ME2 Computing- Coursework summary

Student(s): Pablo Capell Abril – 01938515
Varun Raaghav – 01847960

Words: 984/1000

A) What physics are you trying to model and analyse? (Describe clearly, in words, what physical phenomenon you wish to analyse)

We are analysing the cooling of a computer chip that has overheated in operation. There is no heat generation source as the power from the computer to the chip has been cut-off. The cooling is done on one side through a water-cooling system, where the water is at 10℃. There are two fans on 2 opposite sides of the chip that keep those boundaries at a constant temperature of 30℃. The final boundary is in contact with another chip, which means that the layer is maintained at a constant temperature of 60℃.
The cooling over a period of 2000s is analysed

B) What PDE are you trying to solve, associated with the Physics described in A? (write the PDE)

We are trying to solve the heat diffusion equation for two dimensions in space (x, y) and one dimension in time (t).

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

Where T is the temperature, and α is the thermal diffusivity of the material.

C) Boundary value and/or initial values for my specific problem: (be CONSISTENT with what you wrote in A)

As the chip has overheated and is being cooled down, we first set all the points inside the grid to 120℃ which is the limit of operation for a modern computer chip. We then proceeded to set the boundary conditions. (note, the chip is of width 2mm and height 1mm. The grid is spaced by 15 rows and 30 columns. Δy = 2/30 = 0.06667. For ease of programming, the grid has been made such that Δx = Δy). The Δt is chosen as 8 s

<u>Neumann Boundary Condition</u> (For $T_{x,y=1}$)

$$\left.\frac{\partial T^{k+1}_{x,y=1}}{\partial x}\right|_{y=1} = -\frac{h}{k_t}\times\left(T^{k}_{x,y=1}-T_\infty\right) \qquad (2)$$

where h (heat transfer coefficient of the water) = 10^{-6} (W/mm^{2}K) , k_t (thermal conductivity of copper chip per metre) = 0.385 (W/mmK), and $T_\infty = 10$℃

- *Note here that a variable Neumann boundary condition is used to more accurately represent the heat transfer of the water cooling system, as it is unlikely that $\partial T/\partial x$ is constant for all time. Therefore, using 'k' as the time index: $\frac{\partial T[k+1]}{\partial x} = function(T[k])$, and this function is outlined in equation 2*

<u>Dirichlett Boundary Conditions:</u>

$$T_{x=0,y}\ and\ T_{x=2,y}=30, T_{x,y=0}=60℃ \qquad (3)$$

Also since it is assumed that at time (t) = 0, the water cooling is already turned on. Therefore, the top initial condition will be slighltly below T=120℃. Ie at t=0:

$$T^{ONLY\ at\ t=0}_{x,y=1} = 118℃ \qquad (4)$$

D) What numerical method are you going to deploy and why? (Describe, in words, which method you intend to apply and why you have chosen it as opposed to other alternatives)

We have used the explicit method to solve the PDE. It is known that using this method means that the solution is not conditionally stable – ie. will not converge if the space steps are too small in relation to the time step. Therefore, we ensure conditional stability by checking for the diffusion number (or the Courant number) by using the equation

$$d=\frac{\alpha\times\Delta t}{\Delta y^2} \qquad (5)$$

And the solution is stable if d<0.25
For the grid chosen d= 0.1998. Therefore it is stable and convergent. If the grid was made finer, it would no longer be convergent (assuming Δt is unchanged)

The alternative would be to use implicit methods like Crank-Nicolson to solve the PDE by creating a system of equations at each time step and using linear algebra techniques to solve the system of equations. But it was decided not to use this method as the complexity of doing Crank-Nicolson in 2D means that the system of equations end up being coupled and solving it requires significantly longer code. And the code is not likely to be very efficient as it would be required to populate a matrix in every time step and solve it using matrix inversion or iterative methods such as Gauss-Seidel. Therefore, judging the scope and complexity of this problem it was decided to use an explicit method. The drawback is having not an extremely fine grid mesh, but the chip is modelled at a small 1x2mm size, so a very fine mesh is not necessary to have meaningful solutions.

