CP 1: Transient Heat Conduction

Joseph F. Specht IV

NPRE 349

Prof. Caleb Brooks

4 March 2024

Table of Contents

1. Introduction & Background

This section will introduce the problem we are investigating and go over the required theoretical background to solve said problem.

1.1. Problem Statement

The problem under investigation is as follows:

The heat transfer coefficient for hydrogen flowing over a sphere is to be determined by observing the temperature time history of a sphere fabricated from pure copper. The sphere, which is 20 mm in diameter, is at 75C before it is inserted into the gas stream having a temperature of 27C. A thermocouple on the outer surface of the sphere indicates a temperature of 57C after 97s of the sphere in the hydrogen. [1]

This problem is given in [1] of the references, which is the . From *Fundamentals of Heat and Mass Transfer* by Bergman and Lavine [2], we can find the density of copper to be 8933 $kg \cdot m^{-3}$ and the isobaric specific heat capacity of copper to be 385 $J \cdot kg^{-1} \cdot K^{-1}$.

The solution for this problem will be done in python, specifically Jupyter Notebook, the code for which can be found in 6.2. Appendix B: Python Code.

1.2. Theory

This section will detail the theoretical background needed to perform this computer project, which includes the definition and derivation of various quantities needed.

1.2.1. Characteristic Length

The characteristic length is a helpful quantity used to compare the volume of a geometry to the surface area of the same geometry. The characteristic length is an especially useful quantity in the field of heat transfer as the comparison of volume, which serves to store heat, to surface area, which serves to transfer heat, is an effective method of trans-geometric comparison. The characteristic length is given as follow:

$$L_C = \frac{V}{A} = A \quad (1)$$

Where V is the volume of the geometry and A is the surface area of the geometry in contact with the external fluid.

1.2.2. Biot Number and Lumped Capacitance

The Biot number is a quantity used to determine if simplifications can be made to the temperature gradient inside a material. The Biot number compares the temperature distribution at the boundary of a material to the temperature distribution inside a material and is given as follows:

$$B_i = \frac{hL_c}{k_s} \quad (2)$$

Where h is the heat transfer coefficient of the surrounding fluid, L_c is the characteristic length, and k_s is the thermal conductivity of the material.

It was previously mentioned that the Biot number can be used to simplify the complexity of heat transfer problems, but this only occurs when the Biot number is less than 0.1. Once the condition of $B_i < 0.1$ is satisfied, we can apply lumped capacitance, which states the temperature difference between the surface and edge is so small that it can be considered zero. Lumped capacitance can generally be applied in small geometries or when the heat transfer coefficient is low relative to the thermal conductivity.

1.2.3. Fourier Number

The Fourier number is a dimensionless time parameter that characterizes the time scale for diffusive systems. Specifically in heat transfer, the Fourier number compares the time elapsed to the thermal diffusion time scale, which can be given as follows:

$$F_0 = \frac{\alpha t}{L_c^2} \tag{3}$$

Where α is the thermal diffusivity and t is the time. However, we also need an equation for α, which can be given as follows:

$$\alpha = \frac{k_s}{\rho c_p} \tag{4}$$

Where ρ is the mass density of the material and c_p is the isobaric heat capacity of the material.

2. Derivation of Equations

This section will detail the derivations needed to arrive at the governing equation and to go from said governing equation to the closed form analytical solution and the numerical integration method.

2.1. Governing Equation

To begin, we derived the governing equation for this system from the energy balance equation, which is given as follows:

$$\dot{E}_{stored} = \dot{E}_{in} - \dot{E}_{out} + \dot{E}_{gen} \tag{5}$$

Where Ė refers to the change in energy over time and the subscripts, stored, in, out, and gen refer to the change in energy stored in the volume, change in energy flowing into the system, change of energy flowing out of the system, and change in energy generated in the system respectively. For this investigation, we are considering a copper sphere dropped into a cooler hydrogen fluid, which means there is no energy flowing in or generated. Therefore, Eq. (5) becomes:

$$\dot{E}_{stored} = - \dot{E}_{out} \tag{6}$$

We can find each of the terms in Eq. (6) as follows:

$$\dot{E}_{stored} = \rho c_p V \frac{\partial T}{\partial t} \tag{7}$$

Where T is the temperature of the copper sphere. And,

$$\dot{E}_{out} = hA\left(T_s - T_{surr}\right) \tag{8}$$

Where T_s is the surface temperature and T_{surr} is the surrounding fluid temperature. Eq. (7) and Eq. (8) can then be substituted into Eq. (6) to give:

$$\rho c_p V \frac{\partial T}{\partial t} = - hA\left(T_s - T_{surr}\right) \tag{9}$$

Next, the assumption of a low biot number can be made as the geometry of the problem is small and the heat transfer rate of copper is much higher than the heat transfer rate of hydrogen gas.

