Chapter 3

Finite Element Formulation

As already discussed in section 1.3, finite element methods can be very useful as compared to original theoretical models while analyzing complex loading and boundary conditions. In, finite element methods, a structural domain is sub-divided into smaller sub-domains called *elements*. These elements are connected with each other through computational *nodes* on which solution is calculated and then interpolated over the domain. This topological map of elements is called a *mesh* or *grid*. The accuracy of such methods is directly proportional to the number of elements in the grid; larger the number of elements, greater would be the accuracy.

3.1 Approach of current research

For FE formulation of EHSPAT, a 2-D rectangular element with four corner nodes (see Figure 3.1) has been developed in current research. Numbers written in the figure are node numbers while l_1 and l_2 are length and width of element respectively.

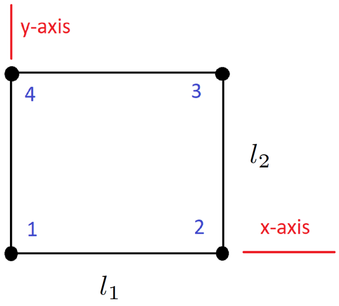

Figure 3.1: Proposed 2-D rectangular element with four corner nodes

Choice of the shape of element stems from the fact that chances of error while evaluating integrals over a rectangular domain are significantly less [33]. In some cases the geometry of domain is not rectangular and has to be approximated using quadrilateral elements, even then a transformation can be performed and results are calculated using rectangular element, then are again transformed back and mapped on the original shape. Current chapter outlines the procedure for developing elemental equations for an individual element. Once the equations are developed, these can be used to analyze any sandwich structure using appropriate computer code.

3.2 Displacement field of the element

We already have seen in chapter 2 that there are eleven primary variables in the formulation of EHSAPT and current element is based on it, therefore displacement field of the element can be written as follows.

$$\left\{U(x,y)\right\} = \left\{\begin{matrix} u^t(x,y) \\ v^t(x,y) \\ w^t(x,y) \\ u^b(x,y) \\ v^b(x,y) \\ w^b(x,y) \\ u^c(x,y) \\ v^c(x,y) \\ w^c(x,y) \\ \psi^c(x,y) \\ \phi^c(x,y) \end{matrix}\right\} \tag{3.1}$$

Dependence of above-written displacement field w.r.t time "t" is not shown here because the focus of current research is to develop a static finite element formulation.

3.2.1 Nodal displacements

In FEM, displacements at the nodes are calculated and then interpolated over the elemental domain Ω^{el} to get displacement distribution over it. These nodal displacements have to be defined in the FE formulation. Based on displacement field given in equation

3.1 and assumptions of EHSAPT, following displacements can be defined at a single node for the proposed element.

$$\left\{U_i\right\} = \begin{Bmatrix} u_i^t \\ v_i^t \\ w_i^t \\ w_{,x_i}^t \\ w_{,y_i}^t \\ u_i^b \\ v_i^b \\ w_i^b \\ w_{,x_i}^b \\ w_{,y_i}^b \\ u_i^c \\ v_i^c \\ w_i^c \\ w_{,x_i}^c \\ w_{,y_i}^c \\ \psi_i^c \\ \phi_i^c \end{Bmatrix} \qquad where \qquad i = 1,2,3,4 \tag{3.2}$$

$u_i^{t,b,b}$, $v_i^{t,b,c}$ and $w_i^{t,b,c}$ are the in-plane displacements in x and y directions and transverse displacements in z direction respectively at the i^{th} node. $w_{,x_i}^{t,b}$ and ψ_i^c are the rotations about y-axis while $w_{,y_i}^{t,b}$ and ϕ_i^c are the rotations about x-axis at the i^{th} node. Proposed element has two rotations each about x-axis (ϕ^c and $w_{,y_i}^c$) and y-axis (ψ^c and $w_{,x_i}^c$) for core at each node. However physically there are only two rotations which are ψ^c and ϕ^c. Derivatives of w^c has no physical meaning and been additionally included to ensure that same shape function can be used for transverse displacements of face sheets and the core. As these derivatives has no physical meaning so these can be set to zero at the boundary conditions other than free. If derivatives of w^c are not included in the nodal displacements then w^c would exhibit C^0 continuity at the element boundaries while face sheets would exhibit C^1 continuity. This would lead to very inaccurate results which was observed during numerical experiments. Similar approach has also been used by Yuan [31].

