[redacted]

[redacted]

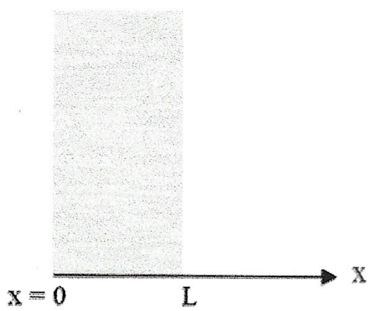
[redacted]

[redacted]

Project description

Write a computer code to numerically solve the 1D rectangular differential equation for steady state:

$$k\,\frac{d^2T}{dx^2} + g = 0$$



Where k = thermal conductivity, g = volumetric heat of generation.

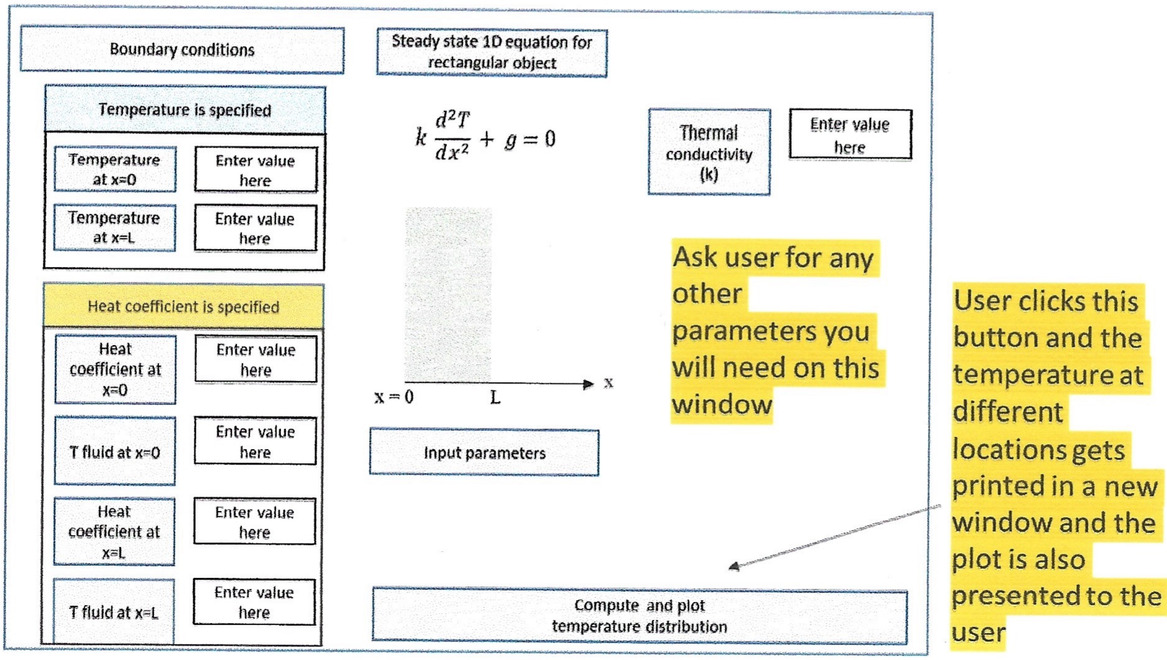
To solve this equation, two boundary conditions are required. The code should implement at least the following two boundary conditions:

i) Temperature is specified at the boundary (at x = 0, or at x = L)
ii) Heat transfer coefficient and temperature of fluid are specified (at x = 0, or at x = L)
iii) A combination of the above two should be possible, for example temperature is specified at x = 0, while convection is specified at x = L

BONUS (20 points extra if you SUCCESSFULLY implement the following)
There is a third boundary condition that can be specified, which is to specify the flux at x = 0 or x = L. If you **successfully implement** this, you will get bonus 20 points. To be considered "successfully", the answer generated should be correct, and the user should be able to select any boundary condition at x = 0, and x = L (including the above two boundary conditions).

