12.56

EE25BTECH11043 - Nishid Khandagre

October 9, 2025

Question

The area enclosed between the curves $y^2 = 4x$ and $x^2 = 4y$ is

Theoretical Solution

The equation of a parabola in Matrix form is

$$\mathbf{x}^\top \mathbf{V}\mathbf{x} + 2\mathbf{u}^\top \mathbf{x} + f = 0 \tag{1}$$

For $y^2 = 4x$:

$$\mathbf{V_1} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \tag{2}$$

$$\mathbf{u_1} = -2\mathbf{e_1} = \begin{pmatrix} -2 \\ 0 \end{pmatrix} \tag{3}$$

$$f_1 = 0 \tag{4}$$

Theoretical Solution

For $x^2 = 4y$:

$$\mathbf{V_2} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \tag{5}$$

$$\mathbf{u_2} = -2\mathbf{e_2} = \begin{pmatrix} 0 \\ -2 \end{pmatrix} \tag{6}$$

$$f_2 = 0 \tag{7}$$

The intersection of two conics with parameters $\mathbf{V_i}$, $\mathbf{u_i}$, f_i, $i = 1, 2$ is defined as

$$\mathbf{X}^T (\mathbf{V}_1 + \mu \mathbf{V}_2)\mathbf{X} + 2(\mathbf{u_1} + \mu \mathbf{u_2})^T \mathbf{X} \ + \ (f_1 + \mu f_2) \ = \ 0 \tag{8}$$

Theoretical Solution

$$\implies \begin{vmatrix} \mathbf{V}_1 + \mu \mathbf{V}_2 & \mathbf{u}_1 + \mu \mathbf{u}_2 \\ (\mathbf{u}_1 + \mu \mathbf{u}_2)^{\mathrm{T}} & f_1 + \mu f_2 \end{vmatrix} = 0 \tag{9}$$

$$\implies \begin{vmatrix} \mu & 0 & -2 \\ 0 & 1 & -2\mu \\ -2 & -2\mu & 0 \end{vmatrix} = 0 \tag{10}$$

Theoretical Solution

$$\implies \begin{vmatrix} \mu & 0 & -2 \\ 0 & 1 & -2\mu \\ -2 & -2\mu & 0 \end{vmatrix} R_3 \leftrightarrow R_3 + \frac{2}{\mu} \times R_1 \begin{vmatrix} \mu & 0 & -2 \\ 0 & 1 & -2\mu \\ 0 & -2\mu & -\frac{4}{\mu} \end{vmatrix} \tag{11}$$

$$R_3 \leftrightarrow R_3 + 2\mu \times R_2 \begin{vmatrix} \mu & 0 & -2 \\ 0 & 1 & -2\mu \\ 0 & 0 & -(\frac{4}{\mu} + 4\mu^2) \end{vmatrix} = 0 \tag{12}$$

Theoretical Solution

$$\implies -(4+4\mu^3) = 0 \tag{13}$$

$$\implies \mu = -1 \tag{14}$$

Substituting the value of $\mu = -1$ in (8) we get points of intersection as

$$\mathbf{x_1} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{15}$$

$$\mathbf{x_2} = \begin{pmatrix} 4 \\ 4 \end{pmatrix} \tag{16}$$

Theoretical Solution

Area of the desired region is given by

$$A = \int_0^4 \left(2\sqrt{x} - \frac{x^2}{4}\right) dx \tag{17}$$

$$A = \left[\frac{4}{3}x^{3/2} - \frac{x^3}{12}\right]_0^4 \tag{18}$$

Theoretical Solution

$$A = \left(\frac{4}{3}(4)^{3/2} - \frac{(4)^3}{12}\right) - (0 - 0) \tag{19}$$

$$A = \left(\frac{4}{3}(8) - \frac{64}{12}\right) \tag{20}$$

$$A = \left(\frac{32}{3} - \frac{16}{3}\right) \tag{21}$$

$$A = \frac{16}{3} \tag{22}$$

Thus, the area enclosed between the curves is $\frac{16}{3}$.

