

University of Wales Trinity Saint David
School of Engineering

2D Diffusion Equation Model using the
Explicit Finite Difference Method
Computational Methods ACMA6012

Lewis Owen Nekrews

2008701

29/01/2024

Table of Contents

1. Introduction.....	2
1.1 Aims and Objectives	3
2. The Constitutive Equation of the Problem.....	4
3. Explicit Finite Difference Form of the Constitutive Equation	5
3.1 First Derivative Approximation: Forward.....	5
3.2 First Derivative Approximation: Backward	6
3.3 First Derivative Approximation: Central.....	7
3.4 Second derivation Approximation	7
3.5 Converting the Second Differential Equation to Linear.....	8
3.6 Discretized form of the Equation.....	9
3.7 Equation Error.....	10
3.8 Stability criteria.....	10
4. The Boundary Condition.....	11
5. Code Details and Explanations.....	12
5.1 Creation of Boundary Conditions.....	13
5.2 Calculating Temperature within the grid space	14
5.3 Comparison to Analytical method	15
6. Validation of the code	16
7. Observing the Results	17
7.1 Sensitivity Analysis Results	18
8. Main Models Limitations.....	19
8.1 Computer & Geometry Limitations	19
8.2 Geometry Considerations	20
8.3 Limitation Due to Coding	20
9. Suggested Improvements and Future Development	21
9.1 Numerical Alternatives	21
10. Conclusion	22
11. References.....	23
12. Appendix	25

1.Introduction

An aluminium plate was to be considered for temperature distribution using the Finite difference method. The user was asked to simulate thermal diffusion, in respect to time, using the stated heat equation until it had reached equilibrium. To solve this through simulation, the brief required the calculations to be done through MATLAB software.

Within the code, it was firstly necessary to define all properties given for the aluminium and to then calculate the thermal diffusivity. The geometry of the plate was also defined within the code which was stated in the brief to be six metres at both length and width.

The region was to be considered two dimensional and have uniform grid spacing, which is equal to change in the x and y direction. This was to be defined with the necessary constraints preventing the user from impeding the geometry of the two-dimensional plate.

It was also considered the initial boundary and constant boundary condition to be stated in the MATLAB code. This is subject to the dimensions given in the aluminium plate having initial domain temperature of zero degrees Celsius.

Additionally, the time step would be an input function in the calculation. The stability criteria must be fulfilled when the user enters a value time step with constraint being the maximum Fourier value, as suggested to be of zero point two five by Black, W.Z and Kreith, F (1980)

The distribution of heat was set to travel in the y direction and was recorded in each time step and at every node of the plate. This was then implemented in a loop command and would break when equilibrium was reached.

The method to calculate the temperature involved converted from the heat equation discussed by Ozisik, M.N (2017) to a linear equation format using the finite difference method. The choice of numerical method was by using Taylor's series to replace the first and second derivative by approximation.

The methods used to simulate and solve the heat transfer would then be compared to the analytical two-dimensional heat equation given by Daileida, R (2012). The total difference in temperature at equilibrium would then be recorded within the code.

The results showed that a close approximation can be created using this method when implementing the stability criteria set. The accuracy of the numerical method was determined by the tolerance set to a lower value, along with lower grid spacing.

The result in both temperature maps were almost identical, with the high temperature slightly offset because of the north boundary condition. As time increased, the diffusion through the plate gradually dissipated to a more central position.

Sensitivity analysis was implemented to aid understanding on how the parameters effect the result of the numerical value. It was noticed that a reduction in the parameter value increased the accuracy of the results at equilibrium in comparison to the analytical solution. The disadvantage proved to be increase the time taken to calculate thermal equilibrium and would make the method less ideal.

The study concluded that the main model's drawbacks were that the amount of truncation error limited the amount of stability that can be achieved. Restrictions to the input parameters was determined by the stability criteria set up in the code, meaning a limitation to the value difference in spacing and time step.

Further work reviewed what could be done as an alternative to the explicit finite difference method and the implicit finite difference method was considered. This method is similar to the explicit as it still replaces the time derivative, however it solves each time level through a set of simultaneous equations, giving the advantage of having no limitations, improving the time step and reducing rounding error.

In conclusion, it could be seen that with the correct parameter chosen by the user, the code can create a close approximation of the temperature distribution at equilibrium using the finite difference method. This was proven when comparing the analytical method when the difference in results showed small margins. Furthermore, there would be greater limitations when using the code to solve engineering problems with a bigger domain as it can be computationally expensive. Additionally, smaller scaled features may require finer mesh for accuracy, meaning longer computational time.

1.1 Aims and Objectives

The aim of the project was to study the temperature distribution of a two-dimensional aluminum plate through time duration based on Explicit Finite Difference Method. Furthermore, to use the approximation method to demonstrate an accurate solution that could potentially be used in an engineering problem. Additionally validate the method used with the analytical solution to prove the code written is sound.

Of equal importance, to gain knowledge and understanding of principles adopting Finite difference method, enabling the solution of future engineering problems in a working environment. To be able to describe and identify the performance of systems through analytical and numerical methods.

2. The Constitutive Equation of the Problem

Heat is normally transferred through more than one mode which could depend on the level of temperature, conduction, convection, or radiation. However, for this report, conduction will be the only mode considered and is defined by Hahm, D.W and Ozisik, M.N (2017) - the transfer of energy through moving electrons within a lattice of a solid body from a region of high temperature to a low temperature. They also describe the heat conduction equation in Equation 3-1 to show heat flow through an object.

$$k \frac{\partial^2 T}{\partial x^2} + k \frac{\partial^2 T}{\partial y^2} + k \frac{\partial^2 T}{\partial z^2} = \rho c_p \frac{\partial T}{\partial t} \quad (\text{Equation 3-1})$$

When discussing the finite difference method used within the study, it was logical to focus on the importance of where the concept developed. This originated from the general equation for conduction and time dependence and is listed in Equation 3-2. Croft, D.R & Stone, J.A.R (1977) mentions, ‘the study of distribution of temperature though solid structure is principally concerned with this partial differential equation’.

$$\text{div} (k \text{ grad } T) + H = \rho c_p \frac{\partial T}{\partial t} \quad (\text{Equation 3-1})$$

For heat transfer equation, the finite difference method can act as a powerful tool to get an approximate solution. Croft, D.R & Stone, J.A.R (1977) mentions the importance of this method in the minimisation of computational effect to gain an accurate estimation of the temperature distribution and the thermal stresses.

These method based on discretization of giving equation, set boundary and initial conditions was similarly mentioned by Ozisik,M.N (2017) in the book Finite Difference Method in Heat Transfer. His research then discusses that the algebraic solution for a two-dimensional parabolic structure which can be shown in Equation 3-3.

$$\frac{\partial T^2}{\partial x^2} + \frac{\partial T^2}{\partial y^2} = \frac{1}{\alpha} \cdot \frac{\partial T}{\partial t} \quad (\text{Equation 3-1})$$

Defining each term involved in the equation and assuming these values are unformal with no internal heat sources. Thermal conductivity is represented as k - assumed constant, with the value T being temperature of the aluminium plate at x . Density is symbolised by the φ , C_p being the specific heat and finally α being thermal diffusivity.