Code specification and expectation

1) When code is run, a window should pop up that will display the equation being solved. The user will interact with the code using this window. An example of this window is shown below.



NOTE : This is just an example window and is not complete.

2) The user should be able to select only one boundary condition for x = 0, and one for x = L. For example, if the user selects more than one boundary conditions for x = 0, then this is not physically possible, and the code cannot give the solution. So, the code should check for this possibility and raise an error window/message alerting the user of this problem.

3) The code should be able to work if g = 0 (meaning no heat of generation) and also for g being a non-zero value.

4) After entering all details, a user should be able to press a button, and the temperature in the object should be presented as computed by the code, and it should also be plotted.

Examples to test code:

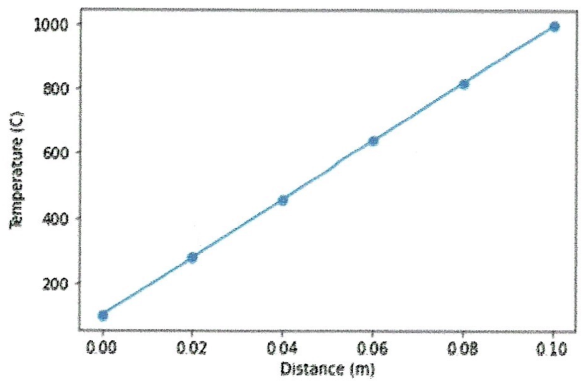
Example 1

BC at x = 0, temperature is specified as 100 C
BC at x = L, temperature is specified as 1000 C

Other parameters are :
k = 23 W/mC
L = 0.1 m
g = 0

Solution

Location (x : m)	Temperature (C)
0	100
0.02	280
0.04	460
0.06	640
0.08	820
0.1	1000



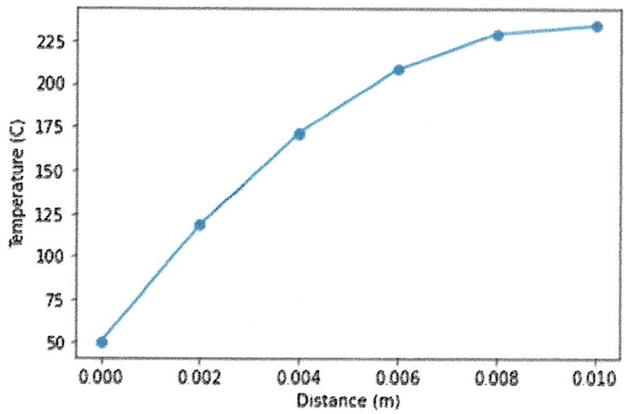
This output is just an example. You can display the results however you like as long as they are clear to understand. Don't forget the labels on x-axis and y-axis.

Example 2

BC at x = 0, temperature is specified = 50 C
BC at x = L, heat transfer coefficient (200 W/m2C) and temperature of fluid are specified (T_∞ = 100 C)

Other parameters are:
k = 18 W/mC
L = 0.01 m
G = 7.2 x 10^7 W/m^3

Location (x : m)	Temperature (C)
0	50
0.002	119
0.004	172
0.006	209
0.008	230
0.01	235



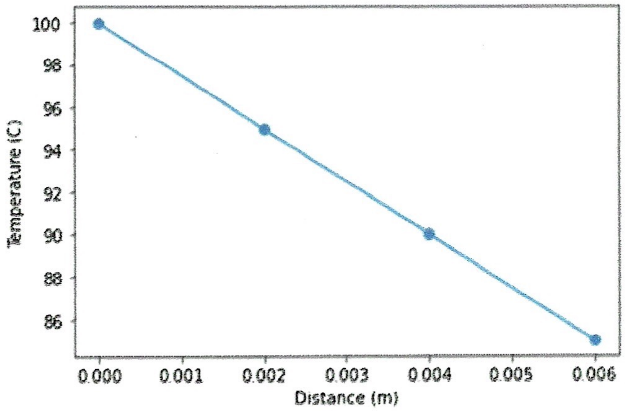
This output is just an example. You can display the results however you like as long as they are clear to understand. Don't forget the labels on x-axis and y-axis.

