Chapter 12. Ordinary Differential Equation Boundary Value (BV) Problems

In this chapter we will learn how to solve ODE boundary value problem. BV ODE is usually given with x being the independent space variable.

\[ y'' + p(x) y' + q(x) y = f(x) \quad a \leq x \leq b \] (1a)

and the boundary conditions (BC) are given at both end of the domain e.g. \( y(a) = \alpha \) and \( y(b) = \beta \). They are generally fixed boundary conditions or Dirichlet Boundary Condition but can also be subject to other types of BC e.g. Neumann BC or Robin BC.

14.1 LINEAR FINITE DIFFERENCE (FD) METHOD

Finite difference method converts an ODE problem from calculus problem into algebraic problem. In FD, \( y' \) and \( y'' \) are expressed as the difference between adjacent \( y \) values, for example,

\[ y'(x) = \frac{y(x+h) - y(x-h)}{2h} \] (1)

which are derived from the Taylor series expansion,

\[ y(x+h) = y(x) + h y'(x) + (h^2/2) y''(x) + \ldots \] (2)

\[ y(x-h) = y(x) - h y'(x) + (h^2/2) y''(x) + \ldots \] (3)
If (2) is added to (3) and neglecting the higher order term ($O(h^3)$), we will get

$$y'(x) = \frac{y(x + h) - 2y(x) + y(x - h)}{h^2}$$  \hspace{1cm} (4)

The difference Eqs. (1) and (4) can be implemented in $[x_1 = a, x_n = b]$ (see Figure) if few finite points $n$ are defined and dividing domain $[a,b]$ into $n-1$ intervals of $h$ which is defined

$$h = \frac{x_n - x_1}{n-1} = x_{i+1} - x_i$$  \hspace{1cm} (5)

FD method gives derivative of $y$ values at point $i$, $y_i$ as follow

$$y'(x_i) \approx \frac{y(x_{i+1}) - y(x_{i-1})}{2h} \hspace{1cm} (6)$$

$$y'(x_i) \approx \frac{y(x_{i+1}) - 2y(x_i) + y(x_{i-1})}{h^2} \hspace{1cm} (7)$$

Substituting (6) & (7) into (1a), we get

$$\begin{bmatrix}
1 - \frac{1}{2} h p(x_i) & \vdots & \vdots \\
\vdots & \ddots & \vdots \\
\vdots & \vdots & 1 - \frac{1}{2} h p(x_{n-1})
\end{bmatrix}
\begin{bmatrix}
y(x_{i-1}) \\
\vdots \\
y(x_{n-1})
\end{bmatrix}
- \begin{bmatrix}
\frac{1}{2} h^2 q(x_i) & \vdots & \vdots \\
\vdots & \ddots & \vdots \\
\vdots & \vdots & \frac{1}{2} h^2 q(x_{n-1})
\end{bmatrix}
\begin{bmatrix}
y(x_i) \\
\vdots \\
y(x_{n-1})
\end{bmatrix}
= \begin{bmatrix}
1 + \frac{1}{2} h p(x_i)
\vdots \\
\vdots \\
1 + \frac{1}{2} h p(x_{n-1})
\end{bmatrix}
\begin{bmatrix}
y(x_{i+1}) \\
\vdots \\
y(x_{n-1})
\end{bmatrix}
+ \begin{bmatrix}
h^2 f(x_{i+1})
\vdots \\
\vdots \\
h^2 f(x_{n-1})
\end{bmatrix}
\hspace{1cm} (8)$$

or can be simplified to,

$$\begin{bmatrix}
1 - \frac{1}{2} h p(x_i) & \vdots & \vdots \\
\vdots & \ddots & \vdots \\
\vdots & \vdots & 1 - \frac{1}{2} h p(x_{n-1})
\end{bmatrix}
\begin{bmatrix}
y_{i-1} \\
\vdots \\
y_{n-1}
\end{bmatrix}
- \begin{bmatrix}
\frac{1}{2} h^2 q(x_i) & \vdots & \vdots \\
\vdots & \ddots & \vdots \\
\vdots & \vdots & \frac{1}{2} h^2 q(x_{n-1})
\end{bmatrix}
\begin{bmatrix}
y_{i} \\
\vdots \\
y_{n-1}
\end{bmatrix}
\begin{bmatrix}
y_{i+1} \\
\vdots \\
y_{n-1}
\end{bmatrix}
+ \begin{bmatrix}
h f(x_{i+1}) \\
\vdots \\
h f(x_{n-1})
\end{bmatrix}
\hspace{1cm} (9)$$