Therefore, the Biot number is assumed to be low and lumped capacitance applies, which gives Eq. (9) the final form:

$$\rho c_p V \frac{\partial \bar{T}(t)}{\partial t} = - hA\left(\overline{T}(t) - T_f\right) \tag{10}$$

Where T bar represents the mean temperature of the copper sphere.

2.2. Analytical Solution

To solve the analytical solution, it helps to perform the following substitution:

$$\theta(t) = \overline{T}(t) - T_{surr} \tag{11}$$

Where θ is the temperature difference, which can be derived to show the derivative of mean temperature is the negative of the derivative of θ. Therefore,

$$\frac{\partial \theta(t)}{\partial t} = - \frac{hA}{\rho c_p V}\theta(t) \tag{12}$$

Next, we will combine all the terms not θ or its derivatives into a time constant τ, which characterizes the characteristic time scale heat flows out of the system.

$$\tau = \frac{\rho c_p V}{hA} \tag{13}$$

This time constant can be substituted back into Eq. (12) to give,

$$\frac{\partial \theta(t)}{\partial t} = - \tau\theta(t) \tag{12}$$

Which can be recognized as a first-order, homogeneous differential equation. As such, the solution to Eq. (12) is easily found as:

$$\theta(t) = \theta_0 exp\left(-\frac{t}{\tau}\right) \tag{13}$$

Substituting Eq. (11) back into Eq. (13) then gives the solution to temperature in the sphere as a function of time.

$$\overline{T}(t) = \left(\overline{T}(0) - T_{surr}\right)exp\left(-\frac{t}{\tau}\right) + T_{surr} \tag{13}$$

We can rearrange Eq. (13) to solve for h, which is an unknown quantity.

$$h = \frac{\rho c_p V}{3t} ln\left(\frac{\overline{T}(0)-T_{surr}}{\overline{T}(t)-T_{surr}}\right) \tag{14}$$

Similarly, Eq. (13) can be rearranged to find the value for time.

$$t = \tau\, ln\left(\frac{\overline{T}(0)-T_{surr}}{\overline{T}(t)-T_{surr}}\right) \tag{15}$$

The equations derived in this section, Eq. (11) - Eq. (15), will be used to solve the parameters of h and final time as is done in 3.1. Finding Parameters and to graph the analytical solution as is done in 3.2. Analytical Results.

2.3. Numerical Solution

To solve this problem is rather trivial, but more complex problems may not have a closed form analytical solution. For problems of this type, we often deploy a method known as finite differencing.

To begin we start by converting derivatives into differentials, which can be done as almost all physical processes, including heat, occur continuously on a macroscopic scale. The separation of derivatives into differentials is given as follows:

$$\frac{\partial T}{\partial t} = \frac{\Delta T}{\Delta t} = \frac{T_{i+1}-T_i}{t_{i+1}-t_i} \tag{16}$$

Where the subscript i denotes the iteration step we are currently on. We can then plug Eq. (16) back into Eq. (10), rearrange to solve for T_{i+1}, and substitute Eq. (3) in to get:

$$T\left(t_{i+1}\right) = -\ F_0\left(T(t_i)\ - T_{surr}\right) + \ T(t_i) \tag{17}$$

3. Results

This section will detail the resultant

3.1. Finding Parameters

In this section, the values for the system, which are given in 1.1. Problem Statement, are substituted into the equations we previously derived to find the unknown parameters we need.

