

Resolutor Gauss-Jordan en Python

Código en Python

El siguiente código implementa el método de Gauss-Jordan para resolver sistemas de ecuaciones lineales, con una interfaz gráfica creada con Tkinter:

```
1 import numpy as np
2 import tkinter as tk
3 from tkinter import messagebox, simpledialog
4
5 def gauss_jordan(x, y, verbose=0):
6     m, n = x.shape
7     augmented_mat = np.zeros(shape=(m, n + 1))
8     augmented_mat[:m, :n] = x
9     augmented_mat[:, m] = y
10    np.set_printoptions(precision=2, suppress=True)
11    if verbose > 0:
12        print('# Matriz aumentada original')
13        print(augmented_mat)
14    outer_loop = [[0, m - 1, 1], [m - 1, 0, -1]]
15    for d in range(2):
16        for i in range(outer_loop[d][0], outer_loop[d][1],
17                        outer_loop[d][2]):
18            inner_loop = [[i + 1, m, 1], [i - 1, -1, -1]]
19            for j in range(inner_loop[d][0], inner_loop[d][1],
20                            inner_loop[d][2]):
21                k = (-1) * augmented_mat[j, i] /
22                    augmented_mat[i, i]
23                temp_row = augmented_mat[i, :] * k
24                if verbose > 1:
25                    print('# Usar la l nea %2i para la
26                          l nea %2i' % (i + 1, j + 1))
27                    print('k=%.2f' % k, '*', augmented_mat[i, :],
28                          '=', temp_row)
29                augmented_mat[j, :] = augmented_mat[j, :] +
30                    temp_row
31            if verbose > 1:
32                print(augmented_mat)
33    for i in range(0, m):
34        augmented_mat[i, :] = augmented_mat[i, :] /
35            augmented_mat[i, i]
```

```

29     if verbose > 0:
30         print('# Normalizar las filas')
31         print(augmented_mat)
32     return augmented_mat[:, n]
33
34 def resolver_con_gui():
35     try:
36         # Obtener dimensiones de la matriz
37         tamaño = simpledialog.askinteger("Entrada", "
38             Ingresa el tamaño de la matriz de coeficientes (
39             n x n):")
40
41         if not tamaño or tamaño < 1:
42             raise ValueError("El tamaño debe ser un número
43                 entero positivo.")
44
45         # Ingresar la matriz de coeficientes
46         coeficientes = []
47         for i in range(tamaño):
48             fila = simpledialog.askstring("Entrada", f"
49                 Ingresa la fila {i+1} de coeficientes,
50                 separada por espacios:")
51             coeficientes.append([float(num) for num in fila.
52                 split()])
53
54         # Ingresar el lado derecho
55         lado_derecho = simpledialog.askstring("Entrada", f"
56             Ingresa el vector del lado derecho, separado por
57             espacios:")
58         lado_derecho = [float(num) for num in lado_derecho.
59             split()]
60
61         coeficientes = np.array(coeficientes)
62         lado_derecho = np.array(lado_derecho)
63
64         # Verificar dimensiones
65         if coeficientes.shape != (tamaño, tamaño) or len(
66             lado_derecho) != tamaño:
67             raise ValueError("Las dimensiones de la matriz
68                 no coinciden.")
69
70         # Resolver
71         solucion = gauss_jordan(coeficientes, lado_derecho,
72             verbose=1)
73         mensaje_resultado = "\n".join([f"x{i+1} = {solucion[
74             i]:.2f}" for i in range(tamaño)])
75         messagebox.showinfo("Solución", f"La solución es:\n
76             {mensaje_resultado}")
77
78     except Exception as e:
79         messagebox.showerror("Error", f"Ocurrió un error:\n
80             {e}")

```

```

        {e}")
65
66 # Configuraci n de la interfaz gr fica
67 root = tk.Tk()
68 root.title("Resolutor Gauss-Jordan")
69
70 etiqueta = tk.Label(root, text=" Bienvenido al Resolutor
    Gauss-Jordan!", font=("Arial", 16))
71 etiqueta.pack(pady=10)
72
73 boton_resolver = tk.Button(root, text="Resolver el Sistema",
    font=("Arial", 14), command=resolver_con_gui)
74 boton_resolver.pack(pady=10)
75
76 boton_salir = tk.Button(root, text="Salir", font=("Arial",
    14), command=root.quit)
77 boton_salir.pack(pady=10)
78
79 root.mainloop()

```