Hence it can be concluded that there are a total of 17 degrees of freedom at each node for the proposed element. As there are four nodes in the proposed element, therefore, total degrees of freedom for each element would be 68 and size of elemental displacement vector would be 68 × 1 as shown.

$$\left\{U^{el}\right\} = \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{Bmatrix} \tag{3.3}$$

3.2.2 Shape function of displacement field of the element

Using equations 3.1-3.3, shape function of the displacement field for proposed element can be written as following. It gives the displacement filed of proposed element at any point (x, y) in the elemental domain Ω^{el}.

$$\left\{U(x,y)\right\} = \begin{Bmatrix} u^t(x,y) \\ v^t(x,y) \\ w^t(x,y) \\ u^b(x,y) \\ v^b(x,y) \\ w^b(x,y) \\ u^c(x,y) \\ v^c(x,y) \\ w^c(x,y) \\ \psi^c(x,y) \\ \phi^c(x,y) \end{Bmatrix} = \left[H(x,y)\right] \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{Bmatrix} \tag{3.4}$$

where x and y are the local co-ordinates of proposed element and $[\mathrm{H}(x, y)]$ is the interpolation matrix which contains interpolation functions used to interpolate the calculated values of displacements over the elemental domain Ω^{el}. Order of interpolation matrix $[\mathrm{H}(x, y)]$ is 11 × 68. Interpolation functions which populate the matrix $[\mathrm{H}(x, y)]$ are calculated individually for each degree of freedom depending whether its derivative is included in the FE formulation or not.

3.2.3 Interpolation functions

There are two types of displacements in current FE formulation based on the inclusion of derivatives of that particular displacement. Displacements whose derivatives are not included in the formulation are $u^{t,b,c}$, $v^{t,b,c}$, ϕ^c and ψ^c. Interpolation functions used to interpolate such variables are called as *Lagrange* interpolation functions. On the other hand, displacements whose derivatives are also included in the formulation are $w^{t,b,c}$. Interpolation functions used to interpolate such variables are called as *Hermite* interpolation functions.

3.2.3.1 Lagrange interpolation functions

Shape function for the displacements whose derivatives are not included can be written as [33]:-

$$c_1 + c_2x + c_3y + c_4xy \tag{3.5}$$

where $c_i (i = 1, 2, 3, 4)$ are the unknown constants which are calculated using boundary conditions which in this case are the values of that particular displacement at four different nodes of the proposed element. Co-ordinates of the nodes for proposed element are given in table 3.1.

Node number	x-coordinate	y-coordinate
1	0	0
2	l_1	0
3	l_1	l_2
4	0	l_2

Table 3.1: Local co-ordinates of the nodes of the proposed element

Using the shape function (equation 3.5) and associated boundary conditions at the boundaries of element (table 3.1), following *Lagrange* interpolation functions can be derived [1] one for each node.

$$\Phi_1 = \frac{(l_1 - x)(l_2 - y)}{l_1\, l_2} \tag{3.6a}$$

[1] An in-house MATLAB code was written to derive these interpolation functions

$$\Phi_2 = \frac{x\,(l_2 - y)}{l_1\,l_2} \tag{3.6b}$$

$$\Phi_3 = \frac{x\,y}{l_1\,l_2} \tag{3.6c}$$

$$\Phi_4 = \frac{(l_1 - x)\,y}{l_1\,l_2} \tag{3.6d}$$

where x and y are the local co-ordinates of the proposed element.

3.2.3.2 Hermite interpolation functions

Similarly shape function for the displacements whose derivatives *w.r.t* x and y are included can be written as [34]:-

$$\begin{gathered} c_1 + c_2x + c_3y + c_4x^2 + c_5y^2 + c_6xy + c_7x^3 \\ +c_8y^3 + c_9x^2y + c_{10}xy^2 + c_{11}x^3y + c_{12}xy^3 \end{gathered} \tag{3.7}$$

Same procedure can be used to derive [2] the 12 *Hermite* interpolation functions. These are four sets of interpolation functions for four different nodes as given in table 3.2.

