ME2 Computing Coursework

Hassan - [redacted]
Yung - [redacted]

Department of Mechanical Engineering
Imperial College London

April 2021

1 Physics Model

We are attempting to model the diffusion of heat through a frozen chicken panini (see Figure 1) heated by a panini press (see Figure 2):

Figure 1: Chicken Panini (Porter, 2019)

Figure 2: Panini Press Gourmet Sandwich Maker MODEL: 25460A (Hamilton Beach, n.d.)

To cook a panini with the panini press, the press is first pre-heated to a desired temperature. The panini is then placed on the cooking griddles, and pressure is applied after closing the lid. This process heats both sides of the panini simultaneously, where the bread is toasted and ingredients are heated up. We are interested in the temperature distribution within the panini as this is a critical factor in the taste experience. A panini that is cold on the inside may result in a negative experience. Conversely, a panini that is too hot may cause burning of the mouth.

For simplicity, the following assumptions were made for the physics model:

- The panini is a perfectly-homogeneous solid cuboid with square-faced bread.
- Boundary conditions are imposed symmetrically, thereby allowing the central cross section of the panini to be analysed in 2D.
- There are no air gaps in between or within the solids.
- Thermal contact resistance is negligible between panini constituents.
- No temperature fluctuations exist at boundaries (uniformly distributed temperatures).

- Press has perfectly-flat surface with uniformly distributed temperature.
- The bread and chicken have differing thermal properties, but these properties are invariant in time and space.
- Panini does not experience deformation or internal property changes (e.g. thermal conductivity, k) over time, temperature or pressure changes.

2 PDE model

The 3D heat diffusion equation in Cartesian coordinates is given by equation (1):

$$\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right)+\frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right)+\frac{\partial}{\partial z}\left(k\frac{\partial T}{\partial z}\right)+\dot{q}=\rho c\frac{\partial T}{\partial t} \tag{1}$$

where T is temperature, x, y, z are the 3D Cartesian coordinates, $\dot{q}$ is internal heat generation, k is thermal conductivity, ρ is material density, c is specific heat capacity at constant pressure, and t is time.

Since thermal conductivity k is assumed to be a constant property regardless of space or time (across a single panini component), k can be treated as a scalar constant. We assume the sandwich is square, and conduct an analysis which is two dimensional in space, thereby allowing $\frac{\partial^2 T}{\partial z^2}=0$. There is no generation of heat within the panini and hence $\dot{q}=0$.

The heat diffusion equation can therefore be simplified to:

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

$$\alpha=\frac{k}{\rho c} \tag{3}$$

where α is the thermal diffusivity defined in equation (3). Finally, equation (2) can be simplified to:

$$\frac{\partial T}{\partial t}-\alpha\left(\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}\right)=0 \tag{4}$$

This equation is in the form of a parabolic PDE bounded in space but open-ended in time.

3 Boundary & Initial Values

3.1 Initial Values

The model defines several properties:

- Bread thickness
- Chicken thickness
- Side length of sandwich
- Initial temperature of entire panini
- The convective heat transfer coefficient of the panini (assumed to be globally constant at 10 W/m^{2}K)
- Thermal properties of the bread and chicken such as conductivity, density and specific heat capacity at constant pressure
- Ambient air temperature
- Panini press plate temperature

3.2 Boundary Values

It is known that the rate of heat flux due to conduction at the surface of a body is equal to the rate of heat flux due to convection at the surface. This is known as a von Neumann boundary condition and is captured in equation (5). The boundary condition is applied along the height of the panini, but varies as the medium changes between bread and chicken.

$$k\frac{\mathrm{d}T}{\mathrm{d}y} = -h\left(T - T_{\infty}\right) \tag{5}$$

Where the sandwich is in contact with the heating plate, no convection is taking place. Rather, a temperature is imposed by the heating plate and as such this is known as a Dirichlet boundary condition.

For a point on the surface of the bread in contact with the chicken, the thermal diffusivity of the bread at this point, α_{bb}, is different to elsewhere in the bread and is approximated in equation (6). Similarly, the thermal diffusivity of the chicken on the chicken-bread boundary, α_{cb} is approximated in equation (7).

$$\alpha_{bb} = \frac{3\alpha_{bread} + \alpha_{chicken}}{4} \tag{6}$$

$$\alpha_{cb} = \frac{3\alpha_{chicken} + \alpha_{bread}}{4} \tag{7}$$

Strictly speaking, these are not boundary conditions in the mathematical sense, but are included here for the sake of completeness.

4 Numerical Method

The explicit finite difference numerical method is applied to solve the relevant heat diffusion equation, where the problem models 3 variables (2 in space, and 1 in time).

The panini is discretised into a matrix representing a mesh, where each element represents a node in the panini. Moving along the columns of the matrices represents moving along the side length of the panini, and moving down the rows in the matrix represents moving up the height of the panini.

The code creates a null 3D array which can be thought of as a list of null matrices. Each matrix will be a snapshot of the panini at a point in time. The size of this 3D array is determined by a user input of mesh size, time step and total time forecast.

Every element in the first matrix is then filled with the initial temperature value of 0°C which can be altered by editing the variable `initial_temperature`. Each subsequent matrix is then calculated using an explicit discretisation detailed in section 5.

The code then checks the convergence of the numerical method and warns the user of any potential issues if they are detected. The numerical method is then deployed for every time instance, first defining boundary conditions and then calculating all internal nodes with an explicit method. A loading bar is shown to update the user of the calculation process. Note that there are two versions of the code: `Panini_Model.py` implements a loading bar that is visibile in all IDEs, with the exception of Jupyter Notebook. In order to run the code with a functional loading bar in Jupyter, use `Panini_Model_Jupyter.ipynb`.

The code then generates a coloured contour plot (heat map) of the final state of the panini, with dotted lines indicating the chicken-bread boundaries. The figure is saved to a file `panini.png` in the console directory.

The code then begins the process of generating an animation of a heat map of the panini changing of the time. The speed of this video is set by default to x2 real time, and this can be altered by changing the variable `speed_factor`. The animation is then saved to the file `panini.mp4`.

Finally, a summary of the parameters is delivered to the user and warnings are administered where necessary, such as when there is a poor temperature distribution, a risk of burning or a still cold panini. The final temperature distribution and temperature distribution animation are then opened, as well as a meme. Note that for the meme to open, the file `Dont-Open-Surprise-Inside.jpg` must exist in the console directory.

5 Discretisation

From Equation 4, letting $T_{xx} = \frac{\partial^2 T}{\partial x^2}$, $T_{yy} = \frac{\partial^2 T}{\partial y^2}$, and $T_t = \frac{\partial T}{\partial t}$, equation becomes:

$$T_t - \alpha \left(T_{xx} + T_{yy}\right) = 0 \tag{8}$$

The problem can be discretised into a 3D grid, with a 2D spatial domain and a 1D time domain. Let the dimensions (x, y, t) correspond with indices (i, j, t) respectively, where subscripts i, j correspond with spatial steps in x, y and superscript t corresponds with time steps in t. The central difference approximations for second order derivatives are applied for T_{xx} and T_{yy}, whereas backward difference approximations for first order derivatives is used for T_t:

$$\begin{cases} T_{xx} \approx \frac{1}{(\Delta x)^2}\left[T^t_{i+1,j} - 2T^t_{i,j} + T^t_{i-1,j}\right] \\ T_{yy} \approx \frac{1}{(\Delta y)^2}\left[T^t_{i,j+1} - 2T^t_{i,j} + T^t_{i,j-1}\right] \\ T_t \approx \frac{1}{\Delta t}\left[T^t_{i,j} - T^{t-1}_{i,j}\right] \end{cases}$$

We can now rewrite the governing equation as:

$$\frac{T^t_{i,j} - T^{t-1}_{i,j}}{\Delta t} = \alpha \left(\frac{T^{t-1}_{i+1,j} - 2T^{t-1}_{i,j} + T^{t-1}_{i-1,j}}{(\Delta x)^2} + \frac{T^{t-1}_{i,j+1} - 2T^{t-1}_{i,j} + T^{t-1}_{i,j-1}}{(\Delta y)^2} \right) \tag{9}$$

In this case, $\Delta x = \Delta y = h$. In other words, the mesh spacing is the same in the x and y direction. The time step, Δt, is referred to as k. Making next-step-in-time $T_{i,j}^{t}$ the subject, the discretised equation can be written as:

$$T_{i,j}^{t} = T_{i,j}^{t-1} - \frac{\alpha k}{h^2}\left(T_{i+1,j}^{t-1} + T_{i-1,j}^{t-1} + T_{i,j+1}^{t-1} + T_{i,j-1}^{t-1} - 4T_{i,j}^{t-1}\right) \tag{10}$$

The problem is fully discretised, with 1 point in the current time step being a function of 5 nodes in the previous time step. There are contributions from neighboring nodes and the node's value from the previous time step. Rewriting by isolating the $T_{i,j}^{t-1}$ term:

$$T_{i,j}^{t} = \frac{\alpha k}{h^2}\left(T_{i+1,j}^{t-1} + T_{i-1,j}^{t-1} + T_{i,j+1}^{t-1} + T_{i,j-1}^{t-1}\right) + \left(1 - \frac{4\alpha k}{h^2}\right)T_{i,j}^{t-1} \tag{11}$$

However, the time-variability in the parabolic PDE introduces stability issues. The Courant condition of stability states that the contribution of the value of the previous node must be positive for as stable and converging solution - this is expressed in equation (12):

$$\left(1 - \frac{4\alpha k}{h^2}\right) > 0 \tag{12}$$

We can then define a relationship between time step and mesh size for convergence:

$$\frac{\alpha k}{h^2} < \frac{1}{4} \tag{13}$$

As there are two sets of properties in the problem we have defined - one for the bread and one for the chicken - there are two Courant criteria that must be fulfilled.

5.1 Discretisation of boundary conditions

Dirichlet boundary conditions do not require discretisation, but must be applied at nodes where $i = 0$ and $i = i_{max}$.

The convective boundary conditions are applied at the sides of the panini by implementing the backward finite difference formula for first order derivatives. For the left side:

$$T_{i,0}^{t} = \frac{h_{conv}T_{\infty} + T_{i,1}^{t-1}k_{cond}}{k_{cond} + h_{conv}h} \tag{14}$$

For the right side:

$$T_{i,j_{max}}^{t} = \frac{h_{conv}T_{\infty} + T_{i,j_{max}-1}^{t-1}k_{cond}}{k_{cond} + h_{conv}h} \tag{15}$$

Depending on the value of i, the value for k_{cond} must vary to account for the appropriate panini constituent.

For a node in the bread region at the boundary between bread and chicken, a weighted average thermal diffusivity is used. For a node in bread, it has three adjacent bread nodes and one adjacent chicken node hence diffusivity is weighted towards the diffusivity of bread. The matter of convergence is not a concern with this formula as the diffusivity must lie between α_{bread} and $\alpha_{chicken}$ and so if the numerical method is already convergent, the chicken-bread condition satisfies the Courant criteria.

For bread nodes at the chicken-bread boundary:

$$T_{i,j}^{t} = \frac{k\alpha_{bb}}{h^2}\left(T_{i+1,j}^{t-1} + T_{i-1,j}^{t-1} + T_{i,j+1}^{t-1} + T_{i,j-1}^{t-1} - 4T_{i,j}^{t-1}\right) \tag{16}$$

For chicken nodes at the chicken-bread boundary:

$$T_{i,j}^{t} = \frac{k\alpha_{cb}}{h^2}\left(T_{i+1,j}^{t-1} + T_{i-1,j}^{t-1} + T_{i,j+1}^{t-1} + T_{i,j-1}^{t-1} - 4T_{i,j}^{t-1}\right) \tag{17}$$

6 Results and Discussion

Sample results can be found at `https://imperiallondon-my.sharepoint.com/:f:/g/personal/sah419_ic_ac_uk/Eoffz0cQgrNHqhSbmo_zBJMBS9dzqtlhy6hUIka7bR5wMg?e=mvkZef`

Figures 3 and 4 illustrate the temperature distribution in identical paninis after 30 seconds. Despite the code taking significantly longer to run for a x10 finer mesh (< 1 second vs one hour), the distributions are very similar. Differences can be seen in the edges, where the finer mesh has produced marginally different contour shapes.

For the 1 mm mesh, the minimum temperature was found to be 20.45°C and the maximum was found to be 116.19°C. In contrast for the 0.1 mm mesh, the minimum temperature was found to be 16.14°C and maximum 145.85°C. Clearly the discretisation of the panini causes loss of information.

