

Universidad Politécnica de Aguascalientes

Ingeniería en Sistemas Computacionales

Manual de Uso

Materia: Matemáticas para ingeniería II

Docente: Isaac Vázquez Mendoza

Equipo 4

Integrantes:

Vázquez Ríos Juan Ramon UP230190

Olmedo Cruz Danna Giselle UP230183

Surdez Espeleta Andrea Mariana UP230188

Martin Cornejo Diego Ernesto UP230346

Grado y Grupo: ISC06A

20 de Noviembre 2025

1. Requisitos

Para usar el programa necesitas:

Windows 10 o Windows 11

Python 3.10 o superior

Pip instalado

Permiso para instalar librerías

Todos los archivos .py del proyecto

2. Instalación de Python

Ir a la página oficial:

<https://www.python.org/downloads/>

Descargar Python 3.x para Windows.

-----Abir el instalador

----- IMPORTANTE: activar la casilla

-----Add Python to PATH

-----Dar clic en instalar ahora

- Cuando termine, abrir el símbolo del sistema (CMD) y verificar:

`python --version`

Debe mostrar algo como:

`Python 3.11.5`

3. Instalación de Librerías Necesarias

numpy

sympy

Matplotlib

pandas

Instalar librerías:

-----Abrir CMD

-----Escribir: `pip install numpy sympy pandas matplotlib`

4. Descarga del Proyecto desde GitHub

Para facilitar la instalación del proyecto, todo el código se encuentra disponible en un repositorio de GitHub.

De esta manera, pueden descargar todos los archivos .py y la estructura completa con un solo clic.

PASO 1 — Entrar al repositorio

<https://github.com/VRJuanRa/Ecuaciones-Diferenciales.git>

PASO 2 — Descargar el proyecto completo

En GitHub: Dar clic en el botón verde Code

Seleccionar:

Download ZIP

PASO 3 — Extraer el ZIP

Ubicar el archivo .zip descargado

Dar clic derecho

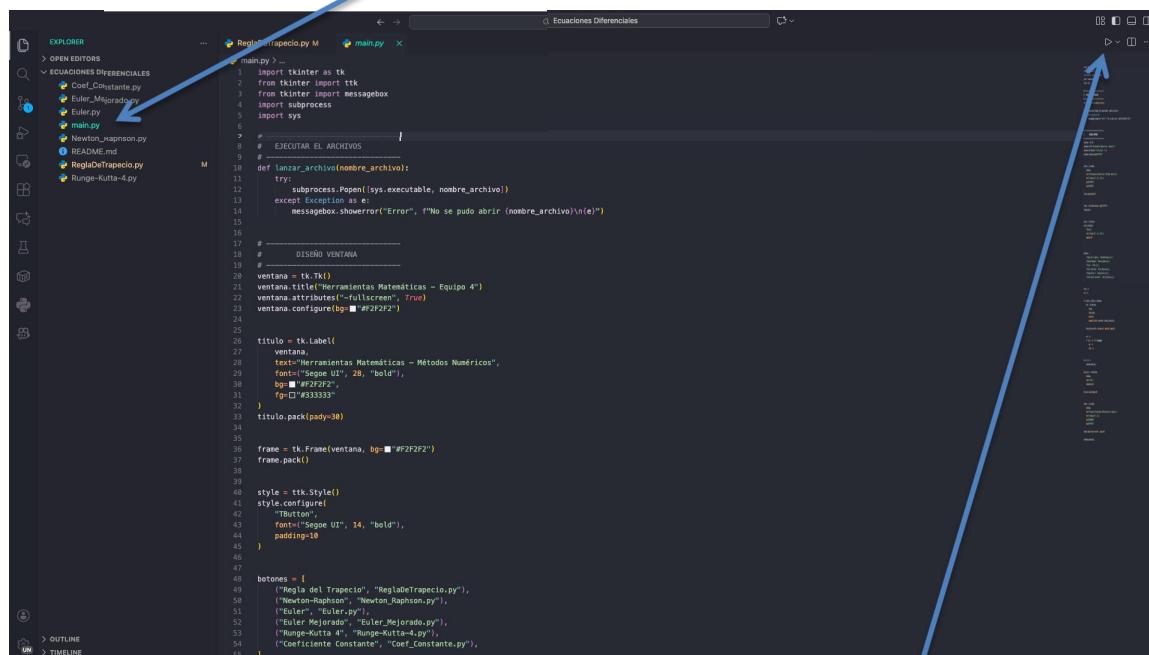
Seleccionar: Extraer aquí

PASO 4 — Abrir el proyecto

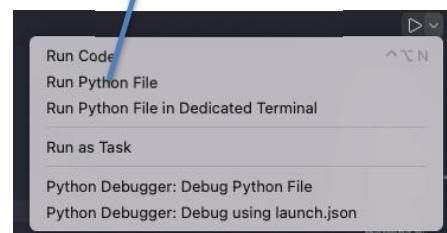
Después de extraerlo, el usuario ya tendrá la carpeta lista y estructurada correctamente.

No necesita mover nada, ni crear carpetas, ni renombrar archivos.

Solo debe abrir Visual Studio seleccionar la carpeta que descomprimiste y se abrirá la carpeta donde le dará clic al main.py y arriba a la derecha está la punta de una flecha que apunta al lado y te desplegará un menú donde le das run python file y se ejecutara por si solo.



Después de eso te abrirá una pantalla como esta





Donde tu ya seleccionas el metodo que deseas usar y se ejecutara por si solo

8. Errores Comunes

Error	Causa	Solución
SympifyError	Función mal escrita	Verificar **, sin, exp
ZeroDivisionError	División por cero	Cambiar x0 o valores
No se pudo abrir archivo	Nombre incorrecto	Revisar nombres .py
matplotlib no funciona	No instalado	pip install matplotlib u otra libreria...

Introducción

Este manual tiene como propósito explicar de forma clara, sencilla y completamente entendible para cualquier usuario el funcionamiento de todos los programas desarrollados por el Equipo 4 dentro del proyecto de Ecuaciones Diferenciales.

Además, incluye un diccionario de datos detallado por cada programa, explicando qué significa cada campo que el usuario debe ingresar.

Agregando tambien los requerimientos para la ejecucion del mismo.

Regla del Trapecio

Este método permite aproximar el valor de una integral definida usando trapecios para estimar el área.

El usuario debe ingresar la función $f(x)$, y los límites a y b .

También debe ingresar el número de particiones n , que determina la precisión.

Luego presiona el botón Calcular Integral.

● ○ ■ Calculadora de Integrales - Regla del Trapecio

Ingrese la función $f(x)$:

Límite inferior (a):

Límite superior (b):

Número de particiones (n):

Calcular Integral

Ejemplo de uso

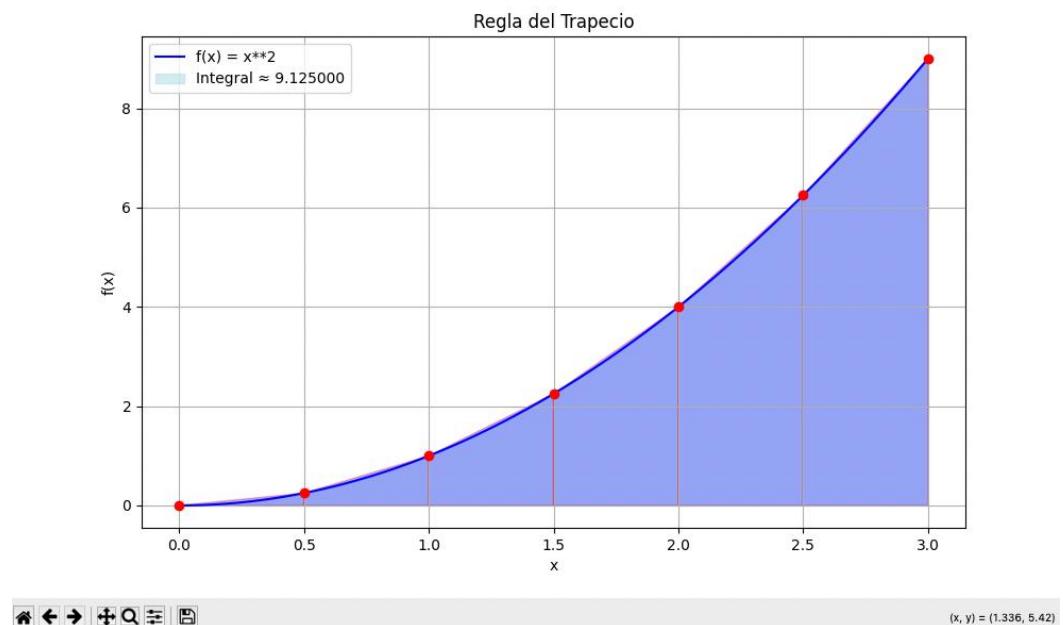
Entrada:

$f(x): x^{**2}$

$a = 0$

$b = 3$

$n = 6$



Diccionario de datos:

En esta sección se explican los valores que el usuario debe ingresar en cada label.

Función f(x): Expresión matemática que se integrará.

$\sin(x)$, $x^{**2} + 3*x + 2$, $\exp(-x^{**2})$

Límite a: Valor inicial del intervalo.

0

Límite b: Valor final del intervalo.

3

Número de segmentos n: Mientras más segmentos, mayor precisión.

6

Newton-Raphson

Este método sirve para encontrar raíces de funciones mediante un proceso iterativo.

Debes ingresar la función $f(x)$ y su derivada $f'(x)$.

x_0 es el valor inicial donde comenzará la aproximación.

El usuario elige cuántas iteraciones máximo se ejecutarán.

La tabla muestra paso a paso cómo se acerca la solución.

También incluye una gráfica donde se visualizan los puntos evaluados.

Newton-Raphson Equipo 4

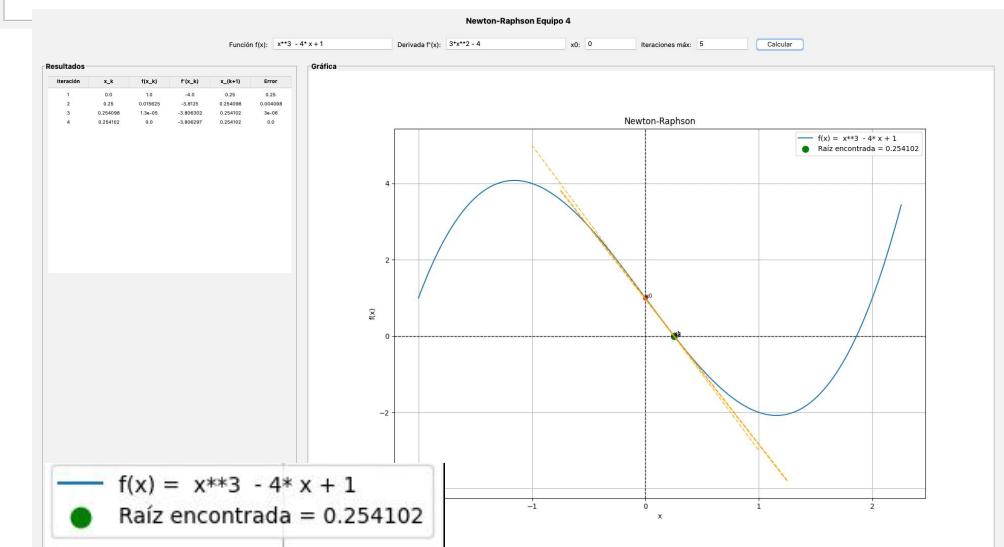
Función f(x): Derivada f'(x): x0: Iteraciones máx: Calcular

Resultados

Iteración	x_k	f(x_k)	f'(x_k)	x_(k+1)	Error
1	0.0	1.0	-4.0	0.25	0.25
2	0.25	0.0025	-3.75	0.254000	0.000025
3	0.254000	1.3e-05	-3.80322	0.254102	3e-06
4	0.254102	0.0	-3.80237	0.254102	0.0

En las imágenes se muestra en donde debes insertar los valores y la tabla donde se mostrarán los resultados en base a la función que diste el valor inicial (x0) y el número de iteraciones que tu diste.

Gráfica



Ejemplo de uso

Entrada:

f(x): $x^{**3} - 4*x + 1$

f'(x): $3*x^{**2} - 4$

x = 0

Interacciones = 5

Diccionario de datos:

f(x): Función de la cual se desea encontrar una raíz.

$$x^{**3} - 4^* x + 1$$

f'(x): Derivada de la función f(x).

$$3*x^{**2} - 4$$

x0: Valor inicial de aproximación.

$$0$$

Iteraciones: Cantidad máxima de pasos del método.

$$5$$

Euler

- Este método aproxima soluciones de ecuaciones diferenciales de la forma $y' = f(x, y)$.
- Se debe ingresar $f(x, y)$, x_0 , y_0 y el paso h .
- A menor h , más precisa la aproximación.
- El programa genera una tabla con los valores de cada iteración y una gráfica.

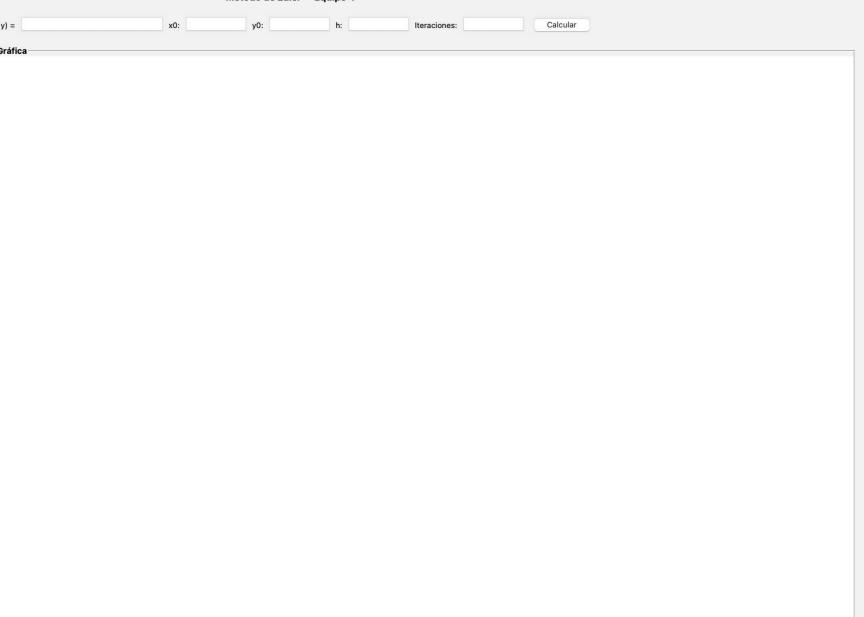
Método de Euler — Equipo 4

f(x, y) =	<input type="text"/>	x0:	<input type="text"/>	y0:	<input type="text"/>	h:	<input type="text"/>	Iteraciones:	<input type="text"/>	<input type="button" value="Calcular"/>
-----------	----------------------	-----	----------------------	-----	----------------------	----	----------------------	--------------	----------------------	---

Método de Euler — Equipo 4

Resultados			
Iter	x _k	y _k	y _(k+1)

Gráfica



Resultados

Iter	x _k	y _k	y _(k+1)

Ejemplo de uso

Entrada:

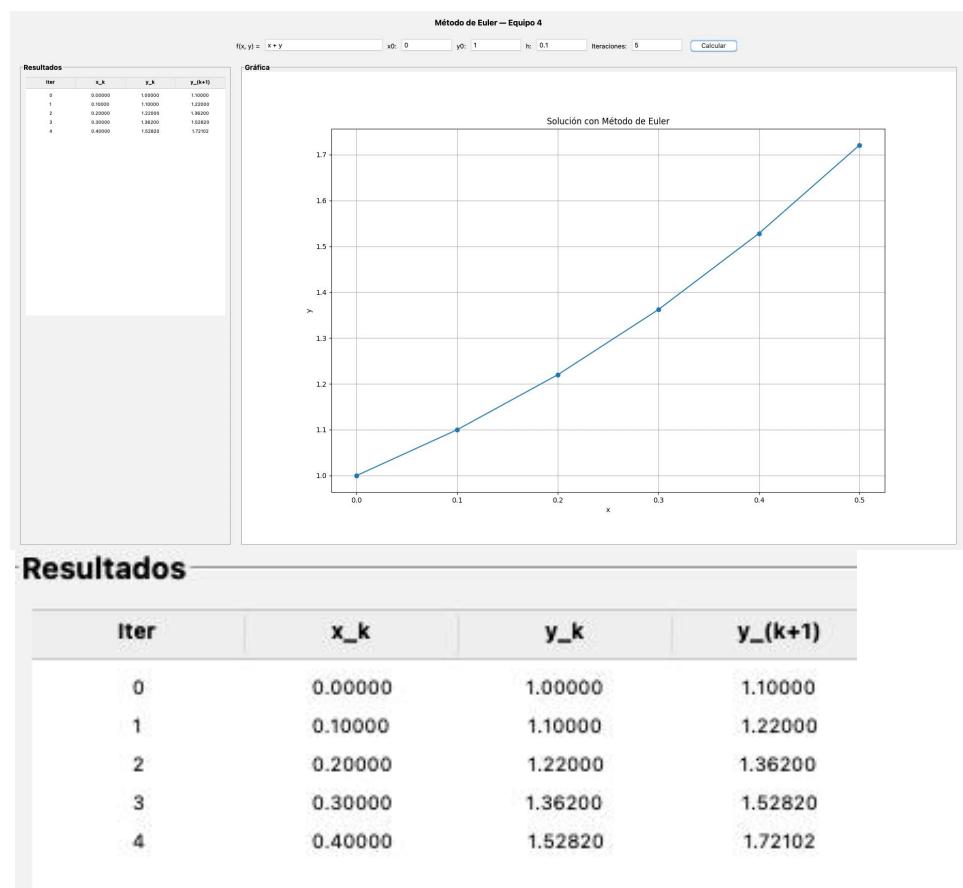
$$y' = x + y$$

X⁰: 0

Y⁰: 1

h = 0.1

Interaciones = 5



Diccionario de datos:

f(x, y): Ecuación diferencial expresada en términos de x y y.

$$y' = x + y$$

x0: Punto inicial del eje x.

0

y0: Valor de y correspondiente al punto x0.

1

h: Tamaño del paso. Controla precisión y velocidad del método

0.1.

Interaciones: Cantidad de pasos a ejecutar.

Euler Mejorado

Euler Mejorado es una versión más precisa del método Euler.

Utiliza un paso de predicción y otro de corrección, promediando las pendientes.

El usuario ingresa la ecuación diferencial, valores iniciales y el paso h.

Se muestran tanto los valores predichos como corregidos para mayor claridad.

Método de Euler Mejorado — Equipo 4

f(x, y) = x0: y0: h: Iteraciones: Calcular

Método de Euler Mejorado — Equipo 4

f(x, y) = x0: y0: h: Iteraciones: Calcular

Resultados

Iter	x_k	y_k	y_pred	y_(k+1)

Gráfica

Página 13 de 18

Ejemplo de uso

Entrada:

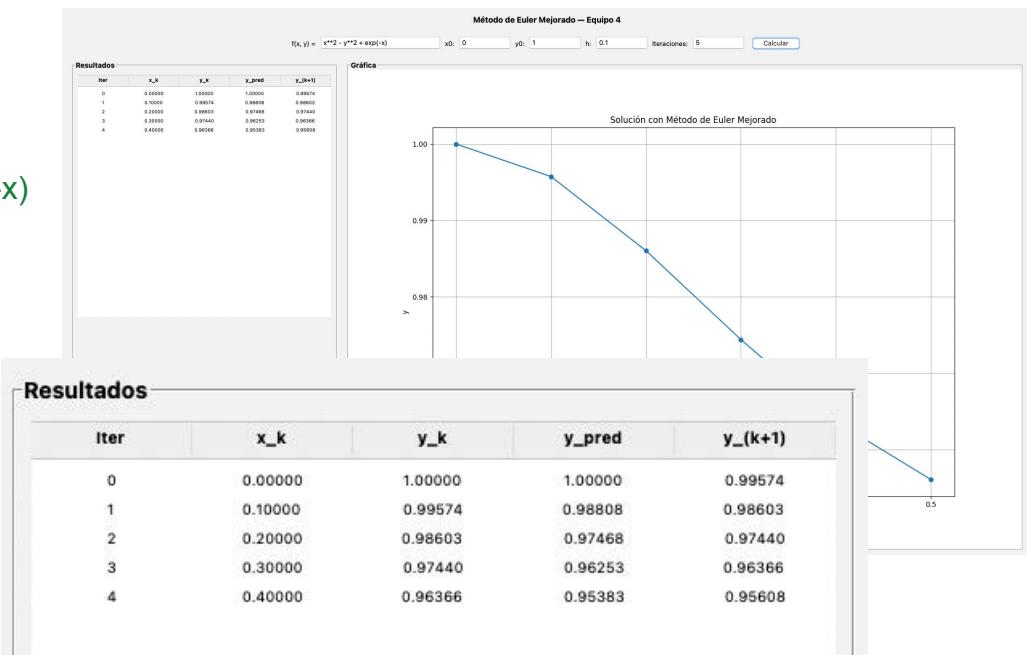
$$y' = x^{**2} - y^{**2} + \exp^{**}(-x)$$

