MM2041 Programming Assignment

Name :- Manab Kumar Jena
(MM20B036)
Name :- Neeraj Barupal
(MM20B044)

Questn no-1 done → Matlab Code:

- .m file attached in ZIP file
- proof of codegeneration attached
- matlab code in txt file attached.
- Assignment also attached.

Programming Assignment

[1] Given → a 2d square grid
→ at steady state.

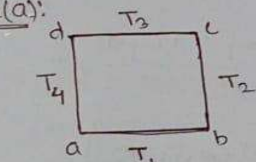
Convention: All temp are taken as degree celcius. (please note).

2cases:-
(a) All sides have constant temperature
(different from each other)

(b) All sides have temperature varies with func of position. ~~[illegible]~~

So, for 2 cases, we would make 2 different codes (matlab)
[all features of codes would be same only input func would varies].

Case(a):



for a steady state condn on 2-d plate from lecture

we know, $\boxed{\dfrac{\partial^2 T}{\partial x^2} + \dfrac{\partial^2 T}{\partial y^2} = 0}$

derived from,

3-d form → $\dfrac{\partial^2 T}{\partial x^2} + \dfrac{\partial^2 T}{\partial y^2} + \dfrac{\partial^2 T}{\partial z^2} + \dfrac{g}{k} = \dfrac{1}{\alpha}\dfrac{dT}{dt}$

vol. heat generatn.
↳ thermal diffusivity
thermal conductivity

In our case,

- it's a 2-d plate on xy plane.
  $\Rightarrow \dfrac{\partial^2 T}{\partial z^2} = 0$
- no heat source ⇒ no heat generatn ⇒ $g = 0$
  $\Rightarrow \dfrac{g}{k} = 0$
- Steady state ⇒ temperature doesnot change with time.
  $\dfrac{\partial T}{\partial t} = 0.$

So, we get laplace equatn. ⇒ $\boxed{\dfrac{\partial^2 T}{\partial x^2} + \dfrac{\partial^2 T}{\partial y^2} = 0}$ --- (1)

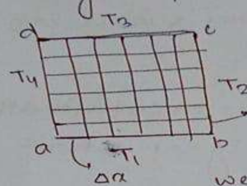
From centred finite difference method,

we know that:

$$\boxed{\frac{\partial^2 T}{\partial x^2} = \frac{T_{i-1,j} - 2T_{i,j} + T_{i+1,j}}{(\Delta x)^2}} \text{ and } \boxed{\frac{\partial^2 T}{\partial y^2} = \frac{T_{i,j-1} - 2 * T_{i,j} + T_{i,j+1}}{(\Delta y)^2}}$$

--- eq(2)

where, (Δx) and (Δy) → are nodes separatⁿ along x and y respectively.



→ Now square grid looks like this with the node.

→ As we are free to chose the number of nodes,

we had taken ~~assumption~~ that $(\Delta x = \Delta y)$ always.

Now, putting all the values (of eqⁿ 2) in eq 1, $\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0\right)$

$$\frac{T_{i-1,j} - 2T_{i,j} + T_{i+1,j}}{(\Delta x)^2} + \frac{T_{i,j-1} - 2T_{i,j} + T_{i,j+1}}{(\Delta x)^2} = 0$$

$$\Rightarrow \boxed{T_{i-1,j} + T_{i+1,j} - 4 * T_{i,j} + T_{i,j+1} + T_{i,j-1} = 0}$$

→ finite difference approximation eqⁿ that we are going to use in code to get all nodal temperature.

$$\boxed{T_{i,j} = \frac{T_{i-1,j} + T_{i+1,j} + T_{i,j+1} + T_{i,j-1}}{4}}$$

generalised eqⁿ.

Now, for case (a):

wall temperature are constant.

Boundary condition that we would use:

(a) all nodes of lower wall $= T_1$

(b) —— do —— right sidewall $= T_2$

(c) —— do —— left —do— $= T_4$

(d) —— do —— upper wall $= T_3$.

Corner nodes would be average of 2 adjacent wall to the corner.

All nodes except this walls node should be in the initial temperature or room temperature.

And now we would apply our centre finite difference eqn to get the temperature of all other nodes.

And slowly it would converge to steady state, so for that to end the loop, we have kept a error func, that would determine the terminatn of loop and steady state temperature of all nodes.

When change in temperature < ϵ_0 (i.e 10^{-3} for our func) we would stop the loop and take respective temperature as the final steady state temperature and then plot the contour graph to represent temperature difference.

Now, for case (b):

Here also all procedure remains same only except that the boundary condn for nodes, where all boundary nodes have equal temperature, here the nodes temperature would depend upon the func.

And after boundary nodes temperature given we could proceed to solve as before from centre finite difference method.

<u>Matlab code details</u>:

for case (a): (const temp on wall)

Give input as.

>> $\underbrace{\text{Temperature\_dist}}_{\text{func name}}$(w, Nx, Tinit, TL, TR, TB, TT)

w → dimension of square plate ($\overbrace{\text{length and breadth}}^{\text{both equal (square)}}$)

Nx → no. of nodes we want on both axis

[Remember: more the node, more would be accuracy but computational time also ↑).

So try to take optimal no. of nodes (like 50-100)

Tinit → Initial temperature of the square domain.

TL, TR, TB, TT → Left, Right, Bottom, Top wall temperature respectively.

- ~~mycode~~ our code/scriptfile/.m file attached along the pdf.
- Example:

dimensi   Tinit

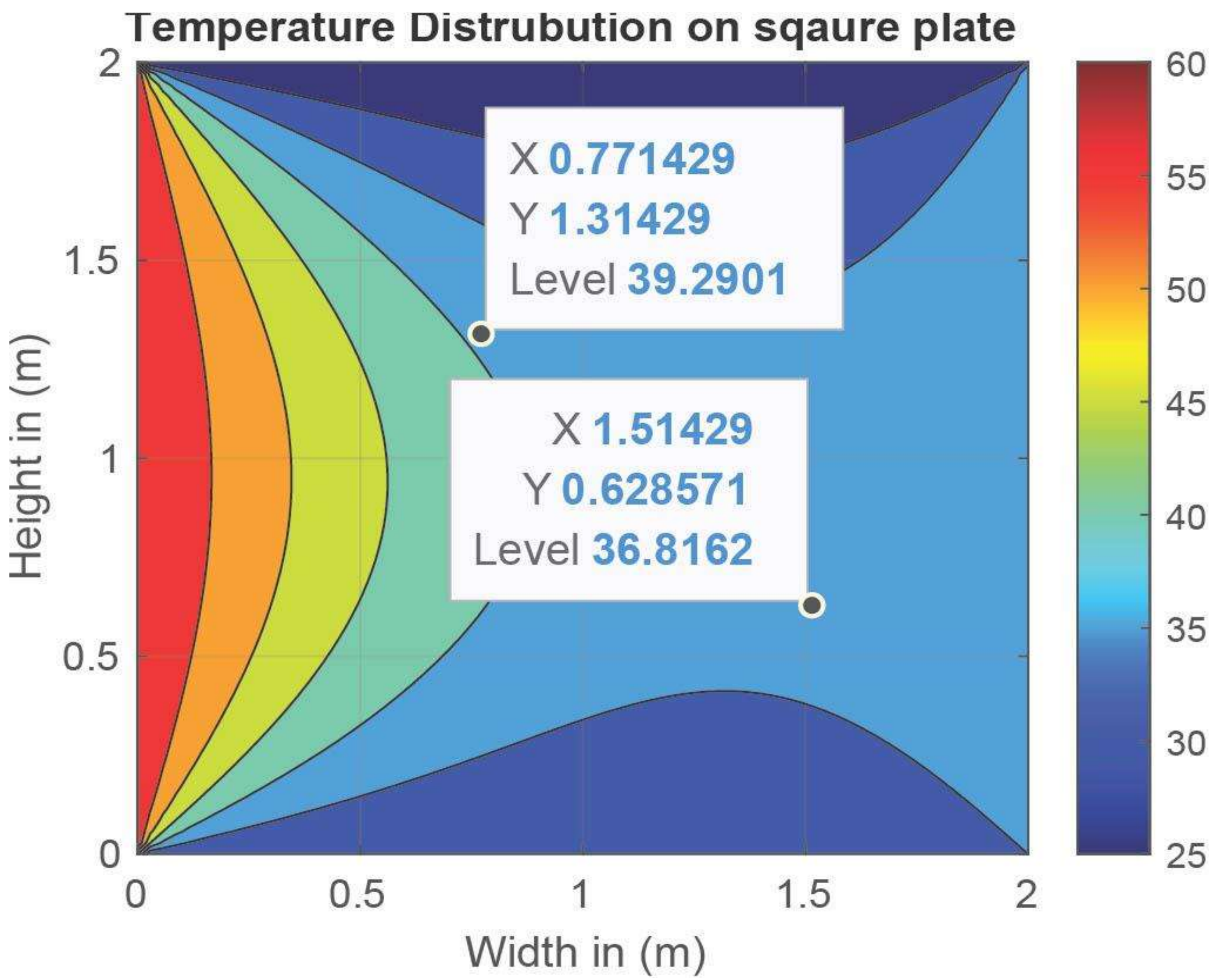
$$\gg \text{Temperaturedist}\ (2,\underbrace{71}_{\text{nodes}},25,\underbrace{60,40,30,25}_{\text{walltemp}})$$

