Multi Freedom Constraints

By
S. Ziaei Rad
Multifreedom Constraints

**Single freedom constraint examples**

\[ u_{x4} = 0 \quad \text{linear, homogeneous} \]
\[ u_{y9} = 0.6 \quad \text{linear, non-homogeneous} \]

**Multi freedom constraint examples**

\[ u_{x2} = \frac{1}{2} u_{y2} \quad \text{linear, homogeneous} \]
\[ u_{x2} - 2u_{x4} + u_{x6} = 0.25 \quad \text{linear, non-homogeneous} \]
\[ (x_5 + u_{x5} - x_3 - u_{x3})^2 + (y_5 + u_{y5} - y_3 - u_{y3})^2 = 0 \quad \text{nonlinear, homogeneous} \]
Sources of Multifreedom Constraints
1- Skew displacement BCs
2- Coupling nonmatched FEM meshes
3- Global-local and multiscale analysis
4- Incompressibility

MFC Application Methods
1- Master-Slave Elimination
2- Penalty Function Augmentation
3- Lagrange Multiplier Adjunction
Example 1D Structure to Illustrate MFCs

Multifreedom constraint:

\[ u_2 = u_6 \quad \text{or} \quad u_2 - u_6 = 0 \]

Linear homogeneous MFC
Example 1D Structure (Cont'd)

Unconstrained master stiffness equations

\[
\begin{bmatrix}
K_{11} & K_{12} & 0 & 0 & 0 & 0 & 0 \\
K_{12} & K_{22} & K_{23} & 0 & 0 & 0 & 0 \\
0 & K_{23} & K_{33} & K_{34} & 0 & 0 & 0 \\
0 & 0 & K_{34} & K_{44} & K_{45} & 0 & 0 \\
0 & 0 & 0 & K_{45} & K_{55} & K_{56} & 0 \\
0 & 0 & 0 & 0 & K_{56} & K_{66} & K_{67} \\
0 & 0 & 0 & 0 & 0 & K_{67} & K_{77}
\end{bmatrix}
\begin{bmatrix}
u_1 \\
u_2 \\
u_3 \\
u_4 \\
u_5 \\
u_6 \\
u_7
\end{bmatrix}
= 
\begin{bmatrix}
f_1 \\
f_2 \\
f_3 \\
f_4 \\
f_5 \\
f_6 \\
f_7
\end{bmatrix}
\]

\[ Ku = f \]
Master Slave Method for Example Structure

Recall: \( u_2 = u_6 \) \quad \text{or} \quad \underline{u_2} - u_6 = 0

Taking \( u_2 \) as master:

\[
\begin{bmatrix}
  u_1 \\
  u_2 \\
  u_3 \\
  u_4 \\
  u_5 \\
  u_6 \\
  u_7 \\
\end{bmatrix} =
\begin{bmatrix}
  1 & 0 & 0 & 0 & 0 & 0 \\
  0 & 1 & 0 & 0 & 0 & 0 \\
  0 & 0 & 1 & 0 & 0 & 0 \\
  0 & 0 & 0 & 1 & 0 & 0 \\
  0 & 0 & 0 & 0 & 1 & 0 \\
  0 & 1 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
  u_1 \\
  u_2 \\
  u_3 \\
  u_4 \\
  u_5 \\
  u_6 \\
  u_7 \\
\end{bmatrix}
\]

or

\[
\underline{u} = \hat{T}\underline{u}.
\]
Forming the Modified Stiffness Equations

Unconstrained master stiffness equation: \( Ku = f \)

Master-slave transformation: \( u = T \hat{u} \)
\[ \hat{K} = T^T K T \]
\[ \hat{f} = T^T f \]

Congruent transformation:

Modified stiffness equations: \( \hat{K} \hat{u} = \hat{f} \)
Modified Stiffness Equations for Example Structure

\( u_2 \) as master and \( u_6 \) as slave DOF.

