

Matgeo Q.9.2.26

Harshvardhan Patidar - AI24BTECH11015
Dept. of Artificial Intelligence

November 13, 2024

1 Problem

2 Solution

- Variables Used
- General equation of a Conic in Matrix form
- Parameters for given parabola
- Matrix Parameters of given Parabola
- Line Parameters
- Points of Intersection
- Calculating Area

3 Plot

4 Codes

- C code
- Python Code

Problem Statement

Find the area of the region included between $y^2 = 9x$ and $y = x$.

Variables Used

Variable	Description
e	Eccentricity of conic
\mathbf{F}	Focus of conic
\mathbf{I}	Identity matrix
$\mathbf{n}^\top \mathbf{x} = c$	Equation of directrix
\mathbf{n}	Slope of normal to directrix
\mathbf{V}	A symmetric matrix given by eigenvalue decomposition

Table: Variables

General equation of a Conic in Matrix Form

The general equation of a conic with directrix $\mathbf{n}^\top \mathbf{x} = c$, Focus \mathbf{F} and eccentricity e is given by

$$g(\mathbf{x}) = \mathbf{x}^\top \mathbf{V} \mathbf{x} + 2\mathbf{u}^\top \mathbf{x} + f = 0 \quad (3.1)$$

where,

$$\mathbf{V} = \|\mathbf{n}\|^2 \mathbf{I} - e^2 \mathbf{n} \mathbf{n}^\top \quad (3.2)$$

$$\mathbf{u} = ce^2 \mathbf{n} - \|\mathbf{n}\|^2 \mathbf{F} \quad (3.3)$$

$$f = \|\mathbf{n}\|^2 \|\mathbf{F}\|^2 - c^2 e^2 \quad (3.4)$$

Parameters for given parabola

For the parabola $y^2 = 9x$, and,

$$\text{directrix is } (-1 \ 0) \mathbf{x} = \frac{9}{4} \quad (3.5)$$

$$\text{Focus } \mathbf{F} = \begin{pmatrix} \frac{9}{4} \\ 0 \end{pmatrix} \quad (3.6)$$

$$\text{and, eccentricity } e = 1. \quad (3.7)$$

Matrix Parameters of given Parabola

We can now find \mathbf{V} , \mathbf{u} , and f to represent given parabola in the matrix equation form given in 3.1.

From 3.2, we have

$$\mathbf{V} = \mathbf{I} - \begin{pmatrix} -1 \\ 0 \end{pmatrix} \begin{pmatrix} -1 & 0 \end{pmatrix} \implies \mathbf{V} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \quad (3.8)$$

$$\mathbf{u} = \frac{9}{4} \begin{pmatrix} -1 \\ 0 \end{pmatrix} - \begin{pmatrix} \frac{9}{4} \\ 0 \end{pmatrix} \implies \mathbf{u} = \begin{pmatrix} -\frac{9}{2} \\ 0 \end{pmatrix} \quad (3.9)$$

$$f = \left(\frac{9}{4}\right)^2 - \left(\frac{9}{4}\right)^2 \implies f = 0 \quad (3.10)$$

Line Parameters

The given line $y = x$ can be represented in matrix form

$$\mathbf{x} = \mathbf{h} + \kappa \mathbf{m} \quad (3.11)$$

where

$$\mathbf{h} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (3.12)$$

$$\mathbf{m} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad (3.13)$$

Points of Intersection

The points of intersection of line and conic are given by

$$\mathbf{x} = \mathbf{h} + \kappa_i \mathbf{m} \quad (3.14)$$

where

$$\kappa_i = \frac{1}{\mathbf{m}^\top \mathbf{V} \mathbf{m}} \left(-\mathbf{m}^\top (\mathbf{V} \mathbf{h} + \mathbf{u}) \pm \sqrt{[\mathbf{m}^\top (\mathbf{V} \mathbf{h} + \mathbf{u})]^2 - g(\mathbf{h}) (\mathbf{m}^\top \mathbf{V} \mathbf{m})} \right) \quad (3.15)$$

On putting values in 3.15, we get

$$\kappa_i = 0, 9 \quad (3.16)$$

Using 3.16 in 3.14, we get points of intersection as $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$ and $\begin{pmatrix} 9 \\ 9 \end{pmatrix}$

Calculating Area

The area between the line and the parabola is given by

$$\begin{aligned}\int_0^9 3\sqrt{x} \, dx - \int_0^9 x \, dx &= \left(2(9)^{3/2} - 2(0)^{3/2}\right) - \left(\frac{(9)^2}{2} - \frac{(0)^2}{2}\right) \\ &= (2 \cdot 27 - 0) - \left(\frac{81}{2} - 0\right) \\ &= \frac{27}{2}\end{aligned}\tag{3.17}$$

So, the area between the given parabola $y^2 = 9x$ and the line $y = x$ is $\frac{27}{2}$.

