

```

1  import numpy as np
2  import matplotlib.pyplot as plt
3
4  def rotation_matrix(roll, pitch, yaw):
5      """
6      Create rotation matrix from Euler angles (roll, pitch, yaw)
7      Using ZYX rotation order.
8      """
9      Rx = np.array([[1, 0, 0],
10                    [0, np.cos(roll), -np.sin(roll)],
11                    [0, np.sin(roll), np.cos(roll)]])
12
13     Ry = np.array([[ np.cos(pitch), 0, np.sin(pitch)],
14                   [0, 1, 0],
15                   [-np.sin(pitch), 0, np.cos(pitch)]])
16
17     Rz = np.array([[np.cos(yaw), -np.sin(yaw), 0],
18                   [np.sin(yaw), np.cos(yaw), 0],
19                   [0, 0, 1]])
20
21     return Rz @ Ry @ Rx
22
23     # Define helix trajectory
24     t = np.linspace(0, 4*np.pi, 100)
25     x = 2 * np.cos(t)
26     y = 2 * np.sin(t)
27     z = t / (2*np.pi) * 5 # Scale z-axis to match approx height 5
28
29     # Orientation: Calculate yaw (heading) from tangent in XY-plane
30     dx_dt = -2 * np.sin(t)
31     dy_dt = 2 * np.cos(t)
32     yaw = np.arctan2(dy_dt, dx_dt)
33
34     # Calculate pitch (angle of ascent)
35     pitch = np.arctan2(np.gradient(z), np.sqrt(dx_dt**2 + dy_dt**2))
36
37     # Assume zero roll for simplicity
38     roll = np.zeros_like(t)
39
40     # Plot setup
41     fig = plt.figure(figsize=(8, 8))
42     ax = fig.add_subplot(111, projection='3d')
43
44     # Plot helix
45     ax.plot(x, y, z, 'k-', linewidth=2, label='Helix Trajectory')
46
47     # Select points to show local frames
48     num_points = 20
49     indices = np.linspace(0, len(t)-1, num_points, dtype=int)
50
51     arrow_length = 0.4
52
53     # Plot local coordinate frames with colored arrows
54     for i in indices:
55         pos = np.array([x[i], y[i], z[i]])
56         R = rotation_matrix(roll[i], pitch[i], yaw[i])
57
58         # Local axes arrows
59         x_axis = R @ np.array([arrow_length, 0, 0])
60         y_axis = R @ np.array([0, arrow_length, 0])
61         z_axis = R @ np.array([0, 0, arrow_length])
62
63         ax.quiver(*pos, *x_axis, color='r', linewidth=1.5)
64         ax.quiver(*pos, *y_axis, color='g', linewidth=1.5)
65         ax.quiver(*pos, *z_axis, color='b', linewidth=1.5)
66
67     # Mark sampled points on the helix

```

```
68 ax.scatter(x[indices], y[indices], z[indices], color='green', s=40)
69
70 # Set labels and grid
71 ax.set_xlabel('X (m)')
72 ax.set_ylabel('Y (m)')
73 ax.set_zlabel('Z (m)')
74 ax.set_title('3D Spiral with Local Orientation Frames')
75 ax.grid(True)
76
77 plt.legend()
78 plt.show()
```