RANDOM ANALYSIS USING MSC/NASTRAN ISHELL MODULE

Mohan Barbela
Senior Technical Representative
The MacNeal-Schwendler Corporation

Abstract

MSC/NASTRAN version 70 has a new DMAP module called ISHELL that allows users to execute an external program from within MSC/NASTRAN. This module is very powerful and can be used to perform various tasks; for example, sorting, margin of safety calculation, and any data block manipulation using an external user written program. This paper briefly describes the ISHELL DMAP module and the procedure to perform random analysis using MSC/NASTRAN ISHELL DMAP module and an external program.

Introduction

Random analysis in MSC/NASTRAN is treated as a data reduction procedure that is applied to frequency response analysis. The output request for a random response can only be made through the XYOUT module. This creates the burden on a user to prepare XYPLOT or XYPRINT entries for each degree of freedom for nodal responses, and for each stress/force component for element responses. This means, even for a small size model, one has to prepare a large number of XYPLOT/XYPRINT entries. The FORTRAN program, usr_random, is written to overcome these limitations. The execution of the program usr_random is invoked through ISHELL DMAP module. The program calculates the random responses of grid displacement, velocity, acceleration of all grids and element stresses/forces for all elements requested in case control data section. The program also creates a neutral result file containing RMS responses that can be read into MSC/PATRAN to plot RMS responses.

ISHELL DMAP Module

MSC/NASTRAN Version 70 introduced a new DMAP module, ISHELL, that allows the users to suspend MSC/NASTRAN, execute an external program, then return to MSC/NASTRAN and continue the solution sequence. The external program can be a shell script or an executable program written in FORTRAN or C. MSC/NASTRAN remains in a 'wait' state until the external process is completed. MSC/NASTRAN's ability to spawn external processes is extremely useful when used in conjunction with its INPUTT2, OUTPUT2, INPUTT4, OUTPUT4 modules. The ISHELL module has been explained in detail in reference 1. The calling syntax and the brief usage is summarized below.

Calling Syntax:

```
ISHELL //Prog_Name/IReturn_code/
        No_of_Integer/No_of_Real/No_of_Complex/No_of_Char/No_of_Unit/
        Integer1/Integer2/Integer3/Integer4/
        Real1/Real2/Real3/Real4/
        Complex1/Complex2/Complex3/Complex4/
        Char1/Char2/Char3/Char4/
        Unit1/Unit2/Unit3/Unit4  $
```

The Parameters are:

Prog_Name : Input, BCD : 8 Character Max : Script or Program Name to be executed.
Pos. 1 : IReturn_Code : Output, Integer : Return Code -0 Successful
Pos. 2 to 6 : No of integer, real, complex, character string, and FORTRAN logical units.
Pos. 7 to 10 : Integer Parameters.

Pos. 11 to 14 : Real Parameters.
Pos. 15 to 22 : Complex Parameters.
Pos. 23 to 26 : String Parameters.
Pos. 27 to 30 : FORTRAN logical Unit Numbers.
Pos. 31 to 34 : Parameters 31 to 34 are not part of ISHELL calling parameters, however, they contain the physical file names associated with FORTRAN logical unit numbers given in position 27 to 30. These names are assigned based on a ASSIGN statement in File Management Section.

Solution Approach

Since random analysis is nothing but a post processing of a frequency response analysis, the approach for this program is simple and straightforward.

- Use MSC/NASTRAN to perform frequency response analysis and to create results OUTPUT2 file.
- Spawn the process from within the MSC/NASTRAN using the ISHELL DMAP module.
- Calculate random response using frequency response results and user input file that defines the random input profile.
- Resume MSC/NASTRAN run.

The external program, usr_random, uses the following approach to calculate random responses.

