

Beam1 - Example1

Contents

1	Beam1 Simply Supported Linear Static Solver	5
	Contents	5
	setup st7model class	5
	setup nfa solver info	5
	setup node restraints	6
	assign input structures to main model structure	6
	run the apis-shell using main function	6
	view frequencies	6
2	Beam1 Fixed-Fixed NFA example	9
	Contents	9
	setup st7model class	9
	setup nfa solver info	9
	setup node restraints	10
	assign input structures to main model structure	10
	run the apis-shell using main function	10
	view frequencies	10
3	Beam1 - update with dynamic info	13
	Contents	13
	Setup structure and get “experimental” data from random spring value	13
	setup optimization	14
4	Beam1 - Assign Boundary Conditions (node restraints) and get node info	23
	Contents	23
	setup st7model class	23
	setup node restraints	23
	assign input structures to main model structure	24
	run the apis-shell using main function	24
	view results	24
5	Beam1 - sensitivity study	27
	Contents	27

setup st7 file info	27
setup nfa info	27
setup node restraints	28
setup spring sensitivity study	28
run the shell	29
view nfa info	30
6 Beam1 - Example1	33
Contents	33
setup st7 file info	33
setup nfa solver info	33
setup node restraints	34
setup discrete springs	34
assign input structures to main model structure	34
run the shell	34
view frequencies	34

Chapter 1

Beam1 Simply Supported Linear Static Solver

jdv 08232016

Contents

- setup st7model class
- setup nfa solver info
- setup node restraints
- assign input structures to main model structure
- run the apis-shell using main function
- view frequencies

setup st7model class

```
sys = st7model();  
sys.pathname = 'C:\Users\John\Documents\MATLAB\repos\st7api\models';  
sys.filename = 'beam1.st7';  
sys.scratchpath = 'C:\Temp';
```

setup nfa solver info

Note that nfa.coords [x,y] is an optional input - nfa will snap or default to all nodes

```

lsa = LSA();
lsa.name = fullfile(sys.pathname,[sys.filename(1:end-4) '.LSA']);
lsa.run = 1;
lsa.inputid = 3; % point load at nodeid=3
lsa.inputcase = 1; % loadcase = 1
lsa.fcase = 1; % freedom case
lsa.force = [0 0 -1e3]; % -1kip load at dof3
lsa.outputid = 1:11; % output at all nodes
lsa.outputcase = ones(size(lsa.outputid));

```

setup node restraints

```

bc = boundaryNode();
bc.nodeid = [1 11]; % boundary nodes
bc.restraint = [1 1 1 0 0 0; 0 1 1 0 0 0]; % [nnodes x 6dof] - simple beam
bc.fcase = ones(size(bc.nodeid)); % [nnodes x 1] - assign to case1

```

assign input structures to main model structure

```

model.sys = sys;
model.lsa = lsa;
model.bc = bc;

```

run the apis-shell using main function

```

fcu = @main;
results = apish(fcu,model);

Loading ST7API.DLL... Done.
    Getting spatial information...      Sorting boundary nodes... Done.Done.
    Running Linear Static Analysis...
Done.

```

view frequencies

```

fprintf('The nodal displacements are: \n');
fprintf('\t\tDZ\n')
fprintf('\t%f\n',results.lsa.resp(:,3));

```

The nodal displacements are:

DZ
0.000000
-0.001363
-0.002434
-0.002888
-0.002881
-0.002535
-0.001973
-0.001316
-0.000687
-0.000207
0.000000

Published with MATLAB® R2014b

Chapter 2

Beam1 Fixed-Fixed NFA example

jdv 08172016

Contents

- setup st7model class
- setup nfa solver info
- setup node restraints
- assign input structures to main model structure
- run the apis-shell using main function
- view frequencies

setup st7model class

```
sys = st7model();  
sys.pathname = 'C:\Users\John\Documents\MATLAB\repos\st7api\models';  
sys.filename = 'beam1.st7';  
sys.scratchpath = 'C:\Temp';
```

setup nfa solver info

Note that nfa.coords [x,y] is an optional input - nfa will snap or default to all nodes

```
nfa = NFA();
nfa.name = fullfile(sys.pathname,[sys.filename(1:end-4) '.NFA']);
nfa.nmodes = 4; % set number of modes to compute
nfa.run = 1;
```

setup node restraints

```
bc = boundaryNode();
bc.nodeid = [1 11]; % fully fixed at nodes 1 and 11
bc.restraint = ones(length(bc.nodeid),6); % [nnodes x 6dof]
bc.fcase = ones(size(bc.nodeid)); % [nnodes x 1]
```

assign input structures to main model structure

```
model.sys = sys;
model.nfa = nfa;
model.bc = bc;
```

run the apis-shell using main function

```
fcfn = @main;
results = apish(fcn,model);

Loading ST7API.DLL... Done.
    Getting spatial information...      Sorting boundary nodes... Done.Done.
    NFA Analysis...
        Populating Mode Shapes... Done.      Running Linear Static Analysis...
Done.
```

view frequencies

```
fprintf('Natural Frequencies [Hz]:\n');
fprintf('\t%f\n',results.nfa.freq);
```

```
Natural Frequencies [Hz]:
    159.787149
    411.388830
    750.091076
    830.361348
```

Published with MATLAB® R2014b

```
function obj = beam1_obj(para,beam,efreq)

    % set spring value
    beam.springs.Kr = [0 para 0];

    % call api to get frequencies due to current para
    opts.keepLoaded = 1; % keeps library loaded
    results = apish(@main,beam);

    % get frequencies
    afreq = results.nfa.freq;

    % form residual for each mode
    for ii = 1:results.nfa.nmodes
        obj(ii) = (afreq(ii)-efreq(ii))/efreq(ii);
    end
    % return sum of squares as objective function value (this isn't
    % necesasry for lsqnonlin)

    obj = sqrt(sum(sum(obj.^2),2));

end
```

Error using beam1_obj (line 4)
Not enough input arguments.

Published with MATLAB® R2014b

Chapter 3

Beam1 - update with dynamic info

Structure: simply supported beam with rotational spring at pinned end. spring value is randomly created within the sensitive range (determined in beam1_sensitivity example)

Experiment: the ‘experimental’ data is the first three modes of the beam

jdv 08182016

Contents

- Setup structure and get “experimental” data from random spring value
- setup optimization

Setup structure and get “experimental” data from random spring value

```
% setup st7 file info
sys = st7model();
sys.pathname = 'C:\Users\John\Documents\MATLAB\repos\st7api\models';
sys.filename = 'beam1.st7';
sys.scratchpath = 'C:\Temp';
% setup nfa info
nfa = NFA();
nfa.name = fullfile(sys.pathname,[sys.filename(1:end-4) '.NFA']);
```

```

nfa.nmodes = 3;
nfa.run = 1;
% setup node restraints
bc = boundaryNode();
bc.nodeid = [1 11];
bc.restraint = zeros(length(bc.nodeid),6); % no restraints
bc.restraint(1,1:3) = 1; % pinned
bc.restraint(11,2:3) = 1; % roller (x kept released)
bc.fcase = ones(size(bc.nodeid));

% spring value:
% generate a random number between 1e7 and 1e11
lb = 1e5; ub = 1e9;
k = (ub-lb)*rand(1,1)+lb;
springs = spring();
springs.nodeid = 1;
springs.Kr = [0 k 0]; %assign spring value
springs.Kfc = 1;

% assign to beam struct and call api
beam.sys = sys;
beam.nfa = nfa;
beam.bc = bc;
beam.springs = springs;
results = apish(@main,beam);

efreq = results.nfa.freq;

Loading ST7API.DLL... Done.
    Getting spatial information...      Sorting boundary nodes... Done.Done.
    NFA Analysis...
        Populating Mode Shapes... Done.

```

setup optimization

```

tic

% use random starting guess
lb = 1e7;
ub = 1e11;
initpara = (ub-lb)*rand(1,1)+lb;

options = PSOSET('SWARM_SIZE', 10 , ...

```

```
'MAX_ITER' , 10 , ...
'TOLFUN'   , 1e-6 , ...
'TOLX'     , 1e-6 , ...
'DISPLAY'  , 'iter');

% create anonymous function that generates the data (residuals) to minimize
% * re-use the beam structure for api calls
obj = @(para)beam1_obj(para,beam,efreq);

[para,fval,exitflag,output] = PSO(obj,initpara,lb,ub,options);

toc

Loading ST7API.DLL... Done.
Getting spatial information...      Sorting boundary nodes... Done.Done.
NFA Analysis...
    Populating Mode Shapes... Done. Loading ST7API.DLL... Done.
Getting spatial information...      Sorting boundary nodes... Done.Done.
NFA Analysis...
    Populating Mode Shapes... Done. Loading ST7API.DLL... Done.
Getting spatial information...      Sorting boundary nodes... Done.Done.
NFA Analysis...
    Populating Mode Shapes... Done. Loading ST7API.DLL... Done.
Getting spatial information...      Sorting boundary nodes... Done.Done.
NFA Analysis...
    Populating Mode Shapes... Done. Loading ST7API.DLL... Done.
Getting spatial information...      Sorting boundary nodes... Done.Done.
NFA Analysis...
    Populating Mode Shapes... Done. Loading ST7API.DLL... Done.
Getting spatial information...      Sorting boundary nodes... Done.Done.
NFA Analysis...
    Populating Mode Shapes... Done. Loading ST7API.DLL... Done.
Getting spatial information...      Sorting boundary nodes... Done.Done.
NFA Analysis...
    Populating Mode Shapes... Done. Loading ST7API.DLL... Done.
Getting spatial information...      Sorting boundary nodes... Done.Done.
NFA Analysis...
    Populating Mode Shapes... Done. Loading ST7API.DLL... Done.
Getting spatial information...      Sorting boundary nodes... Done.Done.
NFA Analysis...
    Populating Mode Shapes... Done. Loading ST7API.DLL... Done.
Getting spatial information...      Sorting boundary nodes... Done.Done.
NFA Analysis...
    Populating Mode Shapes... Done. Warning: repmat(A,M,N) where M or N is a row vector will
future release. Use repmat(A,[M,N]) instead.
```

Nr Iter	Nr Fun Eval	Current best function	Current worst function	Best fun
1	10	0.0802611	0.0830823	0.
Loading ST7API.DLL... Done.				
Getting spatial information...		Sorting boundary nodes... Done.Done.		
NFA Analysis...				
Populating Mode Shapes... Done.		Loading ST7API.DLL... Done.		
Getting spatial information...		Sorting boundary nodes... Done.Done.		
NFA Analysis...				
Populating Mode Shapes... Done.		Loading ST7API.DLL... Done.		
Getting spatial information...		Sorting boundary nodes... Done.Done.		
NFA Analysis...				
Populating Mode Shapes... Done.		Loading ST7API.DLL... Done.		
Getting spatial information...		Sorting boundary nodes... Done.Done.		
NFA Analysis...				
Populating Mode Shapes... Done.		Loading ST7API.DLL... Done.		
Getting spatial information...		Sorting boundary nodes... Done.Done.		
NFA Analysis...				
Populating Mode Shapes... Done.		Loading ST7API.DLL... Done.		
Getting spatial information...		Sorting boundary nodes... Done.Done.		
NFA Analysis...				
Populating Mode Shapes... Done.		Loading ST7API.DLL... Done.		
Getting spatial information...		Sorting boundary nodes... Done.Done.		
NFA Analysis...				
Populating Mode Shapes... Done.		Loading ST7API.DLL... Done.		
Getting spatial information...		Sorting boundary nodes... Done.Done.		
NFA Analysis...				
Populating Mode Shapes... Done.		Loading ST7API.DLL... Done.		
Getting spatial information...		Sorting boundary nodes... Done.Done.		
NFA Analysis...				
Populating Mode Shapes... Done.		Loading ST7API.DLL... Done.		
Getting spatial information...		Sorting boundary nodes... Done.Done.		
NFA Analysis...				
Populating Mode Shapes... Done.		Warning: repmat(A,M,N) where M or N is a row v		
future release. Use repmat(A,[M,N]) instead.				
2	20	0.0802611	0.0829027	0.
Loading ST7API.DLL... Done.				
Getting spatial information...		Sorting boundary nodes... Done.Done.		
NFA Analysis...				
Populating Mode Shapes... Done.		Loading ST7API.DLL... Done.		
Getting spatial information...		Sorting boundary nodes... Done.Done.		
NFA Analysis...				
Populating Mode Shapes... Done.		Loading ST7API.DLL... Done.		
Getting spatial information...		Sorting boundary nodes... Done.Done.		
NFA Analysis...				
Populating Mode Shapes... Done.		Loading ST7API.DLL... Done.		
Getting spatial information...		Sorting boundary nodes... Done.Done.		

[illegible]

```

    Populating Mode Shapes... Done. Loading ST7API.DLL... Done.
    Getting spatial information...      Sorting boundary nodes... Done.Done.
    NFA Analysis...
    Populating Mode Shapes... Done. Loading ST7API.DLL... Done.
    Getting spatial information...      Sorting boundary nodes... Done.Done.
    NFA Analysis...
    Populating Mode Shapes... Done. Warning: repmat(A,M,N) where M or N is a row v
future release. Use repmat(A,[M,N]) instead.
    4          40          0.0047995          0.082538          0.0
Loading ST7API.DLL... Done.
    Getting spatial information...      Sorting boundary nodes... Done.Done.
    NFA Analysis...
    Populating Mode Shapes... Done. Loading ST7API.DLL... Done.
    Getting spatial information...      Sorting boundary nodes... Done.Done.
    NFA Analysis...
    Populating Mode Shapes... Done. Loading ST7API.DLL... Done.
    Getting spatial information...      Sorting boundary nodes... Done.Done.
    NFA Analysis...
    Populating Mode Shapes... Done. Loading ST7API.DLL... Done.
    Getting spatial information...      Sorting boundary nodes... Done.Done.
    NFA Analysis...
    Populating Mode Shapes... Done. Loading ST7API.DLL... Done.
    Getting spatial information...      Sorting boundary nodes... Done.Done.
    NFA Analysis...
    Populating Mode Shapes... Done. Loading ST7API.DLL... Done.
    Getting spatial information...      Sorting boundary nodes... Done.Done.
    NFA Analysis...
    Populating Mode Shapes... Done. Loading ST7API.DLL... Done.
    Getting spatial information...      Sorting boundary nodes... Done.Done.
    NFA Analysis...
    Populating Mode Shapes... Done. Loading ST7API.DLL... Done.
    Getting spatial information...      Sorting boundary nodes... Done.Done.
    NFA Analysis...
    Populating Mode Shapes... Done. Loading ST7API.DLL... Done.
    Getting spatial information...      Sorting boundary nodes... Done.Done.
    NFA Analysis...
    Populating Mode Shapes... Done. Loading ST7API.DLL... Done.
    Getting spatial information...      Sorting boundary nodes... Done.Done.
    NFA Analysis...
    Populating Mode Shapes... Done. Warning: repmat(A,M,N) where M or N is a row v
future release. Use repmat(A,[M,N]) instead.
    5          50          0.00288653          0.0819713          0.0
Loading ST7API.DLL... Done.
    Getting spatial information...      Sorting boundary nodes... Done.Done.
    NFA Analysis...
    Populating Mode Shapes... Done. Loading ST7API.DLL... Done.

```


[illegible]

Populating Mode Shapes... Done. Warning: repmat(A,M,N) where M or N is a row vector will
future release. Use repmat(A,[M,N]) instead.

8	80	0.00119783	0.125437	0.00119783
---	----	------------	----------	------------

Loading ST7API.DLL... Done.

Getting spatial information... Sorting boundary nodes... Done.Done.

NFA Analysis...

Populating Mode Shapes... Done. Loading ST7API.DLL... Done.

Getting spatial information... Sorting boundary nodes... Done.Done.

NFA Analysis...

Populating Mode Shapes... Done. Loading ST7API.DLL... Done.

Getting spatial information... Sorting boundary nodes... Done.Done.

NFA Analysis...

Populating Mode Shapes... Done. Loading ST7API.DLL... Done.

Getting spatial information... Sorting boundary nodes... Done.Done.

NFA Analysis...

Populating Mode Shapes... Done. Loading ST7API.DLL... Done.

Getting spatial information... Sorting boundary nodes... Done.Done.

NFA Analysis...

Populating Mode Shapes... Done. Loading ST7API.DLL... Done.

Getting spatial information... Sorting boundary nodes... Done.Done.

NFA Analysis...

Populating Mode Shapes... Done. Loading ST7API.DLL... Done.

Getting spatial information... Sorting boundary nodes... Done.Done.

NFA Analysis...

Populating Mode Shapes... Done. Loading ST7API.DLL... Done.

Getting spatial information... Sorting boundary nodes... Done.Done.

NFA Analysis...

Populating Mode Shapes... Done. Loading ST7API.DLL... Done.

Getting spatial information... Sorting boundary nodes... Done.Done.

NFA Analysis...

Populating Mode Shapes... Done. Loading ST7API.DLL... Done.

Getting spatial information... Sorting boundary nodes... Done.Done.

NFA Analysis...

Populating Mode Shapes... Done. Warning: repmat(A,M,N) where M or N is a row vector will
future release. Use repmat(A,[M,N]) instead.

9	90	0.007362	0.195115	0.00119783
---	----	----------	----------	------------

Loading ST7API.DLL... Done.

Getting spatial information... Sorting boundary nodes... Done.Done.

NFA Analysis...

Populating Mode Shapes... Done. Loading ST7API.DLL... Done.

Getting spatial information... Sorting boundary nodes... Done.Done.

NFA Analysis...

Populating Mode Shapes... Done. Loading ST7API.DLL... Done.

Getting spatial information... Sorting boundary nodes... Done.Done.

NFA Analysis...

Populating Mode Shapes... Done. Loading ST7API.DLL... Done.

```

Getting spatial information...      Sorting boundary nodes... Done.Done.
NFA Analysis...
    Populating Mode Shapes... Done. Loading ST7API.DLL... Done.
Getting spatial information...      Sorting boundary nodes... Done.Done.
NFA Analysis...
    Populating Mode Shapes... Done. Loading ST7API.DLL... Done.
Getting spatial information...      Sorting boundary nodes... Done.Done.
NFA Analysis...
    Populating Mode Shapes... Done. Loading ST7API.DLL... Done.
Getting spatial information...      Sorting boundary nodes... Done.Done.
NFA Analysis...
    Populating Mode Shapes... Done. Loading ST7API.DLL... Done.
Getting spatial information...      Sorting boundary nodes... Done.Done.
NFA Analysis...
    Populating Mode Shapes... Done. Loading ST7API.DLL... Done.
Getting spatial information...      Sorting boundary nodes... Done.Done.
NFA Analysis...
    Populating Mode Shapes... Done. Loading ST7API.DLL... Done.
Getting spatial information...      Sorting boundary nodes... Done.Done.
NFA Analysis...
    Populating Mode Shapes... Done. Warning: repmat(A,M,N) where M or N is a row v
future release. Use repmat(A,[M,N]) instead.
    10      100      0.000604798      0.0553223      0.00
Maximum number of function evaluations or iterations reached.
Elapsed time is 162.550885 seconds.

Published with MATLAB® R2014b

```

Chapter 4

Beam1 - Assign Boundary Conditions (node restraints) and get node info

jdv 08172016

Contents

- setup st7model class
- setup node restraints
- assign input structures to main model structure
- run the apis-shell using main function
- view results

setup st7model class

```
sys = st7model();  
sys.pathname = 'C:\Users\John\Documents\MATLAB\repos\st7api\models';  
sys.filename = 'beam1.st7';  
sys.scratchpath = 'C:\Temp';
```

setup node restraints

```
bc = boundaryNode();
```


VIEW RESULTS

25

Published with MATLAB® R2014b

Chapter 5

Beam1 - sensitivity study

- multiple runs

jdv 08142016

Contents

- setup st7 file info
- setup nfa info
- setup node restraints
- setup spring sensitivity study
- run the shell
- view nfa info

setup st7 file info

```
sys = st7model();  
sys.pathname = 'C:\Users\John\Documents\MATLAB\repos\st7api\models';  
sys.filename = 'beam1.st7';  
sys.scratchpath = 'C:\Temp';
```

setup nfa info

```
nfa = NFA();  
nfa.name = fullfile(sys.pathname,[sys.filename(1:end-4) '.NFA']);  
nfa.nmodes = 4; % set number of modes to compute
```

```
nfa.run = 1;
```

setup node restraints

```
bc = boundaryNode();
bc.nodeid = [1 11];
bc.restraint = zeros(length(bc.nodeid),6); % no restraints
bc.restraint(1,1:3) = 1; % pinned
bc.restraint(11,2:3) = 1; % roller (x kept released)
bc.fcase = ones(size(bc.nodeid));
```

setup spring sensitivity study

Create rotational springs about the y-axis for boundary nodes

pay attention, this section is tricky

```
springs = spring();
springs.nodeid = 1;
% create unit spring force at desired dof
Kr = [0 1 0];

% create spring range from 10^-2 to 10*12 with 10 increments. start at 5 for
% stability
% note its a row vector
steps = 20;
springrange = logspace(7,11,steps)';

% build model array
for ii = 1:steps
    % the class st7model is not a handle subclass. it is just a value
    % class, like a hard-coded structure. because of this we can create
    % copies of it
    beam(ii).sys = sys;

    % the same goes with the boundaryNodes class
    beam(ii).bc = bc;

    % create new instance of nfa class
    % * this is because nfa subclasses the handle class. handles are
    % persistent. if you create a copy and change it, the original changes
    % too. we we need to create a new instance.
    beam(ii).nfa = NFA();
```

```

beam(ii).nfa.name = strcat(fullfile(sys.pathname,sys.filename(1:end-4)), ...
    '_step',num2str(ii),'.NFA');
beam(ii).nfa.nmodes = 4;
beam(ii).nfa.run = 1;

% springs
% this class is also *not* a subclass of handles. we can use copies.
springs.Kr = Kr*springrange(ii);
springs.Kfc = ones(size(springs.Kr,1),1); % default to freedom case 1
beam(ii).springs = springs;
end

```

run the shell

```
tic
```

```
results = apish(@main,beam);
```

```
toc
```

```
Loading ST7API.DLL... Done.
```

```
Getting spatial information...      Sorting boundary nodes... Done.Done.
```

```
NFA Analysis...
```

```
Populating Mode Shapes... Done.
```

```
Getting spatial information...
```

```
Sorting boundary
```

```
NFA Analysis...
```

```
Populating Mode Shapes... Done.
```

```
Getting spatial information...
```

```
Sorting boundary
```

```
NFA Analysis...
```

```
Populating Mode Shapes... Done.
```

```
Getting spatial information...
```

```
Sorting boundary
```

```
NFA Analysis...
```

```
Populating Mode Shapes... Done.
```

```
Getting spatial information...
```

```
Sorting boundary
```

```
NFA Analysis...
```

```
Populating Mode Shapes... Done.
```

```
Getting spatial information...
```

```
Sorting boundary
```

```
NFA Analysis...
```

```
Populating Mode Shapes... Done.
```

```
Getting spatial information...
```

```
Sorting boundary
```

```
NFA Analysis...
```

```
Populating Mode Shapes... Done.
```

```
Getting spatial information...
```

```
Sorting boundary
```

```
NFA Analysis...
```

```
Populating Mode Shapes... Done.
```

```
Getting spatial information...
```

```
Sorting boundary
```

```
NFA Analysis...
```

```
Populating Mode Shapes... Done.
```

```
Getting spatial information...
```

```
Sorting boundary
```

```
NFA Analysis...
```

```
Populating Mode Shapes... Done.
```

```
Getting spatial information...
```

```
Sorting boundary
```

```
NFA Analysis...
```

```
Populating Mode Shapes... Done.
```

```
Getting spatial information...
```

```
Sorting boundary
```

```

NFA Analysis...
  Populating Mode Shapes... Done.      Getting spatial information... Sorting
NFA Analysis...
  Populating Mode Shapes... Done.      Getting spatial information... Sorting
NFA Analysis...
  Populating Mode Shapes... Done.      Getting spatial information... Sorting
NFA Analysis...
  Populating Mode Shapes... Done.      Getting spatial information... Sorting
NFA Analysis...
  Populating Mode Shapes... Done.      Getting spatial information... Sorting
NFA Analysis...
  Populating Mode Shapes... Done.      Getting spatial information... Sorting
NFA Analysis...
  Populating Mode Shapes... Done.      Getting spatial information... Sorting
NFA Analysis...
  Populating Mode Shapes... Done.      Getting spatial information... Sorting
NFA Analysis...
  Populating Mode Shapes... Done.      Getting spatial information... Sorting
NFA Analysis...
  Populating Mode Shapes... Done.      Getting spatial information... Sorting
NFA Analysis...
  Populating Mode Shapes... Done.      Getting spatial information... Sorting
  Populating Mode Shapes... Done. Elapsed time is 16.013345 seconds.

```

view nfa info

```
plotSpringsVsFreq(results)
```

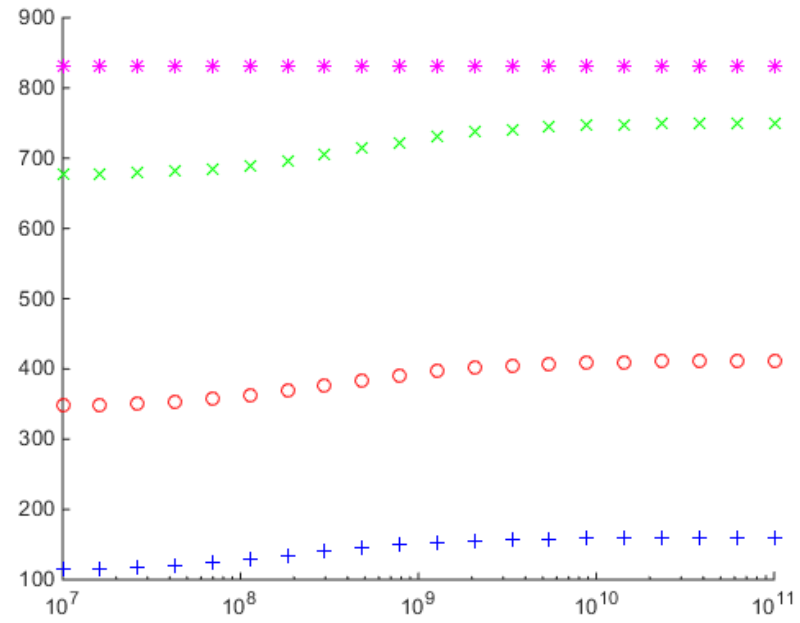
```

% %% plot displaced shapes
%
% dof = results(1).dof;
%
% fh = figure('PaperPositionMode','auto');
% ah = axes();
%
% %% nfa - mode shape vector
% nfa = results(1).nfa;
%
% mode = 5;
% scale = 1;
% z = nfa.U(:,3,mode)*scale;
%
%
% %% plot undeformed shape
% plot(dof.coords(:,1),dof.coords(:,3),...
%      'Marker','.',...

```

```

%      'MarkerEdgeColor','k',...
%      'MarkerFaceColor','k');
%
% hold(ah,'all')
%
% % plot mode
% plot(dof.coords(:,1),z,...
%      'color','b',...
%      'Marker','o',...
%      'MarkerEdgeColor','b',...
%      'MarkerFaceColor','b');
%
% % plot boundaries
% scatter(dof.bcoords(:,1),dof.bcoords(:,2),...
%      'MarkerEdgeColor','m',...
%      'MarkerFaceColor','m');
%
% hold(ah,'off')
%
% xlabel(ah,'Beam Length [ft]');
% ylabel(ah,'Modal Amplitude');
%
% ylim(ah,[-1.5 1.5]);
%
% grid(ah,'on');
% grid(ah,'minor');
%
%
%
%
```



Published with MATLAB® R2014b

Chapter 6

Beam1 - Example1

jdv 08122016

Contents

- setup st7 file info
- setup nfa solver info
- setup node restraints
- setup discrete springs
- assign input structures to main model structure
- run the shell
- view frequencies

setup st7 file info

```
sys = st7model();  
sys.pathname = 'C:\Users\John\Documents\MATLAB\repos\st7api\models';  
sys.filename = 'beam1.st7';  
sys.scratchpath = 'C:\Temp';
```

setup nfa solver info

```
nfa = NFA();  
nfa.name = fullfile(sys.pathname,[sys.filename(1:end-4) '.NFA']);  
nfa.nmodes = 4; % set number of modes to compute  
nfa.run = 1;
```

setup node restraints

```
bc = boundaryNode();
bc.nodeid = [1 11];
bc.restraint = zeros(length(bc.nodeid),6); % no restraints
bc.fcase = ones(size(bc.nodeid));
```

setup discrete springs

Add discrete longitudinal (x) and vertical (z) translational springs. No rotational springs added

```
springs = spring();
springs.Kt = [1e5 0 1e5; 1e5 0 1e5];
springs.Kr = [0 1 0; 0 1 0];
springs.Kfc = ones(size(springs.Kt,1),1);
springs.nodeid = [1 11];
```

assign input structures to main model structure

```
model.sys = sys;
model.nfa = nfa;
model.bc = bc;
model.springs = springs;
```

run the shell

```
fcu = @main;
results = apish(fcu,model);

Loading ST7API.DLL... Done.
    Getting spatial information...      Sorting boundary nodes... Done.Done.
    NFA Analysis...
    Populating Mode Shapes... Done.      Running Linear Static Analysis...
Done.
```

view frequencies

```
fprintf('Natural Frequencies [Hz]:\n');
fprintf('%f\n',results.nfa.freq);
```

Natural Frequencies [Hz]:

51.792673

68.922213

112.907829

215.759094

Published with MATLAB® R2014b