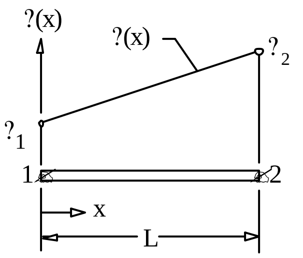
FINITE ELEMENT INTERPOLATION FUNCTIONS AND NUMERICAL INTEGRATION FOR ELEMENT MATRICES

ELEMENT INTERPOLATION FUNCTIONS

One-Dimensional Linear Element



Element distribution,

$$?(x) = N_1 ?_1 + N_2 ?_2$$

$$= ?N_1(x) \quad N_2(x)? \begin{matrix} ? ?_1 ? \\ ? \\ ? ?_2 ? \end{matrix}$$

$$= ?N? ??? $$

(1x2) (2x1)

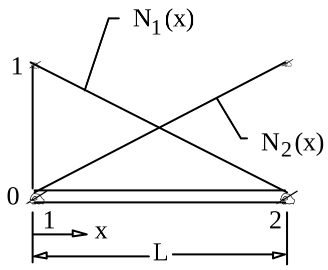
where $N_1(x)$ & $N_2(x)$ are element interpolation functions

$?_1$ & $?_2$ are element nodal quantities

247

ELEMENT INTERPOLATION FUNCTIONS

One-Dimensional Linear Element



$$?(x) = ?N_1(x) \quad N_2(x)? \; ?\,?_1? \; ?\,?_2?$$

Here the element inter-polation functions,

$$N_1(x) = 1 - \frac{x}{L}$$

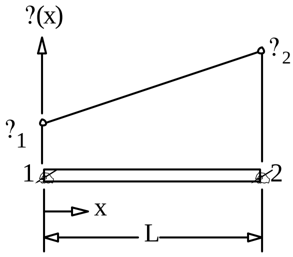
$$N_2(x) = \frac{x}{L}$$

with properties of,

$$N_i = \; ? \; 1 \quad \text{at node } i \qquad ? \; 0 \quad \text{at other nodes}$$

248

DERIVATION OF ELEMENT INTERPOLATION FUNCTIONS



Assume linear distribution,

$$?(x) = ?_1 + ?_2 x$$

$$= ?1 \quad x? \; ?\,?_1? \; ?\,?_2?$$

$$?(x) = ?P? \; ??? \qquad (1x2)\;(2x1)$$

Then apply BC’s,

$$??(x = 0)? \; ??(x = L)? = ??_1? \; ??_2? = ?1 \quad 0? \; ?1 \quad L? \; ??_1? \; ??_2?$$

Or,

$$??? = ?C? \; ??? \qquad (2x1) \quad (2x2)\;(2x1)$$

DERIVATION OF ELEMENT INTERPOLATION FUNCTIONS

Or
$$??? = ?C?^{-1}???$$
$$(2x1) \quad (2x2)\ (1x2)$$

From
$$?(x) = ?P? ??? = ?P? ?C?^{-1}??? = ?N? ???$$
$$(1x2)(2x1) \quad (1x2)\ (2x2)(2x1) \quad (1x2)\ (2x1)$$

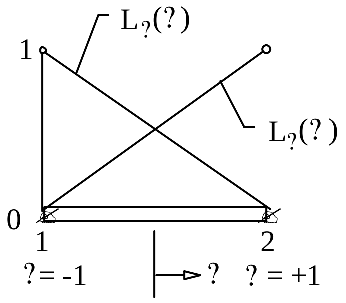
i.e.,
$$?N? = ?P? ?C?^{-1}$$
$$(1x2) \quad (1x2)\ (2x2)$$

Here,
$$?N? = ?1 \quad x? \begin{matrix} ?1 & 0? \\ ?-\frac{1}{L} & \frac{1}{L}? \end{matrix} = ?1-\frac{x}{L} \quad \frac{x}{L}?$$

Note: These element interpolation functions can be derived easily by using Algebraic Manipulation Programs.

TWO-NODE LINEAR ELEMENT

Element interpolation functions in natural coordinate



Let
$$x = \frac{1}{2} L (1 + ?)$$

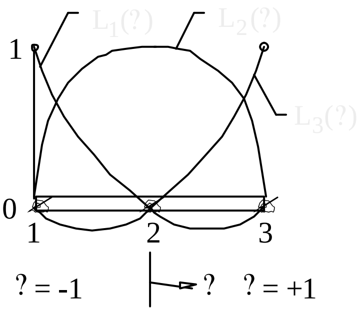
then the element distribution is,
$$?(?) = L_1 ?_1 + L_2 ?_2$$

where the interpolation functions,
$$L_1(?) = \frac{1}{2}(1 - ?)$$
$$L_2(?) = \frac{1}{2}(1 + ?)$$

with the same properties of,
$$L_i = \begin{cases} 1 & \text{at node } i \\ 0 & \text{at other nodes} \end{cases}$$

THREE-NODE QUADRATIC ELEMENT

Element interpolation functions in natural coordinate



$$L_1(?) = \frac{1}{2}(?^2 - ?)$$

$$L_2(?) = 1 - ?^2$$

$$L_3(?) = \frac{1}{2}(?^2 + ?)$$

Again, these element interpolation functions have the properties of,

$$L_i = \begin{cases} 1 & \text{at node } i \\ 0 & \text{at other nodes} \end{cases}$$

