COMP36212 EX2
Numerical Solution of the Heat Conduction Equation

Paul Vautravers
10461739
The University of Manchester

March 26, 2023

Abstract

The one-dimensional heat conduction equation was used to model the problem of heat transport in a metallic rod, during cooling. The heat conduction equation, a parabolic partial differential equation, was solved numerically and analytically for simple boundary and initial conditions. The implicit solver, namely the Crank-Nicolson (CN) scheme, was shown to remedy the issues of stability present with the explicit approach. Consequently, the CN scheme was used to approach a problem where no analytical solution is available and was demonstrated to converge to an exact solution as expected by the Lax convergence theorem.

1 Introduction

The conduction of heat in solids is a widespread phenomenon, whose solution lead to a large volume of novel mathematical techniques and insights into other, seemingly unrelated fields to the original physical problem. The governing equation for heat conduction was first formulated in 1807 by Jean Baptiste Joseph Fourier as the following partial differential equation (PDE):

$$\frac{\partial^2 T}{\partial x^2} = \frac{1}{\kappa}\frac{\partial T}{\partial t}, \tag{1}$$

where $T(x,t)$ is the temperature of the rod at a given position,x, and time,t, and κ is the thermal conductivity of the material, assumed to be constant. Equation 1 is part of a broad class of PDEs known as parabolic PDEs that occur in transient flow problems and have a first order derivative in time proportional to a number of 2nd order spatial derivatives. Fourier's great step in solving this equation was his move away from a description based on discrete bodies, as supported by his contemporaries and predecessors, and towards a continuum approach. In this manner he formulated the transport of heat in the interior of the solid in terms of: heat transport in space, storage of heat in a finite volume and boundary conditions that are assumed to be known. The variables corresponding to heat conductivity and heat capacity have been absorbed into a single constant κ in equation 1, simplifying the problem enormously. Fourier used separation of variables and infinite trigonometric series expansions to develop an analytical solution. This project explores application of the heat conduction equation to modelling the cooling of a metallic rod, length $L = 100$ cm, for 600 seconds after it has been heated and then plunged into cool reservoirs. This situation is reduced into boundary and initial conditions that $T(x = 0, t) = T(x = L, t) = 0$ °C, where L is the length of the bar, and $T(x, t = 0) = 500$ °C. Following Fourier's approach the analytical solution can be found:

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

Despite not reaching a wider audience for 15 years after Fourier's initial publication, it was soon applied to a diverse range of problems including electricity and economics, indicating the significance of being able to resolve this equation for more complex problem conditions [1].

Analytical methods can provide an exact solution to PDEs. However, such solutions only exist for a very restricted range of physical scenarios that can be represented with very simple initial and boundary conditions. Numerical methods circumvent this problem and can deal with a far wider range of problems, making them particularly useful for applications in physics, engineering and derived fields, Generally, numerical rely on discretising the problem space at hand, enabling them to be approximated and solved on computers. It is customary to use the finite difference (FD) approach for solving differential equations. The replacement of exact derivatives with FD approximations allows the development of a finite algebraic system of equations which can be resolved easily with a computer [2]. Forward and backward FD approximations are one-sided and produce first order approximations to the true derivative - the size of the error is on the order of the discrete step size used. Taking the average of the two aforementioned, one-sided approximations gives a centred approximation which yields second order accuracy. As in Leveque's authoritative work on FD methods, we will explore the heat conduction equation through an implicit and explicit finite difference method.

2 Theory

2.1 Explicit Approach

An FD method is considered explicit if it can be marched forward in time, using only previous values in the time discretisation. As Leveque points out, the heat conduction equation can be solved forward in time, allowing such explicit methods to be used. Using a centred difference in space and a forward difference in time, the explicit difference approximation to equation 1, using $T(x,t) = \phi_{(i,j)}$, where i is the position index and j the time index, may be written as:

$$\phi_{i,j+1} = \frac{\kappa \Delta t}{\Delta x^2}(\phi_{(i-1,j)} + \phi_{i,j}(\frac{\Delta x^2}{\kappa \Delta t} - 2) + \phi_{(i+1,j)}). \tag{3}$$

Information on the accuracy of this approach can be gleaned through the use of a Taylor series expansion around the exact solution, a smooth function $U(x,t)$. Comparing this Taylor expansion and the approximation above, we can find the order of residual terms of Δt and Δx, a method called local truncation analysis. Following this analysis, we find that the local truncation error (LTE) is given by: $LTE\ O(\Delta x^2 + \Delta t)$. As such, the method is second order accurate in space and first order accurate in time. This requires that $\Delta t \sim \Delta x^2$ in order to achieve comparable accuracy in time as in space for this method. Often, however, more stringent requirements on step size derive from stability considerations as opposed to accuracy. Simply, stability means that a numerical method does not amplify errors - a solution scheme is stable if its stability condition can be satisfied, perhaps with a restriction on step size [3]. The stability of the explicit scheme above is governed by the condition $\frac{\Delta t}{\Delta x^2} \leq \frac{1}{2}$, placing an impractical restriction on the time discretisation to produce reasonable solutions [2].

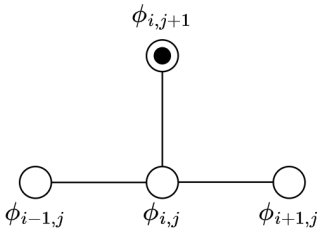
2.2 Implicit Approach: The Crank-Nicolson Method

Alternatively, the problem can also be resolved through an implicit numerical method, which cannot be marched forward in time with only the previous values in time. One such method is the Crank-Nicolson (CN) scheme, which uses a central difference approximation to the derivative at the midpoint of the time interval [4]. This produces a series of simultaneous equations governing the updates for each interior node, from j to $j+1$. These have form:

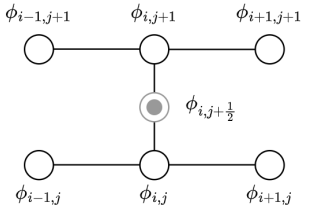
$$\phi_{i,j+1} = \phi_{i,j} + \frac{\Delta t}{2\Delta x^2}(\phi_{i-1,j} - 2\phi_{i,j} + \phi_{i+1,j} + \phi_{i-1,j+1} - 2\phi_{j+1,i} + \phi_{i+1,j+1}). \tag{4}$$

This can be solved as a matrix equation of the following form:

$$\begin{bmatrix} (1+2r) & -r & 0 & 0 \\ -r & (1+2r) & -r & 0 \\ 0 & -r & (1+2r) & -r \\ 0 & 0 & -r & (1+2r) \end{bmatrix} \begin{bmatrix} \phi_{1,j+1} \\ \phi_{2,j+1} \\ \phi_{3,j+1} \\ \phi_{4,j+1} \end{bmatrix} = \begin{bmatrix} r(\phi_{0,j} + \phi_{0,j+1}) + (1-2r)\phi_{1,j} + r\phi_{2,j} \\ r\phi_{1,j} + (1-2r)\phi_{2,j} + r\phi_{3,j} \\ r\phi_{2,j} + (1-2r)\phi_{3,j} + r\phi_{4,j} \\ r(\phi_{5,j} + \phi_{5,j+1}) + (1-2r)\phi_{4,j} + r\phi_{3,j} \end{bmatrix}, \tag{5}$$



(a) The computational molecule for the explicit update scheme, as in equation 3, with the $\phi_{i,j+1}$ in black.



(b) The computational molecule for the implicit CN update scheme, as in equation 4, with the time interval midpoint node $\phi_{i,j+\frac{1}{2}}$ in grey.

Figure 1: Computational molecules for the explicit and Crank-Nicolson update schemes

where $r = \frac{\Delta t}{2\Delta x^2}$ [2]. We observe for the simple equation above, with a system of 4 interior nodes, that the matrix is tridiagonal, allowing it to be solved efficiently with algorithms such as the Thomas algorithm. As the CN scheme uses a centred difference for the time step, it achieves higher accuracy in time than the explicit approach, with local truncation error $LTE\ O(\Delta t^2 + \Delta x^2)$. The implicit CN scheme , however, finds its greatest advantage over the explicit method in its stability. Von Neumann stability analysis can be applied to the CN scheme to demonstrate the simple stability condition that $\Delta t > 0$. This allows $\Delta t = \Delta x$ to be used, ensuring equivalent accuracy in time and space, without risking instability [5] [2].

