EXPERIENCES IN INTERFACING NASTRAN WITH ANOTHER FINITE ELEMENT PROGRAM

By Dennis D. Schwerzler and Robert K. Leverenz

Research Laboratories
General Motors Corporation
Warren, Michigan

SUMMARY

This paper deals with the coupling of NASTRAN to another finite element program which has been developed by the General Motors Research Laboratories for the static analysis of automotive structures. The two programs were coupled together to use the substructuring capability of the in-house program and the normal mode analysis capability of NASTRAN. Modifications had to be made to the NASTRAN program in order to make the coupling feasible. This information can be of use to other NASTRAN users since there are many in-house finite element programs that are specially designed for particular problems or have capabilities not found in NASTRAN. By coupling the two programs together, the capabilities of both programs can be utilized.

INTRODUCTION

An interface program was written to allow an in-house finite element program to be used for the static analysis and the NASTRAN program for the normal mode analysis of automotive structures. By using the in-house finite element program, the following benefits were gained: flexibility of substructuring, an extended element library, an easily modified program for particular problems, and reduced cost of execution. A majority of the geometric data had already been prepared and checked in a format compatible with the in-house program, thus, the coupling of the two programs saved redefining and debugging the data for the NASTRAN format. Since the in-house finite element program does not have dynamic capabilities and the NASTRAN dynamic software is reputed to be one of the best, it becomes a logical choice to couple these two programs rather than develop a dynamic analysis code or use another program.

This paper describes how the two programs are coupled together. The information contained in this paper is based on the use of NASTRAN version 12.1 on an IBM 370/165 computer.

9

EXPLANATION OF INTERFACE PROGRAM DEVELOPED

When a substructure is modeled with the in-house code, the grid point data and the reduced stiffness matrix for the kept grid points are stored in a partitioned data set. When a normal mode analysis of one of these substructures is desired, this data, along with other user supplied data, is loaded into NASTRAN for the dynamic portion of the analysis by means of the interface program.

The input deck to the interface program is the NASTRAN executive control deck, the case control deck, and the bulk data deck, excluding grid point and element data. The interface program reads the input deck, converts any EBCDIC characters to BCD, and stores the card images in an output file for processing by NASTRAN. When processing the bulk data deck, the stiffness matrix for the substructure is inserted in its sorted position by placing the stiffness coefficients on DMIG cards. Likewise, GRID cards are also inserted in the bulk data deck in their correct sorted position.

An additional program processes the original data used to generate the substructure to determine the mass at each grid point by means of the lumped mass method. The mass at each grid point is then partitioned out to the kept grid points surrounding the grid point. This program then punches CØNM2 cards for the masses at each of the kept grid points. These cards are then read in as part of the bulk data input to the interface program.

Dummy rod elements are used to define the shape of the structure for plotting the mode shapes. To avoid changing the stiffness matrix, the rods are given an elastic modulus value of zero.

Grid point constraints imposed on the structure in the in-house finite element program are duplicated in the NASTRAN bulk data deck. This avoided calculating extraneous eigenvalues.

The stiffness values on the DMIG cards are read into the normal mode analysis format by means of the ALTER program shown below.

```
ALTER     26,27

MTRXIN,   ,MATPØØL,EQEXIN,SIL,/STIF,,/V,N,LUSET/V,N,NØM1/C,N,0/C,N,0 $

SMA1      CSTM,MPT,ECPT,GPCT,DIT/KGGY,,GPST/V,N,NØGENL/V,N,NØK4GG $

ADD       KGGY,STIF/KGGX/C,N,(1.0,0.0)/C,N,(1.0,0.0) $

CHKPNT    KGGX,GPST $

ENDALTER
```

This is similar to the alter program given in reference 1.

MODIFICATIONS TO NASTRAN

Using the DMIG bulk data cards for inputting the stiffness matrix into NASTRAN creates several problems. Even though the stiffness matrix is symmetric, the entire matrix has to be read in since the direct matrix input method does not take advantage of symmetry. This requires reading in nearly twice as many cards as would normally be needed. When reading in large matrices, such as a 184 by 184, these extra values consume several minutes of computer time in order to pass through the input file processor.

Another disadvantage of the direct matrix input method is that only single precision values for the stiffness matrix can be read in. For the IBM 370, single precision is only six significant figures, therefore, round-off errors may significantly affect the accuracy of the solution. In order to read in six significant figures in an E field format, the large field format has to be used. A sorting problem developed when the large field format was used on the second DMIG type header card. The interface program outputs the DMIG cards in the correct sorted order such that GJ and CJ were in an increasing numerical order. However, NASTRAN sorted the deck with CJ in a decreasing numerical order. When the cards were input with CJ in a decrasing order, NASTRAN sorted them into the correct order of increasing CJ number. This problem could not be resolved, so it was reported to the NASTRAN system office for further study.

The sorting problem does not occur if the small field format is used for the second type header cards. However, if the small field format is used, a stiffness value must be placed on this header card, and the small field width does not allow inputting a stiffness value of six significant figures in an E field format. To overcome this restriction, a completely constrained fictitious grid point with a zero stiffness value is placed in this field.

After the sorting problem was corrected, problems still occurred in trying to pass a large number of cards (20,000) through the input file processor. Nine minutes of central processing time failed to pass these cards through the XSØRT subroutine. The problem was traced to a double DØ LØØP in the XSØRT subroutine in which a check was made for duplicate continuation cards. It appeared that in this subroutine each continuation card was compared to every other continuation card, resulting in (N)*(N-1)/2 comparisons. Approximately 19,000 of the 20,000 cards were continuation cards resulting in approximately 1.8×10^8 comparisons. At about 5 micro-seconds for each comparison on the IBM 370/165, it would have taken about 900 seconds to complete this cycle. By removing this double DØ LØØP without any other modifications to the subroutine, the 20,000 cards were processed by the XSØRT subroutine in 138 seconds.

Since most of the bulk data deck was created by the interface program which had been thoroughly checked out, and since the data was already sorted, it was felt that most of the checks performed by the XSØRT subroutine were not needed. A very brief, modified version of the XSØRT subroutine was created

especially for use with the interface program (Appendix). This modified subroutine reads in the data either from direct input, a UMF tape, or a check point tape, and outputs the data on the NPTP file tape. This subroutine can only be used for UMF tape or check pointed tape problems that require no corrections to the bulk data deck. The new subroutine further reduced the time to process the 20,000 cards from 138 to 79 seconds.

Initially, an attempt was made to read in the entire mass matrix by means of DMIG cards, using the ALTER program given in reference 1. However, the mass matrix MGG must be opened before the program reaches the ALTER DMAP instruction where the mass values, read in by the ALTER program, are added to MGG. If not, MGG will be an "ill defined matrix," and the addition would not be possible. The mass matrix must be opened by some means at DMAP operation 28, where MGG is formed. The easiest method is to put in a small mass at any grid point. For the problems that were considered only the diagonal mass terms were needed, and it was found to be easier to read in the mass matrix by means of the CØNM2 cards. This method avoids the ill defined mass matrix problem.

OMITTING ROTATIONAL DEGREES OF FREEDOM

At first, all the degrees of freedom of each grid point were passed to NASTRAN for the dynamic analysis. The rotational degrees of freedom were then omitted in NASTRAN, since only the translational degrees of freedom were needed to adequately define the fundamental modes of the structures. The substructuring program was changed so that it performs a Guyan reduction of the stiffness matrix, omitting the rotational degrees of freedom, before passing the stiffness matrix to the interface program. This change has several advantages. Most of the structures considered are made up from several substructures, resulting in stiffness matrices which are densely populated. The Guyan reduction process, which tends to fill up the stiffness matrix, had little effect on the fullness of the stiffness matrix passed, but it did significantly reduce the size of the matrix. The structure which initially required 20,000 cards to define the stiffness matrix was reduced to 6,000 cards with very little change in the eigenvalues and eigenvectors. Since the number of cards was greatly reduced, the input file processor time was reduced from 79 seconds to 27 seconds.

This method has the advantage that the reduction process is performed with the original sixteen significant figures for each stiffness coefficient, instead of the rounded off six significant figures passed to NASTRAN when the rotational degrees of freedom are left in. Because these rotational degrees of freedom are omitted before being passed to NASTRAN, these degrees of freedom have to be constrained in the bulk data deck of the NASTRAN run. Back substitution to obtain deflections for the rotational degrees of freedom is not possible with this method.

SPACE FRAME VERIFICATION PROBLEM

Figure 1 shows a diagram of the structure used to verify the interface program. The space frame structure shown consists of 22 grid points connected together by 32 rod elements. The space frame was modeled entirely in NASTRAN, and the resulting eigenvalues and eigenvectors were compared to those obtained by modeling the space frame in the substructuring program and passing the stiffness matrix over to NASTRAN by means of the interface program. A NASTRAN generated mass matrix was used for both runs, being read in as CØNM2 data in the interface case. The values for the 66 eigenvalues and eigenvectors agreed to four significant figures. This example problem provided an assessment of the loss in accuracy due to the restriction of passing only six significant figures for the stiffness values. A comparison of the static analysis results between the two finite element programs produced exact agreement for the displacements.

The eigenvalues and eigenvectors for the analysis in which the rotational degrees of freedom were omitted before the data was passed to NASTRAN were closer to the values obtained by using NASTRAN directly than those values in which the stiffness values for the rotational degrees of freedom were passed to NASTRAN for omitting.

Omitting the rotational degrees of freedom with the substructuring program for the space frame increased the number of card images passed from 1,473 to 3,674. The reason for the increase was that the stiffness matrix for the space frame was very sparsely populated. The Guyan reduction process, although it reduced the size of the matrix, produced an almost fully populated matrix.

CONCLUDING REMARKS

The interface program has been used on considerably more complicated structures than the space frame shown. The largest structure analyzed was a car body composed of 15 substructures which together contained over 3,000 degrees of freedom. The substructuring program was used to reduce the structure to 186 translational degrees of freedom for passage to NASTRAN for the normal mode analysis. The dynamic analysis of the structure was successfully completed in 425 seconds with the input file processor requiring 137 seconds. The modifications made to the XSØRT subroutine, and the use of the substructure program made this analysis technically feasible and computationally economical.

REFERENCE

1. Jones, T. C. and Pinson, L. D., "Adaptation of NASTRAN to the Analysis of the Viking Space Vehicle," NASTRAN User's Experience, Sept. 12-15, 1971, p. 265, Vol. 1.

APPENDIX

```
      SUBROUTINE XSORT
C*SORT READS BULK DATA CARDS FROM THE INPUT TAPE , ADJUSTS THE FIELDS,
C* PERFORMS AN ALPHA-NUMERIC SORT ON THE CARD IMAGES FROM LEFT TO RIGHT,
C* INSERTS CONTINUATION CARDS IN THEIR PROPER POSITION, AND
C* PLACES THE RESULTING SORTED IMAGES ON THE NEW PROBLEM TAPE.
      INTEGER BIMSK1,BIMSK2,BIMSK3,BIMSK4,BIMSK5,BIMSK6
      INTEGER BKMSK1,BKMSK2,       SHIFTS
      INTEGER STAR,PLUS,DOLLAR,STARL,SLASH,SFTM,BLANK
      INTEGER BK,BUF,CCNT,CORSZ,ECHOS,ECHOU,FCNT,OPTP,
     1 OUTTAP,PTST,PSHIFT,SFTA,STARSW,ORF,ANDF,
     2 TRIAL,TST,UMF,PID
      EXTERNAL  LSHIFT,RSHIFT
      DIMENSION HEADU%32<,HEADS%32<,HEADN%32<,IBLKDA%2<,CDCNT%3<
      DIMENSION BK%4<,MK%4<,IBUF1%20<,IBUF2%20<,IBUF3%2<,KPARNT%2<
      DIMENSION IBUF1A%2<,IBUF2A%2<,NSORT%2<
      COMMON/SYSTEM/IBUFSZ,OUTTAP,NOGO,INTAPE,D1%14<,IECHO,D,IAPPRC,MACH
     1            ,  DUM1 , IUEDIT
      COMMON/OUTPUT/DUM2%96<,HEAD1%32<,HEAD2%32<,HEAD3%32<
      COMMON/XMDMSK/DUM4%5<,ICPFLG
      COMMON/FSORT/BUF%1<
      COMMON//ICOMON
      COMMON/XSRTBD/BIMSK1%6<,BIMSK2%5<,BIMSK3%4<,BIMSK4%4<,BIMSK5%2<,
     1              BIMSK6   ,BKMSK1%8<,BKMSK2   ,SHIFTS%4<,
     2  ICON1,ICON2,STAR,PLUS,DOLLAR,STARL,SLASH,SFTM,MASK,BLANK,MKA,IS
      COMMON /STAPID/ KRAP%12<,KUMF
      EQUIVALENCE %BK%1<,BKMSK1%5<<,%MK%1<,BIMSK2%2<<,
     1            %MKB  ,BIMSK5%1<<,%INF  ,BIMSK2%1<<,
     2            %SFTA ,SHIFTS%2<<,%MKD  ,BIMSK2%2<<,
     3            %MKE  ,BIMSK5%2<<,%MKC  ,BIMSK4%1<<
      DATA HEADU/10*4H     ,4H I N,4H P U,4H T  ,4H B U,4H L K,4H   D,
     1       4H A T,4H A  ,4H D E,4H C K,4H   E,4H C H,4H O  ,9*4H    /
      DATA HEADS/11*4H     ,4H S O,4H R T,4H E D,4H   B,4H U L,4H K  ,
     1       4H D A,4H T A,4H   E,4H C H,4H O  ,10*4H    /
      DATA HEADN/ 3*4H     ,4H    ,4H    ,4H    ,4H .  ,4H 1  ,4H..  ,
     1       4H 2  ,4H..  ,4H 3  ,4H..  ,4H 4  ,4H..  ,4H 5  ,4H..  ,
     2       4H 6  ,4H..  ,4H 7  ,4H..  ,4H 8  ,4H..  ,4H 9  ,4H..  ,
     3       4H10  ,4H.   ,5*4H    /
      DATA CDCNT/4HCARD,4HCOUN,4HT   /,NSORT/4HXSOR,4HT   /
      DATA ECHOU,ECHOS/2*0/
C     DATA BK/4H000 ,4H00  ,4H0   ,4H    /
C     DATA %MK%I<,I#1,4</0777777007777,0777700007777,0770000007777,00/
C     DATA MKA,MKB,INF,SFTA/0000000777777,0377777777777,0777777777777,6/
C     DATA MKC/0007777777777/,MKD/0777777007777/,MKE/0377777007777/
      DATA IEND1,IEND2/4HENDD,4HATA /,IEND/0/,ISEQ/0/,ICCBRK/0/
C     DATA STAR,PLUS,DOLLAR,STARL/4H000*,4H£000,4H$000,4H*000/
      DATA IBLKDA/4HBULK,4HDATA/,NOTSOR/0/,OPTP/4HOPTP/,NPTP/4HNPTP/
      DATA ITAPE1,ITAPE2,ITAPE3,ITAPE4,ITAPE5/301,302,303,304,305/
      DATA KIN/0/,UMF/4HUMF /
C  INITIALIZE XSORT
