Readme:

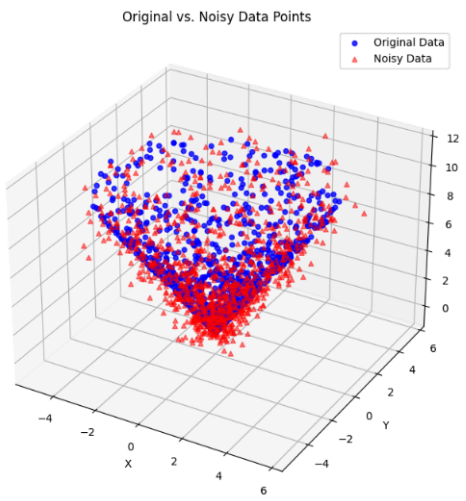
The first task was to generate synthetic data points that lie on a cone's surface.

```python
def generate_cone_points(num_points=1000, height=10, radius=5):
    theta = np.random.uniform(0, 2 * np.pi, num_points)
    z = np.random.uniform(0, height, num_points)
    r = (radius / height) * z
    x = r * np.cos(theta)
    y = r * np.sin(theta)
    return x, y, z
```

- Parameters:
    - theta angles are uniformly distributed around the circle.
    - z values are uniformly distributed along the height of the cone.
    - r (radial distance) varies linearly with z to ensure that points lie on a cone. The slope radius / height ensures the correct tapering of the cone.

```python
noise_scale = 0.5
x_noisy = x + np.random.normal(0, noise_scale, x.shape)
y_noisy = y + np.random.normal(0, noise_scale, y.shape)
z_noisy = z + np.random.normal(0, noise_scale, z.shape)
```

- Noise Addition: Gaussian noise with a mean of 0 and a standard deviation defined by noise_scale is added to each coordinate.



This is the plot for noisy data and original data for noise_scale=0.5.

Fitting a Cone

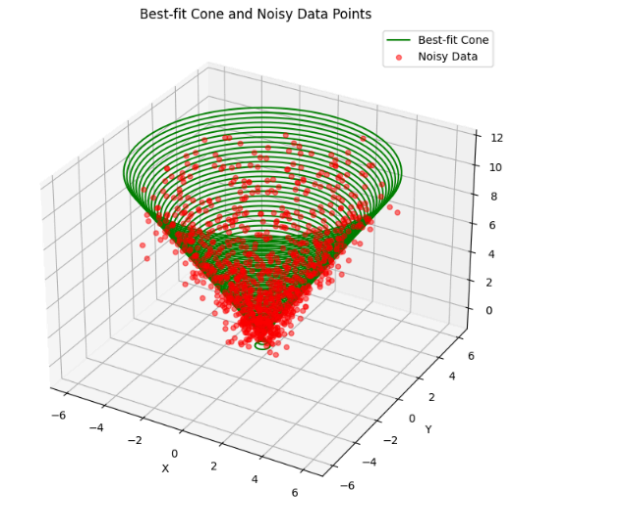
```
def cone_fit(params, x, y, z):
    px, py, pz, angle = params
    r = np.sqrt((x - px)**2 + (y - py)**2)
    ideal_z = pz + r / np.tan(angle)
    return np.sum((z - ideal_z)**2)
```

- Cost Function: Computes the sum of squared distances between the actual z-coordinates and the z-coordinates predicted by a cone with given apex position (px, py, pz) and angle.

Visualization

```
px, py, pz, angle = best_params
for z_level in z_levels:
    r_level = (z_level - pz) * np.tan(angle)
    x_level = px + r_level * np.cos(theta_fit)
    y_level = py + r_level * np.sin(theta_fit)
    z_level_array = np.full_like(x_level, z_level)
    ax.plot(x_level, y_level, z_level_array, color='green', label='Best-fit Cone' if z_level == z_levels[0] else "")
```

Plots circles at different z levels, representing cross-sections of the cone. The radius of each circle depends on its z value and the cone's angle.



Best-fit parameters (px, py, pz, angle): [3.04132180e-02 2.91648919e-06 3.37503075e-02 4.84882546e-01]