3 Explicit Method: Implementation and Results

The single spatial dimension and time dimension allow the problem to be considered in terms of a 2D grid of discrete nodes. A subset of such a grid is shown in figure 1a, illustrating the computational molecule for the explicit update scheme. Python was used to program this project, which in particular enabled ease of generating matrix systems, but which was likely slower than an implementation in C or C++. The 2D $t-x$ grid was initialised using `np.zeros(N_t, N_x)`, with the $j = 0$ row being first set to the initial conditions of $T = 500$ °C, through indices [1,N_x -2] (applying only to the internal nodes). The exterior nodes through all j indices were set to $T = 0$ °C to reflect the cooled boundary conditions. Then, a simple update function was formulated according to 3, written in terms of ϕ_-,ϕ and ϕ_+ and then integrated into a larger function ,`explicit_temp_iter`, that iterates over rows (time) and then over the columns (space), updating the value $\phi_{i,j+1}$ at each iteration using `explicit_update`. The boundary and initial conditions are taken into account without further implementation, as the first interior node can call the value of the left boundary, while iterations can start from $j = 0$ (initial conditions) to generate subsequent values. As such, it was possible to produce plots of the temperature distribution across the bar at different time and position steps given in figure 2. The configuration Δt=100, Δ=20, giving 4 interior nodes (6 total in x) and 7 total time nodes, produces a physical seeming distribution. Decreasing Δt to 50s, we maintain a similar distribution and expect a reduction in error, supported by figure 3. Furthermore, if we consider the error evaluated at t=600 from figure 3, presented in table 1, we see that the error approximately halves between Δ = 100 and Δ=50, expected given the method's first order error in time.

We immediately note from 2 halving the spatial step size does not produce a physical solution. Rather,Δt =100 and Δx = 10 has produced enormous oscillations in Temperature. Furthermore, as expected with an unstable method, these errors grew with time, as shown by the size of oscillations in figure 2, though not presented in figure 3 to not dwarf the errors from the other curves. This, however, was expected, given we have failed to satisfy the stability condition, with $\frac{\Delta t}{\Delta x^2} = 1$.

Δt	Δx	$T(x = 20, t = 600)$	Absolute Error
100.0	20.0	220.962066	9.614814
50.0	20.0	225.046963	5.529917
100.0	10.0	-1995.656788	2226.233668

Table 1: Table providing the temperature numerically calculated at x=20cm and t=600s and the associated numerical error, calculated from the analytical solution.

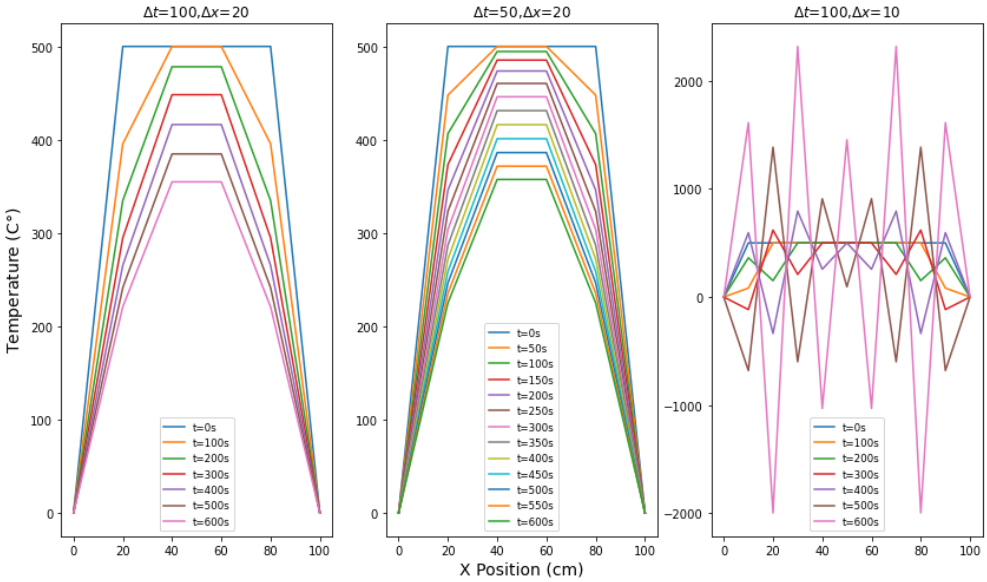


Figure 2: Temperature distribution across the metallic rod, with boundary conditions $T = 0$ °Cand initial value $T = 500$ °C for the interior nodes. The configuration of $\Delta x = 10$ yields an unstable solution that clearly is not physically reasonable

