AKSHAY · J
21105012

FINITE ELEMENT METHOD FOR FLUID DYNAMICS

ASSIGNMENT - II

1> $u_{,xx} = -f.$ $f = qx$

$\frac{\partial^2 u}{\partial x^2} = -f = -qx.$

$\frac{\partial u}{\partial x} = \frac{-qx^2}{2} + c_1$

$u = \frac{-qx^3}{6} + c_1 x + c_2.$

at $x=1$, $u=0$ at $x=0$, $\frac{\partial u}{\partial x} = 0.$

$u = -q/6 + c_1 + c_2 = 0$ $0 = -\frac{q \times 0}{2} + c_1 \Rightarrow c_1 = 0$

$c_1 + c_2 = q/6$ $g, h = 0$

$c_2 = q/6$

$u = q/6 - qx^3/6$

2> $u \in \mathcal{S}$, $\omega \in \mathcal{V}$ $\mathcal{S} = \{u \mid u \in H^1, u(1) = g\}.$

$\int \omega_{,x} u_{,x} dx = \int \omega f dx + h\omega(0)$ $\mathcal{V} = \{\omega \mid \omega \in H^1, \omega(1) = 0\}.$

$a(\omega, u) = (\omega, f) + h\omega(0)$ (weak form)

$\omega^h \in \mathcal{V}^h$, $u^h \in \mathcal{S}^h$, $v^h \in \mathcal{V}^h$

$a(\omega^h, v^h) = (\omega^h, f) + h\omega^h(0) - a(\omega^h, g^h)$ (Galerkin form)

$\omega^h = \sum_{A=1}^{n} c_A N_A$ $N_{n+1}(1) = g$

$v^h = \sum_{B=1}^{n} d_B N_B.$

$a\left(\sum_{A=1}^{n} c_A N_A, \sum_{B=1}^{n} d_B N_B\right) = \left(\sum c_A N_A, f\right) + \sum h c_A N_A(0) - a\left(\sum c_A N_A, N_{n+1}\right) g$

$\sum_A c_A \left[\sum_B a(N_A, N_B) d_B\right] = \sum_A c_A \left[(N_A, f) + hN_A(0) - a(N_A, N_{n+1}) g\right].$

$Kd = F$

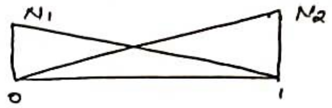
$K = a(N_A, N_B)$ $F = (N_A, f) + hN_A(0) - a(N_A, N_{n+1}) g$

$u^h = \sum_{B=1}^{n+1} d_B N_B$ $d_{N+1} = g$

for $n=1$, $h=1$,

$$N_A = \begin{cases} \dfrac{x - x_{A-1}}{h_{A-1}} & x_{A-1} \le x \le x_A \\ \dfrac{x_{A+1} - x_A}{h_A} & x_A \le x \le x_{A+1} \end{cases}$$

$$N_1 = \{ 1 - x \quad 0 \le x \le 1. \qquad N_2 = \{ x \quad 0 \le x \le 1$$



$$u^h = d_1 N_1 + g N_2.$$

$$K = a(N_A, N_B)$$

$$K = a(N_1, N_1) = \int_0^1 N_{1,x} N_{1,x}\, dx = \int_0^1 -1 \times -1\, dx = \underline{\underline{+1}}.$$

$\underline{\underline{f = qx}}$

$$F = (N_A, f) + h N_A(0) - a(N_A, N_2) g.$$

$$F = \int_0^1 N_1 \times qx + h \times N_1(0) - \int_0^1 (N_{1,x} \times N_{2,x}) g\, dx.$$

$$F = \int_0^1 q(x - x^2) dx + h \times 1 - \int_0^1 -1 \times 1 \times g\, dx.$$

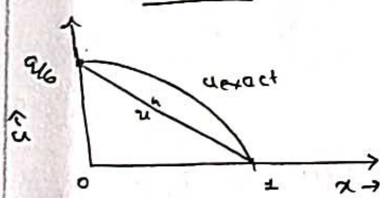
$$F = q(1/2 - 1/3) + h + g = \underline{\underline{q/6 + h + g}}$$

$$d = K^{-1} F = 1 \times (\underline{\underline{q/6}} + h + g)$$

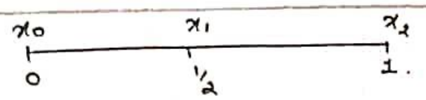
$$u^h = (q/6 + h + g)(1 - x) + gx.$$

$$h = g = 0.$$

$$u^h = \underline{\underline{q/6 (1 - x)}}, \quad u_{exact} = \underline{\underline{q/6 - qx^3/6}}$$

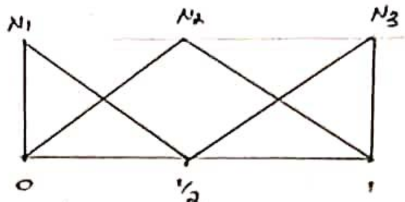


for $n=2$, $h=1/2$



$$N_1 = \begin{cases} 1-2x & 0 \le x \le 1/2. \end{cases} \qquad N_2 = \begin{cases} 2x & 0 \le x \le 1/2 \\ 2(1-x) & 1/2 \le x \le 1 \end{cases}$$

$$N_3 = \begin{cases} 2x-1 & 1/2 \le x \le 1 \end{cases}$$



$u^h = d_1 N_1 + d_2 N_2 + g N_3$.

$$K_{11} = a(N_1, N_1) = \int_0^{1/2} N_{1,x} N_{1,x}\, dx = \int_0^{1/2} -2 \times -2 = \frac{+4}{2} = \underline{\underline{+2}}.$$

$$K_{12} = K_{21} = a(N_1, N_2) = \int_0^{1/2} N_{1,x} N_{2,x}\, dx = \int_0^{1/2} -2 \times 2\, dx = -4/2 = \underline{\underline{-2}}.$$

$$K_{22} = a(N_2, N_2) = \int_0^{1/2} 2 \times 2\, dx + \int_{1/2}^{1} -2 \times -2\, dx = 2 + 2 = \underline{\underline{4}}.$$

$$K = \underline{\underline{\begin{bmatrix} 2 & -2 \\ -2 & 4 \end{bmatrix}}}.$$

$F = (N_A, f) + h N_A(0) - a(N_A, N_3) g$.

$$f_1 = \int_0^{1/2} N_1 \times q x\, dx + h N_1(0) - \int_0^{Q} N_{1,x} N_{3,x}\, g\, dx.$$

$$= \int_0^{1/2} q(x - 2x^2)\, dx + h - \int_0^{Q} \cancel{-2 \times 2} \times g\, dx.$$

$$= \cancel{q(1/2 \cdot 1/4)} \; q\left(\tfrac{1}{2} \times \tfrac{1}{4} - \tfrac{2}{3} \times \tfrac{1}{8}\right) + h$$

$$= \underline{\underline{q/24 + h}}$$

$$f_2 = \int_0^{1/2} 2x \times q x\, dx + \int_{1/2}^{1} 2q(x - x^2)\, dx + h N_2(0) - \int_{1/2}^{1} -2 \times 2 \times g\, dx.$$

$$= \frac{2q}{3}\left(\tfrac{1}{8}\right) + 2q\left(\tfrac{1}{2} \times \tfrac{3}{4} - \tfrac{1}{3} \times \tfrac{7}{8}\right) + h \times 0 + \cancel{4}^{2} \times g \times \tfrac{1}{\cancel{2}}$$

$$= \underline{\underline{q/12 + q/6 + 2g}}. \qquad F = \begin{bmatrix} q/24 + h \\ q/12 + q/6 + 2g \end{bmatrix}$$

$$\underline{\underline{d = K^{-1} F}}$$

$g = h = 0$

$$K^{-1} = \frac{1}{4}\begin{bmatrix} 4 & 2 \\ 2 & 2 \end{bmatrix}$$

$$d = \frac{1}{4}\begin{pmatrix} 4 & 2 \\ 2 & 2 \end{pmatrix}\begin{pmatrix} q/24 + h \\ q/4 + 2g \end{pmatrix} = \frac{1}{4}\begin{bmatrix} q/6 + 4h + q/2 + 4g \\ q/12 + 2h + q/2 + 4g \end{bmatrix} = \frac{1}{4}\begin{bmatrix} q/6 + q/2 \\ q/12 + q/2 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 2/3\, q \\ 7/12\, q \end{bmatrix}$$

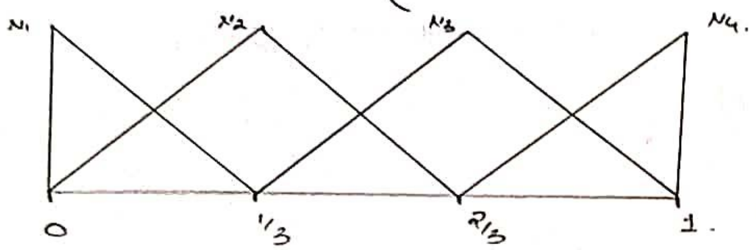
$$u^h = \frac{q}{6}(1-2x) + \frac{1}{48}q \begin{cases} 2x & 0 \le x \le 1/2 \\ 2(1-x) & 1/2 \le x \le 1 \end{cases}$$

$\{0 \le x \le 1/2\}$.

for $n = 3$, $h = 1/3$, $x_0 = 0$, $x_1 = 1/3$, $x_2 = 2/3$, $x_3 = 1$.

$$N_1 = \begin{cases} 1-3x & 0 \le x \le 1/3. \end{cases} \qquad N_2 = \begin{cases} 3x & 0 \le x \le 1/3. \\ 2-3x & 1/3 \le x \le 2/3 \end{cases}$$

$$N_3 = \begin{cases} 3x-1 & 1/3 \le x \le 2/3 \\ 3-3x & 2/3 \le x \le 1 \end{cases} \qquad N_4 = \begin{cases} 3x-2 & 2/3 \le x \le 1 \end{cases}$$



$k_{11} = a(N_1, N_1) = \int_0^{1/3} -3 \times -3 \, dx = \underline{\underline{3}}$

$k_{12} = k_{21} = a(N_1, N_2) = \int_0^{1/3} -3 \times 3 \, dx = -9/3 = \underline{\underline{-3}}$.

$k_{13} = k_{31} = \underline{\underline{0}}$.

$k_{22} = a(N_2, N_2) = \int_0^{1/3} 3 \times 3 \, dx + \int_{1/3}^{2/3} -3 \times -3 \, dx = 9/3 + 9/3 = 18/3 = \underline{\underline{6}}$

$k_{23} = k_{32} = \int_{1/3}^{2/3} -3 \times 3 \, dx = -9 \times 1/3 = \underline{\underline{-3}}$.

$k_{33} = a(N_3, N_3) = \int_{1/3}^{2/3} 3 \times 3 \, dx + \int_{2/3}^{1} -3 \times -3 \, dx = 9/3 + 9/3 = \underline{\underline{6}}$.

$$K = \begin{bmatrix} 3 & -3 & 0 \\ -3 & 6 & -3 \\ 0 & -3 & 6 \end{bmatrix}$$

$$K^{-1} = \frac{1}{3}\begin{bmatrix} 3 & 2 & 1 \\ 2 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$f = (N_A, f) + h N_A(0) - a(N_A, N_4) g$

$\frac{x - x_1}{h}$ $\frac{x - 1/3}{1/3}$ $\frac{1-x}{3}$

$F_1 = (N_1, qx) + hN_1(0) - a(N_1, N_4)g$

$F_1 = \int_0^{1/3} (x - 3x^2) q\,dx + h \times 1 - 0 = q\left[\frac{1}{2} \times \frac{1}{9} - \frac{3}{3} \times \frac{1}{27}\right] + h$

$= q/54 + h$

$F_2 = (N_2, qx) + b N_2(0) - a(\cancel{N_2, N_4}^{0})g$

$F_2 = \int_0^{1/3} 3qx^2\,dx + \int_{1/3}^{2/3} (2x - 3x^2) q\,dx + 0$

$F_2 = 3/3 \times q \times 1/27 + \left[2/2\left(4/9 - 1/9\right) - 3/3\left(8/27 - 1/27\right)\right] q$

