OpenSees

Nonlinear Analysis With Simple Examples

Frank McKenna
UC Berkeley

OpenSees Days 2013

NEES NEEScomm

Outline of Presentation

- Why Nonlinear Analysis
- OpenSees Analysis Options in More Depth
- Transient Integrator

Why Nonlinear Analysis

•**Geometric Nonlinearities** - occur in model when applied load causes large displacement and/or rotation, large strain, or a combo of both

•**Material nonlinearities** - nonlinearities occur when material stress-strain relationship depends on load history (plasticity problems), load duration (creep problems), temperature (thermoplasticity), or combo of all.

•**Contact nonlinearities** - occur when structure boundary conditions change because of applied load.

Nonlinear Analysis is Harder

- It requires **much** more thought when setting up the model
- It requires more thought when setting up the analysis
- It takes more computational time.
- It does not always converge.
- It does not always converge to the correct solution.

BUT if you are using a Finite element code the Problem probably Requires a Nonlinear Analysis

CHECK YOUR MODEL

CHECK YOUR MODEL

CHECK YOUR MODEL

CHECK YOUR MODEL

CHECK YOUR MODEL

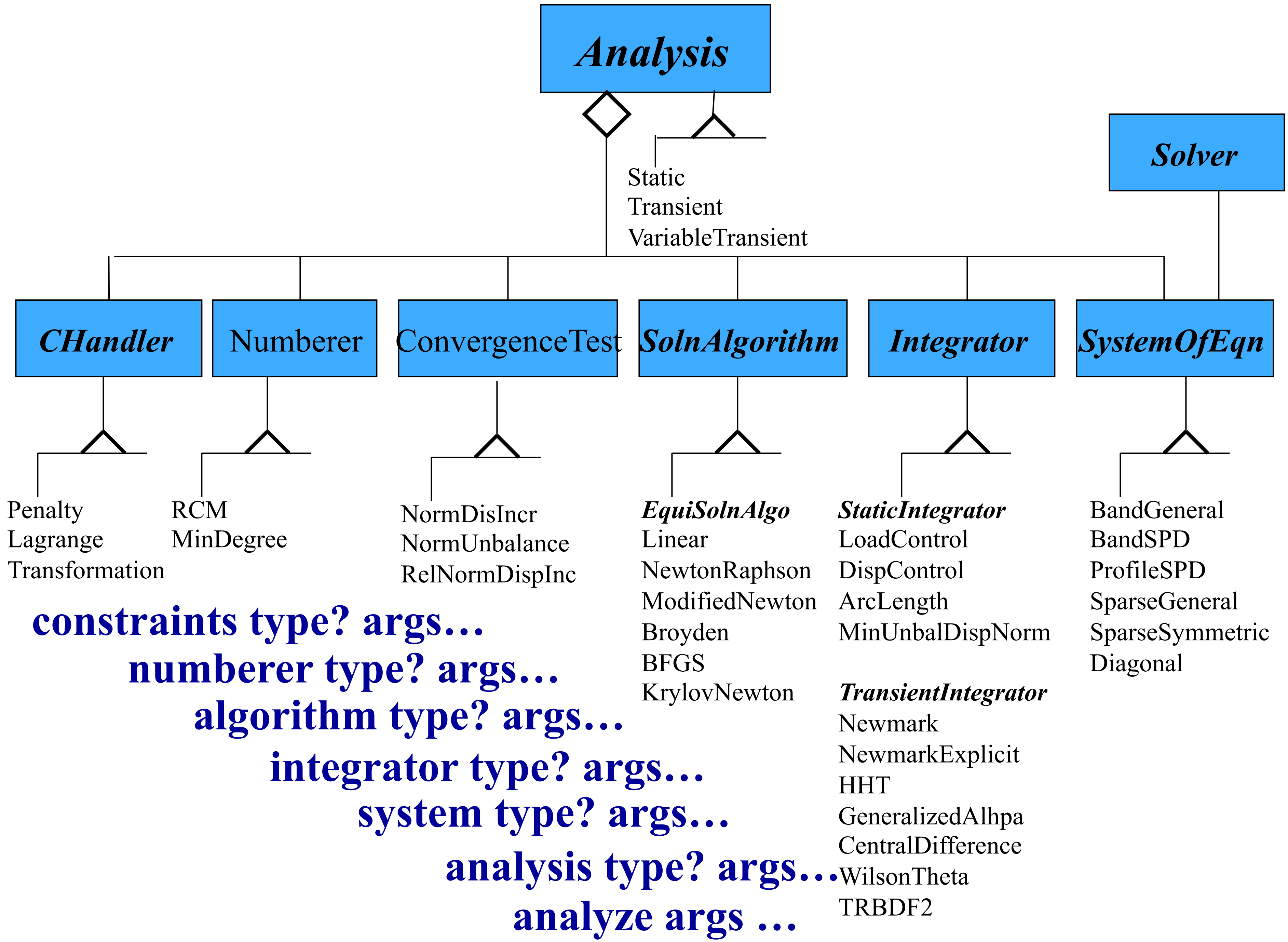
CHECK YOUR MODEL

CHECK YOUR MODEL

CHECK YOUR MODEL

CHECK YOUR MODEL

99% Probability: if Fails in First Step THERE IS A PROBLEM IN YOUR MODEL


Analysis
Solver
Static
Transient
VariableTransient
CHandler
Numberer
ConvergenceTest
SolnAlgorithm
Integrator
SystemOfEqn
Penalty
Lagrange
Transformation
RCM
MinDegree
NormDisIncr
NormUnbalance
RelNormDispInc
EquiSolnAlgo
Linear
NewtonRaphson
ModifiedNewton
Broyden
BFGS
KrylovNewton
StaticIntegrator
LoadControl
DispControl
ArcLength
MinUnbalDispNorm
TransientIntegrator
Newmark
NewmarkExplicit
HHT
GeneralizedAlhpa
CentralDifference
WilsonTheta
TRBDF2
BandGeneral
BandSPD
ProfileSPD
SparseGeneral
SparseSymmetric
Diagonal
constraints type? args…
numberer type? args…
algorithm type? args…
integrator type? args…
system type? args…
analysis type? args…
analyze args …

test command:

- to specify when convergence has been achieved

all look at system: **KU = R**

- Norm Unbalance

$\sqrt{\mathbf{R\hat{}R}} \; \mathbf{< tol}$

test NormUnbalance tol? numIter? <flag?>

- Norm Displacement Increment

$\sqrt{\mathbf{U\hat{}U}} \; \mathbf{< tol}$

test NormDispIncr tol? numIter? <flag?>

- Norm Energy Increment

$\mathbf{\tfrac{1}{2}\,(U\hat{}R) < tol}$

test NormEnergyIncr tol? numIter? <flag?>

- Relative Tests

test RelativeNormUnbalance tol? numIter? <flag?>

test RelativeNormDispIncr tol? numIter? <flag?>

test RelativeNormEnergyIncr tol? numIter? <flag?>

numberer command:

- to specify how the degrees of freedom are numbered

- Plain Numberer

  nodes are assigned dof arbitrarily

  numberer Plain

- RCM Numberer

  nodes are assigned dof using the Reverse Cuthill-McKee algorithm

  numberer RCM

- AMD Numberer

  nodes are assigned dof using the Approx. MinDegree algorithm

  numberer AMD

- numbering has an impact on performance of banded and profile solvers. The sparse solvers all use their own optimal numbering schemes.

algorithm command:

- to specify the steps taken to solve the nonlinear equation

•Linear Algorithm

algorithm Linear

```
theIntegrator->formUnbalance();
theIntegrator->formTangent();
theSOE->solve()
theIntegrator->update(theSOE->getX());
```

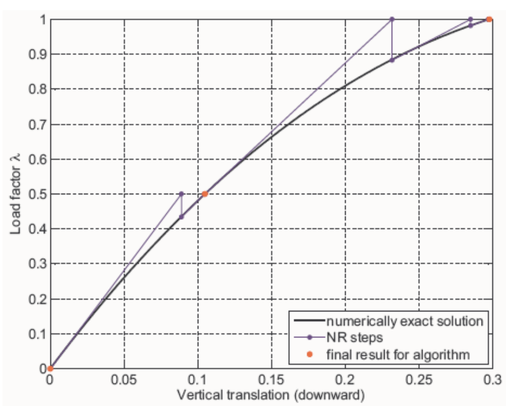
•Newton-Raphson Algorithm

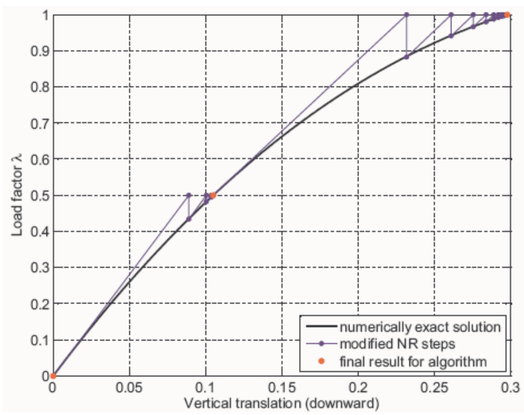
algorithm Newton

```
theIntegrator->formUnbalance();
do {
  theIntegrator->formTangent();
  theSOE->solve()
  theIntegrator->update(theSOE->getX());
  theIntegrator->formUnbalance();
} while (theTest->test() == fail)
```

•Modified Newton Algorithm

algorithm ModifiedNewton <-initial>





•Accelerated Modified Newton Algorithm

algorithm KrylovNewton <-initial>

constraints command:

- to specify how the constraints are enforced

$$\mathbf{U}_c = \mathbf{C}_{rc}\,\mathbf{U}_r$$

$$\mathbf{C}\,\mathbf{U} = \mathbf{0}$$

$$[\mathbf{C}_r \quad \mathbf{C}_c]\text{^}[\mathbf{U}_r \quad \mathbf{U}_c] = 0$$

$$\mathbf{T}\,\mathbf{U}_r = [\mathbf{U}_r \quad \mathbf{U}_c]\text{^}$$

- Transformation Handler

$$K^*\,U_r = R^* \qquad K^*=T\text{^}KT \qquad R^*=T\text{^}R$$

constraints Transformation

in OpenSees currently don't allow retained node in one constraint to be a constrained node in another constraint

- Lagrange Handler

$$\begin{bmatrix} \mathbf{K} & \mathbf{C}\text{^} \\ \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{R} \\ \mathbf{Q} \end{bmatrix}$$

constraints Lagrange

- Penalty Handler

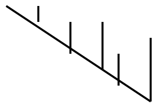
$$[\mathbf{K} + \mathbf{C}\text{^}\alpha\mathbf{C}]\ \mathbf{U} = [\mathbf{R} + \mathbf{C}\text{^}\alpha\mathbf{Q}]$$

constraints Penalty α_{sp}*?* α_{mp}*?*

system command:

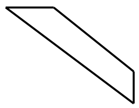
- to specify how matrix equation KU = R is stored and solved

- Profile Symmetric Positive Definite (SPD)



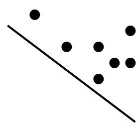
system ProfileSPD

- Banded Symmetric Positive Definite



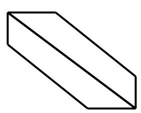
system BandSPD

- Sparse Symmetric Positive Definite



system SparseSPD

- Banded General



system BandGeneral

- Sparse Symmetric

system SparseGeneral

If you have a large system
Use one of these

system Umfpack

integrator command:

-determines the predictive step for time t+δt
-specifies the tangent matrix and residual vector at any iteration
-determines the corrective step based on ΔU

1. Static Integrators for Use in Static Analysis

Nonlinear equation of the form:

$$\mathbf{R(U, \lambda) = \lambda P^* - FR(U)}$$

2. Transient Integrators for Use in Transient Analysis

Nonlinear equation of the form:

$$\mathbf{R(U, \dot{U}, \ddot{U}) = P(t) - F_I(\ddot{U}) - F_R(U, \dot{U})}$$

Static Integrators

$$R(U, \lambda) = \lambda P^* - FR(U)$$

at each step solving for λ

- Load Control

$$\lambda_n = \lambda_{n-1} + \Delta\lambda$$

integrator LoadControl Δλ

*does not require a reference load, i.e. loads in load patterns with Linear series and all other loads constant.