4 Crank-Nicolson Method: Implementation and Results

The update scheme for the implicit Crank-Nicolson approach can be understood with the aid of the computational molecule, shown in figure 1b. This is visibly more complicated than the explicit molecule and illustrates why solutions can not be marched as directly forwards in time as the previous method. The CN scheme solves the system of equations for the whole row of nodes simultaneously, allowing linear algebra to be used. An initialiser for the Crank-Nicolson system of equations was developed,`init_cn_matrix`, following the matrix provided in equation 5, using Python. Next, the Thomas algorithm was implemented to solve such a tridiagonal system. This involved producing a decomposed matrix of the original CN matrix, denoting the elements on the lower off-diagonal [α_2,α_N], the main diagonal [β_1,β_N] and the upper off-diagonal [γ_1,γ_N]. We then have the update rule that, iterating from $i = 2$ to $i = N$, $\alpha_i = \frac{\alpha_i}{\beta_{i-1}}$ and simultaneously $\beta_i = \beta_i - \alpha_i\gamma_{i-1}$. In the same iteration, The forward vector element s_i is replaced such that $s_i = s_i - \alpha_i s_{i-1}$. Finally, the backward substitution can be made to evaluate the new temperature vector,x. First, we have that $x_n = \frac{s_N}{\beta_N}$, before iterating through the remaining terms in reverse (from N-1 to 1) with update $x_i = \frac{s_i - \gamma_i x_{i+1}}{\beta_i}$. This concludes a brief description of the Thomas algorithm as implemented in this project. Whilst `init_cn_matrix` was only called once, the right hand side of equation 5, referred to as `rhs_vec` throughout the code, was updated each timestep. The function `init_rhs_vec` is called at every time step and assigns elements of the vector, corresponding to the interior nodes of the grid. As such, the initial values are incorporated through this formula, where $j = 0$ is used to find the $j + 1 = 1$ x vector with the Thomas algorithm.

The steps considered for explicit process were repeated for the implicit. These plots produced a similar structure to the explicit method for $\Delta t = 100$, Δx =20, as shown in figure 4. However, considering the error calculated from the fixed node $x = 20$cm at $t = 600$s, tabulated in table 2, we immediately see that the absolute error of the implicit method, 1.62, is almost an order less than that of the explicit method. This is anticipated, given CN is second order not first order in time, as the explicit approach is. Additionally, we observe that the third panel of figure 4 produces a stable solution with $\Delta x = 10$ and reveals finer structure to the temperature distribution. This stability is not unexpected, given that the implicit Crank-Nicolson method is unconditionally stable, for $\Delta t > 0$. The stability of this method is also illustrated through figure 5, as we can observe that the error does not grow with time.