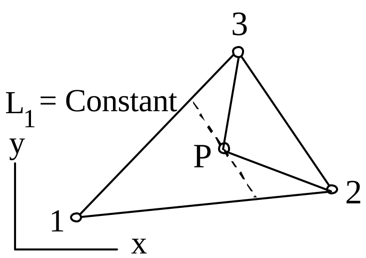
TRIANGULAR ELEMENT

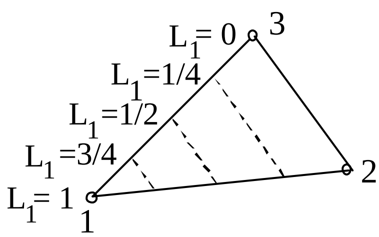
Element interpolation functions in area coordinate

$$? = L_1 ?_1 + L_2 ?_2 + L_3 ?_3$$

where $L_i, i = 1, 2, 3$ are the area coordinates, e.g.,

$$L_1 = \frac{\text{Area } P23}{\text{Area } 123}$$





TRIANGULAR ELEMENT

These area coordinates are related to the Cartesian coordinates as,

$$L_i = \frac{1}{2A}(a_i + b_i x + c_i y) \qquad i = 1, 2, 3$$

where

A = Area of triangle

$$= \frac{1}{2} ? x_2(y_3 - y_1) ? x_1(y_2 - y_3) ? x_3(y_1 - y_2) ?$$

and

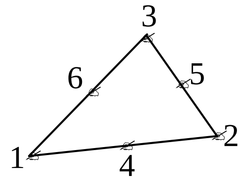
$a_1 = x_2 y_3 - x_3 y_2 \qquad b_1 = y_2 - y_3 \qquad c_1 = x_3 - x_2$

$a_2 = x_3 y_1 - x_1 y_3 \qquad b_2 = y_3 - y_1 \qquad c_2 = x_1 - x_3$

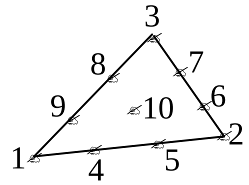
$a_3 = x_1 y_2 - x_2 y_1 \qquad b_3 = y_1 - y_2 \qquad c_3 = x_2 - x_1$

HIGHER-ORDER TRIANGULAR ELEMENTS

6-Node Element



10-Node Element



Corner: $N_i = L_i (2L_i - 1)$ $\qquad N_i = \frac{1}{2}(3L_i - 1)(3L_i - 2)$

Mid-side: $N_4 = 4 L_1 L_2$ $\qquad N_4 = \frac{9}{2} L_1 L_2 (3L_1 - 1)$

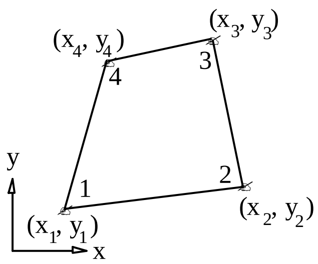
Etc. $\qquad N_5 = \frac{9}{2} L_1 L_2 (3L_1 - 2)$

Etc.

Centroid: $N_{10} = 27 L_1 L_2 L_3$

QUADRILATERAL ELEMENT

Four-node quadrilateral element is widely used because it provides higher solution accuracy compared to triangular element. However, there are several difficulties associated with it as follows,



1. Element distribution is first assummed in the form,

$$\phi(x, y) = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 xy$$

QUADRILATERAL ELEMENT

This results in very complex element interpolation functions $\lfloor N(x, y) \rfloor_{(1x4)}$.

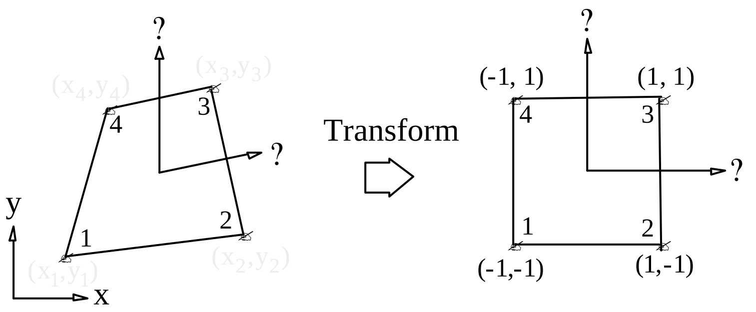
2. The element matrices are thus difficult to evaluate, e.g.,

$$[K] = \int_{\Omega^{(e)}} \left(\left\{\frac{\partial N}{\partial x}\right\}\left\{\frac{\partial N}{\partial x}\right\}^T + \left\{\frac{\partial N}{\partial y}\right\}\left\{\frac{\partial N}{\partial y}\right\}^T \right) d\Omega$$

i.e., need to take derivative of N then integrate over quad shape area.

To reduce these difficulties, the idea is to transform this quad shape into a square shape.

