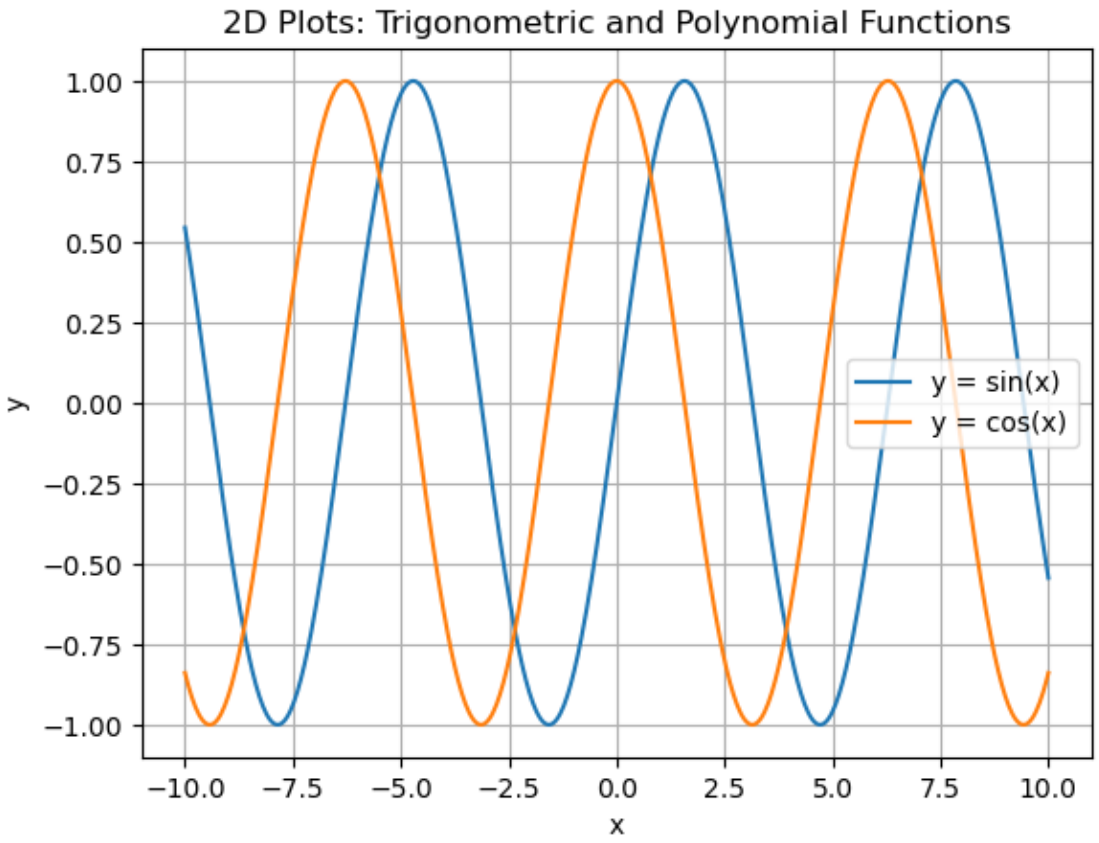
LAb2

January 31, 2025

1.2D Plots (Sine and Cosine)