The transfer function theorem states that if $H_{ja}(f)$ is the frequency response of any physical variable U_j, due an excitation source $Q_a(f)$, then

$$U_j(f) = H_{ja}(f) \cdot Q_a(f)$$

and the power spectral density of the response $S_j(f)$ due to power spectral density of the source, $S_a(f)$ is given by

$$S_j(f) = |H_{ja}(f)|^2 \cdot S_a(f)$$

If the sources are statistically independent, then the power spectral density of the total response is equal to sum of the power spectral densities of the responses due to individual sources. Thus

$$S_j(f) = \sum S_{ja}(f) = \sum |H_{ja}(f)|^2 \cdot S_a(f)$$

If the sources are statistically correlated, the degree of correlation can be given by a cross-spectral density, S_{ab}, and the auto-spectral density of the response can be evaluated from

$$S_j = \sum\sum H_{ja}(f) \, . \, H^*_{jb}(f) . \, S_{ab}(f)$$

where H^*_{jb} is the complex conjugate of H_{jb}.

The RMS response is calculated using

$$S_{rms} = \left[\int S_j(f) \, df \right]^{1/2}$$

The number of zero crossings N_0 is calculated using

$$N_0 = \left[\int (2.\pi.f)^2 * S_j(f) \, df \, / \int S_j(f) \, df \right]^{1/2}$$

The rms response and number of zero crossings are computed using log-log interpolation and integration.

Procedure

1. Create separate random input file which will have only random related input e.g. RANDPS and TABRND1 tables. Refer to the MSC/NASTRAN Quick Reference Guide for more detail.

2. Run MSC/NASTRAN to calculate frequency response (Sol 108 or Sol 111). Request the output for element stresses/forces and grid accelerations/velocities/displacements using SORT2 option. Write the frequency response using OUTPUT2 DMAP statement.

3. Call ISHELL DMAP module from within MSC/NASTRAN. The ISHELL module will pass variables (OUTPUT2 file and random input file created in step 1) to an external c-shell(csh) script USR_RAND.

4. The USR_RAND script file parses the argument list supplied by ISHELL DMAP module and executes the external FORTRAN program usr_random.

5. The program usr_random reads the frequency response from OUTPUT2 file and user defined random input environment and calculates the response under random excitation.

Important features of program usr_random:

- Calculates random response of all output quantities requested using simple standard MSC/NASTRAN output request card, such as stress = all, accel = 99, force = all.

- The random input profile is interpolated using logarithmic interpolation.
- The RMS response and number of positive crossing are calculated using logarithmic integration.
- Three separate output files are created. These files contain:
    - Full Output : Random response at all output frequencies, RMS response and number of positive crossing.
    - Condensed Output : Only RMS response and number of positive crossing.
    - Neutral Result File : RMS response results file that can be read into MSC/PATRAN to create contour plots of RMS accelerations, velocities, and criteria plots of forces and stresses.

Examples

This example shows how to run user written program from MSC/NASTRAN V70. Here the random response of the entire structure is calculated without going through MSC/NASTRAN random module. This example problem shows the usefulness and the power of the ISHELL module. This example shows the procedure for:

1. Saving the harmonic responses on OUTPUT2 unformatted file.
2. Calling the ISHELL module from within MSC/NASTRAN. This will invoke C Shell (csh) script USR_RAND.
3. The USR_RAND script will parse the arguments passed by ISHELL module and get the file names based on their assigned unit numbers in arguments 27 and 28 and get the file names from arguments 31 and 32. The external program usr_random requires two input file names:
    - MSC/NASTRAN OUTPUT2 file (see PARAM,randop2) and the
    - User file that contains random input profile(see PARAM,randinp).
4. The USR_RAND script file will invoke execution of external program usr_random with two file names as input arguments.
5. The external program will read the MSC/NASTRAN OUTPUT2 file and user input file and calculate the random responses of all the grids and elements found in OUTPUT2 file.
6. The program usr_random will create the .rms file that contains rms response and also will create neutral result files .els, .elf, .dis, .acc, and .vel. These files were read in MSC/PATRAN as neutral result files to plot rms responses.

The sample MSC/NASTRAN data deck setup(test.dat), usr_random input file (random_test.inp) and the script file (USR_RAND) are shown in following section.

Figure 1 and 2 shows the random RMS responses under base random excitation. Figure 3 shows the random RMS responses under acoustic loading.