$F_2 = \cancel{q/27} + q/3 - 7\cancel{q}/27 = \underline{\underline{q/3}} - \frac{6q}{27} = q/9$

$F_3 = (N_3, qx) + b\cancel{N_3(0)}^{0} - a(N_3, N_4)g$

$F_3 = \int_{1/3}^{2/3} (3x^2 - x) q\,dx + \int_{2/3}^{1} (3x - 3x^2) q\,dx - \int_{2/3}^{1} -3 \times 3g\,dx.$

$F_3 = \left(3/3\left(8/27 - 1/27\right) - \frac{1}{2}\left(4/9 - 1/9\right)\right] q + \left[3/2\left(1 - 4/9\right) - 3/3\left(1 - 8/27\right)\right] q + 9/3\, g$

$F_3 = 7q/27 - q/6 + 5q/6 - 19q/27 + 3g = \underline{\underline{2q/9 + 3g}}$

$d = K^{-1} f = \frac{1}{3}\begin{bmatrix} 3 & 2 & 1 \\ 2 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}\begin{bmatrix} q/54 \\ q/9 \\ 2q/9 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 14q/18 \\ 23q/27 \\ 19q/54 \end{bmatrix} = \cancel{\begin{bmatrix} 17q/54 \\ 25q/81 \\ 31q/162 \end{bmatrix}} \begin{bmatrix} 1/6\, q \\ 13q/81 \\ 19q/162 \end{bmatrix}$

$u^h = d_1 N_1 + d_2 N_2 + d_3 N_3 + g N_4.$

$u^h = \frac{q}{6}(1 - 3x) + \frac{13q}{81} \times \begin{cases} 3x & 0 \le x \le 1/3 \\ 2 - 3x & 1/3 \le x \le 2/3 \end{cases} + \frac{19q}{162} \times \begin{cases} 3x - 1 & 1/3 \le x \le 2/3. \\ 3 - 3x & 2/3 \le x \le 1. \end{cases}$

$\{0 \le x \le 1/3\}$

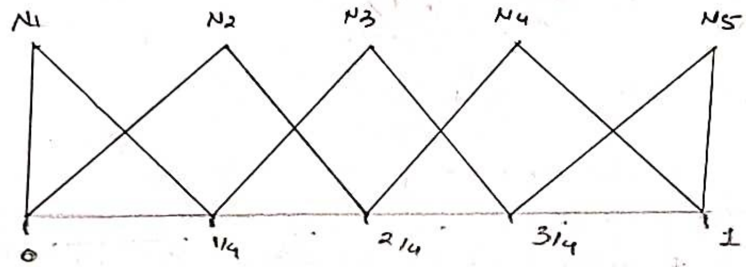
for $n = 4$, $h = 1/4$ $\quad x_0 = 0,\ x_1 = 1/4,\ x_2 = 2/4,\ x_3 = 3/4,\ x_4 = 1$

$$N_1 = \begin{cases} 1 - 4x & 0 \le x \le 1/4 \end{cases} \qquad N_2 = \begin{cases} 4x & 0 \le x \le 1/4 \\ 2 - 4x & 1/4 \le x \le 2/4 \end{cases}$$

$$N_3 = \begin{cases} 4x - 1 & 1/4 \le x \le 2/4 \\ 3 - 4x & 2/4 \le x \le 3/4 \end{cases} \qquad N_4 = \begin{cases} 4x - 2 & 2/4 \le x \le 3/4 \\ 4 - 4x & 3/4 \le x \le 1 \end{cases}$$

$$N_5 = \begin{cases} 4x - 3 & 3/4 \le x \le 1 \end{cases}$$

$\frac{2/4 - x}{1/4}$ $\quad \frac{x - 1/4}{1/4}$ $\quad \frac{x - 2/4}{1/4}$ $\quad \frac{1 - x}{1/4}$



$K_{11} = a(N_1, N_1) = \int_0^{1/4} -4x - 4\,dx = 16/4 = \underline{\underline{4}}$

$K_{12} = a(N_1, N_2) = K_{21} = \int_0^{1/4} -4 \times 4\,dx = \underline{\underline{-4}}$

$K_{13} = K_{14} = 0 = K_{31} = K_{41}$

$K_{22} = a(N_2, N_2) = \int_0^{1/4} 4 \times 4\,dx + \int_{1/4}^{2/4} -4x - 4\,dx = 16/4 + \frac{16}{4} = \underline{\underline{8}}$

$K_{23} = K_{32} = a(N_2, N_3) = \int_{1/4}^{2/4} -4 \times 4\,dx = -16/4 = \underline{\underline{-4}}.$

$K_{24} = K_{42} = 0.$

$K_{33} = a(N_3, N_3) = \int_{1/4}^{2/4} 4 \times 4\,dx + \int_{2/4}^{3/4} -4x - 4\,dx = 16/4 + 16/4 = \underline{\underline{8}}.$

$K_{34} = K_{43} = a(N_3, N_4) = \int_{2/4}^{3/4} -4 \times 4\,dx = -16 \times 1/4 = \underline{\underline{-4}}.$

$K_{44} = \int_{2/4}^{3/4} 4 \times 4\,dx + \int_{3/4}^{1} -4x - 4\,dx = 16/4 + 16/4 = \underline{\underline{8}}.$

$$K = \begin{bmatrix} 4 & -4 & 0 & 0 \\ -4 & 8 & -4 & 0 \\ 0 & -4 & 8 & -4 \\ 0 & 0 & -4 & 8 \end{bmatrix} \qquad K^{-1} = \frac{1}{4}\begin{bmatrix} 4 & 3 & 2 & 1 \\ 3 & 3 & 2 & 1 \\ 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$F = (N_A, f) + h N_A(0) - a(N_A, \overset{0}{\cancel{N_5}}) g$$

$$F_1 = \int_0^{1/4} N_1 q x \, dx + h N_1(0) - 0$$

$$F_1 = \int_0^{1/4} (x - 4x^2) q \, dx + h = \tfrac{1}{2}(1/16) - \tfrac{4}{3}(1/64) = \underline{\underline{q/96}}$$

$$F_2 = \int_0^{1/4} 4x^2 q \, dx + \int_{1/4}^{2/4} (2x - 4x^2) q \, dx + h \times 0 - a(N_2, \overset{0}{\cancel{N_5}}) g$$

$$F_2 = 4/3 \times 1/64 \, q + \left[2/2\,(4/16 - 1/16) - 4/3\,(8/64 - 1/64)\right] q = q/48 + \left[3/16 - 7/48\right] q$$

$$= \underline{\underline{q/16}}$$

$$F_3 = \int_{1/4}^{2/4} (4x^2 - x) q \, dx + \int_{2/4}^{3/4} (3x - 4x^2) q \, dx + 0 - 0$$

$$= \left(4/3\,(8/64 - 1/64) - \tfrac{1}{2}(4/16 - 1/16)\right] q + \left[3/2\,(9/16 - 4/16) - 4/3\left[27/64 - 8/64\right]\right] q$$

$$= 5q/96 + 7q/96 = 12q/96 = \underline{\underline{q/8}}$$

$$F_4 = \int_{2/4}^{3/4} (4x^2 - 2x) q \, dx + \int_{3/4}^{1} (4x - 4x^2) q \, dx + 0 - \int_{3/4}^{1} 4x - 4g \, dx.$$

$$= \left[4/3\left(\frac{27}{64} - \frac{8}{64}\right]\right. - 2/2\left[9/16 - 4/16\right] q + \left(4/2\,(1 - 9/16) - 4/3\,(1 - 27/64)\right] q + \frac{16}{4} g$$

$$= q/12 + 5q/48 + \underline{16g/4.} = \underline{\underline{\cancel{9/48}\, q}} \overset{9/48\, q}{} + \overset{0}{\cancel{16g/4}}$$

$g = \underline{\underline{h = 0}}.$

$$d = k^{-1}f = \frac{1}{4}\begin{bmatrix} 4 & 3 & 2 & 1 \\ 3 & 3 & 2 & 1 \\ 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}\begin{bmatrix} q/96 \\ q/16 \\ q/8 \\ q/48\,q \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 2q/3 \\ 21q/32 \\ 7q/12 \\ 37q/96 \end{bmatrix} = \begin{bmatrix} 1/6\,q \\ 21q/128 \\ 7q/48 \\ 37q/384 \end{bmatrix}$$

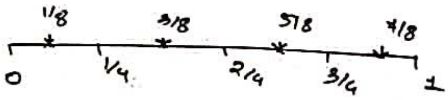
$$u^h = d_1 N_1 + d_2 N_2 + d_3 N_3 + d_4 N_4 + \cancel{g N_5}^{0}$$

$$u^h = \tfrac{1}{6} q(1-4x)\{0 \le x \le 1/4\} + \frac{21q}{128}\begin{cases} 4x & 0 \le x \le 1/4 \\ 2-4x & 1/4 \le x \le 2/4 \end{cases} + \frac{7q}{48}\begin{cases} 4x-1 & 1/4 \le x \le 2/4 \\ 3-4x & 2/4 \le x \le 3/4 \end{cases}$$

$$+ \frac{37q}{384}\begin{cases} 4x-2 & 2/4 \le x \le 3/4. \\ 4-4x & 3/4 \le x \le 1 \end{cases}$$

3> Yes the stiffness matrix is banded (tridiagonal). No matter what is the value of g and h stiffness matrix will be banded. That is 'g' and 'h' has no influence in the bandedness of 'K'. K =

4> $re,x = |u^h,x - u,x| / q/2$.



$$u^h,x = q/6\,(-4)\{0 \le x \le 1/4\} + \frac{21q}{128}\begin{cases} 4 & 0 \le x \le 1/4 \\ -4 & 1/4 \le x \le 2/4 \end{cases} + \frac{7q}{48}\begin{cases} 4 & 1/4 \le x \le 2/4 \\ -4 & 2/4 \le x \le 3/4 \end{cases}$$

$$+ \frac{37q}{384}\begin{cases} 4 & 2/4 \le x \le 3/4. \\ -4 & 3/4 \le x \le 1 \end{cases}$$

$u_{exact},x = -3qx^2/6$.

$$\left.\frac{|u^h,x - u,x|}{(q/2)}\right|_{x=1/8} = \frac{\left|\left(-4q/6 + \frac{21\times 4q}{128}\right) + \left(\frac{3q}{6}(1/64)\right)\right|}{q/2} = \underline{\underline{5.208\times10^{-3}}}$$

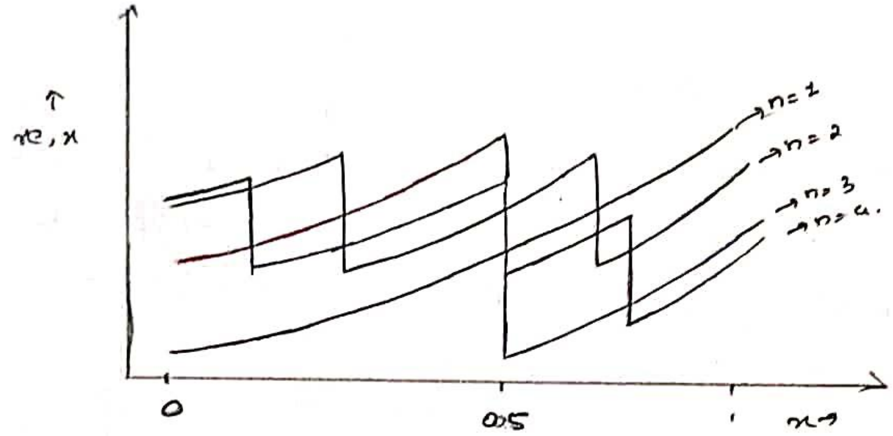
$$\left.\frac{|u^h,x - u,x|}{(q/2)}\right|_{x=3/8} = \frac{\left|-\frac{21\times4q}{128} + \frac{7q}{48}\times4 + \frac{3q}{6}(9/64)\right|}{q/2} = \underline{\underline{5.208\times10^{-3}}}$$

$$\left.\frac{|u^h,x - u'x|}{(q/2)}\right|_{x=5/8} = \frac{\left|\frac{7\times-4q}{48} + \frac{37\times4q}{384} + \frac{3q}{6}(25/64)\right|}{q/2} = \underline{\underline{5.208\times10^{-3}}}$$

$$\frac{|u^h,x - u'x|}{(q/2)}\bigg|_{x=7/8} = \frac{\left|\frac{-4\times37}{384}x + \frac{3x}{6}\left(\frac{49}{64}\right)\right|}{q/2} = \underline{\underline{5.208\times10^{-3}}}$$

They all are equal.

5) for the plot of $\ln(re,x)$ Vs $\ln h$ for each h values the value of $\ln(re,x)$ will be same at the midpoint of each elements.



6) $$re,x = \frac{|u^h,x - u,x|}{q/2}.$$

$$\ln(re,x) = \ln(|u^h,x - u,x|) - \ln(q/2)$$

$$y = mx + c.$$

intercept $\Rightarrow c = -\ln(q/2)$

it gives an indication of magnitude of error when $x=0$.

slope gives an indication amount of reduction in relative error with x. steeper slope indicates higher reduction.

Slopes are adjusted in such a way as to reduce the re,x for higher n values.

B)

```
###AKSHAY J
###21105012
import numpy as np
from matplotlib import pyplot as plt
from scipy.integrate import quad

###GIVEN
n1=10
n2=50
n3=100
n4=3
h1=1/n1
h2=1/n2
h3=1/n3
h4=1/n4
q=1              ##THE VALUE OF q IS TO BE TAKEN TO BE
ONE
Nder1=np.zeros((n1+1,1000))
Nder2=np.zeros((n2+1,1000))
Nder3=np.zeros((n3+1,1000))
Nder4=np.zeros((n4+1,1000))
x=np.linspace(0,1,1000)
xn1=np.linspace(0,1,n1+1)
xn2=np.linspace(0,1,n2+1)
xn3=np.linspace(0,1,n3+1)
xn4=np.linspace(0,1,n4+1)
K1=np.zeros((n1,n1))
K2=np.zeros((n2,n2))
K3=np.zeros((n3,n3))
K4=np.zeros((n4,n4))
F1=np.zeros((n1,1))
F2=np.zeros((n2,1))
F3=np.zeros((n3,1))
F4=np.zeros((n4,1))
ke=np.zeros((2,2))
fe=np.zeros((2,1))
d1=np.zeros((n1,1))
d2=np.zeros((n2,1))
d3=np.zeros((n3,1))
d4=np.zeros((n4,1))
u_exact=np.zeros(1000)
```

```
uexact_slope=np.zeros(1000)
N1=np.zeros((n1+1,1000))
N2=np.zeros((n2+1,1000))
N3=np.zeros((n3+1,1000))
N4=np.zeros((n4+1,1000))
Re1=0
Re2=0
Re3=0
Re4=0

########SETTING THE BASIS FUNCTION VALUE
#N=10
for A in range(1,n1):
    for i in range(1000):
      if x[i] <= h1:
       N1[0,i]=(xn1[1]-x[i])/h1
      if x[i] >= xn1[n1-1]:
       N1[n1,i]=(x[i]-xn1[n1-1])/h1
      if xn1[A-1] <= x[i] and x[i] <= xn1[A]:

        N1[A,i]=(x[i]-xn1[A-1])/h1
      if xn1[A] <= x[i] and x[i] <= xn1[A+1]:

        N1[A,i] = (xn1[A+1] - x[i]) / h1
#N=50
for A in range(1,n2):
    for i in range(1000):
      if x[i] <= h2:
       N2[0,i]=(xn2[1]-x[i])/h2
      if x[i] >= xn2[n2-1]:
       N2[n2,i]=(x[i]-xn2[n2-1])/h2
      if xn2[A-1] <= x[i] and x[i] <= xn2[A]:

        N2[A,i]=(x[i]-xn2[A-1])/h2
      if xn2[A] <= x[i] and x[i] <= xn2[A+1]:

        N2[A,i] = (xn2[A+1] - x[i]) / h2
#N=100
for A in range(1,n3):
    for i in range(1000):
      if x[i] <= h3:
       N3[0,i]=(xn3[1]-x[i])/h3
      if x[i] >= xn3[n3-1]:
       N3[n3,i]=(x[i]-xn3[n3-1])/h3
```

```
      if xn3[A-1] <= x[i] and x[i] <= xn3[A]:

        N3[A,i]=(x[i]-xn3[A-1])/h3
      if xn3[A] <= x[i] and x[i] <= xn3[A+1]:

        N3[A,i] = (xn3[A+1] - x[i]) / h3
#N=2
for A in range(1,n4):
    for i in range(1000):
      if x[i] <= h4:
       N4[0,i]=(xn4[1]-x[i])/h4
      if x[i] >= xn4[n4-1]:
       N4[n4,i]=(x[i]-xn4[n4-1])/h4
      if xn4[A-1] <= x[i] and x[i] <= xn4[A]:

        N4[A,i]=(x[i]-xn4[A-1])/h4
      if xn4[A] <= x[i] and x[i] <= xn4[A+1]:

        N4[A,i] = (xn4[A+1] - x[i]) / h4
####
#N=10
for A in range(1,n1):
    for i in range(1000):
      if xn1[0] <= x[i] and x[i] <= xn1[1]:
        Nder1[0,i] = (-1) / h1
      if xn1[n1 - 1] <= x[i] and x[i] <= xn1[n1]:
        Nder1[n1,i] = (1) / h1
      if xn1[A - 1] <= x[i] and x[i] <= xn1[A]:
                Nder1[A,i] = (1) / h1
      if xn1[A] <= x[i] and x[i] <= xn1[A+1]:
                Nder1[A,i] = (-1) / h1
#N=50
for A in range(1,n2):
    for i in range(1000):
      if xn2[0] <= x[i] and x[i] <= xn2[1]:
        Nder2[0,i] = (-1) / h2
      if xn2[n2 - 1] <= x[i] and x[i] <= xn2[n2]:
        Nder2[n2,i] = (1) / h2
      if xn2[A - 1] <= x[i] and x[i] <= xn2[A]:
                Nder2[A,i] = (1) / h2
      if xn2[A] <= x[i] and x[i] <= xn2[A+1]:
                Nder2[A,i] = (-1) / h2
#N=100
for A in range(1,n3):
```

```
    for i in range(1000):
      if xn3[0] <= x[i] and x[i] <= xn3[1]:
        Nder3[0,i] = (-1) / h3
      if xn3[n3 - 1] <= x[i] and x[i] <= xn3[n3]:
        Nder3[n3,i] = (1) / h3
      if xn3[A - 1] <= x[i] and x[i] <= xn3[A]:
                Nder3[A,i] = (1) / h3
      if xn3[A] <= x[i] and x[i] <= xn3[A+1]:
                Nder3[A,i] = (-1) / h3
#N=2
for A in range(1,n4):
    for i in range(1000):
      if xn4[0] <= x[i] and x[i] <= xn4[1]:
        Nder4[0,i] = (-1) / h4
      if xn4[n4 - 1] <= x[i] and x[i] <= xn4[n4]:
        Nder4[n4,i] = (1) / h4
      if xn4[A - 1] <= x[i] and x[i] <= xn4[A]:
                Nder4[A,i] = (1) / h4
      if xn4[A] <= x[i] and x[i] <= xn4[A+1]:
                Nder4[A,i] = (-1) / h4
##########CALCULATING TE EXACT GRAPH
for i in range(len(x)):
    u_exact[i] = (q / 6) - ((q * x[i] ** 3) / 6)
for i in range(0,len(x)):
    uexact_slope[i]=(-q*x[i]**2)/2

##########FINDING Ke AND Fe AND ASSEMBLING IT TO
GLOBAL K AND F
# N=10
for a in range(len(xn1)-1):
    for i in range(0,2):
        for j in range(0,2):
           ke[i][j]=((-1)**(i+j))/h1
           f= lambda zeta: (h1/8) *q*
(h1*zeta+xn1[a]+xn1[a+1]) *(((-1)**(i+1) * zeta) +1)
           f=quad(f,-1,1)
           fe[i][0]=f[0]
    if a==n1-1:
        K1[a][a] = K1[a][a] + ke[0][0]
        F1[a][0] = F1[a][0] + fe[0][0]
    else:
      K1[a][a] = K1[a][a] + ke[0][0]
      K1[a][a+1] = K1[a][a+1] + ke[0][1]
      K1[a+1][a] = K1[a+1][a] + ke[1][0]
```

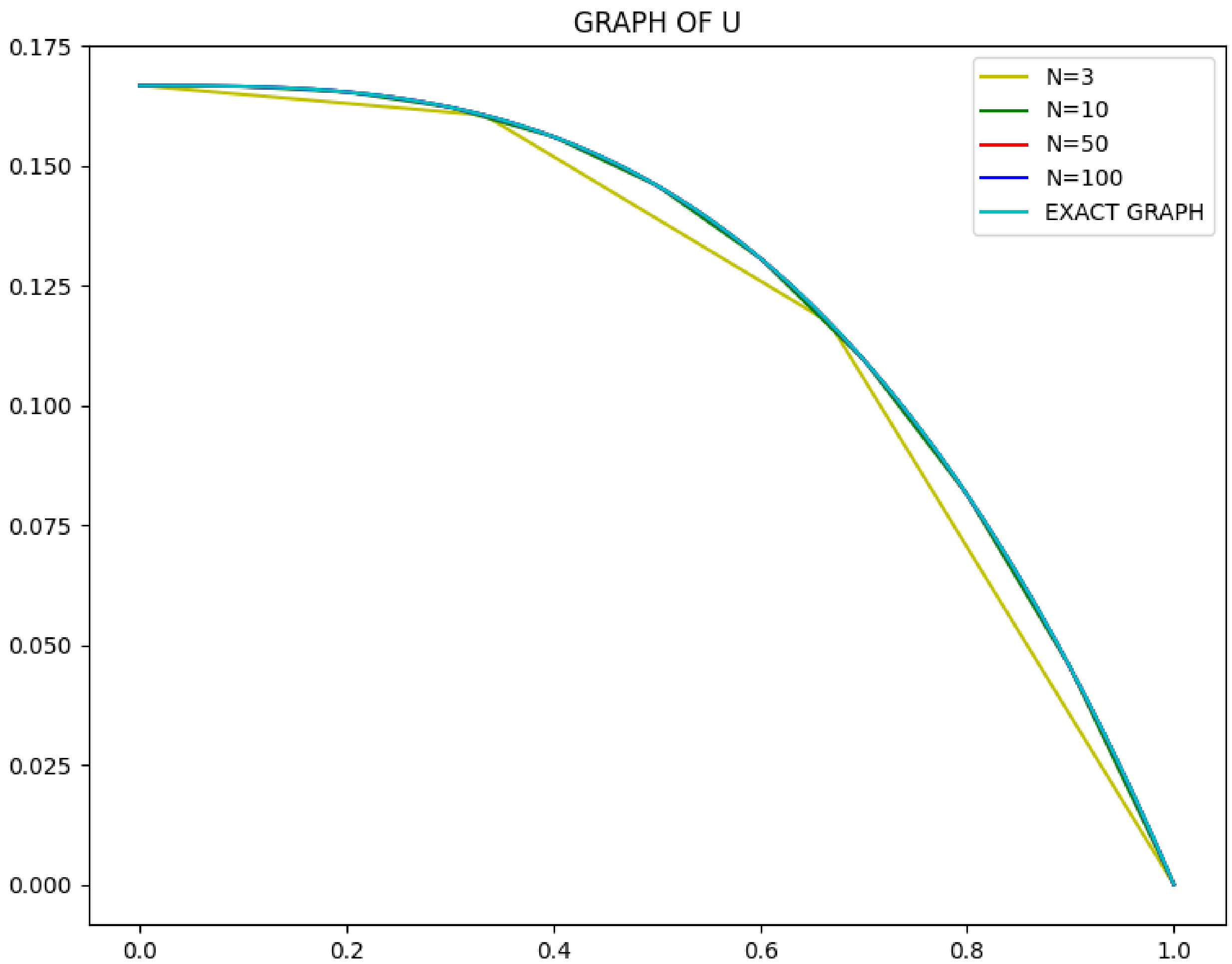
```
      K1[a+1][a+1] = K1[a+1][a+1] + ke[1][1]
      F1[a][0] = F1[a][0] + fe[0][0]
      F1[a+1][0]= F1[a+1][0] + fe[1][0]
ke=np.zeros((2,2))
fe=np.zeros((2,1))
# N=50
for a in range(len(xn2)-1):
    for i in range(0,2):
        for j in range(0,2):
           ke[i][j]=((-1)**(i+j))/h2
           f= lambda zeta: (h2/8) *q*
(h2*zeta+xn2[a]+xn2[a+1]) *(((-1)**(i+1) * zeta) +1)
           f=quad(f,-1,1)
           fe[i][0]=f[0]
    if a==n2-1:
        K2[a][a] = K2[a][a] + ke[0][0]
        F2[a][0] = F2[a][0] + fe[0][0]
    else:
      K2[a][a] = K2[a][a] + ke[0][0]
      K2[a][a+1] = K2[a][a+1] + ke[0][1]
      K2[a+1][a] = K2[a+1][a] + ke[1][0]
      K2[a+1][a+1] = K2[a+1][a+1] + ke[1][1]
      F2[a][0] = F2[a][0] + fe[0][0]
      F2[a+1][0]= F2[a+1][0] + fe[1][0]
ke=np.zeros((2,2))
fe=np.zeros((2,1))
# N=100
for a in range(len(xn3)-1):
    for i in range(0,2):
        for j in range(0,2):
           ke[i][j]=((-1)**(i+j))/h3
           f= lambda zeta: (h3/8) *q*
(h3*zeta+xn3[a]+xn3[a+1]) *(((-1)**(i+1) * zeta) +1)
           f=quad(f,-1,1)
           fe[i][0]=f[0]
    if a==n3-1:
        K3[a][a] = K3[a][a] + ke[0][0]
        F3[a][0] = F3[a][0] + fe[0][0]
    else:
      K3[a][a] = K3[a][a] + ke[0][0]
      K3[a][a+1] = K3[a][a+1] + ke[0][1]
      K3[a+1][a] = K3[a+1][a] + ke[1][0]
      K3[a+1][a+1] = K3[a+1][a+1] + ke[1][1]
      F3[a][0] = F3[a][0] + fe[0][0]
```

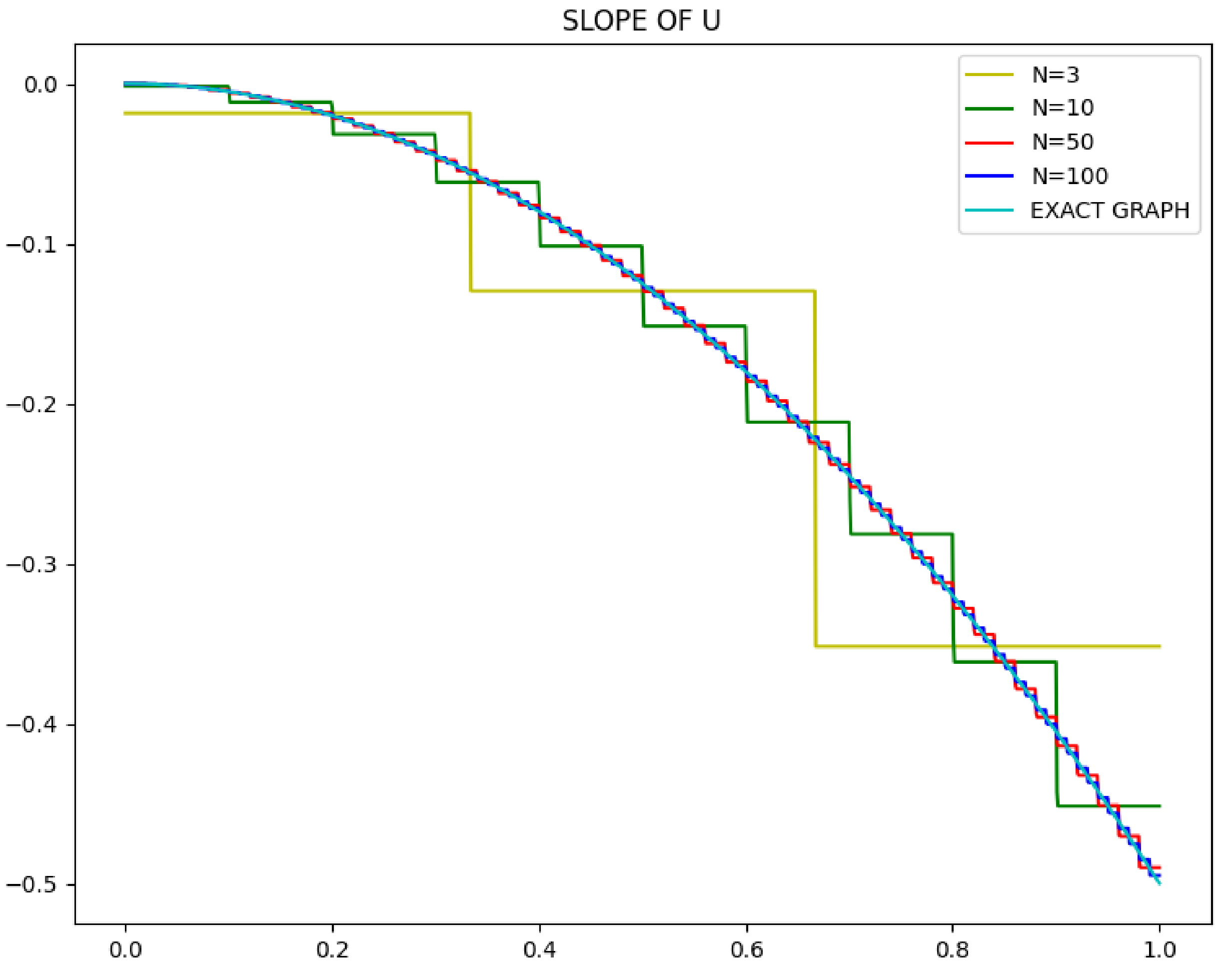
```
      F3[a+1][0]= F3[a+1][0] + fe[1][0]
ke=np.zeros((2,2))
fe=np.zeros((2,1))
# N=2
for a in range(len(xn4)-1):
    for i in range(0,2):
        for j in range(0,2):
           ke[i][j]=((-1)**(i+j))/h4
           f= lambda zeta: (h4/8) *q*
(h4*zeta+xn4[a]+xn4[a+1]) *(((-1)**(i+1) * zeta) +1)
           f=quad(f,-1,1)
           fe[i][0]=f[0]
    if a==n4-1:
        K4[a][a] = K4[a][a] + ke[0][0]
        F4[a][0] = F4[a][0] + fe[0][0]
    else:
      K4[a][a] = K4[a][a] + ke[0][0]
      K4[a][a+1] = K4[a][a+1] + ke[0][1]
      K4[a+1][a] = K4[a+1][a] + ke[1][0]
      K4[a+1][a+1] = K4[a+1][a+1] + ke[1][1]
      F4[a][0] = F4[a][0] + fe[0][0]
      F4[a+1][0]= F4[a+1][0] + fe[1][0]
##########CALCULATING THE D VALUE
d1=np.linalg.pinv(K1) @ F1
d2=np.linalg.pinv(K2) @ F2
d3=np.linalg.pinv(K3) @ F3
d4=np.linalg.pinv(K4) @ F4
###########CALCULATING APPROXIMATE SOLUTION AND
SLOPE
#N=10
uh1=np.zeros(1000)
for n in range(n1):
  for i in range(1000):
      uh1[i]=uh1[i]+d1[n]*N1[n,i]
uh_slope1=np.zeros(1000)
for i in range(1,1000):
      uh_slope1[i]=uh_slope1[i]+(uh1[i]-uh1[i-
1])/(x[i]-x[i-1])
      uh_slope1[0]=uh_slope1[1]
#N=50
uh2=np.zeros(1000)
for n in range(n2):
  for i in range(1000):
      uh2[i]=uh2[i]+d2[n]*N2[n,i]
```

```
uh_slope2=np.zeros(1000)
for i in range(1,1000):
      uh_slope2[i]=uh_slope2[i]+(uh2[i]-uh2[i-
1])/(x[i]-x[i-1])
      uh_slope2[0]=uh_slope2[1]
#N=100
uh3=np.zeros(1000)
for n in range(n3):
  for i in range(1000):
      uh3[i]=uh3[i]+d3[n]*N3[n,i]
uh_slope3=np.zeros(1000)
for i in range(1,1000):
      uh_slope3[i]=uh_slope3[i]+(uh3[i]-uh3[i-
1])/(x[i]-x[i-1])
      uh_slope3[0]=uh_slope3[1]
#N=2
uh4=np.zeros(1000)
for n in range(n4):
  for i in range(1000):
      uh4[i]=uh4[i]+d4[n]*N4[n,i]
uh_slope4=np.zeros(1000)
for i in range(1,1000):
      uh_slope4[i]=uh_slope4[i]+(uh4[i]-uh4[i-
1])/(x[i]-x[i-1])
      uh_slope4[0]=uh_slope4[1]
#############CALCULATING THE RELATIVE ERROR
for i in range(1000):
    Re1 = Re1 + np.abs(uh_slope1[i] -
uexact_slope[i]) / (q / 2)
    Re2 = Re2 + np.abs(uh_slope2[i] -
uexact_slope[i]) / (q / 2)
    Re3 = Re3 + np.abs(uh_slope3[i] -
uexact_slope[i]) / (q / 2)
print('SUM OF RELATIVE ERROR FOR N=10 : ',Re1)
print('SUM OF RELATIVE ERROR FOR N=50 : ',Re2)
print('SUM OF RELATIVE ERROR FOR N=100 : ',Re3)

#############PLOTTING THE GRAPH
#PLOT FOR U
plt.plot(x,uh4,'y',label='N=3')
plt.plot(x,uh1,'g',label='N=10')
plt.plot(x,uh2,'r',label='N=50')
plt.plot(x,uh3,'b',label='N=100')
plt.plot(x,u_exact,'c',label='EXACT GRAPH')
```

```
plt.legend()
plt.title('GRAPH OF U')
plt.show()
#PLOT OF SLOPE
plt.plot(x,uh_slope4,'y',label='N=3')
plt.plot(x,uh_slope1,'g',label='N=10')
plt.plot(x,uh_slope2,'r',label='N=50')
plt.plot(x,uh_slope3,'b',label='N=100')
plt.plot(x,uexact_slope,'c',label='EXACT GRAPH')
plt.legend()
plt.title('SLOPE OF U')
plt.show()
```


GRAPH OF U
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0.0
0.2
0.4
0.6
0.8
1.0
N=3
N=10
N=50
N=100
EXACT GRAPH




```
SUM OF RELATIVE ERROR FOR N=10 :  24.76415237058884
SUM OF RELATIVE ERROR FOR N=50 :  4.757501226148547
SUM OF RELATIVE ERROR FOR N=100 :  2.268513426517292
```

B> It can be seen the amount of point at which u^h and u_{exact} coincides increases with increase in 'n' values. For n=3 other than the nodal points, the graph coincides with exact solution at 2 points. After n=10, the approximate solution almost trace out the exact graph. Even for n=2 it can be found out that the values are exact at the nodes. For n=100 there is no difference in values for approximate and exact solution.

For the slope graph, initially even at the nodes the values are found to be different. With increase in 'n' value the amount of point at which $u^h_{,x}$ coincides with $u_{,x}$ also increases. For n=100 the graph of $u_{,x}$ is almost traced out by $u^h_{,x}$.

It can be seen that with increase in 'n' value sum of relative error goes on decreasing. This is a clear indication that with increase in 'n' value u^h can be a good and accurate approximation to u_{exact}.