Department of Mechanical and Mechatronics Engineering
University of Waterloo

ME 303 – Advanced Engineering Mathematics

PDEs and Heat Equations Analysis

Prepared by:
Ali Muizz
Yasir Ahmed
Vikesh Mistry
Krypton Purnama
Saleem Mohammed Ali

Prepared for:
ME 303 Teaching Team

Submitted on:
Sunday, September 14, 2025

1.0 1D Heat Equation in Spherical Coordinates

The first partial differential equation (PDE) that is analyzed is the generic heat equation where 'u' represents temperature, 't' is time and 'α' is the coefficient of diffusivity.

$$\frac{\partial u}{\partial t} = \alpha \nabla^2 u \tag{1}$$

This section focuses on solving this second-order Laplacian equation numerically to model the boiling of an egg.

1.1 Defining the PDE

To accurately use this PDE, the following assumptions are made about the geometry and material properties of an egg and the cooking process.

1) The egg is a perfect sphere with radius R
2) Assume constant thermal diffusivity of the eggshell, whites and yolk
3) The egg remains completely submerged in constant boiling water

These assumptions make modelling the behaviour less computationally demanding and set simple boundary conditions. First, the right-hand side (RHS) of Equation 1 is expanded using the spherical coordinates $(r, \theta, \emptyset)$.

$$\alpha \nabla^2 u = \alpha \left(\frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial u}{\partial r} \right) + \frac{1}{r^2 \sin\theta} \frac{\partial}{\partial \theta} \left(\sin\theta \frac{\partial u}{\partial \theta} \right) + \frac{1}{r^2 \sin^2\theta} \frac{\partial^2 u}{\partial \phi^2} \right)$$

Since the egg is assumed to be perfectly spherically and completely submerged in water during the cooking process, it transfers heat at the same rate at any given angle. So, $d\theta$ and $d\emptyset$ are equal to 0, creating a purely radial 1D heat equation given below.

$$\frac{\partial u}{\partial t} = \frac{\alpha}{r^2} \frac{\partial}{\partial r} \left(\mathrm{r}^2 \frac{\partial u}{\partial r} \right)$$

Product and Chain Rule:

$$\frac{\partial u}{\partial t} = \frac{\alpha}{r^2} \frac{\partial}{\partial r} \left(\mathrm{r}^2 \frac{\partial u}{\partial r} \right) = \frac{\alpha}{r^2} \left(2r \frac{\partial u}{\partial r} + r^2 \frac{\partial^2 u}{\partial r^2} \right) = \alpha \left(\frac{2}{r} \frac{\partial u}{\partial r} + \frac{\partial^2 u}{\partial r^2} \right)$$

$$\frac{\partial u}{\partial t} = \alpha\left(\frac{2}{r}\frac{\partial u}{\partial r} + \frac{\partial^2 u}{\partial r^2}\right) \tag{2}$$

Equation 2 gives the rate of heating or cooling based as a function of radius. This was solved through numerical approximation as seen in the next section.

1.2 Solving the PDE

Equation 2 is then numerically solved using finite differences in the work below. Note, u_i^k notation will be used where 'k' represents a single time frame, and 'i' is a point along the radius. For example, u_R^0 is the initial condition before the egg is placed in the boiling point.

Euler's Method:

$$\frac{\partial u}{\partial t} = \frac{u_i^{k+1} - u_i^k}{\Delta t}$$

Central Difference:

$$\frac{\partial u}{\partial r} = \frac{u_{i+1}^k - u_{i-1}^k}{2\Delta r}$$

$$\frac{\partial^2 u}{\partial r^2} = \frac{u_{i+1}^k - 2u_i^k + u_{i-1}^k}{(\Delta r)^2}$$

Substitute into Equation 2:

$$\frac{u_i^{k+1} - u_i^k}{\Delta t} = \alpha\left(\frac{2}{r}\frac{u_{i+1}^k - u_{i-1}^k}{2\Delta r} + \frac{u_{i+1}^k - 2u_i^k + u_{i-1}^k}{(\Delta r)^2}\right)$$

$$u_i^{k+1} = u_i^k + \frac{\alpha\Delta t}{(\Delta r)^2}\left(u_{i+1}^k - 2u_i^k + u_{i-1}^k\right) + \frac{\alpha\Delta t}{r\Delta r}\left(u_{i+1}^k - u_{i-1}^k\right) \tag{3}$$

Stability Condition:

$$\left(1 - \frac{2\alpha\Delta t}{(\Delta r)^2}\right)u_i^k \rightarrow \frac{2\alpha\Delta t}{(\Delta r)^2} < 1 \rightarrow \Delta t < \frac{(\Delta r)^2}{2\alpha} \tag{4}$$

With this equation, the boundary conditions (BC) are defined as followed. The first Dirichlet boundary condition follows the third assumption, setting boiling water equal to 100°C.

$$u(R, t) = 100$$

The following Neumann BC is set because at the centre of the egg there is no direction of heat flow. This ensures that heat is not flowing across the centre.

$$\frac{\partial u}{\partial t}(0,t) = 0$$

The initial conditions selected for the model is based on the average temperature of a fridge, assuming the egg has a constant temperature throughout (4°C).

$$IC: u(r,0) = 4$$

These assumptions imply that the water will remain at a constant 100°C after placing the egg in, that the heat transfer is perfectly linear, and that the entire egg has a constant temperature of 4°C throughout right before being boiled, respectively.

1.3 Finding Stable Condition

With these conditions and the derived solution, the model was developed in MATLAB to graph the result, and the code can be found in Appendix A**.** The values used in the code were calculated/researched as shown below. To find radial steps and time steps that fit the stability condition, a value of $\Delta r = 0.1$mm was chosen based on the smallest egg's radius being around 15mm. Choosing a radius step of 0.1mm will plot a minimum of 150 points which was determined to be valid for this solution. Note, the Δr was not reduced as the time step would exceed the computational power available for running the simulations as seen below.

$$\Delta r = 0.1mm = 0.0001m$$

$$\frac{(\Delta r)^2}{2\alpha} = \frac{(0.0001)^2}{2(1.5x10^{-7})} = \frac{1}{30} \approx 0.03, \qquad \Delta t = 0.01$$

Or, selecting a smaller Δr,

$$\Delta r = 0.01mm = 0.00001m$$

$$\frac{(\Delta r)^2}{2\alpha} = \frac{(0.00001)^2}{2(1.5x10^{-7})} = \frac{1}{3000} \approx 0.0003, \qquad \Delta t = 0.0001$$

$$N_t = \frac{1s}{1x10^{-5}s} = 1x10^5\ time\ steps/s$$

To approximate the eggs temperature every second, it would take 10^5 steps. From common knowledge, an egg should take over 5mins to cook, so $\Delta r = 0.01mm$ is very computationally expensive. Thus, instead, a time step of 0.01s will be used. Next the coefficient of diffusivity of an egg was researched to be around, $\alpha = 1.5x10^{-7}m^2/s$ [1]. Based off measurements the average radius of a quail, chicken and ostrich egg was found to be 1.5cm, 2.2cm, and 6.5cm [1]. The amount of estimated temperature points along the radius of each egg 'N' was calculated in the code using the formula above.

$$N = round\left(\frac{R}{dr}\right)$$

1.4 Results and Discussion

This section will provide the simulations demonstrating boiling an egg based on the solution and parameters derived in previous sections. A criterion to determine when the egg is fully cooked is defined as follows:

1) The entire egg must be heated to 80°C for $t \geq 10s$, $u(r,t) \geq 80℃$

With the preliminary setup complete, the 3D heating plot of the egg is shown in Figure 1. Furthermore, Figure 2 shows that a larger radius increases the time to heat the center of each at $(r = 0)$.