E) I am going to discretise my PDE as the following (show the steps from continuous to discrete equation and boundary/initial conditions:

$$\alpha\left(\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}\right)=\frac{\partial T}{\partial t} \tag{6}$$

A central difference approximation is used for the x and y special discretisation:

$$\frac{\partial^2 T}{\partial x^2}=\frac{T_{i+1}-2T_i+T_{i-1}}{\Delta x^2} \tag{7}$$

$$\frac{\partial^2 T}{\partial y^2}=\frac{T_{j+1}-2T_j+T_{j-1}}{\Delta y^2} \tag{8}$$

We then used the forward finite difference scheme (ie. explicit) to solve for the time discretisation (k is the time index):

$$\frac{\partial T}{\partial t}=\frac{T_{i,j}^{k+1}-T_{i,j}^{k}}{\Delta t} \tag{9}$$

Therefore, the PDE is discretised as

$$\frac{T_{i,j}^{k+1}-T_{i,j}^{k}}{\Delta t}=\alpha\left(\frac{T_{i+1,j}^{k}-2T_{i,j}^{k}+T_{i-1,j}^{k}}{\Delta x^2}+\frac{T_{i,j+1}^{k}-2T_{i,j}^{k}+T_{i,j-1}^{k}}{\Delta y^2}\right) \tag{10}$$

For simplicity, it is chosen such that Δx = Δy; and *d* (diffusion number: eqn 5) is such that:

$$d=\frac{\alpha\times\Delta t}{\Delta y^2}$$

Therefore, the final discretised PDE is

$$T_{i,j}^{k+1}=T_{i,j}^{k}+d\times\left(T_{i+1,j}^{k}+T_{i-1,j}^{k}+T_{i,j+1}^{k}+T_{i,j-1}^{k}-4T_{i,j}^{k}\right) \tag{11}$$

F) Plot the numerical results comprehensively and discuss them (discuss how the results describe the physics and comment on any discrepancies or unexpected behaviours). Use multiple types of visual graphs. Present and discuss any outcomes of the grid analysis, as requested in Task 8, too.

Figure 1 and 2 below show the surface and contour plot of the chip at time = 0 s (ie Initial conditions)

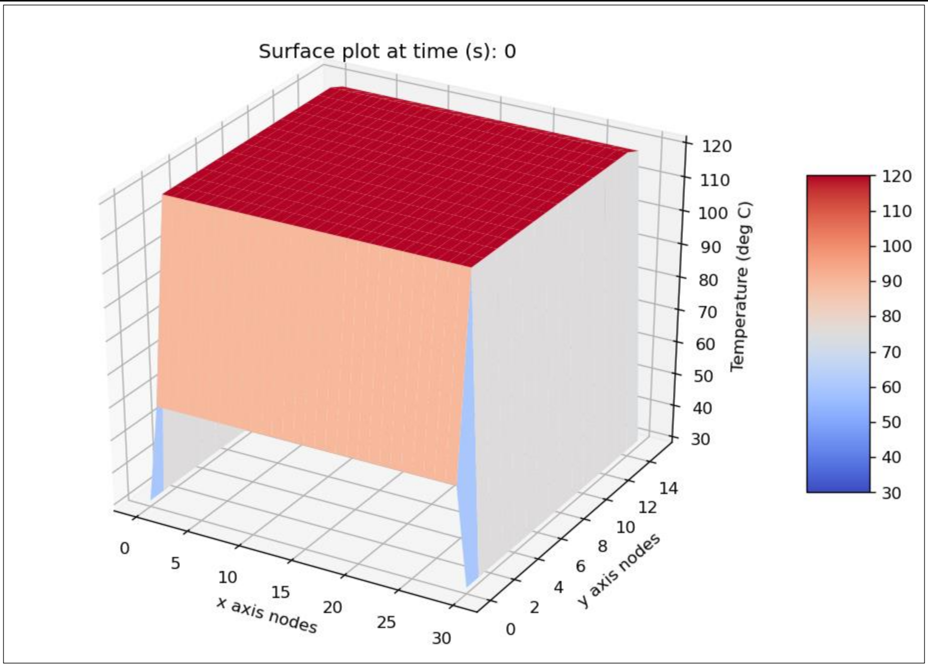

Figure 1: Surface plot of microchip at t=0

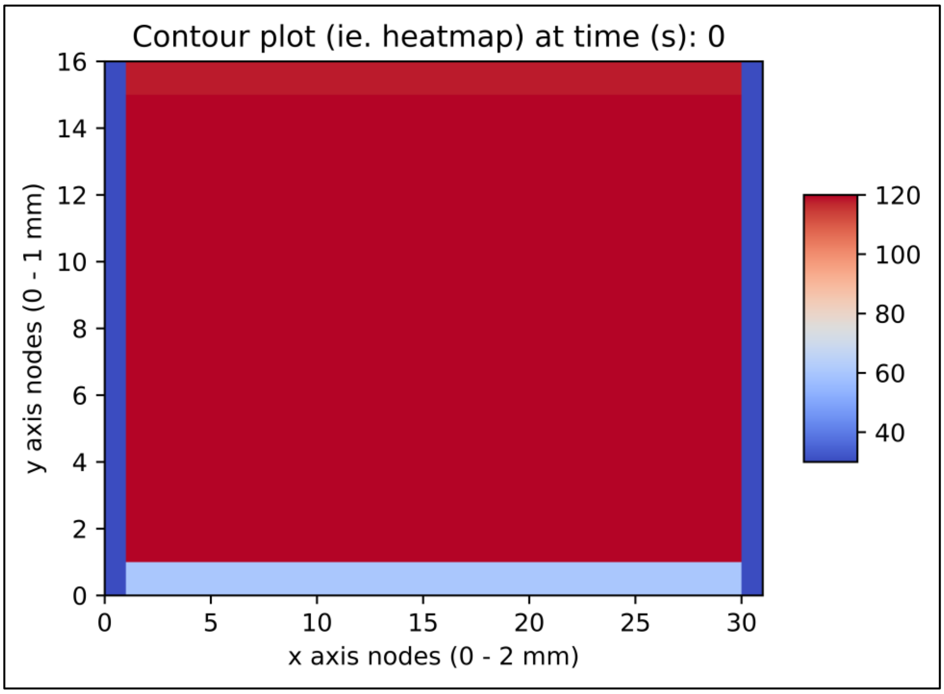

Figure 2: Contour plot of chip at initial conditions

Figure 3-12 below show how the temperature of the chip varies through time. The time at which the graphs are taken are shown in the caption and the title. Both surface and contour plots are shown to aid the visual representation of the temperature changing with time.