\[
\begin{bmatrix}
K_{11} & K_{12} & 0 & 0 & 0 & 0 \\
K_{12} & K_{22} + K_{66} & K_{23} & 0 & K_{56} & K_{67} \\
0 & K_{23} & K_{33} & K_{34} & 0 & 0 \\
0 & 0 & K_{34} & K_{44} & K_{45} & 0 \\
0 & K_{56} & 0 & K_{45} & K_{55} & 0 \\
0 & K_{67} & 0 & 0 & 0 & K_{77}
\end{bmatrix}
\begin{bmatrix}
u_1 \\
u_2 \\
u_3 \\
u_4 \\
u_5 \\
u_7
\end{bmatrix}
= 
\begin{bmatrix}
f_1 \\
f_2 + f_6 \\
f_3 \\
f_4 \\
f_5 \\
f_7
\end{bmatrix}
\]

\( u_6 \) as master and \( u_2 \) as slave DOF.

\[
\begin{bmatrix}
K_{11} & 0 & 0 & 0 & K_{12} & 0 \\
0 & K_{33} & K_{34} & 0 & K_{23} & 0 \\
0 & K_{34} & K_{44} & K_{45} & 0 & 0 \\
0 & 0 & K_{45} & K_{55} & K_{56} & 0 \\
K_{12} & K_{23} & 0 & K_{56} & K_{22} + K_{66} & K_{67} \\
0 & 0 & 0 & 0 & K_{67} & K_{77}
\end{bmatrix}
\begin{bmatrix}
u_1 \\
u_3 \\
u_4 \\
u_5 \\
u_6 \\
u_7
\end{bmatrix}
= 
\begin{bmatrix}
f_1 \\
f_3 \\
f_4 \\
f_5 \\
f_2 + f_6 \\
f_7
\end{bmatrix}
\]

Although they are algebraically equivalent, the latter would be processed faster if a skyline solver is used for the modified equations.
Multiple MFCs

Suppose

\[ u_2 - u_6 = 0, \quad u_1 + 4u_4 = 0, \quad 2u_3 + u_4 + u_5 = 0 \]

take 3, 4 and 6 as slaves:

\[ u_6 = u_2, \quad u_4 = -\frac{1}{4}u_1, \quad u_3 = -\frac{1}{2}(u_4 + u_5) = \frac{1}{8}u_1 - \frac{1}{2}u_5 \]

and put in matrix form:

\[
\begin{bmatrix}
  u_1 \\
  u_2 \\
  u_3 \\
  u_4 \\
  u_5 \\
  u_6 \\
  u_7
\end{bmatrix}
= \begin{bmatrix}
  1 & 0 & 0 & 0 & 0 \\
  0 & 1 & 0 & 0 & 0 \\
  \frac{1}{8} & 0 & -\frac{1}{2} & 0 & 0 \\
  -\frac{1}{4} & 0 & 0 & 0 & 0 \\
  0 & 0 & 1 & 0 & 0 \\
  0 & 1 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  u_1 \\
  u_2 \\
  u_3 \\
  u_4 \\
  u_5 \\
  u_6 \\
  u_7
\end{bmatrix}
\]
Nonhomogeneous MFCs

\[ u_2 - u_6 = 0.2 \]

In matrix form

\[
\begin{bmatrix}
  u_1 \\
  u_2 \\
  u_3 \\
  u_4 \\
  u_5 \\
  u_6 \\
  u_7 \\
\end{bmatrix}
= 
\begin{bmatrix}
  1 & 0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 1 & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 1 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 1 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 1 & 0 & 0 \\
  0 & 1 & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\end{bmatrix}
- 
\begin{bmatrix}
  u_1 \\
  u_2 \\
  u_3 \\
  u_4 \\
  u_5 \\
  u_6 \\
  u_7 \\
\end{bmatrix}
\]
Nonhomogeneous MFCs (cont'd)

\[ u = T\hat{u} - g \]

\[ T^T K T\hat{u} = \hat{K}\hat{u} = \hat{\mathbf{f}} = T^T \mathbf{f} + T^T K \mathbf{g} \]

For the example structure

\[
\begin{bmatrix}
K_{11} & K_{12} & 0 & 0 & 0 & 0 & 0 \\
K_{12} & K_{22} + K_{66} & K_{23} & 0 & K_{56} & K_{67} & 0 \\
0 & K_{23} & K_{33} & K_{34} & 0 & 0 & 0 \\
0 & 0 & K_{34} & K_{44} & K_{45} & 0 & 0 \\
0 & K_{56} & 0 & K_{45} & K_{55} & 0 & 0 \\
0 & K_{67} & 0 & 0 & 0 & K_{77} & 0
\end{bmatrix}
\begin{bmatrix}
u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_7
\end{bmatrix}
= \begin{bmatrix}
f_1 \\ f_2 + f_6 - 0.2K_{66} \\ f_3 \\ f_4 \\ f_5 - 0.2K_{56} \\ f_7 - 0.2K_{67}
\end{bmatrix}
\]

*a modified force vector*
The General Case of MFCs

For implementation in general-purpose programs the master-slave method can be described as follows. The degrees of freedoms in \( \mathbf{u} \) are classified into three types: independent or uncommitted, masters and slaves.

\[
\begin{bmatrix}
K_{uu} & K_{um} & K_{us} \\
K_{mu} & K_{mm} & K_{ms} \\
K_{su} & K_{sm} & K_{ss}
\end{bmatrix}
\begin{bmatrix}
\mathbf{u}_u \\
\mathbf{u}_m \\
\mathbf{u}_s
\end{bmatrix} =
\begin{bmatrix}
\mathbf{f}_u \\
\mathbf{f}_m \\
\mathbf{f}_s
\end{bmatrix}
\]

The MFCs may be written in matrix form as

\[
A_m \mathbf{u}_m + A_s \mathbf{u}_s = \mathbf{g}, \quad \mathbf{u}_s = -A_s^{-1} A_m \mathbf{u}_m + A_s^{-1} \mathbf{g} = \mathbf{T} \mathbf{u}_m + \mathbf{g},
\]

Inserting into the partitioned stiffness matrix and symmetrizing

\[
\begin{bmatrix}
K_{uu} & K_{um}T \\
TTK_{um} & TTK_{mm}T
\end{bmatrix}
\begin{bmatrix}
\mathbf{u}_u \\
\mathbf{u}_m
\end{bmatrix} =
\begin{bmatrix}
\mathbf{f}_u - K_{us} \mathbf{g} \\
\mathbf{f}_m - K_{ms} \mathbf{g}
\end{bmatrix}
\]
Model Reduction

The congruential transformation equation (*) has additional applications beyond the master-slave method. An important one is *model reduction by kinematic constraints*. Through this procedure the number of DOF of a static or dynamic FEM model is reduced by a significant number, typically to 1% to 10% of the original number. This is done by taking a lot of slaves and a few masters. Only the masters are left after the transformation. Often the reduced model is used in subsequent calculations as component of a larger system, particularly during design or in parameter identification.
Model Reduction (cont'd)

Lots of slaves, few masters. Only masters are left. Example of previous slide:

Applying the congruential transformation we get the reduced stiffness equations

\[
\begin{bmatrix}
  u_1 \\
  u_2 \\
  u_3 \\
  u_4 \\
  u_5 \\
  u_6 \\
  u_7
\end{bmatrix}
= 
\begin{bmatrix}
  1 & 0 \\
  5/6 & 1/6 \\
  4/6 & 2/6 \\
  3/6 & 3/6 \\
  2/6 & 4/6 \\
  1/6 & 5/6 \\
  0 & 1
\end{bmatrix}
\begin{bmatrix}
  u_1 \\
  u_7
\end{bmatrix}
\]

