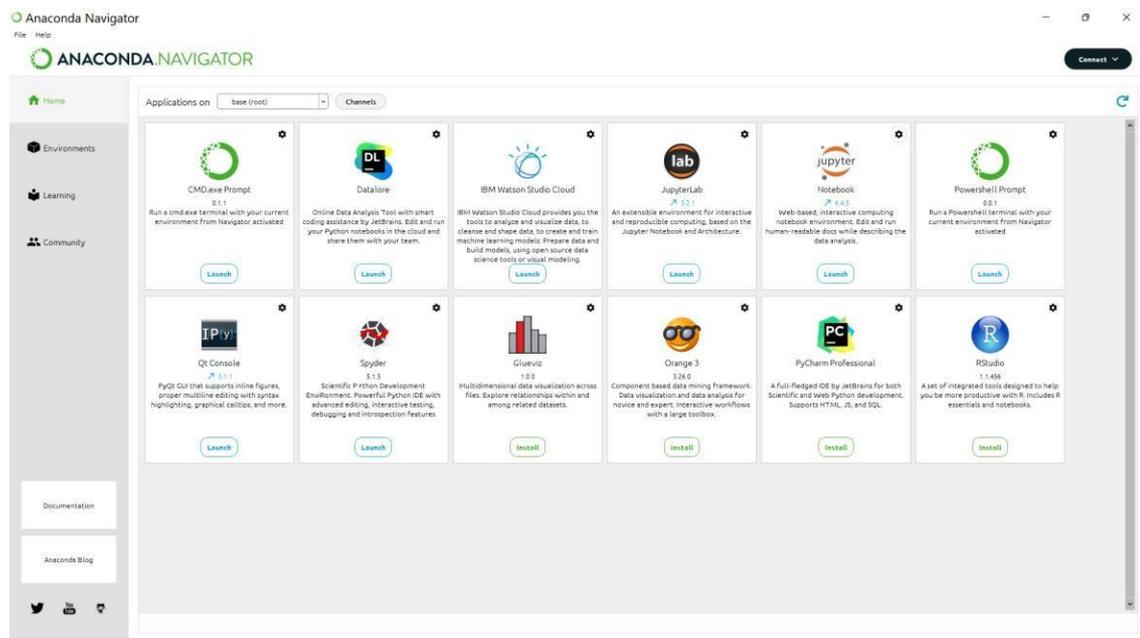


Figura 2-1: Interfaz gráfica Anaconda.

Realizado por: Logroño, D., 2022.

2. Seleccionar Jupyter

Al ingresar a la interfaz gráfica de anaconda, seleccionamos Jupyter, al estar en Jupyter creamos un nuevo archivo (Figura 2).

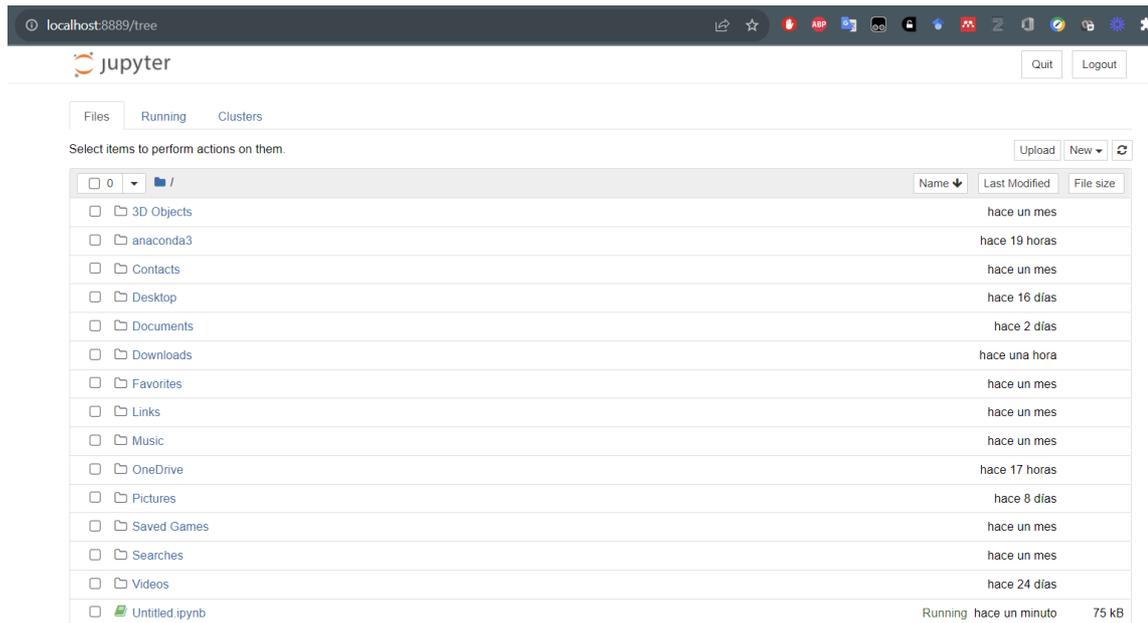


Figura 2-2: Interfaz gráfica Jupyter.
Realizado por: Logroño, D., 2022.