→ Graph obtained is as below

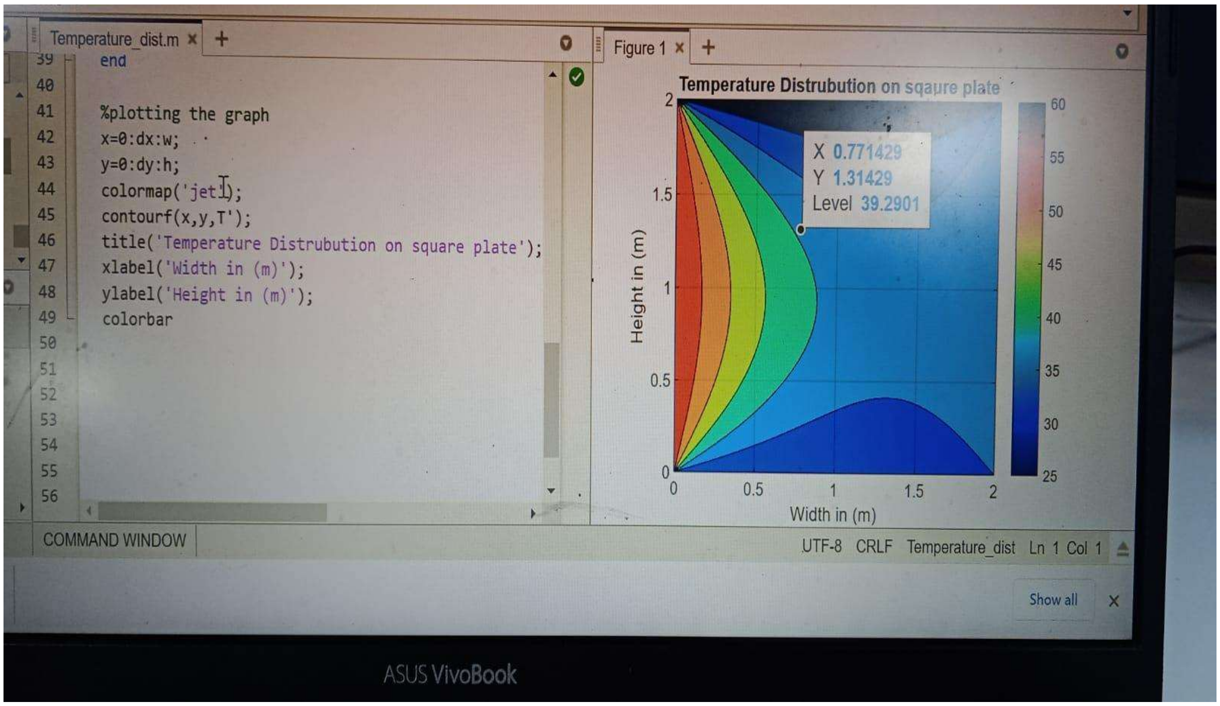
for a point (0.8, 1.341286)
↓ X   ↓ Y
temperature = 38.5799°C

for 2 point as example, I have plotted them in graph

X	Y	Temperature
0.77	1.314	39.29°C
1.514	0.628	36.81°C


Temperature Distrubution on sqaure plate
X 0.771429
Y 1.31429
Level 39.2901
X 1.51429
Y 0.628571
Level 36.8162
Height in (m)
Width in (m)
0
0.5
1
1.5
2
25
30
35
40
45
50
55
60

PROOF FOR CODE GENERATION:



for case (b),

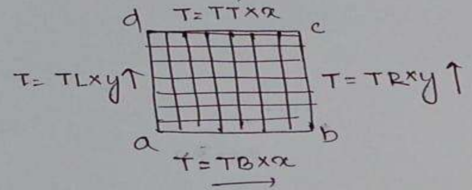
→ temperature can be varied in many ways but for simplicity,

⇒ I have assumed temperature to be linearly varied along wall as,

$\boxed{T = T_0 x}$ or $\boxed{T = T_0 y}$

So, we need this constant for each wall.

⇒ 2nd simplification, we have done to fix the number of nodes (as 50). In 1st case we could change node number



→ here boundary nodes would have different temperature.