3. Explicit Finite Difference Form of the Constitutive Equation

The heat equation was to be solved at all nodes simultaneously. Using a differential equation can prove difficult in solving heat distribution of the aluminum plate so Taylor's series expansion was used. Taylor's series is described by Croft, D.R & Stone, J.A.R (1977) as a method of approximating functions by a polynomial, enabling integration from an analytical perspective. Taking the heat diffusion equation previously shown, it was the method used for the approximation regarding the first and second derivative. As described by Patanker, S (2018) in Equation 3-1 by approximating the derivatives in differential equation where the function is replaced by a polynomial can be shown in Equation 3-1.

$$f(x) = f(x_o) + f'(x_o) \cdot \frac{(x-x_o)}{1!} + f''(x_o) \cdot \frac{(x-x_o)^2}{2!} + f'''(x_o) \cdot \frac{(x-x_o)^3}{3!} + \dots \quad (\text{Equation 3-1})$$

3.1 First Derivative Approximation: Forward

For this problem, it was considered beneficial to rewrite Taylor's series by shifting of the origin to the cartesian system instead 0,0 to x_o . This meant the equation could be rewritten with all calculations measured in respect to x_o . This would then introduce the h values to denote the new origin. One of the three approximations would be chosen for the first derivative in the heat equation. Using Taylor's series, it was necessary to transpose to make f' the subject. This was to understand where the first derivative could be approximated by a linear operator through transposition.

For the approximation given by (Morton and Mayers, 2005) in Equation 3-2 to be valid, the h value must remain a small value and then with the first error term is qualified by $O(h)$ as it carries the largest value, and the others are in respect, negligible.

$$\begin{aligned} f(x_o + h) &= f(x_o) + f'(x_o) \cdot \frac{h^1}{1!} + f''(x_o) \cdot \frac{h^2}{2!} + f'''(x_o) \cdot \frac{h^3}{3!} + \dots \\ f(x_o + h) - f(x_o) &= f'(x_o) \cdot \frac{h^1}{1!} + f''(x_o) \cdot \frac{h^2}{2!} + f'''(x_o) \cdot \frac{h^3}{3!} + \dots \\ \frac{f(x_o + h) - f(x_o)}{h} &= f'(x_o) + f''(x_o) \cdot \frac{h^1}{2!} + f'''(x_o) \cdot \frac{h^2}{3!} + f^{(4)}(x_o) \cdot \frac{h^3}{4!} + \dots \\ f(x_o) &\approx \frac{f(x_o + h) - f'(x_o)}{h} \text{ with first error term gives } f(x_o) \\ &= \frac{f(x_o + h) - f'(x_o)}{h} + O(h) \end{aligned} \quad (\text{Equation 3 - 2})$$

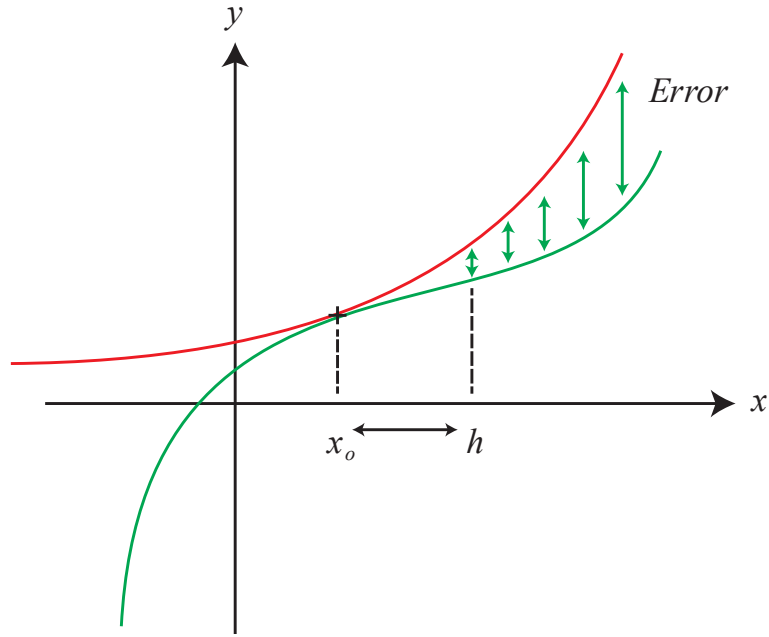


Figure 1 Forward Approximation by the Taylor series show in the increase in error while h value increase Aswell (Morton and Mayers, 2005)

3.2 First Derivative Approximation: Backward

Similarly, the same method can be used with the negative direction to find the backward difference approximation, in order to find an error term. This was done by subtracting $x = x_0 - h$ and making f' the subject which gave the following results with the same level of accuracy and can be seen in Equation 3-3 which was again referenced by Morton and Mayers (2005).

$$f(x_0 - h) = f(x_0) - f'(x_0) \cdot \frac{h^1}{1!} + f''(x_0) \cdot \frac{h^2}{2!} - f'''(x_0) \cdot \frac{h^3}{3!} + \dots$$

$$f(x_0 + h) - f(x_0) = -f'(x_0) \cdot \frac{h^1}{1!} + f''(x_0) \cdot \frac{h^2}{2!} - f'''(x_0) \cdot \frac{h^3}{3!} + \dots$$

$$\frac{f(x_0 - h) - f(x_0)}{h} = -f'(x_0) + f''(x_0) \cdot \frac{h^1}{2!} - f'''(x_0) \cdot \frac{h^2}{3!} + f^{(4)}(x_0) \cdot \frac{h^3}{4!} - \dots$$

$$f(x_0) \approx \frac{f(x_0 - h) - f'(x_0 - h)}{h} \text{ with error gives}$$

$$f(x_0) \approx \frac{f(x_0 - h) - f'(x_0 - h)}{h} + O(h) \quad (\text{Equation 3 - 3})$$

3.3 First Derivative Approximation: Central

The last approximation could be generated by using the previously defined forward and backward equations, creating a third and a smaller error. This could be done by subtracting the second expression from the first creating the following Equation 3-4, resulting in a smaller error term h^2 and is called the central difference approximation, as given by Dewan, N.A (2016).

$$\begin{aligned}
 (1) \quad f(x_0 + h) &= f(x_0) + f'(x_0) \cdot \frac{h^1}{1!} + f''(x_0) \cdot \frac{h^2}{2!} + f'''(x_0) \cdot \frac{h^3}{3!} + \dots \\
 (2) \quad f(x_0 - h) &= f(x_0) - f'(x_0) \cdot \frac{h^1}{1!} + f''(x_0) \cdot \frac{h^2}{2!} - f'''(x_0) \cdot \frac{h^3}{3!} + \dots \\
 f(x_0 + h) - f(x_0 - h) &= 2f'(x_0) \cdot \frac{h^1}{1!} + 2f'''(x_0) \cdot \frac{h^3}{3!} + f^{(5)}(x_0) \cdot \frac{h^5}{5!} + \dots \\
 \frac{f(x_0 + h) - f(x_0 - h)}{2h} &= f'(x_0) + 2f'''(x_0) \cdot \frac{h^2}{3!} + f^{(5)}(x_0) \cdot \frac{h^4}{5!} + \dots \\
 \frac{f(x_0 + h) - f(x_0 - h)}{2h} &= f'(x_0) \text{ with error gives } \dots \\
 f'(x_0) &\approx \frac{f(x_0 + h) - f(x_0 - h)}{2h} + O(h^2) \quad (\text{Equation 3 - 4})
 \end{aligned}$$

3.4 Second derivation Approximation

The only method useful of approximating the second derivative in the heat equation is with central difference approximation. This is done by adding both equation of the following and transposing.

