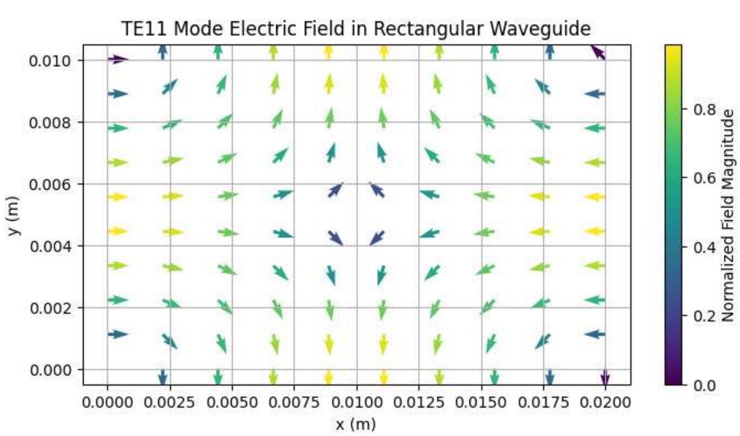
Practical 7

AIM - TE and TM modes

```
//MAYANK BARMAN 8562 (22025558001)
// PYTHON PROGRAM
TE11 mode
import numpy as np
import matplotlib.pyplot as plt
# Waveguide dimensions (in meters)
a = 0.02  # Width along x-axis (e.g., 2 cm)
b = 0.01  # Height along y-axis (e.g., 1 cm)
# Mode indices
m = 1
n = 1
# Create grid
x = np.linspace(0, a, 10)
y = np.linspace(0, b, 10)
X, Y = np.meshgrid(x, y)
# Compute transverse electric field components
Ex = np.cos(m * np.pi * X / a) * np.sin(n * np.pi * Y / b)
Ey = -np.sin(m * np.pi * X / a) * np.cos(n * np.pi * Y / b)
# Normalize for plotting
magnitude = np.sqrt(Ex*2 + Ey*2)
Ex /= magnitude
Ey /= magnitude
# Plot vector field
plt.figure(figsize=(8, 4))
plt.quiver(X, Y, Ex, Ey, magnitude, scale=25) # Changed to pass real parts to quiver
plt.title(f"TE{m}{n} Mode Electric Field in Rectangular Waveguide")
plt.xlabel("x (m)")
plt.ylabel("y (m)")
plt.colorbar(label='Normalized Field Magnitude')
plt.grid(True)
```

output -



```
#TM11 mode
import numpy as np
import matplotlib.pyplot as plt

# Waveguide dimensions (in meters)
a = 0.02  # Width along x-axis (e.g., 2 cm)
b = 0.01  # Height along y-axis (e.g., 1 cm)

# Mode indices
m = 1
n = 1

# Create grid
x = np.linspace(0, a, 10)
y = np.linspace(0, b, 10)
X, Y = np.meshgrid(x, y)


# Compute transverse electric field components
Bx = np.sin(m * np.pi * X / a) * np.cos(n * np.pi * Y / b)
By = -np.cos(m * np.pi * X / a) * np.sin(n * np.pi * Y / b)

# Normalize for plotting
magnitude = np.sqrt(Ex*2 + Ey*2)
Bx /= magnitude
By /= magnitude

# Plot vector field
plt.figure(figsize=(8, 4))
plt.quiver(X, Y, Bx, By, magnitude, scale=25) # Changed to pass real parts to quiver
plt.title(f"TM{m}{n} Mode Electric Field in Rectangular Waveguide")
plt.xlabel("x (m)")
plt.ylabel("y (m)")
plt.colorbar(label='Normalized Field Magnitude')
plt.grid(True)
```

Output -