While the static images do not vary greatly with mesh size, more obvious differences can be observed when comparing animations. The larger mesh size of 1 millimetre took approximately 6 seconds to save the animation to an mp4 file at 150 dpi. In contrast, the code had to be left running overnight (several hours) for an animation to be produced for the 0.1 millimetre mesh size at 50 dpi. The contour plots vary in 'smoothness' as expected by the mesh sizes. It is in the animation where a finer mesh pays dividends.

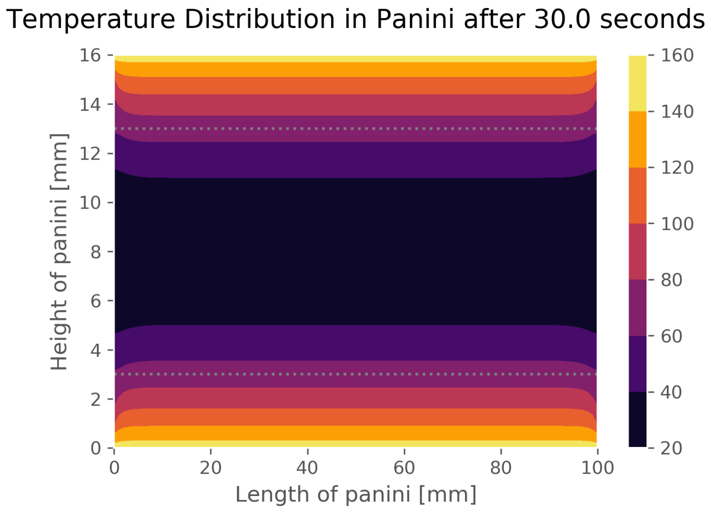

Figure 3: The temperature distribution in the panini after 30 seconds, calculated at $h = 1$ mm and $k = 0.5$ s. Dotted lines indicate the bread-chicken boundary

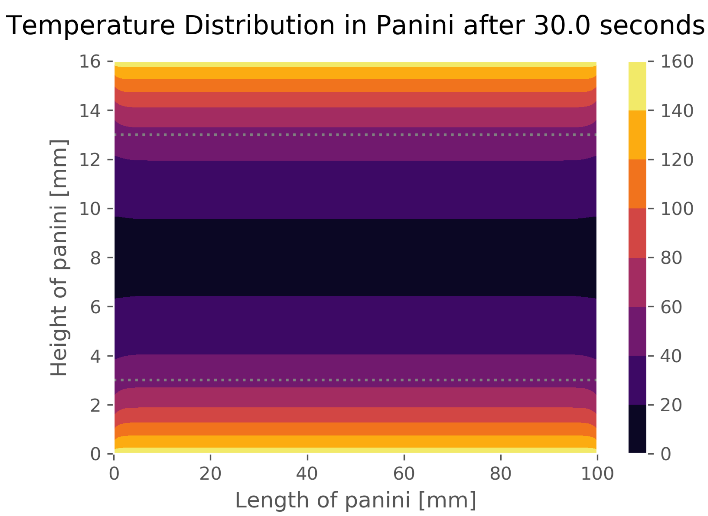

Figure 4: The temperature distribution in the panini after 30 seconds, calculated at $h = 0.1$ mm and $k = 0.005373374$ s. Dotted lines indicate the bread-chicken boundary

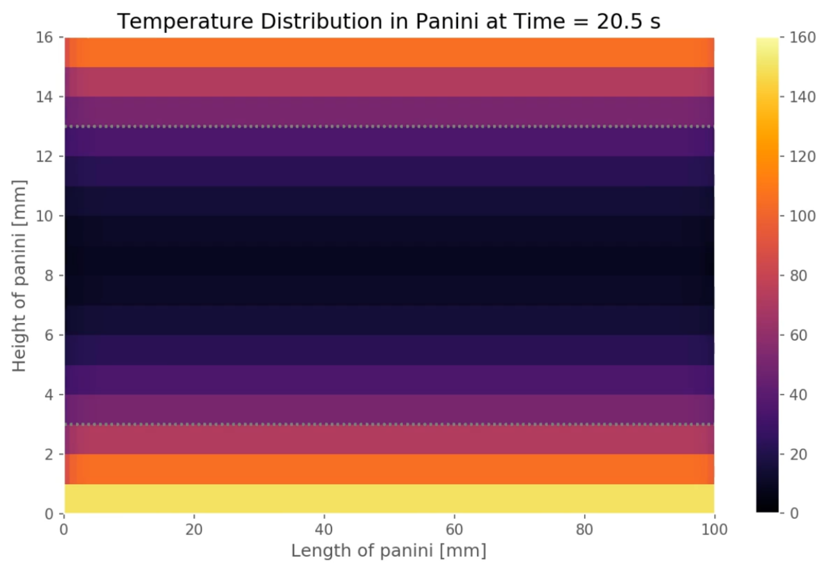

Figure 5: [Animation capture] Temperature distribution in panini after 20.5 seconds at 1 mm mesh size

