An Example Using the ISHELL Module to Print Modal Effective Mass.

by

Ted Rose

Manager, MSC/NASTRAN Training and Support

The MacNeal-Schwendler Corporation

Presented at the 1999 MSC U.K. Conference

Abstract:

It often seems to be several years after the introduction of a new feature before it gains recognition among the user community. This paper is intended to provide an introduction to the ISHELL module, which puts MSC/NASTRAN into a wait state while executing an external program. This module was introduced in V70. The paper also demonstrates using the ISHELL module with a DMAP alter which calculates the modal effective mass for a model and writes it to a file for re-formatting by an eternal program using ISHELL.

Introduction:

DMAP (Direct Matrix Abstraction Programming) is a programming language included with MSC/NASTRAN. By using this language, users can modify existing solutions and even create their own solutions inside the program. In the past, use of DMAP was limited to using existing Modules in MSC/NASTRAN, and creating formatted printout required "herculean" efforts.

In version 70, a new DMAP module (ISHELL) was introduced[1]. When this module is executed, MSC/NASTRAN goes into a wait state while an external program is run (spawned). When the external program is completed, MSC/NASTRAN continues. Because of this, it is possible to execute user-written programs during MSC/NASTRAN processing.

When MSC/NASTRAN goes into the wait state, it sends a command to the computer. Elements in this command are file names for selected files to be passed and parameters from MSC?NASTRAN which might be used by the external program.

Overview:

In versions prior to V70 of MSC/NASTRAN, there was only one way to execute user-supplied routines and programs as a part of MSC/.NASTRAN execution. This was to use the user-modifiable option. In this option, MSC/NASTRAN is shipped as normal with the addition of "object" routines. These object routines can be linked with user-supplied object code to create a customized version of the program. To use this feature requires extensive programming knowledge and a familiarity with MSC/NASTRAN and how to program for it. Although the user-modifiable option is a very powerful tool, it has not gained popularity with our users.

Now there is a feature included in the DMAP option which allows you to suspend MSC/NASTRAN execution while a user-supplied program is executed. This feature is the new ISHELL module and its companion module FORTIO.

FORTIO is a module which releases files from the current MSC/NASTRAN run to make them available for the external program. It can also re-attach files after the external program is completed. By using this, you may use OUTPUT2, OUTPUT4, or any other output files in the external program. Beware – if a file is released and re-attached to a run, MSC/NASTRAN will re-attach it positioned at the start of the file. If you release the '.f06' file and then re-attach it, the program will start writing over whatever was in the file before.

The form of the FORTIO statement is:

FORTIO //operation/unitno/closeopt/s,n,iostatus

This module has no input or output datablocks, it uses only parameters. The parameters used by it are:

Operation - character (no default) – operation to be performed
- EXISTS = check existence of a file
- OPEN = open a file (attach to the current run)
- CLOSE = close a file (release from the current run)

Unitno - integer (no default) – unit number of file

Closeopt - integer (default=2) – option if closing a file
- 1 = rewind
- 2 = close/keep
- 3 = close/delete

Iostatus - integer (output – no default) – status code from the operation
0 = ok
1 = not ok

Errors which occur when processing FORTIO do not terminate the program, they simply set iostatus to 1. It is up to you to catch this and handle it properly.

When the ISHELL module is invoked, it suspends MSC/NASTRAN and sends a command to the operating system. The command contains the information on the ISHELL statement, plus names for four files, if desired.

The form of the ISHELL DMAP statement is:

ISHELL //prgname/s,n,irtn/noin/noreal/nocmpx/nochar/nounit/
int1/int2/int3/int4/
real1/real2/real3/real4/
complex1/complex2/complex3/complex4/
char1/char2/char3/char4/
unit1/unit2/unit3/unit4 $

The module has no input or output datablocks. All input and output is by parameters. The associated parameters are:

Prgname - character – name of the external program to be executed. This name is 8 or fewer characters in length. This is the first term on the command line issued by the program. It may be any command which is recognized by the computer operating system.

Irtn - integar (default = 0) - output parameter - this parameter is set by MSC/NASTRAN to indicate if the external program was found. A value of –1 indicates the program was not found.

Noin - integer parameter (default = 0) – this is the second term on the command line. It is intended to tell the called program how many integer arguments (int1, int2, etc) are being passed. It is not used by MSC/NASTRAN. There are always 4 places for integer arguments on the command line issued.

Noreal -integer (default = 0) – the next term on the command line – this is intended to tell the called program how many real arguments are on the command line. Like noin, it is not used by MSC/NASTRAN.

Nocmpx -integer (default = 0) – similar to noreal – intended to indicate the number of complex arguments passed.

Nochar -integer (default = 0) – similar to noreal – intended to indicate the number of character arguments passed.

Int1-4 – integers(default = 0) – integer parameters passed on the command line.

Real1-4 – real (default = 1.0) – real parameters passed on the command line

Complex1-4 – complex {default = (-1.,0.)} – complex parameters passed on the command line.

Char1-4 -character (default = 'NULLNULL') – character parameters passed on the command line

Unit1-4 -integer - (default = 0) unit numbers for files to be used by the called program. These numbers and the associated file names will be on the command line (spaces 31-34) issued by MSC/NASTRAN. These files must have been ASSIGNed in the FMS section. They must also be 'closed' by the FORTIO statement before execution of this statement.

Starting with V70.6, the command is echoed in the '.f04' file. In previous versions, it is not printed.

These commands are best demonstrated by simple examples.

Example 1 – execute a program which uses no files from MSC/NASTRAN

```
SOL 101
Compile xxxxx    $ subMDAP not shown
Alter yyy   $ line number not shown.
Type parm,,i,n,worked
ISHELL  //'prog1.exe'/s,n,worked $
If(worked=-1)then $    didn't find program
  Message //'program prog1.exe not found – run terminated' $
 Exit $
Endif $
```

The previous example spawns a process running a program named "prog1.exe". If this program is found, it will be run and MSC/NASTRAN will continue the run to completion. If it is not found, the parameter "worked" will have a value of –1 when the module returns and the alter will print out a message and terminate the run.

Example 2 – execute a program using information from OUTPUT2 files written by the current run and read updated information back in using INPUTT2.

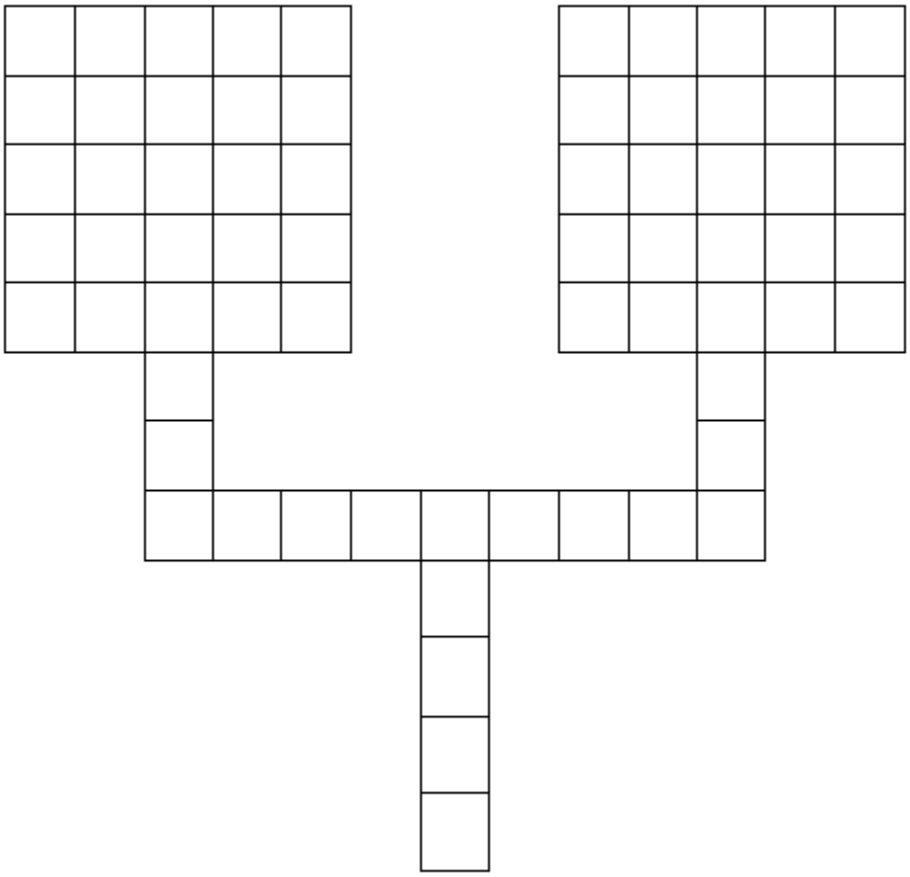
```
ASSIGN OUTPUT2='file.out2', unit=41, delete   $ file to use in transferring data to the program
ASSIGN INPUTT2='file2.inp2', unit=42, delete  $ file to get information from the program in
SOL 101
Compile xxxx $
Alter yy
Mpyad ug,three,/ugthree $
Output2 ugthree,,,//-1/41 $
Type parm,,i,n,closedit,isitok $
FORTIO //'CLOSE'/41//s,n,closedit $ release unit 41
If(closedit=1)then $
  Message //'unit 41 not closed properly – run terminated' $
  Exit $
Endif $
FORTIO //'CLOSE'/42//s,n,closedit $ release unit 42
If(closedit=1)then $
  Message //'unit 42 not closed properly – run terminated' $
  Exit $
Endif $
Message //' files 41 and 42 released ok – executing external program' $
ISHELL //'myprog'/s,n,isitok////////////////////////41/42 $
If(isitok=-1)then $
  Message //'program myprog not executed ok – run teminated' $
  Exit $
Endif $
Message //'program myprog executed ok – reading results into MSC/NASTRAN' $
INPUTT2 /new1,new2,new3,new4,new5/-1/42 $
```

In the above, two files are created by the run ("file.out2" and "file2.inp2"). A matrix named "ugthree" is written into unit 41 ("file.out2") and then units 41 and 42 are closed. At this point, an external program "myprog" is executed. This program reads the matrix from file 41 ("file.out2"), performs some operations

on it, and writes 5 new matrices and/or tables into unit 42 ("file2.inp2"). MSC/NASTRAN then resumes processing and reads the 5 new datablocks from unit 42.

Example: Using ISHELL to Provide Formatted Modal Effective Mass Printout

The following is an example of using the ISHELL module. The model used is the famous "two-headed flyswatter" model used in the MSC/NASTRAN Superelement User's Guide[2]. For demonstration purposes, I am simply calculating the modes and then the modal effective mass.



In this example, I am using an external program to write the modal effective mass in a separate file. There is already an alter (effmassa.vxxx which prints it in the '.f06' file) in the sssalter directory, so this is simply a modification of that alter to use an external program to re-write the results.

Modal effective mass is defined as follows. First a modal participation factor is calculated:

$$\varepsilon = \{\Phi\}^T [M] \, Dm$$

where

Φ = the eigenvector for the mode
M = the mass matrix for the model
Dm= a rigid-body vector in the associated direction

Then the modal effective mass is found as the square of the participation factor. The DMAP alter will calculate the modal effective mass (meff), the modal effective weight (wteff = meff*gravity), and the total mass (mtot) represented by the modes, then write these matrices, along with the eigenvalues, into an OUTPUT4 file. The external program , efm.bas, will read this data, then format and print it in a separate file

The alter – ishella.v705:

```
$    ishella.v705 - demonstrate use of ISHELL module
$
$ alter to calculate modal effective mass and participation factors
$ then write the data to a file for ISHELL to print
$
$  by Ted Rose
$  June, 1999
$  For V70.5
$
compile semodes $
alter 'modefsrs'(,-1)
type parm,,i,n,storit=0
call dbstore bgpdts,cstms,,,//0/0/'DBALL'/s,storit $
$
compile moders, list $
alter 'RETURN'(,-1) $ before return statement
type parm,,i,n,gotit,close11,close6, open6,itworked
type parm,nddl,rs,y,wtmass
type parm,,rs,n,masswt
type parm,,cs,n,masswtc
call dbfetch /bgpdts,cstms,,,/0/0/0/0/s,gotit $
VECPLOT ,,BGPDTS,EQEXINS,CSTMS,,,,/RBTG1/GRDPNT//4 $
trnsp rbtg1/rbg1 $
UPARTN   USET,rbg1/rba,,,/'G'/'A'/'O'/1 $
LAMX, ,LAMA/LAMAMAT/-1 $
smpyad pha,mmaa,rba,,,/partfac/3////1 $
smpyad rba,mmaa,rba,,,/mtot/3////1 $
diagonal partfac/meff/'whole'/2. $
masswt = 1./wtmass $
masswtc = cmplx(masswt,0.) $
add meff,/wteff/masswtc
$
output4 meff,wteff,mtot,lamamat,//-1/11/-1 $ write bcd
$
FORTIO //'close'/11/2/s,n,close11 $
if(close11<>0)then $
  message //'error - cannot close unit 11 - ishell not run' $
  message //'error code returned from FORTio ='/close11 $
else $
ishell //'efm.exe'/s,n,itworked $
 message //'finished with ishell - itworked = '/itworked $
if(itworked<=-1)then $
   message //'ishell failed, itworked = '/itworked $
  else $
   message //'ishell worked - see output above' $
  endif$
endif $
```

Input file ishell.dat

```
assign output4='output.dat', formatted, unit=11, delete
SOL 103
TIME 5
diag 8
include 'ishella.v705'
CEND
TITLE = SAMPLE PROBLEM 1 - minimum database sol 103
SUBTITLE = initial run - create database
$ ECHO = PUNCH,SORT
DISP = ALL
method =1
BEGIN BULK
eigrl,1,-1.,1000.,5
$
$ ADD MODEL DATA
$
INCLUDE 'model.dat'
$
ENDDATA
```

The above calculates the first 5 modes under 1000 hz for this model, then calculates the modal effective mass, writes it into a file named 'output.dat', then executes a program named 'efm.exe', which reads the file, and reformats the results in a new file (The program efm.bas – written in BASIC on my PC – is available upon request, either email me, or ask your local support staff for a copy if you wish).

OUTPUT4 file 'output.dat'

The following is the OUTPUT4 file containing the results of the modal effective mass alter. As this is not a desireable format for reading results, this data will be re-formatted by 'efm.exe'.

```
    6     5     2     2MEFF    1P,5E16.9
    1     1     5
 9.620566452E-25 2.275536591E-24 3.633813121E-25 4.994418613E-25 2.273676541E-24
    2     1     5
 4.341680330E-25 3.201760299E-24 6.543543547E-24 2.065635608E-23 5.479815320E-24
    3     1     5
 1.327433891E-03 2.696747310E-25 1.237296174E-25 1.921092049E-05 1.722595084E-04
    4     1     5
 8.228709818E-02 7.595393249E-25 3.930892307E-25 3.635373902E-05 8.018339922E-04
    5     1     5
 6.047345878E-25 1.485839666E-02 6.954121926E-04 9.363335629E-26 3.205879522E-26
    6     1     5
 3.298605386E-24 7.680593762E-24 1.103475722E-24 1.400110165E-24 7.510538843E-24
    7     1     1
-1.924681099E-31
    6     5     2     2WTEFF   1P,5E16.9
    1     1     5
 3.714504583E-22 8.785855945E-22 1.403016710E-22 1.928347038E-22 8.778674285E-22
    2     1     5
 1.676324524E-22 1.236200941E-21 2.526464800E-21 7.975427404E-21 2.115758903E-21
    3     1     5
 5.125227600E-01 1.041215223E-22 4.777205513E-23 7.417344140E-03 6.650946560E-02
```

```
     4     1     5
 3.177108176E+01 2.932584393E-22 1.517719103E-22 1.403619328E-02 3.095884274E-01
     5     1     5
 2.334882679E-22 5.736832934E+00 2.684989277E-01 3.615187658E-23 1.237791375E-23
     6     1     5
 1.273592868E-21 2.965480346E-21 4.260524207E-22 5.405830986E-22 2.899822073E-21
     7     1     1
-1.049841589E-22
     6     6     6     2MTOT    1P,5E16.9
     1     1     6
 1.570021679E-03 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00
-1.090424767E-02
     2     2     5
 1.570021679E-03 0.000000000E+00 0.000000000E+00 0.000000000E+00-1.253608762E-19
     3     3     3
 1.570021679E-03 1.090424767E-02 1.253608762E-19
     4     3     3
 1.090424767E-02 8.319332317E-02 2.710505431E-19
     5     3     3
 1.253608762E-19 2.710505431E-19 1.562269789E-02
     6     1     6
-1.090424767E-02-1.253608762E-19 0.000000000E+00 0.000000000E+00 0.000000000E+00
 9.881602106E-02
     7     1     1
 3.015140325E+15
     5     5     1     1LAMAMAT 1P,5E16.9
     1     1     5
 1.171679443E+03 3.460238281E+03 3.991923047E+04 6.105859375E+04 1.187518281E+05
     2     1     5
 3.422980499E+01 5.882379150E+01 1.997979736E+02 2.471003723E+02 3.446038818E+02
     3     1     5
 5.447842598E+00 9.362096786E+00 3.179883385E+01 3.932724380E+01 5.484540939E+01
     4     1     5
 1.000000000E+00 1.000000000E+00 1.000000000E+00 1.000000000E+00 1.000000000E+00
     5     1     5
 1.171679443E+03 3.460238281E+03 3.991923047E+04 6.105859375E+04 1.187518281E+05
     6     1     1
 1.171679443E+03
```

