

The Heat Equation

Introduction

In this report, we investigate the solutions to the one-dimensional heat equation. The solutions describe the evolution of the temperature distribution in a rod of length l whose ends are kept at zero temperature from the initial temperature profile given by $f(x)$. We will first find these solutions for various initial conditions analytically using variable separation and the Fourier method. We will then solve these equations numerically, allowing us to plot these results. This allows us to compare the accuracy of the numerical method with the analytical. Finally, we extended our use of numerical methods to solve the heat equation for more complex and interesting initial conditions.

Working Equations-Analytic

The heat equation in one dimension (with that dimension being the x-axis) is given by **Eq. (1)**.

$$u_t - Ku_{xx} = 0 \quad (1)$$

Where $u(x, t)$ is the temperature of the rod given as a function of position and time, and K is the parameter characteristic of the medium for a rod of length l , whose ends are kept at zero temperature, and $x \in [0, l]$. We have the following boundary conditions seen in **Eq. (2)**, sometimes known as the Dirichlet Boundary Conditions.

$$u(0, t) = 0 \quad \text{and} \quad u(l, t) = 0 \quad (2)$$

We also consider a general initial condition presented in **Eq. (3)**.

$$u(x, 0) = f(x) \quad (3)$$

We derived the general solution of the heat equation by applying variable separation and the principle of superposition, and our results are presented below.

$$u(x, t) = \sum_{n=1}^{\infty} B_n \sin\left(\frac{n\pi x}{l}\right) e^{-\left(\frac{n^2\pi^2}{l^2}\right)Kt} \quad (4)$$

The coefficients B_n were then determined by the initial conditions using the Fourier method. We considered two different initial conditions. The first of these we call (a), which is presented formally in **Eq.(5)**. Using elementary integration, we then found the coefficients B_n and substituted them into **Eq.(4)** to yield **Eq.(6)**.

$$f(x) = \begin{cases} 0, & 0 \leq x < a \\ T, & a \leq x \leq b \\ 0, & b < x \leq l \end{cases} \quad \text{where } 0 < a < b < l \quad (5)$$

$$u(x, t) = \sum_{n=1}^{\infty} \frac{2T \left[\cos\left(\frac{n\pi a}{l}\right) - \cos\left(\frac{n\pi b}{l}\right) \right]}{n\pi} \sin\left(\frac{n\pi x}{l}\right) e^{-\left(\frac{n^2\pi^2}{l^2}\right)Kt} \quad (6)$$

Similarly, we considered a second initial condition, which we call (b), which is given by **Eq.(7)**. The general solution in the case of (b) was determined using the same approach and was found to be **Eq.(8)**.

$$f(x) = \frac{4T}{l^2} x(l-x) \quad (7)$$

$$u(x, t) = \sum_{n=1}^{\infty} \frac{32T}{\pi^3 (2n-1)^3} \sin\left(\frac{(2n-1)\pi x}{l}\right) e^{-\left(\frac{(2n-1)^2 \pi^2}{l^2}\right) Kt} \quad (8)$$

It should be noted that **Eq.(5)** is known as a step-function, and **Eq.(8)** is a parabolic continuous function. This will be significant for our results. We will now investigate the behavior and accuracy of these equations using visual support.

Results-Analytic

In this section, we build on our previous analytic work by examining the accuracy of our Fourier series produced for the two initial conditions using Fourier approximations. We consider first the limits of our approximations at $t = 0$, i.e. $u(x, 0)$, then the time evolution for our equations, i.e. $u(x, t)$. The results were produced using Python.

Fig. (1) illustrates the Fourier approximations of **Eq. (5)** by use of $u(x, 0) = f(x)$. We were able to set a variety of terms in the partial sum, N shown in **Eq. (6)**, with $t = 0$, and with parameters shown in the figure caption to create approximations. As we would expect, the accuracy of the approximations increases with N , which is seen in the decrease of the amplitude of the oscillations as N increases. For small values of N , the rectangular shape of $f(x)$, is not well seen, and this error most notably occurs at the discontinuous points of **Eq. (5)** $x = a$, and $x = b$. With large values of N , we notice that the approximation is weakest around $x = a$ and $x = b$, where we see larger amplitudes in the oscillations compared to other intervals of the function. This is the effect of the Gibbs phenomenon, which suggests that at the discontinuity, the partial sums display oscillations that become progressively narrower and move closer to the discontinuity. However, they always overshoot, meaning the approximating curve does not approach the jump in $f(x)$. This is caused by the convergence theorem requiring a piecewise smooth function for the Fourier approximation to converge accurately.

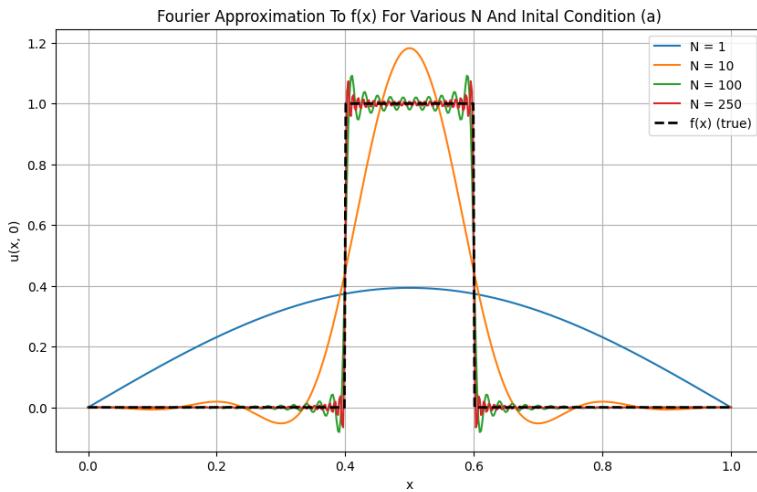


Figure 1: Fourier approximation for the first considered initial condition $f(x) = u(x, 0)$ as seen in **Eq.(5)**. The approximations were produced using **Eq.(6)** and **Eq.(5)** with $t = 0$, $a = 0.4$, $b = 0.6$, $T = 1$, $K = 1$ and $l = 1$ substituted in, and a variety of terms in the partial sum, N ($N = 1$ (blue line), $N = 10$ (yellow line), $N = 100$ (green line) and $N = 250$ (red line)). The figure is produced using Python for $x \in [0, 1]$, and $f(x)$ (**Eq.(5)**) is plotted for reference with a dashed black line. The Fourier approximations of $f(x)$ (**Eq.(5)**) are seen to improve as N increases.

In the case of **Fig. (2)** in a similar manner we illustrate the Fourier approximations of **Eq. (7)** by use of $u(x, 0) = f(x)$. We were able to set a variety of several terms in the partial sum, N shown in **Eq. (8)**, with $t = 0$, and with parameters shown in the figure caption to create a set of approximations. Again, we notice an increase in the accuracy of the approximations as N increases. However, in this case, the approximations converge rapidly, and even for low values of N , the approximations are highly accurate compared to our analysis of **Fig. (1)** we notice no overshooting or noticeable points of lack of convergence. This is due to a lack of effect from Gibbs phenomenon, as **Eq. (7)** is a continuous function unlike **Eq. (5)** we can create an approximation that converges as $N \rightarrow \infty$ with no overshooting.

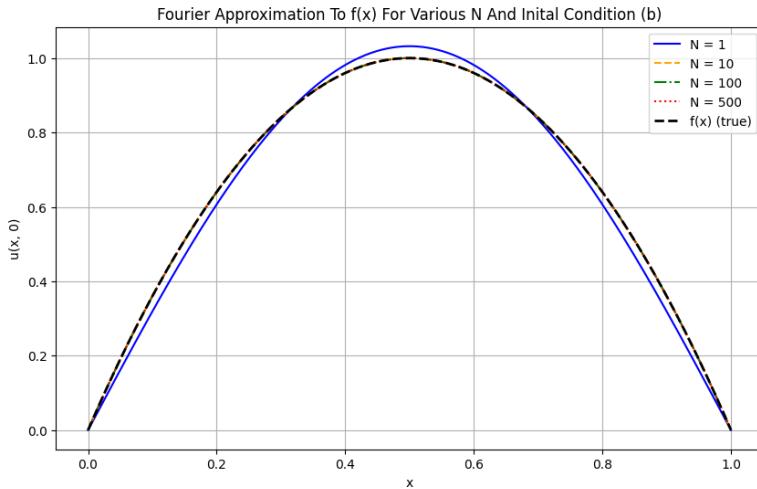


Figure 2: Fourier approximation for the second considered initial condition $f(x) = u(x, 0)$ as seen in **Eq.(7)**. The approximations were produced using **Eq.(8)** with $t = 0$, $a = 0.4$, $b = 0.6$, $T = 1$, $K = 1$ and $l = 1$ substituted in, and a variety of several terms in the partial sum, N ($N = 1$ (blue line), $N = 10$ (yellow dashed line), $N = 100$ (green dashed line) and $N = 250$ (red dashed line)). The figure is produced using Python for $x \in [0, 1]$, and $f(x)$ (**Eq.(7)**) is plotted for reference with a dashed black line. The Fourier approximations of $f(x)$ (**Eq.(7)**) are seen to improve as N increases.

We now describe our results for the time evolution of our $u(x, t)$. **Fig. (3)** illustrates the time evolution for **Eq.(6)**, with a maximum of partial sums N , a variety of time values, and specified parameters, which are all described in the figure caption. Initially, for $t = 0$, our figure replicates what we see for $N = 250$ in **Fig (1)**, this is to be expected a rapidly oscillating curve with an approximately constant temperature in the intervals $[0, a)$, $[a, b]$ and $(b, l]$. When t increases, the oscillations become increasingly smooth and symmetric, approaching the line $u(x, t) = 0$. This is due to the $e^{-\left(\frac{(2n-1)^2 \pi^2}{l^2}\right)Kt}$ in

Eq.(6), which goes to zero as $t \rightarrow \infty$ forcing the sum to approach zero. This agrees with our physical interpretation of the rod: we expect rapid heat dissipation over time in a positionally symmetric manner.

A similar result is seen **Fig. (4)** where we illustrate the time evolution for **Eq.8**, with a maximum of partial sums N , a variety of time values, and specified parameters, which are all described in the figure caption. At $t = 0$, we see the curve plotted for $N = 250$ in **Fig. (2)**, however, this time it is a smooth symmetric curve with no overshooting. The figure again shows that the line $u(x, t) = 0$ is approached as t increases. This is due to the $e^{-\left(\frac{(2n-1)^2 \pi^2}{l^2}\right)Kt}$ which also appears in **Eq.(8)**. This term again forces the sum to zero as $t \rightarrow \infty$. We should note that the smoothness of this progressive approach to $u(x, t) = 0$ is noticeable compared to **Fig. (3)**. This is once caused by the continuity of our initial condition seen in **Eq.(7)**. For this initial condition, we also see what our physical interpretation would suggest: that the rod loses heat rapidly over time in a positionally symmetric manner.

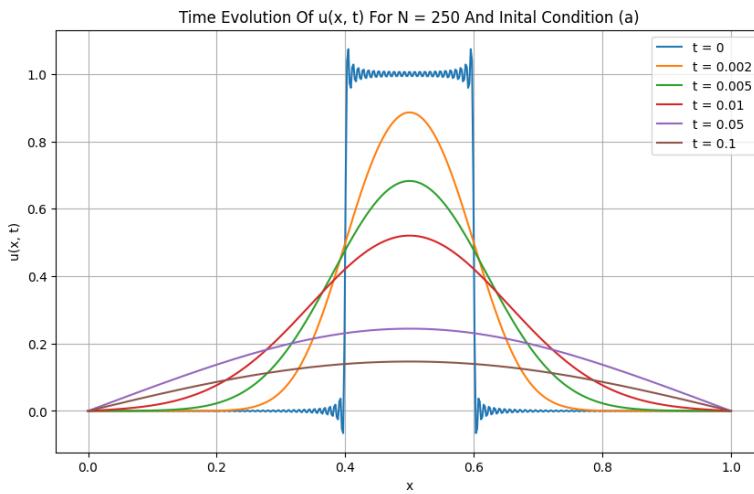


Figure 3: Time evolution of $u(x, t)$ as described by Eq.(6) with a number of partial sum terms of $N = 250$. We used values of $a = 0.4$, $b = 0.6$, $T = 1$, $K = 1$, $l = 1$. The curves shown represent different values of t ($t = 0$ (blue line), $t = 0.002$ (yellow line), $t = 0.005$ (green line), $t = 0.01$ (red line) , $t = 0.05$ (purple line) and $t = 0.1$ (brown line)). The plot was produced using Python for $x \in [0, 1]$. The plots are seen to smoothen after $t = 0$ and approach the line $u(x, t) = 0$ as t increases.

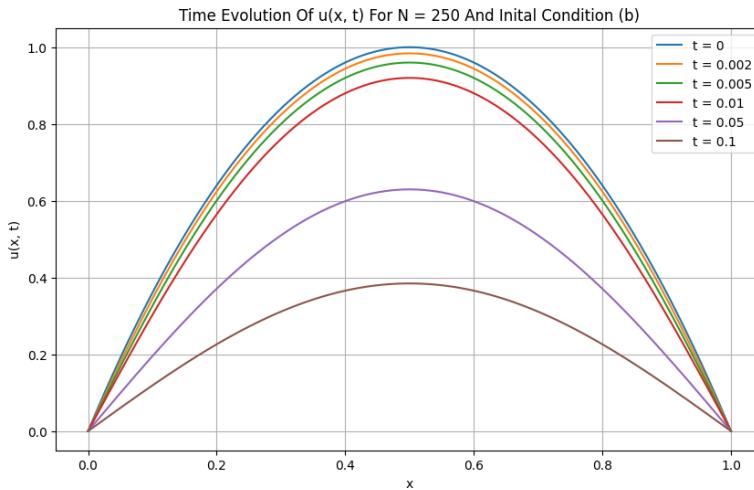


Figure 4: Time evolution of $u(x, t)$ as described by Eq.(8) with a number of partial sum terms of $N = 250$. We used values of $a = 0.4$, $b = 0.6$, $T = 1$, $K = 1$, $l = 1$. The curves shown represent different values of t ($t = 0$ (blue line), $t = 0.002$ (yellow line), $t = 0.005$ (green line), $t = 0.01$ (red line) , $t = 0.05$ (purple line) and $t = 0.1$ (brown line)). The plot was produced using Python for $x \in [0, 1]$. The plots are seen to approach the line $u(x, t) = 0$ as t increases.

The results in this section have shown us that the accuracy of our Fourier approximations increases significantly as N increases and that the solution $u(x, t)$ for both initial conditions rapidly approaches zero as time passes. Most notably, our Fourier approximations were significantly more accurate when applied to a continuous function, causing no visible oscillations near jump points, as in the case of discontinuous functions, giving us a clear visualization of the Gibbs phenomenon. We will now discuss solutions to the heat equation using numerical methods instead of our previous analytical approach.

Working Equations-Numeric

There are many cases where an initial condition causes finding the Fourier coefficients analytically to be cumbersome or impossible. In that case, it is practical to implement numerical methods. In our situation, we will use an approach known as the finite difference method.

We will return to **Eq. (1)**, with boundary conditions given by **Eq. (2)**. We will, this time, consider slightly different initial conditions. We first define a grid $n_s + 1$ spatial points in steps of $h = \frac{l}{n_s}$ that is, $x_i = ih$ (for $i = 0, 1, 2, \dots, n_s$), and a grid of time values in steps of ε described by $t_k = k\varepsilon$ (for $k = 0, 1, 2, \dots$). For convenience, we let $u(x_i, t_k) \equiv u_i^{(k)}$. We then recall the Taylor expansion of $u(x, t)$ in x :

$$\begin{aligned} u(x + h, t) &= u(x, t) + u_x(x, t)h + \frac{1}{2}u_{xx}(x, t)h^2 + \dots, \\ u(x - h, t) &= u(x, t) - u_x(x, t)h + \frac{1}{2}u_{xx}(x, t)h^2 + \dots. \end{aligned}$$

The addition of these two equations and the removal of terms with an order higher than two gives **Eq. (9)**.

$$u_{xx}(x, t) = \frac{1}{h^2} [u(x + h, t) - 2u(x, t) + u(x - h, t)] \quad (9)$$

Using our grids as we defined, we can write this in our new notation as seen in **Eq. (10)**.

$$(u_{xx})_i^{(k)} = \frac{1}{h^2} (u_{i+1}^{(k)} - 2u_i^{(k)} + u_{i-1}^{(k)}) \quad (10)$$

Using a similar concept for the Taylor expansion in t for $u(x, t + \varepsilon)$, and keeping only terms to first order in ε yields **Eq. (11)**.

$$(u_t)_i^{(k)} = \frac{u_i^{(k+1)} - u_i^{(k)}}{\varepsilon} \quad (11)$$

Substituting **Eq. (10)** and **Eq. (11)** into **Eq. (1)**(the heat equation), yields **Eq. (12)**.

$$u_i^{(k+1)} = u_i^{(k)} + \frac{K\varepsilon}{h^2} (u_{i+1}^{(k)} - 2u_i^{(k)} + u_{i-1}^{(k)}) \quad (12)$$

In the case $k = t = 0$, we have $u_i^{(0)} = u(x_i, 0) = f(x_i) = f(ih)$ (which is determined by our initial condition). This then allows us to find $u_i^{(1)}$ for $i = 1, 2, \dots, n_s - 1$. In the cases $i = 0$, and $i = n_s$, we have $u_0^{(1)} = u_{n_s}^{(1)} = 0$ (due to boundary conditions seen in **Eq. (2)**). Now that we know $u_i^{(1)}$ (for $i = 0, 1, 2, \dots, n_s$), we can make use of **Eq. (12)** to find $u_i^{(2)}$,etc. After repeating this process k times, we can find the approximate numerical values of $u(x_i, t_k)$.

This notation can be simplified using matrix form. Let $\mathbf{u}^{(k)}$ be the vector whose components are the values of $u_i^{(k)}$. Then defining the $(n_s + 1) \times (n_s + 1)$ matrix **Eq. (13)**, where $\alpha = \frac{K\varepsilon}{h^2}$ and $\beta = 1 - 2\alpha$.

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ \alpha & \beta & \alpha & \cdots & 0 \\ 0 & \alpha & \beta & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \alpha \\ 0 & \cdots & 0 & \alpha & \beta \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix} \quad (13)$$

We now rewrite **Eq. (12)** as **Eq. (14)**.

$$\mathbf{u}^{(k+1)} = \mathbf{A}\mathbf{u}^{(k)} \quad (14)$$

We start from $\mathbf{u}^{(0)} = (f(x_i))$, each subsequent step forward in time simply requires multiplying the current temperature distribution $\mathbf{u}^{(k)}$ by the matrix \mathbf{A} .

We now apply these methods to produce approximate solutions for $u(x, t)$ for two initial conditions. The first initial condition being **Eq. (7)** with $T = 1$ and $l = 1$ yielding **Eq. (15)**. The second initial

condition is described by **Eq. (16)**. We will keep calling the first of the initial conditions (b) as we are using the same parameters for **Eq. (7)** as in **Fig. (4)**, and the second initial condition will be called (c). We should note that **Eq. (16)** can be described as a Gaussian function.

$$f(x) = 4x(1-x) \quad (15)$$

$$f(x) = e^{-100(x-0.5)^2} \quad (16)$$

Results-Numeric

In this section, we apply our numerical theory to the heat equation **Eq. (1)** with boundary conditions **Eq. (2)**, and initial conditions (b) **Eq. (15)** and (c) **Eq. (16)**. We first created a numerical approximation for $u(x, t)$ at $t = 0.1$ in the initial condition (b) case. We then constructed numerical approximations for the $u(x, t)$ solutions for various t values. These results were produced using Python.

Fig. (5) illustrates the numerical approximation for the solution of the heat equation at time $t = 0.1$, with the initial condition $f(x) = 4x(1 - x)$, and boundary conditions given by **Eq. (2)**. The parameters used for the numerical method and $u(x, t)$ are specified in the figure caption. We notice that the curve $u(x, t = 0.1)$ is visually identical to the curve $t = 0.1$ in **Fig. (4)**. This is a statement of the accuracy of our numerical approximation. Similar to what we see **Fig. (4)**, we see the temperature of the rod decrease in a smooth, symmetric manner, as our physical intuition would suggest.

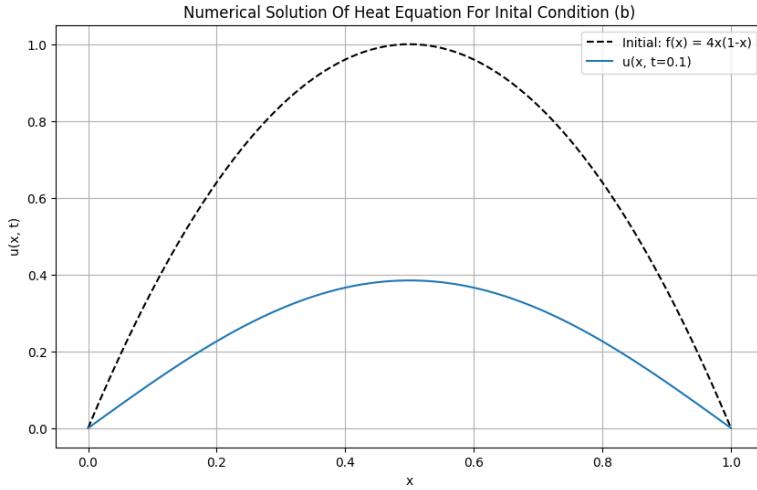


Figure 5: The numerical approximation for the solution to the heat equation $u(x, t)$ (described by **Eq. (8)**), with parameters $K = 1, l = 1$ and $T = 1$. The initial condition $f(x) = 4x(1 - x)$ (see **Eq. (15)**) was plotted using a dashed black line. The numerical approximation for $u(x, t)$ at $t = 0.1$ was plotted using a solid blue line. It was computed using the finite-difference method developed in the previous section with $h = 0.01, \varepsilon = 10^{-5}$, and ten thousand time steps. The plot was produced using Python.

In **Fig. (6)** we illustrate the numerical approximations for the solutions of the heat equation at a variety of times, with the initial condition $f(x) = e^{-100(x-0.5)^2}$, and boundary conditions given by **Eq. (2)**. The parameters used for the numerical method and $u(x, t)$ are specified in the figure caption. The plot shows the expected diffusion of heat. We have a series of symmetric bell-shaped curves that flatten as time passes. Heat dissipation is seen to be more rapid initially and to slow down as the curves approach $u(x, t) = 0$. This is consistent with our physical understanding of the rod, where we would expect it to cool symmetrically rapidly initially and for that cooling to slow down as time goes by.

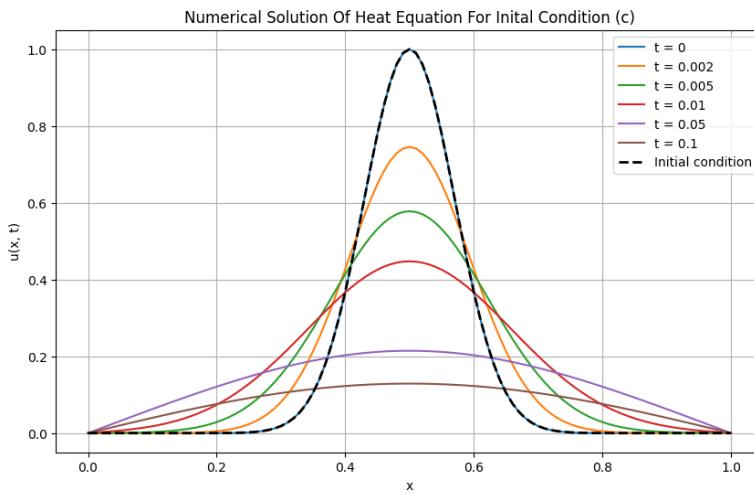


Figure 6: Numerical approximations for the solutions to the heat equation $u(x, t)$ (described by **Eq. (1)**), with boundary conditions given by **Eq. (2)**) and parameters $K = 1$ and $l = 1$. The initial condition $f(x) = e^{-100(x-0.5)^2}$ (see **Eq. (16)**) was plotted using a dashed black line. The numerical approximations for $u(x, t)$ were plotted for a variety of t values ($t = 0$ (blue line), $t = 0.002$ (yellow line), $t = 0.005$ (green line), $t = 0.01$ (red line), $t = 0.05$ (purple line) and $t = 0.1$ (brown line)). It was computed using the finite-difference method developed in the previous section with $h = 0.01$, $\varepsilon = 10^{-5}$, and ten thousand time steps. The plot was produced using Python.

Both figures confirm the accuracy of the finite difference method to our previous results and physical understanding. With this in mind, we now apply our method to one further initial condition.

Extension

In our report, we solved the heat equation with Dirichlet boundary conditions and three different types of initial conditions. We first solved it analytically for a step function, then analytically and numerically for a simple symmetric continuous function, and finally for a Gaussian function. We will now consider using a harmonic function for the initial condition; applying the finite difference method will again solve this. We will call this final initial condition (d), and its general form will be represented by **Eq. (17)**. We have considered four forms of this initial condition: $m = 1, m = 2, m = 3$, and $m = 4$.

$$f(x) = \sin\left(\frac{m\pi x}{l}\right) \text{ for } m \in \mathbb{N} \quad (17)$$

Fig. (7) illustrates the numerical approximations for the solutions to the heat equation, with the initial conditions $f(x) = \sin\left(\frac{m\pi x}{l}\right)$ (for $m = 1, 2, 3, 4$), and boundary conditions described by **Eq. (2)**. The parameters used for the numerical method and $u(x, t)$ are specified in the figure caption. As expected in all four cases, we see heat dissipation over time, with the amplitude of the oscillations decreasing exponentially. We notice that the decay rate increases with the value m . This agrees with the theory beyond this report, which shows the solution to this equation $u(x, t)$ is proportional to $e^{-(\frac{m\pi}{l})^2 t}$. Thus, a larger m causes a quicker heat dissipation of the rod. We also notice the value of m characterizes the number of oscillations produced by our curves; this again makes sense as the frequency of oscillations for an equation of the form **Eq. (17)** is characterized by the value of m .

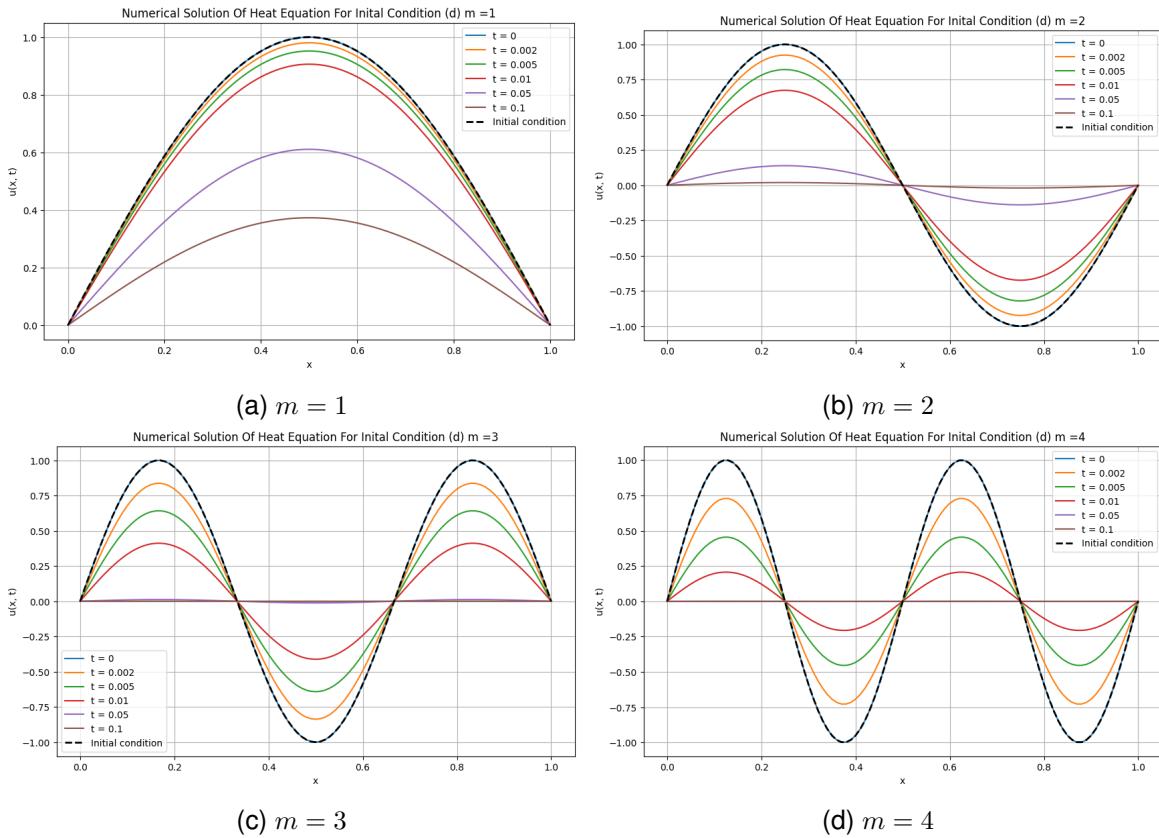


Figure 7: Numerical approximations for the solutions to the heat equation $u(x, t)$ (described by **Eq. (1)**), with boundary conditions given by **Eq. (2)**, and parameters $K = 1$ and $l = 1$. The initial conditions $f(x) = \sin(\frac{n\pi x}{l})$ (in the cases $m = 1, 2, 3, 4$ as labeled see **Eq. (17)**) was plotted using a dashed black line. The numerical approximations for $u(x, t)$ were plotted for a variety of t values ($t = 0$ (blue line), $t = 0.002$ (yellow line), $t = 0.005$ (green line), $t = 0.01$ (red line), $t = 0.05$ (purple line) and $t = 0.1$ (brown line)). It was computed using the finite-difference method developed in a previous section with $h = 0.01$, $\varepsilon = 10^{-5}$, and ten thousand time steps. The plot was produced using Python.

Conclusion

This report investigated the one-dimensional heat equation under Dirichlet boundary conditions using analytical and numerical methods. First, the Fourier series and approximation were used to solve the equation for various initial conditions, revealing effects such as the Gibbs phenomenon and confirming improved accuracy for smooth functions.

The finite-difference method was then applied, which closely matched the analytical results and accurately captured our physical intuition for heat diffusion over time. We then extended the report using harmonic initial conditions and demonstrated that the faster the initial condition oscillates, the more rapid the decay.

Moving forward, we could generate complete analytical solutions for our extension, allowing us to understand our results completely. We may also consider further initial conditions, such as a triangular pulse or a combination of harmonic functions. All of which led to insightful results.