Eq. (9) is then applied to each of the internal nodes, $i = 2, \ldots, n-1$. This will produce a system of linear equations of tri-diagonal form. The system of linear equations can then be solved using the Thomas algorithm (but we will solve using sparse matrix technique).
Example 1

Solve the following 2nd order ODE,

\[ y'' + 7y' + 3y = 1 \]

for \([0,1]\) with Dirichlet boundary condition, \(y(0) = 0\) and \(y(1) = 1\), using FD with \(h = 0.1\) (Note in this example \(p(x) = 7\), \(q(x) = 3\) and \(f(x) = 1\)). The difference equation (9) for this problem is,

\[ (1 - 3.5h)y_{i-1} - (2 - 3h^2)y_i + (1 + 3.5h)y_{i+1} = h^2 \]

for \(h = 0.1\),

\[ 0.65\ y_{i-1} - 1.97\ y_i + 1.35\ y_{i+1} = 0.01 \]

The difference equation is applied to the internal nodes \((i = 2 - 10)\),

Applying (C-1) at \(i = 2 - 10\), we obtain the following system of linear equations

\[
\begin{align*}
  i=2 & \rightarrow -1.97y_2 & + 1.35y_3 & = 0.01 - 0.65y_1 \\
  i=3 & \rightarrow 0.65y_2 & - 1.97y_3 & + 1.35y_4 & = 0.01 \\
  i=4 & \rightarrow 0.65y_3 & - 1.97y_4 & + 1.35y_5 & = 0.01 \\
  i=5 & \rightarrow 0.65y_4 & - 1.97y_5 & + 1.35y_6 & = 0.01 \\
  i=6 & \rightarrow 0.65y_5 & - 1.97y_6 & + 1.35y_7 & = 0.01 \\
  i=7 & \rightarrow 0.65y_6 & - 1.97y_7 & + 1.35y_8 & = 0.01 \\
  i=8 & \rightarrow 0.65y_7 & - 1.97y_8 & + 1.35y_9 & = 0.01 \\
  i=9 & \rightarrow 0.65y_8 & - 1.97y_9 & + 1.35y_{10} & = 0.01 - 1.35y_{11} \\
  i=10 & \rightarrow 0.65y_9 & - 1.97y_{10} & + 1.35y_{11} & = 0.01 - 1.35y_{11}
\end{align*}
\]

Note that nodes \(i = 1 \& 11\) are end nodes, which have specified values and they do not appear in the system. The above system of linear equations can be written in tri-
diagonal matrix form, which can be solved using *Thomas algorithm* but we will solve using sparse technique.

\[
\begin{bmatrix}
-1.97 & 1.35 & 0 & \cdots & 0 \\
0.65 & -1.97 & 1.35 & 0 & \cdots \\
0 & 0.65 & -1.97 & 1.35 & 0 & \cdots \\
0 & \cdots & 0 & 0.65 & -1.97 & 1.35 & 0 & \cdots \\
0 & \cdots & 0 & \cdots & 0 & 0.65 & -1.97 & 1.35 & 0 \\
0 & \cdots & 0 & \cdots & 0 & \cdots & 0 & 0.65 & -1.97 \\
0 & \cdots & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0
\end{bmatrix}
\begin{bmatrix}
y_2 \\
y_3 \\
y_4 \\
y_5 \\
y_6 \\
y_7 \\
y_8 \\
y_9 \\
y_{10}
\end{bmatrix}
= \begin{bmatrix}
0.01 \\
0.01 \\
0.01 \\
0.01 \\
0.01 \\
0.01 \\
0.01 \\
0.01 \\
-1.34
\end{bmatrix}
\]

