

Problem 7.4.30

ee25btech11023-Venkata Sai

September 19, 2025

1 Problem

2 Solution

- Formula
- Outer Product
- Conclusion
- Plot

3 C Code

4 Python Code

Problem

A circle is given by $x^2 + (y - 1)^2 = 1$, another circle C touches it externally and also the X axis, then the locus of its centre is

Formula

As the circle touches X -axis , Distance of a point from x -axis is given by

$$r = \mathbf{n}^T \mathbf{c} \quad (2.1)$$

where \mathbf{n} is the unit vector normal to x -axis

For the given circle with radius r_1 and center c_1

$$x^2 + (y - 1)^2 = 1 \quad (2.2)$$

$$\mathbf{p} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \mathbf{n} \text{ and } r_1 = 1 \quad (2.3)$$

Distance between their centers equal to sum of their radius

$$\|\mathbf{c} - \mathbf{p}\| = r + r_1 \quad (2.4)$$

$$\|\mathbf{c} - \mathbf{n}\| = \mathbf{n}^T \mathbf{c} + 1 \quad (2.5)$$

$$\|\mathbf{c} - \mathbf{n}\|^2 = (\mathbf{n}^T \mathbf{c} + 1)^2 \quad (2.6)$$

$$(\mathbf{c} - \mathbf{n})(\mathbf{c}^T - \mathbf{n}^T) = (\mathbf{n}^T \mathbf{c} + 1)^2 \quad (2.7)$$

Outer Product

$$\mathbf{c}^\top \mathbf{c} + \mathbf{n} \mathbf{n}^\top - \mathbf{c}^\top \mathbf{n} - \mathbf{n}^\top \mathbf{c} = (\mathbf{n}^\top \mathbf{c})^\top (\mathbf{n}^\top \mathbf{c}) + 2\mathbf{n}^\top \mathbf{c} + 1 \quad (2.8)$$

$$\mathbf{c}^\top \mathbf{c} + \|\mathbf{n}\|^2 - 2\mathbf{n}^\top \mathbf{c} = (\mathbf{c}^\top \mathbf{n} \mathbf{n}^\top \mathbf{c}) + 2\mathbf{n}^\top \mathbf{c} + 1 \quad (2.9)$$

$$\mathbf{c}^\top \mathbf{c} + 1 = (\mathbf{c}^\top \mathbf{n} \mathbf{n}^\top \mathbf{c}) + 4\mathbf{n}^\top \mathbf{c} + 1 \quad (2.10)$$

$$\mathbf{c}^\top \mathbf{c} - (\mathbf{c}^\top \mathbf{n} \mathbf{n}^\top \mathbf{c}) = 4\mathbf{n}^\top \mathbf{c} \quad (2.11)$$

$$\mathbf{c}^\top (\mathbf{I} - \mathbf{n} \mathbf{n}^\top) \mathbf{c} = 4\mathbf{n}^\top \mathbf{c} \quad (2.12)$$

$$\mathbf{c}^\top \left(\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \end{pmatrix} \right) \mathbf{c} = 4 \begin{pmatrix} 0 \\ 1 \end{pmatrix}^\top \mathbf{c} \quad (2.13)$$

$$\mathbf{c}^\top \left(\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \right) \mathbf{c} = 4 \begin{pmatrix} 0 & 1 \end{pmatrix} \mathbf{c} \quad (2.14)$$

Conclusion

$$\begin{pmatrix} x & y \end{pmatrix} \left(\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \right) \begin{pmatrix} x \\ y \end{pmatrix} = 4 \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad (2.15)$$

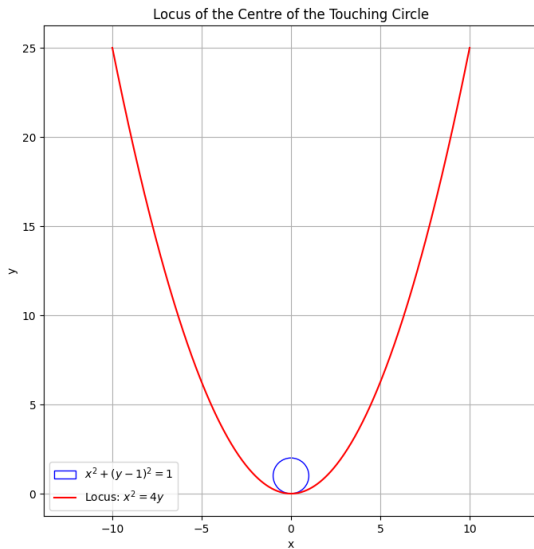
$$\begin{pmatrix} x & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = 4y \quad (2.16)$$

$$x^2 = 4y \quad (2.17)$$

Hence $y \geq 0$. Then

$$\{(x, y) : x^2 = 4y\} \cup \{(x, y) : y \geq 0\} \quad (2.18)$$

Plot



C Code

```
void get_circle_params(double* out_data) {  
    out_data[0] = 0.0;  
    out_data[1] = 1.0;  
    out_data[2] = 1.0;  
}
```


Python Code for Solving

```
1 import ctypes
2 import sympy
3
4 def find_locus_equation():
5
6     lib = ctypes.CDLL('./code.so')
7
8     double_array_3 = ctypes.c_double * 3
9     lib.get_circle_params.argtypes = [ctypes.POINTER(ctypes.
10         c_double)]
11     out_data_c = double_array_3()
12     lib.get_circle_params(out_data_c)
13
14     c1_x, c1_y, r1 = list(out_data_c)
15     c1_center = (c1_x, c1_y)
16
17     h, k = sympy.symbols('h k', real=True, positive=True)
18     r = k
19     lhs = (h - c1_center[0])**2 + (k - c1_center[1])**2
```

Python Code for Solving

```
rhs = (r + r1)**2
equation = sympy.Eq(lhs, rhs.subs(r, k))
locus_eq = sympy.simplify(equation.lhs - equation.rhs)
x, y = sympy.symbols('x y')
final_locus = locus_eq.subs([(h, x), (k, y)])

return sympy.Eq(final_locus, 0)
```

Python Code for Plotting

```
# Code by /sdcard/github/matgeo/codes/CoordGeoVV Sharma
# September 12, 2023
# Revised July 21, 2024
# Released under GNU GPL
# Section Formula

import sys
sys.path.insert(0, '/workspaces/urban-potato/matgeo/codes/
    CoordGeo/')

import numpy as np
import matplotlib.pyplot as plt
from call import find_locus_equation

locus_equation = find_locus_equation()

print(fLocus equation: {locus_equation})
fig, ax = plt.subplots(figsize=(8, 8))

c1 = plt.Circle((0, 1), 1, color='blue', fill=False, label='$x^2
    + (y-1)^2 = 1$')
```

Python Code for Plotting

```
ax.add_patch(c1)

x_vals = np.linspace(-10, 10, 400)
y_vals = x_vals**2 / 4
ax.plot(x_vals, y_vals, 'r-', label=f'Locus:  $x^2=4y$ ')

ax.set_title(Locus of the Centre of the Touching Circle)
ax.set_xlabel(x); ax.set_ylabel(y)
ax.grid(True); ax.axis('equal'); ax.legend()
plt.show()
plt.savefig('fig1.png')
```