2.1

2.1 The Diffusion Equation

Analytic Solution

Question 1

First consider the substitution $\theta(x,t) = \theta_0 + (\theta_1 - \theta_0) F(x,t)$, then the problem becomes

$$\frac{\partial \theta}{\partial t} = K \frac{\partial^2 \theta}{\partial^2 x} \implies \frac{\partial F}{\partial t} = K \frac{\partial^2 F}{\partial^2 x} \tag{1}$$

$$\theta(x,0) = \theta_0 \implies F(x,0) = 0 \tag{2}$$

$$\theta(0,t) = \theta_1 \implies F(0,t) = 1 \tag{3}$$

$$\tag{4}$$

with the boundary condition as $x \to \infty$ becoming

$$\frac{\partial \theta}{\partial x}(x,t) \to 0 \text{ as } x \to \infty \implies \frac{\partial F}{\partial x}(x,t) \to 0 \text{ as } x \to \infty \tag{5a}$$

$$\theta(x,t) \to \theta_0 \text{ as } x \to \infty \implies F(x,t) \to 0 \text{ as } x \to \infty \tag{5b}$$

We are going to derive the dimensions of F, x, t, k with respect to the dimensions of T, L, Θ representing time, distance, and temperature respectively. We can only add terms together of like dimensionality so we require that $[\theta(x,t)] = [\theta_0] = [(\theta_1 - \theta_0) F(x,t)] = \Theta$, so $[(\theta_1 - \theta_0)][F(x,t)] = \Theta[F(x,t)] = \Theta \implies [F(x,t)] = 1$. We already know that $[x] = L$ and $[t] = T$ so the dimensions of all quantities in this problem are:

$$[F] = 1$$
$$[x] = L$$
$$[t] = T$$
$$[k] = L^2 T^{-1}$$

from this we can deduce that $\left[\frac{x}{\sqrt{kt}}\right] = 1$ so F and $\left[\frac{x}{\sqrt{kt}}\right]$ are dimensionless so by Buckingham Pi Theorem $F = f(\xi)$ where $\xi = \frac{x}{\sqrt{kt}}$. We will solve the problem by first noting

$$\frac{\partial f}{\partial t} = -\frac{1}{2t} \xi f'(\xi)$$
$$\frac{\partial^2 f}{\partial^2 x} = \frac{\xi^2}{x^2} f''(\xi)$$

so

$$\begin{aligned}
\frac{\partial f}{\partial t} &= K\frac{\partial^2 f}{\partial^2 x} \\
\implies -\frac{1}{2t}\xi f'(\xi) &= K\frac{\xi^2}{x^2} f''(\xi) \\
\implies -\frac{\xi}{2} f'(\xi) &= f''(\xi) \\
\implies \frac{f'(\xi)}{f''(\xi)} &= -\frac{\xi}{2}
\end{aligned}$$

Therefore

$$\begin{aligned}
f'(\xi) &= Ae^{-\frac{\xi^2}{4}} \\
f(\xi) &= B + A\int_{\xi}^{\infty} e^{-\frac{u^2}{4}} du \\
&= B + A\int_{\frac{\xi}{2}}^{\infty} e^{-u^2} du
\end{aligned}$$

Applying the conditions (2), (3) above

$$\begin{aligned}
F(x,0) &= f(\infty) = B = 0 \\
F(0,t) &= f(0) = \frac{\sqrt{\pi}}{2} A = 1
\end{aligned}$$

so

$$f(\xi) = \frac{2}{\sqrt{\pi}} \int_{\frac{\xi}{2}}^{\infty} e^{-u^2} du$$

Note that

$$\begin{aligned}
\frac{\partial F}{\partial x}(x,t) &= \frac{-2}{\sqrt{\pi K t}} e^{-\frac{x^2}{4Kt}} \to 0 \text{ as } x \to \infty \\
F(x,t) &= \frac{2}{\sqrt{\pi}} \int_{\frac{x}{2\sqrt{kt}}}^{\infty} e^{-u^2} du \to 0 \text{ as } x \to \infty
\end{aligned}$$

so $f(\xi)$ satisfies the boundary conditions (5a) and (5b) defined above.

Question 2

let $U(X,T) = U_t(X,T) + U_s(X)$ where both U_t and U_s satisfy

$$U_T = U_{XX} \text{ for } 0 < T \ , \ 0 < X < 1$$

and U_t satisfies the boundary conditions

$$U_t(0,T) = U_t(1,T) = 0$$

and U_s satisfies the boundary conditions

$$U_s(0) = 1 \ , \ U_s(1) = 0$$

(we can view the term U_s as the steady state solution and U_t the transient state solution). we can see that

$$U_s(x) = 1 - X$$

Let $U_t = \mathcal{X}(X)\mathcal{T}(T)$ Then we have $\mathcal{X}\dot{\mathcal{T}} = \mathcal{X}''\mathcal{T}$ so $\frac{\mathcal{X}''}{\mathcal{X}} = \frac{\dot{\mathcal{T}}}{\mathcal{X}} = -k^2$ and $\mathcal{X}(0) = \mathcal{X}(1) = 0$ therefore

$$\begin{aligned}
\mathcal{X}_n(X) &= A\sin(n\pi X) \ , \quad k = n\pi \\
\mathcal{T}_n(T) &= Be^{-n^2\pi^2 T}
\end{aligned}$$

so the solution $\mathcal{X}_n(X)\mathcal{T}_n(T)$ holds true for all n and any superposition of these solutions satisfies the boundary conditions so we have

$$U_t(X,T)=\sum_{n=1}^{\infty}\mathcal{X}_n(X)\mathcal{T}_n(T)=\sum_{n=1}^{\infty}A_n\sin(n\pi X)e^{-n^2\pi^2T}$$

so

$$U(X,T)=U_s(X)+U_t(X,T)=1-X+\sum_{n=1}^{\infty}A_n\sin(n\pi X)e^{-n^2\pi^2T}$$

we require

$$U(X,0)=1-X+\sum_{n=1}^{\infty}A_n\sin(n\pi X)=0$$

thus

$$\sum_{n=1}^{\infty}A_n\sin(n\pi X)=X-1$$

This is a half sine-series of $f(X)=X-1$ for $0<X<1$ so

$$A_n=2\int_0^1(X-1)\sin(n\pi X)\,dX=-\frac{2}{\pi n}$$

Therefore

$$U(X,T)=1-X-\sum_{n=1}^{\infty}\frac{2}{\pi n}\sin(n\pi X)e^{-n^2\pi^2T}$$

Similarly we consider $U(X,T)=U_t(X,T)+U_s(X)$ where both U_t and U_s satisfy

$$U_T=U_{XX}\text{ for }0<T\ ,\ 0<X<1$$

and U_t satisfies the boundary conditions

$$U_t(0,T)=\frac{\partial U_t}{\partial X}(1,T)=0$$

and U_s satisfies the boundary conditions

$$U_s(0)=1\ ,\ \frac{\partial U_s}{\partial X}(1)=0$$

We can see that

$$U_s(X)=1$$

Let $U_t=\mathcal{X}(X)\mathcal{T}(T)$ Then we have $\mathcal{X}\dot{\mathcal{T}}=\mathcal{X}''\mathcal{T}$ so $\frac{\mathcal{X}''}{\mathcal{X}}=\frac{\dot{\mathcal{T}}}{\mathcal{X}}=-k^2$ and $\mathcal{X}(0)=\mathcal{X}(1)=0$ therefore

