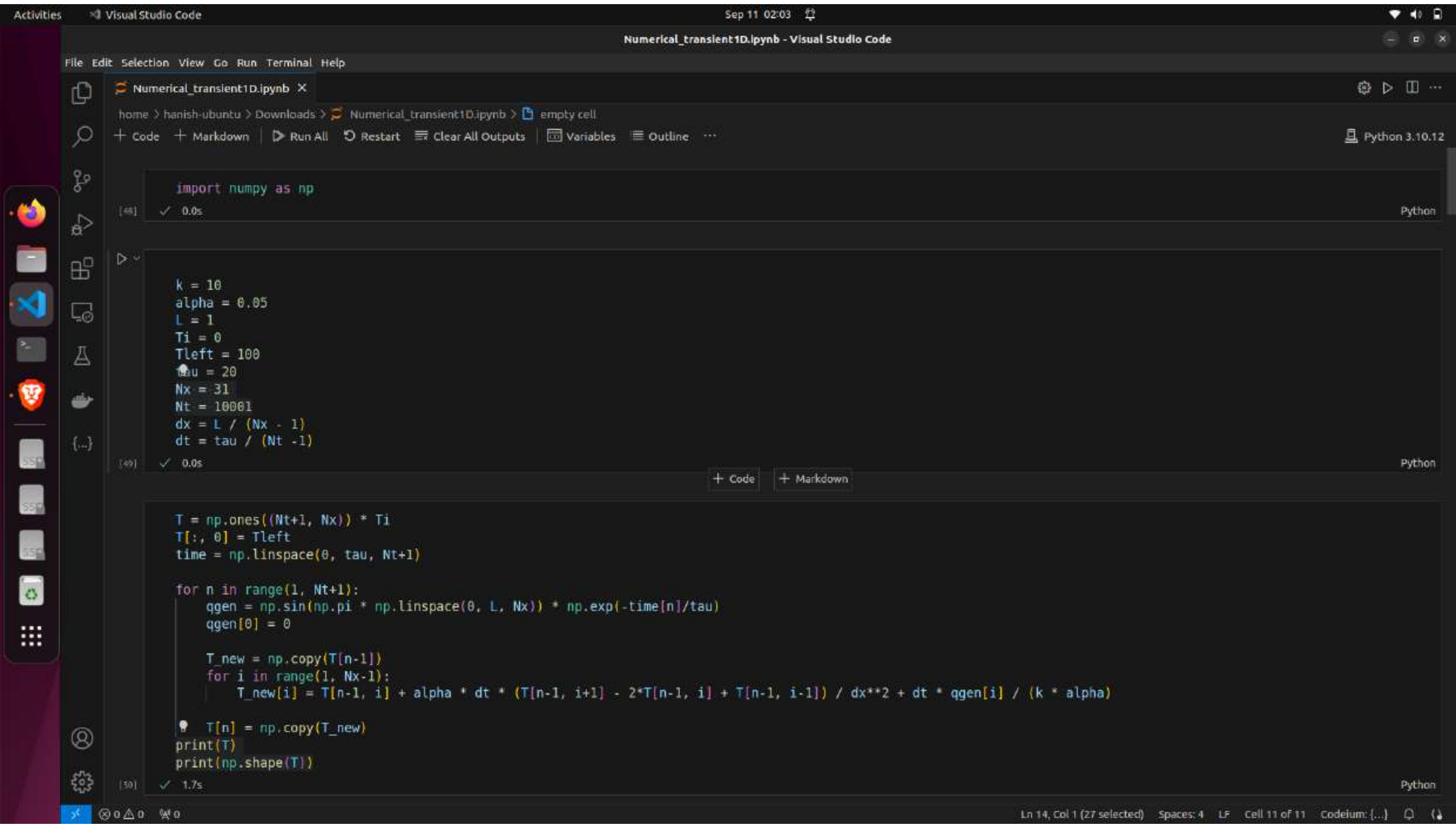

Activities
Visual Studio Code
Sep 11 02:03
Numerical_transient1D.ipynb - Visual Studio Code
File Edit Selection View Go Run Terminal Help
Numerical_transient1D.ipynb
home > hanish-ubuntu > Downloads > Numerical_transient1D.ipynb > empty cell
+ Code + Markdown | Run All Restart Clear All Outputs | Variables Outline ···
Python 3.10.12

```
import numpy as np
```

[48] ✓ 0.0s Python

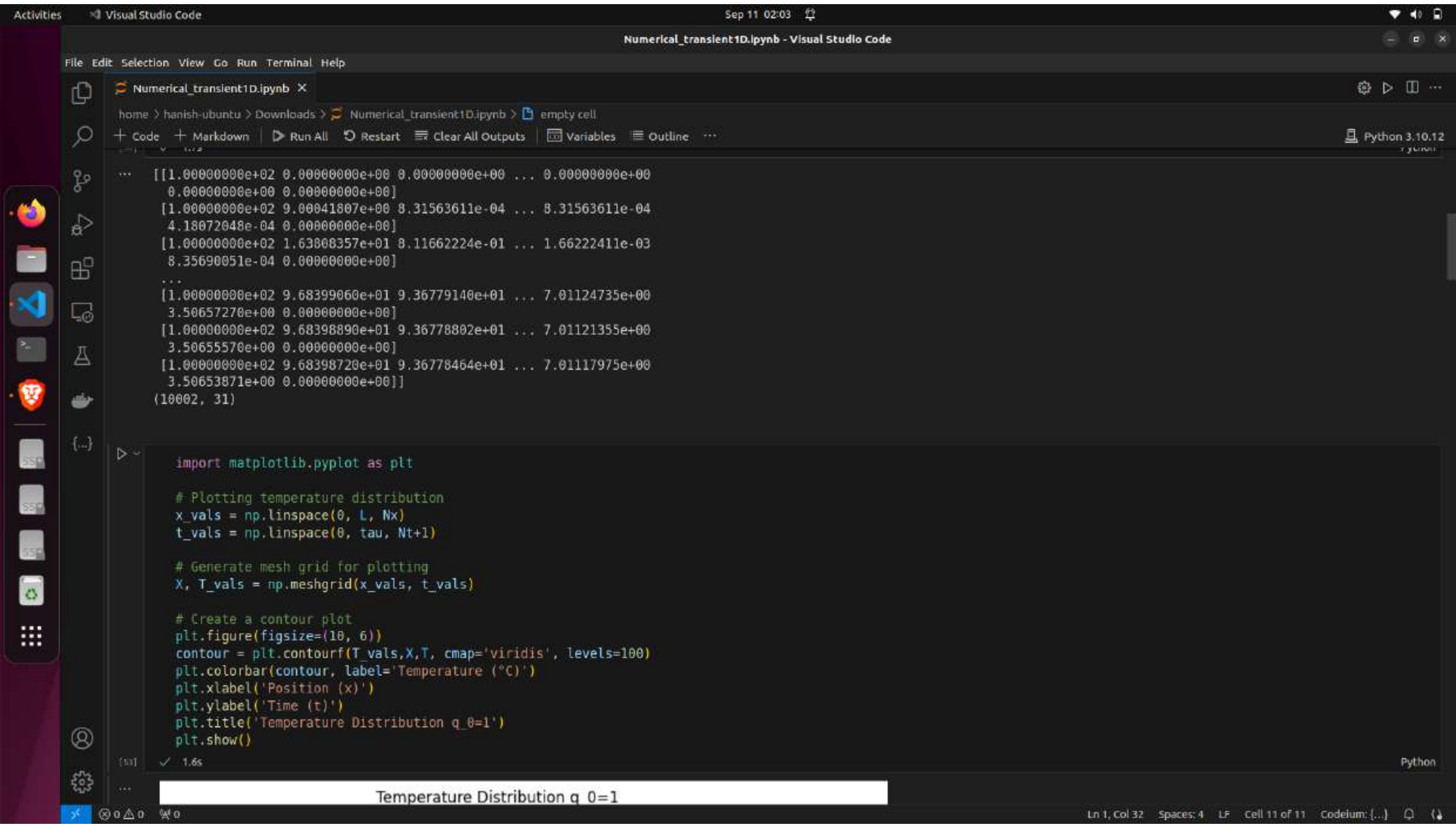
```
k = 10
alpha = 0.05
L = 1
Ti = 0
Tleft = 100
tau = 20
Nx = 31
Nt = 10001
dx = L / (Nx - 1)
dt = tau / (Nt -1)
```

[49] ✓ 0.0s Python

```
T = np.ones((Nt+1, Nx)) * Ti
T[:, 0] = Tleft
time = np.linspace(0, tau, Nt+1)

for n in range(1, Nt+1):
    qgen = np.sin(np.pi * np.linspace(0, L, Nx)) * np.exp(-time[n]/tau)
    qgen[0] = 0

    T_new = np.copy(T[n-1])
    for i in range(1, Nx-1):
        T_new[i] = T[n-1, i] + alpha * dt * (T[n-1, i+1] - 2*T[n-1, i] + T[n-1, i-1]) / dx**2 + dt * qgen[i] / (k * alpha)

    T[n] = np.copy(T_new)
print(T)
print(np.shape(T))
```

[50] ✓ 1.7s Python

⊗ 0 ⚠ 0 0
Ln 14, Col 1 (27 selected) Spaces: 4 LF Cell 11 of 11 Codeium: {...}

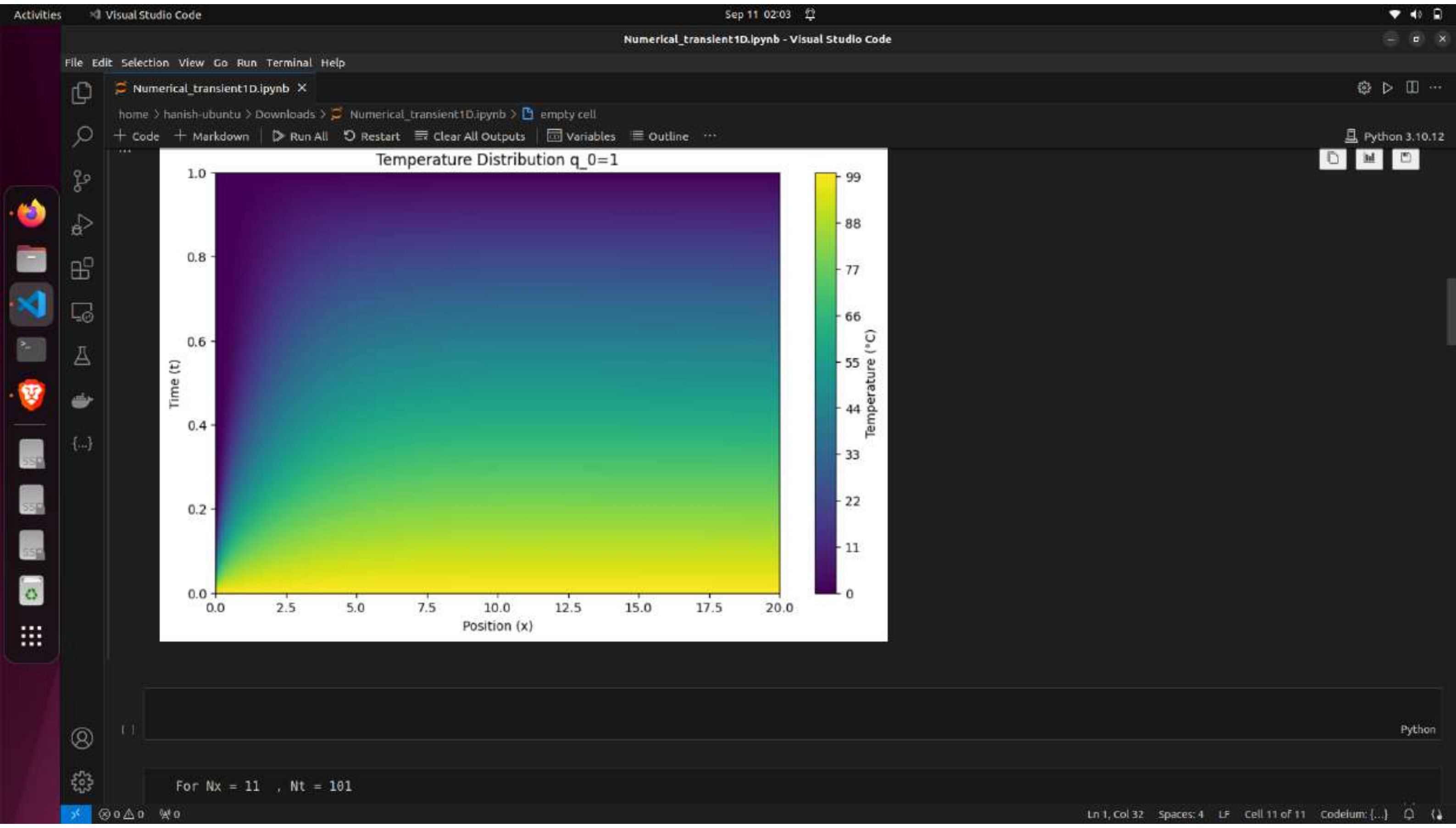

Activities
Visual Studio Code
Sep 11 02:03
Numerical_transient1D.ipynb - Visual Studio Code
File Edit Selection View Go Run Terminal Help
Numerical_transient1D.ipynb
home > hanish-ubuntu > Downloads > Numerical_transient1D.ipynb > empty cell
+ Code + Markdown Run All Restart Clear All Outputs Variables Outline ···
Python 3.10.12

```
[[1.00000000e+02 0.00000000e+00 0.00000000e+00 ... 0.00000000e+00
  0.00000000e+00 0.00000000e+00]
 [1.00000000e+02 9.00041807e+00 8.31563611e-04 ... 8.31563611e-04
  4.18072048e-04 0.00000000e+00]
 [1.00000000e+02 1.63808357e+01 8.11662224e-01 ... 1.66222411e-03
  8.35690051e-04 0.00000000e+00]
 ...
 [1.00000000e+02 9.68399060e+01 9.36779140e+01 ... 7.01124735e+00
  3.50657270e+00 0.00000000e+00]
 [1.00000000e+02 9.68398890e+01 9.36778802e+01 ... 7.01121355e+00
  3.50655570e+00 0.00000000e+00]
 [1.00000000e+02 9.68398720e+01 9.36778464e+01 ... 7.01117975e+00
  3.50653871e+00 0.00000000e+00]]
(10002, 31)
```

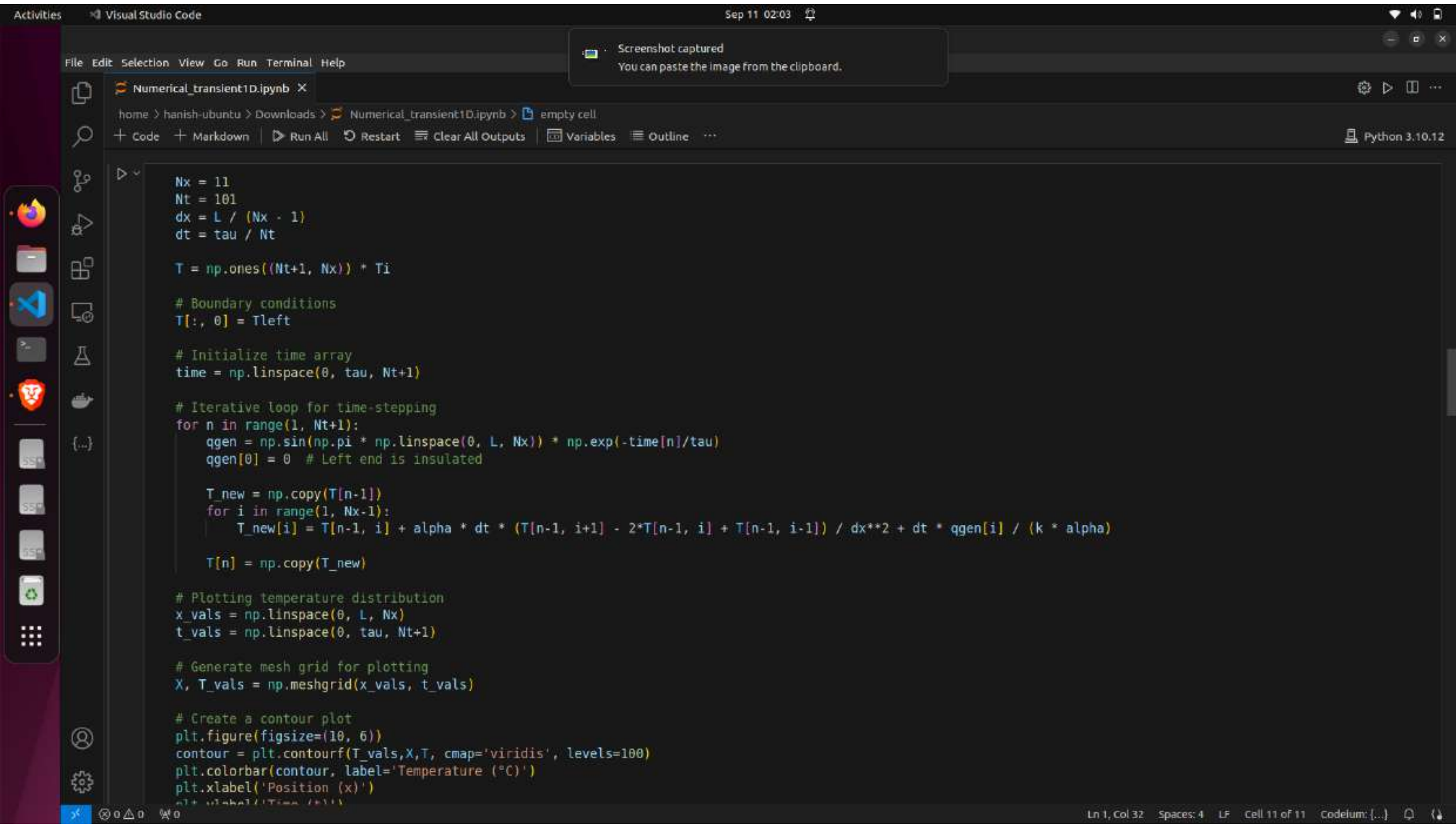
```python
import matplotlib.pyplot as plt

# Plotting temperature distribution
x_vals = np.linspace(0, L, Nx)
t_vals = np.linspace(0, tau, Nt+1)

# Generate mesh grid for plotting
X, T_vals = np.meshgrid(x_vals, t_vals)

# Create a contour plot
plt.figure(figsize=(10, 6))
contour = plt.contourf(T_vals,X,T, cmap='viridis', levels=100)
plt.colorbar(contour, label='Temperature (°C)')
plt.xlabel('Position (x)')
plt.ylabel('Time (t)')
plt.title('Temperature Distribution q_0=1')
plt.show()
```

```
[53] ✓ 1.6s
```

Temperature Distribution q 0=1

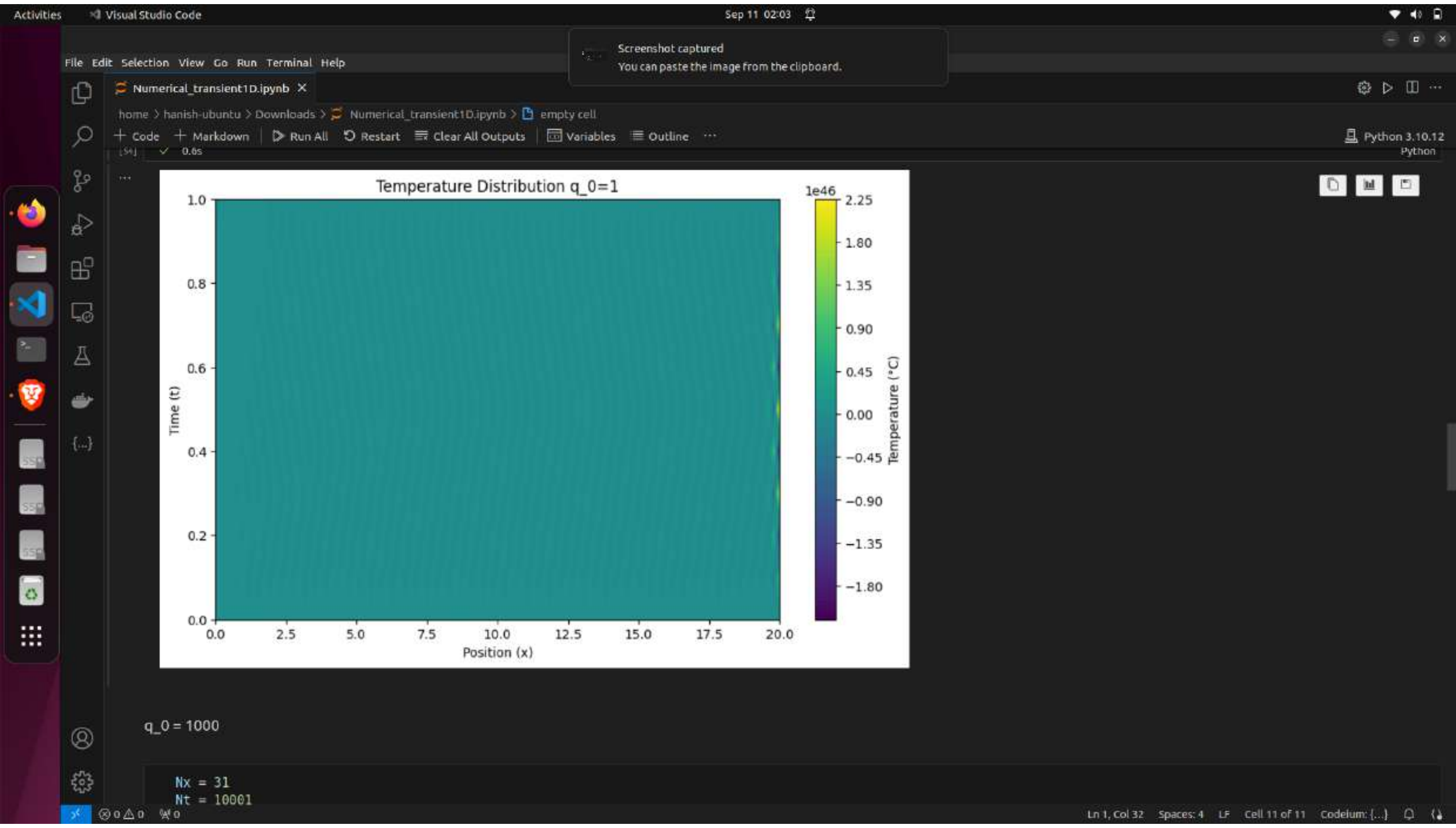

Activities
Visual Studio Code
Sep 11 02:03
Numerical_transient1D.ipynb - Visual Studio Code
File Edit Selection View Go Run Terminal Help
Numerical_transient1D.ipynb
home > hanish-ubuntu > Downloads > Numerical_transient1D.ipynb > empty cell
+ Code + Markdown Run All Restart Clear All Outputs Variables Outline
Python 3.10.12
Temperature Distribution q_0=1
Time (t)
1.0
0.8
0.6
0.4
0.2
0.0
Position (x)
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Temperature (°C)
99
88
77
66
55
44
33
22
11
0
Python
Ln 1, Col 32 Spaces: 4 LF Cell 11 of 11 Codeium: {...}

```
For Nx = 11  , Nt = 101
```


Activities
Visual Studio Code
Sep 11 02:03
File Edit Selection View Go Run Terminal Help
Numerical_transient1D.ipynb
home > hanish-ubuntu > Downloads > Numerical_transient1D.ipynb > empty cell
+ Code + Markdown | Run All Restart Clear All Outputs | Variables Outline ···
Python 3.10.12
Ln 1, Col 32 Spaces: 4 LF Cell 11 of 11 Codeium: {...}

Screenshot captured
You can paste the image from the clipboard.

```python
Nx = 11
Nt = 101
dx = L / (Nx - 1)
dt = tau / Nt

T = np.ones((Nt+1, Nx)) * Ti

# Boundary conditions
T[:, 0] = Tleft

# Initialize time array
time = np.linspace(0, tau, Nt+1)

# Iterative loop for time-stepping
for n in range(1, Nt+1):
    qgen = np.sin(np.pi * np.linspace(0, L, Nx)) * np.exp(-time[n]/tau)
    qgen[0] = 0  # Left end is insulated

    T_new = np.copy(T[n-1])
    for i in range(1, Nx-1):
        T_new[i] = T[n-1, i] + alpha * dt * (T[n-1, i+1] - 2*T[n-1, i] + T[n-1, i-1]) / dx**2 + dt * qgen[i] / (k * alpha)

    T[n] = np.copy(T_new)

# Plotting temperature distribution
x_vals = np.linspace(0, L, Nx)
t_vals = np.linspace(0, tau, Nt+1)

# Generate mesh grid for plotting
X, T_vals = np.meshgrid(x_vals, t_vals)

# Create a contour plot
plt.figure(figsize=(10, 6))
contour = plt.contourf(T_vals,X,T, cmap='viridis', levels=100)
plt.colorbar(contour, label='Temperature (°C)')
plt.xlabel('Position (x)')
plt.ylabel('Time (t)')
```

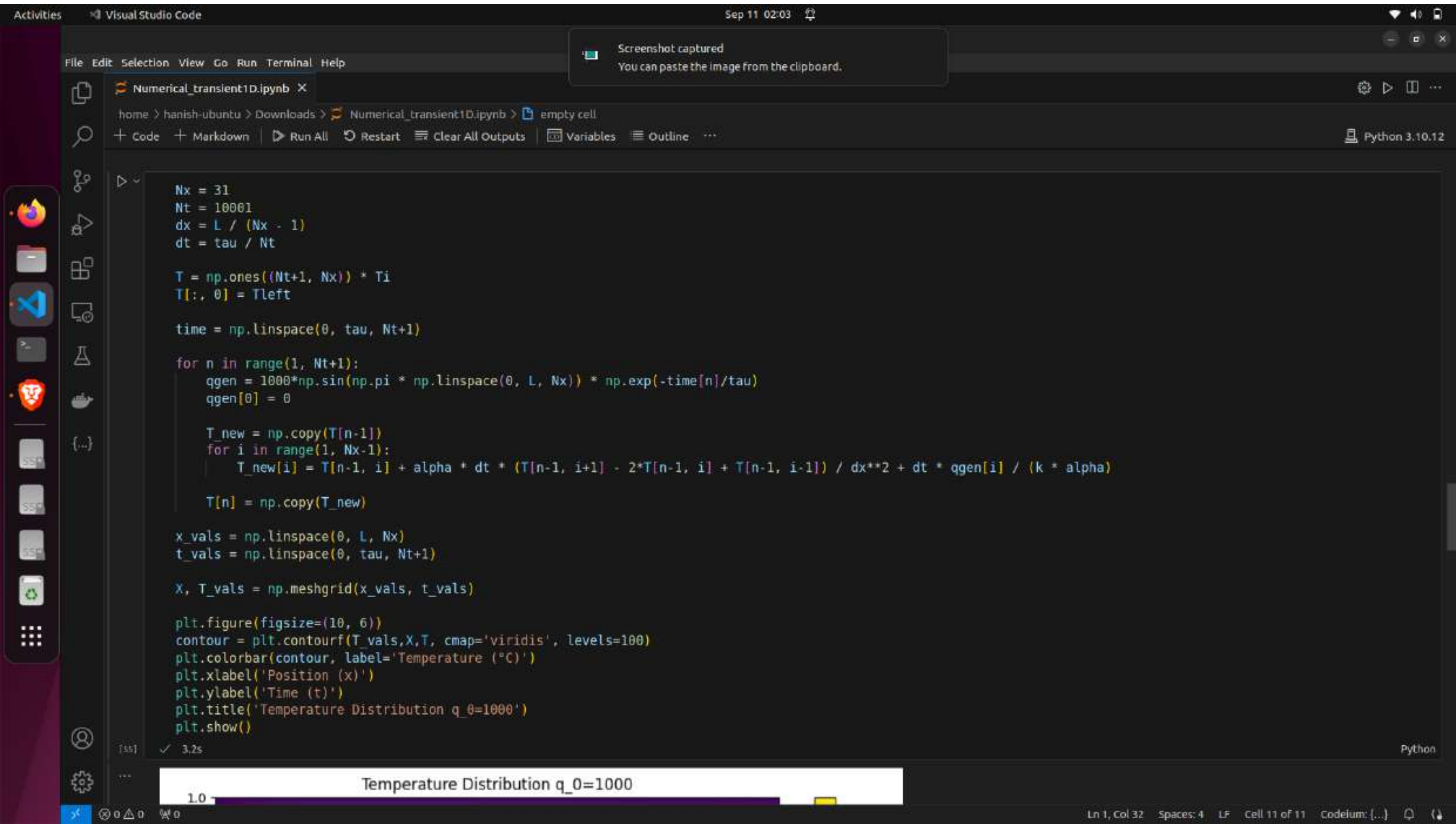


Activities
Visual Studio Code
Sep 11 02:03
File Edit Selection View Go Run Terminal Help
Numerical_transient1D.ipynb
home > hanish-ubuntu > Downloads > Numerical_transient1D.ipynb > empty cell
+ Code + Markdown Run All Restart Clear All Outputs Variables Outline ···
Python 3.10.12
Python
0.6s
Temperature Distribution q_0=1
Time (t)
Position (x)
Temperature (°C)
1e46
1.0
0.8
0.6
0.4
0.2
0.0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
2.25
1.80
1.35
0.90
0.45
0.00
−0.45
−0.90
−1.35
−1.80
Ln 1, Col 32 Spaces: 4 LF Cell 11 of 11 Codeium: {...}

Screenshot captured
You can paste the image from the clipboard.

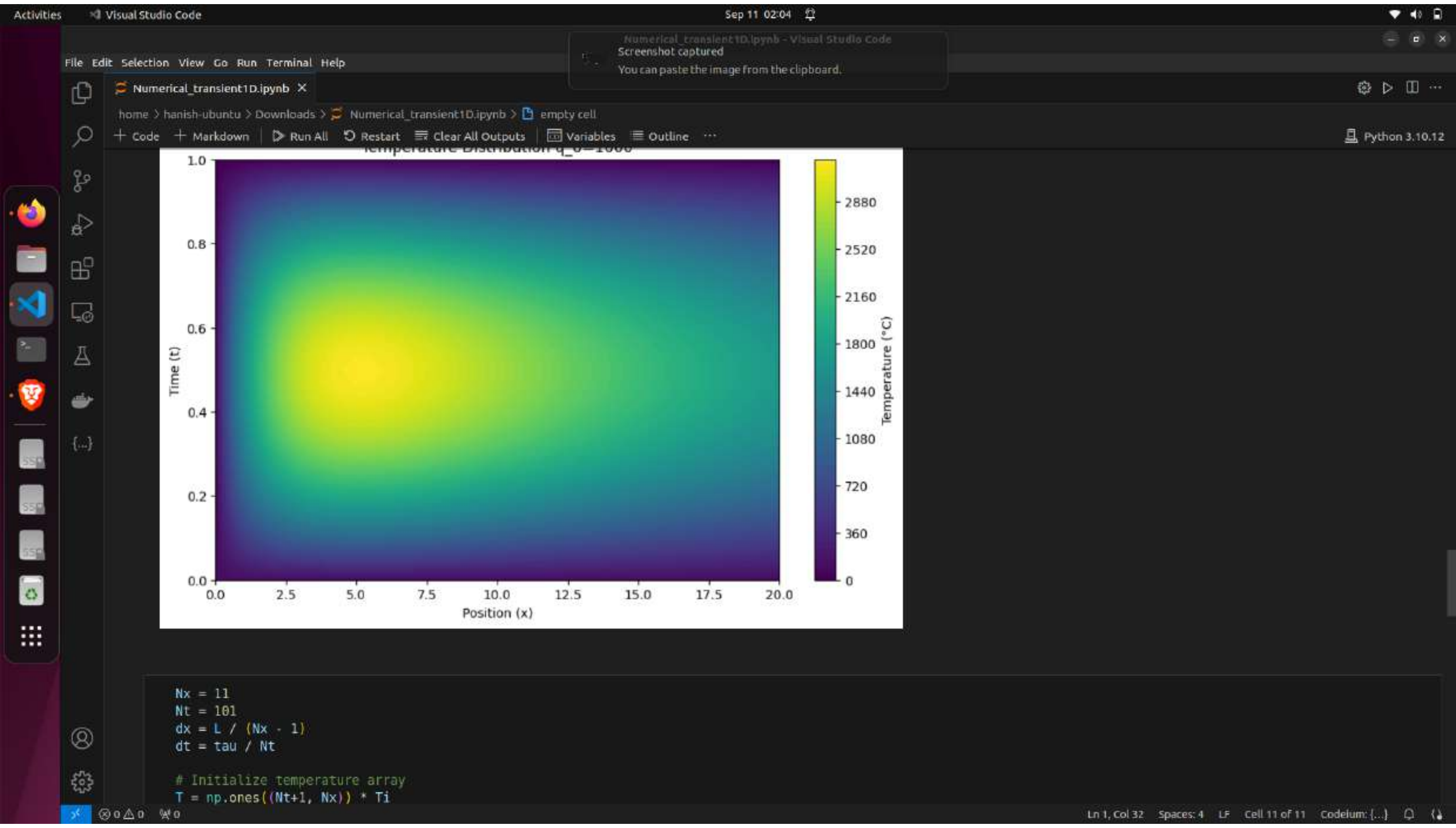
q_0 = 1000

```
Nx = 31
Nt = 10001
```


Activities
Visual Studio Code
Sep 11 02:03
File Edit Selection View Go Run Terminal Help
Numerical_transient1D.ipynb
home > hanish-ubuntu > Downloads > Numerical_transient1D.ipynb > empty cell
+ Code + Markdown Run All Restart Clear All Outputs Variables Outline ···
Python 3.10.12
[55]
✓ 3.2s
Python
Temperature Distribution q_0=1000
1.0
Ln 1, Col 32 Spaces: 4 LF Cell 11 of 11 Codeium: {...}

Screenshot captured
You can paste the image from the clipboard.

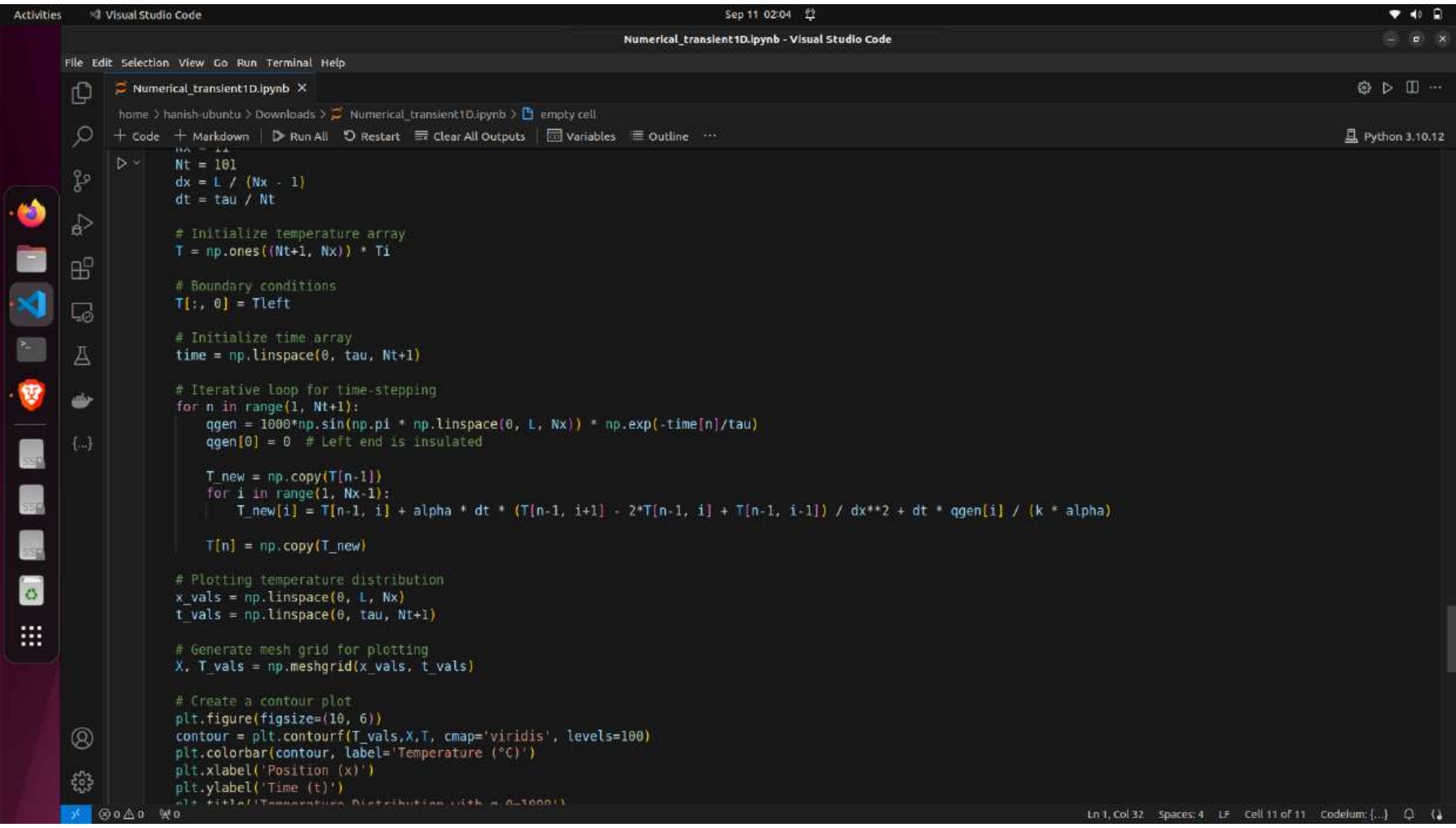
```python
Nx = 31
Nt = 10001
dx = L / (Nx - 1)
dt = tau / Nt

T = np.ones((Nt+1, Nx)) * Ti
T[:, 0] = Tleft

time = np.linspace(0, tau, Nt+1)

for n in range(1, Nt+1):
    qgen = 1000*np.sin(np.pi * np.linspace(0, L, Nx)) * np.exp(-time[n]/tau)
    qgen[0] = 0

    T_new = np.copy(T[n-1])
    for i in range(1, Nx-1):
        T_new[i] = T[n-1, i] + alpha * dt * (T[n-1, i+1] - 2*T[n-1, i] + T[n-1, i-1]) / dx**2 + dt * qgen[i] / (k * alpha)

    T[n] = np.copy(T_new)

x_vals = np.linspace(0, L, Nx)
t_vals = np.linspace(0, tau, Nt+1)

X, T_vals = np.meshgrid(x_vals, t_vals)

plt.figure(figsize=(10, 6))
contour = plt.contourf(T_vals,X,T, cmap='viridis', levels=100)
plt.colorbar(contour, label='Temperature (°C)')
plt.xlabel('Position (x)')
plt.ylabel('Time (t)')
plt.title('Temperature Distribution q_0=1000')
plt.show()
```


Activities
Visual Studio Code
Sep 11 02:04
Numerical_transient1D.ipynb - Visual Studio Code
File Edit Selection View Go Run Terminal Help
Numerical_transient1D.ipynb
home > hanish-ubuntu > Downloads > Numerical_transient1D.ipynb > empty cell
+ Code + Markdown Run All Restart Clear All Outputs Variables Outline
Python 3.10.12
Temperature Distribution q_0=1000
Time (t)
1.0
0.8
0.6
0.4
0.2
0.0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Position (x)
Temperature (°C)
2880
2520
2160
1800
1440
1080
720
360
0
Ln 1, Col 32 Spaces: 4 LF Cell 11 of 11 Codeium: {...}

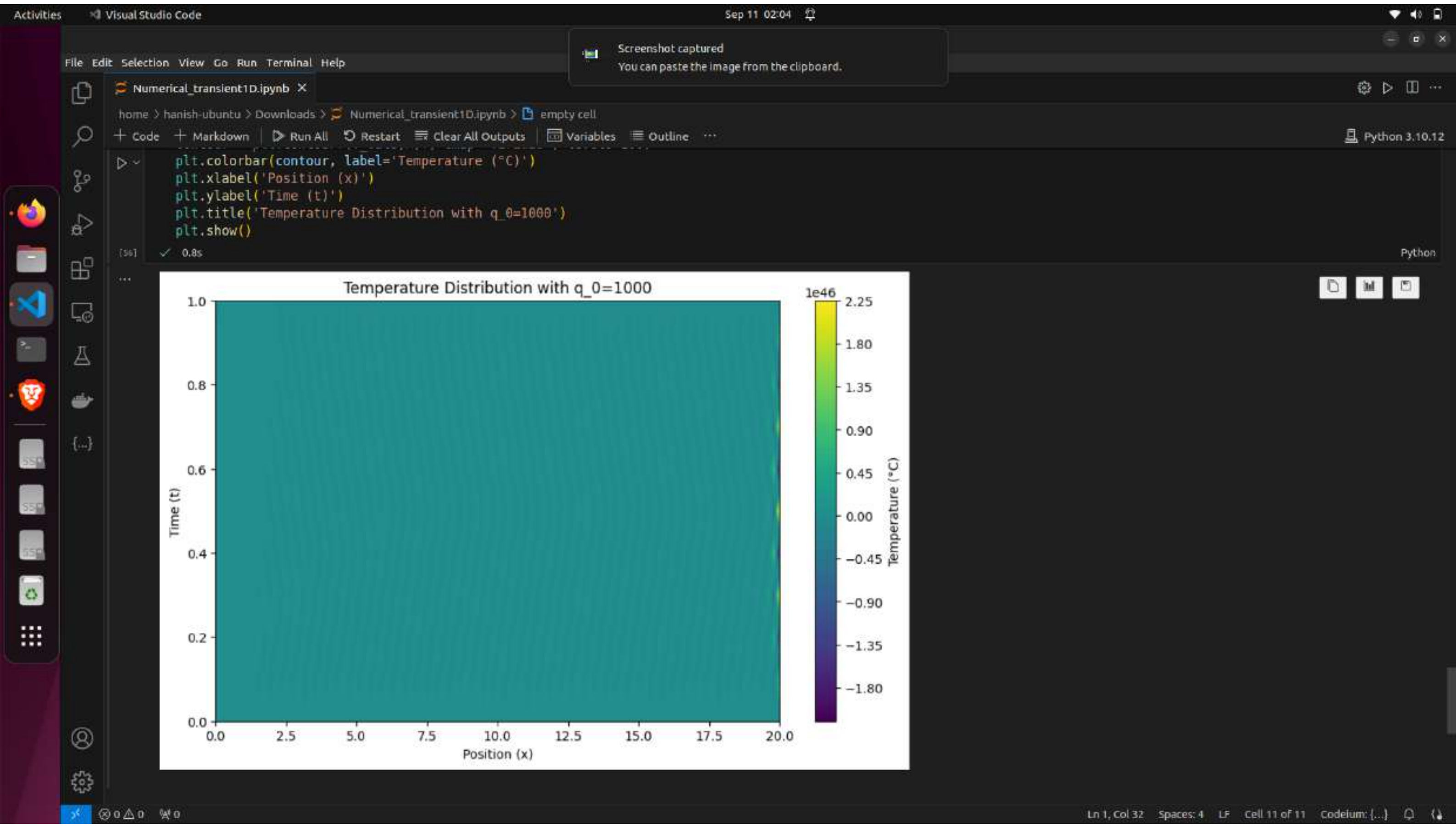
Screenshot captured

You can paste the image from the clipboard.

```
Nx = 11
Nt = 101
dx = L / (Nx - 1)
dt = tau / Nt

# Initialize temperature array
T = np.ones((Nt+1, Nx)) * Ti
```


Activities
Visual Studio Code
Sep 11 02:04
Numerical_transient1D.ipynb - Visual Studio Code
File Edit Selection View Go Run Terminal Help
Numerical_transient1D.ipynb
home > hanish-ubuntu > Downloads > Numerical_transient1D.ipynb > empty cell
+ Code + Markdown Run All Restart Clear All Outputs Variables Outline
Python 3.10.12
Ln 1, Col 32 Spaces: 4 LF Cell 11 of 11 Codeium: {...}

```python
Nx = 11
Nt = 101
dx = L / (Nx - 1)
dt = tau / Nt

# Initialize temperature array
T = np.ones((Nt+1, Nx)) * Ti

# Boundary conditions
T[:, 0] = Tleft

# Initialize time array
time = np.linspace(0, tau, Nt+1)

# Iterative loop for time-stepping
for n in range(1, Nt+1):
    qgen = 1000*np.sin(np.pi * np.linspace(0, L, Nx)) * np.exp(-time[n]/tau)
    qgen[0] = 0  # Left end is insulated

    T_new = np.copy(T[n-1])
    for i in range(1, Nx-1):
        T_new[i] = T[n-1, i] + alpha * dt * (T[n-1, i+1] - 2*T[n-1, i] + T[n-1, i-1]) / dx**2 + dt * qgen[i] / (k * alpha)

    T[n] = np.copy(T_new)

# Plotting temperature distribution
x_vals = np.linspace(0, L, Nx)
t_vals = np.linspace(0, tau, Nt+1)

# Generate mesh grid for plotting
X, T_vals = np.meshgrid(x_vals, t_vals)

# Create a contour plot
plt.figure(figsize=(10, 6))
contour = plt.contourf(T_vals,X,T, cmap='viridis', levels=100)
plt.colorbar(contour, label='Temperature (°C)')
plt.xlabel('Position (x)')
plt.ylabel('Time (t)')
plt.title('Temperature Distribution with q_0=1000')
```



Activities
Visual Studio Code
Sep 11 02:04
File Edit Selection View Go Run Terminal Help
Numerical_transient1D.ipynb
home > hanish-ubuntu > Downloads > Numerical_transient1D.ipynb > empty cell
+ Code + Markdown Run All Restart Clear All Outputs Variables Outline ···
Python 3.10.12
[56]
0.8s
Python
Ln 1, Col 32 Spaces: 4 LF Cell 11 of 11 Codeium: {...}

Screenshot captured
You can paste the image from the clipboard.

```
plt.colorbar(contour, label='Temperature (°C)')
plt.xlabel('Position (x)')
plt.ylabel('Time (t)')
plt.title('Temperature Distribution with q_0=1000')
plt.show()
```

Temperature Distribution with q_0=1000
1.0
0.8
0.6
0.4
0.2
0.0
Time (t)
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Position (x)
1e46
2.25
1.80
1.35
0.90
0.45
0.00
−0.45
−0.90
−1.35
−1.80
Temperature (°C)