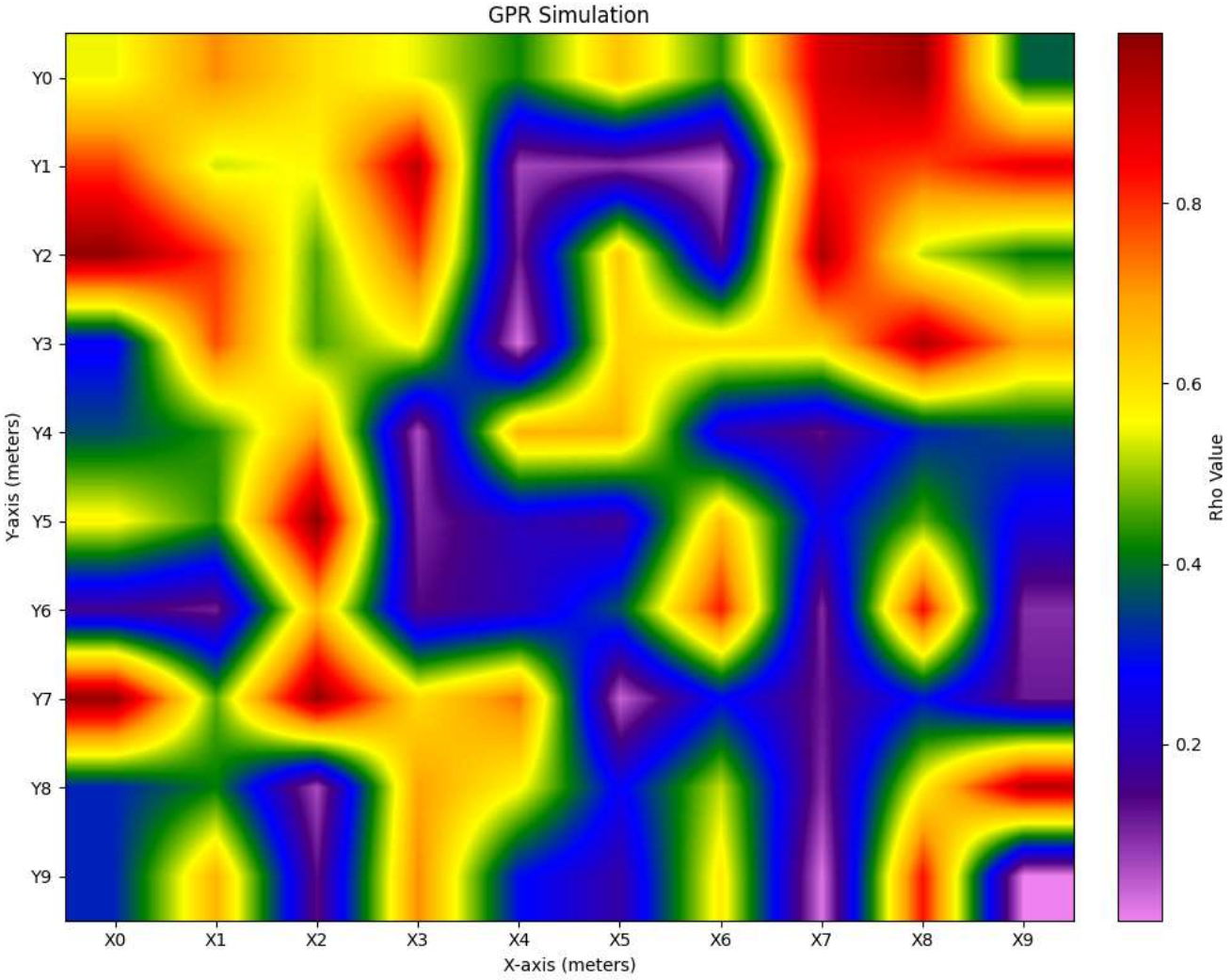
```
from google.colab import files

uploaded = files.upload()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving rho_matrix.csv to rho_matrix.csv

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.colors as mcolors

# Read the CSV file into a DataFrame
df = pd.read_csv('rho_matrix.csv', index_col=0)

# Extract the rho values into a 2D numpy array
rho = df.values

# Define a threshold for detecting metal
metal_threshold = 0.8  # Adjust this threshold as needed

# Find indices where rho is greater than the threshold
metal_indices = np.where(rho > metal_threshold)

# Convert indices to X and Y coordinates
metal_coordinates = [(df.columns[x], df.index[y]) for x, y in zip(*metal_indices)]

# Displaying detected metal coordinates
print("Detected metal at coordinates (X, Y):")
for coord in metal_coordinates:
    print(coord)

# Define the VIBGYOR colormap with a dark red for values close to 1
colors = ["violet", "indigo", "blue", "green", "yellow", "orange", "red", "darkred"]
nodes = [0.0, 0.14, 0.28, 0.42, 0.56, 0.70, 0.85, 1.0]  # Evenly spaced nodes for the colors
cmap = mcolors.LinearSegmentedColormap.from_list("vibgyor", list(zip(nodes, colors)))

# Plotting the rho values with the specified colormap
fig, ax = plt.subplots(figsize=(10, 8))  # Larger figure size
cax = ax.imshow(rho, interpolation='bilinear', cmap=cmap, aspect='auto')  # Smoother color transition
fig.colorbar(cax, label="Rho Value", fraction=0.046, pad=0.04)  # Refined colorbar

# Set the axis labels and title
ax.set_xlabel('X-axis (meters)')
ax.set_ylabel('Y-axis (meters)')
ax.set_title('GPR Simulation')

# Optionally, set the x and y ticks to represent the actual coordinates
ax.set_xticks(np.arange(len(df.columns)))
ax.set_yticks(np.arange(len(df.index)))
ax.set_xticklabels(df.columns)
ax.set_yticklabels(df.index)

# Adjust the padding between and around subplots and display the plot
plt.tight_layout()
plt.show()
```

```
Detected metal at coordinates (X, Y):
('X0', 'Y7')
('X0', 'Y8')
('X1', 'Y3')
('X1', 'Y7')
('X1', 'Y9')
('X2', 'Y0')
('X2', 'Y7')
('X3', 'Y8')
('X5', 'Y2')
('X6', 'Y6')
('X6', 'Y8')
('X7', 'Y0')
('X7', 'Y2')
('X8', 'Y9')
('X9', 'Y8')
```



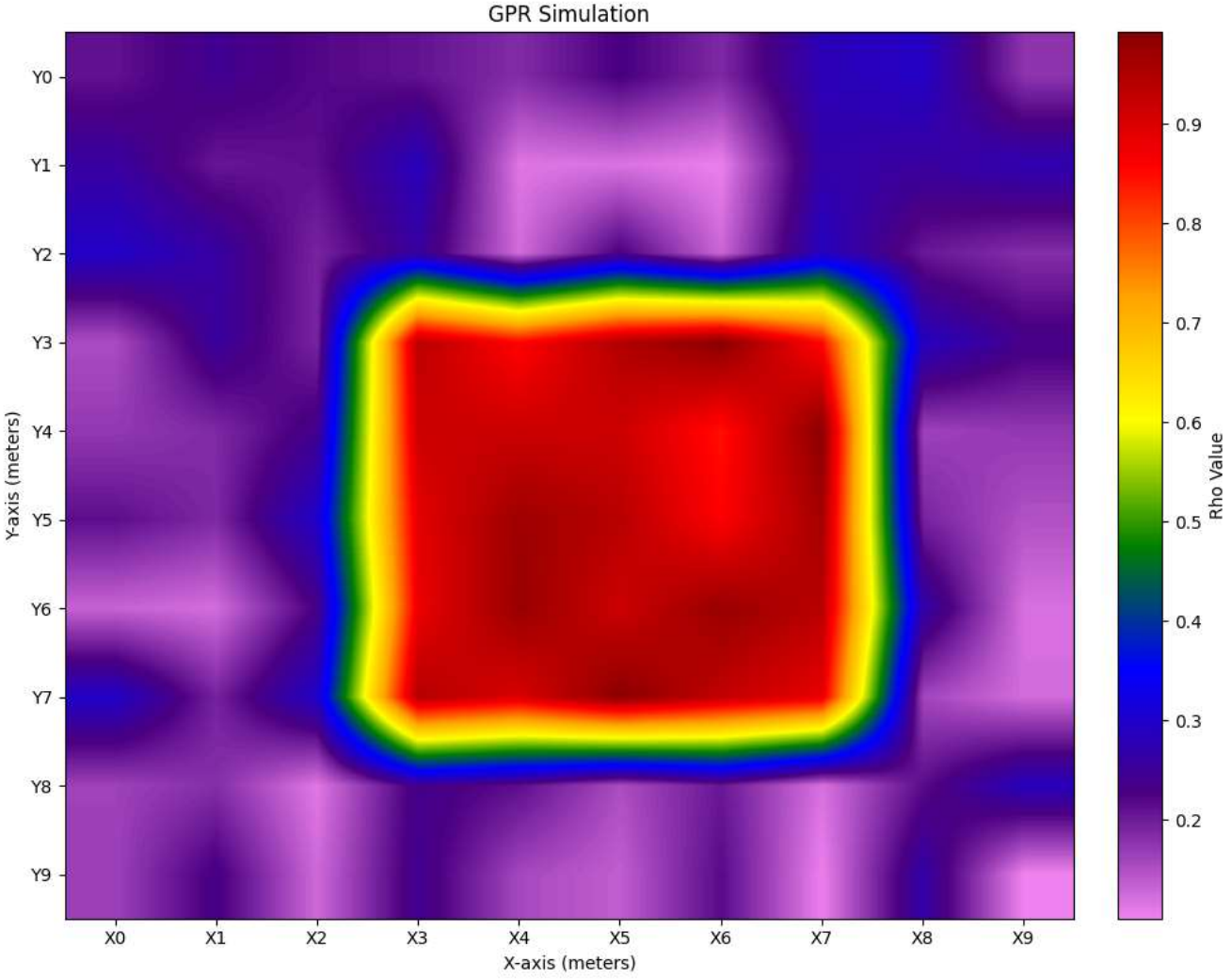

```
from google.colab import files

uploaded = files.upload()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving rho_matrix_10x10_realistic.csv to rho_matrix_10x10_realistic.csv

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.colors as mcolors

# Read the CSV file into a DataFrame
df = pd.read_csv('rho_matrix_10x10_realistic.csv', index_col=0)

# Extract the rho values into a 2D numpy array
rho = df.values

# Define a threshold for detecting metal
metal_threshold = 0.8  # Adjust this threshold as needed

# Find indices where rho is greater than the threshold
metal_indices = np.where(rho > metal_threshold)

# Convert indices to X and Y coordinates
metal_coordinates = [(df.columns[x], df.index[y]) for x, y in zip(*metal_indices)]

# Displaying detected metal coordinates
print("Detected metal at coordinates (X, Y):")
for coord in metal_coordinates:
    print(coord)

# Define the VIBGYOR colormap with a dark red for values close to 1
colors = ["violet", "indigo", "blue", "green", "yellow", "orange", "red", "darkred"]
nodes = [0.0, 0.14, 0.28, 0.42, 0.56, 0.70, 0.85, 1.0]  # Evenly spaced nodes for the colors
cmap = mcolors.LinearSegmentedColormap.from_list("vibgyor", list(zip(nodes, colors)))

# Plotting the rho values with the specified colormap
fig, ax = plt.subplots(figsize=(10, 8))  # Larger figure size
cax = ax.imshow(rho, interpolation='bilinear', cmap=cmap, aspect='auto')  # Smoother color transition
fig.colorbar(cax, label="Rho Value", fraction=0.046, pad=0.04)  # Refined colorbar

# Set the axis labels and title
ax.set_xlabel('X-axis (meters)')
ax.set_ylabel('Y-axis (meters)')
ax.set_title('GPR Simulation')

# Optionally, set the x and y ticks to represent the actual coordinates
ax.set_xticks(np.arange(len(df.columns)))
ax.set_yticks(np.arange(len(df.index)))
ax.set_xticklabels(df.columns)
ax.set_yticklabels(df.index)

# Adjust the padding between and around subplots and display the plot
plt.tight_layout()
plt.show()
```

```
Detected metal at coordinates (X, Y):
('X3', 'Y3')
('X3', 'Y4')
('X3', 'Y5')
('X3', 'Y6')
('X3', 'Y7')
('X4', 'Y3')
('X4', 'Y4')
('X4', 'Y5')
('X4', 'Y6')
('X4', 'Y7')
('X5', 'Y3')
('X5', 'Y4')
('X5', 'Y5')
('X5', 'Y6')
('X5', 'Y7')
('X6', 'Y3')
('X6', 'Y4')
('X6', 'Y5')
('X6', 'Y6')
('X6', 'Y7')
('X7', 'Y3')
('X7', 'Y4')
('X7', 'Y5')
('X7', 'Y6')
('X7', 'Y7')
```




```
from google.colab import files

uploaded = files.upload()
```