1. Llamamos a las librerías “numpy” y “matplotlib”

```
# Esta implementación en python utiliza numpy y matplotlib para calcular y representar
# gráficamente las funciones requeridas.
import numpy as np
import matplotlib.pyplot as plt
```

Figura 2-3: Interfaz gráfica Jupyter.
Realizado por: Logroño, D., 2022.

2. Se definen las funciones a utilizar
3. Se realiza los cálculos aplicando el Método de Runge-Kutta
4. Se grafica la solución.

CAPÍTULO III

3. MARCO DE ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

3.1. ANÁLISIS DE DATOS

3.1.1. Ejemplo 1

Resolver la siguiente ecuación:

$$\frac{d^2x}{dt^2} + p(t) \frac{dx}{dt} + q(t)x = F(t) \quad (37)$$

```
# Esta implementación en python utiliza numpy y matplotlib para calcular y representar
# gráficamente las funciones requeridas.
import numpy as np
import matplotlib.pyplot as plt
```

Figura 3-1: Llamado a librerías “numpy” y “matplotlib”

Realizado por: Logroño, D., 2022.

```
# Definimos las funciones F, p y q.
def F(t):
    F0 = 0.1
    w0 = 0.75
    F = F0 * np.cos(w0*t)
    return F

def p(t):
    p = 0.4
    return p

def q(t):
    q = 0.25
    return q

# Para aprovechar la potencialidad de python utilizaremos un único vector cuyas componentes
# serán la posición y la velocidad. En este caso lo llamamos x.
# Definimos las derivadas de la posición x[0] y de la velocidad x[1]

def dxdt(t,x):
    deriv = np.array([0.,0.])
    deriv[0] = x[1]
    deriv[1] = F(t) - p(t) * x[1] - q(t) * x[0] # La derivada de la velocidad
                                                # se puede obtener a partir de la ecuación diferencial

    return deriv

# Definimos una función que mediante el algoritmo de Runge-Kutta de 4to orden nos devuelve
# la posición y la velocidad en t+h. Se le debe proporcionar x(t) y las derivadas en t.
def rk4(deriv,h,x,t):
    hh = h/2.
    h6 = h/6.
    th = t+hh
    xt = x+hh*deriv
    dxt = dxdt(th,xt)
    xt = x+hh*dxt
    dxm = dxdt(th,xt)
    xt = x+h*dxm
    dxm = dxt+dxm
    dxt = dxdt(t+h,xt)
    conf_th = x+h6*(deriv+dxt+2.*dxm)
    return conf_th
```

Figura 3-2: Definición de funciones

Realizado por: Logroño, D., 2022.

```

# Este es el cuerpo de cálculo
h=0.1 # paso temporal
time=np.zeros(1) # iniciamos un vector para almacenar la coordenada t
x=np.array([0.,0.5]) # condiciones iniciales
traj = x # traj almacenará la posición y la velocidad en función del tiempo
for i in range(1000):
    t = i*h
    time = np.hstack([time,t])
    deriv = dxdt(t,x) # calculamos las derivadas en tiempo t
    x_new = rk4(deriv,h,x,t) # Las nuevas posición y velocidad calculadas mediante RK4
    traj = np.vstack([traj,x_new]) # almacenamos posición y velocidad
    x = x_new # Las nuevas condiciones iniciales para el paso posterior
    # serán las obtenidas luego de la evolución

```

Figura 3-3: Cálculos

Realizado por: Logroño, D., 2022.

```

#Graficamos con matplotlib.pyplot
font = {'family' : 'sans',
        'size' : 18}
plt.rc('font', **font)
plt.figure(figsize=(12,6))
plt.title("posición y velocidad en función del tiempo")
plt.xlabel("tiempo")
plt.plot(time,traj[:,0],'r-', label='posición') #para la trayectoria
plt.plot(time,traj[:,1],'b-', label='velocidad') #para la velocidad
plt.legend(loc='upper right');

```

Figura 3-4: Graficación

Realizado por: Logroño, D., 2022.