$$\begin{aligned}
 (1) \quad f(x_0 + h) &= f(x_0) + f'(x_0) \cdot \frac{h^1}{1!} + f''(x_0) \cdot \frac{h^2}{2!} + f'''(x_0) \cdot \frac{h^3}{3!} + \dots \\
 (2) \quad f(x_0 - h) &= f(x_0) - f'(x_0) \cdot \frac{h^1}{1!} + f''(x_0) \cdot \frac{h^2}{2!} - f'''(x_0) \cdot \frac{h^3}{3!} + \dots \\
 f(x_0 + h) + f(x_0 - h) &= 2f(x_0) + 2f''(x_0) \cdot \frac{h^2}{2!} + f^{(4)}(x_0) \cdot \frac{h^4}{4!} + \dots \\
 f(x_0 + h) - 2f(x_0) + f(x_0 - h) &= f''(x_0) \cdot \frac{h^2}{1!} + f^{(4)}(x_0) \cdot \frac{h^4}{4!} + \dots \\
 \frac{f(x_0 + h) - 2f(x_0) + f(x_0 - h)}{h^2} &= \frac{f''(x_0) \cdot \frac{h^2}{1!} + f^{(4)}(x_0) \cdot \frac{h^4}{4!} + \dots}{h^2} \\
 \frac{f(x_0 + h) - 2f(x_0) + f(x_0 - h)}{h^2} &\approx f''(x_0) + O(h^2) \quad (\text{Equation 3 - 11})
 \end{aligned}$$

3.5 Converting the Second Differential Equation to Linear

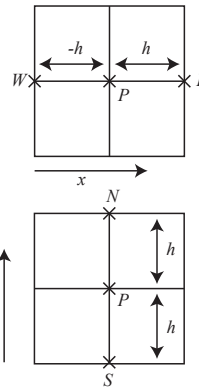
The initial step was to replace the second derivative with respect to x by the central difference approximation. This necessitated replacing f'' with the parameter temperature T and (x_0) with the location point P at each of the nodes in the grid spacing. Next to substitute into the approximation was the temperature and the distance between two nodes shown as $f(x_0 + h)$ which was replaced by temperature at the east direction denoted as T_E . Similarly, this was done for $f(x_0 - h)$ however minus h meant that that it was replaced by temperature west as described in Equation 3-5 by Croft, D.R and Stone, J.A.R (1977). The same thing was repeated when deriving with respect to y when considering the first term $f(x_0 + h)$ which was represented as temperature in the north direction. $f(x_0 - h)$ is temperature to the south and h^2 is the vertical distance between two successive nodes shown in Equation 3-6.

$$(1) \frac{\partial T^2}{\partial x^2} + \frac{\partial T^2}{\partial y^2} = \frac{1}{\alpha} \cdot \frac{\partial T}{\partial t}$$

$$(2) f''(x_0) \approx \frac{f(x_0+h)-2f(x_0)+f(x_0-h)}{h^2}$$

$$\frac{\partial^2 T_p}{\partial x^2} \approx \frac{T_E - 2T_p + T_W}{h^2}$$

$$\frac{\partial^2 T_p}{\partial y^2} \approx \frac{T_N - 2T_p + T_S}{h^2}$$



(Equation 3-5)

(Equation 3-6)

Considering the three choices for the first derivative approximation proven previous, the selected equation would prove impactful on the results of the analysis when studying the heat transfer. The brief suggested choosing the Explicit Finite difference method, the forward difference approximation which leaves us with only one unknown- as described by Kreith, F and Black, W.Z (1980) . While the backward difference approximation would leave multiple unknowns, the explicit method has the advantages of having fewer stability issues. Considering the first function, $f(x_0 + h)$ where the location is moving from a time perspective, h is defined with $t + \Delta t$, where Δt would ask the user to input time step requested in the brief tasks.

$$(3) f(x_0) \approx \frac{f(x_0+h)-f'(x_0)}{h}$$

$$\frac{\partial T}{\partial t} \approx \frac{T_p^{t+\Delta t} - T_p^t}{\Delta t}$$

Equation 3-7

3.6 Discretized form of the Equation

Prediction of the temperature could be made when using the discretized form of the heat equation after a single time step input by the user. This was because there was only one unknown to be solved in the equation as we knew the temperature was initially within the constitutive equation.

This meant if the discretized Equation 3-7 was transposed to make $T_p^{t+\Delta t}$ the subject, the temperature at each node individually could be found. This was mentioned by Kreith, F and Black, W.Z (1980) in the book Basic Heat Transfer and the Equation 3-8 will be the result.

$$\frac{T_E^t - 2T_p^t + T_W^t}{h^2} + \frac{T_N^t - 2T_p^t + T_S^t}{h^2} = \frac{1}{\alpha} \cdot \frac{T_p^{t+\Delta t} - T_p^t}{\Delta t} \quad (\text{Equation 3-7})$$

$$\frac{1}{h^2} (T_E^t + T_W^t + T_N^t + T_S^t - 4T_p^t) = \frac{T_p^{t+\Delta t} - T_p^t}{\alpha \Delta t}$$

$$\frac{\alpha \Delta t}{h^2} (T_E^t + T_W^t + T_N^t + T_S^t - 4T_p^t) = T_p^{t+\Delta t} - T_p^t$$

$$\frac{\alpha \Delta t}{h^2} (T_E^t + T_W^t + T_N^t + T_S^t - 4T_p^t) + T_p^t = T_p^{t+\Delta t} \quad (\text{Equation 3-8})$$

Fourier's number, $\frac{\alpha \Delta t}{h^2}$ is denoted as F_0 Fourier's law as simplified by Kreith, F and Black, W.Z (1980) in literature of basic heat transfer. Also, by dividing the term T_p^t by F_0 to commonality create equation 3-9 allowed the calculation to be factorize and created the final discretized equation that would be used within the code in MATLAB.

$$F_0 (T_E^t + T_W^t + T_N^t + T_S^t - 4T_p^t) + T_p^t = T_p^{t+\Delta t}$$

$$F_0 \left(T_E^t + T_W^t + T_N^t + T_S^t - 4T_p^t \frac{T_p}{F_0} \right) = T_p^{t+\Delta t}$$

$$F_0 \left(T_E^t + T_W^t + T_N^t + T_S^t - T_p \left(\frac{1}{F_0} - 4 \right) \right) = T_p^{t+\Delta t} \quad (\text{Equation 3-9})$$

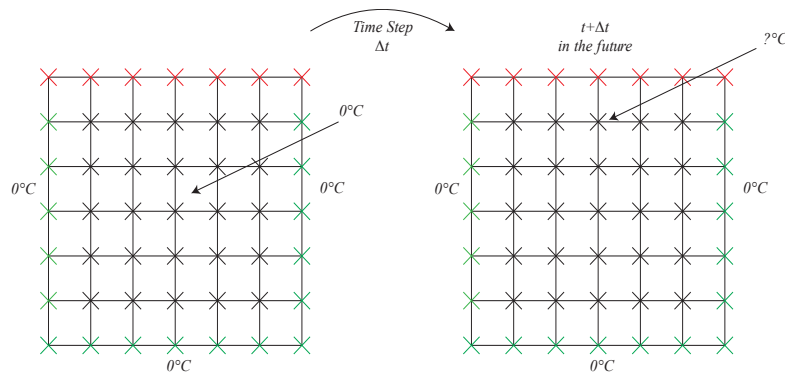


Figure 2 Temperature Calculated at each Nodes Explained by Kreith, F and Black, W.Z (1980)

3.7 Equation Error

The Equation 3-11 is the most common form in textbooks in Heat transfer calculating using finite difference methods where the benefit is control to the propagation of rounding error - as stated by Croft, D.R and Stone, J.A.R (1977) . Another error using this equations is truncation, which is defined by Lewis,R.W, and Morgan, K (1980) as being the value of $O(h)$, the smaller the value the more accurate the calculated result will be. This truncation error is the section that is taken out of the Taylor's series to approximate our first derivative.

3.8 Stability criteria

There is a restriction to size of Fourier number that can be calculated and is referred to the stability limit by Kreith, F and Black,W.Z (1980) page 169. This means anything above is unstable and must remain negative to suit the nodal equation. Kreith, F and Black,W.Z (1980) stated the following limit in Figure 3-1 which is a $1/4$. To further understand this, Equation 3-10 can be transposed to make Δt the subject to prove the equations dependence is on time step size and must be less than $\frac{h^2}{4\alpha}$.

$$0 < \frac{\alpha \Delta t}{h^2} \leq \frac{1}{4} \quad (\text{Equation 3-10})$$

$$0 < \Delta t \leq \frac{h^2}{4\alpha} \quad (\text{Equation 3-11})$$

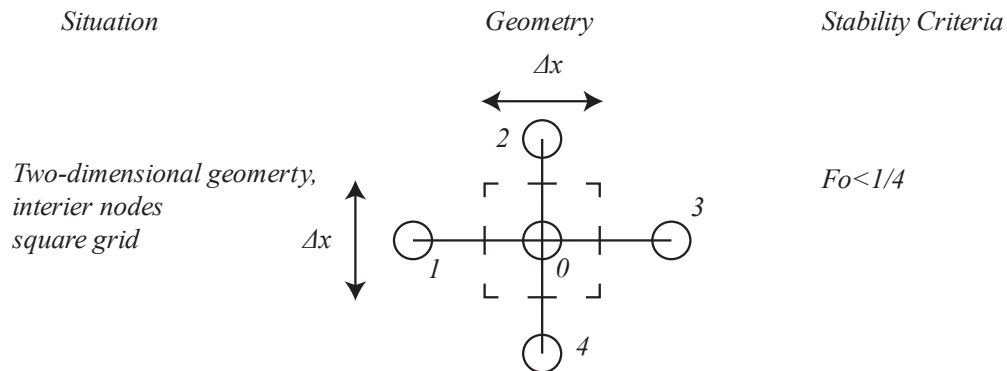


*Figure 3-2 Explicit Nodal Equation and Stability Requirements
Described by Kreith, F and Black,W.Z (1980)*

4. The Boundary Condition

Described as mathematical expressions for the thermal conditions of the boundaries, the initial and boundary conditions for each direction must be stated to calculate the heat transfer of the aluminum plate. Conner, N (2022) stated there must be four boundary conditions for two dimensional problems. The temperature at any point of the aluminum at a specified time depended on the condition of the geometry at the start of the heat conduction process. Such a condition was specified at time $t=0$ and was called the initial condition given in the brief as Equation 4-1.

$$T(x, y, 0) = 0; \quad 0 < x < 6 \text{ \& } 0 < y < 6 \quad (\text{Equation 4-1})$$

The boundary condition is specified temperatures or also called the prescribed temperature, which is stated by Ghoshdastidar, P.S (2004). He states that with this condition the temperature may be constant or a function of time and is also referred to as the Dirichlet condition.

The temperature of the exposed surface could be easily measured directly and the simplest method to specify the thermal condition of the surface was to determine the temperature. The boundary condition was expressed in Equations 4-4 and requirements of the conditions can be seen in the schematic Figure 4-1 where the temperature remained zero, apart from the north boundary.

$$\left. \begin{array}{l} T(0, y, t) = 0 \\ T(6, y, t) = 0 \end{array} \right\} \quad 0 \leq y \leq 6 \text{ \& } 0 \leq t \leq \text{thermal equilibrium} \quad (\text{Equation 4-2})$$

$$\left. \begin{array}{l} T(x, 0, t) = f(x) \\ T(x, 6, t) = 0 \end{array} \right\} \quad 0 \leq y \leq 6 \text{ \& } 0 \leq t \leq \text{thermal equilibrium} \quad (\text{Equation 4-3})$$

$$f(x) = \begin{cases} 12.5 & \text{if } 0 \leq x \leq \left(\frac{2}{3} \times 6\right) \\ -12x + 150 & \text{if } \left(\frac{2}{3} \times 6\right) < x \leq 6 \end{cases} \quad (\text{Equation 4-4})$$

$$T(x, 6, t) = f(x)$$

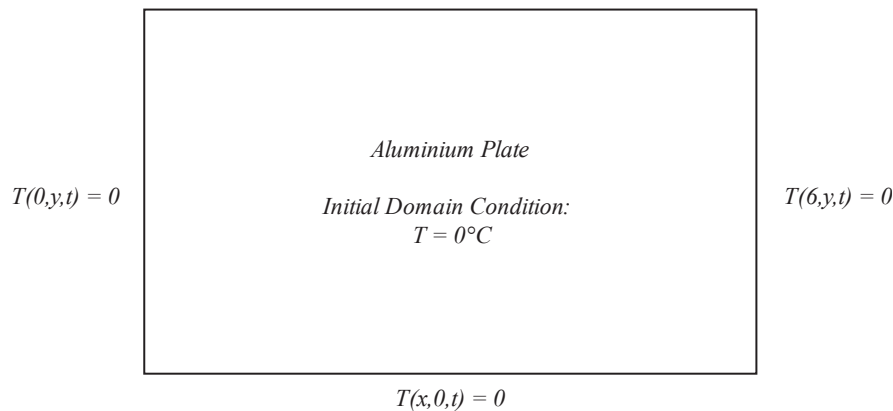


Figure 4-1 Schematic of Two Dimensional plate with Conditions Defined

5.Code Details and Explanations

The initial step when starting the code was to create variables of the aluminum properties that were given in the brief with all values shown in si units. All three aluminum properties were assumed to be constant value during the study and the programmer used those values to create a variable for heat diffusivity α . The geometry of the plate was also allocated variables with thickness of the plate considered negligible.

Next was to request the user to input the values for the grid spacing h between each node. This value would also have to meet certain conditions to satisfy the geometric values previously stated. The code would continue to request a valid value through a loop function. The matrix variables were defined with their number of points for both coordinates divided by the spacing value to create node points.

The second input was requested by the user and defined the time step value. The time step was then used to calculate the Fourier stability criteria and with previous values defined. The stability criteria value was referenced into a while loop to restrict the user from creating unstable results.

```
% aluminium properties
k_heat = 143; % Heat Conductivity
rho = 2.8e3; % density
cp = 795; % Heat capacity
Alpha = k_heat/(rho*cp); % Heat Diffusivity
% Geometry in metres
L = 6;
W = 6;

% Grid spacing
h = input('input grid size: ');
while mod(L,h)~= 0 || h<= 0 || h>= L
    fprintf('not a valid grid spacing - h cannot be %g \n',h)
    h=input('input a better grid size: ');
end

% Matrix Dimensions
Nx = L/h +1; % number of points x direction
Ny = W/h +1; % number of points y direction

% Fourier Number
F0 = Alpha*dt/h^2
while F0 < 0 || F0>0.25
    fprintf('Stability criterion not fulfilled - F0 cannot be %g \n',F0)
    dt = input (' time step, delta t = '); % New Time step user input
    F0 = Alpha*dt/h^2
end
```

Figure 5-1 initial Steps of Code Created In MATLAB including Stability Criteria Implemented

5.1 Creation of Boundary Conditions

Creating the initial and constant boundary condition was the next step in the code. This was initially set with zeros at each node point in array, then for the boundary condition, using a for loop which would introduce a counter where it was then multiplied by the grid spacing. The loop continued across the first row of the matrix using one of the two values given in the brief with a conditional statement. The size of the grid spacing would ultimately determine which value is chosen across the boundary.

