CAPE Laboratory Assignment – 4

Instructor: Dr Debashis Sarkar

18CH10071, Anshuman Agrawal

Q.1

Problem Statement

Consider the following BVP representing a dimensionless form of the diffusion with chemical reaction in a catalyst pore.

$$\frac{d^2c}{dx^2} = 4c$$

At $x = 0$ (the mouth of the pore), the dimensionless concentration, $c = 1$.
At $x = 1$(the pore end), the gradient of the concentration, $dc/dx = 0$.

Solve the BVP using (a) Shooting Method (your code) and (b) MATLAB function `bvp4c`. Plot the concentration profile along pore length x and compare the results.

MATLAB Code

Shooting Method

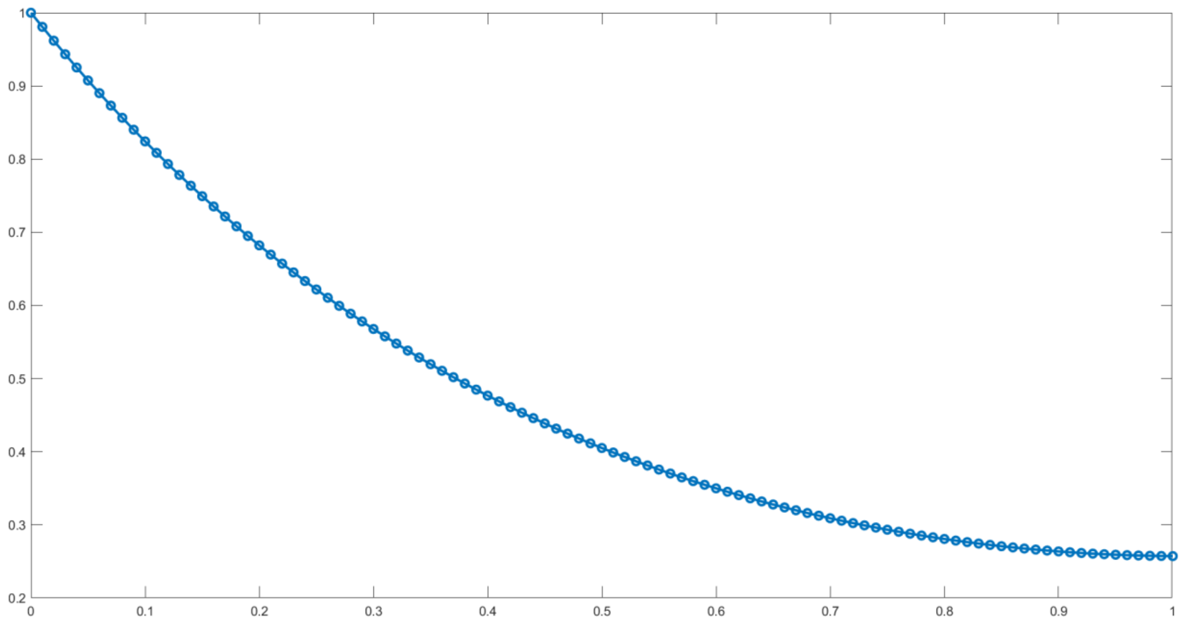
```
clear all;

% defining the problem as 2 variable y1 and y2 with y1 =
c and y2 = dc/dx = dy1/dx
f = @(x,y1,y2) [y2];
g = @(x,y1,y2) [4*y1];
y1(1) = 1; %initial value of c at x = 0
y2(1) = -1.93; % initial guess at x=0 assumed through
trial and error using different values
h = 0.01; % step size for solving the IVP part
x = [0:h:1];
```

```
for i = 1:1/h
    %using euler method to solve the IVP to get matching
values
    y1(i+1) = y1(i) + h*f(x(i),y1(i),y2(i));
    y2(i+1) = y2(i) + h*g(x(i),y1(i),y2(i));
end
plot(x,y1,'-o','LineWidth',2);% plotting concentration
versus pore depth
```

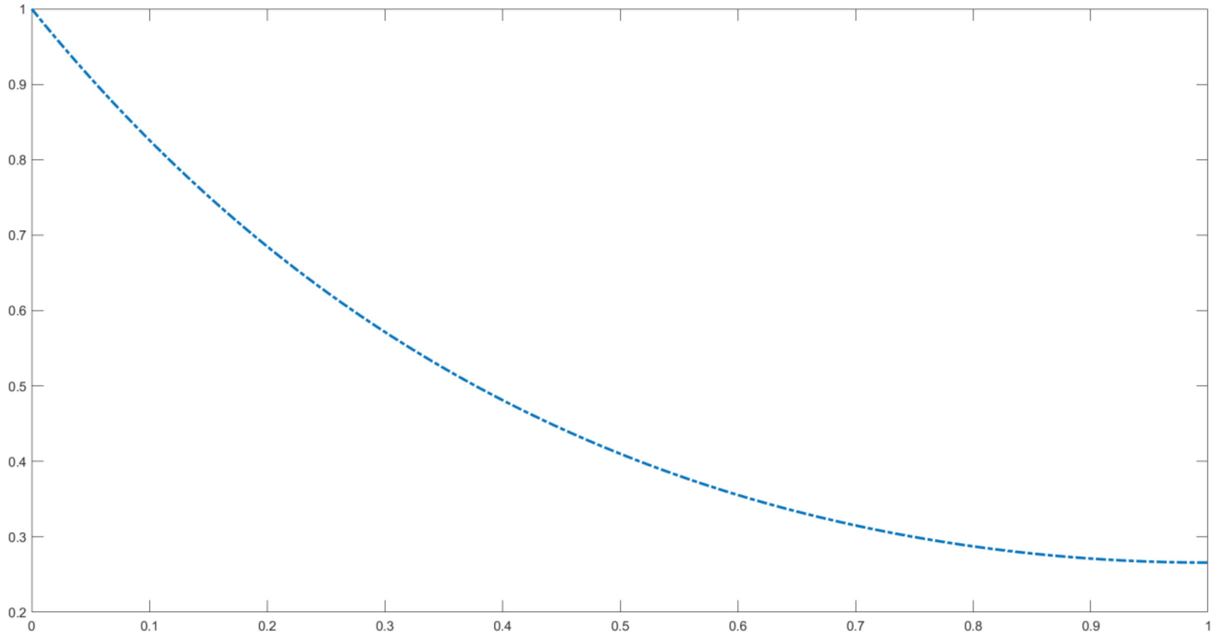
bvp4c method

```
clear all;

% defining the problem as 2 variable y1 and y2 with y1 =
c and y2 = dc/dx = dy1/dx
f= @(x,y) [ y(2) ; 4*y(1)]; % define the 2 variabled
differential equations
bc = @(ya, yb) [ya(1)-1; yb(2)]; % given boundary
conditions
% x=0, c=1; x=1, dc/dx=0
xmesh = linspace(0 , 1 , 100); % intervals
solinit = bvpinit(xmesh, [0 1]);
sol = bvp4c(f, bc,solinit);
c = sol.y;
plot(sol.x, c(1,:),'-.','LineWidth',2);
```

Results



Concentration profile using shooting method



Concentration profile using bvp4c method

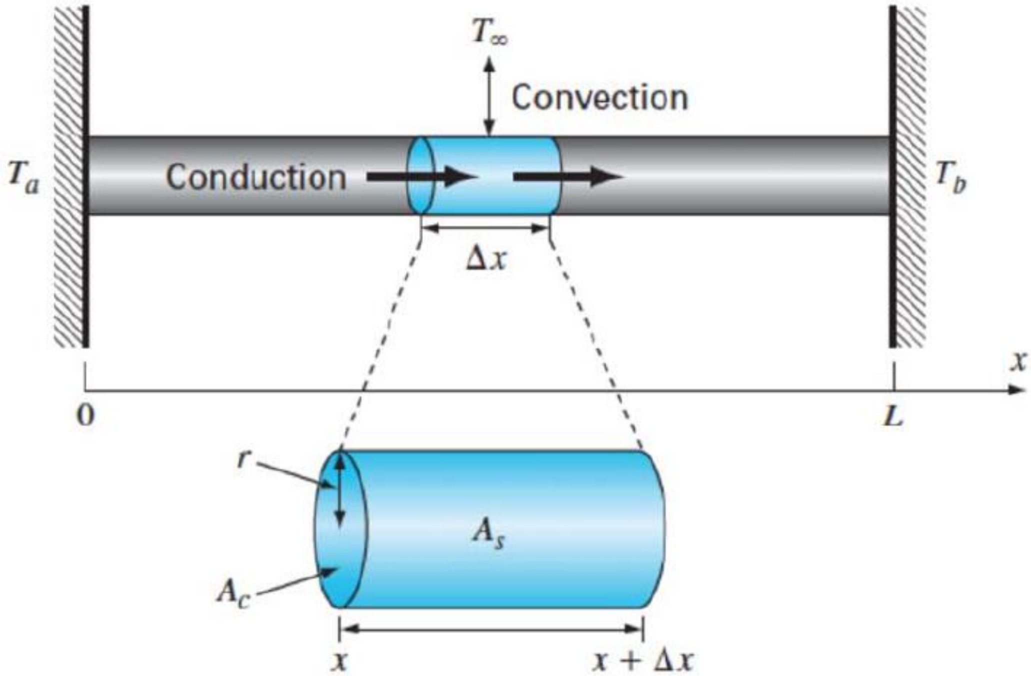
Conclusion

It is noted that the values obtained for dc/ dx at x = 0 using both methods are pretty much comparable. We can't really compare the complexity of both the methods because shooting method involves a lot of trial and error and it might take a really long time until we choose a good enough value to satisfy the condition. Upon trying with several values such 0, 1, -1, -2 and finally when we came upon the -1.9 value, a slight proximity to the given condition was noticed. Therefore, upon guessing nearby values I could arrive at a decent enough result. The *bvp4c* method is a finite difference method and relies more on equations than intuition to reach a final solution and should be the preferred method for boundary value problems. Therefore, we can expect better and more accurate solutions using the direct method of *bvp4c*. The shooting method approach could have been made more accurate but at the cost of computation effort had we used a better explicit method like the Runge – Kutta 4^{th} order method.

Q.2

Problem Statement

Consider the following BVP representing steady state Heat Transfer in a rod of length $L = 10$ m.



$$\frac{d^2T}{dx^2} + h'(T_\infty - T) + \sigma(T_\infty^4 - T^4) = 0$$

Given: $h' = 0.05$ m^{-2}, $\sigma = 2.7 \times 10^{-9}$ K^{-3} m^{-2}

$T_\infty = 200$ K, $T(0) = 300$ K, $T(10) = 400$ K.

Solve the BVP using (a) Shooting Method (your code) and (b) MATLAB function `bvp4c`. Plot the temperature distribution along the length of the rod and compare the results.

MATLAB Code

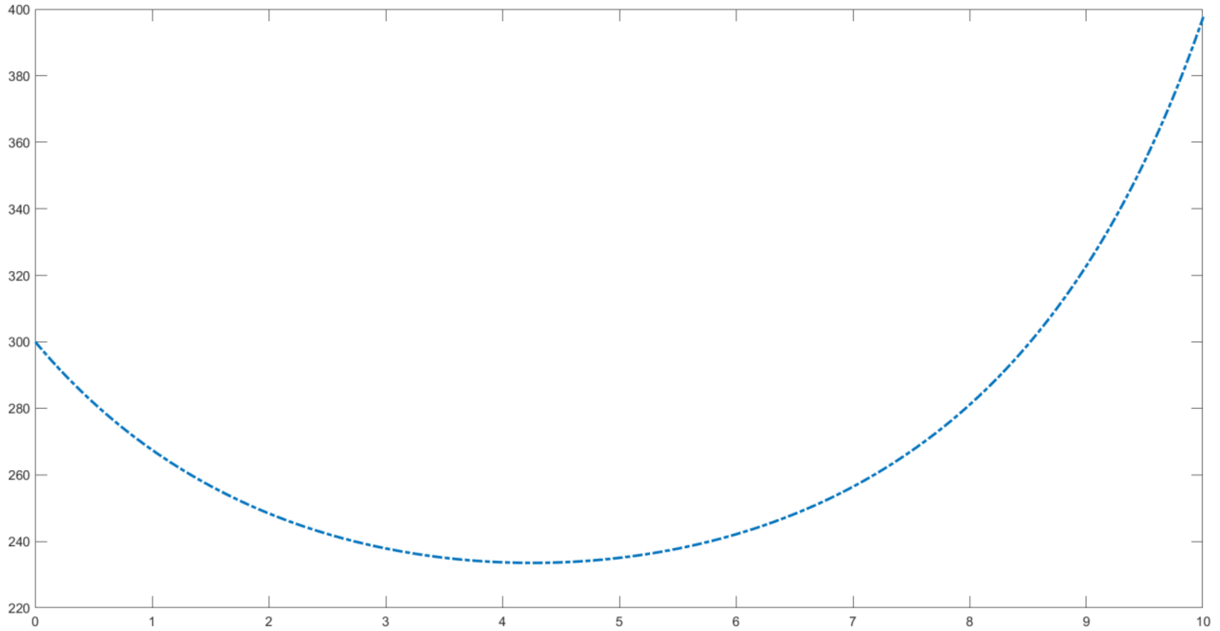
Shooting Method

```
clear all;

h = 0.05;
T0 = 300;
TL = 400;
Tinf = 200;
sigma = 2.7*(10^-9);
% defining differential equations
f = @(x,y1,y2) [y2];
g = @(x,y1,y2)[(-h*(Tinf-y1) - sigma*(Tinf^4 - (y1)^4))];
y1(1) = 300;
y2(1) = -41.75; % initial guess through trial and error
h = 0.01; % step size
x = [0:h:10]; % rod length
for i = 1:10/h
y1(i+1) = y1(i) + h*f(x(i),y1(i),y2(i)); % solving given
system using Explicit Euler method
y2(i+1) = y2(i) + h*g(x(i),y1(i),y2(i));
end
plot(x,y1,'-.','LineWidth',2);
```

