CP1 for NPRE 501

Jin Whan Bae

2017-11-01

Problem Definition

Table 1 lists the constants used in the problem.

Parameter	Value	[Unit]
Diamater	6	[cm]
Radius	**3**	[cm]
Geometry	Sphere	
k	15	$[\frac{W}{mK}]$
Density	8000	$[\frac{kg}{m^3}]$
Specific Heat	500	$[\frac{J}{kgK}]$
α	**3.75e-6**	$[\frac{m^2}{s}]$

Table 1: Problem Constants. Derived constants are in bold.

Differential Equation:

$$\frac{1}{\alpha} \cdot \frac{dT}{dt} = \frac{1}{r^2}\frac{d}{dr}r^2\frac{dT}{dr}$$

Boundary Conditions:

$$T(0,t) = finite \quad OR \quad \frac{dT}{dr}(r=0,t) = 0$$

$$\frac{dT}{dr}(r=R) = 0$$

Initial Condition:

$$T(r,0) = \frac{T_0}{2}(1 - \cos{(\frac{\pi \cdot r}{R})})$$

Analytical Term Study

For the term study, it is speculated that for higher time values, less terms (β_n) would be needed for reasonable convergence, since a higher time value would cause the exponential term, which has β in, less of a contribution. The solved equation from Appendix A proves this mathematically:

$$\overline{T}(r,t) = \sum_{n=1}^{\infty} A_n e^{-\alpha\beta_n^2 t} sin(\beta_n r)$$

The study is conducted where different number of betas are used for each timestep. Seven timesteps are used: $t = 0, 2, 4, 17.7, 31.1, 62.2, 75$. A 'reasonable convergence' is met when

$$\sum_{r=0}^{R} (|r_r^n - r_r^{n+1}|) < 1e - 3$$

Where n is the number of terms used for the temperature profile.

Table 2 organizes the 'reasonable convergence' point of all timesteps.

Time [secs]	Number of Terms for Convergence
0	17
2	7
4	5
17.7	3
31.1	2
62.2	2
75	2

Table 2: Timestep and number of terms for convergence.

As expected, as time increases, the number of terms for convergence becomes minimal. Figure 1, 2, 3, 4, 5, 6, 7 show the convergence study where the temperature profile is plotted with increasing number of terms. Note the visible difference is negligible. Also note the y axis range change for different times.

The converged $T(r,t)$ for all seven cases is shown in fig. 8.

Refer to appendix D for eigenvalue calculations.

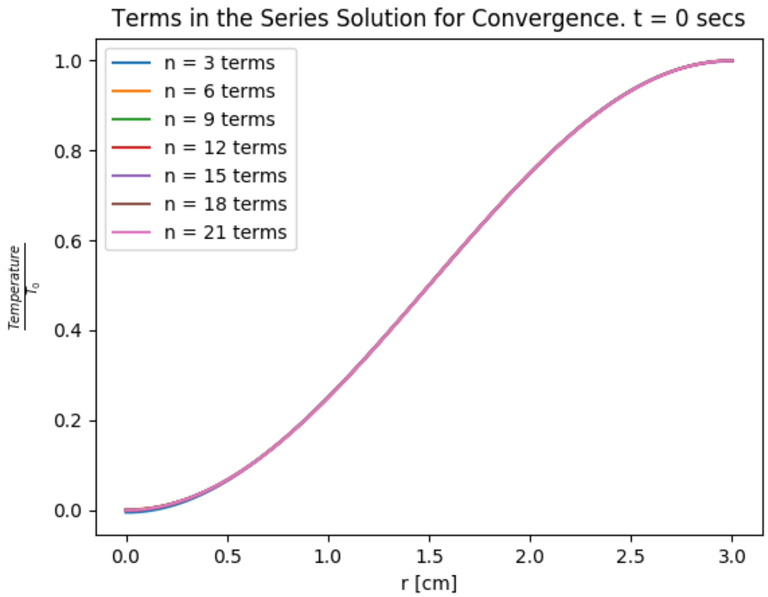

Figure 1: Number of terms convergence study for t = 0.

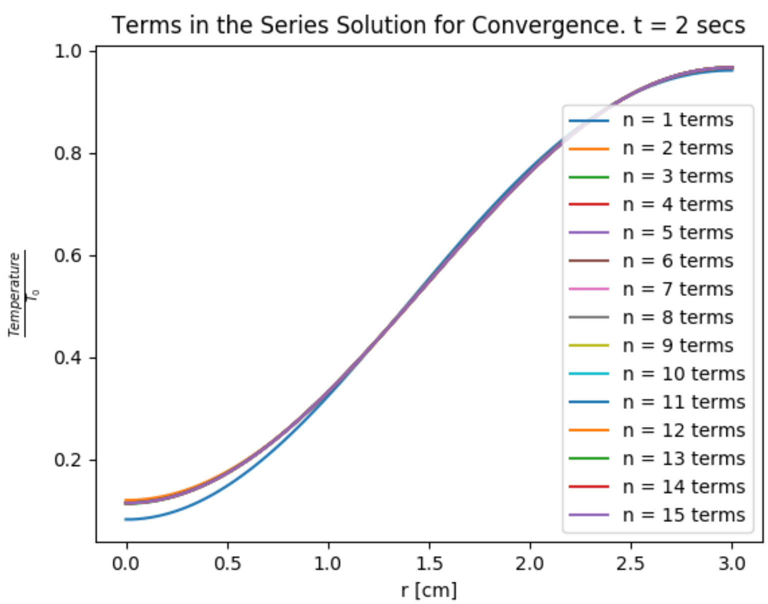

Figure 2: Number of terms convergence study for t = 2.

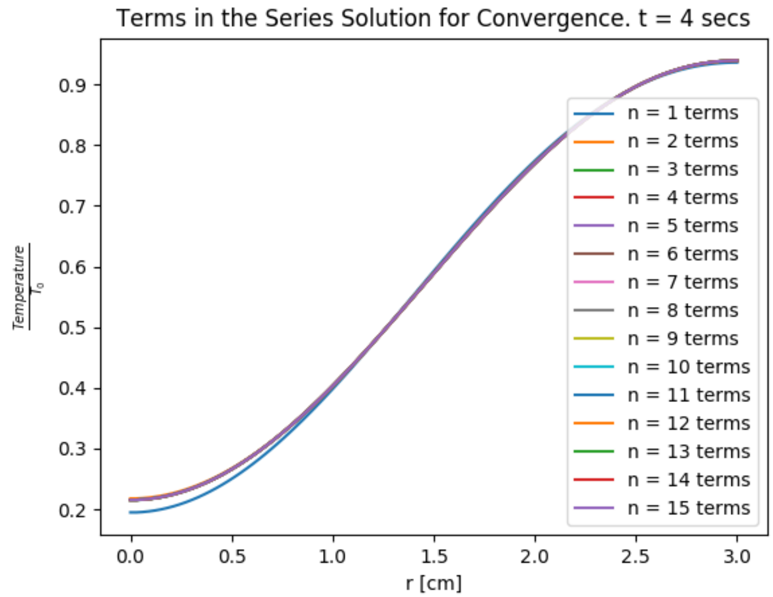

Figure 3: Number of terms convergence study for t = 4.

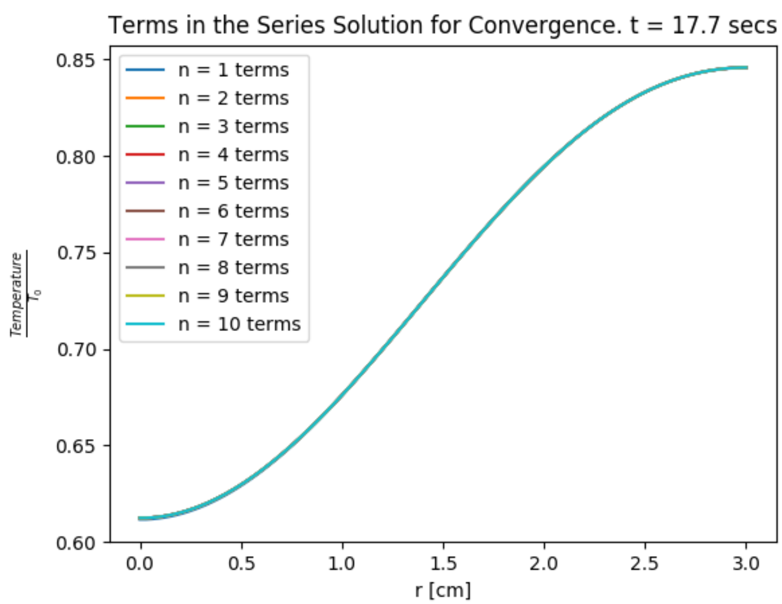

Figure 4: Number of terms convergence study for t = 17.7.

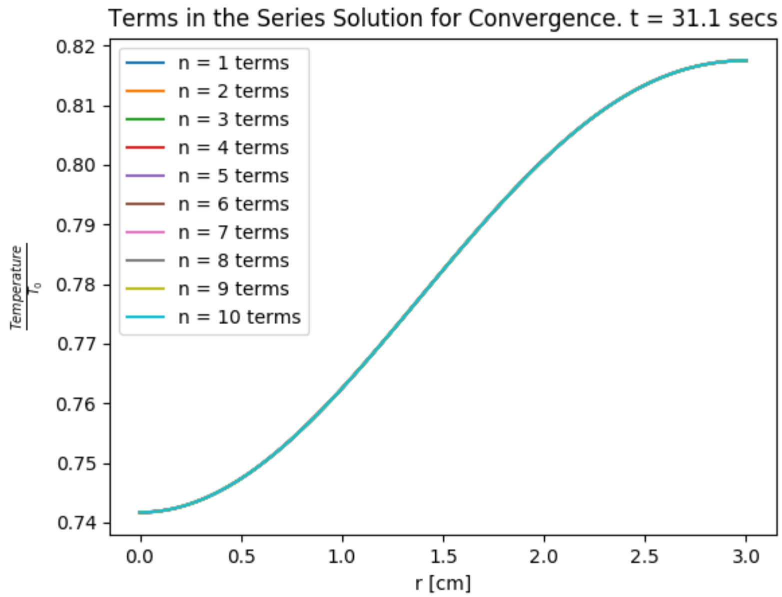

Figure 5: Number of terms convergence study for t = 31.1.

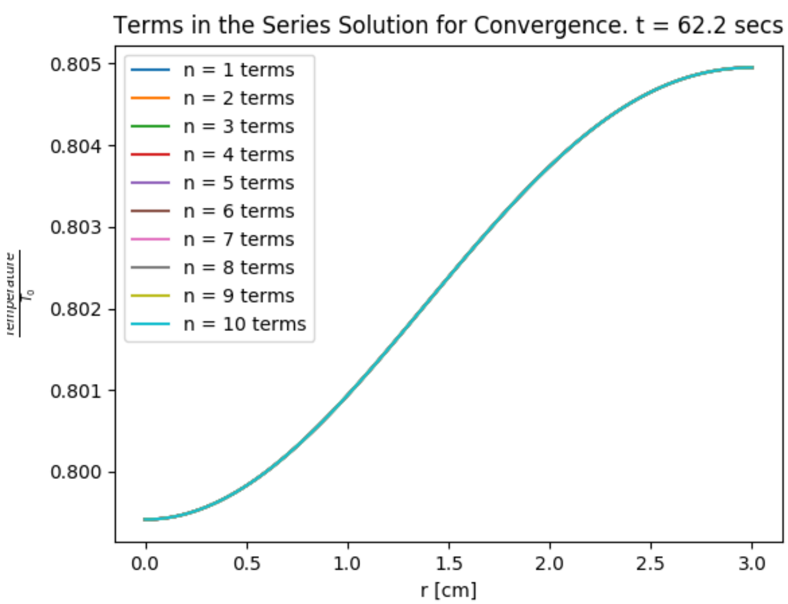

Figure 6: Number of terms convergence study for t = 62.2.

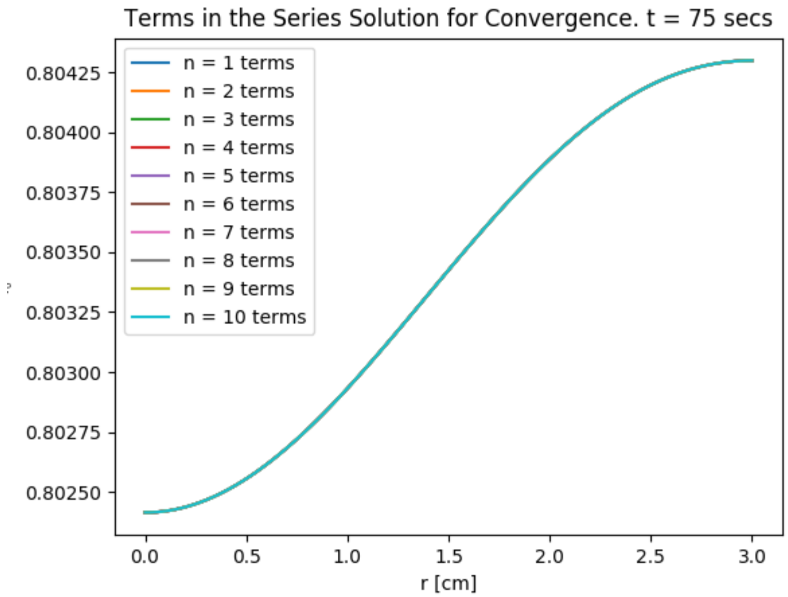

Figure 7: Number of terms convergence study for t = 75.

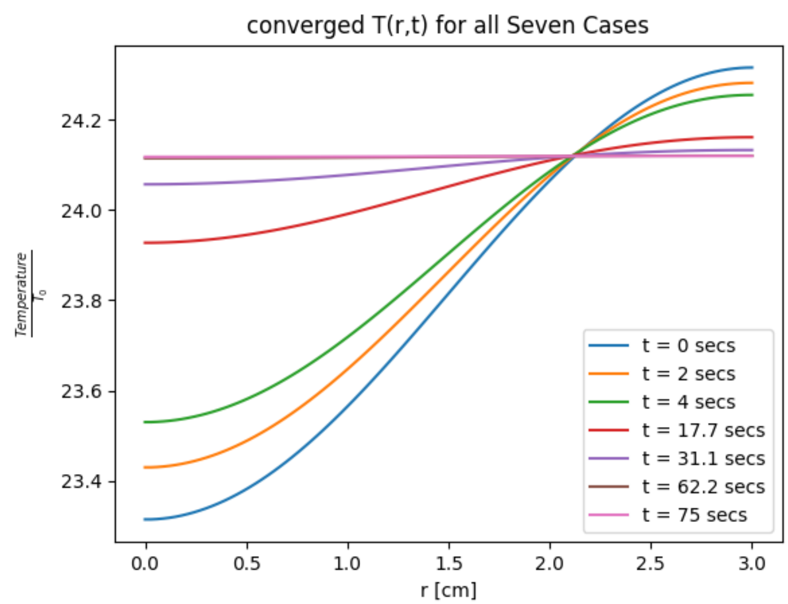

Figure 8: Converged Temperature Profiles for all Seven Timesteps.

Numerical Solution

Considering previous plots, it can be safely assumed that the sphere reaches steady state after 70 seconds. Setting the maximum time to 70 seconds, the grid refinement study is done.

An important thing to note in the numerical solution is the role of $\frac{dt}{dr}$. Varying grid size, thus varying dt and dr may cause this term to either be close to zero or very big, which produces confusing results. The error depends on the ratio of the two terms (eg. the term will go to infinity if dr is much smaller than dt). To prevent this from happening, only one grid size is changed at a time, and the other grid is set so that the $\frac{dt}{dr}$ term does not mess up the entire equation.

For the grid refinement study of spatial terms (r grid), a fixed time is set, and the grid size is varied. For the grid refinement study of temporal terms (t grid), a fixed spatial grid is set and the spatial terms are varied.

Spatial Grid Refinement Study

Figure 9 and 10 show the grid refinement study done on time 10 and 30 seconds.

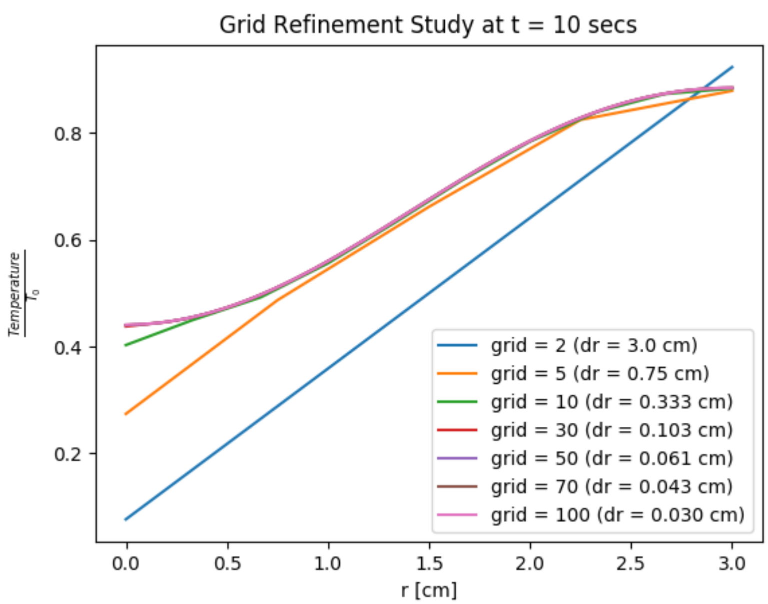

Figure 9: Grid Refinement Study for the temperature at 10 seconds.

With grid size under 5, the temperature profile is far from the converged. However, when the grid size becomes larger than 15 (dr around 0.15cm), the temperature profile converges for both timesteps. After that point, increasing the grid size does not affect the fidelity of the temperature profile much more, which brings the 'Goldilocks' grid size to about 20.

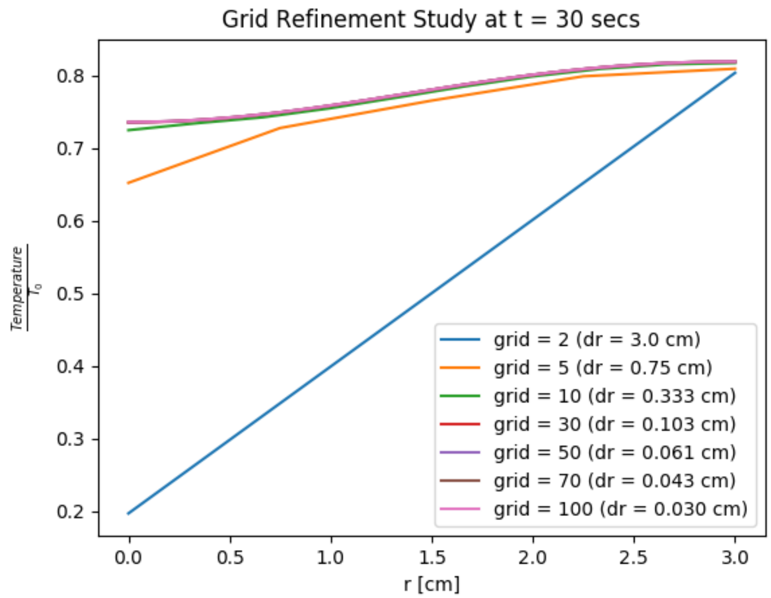

Figure 10: Grid Refinement Study for the temperature at 30 seconds.

Temporal Grid Refinement Study

With the problem with $\frac{dt}{dr}$, the r grid is set as a function of r grid (see code for details) to prevent the equation from blowing up. The time grid points used are 10, 50, 100 and 300, shown in fig. 11, 12, 13, 14.

As the time grid reaches 300 (or $dt = 0.23$), the temperature profiles are a smooth curve, like the analytical solution. Thus, it can be stated that a time grid size of 300 is adequate for obtaining a good solution.

Analytical vs Numerical

For the difference between the analytical and numerical solution, both methods are plotted against each other in a fixed time. The plots are in Figure 15, 16, 17, 18, 19.

As notably so, there is very little difference between the numerical and analytical solution, given ample temporal and spatial grid size.

Conclusion

Through this exercise, a conclusion can be drawn that the numerical and analytical solutions for the temperature profile of the sphere has little difference. A grid refinement study is done both

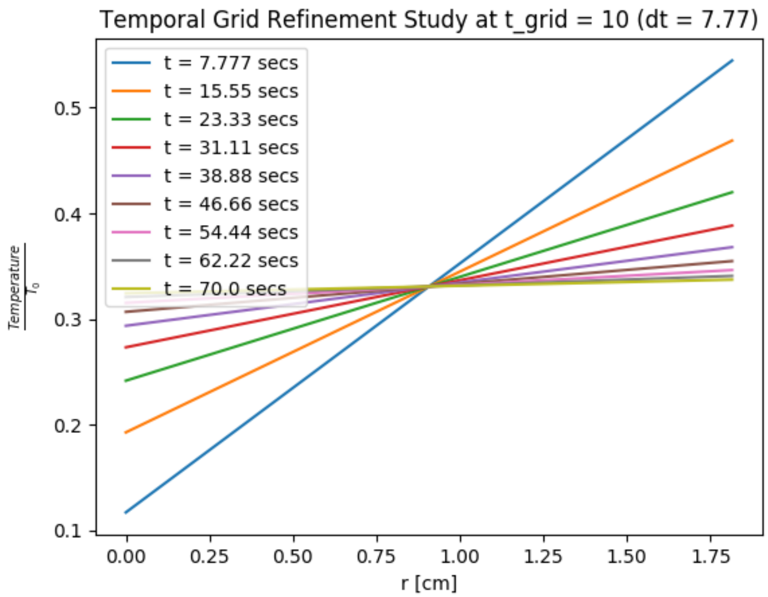

Figure 11: Time Grid Refinement Study for t grid = 10

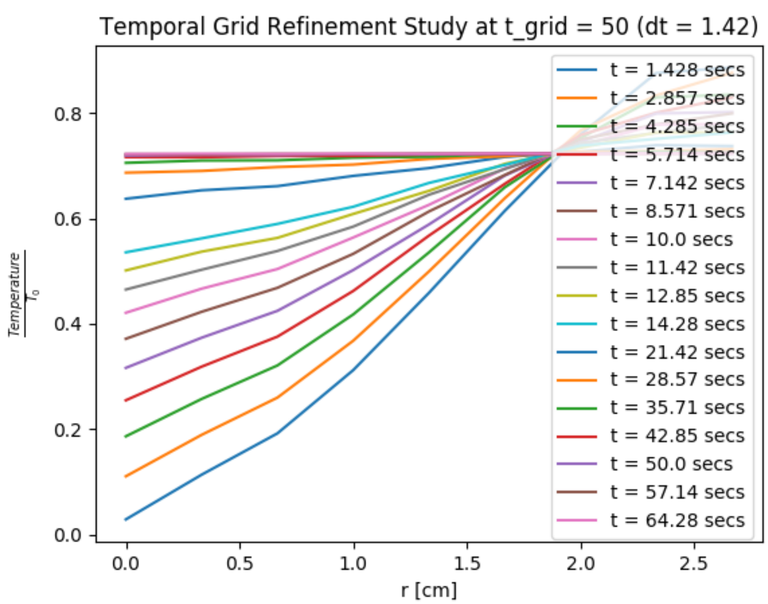

Figure 12: Time Grid Refinement Study for t grid = 50

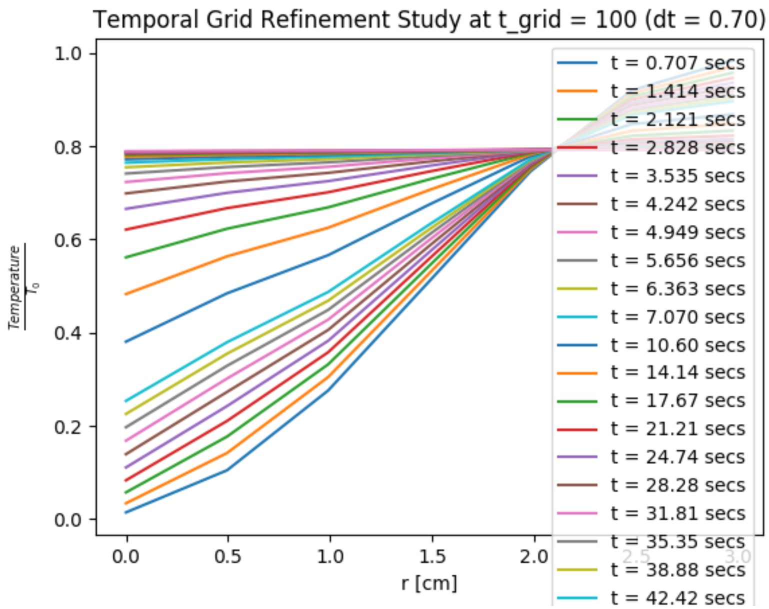

Figure 13: Time Grid Refinement Study for t grid = 100

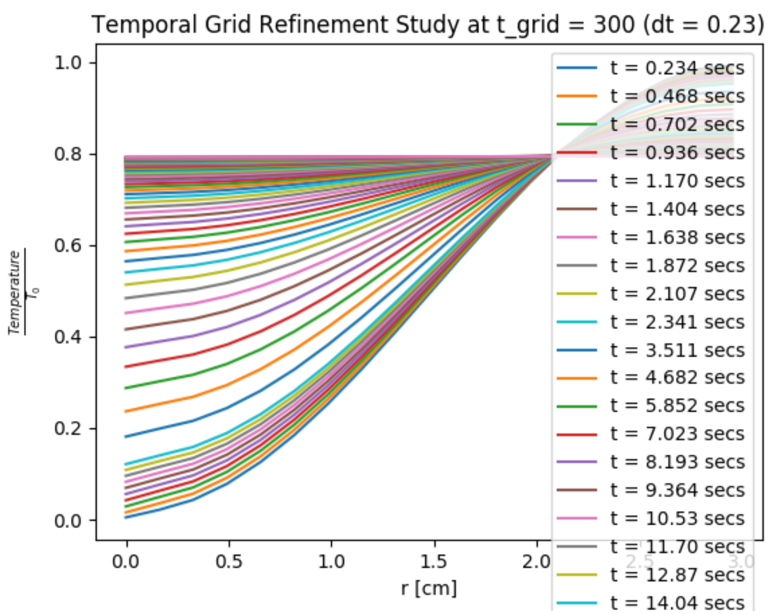

Figure 14: Time Grid Refinement Study for t grid = 300

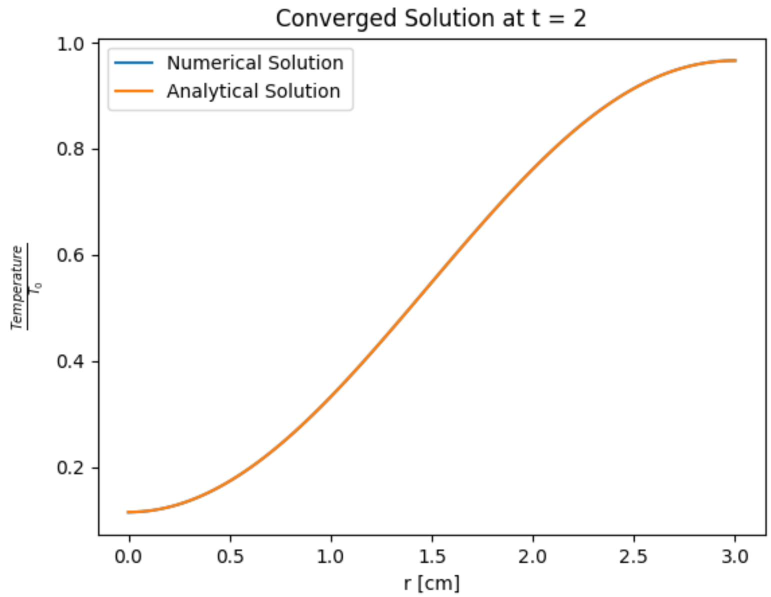

Figure 15: Comparison of Analytical and Numerical Solutions at t = 2s.

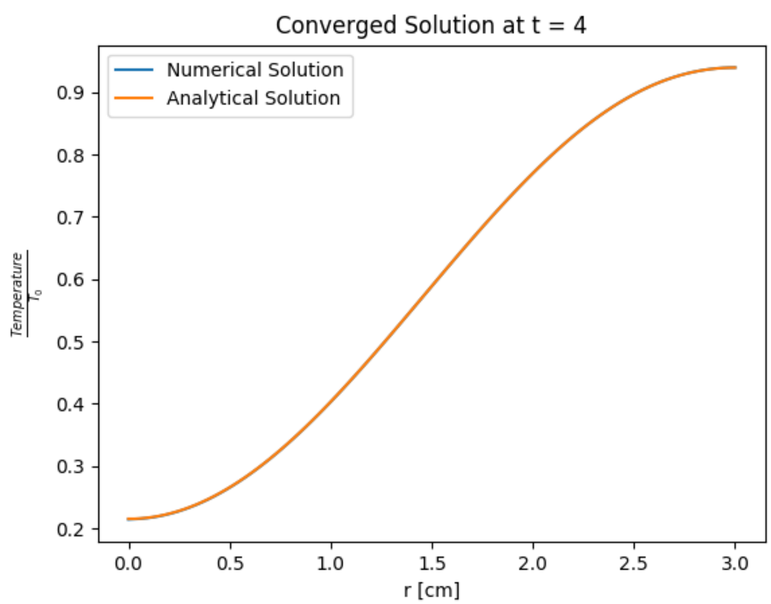

Figure 16: Comparison of Analytical and Numerical Solutions at t = 4s.

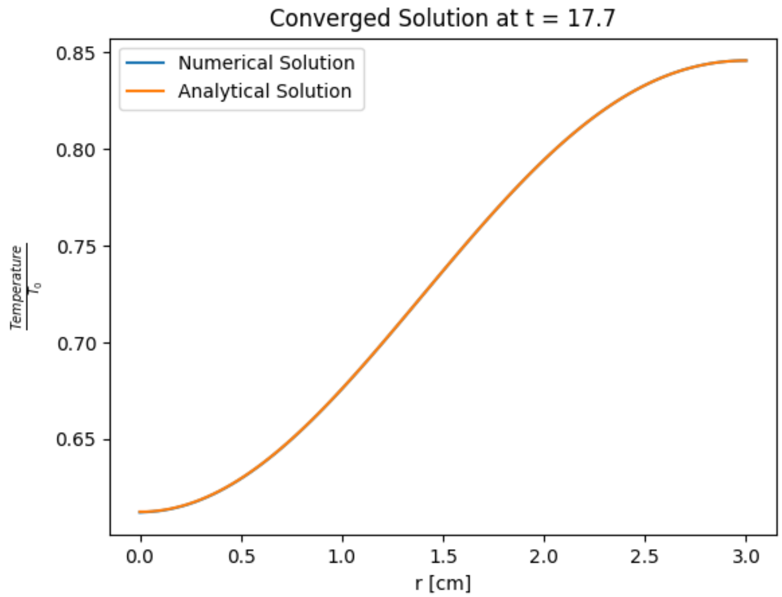

Figure 17: Comparison of Analytical and Numerical Solutions at t = 17.7s.

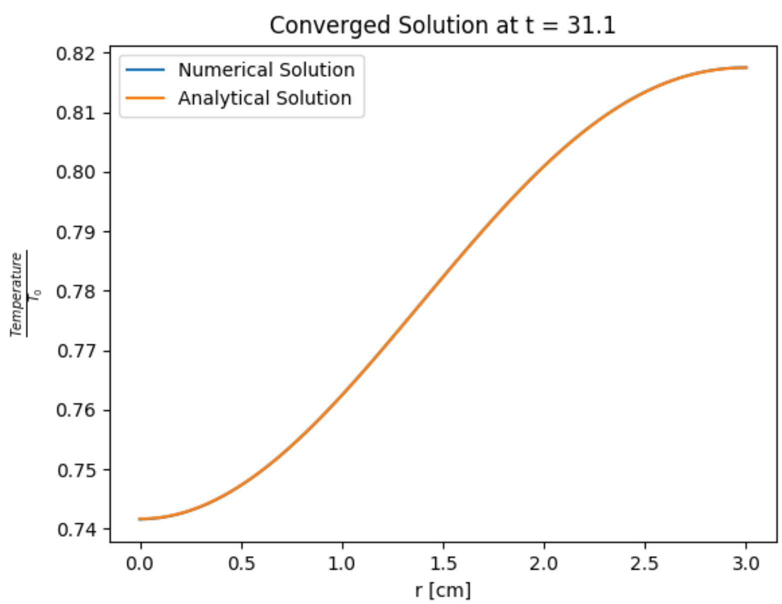

Figure 18: Comparison of Analytical and Numerical Solutions at t = 31.1s.

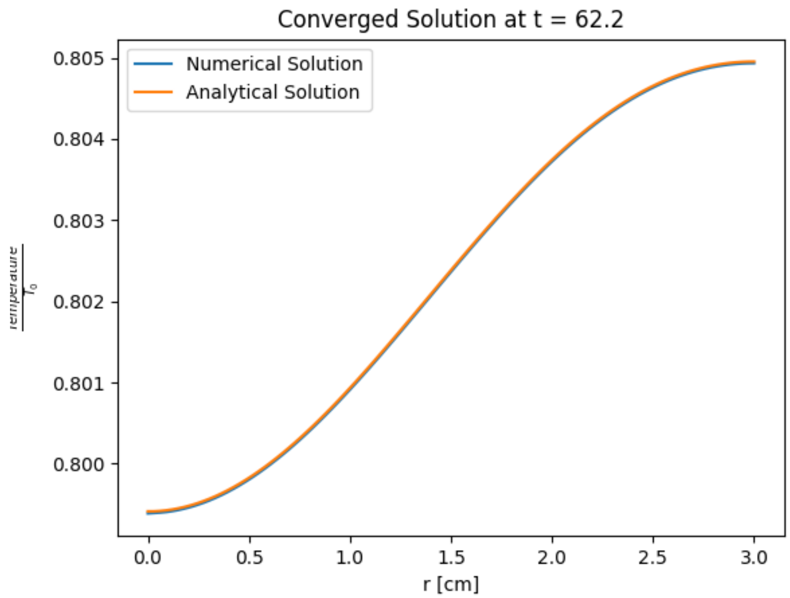

Figure 19: Comparison of Analytical and Numerical Solutions at t = 62.2s.

temporally and spatially, to find a balance between fidelity and efficiency.

Appendix A

Simple Energy Balance

Since there is no heat generation or leakage (r=R is insulated), the steady state temperature is a constant throughout the sphere.

Doing a simple energy balance,

$$\frac{dE}{dt} = E_{gain} - E_{loss}$$

$$\frac{dE}{dt} = E_{gen} + E_{in} - E_{out}$$

Since there is no heat generation, flux in, or flux out (insulated BC), the steady state is a constant temperature in time.

Expressing mathematically for when time goes to infinity:

T_{ss} is when $\frac{dT}{dt} = 0$:

$$0 = \frac{1}{r^2}\frac{d}{dr}r^2(\frac{dT_{ss}}{dr})$$

$$\frac{C_1}{r^2} = \frac{dT_{ss}}{dr}$$

$$-\frac{C_1}{r} + C_2 = T_{ss}$$

applying BC:

at r = 0, finite, makes $C_1 = 0$

$$T_{ss} = C_2$$

the numerical value of this constant can be found doing an energy balance of the system with the initial condition:

$$\int_0^R \frac{T_0}{2}(1 - \cos(\frac{\pi r}{R}))r^2 dr = \int_0^R Cr^2 dr$$

$$\frac{T_0}{2}\int_0^R (r^2 - r^2\cos(\frac{\pi r}{R}))dr = C\frac{R^3}{3}$$

using separation of variables:

$$\frac{T_0}{2}(\frac{R^3}{3} - (\frac{r^2 R}{\pi}\sin(\frac{\pi r}{R})\Big|_0^R - \frac{2R}{\pi}\int_0^R r\sin(\frac{\pi r}{R})))dr = C\frac{R^3}{3}$$

Solving this:

$$\frac{T_0}{2}(\frac{R^3}{3} + \frac{2R^3}{\pi^2}) = C\frac{R^3}{3}$$

$$C = T_{ss} = \frac{3T_0}{2}(\frac{\pi^2+6}{3\pi^2}) \approx 0.804T_0$$

The steady state is plotted in figure fig. 20.

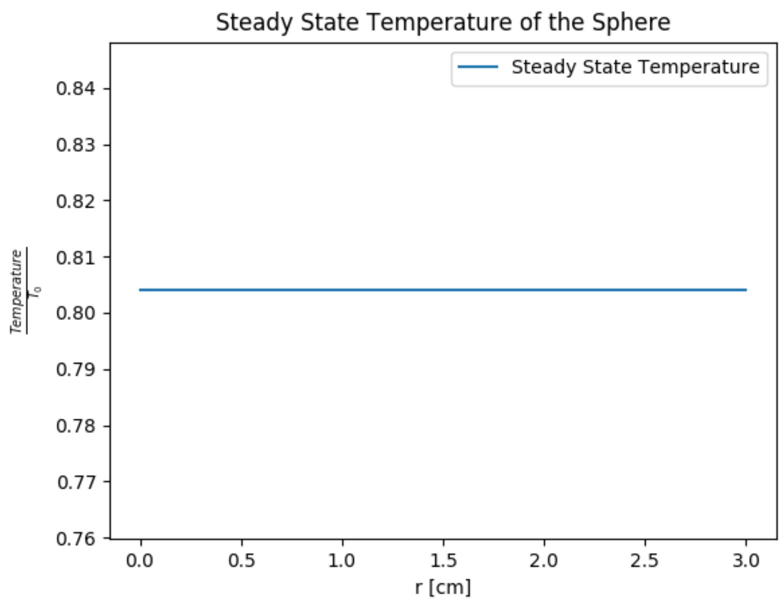

Figure 20: Plot of Steady State Solution for the Sphere.

Sketch of Different Temperature Profiles in Time

The time evolution of the temperature profile starts from t=0, with the two boundary points, $T(0,0) = 0$ and $T(R,0) = T_0$. Through time, the temperature profile flattens out, since the only thing affecting the profile is heat diffusion. The steady state (as time goes to infinity) becomes a constant throughout space. A sketch is made in Figure 21.

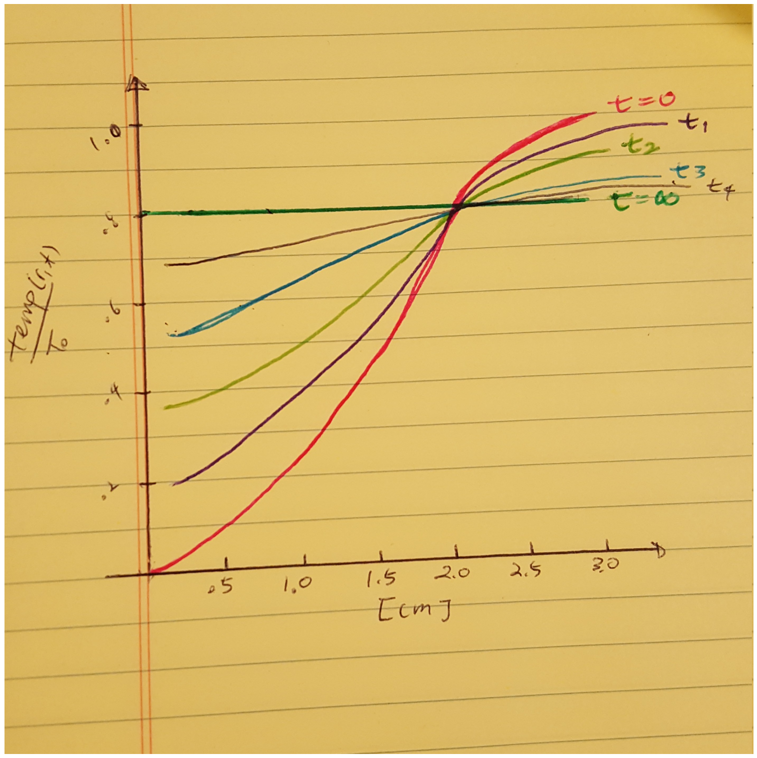

Figure 21: Sketch of Temperature profile over time.

Solution of $T(r,t)$ for $t > 0$

Since the numerical value of T_{ss} has been found through the energy balance, that can be added to the solution for $T(r,t)$ for $t > 0$.

The application and solution for $T(r,t)$ is shown in Appendix B.

Appendix B

Analytical solution for solving T(r,t) directly

$$\frac{1}{\alpha} \cdot \frac{dT}{dt} = \frac{1}{r^2}\frac{d}{dr}r^2\frac{dT}{dr}$$

Boundary Conditions:

$$T(0,t) = finite$$

$$\frac{dT}{dr}(r = R) = 0$$

Initial Condition:

$$T(r,0) = \frac{T_0}{2}(1 - \cos(\frac{\pi \cdot r}{R}))$$

set:

$$T(r,t) = \frac{\overline{T}(r,t)}{r} + T_s s$$

$$\overline{T}(r,t) = T(r,t)r + T_s sr$$

plugging this back into the differential equation:

$$\frac{1}{\alpha} \cdot \frac{d\overline{T}}{dt}\frac{1}{r} = \frac{1}{r^2}\frac{d}{dr}r^2(\frac{d\overline{T}}{dr}\frac{1}{r} - \frac{1}{r^2}\overline{T})$$

$$\frac{1}{\alpha} \cdot \frac{d\overline{T}}{dt} = \frac{1}{r}\frac{d}{dr}(\frac{d\overline{T}}{dr}r - \overline{T})$$

$$\frac{1}{\alpha} \cdot \frac{d\overline{T}}{dt} = \frac{1}{r}(\frac{d^2\overline{T}}{dr^2}r + \frac{d\overline{T}}{dr} - \frac{d\overline{T}}{dr})$$

$$\frac{1}{\alpha} \cdot \frac{d\overline{T}}{dt} = \frac{d^2\overline{T}}{dr^2}$$

where

$$\overline{T}(0,t) = 0$$

$$\frac{d\overline{T}}{dr} = \frac{\overline{T}}{R}$$

and initial condition

$$\overline{T}(r,0) = r\frac{T_0}{2}(1 - cos(\frac{\pi r}{R})) - T_{ss}r$$

This turns into a cartesian problem.
Applying Separation of Variables:

$$\overline{T}(r,t) = \Gamma(t)\Psi(r)$$

Applying the new variables, dividing both sides by $\Gamma(t)\Psi(r)$, and setting it to a new variable $-\beta^2$:

$$\frac{1}{\alpha\Gamma} \cdot \frac{d\Gamma}{dt} = \frac{d^2\Psi}{dr^2}\frac{1}{\Psi} = -\beta^2$$

Solving for Ψ first:

$$\frac{d^2\Psi}{dr^2}\frac{1}{\Psi} = -\beta^2$$

$$\frac{d^2\Psi}{dr^2} + \beta^2\Psi = 0$$

$$\Psi(r) = C_1 sin(\beta r) + C_2 cos(\beta r)$$

Boundary Conditions:

Since:

$$\frac{dT}{dr} = -\frac{\overline{T}}{r^2} + \frac{1}{r}\frac{d\overline{T}}{dr}$$

The boundary conditions become:

$$-\Psi(r=0) = 0$$

$$-k(\frac{\Psi}{R^2} + \frac{d\Psi}{dr}\frac{1}{R}) = 0$$

Applying the first boundary condition, interpreting as $\Psi(r=0) = 0, C_2 = 0$
Applying the second boundary condition,

$$-\frac{C_1 \sin(\beta R)}{R^2} + C_1 \cos(\beta r)\beta = 0$$

$$-\sin(\beta R) + R\beta \cos \beta R = 0$$

$$\tan(\beta R) = R\beta$$

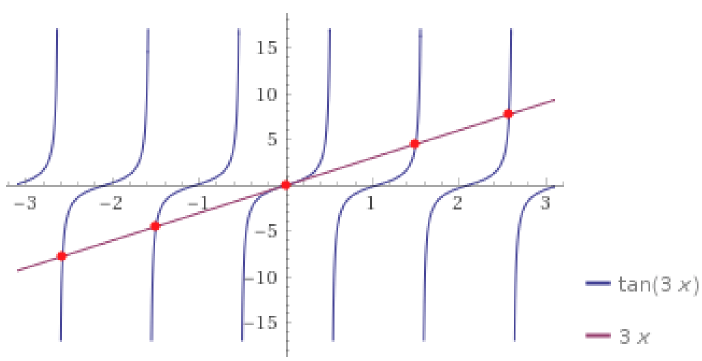

Figure 22: Plot of tan(3x) = 3x.

β is the zeros of that equation (illustrated in fig. 22)
Solving for $\Gamma(t)$:

$$\frac{1}{\alpha\Gamma} \cdot \frac{d\Gamma}{dt} = -\beta_n^2$$

$$\frac{1}{\alpha\Gamma} \cdot \frac{d\Gamma}{dt} = -\beta_n^2 \alpha\Gamma$$

$$\Gamma(t) = A_1 e^{-\alpha\beta_n^2 t}$$

This makes $\overline{T}(r,t)$:

$$\overline{T}(r,t) = \sum_{n=1}^{\infty} A_n e^{-\alpha\beta_n^2 t} sin(\beta_n r)$$

Applying initial condition and orthogonality:

$$T\sum_{n=1}^{\infty} A_n \sin(\beta_n r) = r\frac{T_0}{2}(1-\cos(\frac{\pi r}{R})) - T_{ss} r$$

$$A_n = \frac{\int_0^R (r\frac{T_0}{2}(1-\cos(\frac{\pi r}{R})) - T_{ss} r)\sin(\beta_n r)dr}{\int_0^R (\sin(\beta_n r))^2 dr}$$

Plugging all this into $T(r,t)$:

$$T(r,t) = \sum_{n=1}^{\infty} A_n \frac{\sin(\beta_n r)}{r} e^{-\beta_n^2 \alpha t} + T_{ss}$$

Analytical solution by defining new temperature

$$T'(r,t) = T(r,t) - T_{ss}$$

$$T'(r,t) + T_{ss} = T(r,t)$$

where T_{ss} is:

$$0 = \frac{1}{r^2}\frac{d}{dr} r^2(\frac{dT_{ss}}{dr})$$

$$\frac{C_1}{r^2} = \frac{dT_{ss}}{dr}$$

$$-\frac{C_1}{r} + C_2 = T_{ss}$$

applying BC:
at r = 0, finite, makes $C_1 = 0$

$$T_{ss} = C_2$$

Plugging the new definition into the original differential equation:
differential equation becomes:

$$\frac{1}{\alpha}(\frac{dT'}{dt} + \frac{dT_{ss}}{dt}) = \frac{1}{r^2}\frac{d}{dr}r^2(\frac{dT'}{dr} + \frac{dT_{ss}}{dr})$$

Considering $\frac{dT_{ss}}{dt} = \frac{dT_{ss}}{dr} = 0$,

$$\frac{1}{\alpha} \cdot \frac{dT'}{dt} = \frac{1}{r^2}\frac{d}{dr}r^2 \cdot \frac{dT'}{dr}$$

Boundary Conditions

$$T'(0,t) = finite$$

$$\frac{dT'}{dr}(r = R) = 0$$

Initial Condition:

$$T'(r,0) = \frac{T_0}{2}(1 - \cos{(\frac{\pi \cdot r}{R})}) - C_2$$

set:

$$T'(r,t) = \frac{\overline{T'}}{r}$$

$$\frac{1}{\alpha} \cdot \frac{d\overline{T'}}{dt}\frac{1}{r} = \frac{1}{r^2}\frac{d}{dr}r^2(\frac{d\overline{T'}}{dr}\frac{1}{r} - \frac{1}{r^2}\overline{T'})$$

$$\frac{1}{\alpha} \cdot \frac{d\overline{T'}}{dt} = \frac{1}{r}\frac{d}{dr}(\frac{d\overline{T'}}{dr}r - \overline{T'})$$

$$\frac{1}{\alpha} \cdot \frac{d\overline{T'}}{dt} = \frac{1}{r}(\frac{d^2\overline{T'}}{dr^2}r + \frac{d\overline{T'}}{dr} - \frac{d\overline{T'}}{dr})$$

$$\frac{1}{\alpha} \cdot \frac{d\overline{T'}}{dt} = \frac{d^2\overline{T'}}{dr^2}$$

turns into a cartesian problem.

Applying Separation of Variables:

$$\overline{T'}(r,t) = \Gamma(t)\Psi(r)$$

Boundary Conditions:

$$\Psi(r=0) = finite$$

$$\frac{d\Psi}{dr}(r=R) = 0$$

Applying the new variables, dividing both sides by $\Gamma(t)\Psi(r)$, and setting it to a new variable $-\beta^2$:

$$\frac{1}{\alpha\Gamma} \cdot \frac{d\Gamma}{dt} = \frac{d^2\Psi}{dr^2}\frac{1}{\Psi} = -\beta^2$$

Solving for Ψ first:

$$\frac{d^2\Psi}{dr^2}\frac{1}{\Psi} = -\beta^2$$

$$\frac{d^2\Psi}{dr^2} + \beta^2\Psi = 0$$

$$\Psi(r) = C_1 sin(\beta r) + C_2 cos(\beta r)$$

Applying the first boundary condition, interpreting as $\Psi(r = 0) = 0, C_2 = 0$
Applying the second boundary condition,

$$-\frac{C_1 \sin(\beta R)}{R^2} + C_1 \cos(\beta r)\beta = 0$$

$$-\sin(\beta R) + R\beta \cos \beta R = 0$$

$$\tan(\beta R) = R\beta$$

β is the zeros of that equation.
Solving for $\Gamma(t)$:

$$\frac{1}{\alpha\Gamma} \cdot \frac{d\Gamma}{dt} = -\beta_n^2$$

$$\frac{1}{\alpha\Gamma} \cdot \frac{d\Gamma}{dt} = -\beta_n^2 \alpha\Gamma$$

$$\Gamma(t) = A_1 e^{-\alpha\beta_n^2 t}$$

This makes $\overline{T'}(r, t)$:

$$\overline{T'}(r, t) = A_n e^{-\alpha\beta_n^2 t} sin(\beta_n r)$$

Solving back for $T'(r, t)$:

$$T'(r, t) = \frac{A_n}{r} e^{-\alpha\beta_n^2 t} sin(\beta_n r)$$

Applying initial condition and orthogonality:

$$T'(r, 0) = \frac{A_n}{r} sin(\beta_n r) = \frac{T_0}{2}(1 - \cos(\frac{\pi \cdot r}{R})) + C_2$$

$$A_n = \frac{\int_0^R r^3 sin(\beta_n r)(\frac{T_0}{2} \cdot (1 - \cos\frac{\pi r}{R}) + C_2) dr}{\int_0^R r^2 \sin^2 \beta_n r dr}$$

The analytical solution is:

$$T'(r,t) = \frac{\int_0^R r^3 sin(\beta_n r)(\frac{T_0}{2}\cdot(1-\cos\frac{\pi r}{R}) + T_{ss})dr}{\int_0^R r^2 \sin^2\beta_n r dr} e^{-\alpha\beta_n^2 t} sin(\beta_n r)$$

$$T = \frac{\int_0^R r^3 sin(\beta_n r)(\frac{T_0}{2}\cdot(1-\cos\frac{\pi r}{R}) + T_{ss})dr}{\int_0^R r^2 \sin^2\beta_n r dr} e^{-\alpha\beta_n^2 t} sin(\beta_n r) + T_{ss}$$

Since the solution from the previous question with $n = 0$ covers the steady state case, this problem is redundant and provides no new insight into the problem.

Appendix D - Eigenvalues

The eigenvalues are acquired by a script that assesses zero points from the equation:

```
import numpy as np


def beta_equation(beta):
    y = 3*beta - np.tan(beta * 3)
    return y

R = 3
k = 0.15
rho = 8000. / 1000000.
cp = 500
alpha = k / (rho*cp)
T_0 = 1

points = np.arange(0, 1e2, 1e-5)
pot_betas = []
count = 0
flag = 1
for beta in points:
    if beta_equation(beta) > 0:
        flag = 0
```

```
    elif beta_equation(beta) < 0 and flag == 0:
        pot_betas.append(beta)
        count = count + 1
        flag = 1
betas = pot_betas
```

The acquired eigenvalues are listed in table 3. Only the first 25 terms are shown for brevity.

Term Number (n)	**Eigenvalue** (β_n)
0	0
1	1.49781
2	2.57509
3	3.63471
4	4.68874
5	5.74026
6	6.79044
7	7.83982
8	8.88869
9	9.9372
10	10.98547
11	12.03355
12	13.08148
13	14.12931
14	15.17705
15	16.22472
16	17.27233
17	18.3199
18	19.36742
19	20.41492
20	21.46238
21	22.50982
22	23.55723
23	24.60463
24	25.65201
25	26.69938

Table 3: First 25 eigenvalues acquired.

Appendix D - Numerical Solution

$$\frac{1}{\alpha} \cdot \frac{dT}{dt} = \frac{2}{r}\frac{dT}{dr} + \frac{d^2T}{dr^2}$$

Applying the finite difference method, central for r and explicit for t:

$$\frac{1}{\alpha} \cdot \frac{T_k^u - T_k^{u-1}}{\Delta t} = \frac{2}{r_k}\left(\frac{T_{k+1}^{u-1} - T_{k-1}^{u-1}}{2\Delta r}\right) + \frac{T_{k-1}^{u-1} - 2T_k^{u-1} + T_{k+1}^{u-1}}{\Delta r^2}$$

where u is the temporal step and k is the spacial step.

Solving for T_k^{u+1}:

$$T_k^{u+1} = T_k^u + \alpha\frac{\Delta t}{\Delta r \cdot r}(T_{k+1}^u + T_{k-1}^u) + \alpha * \frac{\Delta t}{(\Delta r)^2}(T_{k+1}^u - 2T_k^u + T_{k-1}^u)$$

Applying neuman boundary condition at r=0:

$$\frac{dT(0,t)}{dr} = 0$$

$$\frac{T_1^u - T_{-1}^u}{2\Delta r} = 0$$

$$T_1^u = T_{-1}^u$$

This gives:

$$T_0^{u+1} = T_0^u + \alpha * \frac{\Delta t}{(\Delta r)^2}(2T_1^u - 2T_0^u)$$

Applying neuman boundary condition at r=R (k=K at r=R):

$$-k\frac{dT(R,t)}{dr} = 0$$

$$\frac{T_{K+1}^u - T_{K-1}^u}{2\Delta r} = 0$$

$$T_{K+1}^u = T_{K-1}^u$$

This gives:

$$T_K^{u+1} = T_K^u + \alpha * \frac{\Delta t}{(\Delta r)^2}(2T_{K-1}^u - 2T_K^u)$$

Numerical Solution Code

```
def numerical (r_grid, t_max):
    # GRID AND STUFF
    R = 3
    k = 0.15
    rho = 8000. / 1000000.
    cp = 500
    alpha = k / (rho*cp)
    T_0 = 1

    t_grid = 1000

    r_list = np.linspace(0, R, r_grid)
    dr = r_list[1] - r_list[0]
    t_list = np.linspace(0, t_max, t_grid)
    dt = t_list[1] - t_list[0]

    if dt/dr > 1e8:
        raise ValueError('Too high of dt/dr dude')

    t = np.zeros((len(r_list), len(t_list)), dtype=float)

    # Apply Initial Condition
    t[:,0] = (T_0 / 2) * (1 - np.cos(np.pi * r_list / R))
    print(t[:,0])

    x = dt/dr

    plt.plot(r_list, t[:,0], label='t = 0 secs')

    for timestep in range(1, len(t_list)):
        t[0,timestep] = t[0,timestep-1] + (alpha * (x/dr) * (2*t[1,timestep-
        t[-1,timestep] = t[-1, timestep-1] + alpha * (x/dr) * (2*t[-2,timest
        for space in range(1, len(r_list)-1):
            first_term = alpha * (x/r_list[space]) * (t[space+1,timestep-1]
            second_term = alpha * (x/dr) * (t[space+1, timestep-1] - (2*t[sp
```

```
            t[space, timestep] = t[space, timestep-1] + first_term + second_

        if timestep%100 ==0:
            plt.plot(r_list, t[:,timestep], label='t = %s secs' %str(t_list

    print(t)
    plt.ylabel(r'$\frac{Temperature}{T_0}$')
    plt.xlabel('r [cm]')
    plt.legend()
    plt.title('Numerical Solution of Time-evolution of Heat Profile')
    plt.savefig('Numerical.png', format='png')
    plt.show()
```

Analytical Solution Code

```
def analytical (r_grid, t_max, t_grid, n):
    R = 3
    k = 0.15
    rho = 8000. / 1000000.
    cp = 500
    alpha = k / (rho*cp)
    T_0 = 1

    #########################
    # finding betas
    points = np.arange(-0.1, 1e3, 1e-4)
    pot_betas = []
    for beta in points:
        if beta_equation(beta) > 0:
            flag = 0
        elif beta_equation(beta) < 0 and flag == 0:
            pot_betas.append(beta)
            flag = 1

    # pick only first n betas
    betas = pot_betas[:n]
```

```
print(betas)
###########################

T_0 = 1
r_list = np.linspace(0, R, r_grid)
dr = r_list[1] - r_list[0]
t_list = np.linspace(0, t_max, t_grid)
dt = t_list[1] - t_list[0]

t = [0] * r_grid
t_compiled_betas = [0] * r_grid
t_tot = (T_0 / 2) * (1 - np.cos(np.pi * r_list / R))
t_ss = 3*T_0 /2 * ((np.pi**2 + 6)/(3*np.pi**2))
print(t_ss)
steady_state = [t_ss] * r_grid
# plot initial condition
plt.plot(r_list, t_tot, label = 't = 0 secs')
plt.plot(r_list, steady_state, label = 'Steady State')

def integrand1(r, b, T_0, R, t_ss):
    return ((r*T_0*0.5*(1-np.cos(np.pi*r / R)) - t_ss * r)*np.sin(b * r
def integrand2(r, b):
    return ((np.sin(b*r))**2)

print(betas)
for time in range(1, len(t_list)):
    t_compiled_betas = [0] * r_grid
    for b in betas[1:]:
        t = [0] * r_grid
        for space in range(1, len(r_list)):
            r = r_list[space]
            integral1 = integrate.quad(integrand1, 0, R, args=(b, T_0,
            integral2 = integrate.quad(integrand2, 0, R, args=(b))
            a = integral1[0] / integral2[0]
            spatial = np.sin(b*r) / r
            temporal = np.exp(-1 * b**2 * alpha * t_list[time])
```

```
            t[space] = a * spatial * temporal
            t[0] = t[1]
        t_compiled_betas = [x+y for x,y in zip(t_compiled_betas, t)]

    t_compiled_betas = [x+t_ss for x in t_compiled_betas]
    plt.plot(r_list, t_compiled_betas, label='t = %s secs' %str(t_list[t
    t_tot = np.vstack((t_tot, t_compiled_betas))

print(t_tot)
plt.ylabel(r'$\frac{Temperature}{T_0}$')
plt.xlabel('r [cm]')
plt.legend()
plt.title('Analytical Solution of Time-evolution of Heat Profile')
plt.savefig('Analytical.png', format='png')
plt.show()
```

-end of report.