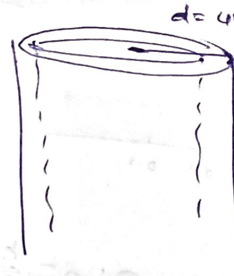
Heat Transfer Assignment 5.



① $h = 125\ W/m^2K$

$\rho = 1000\ kg/m^3$

$c = 4200\ J/kg\text{-}K$

$T_\infty = 90°c$

$T_i = 10°c$

$-E_{out} = E_{st}$

$hA(T - T_\infty) = -\rho V c \frac{dT}{dt}$

$\theta = T - T_\infty$

$\Rightarrow -\frac{hA\theta}{\rho V c} = \frac{d\theta}{dt}$

$\theta = \theta_i e^{-\frac{hA}{\rho Vc} t}$

i) $\frac{\theta}{\theta_i} = 0.99 = e^{-\frac{hA}{\rho Vc} t}$

$t = \frac{\rho Vc \ln \frac{1}{0.99}}{hA}$

$t = \frac{10^3 \times \pi R^2 L \, 4200}{125 \times 2\pi R L}$

$= \frac{10^5 \times 42 \times 20 \times 10^{-2}}{125 \times 2}$

$= 10^4 \times 42/125 = 0.33\ 3.36 \times 10^3\ sec.$

$\frac{T - T_\infty}{T_i - T_\infty} = e^{-\frac{hA}{\rho Vc} t}$

$\frac{0.99 T_\infty - T_\infty}{T_i - T_\infty} = e^{-\frac{hA}{\rho Vc} t}$

$e^{-\frac{hA}{\rho Vc} t} = \frac{0.01 T_\infty\ 90}{80}$

$t = -\ln \frac{0.09}{8} \times \frac{10^3 \times \pi R^2 \times 4200}{2\pi R L}$

$t = 15077.62$ sec

$t = 251.29$ min

ii) $Q = \int_0^\infty hA(T - T_\infty)\,dt$

$= \int_0^\infty hA\, \theta_i e^{-\frac{hA}{\rho Vc} t}\, dt$

$= \frac{hA\theta_i \; e^{-\frac{hAt}{\rho Vc}} \Big|_0^\infty}{-\frac{hA}{\rho Vc}}$

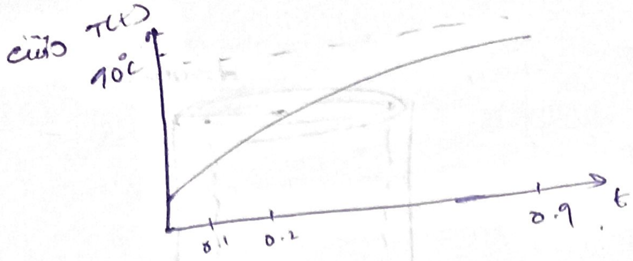
$= \rho V c \theta_i$

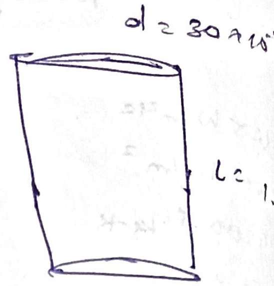
$= 10^3 \times \pi \left(\frac{d}{2}\right)^2 L \times 4200 \times (-80)$

$= -10^3 \times \pi\, 20^2 \times 10^{-4} \times 70 \times 10^{-2} \times 4200 \times 80$

$= 4\pi \times 7 \times 42 \times 10^3$

$= 3694.53 \times 10^3$





(2) $\cancel{\theta = \theta_i e^{-\frac{hA}{\rho V c} t}}$ $\qquad hA \frac{d(T-T_\infty)}{dt} = -\rho V c \frac{dT}{dt}$

$\frac{dT}{dt} = -\lambda (T - T_\infty) \qquad \lambda = \frac{hA}{\rho V c} = \frac{8}{10^3} \times \frac{2\pi R L}{\pi R^2 L \times 4178}$

$= \frac{8 \times 2}{10^3 \times 15 \times 10^{-2} \times 4178}$

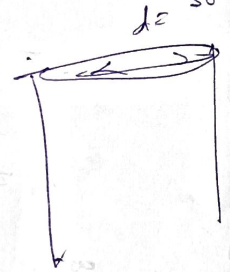
$\int_{t_0}^{t} dt = -\int_{T_0}^{T} \frac{dT}{\lambda (T - T_\infty)} \qquad = 2.55 \times 10^{-5}$

$\cancel{t - t_0 =}\ \Delta t = -\frac{1}{\lambda} \ln \frac{(T - T_\infty)}{T_0 - T_\infty}$

$\Delta t = -\frac{1}{2.55 \times 10^{-5}} \ln (25 - 20)$

$\frac{dT}{dt} = -\lambda (T - T_\infty) = -\lambda \times 5$

$T - T_i = -5 \lambda t$



(2) $\cancel{\Delta t =}\ \theta = \theta_i e^{-\frac{hA}{\rho V c} t}$

$\frac{T - T_\infty}{T_i - T_\infty} = e^{-\frac{8 \times 2\pi R L}{10^3 \times \pi R^2 L \times 4178} t}$

$\frac{25 - 20}{37 - 20} = e^{-\frac{8 \times 2}{10^3 \times 15 \times 10^{-2} \times 4178} t}$

$\frac{5}{17} = e^{-2.55 \times 10^{-5} t}$

$\ln 17/5 = 2.55 \times 10^{-5} t$

$t = \cancel{1.82 \times 10^4}$ sec $\quad 4.799 \times 10^4$

$t = \cancel{212}$ min $\quad 800$ min $= 13$ hr 20 min

Time of death = ~~4:38~~ pm. 3.40 a.m

Q. 3

Using $\frac{T - T_s}{T_i - T_s} = erf\left(\frac{x}{\sqrt{4\alpha t}}\right)$

~~$\frac{57-180}{}$~~

$\frac{40-50}{35-50} = erf\left(\frac{5\times10^{-2}}{2\sqrt{7\times10^{-7}t}}\right)$

$\frac{2}{3} = 0.66 = erf\left(\sqrt{\frac{25\times10^{-4}}{4\times7\times10^{-7}t}}\right)$

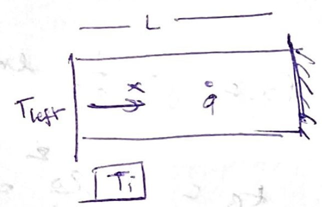
$0.68 = \sqrt{\frac{25\times10^{-4}}{4\times7\times10^{-7}t}}$

$t = \frac{25\times10^{3}}{4\times7\times0.68^2}$

$t = 1.93\times10^{3}$ sec

$t = 32.18$ min

Q. 5



$\frac{k}{\rho c}\frac{\partial^2 T}{\partial x^2} + \frac{q_0}{\rho c} = \frac{\partial T}{\partial t}$

& $\dot{q} = q_0 \sin\left(\frac{\pi x}{L}\right)e^{-t/\tau}$

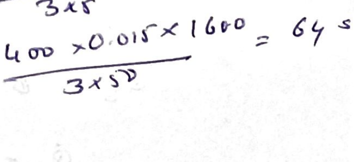
$k=10$, $\alpha=0.05$, $L=1$, $T_i=0$, $T_{left}=100$, $\tau=20$

comparison:
i) For low Nt, temp values diverges and attains unrealistics values, this happens because of high step value of (dt) which leads unstability & divergence → To be continu

Q 4

	ρ	c	ρ	k	h
			1600	170	5
	300	400			50
sphere A				400	
sphere B	30	1600			

$Bi = \frac{h(D_0/3)}{k}$ $\left(\because L_c = \frac{r_0}{3}\right)$



(4)

$Bi_A = \frac{5\times0.15/3}{170} = 1.47\times10^{-3}$ $Bi \ll 0.1$

$Bi_B = \frac{50\times0.015/3}{17} = 0.147$

$\tau_A = \frac{1600\times0.15\times400}{3\times5} = 6400$ s

Thermal time constant $\tau = \frac{\rho V c}{hA}$ ⇒

$\tau_B = \frac{400\times0.015\times1600}{3\times50} = 64$ s

∴ B ~~slower~~ cools slower than A

as $\tau_B < \tau_A$

(b) As $Bi << 0.1$ for sphere A,

$$\frac{\theta}{\theta_i} = e^{-t/\tau} \Rightarrow t = -\tau_A \times \ln\frac{\theta}{\theta_i}$$

$$t = -6400 \ln\frac{415-320}{800-320}$$

$$= 10367.8 \text{ sec}$$

$$t = 2.88 \text{ hrs}$$

for sphere B, $Bi > 0.1$ $\therefore$ using exact solution

$$(Bi)_B = \frac{h r_o}{k} = \frac{50 \times 0.015}{1.7} = 0.441$$

$$\zeta_1 = 1.0992 \qquad C_1 = 1.1278$$

$$\theta^*(r^*=1, F_o) = \frac{T(r_o,t) - T_\infty}{T_i - T_\infty} = \frac{415-320}{800-320} = 0.1979$$

$$\theta_o^* = \theta^* \zeta_1 r^* / \sin(\zeta_1 r^*) = \frac{0.1979 \times 1.0992 \times 1}{\sin(1.099)}$$

$$= 0.2442$$

$$F_o = -\frac{1}{\zeta_1^2} \ln\frac{\theta_o^*}{C_1} = -\frac{1}{1.0992^2} \ln\left(\frac{0.2442}{1.1278}\right) = 1.266$$

$$t_B = F_o \frac{r_o^2}{\alpha} = F_o \frac{\rho c_p}{k} r_o^2 = 1.266 \times \frac{400 \times 1600}{1.7} \times 0.015^2$$

$$= 107 \text{ sec}$$

as fourier no. is > 0.2 therefore one term approx is accurate

(c) Sphere A (~~sphe~~ space isothermal behaviour)

$$E_{in} - E_{out} = \Delta E$$

$$-Q_A = \Delta E = E(t) - E(0)$$

$$Q_A = \rho c V [T - T_i] = 1600 \times 400 \times 4/3 \times \pi \times 0.15^3 (415-800)$$

$$Q_A = 3.48 \times 10^6 \text{ J}$$

sphere B

$$\frac{Q_B}{Q_o} = 1 - \frac{3\theta_o^*}{\zeta_1^3}\left[\sin\zeta_1 - \zeta_1 \cos\zeta_1\right] = 1 - \frac{3 \times 0.2442}{1.0992^3} \times (\sin$$

$$= 0.784$$

$$Q_B = 0.784 Q_o = 0.784 \times 1600 \times \frac{4}{3}\pi\, 0.015^3 \times (800-320) \times 400$$

$$Q_B = 3405 \text{ J.}$$

Q5

case I : $q_0 = 1$

The plot depicts the evalution of temp distribution over time & space in 1D & T_i is uniformly initialised to 0°c

~~Asin~~ $q_0 = 1$ is small and change is gradual

case II : $q_0 = 1000$

The key diff. in this case is amplitude, the larger T_0 which leads to more rapid changes exhibiliting more prononced varianion a higher peak temp.

Assignment3.ipynb - Assignments - Visual Studio Code

Assignment3.ipynb > Decresing Nx or Nt is making Temperature = 0 throughout. This is caused by high step value which makes the approximation in calculation of derivative invalid. > empty cell

```python
import numpy as np
import matplotlib.pyplot as plt
```

```python
k = 10
alpha = 0.05
L = 1
Ti = 0
Tleft = 100
tau = 20
```

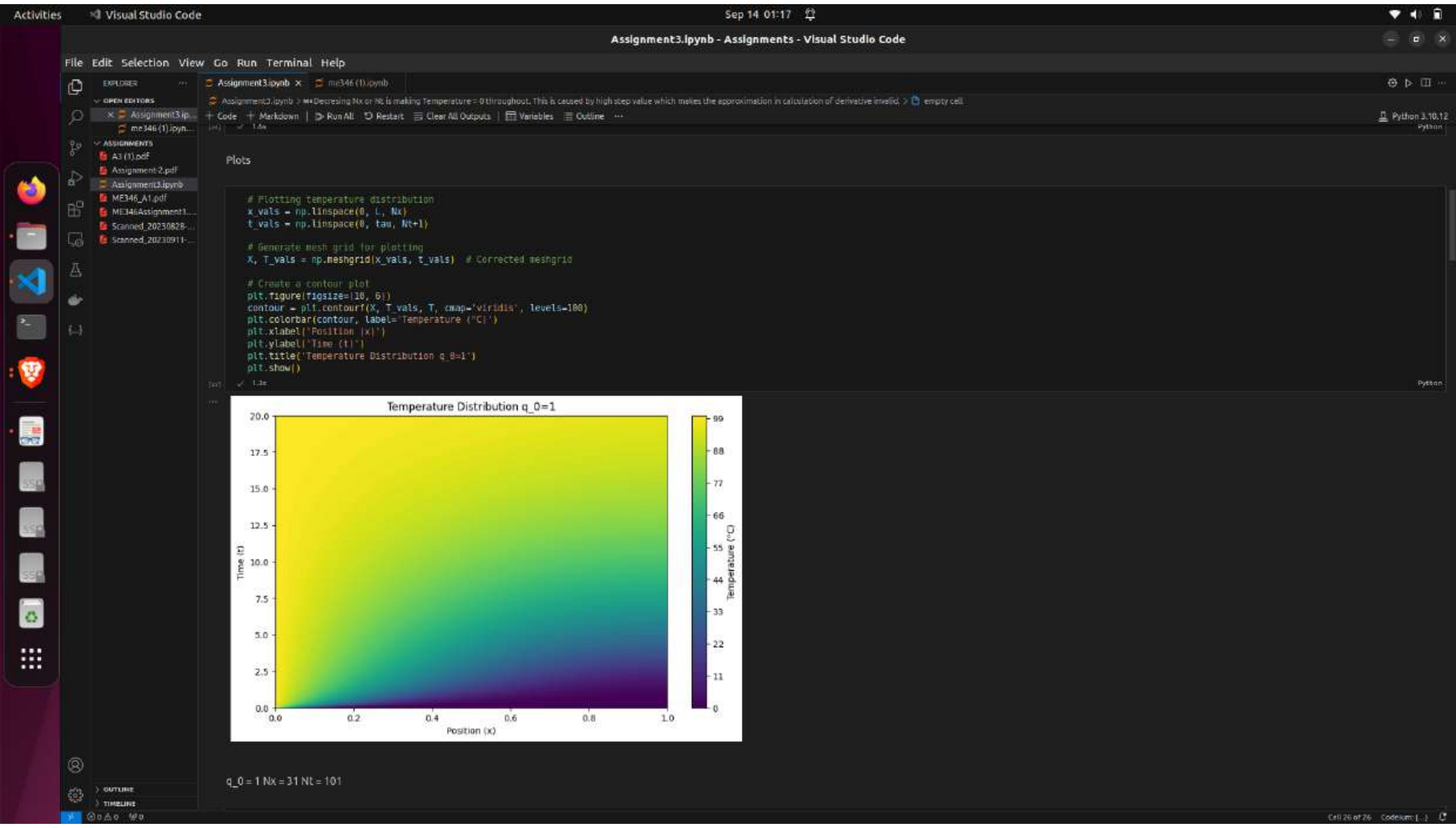
q_0 = 1
Nx = 31
Nt = 10001

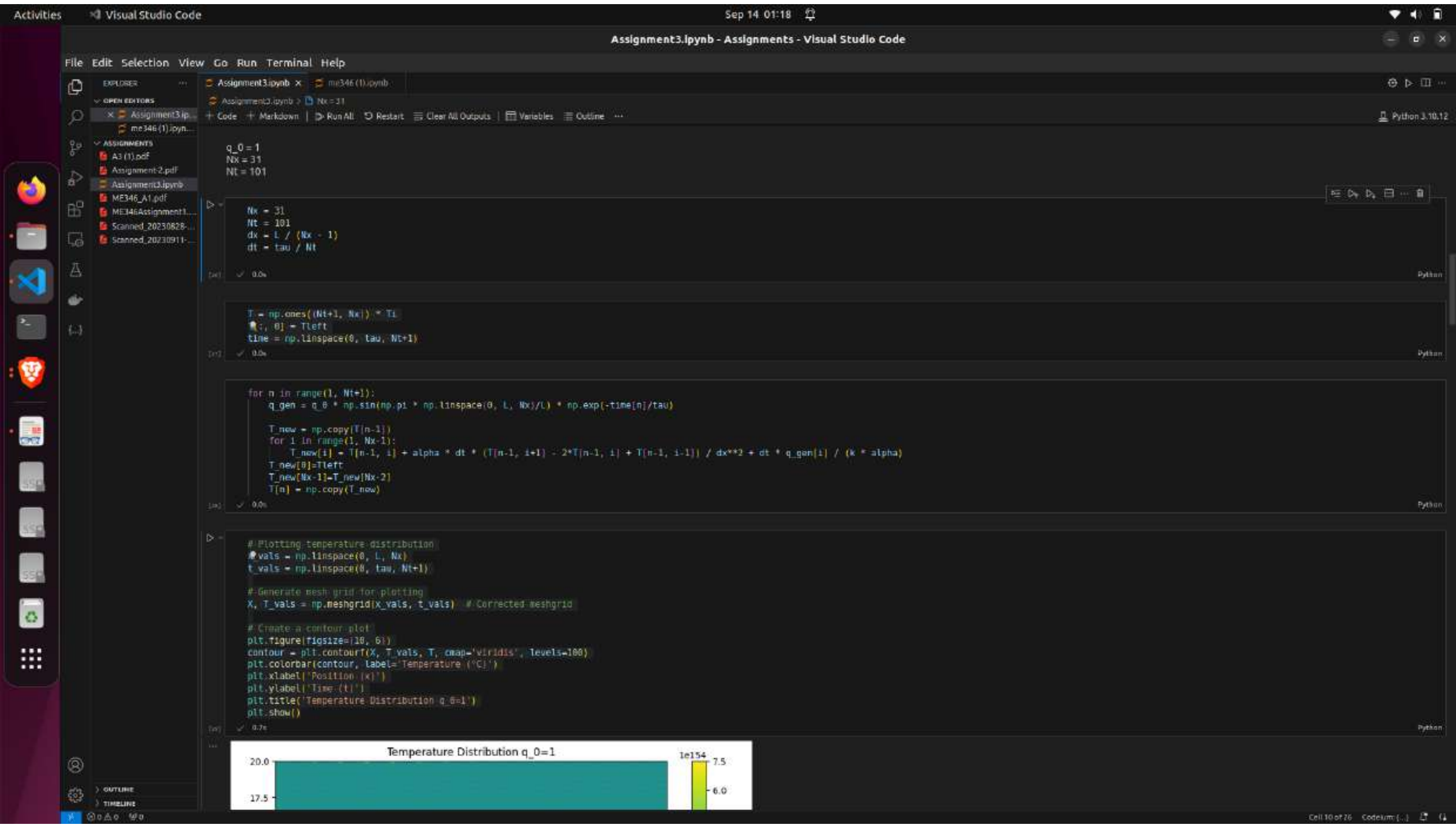
```python
Nx = 31
Nt = 10001
dx = L / (Nx - 1)
dt = tau / Nt

q_0 = 1
```

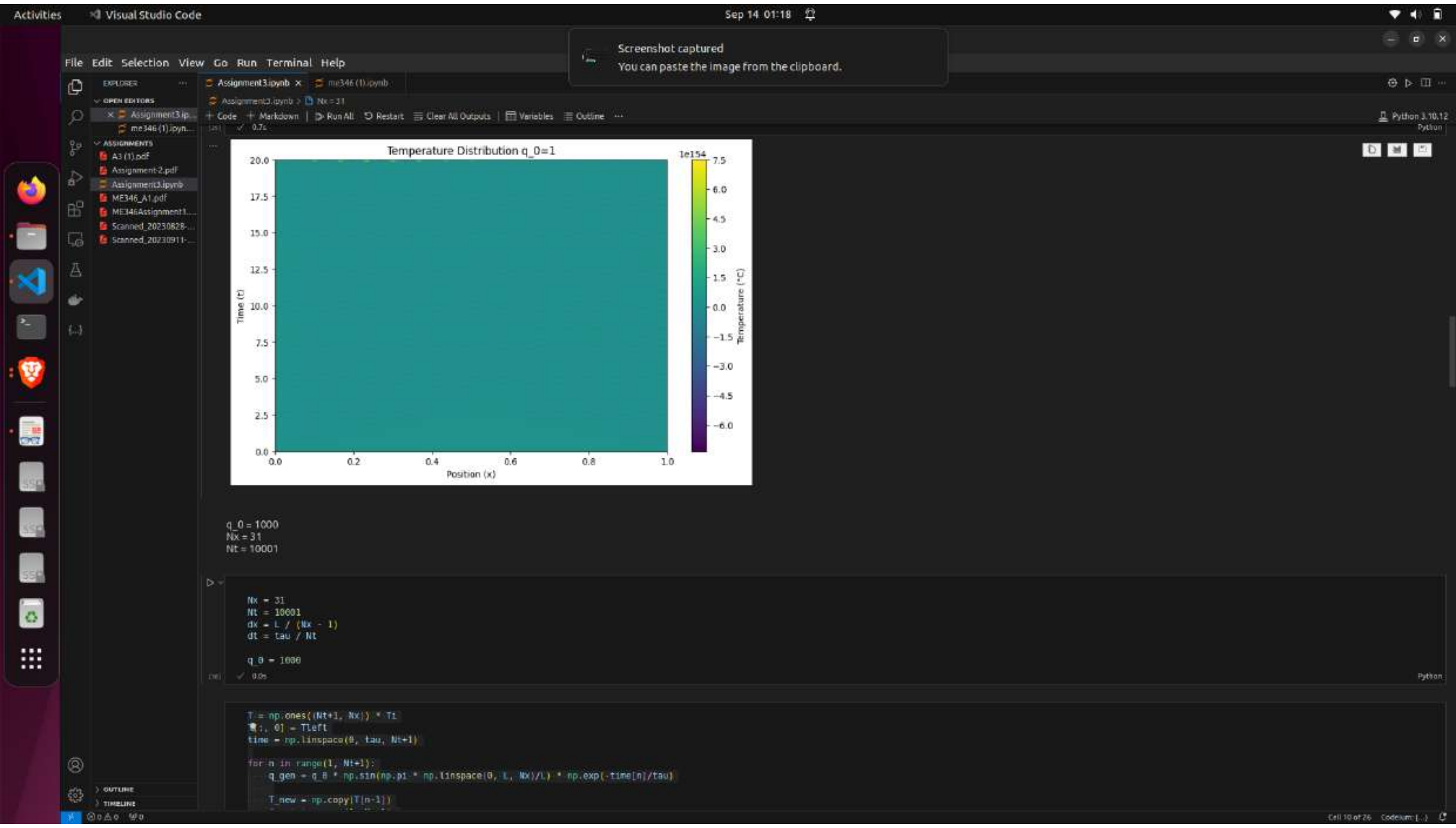
```python
# Initializing temperature array
T = np.ones((Nt+1, Nx)) * Ti

# Boundary conditions
T[:, 0] = Tleft

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

```python
for n in range(1, Nt+1):
    q_gen = q_0 * np.sin(np.pi * np.linspace(0, L, Nx)/L) * np.exp(-time[n]/tau)

    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 * q_gen[i] / (k * alpha)
    T_new[0]=Tleft
    T_new[Nx-1]=T_new[Nx-2]
    T[n] = np.copy(T_new)
```

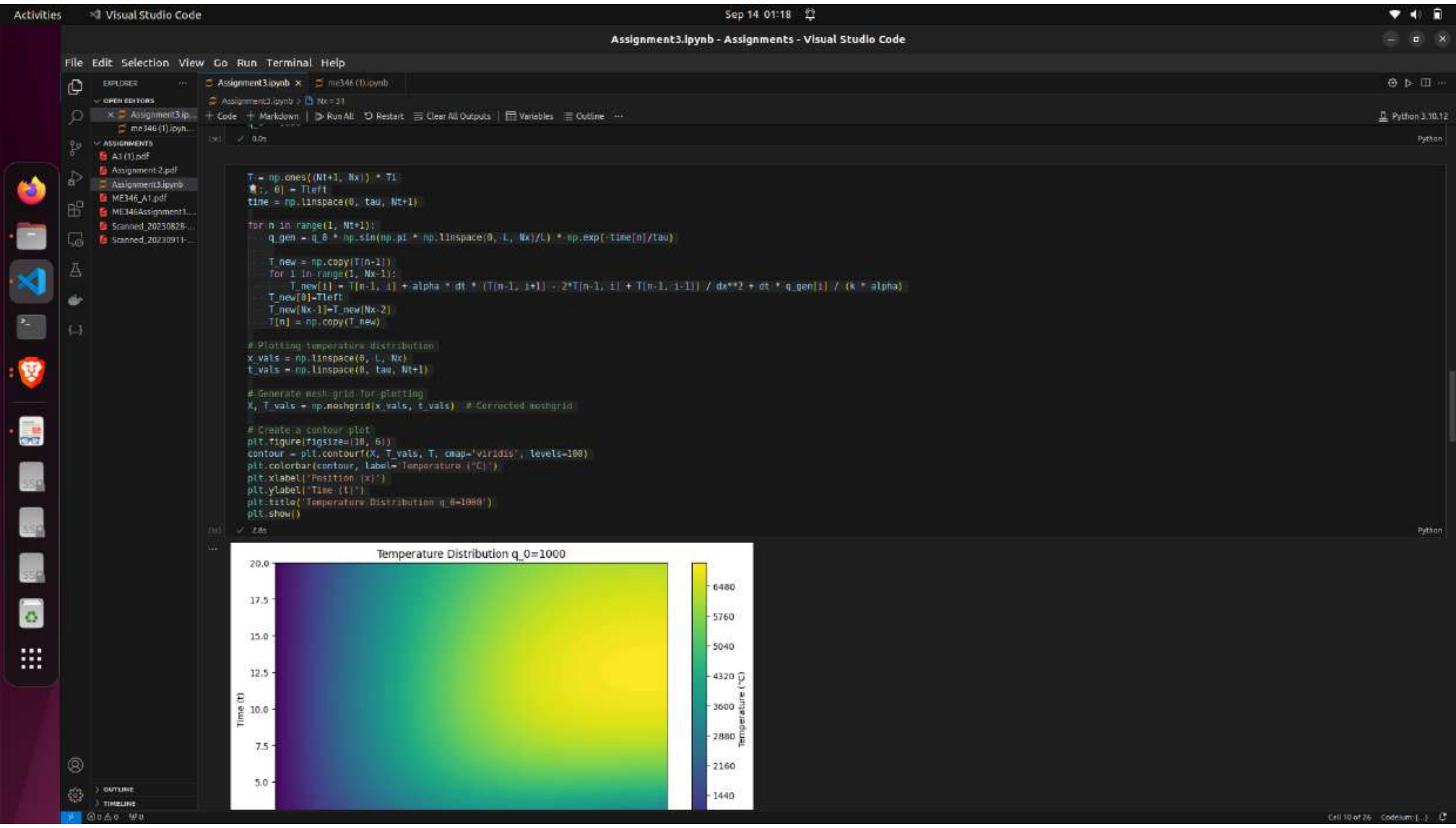
Plots


Activities
Visual Studio Code
Sep 14 01:17
Assignment3.ipynb - Assignments - Visual Studio Code
File Edit Selection View Go Run Terminal Help
Assignment3.ipynb
me346 (1).ipynb
Assignment3.ipynb > Decresing Nx or Nt is making Temperature = 0 throughout. This is caused by high step value which makes the approximation in calculation of derivative invalid. > empty cell
Code Markdown Run All Restart Clear All Outputs Variables Outline
Python 3.10.12
Plots
# 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) # Corrected meshgrid
# Create a contour plot
plt.figure(figsize=(10, 6))
contour = plt.contourf(X, T_vals, 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()
Temperature Distribution q_0=1
Time (t)
Position (x)
Temperature (°C)
q_0 = 1 Nx = 31 Nt = 101

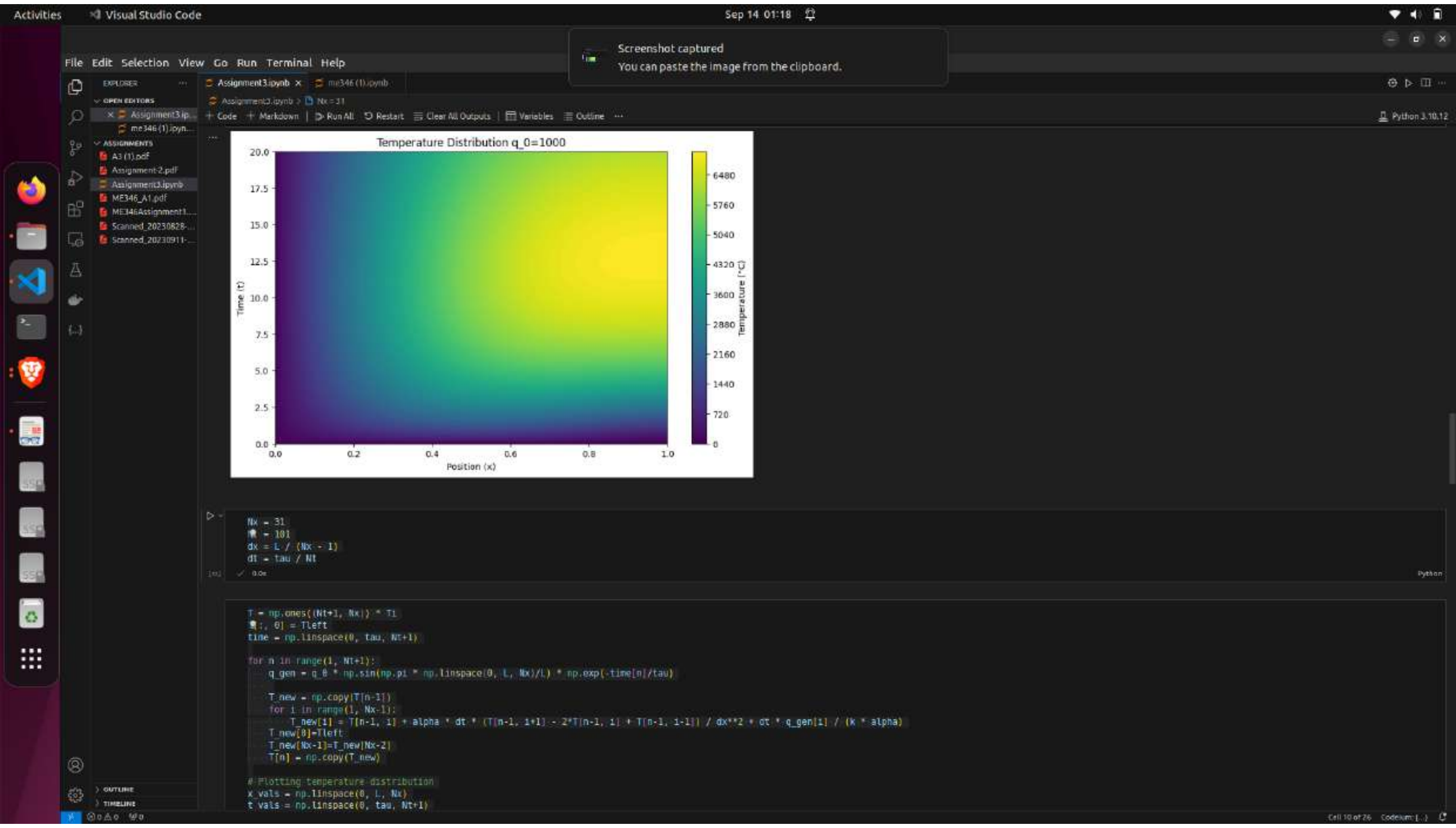

Activities
Visual Studio Code
Sep 14 01:18
Assignment3.ipynb - Assignments - Visual Studio Code
File Edit Selection View Go Run Terminal Help
Assignment3.ipynb
me346 (1).ipynb
q_0 = 1
Nx = 31
Nt = 101
Nx = 31
Nt = 101
dx = L / (Nx - 1)
dt = tau / Nt
T = np.ones((Nt+1, Nx)) * Ti
T[:, 0] = Tleft
time = np.linspace(0, tau, Nt+1)
for n in range(1, Nt+1):
q_gen = q_0 * np.sin(np.pi * np.linspace(0, L, Nx)/L) * np.exp(-time[n]/tau)
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 * q_gen[i] / (k * alpha)
T_new[0]=Tleft
T_new[Nx-1]=T_new[Nx-2]
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) # Corrected meshgrid
# Create a contour plot
plt.figure(figsize=(10, 6))
contour = plt.contourf(X, T_vals, 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()
Temperature Distribution q_0=1
20.0
17.5
1e154
7.5
6.0



Activities
Visual Studio Code
Sep 14 01:18
Screenshot captured
You can paste the image from the clipboard.
File Edit Selection View Go Run Terminal Help
Assignment3.ipynb
Temperature Distribution q_0=1
1e154
Temperature (°C)
Time (t)
Position (x)
q_0 = 1000
Nx = 31
Nt = 10001
Nx = 31
Nt = 10001
dx = L / (Nx - 1)
dt = tau / Nt
q_0 = 1000
T = np.ones((Nt+1, Nx)) * Ti
T[:, 0] = Tleft
time = np.linspace(0, tau, Nt+1)
for n in range(1, Nt+1):
q_gen = q_0 * np.sin(np.pi * np.linspace(0, L, Nx)/L) * np.exp(-time[n]/tau)
T_new = np.copy(T[n-1])



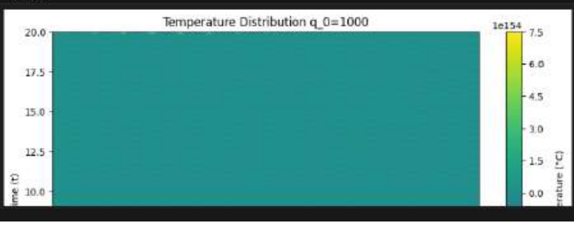
Assignment3.ipynb - Assignments - Visual Studio Code
T = np.ones((Nt+1, Nx)) * Ti
T[:, 0] = Tleft
time = np.linspace(0, tau, Nt+1)
for n in range(1, Nt+1):
q_gen = q_0 * np.sin(np.pi * np.linspace(0, L, Nx)/L) * np.exp(-time[n]/tau)
T_new = np.copy(T[n-1])
for i in range(1, Nx-1):
T_new[i] = T[n-1, i] + alpha * dt * (T[n-1, i+1] - 2*T[n-1, i] + T[n-1, i-1]) / dx**2 + dt * q_gen[i] / (k * alpha)
T_new[0]=Tleft
T_new[Nx-1]=T_new[Nx-2]
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) # Corrected meshgrid
# Create a contour plot
plt.figure(figsize=(10, 6))
contour = plt.contourf(X, T_vals, 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()
Temperature Distribution q_0=1000
Time (t)
Temperature (°C)

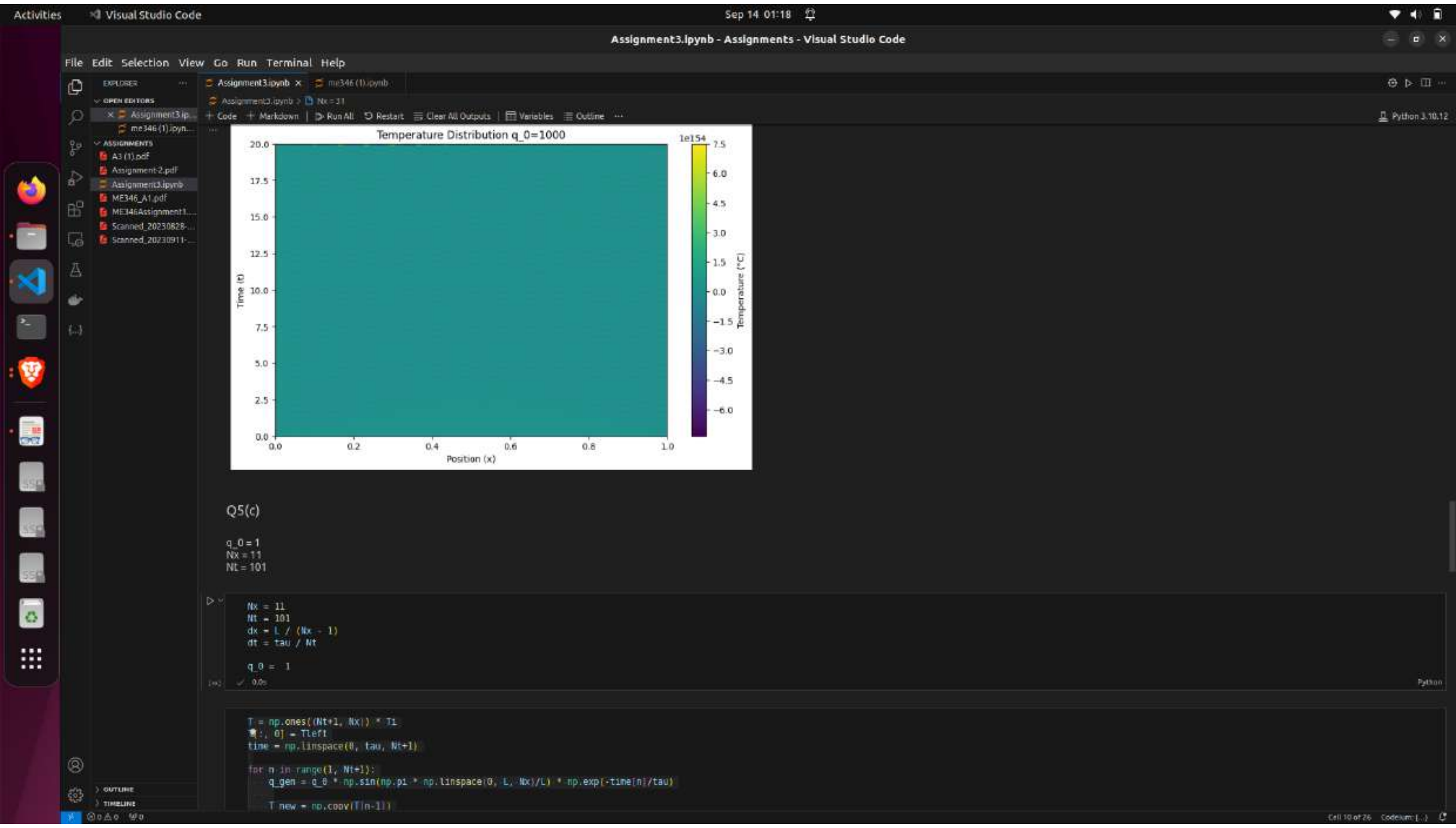

Activities
Visual Studio Code
Sep 14 01:18
Screenshot captured
You can paste the image from the clipboard.
File Edit Selection View Go Run Terminal Help
Assignment3.ipynb
+ Code + Markdown | Run All Restart Clear All Outputs Variables Outline
Python 3.10.12
Temperature Distribution q_0=1000
Time (t)
Position (x)
Temperature (°C)
6480
5760
5040
4320
3600
2880
2160
1440
720
0
Nx = 31
Nt = 101
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):
q_gen = q_0 * np.sin(np.pi * np.linspace(0, L, Nx)/L) * np.exp(-time[n]/tau)
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 * q_gen[i] / (k * alpha)
T_new[0]=Tleft
T_new[Nx-1]=T_new[Nx-2]
T[n] = np.copy(T_new)
# Plotting temperature distribution
x_vals = np.linspace(0, L, Nx)
t_vals = np.linspace(0, tau, Nt+1)

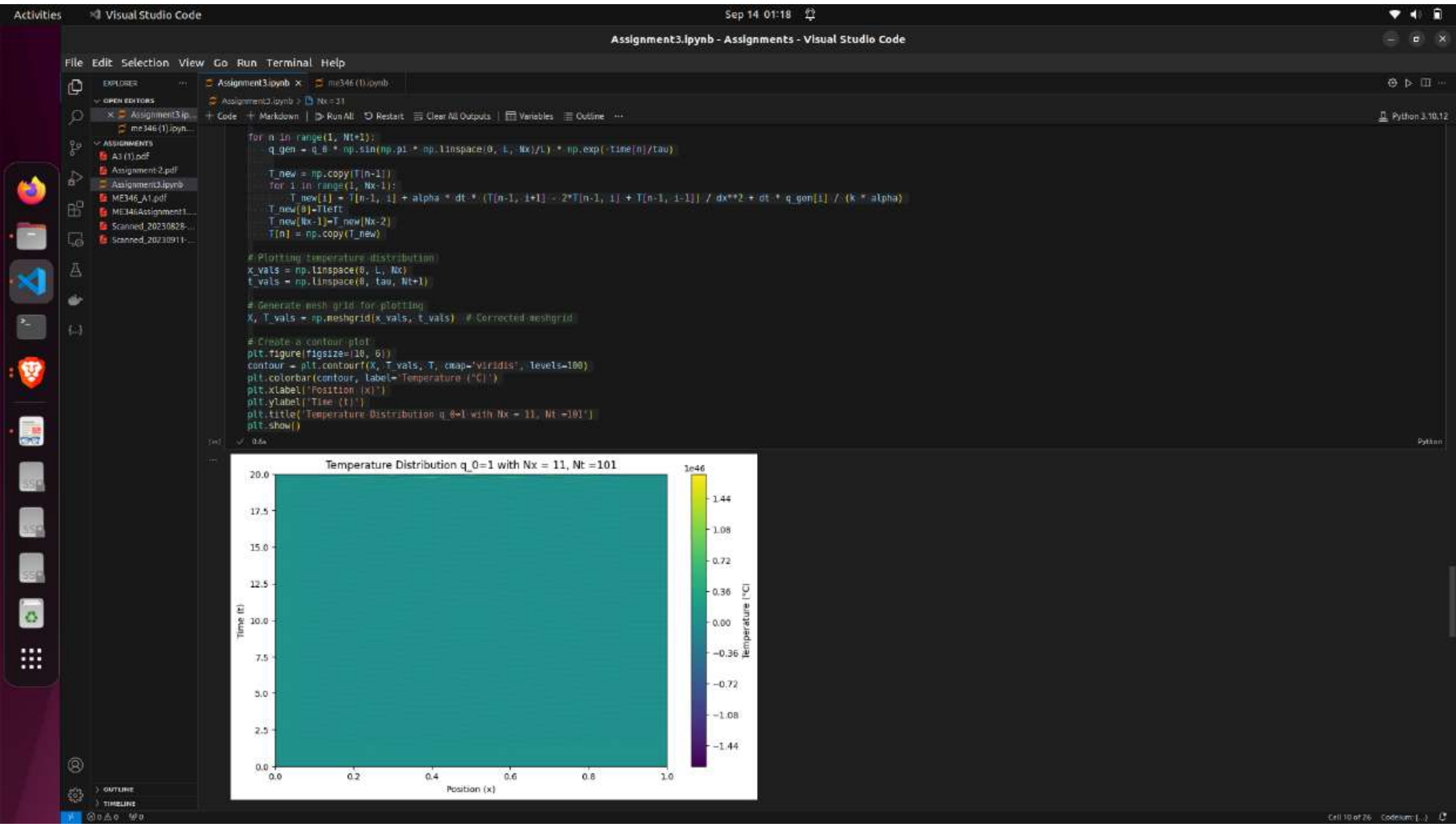
Assignment3.ipynb - Assignments - Visual Studio Code

```python
Nx = 31
Nt = 101
dx = L / (Nx - 1)
dt = tau / Nt
```

```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):
    q_gen = q_0 * np.sin(np.pi * np.linspace(0, L, Nx)/L) * np.exp(-time[n]/tau)

    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 * q_gen[i] / (k * alpha)
    T_new[0]=Tleft
    T_new[Nx-1]=T_new[Nx-2]
    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)  # Corrected meshgrid

# Create a contour plot
plt.figure(figsize=(10, 6))
contour = plt.contourf(X, T_vals, 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 14 01:18
Assignment3.ipynb - Assignments - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
Assignment3.ipynb
me346 (1).ipynb
ASSIGNMENTS
A3 (1).pdf
Assignment-2.pdf
Assignment3.ipynb
ME346_A1.pdf
Scanned_20230828-...
Scanned_20230911-...
Code
Markdown
Run All
Restart
Clear All Outputs
Variables
Outline
Python 3.10.12
Temperature Distribution q_0=1000
Time (t)
Position (x)
Temperature (°C)
1e154
Q5(c)
q_0 = 1
Nx = 11
Nt = 101
Nx = 11
Nt = 101
dx = L / (Nx - 1)
dt = tau / Nt
q_0 = 1
T = np.ones((Nt+1, Nx)) * Ti
T[:, 0] = Tleft
time = np.linspace(0, tau, Nt+1)
for n in range(1, Nt+1):
q_gen = q_0 * np.sin(np.pi * np.linspace(0, L, Nx)/L) * np.exp(-time[n]/tau)
T_new = np.copy(T[n-1])
OUTLINE
TIMELINE

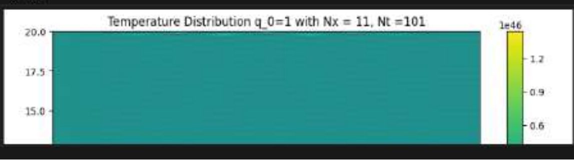


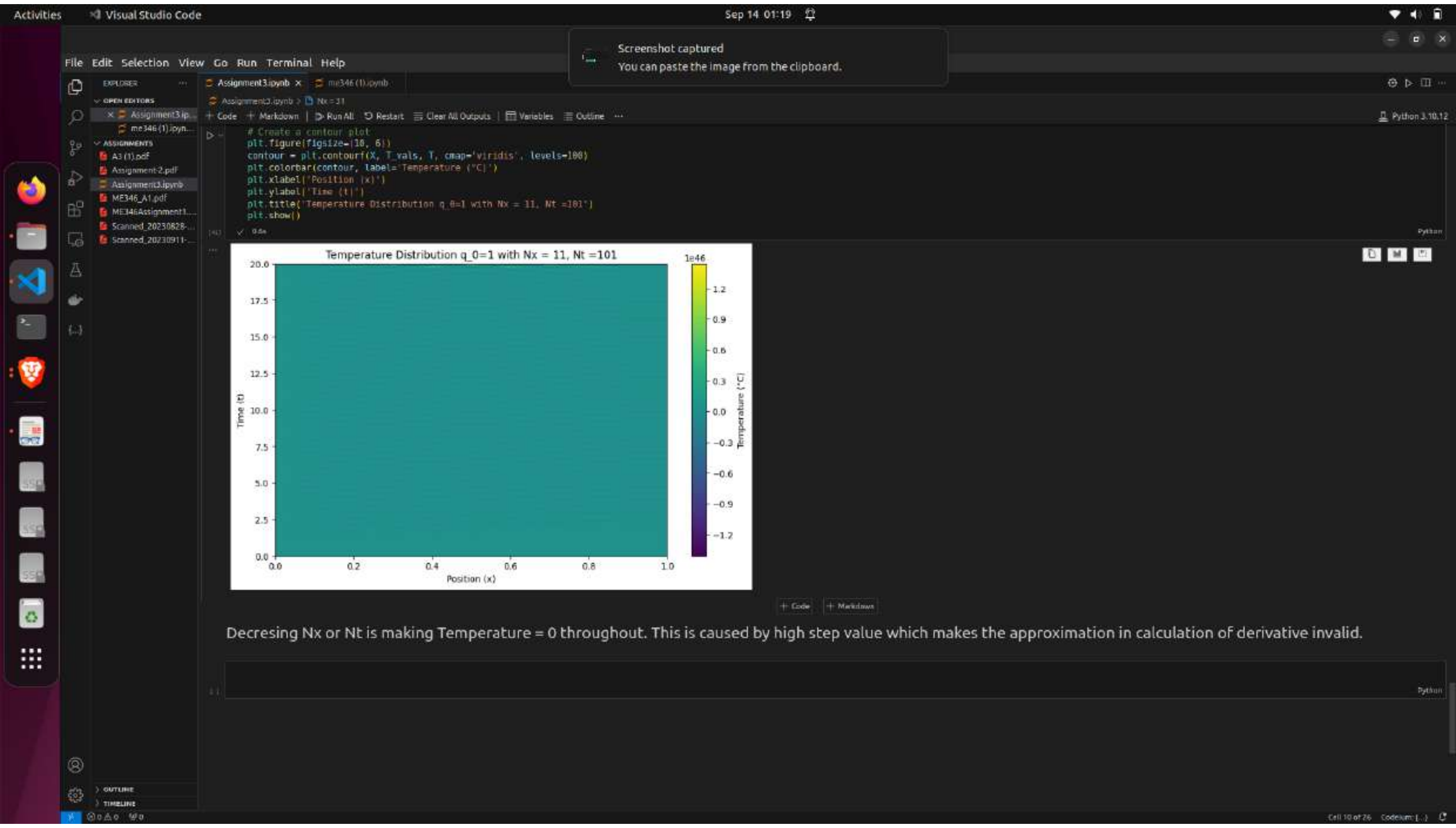
Activities
Visual Studio Code
Sep 14 01:18
Assignment3.ipynb - Assignments - Visual Studio Code
File Edit Selection View Go Run Terminal Help
Assignment3.ipynb
me346 (1).ipynb
Code
Markdown
Run All
Restart
Clear All Outputs
Variables
Outline
Python 3.10.12
for n in range(1, Nt+1):
q_gen = q_0 * np.sin(np.pi * np.linspace(0, L, Nx)/L) * np.exp(-time[n]/tau)
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 * q_gen[i] / (k * alpha)
T_new[0]=Tleft
T_new[Nx-1]=T_new[Nx-2]
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) # Corrected meshgrid
# Create a contour plot
plt.figure(figsize=(10, 6))
contour = plt.contourf(X, T_vals, 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 with Nx = 11, Nt =101')
plt.show()
Temperature Distribution q_0=1 with Nx = 11, Nt =101
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
Time (t)
0.0
0.2
0.4
0.6
0.8
1.0
Position (x)
1e46
1.44
1.08
0.72
0.36
0.00
−0.36
−0.72
−1.08
−1.44
Temperature (°C)

q_0 = 1000
Nx = 11
Nt = 101

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

q_0 =  1000
```

```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):
    q_gen = q_0 * np.sin(np.pi * np.linspace(0, L, Nx)/L) * np.exp(-time[n]/tau)

    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 * q_gen[i] / (k * alpha)
    T_new[0]=Tleft
    T_new[Nx-1]=T_new[Nx-2]
    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)  # Corrected meshgrid

# Create a contour plot
plt.figure(figsize=(10, 6))
contour = plt.contourf(X, T_vals, 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 with Nx = 11, Nt =101')
plt.show()
```




Activities
Visual Studio Code
Sep 14 01:19
Screenshot captured
You can paste the image from the clipboard.
File Edit Selection View Go Run Terminal Help
Assignment3.ipynb
me346 (1).ipynb
Assignment3.ipynb > Nx = 11
Code Markdown Run All Restart Clear All Outputs Variables Outline
Python 3.10.12
# Create a contour plot
plt.figure(figsize=(10, 6))
contour = plt.contourf(X, T_vals, 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 with Nx = 11, Nt =101')
plt.show()
Temperature Distribution q_0=1 with Nx = 11, Nt =101
Time (t)
Position (x)
1e46
Temperature (°C)
Decresing Nx or Nt is making Temperature = 0 throughout. This is caused by high step value which makes the approximation in calculation of derivative invalid.