An Abaqus-Matlab Tutorial for Jointed Systems

International Committee on Jointed Structures Seminar Series

Nidish Narayanaa Balaji

Department of Aerospace Engineering, IIT Madras

March 26, 2025

Table of Contents

Detailed Instructions are hosted through Github in a repository named Nidish96/Abaqus4Joints

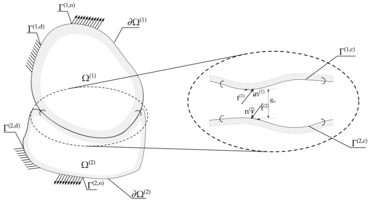
Acknowledgements

- Prof. Matthew Brake
- Dr. Justin Porter, Maeve Karpov
- Prof. Matt Allen

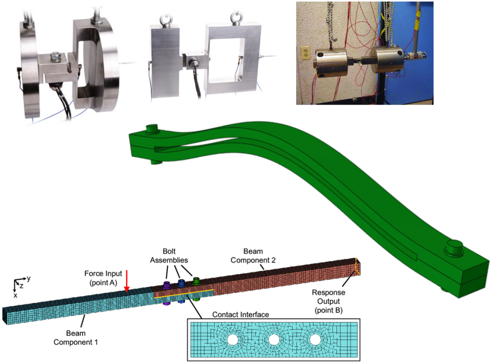
The research needs of the joints community are quite unique and specific

1. Introduction

The Generic Contact Problem



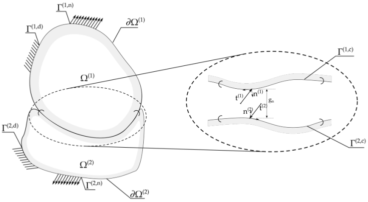
Joints Benchmarks (see jointmechanics.org)



The research needs of the joints community are quite unique and specific

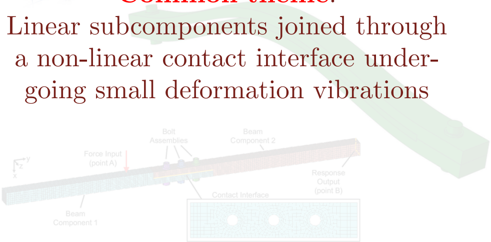
1. Introduction

The Generic Contact Problem



Joints Benchmarks (see jointmechanics.org)

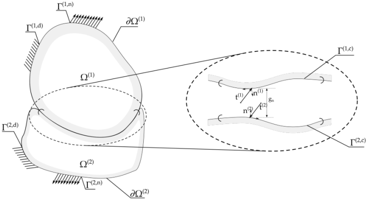
Common theme:
Linear subcomponents joined through a non-linear contact interface undergoing small deformation vibrations



The research needs of the joints community are quite unique and specific

1. Introduction

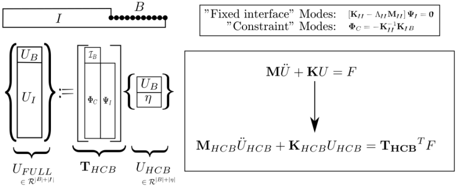
The Generic Contact Problem



Joints Benchmarks (see jointmechanics.org)

Common theme:
Linear subcomponents joined through a non-linear contact interface undergoing small deformation vibrations

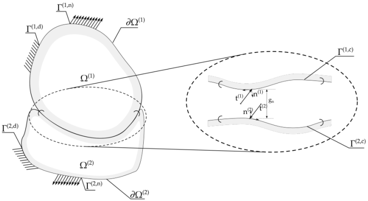
The HCB/CMS Methodology



The research needs of the joints community are quite unique and specific

1. Introduction

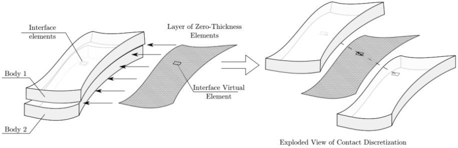
The Generic Contact Problem



Joints Benchmarks (see jointmechanics.org)

Common theme:
Linear subcomponents joined through a non-linear contact interface undergoing small deformation vibrations

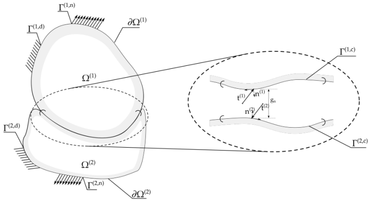
Interface Virtual Element Representations

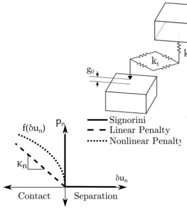


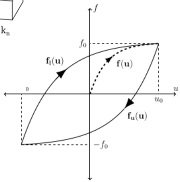
The research needs of the joints community are quite unique and specific

1. Introduction

The Generic Contact Problem



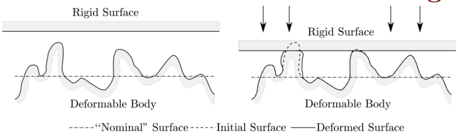




Joints Benchmarks (see jointmechanics.org)

Common theme:
Linear subcomponents joined through a non-linear contact interface undergoing small deformation vibrations

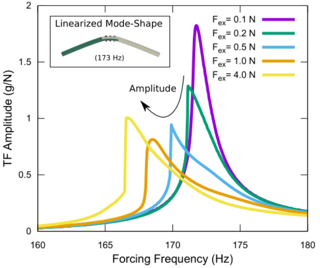
Interface Constitutive Modeling



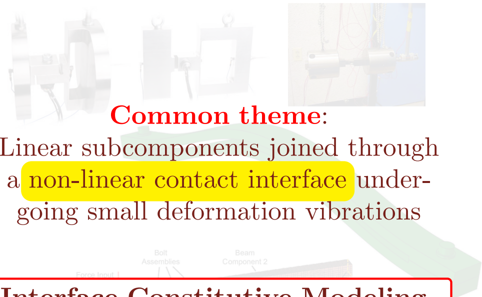
The research needs of the joints community are quite unique and specific

1. Introduction

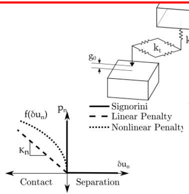
Nonlinear Responses

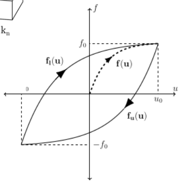


Joints Benchmarks (see jointmechanics.org)

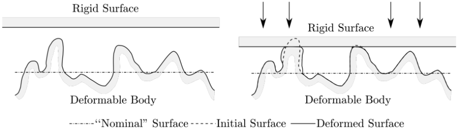


Common theme:
Linear subcomponents joined through a non-linear contact interface undergoing small deformation vibrations





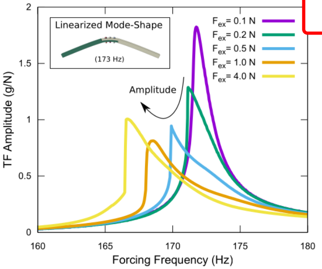
Interface Constitutive Modeling



The research needs of the joints community are quite unique and specific

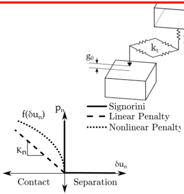
1. Introduction

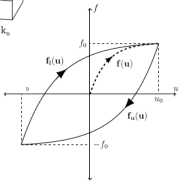
Nonlinear Responses



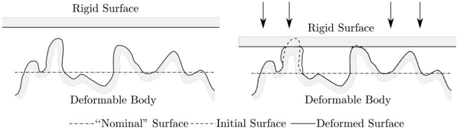
Goal: Off-Load as much as possible to **focus on joints**.

Common theme:
Linear subcomponents joined through a non-linear contact interface undergoing small deformation vibrations





Interface Constitutive Modeling

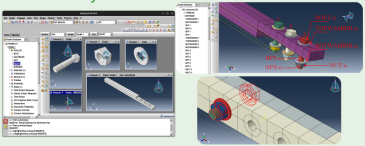


An Overview of the Modeling Pipeline

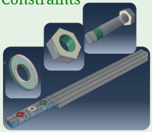
2.Outline of the Steps

Pre-Processing

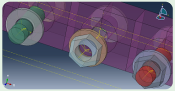
Assembly



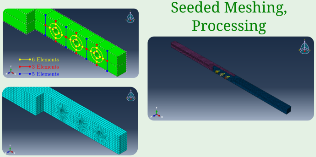
Constraints



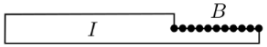
Bolt Prestress as Static Load



Seeded Meshing, Processing



Component Mode Synthesis



"Fixed interface" Modes: $[\mathbf{K}_{II} - \Lambda_{II}\mathbf{M}_{II}]\,\mathbf{\Psi}_I = \mathbf{0}$
"Constraint" Modes: $\mathbf{\Phi}_C = -\mathbf{K}_{II}^{-1}\mathbf{K}_{IB}$

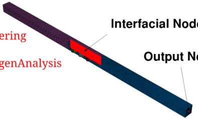
$$\underbrace{\begin{Bmatrix} U_B \\ U_I \end{Bmatrix}}_{U_{FULL} \in \mathcal{R}^{|B|+|I|}} := \underbrace{\begin{bmatrix} \mathcal{I}_B & \\ \Phi_C & \Psi_I \end{bmatrix}}_{\mathbf{T}_{HCB}} \underbrace{\begin{Bmatrix} U_B \\ \eta \end{Bmatrix}}_{U_{HCB} \in \mathcal{R}^{|B|+|\eta|}}$$

$$\mathbf{M}\ddot{U} + \mathbf{K}U = F$$

$$\downarrow$$

$$\mathbf{M}_{HCB}\ddot{U}_{HCB} + \mathbf{K}_{HCB}U_{HCB} = \mathbf{T_{HCB}}^T F$$

+ Reorganize NodeSet numbering
+ Choose recovery nodes
+ Conduct Fixed Interface EigenAnalysis