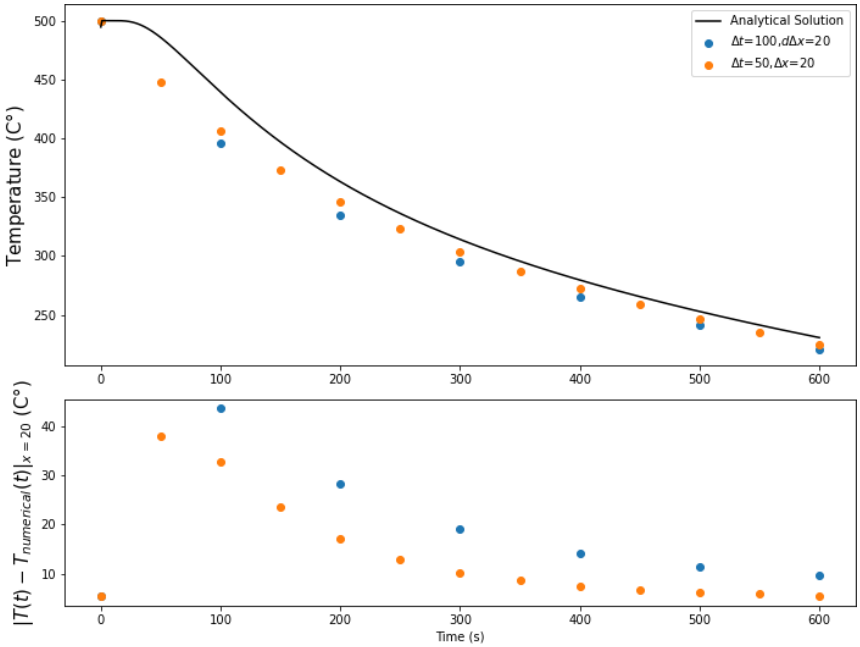


Figure 3: Variation of temperature in time, at fixed position x=20 cm. Comparison with the analytical solution from equation 2 yields the absolute error of the numerical method, shown in the lower panel. We see that the reduction in Δt has improved the error, as supported by table 1

Δt	Δx	$T(x = 20, t = 600)$	Absolute Error
100.0	20.0	228.955176	1.621704
50.0	20.0	229.317966	1.258914
100.0	10.0	229.712404	0.864476

Table 2: Table providing the temperature numerically calculated with the Crank-Nicolson scheme at x=20cm and t=600s and the associated numerical error, calculated from the analytical solution.

5 Application of Crank-Nicolson Scheme to Non-Analytical Problem

Following the implementation of the implicit and explicit methods to solve the heat conduction equation for simple boundary and initial conditions, we conclude that the explicit solution is not suitable for problems where an analytical solution is not available – without prohibitively small step sizes. Furthermore, while the CN method was more complicated to implement, it did not perform markedly slower. This conforms to expectation because, as Leveque points out, the CN update scheme can be solved for a system of m equations with order $O(m)$ work - allowing the CN method to be nearly as efficient per timestep as the explicit approach [2]. This was supported by timing the execution of the explicit and implicit grid solver function. The simple problem with initial conditions as in section 2 and 3 was now solved up till time t=10000 and x=1000, with both methods, 100 times. Each repeat timed the methods separately, allowing a mean time for both methods to be found as t_{exp}= 0.0278 ± 0.0070s and t_{imp} = 0.0761 ± 0.0132s. We observe from this that while the CN scheme is demonstrably slower, it is of the same order of magnitude as the explicit method. Given the explicit scheme requires Δt to be less than half Δx^2, reaching a similar accuracy as the CN scheme will require an order smaller (at least) time, increasing the scale of the problem system by an order. This negates the advantage that the explicit scheme had in computation time, given we have increased the system scale by an order of magnitude for a numerical method that was already only equivalent order in computation time to the more complex CN scheme.

For the last part of the project, cooling of the bar will be considered up to times t=60s, where the initial condition of the interior nodes is given by:

$$T(x, t = 0) = -xg(x - L) + c, \tag{6}$$

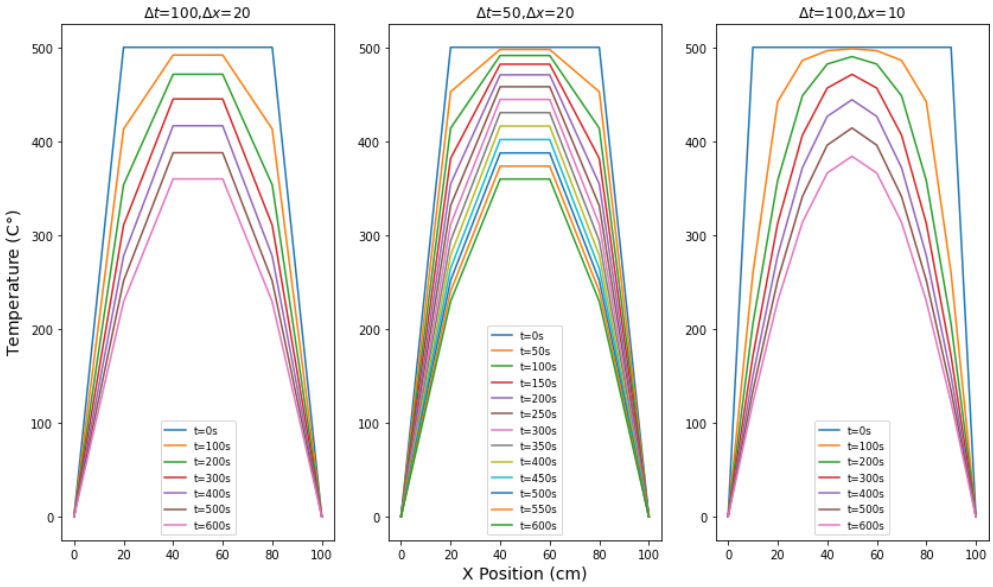


Figure 4: Implicit CN scheme: Temperature distribution across the metallic rod, with boundary conditions $T = 0$ °C and initial value $T = 500$ °C for the interior nodes, stable solution produced with $\Delta x = 10$

where L is the length of the bar as before, g and c are constants equivalent to 0.1 and 400 respectively. Using the Crank-Nicolson scheme as before, and now supplying the initial condition from equation 6 and that $T = 0$ at the boundaries, temperature distributions in time and space are produced and shown in figure 6.

In this case, we do not have an analytical solution for these initial conditions. However, by the Lax equivalence theorem we expect that for a consistent, linear numerical method it will converge to an exact solution if and only if it is stable. Generally, linear methods are only used when discretising linear PDEs - such as the heat conduction equation [2][5]. In this sense, should our implementation of the CN scheme converge to some function, that function will represent the exact solution in the limit that the step tends to zero. Given the CN scheme is unconditionally stable, we can expect this method to converge to some stable function before round off errors become significant. To verify this, $\Delta = \Delta x = \Delta$ t values were used to solve the early behaviour of the rod as before, till t = 60. The values of delta chosen were 20,10,5,2.5 until 0.078125. The difference in the value of the temperature at x=20, at time = 600, was calculated between each successive choice of step and normalised according to the difference for the initial values of Δ, 20 and 10. This series was evaluated according to:

$$\frac{T(\Delta_1) - T(\Delta_0)}{T(\Delta_1) - T(\Delta_0)}, \frac{T(\Delta_2) - T(\Delta_1)}{T(\Delta_1) - T(\Delta_0)}, \frac{T(\Delta_3) - T(\Delta_2)}{T(\Delta_1) - T(\Delta_0)}, ..., \frac{T(\Delta_N) - T(\Delta_{N-1})}{T(\Delta_) - T(\Delta_0)} \tag{7}$$

Plotting the numerical solutions over all time values at x = 20, as in previous sections, we find that the solutions appear to converge, as shown in the upper panel of figure 7, with curves for $\Delta = 1.25$ and $\Delta = 0.078125$ already near indistinguishable. The normalised difference in temperature evaluated at x=20cm and t=60s between successive steps is shown in this panel, with a clear linear decrease against step size, when plotted on double logarithmic axes. This verifies the previous expectation that stability of the CN scheme would guarantee a convergent solution for the heat conduction equation, given it is a linear PDE. As such, we have shown that the CN scheme converges for this problem, evidencing the accuracy of the outputted solution in the absence of an analytical solution.

