Implementation of the Block-Krylov Boundary Flexibility Method of Component Synthesis

Kelly S. Carney
*Lewis Research Center*
*Cleveland, Ohio*

Ayman A. Abdallah
*Analex Corporation*
*Brook Park, Ohio*

and

Arthur A. Hucklebridge
*Case Western Reserve University*
*Cleveland, Ohio*

Prepared for the
1993 MSC World Users' Conference
sponsored by the MacNeal-Schwendler Corporation
IMPLEMENTATION OF THE BLOCK-KRYLOV BOUNDARY FLEXIBILITY METHOD OF COMPONENT SYNTHESIS

Kelly S. Carney
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135

Ayman A. Abdallah
Analex Corporation
5001 Aerospace Parkway
Brook Park, Ohio 44142

Arthur A. Hucklebridge
Case Western Reserve University
Department of Civil Engineering
Cleveland, Ohio 44106

Abstract
A method of dynamic substructuring is presented which utilizes a set of static Ritz vectors as a replacement for normal eigenvectors in component mode synthesis. This set of Ritz vectors is generated in a recurrence relationship, which has the form of a block-Krylov subspace. The initial seed to the recurrence algorithm is based on the boundary flexibility vectors of the component. This algorithm is not load-dependent, is applicable to both fixed and free-interface boundary components, and results in a general component model appropriate for any type of dynamic analysis. This methodology has been implemented in the MSC/NASTRAN normal modes solution sequence using DMAP. The accuracy is found to be comparable to that of component synthesis based upon normal modes. The block-Krylov recurrence algorithm is a series of static solutions and so requires significantly less computation than solving the normal eigenspace problem.

1993 MSC World Users' Conference
Arlington, Virginia
May 24-28, 1993
IMPLEMENTATION OF THE BLOCK-KRYLOV BOUNDARY FLEXIBILITY METHOD OF COMPONENT SYNTHESIS

Kelly S. Carney
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135

Ayman A. Abdallah
Analex Corporation
3001 Aerospace Parkway
Brook Park, Ohio 44142

Arthur A. Hucklebridge
Case Western Reserve University
Department of Civil Engineering
Cleveland, Ohio 44106

Introduction

Component mode synthesis is a methodology for analyzing large structures by separating them into smaller components, which can then be recombined to analyze the entire system. This methodology has become well established and widely used in structural dynamic analysis. The advantages of component mode synthesis include, the lower computation costs associated with the smaller components which are analyzed, and the flexibility of data management gained by working with the discrete components.

The typical component mode synthesis algorithm is briefly described [1]. A large structure is broken into components, with each component having a set of boundary, or interface, points. At these interface points, fixed or free boundary conditions are assumed, and a corresponding set of component normal mode shapes, or eigenvectors, is determined. The eigenvectors are augmented by a set of modes which are associated with the component's boundary flexibility. Depending on whether a fixed or free interface is selected, these modes are the constraint modes or the attachment modes, respectively. The combined set of component normal modes and boundary modes are used to represent the component in subsequent system analysis, by using the following transformation process. The combined set of modes form a coordinate transformation matrix which transforms the physical coordinates of the structural model into a combination of modal coordinates and boundary coordinates. The boundary coordinates are retained in the physical space, so they can be used to couple the components for subsequent system analysis.

A component's size, although smaller than that of the complete structural model, can still be large enough to make computation expensive. The rapid reduction in cost per calculation in today's digital computers has not necessarily led to a reduction in total computation cost. Instead, engineers have exploited the increased computational resources by creating larger structural and component models. The larger models have allowed for more structural details to be included, as well as more refined data recovery, but they may be expensive to formulate and analyze. In order to reduce the computational cost associated with large component models, it is desirable to develop more efficient methods of formulation. Since the solution of the normal eigensystem problem requires the largest computational effort in component formulation, it is logical to develop alternate methods which circumvent the eigensystem solution entirely.

A method, which does circumvent the eigensystem solution, has been defined in literature and is briefly described [2-4]. The boundary flexibility modes, specifically either the same constraint modes or attachment modes that were mentioned previously, are multiplied by the component mass matrix to create a force matrix.
Static analysis is then performed, using this force matrix and the component stiffness matrix, to obtain a matrix, or block, of vector displacements. A recurrence relationship of matrix multiplications, which have been shown to be a Krylov sequence [3,5], then defines a series of matrices, or blocks, of vector displacements. The calculated vectors are orthogonalised, using normalised Gram-Schmidt orthogonalization [6]. These vectors, which can also be thought of as static modes, replace the normal modes in the component formulation methodology. Because the vectors are calculated in blocks and are based on a Krylov sequence, the subspace defined by these vectors is called a block-Krylov subspace.

The primary contribution described in this paper is the incorporation of the above methodology into MSC/NASTRAN. Since the block-Krylov vectors replace the normal modes in typical component mode synthesis, the physical to modal coordinate transformation, the constraint modes, and the attachment modes, are not changed. Therefore, the component mode synthesis formulation currently available in MSC/NASTRAN was utilized, except for the substitution of the normal modes by the block-Krylov vectors. In addition to the contribution described above, the technique of using Gram-Schmidt orthogonalization within each vector block was added to the methodology.

Problem Definition and Theory

Overview

Wilson, Yuan, and Dickens [7] originally proposed the use of Ritz vectors, based upon external loading, for structural dynamic analysis. This formulation reduced an entire structure, not a component. The algorithm begins with a set of externally applied loads. The displacements from the static solution to the applied loads become the initial Ritz vector. That vector is then multiplied by the mass matrix to become the next force vector. This sequence is repeated to form a recurrence relationship. This recurrence relationship is used in the papers discussed below and throughout this work.

Nour-Omid and Clough [5] investigated Wilson, et al.'s methodology and found that the proposed recurrence relationship was actually a Krylov sequence. A Krylov subspace of order j is a vector space defined by

\[ \{ \phi, A \phi, A^2 \phi, \ldots, A^{j-1} \phi \} \]  

where \( \phi \) is a column vector and \( A \) is a square matrix. If \( A \) is \( n \times n \) dimensional, and if \( j = n \), the Krylov vectors span the \( n \)-dimensional space [3], and an exact solution can be produced. In structural dynamics, the Krylov subspace can be defined by the following sequence,

\[ \{ r, K^{-1} M r, (K^{-1} M)^2 r, \ldots, (K^{-1} M)^{j-1} r \} \]  

where \( K \) is the stiffness matrix, \( M \) is the mass matrix, and \( r \) is a starting vector (or in block-Krylov, a set of vectors). This Krylov sequence is identical to the Lanczos eigenvalue extraction algorithm, when applied with complete orthogonalization with respect to both the mass and the stiffness matrices.

The use of Krylov vectors was shown to be applicable to component mode synthesis by Wilson and Bayo [8]. The Ritz vectors calculated were based, once again, upon an external load. Only a formulation for components with fixed interface boundary conditions was presented.

Craig and Hale [3], and Abdallah and Hucklebridge [2], demonstrated a methodology applicable to components with fixed or free interfaces, with or without rigid body modes, and with no applied loading. Components having no applied external loading were formed using the boundary flexibility matrix, multiplied by the mass matrix, to form a force matrix. This force matrix produces a set of vectors, which are referred to as a block. As discussed previously, the boundary flexibility matrix is defined as either the constraint modes
or the attachment modes, depending on whether fixed or free interface conditions are selected. The methodology contained in these two papers is reviewed in the next three parts of this section. Abdallah and Hucklebridge also quantified the advantages, in computational effort, that block-Krylov vectors have over normal eigenvectors.

Yiu and Landress [4] also developed a methodology for forming a component which does not have an external applied load. However, their formulation is applicable to fixed interface components only. A criteria for concluding the recurrence sequence, based upon the rigid body mass and flexibility represented by the calculated Ritz vectors, was proposed.

Fixed Interface Methodology

First, as is standard in component mode synthesis methods, the finite element component mass, m, and stiffness, k, matrices are partitioned into internal and external degrees of freedom, denoted by subscripts i and c, respectively.

\[
m = \begin{bmatrix} m_{cc} & m_{ci} \\ m_{ic} & m_{ii} \end{bmatrix}
\]

\[
k = \begin{bmatrix} k_{cc} & k_{ci} \\ k_{ic} & k_{ii} \end{bmatrix}
\]

The constraint modes are defined by

\[
\Phi_{ic} = -k_{ii}^{-1} k_{ic}
\]

which is the same definition used in standard component mode synthesis.

For Wilson’s method [7], a set of externally applied loads is required to obtain the initial set of Ritz vectors. For the boundary flexibility method, this set of loads is created by multiplying the constraint modes by the mass matrix. (Craig [3] also included the off-diagonal mass matrix in his formulation.) Since the mass matrix is used to create the loads, they can be considered inertia loads. This set of inertia loads are then used to generate the initial set, or block, of Ritz vectors using the following

\[
q_{1}^{**} = k_{ii}^{-1} (m_{ii} \Phi_{ic} + m_{ic})
\]

where the superscript ** indicates that the vectors in the matrix have not been normalized. The first block of vectors is normalized using the following equation. The subscript r, in the following equation, signifies that the block is normalized vector by vector, and there are c vectors within each matrix, or block.

\[
q_{1r} = \frac{q_{1r}^{**}}{\sqrt{(q_{1r}^{**})^T m_{ii} q_{1r}^{**}}}, \quad r = 1, 2, \ldots, c
\]
The subsequent sets of block-Krylov vectors are generated using the recurrence relationship [5, 7], which was defined in Eq. (2),

\[ q_j^* = k_{ii}^{-1} m_{ii} q_{j-1} \]  

(8)

where the superscript * signifies that the vectors have not been orthogonalized or normalized. The additional sets of vectors are orthogonalized, with respect to the mass matrix, with all previous vectors. The process used to perform this orthogonalization is a normalized Gram-Schmidt procedure.

\[ q_{j}^{**} = q_{j}^* - q_{1,j-1}^c \]  

(9)

where

\[ c = q_{1,j-1}^T m_{ii} q_{j}^* \]  

(10)

and \( q_{1,j-1} \) is the concatenation of the previous sets of block-Krylov vectors,

\[ q_{1,j-1} = [q_1, q_2, \ldots, q_{j-1}] \]  

(11)

where all vectors have been normalized as follows.

\[ q_{j} = \frac{q_{j}^{**}}{\sqrt{(q_{j}^{**})^T m_{ii} q_{j}^{**}}} \]  

(12)

The complete set of calculated block-Krylov Ritz vectors is included in the transformation matrix as \( Q \). (The resulting transformation matrix has the same form as that of *Craig-Bampton* component mode synthesis [9], with the Krylov vectors replacing the normal modes.)

\[ \Psi = \begin{bmatrix} I_{cc} & 0 \\ \Phi_{lc} & Q_1 \end{bmatrix} \]  

(13)

The physical mass and stiffness matrices are transformed into the component modal matrices

\[ \mu = \Psi^T m \Psi \]  

(14)

\[ \kappa = \Psi^T k \Psi \]  

(15)

The resulting mass submatrices are

\[ \mu = \begin{bmatrix} \mu_{cc} & \mu_{cl} \\ \mu_{lc} & \mu_{ll} \end{bmatrix} \]  

(16)

where

\[ \mu_{cc} = \Phi_{lc}^T (m_{ii} \Phi_{lc} + m_{lc}) + m_{cl} \Phi_{lc} + m_{cc} \]  

(17)

\[ \mu_{lc} = \mu_{cl}^T = Q_1^T (m_{ii} \Phi_{lc} + m_{lc}) \]  

(18)
\[ \mu_{ii} = I_{ii} = Q_j^T m_{ii} Q_i \]  

The resulting stiffness submatrices are

\[ \kappa = \begin{bmatrix} \kappa_{cc} & \kappa_{cl} \\ \kappa_{lc} & \kappa_{ll} \end{bmatrix} \]  

where

\[ \kappa_{cc} = k_{cl} \Phi_{ic} + k_{cc} \]  

\[ \kappa_{ic} = \kappa_{cl}^T = 0 \]  

\[ \kappa_{ll} = Q_j^T k_{ii} Q_i \]  

The use of constraint modes in the transformation matrix leads to the null off-diagonal partitions of the component stiffness matrix, just as in the approach based upon normal modes.