Node number	Corresponding hermite int. function
1	$\Psi_1,\, \Psi_2,\, \Psi_3$
2	$\Psi_4,\, \Psi_5,\, \Psi_6$
3	$\Psi_7,\, \Psi_8,\, \Psi_9$
4	$\Psi_{10},\, \Psi_{11},\, \Psi_{12}$

Table 3.2: Hermite interpolation functions for each node of the proposed element

$$\Psi_1 = \frac{(l_1 - x)(l_2 - y)\left(l_1^2\,l_2^2 + l_1^2\,l_2\,y - 2\,l_1^2\,y^2 + l_1\,l_2^2\,x - 2\,l_2^2\,x^2\right)}{l_1^3\,l_2^3} \tag{3.8a}$$

$$\Psi_2 = \frac{x\,(l_1 - x)^2\,(l_2 - y)}{l_1^2\,l_2} \tag{3.8b}$$

$$\Psi_3 = \frac{y\,(l_1 - x)\,(l_2 - y)^2}{l_1\,l_2^2} \tag{3.8c}$$

$$\Psi_4 = -\,\frac{x\,(l_2 - y)\left(-l_1^2\,l_2\,y + 2\,l_1^2\,y^2 - 3\,l_1\,l_2^2\,x + 2\,l_2^2\,x^2\right)}{l_1^3\,l_2^3} \tag{3.8d}$$

$$\Psi_5 = -\,\frac{x^2\,(l_1 - x)\,(l_2 - y)}{l_1^2\,l_2} \tag{3.8e}$$

[2] An in-house MATLAB code was written to derive these interpolation functions

$$\Psi_6 = \frac{x\, y\, (l_2 - y)^2}{l_1\, l_2^2} \tag{3.8f}$$

$$\Psi_7 = -\frac{x\, y\, (l_1^2\, l_2^2 - 3\, l_1^2\, l_2\, y + 2\, l_1^2\, y^2 - 3\, l_1\, l_2^2\, x + 2\, l_2^2\, x^2)}{l_1^3\, l_2^3} \tag{3.8g}$$

$$\Psi_8 = -\frac{x^2\, y\, (l_1 - x)}{l_1^2\, l_2} \tag{3.8h}$$

$$\Psi_9 = -\frac{x\, y^2\, (l_2 - y)}{l_1\, l_2^2} \tag{3.8i}$$

$$\Psi_{10} = -\frac{y\, (l_1 - x)\, (-3\, l_1^2\, l_2\, y + 2 l_1^2\, y^2 - l_1\, l_2^2\, x + 2\, l_2^2\, x^2)}{l_1^3\, l_2^3} \tag{3.8j}$$

$$\Psi_{11} = \frac{x\, y\, (l_1 - x)^2}{l_1^2\, l_2} \tag{3.8k}$$

$$\Psi_{12} = -\frac{y^2\, (l_1 - x)\, (l_2 - y)}{l_1\, l_2^2} \tag{3.8l}$$

Where x and y are the local co-ordinates of proposed element. Ψ_1, Ψ_4, Ψ_7 and Ψ_{10} interpolates the calculated transverse displacements $w^{t,b,c}$ from nodes 1, 2, 3 and 4 respectively. Ψ_2, Ψ_5, Ψ_8 and Ψ_{11} interpolates the calculated rotations about y-axis $w_{,x}$ from nodes 1, 2, 3 and 4 respectively. Ψ_3, Ψ_6, Ψ_9 and Ψ_{12} interpolates the calculated rotations about x-axis $w_{,y}$ from nodes 1, 2, 3 and 4 respectively.

3.3 Interpolation matrix *[H]*

Interpolation matrix [H] is used to interpolate the calculated values of displacements at four nodes of the proposed element and the applied load onto the entire elemental domain Ω^{el}. It is a function of local coordinates and dimensions of the proposed element as already shown in section 3.2.2 and equations 3.6 and 3.8. As there are four nodes of the proposed element from which displacements are to be interpolated, therefore, this matrix can be divided into four sub-matrices as shown in equation 3.9. Each sub-matrix corresponds to a same numbered node and has same order i.e. 11×17. However, overall order of interpolation matrix is 11×68.

$$\Big[H(x,y)\Big] = \Big[H_1 \quad H_2 \quad H_3 \quad H_4\Big] \tag{3.9}$$

Position of interpolation functions in interpolation matrix is dependent on the order of variables in equation 3.4. Sub-matrices of the interpolation matrix can be written in notation form using the same order as following.

$$\begin{aligned}\left[H_i\right] = &\left[\begin{matrix}
\Phi_i & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
0 & \Phi_i & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
0 & 0 & \Psi_j & \Psi_{j+1} & \Psi_{j+2} & 0 & 0 & 0 & 0 & 0 & \cdots \\
0 & 0 & 0 & 0 & 0 & \Phi_i & 0 & 0 & 0 & 0 & \cdots \\
0 & 0 & 0 & 0 & 0 & 0 & \Phi_i & 0 & 0 & 0 & \cdots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \Psi_j & \Psi_{j+1} & \Psi_{j+2} & \cdots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots
\end{matrix}\right. \\
&\left.\begin{matrix}
\cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\cdots & \Phi_i & 0 & 0 & 0 & 0 & 0 & 0 \\
\cdots & 0 & \Phi_i & 0 & 0 & 0 & 0 & 0 \\
\cdots & 0 & 0 & \Psi_j & \Psi_{j+1} & \Psi_{j+2} & 0 & 0 \\
\cdots & 0 & 0 & 0 & 0 & 0 & \Phi_i & 0 \\
\cdots & 0 & 0 & 0 & 0 & 0 & 0 & \Phi_i
\end{matrix}\right]\end{aligned} \tag{3.10}$$

where $i = 1, 2, 3, 4$ and $j = 3(i-1)+1$. $\left[H_i\right]$ interpolates the displacements of i^{th} node of the proposed element. Φ are the Lagrange and Ψ are the Hermite interpolation functions.

3.4 Strain field of the element

Equations 2.3 and 2.6 can be written in matrix form to get strain field of the face sheets and core as:-

$$\left\{\epsilon^{t,b}\right\} = \left[L^{t,b}\right]\left\{U(x,y)\right\} \tag{3.11a}$$

$$\left\{\epsilon^c\right\} = \left[L^c\right]\left\{U(x,y)\right\} \tag{3.11b}$$

where $\left[L\right]$ is the differential operator matrix which operates on the displacement field of element. As per EHSAPT formulation, there are three strains in the face sheets therefore order of $\left[L^{t,b}\right]$ is 3×11. Similarly as per EHSAPT formulation, there are six strains in the core therefore order of $\left[L^c\right]$ is 6×11.

From equation 3.4 we know that displacement field of element can be written as following.

$$\left\{U(x,y)\right\} = \left[H\right]\left\{U^{el}\right\} \tag{3.12}$$

By substituting equation 3.12 in 3.11, strain field of the element can be written as following.

$$\left\{\epsilon^{t,b}\right\} = \left[L^{t,b}\right]\left[H\right]\left\{U^{el}\right\} \tag{3.13a}$$

$$\left\{\epsilon^c\right\} = \left[L^c\right]\left[H\right]\left\{U^{el}\right\} \tag{3.13b}$$

3.4.1 Strain Interpolation matrix *[B]*

Now a new matrix called *Strain Interpolation matrix* [3] [B] can be defined using equation 3.14 as following. This matrix will be used to calculate elemental stiffness matrix in the subsequent section.

$$\left[B^{t,b}\right] = \left[L^{t,b}\right]\left[H\right] \tag{3.14a}$$

$$\left[B^c\right] = \left[L^c\right]\left[H\right] \tag{3.14b}$$

Order of $\left[B^{t,b}\right]$ is 3×68 while that of $\left[B^c\right]$ is 6×68. Finally, strain field of the element in form of strain interpolation matrix can be written as following.

$$\left\{\epsilon^{t,b}\right\} = \left[B^{t,b}\right]\left\{U^{el}\right\} \tag{3.15a}$$

$$\left\{\epsilon^c\right\} = \left[B^c\right]\left\{U^{el}\right\} \tag{3.15b}$$

[3] An in-house Mathematica code was written to calculate strain interpolation matrix [B]

Similarly, stresses can be written in terms of strain interpolation matrix using the constitutive relations as following.

$$\left\{\sigma^{t,b}\right\} = \left[C^{t,b}\right]\left[B^{t,b}\right]\left\{U^{el}\right\} \tag{3.16a}$$

$$\left\{\sigma^{c}\right\} = \left[C^{c}\right]\left[B^{c}\right]\left\{U^{el}\right\} \tag{3.16b}$$

3.5 Elemental equations

There are two widely used techniques to develop finite element equations of an FE model. One is *weak form* approach and other is *energy functional* approach. Both are equivalent to each other and yields the same elemental equations at the end.

3.5.1 Weak form approach

This approach uses the governing differential equations of the theoretical model and reduces the order of differentiation until it is evenly divided between primary variable(s) and the weighting function(s) thus making it so-called *"weak"*. The shortcoming of this approach is that it can not be used for those theoretical models whose order of governing differential equations is not even. Further, in current research there are eleven coupled partial differential equations with the same number of primary variables. Therefore, the application of this approach for finite element formulation of EHSAPT was not feasible.