Grid creation started with using both x and y coordinates then meshed in preparation of simulation. Once the grid was meshed, flipping the boundary would simulate the diffusion coming from the north direction. Initialization of time was also required before heat diffusion began and a second matrix from the next temperature.

```
% initialise Conditions
T = zeros (Nx,Ny);
% Boundary Conditions with Given Values in Brief
for i=1:1:Nx
    d=(i-1)*h;
    if d<=(2/3)*L
        T(1,i)=12.5*d;
    else
        T(1,i)=-25*d+150;
    end
end

% Grid Creation using Given h value
x = 0:h:L;
y = 0:h:W;
[X,Y] = meshgrid(x,y); % Meshgrid function created in preparation of Temp Map

% Temperature at first time step using explicit finite difference method
T = flip(T); % Diffusion From the North Boundary
Tnext = T;
Time = 0;
k= 1;
```

Figure 5-2 Boundary Conditions Created and Grid Creation In MATLAB

5.2 Calculating Temperature within the grid space

Using the explicit finite difference equation, the temperature was calculated through a while loop which was repeated if the calculated value was above the tolerance set by the programmer. The tolerance would then be regarded as a sufficient estimate of the temperature at equilibrium.

Two for loops were nested within the while loop for both x and y direction to calculate the finite difference equation, where the value of temperature at the nodal point was collected and at each cardinal direction.

With a condition statement, the distribution of temperature was illustrated through each frame, this would simulate the diffusion of the aluminum plate using the numerical method until a time to reach equilibrium was calculated.

```
% Set Max change in temperature between iterations
while max(max(abs(T - Tnext))) > 1e-4 || max(max(abs(T - Tnext))) ==0
    % Repeat until Equilibrium
    T = Tnext;
    % Temperature at Grid points
    for i=2:Nx-1
        for j = 2:Ny-1
            TN=T(i,j+1); TW=T(i-1,j); TS=T(i,j-1); TE=T(i+1,j); Tp=T(i,j);
            Tnext(i,j) = F0 * (TN+TS+TE+TW-(4-1/F0)*Tp);
        end
    end
    if mod(k,100)==0 % Speed up frame
        contourf(X,Y,Tnext); % Plot Temperature Map
        colorbar;
        title('Temperature Distribution Numerical');
        xlabel('X Axis');
        ylabel('Y Axis');
        f(k/10) = getframe;
    end
    % Time recorded to reach Equilibrium
    Time = k*dt;
    k=k+1;
end

% Display Time Thermal Equilibrium is reached
fprintf('Thermal equilibrium reached at time = %f seconds\n', Time)
```

Figure 5-3 Explicit Finite Difference Equation Introduction into MATLAB with tolerance set by the programmer.

5.3 Comparison to Analytical method

The temperature at equilibrium was compared with the analytical equation value and a difference was created in code to prove the numerical accuracy. This analytical solution was displayed using a temperature map so the user could identify and compare. The matrix 'Tnext' representing numerical and 'TA' as analytical are displayed in the command window in Figure 5-5 for the user to further compare the quantitative difference.

```
% Grid set to zeros
TA = zeros(Nx,Ny);
% Calculate Analytical Method using Given Formula
for i= 1:Nx
    for j = 1:Ny
        for n=1:100
            x=(j-1)*h;
            y=(i-1)*h;
            TA(i,j) = TA(i,j) +
450/pi^2*(sin(2*n*pi/3)/(n^2*sinh(n*pi))*sin(n*pi*x/6)*sinh(n*pi*y/6));
        end
    end
end
% Plot Temperature Map
figure;
contourf(X,Y,TA);
colorbar;
title('Temperature Distribution Analytical');
xlabel('X Axis');
ylabel('Y Axis');
% Comparing The difference in Value for both Analytical To Numerical Results
Difference = max(max(abs(T - TA)));
fprintf('Thermal equilibrium Difference in Value Between Numerical & Analytical =
%f \n', Difference)
```

Figure 5-4 Analytical method instructed into code for comparison and validation

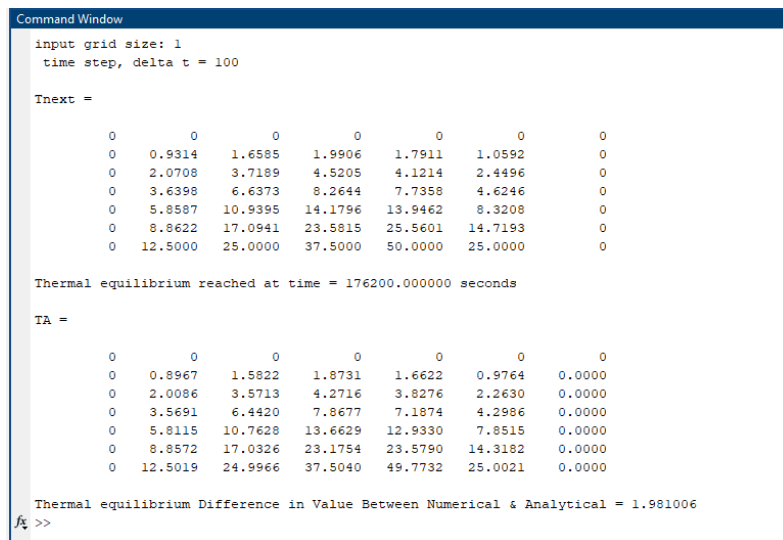


Figure 5-5 Command Window Displaying Results from Code

6. Validation of the code

Verifying the numerical code to analytical solution with example parameters $h=1.0$, time step =100 and tolerance $\times 10^{-4}$ showed that the results are reasonably accurate representations. When studying the results in the numerical and analytical Table 6-1, there are no noticeable extreme differences in temperature with equilibrium at each node. This indicated that the numerical algorithm has no major defections or bugs. However, regarding the solution quality, the difference in solutions could potentially have been from the discretized error term previously implemented through Taylor's series expansion. The stability criteria set in the code proved sufficient in gaining a close estimation. Limited differences in results were recorded and no gradual increase evident, making the numerical scheme stable.

The method of validating the results was to reduce the grid space which lowered the error significantly, as proven figure 6-2. This was suggested by Roy, C.J to increase the number of points in the grid, he stated, proves consistency and convergence. This was tested by the user with a grid space of 0.1 and time step of 10.

Of interest was the difference in the two temperature maps in figure 6-2 which display the continuous curves of the analytical and numerical diffusion. On both maps, the temperature seen was initially higher to the right side due the unequal distribution of heat, the temperature gradually evened out to center as time increased.

