

TRANSIENT THERMAL DIFFUSION MODEL USING THE FINITE DIFFERENCE METHOD

ADVANCED COMPUTATIONAL METHODS - MATLAB

SUPERVISION

Dr Arnaud Marotin

BEng Motorcycle Engineering

Level 6

Semester 1

ANTONIO JANUARIO

SWANSEA

January 2020

CONTENTS

1	INTRODUCTION.....	1
2	THE CONSTITUTIVE EQUATION OF THE PROBLEM	2
3	NUMERICAL METHOD CHOSEN TO DISCRETISE THE CONSTITUTIVE EQUATION	4
4	INITIAL AND BOUNDARY CONDITIONS	6
4.1	ASSIGNMENT PLATE.....	6
4.2	VALIDATION – TEST PLATE	7
5	FINITE DIFFERENCE MODEL (SCRIPT)	8
6	ANALYSIS OF CODE RESULTS.....	11
6.1	VALIDATION – TEST PLATE	11
6.2	ASSIGNMENT PLATE.....	12
7	LIMITATIONS OF THE MODEL	13
8	FUTHER IMPROVEMENTS	14
9	CONCLUSION.....	15
10	BIBLIOGRAPHY	16
11	APPENDIX	17
11.1	Validation _ Januario _ Antonio _ 1805566.m.....	17
11.2	Januario _ Antonio _ 1805566.m.....	19

1 INTRODUCTION

The following assessment was developed for the module of Advanced Computational Methods – MATLAB of the third year – level 6 of the Motorcycle Engineering Degree.

The aim of the presented work is to demonstrate a comprehensive and thorough knowledge of heat transfer calculations using finite difference equations.

It is proposed to calculate the temperatures of a plate at each time step by solving a time-dependent thermal diffusion equation using the finite difference method. This calculation will be performed through a MATLAB model, based on the Explicit Method, developed by the author.

Success will be measured by how accurate and cohesive the results from the test plate - created by the author and with a known solution - are, allowing the validation of the MATLAB model developed.

2 THE CONSTITUTIVE EQUATION OF THE PROBLEM

The problem proposed by the assignment is to calculate the temperature distribution within an aluminium plate through a specific time step until it reaches thermal equilibrium.

The temperature distribution is dictated by the heat transfer, which is the conversion, use and exchange of thermal energy, that could occur through different mechanisms: thermal conduction, thermal convection and thermal radiation.

The mechanism of heat transfer being analysed for this specific problem is the thermal conduction. Thermal conduction is defined as the energy exchange between solids or fluids at rest and this transport of energy is caused by molecular motion due to an internal temperature gradient.

Physical contact between the systems is necessary in order for thermal conduction to happen. For solids, such as the aluminium plate being studied, the heat transfer occurs by a combination of vibration and collision of molecules and diffusion and collisions of free electrons.

The phenomena described above, thermal conduction in solids, is characterised by the following thermal diffusion equation:

$$\nabla^2 T = \frac{1}{\alpha} \cdot \frac{\partial T}{\partial t}$$

Where,

- ∇^2 is the Laplacian,

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

- t is time;
- T is the temperature;
- α is the material's thermal diffusivity ($\text{m}^2 \cdot \text{s}^{-1}$),

$$\alpha = \frac{k}{\rho \cdot c}$$

- k is the thermal conductivity ($\text{W} \cdot \text{m}^{-1} \cdot \text{K}^{-1}$);
- ρ is the density ($\text{kg} \cdot \text{m}^{-3}$);
- c the specific heat ($\text{J} \cdot \text{kg}^{-1} \cdot \text{K}^{-1}$);

The calculations will be performed in the 2-dimensional aluminium plate covered by a uniform grid, with a constant grid spacing – h , as shown in the figure below,

