Numerical Solution of the Heat Conductance Equation

Yecheng Chu
April 2021

Introduction

In the experiment I explored the solution of the heat conductance equation (Equation 1) when modelling the temperature distribution on a bar shown in Fig 1.

$$\frac{\partial^2 T(x,t)}{\partial x^2} = \frac{1}{\kappa}\frac{\partial T(x,t)}{\partial t} \qquad (1)$$

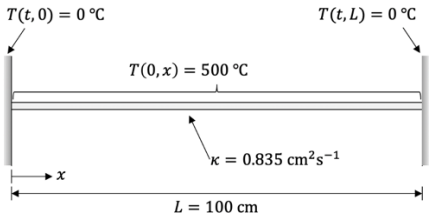


Fig 1: Problem description

The solution used is based on finite difference approximations of derivatives. In the experiment, both explicit and implicit approaches have been adopted. In order to compare their accuracy, an analytical solution is computed using Equation 2.

$$T(x,t) = \frac{2000}{\pi}\sum_{n=1}^{\infty}\frac{1}{n} sin\frac{n\pi x}{L} exp^{-\frac{n^2\pi^2\kappa}{L^2}t} \quad for\ n = 1, 3, 5, \ldots, \infty \qquad (2)$$

Explicit Solution

First, an explicit method is used. The first order derivative part of the equation obtained via forward difference approximation is expressed in Equation 3:

$$\frac{\partial T(x,t)}{\partial t} = \frac{T(i, j+1) - T(i,j)}{\Delta t} \qquad (3)$$

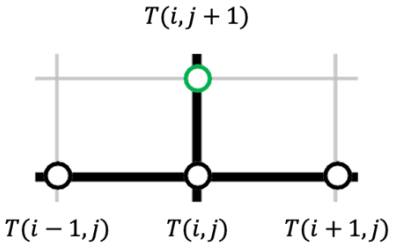
The second order derivative part of the equation is obtained via central difference approximation shown below:

$$\frac{\partial^2 T(x,t)}{\partial x^2} = \frac{T(i-1, j) - 2T(i,j) + T(i+1,j)}{\Delta x^2} \qquad (4)$$

By substituting those formula into Equation 1, I can get a formula. Then I rearrange the formula putting the unknown value $T(i, j+1)$ to the left of the equal sign and all other known values to the right. I would get the update scheme (Equation 5).

$$T(i, j+1) = \gamma T(i-1, j) + (1 - 2\gamma) + \gamma T(i+1, j) \quad where\ \gamma = \frac{\kappa \Delta t}{\Delta x^2} \qquad (5)$$

A computational molecule is shown in Fig 2. In this molecule, all the black nodes are known and are used in Equation 5 to calculate the value of the green unknown node. Update equations are based on Equation 5, which moves the molecule along the bar and do the update for all the unknown points on the row.



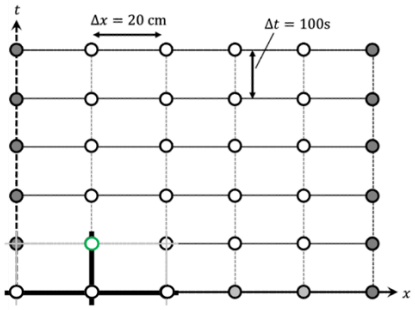


Fig 2 (left): Computational molecule used for explicit approach
Fig 3 (right): Explicit approach implementation

The first step in the implementation is to set up an empty grid based on Δt and Δx. For example, the grid used for this case when $\Delta t = 100s$ and $\Delta x = 20cm$ has the shape of 7 rows and 6 columns. Then the grid needed to be initialized with all values at the last row set to 500 except those boundary points. All values in the left and right boundary of the grid are set to 0. Then a nested loop is used to put the molecule in Fig 2 into the grid to solve the unknown points one by one from bottom left to top right using Equation 5 with different known points. Fig 3 shows how the molecule is used to solve the first unknown point $T(i, j+1)$, after calculating this point, the molecule is moved to right to calculate $T(i+1, j+1)$. It keeps looping until all the points in $j+1$ row are calculated. Then the molecule is moved up to calculate $j+2$ row using the same approach. Finally, all the rows are looped and all points in the grid will be calculated.

The temperature distribution graph is shown in Fig 4. It shows those points at the two ends of the bar have a rapid temperature drop at the beginning but those in the middle does not show a significant drop in temperature. It indicates time is needed to conduct the temperature. However, the two points at the middle remains to be 500°C at $t = 100s$ that is not very reasonable. This error occurs because when I update the temperature of these two points using the computation molecule, the three points I used when $t = 0$ are all in the middle which are 500°C. As a result, it gives an over-estimate of the temperature. Another observation is that this figure is symmetric. This is because the temperatures at two ends are the same and the points I sampled are symmetric in position on the bar.

Fig 5 shows the evolution of temperature with time at $x = 20cm$ using three different Δt and Δx sets. Since the $\Delta t = 100s$ and $\Delta x = 10cm$ case is unstable, I also plot the graph without this case to show the comparison of changing time step size of the grid. In order to get more accurate results, the analytic solution for reference is plotted using a Δt of $50s$.

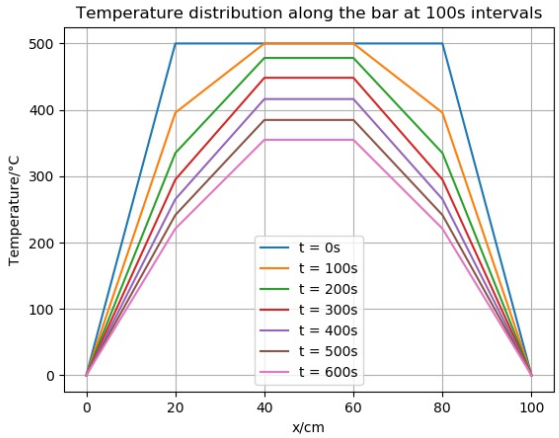


Fig 4: Temperature distribution graph

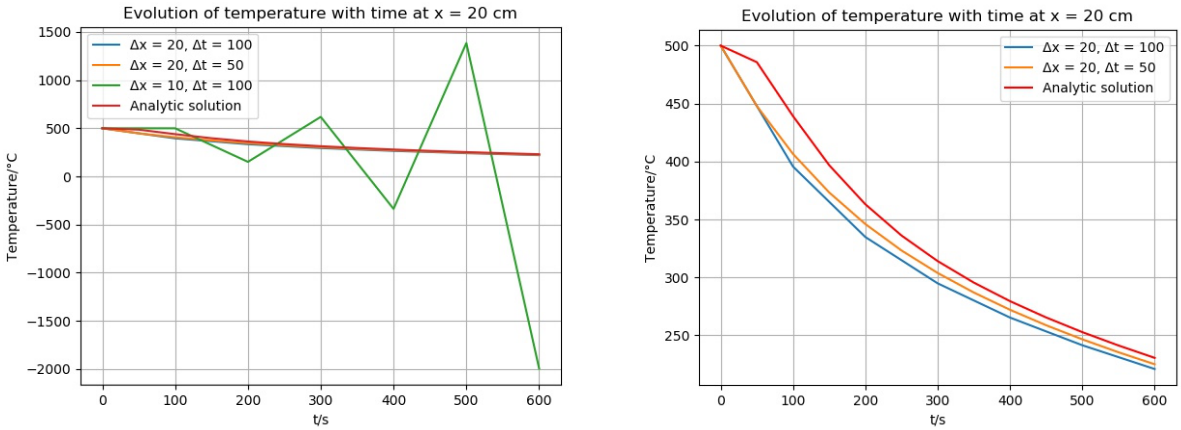


Fig 5: Evolution of temperature at a fixed point

The temperatures at $t = 600s$ and $x = 200cm$ for the three different cases are recorded and have their absolute and relative errors calculated in Table 1. According to both Fig 5 and Table 1, the case when $\Delta t = 100s$ and $\Delta x = 10cm$ is unstable and hence got the worst result. It is shown indicated in lecture notes that for the explicit solution to be stable $\Delta t \leq \frac{\Delta x^2}{2\kappa}$ should be guaranteed. For this case Δt should be smaller or equal to $\frac{10^2}{2\times 0.835} = 59.88$. Clearly, this case is unstable. For the other two cases, Δt should be smaller or equal to 239.52, and the case where $\Delta t = 50s$ has a smaller Δt, so it gets the best result.

Condition	Explicit result	Analytic result	Absolute error	Relative error
$\boldsymbol{\Delta t = 100s\ \Delta x = 20cm}$	220.96cm	230.58cm	9.62cm	4.17%
$\boldsymbol{\Delta t = 50s\ \ \Delta x = 20cm}$	225.05cm		5.53cm	2.40%
$\boldsymbol{\Delta t = 100s\ \Delta x = 10cm}$	-1995.66cm		2226.24cm	965.50%

Table 1: Comparison for different step sizes using explicit method when $t = 600s\ x = 20cm$

Implicit Solution

The implicit solution uses Crank-Nicolson method which the mid-point of the time step is used to do the evaluation. The first order derivative at the mid-point is calculated use a central difference approach which can be express as:

$$\frac{\partial T(x,t)}{\partial t} = \frac{T(i,j+1) - T(i,j)}{2 \times \frac{\Delta t}{2}} = \frac{T(i,j+1) - T(i,j)}{\Delta t} \qquad (6)$$

The second order derivative at the mid-point can be evaluated as the mean of the approximations at the beginning and end of the time interval:

$$\frac{\partial^2 T(x,t)}{\partial x^2} = \frac{1}{2}\left(\frac{T(i-1,j) - 2T(i,j) + T(i+1,j)}{\Delta x^2} + \frac{T(i-1,j+1) - 2T(i,j+1) + T(i+1,j+1)}{\Delta x^2}\right) \qquad (7)$$

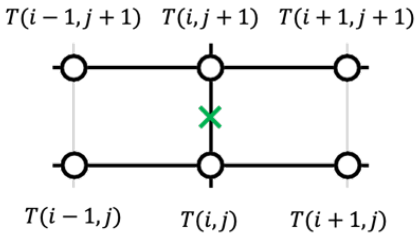
Substitute the terms in Equation 1 using Equation 6 and 7, I can get:

$$\frac{1}{2}\left(\frac{T(i-1,j) - 2T(i,j) + T(i+1,j)}{\Delta x^2} + \frac{T(i-1,j+1) - 2T(i,j+1) + T(i+1,j+1)}{\Delta x^2}\right) = \frac{1}{\kappa}\frac{T(i,j+1) - T(i,j)}{\Delta t} \qquad (8)$$

Then I put the terms involving unknown (points at $j+1$ row) to the left and other known terms to the right. After the rearrangement, the formula looks like this:

$$\frac{\gamma}{2}T(i-1,j+1) - (1+\gamma)T(i,j+1) + \frac{\gamma}{2}T(i+1,j+1) = -\frac{\gamma}{2}T(i-1,j) + (\gamma-1)T(i,j) - \frac{\gamma}{2}T(i+1,j) \quad where\ \gamma = \frac{\kappa \Delta t}{\Delta x^2} \qquad (9)$$

A molecule used for this method is shown in Fig 6. The mid-point used to build Equation is shown by a green cross. The three points below it are the three known points on row j and the three unknown points are on row $j+1$ above the mid-point. In the code, I solve the grid row by row use the similar technique as that in explicit method. The molecule is moved along the bar and the values are recorded using the update scheme shown in Equation 9. Unlike explicit method where there is only one unknow points in the equation, in this molecule three unknown points are needed, except at the two ends, to calculate the mid-point. So, we cannot get the answer directly. The method I choose is to collect nodal equations and formulate as a matrix $AT = b$ in each row of the grid and solve the tri-diagonal system for T.



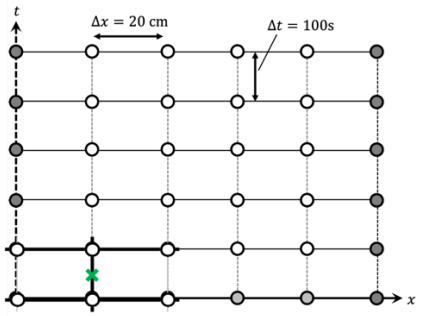


Fig 6 (left): Computational molecule used for implicit approach
Fig 7 (right): Implicit approach implementation

$$\begin{bmatrix} -(\gamma+1) & \frac{\gamma}{2} & 0 & 0 \\ \frac{\gamma}{2} & -(\gamma+1) & \frac{\gamma}{2} & 0 \\ 0 & \frac{\gamma}{2} & -(\gamma+1) & \frac{\gamma}{2} \\ 0 & 0 & \frac{\gamma}{2} & -(\gamma+1) \end{bmatrix} \begin{Bmatrix} T(i,j+1) \\ T(i+1,j+1) \\ T(i+2,j+1) \\ T(i+3,j+1) \end{Bmatrix} = \begin{Bmatrix} -\frac{\gamma}{2}T(i-1,j)+(\gamma-1)T(i,j)-\frac{\gamma}{2}T(i+1,j) \\ -\frac{\gamma}{2}T(i,j)+(\gamma-1)T(i+1,j)-\frac{\gamma}{2}T(i+2,j) \\ -\frac{\gamma}{2}T(i+1,j)+(\gamma-1)T(i+2,j)-\frac{\gamma}{2}T(i+3,j) \\ -\frac{\gamma}{2}T(i+2,j)+(\gamma-1)T(i+3,j)-\frac{\gamma}{2}T(i+4,j) \end{Bmatrix}$$

Fig 8: The tri-diagonal system for the $j+1$ row

Fig 8 shows the tri-diagonal system for the $j+1$ row using the updating scheme in the form of $AT = b$. The implantation is exactly building the A and b part in this figure for each row. After building and initializing the grid as I did before, the code uses the molecule to loop along the bar like in Fig 7 to gather A and b from the row. The points at both ends are zero so there are only two terms in the first and last row of the A matrix in Fig 8. The terms for other rows in matrix A are exactly the same, just in different positions which is due to the moving of the molecule. The terms in b are also similar, it just changes the known points under the molecule when it is moving along the bar. As a result, this method is generally appliable when the Δx changes. When calculating the $j+1$ row, the boundary and initial conditions can be used in b, such as $T(i-1,j) = 0$ and $T(i,j) = 50$. After looping through the bar, I get A and b, then I use *numpy. linalg.solve()* to solve for T. This method is used to solve a linear matrix equation, or system of linear scalar equations[1], and it is tested for simple cases before being implemented to solve T. After that, I move the molecule above and solve for another row using the same method until all the rows are looped and all the temperatures could be solved.

Fig 9 shows the temperature distribution graph and the evolution graph with different Δt and Δx sets for implicit method. It is shown that, the implicit approach using Crank-Nicolson method performs better than the explicit method. This is because in explicit method, each unknown point can be solved by just the three points below it, however, in implicit method, all the unknown points on the same row needed to be solved simultaneously so that all the information at a particular time have been incorporated into the solution.

[1] Documentation: https://numpy.org/doc/stable/reference/generated/numpy.linalg.solve.html

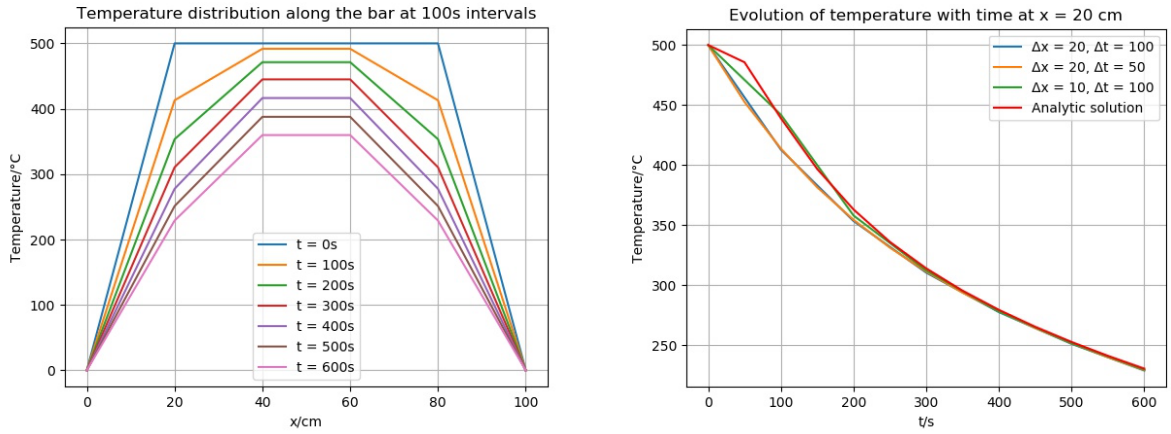


Fig 9: Graphs for implicit method

Condition	Implicit result	Analytic result	Absolute error	Relative error
$\Delta t = 100s \ \Delta x = 20cm$	228.96cm	230.58cm	1.62cm	0.7%
$\Delta t = 50s \ \ \Delta x = 20cm$	229.32cm		1.26cm	0.55%
$\Delta t = 100s \ \Delta x = 10cm$	229.71cm		0.87cm	0.38%

Table 2: Comparison for different step sizes using implicit method when $t = 600s \ x = 20cm$

The comparison between different step sizes used when $t = 600s$ and $x = 20cm$ is shown in Table 2. The results show that for implicit approach, both minimizing the step size of spacing and the step size of time can lead to more accurate results. As the implicit approach is unconditionally stable, it enables a larger time step to be taken for a given accuracy.

Modelling initial temperature change

In this section, the initial temperature changes over the first 60 seconds are modelled. Both explicit and implicit approaches are adopted for comparison. I chose a variety of time step sizes and spacing step sizes to do the experiment. For each method, I record the average of all absolute errors for each time step at the point $x = 20cm$. I also record the time to calculate the temperature of the grid for each method in order to evaluate the computational effort. Though the time taken to calculate the temperature varies a little bit each time, it still gives a good indication of the amount of computation. Different combinations of Δt and Δx have been used and some of the results are shown below in Table 3.

Condition x=20cm	Explicit result avg. error / cm	Explicit time / ms	Implicit result avg. error / cm	Implicit time / ms
$\Delta t = 20s \ \Delta x = 0.5cm$	7.58	0.66	0.67	3.21
$\Delta t = 20s \ \Delta x = 1cm$	7.58	0.33	0.62	1.48
$\Delta t = 20s \ \Delta x = 2cm$	7.58	0.21	0.49	0.82
$\Delta t = 20s \ \Delta x = 5cm$	7.58	0.12	1.69	0.41
$\Delta t = 10s \ \Delta x = 0.5cm$	6.71	1.23	0.19	7.93
$\Delta t = 10s \ \Delta x = 1cm$	6.71	0.62	0.16	2.86
$\Delta t = 10s \ \Delta x = 2cm$	6.71	0.41	0.27	1.38

$\Delta t = 10s\ \Delta x = 5cm$	0.72	0.14	1.97	0.69
$\Delta t = 5s\ \Delta x = 0.5cm$	6.26	2.50	0.04	15.66
$\Delta t = 5s\ \Delta x = 1cm$	6.26	1.22	0.07	5.31
$\Delta t = 5s\ \Delta x = 2cm$	4541.48	0.59	0.33	2.50
$\Delta t = 5s\ \Delta x = 5cm$	0.96	0.25	2.07	1.22

Table 3: Comparison for different step sizes using explicit and implicit methods when $x = 20cm$

According to the results shown, the implicit method generally performs better than the explicit method. So, when choosing the optimal Δt and Δx values, the implicit method is considered. The results shown in Table 3 indicates that for the implicit method, the smallest Δt and Δx value gives best accuracy, however, it also takes longest time to compute. The Δt of my choice is $10s$ as it gives a good balance between accuracy and computation time. The Δx is chosen to be $1cm$, as from $1cm$ to $0.5cm$, the computation time doubles but the improve in accuracy is very limited. While from $2cm$ to $1cm$, the error reduces 40% when the computation time doubles. Fig 10 shows the comparison of the three Δx values when $\Delta t = 10s$, the plot on the bottom is the chosen Δx value. Note that, these plots are not drawn on the same graph because these lines are too close to each other and cannot be distinguished if shown on the same graph. It shows $\Delta x = 1cm$ for implicit method gives a quite good approximation. Finally, the appropriate choice according to my experiment is $\Delta t = 10s$ and $\Delta x = 1cm$.

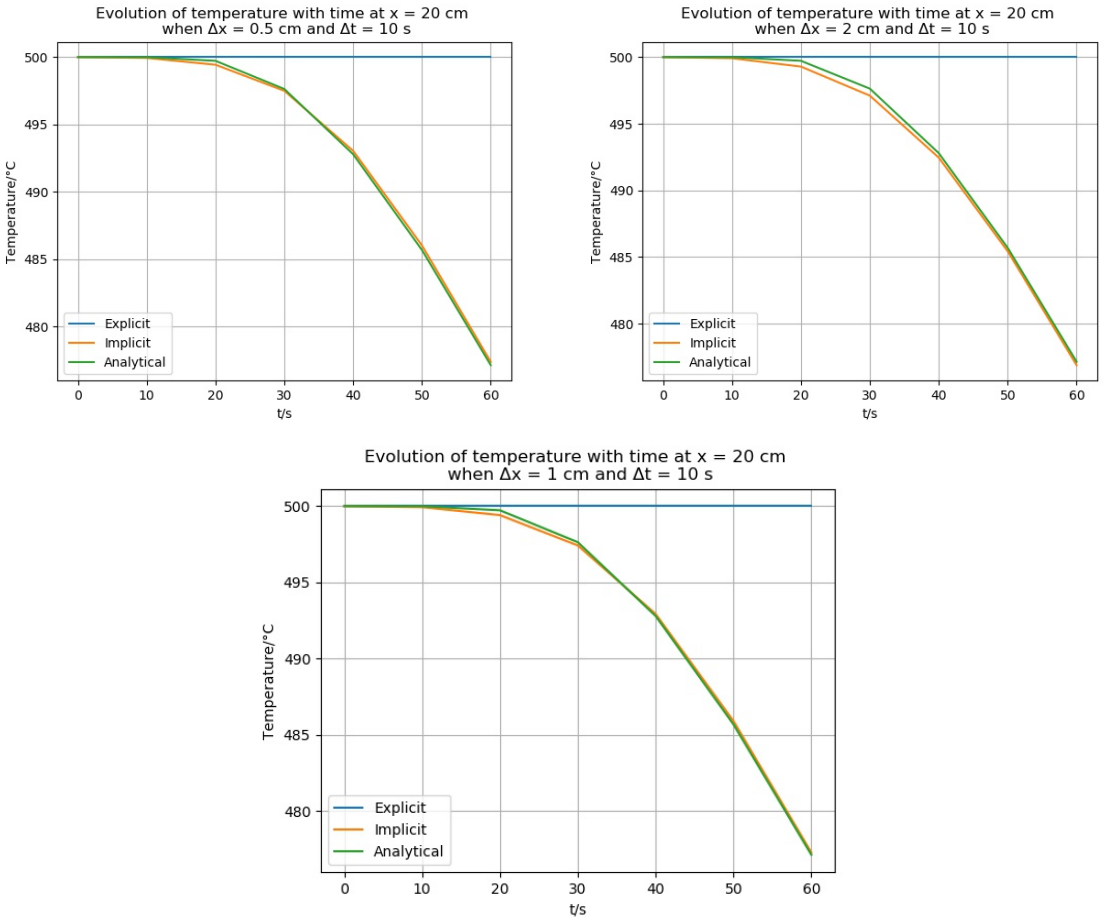


Fig 10: Comparison between three Δx values when $\Delta t = 10s$

Conclusion

In conclusion, in this lab experiment, both the explicit and implicit approaches have been adopted to model the heat conductance equation. In general, the implicit approach has a better performance, and it will not be affected by stability as it is unconditionally stable. However, a drawback of this approach is that it takes more time to compute the solutions. In the last part of the experiment, the optimal value of Δt and Δx have been explored using the implicit approach in order to balance the accuracy and computation time. After comparing the results from different experiments, the compromised value set is chosen to be $\Delta t = 10s$ and $\Delta x = 1cm$.