To solve for h we can substitute said values into Eq. (14).

$$h\ =\ 55.548\ W{\cdot}m^{-2}{\cdot}K^{-1} \tag{15}$$

Similarly, we can substitute the same values into Eq. (15) to get the time limit for graphing.

$$t\ =\ 1749.365\ s \tag{15}$$

This value will be rounded to 1750 s for ease of use, which will be ubiquitous throughout the rest of the calculations. However, h will be applicable in every situation except in 3.2.3. Variation of Heat Transfer Coefficient.

Using a final time where the temperature is found by assuming the final temperature is 0.01 ℃ higher than the ambient temperature. This is a valid assumption as the equation modeling this system takes the form of exponential decay, which approaches the ambient temperature asymptotically. Therefore, we select a temperature slightly higher than the ambient temperature to preserve computational resources and because any temperature below this value is effectively thermalized to ambient temperature.

3.2. Analytical Results

This section will detail and describe the analytical solution as described in 2.2. Analytical Solution.

3.2.1. Analytical Solution

This section will describe the case described in 1.1. Problem Statement, which can be seen in Fig. 1 below.

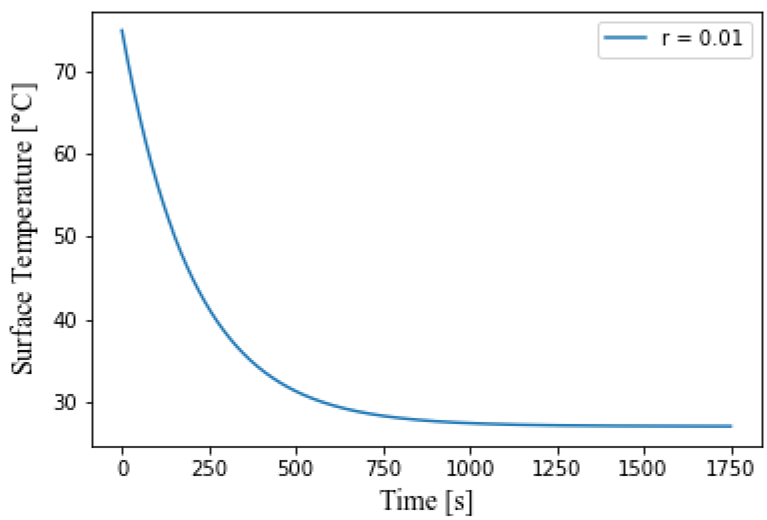


Fig. 1. Analytical Solution.

As can be seen above, the analytical solution follows the expected exponential decay typical of diffusive systems. As was discussed in 3.1. Finding Parameters, the temperature of this model never truly reaches the temperature of the surrounding hydrogen gas.

3.2.2. Variation of Radius

This section will describe the effect of changing the radius of the copper sphere, which is pictured below in Fig. 2.

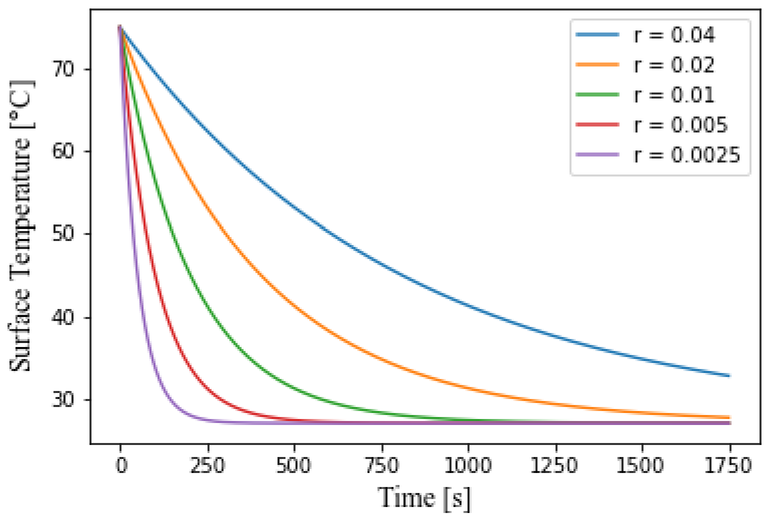


Fig. 2. Effect of Changing Radius in the Analytical Solution.

As detailed above, the temperature decays more quickly as the radius of the sphere decreases. This is to be expected as the characteristic length increases as the radius increases, which means the heat has to "travel" farther the larger the radius.

