FINITE DIFFERENCE METHOD

Rachneet Kaur, M.Sc. Mathematics, Indian Institute of Technology, Hauz Khas, New Delhi – 110016

purplebluemix@gmail.com, **9868894365**

Abstract:

Finite Difference Method (FDM) is a numerical method for approximating the solution of Differential Equations by using Finite Difference formulae to approximate the derivatives. In this paper, I give all the theoretical as well as computational aspects of FDM and the concepts are explained with a wide no. of examples with different types of boundary conditions in each of them. I have also discussed the error analysis for FDMs along with MATLAB codes and the output graphs. Computationally, the error bounds are verified for each of the examples after computing them theoretically, verifying the correctness of the approach.

Keywords:

Discretization, Dirichlet and Neumann Boundary conditions, positive definite, sparse, tridiagonal, homogeneous and non homogeneous boundary conditions.

INTRODUCTION:

Finite Difference Method (FDM) is a numerical method for approximating the solution of Differential Equations by using Finite Difference formulae to approximate the derivatives. FDM uses Taylor series Expansion to approximate solutions of differential equations. We Will Discretize the Domain into smaller Subdomains by Inserting Nodes and Compute the Values of Unknown Function 'u' at these Nodal Points by Replacing the Derivatives by Finite Difference Formulae of Required Order. By Using More No. Of Refinements or Higher Order FD Formulae, More Accurate Solution Is Obtained.

WHY DO WE USE FDM?

(1)The Finite Difference Scheme Is One of the Simplest Forms of Discretization, It is Easy to Understand and implement.
(2) Easy To Code in MATLAB.

COMPARING FDM WITH FEM:

FEM coincides with FDM except that average of f over (x_{j-1},x_{j+1}) is used in FEM unlike $f(x_j)$ in FDM.
FDM is faster than FEM in lower order accuracy problems.
FDM can be applied to solve non – linear problem directly but FEM can't.
FDM: DE ⟹ DISCRITIZATION ⟹ FINITE DIFFERENCE FORMULA REPLACE DERIVATIVES ⟹ MATRIX FORMULATION ⟹ SOLUTION⟹ERROR ANALYSIS.
FEM: DE⟹WEAK FORMULATION⟹FINITE ELEMENT FORMULATION (DISCRITIZATION) ⟹BASIS FUNCTIONS⟹MATRIX FORMULATION⟹SOLUTION⟹ERROR ANALYSIS

WHY DID WE SWITCH TO FEM FROM FDM?

FDM uses a square network of lines to construct the discretization of ODE/PDE, hence it is not very efficient while handling complex geometries in multiple dimensions and hence in those cases FEM is used.

---

1 – DIMENSIONAL FINITE DIFFERENCE METHOD:

- **BVP WITH DRICHLET BOUNDARY CONDITIONS:**

CONSIDER A SECOND ORDER ODE: $-u'' = f(x)\ on\ I = (0,1)$

$$u(0) = \alpha,\ u(1) = \beta$$

(DRICHLET BOUNDARY CONDITIONS)

Here, f(x) is given and u(x) is unknown function to be computed.
We first discretise our domain into smaller subdomains.
Divide the interval (0, 1) into (m+1) subintervals by inserting (m+2) nodes in (0, 1).

Mesh = $\{x_0 = 0 < x_1 < x_2 < \ldots\ldots\ldots\ldots < x_m < x_{m+1} = 1\}$, Mesh width = $h = \frac{1}{m+1}$, $x_j = \frac{j}{m+1}$.

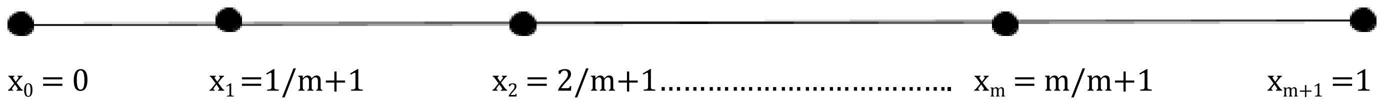
More the no. of refinements, more is the accuracy.

Now, we need to compute the values of u at these nodal points u (x_j) = U_j.

From boundary conditions, we have $U_0 = \alpha$, $U_{m+1} = \beta$.

So, we need to compute $U_1, U_2, \ldots\ldots\ldots, U_m$.

We replace u'' by II order Central Difference formula (three point centred scheme for second derivative):



The above system of m equations in m variables can be written in matrix form as:

$$\frac{1}{h^2}\begin{bmatrix} 2 & -1 & 0 & 0 & \ldots & 0 \\ -1 & 2 & -1 & 0 & \ldots & 0 \\ . & -1 & 2 & & & \\ & & & -1 & & \\ 0 & & 0 & \ldots & -1 & 2 \end{bmatrix} \cdot \begin{bmatrix} U_1 \\ U_2 \\ \vdots \\ U_m \end{bmatrix} = \begin{bmatrix} f(x_1) + \alpha/h^2 \\ f(x_2) \\ \vdots \\ f(x_m) + \beta/h^2 \end{bmatrix}$$

Reducing to **AU = F** where U needs to be computed.

Note that **A is a SPARSE, TRIDIAGONAL, SYMMETRIC AND POSITIVE DEFINITE MATRIX.**

CLEARLY, ONE CAN OBSERVE THAT A IS SPARSE, TRIDIAGONAL AND SYMMETRIC.

<u>A IS POSITIVE DEFINITE AS:</u>

FOR ANY NON ZERO VECTOR v = ($v_1, \ldots\ldots\ldots, v_N$),

CONSIDER $h^2 v^T A v = v_1^2 + (v_2 - v_1)^2 + \ldots\ldots\ldots + (v_N - v_{N-1})^2 + v_N^2 > 0$

Hence, $v^T A v > 0$, concluding that A is a positive definite matrix.

A, being positive definite and symmetric, is Invertible. Hence U = A^{-1}F is the required <u>Unique</u> solution.

<u>EXAMPLE (1)</u>: SOLVE THE FOLLOWING BVP USING FINITE DIFFERENCE METHOD:

$$u'' = e^{x^2} \; on \; I = (0, 1)$$

$$\mathbf{u(0) = 0, u(1) = 0}$$

<u>SOLUTION:</u> We divide I = (0, 1) into 4 equal subintervals:

h = 0.25

Mesh = {0, 0.25, 0.5, 0.75, 1}

$U_0 = 0$, $U_4 = 0$ (From Boundary conditions)

And U_1, U_2, U_3 is computed by using U = A^{-1}.F where A = $\begin{bmatrix} -2 & 1 & 0 \end{bmatrix}$, F = $\begin{bmatrix} e^{0.0625} \end{bmatrix}$

$$16 \cdot \begin{matrix} 1 & -2 & 1 \\ 0 & 1 & -2 \end{matrix} \qquad \begin{matrix} e^{0.25} \\ e^{0.5625} \end{matrix}$$

Calculating, we get U = $\begin{bmatrix} 0 \\ -0.117 \\ -0.168 \\ -0.139 \\ 0 \end{bmatrix}$

- **Consider the problem:** $-u'' + c(x)u = f(x) \text{ on } I = (0,1)$
  $u(0) = \alpha,\ u(1) = \beta$

In this case,

Replacing $U_j'' = [U_{j-1} - 2U_j + U_{j+1}] / h^2$

The Discretization is:

$$\left.[-U_{j-1} + 2U_j - U_{j+1}] / h^2 + c(x_j)U_j = f(x_j) \quad \text{where} \quad j = 1,2,3\ldots\ldots\ldots,m \right\} \quad \text{---------------------(B)}$$

From boundary conditions, we have $U_0 = \alpha$, $U_{m+1} = \beta$.

So, we need to compute $U_1, U_2, \ldots\ldots\ldots, U_m$.

The above system of m equations in m variables can be written in matrix form as:

$$\begin{bmatrix} c(x_1) & 0 & \ldots & 0 \\ 0 & c(x_2) & \ldots & 0 \\ \vdots & & & \vdots \\ 0 & \ldots & 0 & c(x_m) \end{bmatrix} + 1/h^2 \begin{bmatrix} 2 & -1 & 0 & 0 & \ldots & 0 \\ -1 & 2 & -1 & 0 & \ldots & 0 \\ \vdots & -1 & 2 & & & \\ & & & & -1 & \\ 0 & 0 & \ldots & & -1 & 2 \end{bmatrix} \cdot \begin{bmatrix} U_1 \\ U_2 \\ \vdots \\ U_m \end{bmatrix} = \begin{bmatrix} f(x_1) + \alpha/h^2 \\ f(x_2) \\ \vdots \\ f(x_m) + \beta/h^2 \end{bmatrix}$$

Reducing to (B +A) U = F, where U needs to be computed.

Let us say C = B+A,

Clearly, C is SPARSE, TRIDIAGONAL AND SYMMETRIC MATRIX.

Also, C is Positive definite as:

For any non-zero vector v = ($v_1, \ldots\ldots\ldots, v_N$),

CONSIDER $h^2v^TCv = h^2v^TAv + h^2v^TBv$

$$= h^2 (v^TA\,v + \sum c(xi)vi^2\ )$$

$$\geq h^2v^TAv$$

$$= v_1^2 + (v_2 - v_1)^2 + \ldots\ldots\ldots + (v_N - v_{N-1})^2 + v_N^2$$

$$> 0$$

Hence, $v^TCv > 0$, concluding that C is a positive definite matrix.

C, being positive definite and symmetric, is Invertible

Hence U = $C^{-1}F$ is the required <u>Unique</u> solution.

<u>EXAMPLE (2)</u>: SOLVE THE FOLLOWING BVP USING FINITE DIFFERENCE METHOD:

$$-u'' + u = 1 \text{ on } I = (0,1)$$

$$\mathbf{u(0) = 1,\ u(1) = 3}$$

<u>SOLUTION:</u> We divide I = (0, 1) into 4 equal subintervals:

h = 0.25

Mesh = {0, 0.25, 0.5, 0.75, 1}

$U_0 = 1$, $U_4 = 3$(From Boundary conditions)

And U_1, U_2, U_3 is computed by using U = $C^{-1}.F$ where A = $16 . \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$, F = $\begin{bmatrix} 17 \\ 1 \\ 49 \end{bmatrix}$

B = $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

$\Longrightarrow$ C = $\begin{bmatrix} 33 & -16 & 0 \\ -16 & 33 & -16 \\ 0 & -16 & 33 \end{bmatrix}$

Calculating, we get U = $\begin{bmatrix} 1 \\ 1.43 \\ 1.88 \\ 2.33 \\ 3 \end{bmatrix}$

• <u>BVP WITH NEUMANN BOUNDARY CONDITIONS:</u>

CONSIDER A SECOND ORDER ODE: $-u'' + c(x)u = f(x) \text{ on } I = (0,1)$

$$u'(0) = \alpha,\ u'(1) = \beta$$

(NEUMANN BOUNDARY CONDITIONS)

We first discretise our domain into smaller subdomains.
Mesh = $\{x_0 = 0 < x_1 < x_2 < \dots\dots < x_m < x_{m+1} = 1\}$,
Mesh width = h= $\frac{1}{m+1}$, $x_j = \frac{j}{m+1}$.

Now, we need to compute the values of u at these nodal points u (x_j) = U_j.

So, we need to compute $U_0, U_1, U_2, \ldots\ldots\ldots, U_m, U_{m+1}$.

We replace u'' by II order Central Difference formula (three point centred scheme for second derivative):

$$\mathbf{u''(x_j) = U_j'' = [\, U_{j-1} - 2U_j + U_{j+1}\,] / h^2}$$

We get,

$$\mathbf{[\,-U_{j-1} + 2U_j - U_{j+1}\,] / h^2 + c(x_j)U_j = f(x_j) \quad \text{where} \quad j = 0,1,2,3\ldots\ldots\ldots,m,m+1} \qquad \text{-----------------------------}(1)$$

Where we replace u'(0) by I order forward difference formula and u'(1) by I order backward difference formula:

$$U'(0) = U_1 - U_0 / h = \boldsymbol{\alpha}$$

$$\mathbf{\Longrightarrow U_0 = U_1 - h\alpha} \qquad \text{---------------------------------------}(2)$$

$$U'(1) = U_{m+1} - U_m / h = \beta$$

$$\mathbf{\Longrightarrow U_{m+1} = U_m + h\beta} \qquad \text{--}(3)$$

Hence, the Discretization for the BVP with Neumann Boundary conditions is:

$$\left.\begin{aligned}
&\mathbf{[\,-U_{j-1} + 2U_j - U_{j+1}\,] / h^2 + c(x_j)U_j = f(x_j) \quad \text{where} \quad j = 0, 1,2,3\ldots\ldots\ldots,m,m+1}\\
&\mathbf{U_0 = U_1 - h\alpha}\\
&\mathbf{U_{m+1} = U_m + h\beta}
\end{aligned}\right\} \qquad \text{-------------}\ \mathbf{(C)}$$

The above system of m+2 equations in m+2 variables can be written in matrix form as:

$$\begin{bmatrix} 0 \ldots\ldots & 0\ldots\ldots\ldots 0 & \\ 0 & c(x_1)\ 0\ldots\ldots 0 & \\ . & & . \\ 0 & c(x_m) & 0 \\ 0\ldots\ldots\ldots\ldots & 0 \ldots\ldots\ldots & 0 \end{bmatrix} + 1/h^2 \begin{bmatrix} 1 & -1 & 0 & 0 \ldots\ldots\ldots\ldots\ldots\ldots & 0 & \\ -1 & 2 & -1 & 0\ldots\ldots\ldots\ldots\ldots\ldots\ldots & 0 & \\ . & -1 & 2 & & . & \\ & & & 2 & -1 & \\ 0 & & 0\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots & & -1 & 1 \end{bmatrix} \begin{bmatrix} U_0 \\ U_1 \\ \\ \\ U_{m+1} \end{bmatrix} = \begin{bmatrix} -\alpha/h \\ f(x_1) \\ \\ f(x_m) \\ \beta/h \end{bmatrix}$$

Reducing to (B +A) U = F, where U needs to be computed.

In this case, **ERROR = O (h),**

But we can improve the accuracy in the Finite Difference solution if we approximate u' (0) and u' (1) by Central Difference Approximation formulae:

$$u'(0) = u_1 - u_{-1} / 2h = \boldsymbol{\alpha}$$

$$\mathbf{\Longrightarrow u_{-1} = u_1 - 2h\alpha} \qquad \text{--------------------------------------}\ (2)$$

$$u'(1) = u_{N+2} - u_N / 2h = \beta$$

$$\mathbf{\Longrightarrow u_{N+2} = u_N + 2h\beta} \qquad \text{------------------------------------}(3)$$

In the discretization (1), when we substitute (2) (Put j = 0), we get:

$$\mathbf{[-2U_1 + 2U_0] / h^2 + c(x_0)U_0 = f(x_0) - 2\alpha/h} \qquad \text{---------------------------------------}\ (4)$$

And when we substitute (3) (Put j=N+1), we get:

$$\mathbf{[-2U_N + 2U_{N+1}] / h^2 + c(x_{N+1})U_{N+1} = f(x_{N+1}) + 2\beta/h} \qquad \text{---------------------------------------}\ (5)$$

Hence, the Discretization is:

$$\left.\begin{aligned}
&\mathbf{[\,-U_{j-1} + 2U_j - U_{j+1}\,] / h^2 + c(x_j)U_j = f(x_j) \quad \text{where} \quad j = 0,1,2,3\ldots\ldots\ldots,m,m+1}\\
&\mathbf{[-2U_1 + 2U_0] / h^2 + c(x_0)U_0 = f(x_0) - 2\alpha/h}
\end{aligned}\right\} \qquad \text{-----------}\mathbf{(D)}$$

$[-2U_N + 2U_{N+1}] / h^2 + c(x_{N+1})U_{N+1} = f(x_{N+1}) + 2\beta/h$

The above system of m+2 equations in m+2 variables can be written in matrix form as:

$$\begin{bmatrix} c(x_0) \;.....0.......0 \\ 0 \quad c(x_1)\; 0......0 \\ . \qquad\qquad . \\ 0 \qquad c(x_m) \quad 0 \\ 0......0\; ....c(x_{m+1}) \end{bmatrix} + 1/h^2 \begin{bmatrix} 2 & -2 & 0 & 0 \;.................... & 0 \\ -1 & 2 & -1 & 0 \;.................... & 0 \\ . & -1 & 2 & & . \\ & & -1 & 2 & -1 \\ 0 & & 0 \;........................ & -2 & 2 \end{bmatrix} \begin{bmatrix} U_0 \\ U_1 \\ \\ \\ \end{bmatrix} = \begin{bmatrix} f(x_0) - 2\alpha/h \\ f(x_1) \\ \\ f(x_m) \\ U_{m+1} \end{bmatrix}$$

$f(x_{m+1}) + 2\beta/h$

Reducing to (B +A) U = F, where U needs to be computed.

In this case, **ERROR = O (h^2)**

REMARK:-

IF IN THE ABOVE PROBLEM C(x) = 0, THEN IT IS NOT - WELL POSED PROBLEM.

Consider the case when c = 0, in that case A =

$$\begin{bmatrix} 2 & -2 & 0 \;.................... & & & 0 \\ -1 & 2 & -1 & 0 & ....0 & \\ 0 & -1 & 2 & -1 & 0......0 & \\ . & & & & . & \\ . & & & & . & \\ 0 \;................0 & & & 0 & -2 & 2 \end{bmatrix}$$

Which is SINGULAR MATRIX, hence it is NOT INVERTIBLE.

It reflects that the problem we are attempting to solve is not Well Posed, and the differential equation will have either No Solution or Infinitely Many Solutions.

EXAMPLE (3): SOLVE THE FOLLOWING BVP USING FINITE DIFFERENCE METHOD:

$$-u'' + u = 1 \; on \; I = (0, 1)$$

$$u'(0) = 1, \; u'(1) = 3$$

SOLUTION: We divide I = (0, 1) into 2 equal subintervals:

h = 0.5

Mesh = {0, 0.5, 1}

And U_0, U_1, U_2 is computed by using $U = C^{-1}.F$ where A = $4\begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}$ F= $\begin{bmatrix} -2 \\ 1 \\ 6 \end{bmatrix}$

B = $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ \end{bmatrix}$

0 0 0

$$\Longrightarrow \quad C = \begin{bmatrix} 4 & -4 & 0 \\ -4 & 9 & -4 \\ 0 & -4 & 4 \end{bmatrix}$$

Calculating, we get $U = \begin{bmatrix} 4.49 \\ 4.99 \\ 6.49 \end{bmatrix}$

• BVP WITH MIXED BOUNDARY CONDITIONS:

CONSIDER A SECOND ORDER ODE: $-u'' = f(x) \ on \ I = (0,1)$

$$u'(0) = \alpha, \ u(1) = \beta$$

(MIXED BOUNDARY CONDITIONS)

So, we need to compute $U_0, U_1, U_2, \ldots\ldots, U_m$.

From boundary conditions, $U_{m+1} = \beta$

We replace u'' by II order Central Difference formula:

$$u''(x_j) = U_j'' = [\, U_{j-1} - 2U_j + U_{j+1} \,] / \ h^2$$

We get,

$$[\, -U_{j-1} + 2U_j - U_{j+1} \,] / \ h^2 = f(x_j) \quad \text{where} \quad j = 0,1,2,3\ldots\ldots,m \quad \text{-----------------------------(1)}$$

Where we replace u'(0) by I order forward difference formula:

$$U'(0) = U_1 - U_0 / h = \alpha$$

$$\Longrightarrow U_0 = U_1 - h\alpha \quad \text{---------------------------------------(2)}$$

Hence, the Discretization for the BVP with Mixed Boundary conditions is:

$$\left.\begin{aligned} &[\, -U_{j-1} + 2U_j - U_{j+1} \,] / \ h^2 = f(x_j) \quad \text{where} \quad j = 0,1,2,3\ldots\ldots,m \\ &U_0 = U_1 - h\alpha \end{aligned}\right\} \quad \text{-------------------------- (E)}$$

The above system of m+1 equations in m+1 variables can be written in matrix form as:

$$\frac{1}{h^2}\begin{bmatrix} 1 & -1 & 0 & 0 \ldots\ldots & 0 \\ -1 & 2 & -1 & 0 \ldots\ldots & 0 \\ . & -1 & 2 & & . \\ . & & & & -1 \\ 0 & 0 \ldots\ldots & & -1 & 2 \end{bmatrix} \begin{bmatrix} U_0 \\ U_1 \\ \vdots \\ U_m \end{bmatrix} = \begin{bmatrix} -\alpha/h \\ f(x_1) \\ \vdots \\ f(x_m) + \beta/h^2 \end{bmatrix}$$

Reducing to (B +A) U = F, where U needs to be computed.

Error = O(h),

But we can improve the accuracy in the Finite Difference solution if we approximate u' (0) by Central Difference Approximation formulae:

$u'(0) = u_1 - u_{-1} / 2h = \boldsymbol{\alpha}$

$\Rightarrow$ **$u_{-1} = u_1 - 2h\alpha$** -------------------------------------- **(2)**

In the discretization (1), when we substitute (2) (Put j = 0), we get:

$[-2U_1 + 2U_0] / h^2 = f(x_0) - 2\alpha/h$ -------------------------------------- **(4)**

Hence, the Discretization is:

$$\left.\begin{aligned} &[-U_{j-1} + 2U_j - U_{j+1}] / h^2 = f(x_j) \quad \text{where} \quad j = 0,1,2,3\ldots\ldots\ldots,m,m+1 \\ &[-2U_1 + 2U_0] / h^2 = f(x_0) - 2\alpha/h \end{aligned}\right\} \text{-----------(F)}$$

The above system of m+1 equations in m+1 variables can be written in matrix form as:

$$\frac{1}{h^2}\begin{bmatrix} 2 & -2 & 0 & 0\ldots\ldots\ldots\ldots\ldots\ldots 0 \\ -1 & 2 & -1 & 0\ldots\ldots\ldots\ldots\ldots\ldots 0 \\ & -1 & 2 & \\ & & -1 & 2 \end{bmatrix} . \begin{bmatrix} U_0 \\ U_1 \\ \\ U_m \end{bmatrix} = \begin{bmatrix} f(x_0) - 2\alpha/h \\ f(x_1) \\ f(x_m) + \beta/h^2 \end{bmatrix}$$

Reducing to A U = F, where U needs to be computed.

Reducing to A U = F, where U needs to be computed.

In this case, **ERROR = O (h^2)**

- **Consider the problem:** $-u'' + c(x)u = f(x) \ on \ I = (0,1)$

  $u'(0) = \alpha$, $u(1) = \beta$

In this case,

Replacing **$U_j'' = [U_{j-1} - 2U_j + U_{j+1}] / h^2$**

The Discretization is:

$[-U_{j-1} + 2U_j - U_{j+1}] / h^2 + c(x_j)U_j = f(x_j)$ where $j = 0,1,2,3\ldots\ldots\ldots,m$ ----------------------------**(1)**

From boundary conditions, we have $U_{m+1} = \beta$.

So, we need to compute U_0 ,U1, U_2,………., U_m.

Where we replace u'(0) by I order forward difference formula:

$U'(0) = U_1 - U_0 / h = \boldsymbol{\alpha}$

$\Rightarrow U_0 = U_1 - h\alpha$ --(2)

Hence, the Discretization for the BVP with Mixed Boundary conditions is:

$$\left.\begin{array}{l} [-U_{j-1} + 2U_j - U_{j+1}] / h^2 + c(x_j)U_j = f(x_j) \quad \text{where} \quad j = 0,1,2,3 \ldots\ldots\ldots, m \\ U_0 = U_1 - h\alpha \end{array}\right\} \text{---------------- (G)}$$

The above system of m+1 equations in m+1 variables can be written in matrix form as:

$$\left[\begin{array}{ccc} 0 \ldots\ldots & 0 \ldots\ldots\ldots 0 & \\ 0 & c(x_1)\ 0 \ldots\ldots 0 & \\ . & & . \\ 0 & & c(x_m) \end{array}\right] + 1/h^2 \left[\begin{array}{cccccc} 1 & -1 & 0 & 0 \ldots\ldots\ldots\ldots\ldots\ldots & & 0 \\ -1 & 2 & -1 & 0 \ldots\ldots\ldots\ldots\ldots\ldots\ldots & & 0 \\ . & -1 & 2 & & & . \\ 0 & & & & -1 & 2 \end{array}\right] \left[\begin{array}{c} U_0 \\ U_1 \\ \\ U_m \end{array}\right] = \left[\begin{array}{c} -\alpha/h \\ f(x_1) \\ \\ f(x_m) + \beta/h^2 \end{array}\right]$$

Reducing to (B +A) U = F, where U needs to be computed.

In this case, **ERROR = O (h),**

But we can improve the accuracy in the Finite Difference solution if we approximate u' (0) by Central Difference Approximation formulae:

$u'(0) = u_1 - u_{-1} / 2h = \alpha$

$\Rightarrow u_{-1} = u_1 - 2h\alpha$ -- (2)

In the discretization (1), when we substitute (2) (Put j = 0), we get:

$[-2U_1 + 2U_0] / h^2 + c(x_0)U_0 = f(x_0) - 2\alpha/h$ -- (4)

Hence, the Discretization is:

$$\left.\begin{array}{l} [-U_{j-1} + 2U_j - U_{j+1}] / h^2 + c(x_j)U_j = f(x_j) \quad \text{where} \quad j = 0,1,2,3 \ldots\ldots\ldots, m, m+1 \\ [-2U_1 + 2U_0] / h^2 + c(x_0)U_0 = f(x_0) - 2\alpha/h \end{array}\right\} \text{-----------(H)}$$

The above system of m+1 equations in m+1 variables can be written in matrix form as:

$$\left[\begin{array}{ccc} c(x_0) \ldots\ldots & 0 \ldots\ldots\ldots 0 & \\ 0 & c(x_1)\ 0 \ldots\ldots 0 & \\ . & & . \\ 0 & & c\,(x_m) \end{array}\right] + 1/h^2 \left[\begin{array}{cccccc} 2 & -2 & 0 & 0 \ldots\ldots\ldots\ldots\ldots\ldots & & 0 \\ -1 & 2 & -1 & 0 \ldots\ldots\ldots\ldots\ldots\ldots\ldots & & 0 \\ . & -1 & 2 & & & . \\ & & & & -1 & 2 \end{array}\right] \left[\begin{array}{c} U_0 \\ U_1 \\ \\ U_m \end{array}\right] = \left[\begin{array}{c} f(x_0) - 2\alpha/h \\ f(x_1) \\ f(x_m) + \beta/h^2 \end{array}\right]$$

Reducing to (B +A) U = F, where U needs to be computed.

In this case, **ERROR = O (h^2)**

EXAMPLE (4): SOLVE THE FOLLOWING BVP USING FINITE DIFFERENCE METHOD:

$$-u'' = 1 \text{ on } I = (0, 1)$$

$$u(0) = 0,\ u'(1) = 0$$

SOLUTION: We divide I = (0, 1) into 3 equal subintervals:

h = 0.33

Mesh = {0, 1/3,2/3, 1}

$U_0 = 0$

And U_1, U_2, U_3 is computed by using $U = A^{-1}.F$ where A =

$$A = 9\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \qquad F = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

we get U =

$$U = \begin{bmatrix} 0.22 \\ 0.33 \\ 0.33 \end{bmatrix}$$

CODING 1-D FINITE DIFFERENCE METHOD IN MATLAB:

CONSIDER EXAMPLE (1): $u'' = e^{x^2} \; on \; I = (0,1)$

u (0) = 0, u(1) = 0

(**HOMOGENEOUS DRICHLET BOUNDARY CONDITIONS)**

MATLAB CODE:

```
function FDM_1D_EXAMPLE1
h=1/6; %STEP SIZE
N=1/h; %NO. OF INTERVALS
%HOMOGENEOUS DRICHLET BOUNDARY CONDITIONS
alpha=0;
beta=0;
mesh = 0:h:1
f=zeros(N-1,1);
a = zeros(N-1,N-1);
uh = zeros(N+1,1);
f(1)= exp((mesh(2))^2) - (alpha/(h^2));
f(N-1,1)=exp((mesh(end-1))^2) - (beta/(h^2));
for i=2:N-2
    f(i,1)=exp((mesh(i+1))^2);
end
for i=1:N-1
    a(i,i)=-2;
    if i~=N-1
    a(i,i+1)=1;
    a(i+1,i)=1;
    end
end
a = a/(h^2)
 u=a\f;
 uh = [ alpha; u; beta ]
 plot(mesh,uh)
end
```

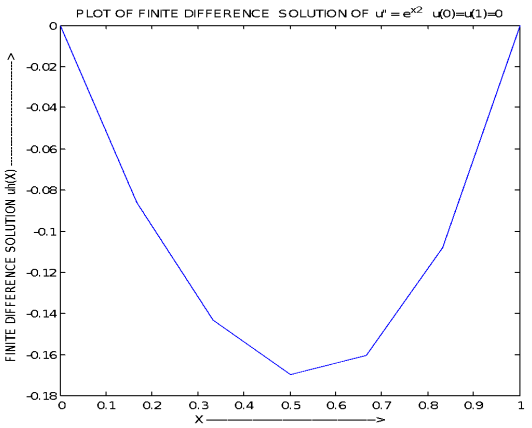
OUTPUT:

a =

-72	36	0	0	0
36	-72	36	0	0
0	36	-72	36	0
0	0	36	-72	36
0	0	0	36	-72

$$u_h = \begin{bmatrix} -0.086040887124182 \\ -0.143521574875095 \\ -0.169960066272066 \\ -0.160731184983267 \\ -0.108179428760947 \end{bmatrix}$$

PLOTTING THE FINITE DIFFEERENCE SOLUTION:



CONSIDER EXAMPLE (2): $u'' = 1 \; on \; I = (0, 1)$

u (0) = 1, u(1) = 3

(**NON -HOMOGENEOUS DRICHLET BOUNDARY CONDITIONS)**

```
function [uh] = FDM_1D_EXAMPLE2
h=1/3; %STEP SIZE
N=1/h; %NO. OF INTERVALS
%NON - HOMOGENEOUS DRICHLET BOUNDARY CONDITIONS
alpha=1;
beta=3;
mesh = 0:h:1
f=zeros(N-1,1);
a = zeros(N-1,N-1);
uh = zeros(N+1,1);
f(1)= 1 - (alpha/(h^2));
f(N-1)= 1- (beta/(h^2));
for i=2:N-2
    f(i)= 1;
end
for i=1:N-1
    a(i,i)=-2;
    if i~=N-1
    a(i,i+1)=1;
    a(i+1,i)=1;
    end
end
a = a/(h^2);
 u=a\f;
 uh = [ alpha; u; beta ]
 plot(mesh,uh,'r')
 hold on
 ezplot('((x^2/2)+(3/2*x)+1)',[0,1])
 hold off
```

uh =

1.000000000000000

1.555555555555556

2.222222222222222

3.000000000000000

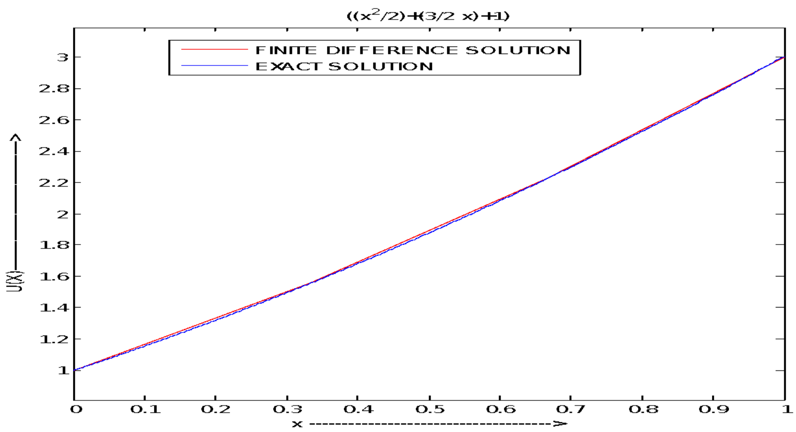


FIGURE: PLOTTING THE FINITE DIFFEERENCE SOLUTION WITH THE EXACT SOLUTION

CONSIDER EXAMPLE (3): $-u'' = 1 \; on \; I = (0,1)$

u (0) = 0, u'(1) = 0

(**HOMOGENEOUS MIXED BOUNDARY CONDITIONS)**

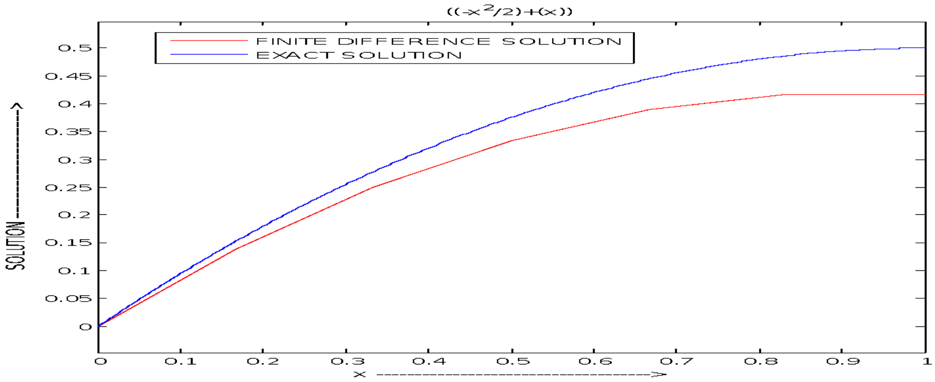
MATLAB CODE:

OUTPUT:

$$a = \begin{bmatrix} 72 & -36 & 0 & 0 & 0 & 0 \\ -36 & 72 & -36 & 0 & 0 & 0 \\ 0 & -36 & 72 & -36 & 0 & 0 \\ 0 & 0 & -36 & 72 & -36 & 0 \\ 0 & 0 & 0 & -36 & 72 & -36 \\ 0 & 0 & 0 & 0 & -36 & 36 \end{bmatrix}$$

$$uh = \begin{bmatrix} 0 \\ 0.138888888888889 \\ 0.250000000000000 \\ 0.333333333333333 \\ 0.388888888888888 \\ 0.416666666666666 \\ 0.416666666666666 \end{bmatrix}$$

FIGURE: <u>PLOTTING THE FINITE DIFFEERENCE SOLUTION WITH THE EXACT SOLUTION</u>



CONSIDER EXAMPLE (4): $-u'' + u = 1 \text{ on } I = (0,1)$

u (0) = 1, u(1) = 3

<u>MATLAB CODE:</u>

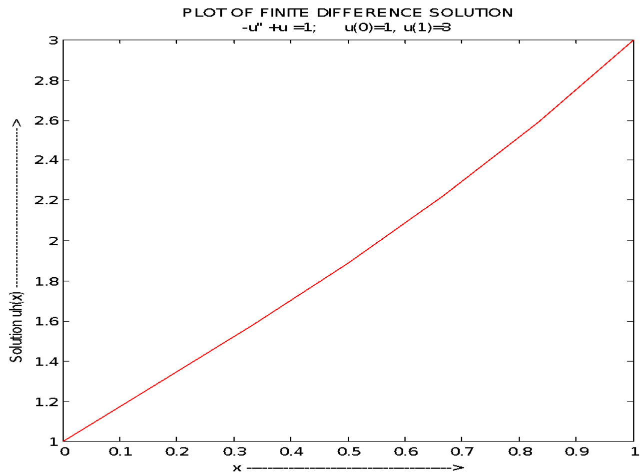
<u>OUTPUT:</u>

$$a = \begin{matrix} 73 & -36 & 0 & 0 & 0 \\ -36 & 73 & -36 & 0 & 0 \\ 0 & -36 & 73 & -36 & 0 \end{matrix}$$

$$\begin{matrix} 0 & 0 & -36 & 73 & -36 \\ 0 & 0 & 0 & -36 & 73 \end{matrix}$$

$$uh = \begin{bmatrix} 1.000000000000000 \\ 1.285054194918955 \\ 1.578026561918992 \\ 1.887055222305668 \\ 2.220724305534167 \\ 2.588302397249727 \\ 3.000000000000000 \end{bmatrix}$$

FIGURE: <u>PLOTTING THE FINITE DIFFEERENCE SOLUTION:</u>



CONSIDER EXAMPLE (5): $-u'' + u = 1 \; on \; I = (0,1)$

u' (0) = 1, u' (1) = 3

(NON - HOMOGENEOUS NEUMANN BOUNDARY CONDITIONS)

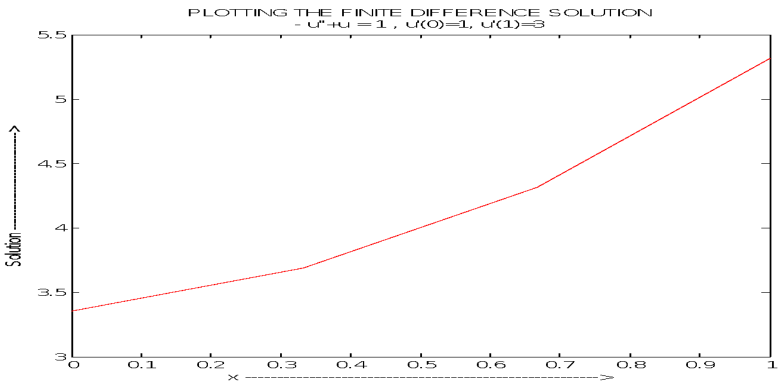
<u>MATLAB CODE:</u>

OUTPUT:

$$a = \begin{bmatrix} 9 & -9 & 0 & 0 \\ -9 & 19 & -9 & 0 \\ 0 & -9 & 19 & -9 \\ 0 & 0 & -9 & 9 \end{bmatrix}$$

$$uh = \begin{bmatrix} 3.350877192982452 \\ 3.684210526315785 \\ 4.315789473684206 \\ 5.315789473684206 \end{bmatrix}$$

FIGURE: PLOTTING THE FINITE DIFFEERENCE SOLUTION:



CONSIDER EXAMPLE (6): $u'' = e^{x} \; on \; I = (0, 1)$

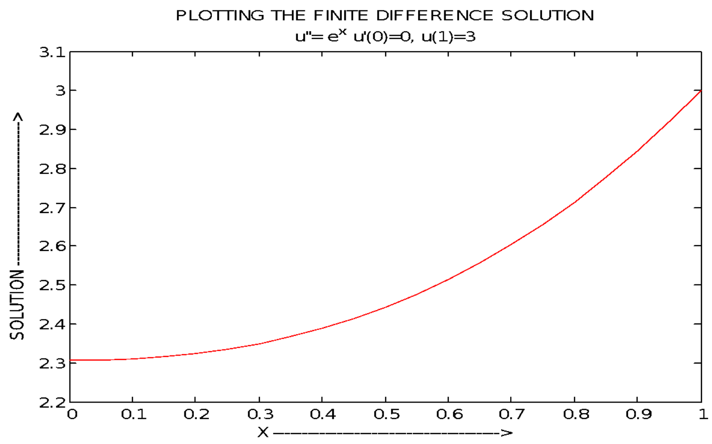
u' (0) = 0, u (1) = 3

(NON – HOMOGENEOUS MIXED BOUNDARY CONDITIONS)

MATLAB CODE:

```
function [uh] = FDM_1D_EXAMPLE3
h=1/20; %STEP SIZE
N=1/h; %NO. OF INTERVALS
%MIXED BOUNDARY CONDITIONS
alpha=0;
beta=3;
mesh = 0:h:1
f=zeros(N,1);
a = zeros(N,N);
uh = zeros(N+1,1);
f(1)= (alpha/(h));
f(N)= exp(mesh(end-1))-(beta/(h^2));
for i=2:N-1
    f(i)= exp(mesh(i));
end
f
for i=1:N
    a(i,i)=-2;
    if i~=N
    a(i,i+1)=1;
    a(i+1,i)=1;
    end
end
a(1,1)=-1;
a = a/(h^2)
 u=a\f;
 uh = [  u; beta ]
 plot(mesh,uh,'r')
```

PLOTTING THE FINITE DIFFERENCE SOLUTION:



ERROR ESTIMATION IN 1-D FINITE DIFFERENCE METHOD WITH MATLAB CODES :

- **COMPUTING ERROR FOR BVP WITH DRICHLET BOUNDARY CONDITIONS:**

$$u'' = f(x) \ on \ I = (0,1)$$

$$\mathbf{u(0) = \alpha, u(1) = \beta}$$

Let U′ be the vector of original values of u at nodal points:

$$U' = \begin{bmatrix} u(x_1) \\ u(x_2) \\ u(x_3) \\ . \\ . \\ u(x_m) \end{bmatrix}$$

Let U be the vector of approximated value of u at nodal points:

$$U = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ . \\ . \\ U_m \end{bmatrix}$$

Then $E = U - U' = \begin{bmatrix} U_1 - u(x_1) \\ U_2 - u(x_2) \\ U_3 - u(x_3) \\ . \\ . \\ U_m - u(x_m) \end{bmatrix}$

Error: $\|E\|_\infty = \max |E_j| = O(h^2)$

THEORITICALLY COMPUTING THE ERROR:-

We compute the LOCAL TRUNCATION ERROR by replacing U_j by $u(x_j)$ in finite difference formula.

The true solution $u(x_j)$ won't satisfy this equation exactly and the discrepancy is the LTE δ_j

$$\Rightarrow \delta = AU' - F \qquad \text{---------------------------------}(1)$$

$$\delta_j = [u(x_{j-1}) - 2u(x_j) + u(x_{j+1})] / h^2 - f(x_j) \text{ for } j = 1, 2, \ldots\ldots, m.$$

By expanding $u(x_{j-1})$ and $u(x_{j+1})$ in taylor series,

$u(x_{j-1}) = u(x_j - h) = u(x_j) - h\, u'(x_j) + h^2/2\, u''(x_j) - h^3/6\, u^{(3)}(x_j) + h^4/24 u^{(4)}(x_j) - h^5/120 u^{(5)}(x_j) + O(h^6)$

$u(x_{j-+1}) = u(x_j + h) = u(x_j) + h\, u'(x_j) + h^2/2\, u''(x_j) + h^3/6\, u^{(3)}(x_j) + h^4/24 u^{(4)}(x_j) + h^5/120 u^{(5)}(x_j) + O(h^6)$

$\Rightarrow \delta_j = u''(x_j) + u^{(4)}(x_j)h^2/12 + O(h^4) - f(x_j)$

But $u'' = f$

$\Rightarrow \delta_j = u^{(4)}(x_j)h^2/12 + O(h^4)$

Hence, **LOCAL TRUNCATION ERROR = O (h^2).**

We compute the GLOBAL TRUNCATION ERROR:

$$AU = F$$
$$AU' = F + \boldsymbol{\delta}$$
$$\Rightarrow \quad A(U-U') = -\boldsymbol{\delta}$$
$$\Rightarrow \quad AE = -\boldsymbol{\delta}$$

Now, it is equivalent to system of equations:

$$\Rightarrow \quad [E_{j-1} - 2E_j + E_{j+1}] / h^2 = -\boldsymbol{\delta}_j$$

$$E_0 = E_{m+1} = 0$$

(Since $U_0 = \boldsymbol{\alpha}$, $U_{m+1} = \beta$ and these are exact values of u at boundary conditions).

It is discretization of ODE: $\quad e''(x) = -\boldsymbol{\delta}(x)$ on $(0, 1)$

$$e(0) = e(1) = 0$$

Since $\delta(x) = u^{(4)}(x)h^2/12$

$\Rightarrow e''(x) = -u^{(4)}(x)h^2/12$

$\Rightarrow e(x) = -u^{(2)}(x)h^2/12 + Cx + D$

Substituting $e(0)=e(1)=0$

$\Rightarrow e(x) = -u^{(2)}(x)h^2/12 + h^2/12([u^{(2)}(1) - u^{(2)}(0)]\, x + u^{(2)}(0))$

$= O(h^2)$

Hence, **GLOBAL TRUNCATION ERROR = O(h^2).**

NUMERICALLY, VERIFYING THAT ERROR = O(h^2) IN MATLAB:

CONSIDER EXAMPLE (2): $\quad u'' = 1 \text{ on } I = (0, 1)$

$$\mathbf{u(0) = 1,\ u(1) = 3}$$

```
function ERROR_EXAMPLE2
hv=[ 1/4   1/8   1/16    1/32  1/64 ];
for i=1:5

    [uhv]= FDM_1D_EXAMPLE2_error(hv(i));
     meshv = 0:hv(i):1;
     Nv(i)=1/hv(i);
       e = zeros(100,1);
       Error(i)=0;
       j=0:0.01:1
       vh= zeros(100,1);
       vh=interp1(meshv,uhv,j);
     for k=1:100
     uxact((k))=(((j(k))^2/2))+3*(j(k)/2)+1;
     e((k)) = (vh((k))- uxact((k)));
     end
     Error(i) = max(abs(e));
end
hv
Nv
Error
figure(1)
plot(Nv,Error,'r')
figure(2)
plot(hv,Error,'b')
end
```

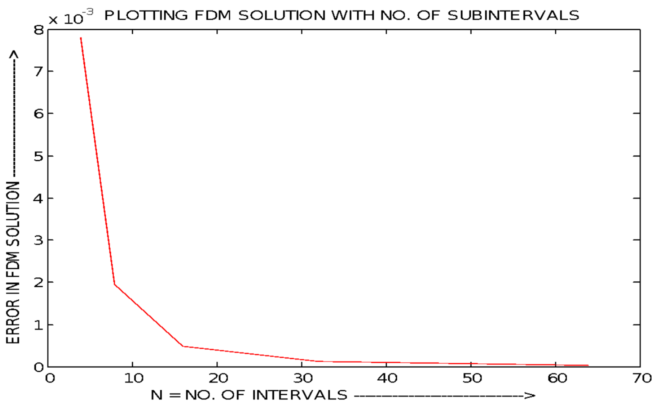
CODE IN MATLAB TO COMPUTE THE ERROR:

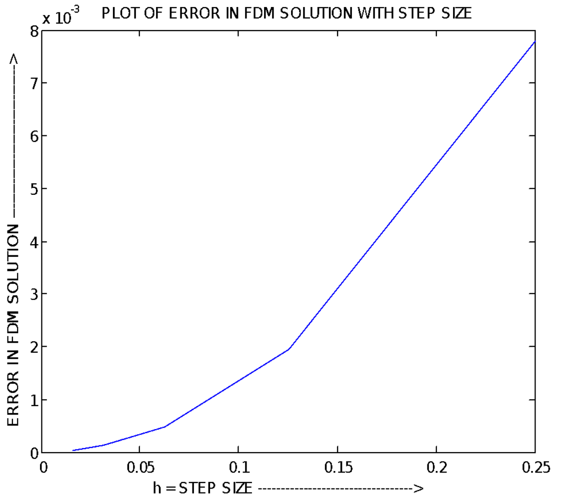
OUTPUT:-

hv = 0.2500   0.1250   0.0625   0.0313   0.0156

Nv = 4   8   16   32   64

Error = 0.0078   0.00195   0.000487   0.000121   0.0000304





From both the figures, it is clear that as we increase the no. of refinements, error in the Finite Difference Solution decreases and approaches to 0 as the no. of refinements are infinite.

Also, as we halve the step size h, Error in the FDM solution decreases by a factor of 4,

Verifying that ERROR = O (h^2).

- **COMPUTING ERROR FOR BVP WITH MIXED BOUNDARY CONDITIONS:**

$$u'' = f(x) \ on \ I = (0,1)$$

$$\mathbf{u'(0) = \alpha, \ u(1) = \beta}$$

THEORITICALLY COMPUTING THE ERROR:-

If we approximate u’ (0) by I order Forward difference formula: u’(0) = [u(h) – u(0)] / h,
Error = δ = [u (h) –u (0)] /h – **α**

Expanding u (h) by taylor series expansion:

$u(h) = u(0)+hu'(0)+h^2/2u''(0) +O\,(h^3)$

$\Longrightarrow\ \delta = [hu'(0)+h^2/2u''(0) +O\,(h^3)]/h - \boldsymbol{\alpha}$

$= u'(0) + h/2\ u''(0) +O(h^2) - \boldsymbol{\alpha}$

But $u'(0) = \boldsymbol{\alpha}$

$\Longrightarrow\ \delta = h/2\ u''(0) +O(h^2)$

Hence, ERROR = O(h)

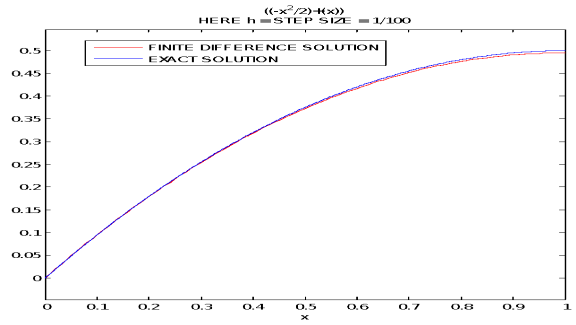
NUMERICALLY, VERIFYING THAT ERROR = O(h) IN MATLAB:

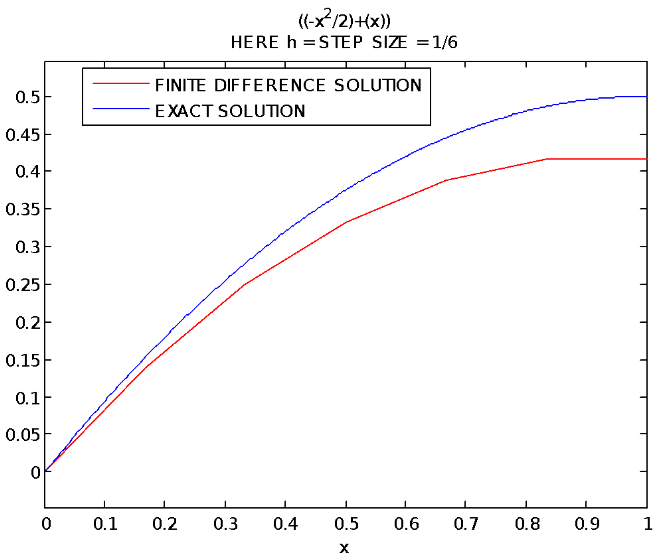
CONSIDER EXAMPLE (3): $-u'' = 1\ on\ I = (0,1)$

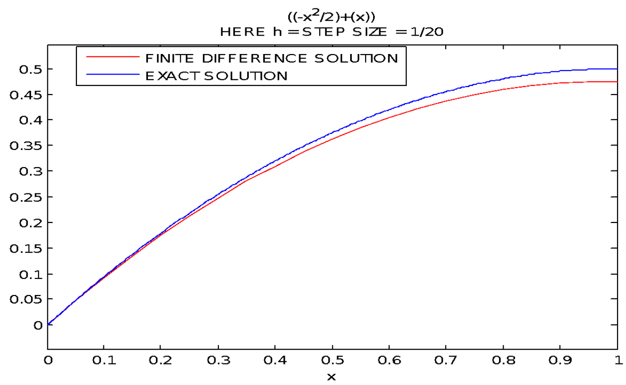
u (0) = 0, u'(1) = 0

```
function ERROR_EXAMPLE3
hv=[ 0.5  0.25 0.25/2 0.25/4 0.25/8 0.25/16 0.25/32];
for i=1:7

    [uhv]= FDM_1D_EXAMPLE3(hv(i));
     meshv = 0:hv(i):1;
     Nv(i)=1/hv(i);
       e = zeros(Nv(i)+1,1);
       Error(i)=0;
     for j=1:Nv(i)+1
     uxact(j)=(-(meshv(j)^2/2))+meshv(j);
     e(j) = (uhv(j)- uxact(j));
     end
     Error(i) = max(abs(e))
end
hv
Nv
Error
figure(1)
plot(Nv,Error,'r')
figure(2)
plot(hv,Error,'b')
end
```

Where we approximated u' by I order forward difference formula.







We plotted the finite difference solution with the exact solution and it is clear that as we decrease the step size, the FDM solution graph and the exact solution graph overlap, Concluding that as the no. of refinements increase, FDM solution approaches the Exact solution.

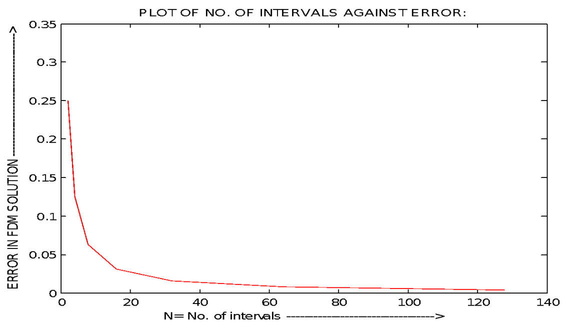
CODE IN MATLAB TO COMPUTE THE ERROR:

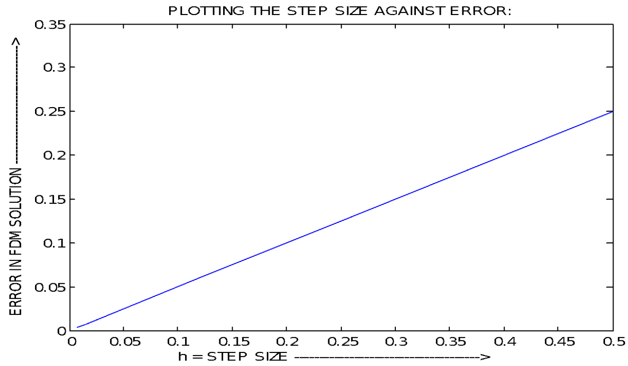
OUTPUT:

hv = 0.5000 0.2500 0.1250 0.0625 0.0313 0.0156 0.0078

Nv = 2 4 8 16 32 64 128

Error = 0.2500 0.1250 0.0625 0.0312 0.0156 0.0078 0.0039





Observe that as the Step size h is halved, Error in FDM solution also halves and hence,

ERROR = O (h).

If we approximate u' (0) by II order Central difference formula: u'(0)= [u(h)-u(-h)] / 2h

Error = δ = [u(h)- u(-h)] / 2h – **α**

Expanding u(h) and u(-h) in taylor series,

$u(h)= u(0)+hu'(0)+h^2/2u''(0)+h^3/6u'''(0)+$……….

$u(-h)= u(0)-hu'(0)+h^2/2u''(0)-h^3/6u'''(0)+$……….

$\Longrightarrow \delta = u'(0) + h^2/6.u''(0) - \boldsymbol{\alpha}$

But $u'(0) = \boldsymbol{\alpha}$

$\Longrightarrow \delta = h^2/6.u''(0) = O(h^2)$

Hence, ERROR $=O(h^2)$.

CONSIDER EXAMPLE (3): $-u'' = 1 \; on \; I = (0,1)$

u (0) = 0, u'(1) = 0

Where we approximated u' by II order central difference formula.

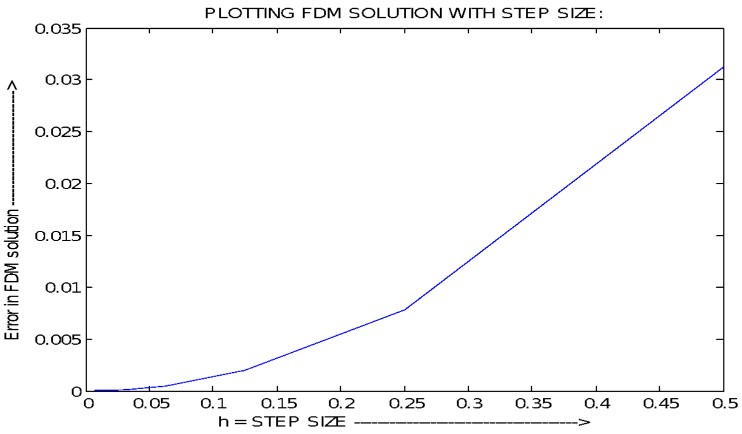
```
function ERROR_EXAMPLE3_2
hv=[1/2  1/4  1/8  1/16   1/32  1/64  1/128];
for i=1:7

   [uhv]= FDM_1D_EXAMPLE3(hv(i));
    meshv = 0:hv(i):1;
    Nv(i)=1/hv(i);
      e = zeros(100,1);
      Error(i)=0;
      j=0:0.01:1;
      vh= zeros(100,1);
      vh=interp1(meshv,uhv,j);
    for k=1:100
    uxact((k))=(-((j(k))^2/2))+(j(k));
    e((k)) = (vh((k))- uxact((k)));
    end
    Error(i) = max(abs(e));
end
hv
Nv
Error
figure(1)
```

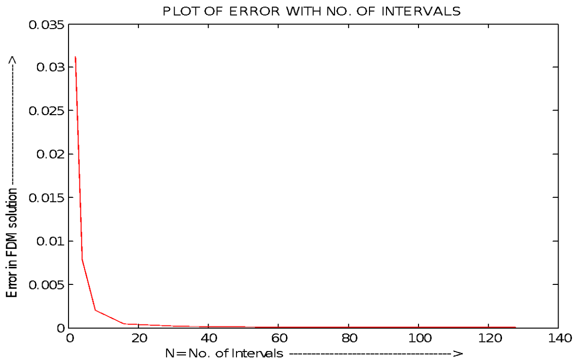
OUTPUT:

hv = 0.5000   0.2500   0.1250   0.0625   0.0313

Nv = 2   4   8   16   32

Error = 0.0313   0.0078   0.0019   0.0005   0.0001





From both the figures, it is clear that as we increase the no. of refinements, error in the Finite Difference Solution decreases and approaches to 0 as the no. of refinements are infinite.

Also observe that as the Step size h is halved, Error in FDM solution decreases by a factor of 4 and hence,

Verifying that ERROR = O (h^2).

References:-

[1] Finite difference methods for ordinary and partial differential equation, Randall. J. LeVegue.

[2] Finite difference numerical methods for partial differential equations, Aitor Bergara.

[3] Neela Nataraj, “Introduction to Finite Difference Method Lecture”, Department of Mathematics, Indian Institute of Technology Bombay