

We used chatGPT to find the correct functions to format the graphs correctly. This includes functions like `figsize` and `set_xlabel()`.

Add axis titles to subplots using python matplotlib



To add axis titles to subplots in Matplotlib, you can use the `set_xlabel` and `set_ylabel` methods for each axis object. Here's a simple example of how to do this in Python using Matplotlib:

```
python Copy code

import numpy as np
import matplotlib.pyplot as plt

# Sample data
x = np.linspace(0, 10, 100)
y1 = np.sin(x)
y2 = np.cos(x)

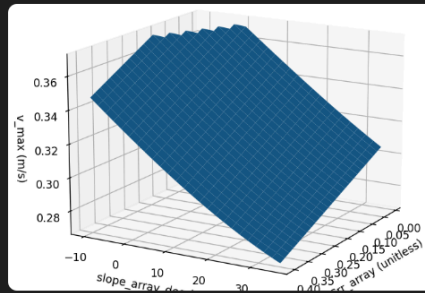
# Create subplots
fig, (ax1, ax2) = plt.subplots(2, figsize=(6, 8))

# Plot data on each subplot
ax1.plot(x, y1)
ax2.plot(x, y2)

# Add axis titles (labels)
ax1.set_xlabel('X-axis (Time)')
ax1.set_ylabel('Y-axis (Amplitude)')
ax1.set_title('Sine Wave')

ax2.set_xlabel('X-axis (Time)')
ax2.set_ylabel('Y-axis (Amplitude)')
```

We also used ChatGPT to figure out how to make our 3D plot more legible:



How do I make the `crr_array` axis units more spread out and legible



To make the `crr_array` axis units more spread out and legible, you can apply two improvements:

1. **Increase spacing between the tick marks:** Reduce the number of ticks on the `crr_array` axis so they don't appear so close together.
2. **Rotate the tick labels:** Rotate the labels on the `crr_array` axis to make them easier to read.

### Updated Code for Better Axis Formatting:

Here's how you can adjust both the tick spacing and label orientation:

```
python Copy code  
  
# Define fewer tick locations on the Crr_array axis to spread them out  
crr_ticks = np.linspace(0.01, 0.4, 5) # Adjust number of ticks  
  
# Set the ticks and labels  
ax.set_xticks(crr_ticks) # Set the tick locations on the Crr axis  
ax.set_xticklabels([f"{tick:.2f}" for tick in crr_ticks], rotation=45) # Rotate labels for
```

how do I keep the ticks already generated by my code and still rotate them



If you want to **keep the ticks already generated by your code** and simply **rotate the labels for better readability**, you can modify just the rotation part without changing the ticks.

To do this, you don't need to change the tick locations (`ax.set_xticks()`), but you can use the `ax.set_xticklabels()` method to rotate the existing labels.

### Here's how to update your code:

Add the following lines after plotting the surface but **before** `plt.show()`:

```
python Copy code  
  
# Rotate the existing tick labels for Crr_array  
for label in ax.get_xticklabels():  
    label.set_rotation(45)
```