3.2.3. Variation of Heat Transfer Coefficient

This section will describe the effect of changing the heat transfer coefficient of the fluid surrounding the copper sphere, which is pictured below in Fig. 3.

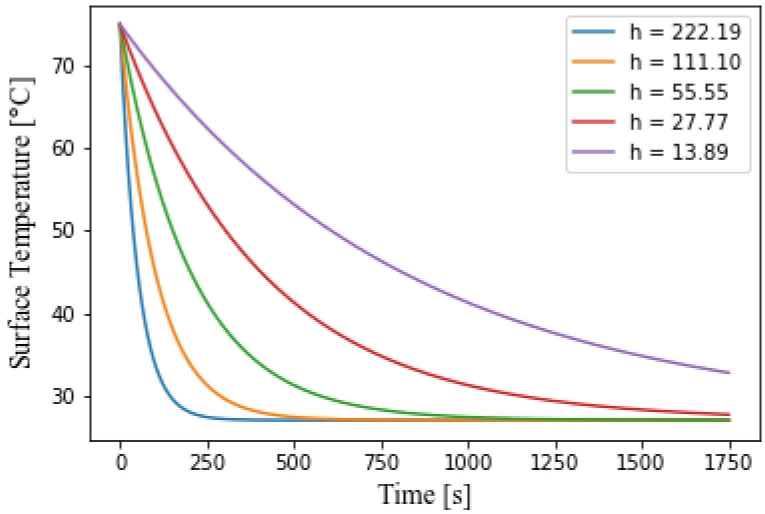


Fig. 3. Effect of Changing the Heat Transfer Coefficient in the Analytical Solution.

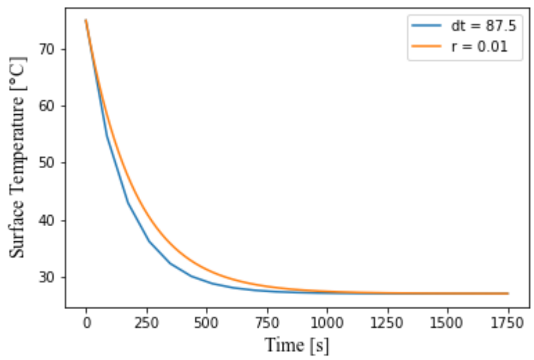
As detailed above, the temperature decays more quickly the higher the heat transfer coefficient of the surrounding fluid. This is to be expected as the heat transfer coefficient describes how quickly the surrounding fluid can remove heat from an object in the fluid stream, thus, leading to an increase in heat transfer and quicker thermal decay the higher the heat transfer coefficient.

3.3. Numerical Results

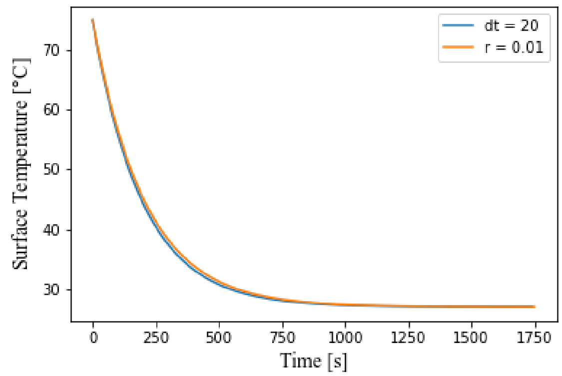
This section will detail and explain the analytical solution as described in 2.3. Numerical Solution.

3.3.1. Optimal Time Step

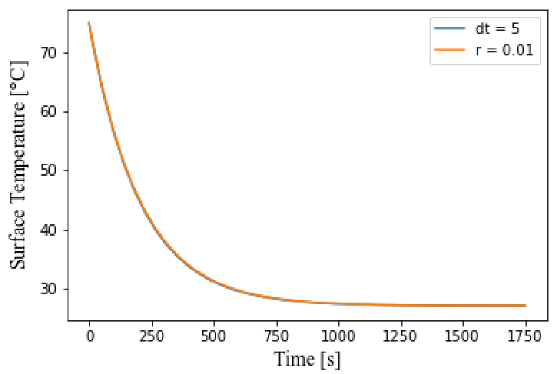
This section will solve the numerical solution with a variety of time steps to determine the optimal balance between accuracy and run-time for the numerical solution, which can be seen below in Fig. 4.



