MERCURY

Optimized Software for (Single Site) Hybrid Simulation

From Pseudo Dynamics to Real Time Hybrid Simulation

Student Manual

Ver. 1

Dae-Hung Kang

Gary Haussmann

Victor E. Saouma

Fast Hybrid Testing Laboratory

http://fht.colorado.edu

Department of Civil Environmental and Architectural Engineering
University of Colorado, Boulder
Boulder, Colorado 80309-0428

Contents

List of Figures

List of Tables

1 Introduction

A reasonable understanding of Structural Analysis requires the knowledge of the inner working of the direct stiffness method which is at the heart of most/all structural analysis computer programs.

Hence, this term project assignment will provide you with an opportunity to examine the “guts” of a very well written analysis code, Mercury, to (finally) put into application your programming skills (Matlab) into action, and to possibly sharpen them.

Mercury[1] is a computer program developed by the Fast Hybrid Testing laboratory of the University of Colorado through funding from both the State of Colorado and the Network for Earthquake Engineering Simulation (NEES). It has capabilities for static and dynamic analysis, linear and nonlinear, and a rich library of elements and constitutive models. Two versions of the code exist, one in Matlab (used for original development), and another in c++ used for deployment in our laboratory. This later version is optimized for speed through parallel processing and multithreading.

A subset of the code has been extracted, and labeled Mercury-edu. You will be given a version of Mercury-edu from which key files or Matlab statements have been deleted. Your term project consists in completing this code.

You will be assessed through the submittal of your term project, and a separate quiz.

It will be assumed that you have some familiarity with Matlab (gained through undergraduate courses and/or the tutorials you were provided). However, to fully understand the syntax of Mercury, you will need to familiarize yourself with two additional constructs. Those are described in the following section.

A final note, those of you interested in deepening your understanding of structural analysis should consider taking the Nonlinear Analysis of Structures course being planned for the Spring and in which you will further explore Mercury.

2 Matlab Revisited

You were given two tutorial files in Matlab, and highlighted in yellow were the sections you should be familiar with to work on your term project. There are nevertheless two Matlab constructs that were not discussed, and which you should be familiar with: `case` and `structure`. Those will be briefly addressed next.

2.1 Switch Case

http://blogs.mathworks.com/pick/2008/01/02/matlab-basics-switch-case-vs-if-elseif/ In the following example, depending on the string value of `eletype` different functions are invoked.

```
switch eletype
        case 'Simple2DTruss'
            [tmpeleinfo] = Simple2DTrussInfo(eleinfo);
        case 'Simple3DTruss'
            [tmpeleinfo] = Simple3DTrussInfo(eleinfo);
        case 'StiffnessBasedBeam'
            [tmpeleinfo] = StiffnessBasedBeamInfo(eleinfo);
        case 'StiffnessBased2DBeamColumn'
            [tmpeleinfo] = StiffnessBased2DBeamColumnInfo(eleinfo);
        case 'StiffnessBased3DBeamColumn'
            [tmpeleinfo] = StiffnessBased3DBeamColumnInfo(eleinfo);
        case 'Grid'
            [tmpeleinfo] = GridInfo(eleinfo);
    end
```

3 Mercury.edu Internals

3.1 Structure Array

Structures are a useful way of grouping MATLAB variables that belong together. In our case, there are a number of information associated with nodes: x and y coordinates, constraints, load. Similarly, associated with an element are: first and second node, Young’s modulus, cross sectional area. Hence rather than storing each one of those data into disjointed arrays (such as x(:) and y(:)), we can group them together into a single entity. This will facilitate data definition, and data transfer to functions.

Some explanation:

- `str.ID(ele(iele).(eletype).enode,6)` corresponds to xxx;
- `str.elecoord(iele,4) = str.nodcoord(ele(iele).(eletype).enode,2)` corresponds to xxx;;
- `str.LM(iele, 8) = str.ID(ele(iele).(eletype).enode,2)` corresponds to xxx;

[1] http://fht.colorado.edu/Mercury

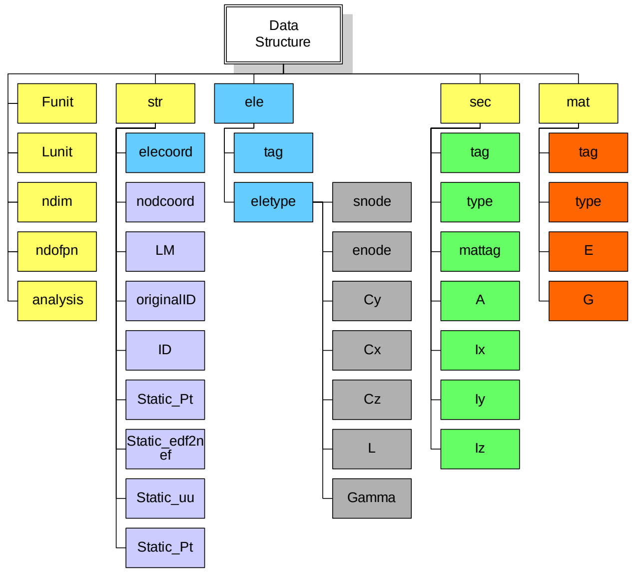

Figure 1: Mercury Data Structure

3.2 Internal Files

Files are internally organized as shown in Fig. 2

4 Mercury-edu USER's MANUAL

This document describes the input for Mercury Education.

4.1 Preface Block

This initial block declares units, spatial dimension of the structure, and the number of degrees of freedom per node.

4.1.1 Unit

The `Unit` declares selected units for analysis.

`Unit` = {'F', 'L'}

`F`: force, and `L`: length units. For example **Unit = {'kN', 'mm'}**

4.1.2 Structural mode

The `StrMode` declares dimension of structure and number of degrees of freedom per node.

`StrMode` = [`ndim`, `ndofpn`]

---

where `ndim` refers to the spatial dimension of the structure [2|3] and `ndofpn` to the number of degrees of freedom per node in global reference [2|3|6]

4.2 Control Block

The control block defines basic information about the structural analysis.