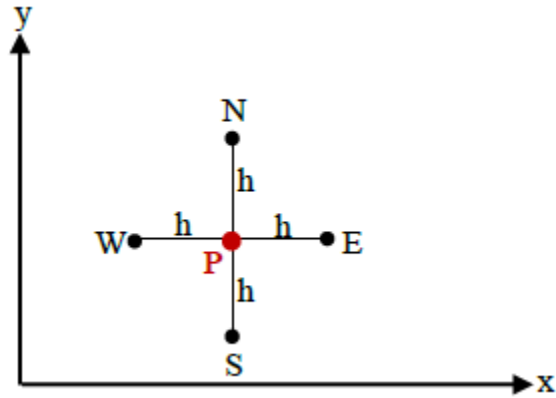


Figure 1. Typical point P with its 4 neighbours in a 2D region with uniform grid spacing

3 NUMERICAL METHOD CHOSEN TO DISCRETISE THE CONSTITUTIVE EQUATION

The Finite Difference Method allows to transform differential equations, as the thermal diffusion equation presented on Chapter 2, into several algebraic equations with solution depending on the initial and boundary conditions. The previous is possible because the derivatives from the differential equation will be substituted by forward, backward and central differences approximations according to Taylor series expansion.

There are different methods that could be applied depending on the accuracy, convergence, stability and speed required by the results. There are Explicit, Implicit, Combined and Alternating Direction Implicit methods.

For the scope of the assignment, the Explicit method was chosen in order to solve the thermal diffusion equation that allows a finite-difference representation of the equation, by using the forward difference approximation to calculate $\partial T / \partial t$ and the central difference approximation to calculate $\nabla^2 T$.

$$\nabla^2 T = \frac{1}{\alpha} \cdot \frac{\partial T}{\partial t}$$

Using the central difference approximation:

$$\nabla^2 T = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}$$

Where,

$$\begin{cases} \frac{\partial^2 T}{\partial x^2} = \frac{T_E^t - T_P^t + T_W^t}{h^2} \\ \frac{\partial^2 T}{\partial y^2} = \frac{T_N^t - T_P^t + T_S^t}{h^2} \end{cases}$$

And using the forward difference approximation:

$$\frac{\partial T}{\partial t} = \frac{T_P^{t+1} - T_P^t}{\Delta t}$$

Substituting in the thermal diffusion equation, it will end up as follows:

$$T_P^{t+1} = F_0 \left(T_E^t + T_W^t + T_N^t + T_S^t - \left(4 - \frac{h^2}{F_0} \right) \cdot T_P^t \right)$$

The Finite Difference equation given above is called the explicit form, where there is only one unknown, T_P^{t+1} , that can be explicitly resolved knowing all the temperatures in previous steps.

Where F_0 , the Fourier number, is equal to:

$$F_0 = \frac{\alpha \cdot \Delta t}{h^2}$$

and the Fourier number cannot exceed the value imposed by the stability criterion:

$$0 < F_0 \leq \frac{1}{4}$$

4 INITIAL AND BOUNDARY CONDITIONS

4.1 ASSIGNMENT PLATE

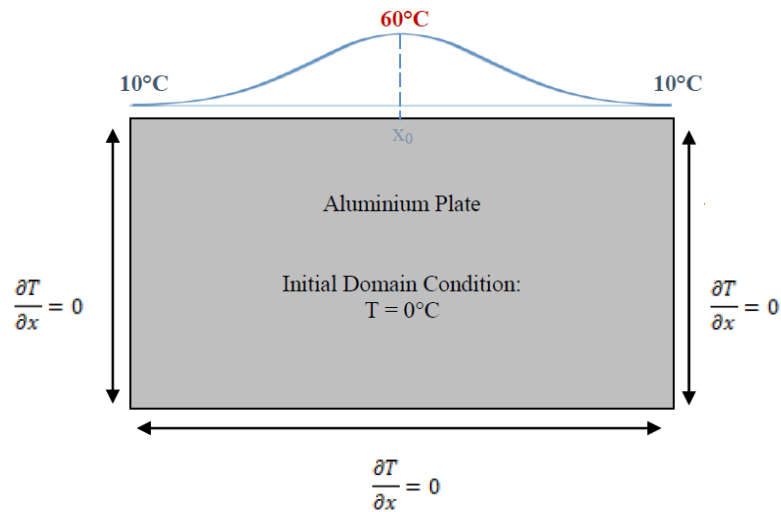


Figure 2. Initial and Boundary Conditions of aluminium plate from assignment brief

Initial Conditions:

- The initial domain condition, when $t = 0$, is the temperature being equal to 0°C in all the domain nodes;

Boundary Conditions:

- North boundary with a Gaussian distribution of temperatures with an amplitude of 60°C ;

Where,

$$T(x) = A \cdot e^{-\frac{(x-x_0)^2}{2\sigma^2}}$$

- Perfectly well insulated West, East and South boundary, meaning that there is no heat transfer / flow through it;

4.2 VALIDATION – TEST PLATE

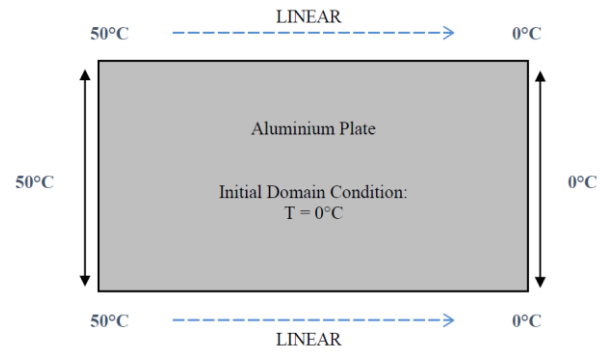


Figure 3. Initial and Boundary Conditions of test aluminium plate used for validation

Initial Conditions:

- The initial domain condition, when $t = 0$, is the temperature being equal to 0°C in all the domain nodes;

Boundary Conditions:

- West boundary with a constant temperature of 50°C ;
- East boundary with a constant temperature of 0°C ;
- North and South boundaries with a linear distribution on temperature from West to East;

5 FINITE DIFFERENCE MODEL (SCRIPT)

The problem proposed by the assignment does not present an analytical solution, reason why it was important to apply the Finite Difference Method to an alternative problem with a known and exact solution, in a way to validate the method used.

Therefore, the script written for the assignment, “Antonio_Januario_180556.m”, was then adapted to solve the transient thermal diffusion problem with the exact solution – “Validation_Antonio_Januario_18055.m”.

Description of the script “Validation_Antonio_Januario_180556.m”:

- The script starts by defining all the constants used in the chosen Finite Difference Method;

```
10 %Constants:
11 %Conductivity (W/m.C); Density (kg/m3); Specific Heat (J/kg.C); Thermal diffusivity (m2/s)
12 k = 143;
13 ro = 2.8*10^3;
14 c = 795;
15 alfa = k/(ro*c);
```

- Then, it asks the user to define the dimension of the plate along the x – axis and y – axis, while opening two cycles asking the user to introduce the grid spacing and the time step to be used;

```
17 fprintf('TEST OF TRANSIENT THERMAL DIFFUSION MODEL USING THE FINITE DIFFERENCE METHOD\n\n')
18 x = input('Define the dimension of the plate along the x-axis(m):');
19 y = input('Define the dimension of the plate along the y-axis(m):');
```

- The second cycle asks for the grid spacing value and analyses if it complies with the plate's dimension, assuring that it is inferior to the plate's x and y axis and that the division of the axis values by the grid value results in an integer value;

```
21 rep1 = true;
22 rep2 = true;
23 while rep1
24     while rep2
25         h = input('Choose the grid spacing(m):');
26         if h > x || h > y || rem(x,h) ~= 0 || rem(y,h) ~= 0
27             fprintf('\nGrid spacing was not correct for the current plate dimensions!\n\n')
28         else
29             rep2=false;
30         end
31     end
```

- The first cycle asks the user to choose a value for the time step and calculates the Fourier number until this value doesn't exceed the value imposed by the stability criterion;

```

32 - t = input('Choose a value for the time step(s):');
33 - %Stability condition
34 - Fo = alfa*t/(h^2);
35 - if Fo<0 || Fo>=1/4
36 -     fprintf('\nStability condition was not met!\n\n')
37 - else
38 -     repl=false;
39 - end
40 - end