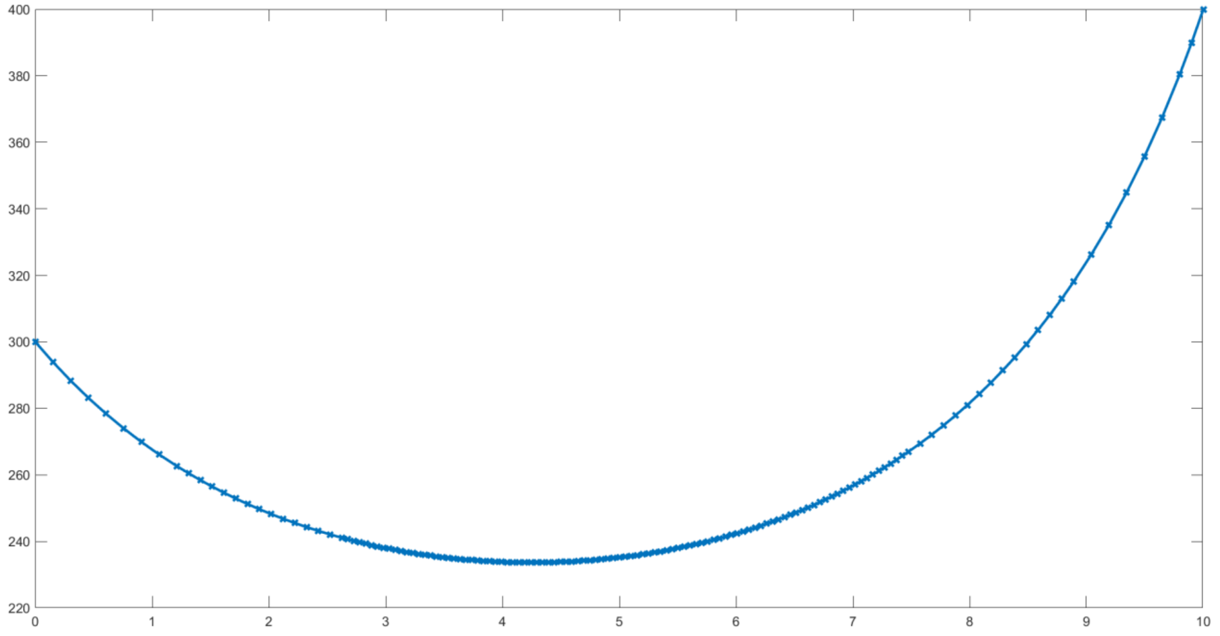
bvp4c method

```
clear all;

h = 0.05;
T0 = 300;
TL = 400;
Tinf = 200;
sigma = 2.7*10^-9;
f = @(x,y) [ y(2) ; (-h*(Tinf-y(1)) - sigma*(Tinf^4 -
(y(1))^4))]; %defining both differential equations
bc = @(ya, yb) [ya(1)-T0; yb(1) - TL]; % given boundary
conditions
xmesh = linspace(0 , 10 , 100); % mesh over which
differnetial equation is to be solved
solinit = bvpinit(xmesh, [0 10]);
sol = bvp4c(f, bc,solinit);
T = sol.y;
plot(sol.x, T(1,:),'-x','LineWidth',2);
```

Results



Temperature distribution obtained using Shooting Method



Temperature distribution obtained using bvp4c method

Conclusion

The *bvp4c* method is clearly a much better method in terms of logic, accuracy and computational effort. Shooting method still carries the disadvantage of arriving at a suitable initial guess after several trials on the given equation. The values in the shooting method seem to be more closely spaced towards the end while *bvp4c* shows that trend around the centre of the x mesh which might be pointing to a slight bias in the former method owing to the initial guess. A possible modification to the shooting method could be the usage of a better explicit method to solve the differential equations so that we can reach much closer to the desired value with the same initial guesses. Unlike the situation in given problem, the radiation heat transfer effects are usually neglected because they are really small in magnitude and don’t cause much change in the overall state of the system. Unless we are dealing with huge temperature differences, blackbodies or unusually large bodies where radiation becomes an important factor, it is safe to neglect it.