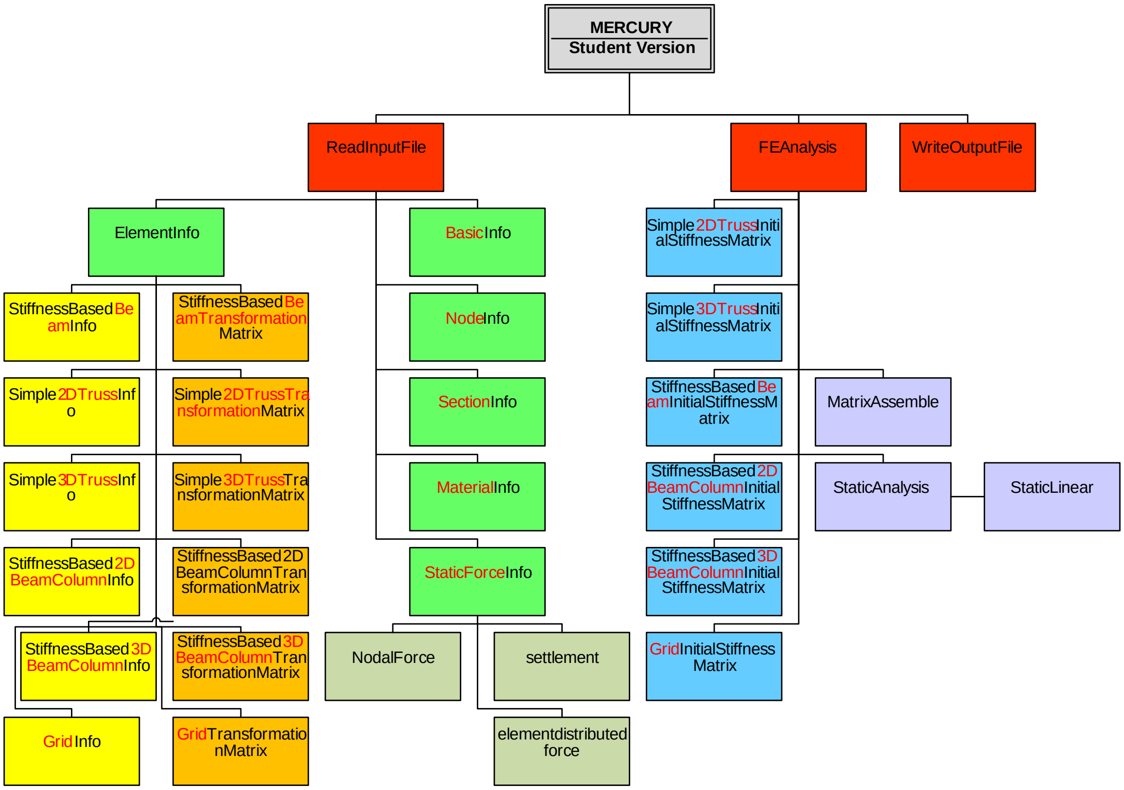

Figure 2: Mercury.edu Files

4.2.1 Analysis

The `Analysis` defines the analysis mode.

```
Analysis = 'AnalysisMode'
```

where `AnalysisMode`: [Static].

4.3 Geometry Block

The geometry block defines nodal coordinates and their constraints assuming a right handed coordinate system.

4.3.1 Nodal coordinates

The `nodcoord` assigns coordinates of nodes.

nodcoord	=	{	$\texttt{nodtag}_1,\ x_1,\ y_1\ [,\ z_1]$	;
			...	;
			$\texttt{nodtag}_i,\ x_i,\ y_i\ [,\ z_i]$	;
			...	;
			$\texttt{nodtag}_n,\ x_n,\ y_n\ [,\ z_n]$	}

for example:

Node	=	{	1,	0.0,	0.0	;
			2,	1.0,	3.0	;
			3,	2.0,	0.0	}

4.3.2 Boundary condition

The `constraint` command assigns boundary conditions to the nodes. Each node has to have as many constraint as d.o.f's per node.

constraint	= {	$\texttt{nodtag}^1,\ id_1^1,\ id_2^1\ [,\ id_3^1,\ id_4^1,\ id_5^1,\ id_6^1]$	;
		...	;
		$\texttt{nodtag}^i,\ id_1^i,\ id_2^i\ [,\ id_3^i,\ id_4^i,\ id_5^i,\ id_6^i]$	;
		...	;
		$\texttt{nodtag}^n,\ id_1^n,\ id_2^n\ [,\ id_3^n,\ id_4^n,\ id_5^n,\ id_6^n]$	}

Where 0 corresponds to a free dof, and 1 to a fixed one. For example:

constraint	=	{	3,	1,	1	;
			5,	1,	0	}

4.4 Element Block

The `elements` command defines element type, nodal connectivity, and basic sectional information. These may vary with the element type.

elements	= {	$\{\ \texttt{eletag}_1,\ \texttt{eletype}_1,\ \texttt{in}_1,\ \texttt{jn}_1,\ \texttt{SecInfo}_1\ \}$	;
		...	;
		$\{\ \texttt{eletag}_i,\ \texttt{eletype}_i,\ \texttt{in}_i,\ \texttt{jn}_i,\ \texttt{SecInfo}_i\ \}$	;
		...	;
		$\{\ \texttt{eletag}_n,\ \texttt{eletype}_n,\ \texttt{in}_n,\ \texttt{jn}_n,\ \texttt{SecInfo}_n\ \}$	}

Where

- $\texttt{eletag}_i$: Sequential integer identifying the i^{th} element
- $\texttt{eletype}_i$: i^{th} element type
- $\texttt{in}_i$: First node
- $\texttt{jn}_i$: Second node

- `SecInfo`$_i$: Basic section information for the element (see below)

where, `eletype`: Simple2DTruss, Simple3DTruss, StiffnessBasedBeam, StiffnessBased2DBeamColumn, StiffnessBased3DBeamColumn, and Grid.

4.5 Section Block

Section block declares section properties. `sections` defines section types, section properties, and basic material information of section. Description on sections may be different depending on types of section.

```
sections   = {  sectag_1, sectype_1, { SecProp_1 }   ;
                ...                                  ;
                sectag_i, sectype_i, { SecProp_i }   ;
                ...                                  ;
                sectag_n, sectype_n, { SecProp_n }   }
```

Where:

- `sectag`$_i$: Sequential integer number identifying section at i^{th} section
- `sectype`$_i$: Section type at i^{th} section
- `SecProp`$_i$: Section properties and basic material information on i^{th} section

4.5.1 General section

General section has only one layer or fiber, and it in `Mercury` only supports elastic material currently. Usually, if section has nonlinear material, user may use multi-layer or multi-fiber section.

```
sectag, 'General', { mattag, A, Ix, Iy, Iz }
```

Where:

- `sectag`: Section tag
- `mattag`: Integer number identifying material with `sectag`
- `A`: Section area with `sectag`
- `Ix`: Moment inertia on x-axis with `sectag`
- `Iy`: Moment inertia on y-axis with `sectag`
- `Iz`: Moment inertia on z-axis with `sectag`

4.6 Material Block

Material block declares material properties.

```
materials   = {  { mattag_1, mattype_1, E_1, G_1 }   ;
                 ...                                 ;
                 { mattag_i, mattype_i, E_i, G_i }   ;
                 ...                                 ;
                 { mattag_n, mattype_n, E_n, G_n }   }
```

where:

- `mattag`$_i$: Consecutive integer number identifying material at i^{th} material
- `mattype`$_i$: Material type at i^{th} material
- `E`$_i$: Young's modulus of a material with `mattag`
- `G`$_i$: Shear modulus of a material with `mattag`

where, `mattype`: Elastic.

4.7 Force Block

Force block declares all the external forces on the structure.

`forces`	= {	`forcetype`$_1$, { `ForceProp`$_1$ }	;
		...	;
		`forcetype`$_i$, { `ForceProp`$_i$ }	;
		...	;
		`forcetype`$_n$, { `ForceProp`$_n$ }	}

Where:

- `forcetype`$_i$: is type of force [`NodalForce| Settlement| ElementDistributedForce`] Force type of i^{th}
- `ForceProp`$_i$: Force properties of i^{th} `forcetype`

4.7.1 Nodal force

'`NodalForce`',	{	`nodtag`$_1$, `gx`$_1$, `magnitude`$_1$	;
		...	;
		`nodtag`$_i$, `gx`$_i$, `magnitude`$_i$	;
		...	;
		`nodtag`$_n$, `gx`$_n$, `magnitude`$_n$	}

Where

- `nodtag`$_i$: Node tag of i^{th} nodal force
- `gx`$_i$: Direction on global external force at node, [1—2—3] for X, or Y or Z global axis.
- `magnitude`$_i$: Magnitude of external force with node tag at i^{th} nodal force

4.7.2 Settlement

'`Settlement`',	{	`nodtag`$_1$, `gx`$_1$, `magnitude`$_1$	;
		...	;
		`nodtag`$_i$, `gx`$_i$, `magnitude`$_i$	;
		...	;
		`nodtag`$_n$, `gx`$_n$, `magnitude`$_n$	}

Where:

- `nodtag`$_i$: Node tag of i^{th} nodal displacement
- `gx`$_i$: Direction on global reference of external displacement with node tag at i^{th} nodal displacement; [1—2—3] for X, or Y or Z global axis.
- `magnitude`$_i$: Magnitude of external displacement with node tag at i^{th} nodal displacement

4.7.3 Element distributed force

'`ElementDistributedForce`',	{	`eletag`$_1$, `lx`$_1$, `magnitude`$_1$	;
		...	;
		`eletag`$_i$, `lx`$_i$, `magnitude`$_i$	;
		...	;
		`eletag`$_n$, `lx`$_n$, `magnitude`$_n$	}

Where:

- `eletag`$_i$: Element tag at i^{th} element distributed force
- `lx`$_i$: Direction on local reference of external force with element tag at i^{th} element distributed force; [1—2—3] for X, or Y or Z global axis.
- `magnitude`$_i$: Magnitude of external force with element tag at i^{th} element distributed force

5 Examples

5.1 Example 1

```
%==========================================================================
%    Mercury Matlab Ver. 1.0 for Education
%    Written    : Daehung Kang
%    Date       : September 1st, 2009
%    File name: ex01.m
%==========================================================================
% Description
% 1. Static analysis
% 2. Nodal force
% 3. Simple 2D truss element
% 5. General section
% 6. Elastic material
%--------------------------------------------------------------------------
% pp. 374, 9-22
% Structural Analysis Sixth Edition (2006) by R.C. Hibbeler
% Published by Pearson Prentice Hall
% A-36 Steel member, Cross-sectional area = 400mm^2, E=200GPa=200kN/mm^2
%--------------------------------------------------------------------------
% Preface
Unit       = {'kN', 'mm'};
%            ndim, ndofpn
StrMode    = [2, 2];
%--------------------------------------------------------------------------
% Control block
Analysis   = 'Static';
%--------------------------------------------------------------------------
% Geometry block
%              nodtag, x, y
nodcoord    = {1,    0, 0;
               2, 1500, 0;
               3, 3000, 0;
               4, 1500, 2000;
               5, 3000, 2000};
%              nodtag, x, y
constraint = {3, 1, 1;
              5, 1, 1};
%--------------------------------------------------------------------------
% Element block
%             {eletag, 'Simple2DTruss', in, jn, sectag}}
elements = { {1, 'Simple2DTruss', 1, 2, 1};
             {2, 'Simple2DTruss', 2, 3, 1};
             {3, 'Simple2DTruss', 1, 4, 1};
             {4, 'Simple2DTruss', 2, 4, 1};
             {5, 'Simple2DTruss', 3, 4, 1};
             {6, 'Simple2DTruss', 4, 5, 1} };
%--------------------------------------------------------------------------
% Section block
%              sectag, 'General', {mattag, A, Ix, Iy, Iz}
sections = { 1, 'General', {1, 400, 0, 0, 0} };
%--------------------------------------------------------------------------
% Material block
%               mattag, 'Elastic', E, G, density
materials = { {1, 'Elastic', 200, 0, 0} };
%--------------------------------------------------------------------------
% Force block
%             'NodalForce', {nodnum, globalaxis, m}
staticforces = { 'NodalForce', {1, 2, -30;
                                2, 2, -20}};
%--------------------------------------------------------------------------
```

5.2 Example 2

```
%==========================================================================
%    Mercury Matlab Ver. 1.0 for Education
%    Written    : Daehung Kang
%    Date       : September 1st, 2009
%    File name: ex02.m
%==========================================================================
% Description
% 1. Static analysis
% 2. Nodal force
% 3. Stiffness-based beam
% 5. General section
% 6. Elastic material
%--------------------------------------------------------------------------
% pp. 377, 9-63
% Structural Analysis Sixth Edition (2006) by R.C. Hibbeler
% Published by Pearson Prentice Hall
% Arbitary A and Iz for stiffness-based beam
% E = 200GPa = 200,000,000kPa, Iz = 9.50*10^6 mm^4 = 9.50*10^6 * 10^-12 m^4
%--------------------------------------------------------------------------
% Preface
Unit       = {'kN', 'm'};
%            ndim, ndofpn
StrMode    = [2, 2];
%--------------------------------------------------------------------------
% Control block
Analysis   = 'Static';
%--------------------------------------------------------------------------
% Geometry block
%              nodtag, x, y
nodcoord    = {1, 0, 0;
               2, 4, 0;
               3, 8, 0;};
%              nodtag, y, tz
constraint = {1, 1, 0;
```

```
                2, 1, 0};
%------------------------------------------------------------------
% Element block
%           {eletag, 'StiffnessBasedBeam', in, jn, sectag}}
elements = { {1, 'StiffnessBasedBeam', 1, 2, 1};
              {2, 'StiffnessBasedBeam', 2, 3, 1};};
%------------------------------------------------------------------
% Section block
%            sectag, 'General', {mattag, A, Ix, Iy, Iz}
sections = { 1, 'General', {1, 400, 0, 0, 9.50*10^6*10^-12} };
%------------------------------------------------------------------
% Material block
%             mattag, 'Elastic', E, G, density
materials = { {1, 'Elastic', 200000000, 0, 0} };
%------------------------------------------------------------------
% Force block
%           'NodalForce', {nodnum, globalaxis, m}
staticforces = { 'NodalForce', {3, 2,-12};};
%------------------------------------------------------------------
```

5.3 Example 3

```
%==================================================================
%    Mercury Matlab Ver. 1.0 for Education
%    Written   : Daehung Kang
%    Date      : September 1st, 2009
%    File name: ex03.m
%==================================================================
% Description
% 1. Static analysis
% 2. Element force and element distributed force
% 3. Stiffness-based beam
% 5. General section
% 6. Elastic material
%------------------------------------------------------------------
% pp. 378, 9-69
% Structural Analysis Sixth Edition (2006) by R.C. Hibbeler
% Published by Pearson Prentice Hall
% Arbitary A and Iz for stiffness-based beam
% E = 13.1 GPa = 13.1 kN/mm^2
% Iz = 120*180^3/12 mm^4
%------------------------------------------------------------------
% Preface
Unit       = {'kN', 'mm'};
%            ndim, ndofpn
StrMode    = [2, 2];
%------------------------------------------------------------------
% Control block
Analysis   = 'Static';
%------------------------------------------------------------------
% Geometry block
%           nodtag, x, y
nodcoord    = {1,     0, 0;
               2, 1500, 0;
               3, 4500, 0};
%           nodtag, y, tz
constraint = {2, 1, 0;
              3, 1, 0};
%------------------------------------------------------------------
% Element block
%           {eletag, 'StiffnessBasedBeam', in, jn, sectag}}
elements = { {1, 'StiffnessBasedBeam', 1, 2, 1};
              {2, 'StiffnessBasedBeam', 2, 3, 1}};
%------------------------------------------------------------------
% Section block
%            sectag, 'General', {mattag, A, Ix, Iy, Iz}
sections = { 1, 'General', {1, 400, 0, 0, 120*180^3/12}};
%------------------------------------------------------------------
% Material block
%             mattag, 'Elastic', E, G, density
materials = { {1, 'Elastic', 13.1, 0, 0} };
%------------------------------------------------------------------
% Force block
%             'ElementForce', {elenum, localaxis, distance from start node, m}
staticforces = { 'ElementForce', {2, 1, 1500, -8}};
%------------------------------------------------------------------
```

5.4 Example 4

```
%==================================================================
%    Mercury Matlab Ver. 1.0 for Education
%    Written   : Daehung Kang
%    Date      : September 1st, 2009
%    File name: ex04.m
%==================================================================
% Description
% 1. Static analysis
% 2. Nodal force
% 3. Simple3DTruss
% 5. General section
% 6. Elastic material
%------------------------------------------------------------------
% indat5 in casap
%------------------------------------------------------------------
% Preface
Unit       = {'kips', 'in'};
%            ndim, ndofpn
StrMode    = [3, 3];
```

```
%-----------------------------------------------------------------------
% Control block
Analysis  = 'Static';
%-----------------------------------------------------------------------
% Geometry block
%          nodtag, x, y, z
nodcoord = {1,  -72,   0,  96;
            2,  144,   0,  96;
            3,   72,   0, -96;
            4, -144,   0, -96;
            5,    0, 288,   0};
%          nodtag, x, y, z
constraint={1, 1, 1, 1;
            2, 1, 1, 1;
            3, 1, 1, 1;
            4, 1, 1, 1};
%-----------------------------------------------------------------------
% Element block
%          {eletag, 'Simple3DTruss', in, jn, sectag}}
elements = { {1, 'Simple3DTruss', 1, 5, 1};
             {2, 'Simple3DTruss', 2, 5, 1};
             {3, 'Simple3DTruss', 3, 5, 1};
             {4, 'Simple3DTruss', 4, 5, 1}};
%-----------------------------------------------------------------------
% Section block
%           sectag, 'General', {mattag, A, Ix, Iy, Iz}
sections = { 1, 'General', {1, 8.4, 0, 0, 0}};
%-----------------------------------------------------------------------
% Material block
%             mattag, 'Elastic', E, G, density
materials = { {1, 'Elastic', 10000, 0, 0} };
%-----------------------------------------------------------------------
% Force block
%          'NodalForce', {nodnum, globalaxis, m}
staticforces = { 'NodalForce', {5, 2, -100;
                                5, 3,  -50}};
%-----------------------------------------------------------------------
```

5.5 Example 5

```
%=======================================================================
%   Mercury Matlab Ver. 1.0 for Education
%   Written   : Daehung Kang
%   Date      : September 1st, 2009
%   File name: ex05.m
%=======================================================================
% Description
% 1. Static analysis
% 2. Element force
% 3. Stiffness-based 2D beam-column
% 5. General section
% 6. Elastic material
%-----------------------------------------------------------------------
% pp. 380, 9-94
% Structural Analysis Sixth Edition (2006) by R.C. Hibbeler
% Published by Pearson Prentice Hall
% E = 29*10^3 kips/in^2 = 29*10^3*12^2 kips/ft^2
% Iz_12 and Iz_34= 600 in^4 = 600*12^-4 ft^4,
% A_12 and A_34 = 100 in^2 = 100*12^-2 ft^2,
% Iz-23 = 900 in^4 = 900*12^-4 ft^4,
% A_23 = 100 in^2 = 100*12^-2 ft^2,
%-----------------------------------------------------------------------
% Preface
Unit       = {'kips', 'ft'};
%          ndim, ndofpn
StrMode    = [2, 3];
%-----------------------------------------------------------------------
% Control block
Analysis  = 'Static';
%-----------------------------------------------------------------------
% Geometry block
%          nodtag, x, y
nodcoord = {1,  0,  0;
            2,  0, 12;
            3, 10, 12;
            4, 10,  0};
%          nodtag, x, y, tz
constraint={1, 0, 1, 0;
            4, 1, 1, 0};
%-----------------------------------------------------------------------
% Element block
%          {eletag, 'StiffnessBased2DBeamColumn', in, jn, sectag}}
elements = { {1, 'StiffnessBased2DBeamColumn', 1, 2, 1};
             {2, 'StiffnessBased2DBeamColumn', 2, 3, 2};
             {3, 'StiffnessBased2DBeamColumn', 3, 4, 1}};

%-----------------------------------------------------------------------
% Section block
%           sectag, 'General', {mattag, A, Ix, Iy, Iz}
sections = { 1, 'General', {1, 100*12^-2, 0, 0, 600*12^-4};
             2, 'General', {1, 100*12^-2, 0, 0, 900*12^-4}};
%-----------------------------------------------------------------------
% Material block
%             mattag, 'Elastic', E, G, density
materials = { {1, 'Elastic', 29*10^3*12^2, 0, 0} };
%-----------------------------------------------------------------------
% Force block
%           'ElementForce', {elenum, localaxis, distance from start node, m}
staticforces = { 'ElementForce', {2, 2, 5, -12}};
%-----------------------------------------------------------------------
```