1.8.27

Vector Geometry

EE25BTECH11010 - Arsh Dhoke

Question

Find the equation of set of points $\mathbf{P}$ such that
$\|\mathbf{A}-\mathbf{P}\|^2+\|\mathbf{B}-\mathbf{P}\|^2=2k^2$,
where $\mathbf{A}(3,4,5)$ and $\mathbf{B}(-1,3,-7)$.

The input parameters for the problem are given in the table below.

Vectors	**Points**
A	$\begin{pmatrix} 3 \\ 4 \\ 5 \end{pmatrix}$
B	$\begin{pmatrix} -1 \\ 3 \\ -7 \end{pmatrix}$

Table: Vectors and their corresponding points

Condition Setup

The condition is

$$\|\mathbf{A}-\mathbf{P}\|^2+\|\mathbf{B}-\mathbf{P}\|^2=2k^2 \tag{1}$$

$$(\mathbf{P}-\mathbf{A})^T(\mathbf{P}-\mathbf{A})+(\mathbf{P}-\mathbf{B})^T(\mathbf{P}-\mathbf{B})=2k^2 \tag{2}$$

$$\mathbf{P}^T\mathbf{P}-(\mathbf{A}+\mathbf{B})^T\mathbf{P}+\frac{\mathbf{A}^T\mathbf{A}+\mathbf{B}^T\mathbf{B}}{2}=k^2 \tag{3}$$

Completing square

$$\left\|\mathbf{P} - \frac{\mathbf{A}+\mathbf{B}}{2}\right\|^2 - \frac{(\mathbf{A}+\mathbf{B})^T(\mathbf{A}+\mathbf{B})}{4} + \frac{\mathbf{A}^T\mathbf{A}+\mathbf{B}^T\mathbf{B}}{2} = k^2 \tag{4}$$

$$(\mathbf{A}+\mathbf{B})^T(\mathbf{A}+\mathbf{B}) = 57,\ \mathbf{A}^T\mathbf{A} = 50,\ \mathbf{B}^T\mathbf{B} = 59 \tag{5}$$

Simplification

Rearranging and substituting values we get:

$$\left\| \mathbf{P} - \frac{\mathbf{A} + \mathbf{B}}{2} \right\|^2 = k^2 - \frac{161}{4} \tag{6}$$

$$\left(\mathbf{P} - \frac{\mathbf{A} + \mathbf{B}}{2} \right)^T \left(\mathbf{P} - \frac{\mathbf{A} + \mathbf{B}}{2} \right) = k^2 - \frac{161}{4} \tag{7}$$

$$k^2 > \frac{161}{4} \tag{8}$$

Plot

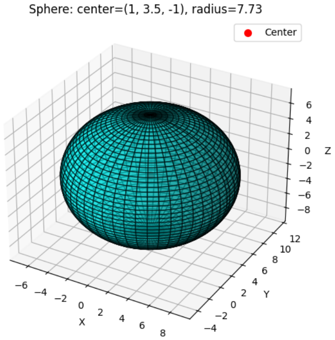

Figure: Graph plotted for $k = 10$ as example.

C Code

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

// Function to solve the locus equation for k=10
void solveSphere() {
    double k = 10.0;

    // Center of sphere (from derivation)
    double Cx = 1.0;
    double Cy = 7.0 / 2.0; // 3.5
    double Cz = -1.0;

    // Radius squared (correct formula)
    double R2 = k * k - 161.0 / 4.0; // 100 - 40.25 = 59.75

    if (R2 <= 0) {
        printf("For k = %.2f, no real sphere exists (radius^2 =
            %.2f)\n", k, R2);
```

C Code

```c
        return;
    }

    double R = sqrt(R2);

    printf("Equation of the sphere:\n");
    printf("(x - %.2f)^2 + (y - %.2f)^2 + (z - %.2f)^2 = %.2f\n",
           Cx, Cy, Cz, R * R);

    printf("Center: (%.2f, %.2f, %.2f)\n", Cx, Cy, Cz);
    printf("Radius: %.2f\n", R);
}

int main() {
    solveSphere(); // for k=10
    return 0;
}
```

Python Code

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

# Define variables
x, y, z, k = sp.symbols('x y z k', real=True)

# Define points
A = sp.Matrix([3, 4, 5])
B = sp.Matrix([-1, 3, -7])
P = sp.Matrix([x, y, z])

# Distances squared
PA2 = (P - A).dot(P - A)
PB2 = (P - B).dot(P - B)

# Equation condition
eq = sp.Eq(PA2 + PB2, 2*k**2)
```