The solution of the above system is

\[
y^T = [y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10}] = [0.6420, 0.9443, 1.0762, 1.1233, 1.1284, 1.1131, 1.0885, 1.0598, 1.0299]
\]

Note that the difference equations at nodes \( i = 2 \) and \( i = n-1 = 10 \) are slightly modified by the end nodes,

\[
i = 2 \quad \left[ -2 \cdot h^2 q(x_i) \right] y_2 + \left[ 1 + \frac{1}{2} h \cdot p(x_i) \right] y_3 = h^2 f(x_i) \left[ 1 + \frac{1}{2} h \cdot p(x_i) \right] y_1 \quad (10)
\]

\[
i = 10 \quad \left[ 1 - \frac{1}{2} h \cdot p(x_i) \right] y_9 - \left[ 2 \cdot h^2 q(x_i) \right] y_{10} = h^2 f(x_i) \left[ 1 + \frac{1}{2} h \cdot p(x_i) \right] y_{11} \quad (11)
\]

The sample MATLAB code (*Example1.m*) I developed to solve this problem is shown below.

---
% Program to solve linear ODE boundary value problems with FD
% Both ends are subjected to Dirichlet boundary conditions
% inputs :
% a,b are starting and ending points
% alfa,beta are the y values at the starting and ending points
% n is the number of interval

---

Kosasih 2012 Chapter 12 ODE Boundary Value Problems 4
% outputs:
% y are the values of y at node points
%***************************************************************************
clear;clf;

%********** INPUT *******************************************************
a = input(’Enter the starting x ’);
b = input(’Enter the ending x ’);
alfa = input(’Enter the y value at starting x ’);
beta = input(’Enter the y value at ending x ’);
n = input(’Enter the number of divisions ’);
pfunc = input(’Enter the function p(x) please: ’); qfunc = input(’Enter the function q(x) please: ’); ffunc = input(’Enter the function f(x) please: ’);
p = inline(pfunc);
q = inline(qfunc);
f = inline(ffunc);

%*********** Constants ************
h = (b - a)/n; % the interval size
x = linspace(a,b,n+1);

%************* Elements of the matrix A
A(1:n-1,1:n-1) = 0.0; % initialize matrix A
for i = 1:n-1 % this loop calculate element for nodes 2 to n
    % in the diagram but in the
    % matrix it is from i = 1 to n-1 
    if i == 1
        A(i,i) = -(2. - q(x(i+1)) * h*h);
        A(i,i+1) = 1 + 0.5 * p(x(i+1)) * h;
        ff(i) = f(x(i+1)) * h*h - (1. - p(x(i+1)) * 0.5 * h) * alfa;
    elseif i == n-1
        A(i,i-1) = 1. - p(x(i+1)) * 0.5 * h;
        A(i,i) = -(2. - q(x(i+1)) * h*h);
        ff(i) = f(x(i+1)) * h*h - (1 + 0.5 * p(x(i+1)) * h) * beta;
    else
        A(i,i-1) = 1. - p(x(i+1)) * 0.5 * h;
        A(i,i) = -(2. - q(x(i+1)) * h*h);
        A(i,i+1) = 1 + 0.5 * p(x(i+1)) * h;
    end
ff(i) = f(x(i+1)) * h*h;
end
end

%************* Solve the system of linear equations using sparse matrix
ASparse = sparse(A);
y = ASparse\ff;

%************* PLOT OF RESULT ********
plot(x, [alfa;y;beta], 'b-')
xlabel('x'), ylabel('y');
legend('numerical results')

Enter the starting x >0
Enter the ending x >1
Enter the y value at starting x >0
Enter the y value at ending x >1
Enter the number of divisions >10
Enter the function file of p(x) please: 7.
Enter the function file of q(x) please: 3.
Enter the function file of f(x) please: 1.