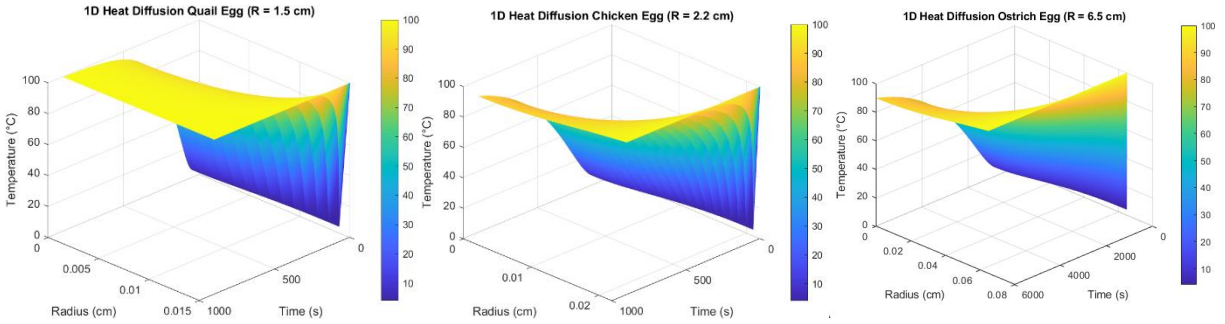


Figure 1: 3D model of the 1D heating of an egg

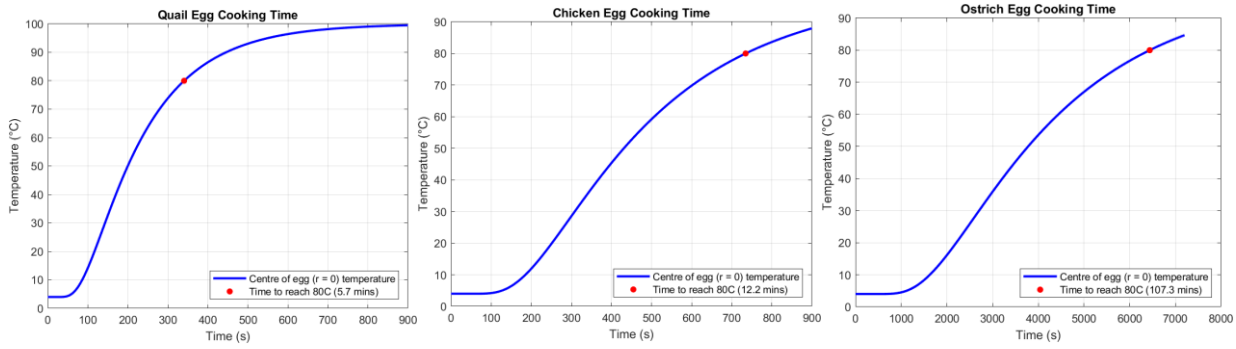


Figure 2: Time to reach 80°C internal temperature

The centre temperature of the egg was plotted against time to determine when it reached 80°C. Based on the previous conditions to cook an egg (add 10 seconds to each of the times listed on the plot), the time to cook each egg is seen in Table 1. To test the accuracy of this model, a chicken egg will be cooked, recording the internal temperature after around 12 minutes. This will be completed in the following section.

Table 1: Cooking time data

Type of Egg	Time to Cook
Quail	5mins 52sec
Chicken	12mins 22sec
Ostrich	107mins 28sec

1.5 Live Experiment

1.5.1 Results

Three chicken eggs with an average diameter of 4.5cm were placed into boiling water directly from the fridge. The eggs were removed at different times: 10-minutes, 12-minutes and 14-minutes. The resulting egg from each experiment is seen in Figure 3. Note, the internal temperature was also recorded after time boiled. Additionally, before the eggs were cut, the internal temperature was recorded for each egg and is listed in Table 2.

Figure 3: The cross-sections of the egg after boiling (from left to right: 10 mins, 12 mins and 14 mins)

Table 2: Experimental data

Time Cooked (minutes)	Internal Temperature (°C)
10	55
12	60
14	65

For the 10-minute boiling experiment, the egg appears to be soft to medium boiled. The egg whites are completely set, and the center of the yolk is runny. The internal temperature was 55°C, not meeting the requirements for a cooked egg.

For the 12-minute boiling experiment, the egg appears to be medium to hard boiled. The center yolk has a darker yellow, undercooked section. It only reached an internal temperature of 60°C, again, not meeting the criterion for a safe to eat egg.

For the 14-minute boiling experiment, the egg appears to be a fully cooked hard-boiled egg with even yolk colouring and fully set whites. The core only reached 65°C which is considered undercooked based on the model. The reasoning for the differences between the live experiment and the derived approximation is discussed in the following section.

1.5.2 Deviation Between Theory and Experiments

There are three main reasons that explain the deviation between the live experiments and the theoretical approximation.

1) The assumption that water temperature remains constant around the egg

The water was brought to a boil and then the three eggs were added simultaneously. Since the eggs were in the fridge for more then 24 hours, they had a thorough temperature of around 4°C. This would drop the water temperature, increasing the time required to reach an internal temperature of 80°C. This can explain why throughout the experiments, none of the eggs reach the target internal temperature. The mathematical model does not account for this as the boundary condition is $u(R,t) = 100$, assuming constant water temperature. This can be improved by making a modular boundary condition that changes based on time and the water temperature. However, this would increase the complexity of the PDE problem and require more processing power.

2) The assumption that the geometry of the eggs are uniform spheres

The mathematical model assumes that the eggs are perfectly spherical with an exact radius R. This is not true in the real world as eggs have an elliptical shape and variable radius. This changes the time that is required to heat the centre of the egg, and accounts for discrepancies between the theory and experiments.

3) Uncalibrated testing equipment

The thermometer used was a typical food-grade thermometer without any prior calibration so the measurements may not be accurate. Before adding the eggs, the thermometer read the boiling water temperature to be 97°C, so it is expected to have around -3°C tolerance.

4) Model simplifications

There were many simplifications in the model that do not accurately describe the boiling of an egg. First, heat diffusivity of the eggshell, egg whites, and yolk were all assumed to be the same, which is not true. Next, there are cases of heat impulses into the system during cooking, such as the heat transfer from the pot to the egg if it touches the bottom of the pot. This is not accounted for with the current model.

In conclusion, the current model underestimates the time needed to hard-boil an egg. This is mainly due to the assumption that there is a constant water temperature of 100°C. After 12 minutes, the egg was still undercooked, reaching only 60°C internally. A model accounting for the factors listed above would provide a closer result to theoretical expectations.

1.6 Future Improvements

As discovered, boiling an egg is a time-consuming process, and one that takes longer than theoretically expected. For this reason, this section explores more economical ways in terms of energy consumption to cook eggs.

In boiling an egg, the medium used to transfer heat from the source to the egg was water. Water can only hold so much heat before it changes form into vapor. To overcome this heat capacity issue, direct conduction can be used instead. Cooking the egg in an enclosed highly conductive metallic container allows the egg to cook without water as a medium. This allows more heat to be transferred to the egg in a shorter period and avoids the formation of water vapor. This reduces the amount of energy and time needed to cook the egg to the desired internal temperature. This can be reflected within the simulation by modifying the boundary condition used from a constant temperature to a convection equation. This allows a more realistic and optimized heat transfer scenario that can reflect the energy savings.

Another method to save energy is with a microwave. A microwave delivers energy directly to the egg using electromagnetic energy to vibrate the water molecules within the egg itself to generate heat. Depending on the efficiency of the microwave, this reduces the total amount of work required as an entire pot of water is not required to boil, rather just the water within the egg is heated. However, with the buildup of steam and pressure within the egg, there is a large chance of the egg exploding. So, although this is potentially a more efficient way of cooking an egg, is not the safest.

The third method of reducing the energy consumption is by boiling water in a kettle, then using low heat to simmer the eggs for a longer time. A kettle is more efficient for boiling water requiring around 1500W of energy and taking around 5 minutes to boil 1 pot of water (1.7L) [2]. While a coil stove requires around 1000W on high power and around 10 minutes to boil a pot of water [3]. Once the water is boiled in the kettle the pot can be placed on the stovetop burner on a lower setting to maintain the water temperature when placing the eggs in. The combination of the kettle and the lowered stovetop heat settings will give the same boiled egg for less energy. However, the downside of this idea is that it will take more time to cook.

By utilizing the above techniques, direct conduction, a microwave or a kettle to boil the water, are all ways to cook an egg in a more energy efficient manner.

2.0 Cartesian 1D Heat Equation

The following PDE was solved numerically and analytically using the conditions below.

PDE

$$u_t = 2u_{xx}, \qquad x \epsilon (0,1), \qquad t \epsilon (0, \infty)$$

BC

$$u(x = 0, t) = 0, \qquad u(x = 1, t) = 2, \qquad t \epsilon [0, \infty)$$

IC

$$u(x, t = 0) = \cos(\pi x), \qquad x \epsilon (0,1)$$

2.1 Numerical Solution

Starting with a numerical approach, Euler's Method and the Central Difference formula was used to approximate the PDE with a first order of accuracy.

Euler's Method:

$$\frac{\partial u}{\partial t} = \frac{u_i^{k+1} - u_i^k}{\Delta t}$$

Central Difference:

$$\frac{\partial^2 u}{\partial x^2} = \frac{u_{i+1}^k - 2u_i^k + u_{i-1}^k}{(\Delta x)^2}$$

Replace in PDE:

$$\frac{u_i^{k+1} - u_i^k}{\Delta t} = 2\left(\frac{u_{i+1}^k - 2u_i^k + u_{i-1}^k}{(\Delta x)^2}\right)$$

$$u_i^{k+1} = u_i^k + \frac{2\Delta t}{(\Delta x)^2}\left(u_{i+1}^k - 2u_i^k + u_{i-1}^k\right) \tag{5}$$

Stability Condition:

$$u_i^k - \frac{2\Delta t}{(\Delta x)^2} 2u_i^k = \left(1 - \frac{4\Delta t}{(\Delta x)^2}\right) u_i^k, \qquad \frac{4\Delta t}{(\Delta x)^2} < 1 \rightarrow \Delta t < \frac{(\Delta x)^2}{4}$$

With the numerical approximation for this PDE given as Equation 5, Δx was selected to solve for the Δt that met the stability condition of this approximation and computational power

available. Note, the choice of grid spacing (Δx) is explained in Section 2.4. The results are plotted and compared to the analytical solution in the following sections.

$$\Delta x = 0.01m$$

$$\Delta t < \frac{(0.01)^2}{4} = 2.5x10^{-5}, \qquad \Delta t = 2x10^{-5}s$$

2.2 Analytical Solution

Using an analytical approach, the same PDE as the previous question was solved. The hand calculations, which the solution below follows, can be found in Appendix B.

First, considering the inhomogeneity of BC at x = 1, the term $v(x,t)$ and $\varphi(x,t)$ are introduced where $u(x,t) = v(x,t) + \varphi(x,t)$. This form displays the inhomogeneous term u as a summation of a homogeneous term v and a shift φ allowing to solve for v using separation of variables.

Utilizing the given u BCs and the 0 BC values of v, due to its homogeneity, values of φ can be found at x = 0 and x = 1.

$$u(x = 0, t) = v(x = 0, t) + \varphi(x = 0, t)$$

$$0 = 0 + \varphi(x = 0, t)$$

$$\varphi(x = 0, t) = 0$$

Using the same approach,

$$\varphi(x = 1, t) = 2$$

From the values of φ at x=0 and x=1, a slope and therefore a linear function for φ is derived:

$$\varphi(x) = 2x$$

Therefore,

$$u(x,t) = v(x,t) + 2x$$

Taking the first derivative with respect to time to achieve the LHS of the PDE and taking the second derivative with respect to x to achieve the u component in the RHS of the PDE respectively yields:

$$u_t(x,t) = v_t(x,t)$$
$$u_{xx}(x,t) = v_{xx}(x,t)$$

Therefore, the PDE can be written as:

$$v_t(x,t) = 2v_{xx}(x,t)$$

Furthermore, since the IC for u is given as:

$$u(x, t = 0) = \cos(\pi x)$$

THE IC for v can be calculated as:

$$v(x, t = 0) = \cos(\pi x) - 2x$$

Thus, the PDE, two Dirichlet BCs, and IC of homogeneous v was acquired, and can be solved using separation of variables.

$$V_t = 2\,V_{xx}$$

BC:

$$V(0,t) = 0, V(1,t) = 0$$

IC:

$$V(x, t = 0) = u(x, t = 0) - \phi(x) = \cos(\pi x) - 2x$$

Separating V into X(x) and T(t):

$$V(x,t) = X(x)T(t) = XT$$

Plugging into the PDE:

$$XT' = 2X''T$$

Separating X(x) and its derivatives from T(t) and its derivatives and equating to a constant k:

$$\frac{X''}{X} = \frac{T'}{\alpha^2 T} = k$$

$$(i), \quad X'' - kx = 0$$
$$(ii), \quad T' - k\alpha^2 T = 0$$

Calculating BCs of X(x) at x = 0 and x = 1 knowing BCs of $V(x,t)$:

$$V = XT$$

BC 1:

$$V(x = 0, t) = 0 = X(0)T(t) \rightarrow X(0) = 0$$

BC 2:

$$V(x = 1, t) = 0 = X(1)T(t) \rightarrow X(1) = 0$$

Considering (i) when; k= 0, k $= \mu^2 > 0$, and $k < \mu^2 0$,

At $k = 0$

$$X'' = 0$$
$$X = ax + b$$

Applying the two BCs for X(x), a trivial solution is computed:

$$0 = a(0) + b \rightarrow b = 0$$
$$0 = a(1) + 0 \rightarrow a = 0$$

At $k = \mu^2 > 0$

$$X'' - \mu^2 X = 0$$
$$X = Ae^{-\mu x} + Be^{\mu x}$$

$$0 = A + B$$
$$0 = Ae^{-\mu} + Be^{\mu}$$

Substituting the first equation into second,

$$B(-e^{-\mu} + e^{\mu}) = 0$$

$$B = 0, A = 0$$

therefore, a trivial solution is achieved.

At $k = -\mu^2 < 0$

$$X'' + \mu^2 X = 0$$

$$X = A\cos(\mu x) + B\sin(\mu x)$$

$$0 = A\cos(0) + B\sin(0) \rightarrow A = 0$$
$$0 = A\cos(\mu) + B\sin(\mu) \rightarrow B\sin(\mu) = 0 \rightarrow \sin(\mu) = 0$$

Therefore,

$$\mu = n\pi, \qquad n = 1,2,3 \ldots$$
$$k_n = -(n\pi)^2, \qquad n = 1,2,3 \ldots$$
$$X_n(x) = B\sin(n\pi x) = \sin(n\pi x)$$

Note that constant B is omitted as it will later be combined with constant Cn found from equation (ii). Plugging k_n into equation (ii):

$$T' - 2kT = 0$$
$$T' + 2(n\pi)^2 T = 0$$

$$T_n(t) = C_n e^{-2(n\pi)^2 t}$$

$$V(x,t) = \sum_{n=1}^{\infty} V_n(x,t) = \sum_{n=1}^{\infty} X_n(x)T_n(t) = \sum_{n=1}^{\infty} \sin(n\pi x)\left(C_n e^{-2(n\pi)^2 t}\right)$$

Considering the IC for V:

$$V(x, t=0) = \cos(\pi x) - 2x$$

$$\sum_{n=1}^{\infty} \sin(n\pi x)\,(C_n) = \cos(\pi x) - 2x$$

Since V is only a summation of sine terms, the Fourier formula, $c_n = \frac{2}{L}\int_0^L f(x)\sin\left(\frac{n\pi x}{L}\right)dx$ can be used to calculate the coefficients C_n

$$C_n = \frac{2}{1}\int_0^1 (\cos(\pi x) - 2x)\sin\left(n\frac{\pi}{1}k\right)dx$$
$$C_n = 2\int_0^1 (\cos(\pi x) - 2x)\sin(n\pi k)\,dx$$
$$C_n = 2\left[\int_0^1 (\cos(\pi x)\sin(n\pi x))dx - \int_0^1 2x\sin(n\pi x)\,dx\right]$$

Left Integral $I_1 = \int_0^1 (\cos(\pi x)\sin(n\pi x))dx$

Since $\cos A \sin B = \frac{1}{2}[\sin(A+B) - \sin(A-B)]$

$$\cos(\pi x)\sin(n\pi x) = \frac{1}{2}\left[\sin\big((n+1)\pi x\big) - \sin\big((1-n)\pi x\big)\right]$$

Plugging into left integral I_1

$$\frac{1}{2}\left[\int_0^1 \sin\big((n+1)\pi x\big)\,dx - \int_0^1 \sin\big((1-n)\pi x\big)\,dx\right]$$

$$\frac{1}{2}\left[\frac{-\cos\big((n+1)\pi x\big)}{(n+1)\pi}\bigg|_0^1 - \frac{-\cos((1-n)\,\pi x)}{(1-n)\pi}\bigg|_0^1\right]$$

$$\frac{1}{2}\left[\frac{-\cos\big((n+1)\pi\big)}{(n+1)\pi} - \left(-\frac{\cos(0)}{(n+1)\pi}\right) - \left[\frac{-\cos((1-n)\,\pi)}{(1-n)\pi} - \frac{-\cos(0)}{(1-n)\pi}\right]\right]$$

$$\frac{1}{2}\left[\frac{-\cos((n+1)\pi)+1}{(n+1)\pi} - \left(\frac{-\cos\big((1-n)\pi\big)+1}{(1-n)\pi}\right)\right]$$

Since $\cos(k\pi) = (-1)^k$

$$I_1 = \frac{1}{2}\left[\frac{-(-1)^{n+1}+1}{(n+1)\pi} - \left(\frac{-(-1)^{1-n}+1}{(1-n)\pi}\right)\right]$$

Simplifying I_1,

$$I_1 = \frac{1}{2\pi}\left[\frac{(-1)^n+1}{n^2-1}\right]$$

Right Integral $I_2 = \int_0^1 2x\sin(n\pi x)\,dx$

Using Integration by parts and letting $u = x, du = dx$

$$v = -\frac{-\cos(n\pi x)}{n\pi}, \qquad dv = \sin(n\pi x)\,dx$$

$$-\frac{x\cos(n\pi x)}{n\pi} - \int_0^1 -\frac{\cos(n\pi x)}{n\pi}\,dx$$

$$\left[\frac{x\cos(n\pi x)}{n\pi} + \frac{1}{n\pi}\frac{\sin(n\pi x)}{n\pi}\right]_0^1$$

$$-\frac{\cos(n\pi)}{n\pi}+\frac{1}{n\pi}\left[\frac{\sin(n\pi)}{n\pi}\right]$$

$$-\frac{\cos(n\pi)}{n\pi}+\frac{\sin(n\pi)}{(n\pi)^2}$$

Since $\cos(n\pi)=(-1)^n$ and $\sin(n\pi)=0$ for n=1,2,3, …

$$-\frac{(-1)^n}{n\pi}+\frac{0}{(n\pi)^2}=-\frac{(-1)^n}{n\pi}$$

$$I_2=2\frac{(-1)^n}{n\pi}$$

$$C_n=2[I_1+I_2]=2\left[\frac{1}{2\pi}\left[\frac{(-1)^n+1}{n^2-1}\right]+2\frac{(-1)^n}{n\pi}\right]$$

$$C_n=\frac{1}{\pi}\left[\frac{(-1)^n+1}{n^2-1}\right]+4\frac{(-1)^n}{n\pi}$$

However, since in this form of I_1, C_1 would be dividing by 0. Therefore, C_1 is calculated separately by substituting n=1 to Cn,

$$2\left[\int_0^1(\cos(\pi x)\sin(\pi x))dx-\int_0^1 2x\sin(\pi x)\,dx\right]$$

Since $\cos A\sin B=\frac{1}{2}[\sin(A+B)-\sin(A-B)]$

$$\cos(\pi x)\sin(\pi x)=\frac{1}{2}\left[\sin\big((n+1)\pi x\big)-\sin\big((1-n)\pi x\big)\right]$$

Plugging into left integral I_1

$$2\left[\int_0^1\sin(2\pi x)-\sin(0)\,dx-4\int_0^1 \mathrm{x}\sin(\pi x)\,dx\right]$$

$$\left[-\frac{1}{2}\cos(2\pi)\right]_0^1-4\left(\left[-\frac{x\cos(\pi x)}{\pi}\right]_0^1-\int_0^1-\frac{\cos(\pi x)}{\pi}dx\right)$$

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

Thus,

$$V_1(x,t) = \sin(\pi x)\left[-\frac{4}{\pi}\right]e^{-2(\pi)^2 t}$$

And therefore,

$$V_n(x,t) = -\frac{4}{\pi}\sin(\pi x)\, e^{-2(\pi)^2 t} + \left[\sum_{n=1}^{\infty} \sin(n\pi x)\left[\frac{1}{\pi}\left[\frac{(-1)^n + 1}{n^2 - 1}\right] + 4\frac{(-1)^n}{n\pi}\right]e^{-2(n\pi)^2 t}\right]$$
$$, for\ n = 2,3,4, \ldots$$

Since $u(x,t) = v(x,t) + \varphi(x,t)$,

$$U_n(x,t) = -\frac{4}{\pi}\sin(\pi x)\, e^{-2(\pi)^2 t} + \sum_{n=1}^{\infty} \sin(n\pi x)\left[\frac{1}{\pi}\left[\frac{(-1)^n + 1}{n^2 - 1}\right] + 4\frac{(-1)^n}{n\pi}\right]e^{-2(n\pi)^2 t} + 2x$$
$$for\ n = 2,3,4, \ldots$$

This is the final analytical function that gives the temperature along the rod at any time. Since infinite terms cannot be summed, $n \in [1,1000]$ was selected as its reduces computational power demand, whilst remaining relatively accurate in comparison to the numerical solution.

2.3 Comparing Numerical and Analytical Solutions

Since both numerical and analytical solutions are valid from $t \in [0, \infty]$, specific time instances were used to compare each method of solving the PDE. The time steps chosen are the following: t = 0.001s, t = 0.01s, t = 0.1s, and t = 10s. Figure 4 shows the results, using the previously derived time step for the numerical solution and $N = 1000$ terms for the analytical solution.

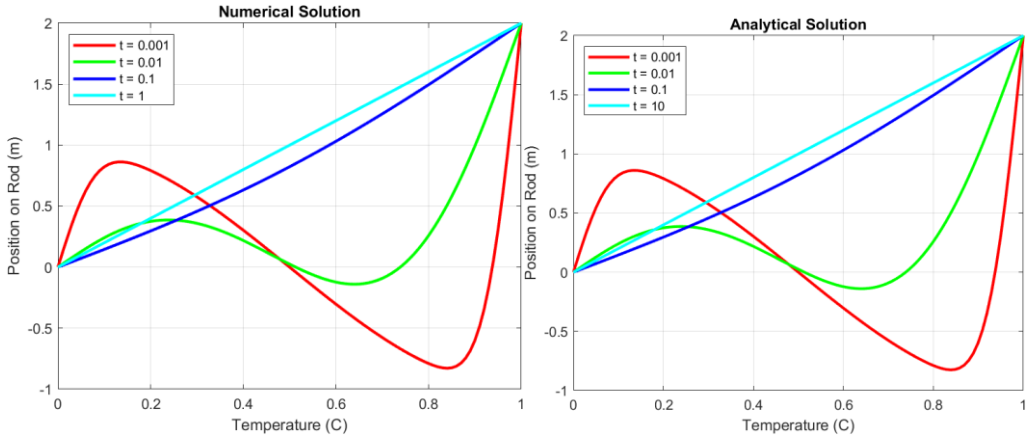


Figure 4: Numerical and analytical solutions

From Figure 4, the two solutions appear near identical. So, the root-means-squared (RMS) of the difference between each solution was plotted at each time instance as seen in Figure 5.

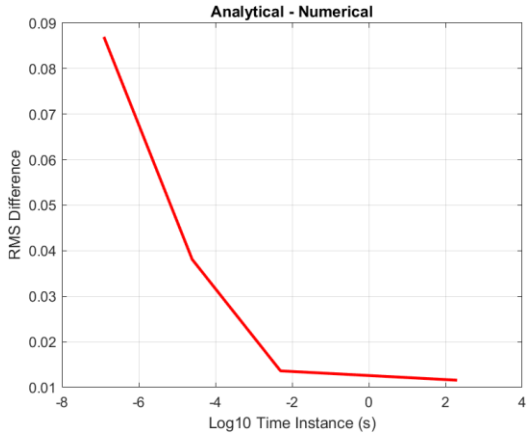


Figure 5: Comparing the analytical and numerical solutions

As seen in Figure 5, there is a very small difference in the solutions with the largest variation being around 0.087 at t = 0.001s and converging to around 0.01 average difference.

2.4 Changing Accuracy of Solutions

Starting with the numerical solution, this section will investigate the effect of the grid spacing on the accuracy of the solution. For this check, a maximum time of 10 seconds and a 1-meter rod length is used.

To observe the effect a grid spacing of $\Delta x = 0.1m$ will have on the numerical solution, it was plotted against the analytical solution as shown in Figure 6 with $n \in [1,1000]$. Since the analytical solution is unaffected by grid spacing and only the number of terms in the summation,

both solutions will use a $\Delta x = 0.1m$ for the 1m rod. Note, with this grid spacing $\Delta t = 2x10^{-3}s$ to meet the stability condition for the numerical number. So, the first-time instance plotted was changed from t = 0.001s to t = 0.002s. The other time instances are the same as defined previously.

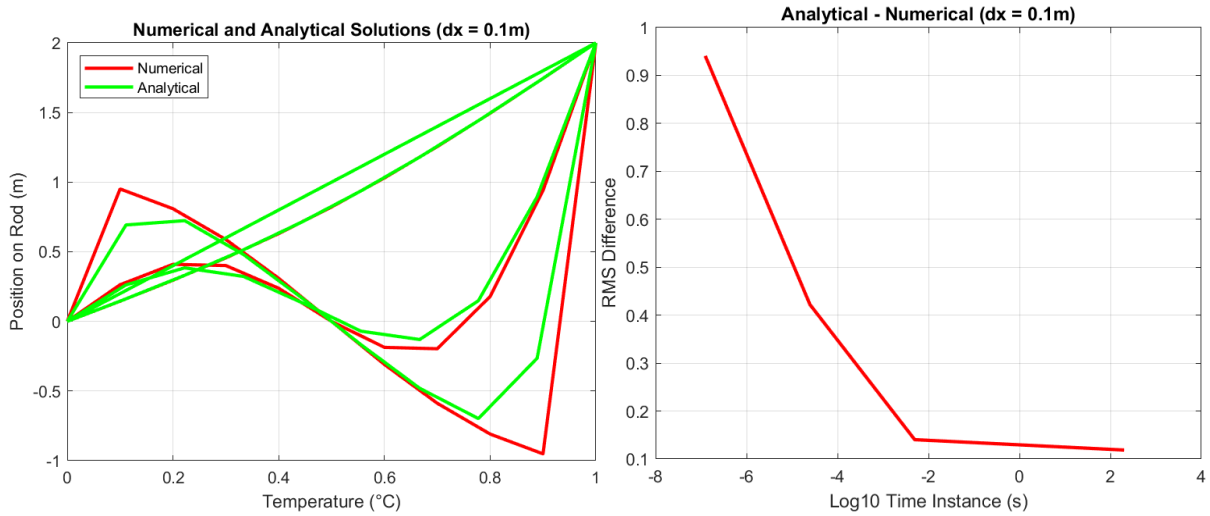


Figure 6: Reduced grid spacing's effect on accuracy of numerical solution

Figure 6 demonstrates that with a reduced grid spacing value of $\Delta x = 0.1m$ the maximum difference jumps to 0.958. So, a grid spacing of $\Delta x = 0.01m$ is a good choice for accuracy with minimal computational power requirements.

In terms of reducing grid spacing, $\Delta x = 0.001m$ would result in exponentially more demanding processing power, shown below.

$$\Delta x = 0.001m, \qquad N_x = \frac{1}{1x10^{-3}} = 10^3 \; steps$$

$$\Delta t < \frac{(0.001)^2}{4} = 2.5x10^{-7}, \qquad \Delta t = 2x10^{-7}, \qquad N_t = \frac{10s}{2x10^{-7}s} = 5x10^7 \; time \; steps$$

This would require a 1 thousand x 50 million matrix to solve every time step which exceeds the computational power available. Thus, the difference in accuracy from $\Delta x = 0.01m$ to $\Delta x = 0.001m$ would not justify using more increased processing power required to solve. As proved previously, using the analytical as a ground truth, the error with grid spacing of $\Delta x = 0.01m$ was less than 0.087 which is already a very low tolerance.

Next, the accuracy of the analytical solution will be compared at different number of summation terms. A ground truth of N = 1000 will be used since there was a very small deviation from the values with this N and the numerical solution at $\Delta x = 0.01m$. Using the same time instances, the function will sum the first 10, 100 and 1000 terms, and is plotted in Figure 7.

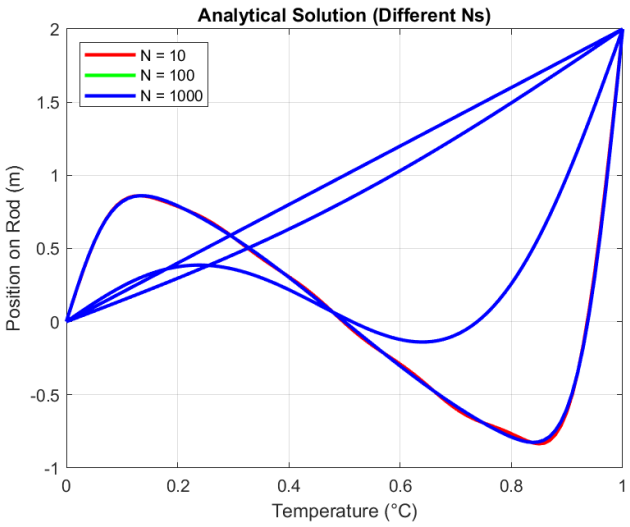


Figure 7: Changing number of summation terms in the analytical solution

There is a small difference when N = 10, with an RMS value of 0.06 at t = 0.01s. When N = 100 the solution is near identical with the RMS value of 0 (rounding). So, the analytical solution is more robust and does not require as much computational power as the numerical approach.

References

[1] S. K. F. D. K. L. K. D. Buay, "How long does it take to boil an egg? Revisited," 04 04 2025. [Online]. Available: https://www.researchgate.net/publication/243414074_How_long_does_it_take_to_boil_an_egg_Revisited.

[2] "Estimating Appliance and Home Electronic Energy Use," 04 04 2025. [Online]. Available: https://www.energy.gov/energysaver/estimating-appliance-and-home-electronic-energy-use.

[3] "Is Induction More Efficient Than Electric Coil or Gas? An Energy Efficiency Comparison Between Stoves," 04 04 2025. [Online]. Available: https://www.centurylife.org/is-induction-more-efficient-than-electric-coil-or-gas-an-energy-efficiency-comparison-between-stoves/.

Statement of Task Division

The tasks in this project were divided among the five group members, ensuring an even distribution of work. Each member contributed to different aspects of the project, as outlined below:

Vikesh Mistry: Developed the numerical solutions and MATLAB code used throughout the project. Wrote sections of the report.

Ali Muizz: Wrote sections of the report, performed edits, and supported MATLAB code development.

Yasir Ahmed: Performed the live experiments for the egg cooking/measurements and wrote it in the report.

Krypton Purnama: Performed the hand calculations for the project and wrote about it within the report. Supported MATLAB code conception.

Saleem Mohammed Ali: Edited the report and provided support for MATLAB coding.

Statement of Generative AI Usage

Generative AI was utilized as a supplementary tool to support work done on this project. It was mainly used for troubleshooting MATLAB code. The final code, report writing, and analysis was completed entirely by the team.

Appendix A – MATLAB Code

1D Heat Equation in Spherical

```
clear; clc;

alpha = 1.5e-7;      % Diffusivity of egg (m^2/s)
R = 0.022;           % Radius (m)
dr = 1e-4;
N = round(R/dr);
dt = 0.01;           % Time step
t_max = 15*60;       % Time (s)
Nt = round(t_max / dt);    % # Time steps

r = linspace(0, R, N)';     % Radius
t = linspace(0, t_max, Nt); % Time grid
U = zeros(N, Nt);           % Temperature storage

% Initial Condition
U(:,1) = 4; % Fridge Temperature

% Boundary condition at r = R
U(end,:) = 100;

% Time loop
for n = 1:Nt-1

   u_new = U(:,n); % Current time step

   % Radial Loop
   for i = 2:N-1

      % Central Difference 1st order
      term1 = (U(i+1,n) - 2*U(i,n) + U(i-1,n)) / (dr*dr);

      % 2nd order
      term2 = (U(i+1,n) - U(i-1,n)) / (r(i) * dr);

      % Final eq
      u_new(i) = U(i,n) + alpha * dt * (term1 + term2);

   end
```

```
    % Apply Neumann BC at r = 0 (du/dr = 0)
    u_new(1) = u_new(2);

    % Store new temperature values
    U(:,n+1) = u_new;

end

[X, Y] = meshgrid(t, r);
col = find(U(1,:) >= 80, 1);
disp(['Time to reach 80C ', num2str(X(1,col)/60,3),' minutes']);
```

GRAPHING

```
clear; clc;

alpha = 1.5e-7;     % Diffusivity of egg (m^2/s)
R = 0.022;          % Radius (m)
dr = 1e-4;
N = round(R/dr);
dt = 0.01;          % Time step
t_max = 15*60;      % Time (s)
Nt = round(t_max / dt);    % # Time steps

r = linspace(0, R, N)'; % Radius
t = linspace(0, t_max, Nt); % Time grid
U = zeros(N, Nt); % Temperature storage

% Initial Condition
U(:,1) = 4; % Fridge Temperature

% Boundary condition at r = R
U(end,:) = 100;


% Time loop
for n = 1:Nt-1

    u_new = U(:,n); % Current time step

    % Radial Loop
```

```
    for i = 2:N-1

        % Central Difference 1st order
        term1 = (U(i+1,n) - 2*U(i,n) + U(i-1,n)) / (dr*dr);

        % 2nd order
        term2 = (U(i+1,n) - U(i-1,n)) / (r(i) * dr);

        % Final eq
        u_new(i) = U(i,n) + alpha * dt * (term1 + term2);

    end

    % Apply Neumann BC at r = 0 (du/dr = 0)
    u_new(1) = u_new(2);

    % Store new temperature values
    U(:,n+1) = u_new;

end

% 3D Surface Plot
% figure;
[X, Y] = meshgrid(t, r);
col = find(U(1,:) >= 80, 1);
disp(['Time to reach 80C ', num2str(X(1,col)/60,3),' minutes']);
surf(X, Y, U, 'EdgeColor', 'none');
 xlabel('Time (s)');
ylabel('Radius (cm)');
zlabel('Temperature (°C)');
title('1D Heat Diffusion in Spherical Coordinates');
colorbar;
grid on;
```

1D Wave Equation Cartesian Coord

NUMERICAL SOLUTION

```
clear; clc;

% Parameters
L = 1;
Tmax = 1;         % Total time
Nx = 101;          % Number of spatial steps
dx = L / (Nx-1);   % Space step size
dt = 0.4 * (dx*dx / 2); % Time step (stability condition)
Nt = round(Tmax / dt); % Number of time steps

x = linspace(0, L, Nx);
t = linspace(0, Tmax, Nt);
U = zeros(Nx, Nt);

% Initial Condition
U(:,1) = cos(pi * x);

% Boundary Conditions
U(1,:) = 0;  % u(0,t) = 0
U(end,:) = 2; % u(1,t) = 2

figure;

for n = 1:Nt-1

   % Get current time step
   u_new = U(:,n);

   % Finite difference
   for i = 2:Nx-1
      u_new(i) = U(i,n) + (2 * dt / dx^2) ...
            * (U(i+1,n) - 2*U(i,n) + U(i-1,n));

   end

   U(:,n+1) = u_new;

end
```

```
timesteps = round([0.001,0.01,0.1,1]./dt);
colors = ['r','g','b','c',"#EDB120"];
numerical = zeros(Nx,length(timesteps));

for k = 1:length(timesteps)

    plot(x, U(:,timesteps(k)), 'Color', colors(k), 'LineWidth', 2);
    hold on;

    %store time values
    numerical(:,k) = U(:,timesteps(k));

end
```

ANALYTICAL SOLUTION

```
x = linspace(0, 1, 100);
t = [0.001, 0.01, 0.1, 10]; % Different time instances
N = 1000; % Number of terms in the summation

[X, T] = meshgrid(x, t);
U = zeros(size(X));

% Compute the series sum
for n = 2:N

    term1 = (((-1)^n + 1)*2*n) / ((n^2 - 1) * pi);
    term3 = 4 * (-1)^n / (n * pi);

    % Avoid division by zero
    if n == 1
        term1 = 0;
    end

    U = U + sin(n * pi * X) .* (term1 + term3) ...
            .* exp(-2 * (n * pi)^2 * T);
end

% Add the linear term 2x
U = -(4/pi)*sin(pi.*X).*exp(-2*pi*pi*T) + U + 2 * X;
```

```
% Plot results
colors = ['m','y','k',"#4DBEEE","#EDB120"];

for i = 1:length(t)

    plot(x, U(i,:), 'Color', colors(i), 'LineWidth', 2);
    hold on;

end

legend('t = 0.001','t = 0.01','t = 0.1','t = 10', ...
           'Location', 'Northwest');
xlabel('Temperature (°C)');
ylabel('Position on Rod (m)');
title('Numerical and Analytical Solutions');
grid on;
hold off;

%store time values
analytical = U;

numerical = transpose(numerical);

error = abs(analytical - numerical(:,1:100));

figure;

for i = 1:length(t)

    error(i,1) = rms(error(i,:));

end

plot(log(t), error(:,1), 'r', 'LineWidth', 2');
xlabel('Log10 Time Instance (s)');
ylabel('RMS Difference');
title('Analytical - Numerical');
grid on;
```

GRID SPACING

```
clear; clc;

% Parameters
L = 1;
Tmax = 1;         % Total time
Nx = 11;          % Number of spatial steps
dx = L / (Nx-1);   % Space step size
dt = 0.4 * (dx*dx / 2); % Time step (stability condition)
Nt = round(Tmax / dt); % Number of time steps

x = linspace(0, L, Nx);
t = linspace(0, Tmax, Nt);
U = zeros(Nx, Nt);

% Initial Condition
U(:,1) = cos(pi * x);

% Boundary Conditions
U(1,:) = 0;  % u(0,t) = 0
U(end,:) = 2; % u(1,t) = 2

figure;

for n = 1:Nt-1

   % Get current time step
   u_new = U(:,n);

   % Finite difference
   for i = 2:Nx-1
      u_new(i) = U(i,n) + (2 * dt / dx^2) ...
            * (U(i+1,n) - 2*U(i,n) + U(i-1,n));

   end

   U(:,n+1) = u_new;

end

timesteps = round([0.002,0.01,0.1,1]./dt);
```

```
colors = ['r','g','b','c',"#EDB120"];
numerical = zeros(Nx,length(timesteps));

for k = 1:length(timesteps)

   plot(x, U(:,timesteps(k)), 'r', 'LineWidth', 2);
   hold on;

   %store time values
   numerical(:,k) = U(:,timesteps(k));

end



%-----------------------------------------------------------------

x = linspace(0, 1, 10);
t = [0.002, 0.01, 0.1, 10]; % Different time instances
N = 1000; % Number of terms in the summation

[X, T] = meshgrid(x, t);
U = zeros(size(X));

% Compute the series sum
for n = 2:N

   term1 = (((-1)^n + 1)*2*n) / ((n^2 - 1) * pi);
   term3 = 4 * (-1)^n / (n * pi);

   % Avoid division by zero
   if n == 1
      term1 = 0;
   end

   U = U + sin(n * pi * X) .* (term1 + term3) ...
            .* exp(-2 * (n * pi)^2 * T);
end

% Add the linear term 2x
U = -(4/pi)*sin(pi.*X).*exp(-2*pi*pi*T) + U + 2 * X;

% Plot results
```

```
colors = ['m','y','k',"#4DBEEE","#EDB120"];

for i = 1:length(t)

    plot(x, U(i,:), 'g', 'LineWidth', 2);
    hold on;

end

legend('Numerical',"","","",'Analytical', ...
           'Location', 'Northwest');
xlabel('Temperature (°C)');
ylabel('Position on Rod (m)');
title('Numerical and Analytical Solutions (dx = 0.1m)');
grid on;
hold off;

%store time values
analytical = U;

numerical = transpose(numerical);

error = abs(analytical - numerical(:,1:10));

figure;

for i = 1:length(t)

    error(i,1) = rms(error(i,:));

end

plot(log(t), error(:,1), 'r', 'LineWidth', 2');
xlabel('Log10 Time Instance (s)');
ylabel('RMS Difference');
title('Analytical - Numerical (dx = 0.1m)');
grid on;
```

ANALYICAL NO TERMS TEST

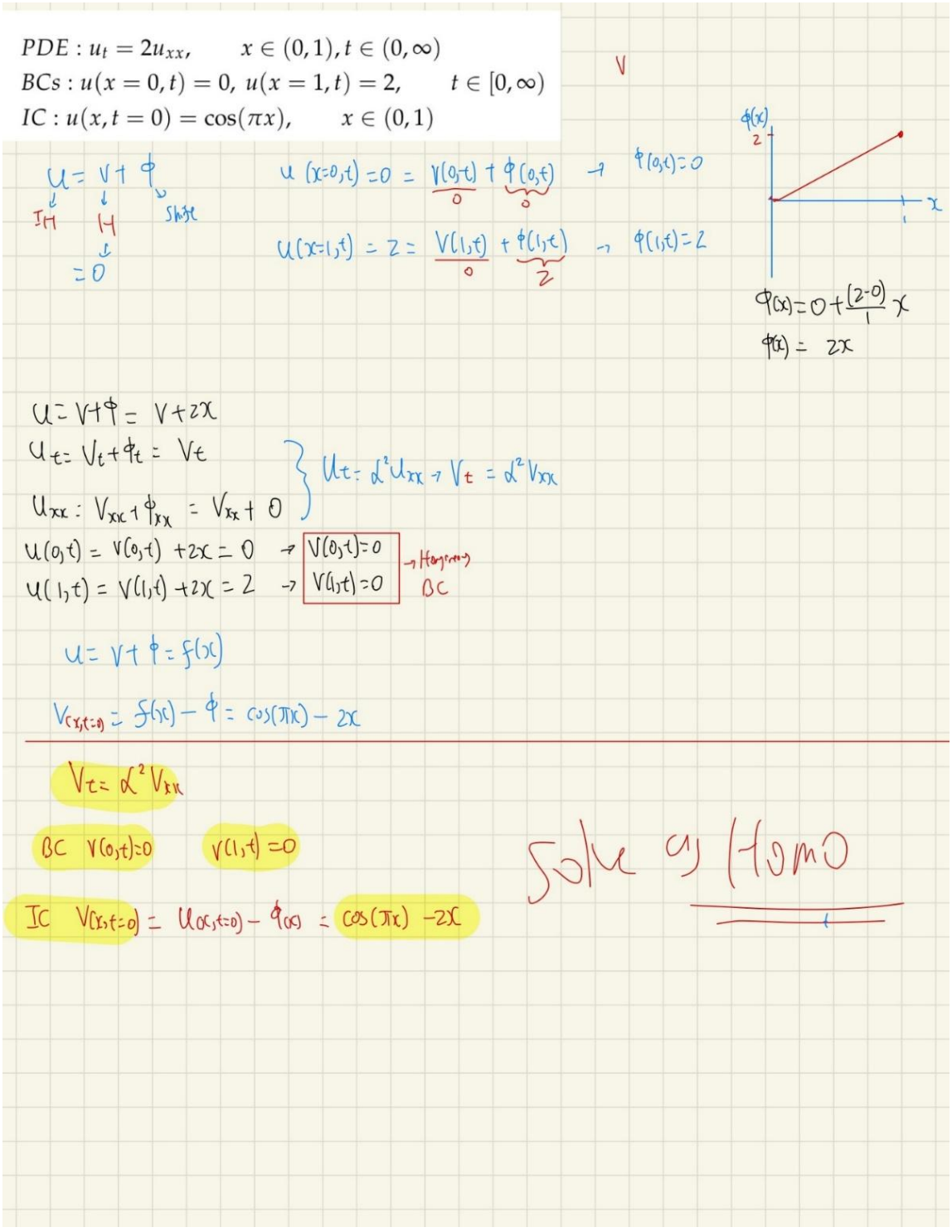
```
clc; clear;

x = linspace(0, 1, 100);
t = [0.001, 0.01, 0.1, 10]; % Different time instances
N = 10; % Number of terms in the summation

[X, T] = meshgrid(x, t);
U = zeros(size(X));

% Compute the series sum
for n = 2:N

   term1 = (((-1)^n + 1)*2*n) / ((n^2 - 1) * pi);
   term3 = 4 * (-1)^n / (n * pi);

   % Avoid division by zero
   if n == 1
      term1 = 0;
   end

   U = U + sin(n * pi * X) .* (term1 + term3) ...
            .* exp(-2 * (n * pi)^2 * T);
end

% Add the linear term 2x
U = -(4/pi)*sin(pi.*X).*exp(-2*pi*pi*T) + U + 2 * X;

% Plot results
figure;
%colors = ['r','g','b','c',"#EDB120"];

for i = 1:length(t)

   plot(x, U(i,:), 'r', 'LineWidth', 2);
   hold on;

end

first_ans = U;

N = 100;
```

```
U = zeros(size(X));

% Compute the series sum
for n = 2:N

   term1 = (((-1)^n + 1)*2*n) / ((n^2 - 1) * pi);
   term3 = 4 * (-1)^n / (n * pi);

   % Avoid division by zero
   if n == 1
      term1 = 0;
   end

   U = U + sin(n * pi * X) .* (term1 + term3) ...
            .* exp(-2 * (n * pi)^2 * T);
end

% Add the linear term 2x
U = -(4/pi)*sin(pi.*X).*exp(-2*pi*pi*T) + U + 2 * X;

% Plot results
for i = 1:length(t)

   plot(x, U(i,:), 'g', 'LineWidth', 2);
   hold on;

end

second_ans = U;

N = 1000;
U = zeros(size(X));

% Compute the series sum
for n = 2:N

   term1 = (((-1)^n + 1)*2*n) / ((n^2 - 1) * pi);
   term3 = 4 * (-1)^n / (n * pi);

   % Avoid division by zero
   if n == 1
      term1 = 0;
   end
```

```
    U = U + sin(n * pi * X) .* (term1 + term3) ...
             .* exp(-2 * (n * pi)^2 * T);
end

% Add the linear term 2x
U = -(4/pi)*sin(pi.*X).*exp(-2*pi*pi*T) + U + 2 * X;

% Plot results
%colors = ['r','g','b','c","#EDB120"];

for i = 1:length(t)

    plot(x, U(i,:), 'b', 'LineWidth', 2);
    hold on;

end

third_ans = U;

first_error = third_ans - first_ans;
first_error = rms(first_error,"all");

second_error = third_ans - second_ans;
second_error = rms(second_error,"all");

legend('N = 10',",",",",",'N = 100',",",",",'N = 1000', ...
          'Location', 'Northwest');
xlabel('Temperature (°C)');
ylabel('Position on Rod (m)');
title('Analytical Solution (Different Ns)');
grid on;
hold off;
```

Appendix B – Hand Calculations

The hand calculations for the analytical solution of the PDE solved in Section 2 is provided in the series of images below.



$V(x,t) = X(x)\,T(t) = XT$ $\quad \alpha = \sqrt{2}$

$V_t = \alpha^2 V_{xx}$

BC $V(0,t)=0$ $\quad V(1,t)=0$

IC $V(x,t=0) = U(x,t=0) - \phi(x) = \cos(\pi x) - 2x$

$XT' = \alpha^2 X'' T$

$\frac{X''}{X} = \frac{T'}{\alpha^2 T} = k$

(1) $X'' - kX = 0$

(2) $T' - k\alpha^2 T = 0$

(1) $V = XT$ <

BC1 $\rightarrow V(x=0,t) = 0 = X(0)\,T(t) \Rightarrow \underline{X(0)=0}$

BC2 $\rightarrow V(x=1,t) = 0 = X(1)\,T(t) \Rightarrow \underline{X(1)=0}$

at $k=0$ $\quad X''=0 \rightarrow X = ax+b = 0$ <

$0 = a(0)+b \rightarrow b=0$

$0 = a(1)+0 \rightarrow a=0$

> trivial

at $k=M^2>0$ $\quad X'' - M^2 X = 0 \rightarrow X = Ae^{-Mx} + Be^{Mx}$ <

$0 = A+B$

$0 = Ae^{-M} + Be^{M}$

$B\left(-e^{-M} + e^{M}\right) = 0$

$e^{-M} = e^{M} \rightarrow -M = M$ Not possible

so $0=0$ & $A=0$

$k = -M^2 < 0 \quad X'' + M^2 X = 0 \rightarrow X = A\cos(Mx) + B\sin(Mx)$

$0 = A\cos(0) + B\sin(0) \rightarrow A = 0$

$0 = A\cos(M) + B\sin(M) \rightarrow B\sin(M) = 0 \rightarrow \sin(M) = 0$

$\mu = n\pi$ $\quad n = 1, 2, 3, \ldots$

$k_n = -(n\pi)^2$

$X_n(x) = B \sin(n\pi x) = \sin(n\pi x)$

(2) $T' - 2kT = 0$

$T' + \underbrace{2(n\pi)^2}_{(\sqrt{2}\,n\pi)^2} T = 0$

$T_n(t) = C_n e^{-2(n\pi)^2 t}$

$V(x,t) = \sum_{n=1}^{\infty} V_{n(x,t)} = \sum_{n=1}^{\infty} \sin(n\pi x)\left(C_n e^{-2(n\pi)^2 t}\right)$

$V(x, t=0) = \cos(\pi x) - 2x$

$\sum_{n=1}^{\infty} \sin(n\pi x)(C_n) = \cos(\pi x) - 2x$.

Since only sine term, use sine fourier where $C_n = \frac{2}{L}\int_0^L (\cos(\pi x) - 2x)\sin(\frac{n\pi x}{L})\,dx$

$C_n = \frac{2}{1}\int_0^1 (\cos(\pi x) - 2x)\sin(n\tfrac{\pi}{1}x)\,dx$, $n \geq 1$ ⚠

$= 2\int_0^1 (\cos(\pi x) - 2x)\sin(n\pi x)\,dx$

$= 2\left[\int_0^1 \cos(\pi x)\sin(n\pi x)\,dx - \int_0^1 2x\sin(n\pi x)\,dx\right]$

Left Integral I_1: $\int_0^1 \cos(\pi x)\sin(n\pi x)\,dx$

since $\cos A \sin B = \frac{1}{2}[\sin(A+B) - \sin(A-B)]$

$\cos(\pi x)\sin(n\pi x)$

$\downarrow$

$\frac{1}{2}\left[\sin((n+1)\pi x) - \sin((1-n)\pi x)\right]$

$\frac{1}{2}\left[\int_0^1 \sin((n+1)\pi x) - \int_0^1 \sin((1-n)\pi x)\right]$

$\frac{1}{2}\left[\frac{-\cos((n+1)\pi x)}{(n+1)\pi}\Big|_0^1 - \frac{-\cos((1-n)\pi x)}{(1-n)\pi}\Big|_0^1\right] =$

$\frac{1}{2}\left[-\frac{\cos((n+1)\pi)}{(n+1)\pi} - \left(-\frac{\cos(0)}{(n+1)\pi}\right) - \left[\frac{-\cos((1-n)\pi)}{(1-n)\pi} - \left(\frac{-\cos(0)}{(1-n)\pi}\right)\right]\right]$

$= \frac{1}{2}\left[\frac{-\cos((n+1)\pi)+1}{(n+1)\pi} - \left[\frac{-\cos((1-n)\pi)+1}{(1-n)\pi}\right]\right]$ since $\cos(k\pi) = (-1)^k$ $\rightarrow$ $\frac{1}{2}\left[\frac{-(-1)^{n+1}+1}{(n+1)\pi} + \left(\frac{(-1)^{1-n}-1}{(1-n)\pi}\right)\right]$, $n \geq 2$

Right Integral I_2: $-\int_0^1 2x\sin(n\pi x)\,dx$

$-2\int_0^1 x\sin(n\pi x)\,dx$

IBP

let $u = x$, $v = -\frac{\cos(n\pi x)}{n\pi}$

$du = dx$, $dv = \sin(n\pi x)\,dx$

$\left[-\frac{x\cos(n\pi x)}{n\pi}\right]_0^1 - \int_0^1 -\frac{\cos(n\pi x)}{n\pi}\,dx$

$= -\left[-\frac{x\cos(n\pi x)}{n\pi} + \frac{1}{n\pi}\frac{\sin(n\pi x)}{n\pi}\right]_0^1$

$= -\frac{\cos(n\pi)}{n\pi} + \frac{1}{n\pi}\left[\frac{\sin(n\pi)}{n\pi}\right]$

$= -\frac{\cos(n\pi)}{n\pi} + \frac{\sin(n\pi)}{(n\pi)^2} = -\frac{(-1)^n}{n\pi} + \frac{0}{(n\pi)^2} = -\frac{(-1)^n}{n\pi}$

$\therefore I_2 = 2\frac{(-1)^n}{n\pi}$

$\frac{1}{2}\left[\frac{(-1)^n+1}{(n+1)\pi} + \frac{(-1)^n+1}{(n-1)\pi}\right]$

$\frac{1}{2\pi}\left[\frac{(-1)^n(n-1) + (n-1) + (-1)^n(n+1) + (n+1)}{n^2-1}\right]$

$\frac{1}{2\pi}\left[\frac{(-1)^n\cdot 2n + 2n}{n^2-1}\right] = \frac{2n[(-1)^n+1]}{\pi[n^2-1]}$

$$C_n = 2\left[I_1 + I_2\right] = 2\left[\frac{n\left[(-1)^n+1\right]}{\pi\left[n^2-1\right]}\right) + \frac{2(-1)^n}{n\pi}\Bigg]$$

$$C_n = \frac{2n\left[(-1)^n+1\right]}{\pi\left[n^2-1\right]} + 4\,\frac{(-1)^n}{n\pi} \qquad n \geq 2$$

$$V_{(x,t)} = C_1 + \sum_{n=2}^{\infty} \sin(n\pi x)\left(\frac{2n\left[(-1)^n+1\right]}{\pi\left[n^2-1\right]} + 4\,\frac{(-1)^n}{n\pi}\right) e^{-2(n\pi)^2 t}$$

$$u_{(x,t)} = C_1 + \left[\sum_{n=2}^{\infty} \sin(n\pi x)\left(\frac{2n\left[(-1)^n+1\right]}{\pi\left[n^2-1\right]} + 4\,\frac{(-1)^n}{n\pi}\right) e^{-2(n\pi)^2 t}\right] + 2x$$

Since $C_n = 2\left[\int_0^1 \cos(\pi x)\sin(n\pi x)\,dx - \int_0^1 2x\sin(n\pi x)\,dx\right]$

$$C_1 = 2\left[\int_0^1 \cos(\pi x)\sin(\pi x)\,dx - \int_0^1 2x\sin(\pi x)\,dx\right]$$

$\cos A \sin B = \frac{1}{2}\left[\sin(A+B) - \sin(A-B)\right]$

$$= \int_0^1 \sin(2\pi x) - \sin(\pi x - \pi x)\,dx - 2\int_0^1 2x\sin(\pi x)\,dx$$

$$= \left[-\frac{1}{2}\cos(2\pi x)\right]_0^1 - 4\int_0^1 x\sin(\pi x)\,dx$$

$u = x$ $\quad V = \frac{-\cos(\pi x)}{\pi}$

$du = dx$ $\quad dV = \sin(\pi x)$

$$= -\frac{1}{2}\cos(2\pi) + \frac{1}{2}\cos(0) - 4\left(\left[-\frac{x\cos(\pi x)}{\pi}\right]_0^1 - \int_0^1 -\frac{\cos(\pi x)}{\pi}\,dx\right)$$

$$= -4\left(-\frac{\cos(\pi)}{\pi} + \left[\frac{\sin(\pi x)}{\pi^2}\right]_0^1\right) = -4\left(\frac{1}{\pi} + \frac{\sin(\pi)}{\pi^2} - \frac{\sin 0}{\pi^2}\right) = -\frac{4}{\pi}$$

$$C_1 = -\frac{4}{\pi}$$

$$u(x,t) = -\frac{4}{\pi}\sin(\pi x)\,e^{-2(\pi)^2 t} + \left[\sum_{n=2}^{\infty}\sin(n\pi x)\left(\frac{2n\left[(-1)^n+1\right]}{\pi\left[n^2-1\right]} + 4\frac{(-1)^n}{n\pi}\right)e^{-2(n\pi)^2 t}\right] + 2x$$