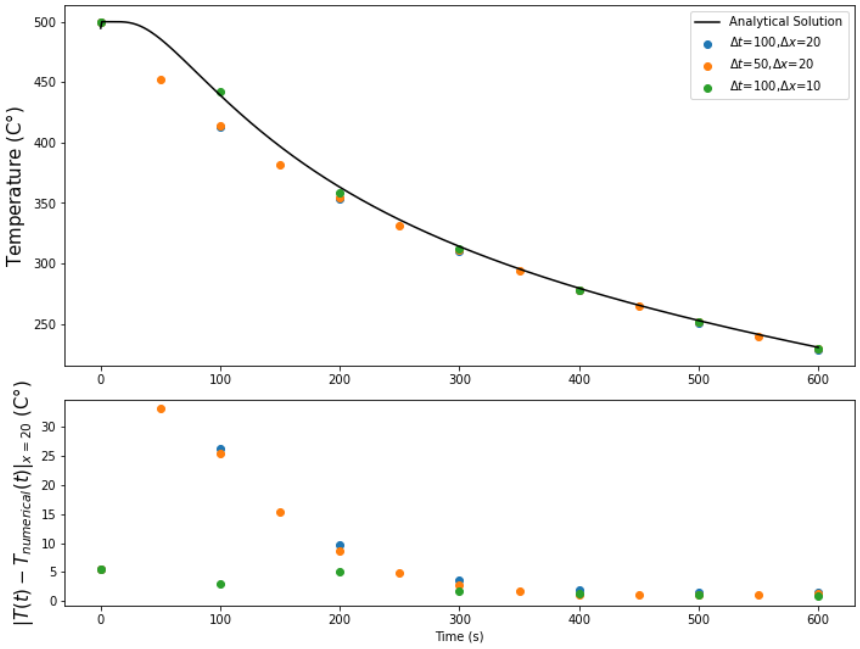


Figure 5: Variation of temperature in time, at fixed position x=20 cm. Comparison with the analytical solution from equation 2 yields the absolute error of the numerical method, shown in the lower panel. We see that the reduction in Δt has improved the error, as supported by table 2

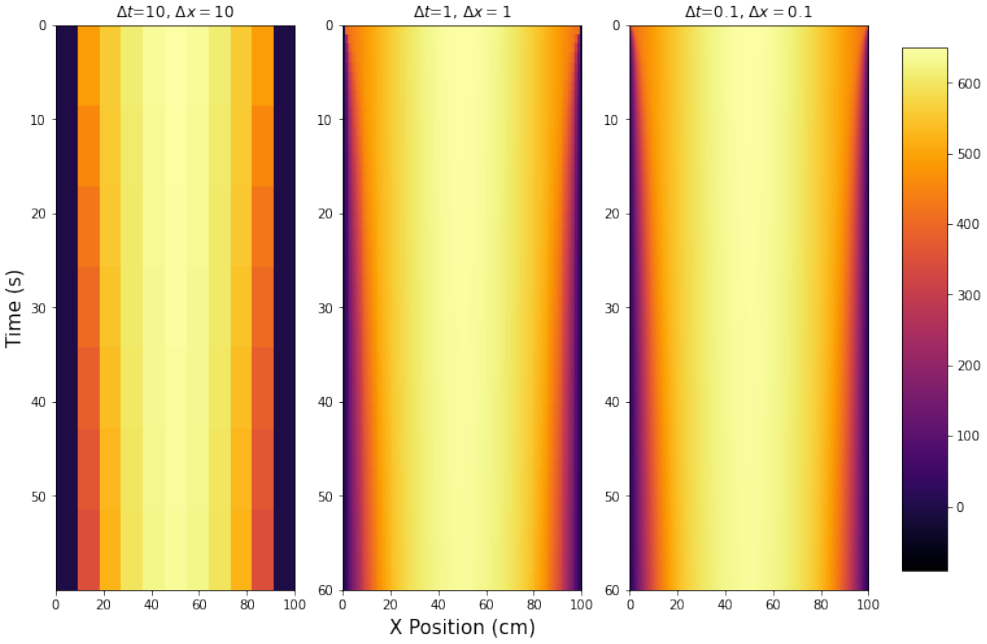


Figure 6: Variation of temperature in time and space, for initial conditions given by equation 6 and boundary conditions of T=0 at each exterior node. We note that the CN scheme continues to produce physical results for Δt=Δx=0.1 as before.