```

- Afterwards, it defines the initial and boundary conditions. It starts by defining the size of the plate and the initial domain conditions, followed by the boundary conditions, already explained on Chapter 4.2 Validation – Test Plate;

```

42 - %CALCULATE TEMPERATURE OF THE PLATE
43 - %Boundary Conditions and Initial Domain Conditions
44 - plate{1,1} = zeros(int8(y/h+1),x/h+1);
45 - plate{1,1}(:,1) = 50;
46 - plate{1,1}(:,x/h+1) = 0;
47 - plate{1,1}(1,:) = 50:-(50/(x/h)):0;
48 - plate{1,1}(int8(y/h+1),:) = 50:-(50/(x/h)):0;

```

- Subsequently, it calculates the temperatures of the plate nodes at each time step until the thermal equilibrium condition is met. To calculate the temperatures of the nodes, the explicit method explained on Chapter 3, Numerical method chosen to discretised the constitutive equation, was used;

```

50 - %Calculate temperature inside the plate at each time step
51 - rep3 = true;
52 - j = 2;
53 - while rep3
54 -     plate{1,j} = plate{1,j-1};
55 -     for k = 2:(size(plate{1,1},1)-1)
56 -         for m = 2:(size(plate{1,1},2)-1)
57 -             plate{1,j}(k,m) = Fo*(plate{1,j-1}(k,m+1)+plate{1,j-1}(k,m-1)+pl
58 -         end
59 -     end

```

- The criteria that was utilised to state that thermal equilibrium was met was that when the value of highest temperature difference of the plate between time steps was inferior to 0.001;

```

61 - %Temperature convergence = thermal equilibrium
62 - Compare = ((plate{1,j})-(plate{1,j-1}));
63 - Compare_max = max(Compare(:));
64 - if Compare_max > 0.001
65 -     j = j+1;
66 - else
67 -     rep3=false;
68 - end
69 - end

```

- Once thermal equilibrium is achieved, the script prints the time (in seconds) that it takes for the plate to achieve the thermal equilibrium and displays a figure with the temperatures inside the plate at each step. The pause function was used to assure that the user was able to observe the temperature changes once the figure was displayed.

```

71 - fprintf('\n\nThe time for the plate to be in thermal equilibrium is %.0fs!',t*j);
72 -
73 - %Display the temperatures inside the plate at each time step
74 - colormap jet
75 - contourf(plate{1,1})
76 - colorbar
77 - title('Validation Plate for the Finite Difference Method')
78 - xlabel('N° of nodes in x-axis')
79 - ylabel('N° of nodes in y-axis')
80 - hold on
81 - for l = 2: j
82 -     contourf(plate{1,l})
83 -     hold on
84 -     pause(1/(10^7))
85 - end

```

6 ANALYSIS OF CODE RESULTS

6.1 VALIDATION – TEST PLATE

To serve as an example, once running the script “Validation_Januario_Antonio_1805566” the settings were defined as following:

VALIDATION – TRANSIENT THERMAL DIFFUSION MODEL USING THE FINITE DIFFERENCE METHOD

```
Define the dimension of the plate along the x-axis(m):1
Define the dimension of the plate along the y-axis(m):0.5
Choose the grid spacing(m):0.1
Choose a value for the time step(s):2
```

Figure 4. MATLAB's user menu from Validation_Januario_Antonio_1805566.m

The time to achieve thermal equilibrium is displayed straight after, as well as the figure with the temperature evolution inside the plate at each step.

The time for the plate to be in thermal equilibrium is 1760s!>>

Figure 5. Print result from Validation_Januario_Antonio_1805566.m

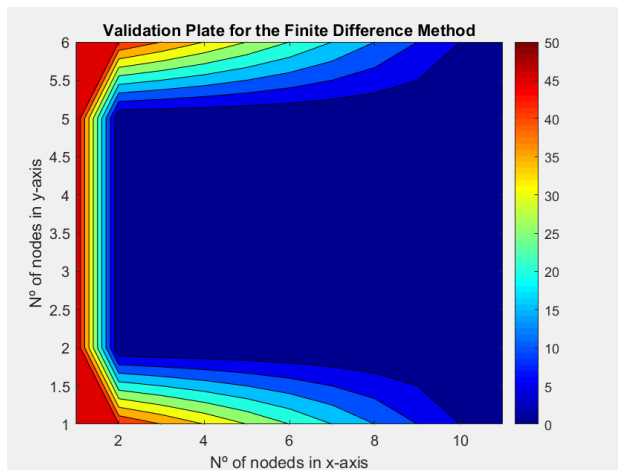


Figure 6. Plate temperatures when $t = 0$ from Validation_Januario_Antonio_180556.m

