Computational Stiffness Method

Hand calculations are central in the classical stiffness method. In that approach, the stiffness matrix is established column-by-column by setting the degrees of freedom of the global structure equal to one, one at a time. In contrast, this document presents the version of the stiffness method that is intended for implementation on the computer. Here, the concept of transformation matrices between element configurations is central.

Configurations

The stiffness matrix and load vector for the global structure is assembled with contributions from the individual elements of the structure. To understand this process it is helpful to think of the elements as going through different "configurations" on the way to the final structure. Five such element configurations are identified in this document:

- **Basic:** In this configuration the element has the minimum number of DOFs to describe any element deformation but not rigid-body motion
- **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
- **All:** This is a structural configuration, in which absolutely all DOFs of the structure are included, even those associated with boundary conditions
- **Final:** In this structural configuration the boundary conditions, as well as other restraints and dependencies are introduced

Transformation Matrices

A transformation matrix defines the relationship between the DOFs in two configurations. Denoting by **u** the vector of DOFs the relationships are written

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

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

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

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

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 **b**asic and **l**ocal configurations. Notice that the "basic-most" configuration appears on the left-hand side in the displacement relationships in Eqs. (1) to (4). The principle of virtual work is 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 basic and local 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:

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

Since $\delta\mathbf{u}_g$ 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 basic configuration is known. This represents the material law at the basic level.

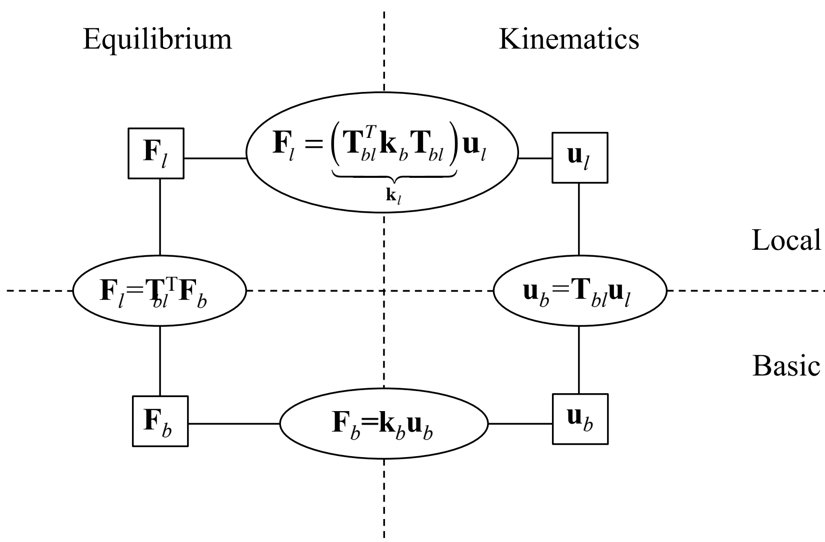

Figure 1: Equilibrium and kinematics between the Basic and Local configurations.

Combined with the known equilibrium and kinematics relationships in Figure 1 the stiffness matrix in the local configuration is:

$$\begin{aligned} \mathbf{F}_l &= \mathbf{T}_{bl}^T \mathbf{F}_b \\ &= \mathbf{T}_{bl}^T \mathbf{k}_b \mathbf{u}_b \\ &= \underbrace{\mathbf{T}_{bl}^T \mathbf{k}_b \mathbf{T}_{bl}}_{\mathbf{k}_l} \mathbf{u}_l \end{aligned} \tag{8}$$

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, schematically written:

$$\mathbf{K} = \mathbf{T}^T\mathbf{k}\mathbf{T} \tag{9}$$

This is sometimes referred to as a contragradient transformation. Notice also that the load vector also transforms according to the transformation matrix. With reference to the left-hand side of Figure 1, the load vector in an "above" configuration is obtained by pre-multiplying the load vector from the "below" configuration by the transpose of the transformation matrix between the two configurations.

