

Informe 1

Título del Informe

Acerca del Autor

Nombre del Autor
CORREO DEL AUTOR
Mes y Año

1.1 Resumen

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec bibendum velit vel pretium ultricies. Praesent vehicula felis vitae lacus bibendum pharetra. Donec non odio feugiat, accumsan nunc vel, interdum purus. Sed semper egestas nisl in ultrices. Curabitur facilisis felis vitae odio egestas, eget pretium quam tempus. Suspendisse enim nisi, ullamcorper semper augue id, volutpat facilisis mauris.

1.2 Explicación del Fenómeno o Modelo

In tempus sapien massa, a interdum ipsum laoreet eget. Vivamus gravida enim ex, porttitor ultricies augue pulvinar ut. Praesent sit amet magna cursus, facilisis tellus ut, rhoncus nisl. Vivamus et lacinia purus, quis vehicula leo. Nulla aliquam pulvinar odio id eleifend. Interdum et

malesuada fames ac ante ipsum primis in faucibus. Morbi cursus ullamcorper augue in efficitur. Aliquam ullamcorper orci vel erat commodo hendrerit. Proin eu semper lorem. Aenean leo lacus, gravida vitae eleifend ac, luctus nec ante.

A continuación se muestra una gráfica de la función.

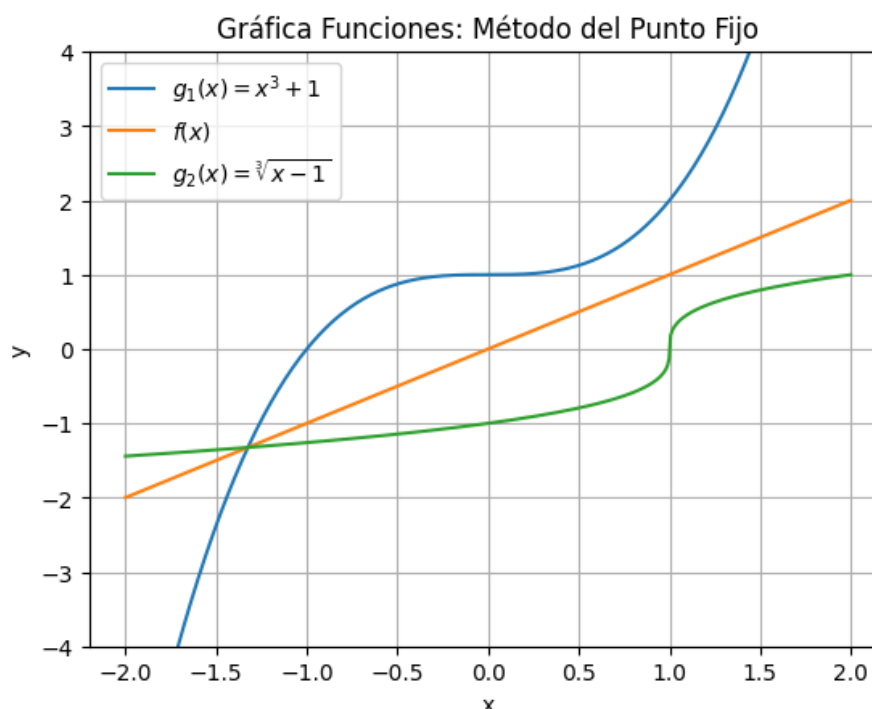


Figura 1.1: Etiqueta de la gráfica

Tenemos un sistema de ecuaciones

$$S = \begin{cases} ax + by &= c \\ dx + ey &= f \end{cases} \quad (1.1)$$

1.3 Datos

Se hace una recopilación de los datos generados por el experimento y se explica su procedencia y la forma de tomarlos. Una tabla de datos del sistema 1.1

x	$f(x)$
1	2
2	3
3	5
4	7
5	11

Cuadro 1.1: Etiqueta de la tabla

1.4 Pseudocódigo

Explicación de los pasos necesarios para resolver el problema.