Example 3

BC at x = 0, flux is specified = 50000 W/m2
BC at x = L, temperature is specified = 85 C

Other parameters are:
k = 20 W/mC
L = 0.6 x 10^{-2} m
G = 0 W/m^3

Location (x : m)	Temperature (C)
0	100
0.002	95
0.004	90
0.006	85



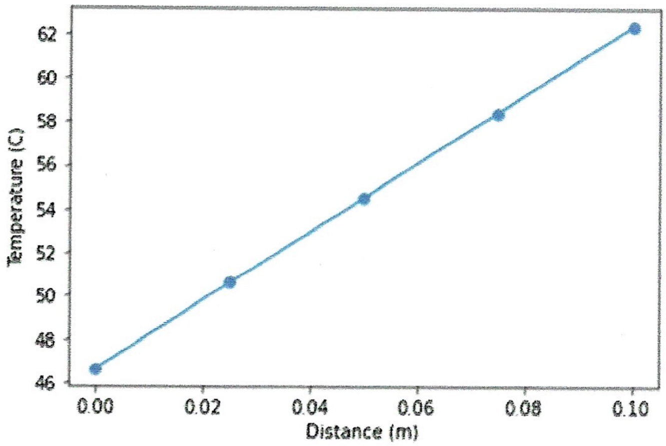
This output is just an example. You can display the results however you like as long as they are clear to understand. Don't forget the labels on x-axis and y-axis.

Example 4

BC at x = 0, heat transfer coefficient (150 W/m2C) and temperature of fluid are specified (T_∞ = 10 C)
BC at x = L, flux is specified = 5500 W/m2

Other parameters are:
k = 35 W/mC
L = 0.1 m
G = 0 W/m^3

Location (x : m)	Temperature (C)
0	46.7
0.025	50.6
0.05	54.5
0.075	58.4
0.1	62.4



This output is just an example. You can display the results however you like as long as they are clear to understand. Don't forget the labels on x-axis and y-axis.

Suggested approach

1) First make code that works without using graphical interface. Test it, and once you know that it works, then put a graphical interface around it.

How to make graphical interface

2) How to make graphical interface in MATLAB:
    a. Use a tool called App Designer
    b. https://www.mathworks.com/videos/app-designer-overview-1510748719083.html
    c. There are other ways, and if you like to use them go ahead.
3) How to make a graphical interface in PYTHON:
    a. Use the inbuilt module called tkinter or other approaches (there are many).
    b. https://www.tutorialsteacher.com/python/create-gui-using-tkinter-python
    c. https://www.youtube.com/watch?v=Y6cir7P3YUk

Scoring Metric

1) Your code will be tested using the example cases given above, and by using other cases. The code should work for all cases. If it does not, you get 15 points for effort made. If you simply write a few lines that are gibberish and I see you did not really make an effort, you get a '0'.
2) If you complete your code WITHOUT graphical interface (and it works) : 30 points
3) If you complete your code WITH graphical interface (and it works) : 50 points
4) If you also successfully implement the bonus boundary condition
    a. 10 bonus points without graphical interface
    b. 20 bonus points with graphical interface

NOTE: If you do not implement the graphical interface, the user should still be asked for input via the console/terminal. The user should not be expected to change the code to run different scenarios. If your code works but you expect user (me) to change code for every scenario, then you will receive 15 points even if the code works.

How to numerically solve the equation

1) See the material on "Finite Difference" attached.

For the nodes that are not at the boundary, the following equation holds.

$$(T_{m-1} - 2T_m + T_{m+1}) + \frac{(\Delta x)^2 \, g_m}{k} = 0$$

This is equation 5-10 of scanned pages (see below).