# Matgeo Q.9.2.26

Harshvardhan Patidar - AI24BTECH11015
Dept. of Artificial Intelligence

November 13, 2024

# 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

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

where,

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

$$\mathbf{u} = ce^2 \mathbf{n} - \|\mathbf{n}\|^2 \mathbf{F} \tag{3.3}$$

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

# Parameters for given parabola

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

$$\text{directrix is } \begin{pmatrix} -1 & 0 \end{pmatrix} \mathbf{x} = \frac{9}{4} \tag{3.5}$$

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

$$\text{and, eccentricity } e = 1. \tag{3.7}$$

## 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

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

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

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

# Line Parameters

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

$$\mathbf{x} = \mathbf{h} + \kappa\mathbf{m} \tag{3.11}$$

where

$$\mathbf{h} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{3.12}$$

$$\mathbf{m} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \tag{3.13}$$

# Points of Intersection

The points of intersection of line and conic are given by

$$\mathbf{x} = \mathbf{h} + \kappa_i \mathbf{m} \tag{3.14}$$

where

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

On putting values in 3.15, we get

$$\kappa_i = 0, 9 \tag{3.16}$$

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)

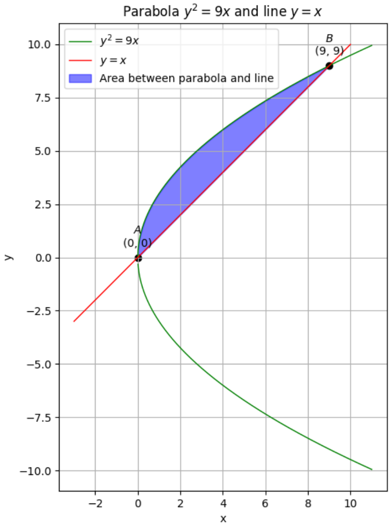


Figure: Area between $y^2 = 9x$ and $y = x$

# C code to produce data I

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "libs/matfun.h"
#include "libs/geofun.h"

void horizontal_parabola_gen(FILE *fptr, double a, double num_points, double ** vertex) {
  double h = vertex[0][0];
  double k = vertex[1][0];

    for (int i = 0; i <= num_points; i++) {
        double x = (11.0 * i) / num_points; // Scale x from 0 to 11
        double y_pos = k + sqrt(4 * a * (x-h)); // Positive y value
        double y_neg = k - sqrt(4 * a * (x-h)); // Negative y value

        // Output the points for the upper and lower curves
        fprintf(fptr, "%lf,%lf\n", x, y_pos); // Upper
        fprintf(fptr, "%lf,%lf\n", x, y_neg); // Lower
    }
```

# C code to produce data II

```c
}


int main() {
    double x1, y1;
    x1 = 0; y1 = 0;  // Vertex of the parabola at the origin
    int m = 2, n = 1;
    double **vertex = createMat(m, n);
    vertex[0][0] = x1;
    vertex[1][0] = y1;

    FILE *fptr;
    fptr = fopen("points.txt", "w");
    if (fptr == NULL) {
        printf("Error opening file!\n");
        return 1;
    }

    double a = 9.0/4.0;  // For y^2 = 9x, we have 4a = 9, thus a = 9/4
    horizontal_parabola_gen(fptr, a, 1000, vertex);
```

# C code to produce data III

```

    fclose(fptr);
    freeMat(vertex, 2);  // Freeing the dynamically allocated memory
    return 0;
}
```

# Python code to Plot curves I

```
import sys  # for path to external scripts
sys.path.insert(0, '/Users/hanumac/Desktop/github/matgeo/codes/coordgeo')
↪  # path to GVV Sir's scripts
import numpy as np
import numpy.linalg as LA
import matplotlib.pyplot as plt

# local imports
from line.funcs import *
from triangle.funcs import *
from conics.funcs import circ_gen

# Load the points from the text file generated by the C code
points = np.loadtxt("points.txt", delimiter=',')

# Extract the x and y coordinates
x = points[:, 0]
y = points[:, 1]

# Separate the positive and negative branches of the parabola
```

# Python code to Plot curves II

```
x_positive = x[y >= 0]  # X values where y is positive
y_positive = y[y >= 0]  # Positive Y values

x_negative = x[y < 0]   # X values where y is negative
y_negative = y[y < 0]   # Negative Y values

# line parameters
A = np.array(([0, 0]))  # point A
B = np.array(([9, 9]))  # point B
m = np.array(([1, 1])).reshape(-1, 1)  # direction vector of line

# generate line points
line_points = line_dir_pt(m, A.reshape(-1, 1), 10, -3)

# Plot
plt.figure()

# Plot the positive branch of the parabola
plt.plot(x_positive, y_positive, label=r'$y^2 = 9x$', color='green',
↪  linewidth=1)
```

# Python code to Plot curves III

```

# Plot the negative branch of the parabola
plt.plot(x_negative, y_negative, color='green', linewidth=1)

plt.gca().set_aspect('equal', adjustable='box')

# Plot the line y = x
plt.plot(line_points[0, :], line_points[1, :], label="$ y = x $", color="red", linewidth=1)

# Fill the area between the parabola and the line from x=0 to x=9
x_fill = np.linspace(0, 9, 500)  # X values between 0 and 9
y_parabola = np.sqrt(9 * x_fill)  # Y values for the positive branch of the parabola
y_line = x_fill  # Y values for the line y = x

plt.fill_between(x_fill, y_line, y_parabola, color='blue', alpha=0.5, label='Area between parabola and line')

# Creating coordinates for labeling
```

# Python code to Plot curves IV

```
tri_coords = np.array([[A[0], B[0]], [A[1], B[1]]])  #array for points
plt.scatter(tri_coords[0, :], tri_coords[1, :], color='black') #Plot points
vert_labels = ['$A$', '$B$']

# Annotate the points using the provided syntax
for i, txt in enumerate(vert_labels):
    plt.annotate(f'{txt}\n({tri_coords[0,i]:.0f}, {tri_coords[1,i]:.0f})',
                 (tri_coords[0,i], tri_coords[1,i]),  # point to label
                 textcoords="offset points",  # position of text
                 xytext=(0, 10),  # distance from text to points (x,y)
                 ha='center')  # horizontal alignment

# Label the axes
plt.xlabel("x")
plt.ylabel("y")
plt.title("Parabola $y^2 = 9x$ and line $y=x$")
plt.grid(True)
plt.legend()
plt.show()
```