1.9.30

Nipun Dasari - EE25BTECH11042

January 9, 2025

Question

Solve the following system of rational equations

$$\frac{10}{x+y} + \frac{2}{x-y} = 4 \tag{1}$$

$$\frac{15}{x+y} - \frac{5}{x-y} = -2 \tag{2}$$

Theoretical Solution

$\frac{A}{x+y} + \frac{B}{x-y} = C$ becomes:

$$c(x^2 - y^2) - (a+b)x + (a-b)y = 0 \tag{3}$$

Matrix form: $\mathbf{x}^\top V \mathbf{x} + 2\mathbf{u}^\top \mathbf{x} = 0$, where:

$$\mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix}, \quad V = \begin{pmatrix} c & 0 \\ 0 & -c \end{pmatrix}, \quad \mathbf{u} = \begin{pmatrix} -(a+b)/2 \\ (a-b)/2 \end{pmatrix} \tag{4}$$

The intersection points of the two hyperbolas lie on a Common chord, $c_1 H_2 - c_2 H_1 = 0$, where $H_1 = 0$ and $H_2 = 0$ are the equations of each of hyperbolas. This results in the linear equation $\mathbf{n}^\top \mathbf{x} = 0$,

$$d = c_1(a_2 + b_2) - c_2(a_1 + b_1) \tag{5}$$

$$e = c_2(a_1 - b_1) - c_1(a_2 - b_2) \tag{6}$$

where d and e are obtained by eliminating the quadratic terms $\mathbf{n} = \begin{pmatrix} d \\ e \end{pmatrix}$

Theoretical Solution

The solution is the non-trivial intersection point of this common chord and either hyperbola

$$y = -\frac{d}{e}x \tag{8}$$

$$c_1\left(x^2 - \left(-\frac{d}{e}x\right)^2\right) - (a_1 + b_1)\,x + (a_1 - b_1)\left(-\frac{d}{e}x\right) = 0 \tag{9}$$

$$\implies x\left(x\left(c_1\left(\frac{e^2 - d^2}{e^2}\right)\right) - \left(\frac{e\,(a_1 + b_1) + d\,(a_1 - b_1)}{e}\right)\right) = 0 \tag{10}$$

$$\therefore x\left(c_1\left(\frac{e^2 - d^2}{e^2}\right)\right) = \left(\frac{e\,(a_1 + b_1) + d\,(a_1 - b_1)}{e}\right) \tag{11}$$

$$x = \frac{e\,(e\,(a_1 + b_1) + d\,(a_1 - b_1))}{c_1\,(e^2 - d^2)} \tag{12}$$

Theoretical Solution

For the given system, the coefficients are:

$$a_1 = 10, b_1 = 2, c_1 = 4 \quad \text{and} \quad a_2 = 15, b_2 = -5, c_2 = -2 \tag{13}$$

Using (10) and (11), we calculate the common chord coefficients:

$$d = 4\,(15-5) - (-2)\,(10+2) = 4\,(10) + 2\,(12) = 64 \tag{14}$$

$$e = (-2)\,(10-2) - 4\,(15 - -5) = -2(8) - 4(20) = -96 \tag{15}$$

Substituting these into the formula for x from (17):

Theoretical Solution

$$x = \frac{(-96)\left((-96)(12) + (64)(8)\right)}{4\left((-96)^2 - (64)^2\right)} \tag{16}$$

$$= \frac{-96\left(-1152 + 512\right)}{4\left(9216 - 4096\right)} \tag{17}$$

$$= \frac{-96\left(-640\right)}{4\left(5120\right)} = \frac{61440}{20480} = 3 \tag{18}$$

Using the value of x from (23) in the formula for y (13):

$$y = -\frac{64}{-96}(3) = \frac{2}{3}3 = 2 \tag{19}$$

Thus, from (23) and (24), the solution is $x = 3$ and $y = 2$.

C Code

```
void get_conic_data(double* data_out) {      //
    Hyperbola 1: V1=[[4,0],[0,-4]], u1=[-6,4]   data_out
    [0] = 4.0; data_out[1] = 0.0;
        data_out[2] = 0.0; data_out[3] = -4.0;
        data_out[4] = -6.0; data_out[5] = 4.0;
        // Hyperbola 2: V2=[[-2,0],[0,2]], u2=[-5,10]
        data_out[6] = -2.0; data_out[7] = 0.0;
        data_out[8] = 0.0; data_out[9] = 2.0;
        data_out[10] = -5.0; data_out[11] = 10.0;
        // Solution Point: [3, 2]
data_out[12] = 3.0; data_out[13] = 2.0;
        }
```

Python Code using shared output

```
# plot_from_so.py
import ctypes
import numpy as np
import matplotlib.pyplot as plt
# Load the shared library
lib = ctypes.CDLL('./5.2.44.so')
# Define the function signature
get_data_func = lib.get_conic_data
get_data_func.argtypes = [np.ctypeslib.ndpointer(dtype=np.double,
    ndim=1, flags='C_CONTIGUOUS')]
get_data_func.restype = None
```

Python Code using shared output

```python
# Create a buffer and call the C function
output_array = np.zeros(14, dtype=np.double)   get_data_func(
    output_array)
# Unpack the data from C
V1 = output_array[0:4].reshape((2, 2))
u1 = output_array[4:6]
V2 = output_array[6:10].reshape((2, 2))
u2 = output_array[10:12]
solution_point = output_array[12:14]
# --- Plotting Code ---
x_vals = np.linspace(-10, 15, 500)
y_vals = np.linspace(-10, 15, 500)
X, Y = np.meshgrid(x_vals, y_vals)
eq1 = V1[0,0]*X**2 + V1[1,1]*Y**2 + 2*(u1[0]*X + u1[1]*Y)
eq2 = V2[0,0]*X**2 + V2[1,1]*Y**2 + 2*(u2[0]*X + u2[1]*Y)
```

