7.4.43

M Chanakya Srinivas- EE25BTECH11036

Problem Statement

A circle passes through three points A, B, C with AC as its diameter. A line through A intersects chord BC at D.
If $\angle DAB = \alpha$, $\angle CAB = \beta$, and the distance between A and the midpoint of DC is d, prove that the area of the circle is

$$\frac{\pi d^2 \cos^2 \alpha}{\cos^2 \alpha + \cos^2 \beta + 2 \cos \alpha \cos \beta \cos(\beta - \alpha)}.$$

Vector Setup

Let

$$\mathbf{A}, \mathbf{B}, \mathbf{C} \in \mathbb{R}^2$$

be position vectors of A, B, C, with AC as the diameter.
The line through A intersects BC at D, and the midpoint of DC is

$$\mathbf{M} = \frac{\mathbf{D} + \mathbf{C}}{2}, \tag{1}$$

$$d = \|\mathbf{A} - \mathbf{M}\|. \tag{2}$$

Cosines of angles:

$$\cos\alpha = \frac{(\mathbf{B} - \mathbf{A})^T(\mathbf{D} - \mathbf{A})}{\|\mathbf{B} - \mathbf{A}\| \, \|\mathbf{D} - \mathbf{A}\|}, \tag{3}$$

$$\cos\beta = \frac{(\mathbf{C} - \mathbf{A})^T(\mathbf{B} - \mathbf{A})}{\|\mathbf{C} - \mathbf{A}\| \, \|\mathbf{B} - \mathbf{A}\|}. \tag{4}$$

Circle Radius

Since AC is the diameter:

$$r = \frac{1}{2}\|\mathbf{A} - \mathbf{C}\|. \tag{5}$$

Let D lie on chord BC parametrically:

$$\mathbf{D} = \mathbf{B} + t(\mathbf{C} - \mathbf{B}), \quad 0 < t < 1. \tag{6}$$

Then midpoint M:

$$\mathbf{M} = \frac{\mathbf{D} + \mathbf{C}}{2} = \frac{1+t}{2}\mathbf{C} + \frac{1-t}{2}\mathbf{B}. \tag{7}$$

Vector from A to M:

$$\mathbf{M} - \mathbf{A} = \frac{1+t}{2}\mathbf{C} + \frac{1-t}{2}\mathbf{B} - \mathbf{A}. \tag{8}$$

Relation between r and d

The squared distance is

$$d^2 = \|\mathbf{M} - \mathbf{A}\|^2 = \left\| \frac{1+t}{2}\mathbf{C} + \frac{1-t}{2}\mathbf{B} - \mathbf{A} \right\|^2 . \tag{9}$$

Solving for $\|\mathbf{A} - \mathbf{C}\|$ in terms of d, α, β:

$$\|\mathbf{A} - \mathbf{C}\|^2 = \frac{4d^2 \cos^2 \alpha}{\cos^2 \alpha + \cos^2 \beta + 2 \cos \alpha \cos \beta \cos(\beta - \alpha)} . \tag{10}$$

Radius squared:

$$r^2 = \frac{1}{4}\|\mathbf{A} - \mathbf{C}\|^2 = \frac{d^2 \cos^2 \alpha}{\cos^2 \alpha + \cos^2 \beta + 2 \cos \alpha \cos \beta \cos(\beta - \alpha)} . \tag{11}$$

Area of Circle

$$\text{Area} = \pi r^2 \tag{12}$$

$$= \frac{\pi d^2 \cos^2\alpha}{\cos^2\alpha + \cos^2\beta + 2\cos\alpha\cos\beta\cos(\beta-\alpha)}. \tag{13}$$

$$\boxed{\text{Area of circle} = \frac{\pi d^2 \cos^2\alpha}{\cos^2\alpha + \cos^2\beta + 2\cos\alpha\cos\beta\cos(\beta-\alpha)}}$$

C Code

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

typedef struct {
    double x, y;
} Vec;

// Vector subtraction
Vec sub(Vec a, Vec b) {
    Vec r = {a.x - b.x, a.y - b.y};
    return r;
}

// Midpoint
Vec midpoint(Vec a, Vec b) {
    Vec r = {(a.x + b.x)/2.0, (a.y + b.y)/2.0};
    return r;
}
```

C Code

```c
// Dot product
double dot(Vec a, Vec b) {
    return a.x*b.x + a.y*b.y;
}

// Norm
double norm(Vec a) {
    return sqrt(dot(a,a));
}

// Circle center (midpoint of AC)
Vec circle_center(Vec A, Vec C) {
    return midpoint(A, C);
}

// Midpoint of DC
Vec midpoint_DC(Vec D, Vec C) {
    return midpoint(D, C);
}
```

Python code through shared output

```
import ctypes
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.patches import Arc

