MM204

Programming Assignment

Abhishek D. Sontakke, Ishaan A. Shende and Jaideep S. Chawla

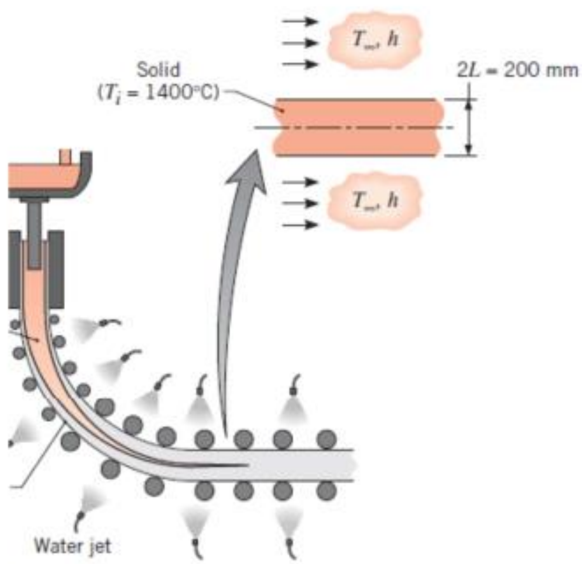
Roll Numbers: 190110001, 190110029 and 190110030

Group Number: **G04**

Abstract

In this report, we present the results for the following problem:

In a thin-slab, continuous casting process, molten steel leaves a mold with a thin solid shell, and the molten material solidifies as the slab is quenched by water jets en route to a section of rollers. Once fully solidified, the slab continues to cool as it is brought to an acceptable handling temperature. It is this portion of the process that is of interest.



Consider a 200-mm-thick solid slab of steel ($\rho = 7800\ \mathrm{kg/m^3}$, $c = 700\ \mathrm{J/kg \cdot K}$, $k = 30\ \mathrm{W/m \cdot K}$), initially at a uniform temperature of $T_i = 1400°C$. The slab is cooled at its top and bottom surfaces by water jets ($T_\infty = 50°C$), which maintain an approximately uniform convection coefficient of $h = 5000\ \mathrm{W/m^2 \cdot K}$ at both surfaces. Using a finite-difference solution with a space increment of $\Delta x = 1$ mm, determine the time required to cool the surface of the slab to 200°C. What is the corresponding temperature at the midplane of the slab? If the slab moves at a speed of $V = 15$ mm/s, what is the required length of the cooling section?

By using a time increment of 50ms,

1. Time required to cool the surface to 200°C is **162.5 s**
2. Corresponding temperature at the mid-plane of the slab is **1365°C**
3. Required length of the cooling section is **2.44 m**

Solution

To find the time required to cool the outer surface to 200° C, we will use the explicit finite-difference method. Given space increment $\Delta x = 1mm$ and since there is symmetry about midplane, there are 101 unknown nodal temperatures. Obviously, implementing an implicit scheme will be computationally heavy, so, we will stick to a forward-difference scheme for this problem.

Finite difference equations [1]

For a specified Δx and Δt, the finite-difference form of the Fourier number and the Biot number are as follows:

$$Fo = \frac{\alpha \Delta t}{(\Delta x)^2}, \qquad Bi = \frac{h\Delta x}{k}$$

Also, we need to take care of the stability criterion,

$$Fo(1 + Bi) \leq \frac{1}{2}$$

[Note that there is no such criterion for the implicit scheme]

1. For a one-dimensional system in x, the explicit scheme for interior nodes (0-99) is:

$$T_m^{p+1} = Fo\left(T_{m+1}^p + T_{m-1}^p\right) + (1 - Fo)T_m^p \ldots (1)$$

where, T_m^{p+1} is the nodal temperature of the mth interior node at (p+1) time

2. For the midplane node (m = 0), we further use the symmetry requirement $T_{m+1}^p = T_{m-1}^p$:

$$T_0^{p+1} = 2FoT_m^{p+1} + (1 - Fo)T_m^p \ldots (2)$$

3. Finally for the surface node (m = 101), energy balance yields:

$$T_{101}^{p+1} = 2Fo\left(T_{99}^p + Bi\, T_\infty\right) + (1 - 2Fo - 2Bi\, Fo)T_{99}^p \ldots (3)$$

