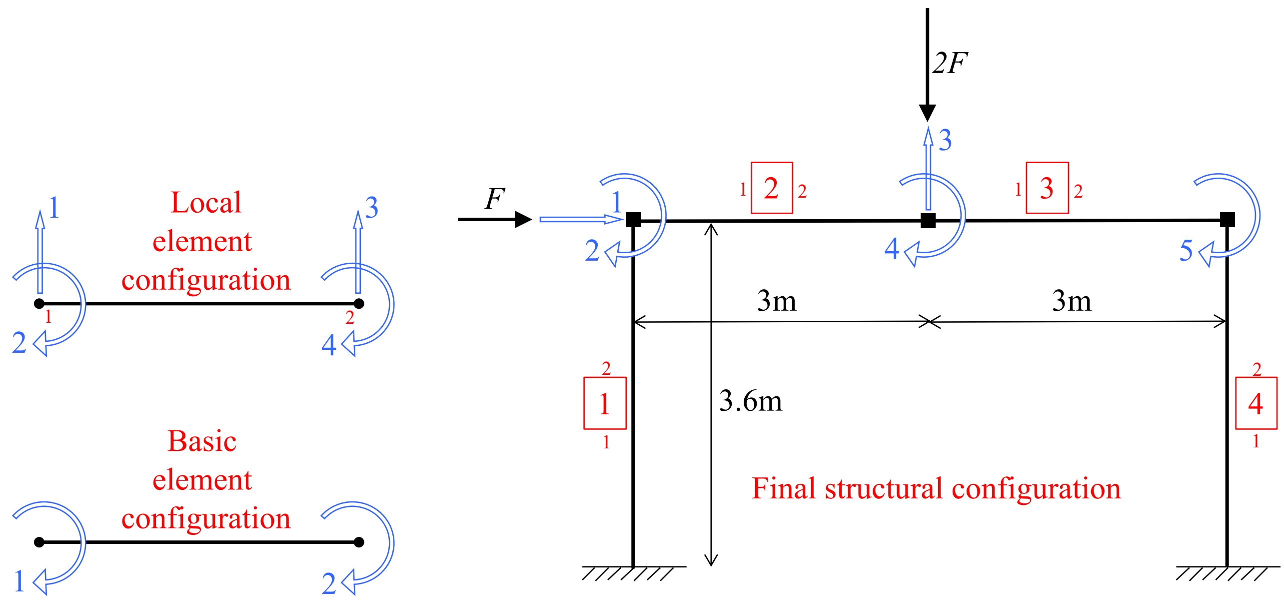
Computational Plastic Capacity Analysis

This example follows the “lower-bound equilibrium approach,” nicely described in Section 6.2.3 of the 2004 book-chapter by Filippou & Fenves entitled Methods of Analysis for Earthquake- Resistant Structures. Other documents on this website describes the lower-bound approach in elasto-plastic analysis for hand calculation of the plastic capacity of cross-sections. The objective in this document is to determine the plastic capacity of the frame shown below, using the matrix structural analysis approach described by Filippou & Fenves.



In this example, axial forces are neglected. As a result, the Basic element configuration has two degrees of freedom, i.e., the end rotations, as shown in the figure above. One the same note, the Local element configuration has four degrees of freedom, giving the following transformation matrix between the Basic and Location configurations:

$$\text{Tbl} = \left\{\left\{-\frac{1}{L}, 1, \frac{1}{L}, 0\right\}, \left\{-\frac{1}{L}, 0, \frac{1}{L}, 1\right\}\right\};$$

```
Tbl // MatrixForm
```

which yields: $\begin{pmatrix} -\frac{1}{L} & 1 & \frac{1}{L} & 0 \\ -\frac{1}{L} & 0 & \frac{1}{L} & 1 \end{pmatrix}$

A generic computer implement of this analysis approach would benefit from establishing the transformation matrix from the Local to the Global element configuration, followed by the transformation from the Global to the Final structural configuration shown in the figure above.

However, in this case-specific example, the transformation from the Local to the Final configuration, for each element, is established by visual inspection of the figure above:

```
Tlf1 = {{0, 0, 0, 0, 0}, {0, 0, 0, 0, 0}, {-1, 0, 0, 0, 0}, {0, 1, 0, 0, 0}};
Tlf1 // MatrixForm
```

which yields: $\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$

```
Tlf2 = {{0, 0, 0, 0, 0}, {0, 1, 0, 0, 0}, {0, 0, 1, 0, 0}, {0, 0, 0, 1, 0}};
Tlf2 // MatrixForm
```

which yields: $\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$

```
Tlf3 = {{0, 0, 1, 0, 0}, {0, 0, 0, 1, 0}, {0, 0, 0, 0, 0}, {0, 0, 0, 0, 1}};
Tlf3 // MatrixForm
```