Python Code

```
print("Expanded equation:")
print(sp.expand(eq))

# Simplify into standard sphere form
expr = sp.expand(PA2 + PB2 - 2*k**2)
expr = sp.simplify(expr)
print("\nSimplified expression = 0:")
print(expr)

# Complete the squares
sphere_eq = sp.together(sp.factor(expr))
print("\nEquation of locus (sphere):")
print(sphere_eq)
```

Python Code

```
# ---- Extract center and radius ----
center = sp.Matrix([1, sp.Rational(7,2), -1])
radius_sq = k**2 - sp.Rational(161,4)

print(f"\nCenter: {center}")
print(f"Radius^2: {radius_sq}")

# ---- Plot the sphere for k=10 ----
k_val = 10
R = float(sp.sqrt(radius_sq.subs(k, k_val)))

# Mesh grid
u = np.linspace(0, 2*np.pi, 100)
v = np.linspace(0, np.pi, 100)
X = float(center[0]) + R*np.outer(np.cos(u), np.sin(v))
Y = float(center[1]) + R*np.outer(np.sin(u), np.sin(v))
Z = float(center[2]) + R*np.outer(np.ones_like(u), np.cos(v))
```

Python Code

```
fig = plt.figure(figsize=(6,6))
ax = fig.add_subplot(111, projection='3d')

ax.plot_surface(X, Y, Z, alpha=0.5, edgecolor='k', linewidth=0.3)
ax.scatter([float(center[0])], [float(center[1])], [float(center
    [2])], s=50, label="Center")

ax.set_xlabel("X")
ax.set_ylabel("Y")
ax.set_zlabel("Z")
ax.set_title("Sphere locus: PA^2 + PB^2 = 2k^2 (k=10)")
ax.legend()

plt.savefig("/home/arsh-dhoke/ee1030-2025/ee25btech11010/matgeo
    /1.8.27/figs/fig1.png")
plt.show()
```

Python+ C Code

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

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

# Define argument and return types
lib.solveSphere.argtypes = [
    ctypes.POINTER(ctypes.c_double),
    ctypes.POINTER(ctypes.c_double),
    ctypes.POINTER(ctypes.c_double),
    ctypes.POINTER(ctypes.c_double)
]
lib.solveSphere.restype = None
```

Python+ C Code

```
# Prepare output variables
Cx = ctypes.c_double()
Cy = ctypes.c_double()
Cz = ctypes.c_double()
R = ctypes.c_double()

# Call the C function (k=10 inside C)
lib.solveSphere(ctypes.byref(Cx), ctypes.byref(Cy), ctypes.byref(
    Cz), ctypes.byref(R))

# Extract results from C
cx, cy, cz, r = Cx.value, Cy.value, Cz.value, R.value

if r < 0:
    print("No real sphere exists for k=10")
    exit()
```

Python+ C Code

```
print(f"Equation of sphere (from C): (x - {cx:.2f})^2 + (y - {cy
    :.2f})^2 + (z - {cz:.2f})^2 = {r**2:.2f}")
print(f"Center: ({cx:.2f}, {cy:.2f}, {cz:.2f}), Radius: {r:.2f}")

# ========================
# Plotting (Matplotlib)
# ========================
u = np.linspace(0, 2*np.pi, 100)
v = np.linspace(0, np.pi, 100)
X = cx + r * np.outer(np.cos(u), np.sin(v))
Y = cy + r * np.outer(np.sin(u), np.sin(v))
Z = cz + r * np.outer(np.ones_like(u), np.cos(v))

fig = plt.figure(figsize=(6,6))
ax = fig.add_subplot(111, projection='3d')

ax.plot_surface(X, Y, Z, alpha=0.5, edgecolor='k', linewidth=0.3)
ax.scatter([cx], [cy], [cz], s=50, label="Center")
```

```
ax.set_xlabel("X")
ax.set_ylabel("Y")
ax.set_zlabel("Z")
ax.set_title("Sphere locus: PA^2 + PB^2 = 2k^2 (k=10, from C)")
ax.legend()

plt.savefig("/home/arsh-dhoke/ee1030-2025/ee25btech11010/matgeo
    /1.8.27/figs/fig1.png")
plt.show()
```