A Unified Finite Element Matrix Condensation Method for Arbitrary Nodal Releases Across All Element Types

Peter Schulze
Structural Engineer
peter@schulzey.com

Abstract

This paper introduces a novel, completely generic matrix condensation method to model arbitrary degree-of-freedom (DOF) releases directly within individual finite element stiffness matrices. While classical static condensation is a staple for handling end releases in one-dimensional beam and frame elements, extending this capability to higher-dimensional continuum elements such as 2D plates/shells and 3D solids has traditionally been obstructed by boundary continuity issues and rotational DOF limitations. Consequently, commercial solvers rely heavily on computationally expensive multipoint constraints (MPCs) or mesh-altering coincident nodes with discrete springs.

To resolve this limitation, we present a unified, element-independent mathematical operator that modifies the isolated element stiffness matrix prior to global assembly. The proposed method makes no underlying assumptions about the element formulation, spatial dimensions, or kinematic fields. It is capable of cleanly releasing any combination of translational or rotational degrees of freedom, including multiple simultaneous releases on a single element, while preserving the structural symmetry and rank stability of the system.

Keywords: Finite Element Analysis; Stiffness Matrix Modification; Static Condensation; Element End Releases; Semi-Rigid Joints; Continuum Elements; Computational Mechanics; Structural Solver Optimization

Introduction

Numerous publications describe the calculation of stiffness matrices for finite element and frame analysis, however none provide a generalised method for modifying a stiffness matrix to accommodate pinned, sliding or spring (semi-rigid) joints. While some publications cover common special cases, such as truss elements that have both ends hinged or members with an axial spring or sliding joint at one end, these approaches are very specific and are therefore of limited use in practical structural analysis.

The industry standard method for modelling releases in finite elements is “node-coupling”, whereby duplicate coincident nodes connected by zero length springs are added at each release location. Full releases can be achieved by using a spring stiffness of zero. While this method is flexible, it adds to the matrix size, requires alteration of the mesh geometry, and can introduce ill-conditioning or convergence problems.

Some common applications of joint releases in structural analysis include: (a) the edge of a slab that transfers vertical forces but not moments to a supporting wall or beam, (b) beams, columns or braces that are hinged at one or both ends and, (c) scaffolding where the connections are semi-rigid in terms of moment transfer.

This paper presents a simple method for adjusting any stiffness matrix to incorporate moment, shear, axial or torsional semi-rigid or full releases. No assumptions are made about the content or structure of the stiffness matrix, making it completely generic and applicable for symmetric or non-symmetric matrices. The method can be applied in less than twenty lines of computer code.

The procedure works with matrices that are in local, global or any other set of axes. Releases can be applied along or about any axis, and multiple releases can be accomodated by simply doing the adjustment multiple times. It is suitable for static, dynamic or buckling analyses, both linear and non-linear.

The method in this paper works by replacing an element's internal spring with an external spring and then eliminating the internal degree of freedom that the spring is attached to. This results in an adjusted stiffness matrix that incorporates the spring and has the same number of degrees of freedom as the original. Full moment, shear, axial or torsional releases can be applied by just setting the spring stiffness to zero.