C Code

```c
#include <math.h>

// Function to calculate the area between y^2 = 4x and x^2 = 4y
double calculateEnclosedArea() {
    // The intersection points are (0,0) and (4,4)
    // The upper curve is y = 2*sqrt(x)
    // The lower curve is y = x*x / 4
    // Integral of (2*sqrt(x) - x*x / 4) from 0 to 4
    // Integral of 2*x^(1/2) is 2 * (x^(3/2) / (3/2)) = (4/3)*x
        ^(3/2)
    // Integral of x^2 / 4 is (1/4) * (x^3 / 3) = x^3 / 12
```

C Code

```c
    // Evaluate at x=4:
    double upper_at_4 = (4.0/3.0) * pow(4.0, 1.5); // (4/3) * 8 =
        32/3
    double lower_at_4 = pow(4.0, 3.0) / 12.0; // 64 / 12 = 16/3

    // Evaluate at x=0 (both are 0)

    return upper_at_4 - lower_at_4; // 32/3 - 16/3 = 16/3
}
```

Python Code (using C shared library)

```
import ctypes
import numpy as np
import matplotlib.pyplot as plt

lib_area = ctypes.CDLL(/Users/nishidkhandagre/matgeo/venv/bin/
    code19.so)

# Define the argument types and return type for the C function
lib_area.calculateEnclosedArea.argtypes = []
lib_area.calculateEnclosedArea.restype = ctypes.c_double

# Call the C function to get the enclosed area
area_result = lib_area.calculateEnclosedArea()

print(fThe area enclosed between the curves y^2 = 4x and x^2 = 4y
     is: {area_result:.4f})
```

Python Code (using C shared library)

```
# Generate points for plotting the curves
x_parabola = np.linspace(0, 5, 400)
x_other_curve = np.linspace(0, 5, 400)

# Curve 1: y^2 = 4x => y = +/- 2*sqrt(x)
y_upper_parabola = 2 * np.sqrt(x_parabola)
y_lower_parabola = -2 * np.sqrt(x_parabola)

# Curve 2: x^2 = 4y => y = x^2 / 4
y_other_curve = x_other_curve**2 / 4

# Plotting
plt.figure(figsize=(8, 8))

# Plot y^2 = 4x as a complete parabola
plt.plot(x_parabola, y_upper_parabola, 'b-', label=r'$y^2 = 4x$')
plt.plot(x_parabola, y_lower_parabola, 'b-')
```

Python Code (using C shared library)

```python
# Plot x^2 = 4y
plt.plot(x_other_curve, y_other_curve, 'r-', label=r'$x^2 = 4y$')

# Fill the enclosed area
x_fill = np.linspace(0, 4, 100)
y_upper_fill = 2 * np.sqrt(x_fill)
y_lower_fill = x_fill**2 / 4
plt.fill_between(x_fill, y_lower_fill, y_upper_fill, color='
    lightgray', alpha=0.5, label='Enclosed Area')

# Intersection points
plt.scatter([0, 4], [0, 4], color='green', s=100, zorder=5, label
    ='Intersection Points')
plt.annotate('(0,0)', (0, 0), textcoords=offset points, xytext
    =(5,5), ha='left')
plt.annotate('(4,4)', (4, 4), textcoords=offset points, xytext
    =(5,5), ha='left')
```

Python Code (using C shared library)

```
plt.gca().set_aspect('equal', adjustable='box')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title(f'Area Enclosed Between $y^2=4x$ and $x^2=4y$')
plt.grid(True)
plt.legend()
plt.ylim(-5, 5)
plt.xlim(-0.5, 5.5)
plt.show()
```