Some Explanation

Since, the cast is initially at a uniform temperature, the temperature of each node is known at $t = 0$ $(p = 0)$ i.e., 1400°C. The calculations begin at $t = \Delta t$ $(p = 1)$, where equations (1)-(3) are applied to the applied to the appropriate node to determine its temperature. Once temperatures are known for $t = \Delta t,$ they are used to determined temperature at $t = 2\Delta t$ $(p = 2)$ using the same equations. In this manner, temperatures are evolved over intervals of Δt until temperature of surface node (T_{101}) reaches 200°C.

The value of p corresponding to $T_{101} = 200°C$ is multiplied by the time interval (Δt) to get the time required (t). We already would have calculated the T_0 (mid-plane temperature) at this step 'p'. The length of cooling section is obtained by multiplying 't' to the casting speed (15mm/s).

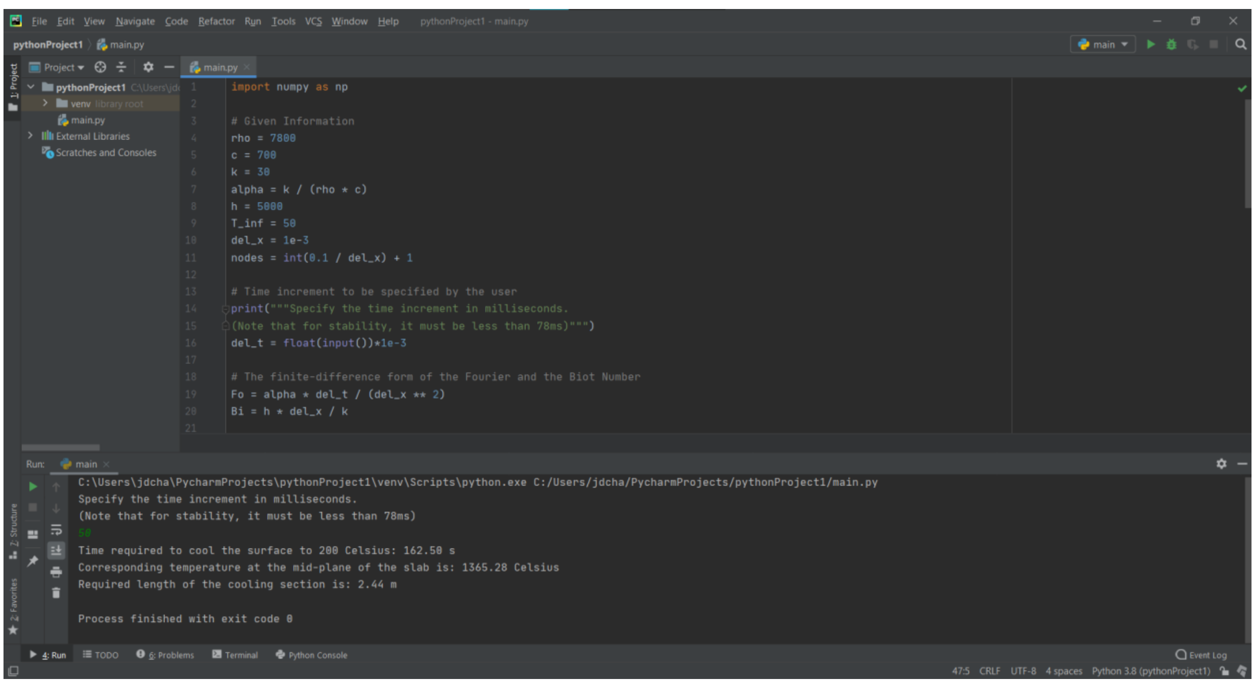
Comments Regarding the Program

1. The program is designed to work out the exact scheme as stated earlier.
2. The user is asked to specify the time increment (in milliseconds) to be used in calculations. (Additional care must be taken that is should be less than 78 ms otherwise the stability criterion will fail)
3. The program then calculates the unknown nodal temperatures iteratively until the surface node temperature reaches 200°C and returns the time taken, corresponding temperature of the midplane and the length of the cooling section.