QUADRILATERAL ELEMENT



Local coordinates Natural coordinates

The relations between these two coordinate systems are,

QUADRILATERAL ELEMENT

$$x = \hat{N}_1 x_1 + \hat{N}_2 x_2 + \hat{N}_3 x_3 + \hat{N}_4 x_4 = \underset{(1x4)}{\lfloor \hat{N} \rfloor} \underset{(4x1)}{\{x\}}$$

$$y = \hat{N}_1 y_1 + \hat{N}_2 y_2 + \hat{N}_3 y_3 + \hat{N}_4 y_4 = \underset{(1x4)}{\lfloor \hat{N} \rfloor} \underset{(4x1)}{\{y\}}$$

where $\hat{N}_i, i = 1, 4$ are called shape functions defined by,

$$\hat{N}_1 = \frac{1}{4}(1 - ?)(1 - ?) \qquad \hat{N}_3 = \frac{1}{4}(1 + ?)(1 + ?)$$

$$\hat{N}_2 = \frac{1}{4}(1 + ?)(1 - ?) \qquad \hat{N}_4 = \frac{1}{4}(1 - ?)(1 + ?)$$

QUADRILATERAL ELEMENT

Since the element distribution is in the form,

$$? = N_1?_1 + N_2?_2 + N_3?_3 + N_4?_4 = ?N? ??? \quad (1x4)(4x1)$$

If the element interpolation functions, $N_i, i = 1,4$ are defined as the element shape functions, $\hat{N}_i, i = 1,4$ then the element is called "isoparametric" element (meaning the same parameters are used for geometry and dependent variable)

QUADRILATERAL ELEMENT

In conclusion, for isoparametric quad element,

$$? = N_1?_1 + N_2?_2 + N_3?_3 + N_4?_4 = ?N? ??? \quad (1x4)(4x1)$$

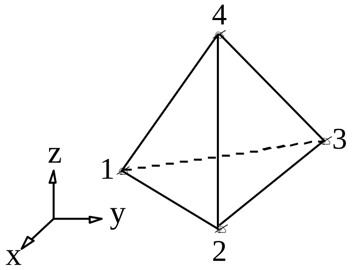
$$= ?N_1 \quad N_2 \quad N_3 \quad N_4?? \begin{matrix} ??_1? \\ ??_2? \\ ??_3? \\ ??_4? \end{matrix}$$

where the element interpolation functions are,

$$N_1 = \frac{1}{4}(1-?)(1-?) \qquad N_3 = \frac{1}{4}(1+?)(1+?)$$

$$N_2 = \frac{1}{4}(1+?)(1-?) \qquad N_4 = \frac{1}{4}(1-?)(1+?)$$

3-D TETRAHEDRAL ELEMENT



Assume element distribution,

$$?(x, y, z) \quad = \quad ?_1 + ?_2x + ?_3y + ?_4z$$

or, in form of element interpolation functions,

$$? \quad = \quad N_1?_1 + N_2?_2 + N_3?_3 + N_4?_4 \quad = \quad ?N? \;\; ??? $$
$$(1x4) \;\; (4x1)$$

TETRAHEDRAL ELEMENT

where

$$N_i \quad = \quad \frac{1}{6V}(a_i + b_ix + c_iy + d_iz) \qquad\qquad i = 1, 2, 3, 4$$

and

$$V \quad = \quad \text{Element volume} \quad = \quad \frac{1}{6}\begin{vmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ 1 & x_3 & y_3 & z_3 \\ 1 & x_4 & y_4 & z_4 \end{vmatrix}$$

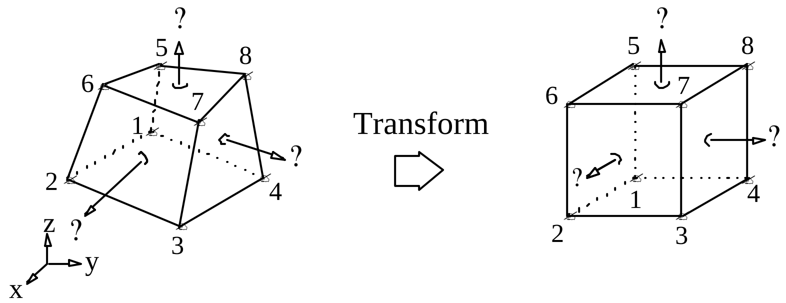
TETRAHEDRAL ELEMENT

$$a_1 = \begin{vmatrix} x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \\ x_4 & y_4 & z_4 \end{vmatrix} \qquad c_1 = -\begin{vmatrix} x_2 & 1 & z_2 \\ x_3 & 1 & z_3 \\ x_4 & 1 & z_4 \end{vmatrix}$$

$$b_1 = -\begin{vmatrix} 1 & y_2 & z_2 \\ 1 & y_3 & z_3 \\ 1 & y_4 & z_4 \end{vmatrix} \qquad d_1 = -\begin{vmatrix} x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \\ x_4 & y_4 & 1 \end{vmatrix}$$

other constants $a_i, b_i, c_i, d_i, \;\; i = 2,3,4$ are determined in the same way using cyclic permutation.