Python Code (Direct)

```
import numpy as np
import matplotlib.pyplot as plt

# Define the functions for the two curves
def curve1_y_positive(x):
    return 2 * np.sqrt(x)

def curve1_y_negative(x):
    return -2 * np.sqrt(x)

def curve2_y(x):
    return x**2 / 4

# Define the range for x
x_parabola1 = np.linspace(0, 10, 400)
x_parabola2 = np.linspace(-6, 6, 400)
```

Python Code (Direct)

```
# Calculate y values for each curve
y1_positive = curve1_y_positive(x_parabola1)
y1_negative = curve1_y_negative(x_parabola1)
y2 = curve2_y(x_parabola2)

# Plotting the curves
plt.figure(figsize=(9, 7))

# Plot y^2 = 4x (full parabola)
plt.plot(x_parabola1, y1_positive, color='blue', label='$y^2 = 4
    x$ (upper part)')
plt.plot(x_parabola1, y1_negative, color='blue', linestyle='--',
    label='$y^2 = 4x$ (lower part)')

# Plot x^2 = 4y (full parabola)
plt.plot(x_parabola2, y2, color='green', label='$x^2 = 4y$')
```

Python Code (Direct)

```python
# Shade the enclosed area between (0,0) and (4,4)
x_fill = np.linspace(0, 4, 100)
y_upper_bound = 2 * np.sqrt(x_fill)
y_lower_bound = x_fill**2 / 4
plt.fill_between(x_fill, y_upper_bound, y_lower_bound, color='
    lightgray', alpha=0.5, label='Enclosed Area')

# Add intersection points
intersection_x = [0, 4]
intersection_y = [0, 4]
plt.scatter(intersection_x, intersection_y, color='red', zorder
    =5, label='Intersection Points')
plt.text(0.1, 0.1, '(0,0)', fontsize=10, verticalalignment='
    bottom')
plt.text(4.2, 4.2, '(4,4)', fontsize=10, verticalalignment='
    bottom')
```

Python Code (Direct)

```
# Add labels and title
plt.xlabel('$x$')
plt.ylabel('$y$')
plt.title('Full Parabolas and Enclosed Area')
plt.legend()
plt.grid(True)
plt.axhline(0, color='black',linewidth=0.5)
plt.axvline(0, color='black',linewidth=0.5)
plt.xlim(-2, 11)
plt.ylim(-7, 7)
plt.gca().set_aspect('equal', adjustable='box')

# Show the plot
plt.savefig(full_parabolas_enclosed_area.png)
plt.show()

print(Figure saved as full_parabolas_enclosed_area.png)
print(fThe calculated enclosed area is: {16/3:.2f} square units)
```

Plot by Python using shared output from C

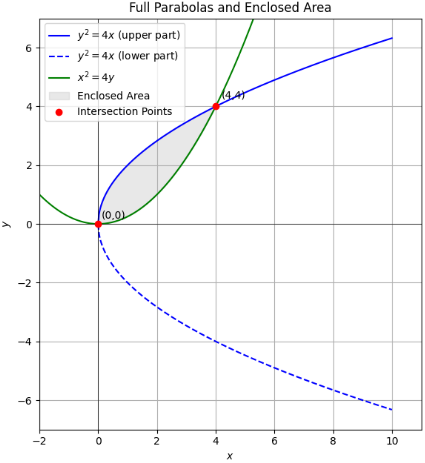

Figure:

Plot by Python only

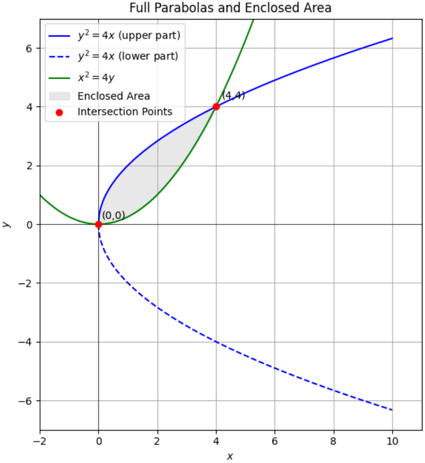

Figure: