Computational Stiffness Method

When the stiffness method is employed in hand calculations, the stiffness matrix is established column-by-column by setting the degrees of freedom of the structure equal to one, one at a time. However, the stiffness method is even more powerful when implemented on the computer. In that case, the structure is essentially seen as an assembly of individual elements. As a result, the stiffness matrix and load vector are established by assembling, i.e., summing the contributions from those elements. That is the topic of this document. The finite element method is a natural name for this approach, but that name is reserved for an extension of the computational stiffness method, covered in another document on this website.

Element Configurations

Figure 1 shows schematically, from left to right, how an individual element contributes to the final structure. To understand the assembly of the final stiffness matrix and load vector it is helpful to label each "configuration," from Basic to Final:

- **Basic:** In this configuration the element has the minimum number of degrees of freedom (DOFs) to describe any element deformation but no rigid-body motion; the forces in this configuration are also independent.
- **Local:** This configuration has enough DOFs to fully describe deformation and rigid-body motion, but the DOFs are in the local coordinate system.
- **Global:** This element configuration is the same as the Local, but the DOFs are now aligned with the global coordinate system; i.e., the orientation of the element in the structure is accounted for.
- **All:** This is a structural configuration, in which absolutely all DOFs of the structure are included, even those fixed by boundary conditions.
- **Final:** In this structural configuration the boundary conditions, as well as other restraints and dependencies are introduced.

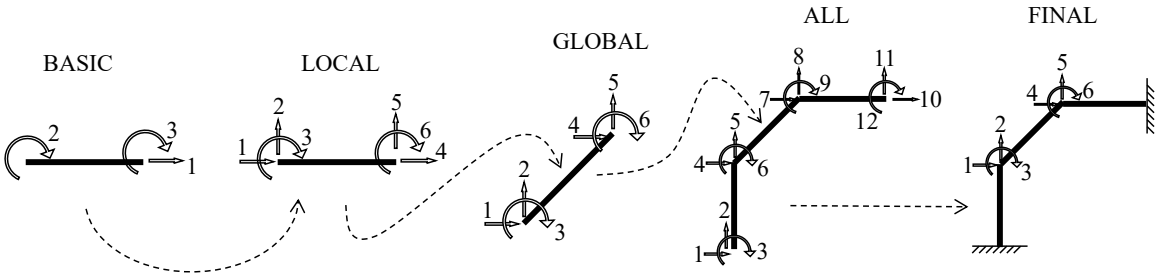

Figure 1: Element configurations.

Transformation Matrices

The configurations in Figure 1 are linked by transformation matrices. In other words, a transformation matrix defines the relationship between the DOFs in two configurations. Denoting by **u** the vector of DOFs the "kinematic compatibility" relationships are written

$$\mathbf{u}_b = \mathbf{T}_{bl}\mathbf{u}_l \tag{1}$$

$$\mathbf{u}_l = \mathbf{T}_{\mathrm{lg}}\mathbf{u}_g \tag{2}$$

$$\mathbf{u}_g = \mathbf{T}_{ga}\mathbf{u}_a \tag{3}$$

$$\mathbf{u}_a = \mathbf{T}_{af}\mathbf{u}_f \tag{4}$$

The subscripts are selected by the first letter of the name of the configuration it refers to. In this way, $\mathbf{T}_{bl}$ refers to the transformation matrix between the Basic and Local configurations, and so forth. Notice that the "basic-most" configuration appears on the left-hand side in Eqs. (1) to (4). The principle of virtual work is now invoked to determine the associated force relationships. When an element or the structure deforms then the virtual work in any of the configurations must be equal. Consider the local and global configurations as an example. By the principle of virtual displacements, equality of virtual work in the two configurations requires:

$$\mathbf{F}_g^T \delta\mathbf{u}_g - \mathbf{F}_l^T \delta\mathbf{u}_l = 0 \tag{5}$$

Introducing the transformation matrix between the two configurations yields:

$$\mathbf{F}_g^T \delta\mathbf{u}_g - \mathbf{F}_l^T \mathbf{T}_{lg}\delta\mathbf{u}_g = \left(\mathbf{F}_g^T - \mathbf{F}_l^T \mathbf{T}_{lg}\right)\delta\mathbf{u}_g = 0 \tag{6}$$

Because $\delta\mathbf{u}_g$ is an arbitrary virtual displacement pattern, the parenthesis in Eq. (6) must be zero for Eq. (6) to hold true. Consequently, because the parenthesis in Eq. (6) is a row vector:

$$\mathbf{F}_g^T - \mathbf{F}_l^T \mathbf{T}_{lg} = 0 \quad \Rightarrow \quad \mathbf{F}_g - \mathbf{T}_{\mathrm{lg}}^T \mathbf{F}_l = 0 \quad \Rightarrow \quad \mathbf{F}_g = \mathbf{T}_{\mathrm{lg}}^T \mathbf{F}_l \tag{7}$$

It is thus observed that the forces transform according to the transposed version of the earlier transformation matrices, with the "final-most" configuration on the left-hand side. Next, the relationship between the forces and displacements in each configuration is investigated. Initially, this relationship may seem obvious: in the stiffness method the forces are related to the displacement by the stiffness matrix. However, the question is how the stiffness matrix in each configuration is established. First, assume that the stiffness matrix in the local configuration is known. This represents the material law at the local level. Combined with the equilibrium relationship as derived in Eq. (7) and kinematic compatibility from Eq. (4) the stiffness matrix in the local configuration is:

$$\begin{aligned} \mathbf{F}_g &= \mathbf{T}_{lg}^{\mathrm{T}} \mathbf{F}_l \\ &= \mathbf{T}_{lg}^{\mathrm{T}} \mathbf{K}_l \mathbf{u}_l \\ &= \underbrace{\mathbf{T}_{lg}^{\mathrm{T}} \mathbf{K}_l \mathbf{T}_{lg}}_{\mathbf{K}_g} \mathbf{u}_g \end{aligned} \tag{8}$$

That is, the stiffness matrix in an "above" configuration is obtained by pre- and post-multiplying the stiffness matrix in the "below" configuration by the transformation matrix between the two configurations; e.g.:

$$\mathbf{K}_g = \mathbf{T}_{lg}^{\mathrm{T}} \mathbf{K}_l \mathbf{T}_{lg} \tag{9}$$

The form of the relationships derived in this section are summarized in Figure 2, using the Local and Global configurations as an example. Equilibrium is on the left-hand side and kinematic compatibility is on the right-hand side. This type of visualization of the three ingredients of any "structural boundary value problem" (equilibrium, compatibility, material law) is found in many of the documents posted on this website.

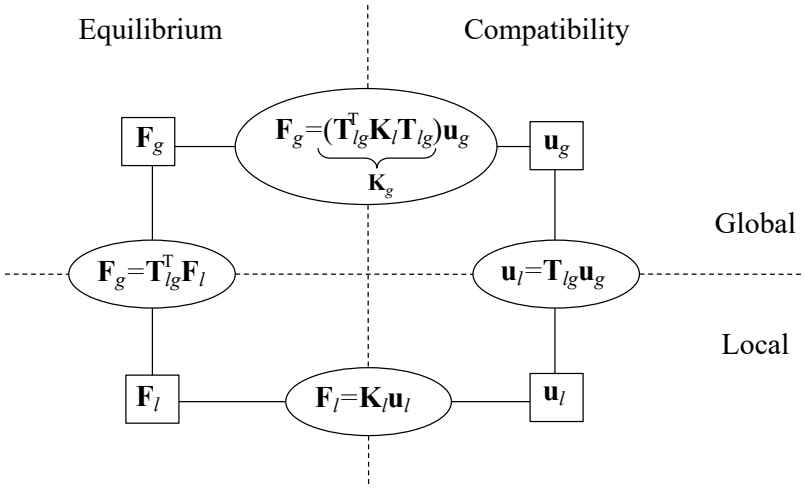

Figure 2: The ingredients of the boundary value problem in the Global configuration.

Figure 3 provides a similar overview for all the configurations, from Basic to Final. In the following sections the transformation matrices are established for the 2D frame element. It turns out that the transformation matrices are set up in a manner similar to the creation of the stiffness matrix in the classical stiffness method; DOFs at set equal to unity, one at a time. For some of the transformations it is possible to establish computer algorithms that are far more efficient than using transformation matrices for establishing the stiffness matrix and load vector. This is particularly the case with the transformation from Global to All, and the introduction of boundary conditions when going from the All to the Final configuration. While the use of transformation matrices is a pedagogical way to establish a consistent methodology, the more efficient algorithms are also mentioned.

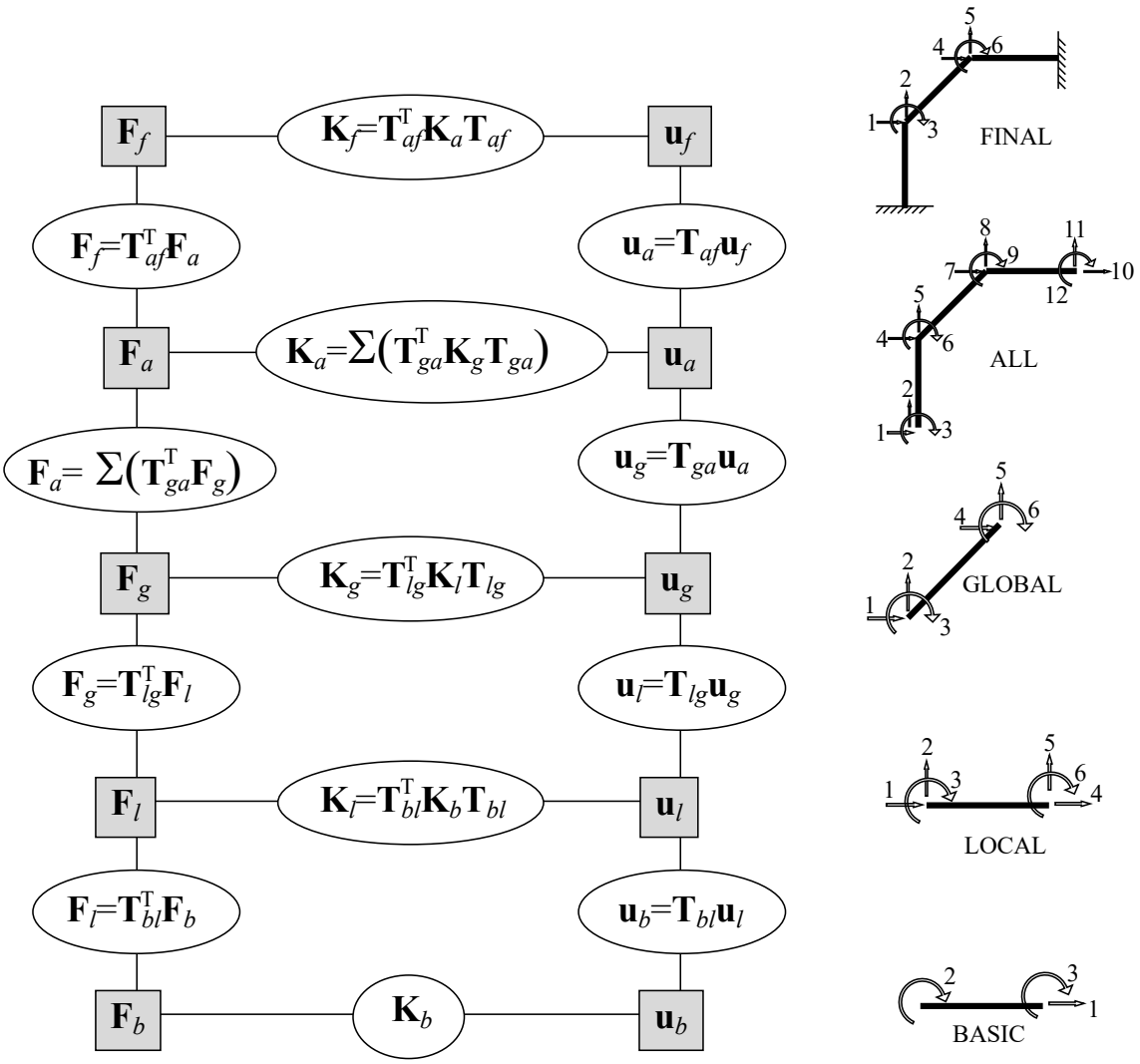

Figure 3: Overview of configurations and matrix relationships.

Basic to Local

The objective in this section is to establish the transformation matrix $\mathbf{T}_{bl}$ in Eq. (1). This is accomplished by setting the local DOFs equal to one, one at a time. By observing the resulting deformations in the basic configuration this establishes the corresponding column of $\mathbf{T}_{bl}$. As an example, consider the 2D frame element in Figure 3. Setting the first local DOF equal to unity, and all others equal to zero, implies a unit shortening of the element. Thus, the value of the first DOF in the basic configuration is -1, while the bending DOFs are zero. This forms the first column of $\mathbf{T}_{bl}$, which in its entirety reads

$$\mathbf{T}_{bl}=\left[\begin{array}{ccc|ccc} -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1/L & 1 & 0 & 1/L & 0 \\ 0 & -1/L & 0 & 0 & 1/L & 1 \end{array}\right] \tag{10}$$

The second and fifth column of this matrix are confusing at first; it helps to draw the deformed shape of the element in the local configuration and identify the end rotations compared with the straight line from one element end to the other.

Local to Global

The transformation into the global coordinate system is a function of the element orientation. For 2D elements, the orientation is represented by the angle θ between the local element axis and the horizontal axis. The columns of the transformation matrix are established by setting the global DOFs equal to unity, one at a time. For the 2D frame element in Figure 3 the result is

$$\mathbf{T}_{lg} = \left[\begin{array}{ccc:ccc} \cos(\theta) & \sin(\theta) & 0 & 0 & 0 & 0 \\ -\sin(\theta) & \cos(\theta) & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \hdashline 0 & 0 & 0 & \cos(\theta) & \sin(\theta) & 0 \\ 0 & 0 & 0 & -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \tag{11}$$

$\mathbf{T}_{lg}$ can be regarded as a matrix of "direction cosines," which is a concept from the more general field of coordinate transformations, described in the notes on vectors and geometry. The direction cosines are

$$c_x = \frac{\Delta x}{L}, \quad c_y = \frac{\Delta y}{L}, \quad c_z = \frac{\Delta z}{L} \tag{12}$$

where Δx is the x-direction distance between the element length. Similar with Δy and Δz. Hence, $\mathbf{T}_{lg}$ can also be written

$$\mathbf{T}_{lg} = \left[\begin{array}{ccc:ccc} c_x & c_y & 0 & 0 & 0 & 0 \\ -c_y & c_x & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \hdashline 0 & 0 & 0 & c_x & c_y & 0 \\ 0 & 0 & 0 & -c_y & c_x & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \tag{13}$$

In computer implementations of the stiffness matrix it is common to see $\mathbf{T}_{bl}$ and $\mathbf{T}_{lg}$ combined; combining Eqs. (10) and (13) yields

$$\mathbf{T}_{bg} = \mathbf{T}_{bl}\mathbf{T}_{lg} = \begin{bmatrix} -c_x & -c_y & 0 & c_x & c_y & 0 \\ \frac{c_y}{L} & -\frac{c_x}{L} & 1 & -\frac{c_y}{L} & \frac{c_x}{L} & 0 \\ \frac{c_y}{L} & -\frac{c_x}{L} & 0 & -\frac{c_y}{L} & \frac{c_x}{L} & 1 \end{bmatrix} \tag{14}$$

It can also be noted that the rotation matrix, **R**, that appears in the theory of coordinate transformations is not exactly the transformation $\mathbf{T}_{lg}$. Rather, the transformation matrix is the transpose of the rotation matrix, which in this case is the same as its inverse. As a result, for a 3D frame element with six DOFs at each end the transformation matrix from the local to the global coordinate system is a 12-by-12 matrix that reads

$$\mathbf{T}_{lg} = \begin{bmatrix} \mathbf{R}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{R}^T \end{bmatrix} \tag{15}$$

where **0** is a three-by-three sub-matrix of zeros and **R** is the rotation matrix established in the notes on vectors and geometry.

Global to All

The objective of this transformation is to link the element DOFs with the structural DOFs. The size of the transformation matrix $\mathbf{T}_{ga}$ is number of element DOFs by number of structural DOFs. $\mathbf{T}_{ga}$ is established in the same manner as all other transformation matrices: one DOF at a time in the All configuration is set equal to unity. For the structure in Figure 3, set DOF number four equal to one. This DOF corresponds to DOF number one for the element below. Hence, in this case the fourth column of the matrix $\mathbf{T}_{ga}$ has value one in the first row:

$$\mathbf{T}_{ga} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{16}$$

The presence of only zeros and ones is a general characteristic of the $\mathbf{T}_{ga}$ matrix, as it is for the next transformation.

All to Final

Fixed boundary conditions are introduced by means of the $\mathbf{T}_{af}$ transformation matrix. The dimension of this matrix is the number of DOFs in the All configuration by the number of

DOFs in the final configuration. The procedure to establish $\mathbf{T}_{af}$ is not new: each DOF in the final configuration is set equal to unity, one at a time. For the structure in Figure 3, setting the first DOF in the final configuration equal to one with the others zero implies that the fourth DOF in the All configuration is equal to one and the other zero. Thus, the first row in $\mathbf{T}_{af}$ has a one in the fourth position:

$$\mathbf{T}_{af} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{17}$$

For the stiffness matrix $\mathbf{K}_a$ the effect of pre- and post-multiplying by $\mathbf{T}_{af}$ is that the rows and columns corresponding to fixed DOFs removed.

Stiffness Matrix Assembly

Now, let's apply all those transformation matrices. The stiffness method requires the assembly of the stiffness matrix, $\mathbf{K}_f$, and the load vector, $\mathbf{F}_f$, for the structure. Once the system of equilibrium equations, $\mathbf{K}_f \mathbf{u}_f = \mathbf{F}_f$, are solved for $\mathbf{u}_f$, the element deformations, $\mathbf{u}_b$, and element forces, $\mathbf{F}_b$, i.e., bending moments, etc. are determined. To assemble the stiffness matrix, the first step is to determine $\mathbf{K}_b$, i.e., the stiffness matrix in the Basic configuration. For truss and frame elements, that is trivial. For example, for the frame element in Figure 3 the basic stiffness matrix is

$$\mathbf{k}_b = \begin{bmatrix} \frac{EA}{L} & 0 & 0 \\ 0 & \frac{4EI}{L} & \frac{2EI}{L} \\ 0 & \frac{2EI}{L} & \frac{4EI}{L} \end{bmatrix} \tag{18}$$

The final stiffness matrix for the structure, $\mathbf{K}_f$, is assembled by pre- and post-multiplying the previous stiffness matrix by transformation matrices:

$$\mathbf{K}_f = \mathbf{T}_{af}^T \left(\sum_{i=1}^{numEl} \mathbf{T}_{ga,i}^T \mathbf{T}_{\mathrm{lg},i}^T \mathbf{T}_{bl,i}^T \mathbf{k}_{b,i} \mathbf{T}_{bl,i} \mathbf{T}_{\mathrm{lg},i} \mathbf{T}_{ga,i} \right) \mathbf{T}_{af} \tag{19}$$

However, the application of $\mathbf{T}_{ga}$ and $\mathbf{T}_{af}$ is computationally inefficient. A far more computationally efficient method to insert $\mathbf{K}_g$ from each element into $\mathbf{K}_f$ is to employ an "ID array." This is described by Filippou and Fenves in their chapter entitled Methods of Analysis for Earthquake-Resistant Structures in the book on earthquake engineering edited by Bozorgnia and Bertero in 2004. An ID array implementation is also given in the G2 Python code posted on this website. For the structure in Figure 3, with node and element numbering shown in Figure 4, the ID array for the three elements, using Option 1 for the DOF numbering, is

$$\mathbf{ID} = \begin{bmatrix} element_1 \\ element_2 \\ element_3 \end{bmatrix} = \begin{bmatrix} node_1, node_2 \\ node_2, node_3 \\ node_3, node_4 \end{bmatrix} = \begin{bmatrix} id_1 \\ id_2 \\ id_3 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 4 & 5 & 6 & 7 & 8 & 9 \\ 7 & 8 & 9 & 10 & 11 & 12 \end{bmatrix} \tag{20}$$

That means a suitable pseudo code to insert the contribution from Element i into $\mathbf{K}_a$ is

$$\mathbf{K}_a(id_i,\ id_i) \mathrel{+}= \mathbf{K}_g \tag{21}$$

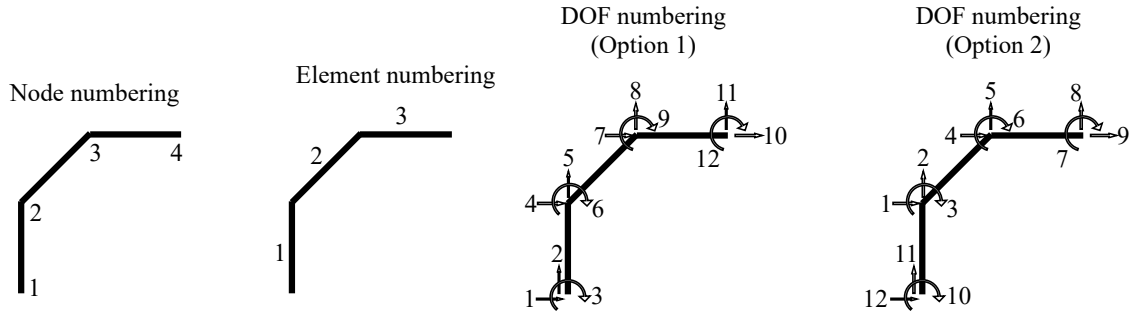

Figure 4: Numbering for ID array.

However, because the goal is to establish $\mathbf{K}_f$ it is more convenient to sort the free DOFs from the fixed DOFs. That means adopting Option 2 in Figure 4. In the G2 code that is done by establishing a DOF vector for the four nodes:

$$\mathbf{DOF} = \begin{bmatrix} node_1 \\ node_2 \\ node_3 \\ node_4 \end{bmatrix} = \begin{bmatrix} 12 & 11 & 10 \\ 1 & 2 & 3 \\ 4 & 5 & 6 \\ 9 & 8 & 7 \end{bmatrix} \tag{22}$$

where it is observed that the six free DOFs are numbered as the first six DOFs. In that case, the ID array is

$$\mathbf{ID} = \begin{bmatrix} 12 & 11 & 10 & 1 & 2 & 3 \\ 1 & 2 & 3 & 4 & 5 & 6 \\ 4 & 5 & 6 & 9 & 8 & 7 \end{bmatrix} \tag{23}$$

Assembling Load Vector & Recovering Element Forces

In the equilibrium equations **Ku**=**F** it may seem at first glance that **F** is the simplest quantity. However, in computational analysis, several **F**-vectors must be considered:

- Load vector at the structural level, containing nodal point loads applied along DOFs. This vector does not exist at the element level.
- Load vector at the structural level, containing element loads, e.g., distributed loads.
- Total load vector at the structural level, containing both previous cases
- Clamping force vector at the element level containing element loads, e.g., distributed loads.
- Force vector at the element level containing member-end forces.

At the structural level, the notation from the introductory document on the stiffness method is adopted. That means that $\mathbf{F}_f$ is the load vector that contains both nodal loads and element loads:

$$\mathbf{K}_f \mathbf{u}_f = \underbrace{\grave{\mathbf{F}}_f - \bar{\mathbf{F}}_f}_{\substack{\text{Total} \\ \text{load} \\ \text{vector}}} = \mathbf{F}_f \tag{24}$$

where $\grave{\mathbf{F}}_f$ are nodal loads and $\bar{\mathbf{F}}_f$ are clamping forces from element loads. The former does not really need an assembly procedure, because the point loads at the nodes are simply entered in the proper position of the load vector. In contrast, the element load vector, $\bar{\mathbf{F}}_f$, is assembled in a manner similar to the stiffness matrix:

$$\bar{\mathbf{F}}_f = \mathbf{T}_{af}^{\mathrm{T}} \left(\sum_{i=1}^{numEl} \mathbf{T}_{ga,i}^{\mathrm{T}} \mathbf{T}_{lg,i}^{\mathrm{T}} \mathbf{T}_{bl,i}^{\mathrm{T}} \bar{\mathbf{F}}_{b,i} \right) \tag{25}$$

where the element clamping forces are

$$\bar{\mathbf{F}}_b = \begin{Bmatrix} 0 \\ -\dfrac{qL^2}{12} \\ \dfrac{qL^2}{12} \end{Bmatrix} \tag{26}$$

for the element addressed in this document, when q is uniformly distributed load. Attention now turns to the recovery of element forces after the structural equilibrium equations, $\mathbf{K}_f \mathbf{u}_f = \mathbf{F}_f$, are solved for $\mathbf{u}_f$. First, the element deformations are obtained:

$$\mathbf{u}_b = \mathbf{T}_{bl} \mathbf{T}_{\mathrm{lg}} \mathbf{T}_{ga} \mathbf{T}_{af} \mathbf{u}_f \tag{27}$$

Note that matrix multiplication is straightforward but computationally costly; therefore, it is reiterated that more efficient algorithms are used in practice instead of $\mathbf{T}_{ga}$ and $\mathbf{T}_{af}$. Once $\mathbf{u}_b$ is determined, the element forces are:

$$\mathbf{F}_b = \mathbf{K}_b \mathbf{u}_b + \bar{\mathbf{F}}_b \tag{28}$$

where it is carefully noted that $\mathbf{F}_b$ at the element level has a different interpretation than $\mathbf{F}_f$ at the structural level. Specifically, $\mathbf{F}_b$ contains the forces along the degrees of freedom

of the element, caused by element deformations and element loads. That contrasts with $\mathbf{F}_f$ at the structural level, which contains all the applied loads on the structure. One way of thinking about this is is: First $\mathbf{K}_f \mathbf{u}_f = \mathbf{F}_f$ gives the deformations $\mathbf{u}_f$. Next $\mathbf{F}_b = \mathbf{K}_b \mathbf{u}_b$ gives the element forces, $\mathbf{F}_b$.