which yields: $\begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$

```
Tlf4 = {{0, 0, 0, 0, 0}, {0, 0, 0, 0, 0}, {-1, 0, 0, 0, 0}, {0, 0, 0, 0, 1}};
Tlf4 // MatrixForm
```

which yields: $\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$

The compound matrices $\mathbf{T}_{bf,i} = \mathbf{T}_{bl}\,\mathbf{T}_{lg,i}$ for the individual elements are:

```
Tbf1 = (Tbl /. L -> 3.6).Tlf1;
Tbf2 = (Tbl /. L -> 3).Tlf2;
Tbf3 = (Tbl /. L -> 3).Tlf3;
Tbf4 = (Tbl /. L -> 3.6).Tlf4;
```

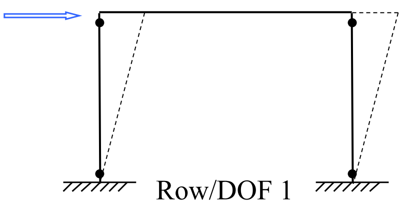
Following the approach outlined by Filippou & Fenves, and explained in the document entitled Computational Plastic Capacity Analysis, those matrices are now stacked in a unique way:

```
TbfStacked = Join[Join[Join[Tbf1, Tbf2, 1], Tbf3, 1], Tbf4, 1];
TbfStacked // MatrixForm
```

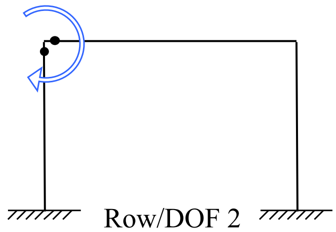
which yields:

$$\begin{pmatrix} -0.277778 & 0. & 0. & 0. & 0. \\ -0.277778 & 1. & 0. & 0. & 0. \\ 0 & 1 & \frac{1}{3} & 0 & 0 \\ 0 & 0 & \frac{1}{3} & 1 & 0 \\ 0 & 0 & -\frac{1}{3} & 1 & 0 \\ 0 & 0 & -\frac{1}{3} & 0 & 1 \\ -0.277778 & 0. & 0. & 0. & 0. \\ -0.277778 & 0. & 0. & 0. & 1. \end{pmatrix}$$

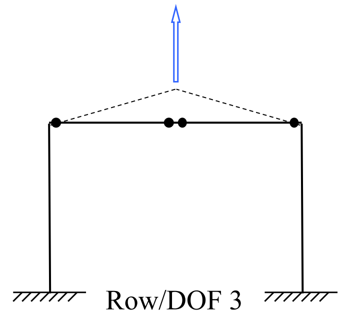
In order to consider equilibrium, it is actually the transpose of the **T**-matrices established above that are required. In fact, a document on the computational stiffness matrix, posted on this website, explains via virtual work why the "equilibrium matrix" is the transpose of the "compatibility matrix." For this reason, the transpose is used in the following. Interestingly, in that matrix, the five rows (a number that necessarily matches the number of Final degrees of freedom) reveals the five *independent* plastic "mechanisms." A non-zero number in a row indicates yielding at that element end. Examination of those numbers reveal the following mechanisms, some associated with element movement, some simply associated with joint (connection) movement. Notice how it is necessary to remember the location of each element end in the structure (shown in the first figure of this document) in order to identify the following plastic hinge locations from the five rows of the transposed of $\mathbf{T}_{\mathrm{bf,stacked}}$:



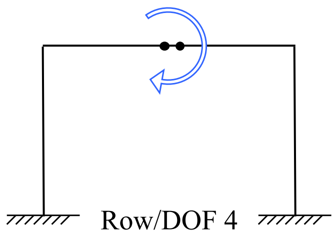
Row/DOF 1
(Element mechanism)



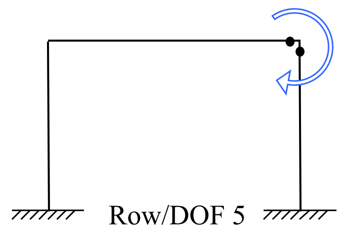
Row/DOF 2
(Joint mechanism)



Row/DOF 3
(Element mechanism)



Row/DOF 4
(Joint mechanism)



Row/DOF 5
(Joint mechanism)

Having addressed element end forces, the time has come to address the external forces. The following forces act along the five Final structural degrees of freedom, shown in the first figure in this document (the total load includes the load factor, i.e., $\lambda \boldsymbol{F}_{\mathrm{f}}$):

```
Ff = {1, 0, -2, 0, 0};
```

The formulation of the linear programming problem that the lower-bound theorem constitutes, benefits from adding the load factor at the beginning of the vector of unknowns, originally the vector of element end forces. That is done here by placing $\mathbf{F}_{\mathrm{f}}$ in front of the transposed of $\mathbf{T}_{\mathrm{bf,stacked}}$. Notice the minus sign, introduced because the equilibrium constraint, $\lambda \mathbf{F}_{\mathrm{f}} = \mathbf{T}_{\mathrm{bl,stacked}} \mathbf{F}_{b,\mathrm{stacked}}$, is written $\lambda \mathbf{F}_{\mathrm{f}} - \mathbf{T}_{\mathrm{bl,stacked}} \mathbf{F}_{b,\mathrm{stacked}} = \mathbf{0}$ as an equality constraint in upcoming linear programming problem.

```
TbfStackedAmended = Join[{Ff}ᵀ, -TbfStackedᵀ, 2];
TbfStackedAmended // MatrixForm
```

which yields:

$$\begin{pmatrix} 1 & 0.277778 & 0.277778 & 0 & 0 & 0 & 0 & 0.277778 & 0.277778 \\ 0 & 0. & -1. & -1 & 0 & 0 & 0 & 0. & 0. \\ -2 & 0. & 0. & -\frac{1}{3} & -\frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0. & 0. \\ 0 & 0. & 0. & 0 & -1 & -1 & 0 & 0. & 0. \\ 0 & 0. & 0. & 0 & 0 & 0 & -1 & 0. & -1. \end{pmatrix}$$

That formulation means that the vector of unknowns in the linear programming problem is:

$$\mathbf{x} = \{\lambda, M_{1,\mathrm{El.1}}, M_{2,\mathrm{El.1}}, M_{1,\mathrm{El.2}}, M_{2,\mathrm{El.2}}, M_{1,\mathrm{El.3}}, M_{2,\mathrm{El.3}}, M_{1,\mathrm{El.4}}, M_{2,\mathrm{El.4}}\}$$

On that note, for the formulation of the linear programming problem, we define the following vector that picks the load factor, λ, as the variable in **x** that we want to maximize:

```
pick = Join[{1}, Table[0, 8]]
```

which yields: {1, 0, 0, 0, 0, 0, 0, 0, 0}

The equality constraints state that there must be equilibrium, i.e., that $\lambda\mathbf{F}_\mathrm{f} - \mathbf{T}_{\mathrm{bl,stacked}}\,\mathbf{F}_{b,\mathrm{stacked}}$, must equal the zero-vector:

```
equalityConstraint = Table[0, {i, 5}, {j, 2}];
equalityConstraint // MatrixForm
```

which yields: $\begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$

The inequality constraints state that the absolute value of the end moments must be less than the yield moment. This is where differing yield moments in different members would be implemented:

```
Mu = 1;
inequalityConstraint =
  Join[{{-1 000 000, 1 000 000}}, Table[{-Mu, Mu}, {i, 8}]];
inequalityConstraint // MatrixForm
```

which yields: $\begin{pmatrix} -1\,000\,000 & 1\,000\,000 \\ -1 & 1 \\ -1 & 1 \\ -1 & 1 \\ -1 & 1 \\ -1 & 1 \\ -1 & 1 \\ -1 & 1 \\ -1 & 1 \end{pmatrix}$

The linear programming problem is here solved via a built-in Mathematic function:

```
LinearProgramming[-pick, TbfStackedAmended, equalityConstraint,
 inequalityConstraint]
```

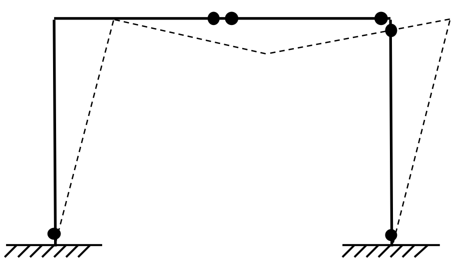
which yields: {0.625, -1., 0.75, -0.75, -1., 1., 1., -1., -1.}

Let's first verify that ultimate capacity of the frame by using the hand calculation approach with $dW_{int} = dW_{ext}$, which yields:

```
Solve[(θ + 2 θ + 2 θ + θ) My == F 3.6 θ + 2 F 3 θ, F]
```

which yields: {{F → 0.625 My}}

Next, let's examine the vector of end moments. The end moments equalling unity, or negative unity, are yielding, because the M_u was previously set equal to unity for all member ends. Translating the vector that came out of the linear programming solution gives the following figure:



Effectively, there are four hinges in that figure. Since the degree of static indeterminacy of the frame is three, that means the structure is unstable after yielding, indicating full collapse.