Presentation - Matgeo

Aryansingh Sonaye
AI25BTECH11032
EE1030 - Matrix Theory

September 29, 2025

Problem Statement

Problem 9.8.34 Find the equation of the line passing through the points of intersection of the circles

$$3x^2 + 3y^2 - 2x + 12y - 9 = 0 \quad \text{and} \quad x^2 + y^2 + 6x + 2y - 15 = 0. \tag{1.1}$$

Description of Variables used

The general conic is

$$\mathbf{x}^\top V\mathbf{x} + 2\mathbf{u}^\top\mathbf{x} + f = 0, \quad \mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix}. \tag{2.1}$$

	C_1	C_2
V	$\begin{pmatrix} 3 & 0 \\ 0 & 3 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
$\mathbf{u}$	$\begin{pmatrix} -1 \\ 6 \end{pmatrix}$	$\begin{pmatrix} 3 \\ 1 \end{pmatrix}$
f	-9	-15

Theoretical Solution

All conics through the intersection points are given by the locus

$$\mathbf{x}^\top (V_1 + \mu V_2)\mathbf{x} + 2(\mathbf{u}_1 + \mu \mathbf{u}_2)^\top \mathbf{x} + (f_1 + \mu f_2) = 0. \tag{2.2}$$

Since we want a line, we eliminate the quadratic part by choosing μ such that

$$V_1 + \mu V_2 = (3 + \mu) I = 0, \tag{2.3}$$

$$\mu = -3. \tag{2.4}$$

Substituting $\mu = -3$, the equation of the required line becomes

$$2(\mathbf{u}_1 - 3\mathbf{u}_2)^\top \mathbf{x} + (f_1 - 3f_2) = 0. \tag{2.5}$$

Theoretical Solution

Now we compute the coefficients:

$$\mathbf{u}_1 - 3\mathbf{u}_2 = \begin{pmatrix} -1 \\ 6 \end{pmatrix} - 3\begin{pmatrix} 3 \\ 1 \end{pmatrix} = \begin{pmatrix} -10 \\ 3 \end{pmatrix}, \tag{2.6}$$

$$f_1 - 3f_2 = -9 - 3(-15) = 36. \tag{2.7}$$

Thus the line is

$$2\begin{pmatrix} -10 & 3 \end{pmatrix}\mathbf{x} + 36 = 0. \tag{2.8}$$

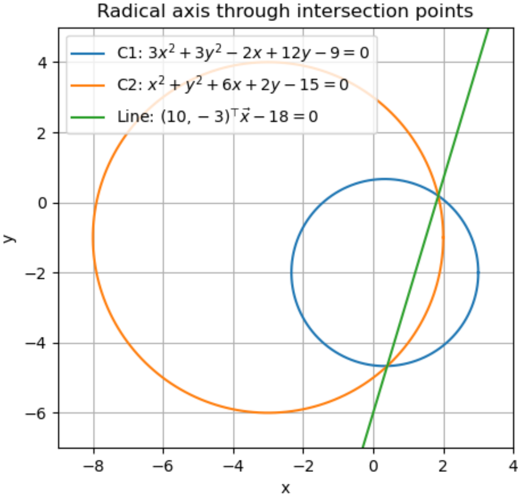
Multiplying throughout by $-\frac{1}{2}$, we obtain

$$\begin{pmatrix} 10 \\ -3 \end{pmatrix}^{\top}\mathbf{x} - 18 = 0. \tag{2.9}$$

$$\boxed{\begin{pmatrix} 10 \\ -3 \end{pmatrix}^{\top}\mathbf{x} - 18 = 0} \tag{2.10}$$

This is the required line through the intersection points.

Plot


Radical axis through intersection points
C1: $3x^2 + 3y^2 - 2x + 12y - 9 = 0$
C2: $x^2 + y^2 + 6x + 2y - 15 = 0$
Line: $(10, -3)^\top \vec{x} - 18 = 0$
4
2
0
−2
−4
−6
−8
−6
−4
−2
0
2
4
x
y

Figure

Code - C

```c
// Compute radical axis of two circles in conic form
// Circle: x^T V x + 2u^T x + f = 0 with V = a I
// Input: a1,u1,f1 and a2,u2,f2
// Output: line L^T x + c = 0 (L is size 2, c scalar)

void radical_axis(
    double a1, const double u1[2], double f1,
    double a2, const double u2[2], double f2,
    double L_out[2], double *c_out
) {
    double mu = -a1 / a2; // value that cancels quadratic part
    L_out[0] = 2.0 * (u1[0] + mu * u2[0]);
    L_out[1] = 2.0 * (u1[1] + mu * u2[1]);
    *c_out = (f1 + mu * f2);
}
```

Code - Python(with shared C code)

The code to obtain the required plot is

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

# 1) load the shared library (same folder)
lib = ct.CDLL("./libradical_simple.so")

# 2) tell ctypes the C signature
# void radical_axis(double a1, const double u1[2], double f1,
# double a2, const double u2[2], double f2,
# double L_out[2], double *c_out);
lib.radical_axis.argtypes = [
    ct.c_double, # a1
    ct.POINTER(ct.c_double), ct.c_double, # u1, f1
    ct.c_double, # a2
    ct.POINTER(ct.c_double), ct.c_double, # u2, f2
```

Code - Python(with shared C code)