Test 1							
Input grid size	1						
Time Step Deltat	100						
Numerical Matrix(Tnext) =							
0	0	0	0	0	0	0	0
0	0.9314	1.6585	1.9906	1.7911	1.0592	0	0
0	2.0708	3.7189	4.5205	4.1214	2.4496	0	0
0	3.6398	6.6373	8.2344	7.7358	4.6246	0	0
0	5.8587	10.9395	14.1796	13.9462	8.3208	0	0
0	8.8622	17.0941	23.5815	25.5601	14.7193	0	0
0	12.5000	25.0000	37.5000	50.0000	25.0000	0	0
Thermal Equilibrium Reach at Time = 176200 seconds							
Analytical Matrix(TA) =							
0	0	0	0	0	0	0	0
0	0.8967	1.5822	1.8731	1.6622	0.9764	0	0
0	2.0086	3.5713	4.2716	3.8276	2.263	0	0
0	3.5691	6.442	7.8677	7.1874	4.2986	0	0
0	5.8115	10.7628	13.6629	12.933	7.8515	0	0
0	8.8572	17.0326	23.1754	23.579	14.3182	0	0
0	12.5019	25.9966	37.5040	49.7732	25.0021	0	0
Thermal equilibrium Difference in Value Between Numerical and Analytical = 1.981006							

Test 2	
Input grid size	1
Time Step Deltat	100
Thermal equilibrium reached at time = 107620 seconds	
Thermal equilibrium Difference in Value Between Numerical & Analytical = 0.279574	

Table 6-1 Testing code & the Results Displayed

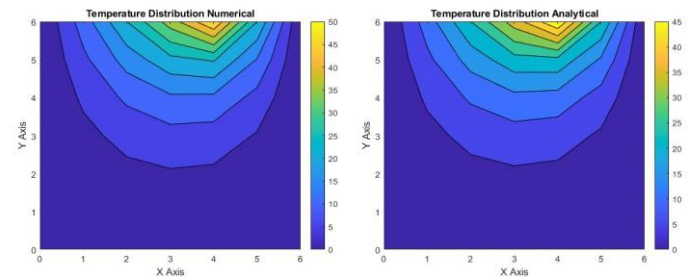


Figure 6-1 Test 1 - Temperature Map Displayed For both numerical and Analytical

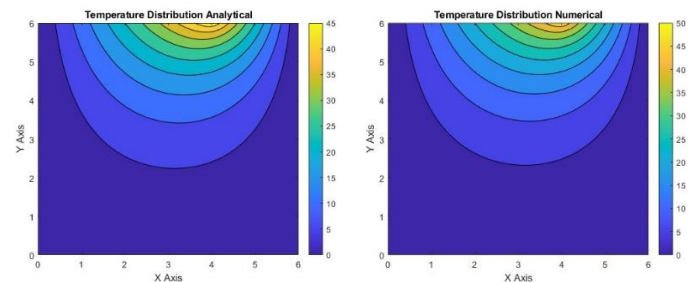


Figure 6-2 Test 2 - Temperature Map Displayed For both numerical and Analytical Showing improved results.

7. Observing the Results

The sensitivity analysis investigated the variations of each parameter included in the previous illustrations. This proved beneficial when identifying the optimum tolerance and aided understanding when evaluating parameters for accurate results.

When studying the effect of the grid spacing h , it can be said that this parameter greatly influenced the outcome of results. Since the spacing chosen by the user determined the number of nodes in the matrix sets, the results showed a greater probability to favor a finer mesh, as its suggested in a study by Finlayon, B.A (2012) to significantly increase accuracy.

Additionally, when looking at the time step values and because the study involved a large area, ideally the grid needed to be finer, to calculate the distribution. With the stability criteria imposed in the brief, this necessitated the need to reduce the time step, making the evaluation more cumbersome, but not necessarily more accurate, as result show in analysis. This was also stated by Crofts, D.R and Stone J.A.R (1977) where the reduction in these parameters would only increase the labour substantially.

The tolerance value, which was not controlled by the user, showed from results that it must be set to a low value or else the nodes would not record any temperature, in the numerical method, and the difference to the analytical would be substantial. In summary, the tolerance set should be small enough to relate to the set time step in order to gain close approximation.

The analysis strategy developed of each stage, first spacing, secondly time step, then lastly the tolerance. This method assisted the programmer in finding the lowest value difference when comparing to analytical when temperature is at equilibrium.

7.1 Sensitivity Analysis Results

Grid Spacing	Time Step(s)(Fixed)	Tolerance (Fixed)	difference	Thermal Equilibrium Time (s)
3	150	0.0001	1.484	209700
2			3.7131	190200
1.5			0.7647	190200
1.2			0.5601	185250
1			1.9852	187800
0.75			0.3167	185700
0.5			0.9997	185100
0.25			0.5004	184500
0.2			0.4004	184350

Table 7-2 Influence of the Grid Spacing Studied

Grid Spacing(Fixed) (m)	Time Step (s)	Tolerance (Fixed)	Change/Difference	Thermal Equilibrium Time (s)
0.75	2000	0.0001	0.323	254000
	1500		0.3228	247500
	1000		0.3225	238000
	800		0.3222	232000
	600		0.3218	224400
	400		0.3209	213200
	200		0.3184	194000
	100		0.3133	174200
	50		0.3031	154350
	25		0.2827	134475
	15		0.2554	119790
	10		0.2211	108140
	5		0.4734	88210

Table 7-2 Influence of the Time step while building on what was found in Grid Spacing Study

Grid Spacing (Fixed)	Time Step (Fixed)	Tolerance	Difference	Thermal Equilibrium Time (s)
0.75	10	0.1	28.4093	10
		0.01	12.4959	7890
		0.001	2.5734	43840
		0.0001	0.2211	108140
		0.00001	0.3133	174350
		0.000001	0.3225	2450550