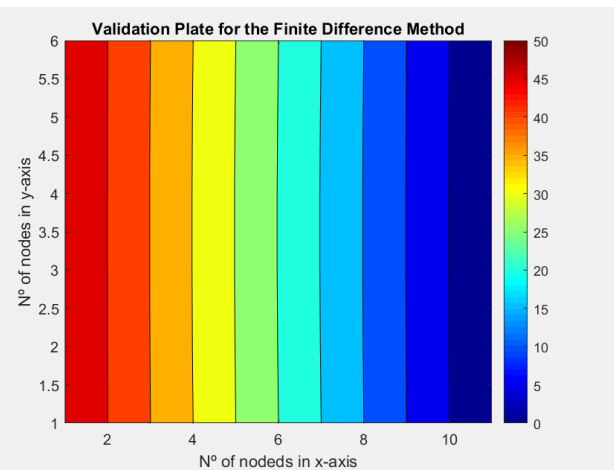


Figure 7. Plate temperatures when thermal equilibrium is reached from Validation_Januario_Antonio_180556.m

The results observed in figure 7, when thermal equilibrium was achieved, were exactly what was expected by the analytical solution of this problem.

These results allow to validate the Finite Difference Method used in this assignment, ensuring that the temperatures result from the assignment's aluminium plate are correct, even though there is no analytical solution for that problem - as it will be observed in the next chapter – 6.2 Assignment Plate.

6.2 ASSIGNMENT PLATE

To serve as an example, once running the script “Januario_Antonio_1805566” the settings were defined as following:

TRANSIENT THERMAL DIFFUSION MODEL USING THE FINITE DIFFERENCE METHOD

Define the dimension of the plate along the x-axis(m):1

Define the dimension of the plate along the y-axis(m):0.5

Choose the grid spacing(m):0.1

Choose a value for the time step(s):2

Figure 8. Matlab's user menu from Januario_Antonio_1805566.m

The time to achieve thermal equilibrium is displayed straight after as well as the figure with the temperature evolution inside the plate at each step.

The time for the plate to be in thermal equilibrium is 6492s!

Figure 9. Print result from Januario_Antonio_1805566.m

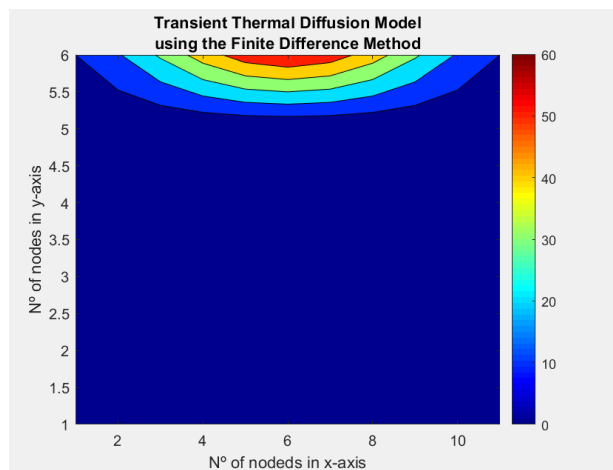


Figure 11. Plate temperatures when $t = 0$ from Januario_Antonio_180556.m

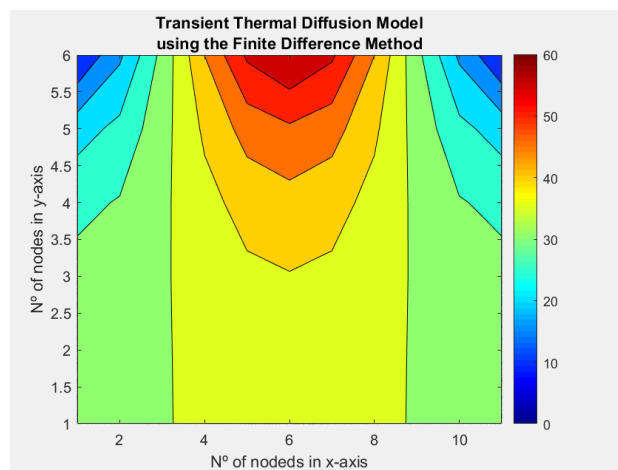


Figure 10. Plate temperatures when thermal equilibrium is reached from Januario_Antonio_180556.m

With the initial and boundary conditions stated on Chapter 4.2 – ASSIGNMENT PLATE, there was no analytical solution that could help predict the heat transfer along the aluminium plate.