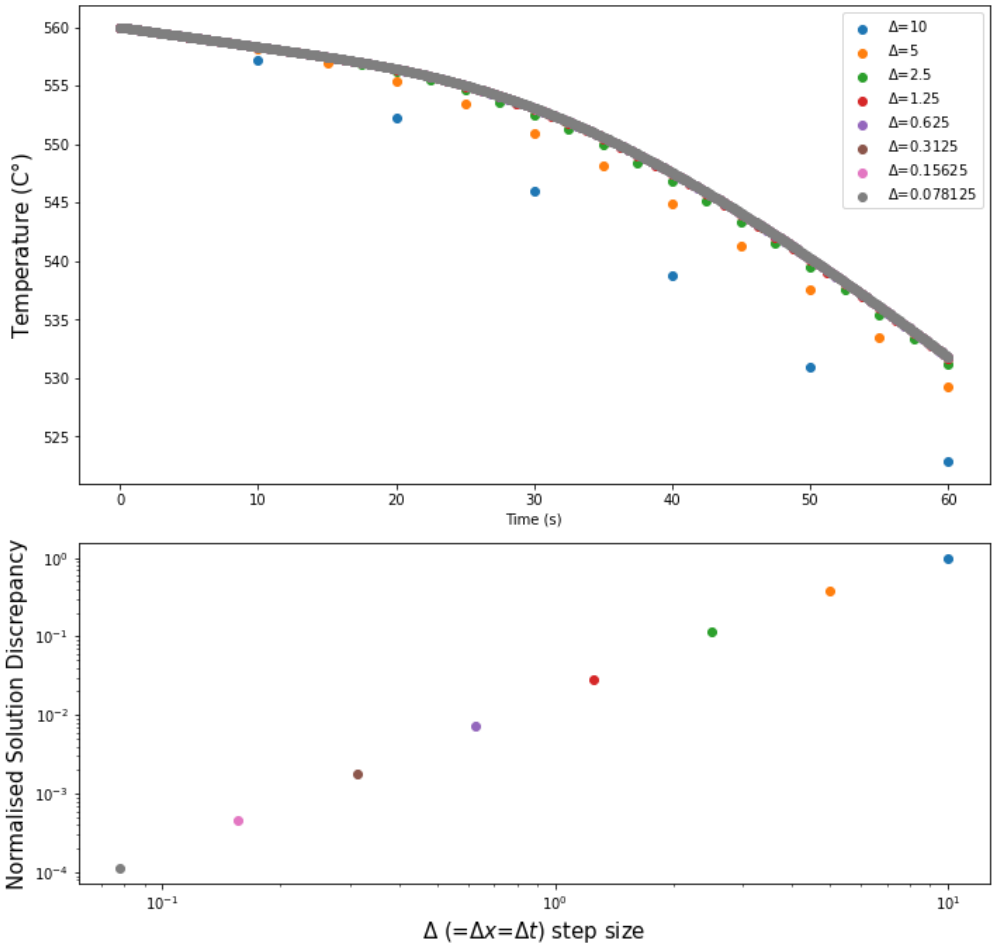


Figure 7: Temperature distribution in time, for x=20cm, for diminishingly small step sizes. The solutions converge onto the same curve, Δ=1.25 being near indistinguishable from Δ= 0.078125. The lower panel shows the log-log plot of the normalised discrepancy between the values $T(x = 20, t = 60)$, calculated for decreasing value of Δ. This log-log plot is linear, showing that the limit of the numerical scheme is the exact solution for Δ tending to 0.

6 Conclusion

We have shown that the heat conduction equation can be solved using explicit and implicit finite difference methods. The error in the explicit numerical method was shown to approximately halve when halving the timestep, in accordance with local truncation error analysis. Application of the implicit Crank-Nicolson method greatly reduced the errors in comparison with the explicit method, also expected from local truncation error analysis. Furthermore, the CN scheme maintained stability whilst the explicit method failed to do so - as expected. This last point encouraged the use of the CN scheme to solve the heat conduction equation for more complex initial conditions, supported by the comparable order of computation time between the explicit and CN methods. Without an analytical solution to compare the CN derived solution to, the latter was shown to converge to a stable curve in the limit Δ tending to zero, as expected by the Lax equivalence theorem.

References

[1] T. N. Narasimhan. Fourier's heat conduction equation: History, influence, and connections. *Reviews of Geophysics*, 37(1):151–172, 1999.

[2] Randall J. Leveque. *Finite difference methods for ordinary and partial differential equations: Steady-state and time-dependent problems*. SIAM, 2007.

[3] R Rosales. Stability of numerical schemes for pdes. https://math.mit.edu/classes/18.306/WWW2007/Notes, Feb 1999. date accessed = 25/03/2023.

[4] J. Crank and P. Nicolson. A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. *Mathematical Proceedings of the Cambridge Philosophical Society*, 43(1):50–67, 1947.

[5] Zhilin Li, Zhonghua Qiao, and Tao Tang. *Numerical solution of differential equations: Introduction to finite difference and finite element methods*. Cambridge University Press, 2017.