**Free Interface Methodology for Components with No Rigid Body Modes**

When allowing the interface points of a component to be free to deflect while forming the component, a somewhat different basis for the initial vector of the Krylov algorithm is required. The attachment modes, rather than the constraint modes are utilized in initial block definition. By definition, the attachment modes are the columns of the flexibility matrix which correspond to the interface degrees of freedom.

\[ g = k^{-1} \]  

\[ g_a = \begin{bmatrix} g_{cc} \\ g_{ic} \end{bmatrix} \]  

The initial block of vectors in the free interface formulation is defined as

\[ q_{i_r}^{**} = k^{-1} m g_a \]  

and is normalized as follows.

\[ q_{i_r} = \frac{q_{i_r}^{**}}{\sqrt{(q_{i_r}^{**}^T m q_{i_r}^{**})}} \quad r = 1, 2, ..., c \]  

Note that the unpartitioned physical mass and stiffness matrices of the component are used in the free interface formulation. The recurrence algorithm then proceeds in the same manner as in the fixed interface methodology.
\begin{align}
q_j^* &= k^{-1} m q_{j-1} \\
q_j^{**} &= q_j^* - q_{1,i-1} c \\
c &= q_{1,i-1}^T m q_i^* \\
q_{ir} &= \frac{q_{ir}^{**}}{\sqrt{(q_{ir}^{**} m q_i^{**})}} \\
r &= 1,2,...,c
\end{align}

Formation of block-Krylov component then follows the normal component mode synthesis techniques which were presented by MacNeal [10] and Rubin [11]. To combine the "Rubin-MacNeal" method with the presented method, the normal eigenvectors are simply replaced with the block-Krylov Ritz vectors, as in the fixed interface methodology. The free interface methodology uses residual flexibility terms, which fully define the stiffness missing from the modal space due to excluded modes and are described below. The flexibility contained in the calculated Krylov vectors is given by the following equation.

\begin{equation}
g_{k} = Q_i (Q_i^T k_i)^{-1} Q_i^T \tag{32}\end{equation}

The unrepresented flexibility, or residual flexibility, is defined as

\begin{equation}
g_{d} = g - g_{k} \tag{33}\end{equation}

The residual flexibility matrix is then partitioned in the same manner as the flexibility matrix was in Eq. (25), when the attachment modes were created for initial vector calculation. The result is the residual attachment modes.

\begin{equation}
g_{ad} = \begin{bmatrix} g_{cc_d} \\ g_{ic_d} \end{bmatrix} \tag{34}\end{equation}

When the residual attachment modes, \(g_{ad}\), are added to the Krylov modes, \(Q_i\), the complete flexibility of the component is represented.

The residual attachment modes and the Krylov modes are used to form the component transformation matrix. This matrix transforms the physical subspace, \(u\), to the modal subspace, \(p\), and is defined by the following equation.

\begin{equation}
\begin{bmatrix} u_c \\ u_i \end{bmatrix} = \begin{bmatrix} g_{cc_d} & Q_{cl} \\ g_{ic_d} & Q_{il} \end{bmatrix} \begin{bmatrix} p_c \\ p_i \end{bmatrix} \tag{35}\end{equation}

In order to provide physical interface degrees of freedom, for use in component coupling, \(p_c\) in the above equation is back-transformed to eliminate it from the right-hand side of the equation. This results in the following transformation matrix,
where
\[
\begin{bmatrix}
   u_c \\
   u_l
\end{bmatrix}
= \begin{bmatrix}
   I_{cc} & 0 \\
   g_{ic} & Q_{il}^* \\
\end{bmatrix}
\begin{bmatrix}
   u_c \\
   P_l 
\end{bmatrix}
\] (36)

\[
g_{ic}^* = g_{ic} g_{cc_d}^{-1} 
\]
(37)

\[
Q_{il}^* = Q_{il} - g_{ic} g_{cc_d}^{-1} Q_{cl} 
\]
(38)

The transformation of the component mass and stiffness matrices then proceeds in a similar manner as shown in Eqs. (14) to (23), with the following differences. The \(Q_{il}^*\) matrix partition replaces the \(Q_l\) matrix partition. The \(g_{ic}\) matrix partition replaces the \(g_{ic}\) matrix partition. In the fixed-interface methodology, the definition of the constraint modes, \(g_{ic}\), leads to terms in the component stiffness matrix which cancel out. In the free-interface methodology, the definition of the transformation submatrices has changed and so this cancellation does not occur. Therefore, Eqs. (21) and (22) are replaced by the following equations.

\[
\gamma_{cc} = g_{ic}^* T (k_{ii} g_{ic}^* + k_{ic}) + k_{cl} g_{ic}^* + k_{cc} 
\]
(39)

and

\[
\gamma_{jc} = \gamma_{cj}^* T = Q_{il}^* T (k_{ii} g_{ic}^* + k_{ic}) 
\]
(40)

### Free Interface Methodology for Components with Rigid Body Modes

When a component has rigid body modes, the associated stiffness matrix is singular. The inverse of the stiffness matrix, the flexibility matrix, cannot be directly obtained, and therefore the attachment modes cannot be directly obtained. To circumvent this problem, Rubin [11] presented the following method for obtaining the residual elastic attachment modes of a component with rigid body modes.

First, the stiffness matrix is constrained from rigid body motion by partitioning out \(r\) degrees of freedom, where \(r\) is the number of rigid body modes.

\[
k = \begin{bmatrix}
   k_{ww} & k_{wr} \\
   k_{rw} & k_{rr}
\end{bmatrix} 
\]

(41)

The remaining partition is then inverted.

\[
g_{ww} = k_{ww}^{-1} 
\]
(42)

This flexibility matrix is then expanded back to \(n (w + r)\) size.

\[
g_c = \begin{bmatrix}
   g_{ww} & 0_{wr} \\
   0_{rw} & 0_{rr}
\end{bmatrix} 
\]
(43)

A square projection matrix is defined by
\[ A = I_{nn} - m \Phi_r \Phi_r^T \]  

(44)

where \( \Phi_r \) is the rigid body modes matrix. The elastic flexibility matrix, \( g_e \), with rigid body motion removed, is shown in reference [11] to be

\[ g_e = A^T g_c A \]  

(45)

Now the analysis proceeds in a similar fashion to the previously discussed methodology for the free interface component with no rigid body motion. The major difference between the two approaches is that the elastic flexibility matrix is used in place of the general flexibility matrix. The inertia relief attachment modes are

\[ g_{n_e} = \begin{bmatrix} g_{cc_e} \\ g_{ic_e} \end{bmatrix} \]  

(46)

The initial block of vectors is calculated using the inertia relief attachment modes and the elastic flexibility matrix.

\[ q_{1}^{**} = g_e m g_{n_e} \]  

(47)

The subsequent block-Krylov Ritz vectors are calculated, orthogonalized, and normalized as shown in Eqs. (28) to (31). The residual elastic flexibility terms are also calculated as shown in the free interface with no rigid body modes discussion, Eqs. (32) to (34). Creation of the transformation matrix, Eqs. (35) to (40), is also similar to when no rigid body modes are present. The one exception is that the rigid body modes must be included in the transformation matrix. Therefore, Eq. (35) is replaced by

\[
\begin{bmatrix}
  u_c \\
  u_i
\end{bmatrix} =
\begin{bmatrix}
  g_{cc_d} & Q_{cl} \\
  g_{ic_d} & Q_{il}
\end{bmatrix}
\begin{bmatrix}
  \Phi_c \\
  \Phi_i
\end{bmatrix}
\begin{bmatrix}
  P_c \\
  P_i
\end{bmatrix}
\]  

(48)

Formation of the final transformation matrix, and subsequently the component mass and stiffness matrices, is then performed as described in the previous section.

**Discussion**

As previously discussed, most component mode synthesis applications use the normal eigenvalues of the substructure to form the component. It is accepted that if all the eigenvalues of a system are used to form the component, an "exact" finite element solution may be obtained. The same fact holds true for components based upon block-Krylov subspaces. Mathematically, this is proven in references [3] and [12]. If \( n \) orthogonal block-Krylov Ritz vectors are used to form a component of a \( n \)-size system, the same "exact" solution as from normal eigenvectors is obtained.

**Orthogonality**

Upon completion of the block-Krylov vector calculations and Gram-Schmidt orthogonalizations, all vectors are re-checked for orthogonality. Orthogonality checking is required because the normalized Gram-Schmidt
orthogonalization algorithm is not always successful at producing independent vectors [6]. The Gram-Schmidt procedure will fail on occasions when vectors, although theoretically independent, are dependent within the numerical constraints of current digital computers. These vectors cannot then be made orthogonal and independent, using the normalized Gram-Schmidt procedure described in the previous sections.

The orthogonality of the block-Krylov vectors is re-checked using the following

$$ L = Q_i^T m_{ii} Q_i \tag{49} $$

If all vectors are orthogonal and normalized with respect to the mass matrix, $L$ will be a l-size identity matrix. The mass matrix used in Eq. (49) is appropriate for fixed interface modes. For the free interface approach, the full physical mass matrix is used.