Table 7-3 The influence of the Tolerance set by the Programmer

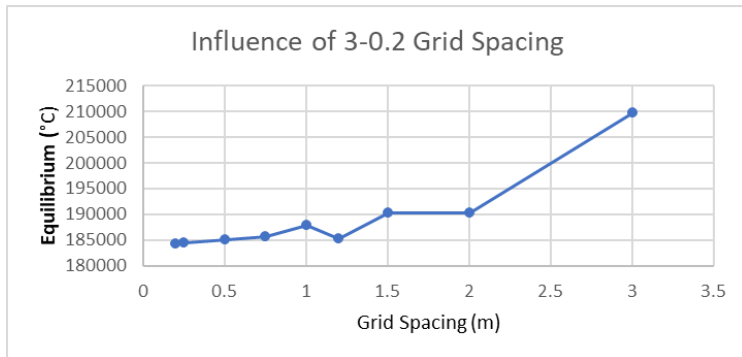


Figure 7-1 Influence of the Grid Spacing Shown in Graph Form

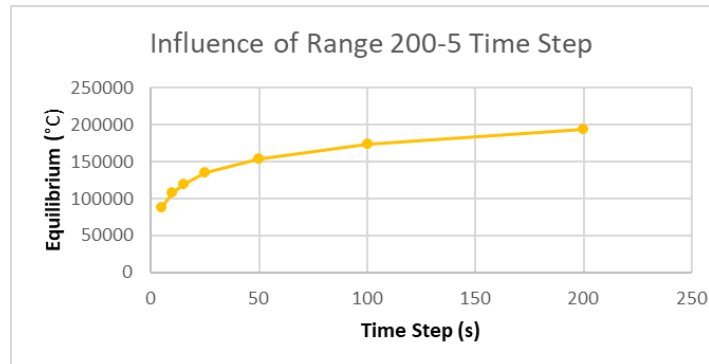


Figure 7-2 Influence of the Time Step Shown in Graph Form

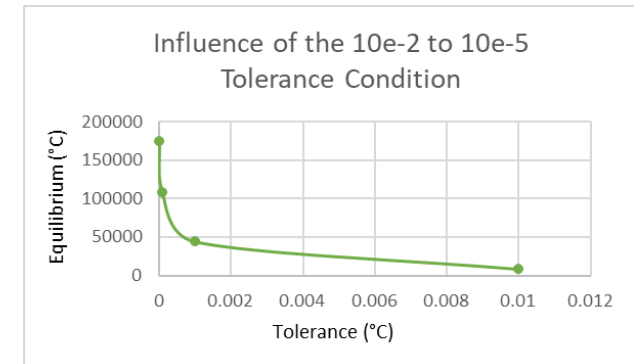


Figure 7-3 Influence of the Tolerance Shown in Graph Form

8. Main Models Limitations

When using numerical methods to solve engineering problems, this would introduce errors into the system. The error would be the discretization between the solution of the original problem and the discrete solution as mentioned by Douglas, N (2018) on his paper on stability, consistency, and convergence. These three are regarded as important concepts when creating a computationally stable scheme.

Consistency, also called truncation error E_p , finite difference scheme replace the partial difference equations scheme at a point P and could be seen in Equation 8-1 as described Li, Z and Tang, T (2017) Although the error is reduced using the explicit method suggested by Smith (1978)

$$E_p = (\text{PDE})_p - (\text{FDM})_p \quad \text{Equation 8-1}$$

Limits to the amount of stability could be another concern when using linear equation for an exact solution as there would be certain amount of truncation implemented at every calculation step, causing round off error. When the exact solution T_p is the same result to the calculated solution T_p^c at a certain point, this gives the Equation 8-2

$$t_p = T_p - T_p^c \quad \text{Equation 8-2}$$

Limitations on convergence depending on the use of Fourier's number were to be considered when studying temperature distribution with the value of Fourier number = 0.25, as suggested previously by Kreith, F and Black, W.Z. (1980) However, Croft, D.R and Stone, J.A.R (1977) described that truncation and round off error are both magnified when these value limitation are set, meaning they cannot be chosen arbitrarily without the result being useless.

8.1 Computer & Geometry Limitations

Round off errors also occur because of computer limitations, digital computers do not give the exact value because of a limitation in representing numbers. As Croft, D.R and Stone, J.A.R (1977) stated - computers can only store the finite number of digits to represent the value and with the truncation of numbers in a certain order, decimal digit will introduce the round off error to each step of the computation. It should be said that round off error may be negligible for the first few calculations, however it could potentially accumulate into a significant change in result.

8.2 Geometry Considerations

Complexity of geometry would be difficult to approximate as the Finite Difference method is mainly studied with regular grid patterns. This was mentioned by Heinzil, R in the study of scientific computing. In addition, the analytic approach becomes too involved and complex. The geometry would be near impossible to obtain a closed form solution according to Kreith , F and Manglik, R.M (2011) in the booklet, Principles of Heat Transfer.

8.3 Limitation Due to Coding

Convergence was the main issue when using this coding method to gain an accurate result. The time step chosen by the user and the grid influence the accuracy of results substantially. Even though the results are in agreement with Fourier's conditional statement, the user could still input a time step that could prove insufficient. The limitation being the number of problems that could be solved using this code without the loss of accuracy and compositional time to provide a solution in comparison to implicit methods.

The code used a while loop to calculate the temperature at each node, within the grid and each time step, and repeated that pattern until equilibrium is reached. The code assumes that the boundary did not change over the time duration. This may not be true when considering the surroundings may affect the temperature of the aluminum during its heat distribution. To implement this, it would be necessary to reapply the conditions at every iteration.

As a consequence of being unable to store an infinite amount of information, the heat transfer process would experience. The method was the average of the surrounding nodal temperature and as described by Raymond,C (2020), the model instead made it discrete by dividing the space up into a series of cells and when assessing the results still show the functions is varying.

This means there is still an infinite amount of information that has not been accounted for within the temperature distribution.

9. Suggested Improvements and Future Development

One of the major drawbacks observed in the study. of using the explicit finite difference method was that the temperature at time level $t + \Delta t$ depended explicitly on the temperature at time level t . Crank-Nicolson method or also known as the implicit finite difference method still replaced the time derivative like the explicit scheme, however, as mentioned by Croft, D.R and Stone, J.A.R (1977) The space derivatives are averaged over $t + \Delta t$ and t .

The solution could be seen in Equations 9-1 where for each time level a set of simultaneous equations must be solved. Its main advantage is discussed by Ozisik, M.N and Orlande, H.R.B (2017)- There are no severe limitations imposed on the time step and the stability due to the lack of rounding error.

$$T'_0 + T'_I + T'_E + T'_W - 2\left(2 + \frac{1}{F_0}\right)T'_p$$

$$= -T_0 - T_I - T_E - T_W + 2\left(2 - \frac{1}{F_0}\right)T_p \quad \text{Equation 9-1}$$

9.1 Numerical Alternatives

For further development when calculating heat diffusion, finite element method could be introduced. This could potentially be suitable alternative numerical method used to obtain an approximate solution for a given heat transfer problem and as described by Larson, M.G and Bengzon, F (2013) it uses polynomial function to approximate the solution within each element by discretizing the domain in smaller interconnected elements.

This process can be seen in Figure 9-1 where similarly to FDM, the geometry is separated into a grid space. This considers calculating at each element and then obtaining the solution with the nodes within the domain.

Larson, M.G and Bengzon, F (2013) further detailed while discussing two dimensional applications where the methodology of building piecewise polynomials on triangulations can be suitable for compute implementation and the study of approximation properties.

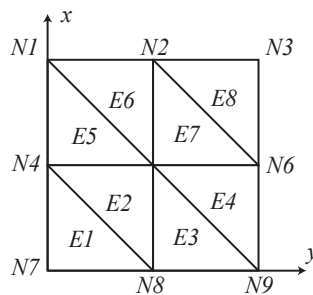


Figure 9-1 Creating the Mesh in Triangular elements as mentioned by Larson, M.G and Bengzon, F (2013)

10. Conclusion

The study considered the heat distribution through a two-dimensional aluminum plate until it reached equilibrium. This was done through a grid form where the temperature was calculated at each node simultaneously, recorded and displayed in a heat map. The time at equilibrium was registered and the distribution value at each node compared to the analytical method. The problem was based around numerical approximation using the explicit finite difference method and the discretized form of the heat equation initially given in the task description.

The results showed that a close solution could be established when remaining within the stability criteria set by the programmer. The code also displayed that when the user inputs significantly smaller value grid spacing, more calculations were recorded, resulting in a closer numerical solution with a smaller difference in temperature to the analytical solution.

Limitations to the code's convergence were identified, the difference in temperature being from the truncation when calculating each node. Furthermore, the stability criteria set meant that the grid size was limited, which, through analysis, showed to increase the accuracy of the results.

The study concludes that basic computational knowledge and understating can lead to accurate approximation to an engineering heat transfer problem which can be validated against the analytical solution.

The study suggests that for future developments, computational methods could be used to solve complex engineering problems when implementing appropriate action. These could be separated into simpler subproblems rather than solving by conventional means, which can be either counterproductive or near impossible due to the amount of data presented.

