Programming Assignment to solve 1-D Transient Heat Transfer Problem using Finite Difference Numerical Method

Submitted for Core Course MM204

For the requirements of the MEMS B.Tech. Program

By

Raghav Jain (19D110013)
Sanskar Jaiswal (19D110017)
Shibashis Sahoo (19D110019)

Guide:
Professor Vishwanathan Nurni

Department of Metallurgical Engineering and Materials Science

INDIAN INSTITUTE OF TECHNOLOGY BOMBAY

(May 2021)

Declaration

We hereby accept that this report is our written submission, and it represents our ideas in our own words. Wherever others' ideas and phrases have been included, we have cited or given their reference with the sources. We also declare that we have followed all the academic honesty principles and integrity. We realize that any violation of the above will be a cause for disciplinary action by the Institute.

Raghav Jain
Sanskar Jaiswal
Shibashis Sahoo

Acknowledgments

We would like to express sincere regards and deep gratitude to our guide Prof. Vishwanathan Nurni, Department of Metallurgical and Materials Science, Indian Institute of Technology Bombay, for his valuable guidance and encouragement. Without his faith in us, We would not have been able to carry out any such project on our own.

We would like to thank our batchmates for clearing any of our doubts to the best of their knowledge, and for keeping us updated with the course contents and timelines.

And last but not least, we would like to express our utmost gratitude and love to our parents and thank them for creating a kind of environment which allows us to keep focusing on our studies, without facing any problems.

Abstract

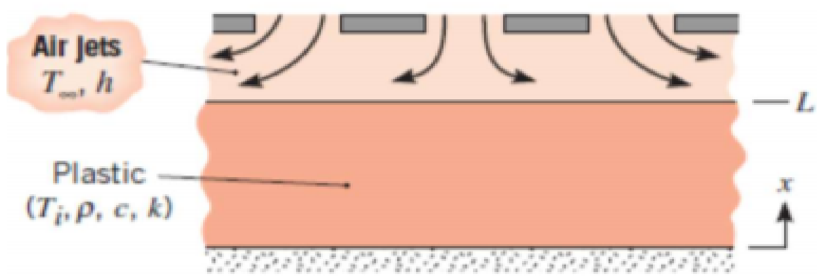
One dimensional transient heat transfer system is the simplest form used to study complex models occurring in real life. The dynamics of Temperature of the system is the most important parameter to understand the working of real world models and come up with better solutions which are more cost effective. In this programming assignment we have tried to solve 1-D transient heat problem involving convection and conduction mode of transfer at the same time using finite difference numerical method. Starting with the balance of heat for a finite size control volume for a short period of time and extending the equations to get the complete Temperature profile of the system with minimal error. Analytically solving second order differential equations is quite intimidating and computationally expensive to solve using a computer. We have considered the explicit form of recurrence relation for the Temperature of the control volume along with satisfying conditional stability to find the time step for the numerical method. Finally we show the temperature profile of Bottom and Top Surface of the slab with respect to time and the change in temperature gradient along the length with time.

Contents

Chapter 1:
Introduction and Problem Statement

According to the problem statement a molded plastic product is cooled by exposing one surface to an array of air jets, while the opposite surface is well insulated.

A molded plastic product ($\rho = 1200\ \text{kg/m}^3$, $c = 1500\ \text{J/kg}\cdot\text{K}$, $k = 0.30\ \text{W/m}\cdot\text{K}$) is cooled by exposing one surface to an array of air jets, while the opposite surface is well insulated. The product may be approximated as a slab of thickness $L = 60$ mm, which is initially at a uniform temperature of $T_i = 80°\text{C}$. The air jets are at a temperature of $T_\infty = 20°\text{C}$ and provide a uniform convection coefficient of $h = 100\ \text{W/m}^2\cdot\text{K}$ at the cooled surface.



Using a finite-difference solution with a space increment of $\Delta x = 6$ mm, determine temperatures at the cooled and insulated surfaces after 1 h of exposure to the gas jets.

Here, we use the finite difference solution and divide the molded plastic into 11 parts of which the top-most and bottom-most control volumes are 3cm thick and the in-between 9 parts are each 6 cm thick.

The following are the assumptions we make -

1. We have assumed the plastic mold to be infinite (i.e. Large enough) in the Y (horizontal) and Z (out of plane of screen) directions and finite in X (vertical) direction.Hence, we assume heat only flows along the thickness

direction (vertical direction) of the mold and it is uniform in the other two directions,thus making it a 1-Dimensional Heat transfer problem

2. Heat transfer Coefficient for air jets on the top surface of the slab is constant.

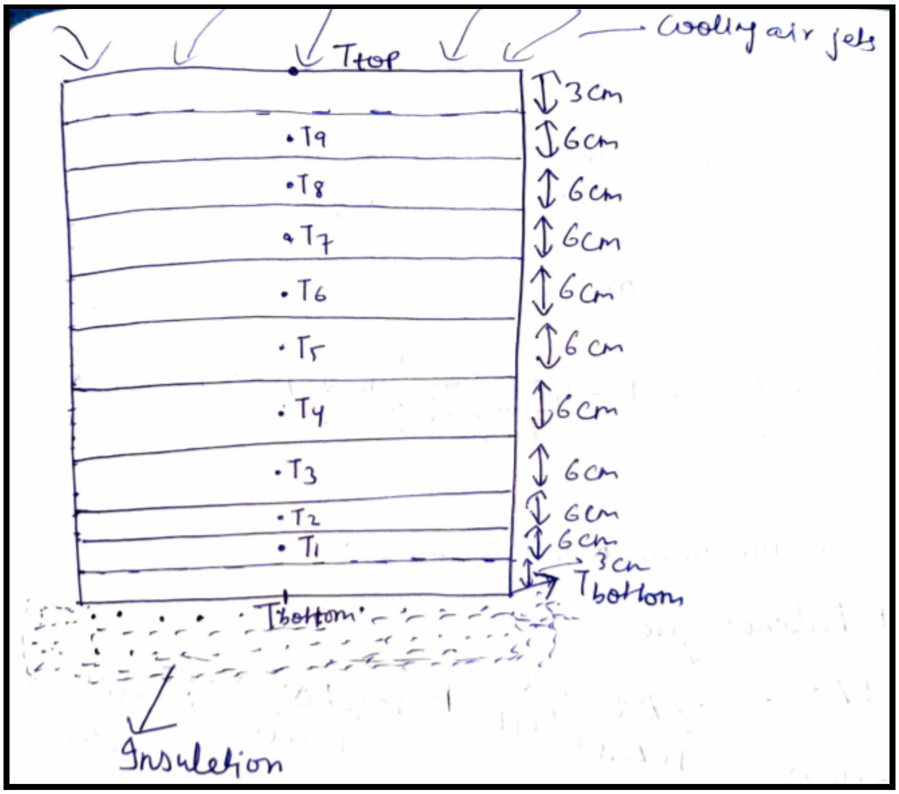
3. The top-most layer has 2 modes of heat transfer(convection with the surrounding at the top and conduction with subsequent layers). Negligible Radiation on the top surface. The middle layer transfers heat through conduction with the layer above and below it.

Considering all these nuances, we shall now proceed to solve the equation, first on paper to simplify the equations as much as possible and then on python to make computations based on these simplified equations.

Our aim will be to solve the equations both implicitly and explicitly. In the chapters 2 and 3 respectively.

Chapter 2:
Problem Solving Approach

Let us write down the equations for the top surface, all middle surfaces and the bottom-most surface individually -



Heat Transfer Equation for the top Surface of the Slab -
Here mode of heat transfer is conduction from the inside the slab and convection from the top surface to the surrounding air.

$$\rho_{slab} C A \frac{\Delta x}{2}\left(\frac{T_{top}^{t+\Delta t} - T_{top}^{t}}{\Delta t}\right) = \frac{-k A\left(T_{top}^{t} - T_{9}^{t}\right)}{\Delta x} - hA\left(T_{top} - T_{\infty}\right)$$

ρ**-** density of the slab material
K - thermal conductivity of the slab
Δt**-** time step
C - specific heat capacity of the slab
Δx**-** thickness of the control volume for operating the numerical method to solve the differential equations.

h - heat transfer coefficient
A - area of the slab
α**=** ρ**/KC -** thermal diffusivity

Notation -
T^t - temperature of a control volume at a time step t
$T^{t+\Delta t}$ - temperature of a control volume at a time step t+Δt
T_{top} - Temperature of the top-most layer
T_{bottom} - Temperature of the bottom-most layer
T_i - Temperature of the ith layer

Hence we can write the above equation as -

$$T_{top}^{t+\Delta t} = \left(\frac{-2k\Delta t}{\rho c\,\Delta x^2} + 1 - \frac{2h\,\Delta t}{\rho c\,\Delta x}\right) T_{top}^{t} + \left(\frac{2h\Delta t}{\rho c\Delta x}\right) T_{\infty} + \frac{2k\Delta t}{\rho c\,\Delta x^2}\, T_9^{t}$$

Heat Transfer Equation for in-between control volumes -

Let us write the heat transfer equation for any general layer say the ith layer (for i being any integer from 1 to 9), then the equations of these layers in between the top-most and bottom-most layer simply becomes:-

Heat Balance gives:-

$$T_i^{t+\Delta t} = \frac{k\Delta t}{\rho c\,\Delta x^2} T_{i+1}^{t} + \left(1 - \frac{2k\Delta t}{\rho c\,\Delta x^2}\right) T_i^{t} + \frac{k\Delta t}{\rho c\,\Delta x^2} T_{i-1}$$

for (i = 1 to 9)

(Note- Here, **T_{i+1} = T_{top}** when i=9, for bottom layer insulation is there.)

Also, the equations must be consistent with the heat balance for the top layer.
Heat Transfer Equation for bottom-most control volumes -

Heet balence:-

$$\rho c A \frac{\Delta x}{2}\, \frac{T_{bottom}^{t+\Delta t} - T_{bottom}^{t}}{\Delta t} = \frac{kA\left(T_1^{t} - T_{bottom}^{t}\right)}{\Delta x}$$

Hence, we get the recurrence relation -

$$\boxed{T_{bottom}^{t+\Delta t} = \left(1 - \frac{2k\Delta t}{\rho c \Delta x^2}\right) T_{bottom}^{t} + \frac{2k\Delta t}{\rho c \Delta x^2} T_1^{t}}$$

Initial temperature of slab according to the given problem= $T_{initial}$ (at t=0)= 80^0C =353K
Temperature of surroundings = T_∞ = 20^0 C = 293K

For a consistent solution in the explicit method, the coefficients of the temperatures of the control volumes at any time 't' must be positive for conditional stability to be satisfied.

Equations for conditional stability (i.e. the coefficients shouldn't be negative) -

$$1 - \frac{2\Delta t k}{\rho c \Delta x^2} > 0 \quad \text{And} \quad 1 - \frac{2\Delta t k}{\rho c \Delta x^2} - \frac{2h\Delta t}{\rho c \Delta x} > 0$$

$$\Downarrow \qquad\qquad\qquad\qquad \Downarrow$$

$$\boxed{\Delta t < \frac{\rho c \Delta x^2}{2k}} \rightarrow (1) \qquad \boxed{\Delta t < \frac{\rho c \Delta x^2}{2(k + \Delta x h)}} \rightarrow (2)$$

Since condition 2 gives a lower value of upper bound, it must be followed and condition 1 will follow as a result.

Plugging in the values of all the constants we get the upper bound on Δt, which lets us calculate the number of iterations required to get the approximate temperature -

$$\Delta t < \frac{1200 \times 1500 \times (6\times10^{-3})^2}{2 \times \left(0.3 + (6\times10^{-3})100\right)}$$

$$\Delta t < 36 \text{ seconds}.$$

Hence, any value less than 36 seconds works for us to get the correct solutions.

Chapter 3:
Solving Implicitly

Now we write the equations to solve the code implicitly which is bound to complicate things a bit in the computational part.

<u>Heat Transfer Equation for the top Surface of the Slab -</u>

$$\frac{\rho c \Delta x}{2}\left(\frac{T_{top}^{t+\Delta t} - T_{top}^{t}}{\Delta t}\right) = -k\left(\frac{T_{top}^{t+\Delta t} - T_{9}^{t+\Delta t}}{\Delta t}\right) - h\left(T_{top}^{t+\Delta t} - T_{\infty}\right)$$

$$\boxed{\left(1 + \frac{2k\Delta t}{\rho c(\Delta x)^2} + \frac{2h\Delta t}{\rho c \Delta x}\right)T_{top}^{t+\Delta t} - \left(\frac{2k\Delta t}{\rho c(\Delta x)^2}\right)T_{9}^{t+\Delta t} = \left(\frac{2h\Delta t}{\rho c \Delta x}\right)T_{\infty} + T_{top}^{t}}$$

<u>Heat Transfer Equation for in-between control volumes -</u>

$$\rho c \Delta x\left(T_{i}^{t+\Delta t} - T_{i}^{t}\right) = \frac{-k}{\Delta x}\left(T_{i}^{t+\Delta t} - T_{i-1}^{t+\Delta t}\right) + k\left(\frac{T_{i+1}^{t+\Delta t} - T_{i}^{t+\Delta t}}{\Delta x}\right)$$

$$(\text{for } i = 1 \text{ to } 9)$$

$$\boxed{\begin{gathered}(\text{for } i = 1 \text{ to } 9)\\ T_{i}^{t+\Delta t}\left(1 + \frac{2k\Delta t}{\rho c(\Delta x)^2}\right) - \left(\frac{k\Delta t}{\rho c(\Delta x)^2}\right)T_{i+1}^{t+\Delta t} - \left(\frac{k\Delta t}{\rho c(\Delta x)^2}\right)T_{i-1}^{t+\Delta t} = T_{i}^{t}\end{gathered}}$$

<u>Heat Transfer Equation for bottom-most control volumes -</u>

$$\frac{T_{bottom}^{t+\Delta t} - T_{bottom}^{t}}{\Delta t}\left(\rho c \frac{\Delta x}{2}\right) = \frac{k}{\Delta x}\left(T_{1}^{t+\Delta t} - T_{bottom}^{t+\Delta t}\right)$$

$$T_{bottom}^{t+\Delta t}\left(1+\frac{2k\Delta t}{\rho c(\Delta x)^2}\right)-\left(\frac{2k\Delta t}{\rho c(\Delta x)^2}\right)T_1^{t+\Delta t} = T_{bottom}^{t}$$

Implicit equation is unconditionally stable, hence we don't require to satisfy any constraint and take any value for the time step Δt. The solution will still be consistent.

To solve the above linear system of equations we can write the equation in matrix form -
Ax = B
We solve for x = $A^{-1}B$

$$\frac{2k\Delta t}{\rho c\Delta x^2} = a \quad \& \quad \frac{2h\Delta t}{\rho c\Delta x} = b$$

$$A = \begin{bmatrix}
1+a & -a & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-a & 1+a & a & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -a & 1+a & a & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -a & 1+a & a & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -a & 1+a & a & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -a & 1+a & a & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -a & 1+a & a & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -a & 1+a & a & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -a & 1+a & a & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -a & 1+a & a \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -a & 1+a+b
\end{bmatrix}_{(11\times 11)}$$

$$B = \begin{bmatrix} T^t_{bottom} \\ T^t_1 \\ T^t_2 \\ T^t_3 \\ T^t_4 \\ T^t_5 \\ T^t_6 \\ T^t_7 \\ T^t_8 \\ T^t_9 \\ bT^t_{\infty} + T^t_{top} \end{bmatrix}_{(11\times 1)}$$

Where , A is a 11x11 matrix
B is 11x1 vector
Hence, x is 11x1 vector

Note: to solve the matrix equation we have used numpy library in Python which has a function to directly solve the matrix equation.