Plot (using Python)

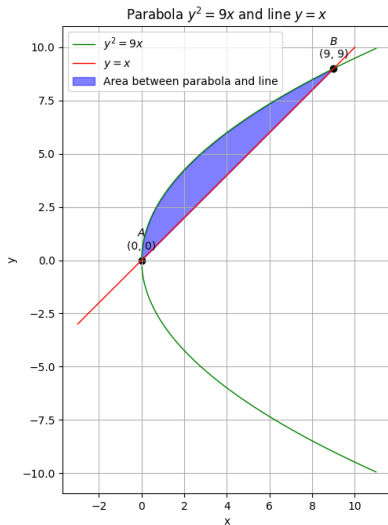


Figure: Area between $y^2 = 9x$ and $y = x$

C code to produce data I

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h>
4  #include "libs/matfun.h"
5  #include "libs/geofun.h"
6
7  void horizontal_parabola_gen(FILE *fptr, double a, double num_points,
↪  double ** vertex) {
8      double h = vertex[0][0];
9      double k = vertex[1][0];
10
11     for (int i = 0; i <= num_points; i++) {
12         double x = (11.0 * i) / num_points; // Scale x from 0 to 11
13         double y_pos = k + sqrt(4 * a * (x-h)); // Positive y value
14         double y_neg = k - sqrt(4 * a * (x-h)); // Negative y value
15
16         // Output the points for the upper and lower curves
17         fprintf(fptr, "%lf,%lf\n", x, y_pos); // Upper
18         fprintf(fptr, "%lf,%lf\n", x, y_neg); // Lower
19     }
```

C code to produce data II

```
20 }
21
22
23 int main() {
24     double x1, y1;
25     x1 = 0; y1 = 0; // Vertex of the parabola at the origin
26     int m = 2, n = 1;
27     double **vertex = createMat(m, n);
28     vertex[0][0] = x1;
29     vertex[1][0] = y1;
30
31     FILE *fptr;
32     fptr = fopen("points.txt", "w");
33     if (fptr == NULL) {
34         printf("Error opening file!\n");
35         return 1;
36     }
37
38     double a = 9.0/4.0; // For  $y^2 = 9x$ , we have  $4a = 9$ , thus  $a = 9/4$ 
39     horizontal_parabola_gen(fptr, a, 1000, vertex);
```

C code to produce data III

```
40  
41     fclose(fptr);  
42     freeMat(vertex, 2); // Freeing the dynamically allocated memory  
43     return 0;  
44 }
```

Python code to Plot curves I

```
1 import sys # for path to external scripts
2 sys.path.insert(0, '/Users/hanumac/Desktop/github/matgeo/codes/coordgeo')
  ↪ # path to my scripts
3 import numpy as np
4 import numpy.linalg as LA
5 import matplotlib.pyplot as plt
6
7 # local imports
8 from line.funcs import *
9 from triangle.funcs import *
10 from conics.funcs import circ_gen
11
12 # Load the points from the text file generated by the C code
13 points = np.loadtxt("points.txt", delimiter=',')
14
15 # Extract the x and y coordinates
16 x = points[:, 0]
17 y = points[:, 1]
18
19 # Separate the positive and negative branches of the parabola
```

Python code to Plot curves II

```
20 x_positive = x[y >= 0] # X values where y is positive
21 y_positive = y[y >= 0] # Positive Y values
22
23 x_negative = x[y < 0] # X values where y is negative
24 y_negative = y[y < 0] # Negative Y values
25
26 # line parameters
27 A = np.array([0, 0]) # point A
28 B = np.array([9, 9]) # point B
29 m = np.array([1, 1]).reshape(-1, 1) # direction vector of line
30
31 # generate line points
32 line_points = line_dir_pt(m, A.reshape(-1, 1), 10, -3)
33
34 # Plot
35 plt.figure()
36
37 # Plot the positive branch of the parabola
38 plt.plot(x_positive, y_positive, label=r'$y^2 = 9x$', color='green',
  ↪ linewidth=1)
```


Python code to Plot curves III

```
39
40 # Plot the negative branch of the parabola
41 plt.plot(x_negative, y_negative, color='green', linewidth=1)
42
43 plt.gca().set_aspect('equal', adjustable='box')
44
45 # Plot the line y = x
46 plt.plot(line_points[0, :], line_points[1, :], label="$ y = x $",
47 ↪ color="red", linewidth=1)
48
49 # Fill the area between the parabola and the line from x=0 to x=9
50 x_fill = np.linspace(0, 9, 500) # X values between 0 and 9
51 y_parabola = np.sqrt(9 * x_fill) # Y values for the positive branch of the
52 ↪ parabola
53 y_line = x_fill # Y values for the line y = x
54
55 plt.fill_between(x_fill, y_line, y_parabola, color='blue', alpha=0.5,
56 ↪ label='Area between parabola and line')
57
58 # Creating coordinates for labeling
```

Python code to Plot curves IV

```
56 tri_coords = np.array([[A[0], B[0]], [A[1], B[1]]]) #array for points
57 plt.scatter(tri_coords[0, :], tri_coords[1, :], color='black') #Plot
   ↪ points
58 vert_labels = ['$A$', '$B$']
59
60 # Annotate the points using the provided syntax
61 for i, txt in enumerate(vert_labels):
62     plt.annotate(f'{txt}\n({tri_coords[0,i]:.0f}, {tri_coords[1,i]:.0f})',
63                 (tri_coords[0,i], tri_coords[1,i]), # point to label
64                 textcoords="offset points", # position of text
65                 xytext=(0, 10), # distance from text to points (x,y)
66                 ha='center') # horizontal alignment
67
68 # Label the axes
69 plt.xlabel("x")
70 plt.ylabel("y")
71 plt.title("Parabola  $y^2 = 9x$  and line  $y=x$ ")
72 plt.grid(True)
73 plt.legend()
74 plt.show()
```