$$\mathcal{X}_n(X)=A\sin\left(\left(n+\frac{1}{2}\right)\pi X\right)\quad,\quad k=\left(n+\frac{1}{2}\right)\pi$$

$$\mathcal{T}_n(T)=Be^{-\left(n+\frac{1}{2}\right)^2\pi^2T}$$

so

$$U(X,T)=U_s(X)+U_t(X,T)=1+\sum_{n=0}^{\infty}B_n\sin\left(\left(n+\frac{1}{2}\right)\pi X\right)e^{-\left(n+\frac{1}{2}\right)^2\pi^2T}$$

we require

$$U(X,0)=1+\sum_{n=0}^{\infty}B_n\sin\left(\left(n+\frac{1}{2}\right)\pi X\right)=0$$

This looks like the odd terms of the half sine series of $f(X) = -1$ for $0 < X < 2$ so if we show that the half sine series of $f(X) = -1$ for $0 < X < 2$ has no terms for n even then we are done, so consider

$$-1 = \sum_{n=1}^{\infty} c_n \sin\left(\frac{n\pi x}{2}\right)$$

then

$$c_n = \int_0^2 -\sin\left(\frac{n\pi x}{2}\right) = \begin{cases} 0 & \text{if } n \text{ even} \\ -\frac{4}{\pi n} & \text{if } n \text{ odd} \end{cases}$$

so

$$0 = 1 + \sum_{n=0}^{\infty} -\frac{4}{\pi(2n+1)} \sin\left(\frac{(2n+1)\pi x}{2}\right) = 1 + \sum_{n=0}^{\infty} -\frac{4}{\pi(2n+1)} \sin\left(\left(n+\frac{1}{2}\right)\pi x\right)$$

for $0 < x < 2$ so $B_n = -\frac{4}{\pi(2n+1)}$ so

$$U(X,T) = 1 - \sum_{n=0}^{\infty} \frac{4}{\pi(2n+1)} \sin\left(\left(n+\frac{1}{2}\right)\pi X\right) e^{-\left(n+\frac{1}{2}\right)^2 \pi^2 T}$$

Programming Task 1

I used the functions defined in the *Analytic Solutions* section of the Program Listings at the end of this project, to produce the graphs and tables below

x	fixed-endpoint-temperature problem	insulated-end problem	semi-infinite bar problem
0.000	1.000000	1.000000	1.000000
0.125	0.854328	0.865040	0.859684
0.250	0.711808	0.735539	0.723674
0.375	0.575109	0.616656	0.595883
0.500	0.446011	0.512987	0.479500
0.625	0.325133	0.428382	0.376759
0.750	0.211841	0.365839	0.288844
0.875	0.104351	0.327479	0.215925
1.000	-0.000000	0.314554	0.157299

Figure 1: Table tabulating information about fixed-endpoint-temperature, insulated-end problem, and semi-infinite bar problem at time $T = 0.25$

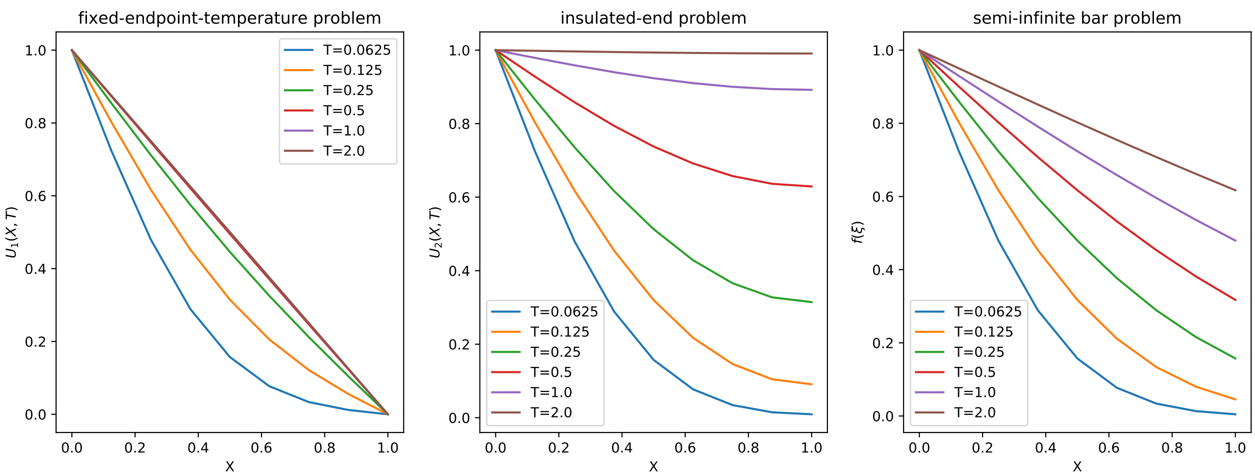

Figure 2: Graph of the non-dimensionalised temperature profiles for $X = 0.125n,\ n = 0, 1, \ldots, 8$ and $T = 0.0625,\ 0.125,\ 0.25,\ 0.5,\ 1.0,\ 2.0$

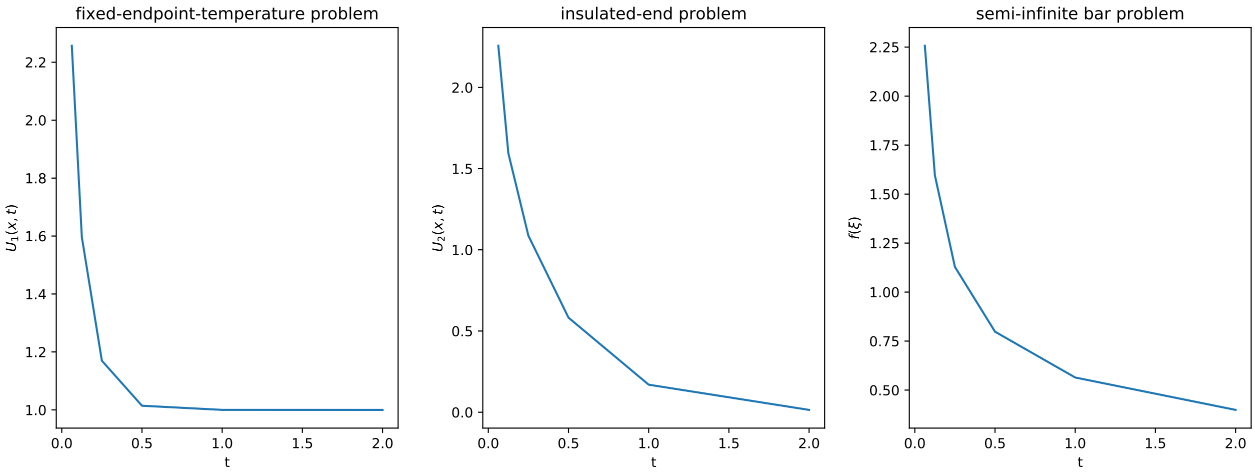

Figure 3: Graph of the non-dimensionalised heat-flux $-U_X$ at $X = 0$ and $T = 0.0625$, 0.125, 0.25, 0.5, 1.0, 2.0

The analytic solutions tabulated and plotted above were found by summing the first 150 terms of the series, we are going to show that this allows for an error less than 2^{-64}, let

$$U_1\left(X,T\right) = 1 - X - \sum_{n=1}^{\infty} \frac{2}{\pi n} \sin\left(n\pi X\right) e^{-n^2\pi^2 T}$$

$$U_1\left(X,T;m\right) = 1 - X - \sum_{n=1}^{m} \frac{2}{\pi n} \sin\left(n\pi X\right) e^{-n^2\pi^2 T}$$

$$U_2\left(X,T\right) = 1 - \sum_{n=0}^{\infty} \frac{4}{\pi\left(2n+1\right)} \sin\left(\left(n+\frac{1}{2}\right)\pi X\right) e^{-\left(n+\frac{1}{2}\right)^2\pi^2 T}$$

$$U_2\left(X,T;m\right) = 1 - \sum_{n=0}^{m} \frac{4}{\pi\left(2n+1\right)} \sin\left(\left(n+\frac{1}{2}\right)\pi X\right) e^{-\left(n+\frac{1}{2}\right)^2\pi^2 T}$$

Consider the error in approximating $U_1\left(X,T\right)$ by the finite sum $U_1\left(X,T;m\right)$

$$\begin{aligned}
\left|U_1\left(X,T\right) - U_1\left(X,T;m\right)\right| &= \left|\sum_{n=m+1}^{\infty} \frac{2}{\pi n} \sin\left(n\pi X\right) e^{-n^2\pi^2 T}\right| \\
&\leq \sum_{n=m+1}^{\infty} \left|\frac{2}{\pi n} \sin\left(n\pi X\right) e^{-n^2\pi^2 T}\right| \\
&\leq \frac{2}{\pi} \sum_{n=m+1}^{\infty} e^{-\frac{\pi^2}{16} n} \\
&= \frac{2e^{-\frac{m+1}{16}\pi^2}}{\pi\left(1 - e^{-\frac{\pi^2}{16}}\right)} \\
&= \frac{2e^{-\frac{m}{16}\pi^2}}{\pi\left(e^{\frac{\pi^2}{16}} - 1\right)} \\
&= \frac{2e^{-\frac{m}{16}\pi^2}}{\pi^3/16} \\
&\leq 2^{1-\frac{m}{2}}
\end{aligned}$$

Noting that this is only valid for $T \geq \frac{1}{16}$. We use the fact that the Taylor expansion of $e^{\frac{\pi^2}{16}} - 1$ contains only positive terms, $\frac{\pi^2}{16} > \frac{1}{2}$, and $\pi^3 > 16$. Consider the error in approximating $U_2\left(X,T\right)$ by the finite sum $U_2\left(X,T;m\right)$

$$
\begin{aligned}
|U_2(x,t) - U_2(x,t;m)| &= \left| \sum_{n=m+1}^{\infty} \frac{4}{\pi(2n+1)} \sin\left(\left(n+\frac{1}{2}\right)\pi X\right) e^{-\left(n+\frac{1}{2}\right)^2 \pi^2 T} \right| \\
&\leq \sum_{n=m+1}^{\infty} \left| \frac{4}{\pi(2n+1)} \sin\left(\left(n+\frac{1}{2}\right)\pi X\right) e^{-\left(n+\frac{1}{2}\right)^2 \pi^2 T} \right| \\
&\leq \frac{4}{\pi} \sum_{n=m+1}^{\infty} e^{-\frac{\pi^2}{16}n} \\
&\leq 2^{2-\frac{m}{2}}
\end{aligned}
$$

So for $m = 150$ we have that the error in approximating $U_1(X,T)$ and $U_2(X,T)$ by the finite sums $U_1(x,t;m)$ and $U_2(x,t;m)$ respectively is less than 2^{-64}

Comparing the temperature profiles of all three problems we see that for T small they all behave similarly, this makes sense as they all start with a bar at temperature $\theta_0 = 0$ and have the point at $X = 0$ instantaneously heated to $\theta_1 = 1$ so for small times the heat would not have propagated far thus the boundary condition at $X = 1$ (or as $X \to \infty$ for the semi-infinite bar) has little impact on the solution. We can also see that all of the problems tend towards steady state solutions which are:

- Fixed-endpoint-temperature problem

  $U_s(X) = 1 - X \ , \quad 0 < X < 1$

- Insulated-end problem

  $U_s(X) = 1 \ , \quad 0 < X < 1$

- Semi-infinite bar problem

  $U_s(X) = 1 \ , \quad 0 < X$

As T becomes larger we can see that for the finite length bar problems, they reach their steady state solution faster than the semi-infinite bar, this makes sense if we consider the steady state solutions for each of the problems and note that the energy needed to get from a bar of length L with constant density starting at a temperature of 0 to a bar with temperature profile $\theta(x)$ is

$$
A \int_0^L \rho\sigma\theta(x)dx
$$

where ρ is the density, σ is the specific heat capacity of the material, and A is the cross-sectional area. so we can see that to reach the stable state solution the fixed-endpoint-temperature problem would require $\frac{\sigma\rho}{2}$ joules of energy, the insulated-end problem requires $\sigma\rho$ joules of energy and the semi-infinite bar would require infinite energy, so the fixed-endpoint-temperature problem would be the first to reach its stable state solution, followed by the insulated-end problem, and lastly the semi-infinite bar problem as the more energy needed to reach the steady state solution, the longer it will take. The energy is provided by the keeping the point at $X = 0$ at temperature $\theta_1 = 1$.
We can also see that as time T gets larger that the heat-flux for the insulated end problem and semi-infinite bar tend to 0, this is because as they reach their steady state, the temperature profile becomes constant, so the energy into each point must equal the energy out, thus a net heat-flux of 0. Whereas for the fixed-endpoint-temperature problem, the temperature profile of the steady state is not constant so a certain amount of energy is loss through the end of the bar ($X = 1$) which must be provided at the start of the bar ($X = 0$), but the energy loss in this case is constant so the heat-flux will tend towards a constant value.

Numerical Integration

Question 3

To incorporate the derivative boundary condition note that

$$\frac{\partial U_2(X,T)}{\partial X} = \frac{U_2(X+\delta X,T) - U_2(X,T)}{\delta X} + O(\delta X) = \frac{U_2(X,T) - U_2(X-\delta X,T)}{\delta X} + O(\delta X)$$

so

$$\frac{\partial U_2(X,T)}{\partial X} = \frac{U_2(X+\delta X,T) - U_2(X-\delta X,T)}{2\delta X} + O(\delta X)$$

Thus

$$\left.\frac{\partial U_2(X,T)}{\partial X}\right|_{X=1} = 0 \implies U_2(1+\delta X,T) - U_2(1-\delta X,T) + O\left((\delta X)^2\right) = 0$$

Which indicates that the boundary condition can be approximated by $U_2(1+\delta X,T) = U_2(1-\delta X,T)$, that is

$$U^m_{N+1} = U^m_{N-1}\ , \quad m \geq 0$$

We choose $U^0_0 = 0.5$ as this is the middle value of the boundary condition at $X = 0$ ($U_2(0,T) = 1\ ,\quad T > 0$) and the initial condition at $T = 0$ ($U_2(X,0) = 0\ ,\quad 0 < X < 1$)

Programming Task 2

I used the functions defined in the *Numerical Integration* section of the Program Listings at the end of this project, to produce the graphs and tables below. Figures 4 to 8 show the output of the program for $C = \frac{1}{12},\ \frac{1}{6},\ \frac{1}{3},\ \frac{1}{2},\ \frac{2}{3}$, and $N = 8,\ 16,\ 32$

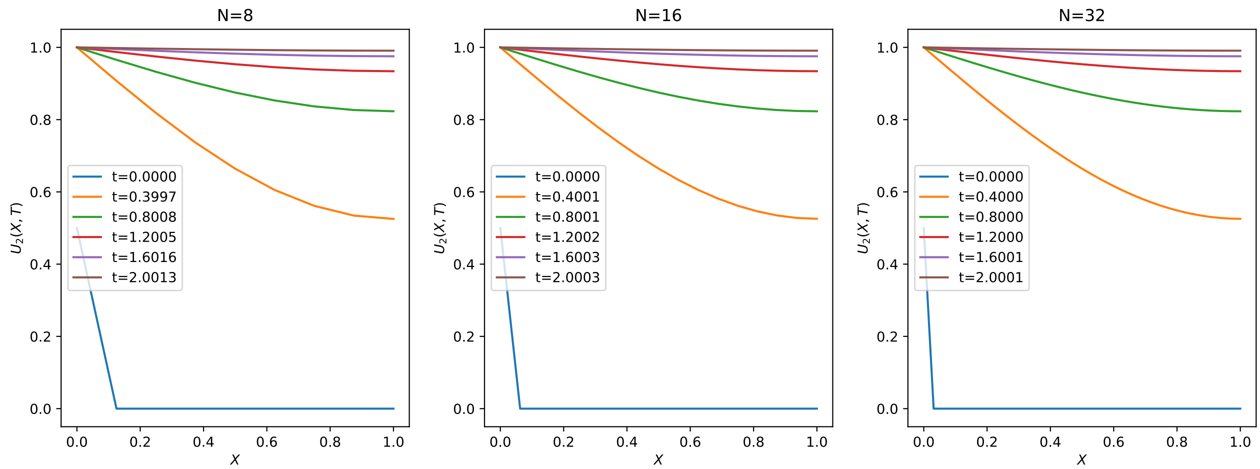

Figure 4: Graph of the numerical solution to the insulated end problem with C=$\frac{1}{12}$

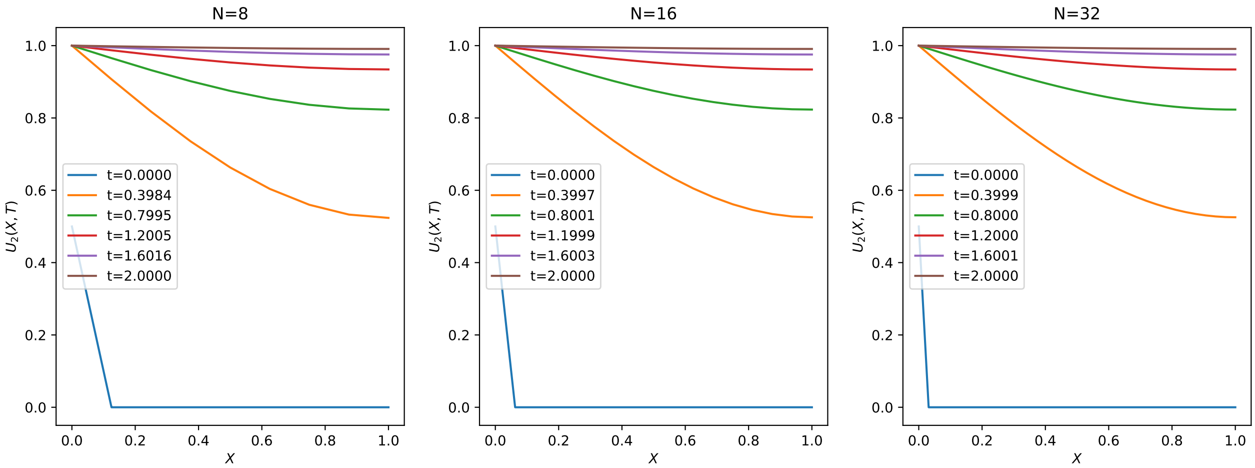

Figure 5: Graph of the numerical solution to the insulated end problem with C=$\frac{1}{6}$

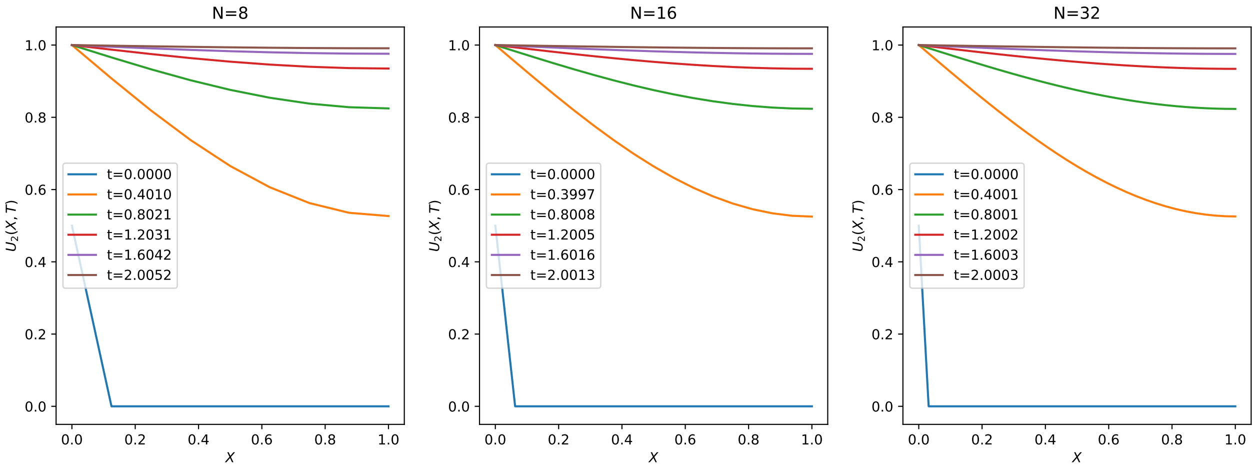

Figure 6: Graph of the numerical solution to the insulated end problem with C=$\frac{1}{3}$