Once a validation test was performed in the previous chapter (6.1 – Validation – Test Plate), the results obtained on figure 10, when thermal equilibrium is met, are correct!

7 LIMITATIONS OF THE MODEL

The limitations of the model are related to the chosen method to solve the problem, which for this assignment was the Explicit Method.

For a plate with a certain dimension and grid spacing, the results obtained are dependent on the time step inserted by the user. This dependency could be observed by analysing the finite difference equation below used to calculate the temperatures in each node.

$$T_P^{t+1} = F_0 \left(T_E^t + T_W^t + T_N^t + T_S^t - \left(4 - \frac{h^2}{F_0} \right) \cdot T_P^t \right)$$

Where,

$$F_0 = \frac{\alpha \cdot \Delta t}{h^2}$$

It was observed that, when increasing the time step, even though the number of iterations that the model needed to reach thermal equilibrium was decreasing, the total time in seconds was increasing.

Time Step (s)	Time to reach thermal equilibrium (s)
1	5389
2	6492
4	7592
6	8238

Another disadvantage of this method, still related with the time step, is that once α and h are defined, the Δt value that can be chosen by the user is limited due to the fact that it needs to respect the following stability criterion:

$$0 < F_0 \leq \frac{1}{4}$$

8 FUTHER IMPROVEMENTS

The model developed by the author could be an object of several improvements. As it was observed in the previous chapter, 7 – Limitations of the model, there are some subjects that could be addressed to improve the model's results.

The model should introduce the calculation of the error related to the use of the Explicit Method, when performing the substitutions of the differential equations by the forward and central difference approximations in the thermal diffusion equation.

The model could indicate the user a range of values for the grid spacing and time step to insert in order to reduce the error induced by the finite difference method used and get results as close as possible to an analytical solution. This will allow the model to be able to calculate a numerical solution closer to the results observed in a real problem scenario.

Depending on the accuracy, convergence, stability and speed of the results intended by the user, the model should allow the user to select the best suitable method to calculate the differential equation from the problem (examples: Explicit, Implicit, Combined and Alternating Direction Implicit method).

The code developed by the author could be more capable, allowing the user to redefine the initial and boundary conditions of the problem being analysed. This feature will allow to analyse multiple different scenarios without the need to change the MATLAB code to do so. An option could also be added to allow the comparison between two or more solutions.

9 CONCLUSION

This project allowed the author to solve a specific heat transfer problem, without an analytical solution, through the application of a finite difference method.

Prior to get the approximated solution for this heat transfer problem, it was required to build a test problem with a known analytical solution in order to validate the finite difference method implemented.

The method implemented was successfully validated, as the results obtained in the test problem perfectly matched the expected by the analytical solution. This meant that this method could be used in the assignment's problem to obtain a coherent solution.

The results obtained from the assignment plate matched the expectations. The temperature around the cold sources, in the northeast and northwest corners, was colder than the rest of the plate and the hot temperature was transferred from the hotter source, with peak temperature in the middle of the north boundary, to the rest of the plate with the heat waves propagating from the middle of the plate to the sides.

It was also observed that the time for the aluminium plate to reach thermal equilibrium was deeply affected by the time step chosen to perform the calculations and that the time step value was limited by the stability criterion.

Nevertheless, the finite difference method used in the model developed by the author, turned out to be a powerful tool to solve heat transfer problems without an analytical solution. In a real engineering problem, where it is common for the analysed problems not to have analytical solutions, the tool developed will allow to for better estimations, quicker progress and consequently less cost.