C##################################################################
      IRESTR # -IAPPRC
      IF%KUMF.LE.0< GO TO 90
      KIN#1
      CALL OPEN%$50,UMF,BUF%1<,2<
C  FIND PARTICULAR BULK DATA FILE ON UMF AS REQUESTED BY USER
```

```
   10 CALL READ%$30,$60,UMF,PID,1,1,IFLG<
      IF%KUMF-PID< 30,80,20
   20 CALL SKPFIL%UMF,1<
      GO TO 10
   30 WRITE%OUTTAP,35< KUMF
   35 FORMAT%52HO***USER FATAL MESSAGE 201,REQUESTED BULK DATA DECK I8,
     1 24H NOT ON USER MASTER FILE<
      NOGO#-1
      CALL CLOSE%UMF,1<
      RETURN
   50 WRITE%OUTTAP,55<
   55 FORMAT%1HO,23X,28H 202,UMF COULD NOT BE OPENED<
      GO TO 1800
   60 WRITE%OUTTAP,65<
   65 FORMAT%1HO,23X,23H 203,ILLEGAL EOR ON UMF<
      GO TO 1800
   80 CALL CLOSE%UMF,2<
C##################################################################################
   90 CALL INITCC
      IF%IECHO.LT.0< GO TO 110
      IF%IECHO.EQ.1< ECHOU# 1
      IF%IECHO.EQ.2< ECHOS# 1
      IF%IECHO.NE.3< GO TO 100
      ECHOU# 1
      ECHOS# 1
  100 IF%ICPFLG.NE.0< ECHOS# 1
  110 CONTINUE
C   START WORKING SORT BUFFER BELOW GINO I/O BUFFERS
      II# 5* IBUFSZ& 1
      IBUFBG# II& 42
      IBUFLG# CORSZ%BUF%1<,ICOMON< - 21
      IF%IBUFLG-IBUFBG.LT.210< CALL MESAGE%-8,0,NSORT<
      ITAPE# ITAPE1
      JTAPE# ITAPE2
      CALL OPEN %$1750,NPTP,BUF%4*IBUFSZ&1<,3<
      CALL WRITE %NPTP,IBLKDA,2,1<
      IF %IRESTR.GT.0 .OR. KIN.GT.0< GO TO 1400
  180 READ %INTAPE,190< IBUF1
      CALL XFADJ1 %IBUF1%1<,LSHIFT,0<
      IF %IBUF1%1<.EQ. IEND1 .AND.IBUF1%2<.EQ.IEND2< GO TO 560
  190 FORMAT %20A4<
      CALL WRITE %NPTP,IBUF1,20,1<
      GO TO 180
  560 CALL EOF %NPTP<
      CALL CLOSE %NPTP,1<
      RETURN
 1400 IF %KIN.GT.0< GO TO 1430
      CALL OPEN %$174C,OPTP,BUF%1<,0<
 1410 CALL READ %$1730,$1710,OPTP,IBUF3,2,1,IFLG<
      IF %IBUF3%1<.EQ.IBLKDA%1<.AND.IBUF3%2<.EQ.IBLKDA%2<< GO TO 1420
      CALL SKPFIL %OPTP,&1<
      GO TO 1410
 1420 CALL READ %$1240,$1710,OPTP,IBUF1,20,1,IFLG<
      CALL WRITE %NPTP,IBUF1,20,1<
      GO TO 1420
 1240 CALL CLOSE %OPTP,1<
      GO TO 560
```

```
1430 CALL OPEN %$50,UMF,BUF%1<,2<
     CALL PEAD%$1250,$1710,UMF,IBUF1,20,1,IFLG<
     CALL WRITE %NPTP,IBUF1,20,1<
     GO TO 1430
1250 CALL CLOSE %UMF,1<
     GO TO 560
1710 WRITE%OUTTAP,1711<
1711 FORMAT%1HO,23X,27H 211,ILLEGAL EOR ON SCRATCH<
     GO TO 1800
1730 WRITE%OUTTAP,1731<
1731 FORMAT%1HO,23X,24H 213,ILLEGAL EOF ON OPTP<
     GO TO 1800
1740 WRITE%OUTTAP,1741<
1741 FORMAT%1HO,23X,29H 214,OPTP COULD NOT BE OPENED<
     GO TO 1800
1750 WRITE%OUTTAP,1751<
1751 FORMAT%1HO,23X,29H 215,NPTP COULD NOT BE OPENED<
     GO TO 1800
1800  WRITE%OUTTAP,1801< PLUS
1801  FORMAT%A1,23H***SYSTEM FATAL MESSAGE<
     CALL MESAGE%-37,0,NSORT<
     NOGO # -1
     RETURN
     END
```

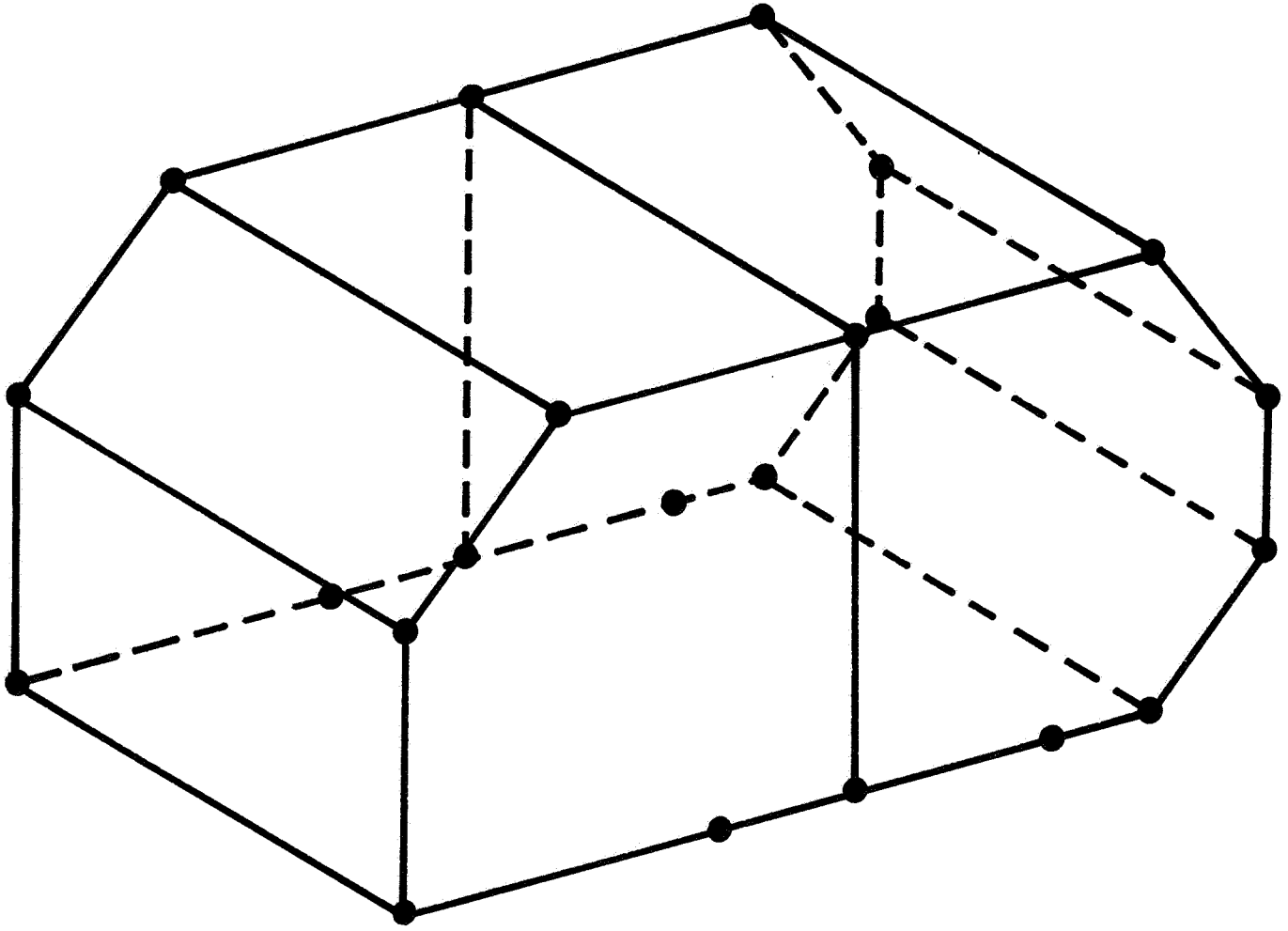

Figure 1. Space frame verification problem.