- Displacement Control

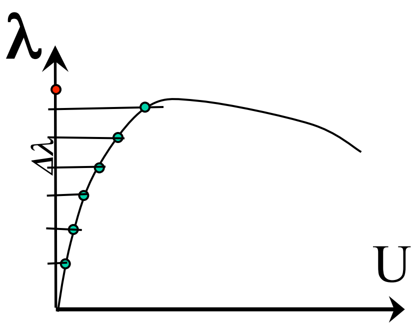
$$Uj_n = Uj_{n-1} + \Delta Uj$$

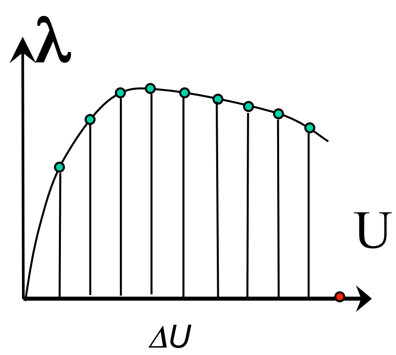
integrator DisplacementControl node dof ΔU

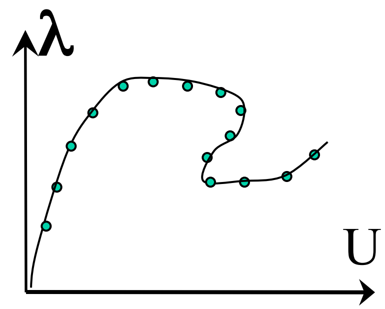
- Arc Length

$$\Delta U_n\text{^}\Delta U_n + \alpha^2 \Delta\lambda_n = \Delta s^2$$

integrator ArcLength α Δs







Transient Integrators

- Explicit:

  $D_{n+1} = f(P_n, D_n, V_n, A_n, D_{n-1}, V_{n-1}, A_{n-1}, \ldots)$

  integrator CentralDifference

  integrator NewmarkExplicit $gamma

  integrator HHTExplicit $alpha

  ….

  1. all Need Linear Algorithm

  2. in absence of damping all require Positive Definite mass matrix.

- Implicit

  $D_{n+1}= f(V_{n+1},A_{n+1}, D_n, V_n, A_n , D_{n-1}, V_{n-1}, A_{n-1}, \ldots)$

  integrator Newmark $gamma $beta

  integrator HHT $alpha

  integrator TRBDF2

  …

Stability & Linear Systems

- Stability (bounded solution) and Accuracy are the most talked about properties of time integration schemes.
- For most integration schemes, the stability and accuracy provisions you read about are provided FOR LINEAR DYNAMICAL SYSTEMS.
- Conditionally Stable: numerical procedure leads to a BOUNDED solution if time step is smaller than some stability limit. Conditional stability requires time step to be inversely proportional to highest frequency.
- Unconditionally Stable: solution is bounded regardless of the time step.

Stability Limits Common Integrators

Central Difference is conditionally stable if:

$$\frac{\Delta t}{T_n} < \frac{1}{\pi} \quad (.318)$$

Newmark is unconditionally stable if:

$$\frac{\Delta t}{T_n} \leq \frac{1}{\pi\sqrt{2}} \frac{1}{\sqrt{\gamma - 2\beta}}$$

Average Acceleration (Trapezoidal) $(\gamma = \frac{1}{2}, \beta = \frac{1}{4})$ ✓

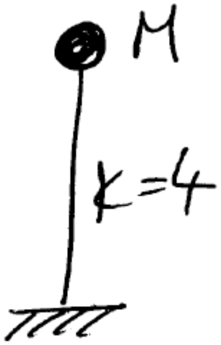
Linear Acceleration $(\gamma = \frac{1}{2}, \beta = \frac{1}{6})$ ✗

But Conditionally stable if: $\frac{\Delta t}{T_n} < 0.55$

Example

(see "Dynamics of Structures" A.K. Chopra, section 5.5)

Free Vibration (ex1.tcl)
(ex2.tcl)



$u(0) = 1$

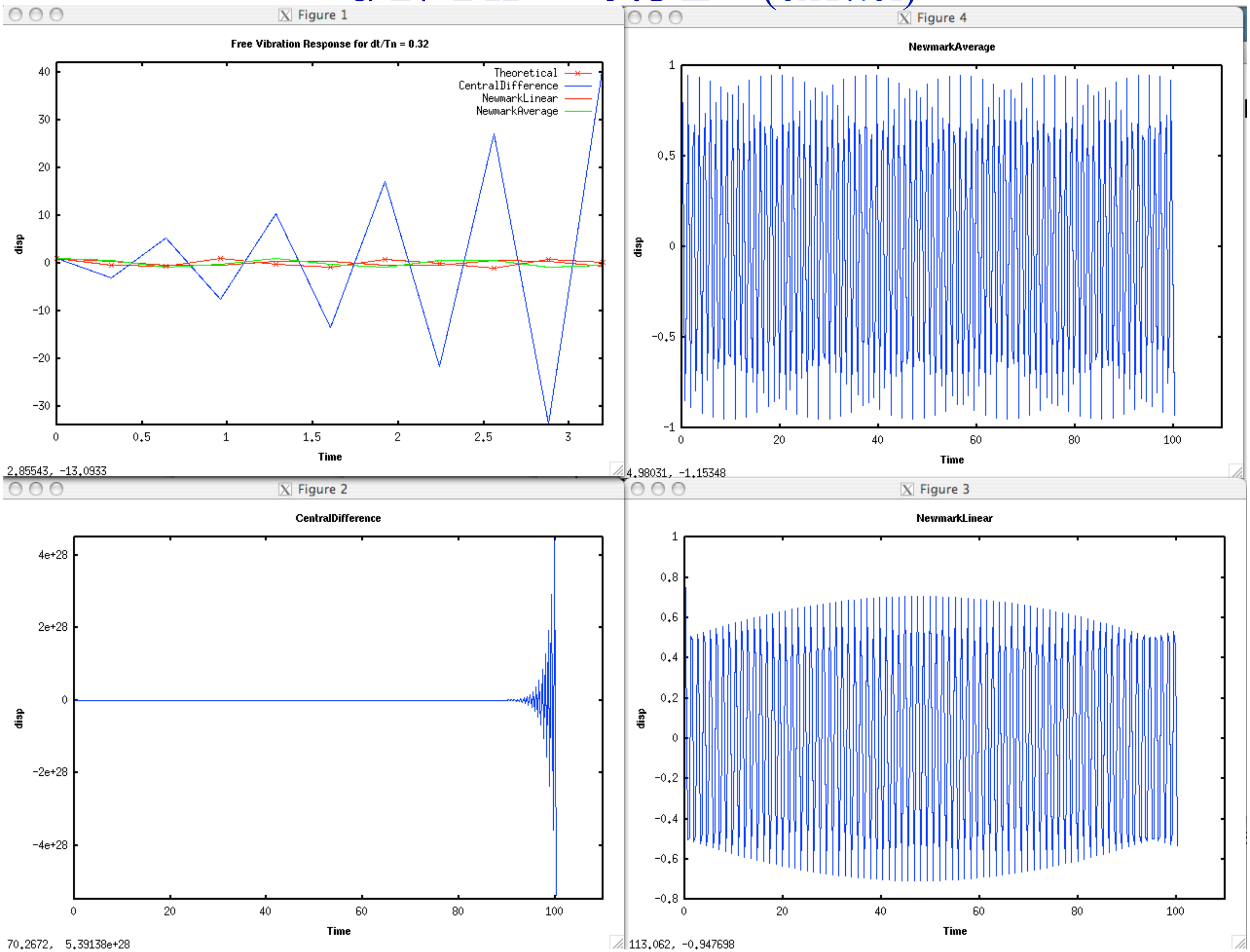
$u'(0) = 0$

$p(t) = \phi$

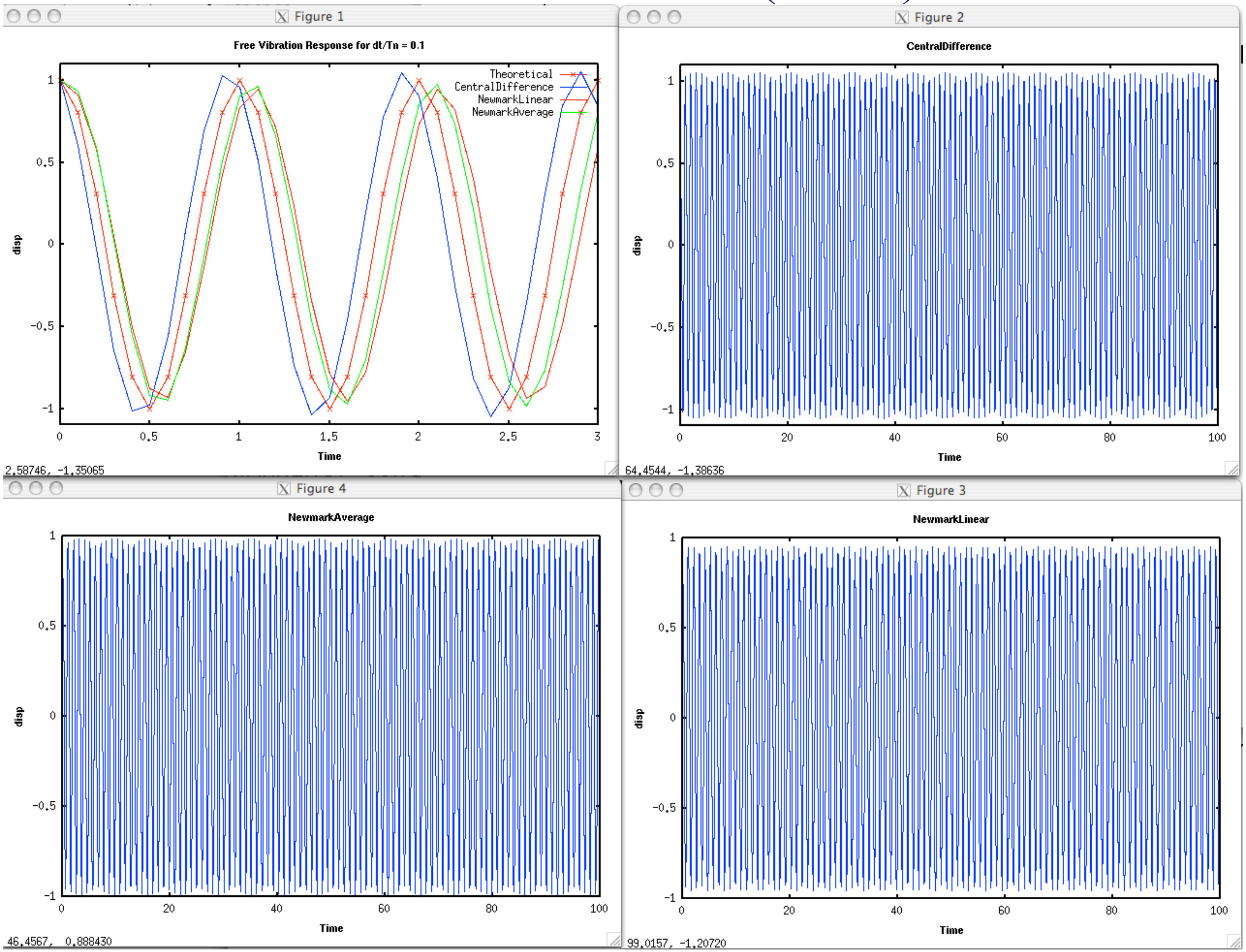
M is chosen to give desired period.

exact soln: $u(t) = u(0)\cos\omega_n t$.