Results file 'effmass.prt' from efm.exe

modal effective mass summary

mode	frequency	mass x	mass y	mass z	iner x	iner y	iner z
1	54.478E-01	96.206E-26	43.417E-26	13.274E-04	82.287E-03	60.473E-26	32.986E-25
2	93.621E-01	22.755E-25	32.018E-25	26.967E-26	75.954E-26	14.858E-03	76.806E-25
3	31.799E+00	36.338E-26	65.435E-25	12.373E-26	39.309E-26	69.541E-05	11.035E-25
4	39.327E+00	49.944E-26	20.656E-24	19.211E-06	36.354E-06	93.633E-27	14.001E-25
5	54.845E+00	22.737E-25	54.798E-25	17.226E-05	80.183E-05	32.059E-27	75.105E-25
total		63.741E-25	36.316E-24	15.189E-04	83.125E-03	15.554E-03	20.993E-24

modal effective weight summary

mode	frequency	weight x	weight y	weight z	iner x	iner y	iner z
1	54.478E-01	37.145E-23	16.763E-23	51.252E-02	31.771E+00	23.349E-23	12.736E-22
2	93.621E-01	87.859E-23	12.362E-22	10.412E-23	29.326E-23	57.368E-01	29.655E-22
3	31.799E+00	14.030E-23	25.265E-22	47.772E-24	15.177E-23	26.850E-02	42.605E-23
4	39.327E+00	19.283E-23	79.754E-22	74.173E-04	14.036E-03	36.152E-24	54.058E-23
5	54.845E+00	87.787E-23	21.158E-22	66.509E-03	30.959E-02	12.378E-24	28.998E-22
total		24.610E-22	14.021E-21	58.645E-02	32.095E+00	60.053E-01	81.055E-22

Fraction of total available weight represented by the calculated modes

weight x	weight y	weight z	iner x	iner y	iner z
40.599E-22	23.131E-21	96.744E-02	99.918E-02	99.559E-02	21.245E-23

This is in a much easier to read form.

Conclusions:

This paper was written for a single purpose, to demonstrate the use of the ISHELL module. I have attempted to show the form of the DMAP statement and a sample application using it. As mentioned earlier, this module allows you to “interrupt” MSC/NASTRAN at any point, run an external program, then continue MSC/NASTRAN, including (if you wish) results from the external program.

By using this module, you can customize MSC/NASTRAN in any way you wish. If there is no MSC-provided module to perform a desired calculation, or if you wish to get results formatted in a different manner than that provided by MSC, this provides you with a tool.

Another use, which has already been used by several clients, is to edit or modify tables used by MSC/NASTRAN during the run. As most of the input data is stored in tables during the run, this allows you to “edit” the input data to modify it during the run.

References

1. MSC/NASTRAN Version 70 Release Guide , The MacNeal-Schwendler Corporation, Los Angeles, Ca, 1997
2. MSC/NASTRAN Superelement Users’ Guide, V70.5, The MacNeal-Schwendler Corporation, Los Angeles, Ca, 1999