

INTERFAZ DE RESOLUCIÓN DE SISTEMAS DE ECUACIONES

ALUMNO: Eddy Kennedy Mamani Hallasi

January 21, 2025

1 Introducción

Este informe presenta una interfaz implementada en Python con Streamlit para resolver sistemas de ecuaciones lineales utilizando tres métodos: Sustitución, Gauss-Jordan y la Regla de Cramer.

La aplicación permite a los usuarios ingresar coeficientes y términos independientes de las ecuaciones, seleccionar un método y obtener soluciones detalladas paso a paso.

A continuación, se incluyen los enlaces relevantes al repositorio de GitHub que contiene el código fuente y a la interfaz en Streamlit:

- **Repositorio en GitHub:** <https://github.com/MetodosDeOptimizacion/ResolucionSistemasEcuaciones>
- **Interfaz en Streamlit:** <https://resolucionssystemasecuaciones-ehivzus2njepd8ywqxy.streamlit.app/>

2 Ejemplo de resolución

Para simplificar la comprensión, asignaremos:

$$x_1 = x$$

$$x_2 = y$$

$$x_3 = z$$

Como ejemplo, consideremos el siguiente sistema de ecuaciones:

$$\begin{aligned}2x + 3y - z &= 5 \\4x - y + 2z &= 6 \\-x + 2y + 3z &= 7\end{aligned}$$

La aplicación resolverá este sistema utilizando los métodos mencionados y mostrará los pasos detallados junto con las soluciones finales para x , y y z .

3 Código de la interfaz

El siguiente es el código completo de la interfaz desarrollada:

```
1 import streamlit as st
2 from sympy import symbols, Eq, solve, Matrix
3
4 # Funci n para resolver por sustituci n
5 def metodo_sustitucion(ecuaciones, variables):
6     pasos = []
7     soluciones = solve(ecuaciones, variables)
8     for i, eq in enumerate(ecuaciones):
9         pasos.append(f"Paso_{i+1}: Usamos la ecuaci n
10             _{eq} para despejar variables.")
11     pasos.append("Finalmente, resolvemos el sistema
12         completo:")
13     for var, val in soluciones.items():
14         pasos.append(f"{var} = {val}")
15     return pasos, soluciones
16
17 # Funci n para resolver por Gauss-Jordan
18 def metodo_gauss_jordan(coeficientes, independientes):
19     pasos = []
20     matriz = Matrix([fila + [ind] for fila, ind in zip(
21         coeficientes, independientes)])
22     pasos.append(f"Matriz inicial aumentada:\n{matriz}")
23     matriz_reducida, pivotes = matriz.rref()
24     pasos.append(f"Matriz escalonada reducida:\n{
25         matriz_reducida}")
26     soluciones = [matriz_reducida[i, -1] for i in range(
27         matriz_reducida.rows)]
```

```

23     for i, sol in enumerate(soluciones):
24         pasos.append(f"x{i+1}={sol}")
25     return pasos, soluciones
26
27 # Funci n para resolver por regla de Cramer
28 def metodo_cramer(coeficientes, independientes):
29     pasos = []
30     matriz_coeficientes = Matrix(coeficientes)
31     matriz_independientes = Matrix(independientes)
32     det_coeficientes = matriz_coeficientes.det()
33
34     if det_coeficientes == 0:
35         pasos.append("El sistema no tiene soluci n
36             nica porque el determinante de la matriz de
37             coeficientes es 0.")
38         return pasos, "El sistema no tiene soluci n
39             nica ."
40
41     pasos.append(f"Determinante de la matriz de
42         coeficientes (A): {det_coeficientes}")
43     soluciones = []
44     for i in range(len(coeficientes)):
45         matriz_sustituida = matriz_coeficientes.copy()
46         matriz_sustituida[:, i] = matriz_independientes
47         det_sustituida = matriz_sustituida.det()
48         pasos.append(f"Matriz con columna {i+1}
49             sustituida:\n{matriz_sustituida}")
50         pasos.append(f"Determinante de la matriz
51             modificada: {det_sustituida}")
52         soluciones.append(det_sustituida /
53             det_coeficientes)
54         pasos.append(f"x{i+1}={det_sustituida}/{
55             det_coeficientes}={soluciones[-1]}")
56
57     return pasos, soluciones
58
59 # Interfaz de Streamlit
60 st.title("Resoluci n de Sistemas de Ecuaciones")
61 st.write("Esta aplicaci n resuelve sistemas de
62     ecuaciones utilizando los m todos de Sustituci n ,

```

```

54         Gauss-Jordan y Regla de Cramer.")
55 # Entrada: N mero de ecuaciones
56 num_ecuaciones = st.number_input("N mero de ecuaciones:
57     ", min_value=2, step=1)
58 if num_ecuaciones:
59     # Entrada: Coeficientes y t rminos independientes
60     st.write("Ingrese los coeficientes y t rminos
61         independientes:")
62
63     coeficientes = []
64     terminos_independientes = []
65
66     for i in range(num_ecuaciones):
67         st.write(f"Ecuaci n {i+1}:")
68         fila = [st.number_input(f"Coeficiente de x {j+1}
69             (Ecuaci n {i+1}):", key=f"coef_{i}_{j}"
70             ) for j in range(num_ecuaciones)]
71         coeficientes.append(fila)
72         terminos_independientes.append(st.number_input(f
73             "T rmino independiente (Ecuaci n {i+1}):"
74             , key=f"indep_{i}"))
75
76 # Selecci n del m todo
77 metodo = st.selectbox("Seleccione el m todo para
78     resolver:", ["Sustituci n", "Gauss-Jordan", "
79     Regla de Cramer"])
80
81 if st.button("Resolver"):
82     variables = symbols('_',).join([f'x_{i+1}' for i
83         in range(num_ecuaciones)])
84
85     if metodo == "Sustituci n":
86         ecuaciones = []
87         for i in range(num_ecuaciones):
88             ecuacion = Eq(sum(coeficientes[i][j] *
89                 variables[j] for j in range(
90                     num_ecuaciones)),
91                 terminos_independientes[i])

```

```

81         ecuaciones.append(ecuacion)
82     pasos, soluciones = metodo_sustitucion(
83         ecuaciones, variables)
84
85     elif metodo == "Gauss-Jordan":
86         pasos, soluciones = metodo_gauss_jordan(
87             coeficientes, terminos_independientes)
88
89     elif metodo == "Regla de Cramer":
90         pasos, soluciones = metodo_cramer(
91             coeficientes, terminos_independientes)
92
93     # Mostrar los pasos y las soluciones
94     st.write("### Pasos del cálculo:")
95     for paso in pasos:
96         st.write(paso)
97
98     st.write("### Soluciones:")
99     if isinstance(soluciones, str):
100         st.write(soluciones)
101     else:
102         for i, solucion in enumerate(soluciones):
103             st.write(f"x{i+1}={solucion}")

```

4 Conclusión

La interfaz desarrollada facilita la resolución de sistemas de ecuaciones lineales de una manera eficiente y didáctica. Gracias a la implementación de los tres métodos mencionados, es posible visualizar tanto los pasos intermedios como las soluciones finales, lo que ayuda a comprender el proceso de cálculo. La herramienta es ideal para estudiantes y profesionales interesados en métodos numéricos y álgebra lineal.