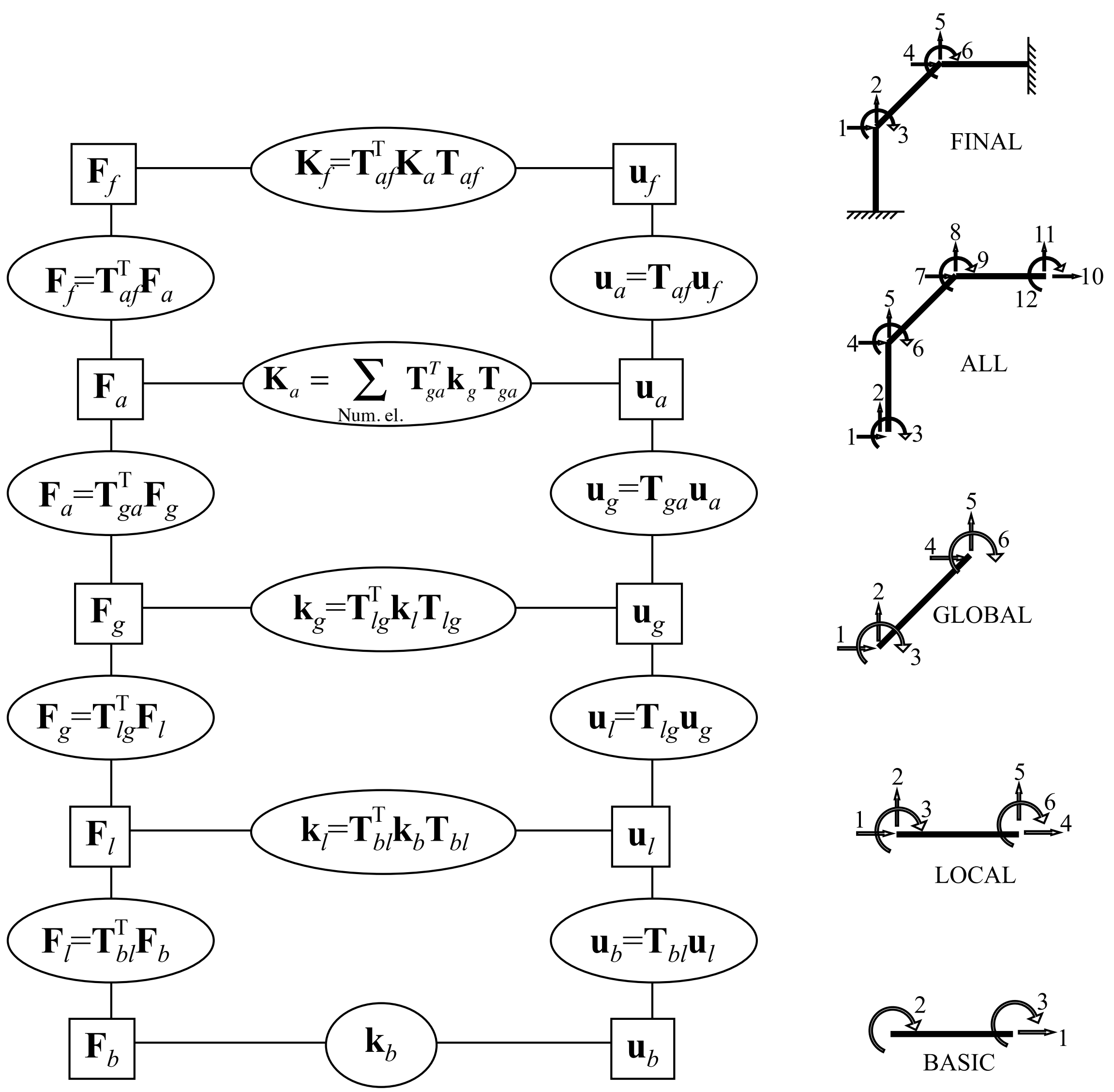

Figure 2: Overview of configurations and matrix relationships.

Figure 2 provides an overview of all the configurations and matrix relationships. In the following sections the transformation matrices are explicitly established, for 2D truss and frame elements. 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.

Analysis Procedure

The analysis procedure associated with Figure 2 is:

1. Establish the stiffness matrix in the basic configuration, $\mathbf{k}_b$
2. Assemble the final stiffness matrix for the final structure, $\mathbf{K}_f$, by repeatedly applying transformation matrices:

$$\mathbf{K}_f = \mathbf{T}_{af}^T \left(\sum_{i=1}^{numEl} \mathbf{T}_{ga,i}^T \mathbf{T}_{\lg,i}^T \mathbf{T}_{bl,i}^T \mathbf{k}_{b,i} \mathbf{T}_{bl,i} \mathbf{T}_{\lg,i} \mathbf{T}_{ga,i} \right) \mathbf{T}_{af} \tag{10}$$

3. Establish the final load vector, $\mathbf{F}_l$, either by inserting point loads at the correct DOF-positions, or by establishing fixed-end forces in the basic or local element configurations, followed by the application of transformation matrices:

$$\mathbf{F}_f = \mathbf{T}_{af}^T \left(\sum_{i=1}^{numEl} \mathbf{T}_{ga,i}^T \mathbf{T}_{\lg,i}^T \mathbf{T}_{bl,i}^T \mathbf{F}_{b,i} \right) \tag{11}$$

4. Solve the system of equilibrium equations:

$$\mathbf{u}_f = \mathbf{K}_f^{-1} \mathbf{F}_f \tag{12}$$

5. For each element, calculate the deformations in the basic configuration by applying transformation matrices:

$$\mathbf{u}_b = \mathbf{T}_{bl} \mathbf{T}_{\lg} \mathbf{T}_{ga} \mathbf{T}_{af} \mathbf{u}_f \tag{13}$$

6. For each element, calculate the forces in the basic element configuration by the basic stiffness relationship, which essentially yields the "homogeneous solution" for the section force diagrams:

$$\mathbf{F}_b = \mathbf{K}_b \mathbf{u}_b \tag{14}$$

7. Add the "particular solution" to the section force diagrams, i.e., the contribution from distributed element loads. Contrary to the moment distribution method and the slope-deflection method, the structure is still considered fully clamped after the stiffness method analysis is completed. Hence, the "particular solution" to be added is the section force diagrams for *fixed-fixed beams*.

Stiffness Matrix and Load Vector in the Basic Configuration

Before invoking the transformation matrices, the starting point of the analysis is to establish the fundamental stiffness matrix and load vector for the element. For the frame element in Figure 2 the basic stiffness matrix is

$$\mathbf{k}_b = \begin{bmatrix} \frac{EA}{L} & 0 & 0 \\ 0 & \frac{4EI}{L} & \frac{2EI}{L} \\ 0 & \frac{2EI}{L} & \frac{4EI}{L} \end{bmatrix} \tag{15}$$

The entries in this matrix are obtained by solving the differential equation for the element, or equivalent methods to determine element deformations. If there are distributed element loads then the load vector is established by entering the fixed-end forces into the basic or local force vector, whichever is more convenient. Thereafter, the sign of the vector is flipped and the final load vector is established by utilizing the transformation matrices that are described in the following.

Basic-Local Transformation

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 row of $\mathbf{T}_{bl}$. As an example, consider the 2D frame element in Figure 2. 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 row of $\mathbf{T}_{bl}$, which in its entirety reads

$$\mathbf{T}_{bl} = \left[\begin{array}{ccc|ccc} -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1/L & 1 & 0 & 1/L & 0 \\ 0 & -1/L & 0 & 0 & 1/L & 1 \end{array}\right] \tag{16}$$

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-Global Transformation

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 2 the result is

$$\mathbf{T}_{lg} = \left[\begin{array}{ccc|ccc} \cos(\theta) & \sin(\theta) & 0 & 0 & 0 & 0 \\ -\sin(\theta) & \cos(\theta) & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & \cos(\theta) & \sin(\theta) & 0 \\ 0 & 0 & 0 & -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \tag{17}$$

$\mathbf{T}_{lg}$ 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 defined as

$$c_x \equiv \frac{dx}{L}, \quad c_y \equiv \frac{dy}{L}, \quad c_z \equiv \frac{dz}{L} \tag{18}$$

Hence, $\mathbf{T}_{lg}$ can also be written

$$\mathbf{T}_{lg} = \left[\begin{array}{ccc|ccc} c_x & c_y & 0 & 0 & 0 & 0 \\ -c_y & c_x & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & c_x & c_y & 0 \\ 0 & 0 & 0 & -c_y & c_x & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \tag{19}$$

It is 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

$$\mathbf{T}_{lg} = \begin{bmatrix} \mathbf{R}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{R}^T \end{bmatrix} \tag{20}$$

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-All Transformation

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). Clearly, when the number of structural DOFs is large then the contragradient transformation in Eq. (9) to obtain the All stiffness matrix is computationally expensive. Therefore, in addition to establishing $\mathbf{T}_{ga}$, it is an objective in this section to look for algorithms that are more efficient for establishing the stiffness matrix and load vector in the All configuration.

$\mathbf{T}_{ga}$ is established by the same procedure as all other transformation matrices: one DOF at a time in the All configuration is set equal to unity. For the structure in Figure 2, 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:

$$\mathbf{T}_{ga} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{21}$$

The presence of only zeros and ones is a general characteristic of the $\mathbf{T}_{ga}$ matrix. It is relatively straightforward to implement an algorithm that places the ones in the correct position in the matrix. This is done in the transformation matrix assembler in ***St***. However, the computational cost increases rapidly with the number of structural DOFs. In the transformation matrix approach, one $\mathbf{T}_{ga}$ matrix is established for each element followed by the following matrix multiplication and summation:

$$\mathbf{K}_a = \left(\mathbf{T}_{ga}^T \mathbf{k}_g \mathbf{T}_{ga}\right)_{\text{Element 1}} + \left(\mathbf{T}_{ga}^T \mathbf{k}_g \mathbf{T}_{ga}\right)_{\text{Element 2}} + \left(\mathbf{T}_{ga}^T \mathbf{k}_g \mathbf{T}_{ga}\right)_{\text{Element 3}} + \cdots \tag{22}$$

Similarly, the load vector is

$$\mathbf{F}_a = \left(\mathbf{T}_{ga}^T \mathbf{F}_g\right)_{\text{Element 1}} + \left(\mathbf{T}_{ga}^T \mathbf{F}_g\right)_{\text{Element 2}} + \left(\mathbf{T}_{ga}^T \mathbf{F}_g\right)_{\text{Element 3}} + \cdots \tag{23}$$

Algorithms that are far more efficient, albeit perhaps less pedagogical, exist. After all, the objective is simply to enter contributions from the element stiffness matrix and load vector into their structural counterparts. This is achieved by entering sub-matrices from $\mathbf{k}_g$ into $\mathbf{K}_a$, and by entering sub-vectors from $\mathbf{F}_g$ into $\mathbf{F}_a$:

$$\mathbf{K}_a = \left[\begin{array}{c|c|c|c|c} & & & & \\ \hline & [\ ] & [\ ] & & \\ \hline & [\ ] & [\ ] & & \\ \hline & & & & \\ \hline & & & & \end{array}\right], \quad \mathbf{F}_a = \left\{\begin{array}{c} \\ \hline \{\ \} \\ \hline \{\ \} \\ \hline \\ \hline \\ \end{array}\right\} \tag{24}$$

All-Final Transformation

Fixed boundary conditions are introduced by means of the $\mathbf{T}_{af}$ transformation matrix. The dimensions of this matrix is given by the number of DOFs in the All configuration and 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 2, 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:

$$\mathbf{T}_{af} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{25}$$

The final stiffness matrix and load vector are

$$\mathbf{K}_f = \mathbf{T}_{af}^T \mathbf{K}_a \mathbf{T}_{af}, \qquad \mathbf{F}_f = \mathbf{T}_{af}^T \mathbf{F}_a \tag{26}$$

Depending on the number of DOFs in the structure, these matrix multiplication operations are unnecessarily computationally expensive, albeit pedagogically appealing. Therefore, instead of carrying out the operations in Eq. (26), consider the effect of these operations. For the stiffness matrix, the row and column that corresponds to the fixed DOF is removed. Similarly for the load vector, the row that corresponds to the fixed DOF is removed. This is the effect of Eq. (26), and this can be accomplished with efficient algorithms.

Shear Wall Transformation

One useful application of matrix structural analysis is "shear wall analysis." The objective of this type of analysis is to determine the forces on individual shear walls due to a global force, F, on a lateral force resisting system consisting of columns and shear walls. The analysis procedure is described earlier in this document and the stiffness matrix for a shear wall element is discussed in the document on frame elements. Figure 3 shows the plan view of a generic building and one of the supporting shear walls. The relationship between the DOFs of the floor, $\mathbf{u}_{floor}$, and the DOFs of the shear wall, $\mathbf{u}_{wall}$, is sought for the analysis. The DOFs of the floor are arbitrarily selected to originate in the lower left corner. Setting the components of $\mathbf{u}_{floor}$ equal to one, one at a time, establishes the columns of the transformation matrix:

$$\mathbf{T}_{wf} = \begin{bmatrix} 1 & 0 & b \\ 0 & 1 & -a \\ 0 & 0 & 1 \end{bmatrix} \tag{27}$$

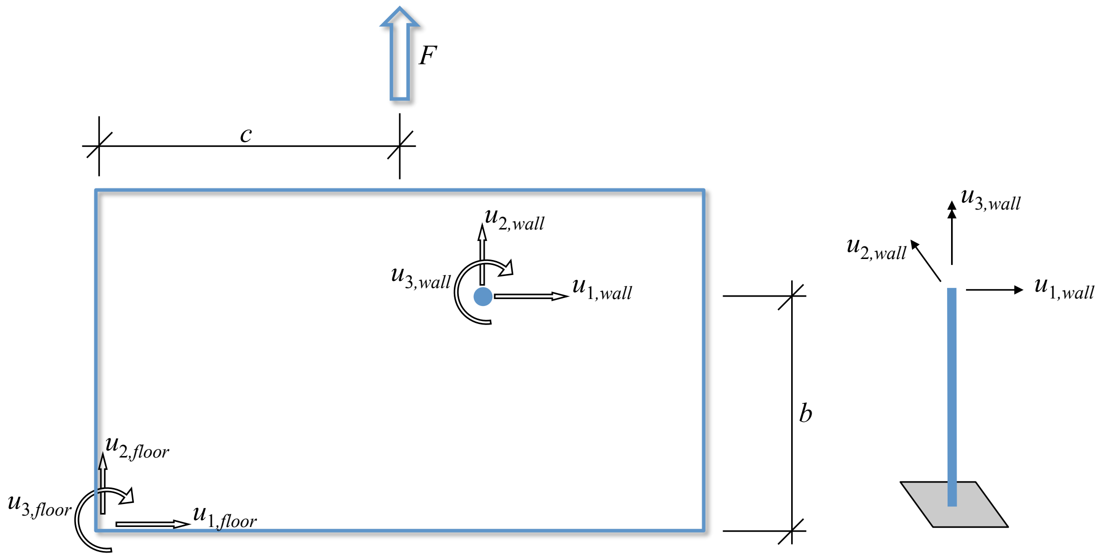

Figure 3: Rigid floor with shear wall.

The load vector for the shown floor is:

$$\mathbf{F} = \begin{Bmatrix} 0 \\ F \\ -c \cdot F \end{Bmatrix} \tag{28}$$

Special Topics

Static Condensation

Static condensation is a technique that is possible in linear structural analysis to remove DOFs from the system of equations without locking them. The DOFs that are removed remain free to translate or rotate, but the value of the translation or rotation remains unknown after the system of equations in solved. Static condensation is particularly useful in the development of "super-elements." A super-element is one that is initially developed with "many" DOFs, some of which are removed by static condensation. Usually, only the DOFs at the boundary of the element are retained. To understand static condensation, consider the following sorted system of equations for an element or a structure

$$\begin{bmatrix} \mathbf{K}_{ii} & \mathbf{K}_{io} \\ \mathbf{K}_{oi} & \mathbf{K}_{oo} \end{bmatrix} \begin{Bmatrix} \mathbf{u}_i \\ \mathbf{u}_o \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}_i \\ \mathbf{F}_o \end{Bmatrix} \tag{29}$$

where subscript i identifies the DOFs that are kept in, while subscript o identifies the DOFs that are to be tossed out. Next, write out the two sub-systems of equations

$$\begin{aligned} \mathbf{K}_{ii}\mathbf{u}_i + \mathbf{K}_{io}\mathbf{u}_o &= \mathbf{F}_i \\ \mathbf{K}_{oi}\mathbf{u}_i + \mathbf{K}_{oo}\mathbf{u}_o &= \mathbf{F}_o \end{aligned} \tag{30}$$

Solve for $\mathbf{u}_o$ in the second sub-system to obtain:

$$\mathbf{u}_o = \mathbf{K}_{oo}^{-1}\left(\mathbf{F}_o - \mathbf{K}_{oi}\mathbf{u}_i\right) \tag{31}$$

Substitution into the first sub-system yields:

$$\mathbf{K}_{ii}\mathbf{u}_i + \mathbf{K}_{io}\left(\mathbf{K}_{oo}^{-1}\left(\mathbf{F}_o - \mathbf{K}_{oi}\mathbf{u}_i\right)\right) = \mathbf{F}_i \tag{32}$$

Re-arrange to obtain the new system of equations in the unknowns $\mathbf{u}_i$:

$$\underbrace{\left(\mathbf{K}_{ii} - \mathbf{K}_{io}\mathbf{K}_{oo}^{-1}\mathbf{K}_{oi}\right)}_{\mathbf{K}}\ \mathbf{u}_i = \underbrace{\mathbf{F}_i - \mathbf{K}_{io}\mathbf{K}_{oo}^{-1}\mathbf{F}_o}_{\mathbf{F}} \tag{33}$$

where the new stiffness matrix, $\mathbf{K}$, and load vector, $\mathbf{F}$, for the super-element or super-structure without the $\mathbf{u}_o$ DOFs are identified.

Settlements and Imposed Deformations

DOFs that experience settlements and imposed displacements are not unknowns. Rather, it is the forces along those DOFs that are unknown. Consequently, the system of equations is reduced by the introduction of these effects. To understand how this is handled, consider the sorted system of equations for the structure

$$\begin{bmatrix} \mathbf{K}_{ii} & \mathbf{K}_{is} \\ \mathbf{K}_{si} & \mathbf{K}_{ss} \end{bmatrix} \begin{Bmatrix} \mathbf{u}_i \\ \mathbf{u}_s \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}_i \\ \mathbf{F}_s \end{Bmatrix} \tag{34}$$

where subscript i identifies the free DOFs, while subscript s identifies the DOFs that are subject to settlement or imposed displacement or rotation. Next, write out the two sub-systems of equations

$$\begin{aligned} \mathbf{K}_{ii}\mathbf{u}_i + \mathbf{K}_{is}\mathbf{u}_s &= \mathbf{F}_i \\ \mathbf{K}_{si}\mathbf{u}_i + \mathbf{K}_{ss}\mathbf{u}_s &= \mathbf{F}_s \end{aligned} \tag{35}$$

The first of these two sub-systems is readily solved by moving the second term in the left-hand side over to the right-hand side:

$$\mathbf{K}_{ii}\mathbf{u}_i = \mathbf{F}_i - \mathbf{K}_{is}\mathbf{u}_s \tag{36}$$

because $\mathbf{u}_s$ is a vector of known displacements. Upon solving for $\mathbf{u}_i$, the second sub-system immediately provides the value of the forces $\mathbf{F}_s$ along the DOFs with imposed deformations.

Springs

Springs are concentrated stiffness values along selected DOFs. They typically represent flexible foundations, i.e., they are meant to be connected to ground. There are two ways to address concentrated springs. One approach is to introduce extra axial bars, i.e., truss elements that are attached to the structure at one end and to a fixed node at the other end. This is a straightforward approach, in which the bar stiffness EA/L is tuned to the desired spring stiffness. This approach also has the advantage that the force in the spring is read from the axial force in the truss member. Another approach is to introduce a special option to allow scalar stiffness values to be inserted into the diagonal of the structural stiffness matrix. This is perhaps simpler for the analyst because it eliminates the need to create an auxiliary node and truss element.