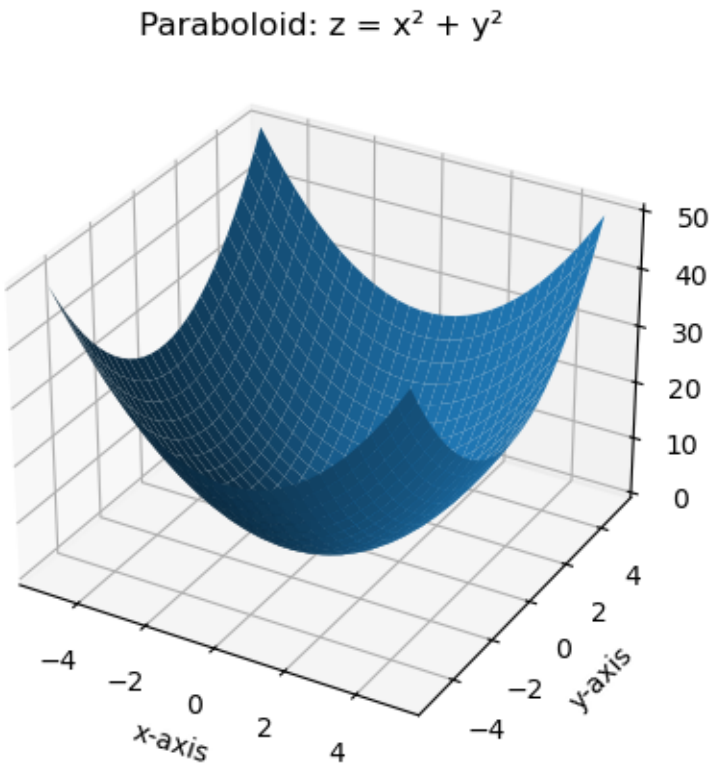
[56]:
```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
# Define the domain
x = np.linspace(-10, 10, 500)

# Functions
y_sin = np.sin(x)
y_cos = np.cos(x)
fig=plt.figure()
ax=fig.add_subplot()
# Plotting
ax.plot(x, y_sin, label="y = sin(x)")
ax.plot(x, y_cos, label="y = cos(x)")

ax.set_xlabel("x")
ax.set_ylabel("y")
ax.set_title("2D Plots: Trigonometric and Polynomial Functions")
ax.legend()
ax.grid()
plt.show()
```



3D Plot (Paraboloid)

[58]:
```
from mpl_toolkits.mplot3d import Axes3D

# Define the grid
x = np.linspace(-5, 5, 30)
y = np.linspace(-5, 5, 30)
x, y = np.meshgrid(x, y)

# Function
z_paraboloid = x**2 + y**2

# 3D Plot
fig = plt.figure()

ax = fig.add_subplot( projection="3d")
ax.plot_surface(x, y, z_paraboloid)
ax.set_title("Paraboloid: z = x² + y²")
ax.set_xlabel("x-axis")
ax.set_ylabel("y-axis")
```

```
ax.set_zlabel("z-axis")

plt.show()
```

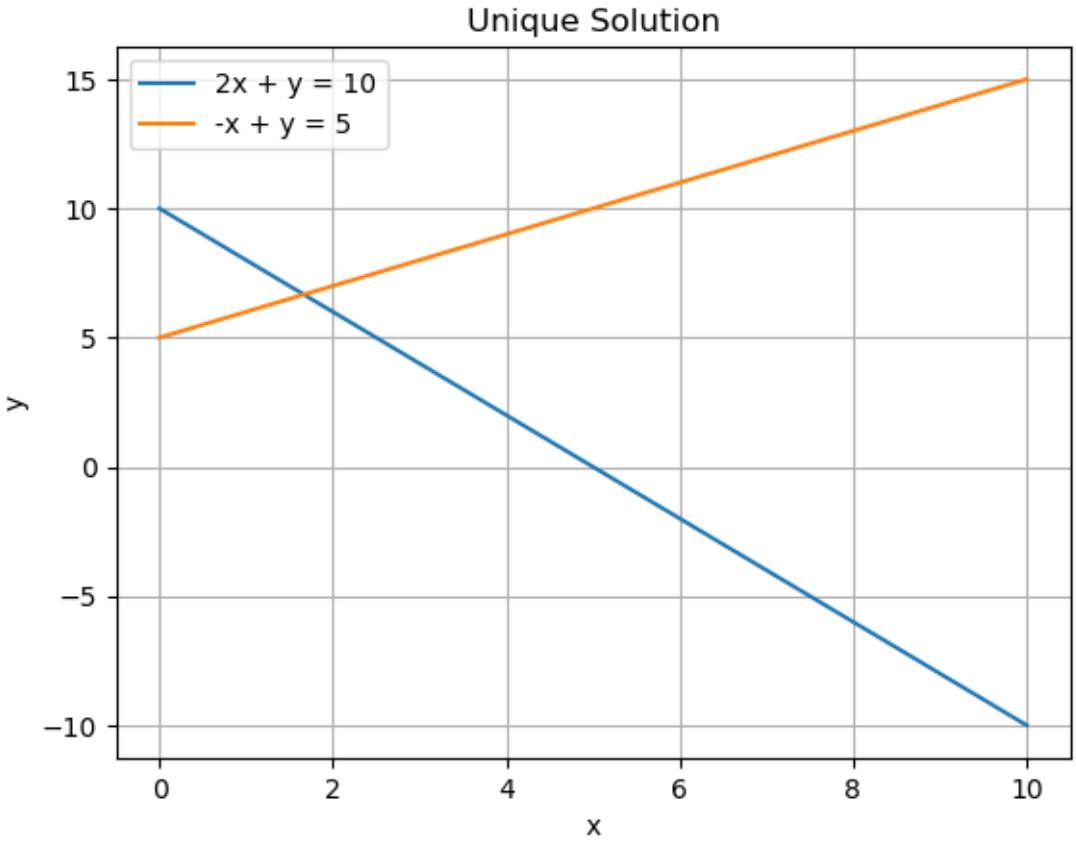


Unique Solution- The system:

$$2x + y = 10 \; and \; -x + y = 5$$

[60]:
```
import numpy as np
import matplotlib.pyplot as plt

# Define the range for x
x = np.linspace(0, 10, 500)

# Equations
y1 = 10 - 2*x  # 2x + y = 10
y2 = x + 5     # -x + y = 5

# Plot
fig=plt.figure()
ax=fig.add_subplot()
ax.plot(x, y1, label="2x + y = 10")
ax.plot(x, y2, label="-x + y = 5")
```

```
ax.set_xlabel("x")
ax.set_ylabel("y")
ax.set_title("Unique Solution")
ax.grid()
ax.legend()
plt.show()
```

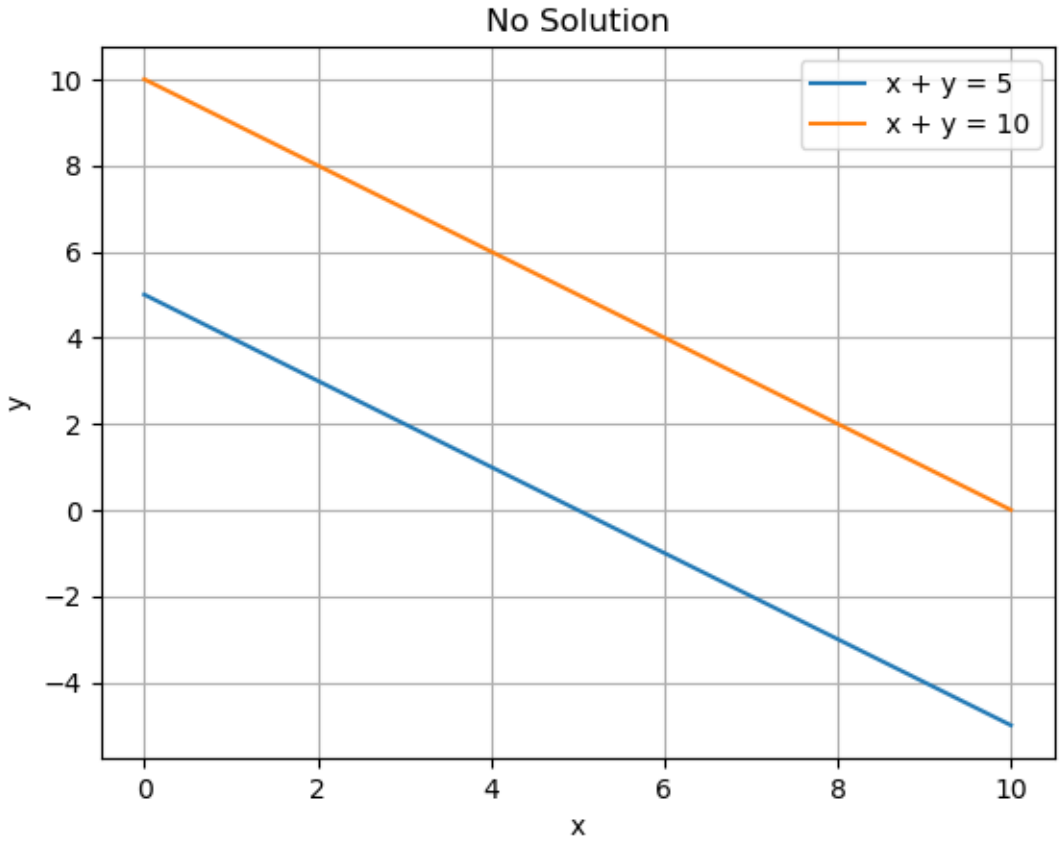


No Solution-The system:

$$x + y = 5 \; and \; x + y = 10$$

[62]:
```
# Define the range for x
x = np.linspace(0, 10, 500)

# Equations
y1 = 5 - x  # x + y = 5
y2 = 10 - x  # x + y = 10

# Plot
fig=plt.figure()
ax=fig.add_subplot()
```

```
ax.plot(x, y1, label="x + y = 5")
ax.plot(x, y2, label="x + y = 10")
ax.set_xlabel("x")
ax.set_ylabel("y")
ax.set_title("No Solution")
ax.grid()
ax.legend()
plt.show()
```

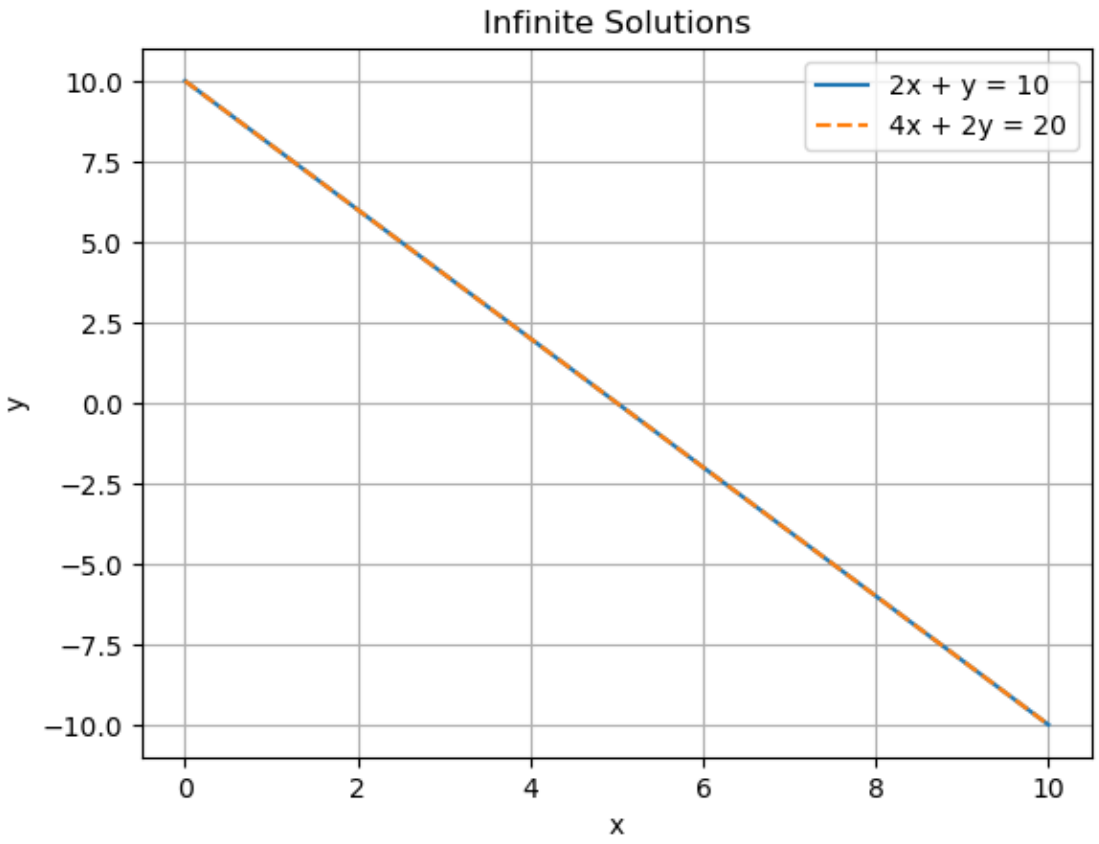


Infinite Solutions The system:

$$2x + y = 10 \quad and \quad 4x + 2y = 20$$

[64]:
```
# Define the range for x
x = np.linspace(0, 10, 500)

# Equations
y1 = 10 - 2*x  # 2x + y = 10
y2 = 10 - 2*x  # 4x + 2y = 20
```

```
# Plot
fig=plt.figure()
ax=fig.add_subplot()
ax.plot(x, y1, label="2x + y = 10")
ax.plot(x, y2, label="4x + 2y = 20", linestyle="dashed")
ax.set_xlabel("x")
ax.set_ylabel("y")
ax.set_title("Infinite Solutions")
ax.grid()
ax.legend()
plt.show()
```

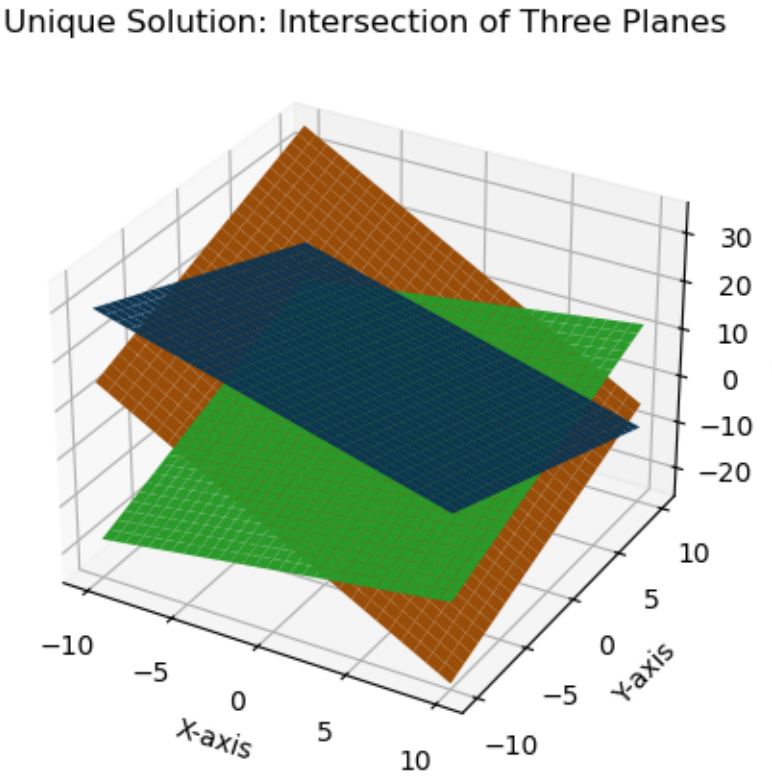


1 3D Visualization of Three Equations with Three Variables

Unique Solution The system:

$$x + y + z = 10, 2x - y + z = 5 \quad and \quad -x + 2y - z = 3$$

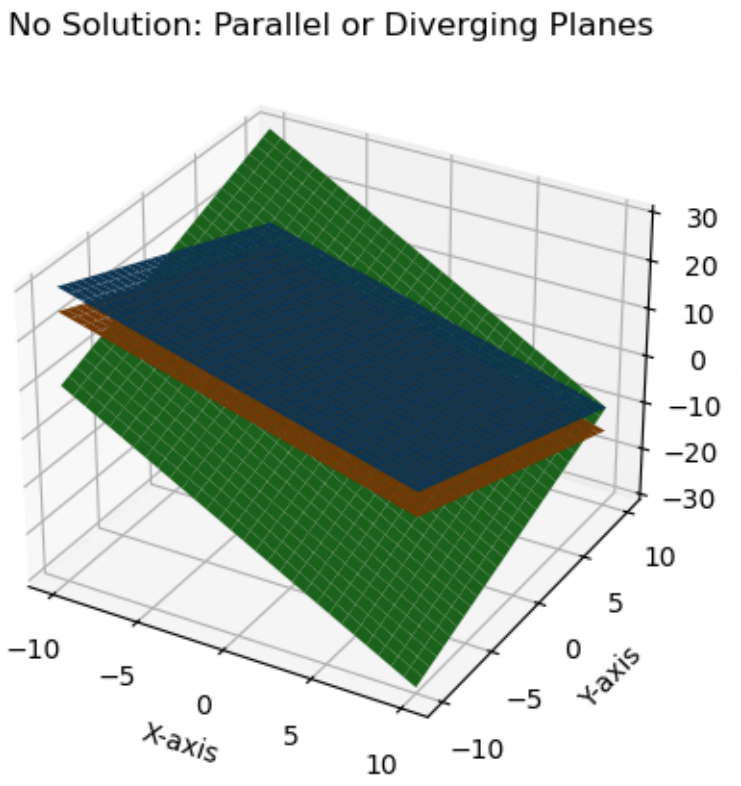
[67]:
```python
from mpl_toolkits.mplot3d import Axes3D

# Define the grid
x, y = np.linspace(-10, 10, 30), np.linspace(-10, 10, 30)
x, y = np.meshgrid(x, y)

# Planes
z1 = 10 - x - y         # x + y + z = 10
z2 = 5 - 2*x + y        # 2x - y + z = 5
z3 = x/2 + y - 3        # -x + 2y - z = 3

# 3D Plot
fig = plt.figure()
ax = fig.add_subplot(projection="3d")
ax.plot_surface(x, y, z1, label="Plane 1")
ax.plot_surface(x, y, z2, label="Plane 2")
ax.plot_surface(x, y, z3, label="Plane 3")

# Labels
ax.set_xlabel("X-axis")
ax.set_ylabel("Y-axis")
ax.set_zlabel("Z-axis")
ax.set_title("Unique Solution: Intersection of Three Planes")
plt.show()
```



No Solution The system:

$$x + y + z = 10, x + y + z = 5 \quad and \quad 2x - y + z = 0$$

[69]:
```
# Planes
z1 = 10 - x - y         # x + y + z = 10
z2 = 5 - x - y          # x + y + z = 5
z3 = -2*x + y           # 2x - y + z = 0

# 3D Plot
fig = plt.figure()
ax = fig.add_subplot( projection="3d")
ax.plot_surface(x, y, z1, label="Plane 1")
ax.plot_surface(x, y, z2, label="Plane 2")
ax.plot_surface(x, y, z3, label="Plane 3")

# Labels
ax.set_xlabel("X-axis")
ax.set_ylabel("Y-axis")
ax.set_zlabel("Z-axis")
ax.set_title("No Solution: Parallel or Diverging Planes")
plt.show()
```

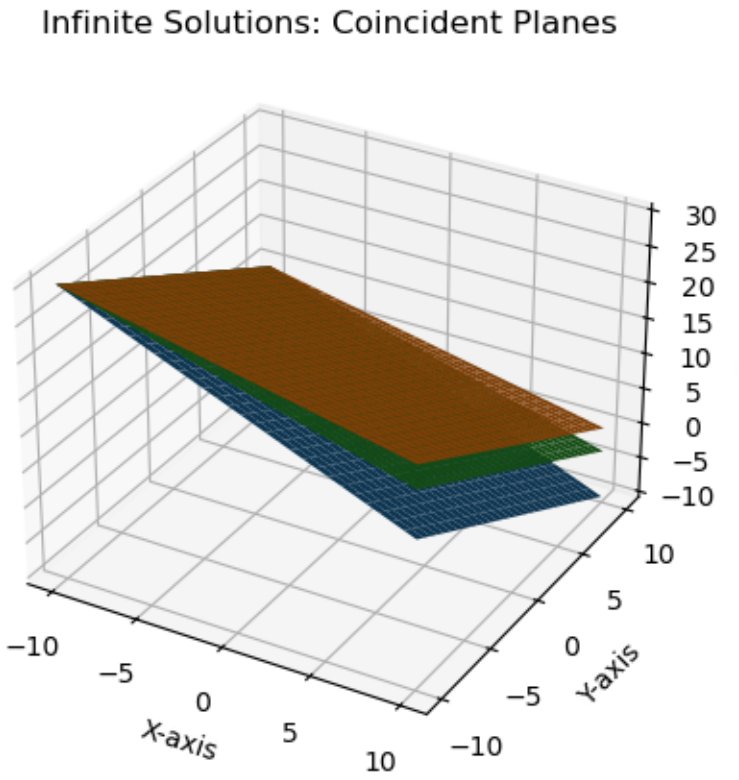


Infinite Solutions The system:

$$x + y + z = 10, 2x + 2y + 2z = 20 \quad and \quad -x - y - z = -10$$

[71]:
```
# Planes
z1 = 10 - x - y          # x + y + z = 10
z2 = (30 - x -2* y)/2          # x + 2y + 2z = 30
z3 = (40 - 2*x - 3*y)/3          # 2x + 3y +3z = 40

# 3D Plot
fig = plt.figure()
ax = fig.add_subplot(projection="3d")
ax.plot_surface(x, y, z1, label="Plane 1")
ax.plot_surface(x, y, z2, label="Plane 2")
ax.plot_surface(x, y, z3, label="Plane 3")

# Labels
ax.set_xlabel("X-axis")
ax.set_ylabel("Y-axis")
ax.set_zlabel("Z-axis")
ax.set_title("Infinite Solutions: Coincident Planes")
```

```
plt.show()
```


Infinite Solutions: Coincident Planes
30
25
20
15
10
5
0
−5
−10
10
5
0
−5
−10
Y-axis
−10
−5
0
5
10
X-axis

[ ]: