EIGENVALUE REANALYSIS USING SUBSPACE ITERATION TECHNIQUES

Gopichand Somayajula, Joseph Stout, and John Tucker
Ford Motor Company, Dearborn, Michigan

Abstract

During the process of designing dynamic elastic systems, it is often necessary to determine the effect of various design changes on the system dynamic characteristics, e.g., natural frequencies and mode shapes. Reanalysis techniques analyze changes with less computational effort. One such technique is a reduced basis method. This method is limited to design changes that do not significantly alter the resultant mode shapes. The proposed method, subspace iteration, allows for large changes in the mode shapes. This technique iterates the baseline solution until certain convergence criteria are met and is executed inside MSC/NASTRAN using a DMAP alter with user-specified parameters. An example demonstrates the accuracy and computational savings of the method.

1 INTRODUCTION

The procedure involved in progressing from a design concept to the final design of a structural component usually involves numerous intermediate design states. Each of these intermediate designs are often determined by analyzing several design modifications and choosing the best design based on the relevant evaluation criteria.

In many cases, the natural frequencies and mode shapes of the structure are important design evaluation criteria. The analytical determination of these frequencies and mode shapes (eigenvalues and eigenvectors) for each of the design levels requires a significant amount of time and money for all but the simplest models. It is very important, then, to use the most efficient calculation procedure available in order to produce a quality design quickly at a minimum cost. It is vital, however, that accuracy is not compromised for efficiency reasons.

2 METHODOLOGY

The procedure used in most approximate eigensolution methods contains an iteration loop. Initial solution vectors are generated and converge to the exact eigenvector as the number of solution iterations approaches infinity. The closer the initial guess is to the correct solution, the fewer the number of iterations required to converge to an accurate solution. When the structure being analyzed is a revision of a previous structure for which an eigensolution was conducted, it seems reasonable for solution efficiency to use the previous eigenvectors as the initial guess in calculating the revised eigensolution. Two different reanalysis methods will be discussed: a single-step Rayleigh-Ritz approach and the subspace iteration technique, which the authors believe is a more generally applicable procedure.

2.1 Rayleigh-Ritz Approach

C. M. Smith[1] described the implementation of a Rayleigh-Ritz eigensolution into MSC/NASTRAN using DMAP. The method he presented uses the eigenvectors of a previous design level as a reduced basis for calculating the eigenvectors and eigenvalues of the revised design. He demonstrated the technique using an example where the CPU time required to generate a solution for a design change was reduced from 27,900 sec. to only 900 sec. The steps involved in this method are:

1. Determine the mass and stiffness matricies for the structure with the proposed design changes; **M, K**.

2. Find the projections of **M** and **K** onto the assumed solution space, $\mathbf{X}_0$, where $\mathbf{X}_0$ is the matrix of eigenvectors from the previous design level eigensolution.

$$\tilde{\mathbf{K}} = \mathbf{X}_0^T \mathbf{K} \mathbf{X}_0 \quad (1)$$
$$\tilde{\mathbf{M}} = \mathbf{X}_0^T \mathbf{M} \mathbf{X}_0 \quad (2)$$

3. Solve the reduced eigenproblem and expand the solution to the full space.

$$\tilde{\mathbf{K}}\mathbf{Q} = \tilde{\mathbf{M}}\mathbf{Q}\lambda \tag{3}$$
$$\mathbf{X} = \mathbf{X_0}\mathbf{Q} \tag{4}$$

where λ is a diagonal matrix containing the approximate eigenvalues, and $\mathbf{X}$ are the approximate eigenvectors.

The Rayleigh-Ritz approximate eigenvalues are upper-bound approximations to the exact eigenvalues. But more importantly, the approximate eigenvectors are constrained to lie in the subspace spanned by the assumed solution space. So, if the actual eignevectors of the revised design cannot be represented by a linear combination of the eigenvectors of the initial design, an incorrect solution will result. The following simple example will illustrate this point.

The cantilever beam in Figure 1 is fixed at one end and supported by a very stiff spring in the vertical direction at its midspan. The stiff spring will provide a pinned connection at the midspan. This configuration is called the constrained case. The beam is also constrained to move only in the x-y plane. The first four mode shapes for this constrained configuration are shown in Figure 2. The next design iteration is to change the stiffness of the midspan support to a very light spring that would provide almost no support. The mode shapes generated by a full analysis are shown in Figure 3. This configuration is called the unconstrained condition. The frequencies for both sets of mode shapes are given in Table 1.

The eigenvalues for the unconstrained case were determined again using a Rayleigh-Ritz approach with the eigenvectors of the constrained case as the basis vectors for the solution space. A very different result was generated. The results were not the same as those obtained in the full analysis, but are the same as the results for the constrained condition. These frequencies, which are given in Table 2, are obviously in error. However, if the Rayleigh-Ritz approach is used with the unconstrained eigenvectors as the assumed solution space in solving for the constrained condition, the correct eigenvalues and eigenvectors are generated. These frequencies are listed in Table 3.

This inconsistency in the performance of the method can be attributted to the selection of base vectors for the solution space. Since the midspan point in the constrained case cannot have a displacement component in the y-direction, no linear combination of these mode shapes (Figure 2) can accurately represent the first, second, or fourth mode shapes of the unconstrained configuration (Figure 3) which have displacements in the y-direction at the midpoint. This mode deficiency does not arrise when solving for the constrained condition using the eigenvectors of the unconstrained case as the basis for the assumed solution space.

This constraint in the method places a large burden upon the analyst. To use this method accurately, the analyst must know whether or not the eigenvectors of the previous solution will span the solution space of the revised eigenvectors. This knowledge is available for only a small subset of possible design changes, e.g., overall material changes. The subspace iteration technique, however, does not have these restrictions.

2.2 Subspace Iteration Technique

The subspace iteration technique developed by Bathe [2] consists of the following steps:

1. Establish the starting vectors, $\mathbf{X_k}$.
2. Use simultaneous Ritz analysis and inverse interation to calculate the eigenvalue and eigenvector approximations.

$$\tilde{\mathbf{K}} = \mathbf{X_k^T K X_k} \tag{5}$$
$$\tilde{\mathbf{M}} = \mathbf{X_k^T M X_k} \tag{6}$$
$$\tilde{\mathbf{K}}\mathbf{Q} = \tilde{\mathbf{M}}\mathbf{Q}\lambda \tag{7}$$
$$\bar{\mathbf{X}}_\mathbf{k} = \mathbf{X_k Q} \tag{8}$$
$$\tilde{\mathbf{K}}\mathbf{X_{k+1}} = \tilde{\mathbf{M}}\bar{\mathbf{X}}_\mathbf{k} \tag{9}$$

3. If the solution in step(2) did not converge repeat step(2), otherwise proceed.
4. Use the Sturm sequence check to verify that the correct number of eigenvalues and eigenvectors were calculated.

The advantage of this method over the Rayleigh-Ritz method is that the starting vectors are not required to span the solution space in order to obtain an accurate answer. The inverse iteration step iterates the starting vectors from spanning a subspace of the solution space to spanning the entire desired solution space. If, however, the starting vectors do span the desired solution space, then the solution procedure will converge in a single step, as seen in the Rayleigh-Ritz solution. If the solution starting vectors do not span the desired solution space, the procedure will iterate until they do.

To illustrate this procedure, the eigenvectors of the constrained cantilever beam are used as the starting vectors in attempting to solve the eigenproblem of the unconstrained configuration. Based on the previous example using the Rayleigh-Ritz technique, it is known that the initial starting vectors do not span the desired solution space. The results of this analysis using a single inverse iteration are shown in Table 4.

The subspace iteration technique, then, shows an obvious benefit in solution accuracy over the Rayleigh-Ritz method. There is some additional CPU required in this technique, but it is well worth the price for the gain in accuracy.

3 IMPLEMENTATION

This section describes the implementation of the subspace iteration technique for eigenvalue reanalysis inside MSC/NASTRAN using a DMAP alter. The natural frequencies and mode shapes of the baseline design should be made available through a NASTRAN database to implement the DMAP program.

The reanalysis task requires two NASTRAN runs:

- A baseline run that extracts the eigenvalues and eigenvectors using the LANCZOS eigenvalue extraction method and saves them in a data base for the reanalysis run. The recommended procedure is to retain three times the number of modes of interest for improving the efficiency of the reanalysis run.

- A reanalysis run for the modified design which predicts the eigenvalues and eigenvectors using the DMAP alter in Sol 63. Modified Givens' eigenvalue extraction is recommended for the reanalysis run. Successful implementation of the DMAP alter requires all of the eigenvalues of the reduced problem at each subspace iteration.

Figure 4 contains the DMAP statements necessary to implement the subspace iteration technique. The eigenvectors and eigenvalues from the baseline run are retrieved using the DBFETCH module. The stiffness and mass matrices of the modified structure are projected onto the assumed solution space (which is the previous eigenvector solution) using SMPYAD modules. The resultant reduced eigenvalue problem is solved using the READ module. The approximate eigenvectors for the modified structure are obtained by mapping the eigenvectors of the reduced problem to the full problem using the MPYAD module. The stiffness matrix of the modified structure is decomposed using DECOMP for use in equation(9) to update the approximate eigenvectors which will be used as base vectors for the solution space in the next iteration.

The iteration loop (LOOPTOP) repeats until it exceeds the user specified maximum iteration limit for the REPT module or until the relative change (RLCHANGE) between two succesive eigenvalue solutions becomes less than the user-specified tolerance. The tolerance is specified through a parameter card (PARAM,FRQTOL,tolerance_value) in the bulk data. The maximum iteration limit is set so that the reanalysis time is less than the full analysis time. If the solution did not satisfy the specified eigenvalue tolerance within the maximum iteration limit, the iteration terminates. The quality of the last eigenvalue solution from the aborted run can be judged based on previous relative change (RLCHANGE) values. One must judge whether or not to accept the end solution. If it is not good enough, a full finite element analysis has to be performed. The program prints the maximum relative change, RLCHANGE, the iteration number, LOOP, and all the eigenvalues at the end of each iteration.

The eigenvalues for the convergence test are extracted from the eigenvalue solution table LAMA using MATGEN, MATMOD and PARTN modules, and the user-supplied range of modes. The range is specified through two parameter cards – PARAM, BMOD, beginning mode number and PARAM, EMOD, ending mode number – in the bulk data. This provision limits the convergence test to only a selected range of modes rather than all the modes. This option becomes powerful when there are rigid body modes. It eliminates rigid body modes, which cause convergence problems, from the convergence test by specifying the first non-zero mode number for BMOD.

Upon convergence, the Sturm sequence check is performed using another matrix decomposition. The Sturm sequence check confirms the complete extraction of modes within the prescribed range. This is a useful check when the design changes are very significant and local in nature. If all of the eigenvalues were extracted within the range, the program prints NO_MISS; otherwise, it prints MISSING. To perform reanalysis the DMAP alter has to be inserted into the executive control deck just before CEND. An example demonstrates the method.

4 NUMERICAL EXAMPLE

The reanalysis procedure was applied to an automotive powertrain to demonstrate two aspects: cost effectiveness and turn-around time for large-size problems. A sketch of the powertrain is shown in Figure 5. The finite element model has 64824 degrees-of-freedom. Two cases were investigated:

1. Predicting the effect of *adding* the knee brace on the natural frequencies using the powertrain without the knee brace as the baseline design. Results are shown in Table 5.

2. Predicting the effect of *removing* the knee brace on the natural frequencies using the powertrain with the knee brace as the baseline design. Results are shown in Table 6.

The results indicate an accurate prediction in both the cases. There is a cost reduction of 47% without the Sturm sequence check and 30% with the Sturm sequence check. The Sturm sequence check can be bypassed for computational savings if the design changes are not large and local in nature. The reanalysis problem was completed overnight using one million words of memory. The regular analysis requires a weekend to allocate the required two million words of memory.

5 CONCLUSIONS

This paper presents an eigenvalue reanalysis method using the subspace iteration method inside MSC/NASTRAN. It is not afflicted with problems associated with the size of design change and the selection of initial basis vectors for the solution space. The implementation procedure is simplified through a DMAP alter. However, it requires careful modeling of the design modifications to implement this method successfully.

Because the method uses eigenvectors of the baseline solution, it restricts the addition/deletion of nodes, alteration of node labels and locations, the explicit usage of rigid elements(like RBAR, RBE1, RBE2, etc.), single point constraints(SPCs) and multi point constraints(MPCs) for structural modifications in order to preserve the compatibility between the mass/stiffness matrices of the modified structure and the eigenvectors of the baseline design solution. Therefore, one has to plan his/her analysis strategy beforehand to get rid of this problem.

The compatibility problem never arises when the effect of thickness changes, cross-section properties, or material changes have to be investigated, but it is important when investigating the effect of adding another component to the existing model. In this case, the nodes and elements of all modifications must be included in the baseline model before generating the solution. The effect of these modifications on the baseline solution can be greatly minimized by specifying very small values for the material properties. The material properties are set to the actual values in the reanalysis problem to investigate the effect of modifications. Even though the method looks very restrictive, one can get around with these problems through a clever modeling strategy.

ACKNOWLEDGEMENTS

The authors thank D. H. Yun and R. G. Delosh of P/T NVH Design Analysis for their support and encouragement. Special thanks to M. J. Felice for his help in providing the powertrain FE model, and to P. M. Petrimoulx for editing this paper.

References

[1] C.M. Smith,The Application of Reanalysis Techniques to Large Finite Element Models through MSC/NASTRAN DMAP. The MSC 1988 World Users Conference Proceedings, Vol. 1.

[2] K. J. Bathe, *Finite Element Procedures in Engineering Analysis.* Prentice-Hall, Inc., Englewood Cliffs, N.J., 1982.

Cantilever Beam Model

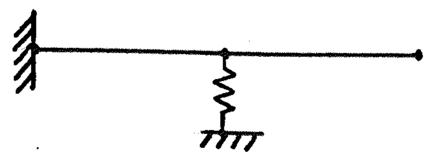


figure 1

Constrained Cantilever Beam Mode Shapes and Frequencies

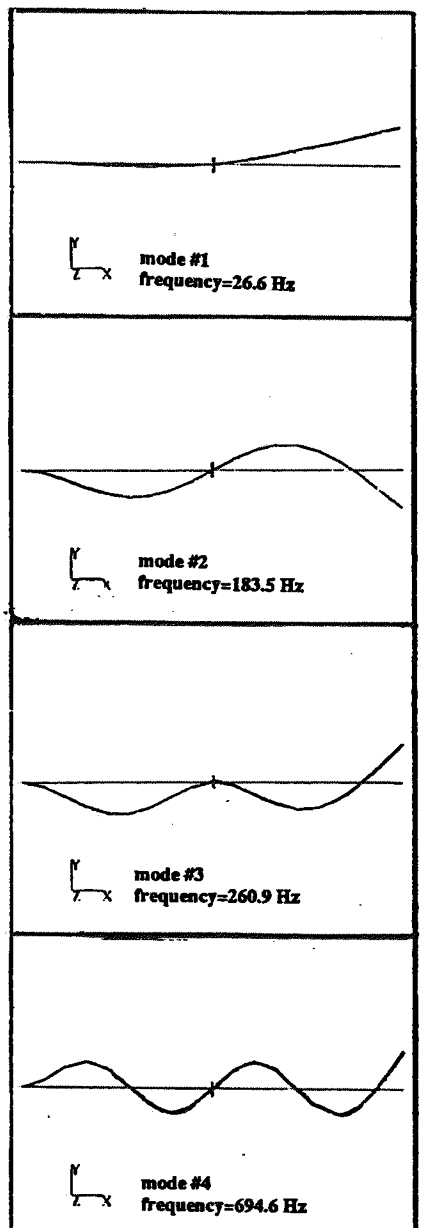


figure 2

Unconstrained Cantilever Beam Mode Shapes and Frequencies

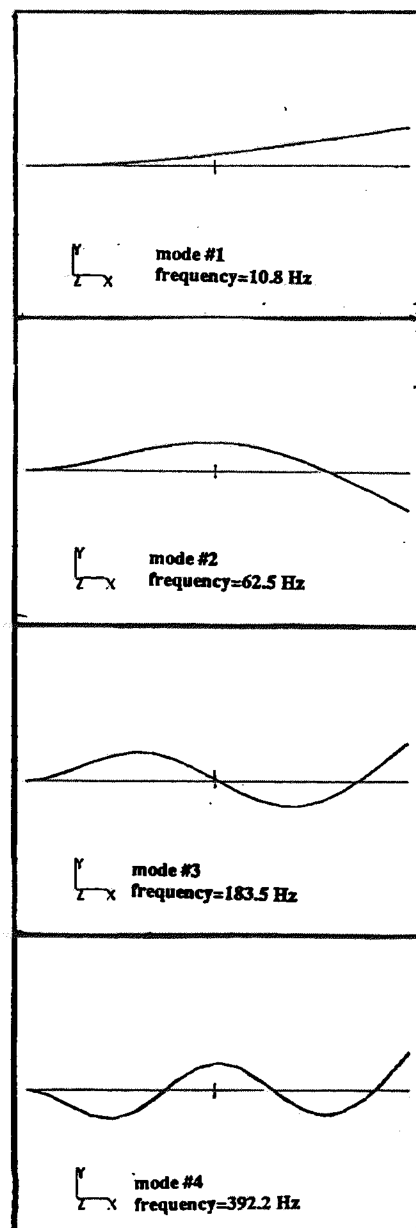


figure 3

MSC/NASTRAN IMPLEMENTATION OF SUBSPACE ITERATION METHOD

(SOL 63, VERSION 65)

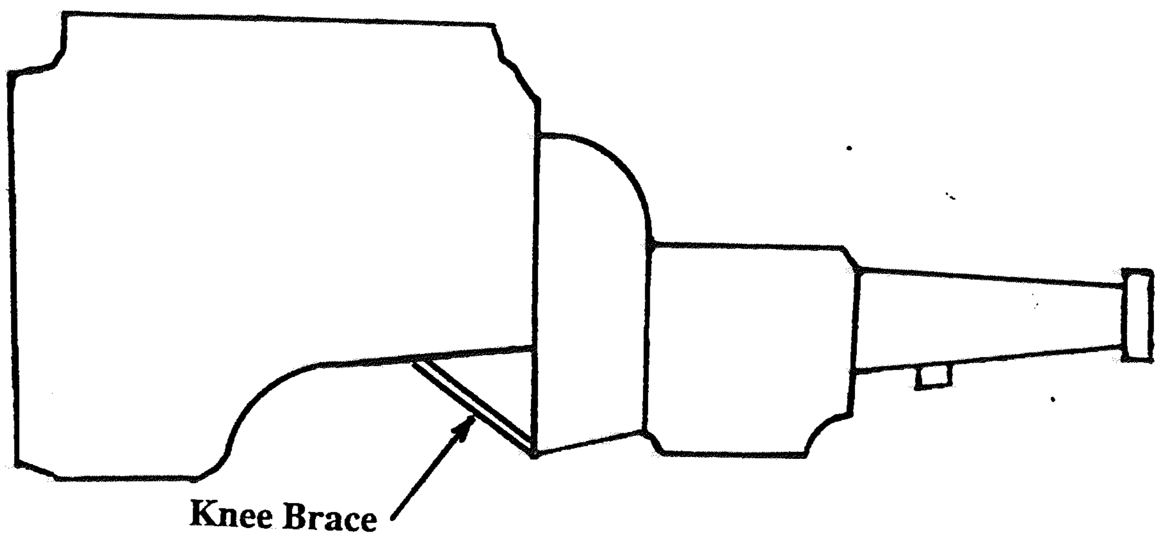
```
ALTER 771 $
JUMP ITERATE $
ALTER 801,801 $
LABEL ITERATE $
EQUIV MMAA,MXX/ALWAYS $
EQUIV MKAA,KXX/ALWAYS $
SETVAL //V,N,LOOP/1 $
SETVAL ///////////V,N,WARN/MISSING $
SETVAL ///////////V,N,INFO/NO_MISS $
DECOMP KXX/LKX,UKX/ $
DBFETCH /PHIA,LAMA,,,/V,Y,SOLID////-2 $
SMPYAD PHIA,KXX,PHIA,,,/KHH/3////1////6 $
SMPYAD PHIA,MXX,PHIA,,,/MHH/3////1////6 $
READ KHH,MHH,MR,DMX,EED,VXCOMPR,CASES/LAMA,PHIZ,MI,
   OEIGS/V,N,READAPP=MODES/S,N,NEIGV $
OFP LAMA// $
MPYAD PHIA,PHIZ,/PHIX $
LAMX, ,LAMA/LMAT/-1 $
PARAML LMAT//TRAILER/2/V,N,IMOD $
PARAM //SUB/V,N,SMOD/V,Y,BMOD=1/1 $
MATGEN ,/YPART/6/IMOD/V,Y,EMOD=3/IMOD $
MATGEN ,/YYPART/6/EMOD/SMOD/EMOD $
MATMOD LMAT,,,,,/OHZ,/1/3 $
PARTN OHZ,,YPART/HZOO,O21,O12,O22/1 $
PARTN HZOO,,YYPART/O11,HZO,O12,O22/1 $
$
LABEL LOOPTOP $
MPYAD MXX,PHIX,/RHS $
FBS LKX,UKX,RHS/PHIXX $
SMPYAD PHIXX,KXX,PHIXX,,,/KBAR/3////1////6 $
SMPYAD PHIXX,MXX,PHIXX,,,/MBAR/3////1////6 $
READ KBAR,MBAR,MR,DMX,EED,VXCOMPR,CASES/LAMA,PHIZ,MI,
   OEIGS/V,N,READAPP/S,N,NEIGV $
MPYAD PHIXX,PHIZ,/PHIX $
LAMX, ,LAMA/LMAT/-1 $
MATMOD LMAT,,,,,/NHZ,/1/3 $
PARTN NHZ,,YPART/HZNN,N21,N12,N22/1 $
PARTN HZNN,,YYPART/N11,HZN,N12,N22/1 $
MATMOD HZO,,,,,/DHZO,/28 $
MODTRL DHZO////3/ $
SOLVE DHZO,/DHZOI/ $
ADD HZO,HZN/DIF/(-1.,0.) $
MATPRN HZO,HZN// $
EQUIV HZN,HZO/ALWAYS $
MPYAD DHZOI,DIF,/ERRV/0 $
NORM ERRV/ERRVN///S,N,RLCHANGE $
PARAMR //LE//RLCHANGE/C,Y,FRQTOL=0.05////V,N,FLAG $
OFP LAMA// $
PRTPARM //0/C,N,RLCHANGE $
PRTPARM //0/C,N,LOOP $
PARAM //ADD/LOOP/LOOP/1 $
COND OUTLOOP,FLAG $
REPT LOOPTOP,10 $
```

Figure 4(cont.)

```
LABEL OUTLOOP $
PARAML LMAT//DMI/1/V,Y,EMOD/V,N,EVL $
PARAMR //NOP/V,N,EPSIG=0.005 $
PARAMR //ADD/V,N,PL/V,N,EVL/V,N,EPSIG $
PARAMR //NOP/V,N,NEGT=-1. $
PARAMR //MPY/V,N,CON/V,N,NEGT/V,N,PL $
PARAMR //COMPLEX//V,N,CON/V,N,ZERRO=0./V,N,FAC $
ADD KXX,MXX/CXX/(1.,0.)/FAC $
MODTRL CXX////6/ $
DECOMP CXX/LCX,UCX/1//////S,N,NBRCHG $
PARAM //LE/V,N,GOOD/V,N,NBRCHG/V,Y,EMOD $
COND NEXTLOOP,GOOD $
PRTPARM //0/C,N,WARN $
JUMP COMPLETE $
LABEL NEXTLOOP $
PRTPARM //0/C,N,INFO $
LABEL COMPLETE $
ALTER 806,806 $
JUMP LBLNOEXP $
CEND
TITLE=
METHOD=99
BEGIN BULK
$
$ USER SUPPLIED PARAMETERS
$
PARAM,BMOD,7
PARAM,EMOD,8
PARAM,FRQTOL,.05
$
$ EIGENVALUE EXTRACTION USING MODIFIED GIVENS' METHOD.
$
EIGR,99,MGIV,,,,8,,,+EIGR
+EIGR,MASS
$
```

Figure 4

Powerplant with Knee Brace



Powerplant without Knee Brace

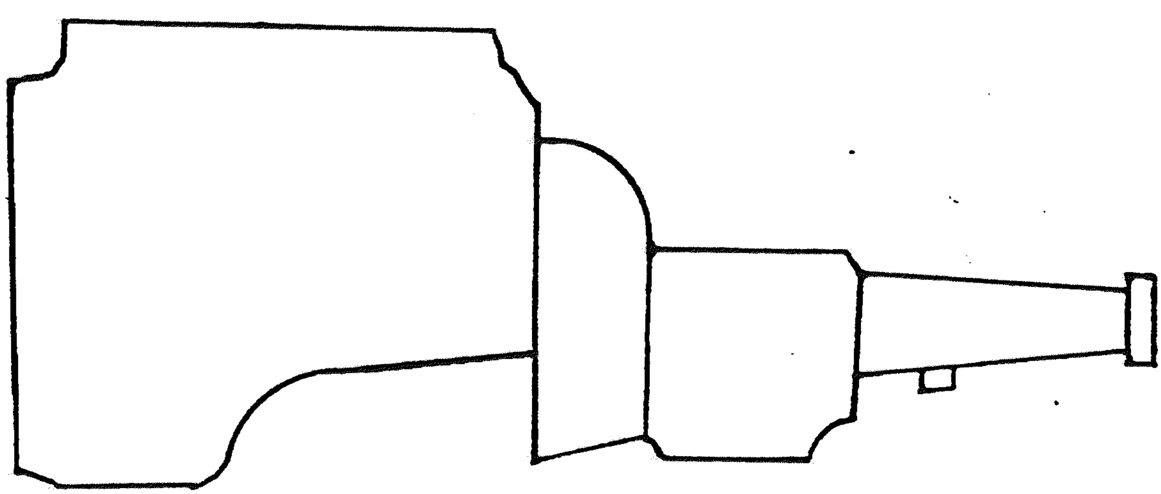


Figure 5

Frequency Comparison

Full Analysis

Constrained	mode #	Unconstranined
26.6 Hz	1	10.8 Hz
183.5	2	62.5
260.9	3	183.5
694.6	4	392.2

table 1

Frequency Comparison

Rayleigh-Ritz Approach

Constrained --> Unconstrained	
(26.6)	26.6 Hz
(183.5)	183.5
(260.9)	260.9
(694.6)	694.6

table 2

Frequency Comparison

Rayleigh-Ritz Approach

Unconstrained --> Constrained	
(10.8)	26.6 Hz
(62.5)	183.5
(183.5)	260.9
(392.2)	694.6

table 3

Frequency Comparison

Subspace Iteration Technique

Constrained --> Unconstrained	
(26.6)	10.8 Hz
(183.5)	62.5
(260.9)	183.5
(694.6)	392.2

table 4

Table 5

Predicting the effect of adding the knee brace on the natural frequencies and mode shapes of the powertrain using the results of the powertrain without knee brace.

MODE	BASE LINE	COMBINED RBS/SUBSPACE			FULL ANALYSIS	%Chg.*
	(No Knee) (brace)	IT 1	IT 2	IT 3	(With Knee brace)	
1	0.	0.	0.	0.	0.	
2	0.	0.	0.	0.	0.	
3	0.	0.	0.	0.	0.	
4	0.	0.	0.	0.	0.	
5	0.	0.	0.	0.	0.	
6	0.	0.	0.	0.	0.	
7	0.	0.	0.	0.	0.	
8	37.	38.	37.	37.	37.	
9	98.	176.	162.	161.	161.	
10	155.	180.	167.	167.	167.	
11	190.	219.	202.	201.	201.	
12	197.	240.	206.	205.	206.	
13	240.	265.	250.	249.	249.	
14	268.	272.	270.	270.	270.	
15	272.	279.	273.	273.	272.	
16	301.	301.	301.	301.	301.	
17	303.	351.	327.	326.	326.	
18	331.	445.	368.	345.	341.	1.5
19	378.	458.	445.	440.	427.	3.0
20	449.	484.	456.	455.	451.	
21	459.	488.	485.	485.	460.	5.4
22	484.	506.	506.	493.	485.	1.6
23	506.	529.	522.	506.	506.	
24	515.	538.	529.	529.	529.	
25	529.	543.	538.	538.	539.	
26	537.	564.	549.	544.	541.	
27	541.	576.	562.	560.	566.	1.1
28	565.	581.	572.	571.	575.	

CPU time	without final check**	3800 sec	7200 sec.
	with final check	5000 sec	
Memory		1M Cray words	2M Cray words.
Turn around time		Overnight	Weekend

* % Changes are shown only for those greater than 1.

** Strum sequence check to make sure all the eigenvalues are extracted in the specified range.

Table 6

Predicting the effect of removing the knee brace on the natural frequencies and mode shapes of the powertrain using the results of the powertrain with knee brace.

MODE	BASE LINE (W/ Knee) (brace)	COMBINED RBS/SUBSPACE IT 1	IT 2	IT 3	FULL ANALYSIS (No Knee brace)	%Chg.*
1	0.	0.	0.	0.	0.	
2	0.	0.	0.	0.	0.	
3	0.	0.	0.	0.	0.	
4	0.	0.	0.	0.	0.	
5	0.	0.	0.	0.	0.	
6	0.	0.	0.	0.	0.	
7	0.	0.	0.	0.	0.	
8	37.	37.	37.	37.	37.	
9	161.	152.	98.	98.	98.	
10	167.	165	155.	155.	155.	
11	201.	198.	190.	190.	190.	
12	206.	202.	197.	197.	197.	
13	249.	248.	240.	240.	240.	
14	270.	270.	268.	268.	268.	
15	272.	273.	272.	272.	272.	
16	301.	301.	301.	301.	301.	
17	326.	323.	303.	303.	303.	
18	341.	336.	331.	331.	331.	
19	427.	420.	380.	378.	378.	
20	451.	451.	449.	449.	449.	
21	460.	462.	459.	459.	459.	
22	485.	484.	484.	484.	484.	
23	506.	506.	505.	506.	506.	
24	528.	526.	516.	515.	515.	
25	529.	529.	529.	529.	529.	
26	539.	540.	538.	537.	537.	
27	541.	541.	549.	541.	541.	
28	566.	564.	561.	565.	565.	
29	575.	575.	565.	573.	572.	
30	590.	590.	589.	589.	589.	
31	598.	598.	596.	596.	592.	

CPU time	without final check**	3800 sec	7200 sec.
	with final check	5000 sec	
Memory		1M Cray words	2M Cray words.
Turn around time		Overnight	Weekend

* % Changes are shown only for those greater than 1.

** Strum sequence check to make sure all the eigenvalues are extracted in the specified range.