Now suppose the ODE Eq. (1)
\[ y'' + p(x) y' + q(x) \quad y = f(x) \quad a \leq x \leq b \] (1)

is now subject to boundary conditions \( y(a) = \alpha \) and \( y'(b) = \beta \). The boundary at \( b \) is a derivative boundary condition or *Neumann* Boundary Condition. For the problem the value of \( y(b) \) must be calculated (part of the solution) so difference equations must be written for \( i = 1, 2, \ldots, n \). For \( i = 1, 2, 1, 2, \ldots, n-1 \) according to the FD equation (9) is but for \( i = n \) it is different as \( y''(b) \) needs \( y(x_n+h) \) as shown below.

\[
y'(x_n) = \frac{y(x_n + h) - 2y(x_n) + y(x_n - h)}{h^2} = \frac{y_{n+1} - 2y_n + y_{n-1}}{h^2} \quad (12)
\]

It is clear from (12) that \( y(b+h) \) is needed, and this is outside the domain as shown in the figure below.

To calculate \( y_{n+1} \), we will make use of the given \( y'(x_n) = \beta \) and following (6),

\[
y'_n \approx \frac{y_{n+1} - y_{n-1}}{2h} \quad (13)
\]

So,

\[
y_{n+1} = y_{n-1} + 2h y'_n \quad (14)
\]

If (14) is substituted into (12)

\[
y'_n = \frac{2h y'_n - 2y_n + 2y_{n-1}}{h^2} \quad (15)
\]

and substituting (15) and \( y'_n = \beta \) into (1), we get

\[
2y_{n-1} - \left(2 - h^2 q(x_n)\right)y'_n = h^2 f(x_n) - 2h\beta - h^2 p(x_n)\beta
\]

(16)
Eq. (16) is the FD equation applied at node \( i = n \).

--- Example 2 -----------------------------------------------

Solve the following 2nd order ODE,

\[
y'' + 5y' + 4y = 1
\]

For \([0,1]\) and boundary conditions \( y(0) = 1 \) and \( y'(1) = 0 \). Use \( h = 0.1 \), in this problem \( p(x) = 5 \), \( q(x) = 4 \) and \( f(x) = 1.0 \). The FD equations are given by (9),

\[
(1 - 2.5h)y_{i-1} - (2 - 4h^2)y_i + (1 + 2.5h)y_{i+1} = h^2
\]

For \( h = 0.1 \),

\[
0.75y_{i-1} - 1.96y_i + 1.25y_{i+1} = 0.01 \quad \text{...(C-1)}
\]

(C-1) is applicable for \( i = 2 \) – 10, but for \( i = 11 \), (16) is used

\[
2y_9 - 1.96y_{10} = 0.01 \quad \text{...(C-2)}
\]

The system of linear equations to be solved is

\[
\begin{array}{cccc}
i = 2 & \rightarrow & -1.96y_2 & 1.25y_3 \\
i = 3 & \rightarrow & 0.75y_2 & -1.96y_3 & 1.25y_4 \\
i = 4 & \rightarrow & 0.75y_3 & -1.96y_4 & 1.25y_5 \\
i = 5 & \rightarrow & 0.75y_4 & -1.96y_5 & 1.25y_6 \\
i = 6 & \rightarrow & 0.75y_5 & -1.96y_6 & 1.25y_7 \\
i = 7 & \rightarrow & 0.75y_6 & -1.96y_7 & 1.25y_8 \\
i = 8 & \rightarrow & 0.75y_7 & -1.96y_8 & 1.25y_9 \\
i = 9 & \rightarrow & 0.75y_8 & -1.96y_9 & 1.25y_{10} \\
i = 10 & \rightarrow & 0.75y_9 & -1.96y_{10} & 1.25y_{11} \\
i = 11 & \rightarrow & 0.75y_9 & -1.96y_{10} & 2y_{10} - 1.96y_{11}
\end{array}
\]

= 0.01 - 0.75y_1

= 0.01

= 0.01

= 0.01

= 0.01

= 0.01

= 0.01

= 0.01

= 0.01

In matrix form,

Kosasih 2012 Chapter 12 ODE Boundary Value Problems 8
The solution vector is

\[ y^T = [0.7051, 0.5136, 0.3903, 0.2628, 0.2329, 0.2155, 0.2062, 0.2020, 0.2011] \]

As a practice, modify the code used in Example 1 to solve this one.