10 BIBLIOGRAPHY

- Marotin, A. (2019). 15. Thermal Diffusion Model using the Finite Difference Method
- Croft, D., & Lilley, D. G. (1986). Heat Transfer Calculation Using Finite Difference Equations. Sheffield Hallam University Press.
- J.N.Reddy, An Introduction to the Finite Element Method, Third Edition

11 APPENDIX

In this chapter it will be attached the full script of both models used to develop the current assignment.

11.1 Validation _Januario _Antonio _1805566.m

```
Validation_Januario_Antonio_1805566.m  X  +
1 - clear all
2 - clc
3 - %MATLAB CODE TEST
4 - %Matlab Assignment
5 - %Student: Antonio Januario
6 - %Number: 1805566
7 - %TRANSIENT THERMAL DIFFUSION MODEL using the Finite Difference Method
8 - %TEST - CODE TO VALIDATE FINITE DIFFERENCE METHOD USED
9
10 %Constants:
11 %Conductivity (W/m.C); Density (kg/m3); Specific Heat (J/kg.C); Thermal diffusivity (m2/s))
12 - k = 143;
13 - ro = 2.8*10^3;
14 - c = 795;
15 - alfa = k/(ro*c);
16
17 - fprintf('VALIDATION - TRANSIENT THERMAL DIFFUSION MODEL USING THE FINITE DIFFERENCE METHOD\n\n')
18 - x = input('Define the dimension of the plate along the x-axis(m):');
19 - y = input('Define the dimension of the plate along the y-axis(m):');
20
21 - repl = true;
22 - rep2 = true;
23 - while repl
24 -     while rep2
25 -         h = input('Choose the grid spacing(m):');
26 -         if h > x || h > y || rem(x,h) ~= 0 || rem(y,h) ~= 0
27 -             fprintf('\nGrid spacing was not correct for the current plate dimensions!\n\n')
28 -         else
29 -             rep2=false;
30 -         end
31 -     end
32 -     t = input('Choose a value for the time step(s):');
33 -     %Stability condition
34 -     Fo = alfa*t/(h^2);
35 -     if Fo<0 || Fo>=1/4
36 -         fprintf('\nStability condition was not met!\n\n')
37 -     else
38 -         repl=false;
39 -     end
40 - end
41
42 %CALCULATE TEMPERATURE OF THE PLATE
43 %Boundary Conditions and Initial Domain Conditions
44 - plate{1,1} = zeros(int8(y/h+1),x/h+1);
45 - plate{1,1}(:,1) = 50;
46 - plate{1,1}(:,x/h+1) = 0;
47 - plate{1,1}(1,:) = 50:- (50/(x/h)):0;
48 - plate{1,1}(int8(y/h+1),:) = 50:- (50/(x/h)):0;
49
```

```

50 %Calculate temperature inside the plate at each time step
51 rep3 = true;
52 j = 2;
53 while rep3
54     plate{1,j} = plate{1,j-1};
55     for k = 2:(size(plate{1,1},1)-1)
56         for m = 2: (size(plate{1,1},2)-1)
57             plate{1,j}(k,m) = Fo*(plate{1,j-1}(k,m+1)+plate{1,j-1}(k,m-1) ...
58                 +plate{1,j-1}(k-1,m)+plate{1,j-1}(k+1,m) - (4-1/Fo)*plate{1,j-1}(k,m));
59         end
60     end
61
62     %Temperature convergence = thermal equilibrium
63     Compare = ((plate{1,j})-(plate{1,j-1}));
64     Compare_max = max(Compare(:));
65     if Compare_max > 0.001
66         j = j+1;
67     else
68         rep3=false;
69     end
70 end
71
72 fprintf('\n\nThe time for the plate to be in thermal equilibrium is %.0fs!',t*j);
73
74 %Display the temperatures inside the plate at each time step
75 colormap jet
76 contourf(plate{1,1})
77 colorbar
78 title('Validation Plate for the Finite Difference Method')
79 xlabel('N° of nodes in x-axis')
80 ylabel('N° of nodes in y-axis')
81 hold on
82 for l = 2: j
83     contourf(plate{1,l})
84     hold on
85     pause(1/(10^7))
86 end