# Load shared library
lib = ctypes.CDLL(./libgeom.so)

class Vec(ctypes.Structure):
    _fields_ = [(x, ctypes.c_double), (y, ctypes.c_double)]

# Bind functions
lib.circle_center.argtypes = [Vec, Vec]
lib.circle_center.restype = Vec

lib.midpoint_DC.argtypes = [Vec, Vec]
lib.midpoint_DC.restype = Vec

# Shift circle upward for clarity
```

Python code through shared output

```
O = lib.circle_center(A,C)
M = lib.midpoint_DC(D,C)

# Convert to numpy
def vec2np(v): return np.array([v.x, v.y])
An,Bn,Cn,Dn,On,Mn = map(vec2np,[A,B,C,D,O,M])

# Circle
r = np.linalg.norm(Cn - An)/2
theta = np.linspace(0,2*np.pi,400)
circle_x = On[0] + r*np.cos(theta)
circle_y = On[1] + r*np.sin(theta)

fig, ax = plt.subplots(figsize=(6,6))
ax.plot(circle_x, circle_y, 'b')

# Points
for P,name in zip([An,Bn,Cn,Dn,Mn,On],['A','B','C','D','M','O']):
    ax.plot(P[0],P[1],'ro')
```

Python code through shared output

```python
# Lines
ax.plot([Bn[0],Cn[0]],[Bn[1],Cn[1]],'g--',label=$BC$)
ax.plot([An[0],Dn[0]],[An[1],Dn[1]],'m--',label=$AD$)
ax.plot([An[0],Cn[0]],[An[1],Cn[1]],'k-', label=$AC$)
ax.plot([An[0],Bn[0]],[An[1],Bn[1]],'c-', label=$AB$) # New AB

# Function to draw arcs for angles
def draw_angle(ax, center, p1, p2, radius=0.25, label=):
    ang1 = np.degrees(np.arctan2(p1[1]-center[1], p1[0]-center
        [0]))
    ang2 = np.degrees(np.arctan2(p2[1]-center[1], p2[0]-center
        [0]))
    arc = Arc(center, 2*radius, 2*radius, angle=0,
              theta1=min(ang1,ang2), theta2=max(ang1,ang2),
```

Python code through shared output

```
               color='k')
    ax.add_patch(arc)
    mid = (ang1+ang2)/2
    ax.text(center[0]+radius*np.cos(np.radians(mid)),
            center[1]+radius*np.sin(np.radians(mid)),
            label, fontsize=14, color=red)

# Angles at A
draw_angle(ax, An, Dn, Bn, radius=0.3, label=r$\alpha$)
draw_angle(ax, An, Cn, Bn, radius=0.5, label=r$\beta$)

# Aesthetics
ax.set_aspect(1)
ax.set_xlim(-0.5,2.5)
ax.set_ylim(0,2.5)
ax.legend(fontsize=10, loc=upper right)
plt.show()
```

Only Python code

```
import numpy as np
import matplotlib.pyplot as plt
from numpy import linalg as LA
from conics.funcs import circ_gen
from line.funcs import line_gen
from matplotlib.patches import Arc

# === Step 1: Circle with diameter AC ===
A = np.array([0.0,0.0]).reshape(-1,1)
C = np.array([4.0,0.0]).reshape(-1,1)
O = (A + C)/2
R = LA.norm(C - O)

# === Step 2: Choose B on circle (not collinear with AC) ===
theta = np.deg2rad(60)
B = O + R * np.array([[np.cos(theta)], [np.sin(theta)]])
```

Only Python code

```
# === Step 3: Angle alpha at A ===
alpha = np.deg2rad(30) # angle DAB
slope_AD = np.tan(alpha)

# === Step 4: Compute intersection D on BC segment ===
dx = C[0,0] - B[0,0]
dy = C[1,0] - B[1,0]
t = (slope_AD*(A[0,0]-B[0,0]) + (B[1,0]-A[1,0])) / (dy - slope_AD
    *dx)
t = np.clip(t, 0.15, 0.85) # ensure D inside BC
D = (B + t*(C-B)).reshape(-1,1)

# === Step 5: Midpoint M of DC ===
M = (D + C)/2

# === Step 6: Compute distance d = AM ===
d = LA.norm(A - M)
```

Only Python code

```
# === Step 7: Generate circle and lines using your functions ===
x_circ = circ_gen(O,R)
x_AB = line_gen(A,B)
x_AC = line_gen(A,C)
x_BC = line_gen(B,C)
x_AD = line_gen(A,D)
x_DM = line_gen(D,M)