1.5 Código en Python

Explicación y referencia del código en Python, ya sea realizado en Google Colab, Visual Studio, Spider, etc.

Para ello existe el comando Google Colab Rincón, J (1)

Programa de Google Colab

Este programa se ha desarrollado usando Google Colab. Para ver el programa haz clic en el logo.



```
1 # Crear un *programa* en **Python
   ** que grafique el polinomio  $P_{\{3\}}$ 
    $\left(x\right)=x^{\{3\}}-10x$ 
```

```

2
3 # Importar las librerias
  necesarias
4 import numpy as np
5 import matplotlib.pyplot as plt
6
7 # Definir el polinomio
8 def poly(x):
9     return x**3 - 10*x**2 + 31*x -
      30
10
11 # Crear el dominio de graficación
12 x = np.linspace(1, 6, 100)
13
14 # Evaluar el polinomio en x
15 y = poly(x)
16
17 # Graficar y configurar la gráfica
18 plt.plot(x, y)
19 plt.xlabel( '$x$ ')
20 plt.ylabel( '$P_n(x)$ ')
21 plt.title( 'Gráfico del Polinomio
    $P_n(x)$ ')
22 plt.grid( linestyle='--' )
23 plt.show()
```

1.6 Simulaciones

Se muestran imágenes de las gráficas de las simulaciones y los respectivos resultados.

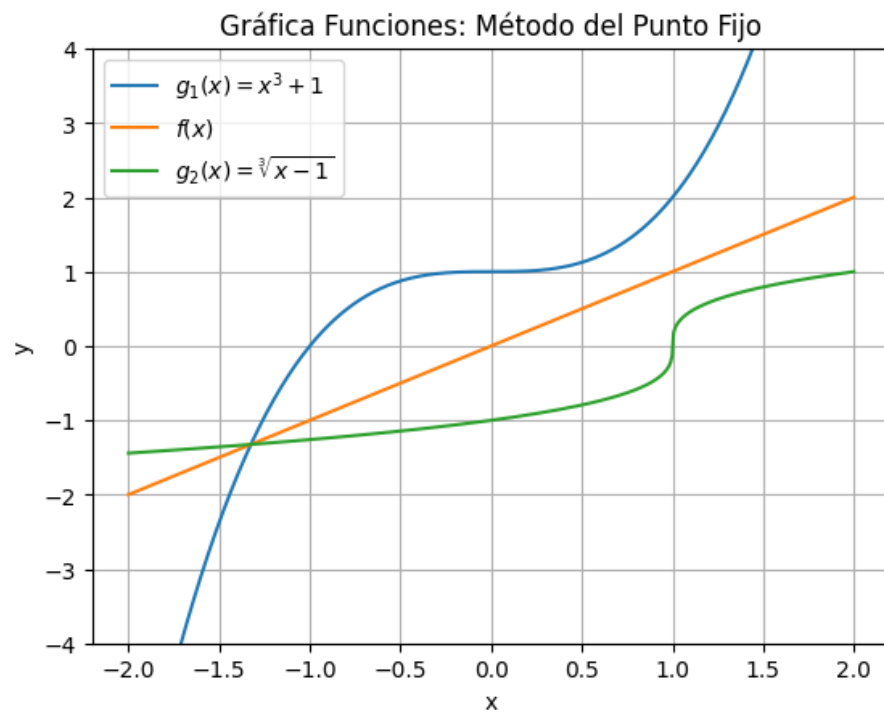


Figura 1.2: Etiqueta de la gráfica

1.7 Conclusiones

Se generan las conclusiones al problema.

1. Conclusión 1
2. Conclusión 2
3. Conclusión 3

Bibliografía

- [1] Rincón,
J. (2012). *Gráfica de
funciones polares*.
Editorial Elizcom
- [2] Rincón,
J. (2012). *Gráfica de
funciones polares*.
Editorial Elizcom
- [3] Rincón,
J. (2012). *Gráfica de
funciones polares*.
Editorial Elizcom