Another possible boundary condition is Robin Boundary Condition. In this case the BC is expressed as an expression,

\[ A_2 y_n + B_2 y'_n = \beta \]  \hspace{1cm} (17)

Using backward difference to express \( y'_n \),

\[ A_2 y_n + B_2 \left( \frac{y_n - y_{n-1}}{h} \right) = \beta \]  \hspace{1cm} (18)

Or,

\[ y_n = \frac{\beta h + B_2 y_{n-1}}{A_2 h + B_2} \]  \hspace{1cm} (19)

--- Example 3 ---

Solve the following 2nd order ODE,

\[ y'' + 5y' + 4y = 1 \]
For \([0,1]\) and boundary conditions: \(y(0)=0\) and \(y(1)-0.5y'(1)=0.5\). Use FD method with \(h=0.1\). For \(i = 2 – 10\) the FD equations are the same as previous example, but for \(i = 11\) the FD equation is calculated (19) with \(A_2 = 1, B_2 = -0.5\) dan \(\beta = 0.5\).

\[i = 11 \rightarrow -1.25 y_{10} + y_{11} = -0.125\]

The formed system of linear equations is

\[
\begin{bmatrix}
-1.96 & 1.25 & 0 & \cdots & 0 \\
0.75 & -1.96 & 1.25 & 0 & \cdots \\
0 & 0.75 & -1.96 & 1.25 & 0 & \cdots \\
0 & \cdots & 0.75 & -1.96 & 1.25 & 0 & \cdots \\
0 & \cdots & \cdots & 0.75 & -1.96 & 1.25 & 0 & \cdots \\
0 & \cdots & \cdots & \cdots & 0.75 & -1.96 & 1.25 & 0 & \cdots \\
0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0.75 & -1.96 & 1.25 \\
0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
y_2 \\
y_3 \\
y_4 \\
y_5 \\
y_6 \\
y_7 \\
y_8 \\
y_9 \\
y_{10} \\
y_{11} \\
\end{bmatrix}
= \begin{bmatrix}
0.0 \\
0.01 \\
0.01 \\
0.01 \\
0.01 \\
0.01 \\
0.01 \\
0.01 \\
-0.125 \\
\end{bmatrix}
\]

The solution is

\[y^T = [0.2198, 0.3446, 0.4165, 0.4543, 0.4704, 0.473, 0.4675, 0.4572, 0.4444, 0.4305]\]

As a practice, modify the code used in Example 1 to solve this one.

\[\begin{align*}
14.2 & \text{ NON-LINEAR FINITE DIFFERENCE (FD) METHOD} \\
\text{The solution of non-linear ODE using FD method is similar to the linear FD} \\
&\text{except for non-linear ODE the solutions is obtained iteratively. In non-linear} \\
&\text{problem} p(x,y) \text{ and} q(x,y) \text{ are function of} x \text{ and} y \text{ or derivatives of} y, \text{ so at each} \\
&\text{iteration} k^{th}, p(x,y) \text{ and} q(x,y) \text{ are calculated using the} y \text{ values at} (k-1)^{th} \text{ iteration. In} \\
&\text{other words we linearize the ODE into,} \\
y^{(k)} + p(x, y^{(k-1)}) y^{(k)} + q(x, y^{(k-1)}) y^{(k)} = f(x)
\end{align*}\]

