9.4.39

Harsha-EE25BTECH11026

September 10,2025

Question

The difference of squares of two numbers is 180. The square of the smaller number is 8 times the larger number. Find the two numbers.

Theoretical Solution

Let x and y be the 2 numbers such that $x > y$.
The given equations are,

$$x^2 - y^2 = 180 \tag{1}$$

$$y^2 = 8x \tag{2}$$

As the given equations are homogeneous, converting them into quadratic form,

$$\implies \mathbf{x}^\top \mathbf{F_1} \mathbf{x} + c = 0 \tag{3}$$

where $\mathbf{x}^\top = \begin{pmatrix} x & y \end{pmatrix}$ and $\mathbf{F_1} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ and $c = -180$

Theoretical Solution

$$\mathbf{x}^\top \mathbf{F_2} \mathbf{x} + 2\mathbf{b}^\top \mathbf{x} = 0 \tag{4}$$

where $\mathbf{x}^\top = \begin{pmatrix} x & y \end{pmatrix}^\top$, $\mathbf{F_2} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$ and $\mathbf{b} = \begin{pmatrix} -4 \\ 0 \end{pmatrix}$.
To identify the intersection of conics, we can employ the approach of degenerating the conics.

To work with degeneracy in matrix form we form the standard augmented 3×3 matrix for each conic:

$$\mathbf{M_i} = \begin{pmatrix} \mathbf{F_i} & \mathbf{b_i} \\ \mathbf{b_i}^\top & c_i \end{pmatrix} \tag{5}$$

$$\implies \mathbf{M_1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -180 \end{pmatrix} \quad \mathbf{M_2} = \begin{pmatrix} 0 & 0 & -4 \\ 0 & 1 & 0 \\ -4 & 0 & 0 \end{pmatrix} \tag{6}$$

$$\therefore \mathbf{x}^\top \left(\mathbf{M_1} + \lambda \mathbf{M_2}\right) \mathbf{x} = 0 \tag{7}$$

To degenerate the conic into a line, we can find the solutions of λ when $\|\mathbf{M_1} + \lambda \mathbf{M_2}\| = 0$

$$\therefore \|\mathbf{M_1} + \lambda \mathbf{M_2}\| = 0 \tag{8}$$

$$\implies (\lambda - 1)\left(4\lambda^2 + 45\right) = 0 \tag{9}$$

$$\therefore \lambda = 1 \tag{10}$$

Theoretical Solution

Substituting λ in the equation,

$$\mathbf{x}^\top (\mathbf{M_1} + \mathbf{M_2})\,\mathbf{x} \tag{11}$$

$$\implies x^2 - 8x - 180 = 0 \tag{12}$$

$$\implies x = 18, -10 \tag{13}$$

for $x = -10$, there is no real solution of y,

$$\implies y = \pm 12 \tag{14}$$

$$\therefore \text{The two numbers are } (18, 12) \; and \; (18, -12) \tag{15}$$

C Code -Finding the intersection of conics

```
#include <stdio.h>
#include <math.h>

void solve_conics(double *results) {
    // Quadratic in x: x^2 - 8x - 180 = 0
    double a = 1, b = -8, c = -180;
    double disc = b*b - 4*a*c;

    if (disc < 0) {
        // No real solution
        results[0] = results[1] = results[2] = results[3] = NAN;
        return;
    }

    // Roots of quadratic
    double sqrt_disc = sqrt(disc);
    double x1 = (-b + sqrt_disc) / (2*a);
    double x2 = (-b - sqrt_disc) / (2*a);
```

C Code -Finding the intersection of conics

```
    // Solutions
    int idx = 0;
    double xs[2] = {x1, x2};
    for (int i=0; i<2; i++) {
        double x = xs[i];
        if (x < 0) continue; // y^2 = 8x requires x >= 0
        double y2 = 8*x;
        double y = sqrt(y2);
        results[idx++] = x;
        results[idx++] = y;
        results[idx++] = x;
        results[idx++] = -y;
    }

    while (idx < 4) {
        results[idx++] = NAN;
    }
}
```

Python+C code

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

# Load C shared library
lib = ctypes.CDLL("./libconics.so")

# Define return type
lib.solve_conics.argtypes = [ctypes.POINTER(ctypes.c_double)]

# Prepare results array
results = (ctypes.c_double * 4)()
lib.solve_conics(results)

vals = list(results)
points = [(vals[i], vals[i+1]) for i in range(0, len(vals), 2) if
    not np.isnan(vals[i])]
print("Solutions :", points)
```

Python+C code

```
# PLOT
fig, ax = plt.subplots(figsize=(6,6))
ax.set_xlabel("x")
ax.set_ylabel("y")
ax.set_title("Intersection of Hyperbola and Parabola")
ax.grid(True)

# Hyperbola: x^2 - y^2 = 180 -> y = sqrt(x^2 - 180)
xh = np.linspace(-40, 40, 800)
for sign in [1, -1]:
    yh = sign*np.sqrt(np.maximum(xh**2 - 180, 0))
    ax.plot(xh, yh, 'r', label="Hyperbola" if sign==1 else "")

# Parabola: y^2 = 8x -> y = sqrt(8x)
xp = np.linspace(0, 40, 400)
for sign in [1, -1]:
    yp = sign*np.sqrt(8*xp)
    ax.plot(xp, yp, 'b', label="Parabola" if sign==1 else "")
```

Python+C code

```
# Intersection points from C
for (px, py) in points:
    ax.plot(px, py, 'ko', markersize=8)
    ax.text(px+0.5, py+0.5, f"({px:.0f},{py:.0f})")

ax.legend()
plt.savefig("/home/user/Matrix Theory: workspace/
    Matgeo_assignments/9.4.39/figs/Figure_1.png")
plt.show()
```

Python code

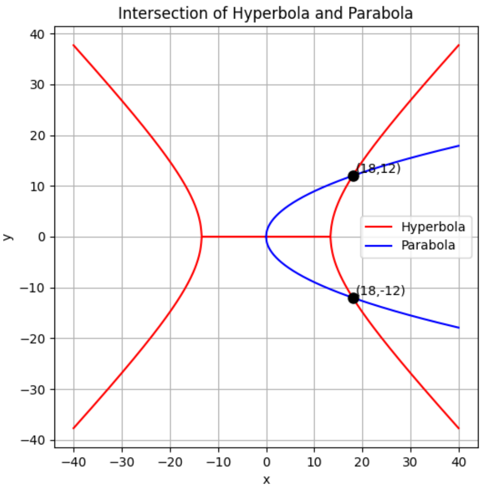
```
import sympy as sp
import numpy as np
import matplotlib.pyplot as plt

# Variables
x, y = sp.symbols('x y', real=True)
# Equations
eq1 = sp.Eq(x**2 - y**2, 180) # Hyperbola
eq2 = sp.Eq(y**2, 8*x) # Parabola
# Solve system
solutions = sp.solve([eq1, eq2], [x, y], dict=True)
print("Solutions:")
for sol in solutions:
    print(sol)
# Extract real solutions
real_solutions = [(float(sol[x]), float(sol[y])) for sol in
    solutions if sol[y].is_real]
```

```
# Setup plot
fig, ax = plt.subplots(figsize=(6,6))
ax.set_xlabel("x")
ax.set_ylabel("y")
ax.set_title("Intersection of Hyperbola and Parabola")
ax.grid(True)

# Range for plotting
xx = np.linspace(-20, 40, 400)

# Hyperbola: x^2 - y^2 = 180 -> y = sqrt(x^2 - 180)
xh = np.linspace(-40, 40, 800)
for sign in [1, -1]:
    yh = sign*np.sqrt(np.maximum(xh**2 - 180, 0)) # avoid
        negatives under sqrt
    ax.plot(xh, yh, 'r', label="Hyperbola" if sign==1 else "")
```

Python code

```python
# Parabola: y^2 = 8x -> y = sqrt(8x)
xp = np.linspace(0, 40, 400) # parabola domain x>=0
for sign in [1, -1]:
    yp = sign*np.sqrt(8*xp)
    ax.plot(xp, yp, 'b', label="Parabola" if sign==1 else "")
# Mark intersection points
for (px, py) in real_solutions:
    ax.plot(px, py, 'ko', markersize=8)
    ax.text(px+0.5, py+0.5, f"({px:.0f},{py:.0f})")

ax.legend()
plt.savefig("/home/user/Matrix Theory: workspace/
    Matgeo_assignments/9.4.39/figs/Figure_1.png")
plt.show()
```

Plot


Intersection of Hyperbola and Parabola
40
30
20
10
0
−10
−20
−30
−40
y
−40
−30
−20
−10
0
10
20
30
40
x
(18,12)
(18,-12)
Hyperbola
Parabola