

```

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

# Degree of the fit curve
degree = 10

# Given data for t (in sec) and r (in mi)
data_r = np.array([
    [0, 22.6],
    [5, 18.6],
    [10, 16.3],
    [15, 15.0],
    [20, 14.1],
    [30, 13.0],
    [40, 12.5],
    [50, 12.0],
    [60, 11.8],
    [70, 11.7],
    [80, 11.6],
    [90, 11.6],
    [100, 11.6],
    [110, 11.7],
    [120, 11.8]
])

# Time and distance at which we will calculate velocity
# Change the corresponding values of time and distance at which you need to
know the velocity and acceleration

t_val = 40
r_val = 12.5

t = data_r[:, 0] # First column (time, in seconds)
r = data_r[:, 1] # Second column (distance, in miles)

p_r = np.polyfit(t, r, degree) # Gives coefficients of the fit polynomial
curve
p_r_dot = np.polyder(p_r) # Taking derivative of the fit curve (r dot)
p_r_ddot = np.polyder(p_r_dot) # Taking derivative of the derivative of
fit curve (r double dot)

r_dot_val = np.polyval(p_r_dot, t_val) # Getting r dot at t = 40 sec
r_ddot_val = np.polyval(p_r_ddot, t_val) # Getting r double dot at t = 40
sec

# Plotting the fit curve (r - t graph)
r_fit_val = np.polyval(p_r, t) # Gets r values using fit curve a t given t
values
plt.plot(t, r, 'o', label='Data')
plt.plot(t, r_fit_val, '-', label=f'Polynomial Fit (degree {degree})')
plt.xlabel('t (in seconds)')
plt.ylabel('r (in miles)')
plt.title(f'Polynomial Fit (degree {degree})')
plt.legend()
plt.grid(True)
plt.show()

# # Given data for t (in sec) and theta (in radians (degrees are converted

```

```

to radians)
data_theta = np.array([[0, 110.5*math.pi/180],
                        [5, 100.0*math.pi/180],
                        [10, 91.0*math.pi/180],
                        [15, 83.7*math.pi/180],
                        [20, 77.7*math.pi/180],
                        [30, 67.7*math.pi/180],
                        [40, 58.6*math.pi/180],
                        [50, 52.0*math.pi/180],
                        [60, 45.0*math.pi/180],
                        [70, 38.5*math.pi/180],
                        [80, 32.8*math.pi/180],
                        [90, 27.0*math.pi/180],
                        [100, 21.6*math.pi/180],
                        [110, 16.8*math.pi/180],
                        [120, 12.0*math.pi/180]])

t = data_theta[:, 0] # First column (time, in seconds)
theta = data_theta[:, 1] # Second column (theta, in radians)

p_theta = np.polyfit(t, theta, degree) # Gives coefficients of the fit
polynomial curve
p_theta_dot = np.polyder(p_theta) # Taking derivative of the fit curve
(theta dot)
p_theta_ddot = np.polyder(p_theta_dot) # Taking derivative of the
derivative of fit curve (theta double dot)

theta_dot_val = np.polyval(p_theta_dot, t_val) # Getting theta dot at t =
40 sec
theta_ddot_val = np.polyval(p_theta_ddot, t_val) # Getting theta double
dot at t = 40 sec

# Plotting the fit curve (theta - t graph)
r_fit_theta = np.polyval(p_theta, t) # Gets theta values using fit curve a
t given t values
plt.plot(t, theta, 'o', label='Data')
plt.plot(t, r_fit_theta, '-', label=f'Polynomial Fit (degree {degree})')
plt.xlabel('t (in seconds)')
plt.ylabel('theta (in radians)')
plt.title(f'Polynomial Fit (degree {degree})')
plt.legend()
plt.grid(True)
plt.show()

print(theta_dot_val, theta_ddot_val, r_dot_val, r_ddot_val)

# Calculating velocity (using radial coordinates)
print(f'Here, e_r and e_theta are unit vectors along radian and angular
direction as usual.')
print()

print(f'The velocity at t = {t_val}s is')
print(f'({r_dot_val}) e_r + ({r_val * theta_dot_val}) e_theta (in mi/sec)')
print()
vel_mag = (r_dot_val ** 2 + (r_val * theta_dot_val) ** 2) ** 0.5
print(f'The magnitude of velocity is {vel_mag} (in mi/sec)')

print()

```

```
# Calculating acceleration (using radial coordinates)
print(f'The acceleration at t = {t_val}s is')
print(f'({r_ddot_val - r_val * theta_dot_val ** 2}) e_r + ({r_val *
theta_ddot_val + 2 * r_dot_val * theta_dot_val}) e_theta (in mi/sec^2)')
print()
acc_mag = ((r_ddot_val - r_val * theta_dot_val ** 2) ** 2 + (r_val *
theta_ddot_val + 2 * r_dot_val * theta_dot_val) ** 2) ** 0.5
print(f'The magnitude of acceleration is {acc_mag} (in mi/sec^2)')
```