# PH3205-Computational Physics

Spring 2022

Bipradeep Saha (19MS135)

*Indian Institute of Science Education and Research, Kolkata,*
*Mohanpur, West Bengal, 741246, India.*

*April 3, 2022*

# Class Work 3

## Aim

- In the class we implemented shooting method for solving the Boundary value but this time for non-linear ODEs.
- We also implemented finite difference method in class.

## Solution

### Shooting Method For Nonlinear ODE

The given ODE for temperature distribution on a rod is:

$$\frac{d^2T}{dx^2} = -H\left(T_\infty - T\right) - \sigma^4\left(T_\infty^4 - T^4\right)$$

where the boundary conditions are:

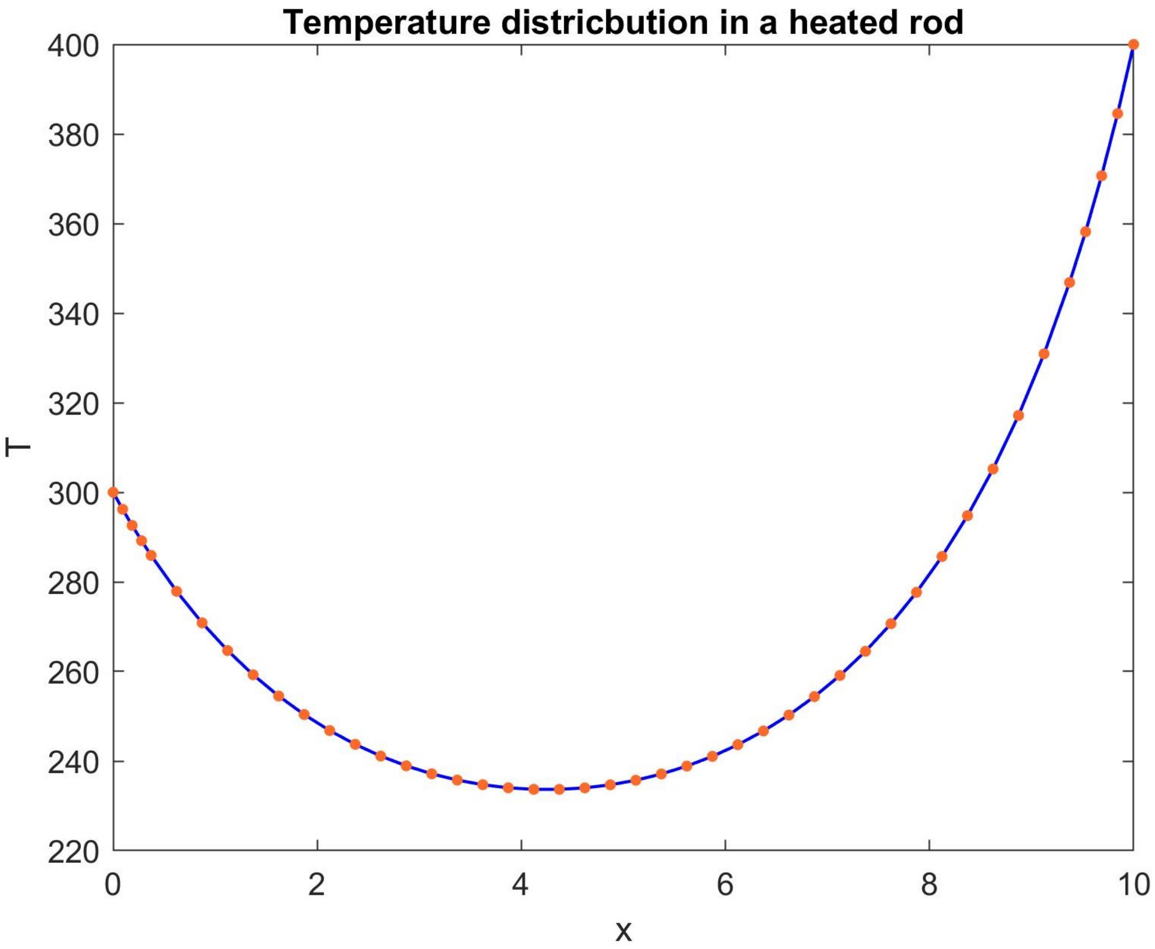
$$T(x=0) = 300^\circ C \qquad T(x=L) = 400^\circ C$$