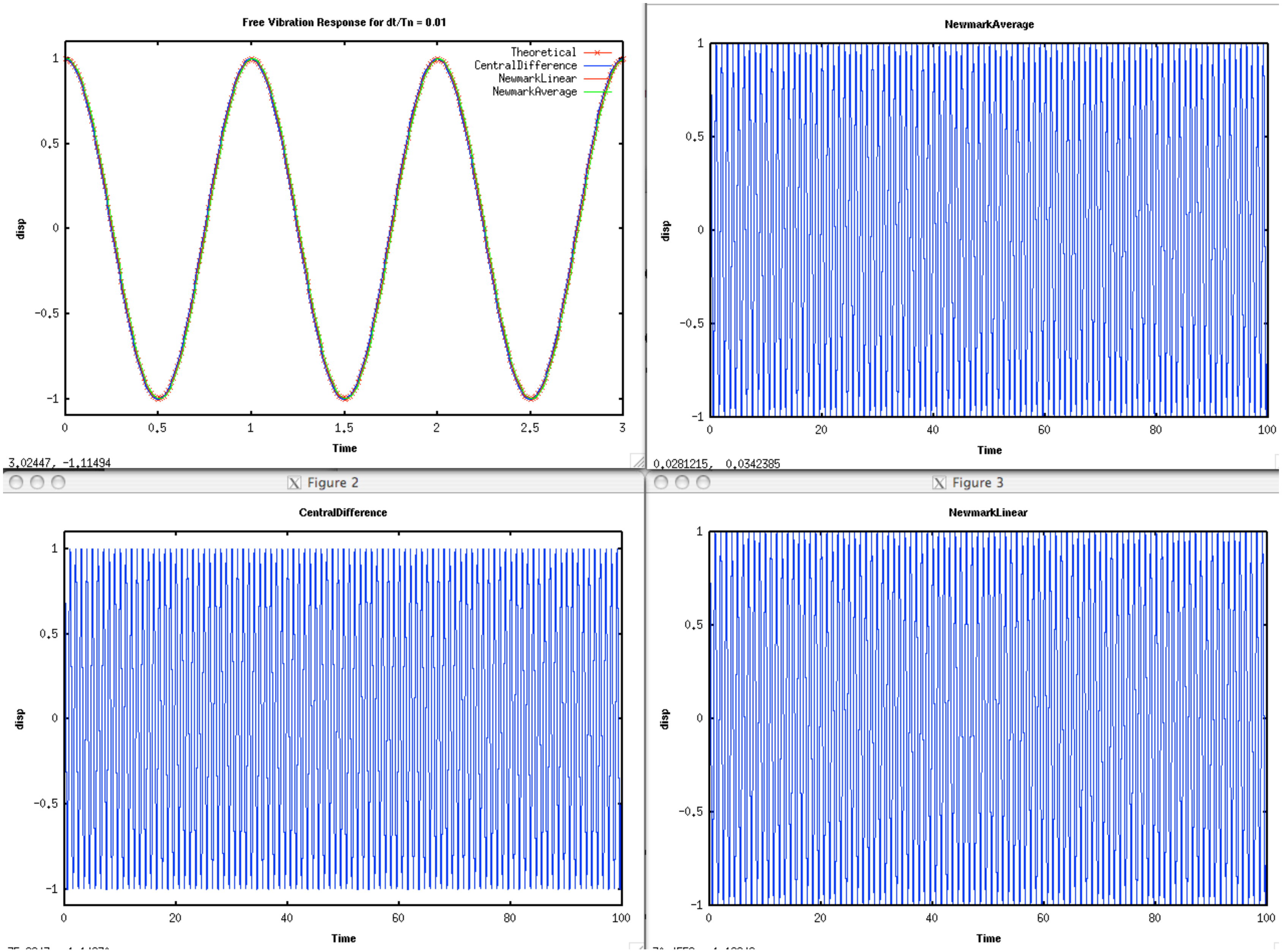
dT/Tn = 0.32 (ex1.tcl)



dT/Tn = 0.1 (ex1.tcl)



dT/Tn = 0.01 (ex1.tcl)



THINGS TO THINK ABOUT

- Computer models of large real structures contain a large number of periods. Some of these periods are smaller than the typical time step used (that from say the earthquake record). It is typically advised to select an algorithm that is unconditionally stable.
- There are situations when you might want to use a conditionally stable algorithm, e.g. convergence problems, accuracy, model size. In these cases you need to select the appropriate time step to ensure that higher frequencies do not cause instability or use other form of damping, e.g. Rayleigh damping.

Stability & Nonlinear Systems

- For nonlinear systems stability is the most important concern.

- Algorithms that are stable for linear dynamical systems ARE NOT NECESSARY STABLE in nonlinear case.

- A sufficient condition in non-linear systems for stability is the conservation of total energy within a step, expressed:

$$U_{n+1} - U_n + K_{n+1} - K_n <= W_{ext}$$

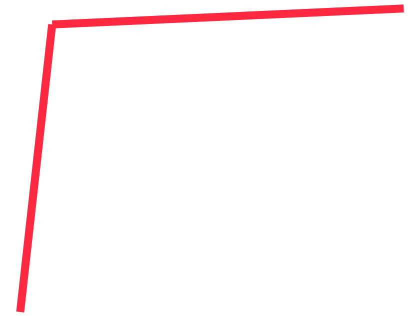
where U = strain energy and K = kinetic energy

- There are 3 groups of algorithms which ATTEMPT to satisfy this criterion:
    1. Numerical Dissipation
    2. Enforced Conservation of Energy
    3. Algorithmic Conservation of Energy

} **Neither Types Are Available in OpenSees**

Observation

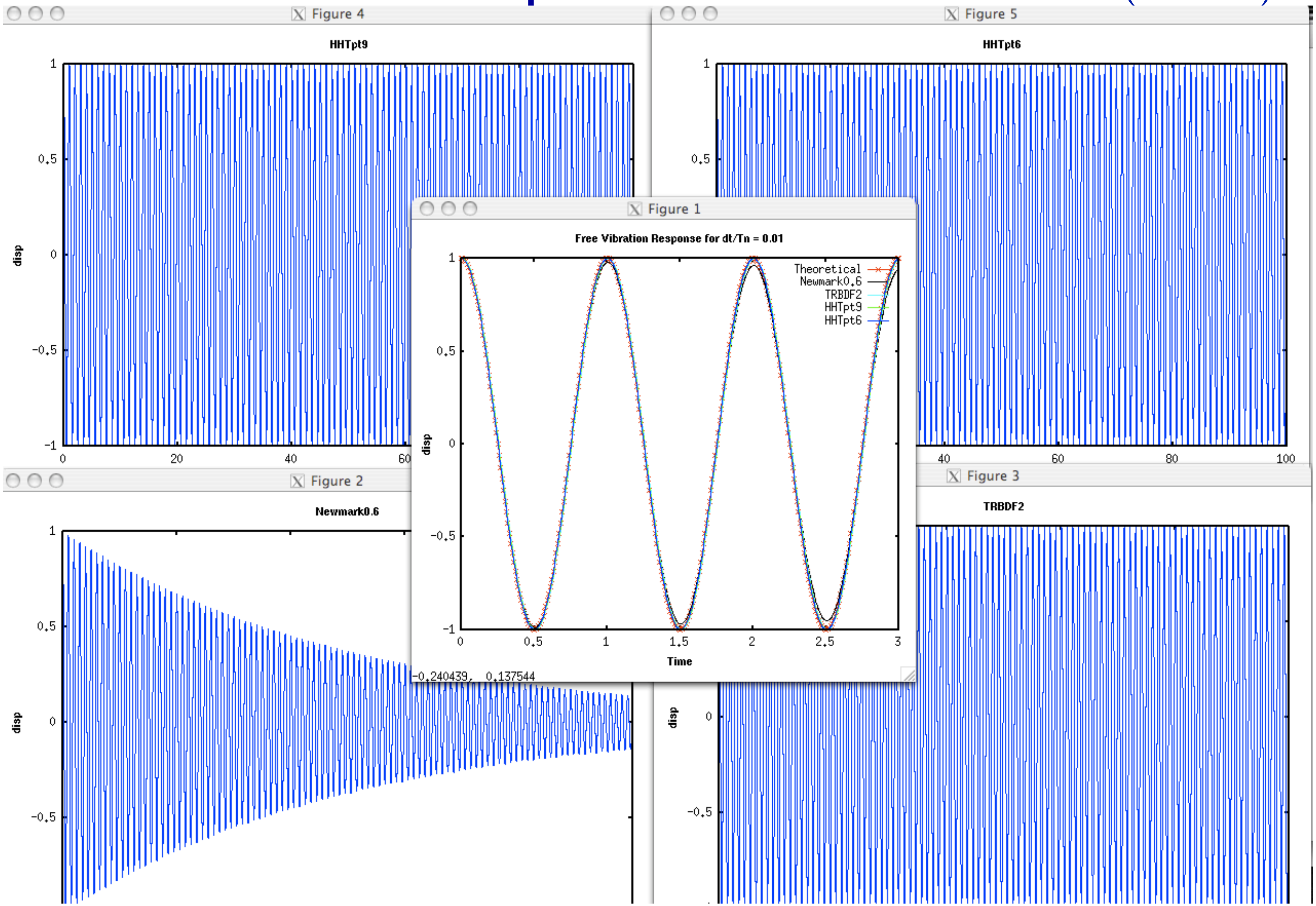
- Cutting dT does not always work as a way to achieve an accurate solution for non-smooth nonlinear problems (discontinuities cause problems):
  - if implicit: may not converge (flip-flop in Newton Raphson,…)
  - if explicit: error introduced can be significant.
- But if all else fails, and you can stomach the wait of the extra computation time required, it is an option to try.



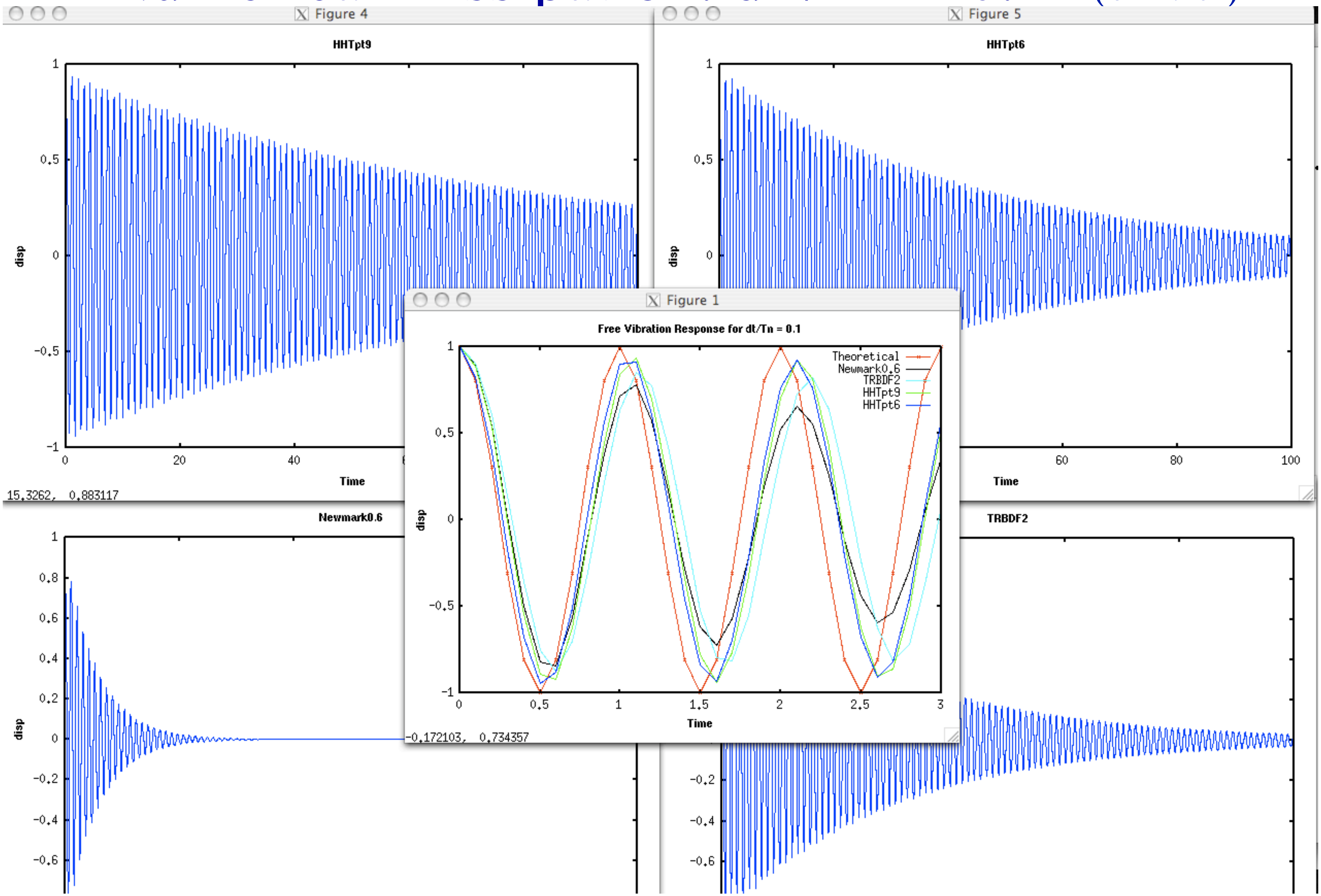
Dissipation Algorithms

- They were developed for large linear systems where typically only the low modes of response are of interest and the engineer wants to remove the high frequency noise (sometimes you don't want to do this!)
- These controlled dissipation of high frequency modes is used in an ATTEMPT to conserve energy.
- For nonlinear systems they do not guarantee the dissipation of enough energy to always satisfy the conservation of energy.
- EXAMPLES: Newmark ($\gamma > 0.5$), HHT, TRBDF2

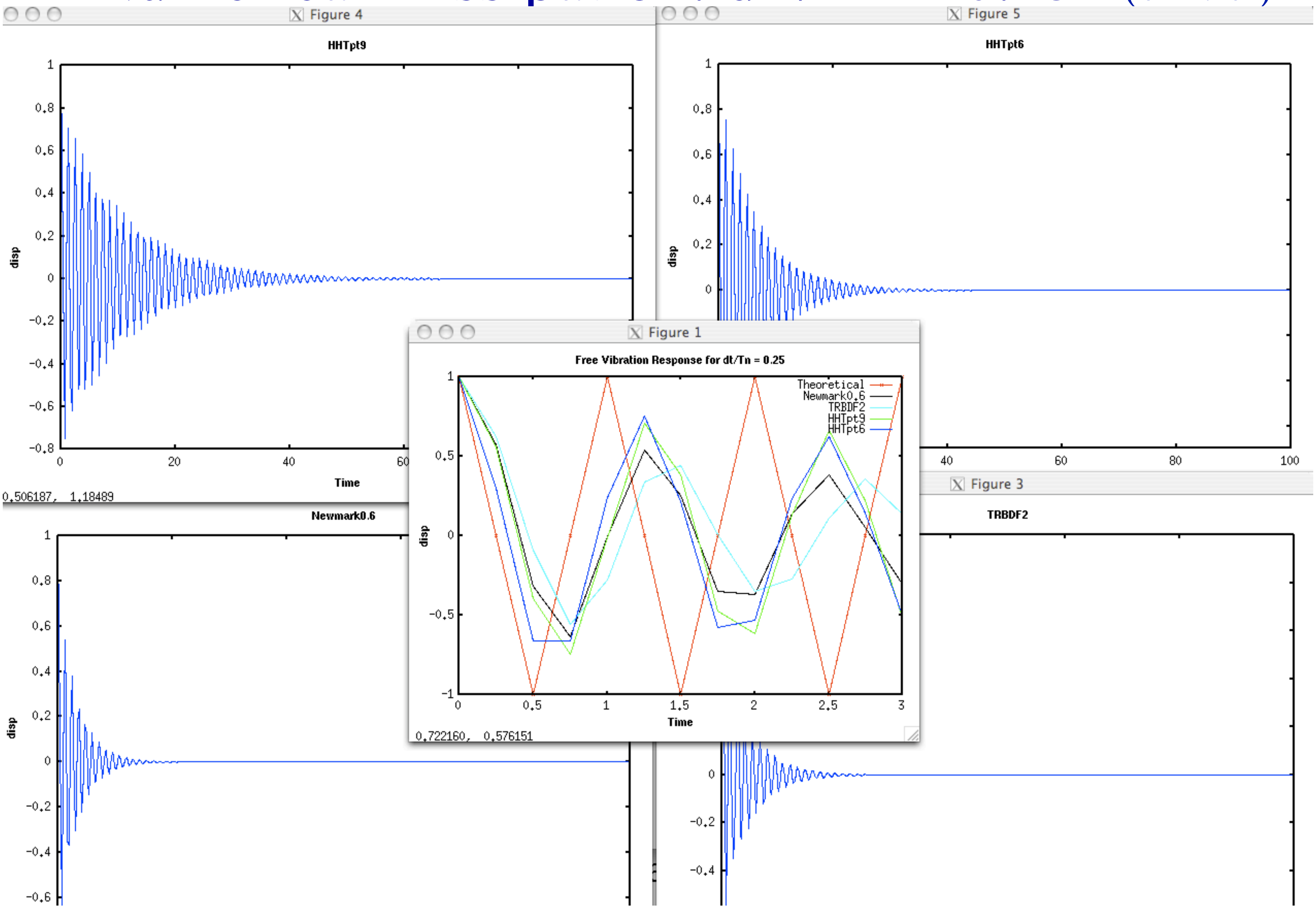
Numerical Dissipation: dT/Tn = 0.01 (ex2.tcl)



Numerical Dissipation: dT/Tn = 0.1 (ex2.tcl)



Numerical Dissipation: dT/Tn = 0.25 (ex2.tcl)



Example

(see "Dynamics of Structures" A.K. Chopra, section 3.1)

Harmonic Vibration (ex3.tcl)

$u(0) = 0$

$u'(0) = 0$

$p(t) = \sin \omega t$

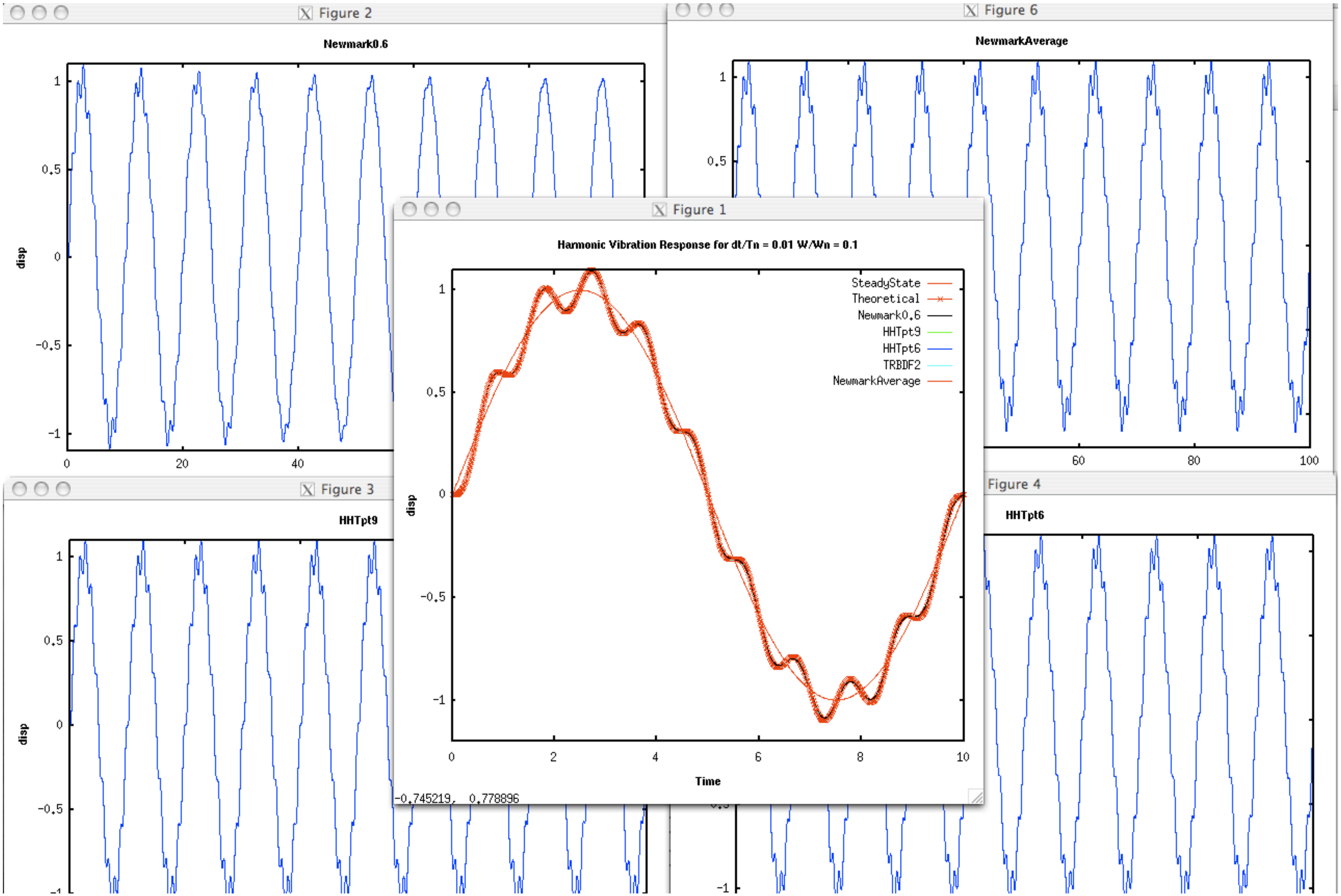
M again is chosen to give desired period

p is chosen such that $p/k = 1$.

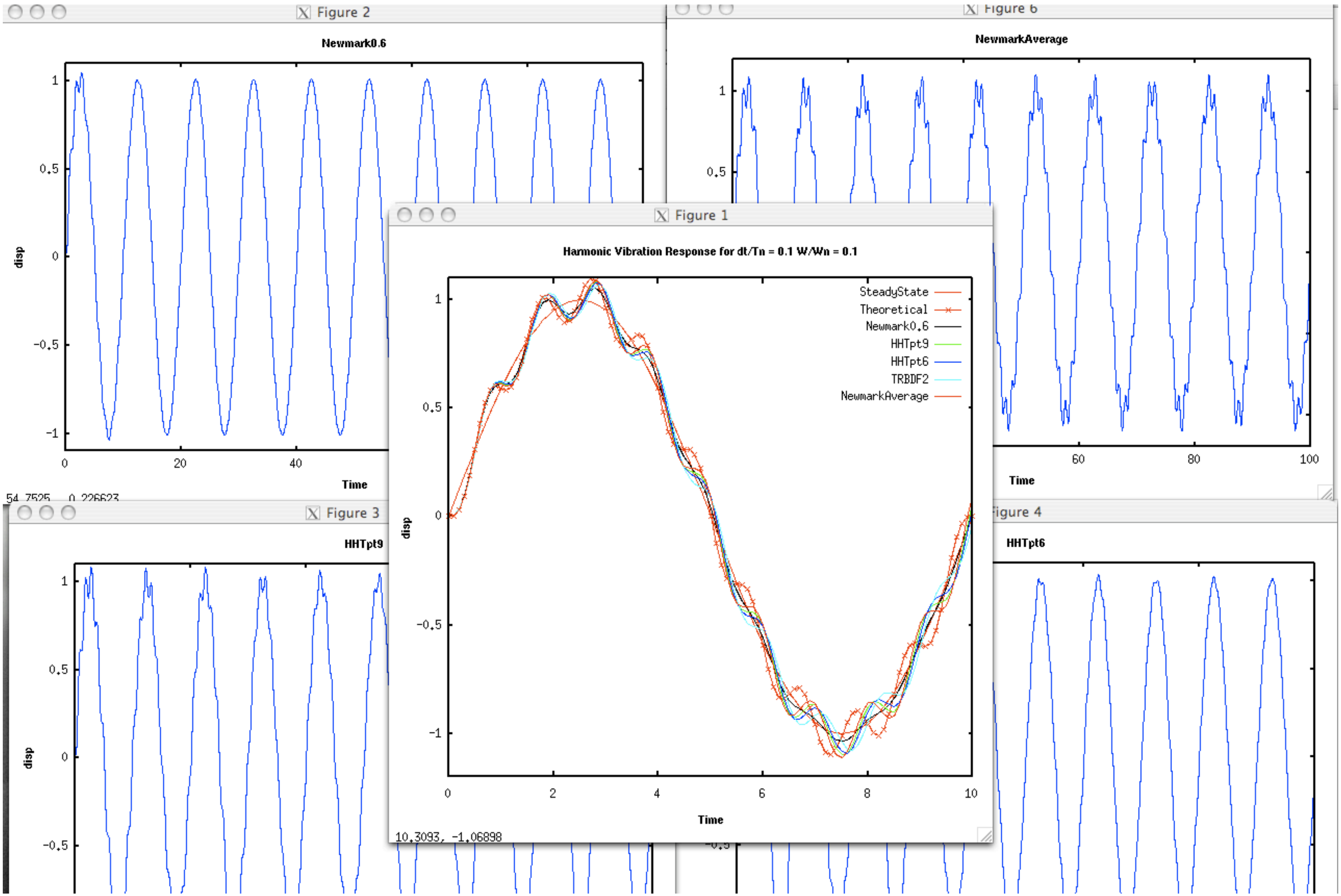
<u>exact soln</u>:

$$u(t) = \frac{p_0}{k} \frac{1}{1-(\omega/\omega_n)^2} \left(\sin \omega t - \frac{\omega}{\omega_n} \sin \omega_n t \right)$$

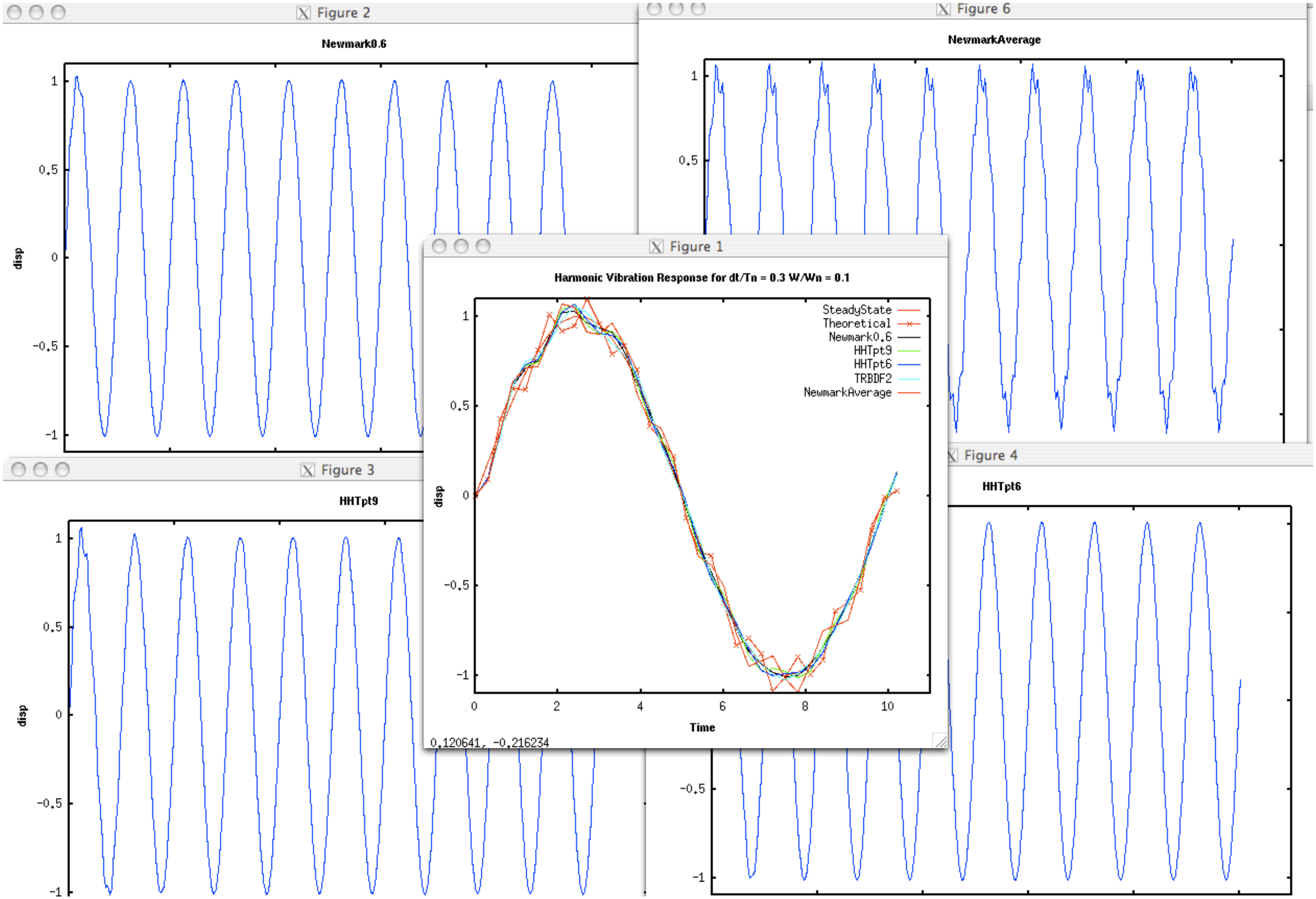
T/Tn = 0.1; dT/Tn = 0.01 (ex3.tcl)



T/Tn = 0.1; dT/Tn = 0.1 (ex3.tcl)



T/Tn = 0.1; dT/Tn = 0.3 (ex3.tcl)



Remember

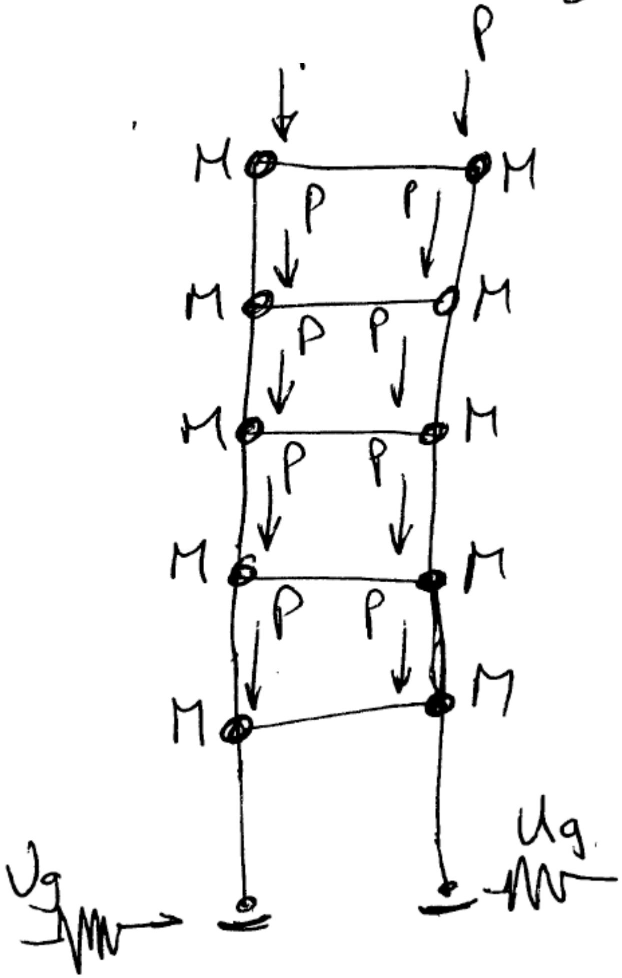
- When using dissipation to damp out higher frequencies, the choice of dT is as important as choice of integrator parameters.

- Why damp out higher frequencies?
  1. Not interested in spurious modes
  2. Contact
  3. (I know I am repeating but again) In nonlinear problems try to remove energy and hopefully allow conservation of energy (not guaranteed)

EXAMPLE

(Andreas Schellenberg, Rutherford and Chekenne)

– Building on friction pendulum bearing subjected to earthquake.



RESULTS

Periods Start: 0.50 sec to 0.0015 sec

Periods Just Before Impact: 380590.94 sec to 0.0045 sec

Periods if successful Analysis: 1.29 sec to 0.0015

Just before point of failure

Eigenvalues at end of transient:

n	lambda	omega	period
0	2.725479e-10	1.6509024804633374e-5	380590.9423199949
1	4.841790e-10	2.2004067805749007e-5	285546.5345156761
2	1.900059e-09	4.358966620656781e-5	144143.9188225047
3	2.437622e+03	49.37227967189686	0.12726139746704931
4	5.094668e+03	71.37694305586363	0.08802822085364471
5	9.174452e+03	95.78335972391028	0.06559787968693609
			0.05515718416484677
			0.04385979937343223
			0.016858990250615526
			0.016472394298701693
			0.008726856690187374
			0.00867526394323706
		3	0.006170819544074681
		3	0.006170819544074681
		7	0.006170057818824233
		4	0.006166930562842862
			0.00615859764023062
17	1.042444e+06	1021.0014691468372	0.006153943453607273
18	1.046875e+06	1023.1690964840562	0.006140906062126648
19	1.055707e+06	1027.476033783757	0.006115164831671342
20	1.555125e+06	1247.0465107605248	0.005038453059258967
21	1.560327e+06	1249.130497586221	0.0050300471562587
22	1.934602e+06	1390.8997088215958	0.00451735324073283
23	1.939489e+06	1392.6553773277867	0.00451165838258974

At end of successful analysis

Eigenvalues at end of transient:

n	lambda	omega	period
1	2.365794e+01	4.863942845058935	1.291788474357259
2	2.444645e+03	49.443351423624186	0.12707846709957166
3	3.354993e+03	57.92230140455401	0.10847609909860358
4	7.038561e+03	83.89613221120506	0.07489243117146242
5	1.297724e+04	113.91768958331274	0.05515548401799733
6	2.045299e+04	143.01395036848677	0.0439340728019222
		775	0.029451923488762022
		442	0.016661966331203787
		365	0.010499457351349708
		4962	0.008700707004971624
		6294	0.006672111636214162
		19744	0.006170795735890136
		92478	0.006170010220361572
		98169	0.0061668889774234605
		89512	0.0061622426545781695
		84738	0.006158571014872712
17	1.046880e+06	1023.1715598700259	0.006140891397328815
18	1.055708e+06	1027.4765204129972	0.006115161935431908
19	1.465409e+06	1210.5407882430068	0.005190395373871768
20	1.557795e+06	1248.1165810932887	0.005034133351289849
21	1.909504e+06	1381.8480379549699	0.00454694375546404
22	1.937262e+06	1391.855595958144	0.004514250850034686
23	1.823510e+07	4270.257603470779	0.001471383202285675

For a dT=0.001

Newmark Average Acceleration and HHT 0.9 failed

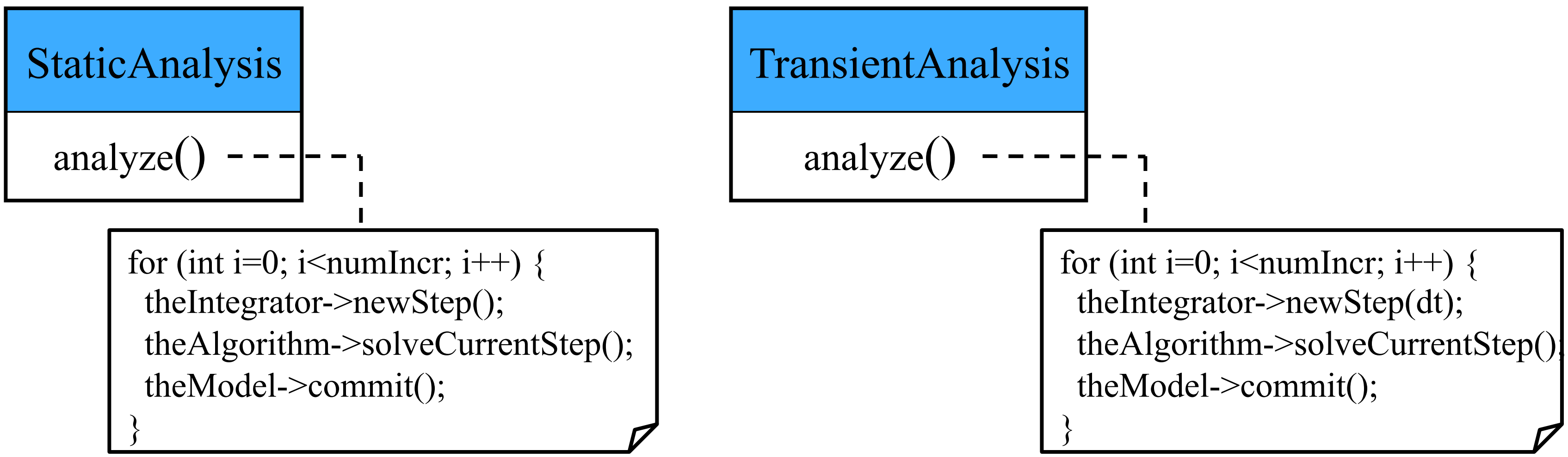
HHT 0.6, TRBDF2, and Newmark 0.6 0.3025 worked

Max recorded roof displacements: 6.03, 5.88, 5.87 respectively

analysis command:

- Static Analysis — *analysis Static*
- Transient Analysis — *analysis Transient*

  - both incremental solution strategies



- Eigenvalue
  - general eigenvalue problem

    $$(\mathbf{K}-\lambda \mathbf{M})\mathbf{\Phi}=\mathbf{0}$$

    eigen numModes? -general

  - standard eigenvalue problem

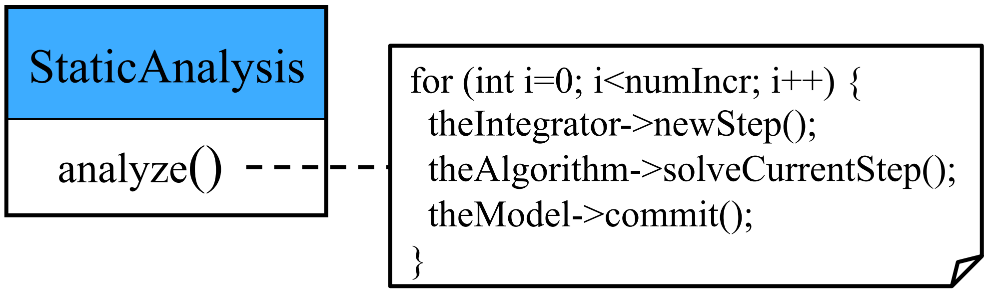
    $$(\mathbf{K}-\lambda)\mathbf{\Phi}=\mathbf{0}$$

    eigen numModes? -standard

analyze command:

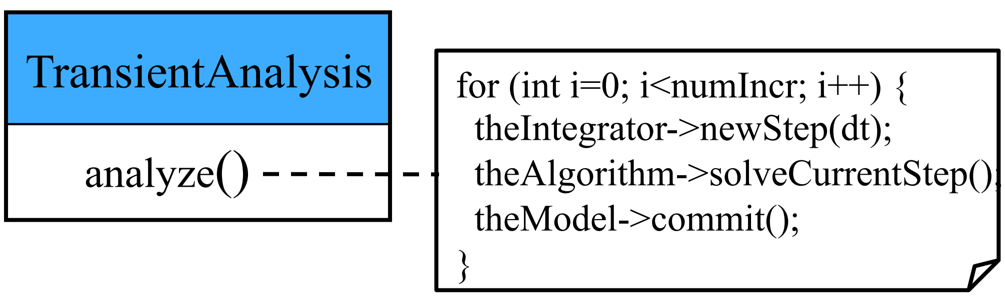
- to perform the static/transient analysis

• Static Analysis



analyze numIter?

• Transient Analysis

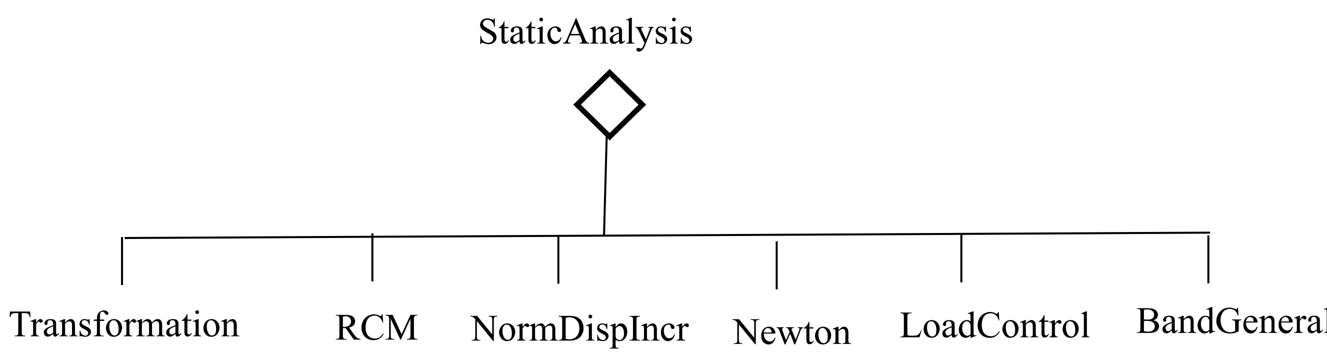


analyze numIter? Δt?

Example Static Analysis:

- Static Nonlinear Analysis with LoadControl

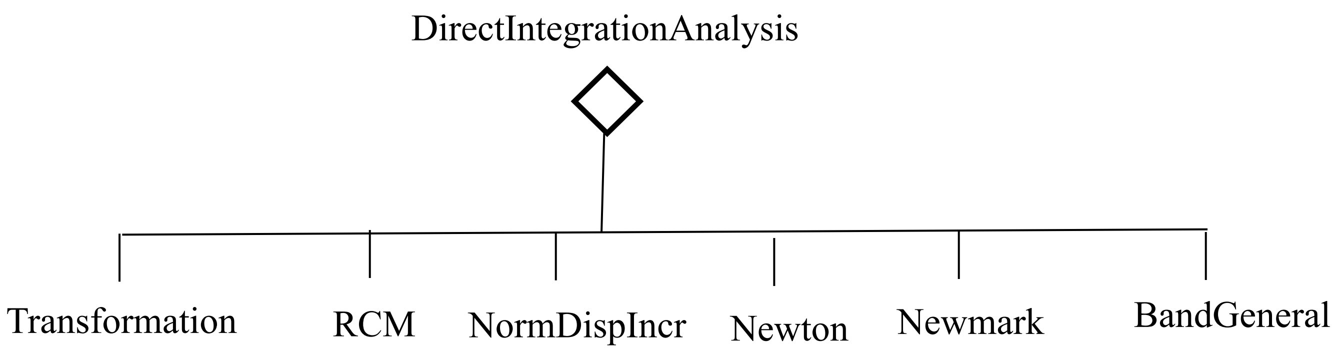
```
constraints Transformation
numberer RCM
system BandGeneral
test NormDispIncr 1.0e-6 6 2
algorithm Newton
integrator LoadControl 0.1
analysis Static
analyze 10
```



Example Dynamic Analysis:

- Transient Nonlinear Analysis with Newmark

```
constraints Transformation
numberer RCM
system BandGeneral
test NormDispIncr 1.0e-6 6 2
algorithm Newton
integrator Newmark 0.5 0.25
analysis Transient
analyze 2000 0.01
```



Remember that nonlinear analysis does not always converge

CHECK YOUR MODEL

CHECK YOUR MODEL
CHECK YOUR MODEL
CHECK YOUR MODEL
CHECK YOUR MODEL
CHECK YOUR MODEL
CHECK YOUR MODEL
CHECK YOUR MODEL
CHECK YOUR MODEL

Commands that Return Values

•analyze command

The analyze command returns 0 if successful.
It returns a negative number if not

set ok [analyze numIter <Δt>]

•getTime command

The getTime command returns pseudo time in Domain.

set currentTime [getTime]

•nodeDisp command

The nodeDisp command returns a nodal displacement.

set disp [nodeDisp node dof]

Example Usage – Displacement Control

```
set maxU 15.0; set dU 0.1
constraints transformation
numberer RCM
system BandGeneral
test NormDispIncr 1.0e-6 6 2
algorithm Newton
integrator DispControl  3 1 $dU
analysis Static
set ok 0
set currentDisp 0.0
while {$ok == 0 && $currentDisp < $maxU} {
        set ok [analyze 1]
        if {$ok != 0} {
          test NormDispIncr 1.0e-6 1000 1
          algorithm ModifiedNewton –initial
          set ok [anal;yze 1]
          test NormDispIncr 1.0e-6 6 2
          algorithm Newton
        }
        set currentDisp [nodeDisp 3 1]
}
```

Example Usage – Transient Analysis

```
set tFinal 15.0;
constraints  Transformation
numberer RCM
system BandGeneral
test NormDispIncr 1.0e-6 6 2
algorithm Newton
integrator Newmark 0.5 0.25
analysis Transient
set ok 0
set currentTime 0.0
while {$ok == 0 && $currentTime < $tFinal} {
        set ok [analyze 1 0.01]
        if {$ok != 0} {
          test NormDispIncr 1.0e-6 1000 1
          algorithm ModifiedNewton –initial
          set ok [analyze 1 0.01]
          test NormDispIncr 1.0e-6 6 2
          algorithm Newton
        }
        set currentTime [getTime]
}
```

Still Not Working!

1. Search the Message Board
2. Post Problem on the Message Board

To check which scale of elcentro earthquake makes the SDF inelastic, the following file was used. By trial and error, scale 10 was found in OpenSees, which is inconsistent with the results(scale 4) from using other programs.

Is there anything wrong in this file?

```
# create ModelBuilder (with two-dimensions and 2 DOF/node)
model BasicBuilder -ndm 1 -ndf 1


# Define geometry for model
# ------------------------
puts "Define geometry for model"
set k1 2.75
set uy 1.35
```

i suggest you check your other input files .. if you have a look at chopra's book he plots the respone spectrum for this e.q. .. for a period of 0.1, D for an elastic system is with 0% damping is about .11 (fig 6.8.1 in my version) .. so you need a scale factor of about 12 [1.35/.11] to reach the ultimate.
(note using Newmark 0.5 0.25 you get .11)

to compute the scale factor for yield i suggest you also stop playing with trying to predict the scale factor & just divide yield disp by the max response from elastic system.

Segmentation Faults, etc:

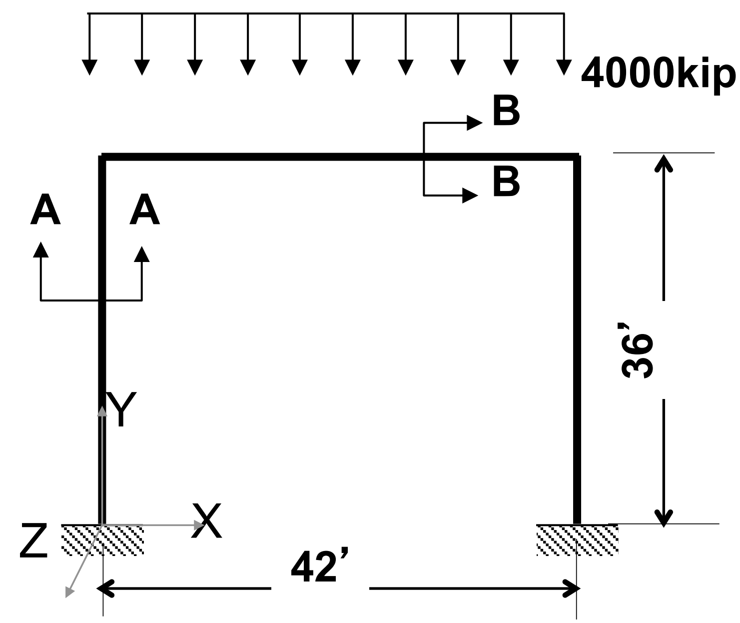
- Email: fmckenna@ce.berkeley.edu

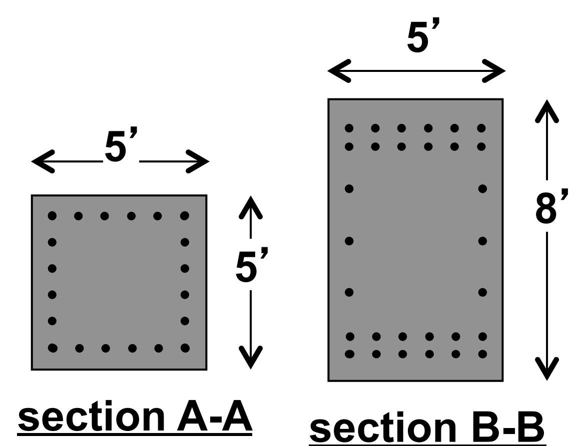
NOTE: Zip up your files in 1 directory and send them to us

Model

RCFrame.tcl

```
model Basic -ndm 2 -ndf 3
set ft 12.0
set in 1.0
set cm 0.3937
# Set parameters for overall model geometry
set width    [expr 42.0*$ft]
set height   [expr 36.0*$ft]

# Create nodes
node  1      0.0     0.0
node  2    $width     0.0
node  3      0.0 $height
node  4    $width $height

# set boundary conditions
fix   1     1    1    1
fix   2     1    1    1

# create materials for sections
uniaxialMaterial Concrete01  1  -6.0 -0.004 -5.0 -0.014; # core
niaxialMaterial Concrete01  2  -5.0 -0.002  0.0 -0.006; # cover
uniaxialMaterial Steel01  3 60.0 30000.0 0.01

# create sections
set bWidth  [expr 5.0*$ft]; set bDepth  [expr 5.0*$ft]; set cover  1.5
set As      0.60;     # area of no. 7 bars

source RCsection2D.tcl
RCsection2D 1 $bWidth $bDepth $cover 1 2 3 6 3 \
   $As 10 10 2
#create geometric transformations
geomTransf PDelta 1
geomTransf Linear 2

# Create the coulumns usind distributed plasticity  elements
set np 5
set eleType forceBeamColumn
element $eleType  1   1   3   $np   1      1
element $eleType  2   2   4   $np   1      1
# Create the beam element using elastic element
set d [expr 8.0*$ft];          # Beam Depth
set b [expr 5.0*$ft];          # Beam Width
set Eb 3600.0; set Ab [expr $b*$d];; set Izb [expr ($b*pow($d,3))/12.0];
section Elastic 2   $Eb $Ab $Izb;       # elastic beam section
#element elasticBeamColumn   3   3   4   $Ab   $Eb  $Izb    2
element $eleType  3   3   4   $np    2      2
```





RCFrameGravity,tcl

Gravity Load Analysis

```
# first source in the model
source RCFrame.tcl

# Create the gravity loads
set W 4000.0;
timeSeries Linear 1
pattern Plain 1 1 {
   eleLoad -ele 3 -type -beamUniform [expr -$W/$width]
}

# create the analysis
system BandGeneral
constraints Transformation
numberer RCM
test NormDispIncr 1.0e-12  10 3
algorithm Newton
integrator LoadControl 0.1
analysis Static

# perform the analysis
analyze 10
```

Terminal — bash — 87×22

```
         OpenSees -- Open System For Earthquake Engineering Simulation
        Pacific Earthquake Engineering Research Center -- 2.3.0.alpha

            (c) Copyright 1999,2000 The Regents of the University of California
                                 All Rights Reserved
    (Copyright and Disclaimer @ http://www.berkeley.edu/OpenSees/copyright.html)



 Node: 3
        Coordinates  : 0 432
        Disps: 0.00201291 -0.0673474 -0.00191622
         unbalanced Load: 0 0 0
        ID : 3 4 5


Element: 1 Type: ForceBeamColumn2d      Connected Nodes: 1 3
        Number of Sections: 5   Mass density: 0
Lobatto
        End 1 Forces (P V M): 2000 -165.625 -24646.3
        End 2 Forces (P V M): -2000 165.625 -46903.7
examples>
```

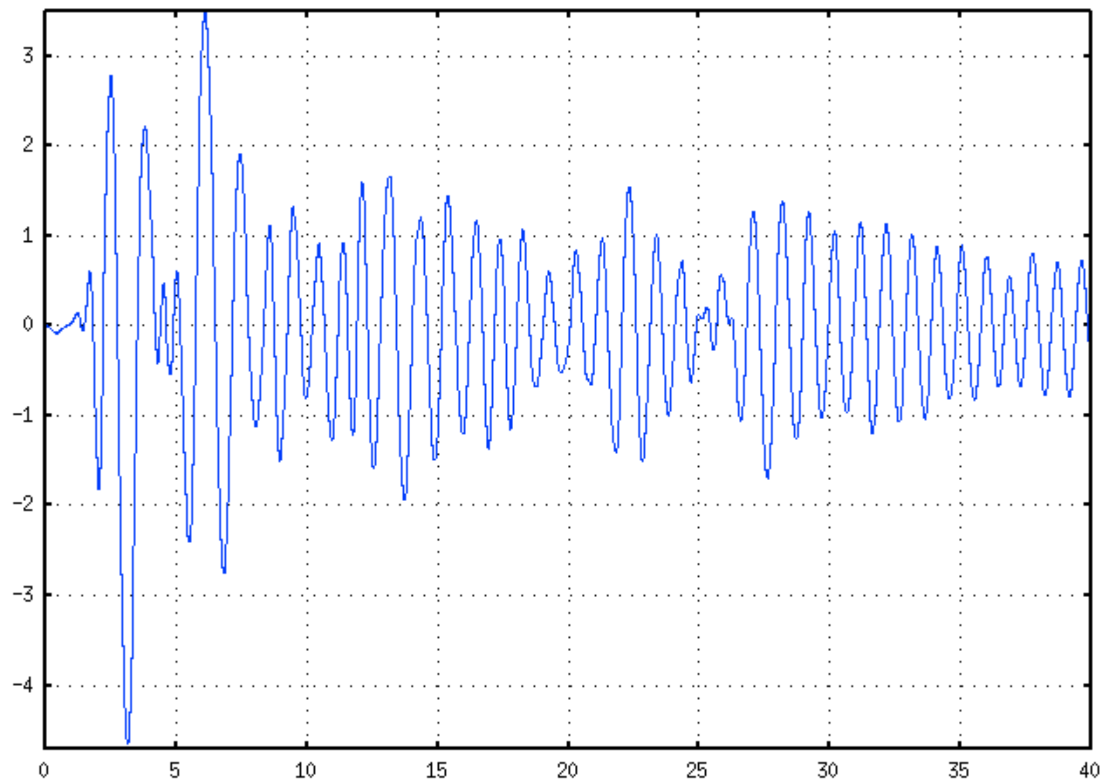
Transient Analysis - Uniform Excitation

```
source RCFrameGravity.tcl
puts "Gravity load analysis completed"

# Set the gravity loads to be constant
#   & reset the time in the domain
loadConst -time 0.0

# Define nodal mass
set g 386.4
set m [expr ($W/2.0)/$g];

#    tag   MX    MY    RZ
mass  3    $m   $m    1.0e-16
mass  4    $m   $m    1.0e-16

# Define dynamic loads
set record IELC180
source ReadRecord.tcl
ReadRecord $record.AT2  $record.dat dT nPts
timeSeries Path 2 -filePath $record.dat -dt $dT
pattern UniformExcitation  2 1 -accel 2
rayleigh 0.0 0.0 0.0 0.0

#create a recorder
recorder Node -time -file disp.out -node 3 4 -dof 1 2 3 disp

# remove old analysis
wipeAnalysis
```

```
#create the analysis
system BandGeneral
constraints Plain
test NormDispIncr 1.0e-8  10
algorithm Newton
numberer RCM
integrator Newmark  0.5  0.25
analysis Transient

set tFinal [expr $nPts * $dT]
set tCurrent [getTime]
set ok 0
# perofrm the analysis
while {$ok == 0 && $tCurrent < $tFinal} {
    set ok [analyze 1 $dT]
    # if the analysis fails try initial tangent iteratio
    if {$ok != 0} {
        puts "regular newton failed .. lets try anothe
        test NormDispIncr 1.0e-8  1000 1
        algorithm ModifiedNewton -initial
        set ok [analyze 1 $dT]
        test NormDispIncr 1.0e-12  10
        algorithm Newton
    }
    set tCurrent [getTime]
}
# Print a message to indicate if analysis succesful
if {$ok == 0} {
   puts "Transient analysis completed SUCCESSF
} else {
    puts "Transient analysis completed FAILED";
}
```

Terminal — bash — 87×27

```
examples> OpenSees RCFrameUniform.tcl


         OpenSees -- Open System For Earthquake Engineering Simulation
        Pacific Earthquake Engineering Research Center -- 2.3.0.alpha

            (c) Copyright 1999,2000 The Regents of the University of California
                                All Rights Reserved
    (Copyright and Disclaimer @ http://www.berkeley.edu/OpenSees/copyright.html)



 Node: 3
        Coordinates  : 0 432
        Disps: 0.00201291 -0.0673474 -0.00191622
         unbalanced Load: 0 0 0
        ID : 3 4 5


Element: 1 Type: ForceBeamColumn2d      C
        Number of Sections: 5   Mass dens
Lobatto
        End 1 Forces (P V M): 2000 -165.6
        End 2 Forces (P V M): -2000 165.6
Gravity load analysis completed
Transient analysis completed SUCCESSFULLY
examples> 
```



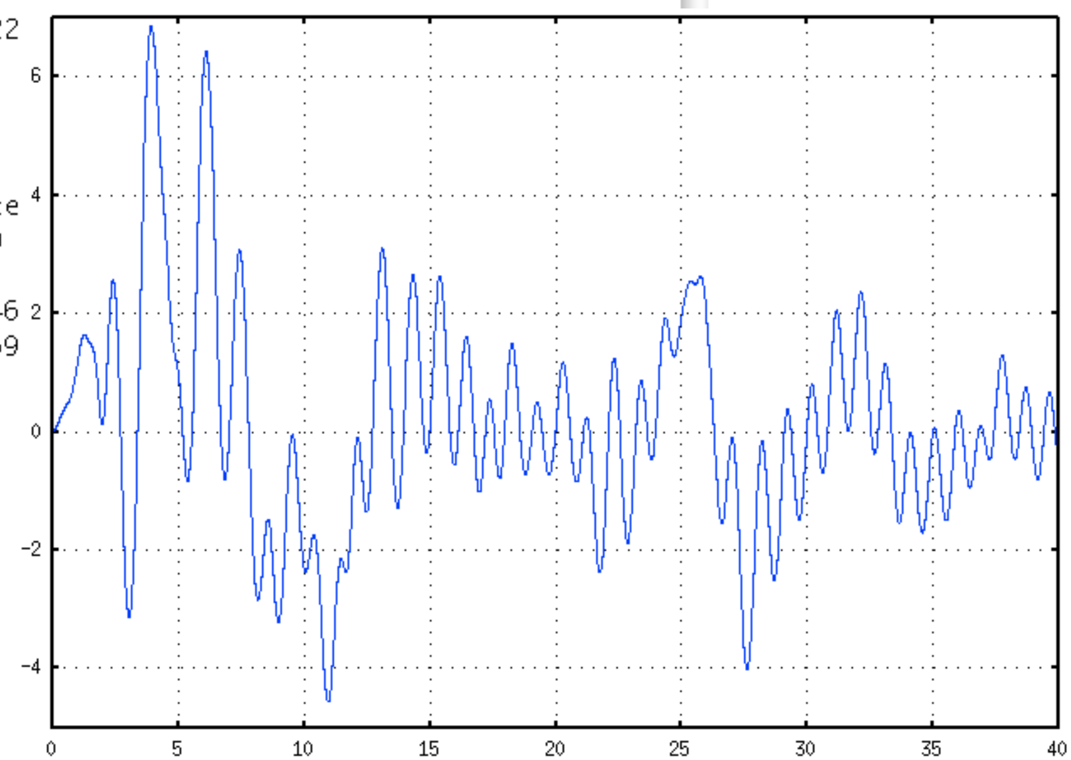
Transient Analysis - MultiSupport Excitation

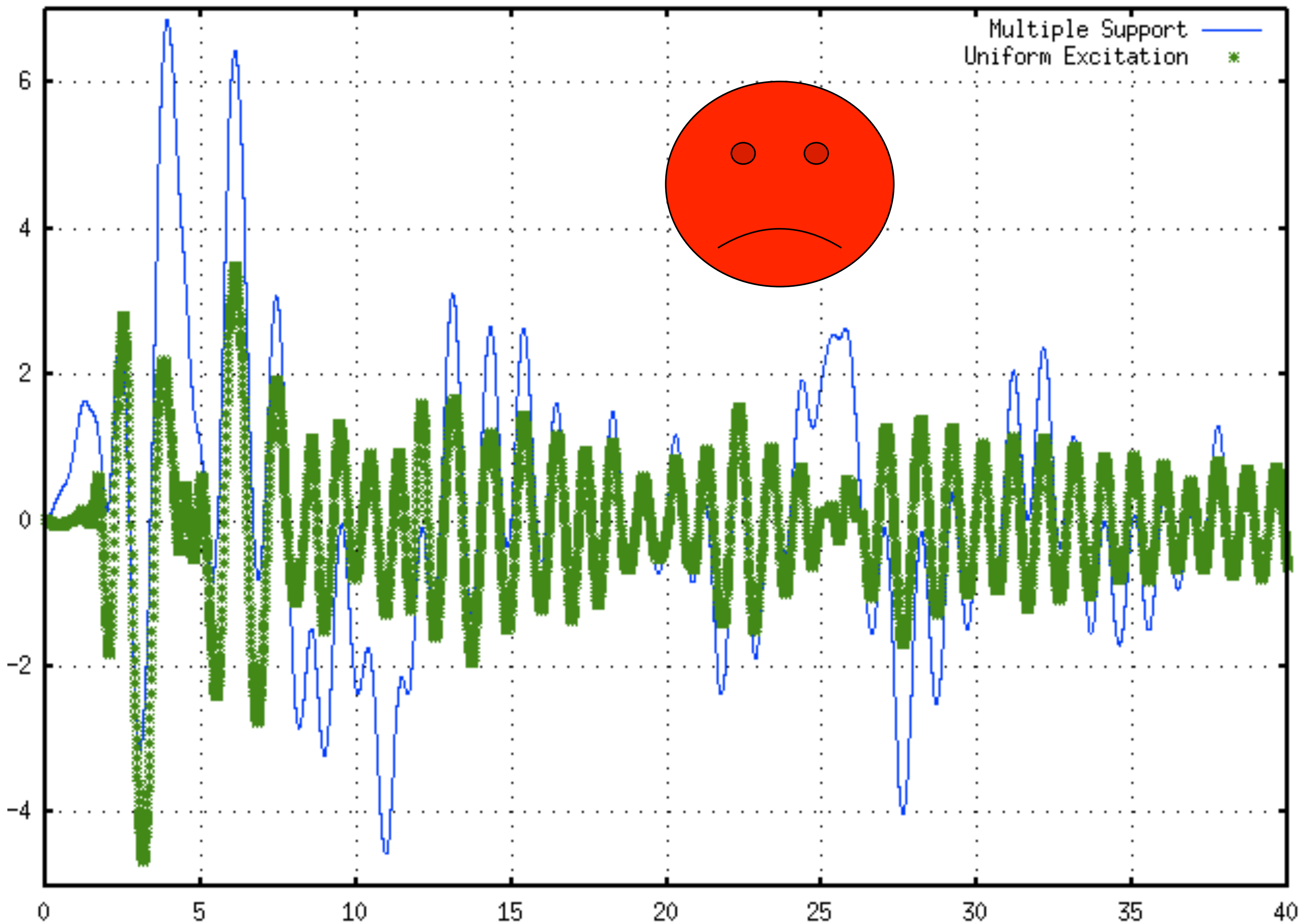
```
source RCFrameGravity.tcl

# Set the gravity loads to be constant
#  & reset the time in the domain
loadConst -time 0.0

# Define nodal mass
set m [expr ($W/2.0)/$g];

#    tag   MX    MY    RZ
mass  3    $m   $m    1.0e-16
mass  4    $m   $m    1.0e-16

# Define dynamic loads
# Set some parameters
set record IELC180
# Source in TCL proc to read PEER SMD record
source ReadRecord.tcl
ReadRecord $record.DT2  $record.dat dT nPts
timeSeries Path 2 -filePath $record.dat -dt $dT -factor $cm
pattern MultiSupport  2   {
   groundMotion 5 Plain -disp 2
   imposedMotion 1 1 5
   imposedMotion 2 1 5
}
recorder Node -time -file multi.out -node 1 3 -dof 1 disp
```

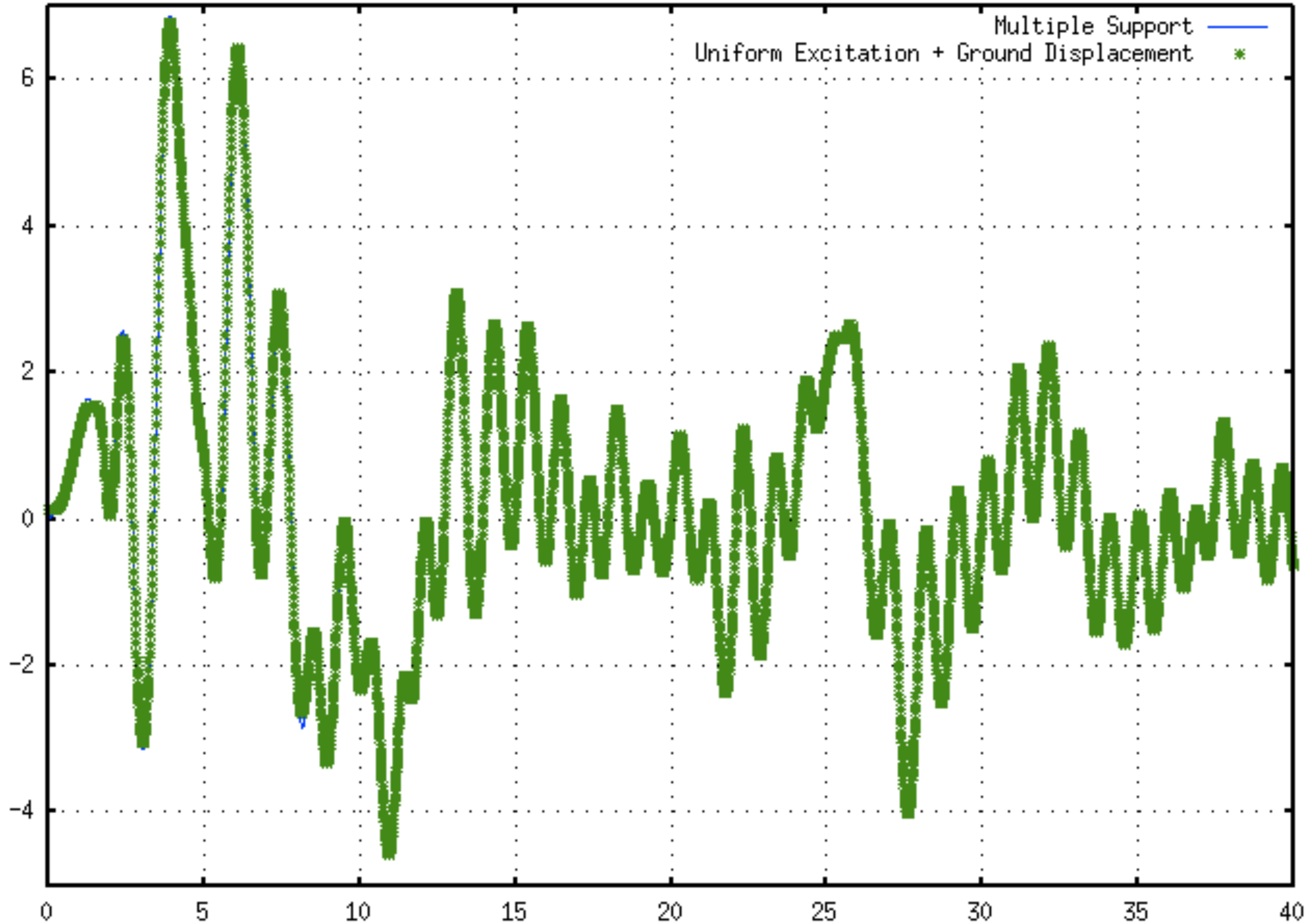
```
rayleigh 0.0 0.0 0.0 0.0

remove sp 1 1
remove sp 2 1

wipeAnalysis

#create the analysis
system BandGeneral
constraints Plain
test NormDispIncr 1.0e-8  10
algorithm Newton
numberer RCM
integrator Newmark  0.5  0.25
analysis Transient

set tFinal [expr $nPts * $dT]
set tCurrent [getTime]
set ok 0
# perofrm the analysis
while {$ok == 0 && $tCurrent < $tFinal} {
   set ok [analyze 1 $dT]
   # if the analysis fails try initial tangent iteratic
   if {$ok != 0} {
      puts "regular newton failed .. lets try anoth
      test NormDispIncr 1.0e-8  1000 1
      algorithm ModifiedNewton -initial
      set ok [analyze 1 $dT]
      test NormDispIncr 1.0e-12  10
      algorithm Newton
   }
   set tCurrent [getTime]
}
# Print a message to indicate if analysis succesfu
if {$ok == 0} {
  puts "Transient analysis completed SUCCESS
} else {
   puts "Transient analysis completed FAILED";
}
```

Terminal — bash — 87×27

```
examples> OpenSees RCFrameMulti.tcl


         OpenSees -- Open System For Earthquake Engineering Simulation
        Pacific Earthquake Engineering Research Center -- 2.3.0.alpha

            (c) Copyright 1999,2000 The Regents of the University of California
                              All Rights Reserved
    (Copyright and Disclaimer @ http://www.berkeley.edu/OpenSees/copyright.html)



 Node: 3
        Coordinates  : 0 432
        Disps: 0.00201291 -0.0673474 -0.00191622
         unbalanced Load: 0 0 0
        ID : 3 4 5


Element: 1 Type: ForceBeamColumn2d      Connecte
        Number of Sections: 5   Mass density: 0
Lobatto
        End 1 Forces (P V M): 2000 -165.625 -246
        End 2 Forces (P V M): -2000 165.625 -469
Gravity load analysis completed
Transient analysis completed SUCCESSFULLY
examples>
```




Multiple Support
Uniform Excitation
6
4
2
0
-2
-4
0
5
10
15
20
25
30
35
40



Multiple Support
Uniform Excitation + Ground Displacement
6
4
2
0
-2
-4
0
5
10
15
20
25
30
35
40

Parameter Study - Response Spectra

```
source READSMDFile.tcl
modelBuilder BasicBuilder -ndm 1 -ndf 1

# set a bunch of parameters
set PI 3.14159265
set g 386.4
set TnMin 0.1;  #min period
set TnMax 2.0; #max period
set TnIncr 0.1; #period incr
set M 1.0;         #mass
set A 1.0;          #area
set L 1.0;          #length
set motion ELCENTRO
set outFilename spectrum.dat

# open output file
Set outFileID [open $outFilename w]

#create accel series
ReadSMDFIle $motion.AT2 $motion.acc dt
Set accelSeries "Path -filePath $motion.acc \
       -dt $dt -factor $g"

# loop over period range
Set Tn $TnMin
while {$Tn <= $TnMax} {
    wipe
     set w [expr 2.0 * $PI / $Tn]
     set K [expr $w * $w * $M]
     set E [expr $k * $l / $A
      node 1 0.0
      node 2 $l -mass $M
      fix 1 1
      uniaxialMaterial Elastic 1 $E
      element truss 1 1 2 $A 1
      pattern UniformExcitation 2 1 -accel $accelSeries
      rayleigh 0.0 0.0 0.0 0.0 0.0

      recorder EnvelopeNode -file envelope.out -node 2 -dof 1 disp
      system ProfileSPD
      test NormDispIncr 1.0e-16 10
     algorithm Newton
     integrator Newmark 0.5 0.25
     analysis Transient
      analyze 2000 $dt

     if [catch {open envelope.out r} inFileID]
       puts puts "ERROR - could not open file"

    set min [gets $inFileID]
    set max [gets $inFileID]
    set absMax [gets $inFileID]
    close $inFileID
    puts $outFileID "$Tn $absmax"
    set Tn [expr $Tn + $TnIncr]
}
close $outFileID
```

Terminal — OpenSees — 82x38

```
cee-84-111:~/OpenSees/EXAMPLES/ExampleScripts/ExampleScripts fmk$ ~/bin/OpenSees


         OpenSees -- Open System For Earthquake Engineering Simulation
        Pacific Earthquake Engineering Research Center -- Version 1.6.0

            (c) Copyright 1999 The Regents of the University of California
                                 All Rights Reserved


OpenSees > source ResponseSpectra.tcl
OpenSees > cat spectrum.dat
0.1 0.0706084
0.2 0.419001
0.3 0.753439
0.4 1.47281
0.5 2.68804
0.6 3.0994
0.7 3.37357
0.8 3.70962
0.9 6.24449
1.0 5.9645
1.1 4.9327
1.2 4.75759
1.3 3.94977
1.4 4.41569
1.5 4.72872
1.6 5.93379
1.7 6.3168
1.8 6.72183
1.9 7.40134
2.0 7.47503
```