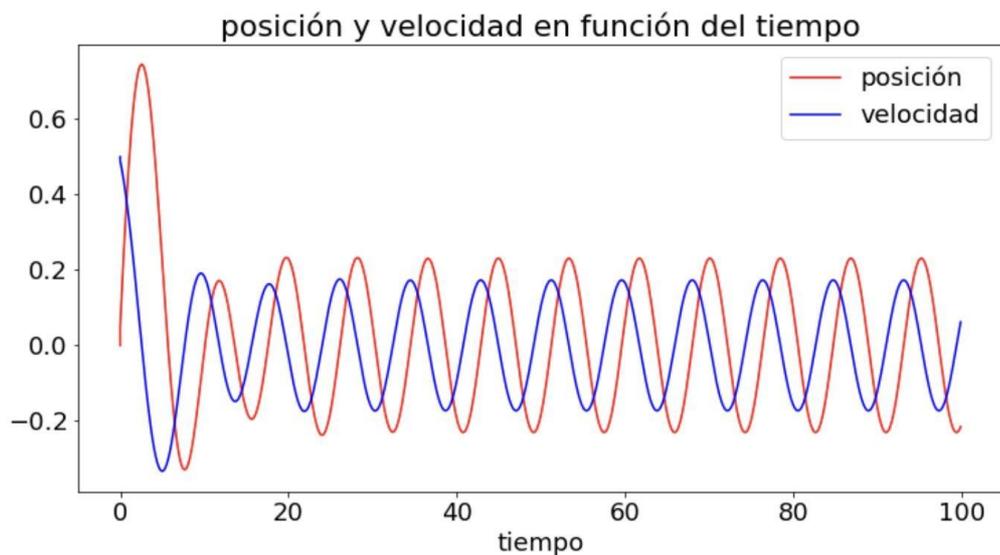


Gráfico 3-1: Resultados

Realizado por: Logroño, D., 2022.

A continuación, se puede observar los códigos que fueron utilizados:

```
# Esta implementación en python utiliza numpy y matplotlib para calcular y representar
# gráficamente las funciones requeridas.
import numpy as np
import matplotlib.pyplot as plt
# Definimos las funciones F, p y q.
def F(t):
    F0 = 0.1
    w0 = 0.75
    F = F0 * np.cos(w0*t)
    return F
def p(t):
    p = 0.4
    return p
def q(t):
    q = 0.25
    return q
# Para aprovechar la potencialidad de python utilizaremos un único vector cuyas componentes
# serán la posición y la velocidad. En este caso lo llamamos x.
# Definimos las derivadas de la posición x[0] y de la velocidad x[1]
def dxdt(t,x):
    deriv = np.array([0.,0.])
    deriv[0] = x[1]
    deriv[1] = F(t) - p(t) * x[1] - q(t) * x[0] # la derivada de la velocidad
    # se puede obtener a partir de la ecuación diferencial
    return deriv
# Definimos una función que mediante el algoritmo de Runge-Kutta de 4to orden nos devuelve
# la posición y la velocidad en t+h. Se le debe proporcionar x(t) y las derivadas en t.
def rk4(deriv,h,x,t):
    hh = h/2.
    h6 = h/6.
    th = t+hh
```

```

xt = x+hh*deriv
dxt = dxdt(th,xt)
xt = x+hh*dxt
dxm = dxdt(th,xt)
xt = x+h*dxm
dxm = dxt+dxm
dxt = dxdt(t+h,xt)
conf_th = x+h6*(deriv+dxt+2.*dxm)
return conf_th
# Este es el cuerpo de cálculo

h=0.1 # paso temporal
time=np.zeros(1) # iniciamos un vector para almacenar la coordenada t
x=np.array([0.,0.5]) # condiciones iniciales
traj = x # traj almacenará la posición y la velocidad en función del tiempo
for i in range(1000):
    t = i*h
    time = np.hstack([time,t])
    deriv = dxdt(t,x) # calculamos las derivadas en tiempo t
    x_new = rk4(deriv,h,x,t) # las nuevas posición y velocidad calculadas mediante RK4
    traj = np.vstack([traj,x_new]) # almacenamos posición y velocidad
    x = x_new # las nuevas condiciones iniciales para el paso posterior
    # serán las obtenidas luego de la evolución
#Graficamos con matplotlib.pyplot
font = {'family' : 'sans', 'size' : 18}
plt.rc('font', **font)
plt.figure(figsize=(12,6))
plt.title("posición y velocidad en función del tiempo")
plt.xlabel("tiempo")
plt.plot(time,traj[:,0], 'r-', label='posición') #para la trayectoria
plt.plot(time,traj[:,1], 'b-', label='velocidad') #para la velocidad
plt.legend(loc='upper right');

```

3.1.2. Ejemplo 2

Resolver la siguiente ecuación diferencial:

$$y'' + y = 0$$

```
import numpy as np
from scipy.integrate import odeint
# Ecuación diferencial de segundo orden
def diff_equation(y_list,x):
    #y'' + y = 0 Para el segundo orden, debe reemplazarse con dos ecuaciones diferenciales de primer orden
    # Sety '= z
    # Entonces z '= y' '= - y
    # La solución manual también es fácil de saber es y = C1sin (x) + C2cos (x)
    y,z=y_list
    return np.array([z,-y])
x=np.linspace(0,np.pi*2,num=100)
y0=[1,1]#y(0)=1,y'(0)=1
result=odeint(diff_equation,y0,x)
plt.plot(x,result[:,0],'r-', label = 'y') # La imagen de y, y = cos (x) + sin (x)
plt.plot(x,result[:,1],'b-', label = 'z') # La imagen de z, que es La imagen de y ', z = -sin (x) + cos (x)
plt.legend()
plt.grid()
plt.show()
```

Figura 3-5: Ejecución

Realizado por: Logroño, D., 2022.