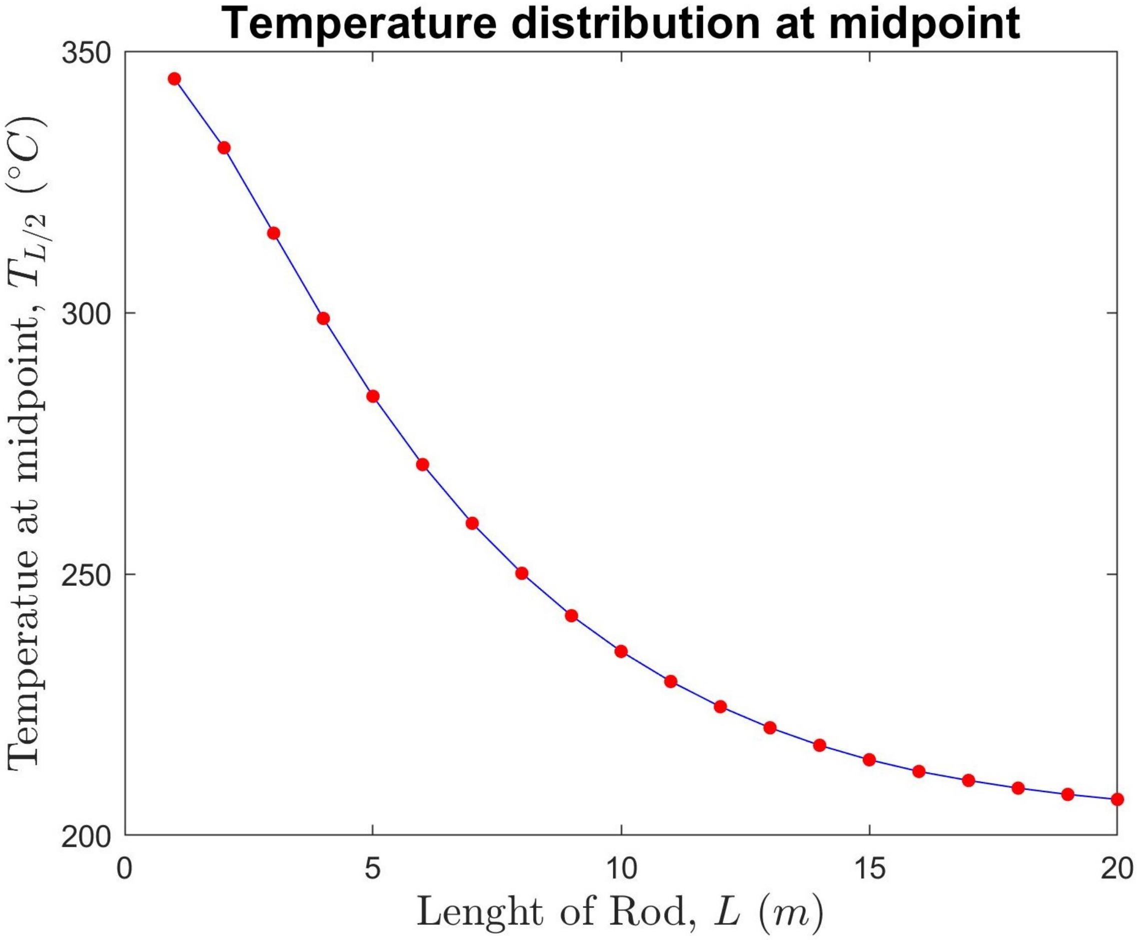
where L is the length of the rod, and rest of the constants were provided to us.

We implemented the shooting method in *Matlab* in a similar way we did for the last assignment, we defined a function $bar\_temp(x,y)$ which returns the system of 1st order ODE for the problem, and then we defined an objective function: $bar\_res(Icguess, L)$ which returns the difference between the end value from the numerical solution and the given boundary value for a given $\frac{dT}{dx}(x=0)$ and the length of the rod.

At first we ran the code for fixed L=10m and plotted the temperature distribution($CW3\_11.jpg$). Then we varied the length of the rod from L=1-20m and plotted the temperature of the rod at the midpoint($CW3\_12.jpg$).

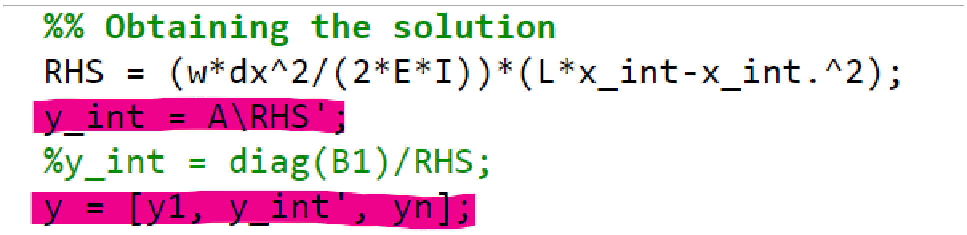
The code is well commented and is in the file: Classwork.m.
The output of the code is the next page.


Temperature districbution in a heated rod
T
x
400
380
360
340
320
300
280
260
240
220
0
2
4
6
8
10


Temperature distribution at midpoint
Temperatue at midpoint, $T_{L/2}$ $(^\circ C)$
Lenght of Rod, $L$ $(m)$
350
300
250
200
0
5
10
15
20

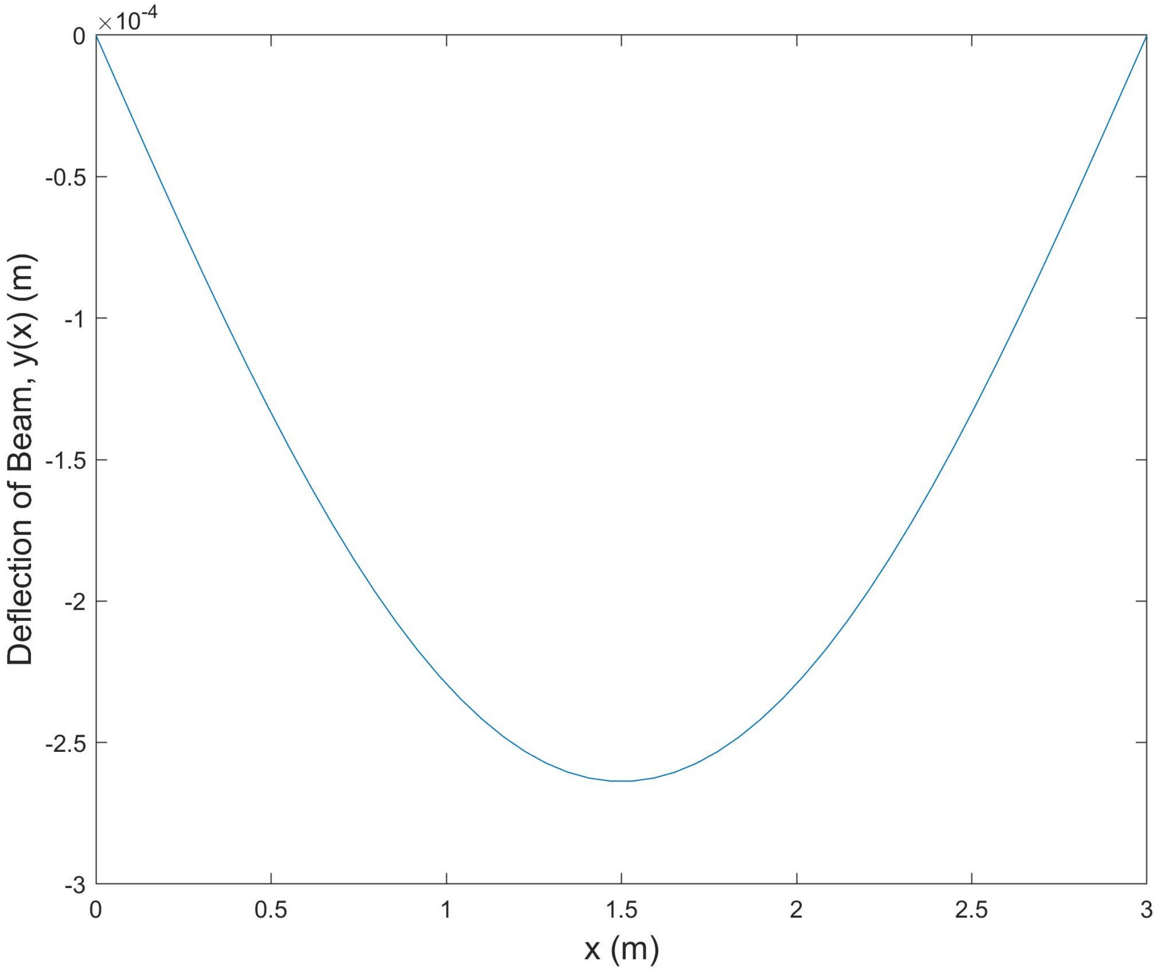
### Finite Difference Method

In this we solved the ODE for deflection of beam using finite difference method. The code was provided to us and we are asked to fix the code. The corrected lines are:



```
%% Obtaining the solution
RHS = (w*dx^2/(2*E*I))*(L*x_int-x_int.^2);
y_int = A\RHS';
%y_int = diag(B1)/RHS;
y = [y1, y_int', yn];
```

The issue was that, the matrix dimensions were not correct. The corrected is in the file: *Classwork_FD.m*
The output of the code is: (*CW3_2.jpg*)


×10^-4
Deflection of Beam, y(x) (m)
x (m)