

Question-9-9.3-13

EE24BTECH11030 - J.KEDARANANDA

Question

Find the area bounded by the ellipse $x^2 + 4y^2 = 16$ and the ordinates $x = 0$ and $x = 2$.

Inputs

Variable	Description
$\vec{x}_1, \vec{x}_2, \vec{x}_3, \vec{x}_4$	Intersection points
\vec{h}	Point on the given line
\vec{m}	Direction vector of given line
A	Area of the region

Formulas

Conic	Expression
ellipse	$\vec{x}^\top \vec{V} \vec{x} + 2\vec{u}^\top \vec{x} + f = 0$
line	$\vec{x} = \vec{h} + \kappa \vec{m}$

Solution

$$\vec{V} = \begin{pmatrix} \frac{1}{16} & 0 \\ 0 & \frac{1}{4} \end{pmatrix} \quad (1)$$

$$\vec{u} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (2)$$

$$f = -1 \quad (3)$$

For the given line $x = 2$, the values of \vec{h} and \vec{m} are:

$$\vec{h} = \begin{pmatrix} 2 \\ 0 \end{pmatrix} \quad (4)$$

$$\vec{m} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (5)$$

Solution

The intersection points are:

$$\vec{x}_1 = \begin{pmatrix} 2 \\ \sqrt{3} \end{pmatrix} \quad (6)$$

$$\vec{x}_2 = \begin{pmatrix} 2 \\ -\sqrt{3} \end{pmatrix} \quad (7)$$

$$\vec{x}_3 = \begin{pmatrix} 0 \\ 2 \end{pmatrix} \quad (8)$$

$$\vec{x}_4 = \begin{pmatrix} 0 \\ -2 \end{pmatrix} \quad (9)$$

$$y = \pm \sqrt{4 - \frac{1}{4}x^2} \quad (10)$$

Solution

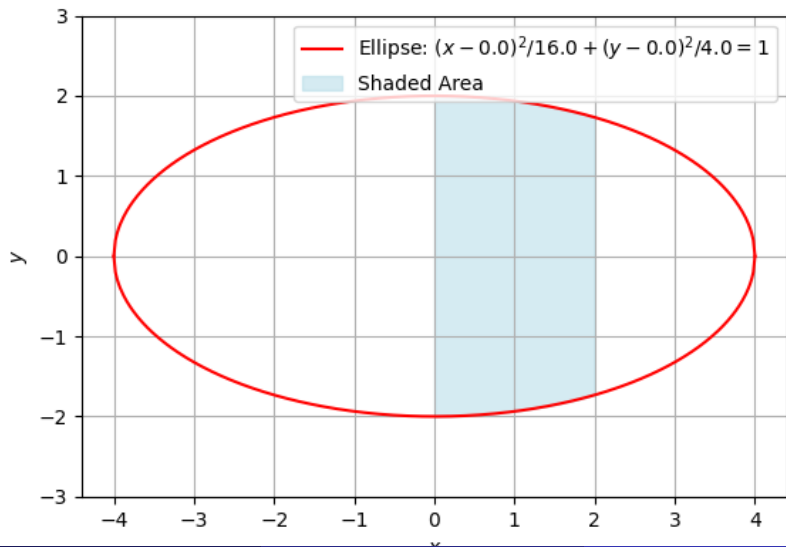
The area A between the curves from $x = 0$ to $x = 2$ is given by:

$$A = 4 \int_0^2 \sqrt{1 - \frac{x^2}{16}} dx \quad (11)$$

Simplifying, we get:

$$A = 2\sqrt{3} + \frac{4\pi}{3} \quad (12)$$

Diagram



C-Code

```
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/socket.h>
#include <netinet/in.h>

#include "libs/matfun.h"
#include "libs/geofun.h"

int main() {
    FILE *file;
    file = fopen("data.txt", "w");
```

```
        if (file == NULL) {
printf("Error opening file!\n");
return 1; // Return error code if file couldn't
        be opened
}
// Ellipse parameters
double a = 4.0; // semi-major axis
double b = 2.0; // semi-minor axis
// Store the parameters
fprintf(file, "%lf\n%lf\n%lf\n%lf\n", a, b, 0.0,
        0.0); // a, b, h, k
fclose(file); // Close the file
return 0; // Return success
}
```

```
# Code by GVV Sharma  
# Modified for Problem Solution  
# Released under GNU GPL  
# Calculating area enclosed between curves  
import sys # For path to external scripts  
sys.path.insert(0, '/home/kedar/assignments/matgeo/  
codes/CoordGeo') # Path to my scripts  
  
import numpy as np  
import matplotlib.pyplot as plt  
from scipy.integrate import quad
```

```
# Read the values from the C-generated text file using  
numpy.loadtxt  
try:  
    data = np.loadtxt('data.txt')  
except OSError:  
    print("Error: Unable to read 'data.txt'. Please  
        ensure the file exists.")  
    sys.exit(1)  
  
# Extracting ellipse parameters  
a = data[0]    # semi-major axis  
b = data[1]    # semi-minor axis  
h = data[2]    # center x-coordinate (h)  
k = data[3]    # center y-coordinate (k)
```

```
# Ellipse equations
def ellipse_upper(x, a, b, h, k):
    """Returns the upper part of the ellipse."""
    return k + b * np.sqrt(1 - ((x - h) * 2) / (a * 2)
        )

def ellipse_lower(x, a, b, h, k):
    """Returns the lower part of the ellipse."""
    return k - b * np.sqrt(1 - ((x - h) * 2) / (a * 2)
        )
```

Python-Code

```
# Define the limits of integration
x_min = 0
x_max = 2

# Compute the area between the curves using
    integration
def area_between_curves(x):
    """Calculates the difference between the upper and
        lower ellipse."""
    return ellipse_upper(x, a, b, h, k) -
        ellipse_lower(x, a, b, h, k)

# Perform the integration from x_min to x_max
area, _ = quad(area_between_curves, x_min, x_max)

print(f"Area enclosed between the ellipse and the
    ordinates: {area:.4f}")
```

```
# Visualization
# Generating points for the ellipse to show the
    complete ellipse
x_vals = np.linspace(h - a, h + a, 400)
y_upper = ellipse_upper(x_vals, a, b, h, k)
y_lower = ellipse_lower(x_vals, a, b, h, k)

# Plot the ellipse's upper and lower parts
plt.plot(x_vals, y_upper, label=f'Ellipse:  $(x - \{h\})^2/\{a^2\} + (y - \{k\})^2/\{b^2\} = 1$ ', color='r')
plt.plot(x_vals, y_lower, color='r')
```

```
# Shade the area between the ellipse and the x-axis
    between x=0 and x=2
y_fill_upper = ellipse_upper(np.linspace(x_min, x_max,
    100), a, b, h, k)
y_fill_lower = ellipse_lower(np.linspace(x_min, x_max,
    100), a, b, h, k)

# Fill the area
plt.fill_between(np.linspace(x_min, x_max, 100),
    y_fill_upper, y_fill_lower, color='lightblue',
    alpha=0.5, label='Shaded Area')

# Labels and plot settings
plt.xlabel('$x$')
plt.ylabel('$y$')
plt.grid(True)
plt.legend(loc='upper right')
```



```
# Set the limits for y-axis to ensure complete  
visibility of the ellipse  
plt.ylim(k - b - 1, k + b + 1) # Slightly adjust as  
needed for aesthetics  
  
# Set equal aspect ratio to avoid distortion  
plt.gca().set_aspect('equal', adjustable='box')  
  
# Show the plot  
plt.show()
```