9.4.38

Vishwambhar - EE25BTECH11025

11th october, 2025

Question

The diagonal of a rectangular field is 60 metres more than the shorter side. If the longer side is 30 metres more than the shorter side, find the sides of the field.

Let

Let:
The shorter side of the rectangle be x
The longer side of the rectangle be y
Then the diagonal of the rectangle will be $\sqrt{x^2 + y^2}$

Given

Given:

$$\sqrt{x^2 + y^2} = x + 60 \tag{1}$$

$$y^2 - 120x - 3600 = 0 \tag{2}$$

$$-x + y = 30 \tag{3}$$

Quadratic Form

Writing equation (3) in conic/quadratic form:

$$\mathbf{x}^\top A\mathbf{x} + 2\mathbf{u}^\top \mathbf{x} + c = 0 \tag{4}$$

$$\tag{5}$$

where,

$$A = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \tag{6}$$

$$\mathbf{u} = \begin{pmatrix} -60 \\ 0 \end{pmatrix} \tag{7}$$

$$c = -3600 \tag{8}$$

Parametric form

Writing equation(4) in parametric form:

$$\mathbf{x} = \mathbf{p} + t\mathbf{m} \tag{9}$$

where,

$$\mathbf{p} = \begin{pmatrix} 0 \\ 30 \end{pmatrix} \tag{10}$$

$$\mathbf{m} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \tag{11}$$

Substituting (9) in (4), we get:

$$pt^2 + qt + r = 0 \tag{12}$$

$$\tag{13}$$

where,

$$p = \mathbf{m}^\top A\mathbf{m} \tag{14}$$

$$q = 2\left(\mathbf{p}^\top A\mathbf{m} + \mathbf{u}^\top\mathbf{m}\right) \tag{15}$$

$$r = \mathbf{p}^\top A\mathbf{p} + 2\mathbf{u}^\top\mathbf{p} \tag{16}$$

Sridharacharya's Formula

By using Sridharacharya's formula,

$$t = \frac{1}{\mathbf{m}^\top A\mathbf{m}} \left(-\mathbf{m}^\top (A\mathbf{p} + \mathbf{u}) \pm \sqrt{\left[\mathbf{m}^\top (A\mathbf{p} + \mathbf{u})\right]^2 - (\mathbf{m}^\top A\mathbf{m}) (\mathbf{p}^\top A\mathbf{p} + 2\mathbf{u}^\top \mathbf{p}} \right. \tag{17}$$

Substituting (17) in (9) we get:

$$\mathbf{x} = \mathbf{p} + \frac{1}{\mathbf{m}^\top A\mathbf{m}} \left(-\mathbf{m}^\top (A\mathbf{p} + \mathbf{u}) + \sqrt{\left[\mathbf{m}^\top (A\mathbf{p} + \mathbf{u})\right]^2 - (\mathbf{m}^\top A\mathbf{m}) (\mathbf{p}^\top A\mathbf{p} +} \right. \tag{18}$$

$$\mathbf{x} = \mathbf{p} + \frac{1}{\mathbf{m}^\top A\mathbf{m}} \left(-\mathbf{m}^\top (A\mathbf{p} + \mathbf{u}) - \sqrt{\left[\mathbf{m}^\top (A\mathbf{p} + \mathbf{u})\right]^2 - (\mathbf{m}^\top A\mathbf{m}) (\mathbf{p}^\top A\mathbf{p} +} \right. \tag{19}$$

After substituting values in equation (18) and (19), we get:

$$\mathbf{p}_1 = \begin{pmatrix} 90 \\ 120 \end{pmatrix} \tag{20}$$

$$\mathbf{p}_2 = \begin{pmatrix} -30 \\ 0 \end{pmatrix} \tag{21}$$

Since the side of the rectangle cannot be negative. The correct vector is $\mathbf{p}_1$.
Therefore,

$$x = 90 \tag{22}$$

$$y = 120 \tag{23}$$

C Code

```c
#include<stdio.h>

void give_data(double *A, double *u, double *c, double *p, double
     *m, double *points){
    A[0] = 0; A[1] = 0; A[2] = 0; A[3] = 1;
    u[0] = -60; u[1] = 0;
    c[0] = -3600;
    p[0] = 0; p[1] = 30;
    m[0] = 1; m[1] = 1;
    points[0] = 1; points[1] = -120; points[2] = -3600; points[3]
         = 1; points[4] = 30;
}
```