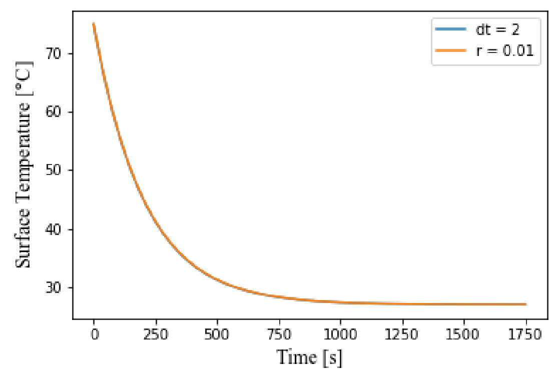
(a)



(b)



(c)



(d)

Fig. 4. Numerical Solution with Various Time Steps.

As can be seen in Fig. 4, the accuracy of the numerical solution increases with a decrease in the time step. However, once the numerical solution has a sufficiently accurate depiction of the analytical solution, which is ~2 seconds in this case, any timestep shorter than that will only serve to increase run time and only minimally improve the results. Conversely, the higher the time step, the lower the agreement of the numerical solution with the analytical solution.

There is an interesting phenomenon when the time step is so low that the decrease in temperature occurs over such a long period that the temperature ends up lower than the ambient temperature. This time step makes the temperature appear to follow the form of a damped harmonic oscillator, but this is merely a graphical curiosity and not indicative of the true system.

3.3.2. Various Time Steps

This section will give a variety of time steps together to show how their variation affects the decay of the numerical solution, which is given below in Fig. 5.

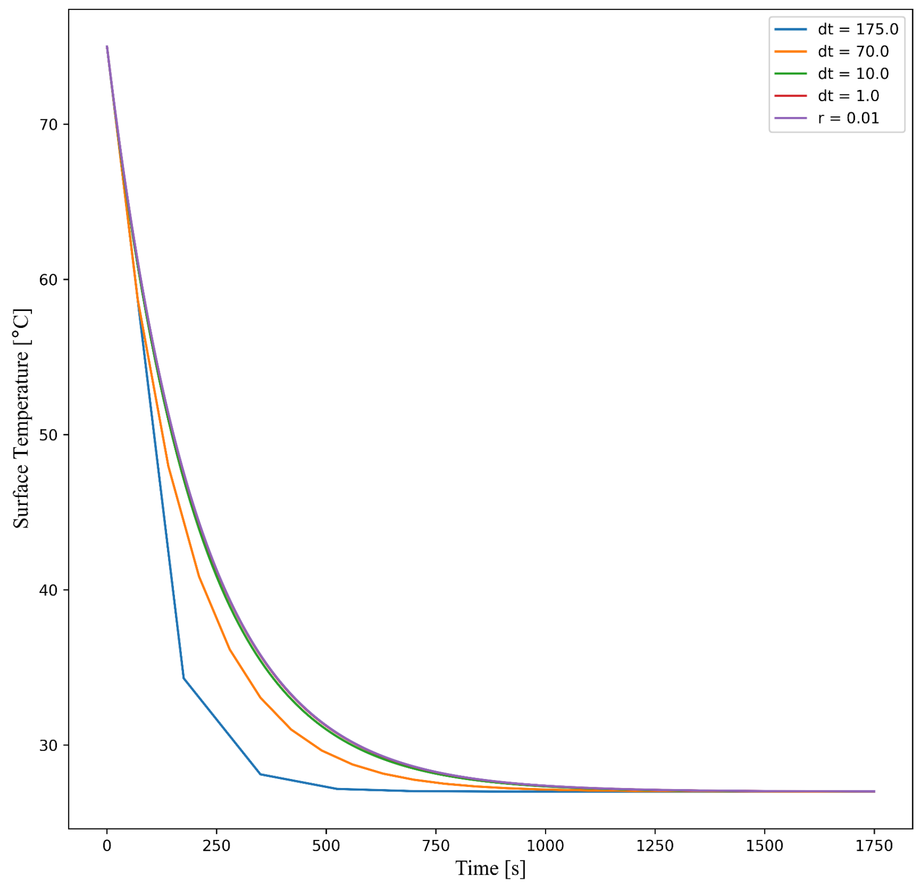


Fig. 5. Various Time Steps for the Analytical Solution.