# === Step 8: Plot setup ===
plt.figure(figsize=(8,8))
plt.plot(x_circ[0,:], x_circ[1,:], 'b', lw=2, label='Circle')
plt.plot(x_AB[0,:], x_AB[1,:], 'g', label='AB')
plt.plot(x_AC[0,:], x_AC[1,:], 'k', label='AC (diameter)')
plt.plot(x_BC[0,:], x_BC[1,:], 'c', lw=2, label='BC (chord)')
plt.plot(x_AD[0,:], x_AD[1,:], 'r', lw=2, label='AD')
plt.plot(x_DM[0,:], x_DM[1,:], 'm--', lw=2, label='DC midpoint
    line')
```

Only Python code

```
# === Step 9: Plot points ===
points = {'A':A,'B':B,'C':C,'D':D,'M':M,'O':O}
for name,p in points.items():
    plt.scatter(p[0,0], p[1,0], s=70, zorder=5)
    offset = 0.15
    if name in ['D','B','M']:
        offset = 0.25
    plt.text(p[0,0]+offset, p[1,0]+offset, name, fontsize=12,
         fontweight='bold')

# === Step 10: Draw angles alpha (DAB) and beta (CAB) clearly ===
def plot_angle(center, p1, p2, radius, color, label):
    v1, v2 = p1-center, p2-center
    ang1 = np.degrees(np.arctan2(v1[1,0], v1[0,0]))
    ang2 = np.degrees(np.arctan2(v2[1,0], v2[0,0]))
    if ang2 < ang1: ang1, ang2 = ang2, ang1
    arc = Arc((center[0,0], center[1,0]), 2*radius, 2*radius,
              angle=0, theta1=ang1, theta2=ang2, color=color, lw
                 =3)
```

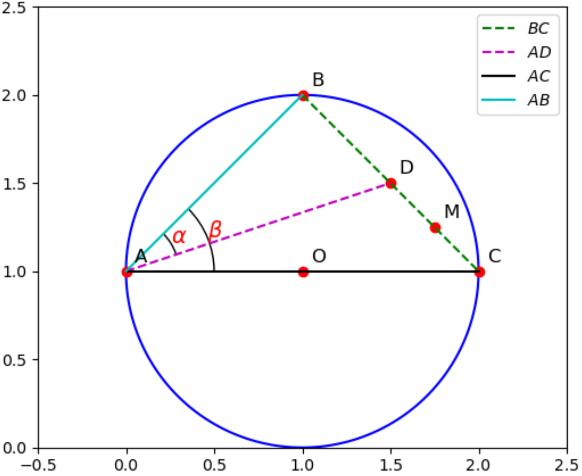
Only Python code

```
    plt.gca().add_patch(arc)
    mid = np.radians((ang1+ang2)/2)
    plt.text(center[0,0]+1.4*radius*np.cos(mid),
             center[1,0]+1.4*radius*np.sin(mid),
             label, fontsize=16, color=color, fontweight='bold')

plot_angle(A,D,B,0.5,'r',r'$\alpha$') # DAB
plot_angle(A,C,B,0.8,'b',r'$\beta$') # CAB
```

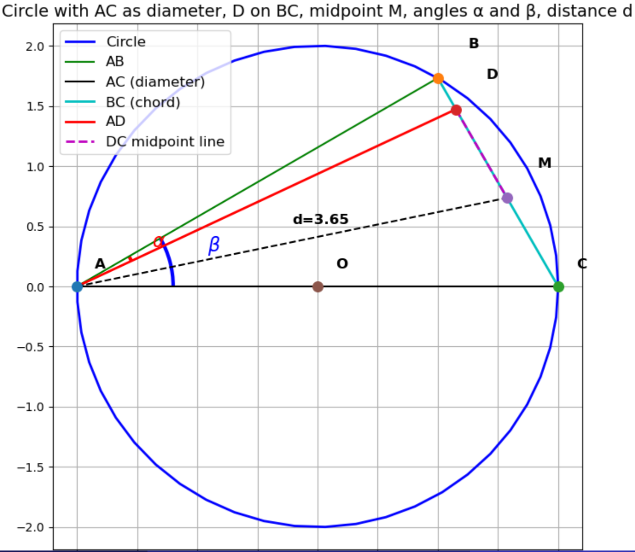
Only Python code

```
# === Step 11: Annotate distance d = AM ===
plt.plot([A[0,0], M[0,0]], [A[1,0], M[1,0]], 'k--', lw=1.5)
mid_label = (A + M)/2
plt.text(mid_label[0,0], mid_label[1,0]+0.15, f'd={d:.2f}',
    fontsize=12, fontweight='bold', color='k')

# === Step 12: Finalize plot ===
plt.axis('equal')
plt.grid(True)
plt.legend(fontsize=12)
plt.title('Circle with AC as diameter, D on BC, midpoint M,
    angles  and , distance d', fontsize=14)
plt.show()
```

PLOTS



PLOTS



Circle with AC as diameter, D on BC, midpoint M, angles α and β, distance d
Circle
AB
AC (diameter)
BC (chord)
AD
DC midpoint line
B
D
M
d=3.65
α
β
A
O
C
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0