HEXAHEDRAL ELEMENT



Local coordinates   Natural coordinates

Element distribution is in the form,

$$? = \overset{8}{\underset{i=1}{?}} N_i \, ?_i = ?N? \;\; ??? $$

$$(1x8) \quad (8x1)$$

HEXAHEDRAL ELEMENT

$$? \;=\; \underset{(1x8)}{?N?}\;\underset{(8x1)}{???} \;=\; N_1?_1 + N_2?_2 + N_3?_3 + N_4?_4 + N_5?_5 + N_6?_6 + N_7?_7 + N_8?_8$$

where the element interpolation functions are,

$$N_1 = \frac{1}{8}(1-?)(1-?)(1-?) \qquad N_5 = \frac{1}{8}(1-?)(1-?)(1+?)$$

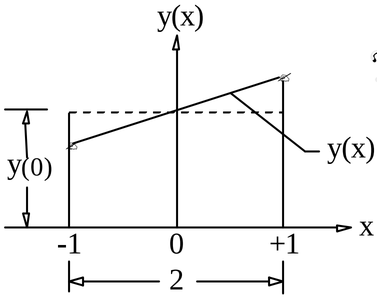
$$N_2 = \frac{1}{8}(1+?)(1-?)(1-?) \qquad N_6 = \frac{1}{8}(1+?)(1-?)(1+?)$$

$$N_3 = \frac{1}{8}(1+?)(1+?)(1-?) \qquad N_7 = \frac{1}{8}(1+?)(1+?)(1+?)$$

$$N_4 = \frac{1}{8}(1-?)(1+?)(1-?) \qquad N_8 = \frac{1}{8}(1-?)(1+?)(1+?)$$

NUMERICAL INTEGRATION REVIEW (GAUSS INTEGRATION FORMULAS)

Compute $?_{-1}^{1} y(x)\,dx$ for $y(x) = ax + b$.



From the figure,

$$?_{-1}^{1} y(x)\,dx \;=\; \underset{\text{Weight}}{2} \;*\; \underset{\text{Gauss point location}}{y(0)}$$

= Area under straight line

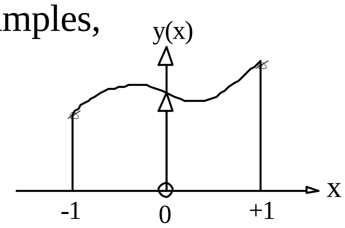
NUMERICAL INTEGRATION REVIEW

For general function y(x),

$$\int_{-1}^{1} y(x)\,dx \approx \sum_{i=1}^{NG} W_i\, y(x_i)$$

where NG = Number of Gauss points
W_i = Weights
x_i = Gauss point locations

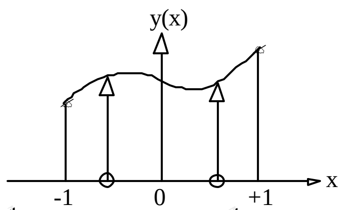
For examples,



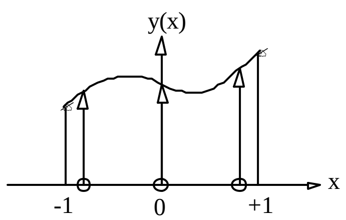
$$\int_{-1}^{1} y(x)\,dx \approx 2\,y(0)$$

NG	x_i	W_i
1	0	2

NUMERICAL INTEGRATION REVIEW



$$\int_{-1}^{1} y(x)\,dx = 1\,y\left(-\frac{1}{\sqrt{3}}\right) + 1\,y\left(+\frac{1}{\sqrt{3}}\right)$$



$$\int_{-1}^{1} y(x)\,dx = \frac{5}{9}\,y(-\sqrt{.6}) + \frac{8}{9}\,y(0) + \frac{5}{9}\,y(\sqrt{.6})$$

NG	x_i	W_i
2	$-\frac{1}{\sqrt{3}} = -.577$	1
	$+\frac{1}{\sqrt{3}} = +.577$	1
3	$-\sqrt{.6} = -.775$	$\frac{5}{9} = .555$
	0	$\frac{8}{9} = .888$
	$+\sqrt{.6} = +.775$	$\frac{5}{9} = .555$

NUMERICAL INTEGRATION REVIEW

Note that, the exact integration is obtained if the integrand, y(x), is a polynomial of order $2*NG - 1$ or less.

No. of Gauss pts. (NG)	Polynomial order (2∗NG - 1)	Curve y(x)
1	1	Linear
2	3	Cubic
3	5	5^{th} order

NUMERICAL INTEGRATION REVIEW

For double integration,

$$\int_{-1}^{1}\int_{-1}^{1} f(\xi,\eta)\, d\xi\, d\eta = \int_{-1}^{1} \left[\sum_{j=1}^{NG} W_j f(\xi,\eta_j) \right] d\xi$$

$$= \sum_{i=1}^{NG} \sum_{j=1}^{NG} W_i W_j f(\xi_i,\eta_j)$$

If 2 Gauss pts. in each direction are used, then

$$\int_{-1}^{1}\int_{-1}^{1} f(\xi,\eta)\, d\xi\, d\eta = W_1 W_1 f(\xi_1,\eta_1) + W_1 W_2 f(\xi_1,\eta_2)$$

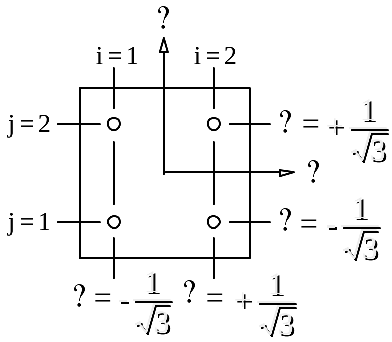
$$+ W_2 W_1 f(\xi_2,\eta_1) + W_2 W_2 f(\xi_2,\eta_2)$$

NUMERICAL INTEGRATION REVIEW

with proper Gauss weights and locations,

$$?_{-1}^{1} ?_{-1}^{1} f(?,?)\, d?\, d? = (1)(1)\, f(-\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}) + (1)(1)\, f(-\frac{1}{\sqrt{3}}, +\frac{1}{\sqrt{3}})$$

$$+ (1)(1)\, f(+\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}) + (1)(1)\, f(+\frac{1}{\sqrt{3}}, +\frac{1}{\sqrt{3}})$$

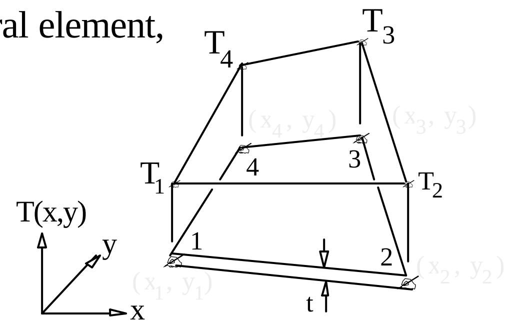


EVALUATION OF ELEMENT MATRICES USING NUMERICAL INTEGRATION

Example   Evaluate heat conduction matrix,

$$\underset{(4x4)}{[K_c]} = ?_A k \left(\underset{(4x1)}{\left\{\frac{?N}{?x}\right\}} \underset{(1x4)}{\left\{\frac{?N}{?x}\right\}} ? \underset{(1x4)}{\left\{\frac{?N}{?y}\right\}} \underset{(1x4)}{\left\{\frac{?N}{?y}\right\}} \right) t\, dA$$

for quadrilateral element,



EVALUATION OF CONDUCTION MATRIX

Note that the conduction matrix can be written in the form,

$$\underset{(4\times4)}{[K_c]} = \iint_A k \underset{(4\times2)}{[B]^T} \underset{(2\times4)}{[B]} t \, dx \, dy$$

where $[B] = [B(x, y)]$ is the temperature-gradient interpolation matrix,

$$\underset{(2\times1)}{\begin{Bmatrix} \dfrac{\partial T}{\partial x} \\ \dfrac{\partial T}{\partial y} \end{Bmatrix}} = \underset{(2\times4)}{\underbrace{\begin{bmatrix} \dfrac{\partial N_1}{\partial x} & \dfrac{\partial N_2}{\partial x} & \dfrac{\partial N_3}{\partial x} & \dfrac{\partial N_4}{\partial x} \\ \dfrac{\partial N_1}{\partial y} & \dfrac{\partial N_2}{\partial y} & \dfrac{\partial N_3}{\partial y} & \dfrac{\partial N_4}{\partial y} \end{bmatrix}}_{[B(x,\,y)]}} \underset{(4\times1)}{\begin{Bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{Bmatrix}}$$

EVALUATION OF CONDUCTION MATRIX

But the element temperature distribution was previously assummed in form of natural coordinates, ξ - η , i.e.,

$$T = N_1T_1 + N_2T_2 + N_3T_3 + N_4T_4 = \underset{(1\times4)}{[N(\xi, \eta)]} \underset{(1\times4)}{\{T\}}$$

Thus to derive matrix $[B]$, we apply chain rule,

$$\frac{\partial T}{\partial \xi} = \frac{\partial T}{\partial x}\frac{\partial x}{\partial \xi} + \frac{\partial T}{\partial y}\frac{\partial y}{\partial \xi}$$

and

$$\frac{\partial T}{\partial \eta} = \frac{\partial T}{\partial x}\frac{\partial x}{\partial \eta} + \frac{\partial T}{\partial y}\frac{\partial y}{\partial \eta}$$

EVALUATION OF CONDUCTION MATRIX

Or, in matrix form,

$$\begin{Bmatrix} \frac{?T}{??} \\ \frac{?T}{??} \end{Bmatrix} = \underbrace{\begin{bmatrix} \frac{?x}{??} & \frac{?y}{??} \\ \frac{?x}{??} & \frac{?y}{??} \end{bmatrix}}_{[J]} \begin{Bmatrix} \frac{?T}{?x} \\ \frac{?T}{?y} \end{Bmatrix}$$

(2x2)

where the Jacobian matrix,

$$[J] = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} = \begin{bmatrix} \frac{?x}{??} & \frac{?y}{??} \\ \frac{?x}{??} & \frac{?y}{??} \end{bmatrix}$$

EVALUATION OF CONDUCTION MATRIX

The Jacobian matrix,

$$[J] = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} = \begin{bmatrix} \frac{?}{??} \sum_{i?1}^{4} N_i x_i & \frac{?}{??} \sum_{i?1}^{4} N_i y_i \\ \frac{?}{??} \sum_{i?1}^{4} N_i x_i & \frac{?}{??} \sum_{i?1}^{4} N_i y_i \end{bmatrix}$$

As an example,

$$J_{11} = \frac{?N_1}{??} x_1 + \frac{?N_2}{??} x_2 + \frac{?N_3}{??} x_2 + \frac{?N_4}{??} x_4$$

$$= -\frac{1}{4}(1-?)x_1 + \frac{1}{4}(1-?)x_2 + \frac{1}{4}(1+?)x_3 - \frac{1}{4}(1+?)x_4$$

EVALUATION OF CONDUCTION MATRIX

Thus the temperature gradients wrt. x and y are,

$$\begin{Bmatrix} \dfrac{\partial T}{\partial x} \\ \dfrac{\partial T}{\partial y} \end{Bmatrix} = [J]^{-1} \begin{Bmatrix} \dfrac{\partial T}{\partial \xi} \\ \dfrac{\partial T}{\partial \eta} \end{Bmatrix} = [J^*] \begin{Bmatrix} \dfrac{\partial T}{\partial \xi} \\ \dfrac{\partial T}{\partial \eta} \end{Bmatrix}$$

where the Jacobian matrix inverse, $[J]^{-1}$, is,

$$[J]^{-1} = \begin{bmatrix} J^*_{11} & J^*_{12} \\ J^*_{21} & J^*_{22} \end{bmatrix} = \frac{1}{|J|} \begin{bmatrix} J_{22} & -J_{12} \\ -J_{21} & J_{11} \end{bmatrix}$$

EVALUATION OF CONDUCTION MATRIX

Thus,

$$\begin{Bmatrix} \dfrac{\partial T}{\partial x} \\ \dfrac{\partial T}{\partial y} \end{Bmatrix} = \begin{bmatrix} J^*_{11} & J^*_{12} \\ J^*_{21} & J^*_{22} \end{bmatrix} \begin{Bmatrix} \dfrac{\partial T}{\partial \xi} \\ \dfrac{\partial T}{\partial \eta} \end{Bmatrix}$$

$$= \underbrace{\begin{bmatrix} J^*_{11} & J^*_{12} \\ J^*_{21} & J^*_{22} \end{bmatrix} \begin{bmatrix} \dfrac{\partial N_1}{\partial \xi} & \dfrac{\partial N_2}{\partial \xi} & \dfrac{\partial N_3}{\partial \xi} & \dfrac{\partial N_4}{\partial \xi} \\ \dfrac{\partial N_1}{\partial \eta} & \dfrac{\partial N_2}{\partial \eta} & \dfrac{\partial N_3}{\partial \eta} & \dfrac{\partial N_4}{\partial \eta} \end{bmatrix}}_{[B(\xi,\eta)]} \begin{Bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{Bmatrix}$$

$$= [B(\xi,\eta)]\{T\}$$

EVALUATION OF CONDUCTION MATRIX

i.e., the temperature-gradient interpolation matrix, ?B?, is now known,

$$?B(?,?)? \;=\; \begin{matrix} ?B_{11} & B_{12} & B_{13} & B_{14}? \\ ?B_{21} & B_{22} & B_{23} & B_{24}? \end{matrix}$$

$$= \begin{matrix} ?J^{*}_{11} & J^{*}_{12}? \\ ?J^{*}_{21} & J^{*}_{22}? \end{matrix} \; \begin{matrix} ?\frac{?N_1}{??} & \frac{?N_2}{??} & \frac{?N_3}{??} & \frac{?N_4}{??}? \\ ?\frac{?N_1}{??} & \frac{?N_2}{??} & \frac{?N_3}{??} & \frac{?N_4}{??}? \end{matrix}$$

EVALUATION OF CONDUCTION MATRIX

As an example,

$$B_{11} = J^{*}_{11}\frac{?N_1}{??} + J^{*}_{12}\frac{?N_1}{??}$$

$$B_{11} = \left(\frac{J_{22}}{|J|}\right)\left(-\frac{1}{4}(1-?)\right) + \left(-\frac{J_{12}}{|J|}\right)\left(-\frac{1}{4}(1-?)\right) = B_{11}(?,?)$$

Thus, the conduction matrix can be now evaluated,

$$\underset{(4x4)}{?K_c?} = ?^{1}_{-1} ?^{1}_{-1} k \underset{(4x2)}{?B(?,?)?^{T}} \underset{(2x4)}{?B(?,?)?} t\,|J|\,d?\,d?$$

where $|J| d? \, d? = dx\, dy$. After applying Gauss integration, the conduction matrix becomes,

EVALUATION OF ELEMENT MATRICES

Element conduction matrix,

$$\underset{(4\times4)}{[K_c]} = \sum_{i=1}^{NG}\sum_{j=1}^{NG} W_i\, W_j\, k\, \underset{(4\times2)}{[B(\xi_i,\eta_j)]^T}\, \underset{(2\times4)}{[B(\xi_i,\eta_j)]}\, t\, |J(\xi_i,\eta_j)|$$

Similarly, other element matrices can be evaluated, such as the load vector from internal heat generation,

$$\underset{(4\times1)}{\{Q_Q\}} = \int_A Q\, \underset{(4\times1)}{\{N\}}\, t\, dx\, dy$$

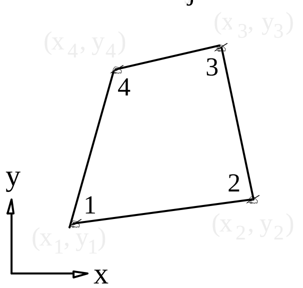
$$= \sum_{i=1}^{NG}\sum_{j=1}^{NG} W_i\, W_j\, Q\, \underset{(4\times1)}{\{N(\xi_i,\eta_j)\}}\, t\, |J(\xi_i,\eta_j)|$$

etc.

DEVELOPMENT OF COMPUTER PROGRAM

Write a computer program to evaluate the conduction matrix,

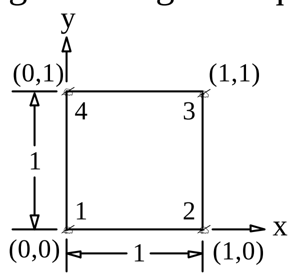
$$[K_c] = \sum_{i=1}^{NG}\sum_{j=1}^{NG} W_i\, W_j\, [B(\xi_i,\eta_j)]^T [B(\xi_i,\eta_j)]\, |J(\xi_i,\eta_j)|$$



for isoparametric quadrilateral element, using 1, 2, and 3 Gauss points in each direction.

DEVELOPMENT OF COMPUTER PROGRAM

Then check the program using unit square element,



which has the exact solution of,

$$[K_c] = \begin{bmatrix} .6667 & -.1667 & -.3333 & -.1667 \\ & .6667 & -.1667 & -.3333 \\ & & .6667 & -.1667 \\ \text{Sym} & & & .6667 \end{bmatrix}$$

DEVELOPMENT OF COMPUTER PROGRAM

Main Program

<u>Input</u>: Read in nodal coordinates x_i, y_i, i = 1, 2, 3, 4

<u>Output</u>: Print out the computed conduction matrix

↓

Subroutine KCMAT

- Set up Weights and Gauss point locations
- Call Subroutine *BJ* for $[B]$ and $[J]$ matrices
- Complete $[K_c]$ and return to *Main Program*

↓

Subroutine BJ

- Compute $[B]$, $[J]$, $[J]^{-1}$, $|J|$ at each Gauss point and return these values to Subroutine *KCMAT*

COMPUTER PROGRAM LISTING

```
      PROGRAM  QUAD
C
C      A PROGRAM TO COMPUTE CONDUCTION MATRIX FOR 4-NODE QUADRI-
C      LATERAL FINITE ELEMENT USING 1, 2 AND 3 GAUSS POINTS IN
C      EACH DIRECTION FOR NUMERICAL INTEGRATION.
C
      DIMENSION  X(4), Y(4), AKC(4,4)
      DATA  X/ 0., 1., 1., 0./
      DATA  Y/ 0., 0., 1., 1./
C
      OPEN(UNIT=7, FILE='KCMAT.OUT', STATUS='NEW')
C
C      LOOP FOR NUMERICAL INTEGRATION WITH 1, 2, 3 GAUSS POINTS:
C
      DO 10  L=1,3
      NG = L
C
C      COMPUTE CONDUCTION MATRIX:
C
      CALL KCMAT(NG, X, Y, AKC)
C
      WRITE(7,100)  NG
  100 FORMAT(//, '  CONDUCTION MATRIX USING', I2,
     *   ' GAUSS POINT(S) IN EACH DIRECTION:', /)
      DO 20  I=1,4
      WRITE(7,200)  (AKC(I,J), J=1,4)
      IF(I.LE.3)  WRITE(7,300)
  300 FORMAT(2X, '[',        57X, ']')
  200 FORMAT(2X, '[', 4E14.6, 1X, ']')
   20 CONTINUE
   10 CONTINUE
C
      STOP
      END
C
C---------------------------------------------------------------
C
      SUBROUTINE KCMAT(NG, X, Y, AKC)
C
C      COMPUTE CONDUCTION MATRIX FOR 4-NODE QUADRILATERAL ELEMENT
C
      DIMENSION  X(4), Y(4), AKC(4,4)
      DIMENSION  XG(3,3), WG(3,3), BMAT(2,4)
C
      DO 10  I=1,4
      DO 10  J=1,4
      AKC(I,J) = 0.
   10 CONTINUE
C
      DO 20  I=1,3
      DO 20  J=1,3
      XG(I,J) = 0.
```

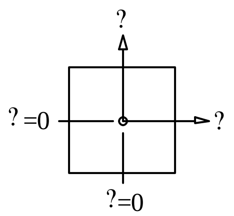

COMPUTER PROGRAM LISTING

```
      WG(I,J) = 0.
   20 CONTINUE
C
C      ASSIGN GAUSS POINT LOCATIONS  XG(I,J)  AND THEIR WEIGHTS
C      WG(I,J)  WHERE THE INDEX  I  DENOTES THE GAUSS POINT
C      LOCATIONS AND THE INDEX  J  DENOTES THE CASES FOR THE
C      NUMBER OF GAUSS POINTS BEING CONSIDERED:
C
      XG(1,1) = 0.
      WG(1,1) = 2.
C
      XG(1,2) = -.5773502691896
      XG(2,2) = +.5773502691896
      WG(1,2) = 1.
      WG(2,2) = 1.
C
      XG(1,3) = -.7745966692415
      XG(2,3) = 0.
      XG(3,3) = +.7745966692415
      WG(1,3) =  .5555555555555
      WG(2,3) =  .8888888888888
      WG(3,3) =  .5555555555555
C
C      COMPUTE CONDUCTION MATRIX:
C
      DO 30  I=1,4
      DO 30  J=1,4
C
C      LOOP OVER NUMBER OF GAUSS POINTS FOR SUMMATION:
C
      DO 30  K=1,NG
      DO 30  L=1,NG
C
C      COMPUTE MATRIX [B] AND DETERMINANT OF JACOBIAN |J|:
C
      CALL BJ(NG, K, L, X, Y, XG, BMAT, DETJAC)
C
      AKC(I,J) = AKC(I,J) + WG(K,NG)*WG(L,NG)*
     1   (BMAT(1,I)*BMAT(1,J) + BMAT(2,I)*BMAT(2,J))*DETJAC
   30 CONTINUE
C
C      COMPLETE THE CONDUCTION MATRIX BY SYMMETRY:
C
      DO 40  I=1,4
      DO 40  J=I,4
      AKC(J,I) = AKC(I,J)
   40 CONTINUE
C
      RETURN
      END
C
C---------------------------------------------------------------
```