Summary

This paper shows the usefulness and power of the MSC/NASTRAN ISHELL DMAP module. The concept presented here can be extended to perform maximum/minimum search, margin of safety calculations, etc.

References

1. MSC/NASTRAN *Version 70 Release Guide*, The MacNeal-Schwendler Corporation, Los Angeles, CA, July 1997.
2. Bendat, J. S., MSC/NASTRAN Random Vibration Analysis and Applications, The MacNeal-Schwendler Corporation, Los Angeles, CA, July 1987.
3. "Random Analysis of a Simple Structure", Application Note, MSC/NASTRAN *Application Manual*, The MacNeal-Schwendler Corporation, Los Angeles, CA, April 1982.
4. MSC/NASTRAN *Handbook for Dynamic Analysis, Version 63,* The MacNeal-Schwendler Corporation, Los Angeles, CA, June 1983.
5. MSC/NASTRAN *Basic Dynamic Analysis User's Guide Version 68,* The MacNeal-Schwendler Corporation, Los Angeles, CA, December 1993.

MSC/NASTRAN Input File with DMAP(test.dat).

```
$ First assign statement is to save Harmonic Response(Transfer Function).
$ Second assign statement will be used only by external program.  In this case
$ it is used for passing the input file name to external program.
$ The file name can be also passed using param, however the file name
$ will be restricted to 8 characters and must be in upper case.
$
assign output2='test.f17',unit=17,form=unformatted,  status=unknown
assign userfile = 'random_test.inp', unit=16, form = formatted, status = old
$
INIT MASTER(S)
ID MSC, ABU
TIME 100
DIAG 5,6,8,56
SOL 111 $ Modal Frequency Response
$
echooff
compile sedrcvr souin=mscsou,nolist,noref
alter 2
$
$ Param,random,n          n = 0  do not execute usr_random,
$                         n = 1  execute usr_random
type parm,,i,y,random=0
type parm,,i,y,irtn        $  Return code. 0 Successful.
$
$ Param,randop2,nunit   nunit = unit number for output2.  Dafault 18.
$
type parm,,i,y,randop2 =18
$
$ Param,randinp,nuint    nunit = unit number for user input file. Default 19.
$
type parm,,i,y,randinp =19
$
$  See PARAM,randop2 and randinp for reassignment.
$  Save the Harmonic Response due to unit 'g' input on
$  unit randop2.
$
MALTER '(OUGV2, OES2, OEF2, ETC.)'(,-1)
if (random > 0) then
   output2 ougv2,oef2,oes2,, // -1/randop2 $
   output2 ,,,,//-9/randop2 $
   message //'   '/ $
   message //'  Running usr_random......  '/ $
   message //'   '/ $
$
$ Following ishell parameter pass the unit number of the
$ two file in variable number 27 and 28.  Physical file
$ names associated with will be
$
ishell //'USR_RAND'/irtn/ 0/ 0/ 0/ 1/ 1/
$                          1  2  3  4  5  6
           / / / / / / / / / / /
```

```
$     variables 7 thru 18
           /  /  /  /randop2/randinp/  /  / $
$          23 24 25  26      27     28 29 30
   if (irtn <> 0) then $
      message //'   '/ $
      message //'USR_RANDOM failed with return code '/ irtn $
      message //'   '/ $
   else
      message //'   '/ $
      message //'USR_RANDOM ran successfully.'/ $
      message //'   '/ $
   endif  $  endif for irtn
endif     $  endif for random
echoon
CEND
$
TITLE = CANTILEVERED BEAM MADE OF PLATES
$
$ Output request : Random response will be calculated for
$ all output quantities requested here.
$
$ Note : Plot option to suppress print output.
$      SORT2 is required for usr_random
$
acceleration(sort2,real,plot) = all
displacement(sort2,real,plot) = all
force(sort2,real,plot) = all
stress(sort2,real,plot) = all
echo = sort
spc = 77
SDAMP = 11102
FREQ = 604
METHOD = 219

SUBCASE  11101 $ Modal Frequency Response w/Random
   subtitle = Excitation in Z Direction
   DLOAD = 11103
SUBCASE  22201 $ Modal Frequency Response w/Random
   subtitle = Excitation in Y Direction
   DLOAD = 22203

BEGIN BULK
$            \201 202  203  204  205  206  207  208  209  210  211
$  Y         \*----*----*----*----*----*----*----*----*----*----*
$  ^         \|    |    |    |    |    |    |    |    |    |    |
$  |     99 * | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
$  |         \|    |    |    |    |    |    |    |    |    |    |
$  +--->X    \*----*----*----*----*----*----*----*----*----*----*
$            \101 102  103  104  105  106  107  108  109  110  111
$
$
param,wtmass,.002588
param,autospc,yes
param,grdpnt,0
```

```
$
$ Following param card controls the execution of usr_random.
$
$ Param,random,0(default).  0 Do not execute usr_random
$                           1 Execute usr_random through ISHELL
$ Param,randop2,nunit (default=18) unit no for output2.
$ Param,randinp,nunit (default=19) unit no for user input file.
$
param,random,1
param,randop2,17
param,randinp,16
$
grid,99,,0.0,0.5,0.0
grid    101             0.      0.      0.
grid    102             1.      0.      0.
=       *1      =       *1.     ==
=8
grid    201             0.      1.      0.
grid    202             1.      1.      0.
=       *1      =       *1.     ==
=8
cquad4  11      1       101     102     202     201
=       *1      =       *1      *1      *1      *1
=8
cbar,201,11,101,102,0.0,1.0
=,*1,=,*1,*1,=,=
=8
cbeam,301,31,201,202,0.0,1.0
=,*1,=,*1,*1,=,=
=8
pshell  1       1       .1      1
pbeam,31,12,.01,.011,.022,,.033,.1
+,.2,.2,.2,-.2,-.2,.2,-.2,-.2
pbar,11,12,.05,.05,.05,0.05,1.0
+,.2,.2,.2,-.2,-.2,.2,-.2,-.2
mat1,12,29.e6,,.3,.25
mat1    1       10.e6           .3      0.1     1.e-6   0.
eigrl,219,-.1,2000.
tabdmp1 11102   crit
        0.      .04     99999.  0.04    endt
rload1  11103   11105                   11106
darea,11105,99,3,1.e8
rload1  22203   22205                   11106
darea,22205,99,2,1.e8
$  Unit G loading
tabled1 11106
        0.      1.0     10.     1.0     2000.   1.0     ENDT
$ Output frequencies
freq1,604,100.,50.,30
freq4,604,10.,2000.,.20,4
rbar,77,99,101,123456
rbar,78,99,201,123456
conm2,999,99,,1.e8
```

```
suport,99,123
spc1,77,456,99
enddata
```

User input file (random_test.inp).

```
$
randps,101,11101,11101,1.0,0.,111
randps,101,11101,22201,0.6,0.5,211
$
tabrnd1 111
+,0.0,0.2,2000.,0.2,endt
$
tabrnd1,211
+,   0.0,  0.1, 100.0, 0.1, 200., 0.1, 300.0, 1.0, +
+, 400.0,  0.1, 500.0, 0.1, 600., 0.1, 700.0, 1.0, +abc
+abc,900.0,0.1 2000.0, 0.1, endt
$
```

Script File: USR_RAND

```
#!/bin/csh -f
echo ' '
echo '     usr_random begin'
#    For c shell, ishell pass all variables separated by space or comma as one
#    argument.
set xvar = ($argv[1])
#
#    Get file name from variable number 31 and 32
#
set nas_file   = $xvar[31]
set input_file = $xvar[32]
#
#    Remove  ‘  from file name  i.e. ‘random_test.inp’  -->  random_test.inp
#
set input_file = ‘eval echo $input_file‘
set nas_file   = ‘eval echo $nas_file‘
#
#   ‘  can also be removed using following statements
# set input_file = `echo $input_file | sed "s/\'//g"`
# set nas_file   = `echo $nas_file   | sed "s/\'//g"`
#
#    Run external program usr_random
#
usr_random  -n $nas_file  -d  $input_file
#
echo '     usr_random end'
echo ' '
exit
```

Fig : 1. Sample RMS Stress(X-comp) Response Plots : Test Case.

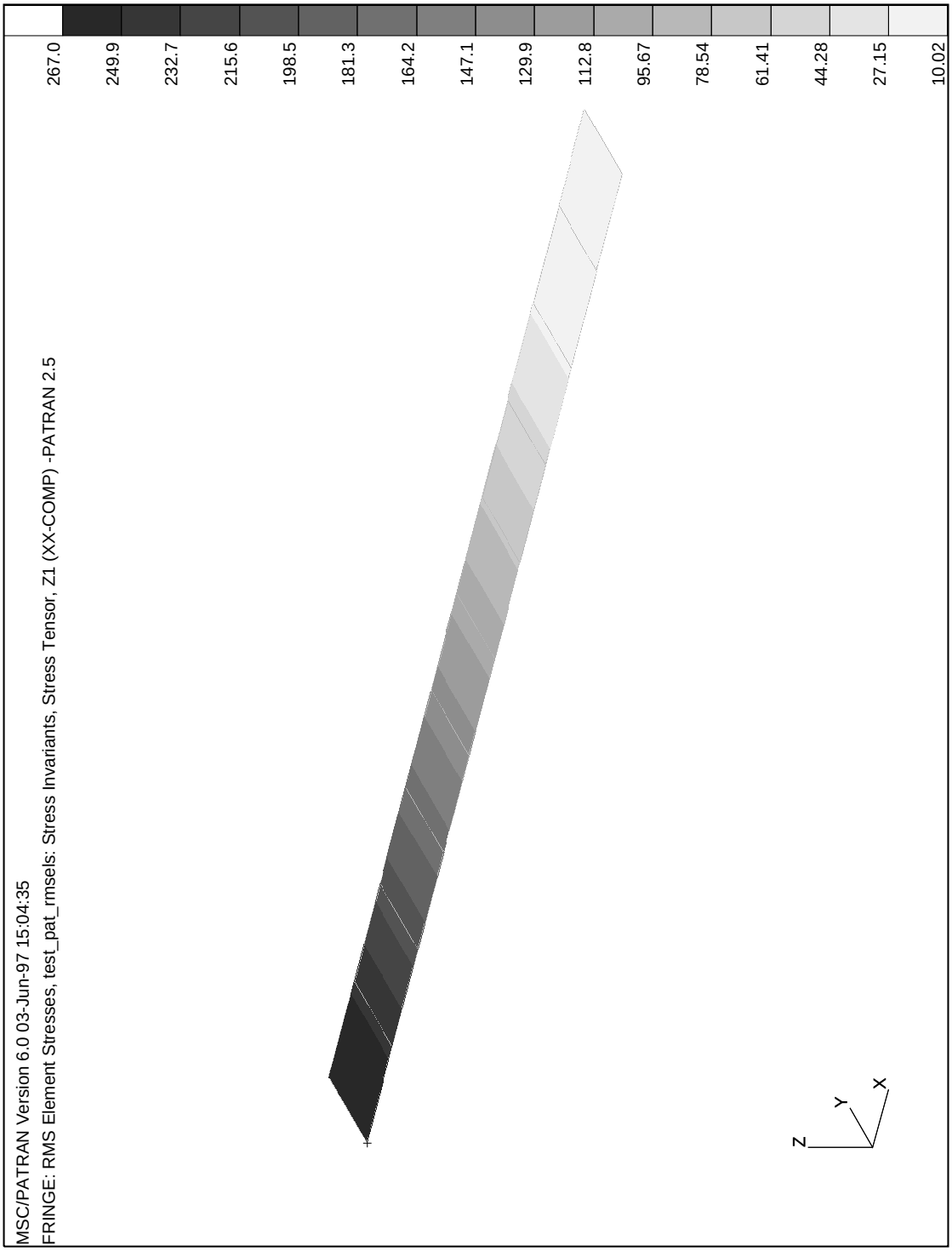

Fig : 2. Sample RMS Acceleration(y) Response Plots : Electronic Box.

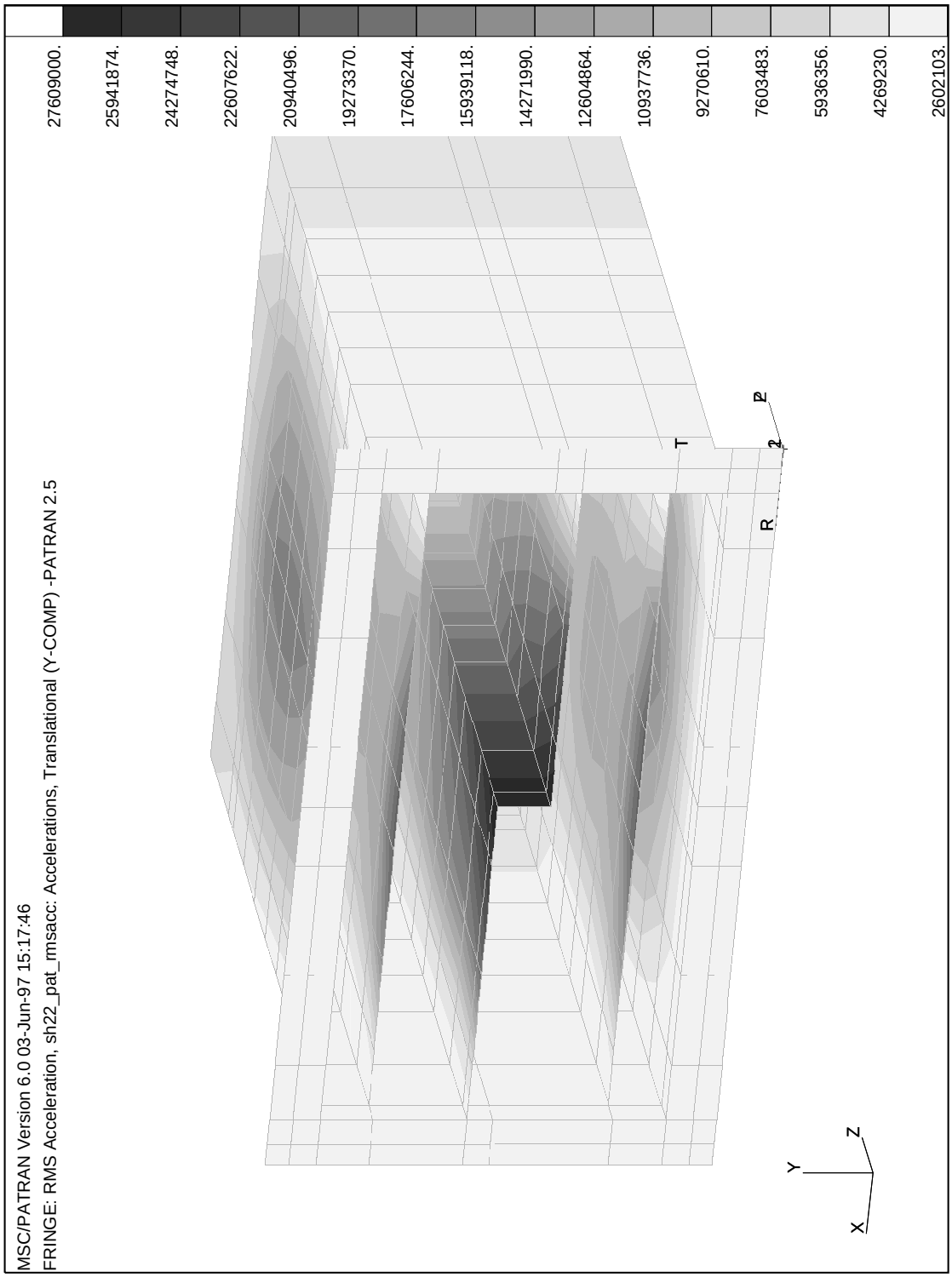

Fig : 3. Sample RMS Stress(xx) Response Plots under acoustic loading.