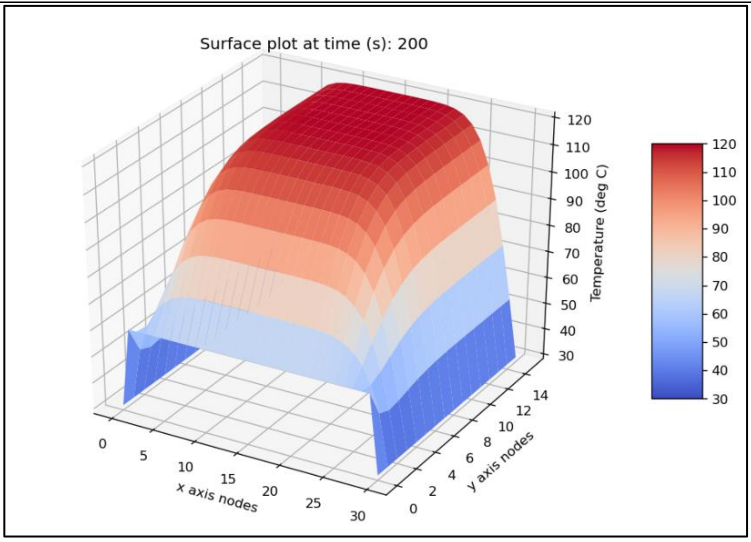

Figure 3: Surface plot at t=200 s

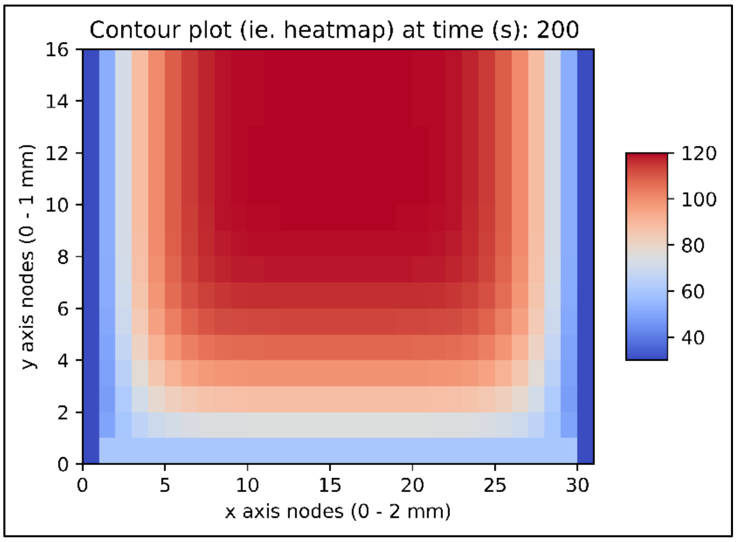

Figure 4: Contour plot at t=200 s

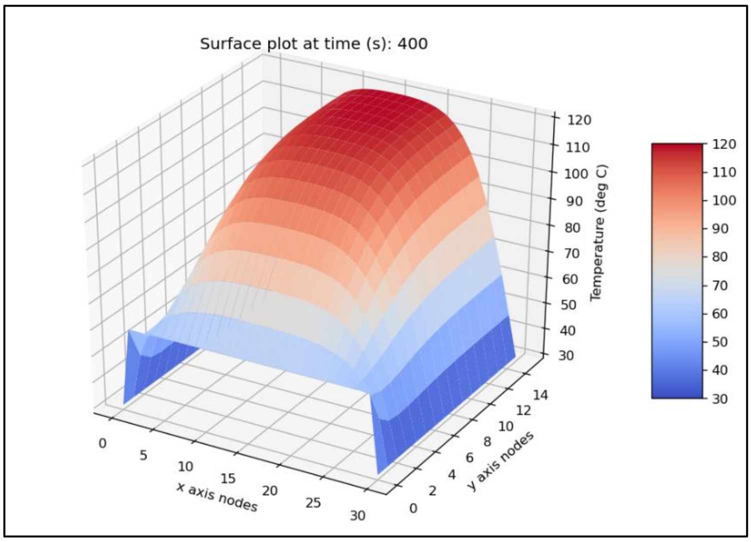

Figure 5: Surface plot at t=400 s

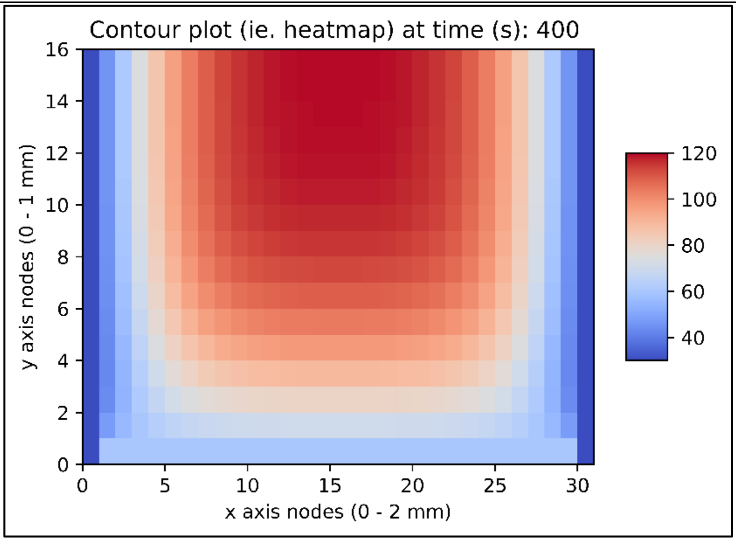

Figure 6: Contour plot at t=400 s

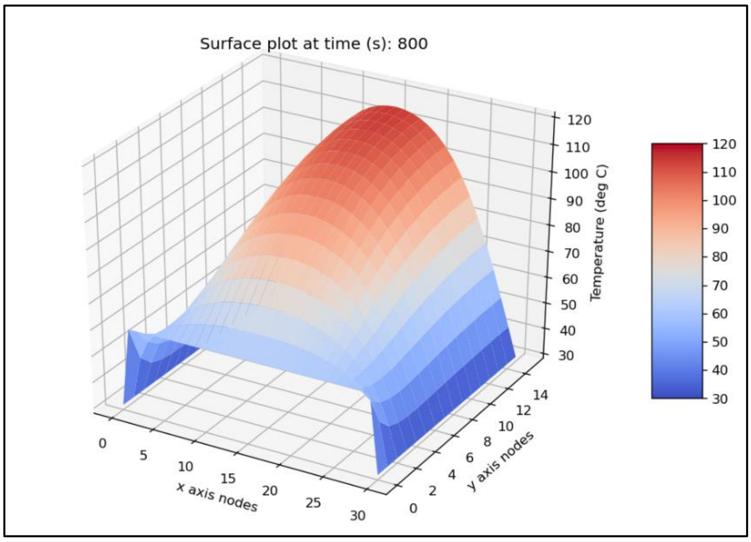

Figure 7: Surface plot at t=800 s

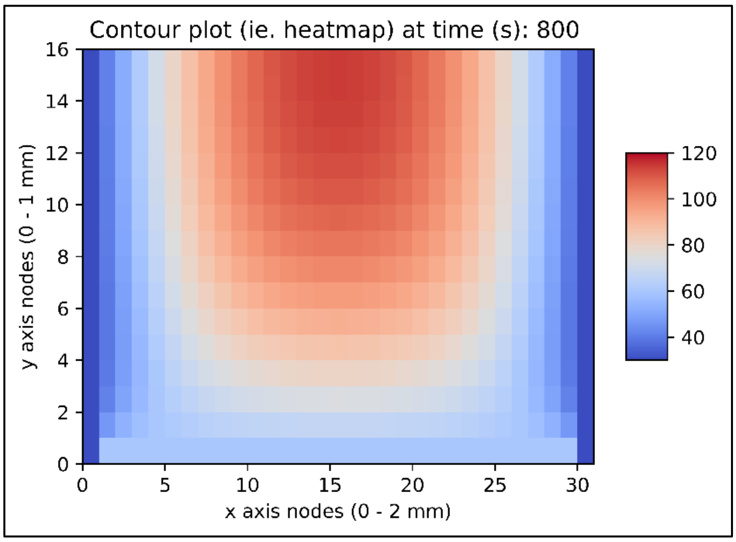

Figure 8: Contour plot at t=800 s

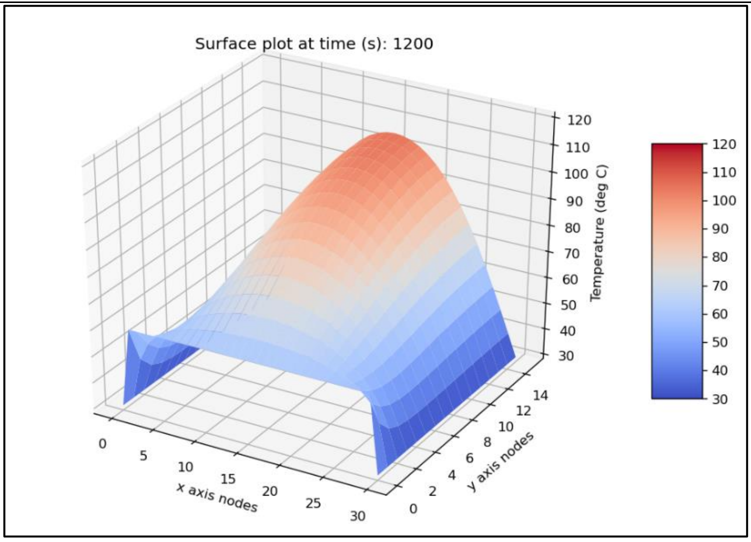

Figure 9: Surface plot at t=1200 s

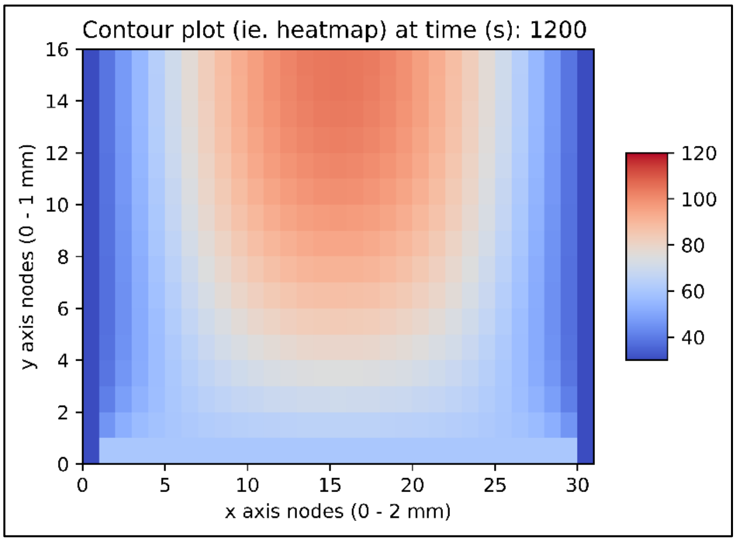

Figure 10: Contour plot at t=1200 s

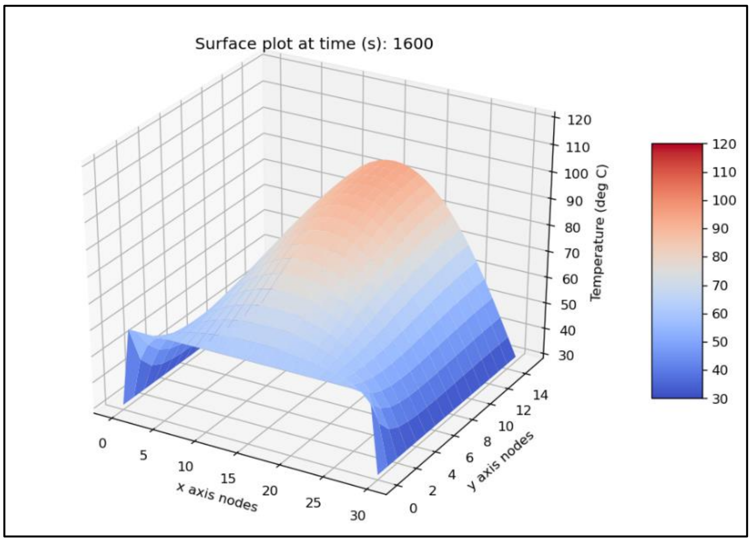

Figure 11: Surface plot at t=1600 s

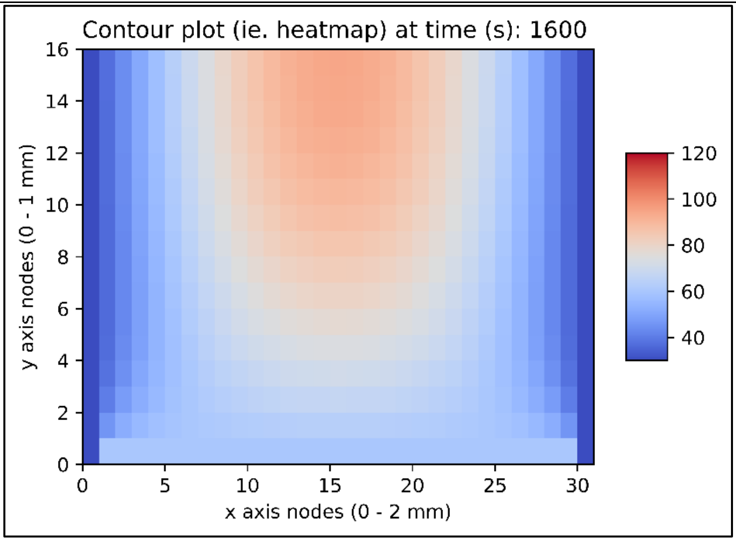

Figure 12: Contour plot at t=1600 s

And finally, the temperature distribution after 2000s is shown below. For this project, the final timestep is 2000s. However, this can be easily adjusted if required in the python code.

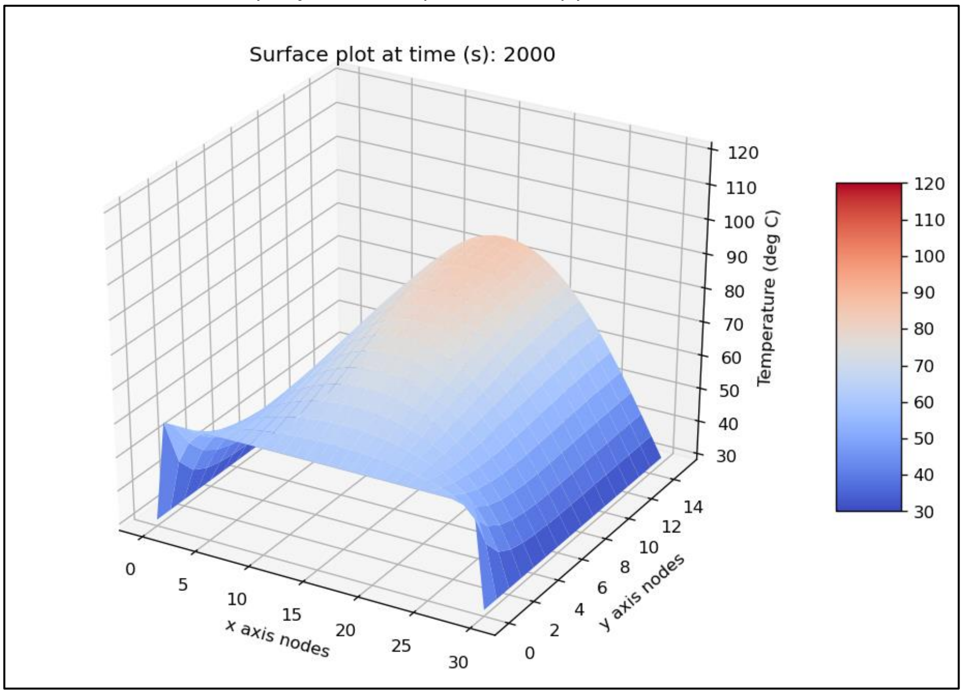

Figure 13: Surface plot – final conditions

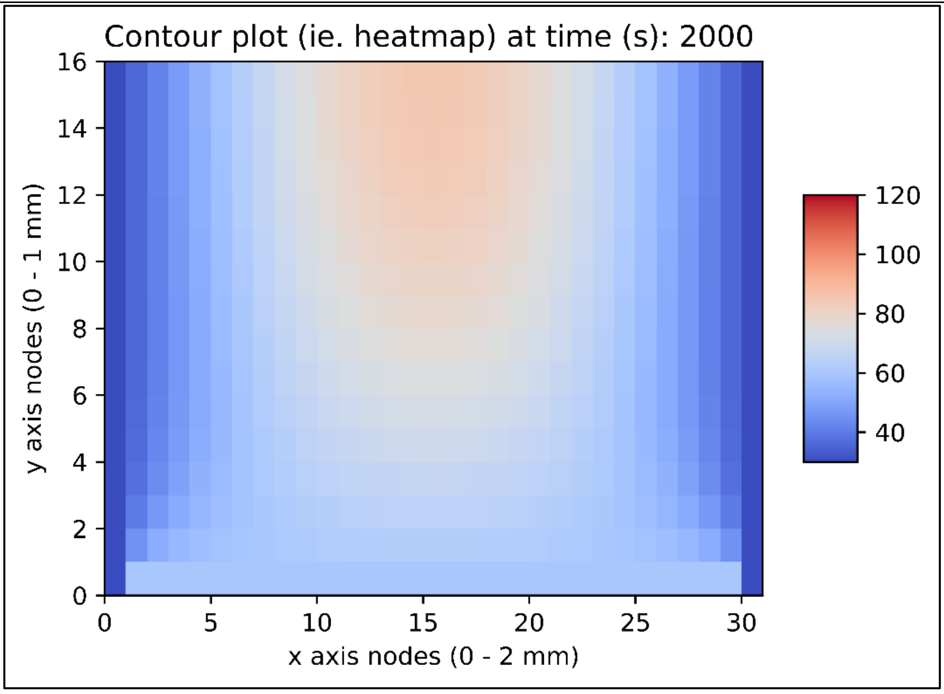

Figure 14: Contour plot – final conditions

As seen from the figures above, the Dirichlet conditions for the left, right and bottom nodes stay constant through time. The top layer is a neumann condition, which is approximated as a function of the previous time step. The grid size used in the solutions above is the maximum allowable fineness. Decreasing Δx will result in an unstable solution and as d>0.25. To increase mesh grid fineness and maintain convergence, Δt must be significantly decreased and this is not recommended as it takes a much longer time to solve the PDE. This is a consequence of the explicit method. Figure 15 shows the final solution if it was unstable when the chosen Δx = 0.0333. Δt is unchanged (Δt=8):