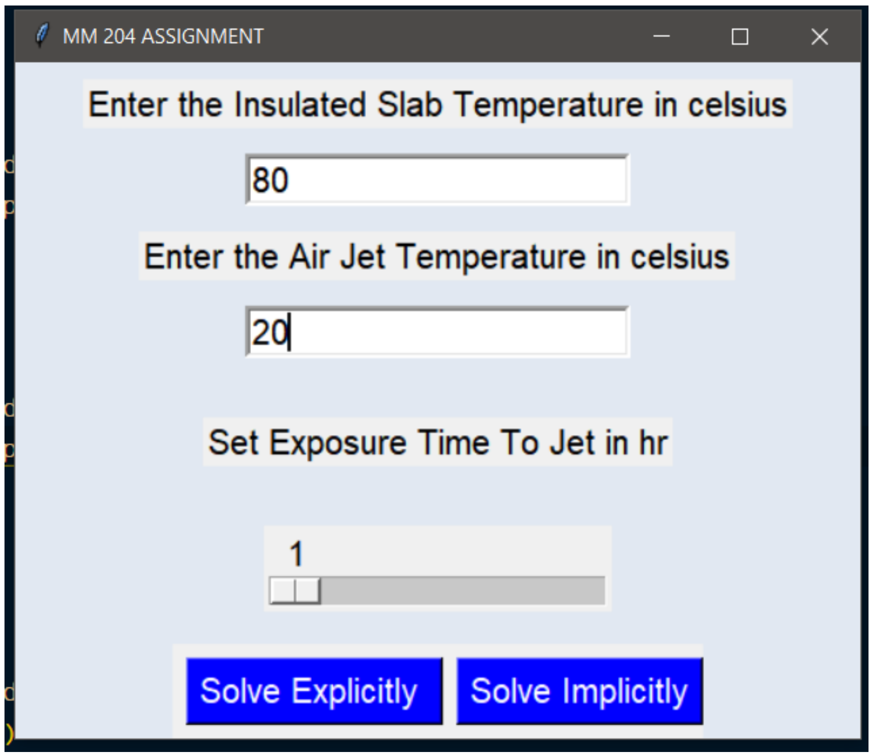
Chapter 4:
Python Code

- Here We have made a small GUI(Graphical User Interface) using Python.

(In the code folder, there is a file named "GUI.py" when the user runs that program, a small window will pop out).

- It takes inputs from the user for Insulated Surface Temp, Cooled Surface Temp in Celsius, and Time in hr.

- After that the user can choose either explicit or implicit methods for solving it.



(In this example we are giving the default values provided in the question)

- The Result window shows the solution of both explicit and implicit methods. In addition it also shows two graphs :-
  - Variation of both surface temp with time
  - Variation of temp in the layers of the slab with distance

```python
import numpy as np
import matplotlib.pyplot as plt

h = 100  # Heat thermal Coefficient of air
k = 0.3  # Thermal conductivity
cp = 1500  # specific heat capacity of the material
rho = 1200  # density of the slab's material
dx = 6e-3  # 6 mm - length of the control volume considered, the boundary control volumes are 3 mm, half the size of inner control volume
dt = 30  # in second as delta t must be less than 36 second comes from explicit equation conditional stability rule
a = (2*k*dt)/((dx**2)*rho*cp)
b = (2*h*dt)/(dx*rho*cp)
T_inf = 293
T_all = np.zeros(11)
T_all += 353
T_bottom = [353]
T_top = [353]
Time = np.linspace(0, 3600, 121)
dist = np.linspace(0, 60, 11)

for i in range(int(3600/dt)):
    T_all = [(1-a-b)*T_all[10]+b*T_inf+a*T_all[8] if j == 10 else (1-a)*T_all[0]+a*T_all[1]
             if j == 0 else (a/2)*T_all[j+1]+(1-a)*T_all[j]+(a/2)*T_all[j-1] for j in range(11)]
    T_bottom.append(T_all[0])
    T_top.append(T_all[10])
    if i % 8 == 0:
        plt.plot(dist, T_all)
        plt.xlabel("Distance from bottom(in cm)")
        plt.ylabel("Temperature (in K)")
        plt.title("Variation of temperature profile of slab with time")


print(f"Insulated Surface Temperature :{T_all[0]} \n",
      f"Cooled Surface Temperature :{T_all[10]}")

plt.show()
plt.plot(Time, T_bottom)
plt.xlabel("Time(in seconds)")
plt.ylabel("Temperature of Bottom(in K)")
plt.title("Variation of Bottom surface Temperature with time")
plt.show()
plt.plot(Time, T_top)
plt.xlabel("Time(in seconds)")
plt.ylabel("Temperature of Top(in K)")
plt.title("Variation of Top surface Temperature with time")
plt.show()
```