Python Code using shared output

```
plt.figure(figsize=(10, 10))
plt.contour(X, Y, eq1, levels=[0], colors='red')
plt.contour(X, Y, eq2, levels=[0], colors='blue')
plt.plot(x_vals, (2/3)*x_vals, 'g--', label='Common Chord')
plt.plot(solution_point[0], solution_point[1], 'ko', markersize
    =10, label=f'Solution from C: ({solution_point[0]}, {
    solution_point[1]})')
plt.title('Plot from C Shared Library Data', fontsize=16)
plt.xlabel('x-axis'); plt.ylabel('y-axis')
plt.grid(True, linestyle='--'); plt.axhline(0, color='k', lw=0.5)
    ; plt.axvline(0, color='k', lw=0.5)
plt.gca().set_aspect('equal', adjustable='box'); plt.xlim(-5, 10)
    ; plt.ylim(-5, 10)
plt.legend()
plt.savefig('so_python_plot.png')
print(Plot saved to so_python_plot.png)
plt.show()
```

Python Code

```
# Code to plot the solution of the system of rational equations
import numpy as np
import matplotlib.pyplot as plt
# --- Define the parameters for the two hyperbolas ---
# Hyperbola 1: 4(x^2 - y^2) - 12x + 8y = 0
V1 = np.array([[4, 0], [0, -4]])
u1 = np.array([-6, 4])
f1 = 0
# Hyperbola 2: -2(x^2 - y^2) - 10x + 20y = 0
V2 = np.array([[-2, 0], [0, 2]])
u2 = np.array([-5, 10])
f2 = 0
```

Python Code

```
# --- Set up the plotting grid ---
# Generate a grid of points to evaluate the equations on
x_vals = np.linspace(-10, 15, 500)
y_vals = np.linspace(-10, 15, 500)
X, Y = np.meshgrid(x_vals, y_vals)
# --- Define the hyperbola equations ---
# Equation is x^T V x + 2u^T x + f = 0
# For a point (x,y), the vector is [x, y]
# So x^T V x becomes V[0,0]*x^2 + V[1,1]*y^2
# and 2u^T x becomes 2*(u[0]*x + u[1]*y)
eq1 = V1[0,0]*X**2 + V1[1,1]*Y**2 + 2*(u1[0]*X + u1[1]*Y) + f1
eq2 = V2[0,0]*X**2 + V2[1,1]*Y**2 + 2*(u2[0]*X + u2[1]*Y) + f2
```

Python Code

```
# --- Create the Plot ---
plt.figure(figsize=(10, 10))
# Plot the hyperbolas by finding where their equations equal zero
plt.contour(X, Y, eq1, levels=[0], colors='red', linewidths=2)
plt.contour(X, Y, eq2, levels=[0], colors='blue', linewidths=2)
# --- Plot the Common Chord and Solution ---
# The common chord is 64x - 96y = 0, which simplifies to 2x - 3y = 0
# So, y = (2/3)x
plt.plot(x_vals, (2/3)*x_vals, 'g--', label='Common Chord: $2x - 3y = 0$')
# The solution point
solution_point = np.array([3, 2])
plt.plot(solution_point[0], solution_point[1], 'ko', markersize=10, label='Solution (3, 2)')
```

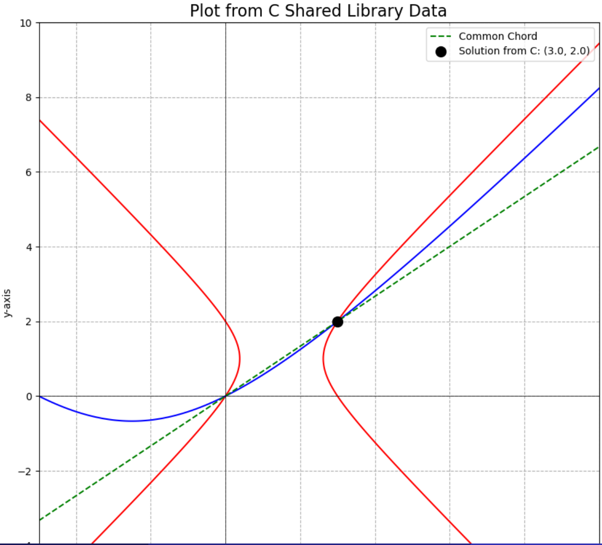
Python Code

```
plt.title('Intersection of Hyperbolas', fontsize=16)
plt.xlabel('x-axis', fontsize=12)
plt.ylabel('y-axis', fontsize=12)
plt.grid(True, which='both', linestyle='--', linewidth=0.5)
plt.axhline(0, color='black', linewidth=0.5)
plt.axvline(0, color='black', linewidth=0.5)
plt.gca().set_aspect('equal', adjustable='box')
plt.xlim(-5, 10)
plt.ylim(-5, 10)
```

Python Code

```python
# Create a legend
plt.legend(handles=[
plt.Line2D([0], [0], color='red', lw=2, label='Hyperbola 1'),
plt.Line2D([0], [0], color='blue', lw=2, label='Hyperbola 2'),
plt.Line2D([0], [0], color='g', linestyle='--', label='Common
    Chord'),
plt.Line2D([0], [0], marker='o', color='k', linestyle='',
    markersize=8, label='Solution (3, 2)')
])
# Save and show the plot
plt.savefig('hyperbola_intersection.png')
plt.show()
```

Plot by python using shared output from c



Plot by python