X⁰: 0

Y⁰: 1

h = 0.1

Iteraciones = 5



Diccionario de datos:

f(x, y): Función de la ecuación diferencial.

$$y' = x^{**2} - y^{**2} + \exp^{**}(-x)$$

x0: Valor inicial de x.

0

y0: Valor inicial de y.

1

h: Tamaño del paso.

0.1

Iteraciones: Cantidad de repeticiones.

5

Predicción: Primer estimación del siguiente valor.

Corrección: Valor final usando promedio de pendientes.

Runge-Kutta 4

RK4 es uno de los métodos numéricos más precisos para resolver EDO de primer orden.

El usuario ingresa $f(x, y)$, x_0 , y_0 , h e iteraciones.

El programa calcula k_1 , k_2 , k_3 y k_4 internamente.

La gráfica muestra la aproximación suave y precisa que caracteriza a RK4.

Ejemplo de uso

Entrada:

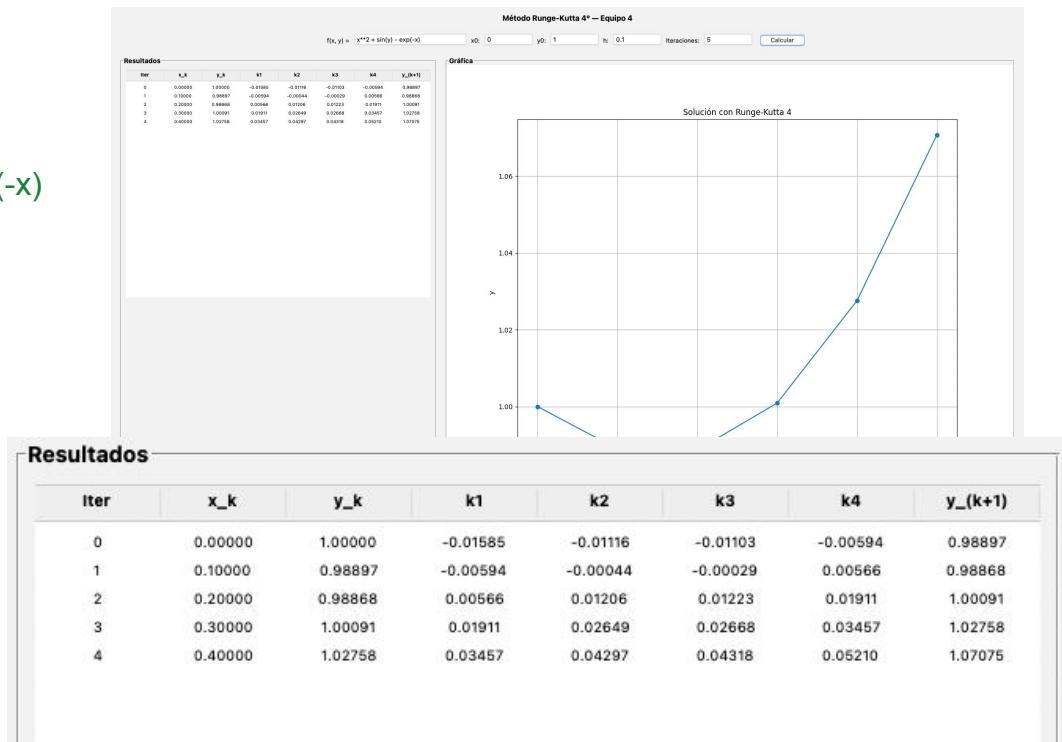
$$y' = x^2 + \sin(y) + \exp(-x)$$

X⁰: 0

Y⁰: 1

h = 0.1

Iteraciones = 5



Diccionario de datos:

f(x, y): Función diferencial.

$$y' = x^2 + \sin(y) + \exp(-x)$$

x0: Valor inicial.

0

y0: Valor inicial.

1

h: Paso.

0.1

Iteraciones: Cuántas veces avanza el método.

5

k1-k4: Pendientes intermedias que determinan el valor final.

Ecuación de Coeficiente Constante

Este programa resuelve ecuaciones de la forma $y' = a \cdot y + b$.

El usuario ingresa los coeficientes a y b , así como x_0 , y_0 y el paso h .

Es ideal para estudiar ecuaciones lineales con comportamiento exponencial.

Método de Euler — $a \cdot y + b$ (Coeficientes Constantes)

a:	<input type="text"/>	b:	<input type="text"/>	x0:	<input type="text"/>	y0:	<input type="text"/>	h:	<input type="text"/>	Iteraciones:	<input type="text"/>	<input type="button" value="Calcular"/>
----	----------------------	----	----------------------	-----	----------------------	-----	----------------------	----	----------------------	--------------	----------------------	---

Método de Euler — $a \cdot y + b$ (Coeficientes Constantes)

a:	<input type="text"/>	b:	<input type="text"/>	x0:	<input type="text"/>	y0:	<input type="text"/>	h:	<input type="text"/>	Iteraciones:	<input type="button" value="Calcular"/>
----	----------------------	----	----------------------	-----	----------------------	-----	----------------------	----	----------------------	--------------	---

Resultados

Iter	x_{-k}	y_{-k}	$y_{-(k+1)}$

Gráfica

Resultados

Iter	x_{-k}	y_{-k}	$y_{-(k+1)}$

Ejemplo de uso

Entrada:

$a = 4$

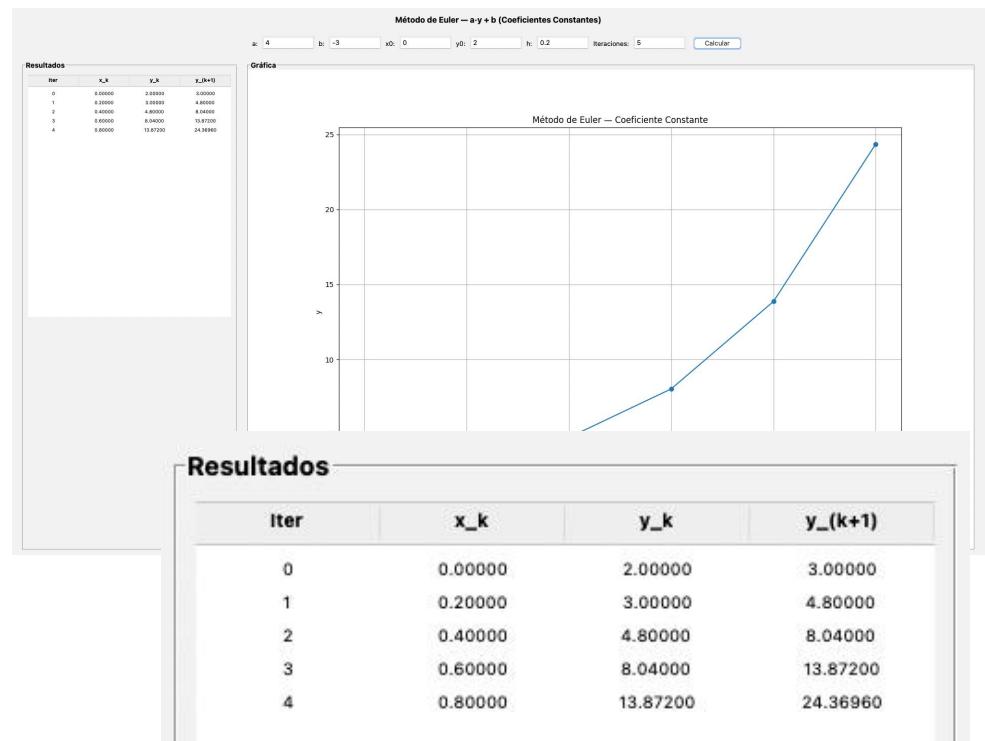
$b = -3$

$X^0: 0$

$Y^0: 2$

$h = 0.2$

Iteraciones = 5



Diccionario de datos:

a: Coeficiente multiplicado por y.

4

b: Coeficiente constante.

-3

x_0 : Inicio de x.

0

y_0 : Valor inicial de y.

2

h: Paso.

0.2

Iteraciones: Número de aproximaciones.

5