DOF Dependencies

A DOF dependency means that one DOF is equal to—or linearly dependent on—another DOF. One example is when the axial deformation of a horizontal frame element is neglected; then both horizontal end displacements are the same, i.e., dependent. Another example is an inclined roller support, in which case the vertical displacement is linearly related to the horizontal displacement. It is dangerous to address this problem by setting a member stiffness, e.g., axial stiffness, to some very high number. This may lead to poor conditioning of the stiffness matrix and, as a result, inaccuracies in the solution of the system of equations. A better approach is to introduce DOF dependencies by a special-purpose transformation matrix. For this purpose, consider the following relationship between DOFs in the All configuration:

$$\underbrace{\begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \end{Bmatrix}}_{\mathbf{u}_a} = \underbrace{\begin{bmatrix} 0 & \alpha & 0 & \\ 0 & 1 & 0 & \cdots \\ 0 & 0 & 1 & \\ & \vdots & & \ddots \end{bmatrix}}_{\mathbf{T}_d} \underbrace{\begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \end{Bmatrix}}_{\mathbf{u}_a} \tag{37}$$

where the transformation matrix $\mathbf{T}_d$ (d stands for dependency) states that DOF number one is equal to α times DOF number to, i.e.,

$$u_1 = \alpha \cdot u_2 \tag{38}$$

In other words, u_2 is the independent DOF and u_1 is the dependent DOF. As usual, the new stiffness matrix and load vector are obtained by

$$\mathbf{K}_a = \mathbf{T}_d^T \mathbf{K}_a \mathbf{T}_d, \qquad \mathbf{F}_a = \mathbf{T}_d^T \mathbf{F}_a \tag{39}$$

where the modified stiffness matrix has retained its dimension but has zeros in the row and column that correspond to the dependent DOF. Similarly, the modified load vector has a zero entry in the component that corresponds to the dependent DOF. Thus, the dependent DOF must be removed from the system of equations by the transformation from the All configuration to the Final configuration, i.e., by the $\mathbf{T}_{af}$ transformation matrix.

Buckling Loads and Modes

When the stiffness matrix is amended with "geometric stiffness" terms it is possible to compute the buckling loads, and associated displaced shapes, i.e., "modes," of the structure. There will be as many buckling loads as there are DOFs, but only the smallest is relevant in practice because it is the governing buckling load. The geometric stiffness matrix for each element type is established in separate documents, while the procedure to compute the buckling loads and modes is presented here. Upon assembling the final structural stiffness matrix, including geometric stiffness contributions, the system of equilibrium equations is written

$$\left(\mathbf{K} - P \cdot \mathbf{K}_G\right)\mathbf{u} = \mathbf{F} \tag{40}$$

The factorization of the axial force level P as a multiplier of the geometric stiffness matrix of the structure is noted. This is possible when the considered axial forces are from one source, such as gravity. It is also necessary that the formulation be based on the second-order linearized theory. The use of exact "Livesley functions," i.e., the exact solution of the differential equation, prevents the form in Eq. (40). However, when the form in Eq. (40) is possible, the buckling loads, P_{cr}, of the system can be computed. First, remove the other loads that are represented by the external load vector, i.e., set $\mathbf{F}=\mathbf{0}$. Next, recognize that the remaining system of equations is an eigenvalue problem; it is homogeneous with non-trivial solutions only when the determinant of the coefficient matrix is zero. Hence, the equation

$$\det\left(\mathbf{K} - P \cdot \mathbf{K}_G\right) = 0 \tag{41}$$

is solved to obtain the critical values of the axial load level, which are the buckling loads, P_{cr}. There will be as many buckling loads as there are degrees of freedom, but usually only the lowest value is relevant in design. Each buckling load has a corresponding buckling mode. While the buckling loads are the eigenvalues, the buckling modes are the eigenvectors. The modes represent the displaced shape of the structure when it buckles at the corresponding buckling load. The amplitude of the deformation is not uniquely determined, but the shape is obtained by setting one component of **u** equal to unity and solving for the others. Software applications that solve eigenvalue problems do this automatically.