Matrix Extraction

+ Use the provided python script to convert the mtx files into MATLAB mat files

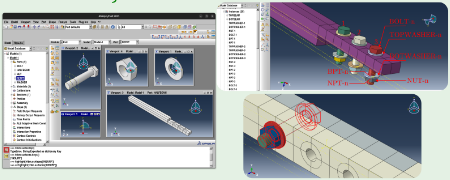
```
** ----------------------------------------------------------------
**
** STEP: HCBCMS
**
*Step, name=HCBCMS, nlgeom=NO
*Substructure Generate, overwrite, type=Z1, recovery matrix=YES,
      nset=OutNodes, mass matrix=YES
*Select Eigenmodes, generate
1, 20, 1
*Damping Controls, structural=COMBINED, viscous=COMBINED
*Retained Nodal Dofs, sorted=NO
BOTS_NDS, 1, 3
TOPS_NDS, 1, 3
**
** LOAD CASES
**
*Substructure Load Case, name=LCASE
** Name: BoltLoad-1   Type: Concentrated force   Scale factor: 1
*Cload
BPT-1_Set-1, 3, 1.
** Name: BoltLoad-2   Type: Concentrated force   Scale factor: 1
*Cload
BPT-2_Set-1, 3, 1.
** Name: BoltLoad-3   Type: Concentrated force   Scale factor: 1
*Cload
BPT-3_Set-1, 3, 1.
** Name: NutLoad-1   Type: Concentrated force   Scale factor: 1
*Cload
NPT-1_Set-1, 3, -1.
** Name: NutLoad-2   Type: Concentrated force   Scale factor: 1
*Cload
NPT-2_Set-1, 3, -1.
** Name: NutLoad-3   Type: Concentrated force   Scale factor: 1
*Cload
NPT-3_Set-1, 3, -1.
*Substructure Matrix Output, FILE NAME=Modelmats, MASS=YES,
      STIFFNESS=YES, SLOAD=YES, RECOVERY MATRIX=YES
*End Step
```

An Overview of the Modeling Pipeline

2.Outline of the Steps

Pre-Processing

Assembly



Load

Seeded Meshing, Processing

Component Mode Synthesis

"Fixed interface" Modes: $[K_{II} - \lambda_i M_{II}]\Phi_I = 0$
"Constraint" Modes: $\Phi_C = -K_{II}^{-1} K_{IB}$

$$\underbrace{\begin{Bmatrix} U_B \\ U_I \end{Bmatrix}}_{U_{FULL}} := \underbrace{\begin{bmatrix} I_B & \\ \Phi_C & \Phi_I \end{bmatrix}}_{T_{HCB}} \underbrace{\begin{Bmatrix} U_B \\ \eta \end{Bmatrix}}_{U_{HCB}}$$

$$M\ddot{U} + KU = F$$

$$M_{HCB}\ddot{U}_{HCB} + K_{HCB}U_{HCB} = T_{HCB}^T F$$

+ Reorganize NodeSet numbering
+ Choose recovery nodes
+ Conduct Fixed Interface EigenAnalysis

Matrix Extraction

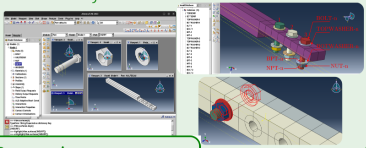
+ Use the provided python script to convert the mtx files into MATLAB mat files

An Overview of the Modeling Pipeline

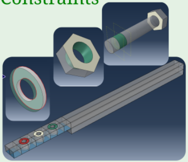
2.Outline of the Steps

Pre-Processing

Assembly



Constraints



Bolt Prestress as Static Load

Seeded Meshing, Processing

Component Mode Synthesis

"Fixed interface" Modes: $[K_{II} - \lambda_i M_{II}]\Phi_I = 0$
"Constraint" Modes: $\Phi_C = -K_{II}^{-1}K_{IB}$

$$M\ddot{U} + KU = F$$

$$M_{HCB}\ddot{U}_{HCB} + K_{HCB}U_{HCB} = T_{HCB}^T F$$

+ Reorganize NodeSet numbering
+ Choose recovery nodes
+ Conduct Fixed Interface EigenAnalysis

Matrix Extraction

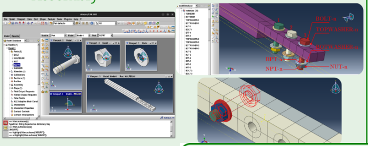
+ Use the provided python script to convert the mtx files into MATLAB mat files

An Overview of the Modeling Pipeline

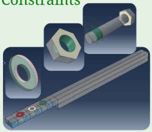
2.Outline of the Steps

Pre-Processing

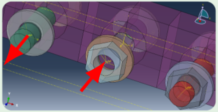
Assembly



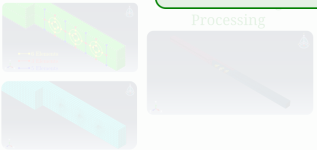
Constraints



Bolt Prestress as Static Load



Processing



Component Mode Synthesis

"Fixed interface" Modes:
"Constraint" Modes:

$$M\ddot{U} + KU = F$$

$$M_{HCB}\ddot{U}_{HCB} + K_{HCB}U_{HCB} = T_{HCB}{}^{T}F$$

+ Reorganize NodeSet numbering
+ Choose recovery nodes
+ Co

"Pull the Bolts, Push the Washers"

Matrix Extraction

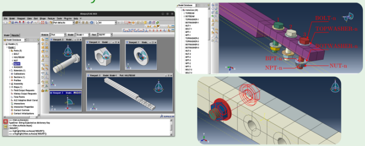
+ Use the provided python script to convert the mtx files into MATLAB mat files

An Overview of the Modeling Pipeline

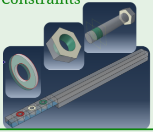
2.Outline of the Steps

Pre-Processing

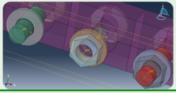
Assembly



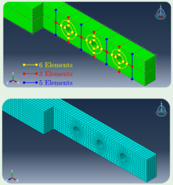
Constraints

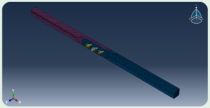


Bolt Prestress as Static Load



Seeded Meshing, Processing





Component Mode Synthesis

"Fixed interface" Modes: $[K_{II} - \lambda_{II} M_{II}]\Phi_I = 0$

"Constraint" Modes: $\Phi_C = -K_{II}^{-1} K_{IB}$

$$M\ddot{U} + KU = F$$

$$M_{HCB}\ddot{U}_{HCB} + K_{HCB}U_{HCB} = T_{HCB}^T F$$

+ Reorganize NodeSet numbering
+ Choose recovery nodes
+ Conduct Fixed Interface EigenAnalysis

Matrix Extraction

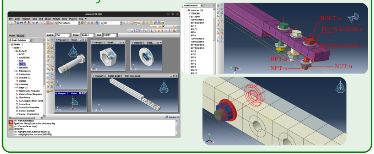
+ Use the provided python script to convert the mtx files into MATLAB mat files

An Overview of the Modeling Pipeline

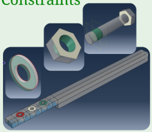
2.Outline of the Steps

Pre-Processing

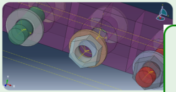
Assembly



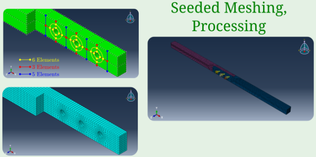
Constraints



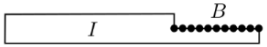
Bolt Prestress as Static Load



Seeded Meshing, Processing



Component Mode Synthesis



"Fixed interface" Modes: $[\mathbf{K}_{II} - \lambda_I \mathbf{M}_{II}]\,\boldsymbol{\Psi}_I = \mathbf{0}$
"Constraint" Modes: $\boldsymbol{\Phi}_C = -\mathbf{K}_{II}^{-1}\mathbf{K}_{IB}$

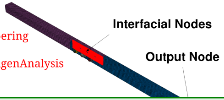
$$\underbrace{\begin{Bmatrix} U_B \\ U_I \end{Bmatrix}}_{\substack{U_{FULL} \\ \in \mathcal{R}^{|B|+|I|}}} := \underbrace{\begin{bmatrix} \mathcal{I}_B & \\ \Phi_C & \Psi_I \end{bmatrix}}_{\mathbf{T}_{HCB}} \underbrace{\begin{Bmatrix} U_B \\ \eta \end{Bmatrix}}_{\substack{U_{HCB} \\ \in \mathcal{R}^{|B|+|\eta|}}}$$

$$\mathbf{M}\ddot{U} + \mathbf{K}U = F$$

$$\downarrow$$

$$\mathbf{M}_{HCB}\ddot{U}_{HCB} + \mathbf{K}_{HCB}U_{HCB} = \mathbf{T_{HCB}}^T F$$

+ Reorganize NodeSet numbering
+ Choose recovery nodes
+ Conduct Fixed Interface EigenAnalysis



Matrix Extraction

+ Use the provided python script to convert the mtx files into MATLAB mat files

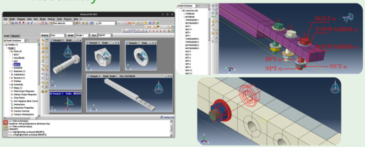
```
** ----------------------------------------------------------------
**
** STEP: HCBCMS
**
*Step, name=HCBCMS, nlgeom=NO
*Substructure Generate, overwrite, type=Z1, recovery matrix=YES,
      nset=OutNodes, mass matrix=YES
*Select Eigenmodes, generate
1, 20, 1
*Damping Controls, structural=COMBINED, viscous=COMBINED
*Retained Nodal Dofs, sorted=NO
BOTS_NDS, 1, 3
TOPS_NDS, 1, 3
**
** LOAD CASES
**
*Substructure Load Case, name=LCASE
** Name: BoltLoad-1   Type: Concentrated force   Scale factor: 1
*Cload
BPT-1_Set-1, 3, 1.
** Name: BoltLoad-2   Type: Concentrated force   Scale factor: 1
*Cload
BPT-2_Set-1, 3, 1.
** Name: BoltLoad-3   Type: Concentrated force   Scale factor: 1
*Cload
BPT-3_Set-1, 3, 1.
** Name: NutLoad-1   Type: Concentrated force   Scale factor: 1
*Cload
NPT-1_Set-1, 3, -1.
** Name: NutLoad-2   Type: Concentrated force   Scale factor: 1
*Cload
NPT-2_Set-1, 3, -1.
** Name: NutLoad-3   Type: Concentrated force   Scale factor: 1
*Cload
NPT-3_Set-1, 3, -1.
*Substructure Matrix Output, FILE NAME=Modelmats, MASS=YES,
      STIFFNESS=YES, SLOAD=YES, RECOVERY MATRIX=YES
*End Step
```

An Overview of the Modeling Pipeline

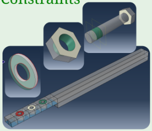
2.Outline of the Steps

Pre-Processing

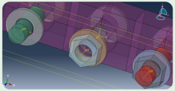
Assembly



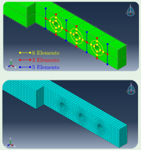
Constraints

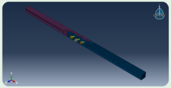


Bolt Prestress as Static Load

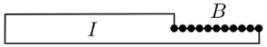


Seeded Meshing, Processing





Component Mode Synthesis



"Fixed interface" Modes: $[\mathbf{K}_{II} - \Lambda_{II}\mathbf{M}_{II}]\,\mathbf{\Psi}_I = \mathbf{0}$

"Constraint" Modes: $\mathbf{\Phi}_C = -\mathbf{K}_{II}^{-1}\mathbf{K}_{IB}$

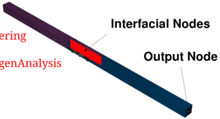
$$\underbrace{\begin{Bmatrix} U_B \\ U_I \end{Bmatrix}}_{U_{FULL} \in \mathcal{R}^{|B|+|I|}} := \underbrace{\begin{bmatrix} \mathcal{I}_B & \\ \Phi_C & \Psi_I \end{bmatrix}}_{\mathbf{T}_{HCB}} \underbrace{\begin{Bmatrix} U_B \\ \eta \end{Bmatrix}}_{U_{HCB} \in \mathcal{R}^{|B|+|\eta|}}$$

$$\mathbf{M}\ddot{U} + \mathbf{K}U = F$$

$$\downarrow$$

$$\mathbf{M}_{HCB}\ddot{U}_{HCB} + \mathbf{K}_{HCB}U_{HCB} = \mathbf{T_{HCB}}^T F$$

+ Reorganize NodeSet numbering
+ Choose recovery nodes
+ Conduct Fixed Interface EigenAnalysis



Matrix Extraction

+ Use the provided python script to convert the mtx files into MATLAB mat files

```
** ----------------------------------------------------------------
**
** STEP: HCBCMS
**
*Step, name=HCBCMS, nlgeom=NO
*Substructure Generate, overwrite, type=Z1, recovery matrix=YES,
      nset=OutNodes, mass matrix=YES
*Select Eigenmodes, generate
1, 20, 1
*Damping Controls, structural=COMBINED, viscous=COMBINED
*Retained Nodal Dofs, sorted=NO
BOTS_NDS, 1, 3
TOPS_NDS, 1, 3
**
** LOAD CASES
**
*Substructure Load Case, name=LCASE
** Name: BoltLoad-1   Type: Concentrated force   Scale factor: 1
*Cload
BPT-1_Set-1, 3, 1.
** Name: BoltLoad-2   Type: Concentrated force   Scale factor: 1
*Cload
BPT-2_Set-1, 3, 1.
** Name: BoltLoad-3   Type: Concentrated force   Scale factor: 1
*Cload
BPT-3_Set-1, 3, 1.
** Name: NutLoad-1   Type: Concentrated force   Scale factor: 1
*Cload
NPT-1_Set-1, 3, -1.
** Name: NutLoad-2   Type: Concentrated force   Scale factor: 1
*Cload
NPT-2_Set-1, 3, -1.
** Name: NutLoad-3   Type: Concentrated force   Scale factor: 1
*Cload
NPT-3_Set-1, 3, -1.
*Substructure Matrix Output, FILE NAME=Modelmats, MASS=YES,
      STIFFNESS=YES, SLOAD=YES, RECOVERY MATRIX=YES
*End Step
```

2.1. Relative Coordinates Pipeline

Outline of the Steps

- Suppose that $\underline{u}_T, \underline{u}_B, \underline{u}_I$ are the vectors of top, bottom, and internal nodal DoFs.
- Then the governing equations look like:

$$\begin{bmatrix} \underline{\underline{M}}_{TT} & \underline{\underline{M}}_{TB} & \underline{\underline{M}}_{TI} \\ & \underline{\underline{M}}_{BB} & \underline{\underline{M}}_{BI} \\ \text{sym} & & \underline{\underline{M}}_{II} \end{bmatrix} \begin{bmatrix} \ddot{\underline{u}}_T \\ \ddot{\underline{u}}_B \\ \ddot{\underline{u}}_I \end{bmatrix} + \begin{bmatrix} \underline{\underline{K}}_{TT} & \underline{\underline{K}}_{TB} & \underline{\underline{K}}_{TI} \\ & \underline{\underline{K}}_{BB} & \underline{\underline{K}}_{BI} \\ \text{sym} & & \underline{\underline{K}}_{II} \end{bmatrix} \begin{bmatrix} \underline{u}_T \\ \underline{u}_B \\ \underline{u}_I \end{bmatrix} + \begin{bmatrix} \underline{F}_T^{(c)} \\ \underline{F}_B^{(c)} \\ \underline{0} \end{bmatrix} = \underline{F}_e(t)$$

2.1. Relative Coordinates Pipeline

Outline of the Steps

- Suppose that $\underline{u}_T, \underline{u}_B, \underline{u}_I$ are the vectors of top, bottom, and internal nodal DoFs.
- Then the governing equations look like:

$$\begin{bmatrix} \underline{\underline{M}}_{TT} & \underline{\underline{M}}_{TB} & \underline{\underline{M}}_{TI} \\ & \underline{\underline{M}}_{BB} & \underline{\underline{M}}_{BI} \\ \text{sym} & & \underline{\underline{M}}_{II} \end{bmatrix} \begin{bmatrix} \underline{\ddot{u}}_T \\ \underline{\ddot{u}}_B \\ \underline{\ddot{u}}_I \end{bmatrix} + \begin{bmatrix} \underline{\underline{K}}_{TT} & \underline{\underline{K}}_{TB} & \underline{\underline{K}}_{TI} \\ & \underline{\underline{K}}_{BB} & \underline{\underline{K}}_{BI} \\ \text{sym} & & \underline{\underline{K}}_{II} \end{bmatrix} \begin{bmatrix} \underline{u}_T \\ \underline{u}_B \\ \underline{u}_I \end{bmatrix} + \begin{bmatrix} \underline{F}_T^{(c)} \\ \underline{F}_B^{(c)} \\ \underline{0} \end{bmatrix} = \underline{F}_e(t)$$

- Under the relative coordinate transformation,

$$\begin{bmatrix} \underline{u}_T \\ \underline{u}_B \\ \underline{u}_I \end{bmatrix} = \begin{bmatrix} \underline{\underline{I}}_T & \underline{\underline{I}}_T & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{I}}_T & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{0}} & \underline{\underline{I}}_I \end{bmatrix} \begin{bmatrix} \Delta\underline{u} \\ \underline{u}_B \\ \underline{u}_I \end{bmatrix}, \qquad \Delta\underline{u} = \underline{u}_T - \underline{u}_B.$$

2.1. Relative Coordinates Pipeline

Outline of the Steps

- Suppose that $\underline{u}_T, \underline{u}_B, \underline{u}_I$ are the vectors of top, bottom, and internal nodal DoFs.
- Then the governing equations look like:

$$\begin{bmatrix} \underline{\underline{M}}_{TT} & \underline{\underline{M}}_{TB} & \underline{\underline{M}}_{TI} \\ & \underline{\underline{M}}_{BB} & \underline{\underline{M}}_{BI} \\ \text{sym} & & \underline{\underline{M}}_{II} \end{bmatrix} \begin{bmatrix} \underline{\ddot{u}}_T \\ \underline{\ddot{u}}_B \\ \underline{\ddot{u}}_I \end{bmatrix} + \begin{bmatrix} \underline{\underline{K}}_{TT} & \underline{\underline{K}}_{TB} & \underline{\underline{K}}_{TI} \\ & \underline{\underline{K}}_{BB} & \underline{\underline{K}}_{BI} \\ \text{sym} & & \underline{\underline{K}}_{II} \end{bmatrix} \begin{bmatrix} \underline{u}_T \\ \underline{u}_B \\ \underline{u}_I \end{bmatrix} + \begin{bmatrix} \underline{F}_T^{(c)} \\ \underline{F}_B^{(c)} \\ \underline{0} \end{bmatrix} = \underline{F}_e(t)$$

- Under the relative coordinate transformation,

Transformed Equations of Motion

$$\begin{bmatrix} \underline{\underline{M}}_{TT} & \underline{\underline{M}}_{TT} + \underline{\underline{M}}_{TB} & \underline{\underline{M}}_{TI} \\ & \begin{matrix} \underline{\underline{M}}_{TT} + \underline{\underline{M}}_{BB} + \\ \underline{\underline{M}}_{TB} + \underline{\underline{M}}_{TB}^T \end{matrix} & \underline{\underline{M}}_{TI} + \underline{\underline{M}}_{BI} \\ \text{sym} & & \underline{\underline{M}}_{II} \end{bmatrix} \begin{bmatrix} \underline{\ddot{u}}_T \\ \underline{\ddot{u}}_B \\ \underline{\ddot{u}}_I \end{bmatrix} + \begin{bmatrix} & & \\ & \overline{K} & \\ & & \end{bmatrix} \begin{bmatrix} \underline{u}_T \\ \underline{u}_B \\ \underline{u}_I \end{bmatrix} + \begin{bmatrix} \underline{F}_T^{(c)} \\ \underline{0} \\ \underline{0} \end{bmatrix} = \underline{F}_e(t).$$

2.1. Relative Coordinates Pipeline

Outline of the Steps

- Suppose that $\underline{u}_T, \underline{u}_B, \underline{u}_I$ are the vectors of top, bottom, and internal nodal DoFs.
- Then the governing equations look like:

$$\begin{bmatrix} \underline{\underline{M}}_{TT} & \underline{\underline{M}}_{TB} & \underline{\underline{M}}_{TI} \\ & \underline{\underline{M}}_{BB} & \underline{\underline{M}}_{BI} \\ \text{sym} & & \underline{\underline{M}}_{II} \end{bmatrix} \begin{bmatrix} \underline{\ddot{u}}_T \\ \underline{\ddot{u}}_B \\ \underline{\ddot{u}}_I \end{bmatrix} + \begin{bmatrix} \underline{\underline{K}}_{TT} & \underline{\underline{K}}_{TB} & \underline{\underline{K}}_{TI} \\ & \underline{\underline{K}}_{BB} & \underline{\underline{K}}_{BI} \\ \text{sym} & & \underline{\underline{K}}_{II} \end{bmatrix} \begin{bmatrix} \underline{u}_T \\ \underline{u}_B \\ \underline{u}_I \end{bmatrix} + \begin{bmatrix} \underline{F}_T^{(c)} \\ \underline{F}_B^{(c)} \\ \underline{0} \end{bmatrix} = \underline{F}_e(t)$$

- Under the relative coordinate transformation,

Transformed Equations of Motion

$$\begin{bmatrix} \underline{\underline{M}}_{TT} & \underline{\underline{M}}_{TT} + \underline{\underline{M}}_{TB} & \underline{\underline{M}}_{TI} \\ & \underline{\underline{M}}_{TT} + \underline{\underline{M}}_{BB} + \underline{\underline{M}}_{TB} + \underline{\underline{M}}_{TB}^T & \underline{\underline{M}}_{TI} + \underline{\underline{M}}_{BI} \\ \text{sym} & & \underline{\underline{M}}_{II} \end{bmatrix} \begin{bmatrix} \underline{\ddot{u}}_T \\ \underline{\ddot{u}}_B \\ \underline{\ddot{u}}_I \end{bmatrix} + \begin{bmatrix} \overline{K} \end{bmatrix} \begin{bmatrix} \underline{u}_T \\ \underline{u}_B \\ \underline{u}_I \end{bmatrix} + \begin{bmatrix} \underline{F}_T^{(c)} \\ \underline{0} \\ \underline{0} \end{bmatrix} = \underline{F}_e(t).$$

2.1. Relative Coordinates Pipeline

Outline of the Steps

- Suppose that $\underline{u}_T, \underline{u}_B, \underline{u}_I$ are the vectors of top, bottom, and internal nodal DoFs.
- Then the governing equations look like:

$$\begin{bmatrix} \underline{\underline{M}}_{T} \cdots \\ \\ \text{sym} \cdots \end{bmatrix} \cdots = \underline{F}_e(t)$$

It is clearly advantageous (when possible) to use the relative coordinate transformation as early as possible in the CMS process.
Thankfully, this is possible fully in Abaqus itself!

- Under the relative coordinate transformation,

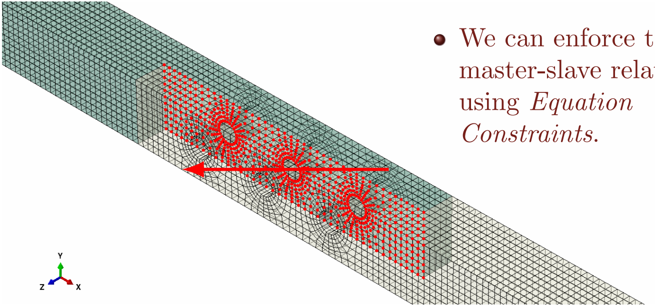
Transformed Equations of Motion

$$\begin{bmatrix} \underline{\underline{M}}_{TT} & \underline{\underline{M}}_{TT} + \underline{\underline{M}}_{TB} & \underline{\underline{M}}_{TI} \\ & \begin{matrix} \underline{\underline{M}}_{TT} + \underline{\underline{M}}_{BB} + \\ \underline{\underline{M}}_{TB} + \underline{\underline{M}}_{TB}^{T} \end{matrix} & \underline{\underline{M}}_{TI} + \underline{\underline{M}}_{BI} \\ \text{sym} & & \underline{\underline{M}}_{II} \end{bmatrix} \begin{bmatrix} \underline{\ddot{u}}_T \\ \underline{\ddot{u}}_B \\ \underline{\ddot{u}}_I \end{bmatrix} + \begin{bmatrix} & & \\ & \overline{K} & \\ & & \end{bmatrix} \begin{bmatrix} \underline{u}_T \\ \underline{u}_B \\ \underline{u}_I \end{bmatrix} + \begin{bmatrix} \underline{F}_T^{(c)} \\ \underline{0} \\ \underline{0} \end{bmatrix} = \underline{F}_e(t).$$

2.1. Relative Coordinates Pipeline

Outline of the Steps

- By adding Reference Points to the model at the location of the interfacial nodes, we can create **slave nodes whos DoF will be the relative DoF of the corresponding nodes on the interface**.
- This can be done quite easily with Python scripting (the website has a script for the BRB which can be adapted to arbitrary contexts).



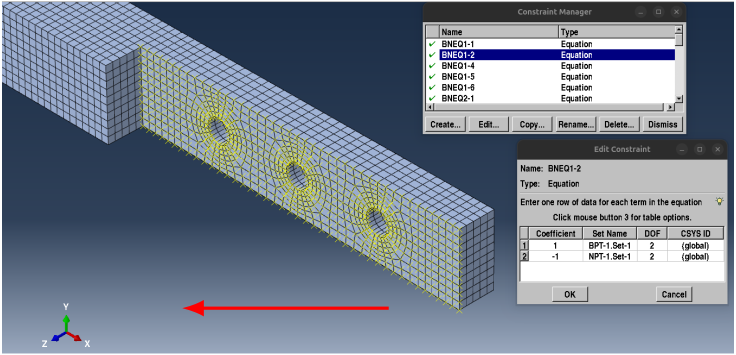
- We can enforce the master-slave relationship using *Equation Constraints.*

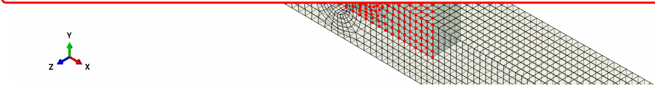
2.1. Relative Coordinates Pipeline

Outline of the Steps

- By adding Reference Points to the model at the location of the interfacial no ... ve D...
- Th ... as a scr...

...And this is what it looks like on ABAQUS.



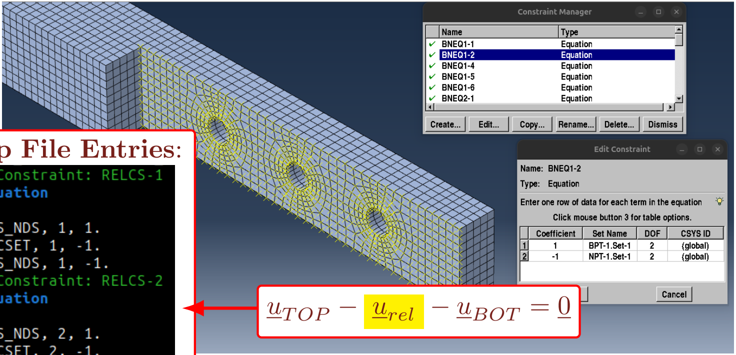


2.1. Relative Coordinates Pipeline

Outline of the Steps

- By adding Reference Points to the model at the location of the interfacial no… …ve D…
- Th… …as a scr…

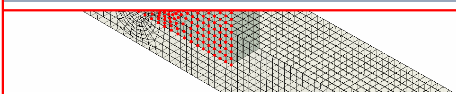
...And this is what it looks like on ABAQUS.



$$\underline{u}_{TOP} - \boxed{\underline{u}_{rel}} - \underline{u}_{BOT} = \underline{0}$$

.inp File Entries:

```
** Constraint: RELCS-1
*Equation
3
TOPS_NDS, 1, 1.
RELCSET, 1, -1.
BOTS_NDS, 1, -1.
** Constraint: RELCS-2
*Equation
3
TOPS_NDS, 2, 1.
RELCSET, 2, -1.
BOTS_NDS, 2, -1.
** Constraint: RELCS-3
*Equation
3
TOPS_NDS, 3, 1.
RELCSET, 3, -1.
BOTS_NDS, 3, -1.
```



Nonlinear Analysis in Under 100 Lines of MATLAB Code!

3. Nonlinear Analysis in MATLAB

4. Outro and Recommendations

- When we're interested in working on friction modeling at the interfacial level, it makes sense to off-load everything else to a software.
- Try to think of ways to utilize the ABAQUS solvers as much as possible (for CMS modal analysis, etc.).

Some Drawbacks of this Pipeline

- Applicability is **strictly restricted to small displacement contact**.
- Geometrical Nonlinearities can't be present. If so, try an ICE fitting coupled with the relative coordinates approach.

Thank You!

All the detailed instructions are hosted through Github in a repository named Nidish96/Abaqus4Joints

Comments, suggestions and contributions welcome!
Please email me at nidish@iitm.ac.in.

References I