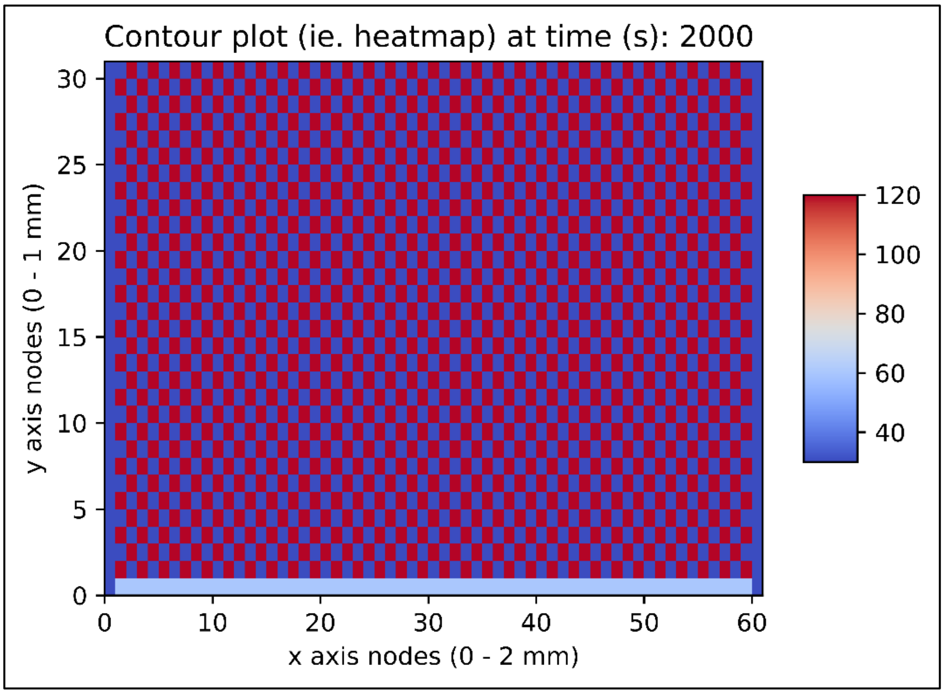

Figure 15: unstable final contour plot - Δx = 0.04mm

To make this mesh grid stable, Δt is decreased to 2s from 8. The stable version of this mesh size is shown below:

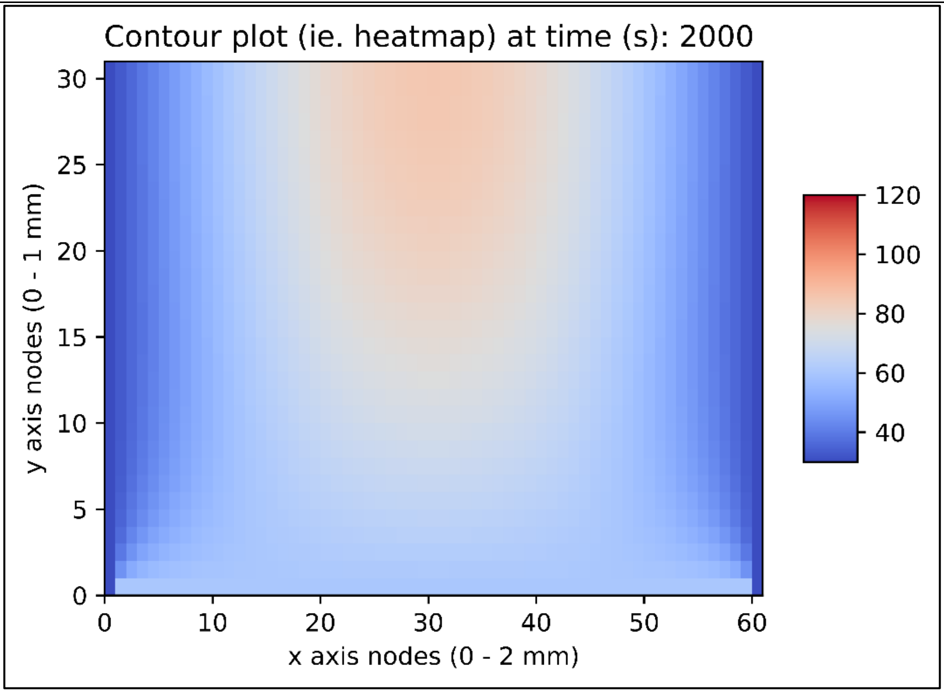

Figure 16: Stable contour plot if Δx = 0.04mm and Δt = 2s

G) Other remarks (limits of the model, convergence problems, possible alternative approaches, anything you find relevant and important to mention):

Since an explicit method is used to solve the PDE, the stability and convergence of the problem play a significant impact when deciding how fine the mesh grid must be, and is limited by the diffusion number (d) being less than 0.25. To increase the mesh grid fineness by a factor of 2 (ie. Δx), the Δt must be decreased by a factor of $2^2 = 4$. This comes at the cost of computational power as it takes significantly longer to solve the problem. The solution to this would be using an implicit method like Crank-Nicolson, which again however comes at the cost of computational power as linear algebra techniques are needed to solve the system of equations.