5 slaves \[\rightarrow\] 2 masters

\[u = T \hat{u} \rightarrow \hat{K}\hat{u} = T^T K T \hat{u} = T^T f = \hat{f},\]

Applying the congruential transformation we get the reduced stiffness equations

\[
\begin{bmatrix}
  \hat{K}_{11} & \hat{K}_{17} \\
  \hat{K}_{17} & \hat{K}_{77}
\end{bmatrix}
\begin{bmatrix}
  u_1 \\
  u_7
\end{bmatrix}
= 
\begin{bmatrix}
  \hat{f}_1 \\
  \hat{f}_7
\end{bmatrix}
\]

where
Model Reduction (cont'd)

\[ \hat{K}_{11} = \frac{1}{36} (36K_{11} + 60K_{12} + 25K_{22} + 40K_{23} + 16K_{33} + 24K_{34} + 9K_{44} + 12K_{45} + 4K_{55} + 4K_{56} + K_{66}) \]
\[ \hat{K}_{17} = \frac{1}{36} (6K_{12} + 5K_{22} + 14K_{23} + 8K_{33} + 18K_{34} + 9K_{44} + 18K_{45} + 8K_{55} + 14K_{56} + 5K_{66} + 6K_{67}) \]
\[ \hat{K}_{77} = \frac{1}{36} (K_{22} + 4K_{23} + 4K_{33} + 12K_{34} + 9K_{44} + 24K_{45} + 16K_{55} + 40K_{56} + 25K_{66} + 60K_{67} + 36K_{77}) \]
\[ \hat{f}_1 = \frac{1}{6} (6f_1 + 5f_2 + 4f_3 + 3f_4 + 2f_5 + f_6), \quad \hat{f}_7 = \frac{1}{6} (f_2 + 2f_3 + 3f_4 + 4f_5 + 5f_6 + 6f_7). \]

Assessment of Master-Slave Method

**ADVANTAGES**

1- exact if precautions taken

2- easy to understand

3- retains positive definiteness

4- important applications to model reduction

**DISADVANTAGES**

1- requires user decisions

2- messy implementation for general MFCs

3- sensitive to constraint dependence

4- restricted to linear constraints
Condensation Techniques

- Guyan Reduction
- Improved Reduction System (IRS)
- Dynamic Reduction
- System Equivalent Reduction Expansion Process (SEREP).
Reduction Techniques

\[ \{x\}_{N \times l} = \begin{bmatrix} x_n \\ x_s \end{bmatrix}_{N \times l} = [T]_{N \times n} \{x\}_{n \times l} \]

Master Dofs

Slave Dofs

Transformation Matrix

\[ [M]_{n \times n} = [T]^T [M]_{N \times N} [T] \]
\[ [K]_{n \times n} = [T]^T [K]_{N \times N} [T] \]
\[ \{ \ddot{x}_s \} = -[K]^{-1}[K]\{x_n\} + [K]\{f_s\} \]

Consider the lower row of the above equation

Ignore the inertia term:

\[ [M]\{\ddot{x}_s\} + [K]\{x_s\} = \{f_s\} \]
Guyan Reduction

Assuming that there are no external forces at the slave DOFs

\[ \{x_s\} = -[K_{ss}]^{-1}[K_{sn}]{x_n} \]

Then

\[ [T] = \begin{bmatrix} [I] \\ -[K_{ss}]^{-1}[K_{sn}] \end{bmatrix} \]

Guyan Transformation matrix
Guyan Reduction

- Since the inertia terms are neglected, this technique is also called static reduction.
- Guyan reduction depends heavily on the selection of the master degrees of freedom.
- A poor selection yielding inaccurate models.
Penalty Function Method
(Physical Interpretation)

Recall the example structure under the homogeneous MF $u_2 = u_6$

"penalty element" of axial rigidity $w$

$$w \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} u_2 \\ u_6 \end{bmatrix} = \begin{bmatrix} f_2^{(7)} \\ f_6^{(7)} \end{bmatrix}$$
Penalty Function Method (cont'd)

Upon merging the penalty element the modified stiffness equations are

\[
\begin{bmatrix}
K_{11} & K_{12} & 0 & 0 & 0 & 0 & 0 \\
K_{12} & K_{22} + w & K_{23} & 0 & 0 & -w & 0 \\
0 & K_{23} & K_{33} & K_{34} & 0 & 0 & 0 \\
0 & 0 & K_{34} & K_{44} & K_{45} & 0 & 0 \\
0 & 0 & 0 & K_{45} & K_{55} & K_{56} & 0 \\
0 & -w & 0 & 0 & K_{56} & K_{66} + w & K_{67} \\
0 & 0 & 0 & 0 & 0 & K_{67} & K_{77}
\end{bmatrix}
\begin{bmatrix}
u_1 \\
u_2 \\
u_3 \\
u_4 \\
u_5 \\
u_6 \\
u_7
\end{bmatrix}
= 
\begin{bmatrix}
f_1 \\
f_2 \\
f_3 \\
f_4 \\
f_5 \\
f_6 \\
f_7
\end{bmatrix}
\]

This modified system is submitted to the equation solver. Note that \( \mathbf{u} \) retains the same arrangement of DOFs.
But which penalty weight to use?

If a finite weight \( w \) is chosen the constraint \( u_2 = u_6 \) is approximately satisfied in the sense that one gets \( u_2 - u_6 = eg \), where \( eg = 0 \). The “gap error” \( eg \) is called the \textit{constraint violation}. The magnitude \( |eg| \) of this violation depends on \( w \): the larger \( w \), the smaller the violation.

More precisely, it can be shown that \( |eg| \) becomes proportional to \( 1/w \) as \( w \) gets to be sufficiently large. However, this is misleading. As the penalty weight \( w \) tends to infinity the modified stiffness matrix becomes more and more \textit{ill-conditioned with respect to inversion}.

Obviously we have two effects at odds with each other. Making \( w \) larger reduces the constraint violation error but increases the solution error. The best \( W \) is that which makes both errors roughly equal in absolute value. This tradeoff value is difficult to find aside of systematically running numerical experiments. In practice the heuristic \textit{square root rule} is often followed.

This rule can be stated as follows. Suppose that the largest stiffness coefficient, before adding penalty elements, is of the order of \( 10^k \) and that the working machine precision is \( p \) digits. Then choose penalty weights to be of order \( 10^{k+p/2} \) with the proviso that such a choice would not cause arithmetic overflow.
But which penalty weight to use?

The name “square root” arises because the recommended $w$ is in fact $10^k \sqrt{10^p}$. Thus it is seen that the choice of penalty weight by this rule involves knowledge of both stiffness magnitudes and floating-point hardware properties of the computer used.

Rough guideline: "square root rule";
Penalty Function Method - General MFCs

\[ 3u_3 + u_5 - 4u_6 = 1 \]

Premultiply both sides by \( b' \):

\[ \begin{bmatrix} 3 & 1 & -4 \end{bmatrix} \begin{bmatrix} u_3 \\ u_5 \\ u_6 \end{bmatrix} = 1 \]

Scale by \( w \) and merge:

\[
\begin{bmatrix}
K_{11} & K_{12} & 0 & 0 & 0 & 0 & 0 \\
K_{12} & K_{22} & 0 & 0 & 0 & 0 & 0 \\
0 & K_{23} & K_{33} + 9w & K_{34} & 3w & -12w & 0 \\
0 & 0 & K_{34} & K_{44} & K_{45} & 0 & 0 \\
0 & 0 & 3w & K_{45} & K_{55} + w & K_{56} - 4w & 0 \\
0 & 0 & -12w & K_{56} - 4w & K_{66} + 16w & K_{67} & K_{77} \\
0 & 0 & 0 & 0 & 0 & K_{67} & K_{77}
\end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 + 3w \\ f_4 \\ f_5 + w \\ f_6 - 4w \\ f_7 \end{bmatrix}
\]
The rule comes from the following mathematical theory. Suppose we have a set of \( m \) linear MFCs.

\[
\mathbf{a}_p \mathbf{u} = b_p, \quad p = 1, \ldots, m
\]

where \( \mathbf{u} \) contains all degrees of freedom and each \( \mathbf{a}_p \) is a row vector with same length as \( \mathbf{u} \). To incorporate the MFCs into the FEM model one selects a weight \( w_p > 0 \) for each constraints and constructs the so-called Courant quadratic penalty function or “penalty energy”

\[
P = \sum_{p=1}^{m} P_p, \quad \text{with} \quad P_p = \mathbf{u}^T \left( \frac{1}{2} \mathbf{a}_p^T \mathbf{a}_p \mathbf{u} - w_p \mathbf{a}_p^T b_p \right) = \frac{1}{2} \mathbf{u}^T \mathbf{K}^{(p)} \mathbf{u} - \mathbf{u}^T \mathbf{f}^{(p)}
\]

where \( \mathbf{K}^{(p)} = w_p \mathbf{a}_p^T \mathbf{a}_p \) and \( \mathbf{f}^{(p)} = w_p \mathbf{a}_p^T b_i \).
Theory of Penalty Function Method - General MFCs

Next, $P$ is added to the potential energy function
$$\Pi = \frac{1}{2} u^T Ku - u^T f$$
to form the augmented potential energy
$$\Pi_a = \Pi + P$$
Minimization of $\Pi_a$ with respect with $u$ yields
$$\left( Ku + \sum_{p=1}^{m} K^{(p)} \right) u = f + \sum_{p=1}^{m} f^{(p)}$$

To use a even more compact form we may write the set of multifreedom constraints as $Au = b$. Then the penalty augmented system can be written compactly as
$$\left( K + A^T W A \right) u = f + W A^T b,$$
where $W$ is a diagonal matrix of penalty weights. This compact form, however, conceals the structure of the penalty elements.
Assessment of Penalty Function Method

**ADVANTAGES**

1- general application (inc' nonlinear MFCs)
2-easy to implement using FE library and standard assembler
3-no change in vector of unknowns
4-retains positive definiteness
5-insensitive to constraint dependence

**DISADVANTAGES**

1- selection of weight left to user
2-accuracy limited by ill-conditioning
Lagrange Multiplier Method

Physical Interpretation

\[
\begin{bmatrix}
K_{11} & K_{12} & 0 & 0 & 0 & 0 & 0 \\
K_{12} & K_{22} & K_{23} & 0 & 0 & 0 & 0 \\
0 & K_{23} & K_{33} & K_{34} & 0 & 0 & 0 \\
0 & 0 & K_{34} & K_{44} & K_{45} & 0 & 0 \\
0 & 0 & 0 & K_{45} & K_{55} & K_{56} & 0 \\
0 & 0 & 0 & 0 & K_{56} & K_{66} & K_{67} \\
0 & 0 & 0 & 0 & 0 & K_{67} & K_{77}
\end{bmatrix}
\begin{bmatrix}
u_1 \\
u_2 \\
u_3 \\
u_4 \\
u_5 \\
u_6 \\
u_7
\end{bmatrix}
= 
\begin{bmatrix}
f_1 \\
f_2 - \lambda \\
f_3 \\
f_4 \\
f_5 \\
f_6 + \lambda \\
f_7
\end{bmatrix}
\]
Lagrange Multiplier Method (cont'd)

Because $\lambda$ is unknown, it is passed to the LHS and appended to the node-displacement vector:

\[
\begin{bmatrix}
K_{11} & K_{12} & 0 & 0 & 0 & 0 & 0 & 0 \\
K_{12} & K_{22} & K_{23} & 0 & 0 & 0 & 0 & 1 \\
0 & K_{23} & K_{33} & K_{34} & 0 & 0 & 0 & 0 \\
0 & 0 & K_{34} & K_{44} & K_{45} & 0 & 0 & 0 \\
0 & 0 & 0 & K_{45} & K_{55} & K_{56} & 0 & 0 \\
0 & 0 & 0 & 0 & K_{56} & K_{66} & K_{67} & -1 \\
0 & 0 & 0 & 0 & 0 & K_{67} & K_{77} & 0 \\
\end{bmatrix}
\begin{bmatrix}
u_1 \\
u_2 \\
u_3 \\
u_4 \\
u_5 \\
u_6 \\
u_7 \\
\lambda \\
\end{bmatrix}
= 
\begin{bmatrix}
f_1 \\
f_2 \\
f_3 \\
f_4 \\
f_5 \\
f_6 \\
f_7 \\
\end{bmatrix}
\]

This is now a system of 7 equations and 8 unknowns. Needs an extra equation: the MFC.
Lagrange Multiplier Method (cont'd)

Append MFC as additional equation:

\[
\begin{bmatrix}
K_{11} & K_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
K_{12} & K_{22} & K_{23} & 0 & 0 & 0 & 0 & 1 \\
0 & K_{23} & K_{33} & K_{34} & 0 & 0 & 0 & 0 \\
0 & 0 & K_{34} & K_{44} & K_{45} & 0 & 0 & 0 \\
0 & 0 & 0 & K_{45} & K_{55} & K_{56} & 0 & 0 \\
0 & 0 & 0 & 0 & K_{56} & K_{66} & K_{67} & -1 \\
0 & 0 & 0 & 0 & 0 & K_{67} & K_{77} & 0 \\
0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\
\end{bmatrix}
\begin{bmatrix}
u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ \lambda \end{bmatrix}
= \begin{bmatrix}
f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ 0 \end{bmatrix}
\]

This is the *multiplier-augmented system*. The new coefficient matrix is called the *bordered stiffness*. 
Lagrange Multiplier Method - Multiple MFCs

Three MFCs: \( u_2 - u_6 = 0, \quad 5u_2 - 8u_7 = 3, \quad 3u_3 + u_5 - 4u_6 = 1 \)

Recipe step #1: append the 3 constraints

\[
\begin{bmatrix}
K_{11} & K_{12} & 0 & 0 & 0 & 0 & 0 \\
K_{12} & K_{22} & K_{23} & 0 & 0 & 0 & 0 \\
0 & K_{23} & K_{33} & K_{34} & 0 & 0 & 0 \\
0 & 0 & K_{34} & K_{44} & K_{45} & 0 & 0 \\
0 & 0 & 0 & K_{45} & K_{55} & K_{56} & 0 \\
0 & 0 & 0 & 0 & K_{56} & K_{66} & K_{67} \\
0 & 0 & 0 & 0 & 0 & K_{67} & K_{77} \\
0 & 1 & 0 & 0 & 0 & -1 & 0 \\
0 & 5 & 0 & 0 & 0 & 0 & -8 \\
0 & 0 & 3 & 0 & 1 & -4 & 0 \\
\end{bmatrix}
\begin{bmatrix}
u_1 \\
u_2 \\
u_3 \\
u_4 \\
u_5 \\
u_6 \\
u_7
\end{bmatrix}
= 
\begin{bmatrix}
f_1 \\
f_2 \\
f_3 \\
f_4 \\
f_5 \\
f_6 \\
f_7
\end{bmatrix}
\]
Lagrange Multiplier Method - Multiple MFCs

*Recipe step #2:*
append multipliers, symmetrize and fill

\[
\begin{pmatrix}
K_{11} & K_{12} & 0 & 0 & 0 & 0 & 0 & 0 \\
K_{12} & K_{22} & K_{23} & 0 & 0 & 0 & 0 & 0 \\
0 & K_{23} & K_{33} & K_{34} & 0 & 0 & 0 & 0 \\
0 & 0 & K_{34} & K_{44} & K_{45} & 0 & 0 & 0 \\
0 & 0 & 0 & K_{45} & K_{55} & K_{56} & 0 & 0 \\
0 & 0 & 0 & 0 & K_{56} & K_{66} & K_{67} & 0 \\
0 & 0 & 0 & 0 & 0 & K_{67} & K_{77} & 0 \\
0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 5 & 0 & 0 & 0 & 0 & -8 & 0 \\
0 & 0 & 3 & 0 & 1 & -4 & 0 & 0 \\
\end{pmatrix}
\begin{pmatrix}
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
\end{pmatrix}
= 
\begin{pmatrix}
1 & 5 & 0 \\
0 & 0 & 3 \\
0 & 0 & 0 \\
0 & 0 & 1 \\
-1 & 0 & -4 \\
0 & -8 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
\end{pmatrix}
\begin{pmatrix}
u_1 \\
u_2 \\
u_3 \\
u_4 \\
u_5 \\
u_6 \\
u_7 \\
\lambda_1 \\
\lambda_2 \\
\lambda_3 \\
\end{pmatrix}
= 
\begin{pmatrix}
f_1 \\
f_2 \\
f_3 \\
f_4 \\
f_5 \\
f_6 \\
f_7 \\
0 \\
0 \\
1 \\
\end{pmatrix}
The recipe illustrated in previous slides comes from a well known technique of variational calculus. Using the matrix notation we write, the set of $m$ MFCs by $Au=b$, where $A$ is $m \times n$. The potential energy of the unconstrained finite element model is

$$\Pi = \frac{1}{2} u^T Ku - u^T f.$$  

To impose the constraint, adjoin $m$ Lagrange multipliers collected in vector $\lambda$ and form the Lagrangian

$$L(u, \lambda) = \Pi + \lambda^T (Au - b) = \frac{1}{2} u^T Ku - u^T f + \lambda^T (Au - b).$$

Extremization of $\Pi$ with respect to $u$ and yields the multiplier-augmented form

$$\begin{bmatrix}
K & A^T \\
A & 0
\end{bmatrix}
\begin{bmatrix}
u \\
\lambda
\end{bmatrix}
= 
\begin{bmatrix}
f \\
b
\end{bmatrix}$$
Assessment of Lagrange Multiplier Method

ADVANTAGES
1- general application
2- exact
3- no user decisions ("black box")

DISADVANTAGES
1- difficult implementation
2- additional unknowns
3- loses positive definiteness
4- sensitive to constraint dependence
## MFC Application Methods: Assessment Summary

<table>
<thead>
<tr>
<th>Feature</th>
<th>Master-Slave Elimination</th>
<th>Penalty Function</th>
<th>Lagrange Multiplier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generality</td>
<td>fair</td>
<td>excellent</td>
<td>excellent</td>
</tr>
<tr>
<td>Ease of implementation</td>
<td>poor to fair</td>
<td>good</td>
<td>fair</td>
</tr>
<tr>
<td>Sensitivity to user decisions</td>
<td>high</td>
<td>high</td>
<td>small to none</td>
</tr>
<tr>
<td>Accuracy</td>
<td>variable</td>
<td>mediocre</td>
<td>excellent</td>
</tr>
<tr>
<td>Sensitivity as regards constraint dependence</td>
<td>high</td>
<td>none</td>
<td>high</td>
</tr>
<tr>
<td>Retains positive definiteness</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
</tbody>
</table>
Multi Freedom Constraints

The End

Which methods to use?