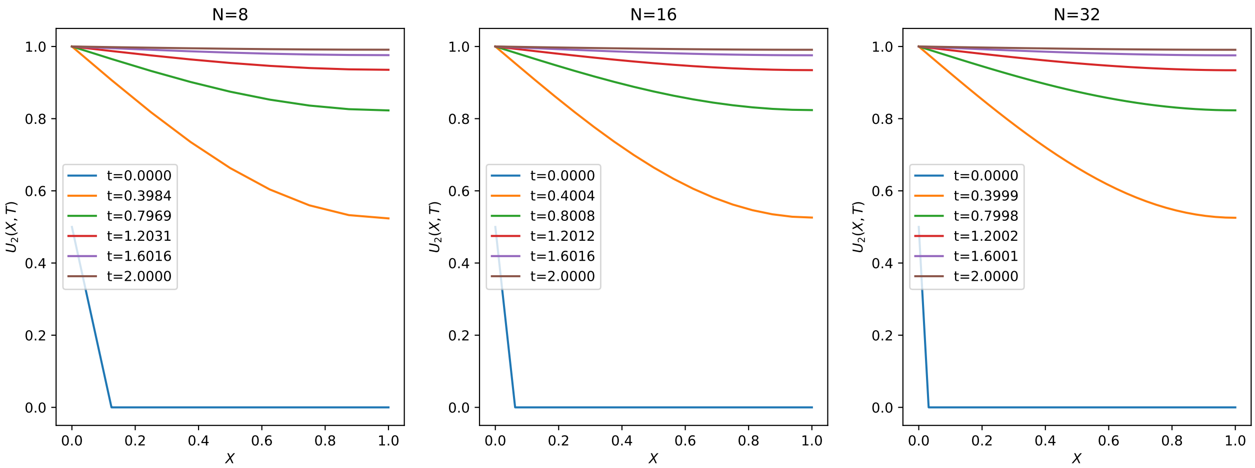

Figure 7: Graph of the numerical solution to the insulated end problem with C=$\frac{1}{2}$

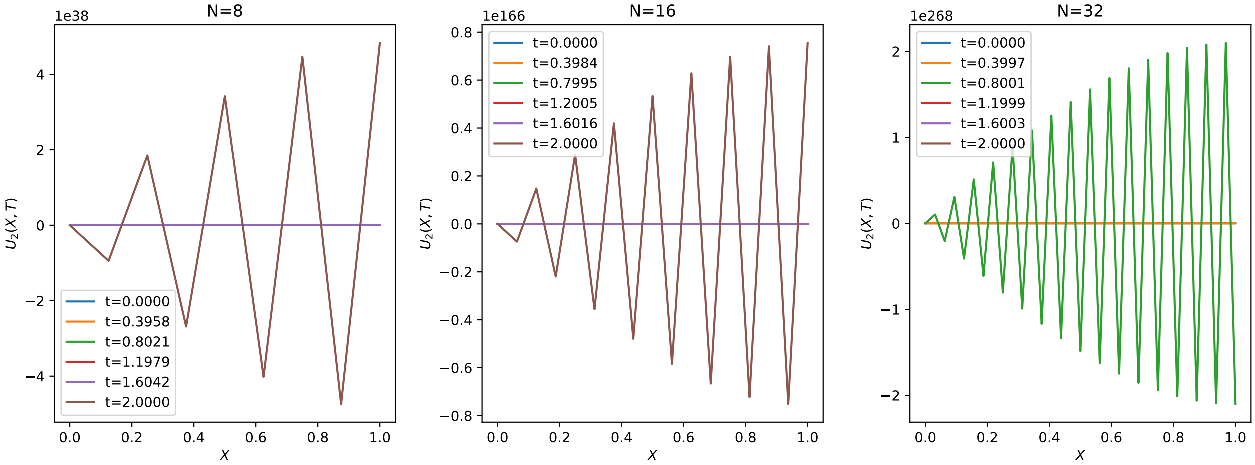

Figure 8: Graph of the numerical solution to the insulated end problem with C=$\frac{2}{3}$

(i)

X	Analytical Solution	Numerical Solution	Error
0.000	1.000000	1.000000	0.000000
0.125	0.802743	0.803650	0.000907
0.250	0.617534	0.618317	0.000783
0.375	0.454407	0.455017	0.000610
0.500	0.320010	0.318695	0.001315
0.625	0.217259	0.214294	0.002964
0.750	0.146034	0.141022	0.005012
0.875	0.104567	0.098083	0.006484
1.000	0.091001	0.084198	0.006803

Figure 9: Table of the analytical solution, numerical solution, and the error for C=$\frac{1}{2}$ and $N = 8$ at T=0.125

X	Analytical Solution	Numerical Solution	Error
0.000	1.000000	1.000000	0.000000
0.125	0.865040	0.864948	0.000092
0.250	0.735539	0.735220	0.000319
0.375	0.616656	0.616141	0.000515
0.500	0.512987	0.511997	0.000990
0.625	0.428382	0.427075	0.001307
0.750	0.365839	0.364022	0.001817
0.875	0.327479	0.325489	0.001989
1.000	0.314554	0.312353	0.002201

Figure 10: Table of the analytical solution, numerical solution, and the error for C=$\frac{1}{2}$ and $N = 8$ at T=0.25

X	Analytical Solution	Numerical Solution	Error
0.000	1.000000	1.000000	0.000000
0.125	0.927660	0.927776	0.000116
0.250	0.858101	0.858301	0.000200
0.375	0.793997	0.794325	0.000327
0.500	0.737812	0.738177	0.000366
0.625	0.691703	0.692189	0.000486
0.750	0.657443	0.657917	0.000474
0.875	0.636346	0.636916	0.000570
1.000	0.629223	0.629734	0.000511

Figure 11: Table of the analytical solution, numerical solution, and the error for C=$\frac{1}{2}$ and $N = 8$ at T=0.5

X	Analytical Solution	Numerical Solution	Error
0.000	1.000000	1.000000	0.000000
0.125	0.978935	0.979136	0.000201
0.250	0.958679	0.959065	0.000386
0.375	0.940011	0.940583	0.000572
0.500	0.923649	0.924363	0.000714
0.625	0.910220	0.911077	0.000856
0.750	0.900242	0.901175	0.000933
0.875	0.894098	0.895108	0.001010
1.000	0.892023	0.893033	0.001010

Figure 12: Table of the analytical solution, numerical solution, and the error for C=$\frac{1}{2}$ and $N = 8$ at T=1.0

(ii)

The necessary plot is shown in figure 13

As we can in figures 4 to 8, the results reach a stable solution for $C \leq \frac{1}{2}$ which agrees with the theoretical stability condition outlined in William F. Ames. *Numerical Methods for Partial Differential Equations*, 2nd Edition, Academic Press, Pages 43-46. Also from the figure 14 we can see that for $C = \frac{1}{6}$ the solution is more accurate than for other C and in fact we see that error $\propto \frac{1}{N^2}$ for $C \neq \frac{1}{6}$ and is directly proportional to $\frac{1}{N^4}$ for $C = \frac{1}{6}$ which agrees with the theoretical accuracy as we find that

$$\max_{n,m} |U_2 (n\delta X, m\delta T) - U_n^m| \leq \frac{AT}{N^2} (C + 1)$$

for $C \neq \frac{1}{6}$ and

$$\max_{n,m} |U_2 (n\delta X, m\delta T) - U_n^m| \leq \frac{AT}{N^4} \left(C^2 + 1\right)$$

We also see from figure 14 that as $C \to 0$ decreases (ignoring $C = \frac{1}{6}$) and $N \to \infty$ that the results become more accurate which agrees with the theoretical result by the above.