How to run the code

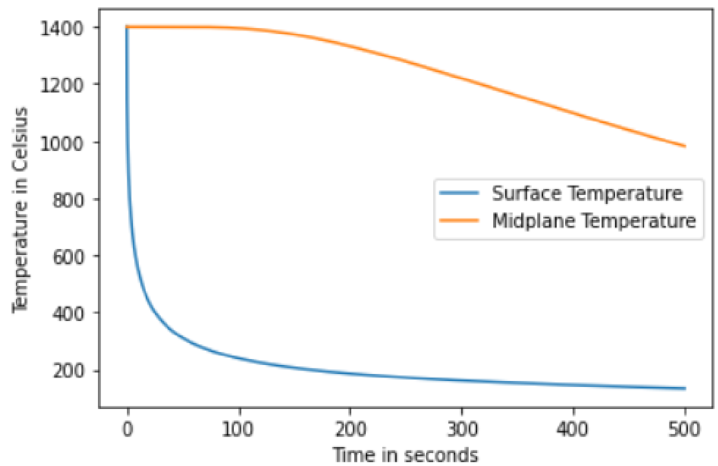
The code is presented as a .py file and can be run on any Python IDE. (the appropriate libraries must be installed).

The following snippet shows one such run on PyCharm Community Edition 2020.2.3



Discussion

1. Overall Fourier number $Fo = \alpha t/L^2 = 0.089$ ($2L = 200mm$). Since it is less than 0.2, the approximate one-term solution for 1-D plane wall cannot be used for verification.
2. We can also plot the temperature distribution for the surface and midplane nodes. The code snippet that accomplishes this is also attached:




```
# Plotting
import matplotlib.pyplot as plt

T_prev = 1400*np.ones(nodes)
T_next = 1400*np.ones(nodes)
Midplane_temp, Surface_temp, timesteps = [1400],[1400],[0]

step = 0
while(step <= 500/del_t):
  step+=1
  timesteps.append(step*del_t)
  T_next[0] = T_prev[1]
  Midplane_temp.append(T_next[0])
  for i in range(1,nodes-1):
    T_next[i] = Fo*(T_prev[i-1]+T_prev[i+1])+(1-2*Fo)*T_prev[i]
  T_next[nodes-1] = 2*Fo*(T_prev[nodes-2]+Bi*T_inf)+(1-2*Fo-2*Bi*Fo)*T_prev[nodes-1]
  Surface_temp.append(T_next[nodes-1])
  T_prev[:] = T_next

plt.plot(timesteps, Surface_temp, label="Surface Temperature")
plt.plot(timesteps, Midplane_temp, label = "Midplane Temperature")
plt.xlabel("Time in seconds")
plt.ylabel("Temperature in Celsius")
plt.legend()
```

3. Although the value of Δx is specified in the problem statement, it can be varied to obtain more accurate results. As a general rule, results get more accurate as you increase the number of nodes (decrease Δx) *as long as it is within the stability criterion.*

   The table(s) below summarizes the results for different values of Δx:

del_x (mm)	del_t (ms)	Time (s)	Mid-plane T (°C)	Length(m)
50	50	101.15	1370.3915	1.5172
25	50	147.05	1363.4109	2.2058
20	50	153.45	1363.1734	2.3018
10	50	160.45	1364.5088	2.4068
5	50	162.05	1365.0544	2.4308
2	50	162.45	1365.2492	2.4367
1	50	162.50	1365.2825	2.4375
0.8	50	162.55	1365.2541	2.4383

Footnotes

Reference [1]:
Incropera, Frank P, and David P. DeWitt. Fundamentals of Heat and Mass Transfer. New York: J. Wiley, 2002. Print. Turabian (6th ed.).