COMPUTER PROGRAM LISTING

```
C
      SUBROUTINE BJ(NG, K, L, X, Y, XG, BMAT, DETJAC)
C
C     COMPUTE THE TEMPERATURE GRADIENT INTERPOLATION MATRIX [B]
C      AND THE DETERMINANT OF JACOBIAN |J|
C
      DIMENSION  X(4), Y(4), XG(3,3), BMAT(2,4)
      DIMENSION  DNDA(4), DNDB(4), AJ(2,2), AJI(2,2)
C
      A = XG(K,NG)
      B = XG(L,NG)
C
      AP = 1. + A
      AM = 1. - A
      BP = 1. + B
      BM = 1. - B
C
C     COMPUTE DERIVATIVES OF ELEMENT INTERPOLATION FUNCTIONS:
C
      DNDA(1) = -.25*BM
      DNDA(2) = -DNDA(1)
      DNDA(3) =  .25*BP
      DNDA(4) = -DNDA(3)
C
      DNDB(1) = -.25*AM
      DNDB(2) = -.25*AP
      DNDB(3) = -DNDB(2)
      DNDB(4) = -DNDB(1)
C
C     COMPUTE JACOBIAN MATRIX [J], ITS DETERMINANT AND INVERSE:
C
      AJ(1,1)=DNDA(1)*X(1)+DNDA(2)*X(2)+DNDA(3)*X(3)+DNDA(4)*X(4)
      AJ(1,2)=DNDA(1)*Y(1)+DNDA(2)*Y(2)+DNDA(3)*Y(3)+DNDA(4)*Y(4)
      AJ(2,1)=DNDB(1)*X(1)+DNDB(2)*X(2)+DNDB(3)*X(3)+DNDB(4)*X(4)
      AJ(2,2)=DNDB(1)*Y(1)+DNDB(2)*Y(2)+DNDB(3)*Y(3)+DNDB(4)*Y(4)
C
      DETJAC = AJ(1,1)*AJ(2,2) - AJ(2,1)*AJ(1,2)
C
      AJI(1,1) =  AJ(2,2)/DETJAC
      AJI(1,2) = -AJ(1,2)/DETJAC
      AJI(2,1) = -AJ(2,1)/DETJAC
      AJI(2,2) =  AJ(1,1)/DETJAC
C
C     COMPUTE MATRIX [B]:
C
      DO 10  I=1,2
      DO 10  J=1,4
      BMAT(I,J) = AJI(I,1)*DNDA(J) + AJI(I,2)*DNDB(J)
   10 CONTINUE
C
      RETURN
      END
```


RESULTS FROM COMPUTER PROGRAM

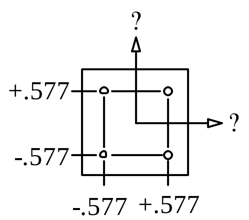


NG = 1

```
CONDUCTION MATRIX USING 1 GAUSS POINT(S) IN EACH DIRECTION:

[    .500000E+00    .000000E+00  -.500000E+00    .000000E+00 ]
[                                                            ]
[    .000000E+00    .500000E+00    .000000E+00  -.500000E+00 ]
[                                                            ]
[   -.500000E+00    .000000E+00    .500000E+00    .000000E+00 ]
[                                                            ]
[    .000000E+00  -.500000E+00    .000000E+00    .500000E+00 ]
```

Overall error ? 30%



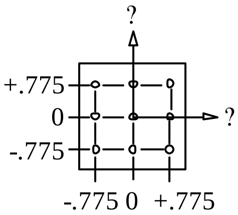
NG = 2

```
CONDUCTION MATRIX USING 2 GAUSS POINT(S) IN EACH DIRECTION:

[    .666667E+00  -.166667E+00  -.333333E+00  -.166667E+00 ]
[                                                          ]
[   -.166667E+00    .666667E+00  -.166667E+00  -.333333E+00 ]
[                                                          ]
[   -.333333E+00  -.166667E+00    .666667E+00  -.166667E+00 ]
[                                                          ]
[   -.166667E+00  -.333333E+00  -.166667E+00    .666667E+00 ]
```

Exact solution

RESULTS FROM COMPUTER PROGRAM



NG = 3

```
CONDUCTION MATRIX USING 3 GAUSS POINT(S) IN EACH DIRECTION:

[    .666667E+00  -.166667E+00  -.333333E+00  -.166667E+00 ]
[                                                          ]
[   -.166667E+00   .666667E+00  -.166667E+00  -.333333E+00 ]
[                                                          ]
[   -.333333E+00  -.166667E+00   .666667E+00  -.166667E+00 ]
[                                                          ]
[   -.166667E+00  -.333333E+00  -.166667E+00   .666667E+00 ]
```

Exact solution

Notes:

- ✍ For rectangular element shape, NG ? 2 yields exact conduction matrix.
- ✍ For general quad shape, the more NG used, the higher solution accuracy.
- ✍ NG=2 is used in most of finite element programs