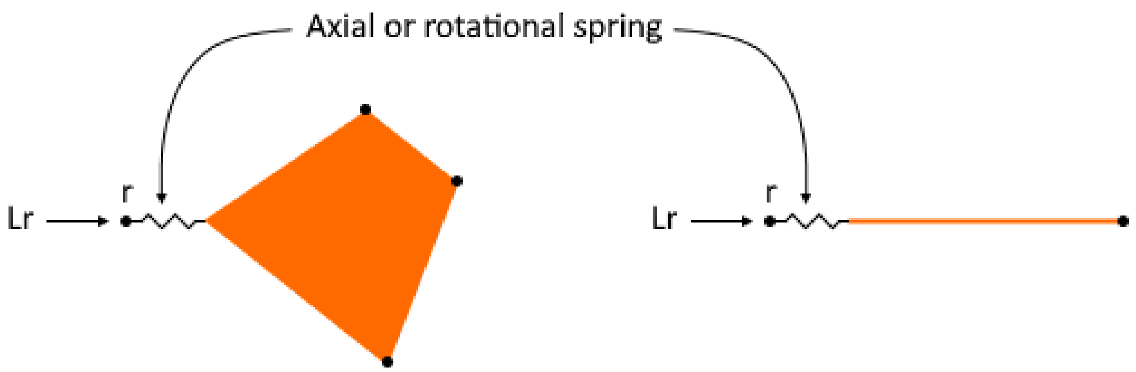

Figure 1 – Finite Element or Frame Member with Internal Spring Release

Figure 1 shows the finite element or frame member with an internal spring release applied to degree of freedom *r*. The stiffness of the element must be adjusted to incorporate the spring and that is the focus of this paper.

The stiffness matrix $[K]$ for a typical finite element or frame member relates the load vector $[L]$ to the displacement vector $[U]$ according to $[L] = [K][U]$ as follows:

$$\begin{bmatrix} L_1 \\ L_2 \\ L_3 \\ \vdots \\ L_n \end{bmatrix} = \begin{bmatrix} K_{11} & K_{12} & K_{13} & \cdots & K_{1n} \\ K_{21} & K_{22} & K_{23} & \cdots & K_{2n} \\ K_{31} & K_{32} & K_{33} & \cdots & K_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ K_{n1} & K_{n2} & K_{n3} & \cdots & K_{nn} \end{bmatrix} \cdot \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ \vdots \\ U_n \end{bmatrix} \tag{1}$$

Similarly, a single degree of freedom spring with stiffness S can be shown in matrix form as follows:

$$\begin{bmatrix} L_r \\ L_e \end{bmatrix} = \begin{bmatrix} +S & -S \\ -S & +S \end{bmatrix} \cdot \begin{bmatrix} U_r \\ U_e \end{bmatrix} \tag{2}$$

where *r* is the element's internal degree of freedom that the spring is attached to and *e* is the external degree of freedom at the other end of the spring.

They can be combined as shown in figure 2 below with the internal spring replaced by an equivalent external spring and new degree of freedom *e*.

Figure 2 – External Node Added and Internal Spring Changed to External

The stiffness matrix $[K']$ for this combined element is:

$$\begin{bmatrix} L_1 \\ L_2 \\ L_3 \\ \vdots \\ L_r \\ \vdots \\ L_n \\ L_e \end{bmatrix} = \begin{bmatrix} K_{11} & K_{12} & K_{13} & \cdots & K_{1r} & \cdots & K_{1n} & 0 \\ K_{21} & K_{22} & K_{23} & \cdots & K_{2r} & \cdots & K_{2n} & 0 \\ K_{31} & K_{32} & K_{33} & \cdots & K_{3r} & \cdots & K_{3n} & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & 0 \\ K_{r1} & K_{r2} & K_{r3} & \cdots & K_{rr}+S & \cdots & K_{rn} & -S \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & 0 \\ K_{n1} & K_{n2} & K_{n3} & \cdots & K_{nr} & \cdots & K_{nn} & 0 \\ 0 & 0 & 0 & 0 & -S & 0 & 0 & S \end{bmatrix} \cdot \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ \vdots \\ U_r \\ \vdots \\ U_n \\ U_e \end{bmatrix} \tag{3}$$

The Method

The focus of this paper is to find a simple method that eliminates degree of freedom *r* so that matrix $[K']$ incorporates the spring and is the same size as the original matrix $[K]$. This is illustrated in figure 3 below.

Figure 3 – Internal Degree of Freedom Eliminated

From matrix (2):

$$L_e = -SU_r + SU_e = S(U_e - U_r) \rightarrow U_r = U_e - \frac{L_e}{S} \qquad (4)$$

Any external load applied to *r* as L_r is now applied to *e* as L_e, and so $L_r = 0$ (5)

From matrix (3):

$L_1 = K_{11}U_1 + K_{12}U_2 + K_{13}U_3 + \cdots + K_{1r}U_r + \cdots + K_{1n}U_n$

$L_2 = K_{21}U_1 + K_{22}U_2 + K_{23}U_3 + \cdots + K_{2r}U_r + \cdots + K_{2n}U_n$

$L_3 = K_{31}U_1 + K_{32}U_2 + K_{33}U_3 + \cdots + K_{3r}U_r + \cdots + K_{3n}U_n$

$L_r = K_{r1}U_1 + K_{r2}U_2 + K_{r3}U_3 + \cdots + (K_{rr} + S)U_r + \cdots + K_{rn}U_n - SU_e = 0$

$L_n = K_{n1}U_1 + K_{n2}U_2 + K_{n3}U_3 + \cdots + K_{nr}U_r + \cdots + K_{nn}U_n$

Eliminating U_r by substituting $U_r = U_e - \frac{L_e}{S}$ from equation (4) and setting $L_r = 0$ as per equation (5):

$$L_1 = K_{11}U_1 + K_{12}U_2 + K_{13}U_3 + \cdots + K_{1r}\left(U_e - \frac{L_e}{S}\right) + \cdots + K_{1n}U_n \qquad (6)$$

$$L_2 = K_{21}U_1 + K_{22}U_2 + K_{23}U_3 + \cdots + K_{2r}\left(U_e - \frac{L_e}{S}\right) + \cdots + K_{2n}U_n \qquad (7)$$

$$L_3 = K_{31}U_1 + K_{32}U_2 + K_{33}U_3 + \cdots + K_{3r}\left(U_e - \frac{L_e}{S}\right) + \cdots + K_{3n}U_n \qquad (8)$$

$$L_r = K_{r1}U_1 + K_{r2}U_2 + K_{r3}U_3 + \cdots + (K_{rr} + S)\left(U_e - \frac{L_e}{S}\right) + \cdots + K_{rn}U_n - SU_e = 0 \qquad (9)$$

$$L_n = K_{n1}U_1 + K_{n2}U_2 + K_{n3}U_3 + \cdots + K_{nr}\left(U_e - \frac{L_e}{S}\right) + \cdots + K_{nn}U_n \qquad (10)$$

From equation (9):

$$K_{r1}U_1 + K_{r2}U_2 + K_{r3}U_3 + \cdots + (K_{rr} + S)\left(U_e - \frac{L_e}{S}\right) + \cdots + K_{rn}U_n - SU_e = 0$$

$$K_{r1}U_1 + K_{r2}U_2 + K_{r3}U_3 + \cdots + K_{rr}U_e + SU_e - \frac{L_e}{S}(K_{rr} + S) + \cdots + K_{rn}U_n - SU_e = 0$$

$$\therefore \frac{L_e}{S}(K_{rr} + S) = K_{r1}U_1 + K_{r2}U_2 + K_{r3}U_3 + \cdots + K_{rr}U_e + \cdots + K_{rn}U_n$$

$$\therefore \frac{L_e}{S} = \frac{1}{K_{rr}+S}(K_{r1}U_1 + K_{r2}U_2 + K_{r3}U_3 + \cdots + K_{rr}U_e + \cdots + K_{rn}U_n) \qquad (11)$$

Substituting $\frac{L_e}{S}$ from equation (11) into equation (6):

$$L_1 = K_{11}U_1 + K_{12}U_2 + K_{13}U_3 + \cdots + K_{1r}U_e + \cdots + K_{1n}U_n$$
$$- \frac{K_{1r}}{K_{rr}+S}(K_{r1}U_1 + K_{r2}U_2 + K_{r3}U_3 + \cdots + K_{rr}U_e + \cdots + K_{rn}U_n)$$

$$L_1 = U_1\left(K_{11} - \frac{K_{1r}K_{r1}}{K_{rr} + S}\right) + U_2\left(K_{12} - \frac{K_{1r}K_{r2}}{K_{rr} + S}\right) + U_3\left(K_{13} - \frac{K_{1r}K_{r3}}{K_{rr} + S}\right) + \cdots + U_e\left(K_{1r} - \frac{K_{1r}K_{rr}}{K_{rr} + S}\right) + \cdots + U_n\left(K_{1n} - \frac{K_{1r}K_{rn}}{K_{rr} + S}\right)$$

Similarly for equations (7), (8) and (10):

$$L_2 = U_1\left(K_{21} - \frac{K_{2r}K_{r1}}{K_{rr}+S}\right) + U_2\left(K_{22} - \frac{K_{2r}K_{r2}}{K_{rr}+S}\right) + U_3\left(K_{23} - \frac{K_{2r}K_{r3}}{K_{rr}+S}\right) + \cdots + U_e\left(K_{2r} - \frac{K_{2r}K_{rr}}{K_{rr}+S}\right) + \cdots + U_n\left(K_{2n} - \frac{K_{2r}K_{rn}}{K_{rr}+S}\right)$$

$$L_3 = U_1\left(K_{31} - \frac{K_{3r}K_{r1}}{K_{rr}+S}\right) + U_2\left(K_{32} - \frac{K_{3r}K_{r2}}{K_{rr}+S}\right) + U_3\left(K_{33} - \frac{K_{3r}K_{r3}}{K_{rr}+S}\right) + \cdots + U_e\left(K_{3r} - \frac{K_{3r}K_{rr}}{K_{rr}+S}\right) + \cdots + U_n\left(K_{3n} - \frac{K_{3r}K_{rn}}{K_{rr}+S}\right)$$

$$L_n = U_1\left(K_{n1} - \frac{K_{nr}K_{r1}}{K_{rr}+S}\right) + U_2\left(K_{n2} - \frac{K_{nr}K_{r2}}{K_{rr}+S}\right) + U_3\left(K_{n3} - \frac{K_{nr}K_{r3}}{K_{rr}+S}\right) + \cdots + U_e\left(K_{nr} - \frac{K_{nr}K_{rr}}{K_{rr}+S}\right) + \cdots + U_n\left(K_{nn} - \frac{K_{nr}K_{rn}}{K_{rr}+S}\right)$$

Rearranging equation (11) for L_e:

$$\frac{L_e}{S} = \frac{1}{K_{rr}+S}(K_{r1}U_1 + K_{r2}U_2 + K_{r3}U_3 + \cdots + K_{rr}U_e + \cdots + K_{rn}U_n)$$

$$L_e = U_1\frac{SK_{r1}}{K_{rr}+S} + U_2\frac{SK_{r2}}{K_{rr}+S} + U_3\frac{SK_{r3}}{K_{rr}+S} + \cdots + U_e\frac{SK_{rr}}{K_{rr}+S} + \cdots + U_n\frac{SK_{rn}}{K_{rr}+S}$$

$$L_e = U_1\left(K_{r1} - \frac{K_{rr}K_{r1}}{K_{rr}+S}\right) + U_2\left(K_{r2} - \frac{K_{rr}K_{r2}}{K_{rr}+S}\right) + U_3\left(K_{r3} - \frac{K_{rr}K_{r3}}{K_{rr}+S}\right) + \cdots + U_e\left(K_{rr} - \frac{K_{rr}K_{rr}}{K_{rr}+S}\right) + \cdots + U_n\left(K_{rn} - \frac{K_{rr}K_{rn}}{K_{rr}+S}\right)$$

Presenting the above equations in matrix form:

$$\begin{bmatrix} L_1 \\ L_2 \\ L_3 \\ \vdots \\ L_e \\ \vdots \\ L_n \end{bmatrix} = \begin{bmatrix} K_{11} - \frac{K_{1r}K_{r1}}{K_{rr}+S} & K_{12} - \frac{K_{1r}K_{r2}}{K_{rr}+S} & K_{13} - \frac{K_{1r}K_{r3}}{K_{rr}+S} & \cdots & K_{1r} - \frac{K_{1r}K_{rr}}{K_{rr}+S} & \cdots & K_{1n} - \frac{K_{1r}K_{rn}}{K_{rr}+S} \\ K_{21} - \frac{K_{2r}K_{r1}}{K_{rr}+S} & K_{22} - \frac{K_{2r}K_{r2}}{K_{rr}+S} & K_{23} - \frac{K_{2r}K_{r3}}{K_{rr}+S} & \cdots & K_{2r} - \frac{K_{2r}K_{rr}}{K_{rr}+S} & \cdots & K_{2n} - \frac{K_{2r}K_{rn}}{K_{rr}+S} \\ K_{31} - \frac{K_{3r}K_{r1}}{K_{rr}+S} & K_{32} - \frac{K_{3r}K_{r2}}{K_{rr}+S} & K_{33} - \frac{K_{3r}K_{r3}}{K_{rr}+S} & \cdots & K_{3r} - \frac{K_{3r}K_{rr}}{K_{rr}+S} & \cdots & K_{3n} - \frac{K_{3r}K_{rn}}{K_{rr}+S} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ K_{r1} - \frac{K_{rr}K_{r1}}{K_{rr}+S} & K_{r2} - \frac{K_{rr}K_{r2}}{K_{rr}+S} & K_{r3} - \frac{K_{rr}K_{r3}}{K_{rr}+S} & \cdots & K_{rr} - \frac{K_{rr}K_{rr}}{K_{rr}+S} & \cdots & K_{rn} - \frac{K_{rr}K_{rn}}{K_{rr}+S} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ K_{n1} - \frac{K_{nr}K_{r1}}{K_{rr}+S} & K_{n2} - \frac{K_{nr}K_{r2}}{K_{rr}+S} & K_{n3} - \frac{K_{nr}K_{r3}}{K_{rr}+S} & \cdots & K_{nr} - \frac{K_{nr}K_{rr}}{K_{rr}+S} & \cdots & K_{nn} - \frac{K_{nr}K_{rn}}{K_{rr}+S} \end{bmatrix} \cdot \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ \vdots \\ U_e \\ \vdots \\ U_n \end{bmatrix}$$

It is apparent from the above that the adjusted stiffness matrix $[K']$ is simply equal to the original matrix $[K]$ minus a release matrix $[R]$ as follows:

$$[L] = [K'][U] = ([K] - [R])[U]$$

Where the release matrix $[R]$ is:

$$[R] = \frac{1}{K_{rr}+S}\begin{bmatrix} K_{1r}K_{r1} & K_{1r}K_{r2} & K_{1r}K_{r3} & \cdots & K_{1r}K_{rr} & \cdots & K_{1r}K_{rn} \\ K_{2r}K_{r1} & K_{2r}K_{r2} & K_{2r}K_{r3} & \cdots & K_{2r}K_{rr} & \cdots & K_{2r}K_{rn} \\ K_{3r}K_{r1} & K_{3r}K_{r2} & K_{3r}K_{r3} & \cdots & K_{3r}K_{rr} & \cdots & K_{3r}K_{rn} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ K_{rr}K_{r1} & K_{rr}K_{r2} & K_{rr}K_{r3} & \cdots & K_{rr}K_{rr} & \cdots & K_{rr}K_{rn} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ K_{nr}K_{r1} & K_{nr}K_{r2} & K_{nr}K_{r3} & \cdots & K_{nr}K_{rr} & \cdots & K_{nr}K_{rn} \end{bmatrix}$$

It can be seen from matrix $[R]$ that all of its terms R_{ij} can be obtained from:

$$R_{ij} = \frac{K_{ir}K_{rj}}{K_{rr}+S} \quad (12)$$

And the stiffness matrix $[K]$ can therefore be adjusted as follows:

$$K'_{ij} = K_{ij} - \frac{K_{ir}K_{rj}}{K_{rr}+S} \quad (13)$$

Pinned or hinged releases can be applied by just setting the spring stiffness S to zero.

Multiple releases can be accomodated by perfoming the equation (13) adjustment multiple times, once for each release.

Conclusion

Equation (13) can be expressed in psuedo-code as follows, where *RelDF* is the degree of freedom that the spring release is applied to, *Stiff* is the spring stiffness and *n* is the size of the matrix. For hinge releases *Stiff* should be set to zero.

```
Fact = K(RelDF,RelDF) + Stiff
If (Fact <> 0) Then
        Assemble the release matrix [R]
        For Row = 1 to n
                For Col = 1 to n
                        R(Row,Col) = K(Row,RelDF) * K(RelDF,Col) / Fact
                Next
        Next

        Adjust the stiffness matrix [K] = [K] - [R]
        For Row = 1 to n
                For Col = 1 to n
                        K(Row,Col) = K(Row,Col) – R(Row,Col)
                Next
        Next
End If
```

It is clear from the above psuedo-code that the method is completely generalised. Furthermore, due to the simplicity of the solution, it can be incorporated into any FEA or frame analysis program with very little effort.