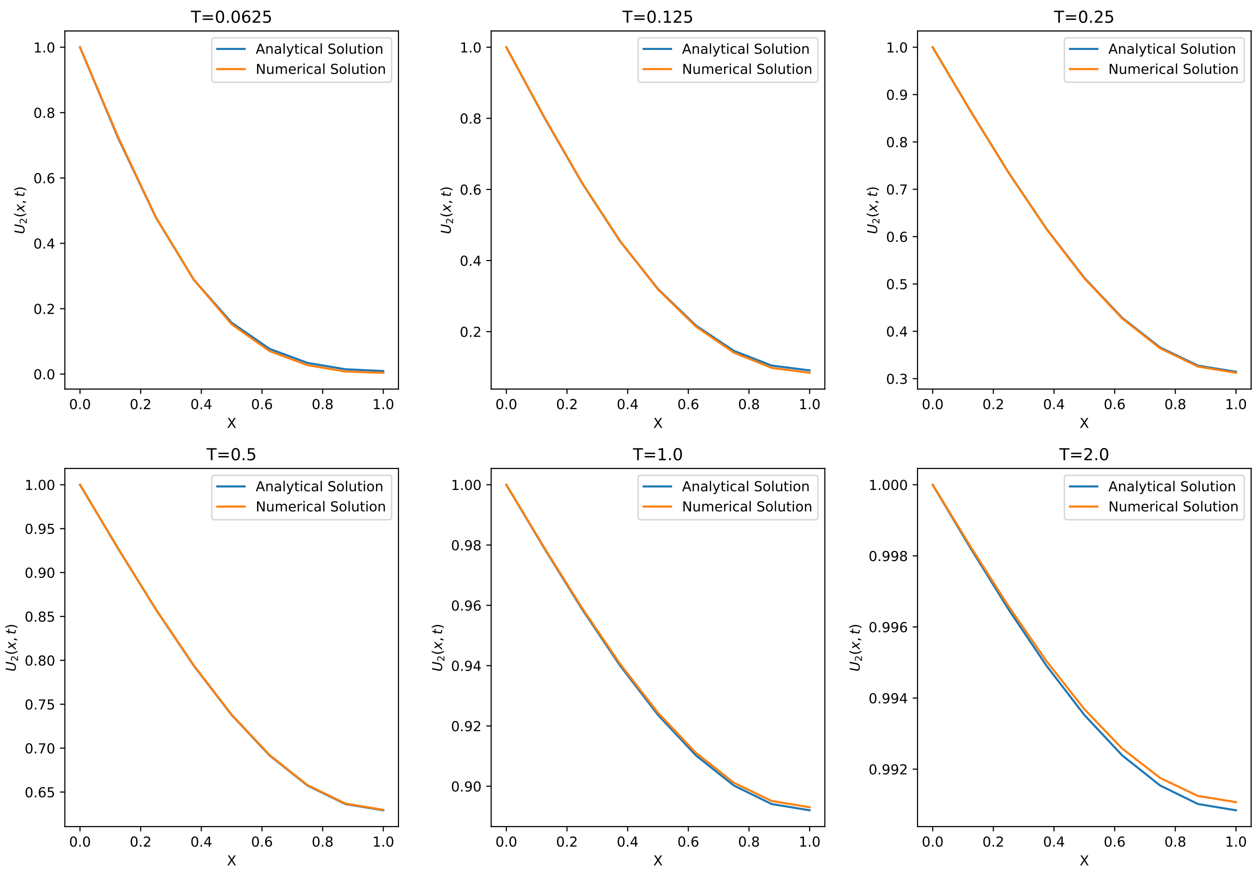

Figure 13: Graph of Analytic and Numerical solutions for C=$\frac{1}{2}$ and $N = 8$

C	$N = 8$	$N = 16$	$N = 32$
$\frac{1}{12}$	2.859815e-05	7.123668e-06	1.779301e-06
$\frac{1}{6}$	1.056440e-07	6.594871e-09	4.120564e-10
$\frac{1}{3}$	2.313432e-04	5.822777e-05	1.458153e-05

Figure 14: The maximum error in the numerical solution for $C = \frac{1}{12},\ \frac{1}{6},\ \frac{1}{3}$ and $N = 8,\ 16,\ 32$ at T=1.0

Program Listing

Helper Functions

These functions were created in order to make parts of the project less cumbersome
The following is a list of all the imports used

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.special import erfc
import pandas as pd
```

The following was used to draw tables and save them to a latex file

```
def draw_table(table, file_name, column_format):
    template = r'''\documentclass[preview]{{standalone}}
    \usepackage{{booktabs}}
    \begin{{document}}
    {}
    \end{{document}}
    '''
    dataframe = pd.DataFrame.from_dict(table)
    with open(file_name+".tex", 'w') as file:
        file.write(template.format(dataframe.to_latex(
                    escape=False,
                    column_format = column_format,
                    index = False)))
```

Analytic Solutions

The following 3 functions define solutions to the fixed-endpoint-temperature problem, insulated-end problem, and semi-infinite bar problem respectively

```
def U1(X,T, N=150):
    answer = 1-X
    for n in range(1, N+1):
        term = -2*np.sin(n*np.pi*X)*np.e**-(n**2*np.pi**2*T)/(np.pi*n)
        answer += term
    return answer

def U2(X,T, N=150):
    answer = 1
    for n in range(0, N+1):
        term = -2*np.sin((n+0.5)*np.pi*X)*np.e**-((n+0.5)**2*np.pi**2*T)
            /(np.pi*(n+0.5))
        answer += term
    return answer

def U3(X,T, K = 1):
    return erfc(0.5*X/(K*T)**0.5) # assuming K = 1
```

The following 3 functions define the heatflux forthe fixed-endpoint-temperature problem, insulated-end problem, and semi-infinite bar problem respectively

```
def U1_X(X,T, N=150):
    answer = -1
```

```
    for n in range(1, N+1):
        term = -2*np.cos(n*np.pi*X)*np.e**-(n**2*np.pi**2*T)
        answer += term
    return answer

def U2_X(X,T, N=150):
    answer = 0
    for n in range(1, N+1):
        term = -2*np.cos((n-0.5)*np.pi*X)*np.e**-((n-0.5)**2*np.pi**2*T)
        answer += term
    return answer

def U3_X(X,T, K = 1):
    return -np.e**(-X**2/(4*K*T))/(np.pi*K*T)**0.5 # assuming K = 1
```

The following draws all the figures seen in question 2