```
    ct.POINTER(ct.c_double), # L_out
    ct.POINTER(ct.c_double) # c_out
]
lib.radical_axis.restype = None

# 3) problem data (Problem 9.8.34)
a1 = 3.0
u1 = np.array([-1.0, 6.0], dtype=np.double)
f1 = -9.0
a2 = 1.0
u2 = np.array([3.0, 1.0], dtype=np.double)
f2 = -15.0

# 4) outputs
L = np.zeros(2, dtype=np.double)
c = ct.c_double() # <-- IMPORTANT: ctypes double, so we can pass
    byref
```

Code - Python(with shared C code)

```
# 5) call C
lib.radical_axis(
    a1, u1.ctypes.data_as(ct.POINTER(ct.c_double)), f1,
    a2, u2.ctypes.data_as(ct.POINTER(ct.c_double)), f2,
    L.ctypes.data_as(ct.POINTER(ct.c_double)),
    ct.byref(c) # pass the address of the ctypes double
)

print("Raw line from C: L =", L, " c =", c.value)

# 6) scale to nice integers: (10, -3)^T x - 18 = 0
scale = -0.5 # because C gives L=[-20,6], c=36
L_scaled = L * scale
c_scaled = c.value * scale
print(f"Vector form: ({int(L_scaled[0])} \\\\ {int(L_scaled[1])})^T x {
    c_scaled:+.0f} = 0")
```

Code - Python(with shared C code)

```
# 7) plot circles + line
def circle_center_radius(a, u, f):
    # V = a I; center = -u/a; r^2 = ||u||^2/a^2 - f/a
    center = -u / a
    r2 = (u @ u) / (a*a) - f / a
    return center, np.sqrt(r2)

c1, r1 = circle_center_radius(a1, u1, f1)
c2, r2 = circle_center_radius(a2, u2, f2)
theta = np.linspace(0, 2*np.pi, 400)
x1 = c1[0] + r1*np.cos(theta); y1 = c1[1] + r1*np.sin(theta)
x2 = c2[0] + r2*np.cos(theta); y2 = c2[1] + r2*np.sin(theta)

xmin = min(c1[0]-r1, c2[0]-r2) - 1
xmax = max(c1[0]+r1, c2[0]+r2) + 1
ymin = min(c1[1]-r1, c2[1]-r2) - 1
ymax = max(c1[1]+r1, c2[1]+r2) + 1
```

Code - Python(with shared C code)

```
xx = np.linspace(xmin, xmax, 600)
if abs(L_scaled[1]) > 1e-12:
    yy = (-c_scaled - L_scaled[0]*xx) / L_scaled[1]
else:
    xx = np.full_like(xx, -c_scaled / L_scaled[0])
    yy = np.linspace(ymin, ymax, 600)
plt.figure()
plt.plot(x1, y1, label="C1: $3x^2+3y^2-2x+12y-9=0$")
plt.plot(x2, y2, label="C2: $x^2+y^2+6x+2y-15=0$")
plt.plot(xx, yy, label=r"Line: $(10,-3)^\top \vec{x} - 18 = 0$")
plt.gca().set_aspect('equal', adjustable='box')
plt.xlim(xmin, xmax); plt.ylim(ymin, ymax)
plt.grid(True); plt.legend()
plt.title("Radical axis through intersection points")
plt.xlabel("x"); plt.ylabel("y")
plt.savefig("radical.png")
plt.show()
```

Code - Python only

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

# Circles:
# C1: 3x^2 + 3y^2 - 2x + 12y - 9 = 0
a1 = 3.0
u1 = np.array([-1.0, 6.0])
f1 = -9.0
# C2: x^2 + y^2 + 6x + 2y - 15 = 0
a2 = 1.0
u2 = np.array([3.0, 1.0])
f2 = -15.0

# ---- Find radical axis (line through intersections) ----
mu = -a1/a2
L = 2*(u1 + mu*u2) # vector
c = f1 + mu*f2 # scalar
```

Code - Python only

```
print("Line (raw):", L, " . x + ", c, " = 0")

# Scale to nice integers
L = -0.5*L
c = -0.5*c
print("Final line:", f"({int(L[0])} \\\\ {int(L[1])})^T x {int(c):+d} = 0")

# ---- Plot circles and line ----
theta = np.linspace(0, 2*np.pi, 400)

# Circle 1
center1 = -u1/a1
r1 = np.sqrt((u1@u1)/(a1*a1) - f1/a1)
x1 = center1[0] + r1*np.cos(theta)
y1 = center1[1] + r1*np.sin(theta)
```

Code - Python only

```
# Circle 2
center2 = -u2/a2
r2 = np.sqrt((u2@u2)/(a2*a2) - f2/a2)
x2 = center2[0] + r2*np.cos(theta)
y2 = center2[1] + r2*np.sin(theta)
# Line points
xx = np.linspace(min(center1[0]-r1, center2[0]-r2)-1,
                 max(center1[0]+r1, center2[0]+r2)+1, 600)
yy = (-c - L[0]*xx)/L[1]

plt.plot(x1, y1, label="C1")
plt.plot(x2, y2, label="C2")
plt.plot(xx, yy, label="Radical axis")
plt.gca().set_aspect('equal', adjustable='box')
plt.legend(); plt.grid(True)
plt.savefig("newradical.png")
plt.show()
```