Matlab Code

initialtemp (in °C)

$\underbrace{\text{Temperature\_dist\_2}}_{\text{func name}}\ (\underbrace{w}_{\text{dimension (in m)}},\ Tinit,\ \underbrace{TL, TR, TB, TT}_{\text{constant of variable temp func along wall.}})$

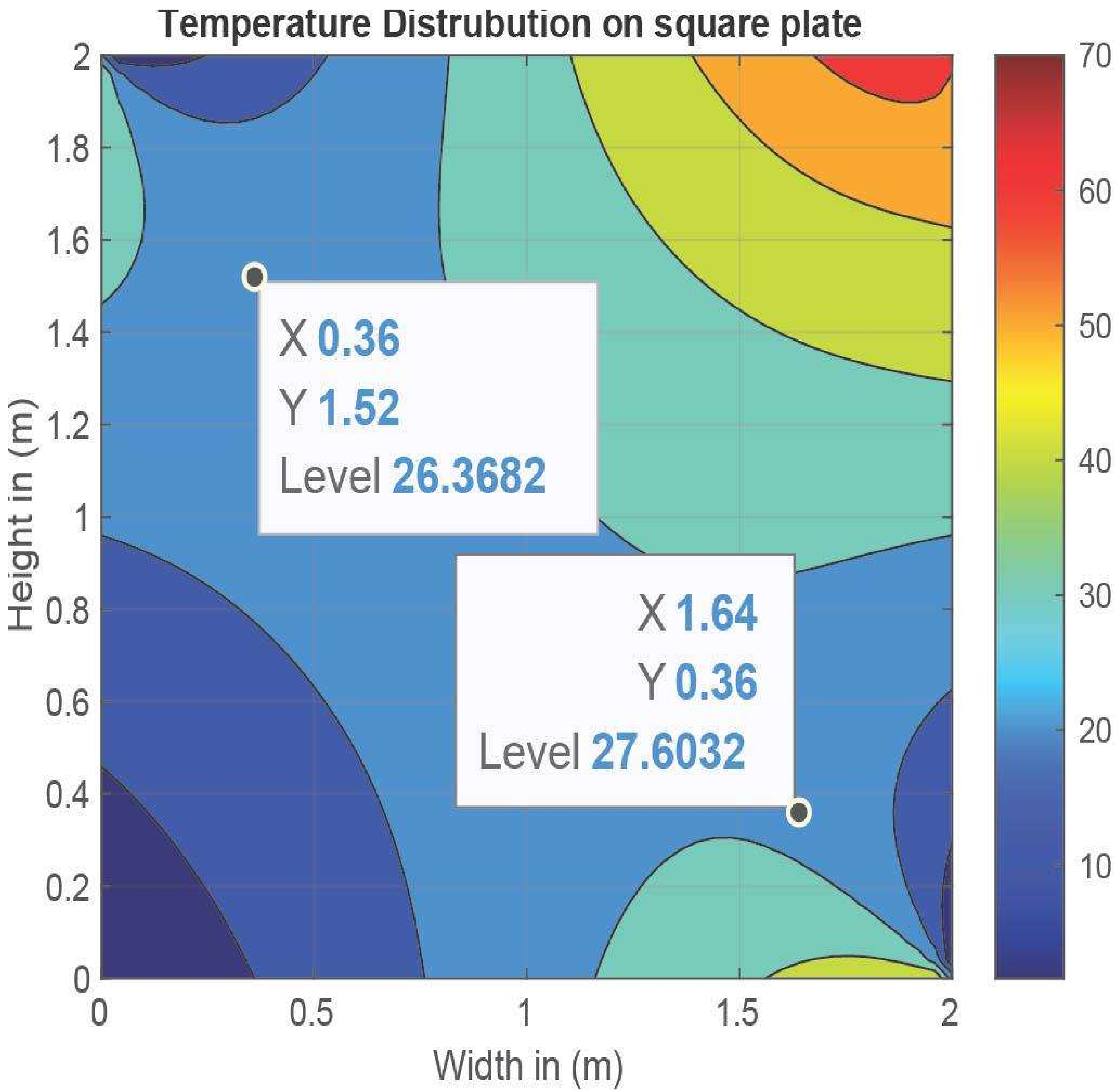
Example

Temperature_dist_2 (2, 0, 20, 30, 25, 35)

Graph → ↓ below.

2 points as example:-

X	Y	Temperature
0.36	1.52	26.3682
1.64	0.36	27.6032


Temperature Distrubution on square plate
X 0.36
Y 1.52
Level 26.3682
X 1.64
Y 0.36
Level 27.6032
Height in (m)
Width in (m)
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
0.5
1.5
10
20
30
40
50
60
70

PROOF FOR CODE GENERATION: