Mechanics calculations on Earthquake stops – part 2

Justin Greenhalgh, others
May/June 2006

Version 01 completed shortly after PDR#3

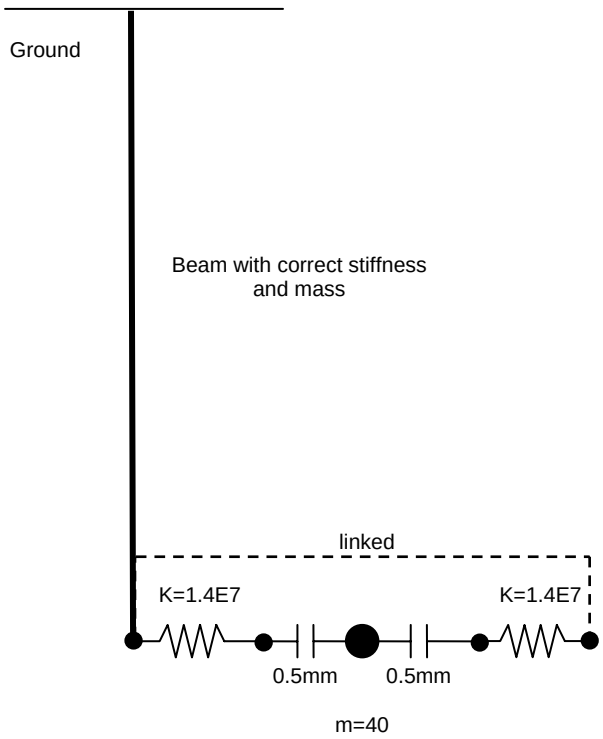
1. BACKGROUND

This note continues the work reported in T060053

2. ADDITION OF A BEAM TO THE MODEL

A weakness of the previous approach was that a 12kg mass was impacting directly on the mass (via the earthquake stops). I felt this to be rather sever, and so decided to implement a simple beam to simulate the structure.

The general idea is shown here:

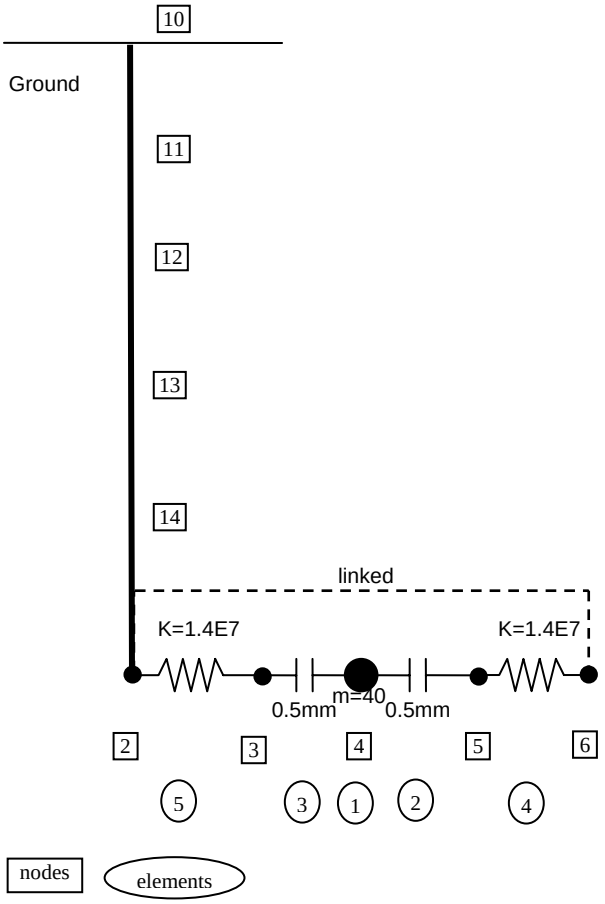


Working from the length and static deflection of the structure we can derive a second moment of area. Working from the length and mass of the structure we can derive a lineal mass density (most easily expressed as a cross-sectional area and volumetric density). Since the choice of cross-section is arbitrary, I chose to use a square cross-section with dimensions that would give the correct second moment of area. Appendix 1 has a macro used to test just the beam part of the model. The model had 5 nodes along the length of the beam.

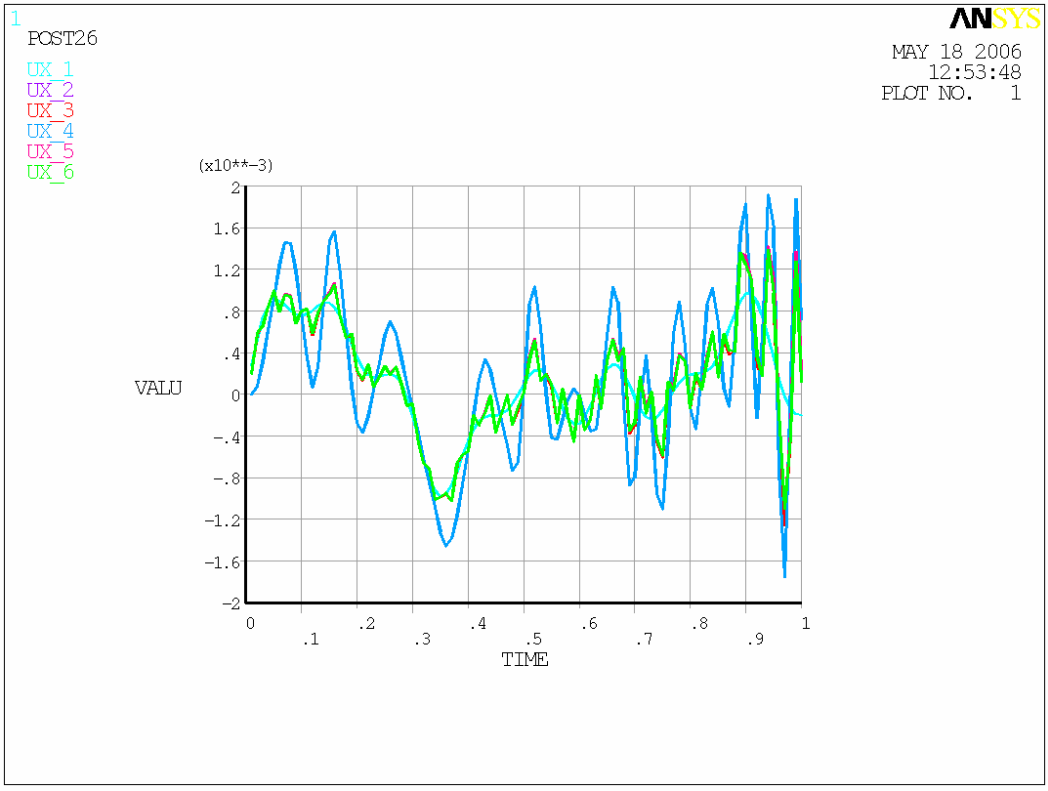
The parameter set given in appendix 1 (length=1.7m, static stiffness 2E6 N/m as before, mass 40kg) gave a natural frequency for the beam of 65 Hz, which is about right.
I obtained the appropriate parameters for the currently-proposed structure:

Length = 1.7m
Mass = 17 + 24 + 9.6 (sleeve + upper + lower) = 52 kg
Static stiffness = xxxx (to a stiff point at the corner of the sleeve)
Static stiffness = yyyy (to the attachment point of the EQ stop)

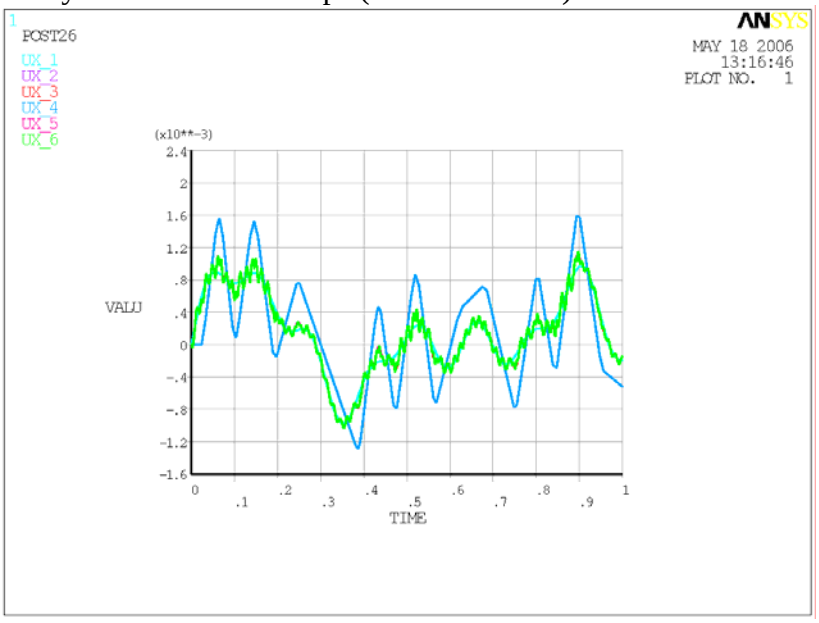
Combining the macros gives this model, and the macro in appendix 2.



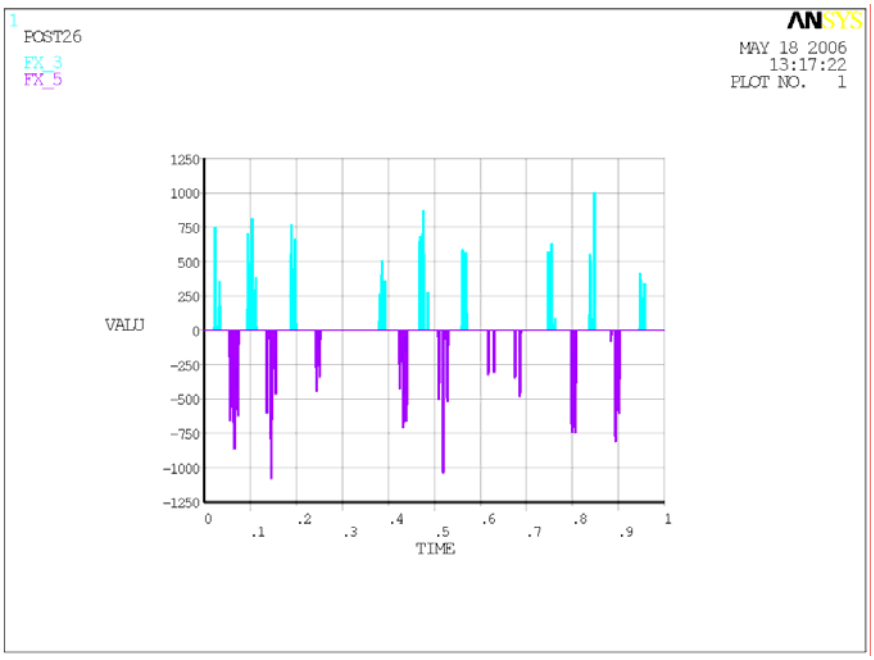
With just 100 time steps, the results look rather as before until the coarseness of the time sep leads to large movements towards the end of the run:



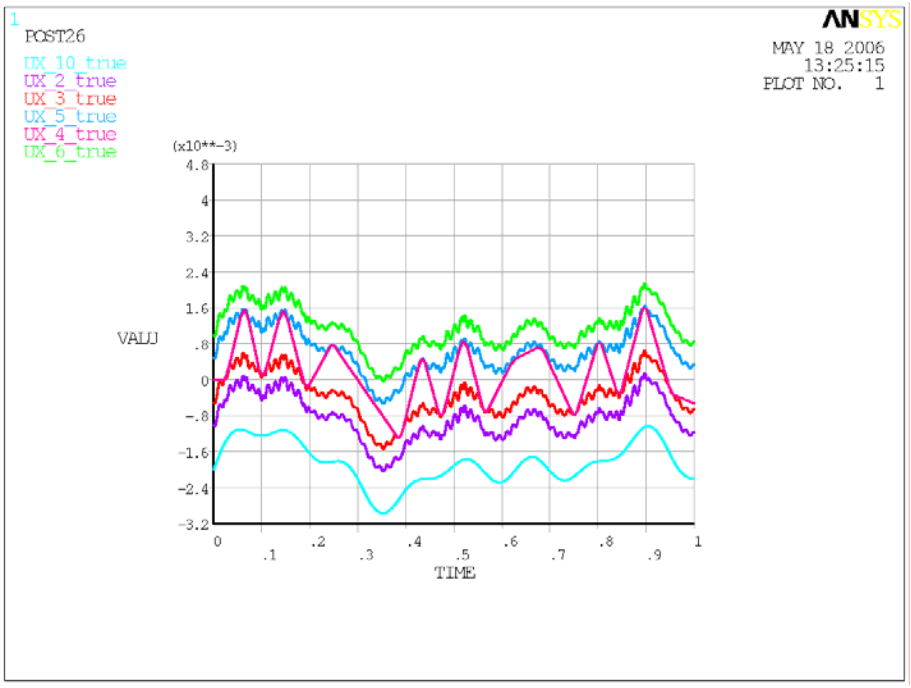
So try with 5000 time steps (time for lunch!)



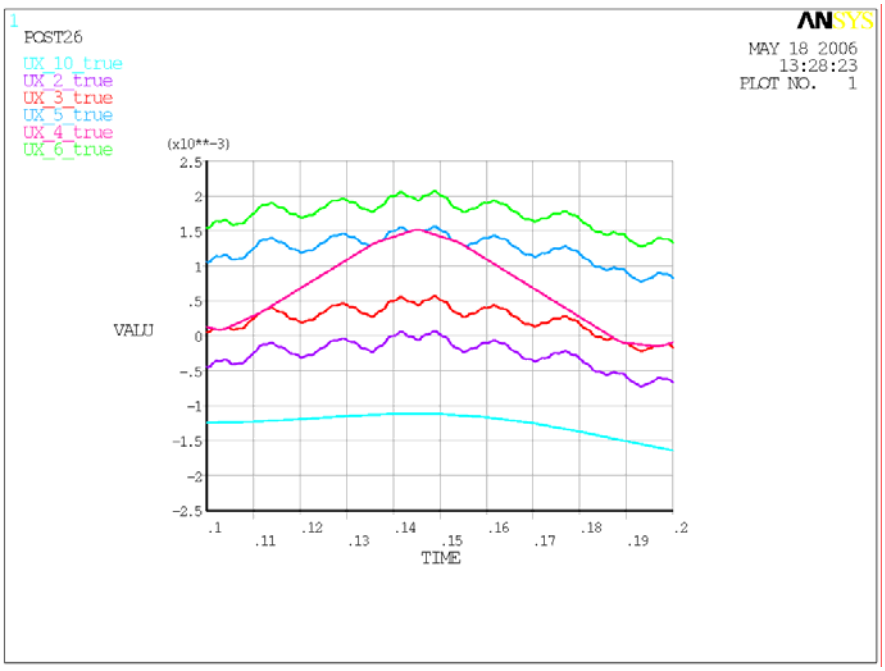
Looks good. Forces, however, are much as before at around 1000N:



For those who like the “opened up” plot, it is now included in the macro:



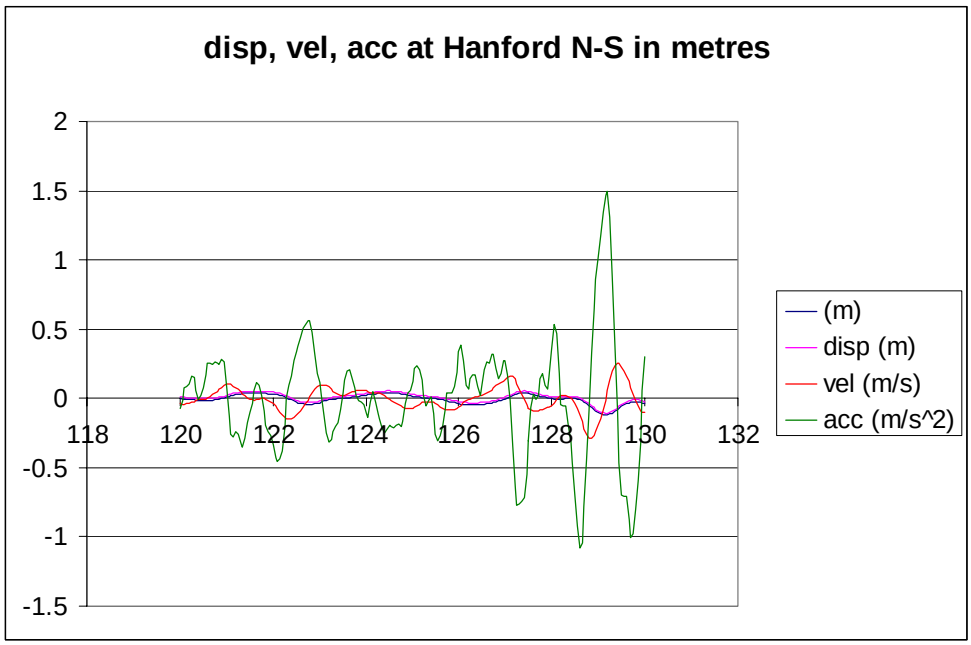
And zoomed in:



3. USING REAL EARTHQUAKE DATA

Many thanks to Dennis Coyne for working hard to extract data about the Montana Earthquake of July 2006. He supplied a spreadsheet with displacement, velocity and acceleration for each of three directions at Hanford.

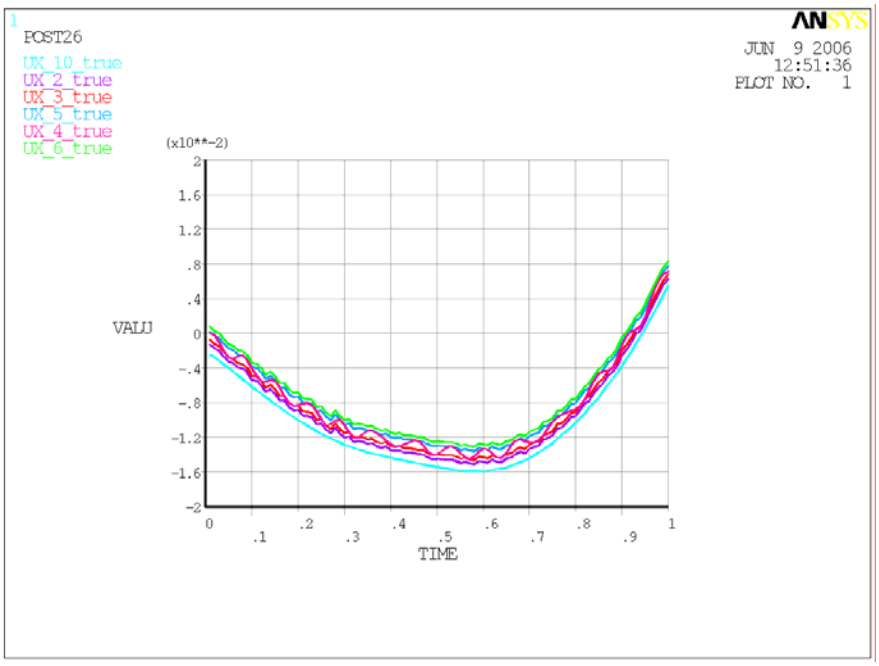
I chose to work with the North-South data, simply because it contained the highest acceleration peak. This is in a sense an arbitrary choice; the peak was scaled in the next step. I sliced 10 seconds of data near the peak (t=120 to 130 seconds). The peak acceleration was 7.36 mm/sec^2, and needed data with a peak of 1.5 m/sec^2 (E010613 section 10 paragraph 2). So I factored all the data (position (mm), velocity (mm/sec), acceleration(mm/sec^2)) by (1.5/7.36) to give results in meters at the required magnitude. Here are the data (the dark blue displacement curve has been offset to start at zero):



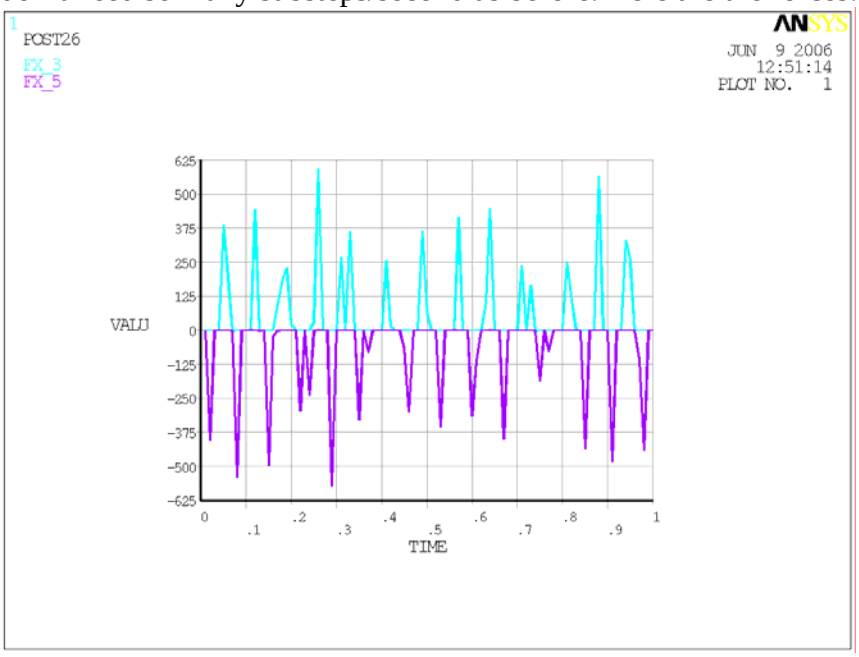
(NOTE that the max displacement is ~120 mm!)

3.1 Run for one second of data

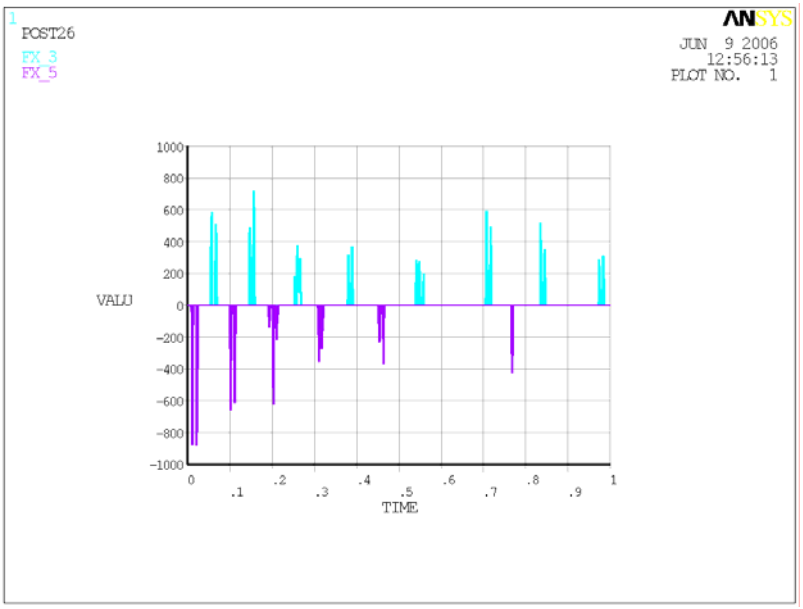
I tried running the first second in just 100 steps to check the data reading was OK, with this result:



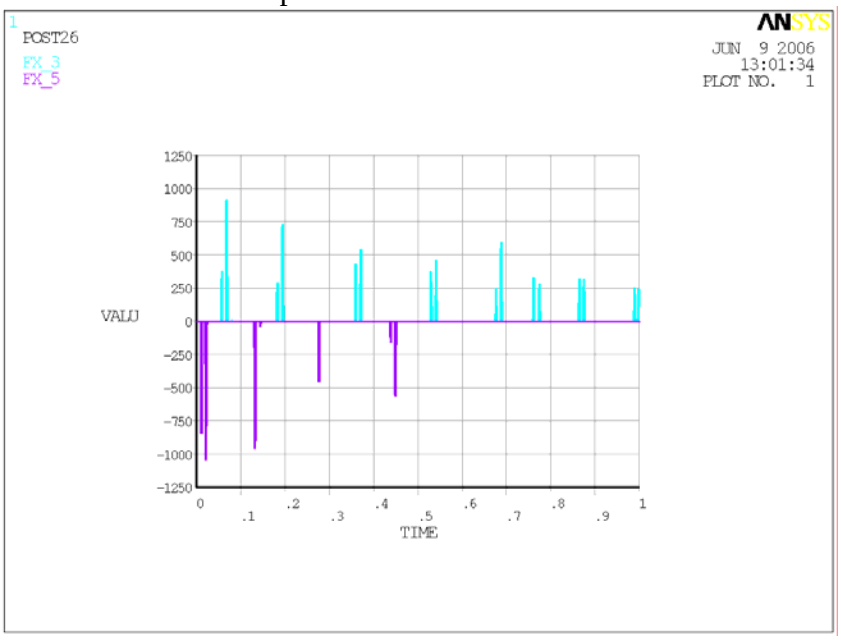
So it seems that the real earthquake is much gentler in terms of accelerations and so it appears I don't need so many substeps/second as before. Here are the forces:



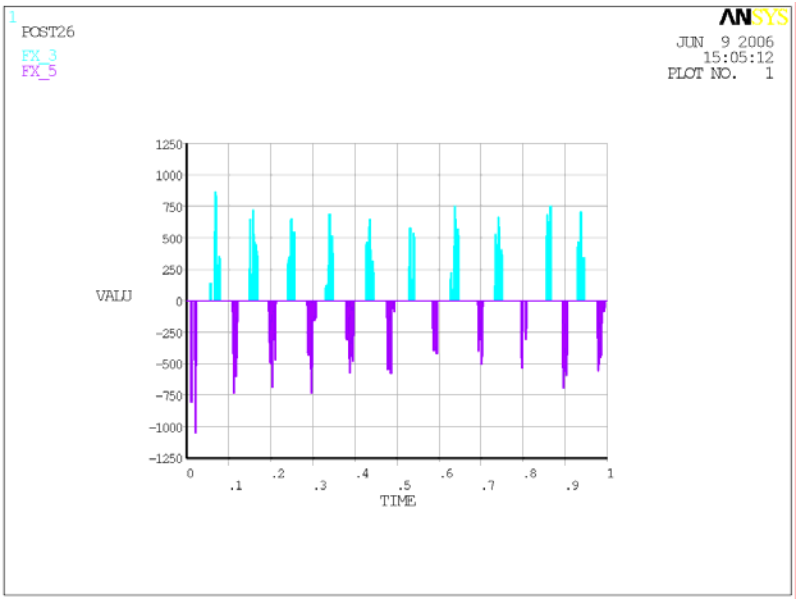
But with 1000 substeps for the first second, here are the forces:



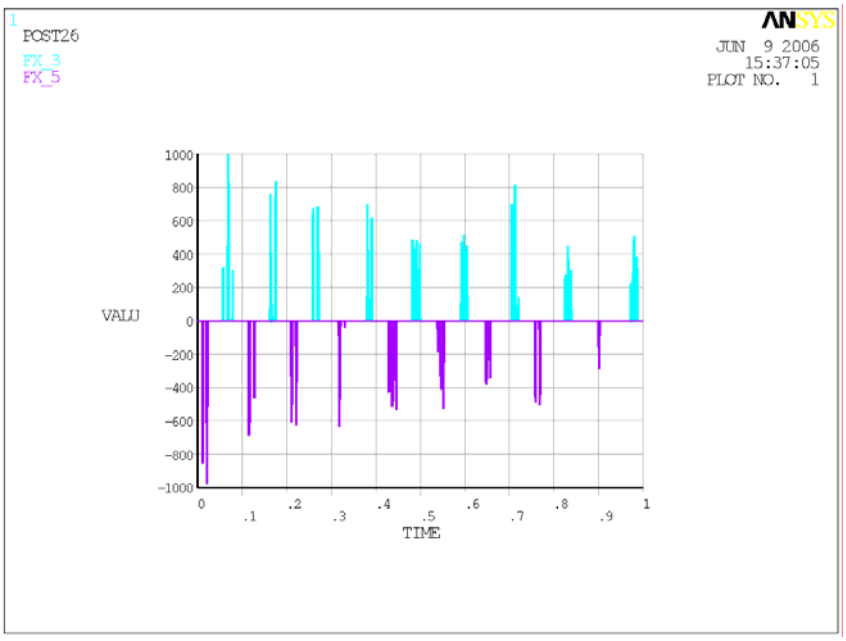
And with 5000 substeps:



OK – time for lunch. 10,000 substeps (needed two steps of 5,000 substeps each to avoid memory pointer error):



20,000 supsteps (4 steps, 5000 substeps each):



Hmmmm. Although the exact form of the contact is different each time, the magnitude of the forces is not changing. So settle on 5,000 substeps per second and press on to a full solution for the ten seconds.

3.2 Run with ten seconds of data.

```
/solve
solcontrol,0    !used in VM81, no idea why, if ommitted solution fails.
ANTYPE,4  !transient
!*
TRNOPT,FULL   !Full analysis - no shortcuts
LUMPM,0
!*
```

```
OUTRES,ALL,1
kbc,0  !Ramped BC

NSUBST,5000
Autots,off
*do,jtime,1,10
TIME,jtime
solve
*enddo

FINISH
```

Sadly ANSYS now crashes in the post-processor, so though I have results I cannot see them. More work required.

4. CONCLUSIONS SO FAR

This is obviously work in progress. From what has been done, it seems to me that forces of order 1000N are about right. The plan to bring this to closure is

- Complete the 10 second data run if possible. Extend the model to look at stops between the masses.
- Make some tests of candidate silica tips on silica. Static loading should be enough to see if there any damage.
- If the results suggest that a simple silica-tipped stop might be subject to damage then a two-stage stop design will be needed.

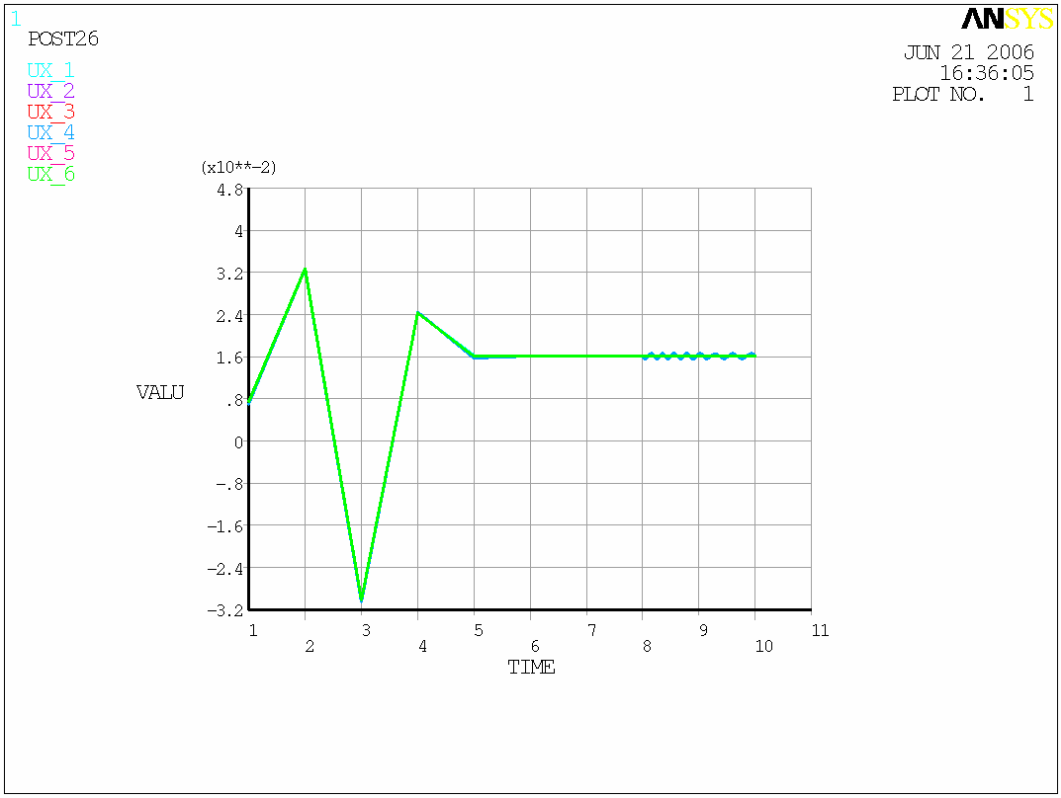
5. BACK TO ANALYSIS.

To avoid the crashing problem, tried saving results only at the end of each second up to the final two seconds:

```
OUTRES,ALL,LAST
kbc,0  !Ramped BC

NSUBST,5000
Autots,off
*do,jtime,1,8
TIME,jtime
solve
*enddo

OUTRES,ALL,1

NSUBST,5000
Autots,off
*do,jtime,9,10
TIME,jtime
solve
```

*enddo

Very odd results:



The displacements at the integer seconds should be (from Montana.txt)

1	0.007464859
2	0.03263021
3	-0.03021859
4	0.024381646
5	0.016145306
6	-0.036850036

Etc

So it seems that the input was not read correctly beyond 5 seconds.

Try with fewer substeps:
NSUBST,100

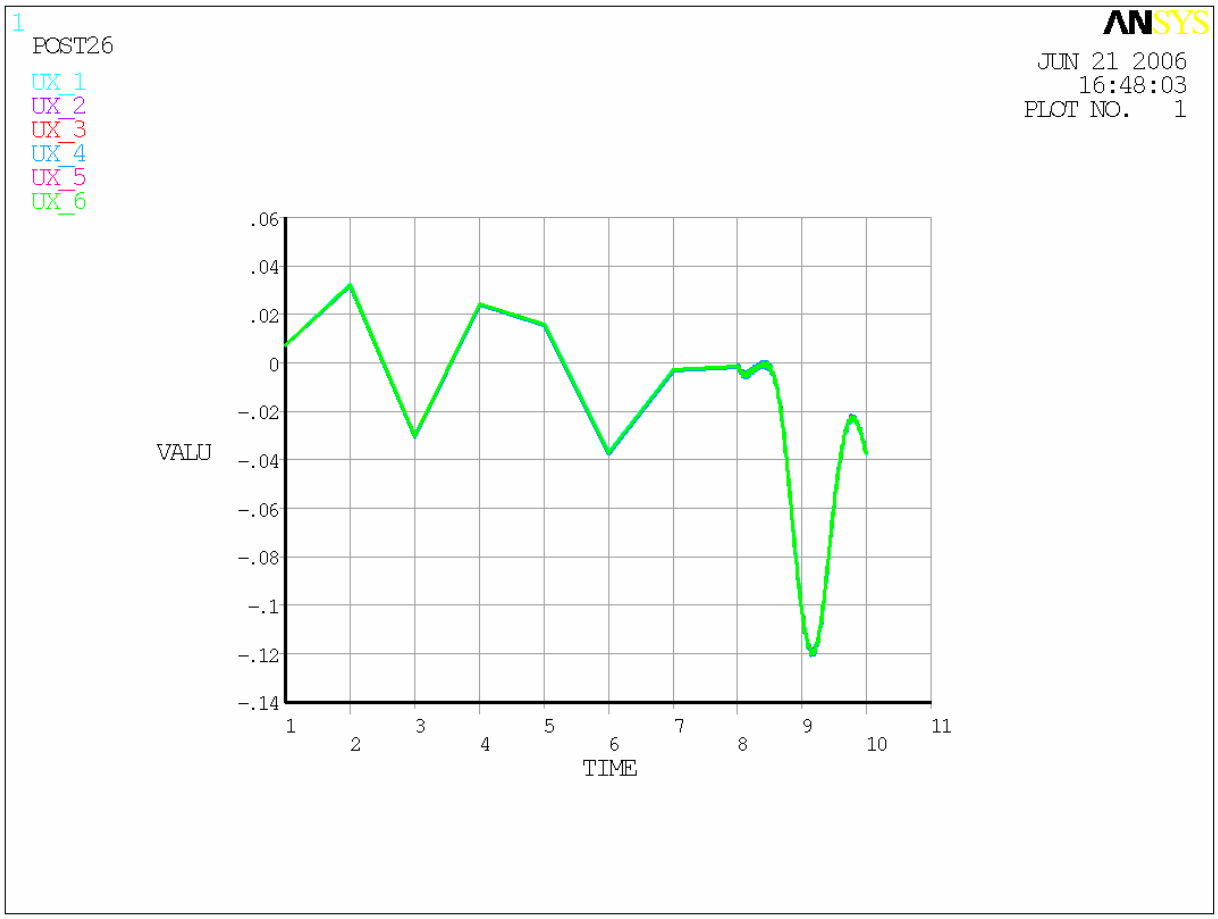
(INPUT FILE RECORDED IN APPENDIX 3)

Aha – here's the culprit:
*DIM, gdisp,TABLE, 101, 1, 1, TIME, ,
!*DIM, Par, Type, IMAX, JMAX, KMAX, Var1, Var2, Var3, CSYSID

The array needs to be bigger to cope with all the input values.
*DIM, gdisp,TABLE, 250, 1, 1, TIME, ,

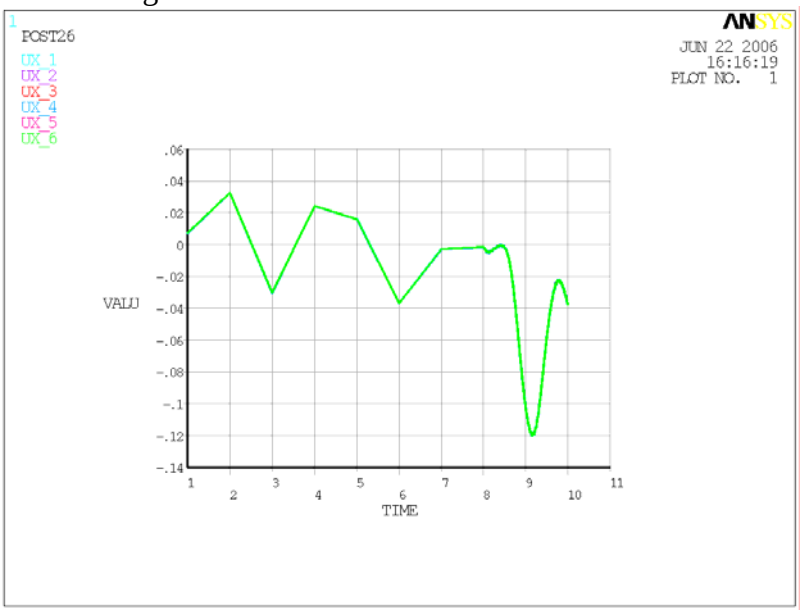
No – it has to be exactly the right size:
*DIM, gdisp,TABLE, 201, 1, 1, TIME, ,

Aha – this is more like it:

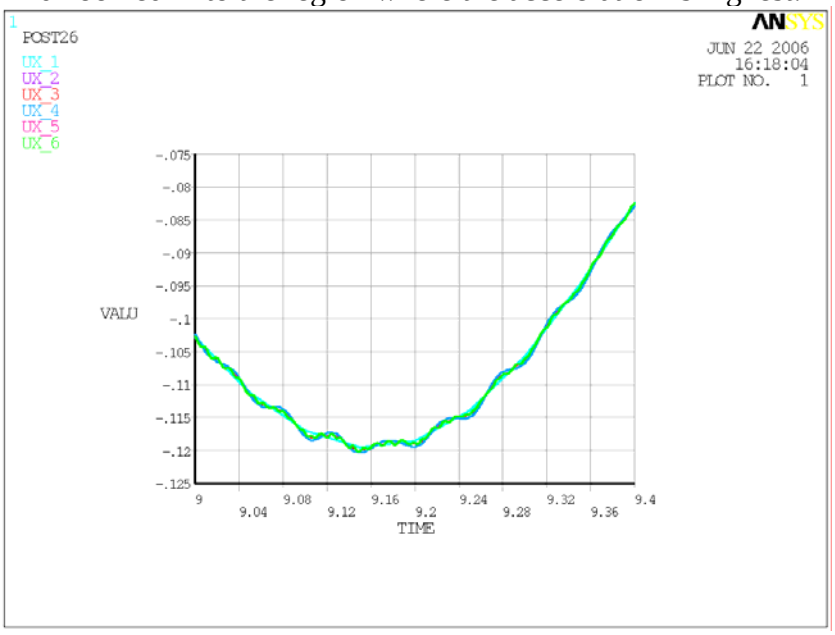


So now revert to 5000 substeps…

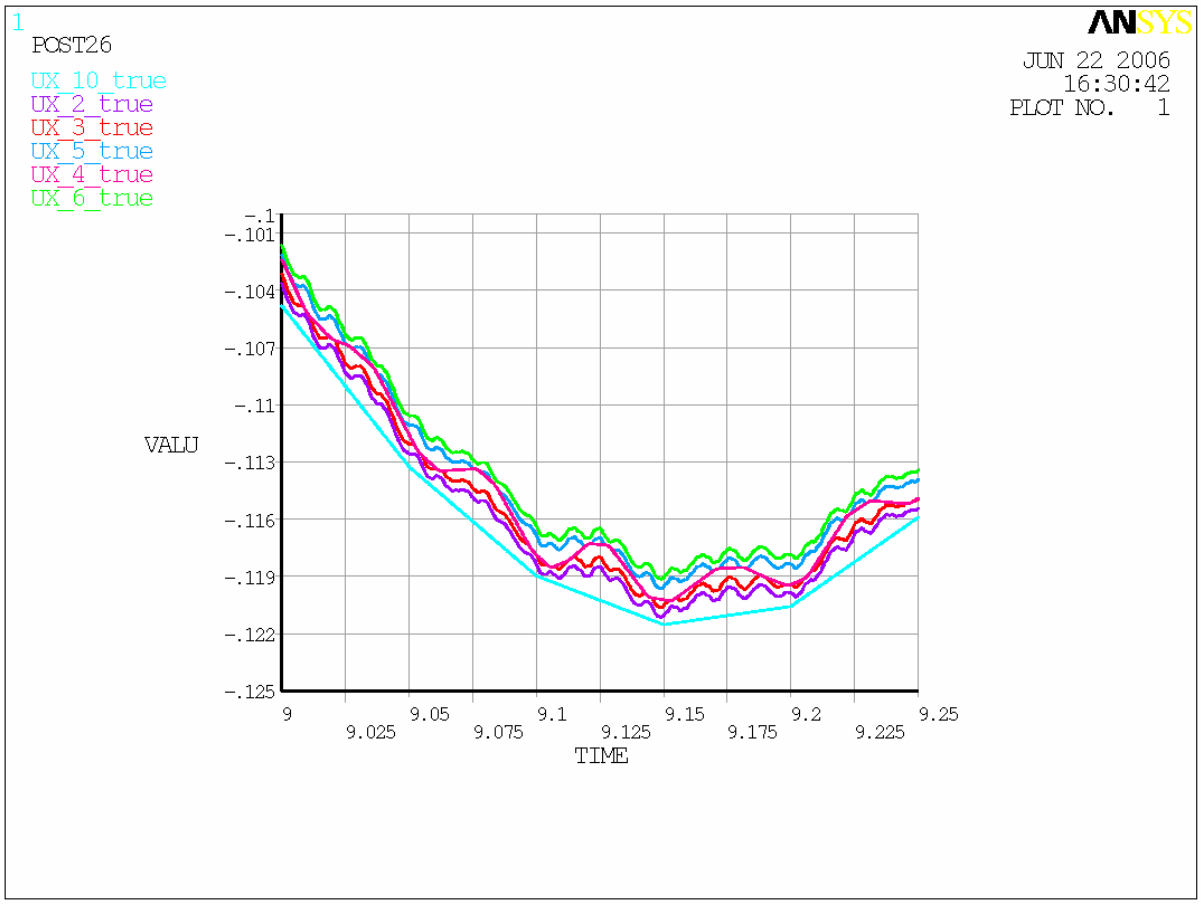
This looks good:



And zoomed in to the region where the acceleration is highest:

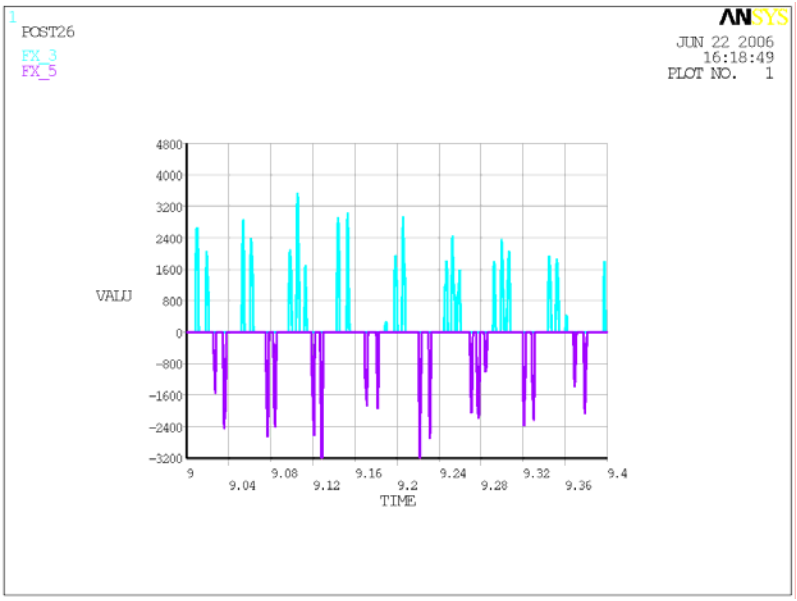


And the view with “true” positions, zoomed in:



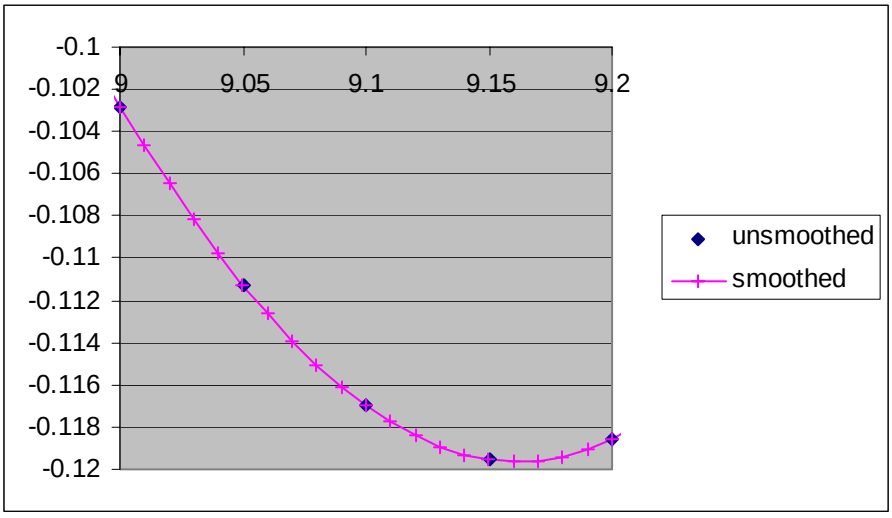
Hmm – how significant is it that the bounces seem to coincide with the (0.05 sec) discretization of the displacement data? To test this I would need to make a smoothed version of the displacement data with finer granularity.

The forces are high:



6. SMOOTHED INPUT DATA

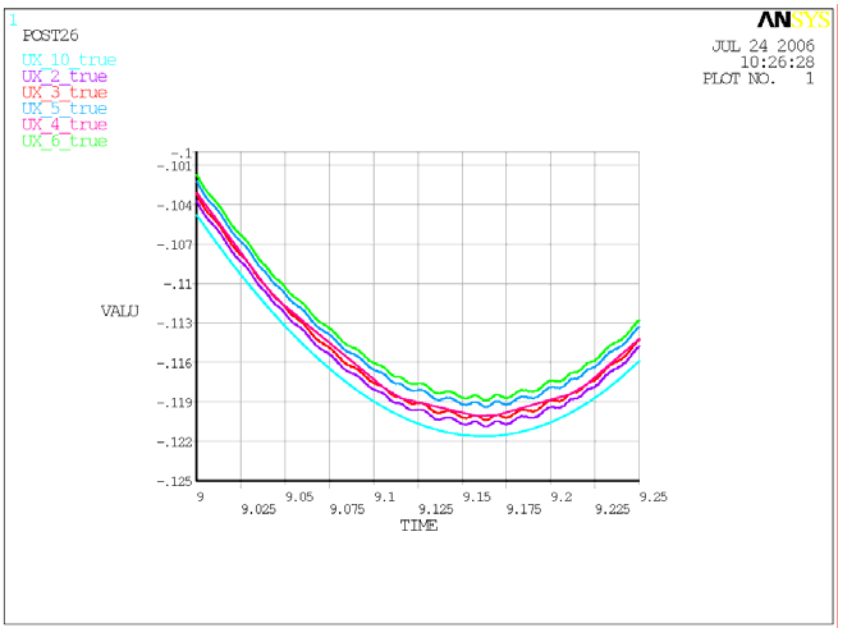
Using a spline routine I have smoothed the input data:



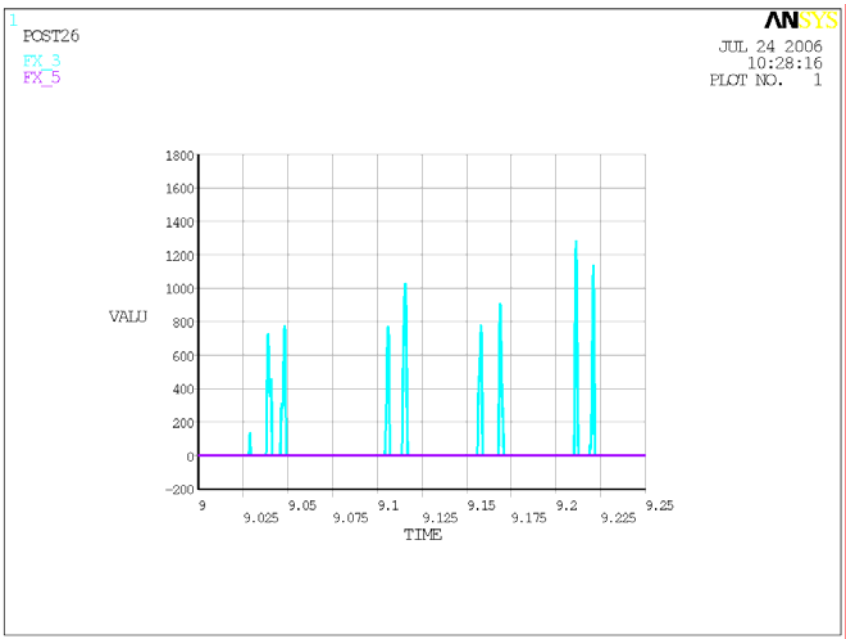
So try running as above but with this new dataset:

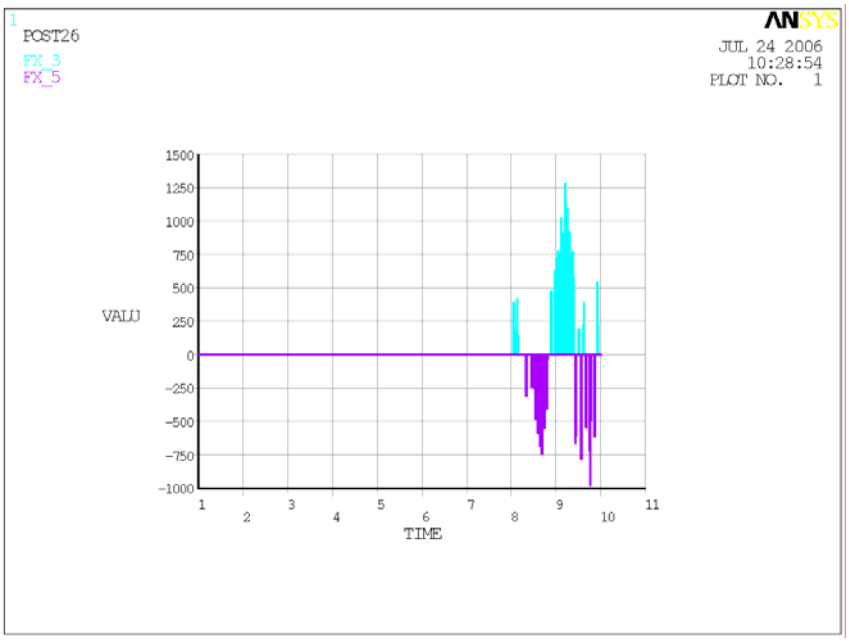
```
!*
*DIM,  gdisp,TABLE, 1001,  1,   1,    TIME, ,
!*DIM, Par,  Type, IMAX, JMAX, KMAX, Var1, Var2, Var3, CSYSID
!*
*TREAD,GDISP,'montana1001','txt' ,' ', 1,
!*TREAD, Par, Fname,     Ext  , --, NSKIP
D,10,UX,%GDISP%
```

Results look rather gentler,



And the forces are smaller:





7. CONCLUSION FROM WORK SO FAR

I conclude from all of the above that the tool for looking at forces in the stops for a given "ground" motion is now basically working.

To do next:

- Firm up on details of model (stiffness of beam etc)
- Give more realistic numbers for spring constant and damping in rubber
- Decide how to handle the stops between masses
- Decide how to get at "ground" motion for the Seismic platform.

Appendix 1 – simple beam model

```
FINISH  ! Make sure we are at BEGIN level
/CLEAR
*abbr,doit,doit
*abbr,jreplo,/replot
/config,nres,5000
/PREP7

! to make a simple beam model with given stiffness and masS.
!
! Parameters
!
length=1.7 !m
youngs=70E9 !Pa
stiffness=2E6 !N/m
beamI=stiffness*length**3/(3*youngs)
beamY=(beamI*12)**0.25
beamA=beamY**2
mass=50 !check with Tim
density=mass/(length*beamA)

/triad,lbot  !Move co-ord sys triad out of the way

! element types

ET,5,BEAM3
!*
KEYOPT,5,6,0
KEYOPT,5,9,0
KEYOPT,5,10,0
!*


!Real constants
!
R,10,beamA,beamI,beamY


!materials
MPTEMP,,,,,,,,
MPTEMP,1,0
MPDATA,EX,1,,youngs
MPDATA,PRXY,1,,0.3
MPDATA,DENS,1,,density

!nodes
N,10,-0.002,length
N,14,-.002,length/5
N,13,-.002,2*length/5
N,12,-.002,3*length/5
N,11,-.002,4*length/5
,2,-.002
!,3,-.0005
!,4,0
!,5,.0005
!,6,.002


!elements
type,5
real,10
mat,1
e,10,11
e,11,12
e,12,13
e,13,14
e,14,2


! fix bottom nodes in Y
!NSEL,S,NODE,,2,6
!fix all nodes in y
nsel,all
D,all,UY,0
!fix node 10 in rotation
d,10,rotz,0

NSEL,ALL

!temp - fix node 10 in x
d,10,ux,0

FINISH
/SOL
!*
ANTYPE,2
!*
!*
MSAVE,0
!*
MODOPT,LANB,10
EQSLV,SPAR
MXPAND,0, , ,0
LUMPM,0
PSTRES,0
!*
MODOPT,LANB,10,1,2000, ,OFF


/eof
```

8. APPENDIX 2 - MACROS WITH BEAM AND REST OF MODEL

```
FINISH  ! Make sure we are at BEGIN level
/CLEAR
*abbr,doit,doit
*abbr,jreplo,/replot
/config,nres,5000
/PREP7

! to make a simple beam model with given stiffness and masS.
!
! Parameters
!
length=1.7 !m
youngs=70E9 !Pa
stiffness=2E6 !N/m
beamI=stiffness*length**3/(3*youngs)
beamY=(beamI*12)**0.25
beamA=beamY**2
mass=50 !check with Tim
density=mass/(length*beamA)

MTM = 40 !Mass of test mass
KPad = 1.4E7 !stiffness of polymer pad
bPad = 2.4E3 !damping constant of pad

/triad,lbot  !Move co-ord sys triad out of the way

! element types

ET,2,CONTAC12   !Gap
!*
KEYOPT,2,1,0  !Friction type only valkid if mu>0
KEYOPT,2,2,0  !0=orientation angle based on theta real const
make theta = 0
KEYOPT,2,3,1  !0=no weak SPRing on open gap
KEYOPT,2,4,1  !1=use node location for initial gap
KEYOPT,2,7,0  !connected with optimised solution time
!*

ET,3,MASS21    !Point mass
!*
KEYOPT,3,1,0  !0=Real consts are mass and inertia
KEYOPT,3,2,0  !0=elem coord system parallel to global
KEYOPT,3,3,4  !2D mass no rotary intertia

ET,4,COMBIN14   !spring for the pads
!*
KEYOPT,4,1,1  !0=linear spring; 1 required for damping
KEYOPT,4,2,1  !1 = 1D longitudinal in UX
KEYOPT,4,3,0  !2 = 2D rather than 3D; Z=0 throughout
!*

ET,5,BEAM3
!*
KEYOPT,5,6,0
KEYOPT,5,9,0
KEYOPT,5,10,0
!*

!Real constants
!
! For the test mass
!R,n,mass
R,1,MTM
!

!for the gap
!
!R,n,theta,kn  ,intf,start,ks
R, 2,90   ,2e10,    ,

!For the polymer pad
!
!R,n,K,CV1,CV2
R,3,KPad,,bPad

!For the beam
!
R,10,beamA,beamI,beamY


!material for the beam
MPTEMP,,,,,,,,
MPTEMP,1,0
MPDATA,EX,1,,youngs
MPDATA,PRXY,1,,0.3
MPDATA,DENS,1,,density

!nodes
N,10,-0.002,length
N,14,-.002,length/5
N,13,-.002,2*length/5
N,12,-.002,3*length/5
N,11,-.002,4*length/5
,2,-.002
,3,-.0005
,4,0
,5,.0005
,6,.002


!elements

! Test mass
type,3
real,1
E,4

!gaps
type,2
real,2
E,5,4
E,4,3

!pads
type,4
real,3
E,5,6
E,3,2

!beam
type,5
real,10
mat,1
e,10,11
e,11,12
e,12,13
e,13,14
e,14,2

!fix all nodes in y
nsel,all
D,all,UY,0
!fix node 10 in rotation
d,10,rotz,0

!couple nodes 2 and 6 in UX:
CP,  1,    UX,  2,     6
!CP, NSET, Lab, NODE1, NODE2, NODE3, NODE4, NODE5,
NODE6, NODE7, NODE8, NODE9, NODE10

!*
*DIM,  gdisp,TABLE, 101,  1,   1,    TIME, ,
```

```
!*DIM, Par,  Type, IMAX, JMAX, KMAX, Var1, Var2, Var3,
CSYSID
!*
*TREAD,GDISP,'timehist','txt' ,' ', 1,
!*TREAD, Par, Fname,     Ext  , --, NSKIP
D,10,UX,%GDISP%

FINISH
!/eof

/solve
solcontrol,0    !used in VM81, no idea why, if ommitted solution
fails.
ANTYPE,4  !transient
!*
TRNOPT,FULL   !Full analysis - no shortcuts
LUMPM,0
!*

OUTRES,ALL,1
kbc,0  !Ramped BC

NSUBST,5000
Autots,off
TIME,1
solve

FINISH
/POST26
numvar,200

!*
NSOL,8,10,U,X,UX_1
STORE,MERGE
!*
NSOL,2,2,U,X,UX_2
STORE,MERGE
!*
NSOL,3,3,U,X,UX_3
STORE,MERGE
!*
NSOL,4,4,U,X,UX_4
STORE,MERGE
!*
NSOL,5,5,U,X,UX_5
STORE,MERGE
!*
NSOL,6,6,U,X,UX_6
STORE,MERGE


FORCE,TOTAL
ESOL,9,5,3 ,F,X,FX_3
STORE,MERGE
FORCE,TOTAL
!*
!*
ESOL,10,4,5 ,F,X,FX_5
STORE,MERGE



XVAR,1
PLVAR,8,2,3,4,5,6

!
! Name: UX_10_true
! ID:  13
! Function: nsol(10 ,U,X)-.002
NSOL,200,10,U,X
FILLDATA,198,,,,-.002,0
REALVAR,198,198
PROD,197,198,194
ADD,13,200,197,,UX_10_true
!
! Name: UX_2_true
! ID:  14
! Function: nsol(2 ,U,X)-.001
NSOL,200,2,U,X
FILLDATA,198,,,,-.001,0
REALVAR,198,198
PROD,197,198,194
ADD,14,200,197,,UX_2_true
!
! Name: UX_3_true
! ID:  15
! Function: nsol(3 ,U,X)-.0005
NSOL,200,3,U,X
FILLDATA,198,,,,-.0005,0
REALVAR,198,198
PROD,197,198,194
ADD,15,200,197,,UX_3_true
!
! Name: UX_4_true
! ID:  17
! Function: nsol(4 ,U,X)
NSOL,17,4,U,X,UX_4_true
!
! Name: UX_5_true
! ID:  16
! Function: nsol(5 ,U,X)+.0005
NSOL,200,5,U,X
FILLDATA,198,,,,.0005,0
REALVAR,198,198
ADD,16,200,198,,UX_5_true
!
! Name: UX_6_true
! ID:  18
! Function: nsol(6 ,U,X)+.001
NSOL,200,6,U,X
FILLDATA,198,,,,.001,0
REALVAR,198,198
ADD,18,200,198,,UX_6_true
!



:END
```

9. APPENDIX 3 – MACRO WITH INPUT FILE, SEE SECTION 5.

```
FINISH  ! Make sure we are at BEGIN level
/CLEAR
*abbr,doit,doit
*abbr,jreplo,/replot
/config,nres,51000
/PREP7

! to make a simple beam model with given stiffness and masS.
!
! Parameters
!
length=1.7 !m
youngs=70E9 !Pa
stiffness=2E6 !N/m
beamI=stiffness*length**3/(3*youngs)
beamY=(beamI*12)**0.25
beamA=beamY**2
mass=50 !check with Tim
density=mass/(length*beamA)

MTM = 40 !Mass of test mass
KPad = 1.4E7 !stiffness of polymer pad
bPad = 2.4E3 !damping constant of pad

/triad,lbot  !Move co-ord sys triad out of the way

! element types

ET,2,CONTAC12   !Gap
!*
KEYOPT,2,1,0  !Friction type only valkid if mu>0
KEYOPT,2,2,0  !0=orientation angle based on theta real const
make theta = 0
KEYOPT,2,3,1  !0=no weak SPRing on open gap
KEYOPT,2,4,1  !1=use node location for initial gap
KEYOPT,2,7,0  !connected with optimised solution time
!*

ET,3,MASS21    !Point mass
!*
KEYOPT,3,1,0  !0=Real consts are mass and inertia
KEYOPT,3,2,0  !0=elem coord system parallel to global
KEYOPT,3,3,4  !2D mass no rotary intertia

ET,4,COMBIN14   !spring for the pads
!*
KEYOPT,4,1,1  !0=linear spring; 1 required for damping
KEYOPT,4,2,1  !1 = 1D longitudinal in UX
KEYOPT,4,3,0  !2 = 2D rather than 3D; Z=0 throughout
!*

ET,5,BEAM3
!*
KEYOPT,5,6,0
KEYOPT,5,9,0
KEYOPT,5,10,0
!*

!Real constants
!
! For the test mass
!R,n,mass
R,1,MTM
!

!for the gap
!
!R,n,theta,kn  ,intf,start,ks
R, 2,90   ,2e10,    ,

!For the polymer pad
!
!R,n,K,CV1,CV2
R,3,KPad,,bPad

!For the beam
!
R,10,beamA,beamI,beamY


!material for the beam
MPTEMP,,,,,,,,
MPTEMP,1,0
MPDATA,EX,1,,youngs
MPDATA,PRXY,1,,0.3
MPDATA,DENS,1,,density

!nodes
N,10,-0.002,length
N,14,-.002,length/5
N,13,-.002,2*length/5
N,12,-.002,3*length/5
N,11,-.002,4*length/5
,2,-.002
,3,-.0005
,4,0
,5,.0005
,6,.002


!elements

! Test mass
type,3
real,1
E,4

!gaps
type,2
real,2
E,5,4
E,4,3

!pads
type,4
real,3
E,5,6
E,3,2

!beam
type,5
real,10
mat,1
e,10,11
e,11,12
e,12,13
e,13,14
e,14,2

!fix all nodes in y
nsel,all
D,all,UY,0
!fix node 10 in rotation
d,10,rotz,0

!couple nodes 2 and 6 in UX:
CP,  1,   UX,  2,    6
!CP, NSET, Lab, NODE1, NODE2, NODE3, NODE4, NODE5,
NODE6, NODE7, NODE8, NODE9, NODE10

!*
*DIM,  gdisp,TABLE, 101,  1,   1,    TIME, ,
!*DIM, Par,  Type, IMAX, JMAX, KMAX, Var1, Var2, Var3,
CSYSID
!*
```

```
*TREAD,GDISP,'montana','txt' ,' ', 1,
!*TREAD, Par, Fname,     Ext  , --, NSKIP
D,10,UX,%GDISP%

FINISH
!/eof

/solve
solcontrol,0    !used in VM81, no idea why, if ommitted solution
fails.
ANTYPE,4  !transient
!*
TRNOPT,FULL   !Full analysis - no shortcuts
LUMPM,0
!*

OUTRES,ALL,LAST
kbc,0  !Ramped BC

NSUBST,100
Autots,off
*do,jtime,1,8
TIME,jtime
solve
*enddo

OUTRES,ALL,1

NSUBST,100
Autots,off
*do,jtime,9,10
TIME,jtime
solve
*enddo


FINISH
/POST26
numvar,200

!*
NSOL,8,10,U,X,UX_1
STORE,MERGE
!*
NSOL,2,2,U,X,UX_2
STORE,MERGE
!*
NSOL,3,3,U,X,UX_3
STORE,MERGE
!*
NSOL,4,4,U,X,UX_4
STORE,MERGE
!*
NSOL,5,5,U,X,UX_5
STORE,MERGE
!*
NSOL,6,6,U,X,UX_6
STORE,MERGE


FORCE,TOTAL
ESOL,9,5,3 ,F,X,FX_3
STORE,MERGE
FORCE,TOTAL
!*
!*
ESOL,10,4,5 ,F,X,FX_5
STORE,MERGE



XVAR,1
PLVAR,8,2,3,4,5,6

!
! Name: UX_10_true
! ID:  13
! Function: nsol(10 ,U,X)-.002
NSOL,200,10,U,X
FILLDATA,198,,,,-.002,0
REALVAR,198,198
PROD,197,198,194
ADD,13,200,197,,UX_10_true
!
!
! Name: UX_2_true
! ID:  14
! Function: nsol(2 ,U,X)-.001
NSOL,200,2,U,X
FILLDATA,198,,,,-.001,0
REALVAR,198,198
PROD,197,198,194
ADD,14,200,197,,UX_2_true
!
! Name: UX_3_true
! ID:  15
! Function: nsol(3 ,U,X)-.0005
NSOL,200,3,U,X
FILLDATA,198,,,,-.0005,0
REALVAR,198,198
PROD,197,198,194
ADD,15,200,197,,UX_3_true
!
! Name: UX_4_true
! ID:  17
! Function: nsol(4 ,U,X)
NSOL,17,4,U,X,UX_4_true
!
! Name: UX_5_true
! ID:  16
! Function: nsol(5 ,U,X)+.0005
NSOL,200,5,U,X
FILLDATA,198,,,,.0005,0
REALVAR,198,198
ADD,16,200,198,,UX_5_true
!
! Name: UX_6_true
! ID:  18
! Function: nsol(6 ,U,X)+.001
NSOL,200,6,U,X
FILLDATA,198,,,,.001,0
REALVAR,198,198
ADD,18,200,198,,UX_6_true
!



:END
```