Area between two curves by using matrix method

Prajwal
EE24BTECH11051
Electrical Enggineering ,
IIT Hyderabad.

November 6, 2024

Problem Statement

Using integration, find the area of the region bounded by the parabolas $y^2 = 4x$ and $x^2 = 4y$.

Variable	Description
V_1, u_1, f_1	Parameters of Parabola
V_2, u_2, f_2	Parameters of Parabola
P_1, P_2	Points of intersection
A	Area between the conics

Table: Variables Used

Conic Parameters

The conic parameters of circle $y^2 = 4x$ are :

$$V_1 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, u_1 = \begin{pmatrix} -2 \\ 0 \end{pmatrix}, f_1 = 0$$

Conic parameters of parabola $x^2 = 4y$ can be expressed as :

$$V_2 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, u_2 = \begin{pmatrix} 0 \\ -2 \end{pmatrix}, f_2 = 0$$

Intersection of conics

The intersection of two conics with parameters $V_i, u_i, f_i (i = 1, 2)$ is defined as :

$$x^\top (V_1 + \mu V_2)x + 2(u_1 + \mu u_2)^\top x + (f_1 + \mu f_2) = 0$$

On solving we get the points of intersection are :

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 4 \\ 4 \end{pmatrix}$$

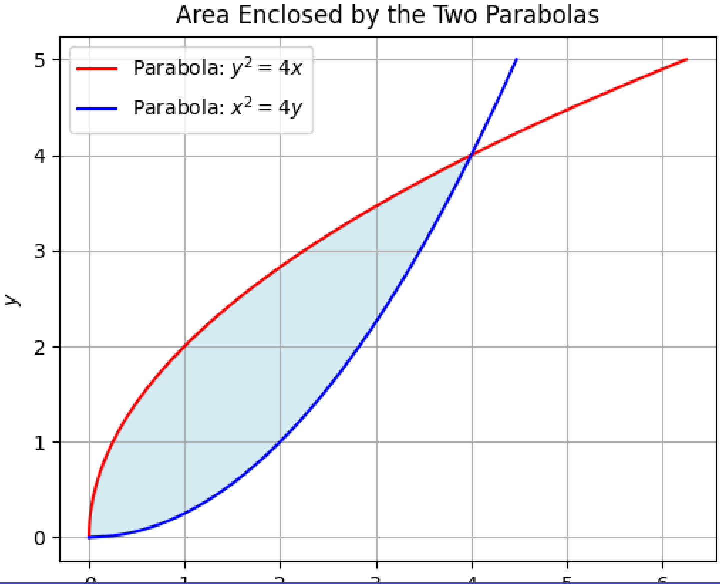
Area calculation

Area between the curves is,

$$2\int_0^4 \left(\sqrt{4x} - \frac{x^2}{4}\right)\, dy \tag{3.1}$$

By solving the integration, we get area is equal to 5.33 sq.units

Figure



Area Enclosed by the Two Parabolas
Parabola: $y^2 = 4x$
Parabola: $x^2 = 4y$
y
0
1
2
3
4
5

C Code

```
#include <stdio.h>

int main() {
    // Define parabola parameters
    double p1 = 1.0; // Parameter for y^2 = 4px
    double p2 = 1.0; // Parameter for x^2 = 4py

    // Open the file to write the parameters
    FILE *file = fopen("data.txt", "w");
    if (file == NULL) {
        printf("Error opening file!\n");
        return 1;
        }
```

```
    // Write the parabola parameters to the file
    fprintf(file, "%f\n", p1); // Parameter for the first parabola
    fprintf(file, "%f\n", p2); // Parameter for the second parabola

    // Close the file
    fclose(file);

    printf("Data written to data.txt\n");

    return 0;
}
```

Python Code for Plotting

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import quad
from scipy.optimize import fsolve

# Read the values from the C-generated text file using numpy.loadtxt
data = np.loadtxt('data.txt')

# Extracting parabola parameters
p1 = data[0] # Parameter for y^2 = 4x
p2 = data[1] # Parameter for x^2 = 4y

# Parabola equation: y^2 = 4px, so x = y^2 / (4p)
def parabola1(y, p):
    return y**2 / (4 * p)
```

```
def parabola2(x, p):
    return x**2 / (4 * p)

# Find the points of intersection between the two parabolas
def find_intersections(p1, p2):
    def intersection_eq(y):
        return parabola2(parabola1(y, p1), p2) - y # Solve for
            intersection

    y_int1 = fsolve(intersection_eq, 0)[0] # Initial guess
    y_int2 = fsolve(intersection_eq, 4)[0] # Initial guess for the other
        intersection
    return y_int1, y_int2

# Get the intersection points
y_int1, y_int2 = find_intersections(p1, p2)
```

```
# Compute the area between the curves using integration
def area_between_curves(y):
    x1 = parabola1(y, p1) # x from the first parabola
    x2 = 2 * np.sqrt(y) # From the second parabola (x = 2*sqrt(y))
    return x2 - x1 # Area between the two parabolas

# Perform the integration from y_int1 to y_int2
area, _ = quad(area_between_curves, y_int1, y_int2)




# Visualization
# Generating points for the parabolas
y_vals = np.linspace(0, y_int2 + 1, 400) # Extend range a bit above the
    highest intersection
x_parabola1 = parabola1(y_vals, p1)
x_parabola2 = 2 * np.sqrt(y_vals) # From the second parabola
```

```
# Plot the curves
plt.plot(x_parabola1, y_vals, label=r'Parabola: $y^2 = 4x$', color='r')
plt.plot(x_parabola2, y_vals, label=r'Parabola: $x^2 = 4y$', color='b')

# Fill the area between the curves
plt.fill_betweenx(y_vals, x_parabola1, x_parabola2, where=(x_parabola2
    >= x_parabola1), color='lightblue', alpha=0.5)

# Labels and plot settings
plt.xlabel('$x$')
plt.ylabel('$y$')
plt.title('Area Enclosed by the Two Parabolas')
plt.grid(True)
plt.legend()
```

```
# Set equal aspect ratio to avoid distortion
plt.gca().set_aspect('equal', adjustable='box')

# Show the plot
plt.show()
```