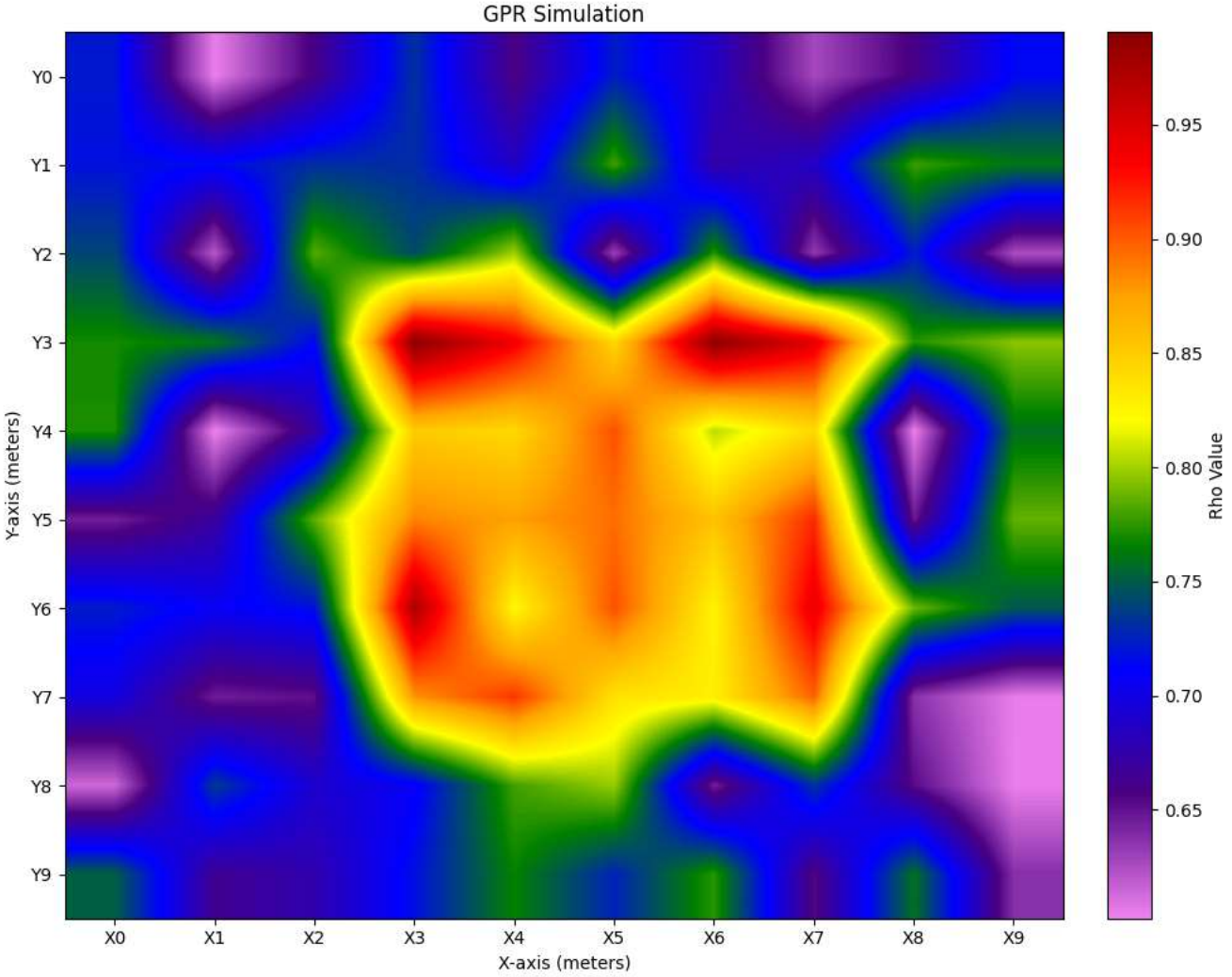
Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving rho_matrix_10x10_higher_non_metal.csv to rho_matrix_10x10_higher_non_metal.csv

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.colors as mcolors


df = pd.read_csv('rho_matrix_10x10_higher_non_metal.csv', index_col=0)
```

```
rho = df.values


metal_threshold = 0.8


metal_indices = np.where(rho > metal_threshold)


metal_coordinates = [(df.columns[x], df.index[y]) for x, y in zip(*metal_indices)]


print("Detected metal at coordinates (X, Y):")
for coord in metal_coordinates:
    print(coord)


colors = ["violet", "indigo", "blue", "green", "yellow", "orange", "red", "darkred"]
nodes = [0.0, 0.14, 0.28, 0.42, 0.56, 0.70, 0.85, 1.0]
cmap = mcolors.LinearSegmentedColormap.from_list("vibgyor", list(zip(nodes, colors)))


fig, ax = plt.subplots(figsize=(10, 8))
cax = ax.imshow(rho, interpolation='bilinear', cmap=cmap, aspect='auto')
fig.colorbar(cax, label="Rho Value", fraction=0.046, pad=0.04)


ax.set_xlabel('X-axis (meters)')
ax.set_ylabel('Y-axis (meters)')
ax.set_title('GPR Simulation')


ax.set_xticks(np.arange(len(df.columns)))
ax.set_yticks(np.arange(len(df.index)))
ax.set_xticklabels(df.columns)
ax.set_yticklabels(df.index)

plt.tight_layout()
plt.show()
```

```
Detected metal at coordinates (X, Y):
('X3', 'Y3')
('X3', 'Y4')
('X3', 'Y5')
('X3', 'Y6')
('X3', 'Y7')
('X4', 'Y3')
('X4', 'Y4')
('X4', 'Y5')
('X4', 'Y6')
('X4', 'Y7')
('X5', 'Y3')
('X5', 'Y4')
('X5', 'Y5')
('X5', 'Y6')
('X5', 'Y7')
('X6', 'Y3')
('X6', 'Y4')
('X6', 'Y5')
('X6', 'Y6')
('X6', 'Y7')
('X7', 'Y3')
('X7', 'Y4')
('X7', 'Y5')
('X7', 'Y6')
('X7', 'Y7')
```




```
from google.colab import files

uploaded = files.upload()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving rho_matrix_10x10_gradient.csv to rho_matrix_10x10_gradient.csv
```