11. References

Croft, D.R & Stone, J.A.R (1977): Heat Transfer Calculations using Finite Difference Equations – The Basic Idea - Page 1, Sheffield Hailam University Press, Sheffield

Croft, D.R & Stone, J.A.R (1977): Heat Transfer Calculations using Finite Difference Equations – The finite Difference Method - Basic Ideas - Page 40, Sheffield Hailam University Press, Sheffield

Croft, D.R & Stone, J.A.R (1977): Heat Transfer Calculations using Finite Difference Equations – The finite Difference Method - Basic Ideas - Page 44, Sheffield Hailam University Press, Sheffield

Croft, D.R & Stone, J.A.R (1977): Heat Transfer Calculations using Finite Difference Equations – Error Accuracy- Page 99, Sheffield Hailam University Press, Sheffield

Croft, D.R & Stone, J.A.R (1977): Heat Transfer Calculations using Finite Difference Equations – Transient Problems- Page 62, Sheffield Hailam University Press, Sheffield

Daileda, Rc (2012) The Two Dimensional Heat Equation – Complete Analytical Solution- Slide 25, [Online] <http://ramanujan.math.trinity.edu/rdaileda/teach/s17/m3357/lectures/lecture12.pdf> , Accessed 16/01/2024, Trinity University

Dewan. N.A (2016): Truncation Error and Taylor Series – Chapter 4, Page 82, Dept. of EEE, BUET, [Online] <https://dewan.buet.ac.bd/EEE423/CourseMaterials/TaylorSeries.pdf> , Accessed 15/01/2024

Finlayon, B.A Introduction to Chemical Engineering Computing – Finite Difference Method using MATLAB, Second Edition, Wiley, New Jersey

Heinzl, R (2007) Concept of Scientific Computing – Finite Differences- Slide 3, [Online] <https://www.iue.tuwien.ac.at/phd/heinzl/node27.html> , Accessed 19/01/2024, Vienna University of Technology

Kreith, F, Black, W,Z (1980) :Basic Heat Transfer – Forward Difference in Time, Page 166, Harper & Row Publishers, New York

Kreith, F, Manglik, R.M, Bohn (2011), M.S : Principles of Heat Transfer – Remarks regarding Geometry – Page 150, Seventh Edition, Cengage Learning – Stamford

Larson, M.G and Bengzon, F (2013) The finite Element Method – Theory, Implementation and Applications – Page 45, Springer, London

Li, Z. Qiao, Z. Tang, T (2017): Numerical Solutions Of Differential Equations – Simple Example Of a Finite Difference Method – Truncation Error, Page 22, Cambridge University Press

Morton, K. and Mayers, D. (2005) Numerical Solution of Partial Differential Equations. Page 11, 2nd Edition Cambridge University Press

Morton, K. and Mayers, D. (2005) Numerical Solution of Partial Differential Equations. 2nd Edition Cambridge University Press

Ozisik, M.N & Orlande, H.R.B (2017) : Finite Difference Methods in Heat Transfer – Implicit Method, Page 13, (2nd Edition) CRC Press, Boca Raton

Ozisik, M.N & Orlande, H.R.B (2017) : Finite Difference Methods in Heat Transfer – Parabolic Systems Heat Equation, Page 5, (2nd Edition) CRC Press, Boca Raton

Patankar, S (2018) : Numerical Heat Transfer and Fluid Flow – Methods of Deriving the Discretization Equation, Chapter 3, CRC Press, Boca Raton

Rumpf, R.C (2020) Computational Methods in Engineering- introduction to Two Dimensional finite Difference Method- Slide 2, [Online] <https://empossible.net/wp-content/uploads/2020/09/Lecture-Introduction-to-Two-Dimensional-FDM.pdf>, Accessed 19/01/2024, El Paso, Texas

Shabany, Y (2009) Heat Transfer – Principle of Conservation of Energy - Page 54-,CRC Press, Boca Raton

12. Appendix

```
clc
close all
clear
format short
% aluminium properties
k_heat = 143; % Heat Conductivity
rho = 2.8e3; % Density
cp = 795; % Heat capacity
Alpha = k_heat/(rho*cp); % Heat Diffusivity
% Geometry
L = 6;
W = 6;

% Grid spacing
h = input('input grid size: ');
while mod(L,h)~= 0 || h<= 0 || h>= L
    fprintf('not a valid grid spacing - h cannot be %g \n',h)
    h=input('input a better grid size: ');
end

% Matrix Dimensions
Nx = L/h +1; % number of points x direction
Ny = W/h +1; % number of points y direction

% Time step
dt = input (" time step, delta t = "); % Time step user input

% Fourier Number
F0 = Alpha*dt/h^2;
while F0 < 0 || F0>0.25
    fprintf('Stability criterion not fulfilled - F0 cannot be %g \n',F0)
    dt = input (' time step, delta t = '); % New Time step user input
    F0 = Alpha*dt/h^2;
end

% Initialise Conditions
T = zeros (Nx,Ny);
% Boundary Conditions with Given Values in Brief
for i=1:1:Nx
    d=(i-1)*h;
    if d<=(2/3)*L
        T(1,i)=12.5*d;
    else
        T(1,i)=-25*d+150;
    end
end
% Grid Creation using Given h value
x = 0:h:L;
y = 0:h:W;
[X,Y] = meshgrid(x,y); % Meshgrid function created in preparation of Temp Map

% Temperature at first time step using explicit finite difference method
T = flip(T); % Diffusion From the North Boundary
Tnext = T;
Time = 0;
k= 1;
```

```

% Set Max change in temperature between iterations
while max(max(abs(T - Tnext))) > 1e-4 || max(max(abs(T - Tnext))) ==0
    % Repeat until Equilibrium
    T = Tnext;
    % Temperature at Grid points
    for i=2:Nx-1
        for j = 2:Ny-1
            TN=T(i,j+1); TW=T(i-1,j); TS=T(i,j-1); TE=T(i+1,j); Tp=T(i,j);
            Tnext(i,j) = F0 * (TN+TS+TE+TW-(4-1/F0)*Tp);
        end
    end
    if mod(k,100)==0 % Speed up frame
        contourf(X,Y,Tnext); % Plot Temperature Map
        colorbar;
        title('Temperature Distribution Numerical');
        xlabel('X Axis');
        ylabel('Y Axis');
        f(k/10) = getframe;
    end
    % Time recorded to reach Equilibrium
    Time = k*dt;
    k=k+1;
end
Tnext
% Display Time Thermal Equilibrium is reached
fprintf('Thermal equilibrium reached at time = %f seconds\n', Time)

% Grid set to zeros
TA = zeros(Nx,Ny);
% Calculate Analytical Method using Given Formula
for i= 1:Nx
    for j = 1:Ny
        for n=1:100
            x=(j-1)*h;
            y=(i-1)*h;
            TA(i,j) = TA(i,j) +
450/pi^2*(sin(2*n*pi/3)/(n^2*sinh(n*pi))*sin(n*pi*x/6)*sinh(n*pi*y/6));
        end
    end
end
TA
% Plot Temperature Map
figure;
contourf(X,Y,TA);
colorbar;
title('Temperature Distribution Analytical');
xlabel('X Axis');
ylabel('Y Axis');

% Comparing The difference in Value for both Analytical To Numerical Results
Difference = max(max(abs(T - TA)));
fprintf('Thermal equilibrium Difference in Value Between Numerical & Analytical =
%f \n', Difference)

```

