MatGeo Presentation - Problem 10.7.8

EE25BTECH11064 - Yojit Manral

Question

The area (in sq. units) of the quadrilateral formed by the tangents at the end points of the latera recta to the ellipse $\frac{x^2}{9} + \frac{y^2}{5} = 1$ is

(a) $\frac{27}{2}$ (b) 27 (c) $\frac{27}{4}$ (d) 18

Solution

→ The parameters of the given conic are

$$\mathbf{V} = \begin{pmatrix} 5/9 & 0 \\ 0 & 1 \end{pmatrix}, \mathbf{u} = 0, \ f = -5 \tag{0.1}$$

→ Also, using eigenvalue decomposition

$$\mathbf{P} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \mathbf{D} = \begin{pmatrix} 5/9 & 0 \\ 0 & 1 \end{pmatrix} \implies \lambda_1 = 5/9 \text{ and } \lambda_2 = 1 \tag{0.2}$$

→ This gives us very useful information

$$e = \sqrt{1 - \frac{\lambda_1}{\lambda_2}} = \frac{2}{3} \tag{0.3}$$

$$\mathbf{F} = \pm e \sqrt{\frac{|\mathbf{u}^T \mathbf{V}^{-1} \mathbf{u} - f|}{\lambda_2 (1 - e^2)}} \mathbf{e_1} = \pm \begin{pmatrix} 2 \\ 0 \end{pmatrix} \tag{0.4}$$

→ Given that the points of contact are the end points of the latera recta. In the first quadrant, we have

$$\mathbf{q} = \begin{pmatrix} 2 \\ 5/3 \end{pmatrix} \tag{0.5}$$

Solution

$\rightarrow$ If the point of contact is $\mathbf{q}$, the equation of tangent to the conic is

$$(\mathbf{Vq}+\mathbf{u})^T\mathbf{x}+\mathbf{u}^T\mathbf{q}+f=0 \tag{0.6}$$

$$\implies \begin{pmatrix}2 & 3\end{pmatrix}\mathbf{x}=9 \tag{0.7}$$

$\rightarrow$ Thus, we get the area of the quadrilateral in the 1^{st} quadrant to be

$$\Delta_1 = \text{Area of } \triangle = \frac{1}{2}x_{int}y_{int} = \frac{1}{2}\times\frac{9}{2}\times 3 \tag{0.8}$$

$$\Delta_1 = \frac{27}{4}sq.units \tag{0.9}$$

$\rightarrow$ Since an ellipse has a two-fold symmetry, total area of the quadrilateral must be 4 times the area in the 1^{st} quadrant

$$\Delta_{Total} = 4\times\Delta_1 = 27sq.units \tag{0.10}$$

$\rightarrow$ Therefore, (b) 27 is the correct option

Solution

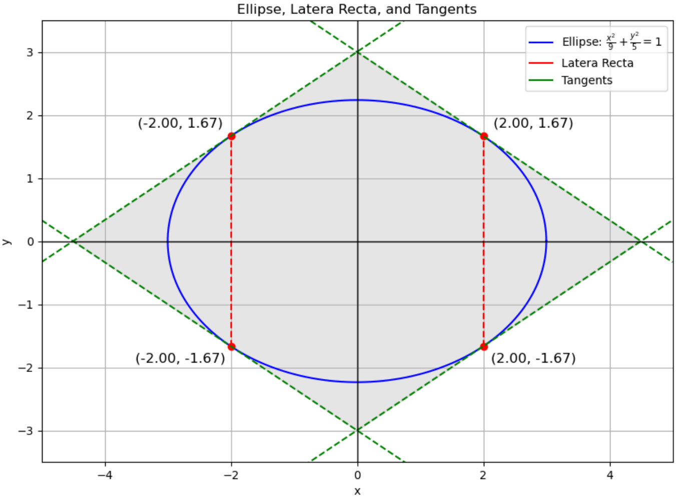

Figure: Plot of ellipse and quadrilateral

File: plot.py

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.lines import Line2D

# Ellipse parameters
a = 3 # semi-major axis
b = np.sqrt(5) # semi-minor axis

# Generate points for the ellipse
theta = np.linspace(0, 2 * np.pi, 400)
x = a * np.cos(theta)
y = b * np.sin(theta)

# Points for the latus rectum
latus_rectum_points = [(2, 5/3), (2, -5/3), (-2, 5/3), (-2, -5/3)]

# Tangent line equation function
def tangent_line(x1, y1):
    """Return the equation of the tangent line at (x1, y1) on the ellipse."""
    return lambda x: (1 - (x * x1 / a**2)) * (b**2 / y1)

# Set up the plot
fig, ax = plt.subplots(figsize=(8, 8))

# Plot the ellipse
ax.plot(x, y, color="blue")
ellipse = Line2D([0], [0], color='b')
```

File: plot.py

```
#Mark the Latera Recta
ax.plot([2, 2], [-5/3, 5/3], 'r--')
ax.plot([-2, -2], [-5/3, 5/3], 'r--')
latera_recta = Line2D([0], [0], color='r', linestyle='-')

# Mark and label the latus rectum points with two decimal places
for (x1, y1) in latus_rectum_points:
    ax.plot(x1, y1, 'ro') # Mark the point
    ax.text(x1+0.8*(x1/np.abs(x1)), y1+0.2*(y1/np.abs(y1)), f'({x1:.2f}, {y1:.2f})', fontsize=12, ha='
           center', va='center')

# Plot tangents at the endpoints of the latus rectum
for (x1, y1) in latus_rectum_points:
    # Get tangent function
    tangent = tangent_line(x1, y1)

    # Create x values for the tangent lines
    x_vals = np.linspace(-5, 5, 100)
    y_vals = tangent(x_vals) # Calculate corresponding y values

    # Plot the tangent lines
    ax.plot(x_vals, y_vals, 'g--')

    #Fill in the quadrilateral
    quad_x = np.linspace(0, 4.5, 90) if x1>0 else np.linspace(-4.5, 0, 90)
    quad_y1 = tangent(quad_x)
    plt.fill_between(quad_x, quad_y1, color='gray', alpha=0.2)
tangents = Line2D([0], [0], color='g', linestyle='-')
```

File: plot.py

```
# Set axes limits
ax.set_xlim(-5, 5)
ax.set_ylim(-3.5, 3.5)

# Add labels and title
ax.set_xlabel("x")
ax.set_ylabel("y")
ax.axhline(0, color='black', linewidth=1)
ax.axvline(0, color='black', linewidth=1)
ax.set_title("Ellipse, Latera Recta, and Tangents")

# Show grid
ax.grid(True)

# Show the plot
plt.legend(handles=[ellipse, latera_recta, tangents], labels=['Ellipse: $\\frac{x^2}{9} + \\frac{y^2}{5} =
      1$', 'Latera Recta', 'Tangents'])
plt.gca().set_aspect('equal', adjustable='box')
plt.show()
```