As detailed above, the various time steps affect the accuracy of the numerical solution as compared to the analytical solution, which is given by the $r = 0.01$m curve. The larger the time step, the more quickly the numerical solution decays, which is to be expected as the rate of heat transfer is proportional to the temperature difference between the sphere and the surrounding fluid. At the first time step, this temperature difference is the highest, thus, the rate of heat transfer will be the highest. This high heat transfer rate will be used for the duration of the time step length, which leads to the temperature decaying more quickly as this heat transfer rate is truly lower as time progresses.

4. Conclusions

In this experiment, it was found that for a copper spherical geometry surrounded by a cooler hydrogen fluid, radius and heat transfer coefficient affect the rate of heat transfer. During this investigation, it was found that h, the heat transfer coefficient for hydrogen gas, was 55.548 $W \cdot m^{-2} \cdot K^{-1}$. It was also found that increasing the radius of the sphere increases L_c, the characteristic length, which, in turn, decreases the rate of heat transfer. Another finding was an increase in h caused the system to thermalize to the surrounding fluid quicker. For a numerical solution, it was found that the optimal time step for this problem was roughly 2 seconds. Optimal meaning the accuracy of the numerical solution aligned strongly with the analytical solution while not drastically increasing the run time of the code. If the goal is maximizing the heat transfer rate between a geometry surrounded by a coolant, the heat transfer coefficient of the fluid should be as high as possible and the characteristic length of the geometry should be as low as possible.

5. References

[1] Caleb Brook. UNIVERSITY OF ILLINOIS DEPARTMENT OF NUCLEAR, PLASMA AND RADIOLOGICAL ENGINEERING NPRE 349 – Introduction to NPRE Heat Transfer. GradeScope. Retrieved March 4, 2024.

[2] Bergman, Theodore L., and Adrienne S. Lavine. *Fundamentals of Heat and Mass Transfer*. 8th ed., John Wiley & Sons, Inc., 2017.

6. Appendix

6.1. Appendix A: Output File

Calculated H value:

55.5477261759478

Max time and rounded time:

1749.36522813319

1750

6.2. Appendix B: Python Code

```
import numpy as np

import numpy.linalg as la

from matplotlib import pyplot as plt

from tabulate import tabulate

from sympy import *

from sympy.plotting import plot


#Plot Variables

font = {'fontname':'Times New Roman'}

size = {'fontsize':'14'}

save_path = r'C:\Users\reape\Desktop\Sp24\NPRE 349\CP 1\Plots\\'



#Analytical Solution


T, Ts, T0, rho, cp, Lc, h, t, r = S('T, T_surrounding, Tbar_0, rho, c_p, L_c, h, t, r')


Lc = r / 3

tau = rho * cp * Lc / h


T_an = Ts + (T0 - Ts) * exp(-t/tau)
```

```
time_from_temp = tau * log((T0 - Ts) / (T - Ts))

h_from_time = rho * cp * r * log((T0 - Ts) / (T - Ts)) / (3*t)


display(T_an, time_from_temp)




#Substitution

T_an_sub1 = T_an.subs([(Ts, 27), (T0, 75), (rho, 8933), (cp, 385)])

tfT_sub = time_from_temp.subs([(Ts, 27), (T0, 75), (rho, 8933), (cp, 385)])

hft_sub = h_from_time.subs([(Ts, 27), (T0, 75), (rho, 8933), (cp, 385)])


#Finding H

h_H = hft_sub.subs([(r, 20e-3 / 2), (T, 57), (t, 97)])


T_an_sub2 = T_an_sub1.subs(h, h_H)

tfT_sub = tfT_sub.subs(h, h_H)


#Finding Time limit

t_max = tfT_sub.subs([(r, 20e-3 / 2), (T, 27.01)]).evalf()

t_limit = int(t_max) + 1


#Displaying

print('Calculated H value')
```

```
display(h_H.evalf())
print('t_max and Rounded t')
display(t_max, t_limit)

#Output File
g = open('NPRE 349_CP 1_Output', 'a')
g.write('Calculated H value:\n')
g.write(str(h_H.evalf()))
g.write('\n')
g.write('Max time and rounded time:\n')
g.write(str(t_max))
g.write('\n')
g.write(str(t_limit))


#Defining Analytical Plotting

def analytical_plotting_r(radius):
    t_an_plot = np.arange(0,t_limit + 1, 1)
    temp_an_plot = np.zeros(len(t_array_A1))
    for i in range(len(t_an_plot)):
        temp_an_plot[i] = T_an_sub2.subs([(r, radius), (t, t_an_plot[i])]).evalf()
    plt.plot(t_an_plot, temp_an_plot, label = "r = {}".format(radius))
```

```
def analytical_plotting_h(heat_trans_coeff):
    t_an_plot = np.arange(0,t_limit + 1, 1)
    temp_an_plot = np.zeros(len(t_array_A1))
    for i in range(len(t_an_plot)):
        temp_an_plot[i] = T_an_sub1.subs([(h, heat_trans_coeff), (r, 20e-3/2), (t,
t_an_plot[i])]).evalf()
    plt.plot(t_an_plot, temp_an_plot, label = "h = {}".format(round(heat_trans_coeff, 2)))


#Plotting the Analytical Solution
analytical_plotting_r(20e-3/2)


plt.legend()
plt.ylabel('Surface Temperature [$\degree$C]',**font, **size)
plt.xlabel('Time [s]', **font, **size)
plt.savefig(save_path+'A1 Constant Radius')
plt.show()


#Plotting Varius Radii
initial_radius_A2 = 20e-3 / 2
radius_array_A2 = np.array([4, 2, 1, 1/2, 1/4]) * initial_radius_A2
```

```
for radius_A2 in radius_array_A2:
    analytical_plotting_r(radius_A2)


plt.legend()
plt.ylabel('Surface Temperature [$\degree$C]',**font, **size)
plt.xlabel('Time [s]', **font, **size)
plt.savefig(save_path+'A2 Changing Radius')
plt.show()


#Plotting Varius h
h_array_A2 = np.array([4, 2, 1, 1/2, 1/4]) * h_H


for h_A2 in h_array_A2:
    analytical_plotting_h(h_A2.evalf())


plt.legend()
plt.ylabel('Surface Temperature [$\degree$C]',**font, **size)
plt.xlabel('Time [s]', **font, **size)
plt.savefig(save_path+'A2 Changing h')
plt.show()
```

```
#Numerical Solution Variables

k, dt = S('k, dt')

alpha = k/rho/cp
F0 = (alpha / Lc**2) * dt
Bi = h*Lc/k

dT = -Bi * F0 * (T0 - Ts)

#Substituting Values into dT
dT_sub = dT.subs([(h, h_H.evalf()), (T_s, 27), (cp, 385), (r, 20e-3/2), (rho, 8933)])

#Defining Numerical Function

def numerical_plotting(dt_input):
    time = np.arange(0,t_limit + 1, dt_input)
    temp = np.zeros(len(time))
    temp[0] = 75
    for i in range(1,len(time)):
        temp[i] = temp[i-1] + dT_sub.subs([(T0, temp[i-1]), (dt, dt_input)])
    plt.plot(time, temp, label = "dt = {}".format(round(dt_input, 2)))
```

```
#Plotting the Numerical Solution

#dt = 2
numerical_plotting(2)
analytical_plotting_r(20e-3/2)

plt.legend()
plt.ylabel('Surface Temperature [$\degree$C]',**font, **size)
plt.xlabel('Time [s]', **font, **size)
plt.savefig(save_path+'B Numerical Solution, dt=2')
plt.show()


#Plotting the Numerical Solution for various dt
dt_array = t_limit / np.array([10,25,175,1750])

plt.figure(figsize=(10,10),dpi=600)

for dt_value in dt_array:
    numerical_plotting(dt_value)
```

```
analytical_plotting_r(20e-3/2)


plt.legend()

plt.ylabel('Surface Temperature [$\degree$C]',**font, **size)

plt.xlabel('Time [s]', **font, **size)

plt.savefig(save_path+'B Various Time Steps')

plt.show()
```