```
X_Values = [0.125*n for n in range(9)]
T_Values = [0.0625, 0.125, 0.25, 0.5, 1.0, 2.0]

grid_1 = [U1(X, 0.25) for X in X_Values]
grid_2 = [U2(X, 0.25) for X in X_Values]
grid_3 = [U3(X, 0.25) for X in X_Values]

Table_PT1 = {r"$x$":X_Values,
             r"\multicolumn{1}{|p{3cm}|}{\centering fixed-endpoint-
                temperature problem}":["{0:.6f}".format(y) for y in
                grid_1],
             r"\multicolumn{1}{|p{3cm}|}{\centering insulated-end problem
                }":["{0:.6f}".format(y) for y in grid_2],
             r"\multicolumn{1}{|p{3cm}|}{\centering semi-infinite bar
                problem}":["{0:.6f}".format(y) for y in grid_3]}

draw_table(Table_PT1, "table_PT1", "|c|l|l|l|")

grid_4 = [[U1(X, T) for X in X_Values] for T in T_Values]
grid_5 = [[U2(X, T) for X in X_Values] for T in T_Values]
grid_6 = [[U3(X, T) for X in X_Values] for T in T_Values]


fig = plt.figure(figsize=(13,5))


for i in range(len(grid_4)):
    ax1 = fig.add_subplot(1, 3, 1)
    ax1.plot(X_Values, grid_4[i], label="T="+str(T_Values[i]))
    ax1.legend()
    ax1.set_title("fixed-endpoint-temperature problem")
    ax1.set_xlabel("X")
    ax1.set_ylabel(r"$U_1\left(X,T\right)$")


for i in range(len(grid_5)):
    ax2 = fig.add_subplot(1, 3, 2)
```

```
    ax2.plot(X_Values, grid_5[i], label="T="+str(T_Values[i]))
    ax2.legend()
    ax2.set_title("insulated-end problem")
    ax2.set_xlabel("X")
    ax2.set_ylabel(r"$U_2\left(X,T\right)$")


for i in range(len(grid_6)):
    ax3 = fig.add_subplot(1, 3, 3)
    ax3.plot(X_Values, grid_6[i], label="T="+str(T_Values[i]))
    ax3.legend()
    ax3.set_title("semi-infinite bar problem")
    ax3.set_xlabel("X")
    ax3.set_ylabel(r"$f\left(\xi\right)$")

plt.tight_layout()
fig.savefig("PT1_1.pdf")

grid_7 = [-U1_X(0, T) for T in T_Values]
grid_8 = [-U2_X(0, T) for T in T_Values]
grid_9 = [-U3_X(0, T) for T in T_Values]

fig = plt.figure(figsize=(13,5))

ax1 = fig.add_subplot(1, 3, 1)
ax1.plot(T_Values, grid_7)
ax1.set_title("fixed-endpoint-temperature problem")
ax1.set_xlabel("t")
ax1.set_ylabel(r"$U_1\left(x,t\right)$")


ax2 = fig.add_subplot(1, 3, 2)
ax2.plot(T_Values, grid_8)
ax2.set_title("insulated-end problem")
ax2.set_xlabel("t")
ax2.set_ylabel(r"$U_2\left(x,t\right)$")


ax3 = fig.add_subplot(1, 3, 3)
ax3.plot(T_Values, grid_9)
ax3.set_title("semi-infinite bar problem")
ax3.set_xlabel("t")
ax3.set_ylabel(r"$f\left(\xi\right)$")


plt.tight_layout()
fig.savefig("PT1_2.pdf")
```

Numerical Integration

The following is used to calculate the numerical solution for the insulated-end problem given the Courant number C and number of equidistant x-coordinates N

```
def U2_num(C, N, maxT = 2):
```

```
    dx = 1/N
    dt = C*dx*dx

    T_Values = [0]
    X_Values = [j*dx for j in range(0, N+1)]

    U = [[0.5] + [0 for _ in range(N)]]
    while T_Values[-1] < maxT:
        U_T = [1]
        T_Values.append(T_Values[-1] + dt)
        U_prev = U[-1]
        for i in range(1,N):
            U_i = U_prev[i] + C*(U_prev[i+1]-2*U_prev[i] + U_prev[i-1])
            U_T.append(U_i)
        U_N = U_prev[N] + C*(2*U_prev[N-1]-2*U_prev[N])
        U_T.append(U_N)
        U.append(U_T)
    return U, X_Values, T_Values
```

The following draws all the figures seen in question 3

```
C_Values = [1/12, 1/6, 1/3, 1/2, 2/3]
N_Values = [8,16,32]

for i in range(len(C_Values)):
    fig = plt.figure(figsize=(13,5))
    axs = [fig.add_subplot(1, 3, l+1)  for l in range(len(N_Values))]
    C = C_Values[i]
    for j in range(len(N_Values)):
        ax = axs[j]
        N = N_Values[j]
        ax.set_title("N={}".format(N))
        grid, X_Vals, T_Vals = U2_num(C,N)
        for k in range(5):
            ax.plot(X_Vals, grid[k*len(grid)//5], label="t={:.4f}".format
                (T_Vals[k*len(grid)//5]))
        ax.plot(X_Vals, grid[-1], label="t={:.4f}".format(T_Vals[-1]))
        ax.legend()
        ax.set_xlabel(r"$X$")
        ax.set_ylabel(r"$U_2\left(X,T\right)$")
    plt.tight_layout()
    plt.show()
    fig.savefig("PT2_{}.pdf".format(i+1))

U2_ana_T_00625 = [U2(n/8, 0.0625) for n in range(9)]
U2_ana_T_0125 = [U2(n/8, 0.125) for n in range(9)]
U2_ana_T_025 = [U2(n/8, 0.25) for n in range(9)]
U2_ana_T_05 = [U2(n/8, 0.5) for n in range(9)]
U2_ana_T_10 = [U2(n/8, 1.0) for n in range(9)]
U2_ana_T_20 = [U2(n/8, 2.0) for n in range(9)]

U2_num_grid, _, T_num_Values = U2_num(1/2, 8)

U2_num_T_00625 = U2_num_grid[T_num_Values.index(0.0625)]
```

```
U2_num_T_0125 = U2_num_grid[T_num_Values.index(0.125)]
U2_num_T_025 = U2_num_grid[T_num_Values.index(0.25)]
U2_num_T_05 = U2_num_grid[T_num_Values.index(0.5)]
U2_num_T_10 = U2_num_grid[T_num_Values.index(1.0)]
U2_num_T_20 = U2_num_grid[T_num_Values.index(2.0)]


Table_00625 = {"$X$":[n/8 for n in range(9)],
              "Analytical Solution":U2_ana_T_00625,
              "Numerical Solution":U2_num_T_00625,
              "Error":[abs(U2_ana_T_00625[i]-U2_num_T_00625[i]) for i in
                  range(9)]}

Table_0125 = {"$X$":[n/8 for n in range(9)],
              "Analytical Solution":U2_ana_T_0125,
              "Numerical Solution":U2_num_T_0125,
              "Error":[abs(U2_ana_T_0125[i]-U2_num_T_0125[i]) for i in
                 range(9)]}

Table_025 = {"$X$":[n/8 for n in range(9)],
              "Analytical Solution":U2_ana_T_025,
              "Numerical Solution":U2_num_T_025,
              "Error":[abs(U2_ana_T_025[i]-U2_num_T_025[i]) for i in
                 range(9)]}

Table_05 = {"$X$":[n/8 for n in range(9)],
              "Analytical Solution":U2_ana_T_05,
              "Numerical Solution":U2_num_T_05,
              "Error":[abs(U2_ana_T_05[i]-U2_num_T_05[i]) for i in range
                 (9)]}

Table_10 = {"$X$":[n/8 for n in range(9)],
              "Analytical Solution":U2_ana_T_10,
              "Numerical Solution":U2_num_T_10,
              "Error":[abs(U2_ana_T_10[i]-U2_num_T_10[i]) for i in range
                 (9)]}

Table_20 = {"$X$":[n/8 for n in range(9)],
              "Analytical Solution":U2_ana_T_20,
              "Numerical Solution":U2_num_T_20,
              "Error":[abs(U2_ana_T_20[i]-U2_num_T_20[i]) for i in range
                 (9)]}


draw_table(Table_0125, "table_0125", "|c||l|l|l|")
draw_table(Table_025, "table_025", "|c||l|l|l|")
draw_table(Table_05, "table_05", "|c||l|l|l|")
draw_table(Table_10, "table_10", "|c||l|l|l|")


fig = plt.figure(figsize=(13,9))

ax = fig.add_subplot(2, 3, 1)
```

```
ax.plot([n/8 for n in range(9)], Table_00625["Analytical Solution"],label
    ="Analytical Solution")
ax.plot([n/8 for n in range(9)], Table_00625["Numerical Solution"],label
    ="Numerical Solution")
ax.set_xlabel("X")
ax.set_ylabel(r"$U_2\left(x,t\right)$")
ax.legend()
ax.set_title("T=0.0625")

ax = fig.add_subplot(2, 3, 2)
ax.plot([n/8 for n in range(9)], Table_0125["Analytical Solution"],label
    ="Analytical Solution")
ax.plot([n/8 for n in range(9)], Table_0125["Numerical Solution"],label="
    Numerical Solution")
ax.set_xlabel("X")
ax.set_ylabel(r"$U_2\left(x,t\right)$")
ax.legend()
ax.set_title("T=0.125")

ax = fig.add_subplot(2, 3, 3)
ax.plot([n/8 for n in range(9)], Table_025["Analytical Solution"],label="
    Analytical Solution")
ax.plot([n/8 for n in range(9)], Table_025["Numerical Solution"],label="
    Numerical Solution")
ax.set_xlabel("X")
ax.set_ylabel(r"$U_2\left(x,t\right)$")
ax.legend()
ax.set_title("T=0.25")

ax = fig.add_subplot(2, 3, 4)
ax.plot([n/8 for n in range(9)], Table_05["Analytical Solution"],label="
    Analytical Solution")
ax.plot([n/8 for n in range(9)], Table_05["Numerical Solution"],label="
    Numerical Solution")
ax.set_xlabel("X")
ax.set_ylabel(r"$U_2\left(x,t\right)$")
ax.legend()
ax.set_title("T=0.5")

ax = fig.add_subplot(2, 3, 5)
ax.plot([n/8 for n in range(9)], Table_10["Analytical Solution"],label="
    Analytical Solution")
ax.plot([n/8 for n in range(9)], Table_10["Numerical Solution"],label="
    Numerical Solution")
ax.set_xlabel("X")
ax.set_ylabel(r"$U_2\left(x,t\right)$")
ax.legend()
ax.set_title("T=1.0")

ax = fig.add_subplot(2, 3, 6)
ax.plot([n/8 for n in range(9)], Table_20["Analytical Solution"],label="
    Analytical Solution")
ax.plot([n/8 for n in range(9)], Table_20["Numerical Solution"],label="
```

```
    Numerical Solution")
ax.set_xlabel("X")
ax.set_ylabel(r"$U_2\left(x,t\right)$")
ax.legend()
ax.set_title("T=2.0")
plt.tight_layout()
plt.savefig("PT2_ii.pdf")

U2_num_1, X_Values_1, T_Values_1 = U2_num(1/12, 8) # C = 1/12, N = 8
U2_num_2, X_Values_2, T_Values_2 = U2_num(1/12, 16) # C = 1/12, N = 16
U2_num_3, X_Values_3, T_Values_3 = U2_num(1/12, 32) # C = 1/12, N = 32

max_error_1 = max([abs(U2(n/8,2)-U2_num_1[-1][n]) for n in range(9)])
max_error_2 = max([abs(U2(n/16,2)-U2_num_2[-1][n]) for n in range(17)])
max_error_3 = max([abs(U2(n/32,2)-U2_num_3[-1][n]) for n in range(33)])

U2_num_4, X_Values_4, T_Values_4 = U2_num(1/3, 8) # C = 1/3, N = 8
U2_num_5, X_Values_5, T_Values_5 = U2_num(1/3, 16) # C = 1/3, N = 16
U2_num_6, X_Values_6, T_Values_6 = U2_num(1/3, 32) # C = 1/3, N = 32

max_error_4 = max([abs(U2(n/8,2)-U2_num_4[-1][n]) for n in range(9)])
max_error_5 = max([abs(U2(n/16,2)-U2_num_5[-1][n]) for n in range(17)])
max_error_6 = max([abs(U2(n/32,2)-U2_num_6[-1][n]) for n in range(33)])

U2_num_7, X_Values_7, T_Values_7 = U2_num(1/6, 8) # C = 1/6, N = 8
U2_num_8, X_Values_8, T_Values_8 = U2_num(1/6, 16) # C = 1/6, N = 16
U2_num_9, X_Values_9, T_Values_9 = U2_num(1/6, 32) # C = 1/6, N = 32

max_error_7 = max([abs(U2(n/8,2)-U2_num_7[-1][n]) for n in range(9)])
max_error_8 = max([abs(U2(n/16,2)-U2_num_8[-1][n]) for n in range(17)])
max_error_9 = max([abs(U2(n/32,2)-U2_num_9[-1][n]) for n in range(33)])

Table_errors = {"C":[r"$\frac{1}{12}$",r"$\frac{1}{6}$",r"$\frac{1}{3}$
    "],
                r"$N=8$": [max_error_1, max_error_7, max_error_4],
                r"$N=16$": [max_error_2, max_error_8, max_error_5],
                r"$N=32$": [max_error_3, max_error_9, max_error_6],}

draw_table(Table_errors, "table_errors", "|c||l|l|l|")
```