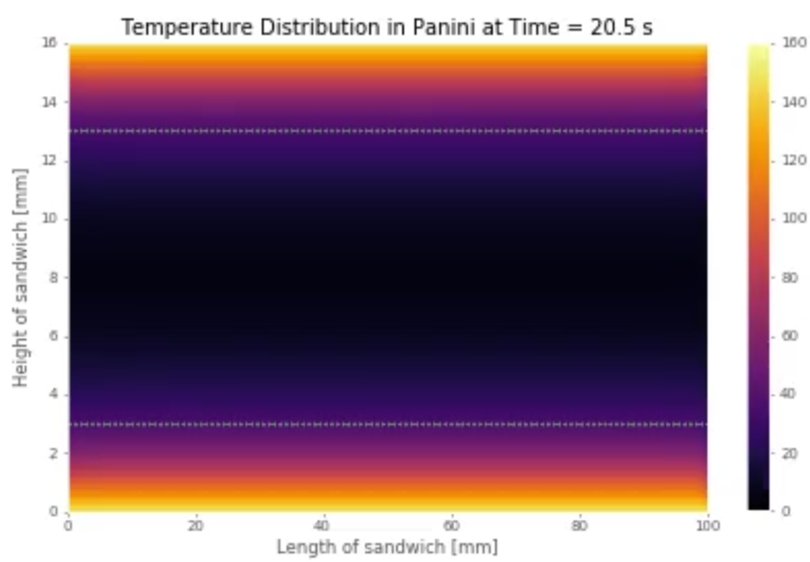

Figure 6: [Animation capture] Temperature distribution in panini after 20.5 seconds at 0.1 mm mesh size. Poor image quality is due to a lower dpi.

7 Additional Remarks

- Many assumptions listed under *Section 1: Physics Model* are not applicable in real life scenarios. For example, homogenous bread solids do not exist in real-life, where actual solid contain defects with inhomogenous properties over space.
- The finite difference method (an explicit method) is not often used in comparison to the Crank-Nicolson method (an implicit method) due to computational efficiency and stability reasons. Comparison shown as follows:
  - *Explicit methods*: Stability problems which limit maximum time-step taken relative to space-steps; Computationally inefficient, where halving space mesh size increases number of operations by 4 times.
  - *Implicit Methods*: Stability problems do not exist, and is computationally efficient. However, implementation of implicit methods are more complex.
- Alternatively, implicit methods such as the Crank-Nicolson method can be applied for parabolic PDE's. The choice for time/space-steps are no longer limited by the Courant condition, and the solution is also 2nd order accurate in both space and time. Furthermore, the Crank-Nicolson method can also be applied for non-uniform meshes, allowing for a larger range of applications. Despite the increased complexity of implementing implicit methods, they overcome the shortcomings of the explicit methods (e.g. inaccuracy, computational inefficiency).
- The code asks for a user input for mesh size. The code returns dimension errors if the mesh size is not a factor of both the side length and height of the sandwich. A better implementation may be a check for this where the user input is rounded so that this error does not occur. Similarly, mesh sizes larger than the thickness of the bread result in errors due to the indexing of boundary conditions.
- Floating point errors were encountered initially during the process, where small yet slightly-inaccurate decimals were summed and multiplied by large numbers (~ 1000) for unit conversion. This was resolved by converting units early in the process to avoid such errors, but more extensive code testing is required to eliminate the issue altogether.

8 Works Cited

Hamilton Beach. "Hamilton Beach Panini Press Gourmet Sandwich Maker - 25460A." Accessed April 4, 2021. https://hamiltonbeach.com/panini-press-gourmet-sandwich-maker-25460a.

Porter, Marie. "Homemade Pesto Chicken Panini." *Celebration Generation* [blog], September 19, 2020. https://celebrationgeneration.com/homemade-pesto-chicken-panini/.