As in any iterative techniques, we must start from an initial estimate \(y^{(0)}\). The values of the initial estimate can affect the rate of convergence, so we must prudently guess \(y^{(0)}\). The simplest technique is to assume linear \(y(x)\) as shown in the following figure.
Then iterations are done until the left side differ within a certain tolerance from the right side. The above method involves solving linear FD equations again and again. A more explicit technique can also be obtained, we start by writing the ODE again,

\[ y'' = f(x) - p(x, y)y' - q(x, y)y \]  \hspace{1cm} (21)

After expressing \( y' \) and \( y'' \) using (6) and (7) and substituting them into (21),

\[ \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} = f(x_i) - p(x_i, y_i)\frac{y_{i+1} - y_{i-1}}{2h} - q(x_i, y_i)y_i \]  \hspace{1cm} (22)

Rearranging (22), \( y_i \) can be explicitly expressed as

\[ y_i = \frac{y_{i+1} + y_{i-1} - h^2 f(x_i) + h p(x_i, y_i)\frac{y_{i+1} - y_{i-1}}{2h} + h^2 q(x_i, y_i)y_i}{2} \]  \hspace{1cm} (23)

Note that \( y_i \) can be calculated iteratively if all \( y_s \) on the right hand side are the known from the previous calculation or initially assumed at the beginning of the iteration.
\[ y_i^{(k+1)} = \frac{1}{4}[2y_i^{(k)} + 2y_{i+1}^{(k)} - 2h^2 f(x_i) + h p(x_i, y_{i}^{(k)}) (y_{i+1}^{(k)} - y_i^{(k)}) + 2h^2 q(x_i, y_{i}^{(k)}) y_i^{(k)}] \] (24)

Fausett suggested that convergence can be accelerated if we add \( \omega y_i \) to both side of (23),

\[ y_i = \frac{1}{4(1+\omega)}[2y_{i+1}^{(k)} + 4\omega y_i^{(k)} + 2y_{i-1}^{(k)} - 2h^2 f(x_i) + h p(x_i, y_i^{(k)}) (y_{i+1}^{(k)} - y_{i-1}^{(k)}) + 2h^2 q(x_i, y_{i}^{(k)}) y_i^{(k)}] \] (25)

And \( y_i \) is calculated iteratively,

\[ y_i^{(k+1)} = \frac{1}{4(1+\omega)}[2y_{i+1}^{(k)} + 4\omega y_i^{(k)} + 2y_{i-1}^{(k)} - 2h^2 f(x_i) + h p(x_i, y_i^{(k)}) (y_{i+1}^{(k)} - y_{i-1}^{(k)}) + 2h^2 q(x_i, y_{i}^{(k)}) y_i^{(k)}] \] (26)

The iteration process is stopped when a specified convergence criteria has been reached. An acceptable convergence criteria is

\[ \|y_i^{k+1} - y_i^k\|_{\infty} < \varepsilon \] (27)

--- Example 4 ---

Solve the following non-linear ODE,

\[ y'' + (1 + y)y' + (1 + y)y = 1 \]

For \([0,1]\) and boundary conditions \( y(0) = 0 \) and \( y(1) = 1 \). \( p(x,y) = q(x,y) = 1 + y \) and \( f(x) = 1 \). Use the explicit method with \( h = 0.1 \) and initial estimate \( y_i^{(0)} \) for \( i = 2-10 \) which assumed linear \( y^{(0)} = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9] \). The explicit form of \( y \) (Eq. 26) for this example is,

\[ y_i^{(k+1)} = \frac{1}{4(1+\omega)}[2y_{i+1}^{(k)} + 4\omega y_i^{(k)} + 2y_{i-1}^{(k)} - 2h^2 + h(1+y_i^{(k)}) (y_{i+1}^{(k)} - y_i^{(k)}) + 2h^2 (1+y_i^{(k)}) y_i^{(k)}] \]

\[ = \frac{1}{4(1+\omega)}[2y_{i+1}^{(k)} + 4\omega y_i^{(k)} + 2y_{i-1}^{(k)} - 0.02 + 0.1(1+y_i^{(k)}) (y_{i+1}^{(k)} - y_i^{(k)}) + 0.02(1+y_i^{(k)}) y_i^{(k)}] \]

The result is shown in Figure below.
14.3 Tutorial Questions

1. Find the temperature profile inside the tube wall. Hot fluid flows inside the tube such that the inside temperature is 100°C. The differential equation for the temperature distribution is given by

\[
\frac{d^2 T}{dr^2} + \frac{1}{r} \frac{dT}{dr} = 0
\]

The boundary conditions are \( T(100) = 100°C \) and \( T(150) = 0°C \). This is a linear 2nd order linear ODE with Dirichlet boundary conditions, so use linear FD technique with \( h = 10 \).

2. Deflection of prismatic simply supported beam shown below
is given by the following equation.

\[ EI \frac{d^2y}{dx^2} = -\frac{qLx}{2} + \frac{qx^2}{2} \]

Where \( V \) is uniformly distributed load, \( L \) is the length of the beam, \( I \) is the moment of inertia of the beam cross section (\( I = \frac{wh^3}{12} \), \( w \) is width and \( h \) is height of the beam) and \( E \) is the modulus of elasticity of the beam. For the case in hand \( E = 10 \text{GPa}, \)
\( L = 2.0 \text{m}, w = 5.0 \text{cm}, h = 10.0 \text{cm}, V = 1500 \text{N/m} \) and \( I = 4.166 \times 10^{-6} \text{m}^4 \), Find \( y(x) \).

It is clear from the problem that the boundary conditions are Dirichlet i.e., \( y(0) = y(2) = 0. \). Rewrite the governing equation

\[ y^* = \frac{Vx}{2EI} (x - L) = 0.018x(x - 2) \]

From the above, \( p(x) = 0, q(x) = 0 \) and \( f(x) = 0.018 x (x - 2) \). Divide the beam into 10 interval giving \( h = 0.2 \) and generate plot of the beam deflection.

3. Use linear shooting method and linear Finite Difference Method to find the solution of
<table>
<thead>
<tr>
<th>ODE</th>
<th>Range</th>
<th>Boundary conditions</th>
<th>Analytical solution</th>
<th>h</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y'' = 4(y - x)$</td>
<td>[0,1]</td>
<td>$y(0) = 0$ \ $y(1) = 2$</td>
<td>$y(x) = \frac{e^2}{e^4 - 1}(e^{2x} - e^{-2x}) + x$</td>
<td>0.25</td>
</tr>
</tbody>
</table>

4. Use FD method (*Neumann Boundary conditions*) to find the solution of

<table>
<thead>
<tr>
<th>ODE</th>
<th>Range</th>
<th>Boundary conditions</th>
<th>h</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y'' = -5y' - 4y + 1$</td>
<td>[0,1]</td>
<td>$y(0) = 1$ \ $y'(1) = 0$</td>
<td>0.25</td>
</tr>
</tbody>
</table>

5. Use FD method (*Robin Boundary conditions*) to find the solution of

<table>
<thead>
<tr>
<th>ODE</th>
<th>Range</th>
<th>Boundary conditions</th>
<th>h</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y'' = -4y' - 6.25y + e^x$</td>
<td>[0,1]</td>
<td>$y(0) = 1$ \ $y(1) - 0.5y'(1) = 0.5$</td>
<td>0.25</td>
</tr>
</tbody>
</table>

6. Use non-linear FD method to find the solution of

<table>
<thead>
<tr>
<th>ODE</th>
<th>Range</th>
<th>Boundary conditions</th>
<th>Analytical solution</th>
<th>h</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y'' = y^3 - y y'$</td>
<td>[1,2]</td>
<td>$y(1) = \frac{1}{2}$ \ $y(2) = \frac{1}{3}$</td>
<td>$y(x) = \frac{1}{x+1}$</td>
<td>0.25</td>
</tr>
</tbody>
</table>

Comment on the use of different $\omega$ say from 0.5 – 1.5. Adopt the following convergence criteria

$$\left\| y_i^{k+1} - y_i^k \right\|_{\infty} < 10^{-5}$$