Python Code For Explicit Solution

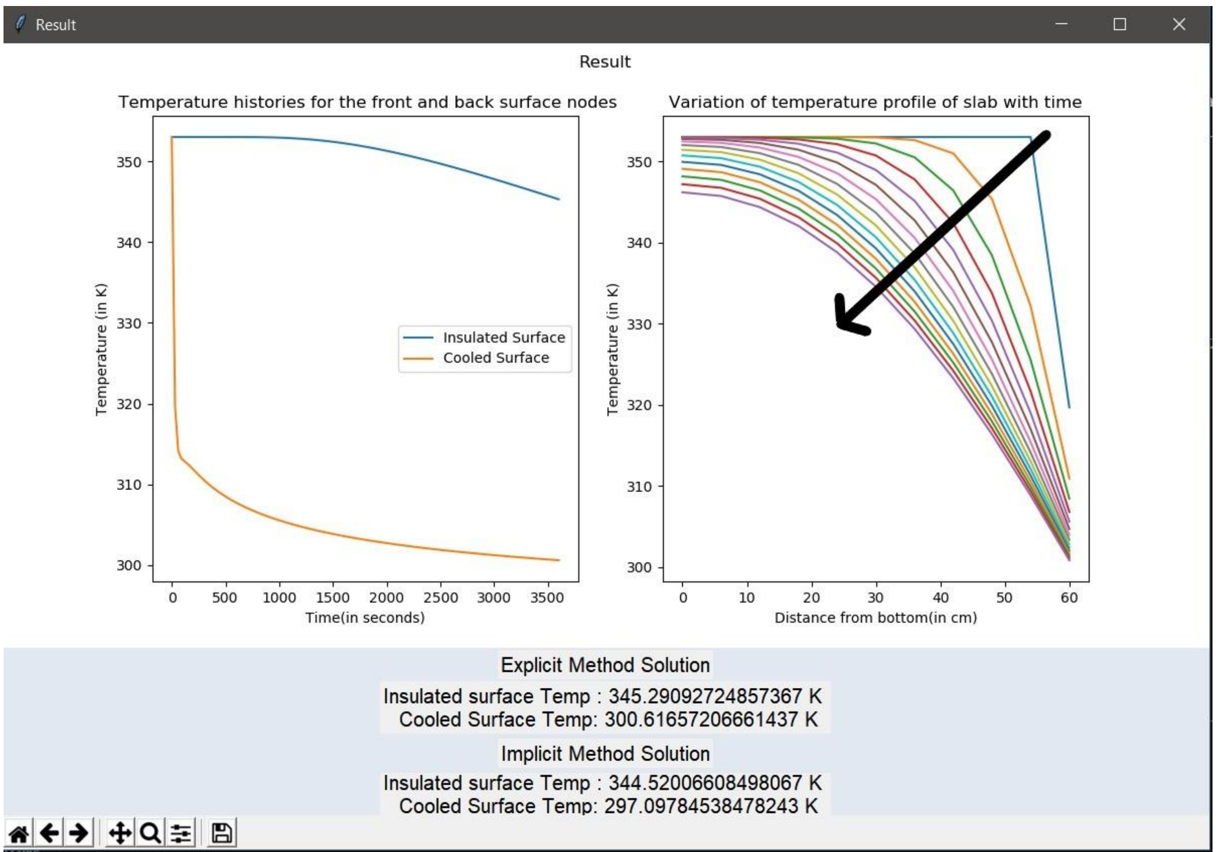
```python
import numpy as np
import matplotlib.pyplot as plt
h = 100  # Heat thermal Coefficient of air
k = 0.3  # Thermal conductivity
cp = 1500  # specific heat capacity of the material
rho = 1200  # density of the slab's material
dx = 6e-3  # 6 mm - length of the control volume considered, the boundary control volumes are 3 mm, half the size of inner control volume
dt = 30  # in second as delta t must be less than 36 second comes from explicit equation conditional stability rule
a = (2*k*dt)/((dx**2)*rho*cp)
b = (2*h*dt)/(dx*rho*cp)
T_inf = 293
T_all = np.zeros(11)
T_all += 353
for i in range(int(3600/dt)):
    A = np.zeros((11, 11))
    B = np.zeros(11)
    for i in range(11):
        if i == 0:
            A[i, i] = 1+a
            A[i, i+1] = -a
            B[i] = T_all[i]
        elif i == 10:
            A[i, i-1] = -a
            A[i, i] = 1+a+b
            B[i] = T_all[i]+b*T_inf
        else:
            A[i, i-1] = -(a/2)
            A[i, i] = 1+a
            A[i, i+1] = -(a/2)
            B[i] = T_all[i]

    T_all = np.linalg.solve(A, B)
print(f"Insulated Surface Temperature :{T_all[0]} \n",
      f"Cooled Surface Temperature :{T_all[10]}")
```

Python Code For Implicit Solution

Chapter 5:
Results and Discussions

Although the top surface temperature rapidly approaches that of the air jet, there is a significant lag in the thermal response of the back surface, which could be easily seen from the plots of temperature profile of both the surfaces. The steep fall in the graph for cooled surface is easily interpretable. The different thermal responses are due to the small value of thermal diffusivity and large value of Biot number. To improve the result of the numerical method we have used, a smaller value of Δxshould be used.



Black arrow shows direction of increasing time, and the graph demonstrates change in temperature as a distance from bottom with time.

Hence, we can find out the temperature profile of the plastic mold with time with the help of the Python code and application developed by us.

THE END

THANKS A LOT FOR YOUR TIME