Python code 1

```
import ctypes as ct
import numpy as np
from numpy.lib import scimath as np_scimath
lib = ct.CDLL("./problem.so")
lib.give_data.argtypes = [
    ct.POINTER(ct.c_double), ct.POINTER(ct.c_double), ct.POINTER(
        ct.c_double),
    ct.POINTER(ct.c_double), ct.POINTER(ct.c_double), ct.POINTER(
        ct.c_double)
]
pointsA = ct.c_double * 4
pointsu = ct.c_double * 2
pointsc = ct.c_double * 1
pointsp = ct.c_double * 2
pointsm = ct.c_double * 2
points = ct.c_double * 5
```

Python code 1

```
A = pointsA()
u = pointsu()
c = pointsc()
p = pointsp()
m = pointsm()
data = points()
lib.give_data(A, u, c, p, m, data)
A = np.array([[A[0], A[1]], [A[2], A[3]]])
u = np.array([[u[0]], [u[1]]])
p = np.array([[p[0]], [p[1]]])
m = np.array([[m[0]], [m[1]]])
c = c[0]
```

Python code 1

```
a1 = float(m.T @ A @ m)
b1 = float(2 * (p.T @ A @ m + u.T @ m))
c1 = float(p.T @ A @ p + 2 * u.T @ p + c)
D = b1**2 - 4 * a1 * c1
t1 = (-b1 + np_scimath.sqrt(D)) / (2 * a1)
t2 = (-b1 - np_scimath.sqrt(D)) / (2 * a1)
x1 = p + t1 * m
x2 = p + t2 * m
print("Intersection 1:", x1)
print("Intersection 2:", x2)
def send_data():
    return data, x1[0,0], x1[1,0], x2[0,0], x2[1,0]
```

Python code 2

```
import matplotlib.pyplot as plt
from call import send_data
import numpy as np
from numpy.lib import scimath
data , Ax, Ay, Bx, By = send_data()
x = np.linspace(-30, 150, 1000)
y = scimath.sqrt((-data[1]*x)-data[2])
A = np.linspace(-30, 150, 1000)
B = -(scimath.sqrt((-data[1]*A)-data[2]))
X = np.linspace(-100, 150, 20)
Y = X+30
```

Python code 2

```
plt.plot(x, y, "-r")
plt.plot(X, Y, "-g")
plt.plot(A, B, "-r")
plt.plot(Ax,Ay, "ko")
plt.text(Ax+0.1, Ay+0.1, f"({Ax:.0f},{Ay:.0f})", color = "black",
     fontsize = 12)
plt.plot(Bx,By, "ko")
plt.text(Bx+0.1, By+0.1, f"({Bx:.0f},{By:.0f})", color = "black",
     fontsize = 12)
plt.text(86.3, -115.3, r'$y^2=120(x+30)$', color = "black",
    fontsize = 12)
plt.text(-92.7, -60, "y=x+30", color = "black", fontsize = 12)
```

Python code 2

```
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.grid(True)
plt.axis("equal")
plt.savefig("../figs/plot.png")
plt.show()
```

Plot

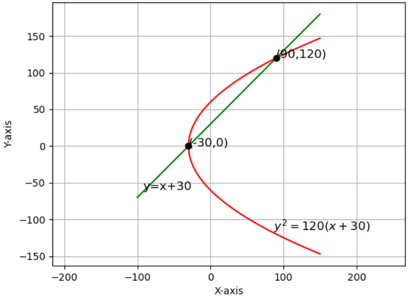

Figure: Plot of the parabola and the line