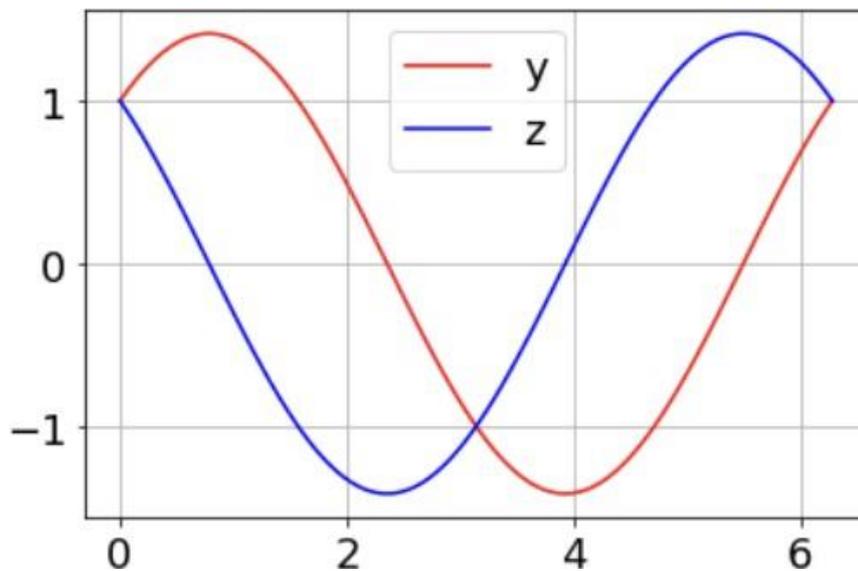


Gráfico 3-2: Resultados

Realizado por: Logroño, D., 2022.

A continuación, se puede observar los códigos que fueron utilizados.

```
import numpy as np
import matplotlib.pyplot as plt
```

```

from scipy.integrate import odeint
# Ecuación diferencial de segundo orden
def diff_equation(y_list,x):
#y'' + y = 0 Para el segundo orden, debe reemplazarse con dos ecuaciones diferenciales de primer
orden
# Sety '= z
# Entonces z '= y' '= - y
# La solución manual también es fácil de saber es y = C1sin (x) + C2cos (x)
y,z=y_list
return np.array([z,-y])
x=np.linspace(0,np.pi*2,num=100)
y0=[1,1]#y(0)=1,y'(0)=1
result=odeint(diff_equation,y0,x)
plt.plot(x,result[:,0],'r-', label = 'y') # La imagen de y, y = cos (x) + sin (x)
plt.plot(x,result[:,1],'b-', label = 'z') # La imagen de z, que es la imagen de y ', z = -sin (x) + cos
(x)
plt.legend()
plt.grid()
plt.show()

```

3.1.3. Ejemplo 3

Resuelva la siguiente ecuación diferencial de segundo orden para el ángulo theta de un péndulo sobre el que actúa la gravedad con fricción:

$$\theta''(t) + b*\theta'(t) + c*\sin(\theta(t)) = 0$$

donde b y c son constantes positivas y un número primo (') denota una derivada. Para resolver esta ecuación con odeint, primero debemos convertirla a un sistema de ecuaciones de primer orden. Al definir la velocidad angular , obtenemos el sistema: **$\omega(t) = \theta'(t)$**

$$\begin{aligned} \theta'(t) &= \omega(t) \\ \omega'(t) &= -b*\omega(t) - c*\sin(\theta(t)) \end{aligned}$$

Sea y el vector [theta , omega]. Implementamos este sistema en Python:

```

▶ import numpy as np
from scipy.integrate import odeint
def pend(y, t, b, c):
    theta, omega = y
    dydt = [omega, -b*omega - c*np.sin(theta)]
    return dydt
b = 0.25
c = 5.0
y0 = [np.pi - 0.1, 0.0]
t = np.linspace(0, 10, 101)
from scipy.integrate import odeint
sol = odeint(pend, y0, t, args=(b, c))
import matplotlib.pyplot as plt
plt.plot(t, sol[:, 0], 'b', label='theta(t)')
plt.plot(t, sol[:, 1], 'g', label='omega(t)')
plt.legend(loc='best')
plt.xlabel('t')
plt.grid()
plt.show()

```

Figura 3-6: Ejecución

Realizado por: Logroño, D., 2022.

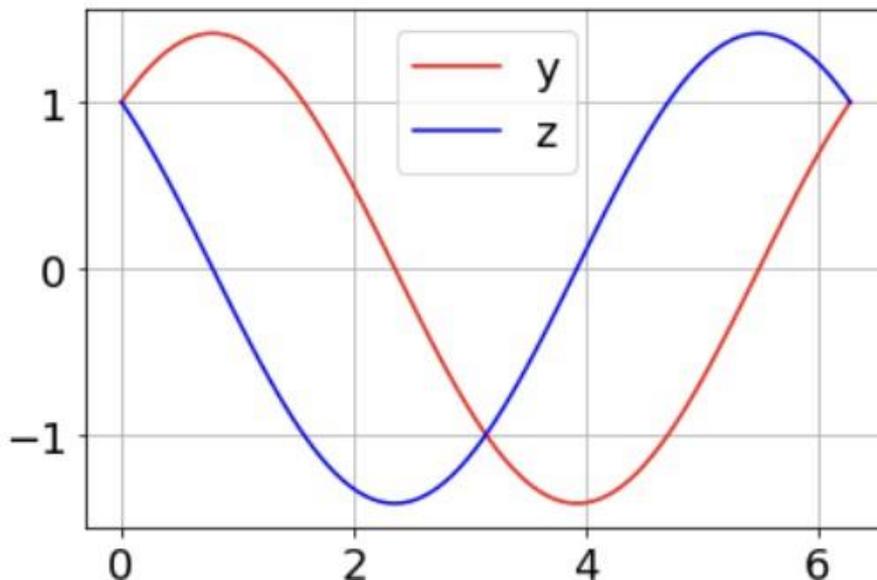


Gráfico 3-3: Resultados

Realizado por: Logroño, D., 2022.

A continuación, se puede observar los códigos que fueron utilizados.

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint
def pend(y, t, b, c):
    theta, omega = y
    dydt = [omega, -b*omega - c*np.sin(theta)]
    return dydt
b = 0.25
c = 5.0
y0 = [np.pi - 0.1, 0.0]
>>> t = np.linspace(0, 10, 101)
>>> from scipy.integrate import odeint
>>> sol = odeint(pend, y0, t, args=(b, c))
>>> import matplotlib.pyplot as plt
>>> plt.plot(t, sol[:, 0], 'b', label='theta(t)')
>>> plt.plot(t, sol[:, 1], 'g', label='omega(t)')
>>> plt.legend(loc='best')
>>> plt.xlabel('t')
>>> plt.grid()
>>> plt.show()
```

3.1.4 Ejemplo 4

Aplicación de balance de fuerzas en estado no estacionario.

Un objeto de masa M se fija al extremo inferior de un sistema amortiguador – resorte. El extremo superior del resorte se fija a una estructura en reposo, como se ilustra en la figura. El amortiguador ejerce una fuerza de $F_B = -B \frac{dy}{dt}$ sobre la caja y el resorte otra fuerza de $F_K = -Ky$ donde B y

K son las constante de amortiguación y de resorte respectivamente. Aplicando la Segunda Ley de Newton, se obtiene la siguiente ecuación de segundo orden que describe el movimiento del sistema:

$$M \frac{d^2y}{dt^2} + B \frac{dy}{dt} + Ky = 0$$

$$y(0) = C y'(0) = 0$$

Resuelve la ecuación diferencial de segundo orden, cuando:

- $M=0.5\text{Kg}$ $B=10\text{Ns/m}$ $K=100\text{N/m}$ y $C=1$
- $M=10\text{Kg}$ $B=2.5\text{Ns/m}$ $K=8.6\text{N/m}$ y $C=20$

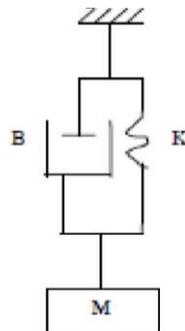


Figura 3. Ilustración del sistema masa – resorte – amortiguador.

Procedimiento: Convierta la ecuación diferencial de segundo orden en un sistema de ecuaciones diferencial de primer orden

$$y' = z$$
$$z' = -\frac{B}{M} z - \frac{K}{M} y$$

Condiciones límites: $y(0) = C$ $z(0) = 0$

a, El siguiente programa resuelve el sistema de ecuaciones diferenciales de primer orden del sistema masa – resorte – amortiguador

Codigo del programa:

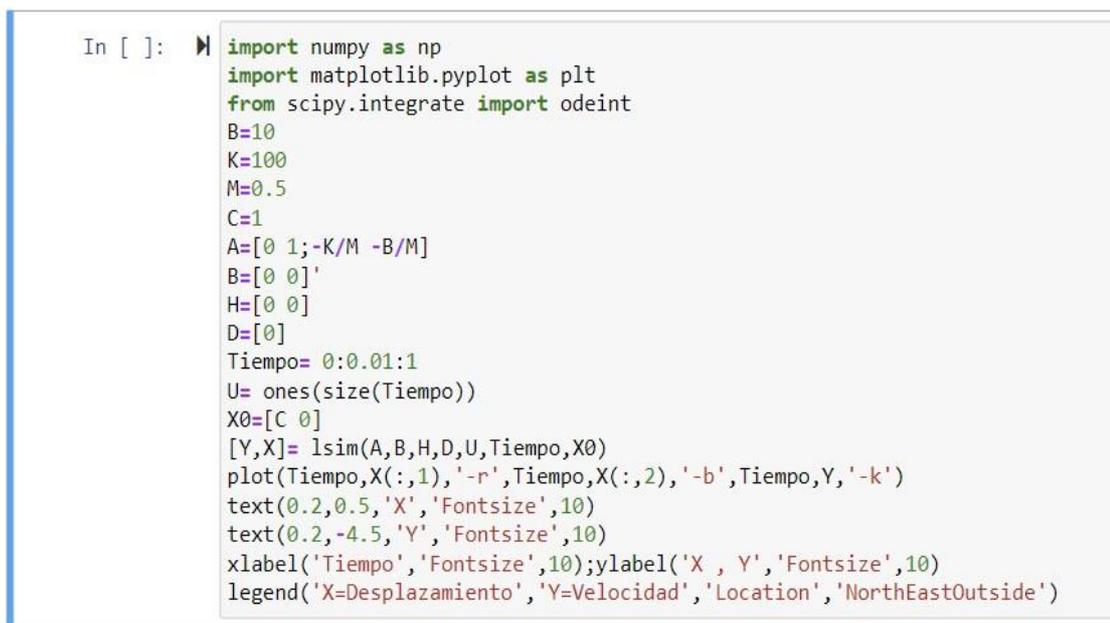
```
import numpy as np
```

```
import matplotlib.pyplot as plt
from scipy.integrate import odeint
```

```

B=10
K=100
M=0.5
C=1
A=[0 1;-K/M -B/M]
B=[0 0]'
H=[0 0]
D=[0]
Tiempo= 0:0.01:1
U= ones(size(Tiempo))
X0=[C 0]
[Y,X]= lsim(A,B,H,D,U,Tiempo,X0)
plot(Tiempo,X(:,1),'-r',Tiempo,X(:,2),'-b',Tiempo,Y,'-k')
text(0.2,0.5,'X','FontSize',10)
text(0.2,-4.5,'Y','FontSize',10)
xlabel('Tiempo','FontSize',10);ylabel('X , Y','FontSize',10)
legend('X=Desplazamiento','Y=Velocidad','Location','NorthEastOutside')

```



```

In [ ]: ▶ import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint
B=10
K=100
M=0.5
C=1
A=[0 1;-K/M -B/M]
B=[0 0]'
H=[0 0]
D=[0]
Tiempo= 0:0.01:1
U= ones(size(Tiempo))
X0=[C 0]
[Y,X]= lsim(A,B,H,D,U,Tiempo,X0)
plot(Tiempo,X(:,1),'-r',Tiempo,X(:,2),'-b',Tiempo,Y,'-k')
text(0.2,0.5,'X','FontSize',10)
text(0.2,-4.5,'Y','FontSize',10)
xlabel('Tiempo','FontSize',10);ylabel('X , Y','FontSize',10)
legend('X=Desplazamiento','Y=Velocidad','Location','NorthEastOutside')

```

Figura 3-7: Ejecución

Realizado por: Logroño, D., 2022.

Resultado de a:

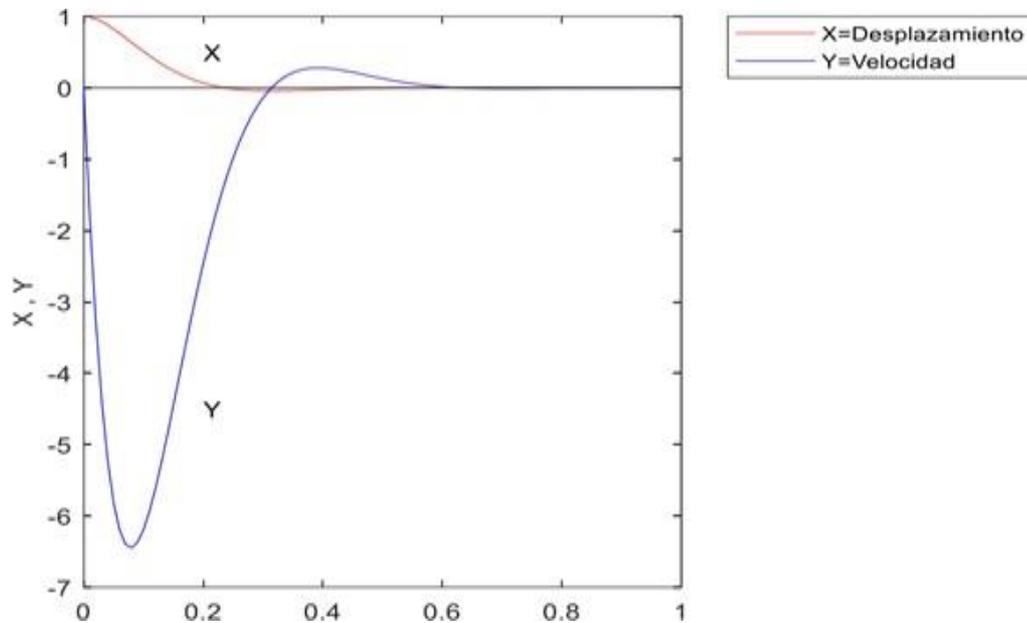


Gráfico 3-4: Solución gráfica de la ecuación diferencial $MY'' + BY' + KY = 0$ (parte a)

Realizado por: Logroño, D., 2022.

b) El siguiente programa resuelve el sistema de E.D.O masa-resorte – amortiguador.

Código del programa:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint
B=2.5
K=8.6
M=10
A=[0 1;-K/M -B/M]
B=[0 0]'
H=[0 0]
D=[0]
Tiempo=(0:0.1:5)'
U= ones(size(Tiempo))
X0=[20 0]
[Y,X]= lsim(A,B,H,D,U,Tiempo,X0)
```

```

plot(Tiempo,X(:,1),'-d',Tiempo,X(:,2),'-o',Tiempo,Y,'-')
disp([Tiempo X])
text(1.5,10,'X','FontSize',15)
text(1,-10,'Y','FontSize',15)
xlabel('Tiempo','FontSize',15)
ylabel('X y Y','FontSize',15)
legend('X=Desplazamiento','Y=Velocidad')

```

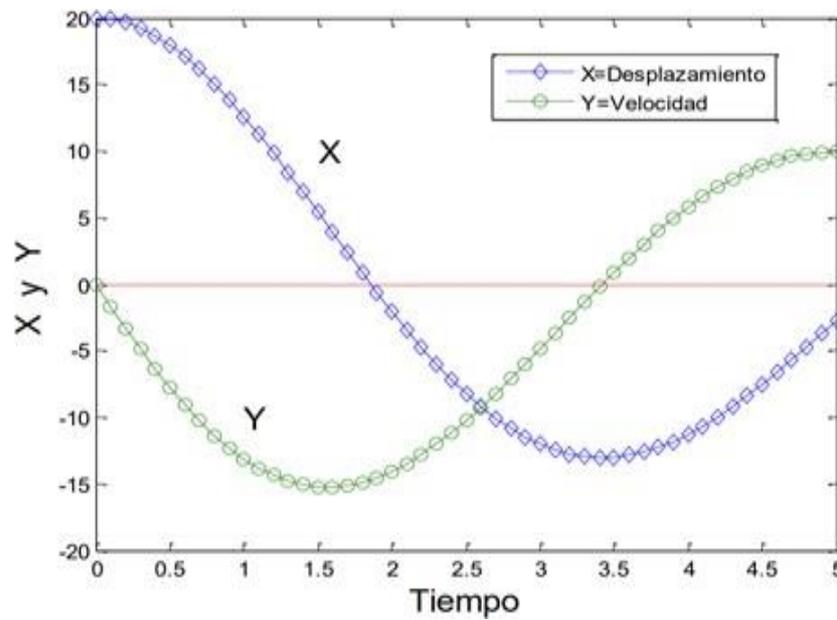


Gráfico 3-5: Solución gráfica de la ecuación diferencial $MY'' + BY' + KY = 0$ (parte b)

Realizado por: Logroño, D., 2022.

CONCLUSIONES

- ✓ Se desarrolló ejercicios de ecuaciones diferenciales de segundo orden utilizando Python es una herramienta de física computacional muy poderosa mediante sus librerías “numpy” y “matplotlib” ya que “mumpy” realiza EDO de primer orden la cual permitió realizar los cálculos de las EDO al aplicar el método de Runge- Kutta, y “matplotlib” permite graficar de forma idónea los resultados obtenidos. lo cual permitirá a los estudiantes de la carrera de Física en su trayectoria académica resolver ejercicios en los que interfieren ecuaciones diferenciales de una manera idónea y rápida.
- ✓ En el capítulo 2 se muestran cómo desarrollar paso a paso las ecuaciones diferenciales de segundo orden, y los ejemplos de aplicación en el capítulo 3 que ayudan a visualizar el método aplicado
- ✓ En el capítulo 3 se presentan los resultados del desarrollo de ecuaciones diferenciales con los métodos propuestos, así también se presentan sus resultados de manera gráfica para que los estudiantes pueden visualizar de mejor manera las soluciones de las ecuaciones diferenciales.
- ✓ Los ejercicios realizados han demostrado la eficacia del programa Python y de los métodos utilizados para la resolución de ecuaciones diferenciales de segundo orden. Lo cual se convertirá en una gran herramienta para los estudiantes de física y demás ciencias de nuestra casa de estudios.

RECOMENDACIONES

- ✓ Se recomienda capacitar a la planta docente y estudiantes sobre el uso del software Python con sus respectivas librerías y diferentes tipos de uso, para así facilitar la eficiente resolución de ecuaciones diferenciales de segundo orden en física computacional y en otras áreas concernientes a la carrera.
- ✓ Se recomienda elaborar un manual, mismo que contenga los pasos necesarios para la resolución de ecuaciones diferenciales de segundo orden en física computacional, mismo que servirá como una guía práctica para los alumnos de la carrera de física y matemática.
- ✓ Es recomendable la enseñanza basada en Software Python ya que al ser una programación más sencilla que otras, y estar basado en software libre su implementación va a ser mucho más sencilla y su aprendizaje mucho más amable con los estudiantes.

BIBLIOGRAFÍA

ACOSTA, JESÚS HUMBERTO CUEVAS; SOLÍS, MARÍA CRISTINA; CONTRERAS, IXRAEL SILVA. Programación computacional y análisis de datos en educación estadística. Areté: Revista Digital del Doctorado en Educación de la Universidad Central de Venezuela, 2019, vol. 5, no 9, p. 11-27.

ARAVENA, MARÍA; CAAMAÑO, CARLOS; GIMÉNEZ, JOAQUÍN. Modelos matemáticos a través de proyectos. Revista latinoamericana de investigación en matemática educativa, 2008, vol. 11, no 1, p. 49-92.

ARCE, YHONY MAMANI, ET AL. Solución analítica y numérica de la ecuación de Laplace utilizando Python. Revista de Investigación de Física, 2021, vol. 24, no 2, p. 40-48.

AREA MOREIRA, MANUEL. La innovación pedagógica con TIC y el desarrollo de las competencias informacionales y digitales. Revista de Investigación en la Escuela, 64, 5- 17., 2008.

BLANCO GÓMEZ, ANDREA. Librería de Python para fusión espacio-temporal de imágenes de teledetección. 2021.

COSTA, V.; TORROBA, PATRICIA; DEVECE, EUGENIO. Articulación en la enseñanza en carreras de ingeniería: el movimiento armónico simple y las ecuaciones diferenciales de segundo orden lineal. Latin American Journal of Physics Education, 2013, vol. 7, no 3, p. 350-356.

GARCÍA ALMEIDA, DANIELA IRANIA, et al. APLICACIÓN MÓVIL PARA RESOLVER ECUACIONES DIFERENCIALES ORDINARIAS (EDO). 2020.

HERNÁNDEZ, JESÚS DANIEL ARIAS; LÓPEZ, ANDRÉS FERNANDO JIMÉNEZ; CASTRO, HERNÁN OSWALDO PORRAS. Desarrollo de aplicaciones en python para el aprendizaje de física computacional. Ingeniería Investigación y Desarrollo: I2+ D, 2016, vol. 16, no 1, p. 72-82.

JOHANSSON, ROBERT. Numerical Python: A Practical Techniques Approach for Industry. Apress, 2015.

OVALLE, DIEGO ARÉVALO; YERMANOS, MIGUEL ÁNGEL BERNAL; RESTREPO, JAIME ANDRÉS POSADA. Matemáticas para Ingeniería: Métodos numéricos con Python. Institución Universitaria Politécnico Grancolombiano, 2017.

RECALDE, LUIS CORNELIO; HENAO, SARA MARCELA. Los obstáculos epistemológicos en el desarrollo histórico de las ecuaciones diferenciales ordinarias. Revista EIA, 2018, vol. 15, no 29, p. 59-70.

SANTILLÁN-LIMA, J. C., et al. El Impacto De Los Servicios De Telecomunicaciones y Las Tics En Las Necesidades De La Educación Superior. En IV Congreso Internacional de Ciencia Tecnología Innovación y Emprendimiento CITE. 2017.

SANTORUM VARELA, ALEJANDRO. Desarrollo de la librería scikit-rmt en Python para la simulación y el análisis de matrices aleatorias. 2021. Tesis de Licenciatura.

SEGUNDO, GUSTAVO. Solución numérica de la ecuación de Laplace. Anales del Instituto de Profesores “Artigas”, 2010.

SOLANO, J. Introducción a la programación en Python. Editorial Tecnológica del ITCR: Costa Rica, 2011.

TOOMEY, DAN. JUPYTER COOKBOOK: Over 75 recipes to perform interactive computing across Python, R, Scala, Spark, JavaScript, and more. Packt Publishing Ltd, 2018.

ZAMORA RAMOS, ALFONZO. Ecuaciones Diferenciales Parciales. Cuajimalpa: Casa Abierta del tiempo. 2011

ZAMORANO RUIZ, JUAN, et al. Comparación y análisis de métodos de clasificación con las bibliotecas scikit-learn y TensorFlow en Python. 2019.

ZETTERQVIST, LENA. Applied problems and use of technology in an aligned way in basic courses in probability and statistics for engineering students—a way to enhance understanding and increase motivation. Teaching Mathematics and its Applications: An International Journal of the IMA, 2017, vol. 36, no 2, p. 108-122.

ZILL, DENNIS G.; CULLEN, MICHAEL R. Ecuaciones diferenciales. McGraw-Hill Interamericana, 2013.



epoch

Dirección de Bibliotecas y
Recursos del Aprendizaje

*UNIDAD DE PROCESOS TÉCNICOS Y ANÁLISIS BIBLIOGRÁFICO Y
DOCUMENTAL*

REVISIÓN DE NORMAS TÉCNICAS, RESUMEN Y BIBLIOGRAFÍA

Fecha de entrega: 26 / 05 / 2022

INFORMACIÓN DEL AUTOR/A (S)
Nombres – Apellidos: Diego Alejandro Logroño Torres
INFORMACIÓN INSTITUCIONAL
Facultad: Ciencias
Carrera: Física
Título a optar: Físico
f. Analista de Biblioteca responsable: Ing. Leonardo Medina Ñuste MSc.


D.B.R.A.I.
Ing. Cristian Castillo



0989-DBRA-UTP-2022