```

11.2 Januario _ Antonio _ 1805566.m

```
Januario_Antonio_1805566.m +
1 - clear all
2 - clc
3 - %Matlab Assignment
4 - %Student: Antonio Januario
5 - %Number: 1805566
6 - %TRANSIENT THERMAL DIFFUSION MODEL using the Finite Difference Method
7 -
8 - %Constants:
9 - %Conductivity (W/m.C); Density (kg/m3); Specific Heat (J/kg.C); Thermal diffusivity (m2/s))
10 - k = 143;
11 - ro = 2.8*10^3;
12 - c = 795;
13 - alfa = k/(ro*c);
14 -
15 - fprintf('TRANSIENT THERMAL DIFFUSION MODEL USING THE FINITE DIFFERENCE METHOD\n\n')
16 - x = input('Define the dimension of the plate along the x-axis(m):');
17 - y = input('Define the dimension of the plate along the y-axis(m):');
18 -
19 - rep1 = true;
20 - rep2 = true;
21 - while rep1
22 -     while rep2
23 -         h = input('Choose the grid spacing(m):');
24 -         if h > x || h > y || rem(x,h) ~= 0 || rem(y,h) ~= 0
25 -             fprintf('\nGrid spacing was not correct for the current plate dimensions!\n\n')
26 -         else
27 -             rep2=false;
28 -         end
29 -     end
30 -     t = input('Choose a value for the time step(s):');
31 -
32 -     %Stability condition
33 -     Fo = alfa*t/(h^2);
34 -     if Fo<0 || Fo>=1/4
35 -         fprintf('\nStability condition was not met!\n\n')
36 -     else
37 -         rep1=false;
38 -     end
39 - end
```

```

40
41 %CALCULATE TEMPERATURE OF THE PLATE
42 %Boundary Conditions and Initial Domain Conditions
43 plate{1,1} = zeros(int8(y/h+2),int8(x/h+3));
44 x_vector = -x/2:h:x/2;
45 %North domain conditions using standard deviation formula
46 sig = ((-(0-(x/2))^2)/(2*log(10/60)))^(1/2);
47 for i = 1:(length(x_vector))
48     plate{1,1}(1,i+1) = 60 * exp(-(x_vector(i)-0)^2/(2*sig^2));
49 end
50
51 %Calculate temperature inside the plate at each time step
52 rep3 = true;
53 j = 2;
54 while rep3
55     plate{1,j} = plate{1,j-1};
56     for k = 2:(size(plate{1,1},1)-1)
57         for m = 2:(size(plate{1,1},2)-1)
58             plate{1,j}(k,m) = Fo*(plate{1,j-1}(k,m+1)+plate{1,j-1}(k,m-1) ...
59                 +plate{1,j-1}(k-1,m)+plate{1,j-1}(k+1,m)-(4-1/Fo)*plate{1,j-1}(k,m));
60         end
61     end
62     plate{1,j}(:,1) = plate{1,j}(:,3);
63     plate{1,j}(:,m+1) = plate{1,j}(:,m-1);
64     plate{1,j}(k+1,:) = plate{1,j}(k-1,:);
65
66     %Temperature convergence = thermal equilibrium
67     Compare = ((plate{1,j})-(plate{1,j-1}));
68     Compare_max = max(Compare(:));
69     if Compare_max > 0.001
70         j = j+1;
71     else
72         rep3=false;
73     end
74 end
75 fprintf('\n\nThe time for the plate to be in thermal equilibrium is %.0fs!',t*j);
76
77 %Display the temperatures inside the plate at each time step
78 colormap jet
79 contourf(plate{1,1}(((size(plate{1,1},1))-1):-1:1,((size(plate{1,1},2))-1):-1:2))
80 colorbar
81 title({'Transient Thermal Diffusion Model','using the Finite Difference Method'})
82 xlabel('N° of nodeds in x-axis')
83 ylabel('N° of nodes in y-axis')
84 hold on
85 for l = 2: j
86     contourf(plate{1,1}(((size(plate{1,1},1))-1):-1:1,((size(plate{1,1},2))-1):-1:2))
87     hold on
88     pause(1/(10^8))
89 end

```