

8.2.3

Naman Kumar-EE25BTECH11041

1 Oct,2025

Question)

$$y^2 = -8x \quad (1)$$

Solution

Since it is a parabola we have

Eccentricity	$e=1$
Eigenvalue	$\lambda_1 = 0$
Determinant	$ V = 0$

General equation of conic

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

Solution

For the equation (1), we can write

$$\mathbf{x}^T \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \mathbf{x} + 2 \begin{pmatrix} 4 \\ 0 \end{pmatrix}^T \mathbf{x} = 0 \quad (3)$$

v	$\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$
u	$4e_1$
f	0

Solution

using general equations we know for any conic

$$\mathbf{V} = \|\mathbf{n}\|^2 \mathbf{I} - e^2 \mathbf{n} \mathbf{n}^T \quad (4)$$

$$\mathbf{u} = c e^2 \mathbf{n} - \|\mathbf{n}\|^2 \mathbf{F} \quad (5)$$

$$f = \|\mathbf{n}\|^2 \|\mathbf{F}\|^2 - c^2 e^2 \quad (6)$$

Let

\mathbf{n}	$\begin{pmatrix} a \\ b \end{pmatrix}$	normal vector to directrix
\mathbf{F}	$\begin{pmatrix} g \\ h \end{pmatrix}$	focus
c		be constant of directrix

Solution

Firstly in (4)

$$\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} = \mathbf{n}^T \mathbf{n} \mathbf{l} - (1)^2 \mathbf{n} \mathbf{n}^T \quad (7)$$

$$\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} a^2 + b^2 & 0 \\ 0 & a^2 + b^2 \end{pmatrix} - \begin{pmatrix} a^2 & ab \\ ab & b^2 \end{pmatrix} \quad (8)$$

$$\mathbf{n} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \mathbf{e}_1 \quad (9)$$

In (5)

$$4\mathbf{e}_1 = c(1)^2\mathbf{e}_1 - (1)^2 \begin{pmatrix} g \\ h \end{pmatrix} \quad (10)$$

$$\begin{pmatrix} g \\ h \end{pmatrix} = (c - 4)\mathbf{e}_1 = (c - 4) \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad (11)$$

$$\mathbf{F} = \begin{pmatrix} c - 4 \\ 0 \end{pmatrix} \quad (12)$$

In (6)

$$0 = (1)^2 \begin{pmatrix} c-4 \\ 0 \end{pmatrix}^T \begin{pmatrix} c-4 \\ 0 \end{pmatrix} - c^2(1) \quad (13)$$

$$c^2 + 16 - 8c - c^2 = 0 \quad (14)$$

$$c = 2 \quad (15)$$

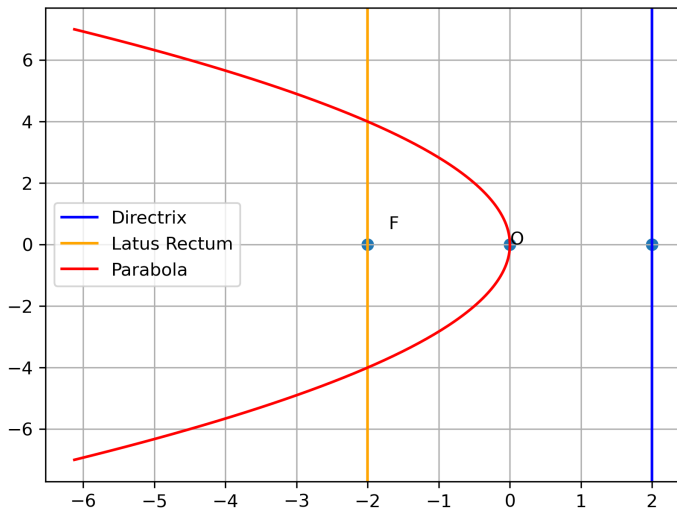
$$\mathbf{F} = -2\mathbf{e}_1 \quad (16)$$

Directrix is

$$\mathbf{n}^T \mathbf{x} = c \quad (17)$$

$$\mathbf{e}_1^T \mathbf{x} = 2 \quad (18)$$

Figure



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

y = np.linspace(-7,7,200)
x = -y*y/8

plt.axvline(x=2, color='blue', label="Directrix")
plt.axvline(x=-2, color='orange', label="Latus Rectum")

xp = np.array([0,2,-2])
yp = np.array([0,0,0])
plt.scatter(xp,yp)
```

```
plt.annotate('O', xy=(0, 0))
plt.annotate('F', xy=(-2, 0), xytext=(-1.7,0.5))

plt.plot(x,y, label="Parabola", color='red')
plt.grid()
plt.legend()
plt.savefig("figure.png", dpi=300)
plt.show()
```

```
// main.c
#include <stdio.h>
#include <math.h>
void generate_points(const char *filename, int n) {
    FILE *fp = fopen(filename, "w");
    if (!fp) {
        printf("Error opening file!\n");
        return;
    }

    double xmin = -10.0; // starting x
    double xmax = 0.0; // parabola defined for x <= 0
```

```
double step = (xmax - xmin) / n;

for (int i = 0; i <= n; i++) {
    double x = xmin + i * step;
    double y_val = sqrt(-8.0 * x);

    fprintf(fp, "%lf %lf\n", x, y_val);
    fprintf(fp, "%lf %lf\n", x, -y_val);
}

fclose(fp);
}

int main() {
    generate_points("points.txt", 200); // generate 200 steps
    printf("Points written to points.txt\n");
    return 0;
}
```

```
#ifdef TEST_C
int main(){
    SolutionSet S = solve_equations();
    for(int i=0; i<S.count; i++){
        printf("Solution %d: (%.2f, %.2f)\n", i+1, S.sols[i].x, S
            .sols[i].y);
    }
    return 0;
}
#endif
```

Python code with shared object

```
# main.py
import matplotlib.pyplot as plt

# Read points from file
x_vals = []
y_vals = []

with open("points.txt", "r") as f:
    for line in f:
        x, y = map(float, line.strip().split())
        x_vals.append(x)
        y_vals.append(y)
```


Python code with shared object

```
# Plot parabola
plt.figure(figsize=(6,6))
plt.plot(x_vals, y_vals, 'b.', markersize=2, label=r"$y^2=-8x$")
plt.axhline(0, color='k', linewidth=0.8)
plt.axvline(0, color='k', linewidth=0.8)
plt.xlabel("x-axis")
plt.ylabel("y-axis")
plt.title("Parabola:  $y^2 = -8x$ ")
plt.legend()
plt.grid(True)
plt.axis("equal")
plt.show()
```