3.5.2 Energy functional approach

This approach uses the strain field, interpolation matrix and constitutive relations of the element to develop elemental equations, therefore, for FE formulation of the EHSAPT, this technique was more feasible. As current FE formulation focuses on the static model, therefore, *principal of virtual work* was used whose statement is given as:-

$$\delta(W_I - W_E) = 0 \tag{3.17}$$

where W_I is the internal work done and W_E is the external work done given as following.

$$\delta W_I = \int_{V_e} \left[(\delta\epsilon^{t,b,c})^T \sigma^{t,b,c}\right] \, dV \tag{3.18a}$$

$$\delta W_E = \int_{\Omega^{el}} \left[\Big(\delta U(x,y) \Big)^T q \right] dxdy \tag{3.18b}$$

q in equation 3.18b is externally applied distributed load over the elemental domain Ω^{el}.

Substituting equations 3.18a and 3.18b in 3.17 will yield the following expression.

$$\int_{V_e} \left[(\delta\epsilon^{t,b,c})^T \sigma^{t,b,c} \right] dV - \int_{\Omega^{el}} \left[\Big(\delta U(x,y) \Big)^T q \right] dxdy = 0 \tag{3.19a}$$

Now substituting $\epsilon^{t,b,c}$, $\sigma^{t,b,c}$ and $U(x,y)$ in equation 3.19a will give the following form of principle of virtual work.

$$\begin{aligned}
&\int_{V_e} \left[\Big(\delta \Big\{ U^{el} \Big\} \Big)^T \Big[B^t \Big]^T \Big[C^t \Big] \Big[B^t \Big] \Big\{ U^{el} \Big\} \right] dV + \\
&\int_{V_e} \left[\Big(\delta \Big\{ U^{el} \Big\} \Big)^T \Big[B^b \Big]^T \Big[C^b \Big] \Big[B^b \Big] \Big\{ U^{el} \Big\} \right] dV + \\
&\int_{V_e} \left[\Big(\delta \Big\{ U^{el} \Big\} \Big)^T \Big[B^c \Big]^T \Big[C^c \Big] \Big[B^c \Big] \Big\{ U^{el} \Big\} \right] dV - \\
&\qquad\qquad \int_{\Omega^{el}} \left[\Big(\delta U^{el} \Big)^T \Big[H \Big]^T q \right] dxdy = 0
\end{aligned} \tag{3.19b}$$

Re-arranging equation 3.19b will yield the following.

$$\begin{aligned}
\delta \Big\{ U^{el} \Big\}^T \Bigg(&\int_{V_e} \Big[B^t \Big]^T \Big[C^t \Big] \Big[B^t \Big] \Big\{ U^{el} \Big\} \; dV + \int_{V_e} \Big[B^b \Big]^T \Big[C^b \Big] \Big[B^b \Big] \Big\{ U^{el} \Big\} \; dV + \\
&\int_{V_e} \Big[B^c \Big]^T \Big[C^c \Big] \Big[B^c \Big] \Big\{ U^{el} \Big\} \; dV - \int_{\Omega^{el}} \Big[H \Big]^T q \; dxdy \Bigg) = 0
\end{aligned} \tag{3.19c}$$

As per definition of the virtual displacement, $\delta \Big\{ U^{el} \Big\}^T \neq 0$ therefore, terms in bracket would be equal to zero which after some re-arrangement can be written as following.

$$\begin{aligned}
\Big\{ U^{el} \Big\} \Bigg(&\int_{V_e} \Big[B^t \Big]^T \Big[C^t \Big] \Big[B^t \Big] \; dV + \int_{V_e} \Big[B^b \Big]^T \Big[C^b \Big] \Big[B^b \Big] \; dV + \\
&\int_{V_e} \Big[B^c \Big]^T \Big[C^c \Big] \Big[B^c \Big] \; dV \Bigg) = \int_{\Omega^{el}} \Big[H \Big]^T q \; dxdy
\end{aligned} \tag{3.19d}$$

If equation 3.19d is observed closely then it has the form $\left\{F\right\} = \left[K\right]\left\{x\right\}$ where,

$$\left[K^{el}\right] = \int_{V_e}\left[B^t\right]^T\left[C^t\right]\left[B^t\right]\ dV + \int_{V_e}\left[B^b\right]^T\left[C^b\right]\left[B^b\right]\ dV + \int_{V_e}\left[B^c\right]^T\left[C^c\right]\left[B^c\right]\ dV \tag{3.20a}$$

$$\left\{F^{el}\right\} = \int_{\Omega^{el}}\left[H\right]^T q\ dxdy \tag{3.20b}$$

$$\left\{x^{el}\right\} = \left\{U^{el}\right\} \tag{3.20c}$$

3.5.3 Elemental Stiffness Matrix

In equation 3.20a, it can be observed that the elemental stiffness matrix is composed of three portions with contribution from each layer. Layer-wise contribution in elemental stiffness matrix can be written as follows.

$$\left[K^t\right] = \int_0^{l_1}\int_0^{l_2}\int_{+c}^{c+f^t}\left[B^t\right]^T\left[C^t\right]\left[B^t\right]dz\,dy\,dx \tag{3.21a}$$

$$\left[K^b\right] = \int_0^{l_1}\int_0^{l_2}\int_{-c-f^b}^{-c}\left[B^b\right]^T\left[C^b\right]\left[B^b\right]dz\,dy\,dx \tag{3.21b}$$

$$\left[K^c\right] = \int_0^{l_1}\int_0^{l_2}\int_{-c}^{+c}\left[B^c\right]^T\left[C^c\right]\left[B^c\right]dz\,dy\,dx \tag{3.21c}$$

Therefore, the final form of elemental stiffness matrix [4] of the proposed element can be written as follows.

$$\left[K^{el}\right] = \left[K^t\right] + \left[K^b\right] + \left[K^c\right] \tag{3.22}$$

3.5.4 Elemental Load vector

Similar to the stiffness matrix, elemental load vector of the proposed element also has a contribution from each individual layer and can be written in matrix form as follows.

[4] An in-house Mathematica code was written to calculate stiffness matrix of each layer in symbolic form and then these were exported to MATLAB for implementation

$$\left\{F^{el}\right\} = \int_0^{l_1}\int_0^{l_2} \left[H\right]^T \left[D_f^{t,b,c}\right]^T \left\{q^{t,b,c}\right\} dy\,dx \tag{3.23}$$

$\left\{q\right\}$ contains the distributed loads for different layers and it can be written as:-

$$\left\{q\right\} = \left\{\begin{matrix} Q_x \\ Q_y \\ Q_z \\ M_{,x} \\ M_{,y} \end{matrix}\right\} \tag{3.24}$$

where Q_x is the applied distributed load in x-direction, Q_y in y-direction and Q_z in z-direction. $M_{,x}$ is the applied distributed moment about y-axis and $M_{,y}$ about x-axis. $\left[D_f^{t,b,c}\right]$ is an operator which has been introduced to make the order of matrices consistent for multiplication and can be defined as following.

$$\left[D_f^{t}\right] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\partial}{\partial x} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\partial}{\partial y} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.25a}$$

$$\left[D_f^{b}\right] = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{\partial}{\partial x} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{\partial}{\partial y} & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.25b}$$

$$\left[D_f^{c}\right] = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.25c}$$

Any concentrated load on a node can be added directly in the global load vector at the relevant position.

3.6 Assembly of elements

Up til now, equations for an individual element has been developed, however, real-world problems cannot be analyzed using a single element. The accuracy of the analysis is directly proportional to the number of elements. Therefore, there are different methods which are used to assemble the global stiffness matrix and load vector. The one used in current research is *boolean matrix* method whose equation for a particular element can be written as:-

$$\left\{U^e\right\} = \left[A^e\right]\left\{U\right\} \tag{3.26}$$

where $\left\{U\right\}$ is the vector which contains all global displacements of the problem, $\left\{U^e\right\}$ is the vector which contains displacements of that particular element and $\left[A^e\right]$ is the boolean matrix for that particular element. This boolean matrix is populated only by *zeros* and *ones* depending upon the element being considered. After developing boolean matrices [5] for each element in the grid, following equations can be used to develop the global stiffness matrix and load vector of the problem.

$$\left[K\right] = \sum_{e=1}^{n}\left[A^e\right]^T\left[K^{el}\right]\left[A^e\right] \tag{3.27a}$$

$$\left\{F\right\} = \sum_{e=1}^{n}\left[A^e\right]^T\left\{F^{el}\right\} \tag{3.27b}$$

"n" is number of elements in the grid.

3.7 Equilibrium equation

After developing global stiffness matrix (see equation 3.27a) and load vector (see equation 3.27b), equilibrium equation using proposed element can be written as following which can be used to calculate the global displacements.

$$\left\{U\right\} = \left[K\right]^{-1}\left\{F\right\} \tag{3.28}$$

[5] A computer code can be written to generate a boolean matrix for each element in the grid