Several solutions to the numerical dependence problem are being investigated. These potential solutions include selective re-orthogonalization, the modified Gram-Schmidt procedure [6], and Kahan's orthogonalization procedure [13].

During examination of the boundary flexibility method, it was found that the Ritz vectors within the Krylov blocks were dependent on each other. In fact, it was determined that there is no theoretical basis why the vectors within the initial block should be independent. The boundary flexibility algorithm, as presented in reference [2] and discussed in section two, makes no orthogonality check of the vectors within the Krylov block. Furthermore, the subsequent Gram-Schmidt procedure is ineffective because the blocks are orthogonalized with respect to a set of vectors that are not orthogonal.

The problem of dependent vectors within the block can be eliminated by modifying the boundary flexibility algorithm. In order to make all the vectors in the solution orthogonal, each Krylov block was partitioned into its individual vectors. The Gram-Schmidt orthonormalizing procedure, shown in Eqs. (9) to (12), was then applied vector by vector to each Krylov block. The constituent vectors are then reassembled and the analysis proceeds as before. An alternate solution to the problem of dependent vectors within the blocks, is shown in reference [12]. In this solution a singular value decomposition is performed on the $q_j^* T m q_j^{**}$ subspace. The resulting similarity transformation orthogonalizes the block.

Modal Selection

A subject which requires future investigation is the selection of block-Krylov component modes. The Krylov vectors are modes, although they are static modes (as are constraint and attachment modes), in contrast to the normal modes based upon the eigenvalues. The simplest approach to modal selection is modal truncation, based upon the numerical value of the natural frequency. However, modal truncation cannot be used directly with Krylov modes, because there are no eigenvalues associated with them. Modal selection techniques more sophisticated than modal truncation will also be investigated. The algorithm currently implemented has no modal selection capability. The Krylov modes produced are the Krylov modes used, in component formation.

Example

Programming

The previously described algorithms were implemented in MSC/NASTRAN [14]. The use of a standard, commercially available computer program allows the results of this work to be easily used by other structural dynamists. Adding these methodologies to MSC/NASTRAN is allowed through the use of the internal programming language called DMAP (Direct Matrix Abstraction Programming). The standard solution sequences of MSC/NASTRAN are written in DMAP, and the source code of MSC/NASTRAN is available.
at the DMAP level. For example, Eqs. (3) to (5), and Eqs. (18) to (23) are currently contained in the standard MSC/NASTRAN normal modes solution. Equations (6) to (12) were coded using DMAP, and were then incorporated into the MSC/NASTRAN solution sequence. A DMAP listing of the present methodology is contained in the Appendix.

Definition

An example case of a cantilevered beam was derived from a finite element model of the Space Station Freedom photovoltaic array central mast. The length of the beam was 1179.9 in. The modulus of elasticity, \( E \), was \( 10.1\times10^6 \) lb/in.\(^2\) and the moment of inertia of the cross section was 108.9 in.\(^4\). Its weight per unit length was 0.2296 lb/in. The cantilevered beam was modeled with eleven nodes and 10 beam finite elements. A variety of boundary conditions, described in the next section, were imposed upon this beam.

Results

A component representation of the 10 element beam was created using the boundary flexibility method with block-Krylov iteration. The fixed interface approach, with two Krylov blocks and constraint modes, was used to form the component. The interface of the component consisted of one node and 6 degrees of freedom. The number of constraint nodes is equal to the number of interface degrees of freedom, and the size of the Krylov block is equal to the number of constraint modes. Therefore, each Krylov block contained six vectors. Since the component was formed with two Krylov blocks, it contained a total of 12 generalized coordinates.

Plots of the lateral Krylov vectors, which represented the cantilevered beam, are shown in Figs. 1 to 4. The unorthogonalised vectors, as output by Eqs. (6) and (8) are shown in Figs. 1 to 2(a) and Figs. 3 to 4(a), respectively. The first normalized vector, as output by Eq. (7), is given in Fig. 1(b). (The first vector does not need to be orthogonalized.) The remaining orthogonalized and normalized vectors, as output by Eq. (12), are given in Figs. 2 to 4(b). The first two unorthogonalized Ritz vectors (plotted in Figs. 1 to 2(a)), which are in the first Krylov block, appear to be nearly identical. The first mode is similar to the classic first bending normal mode shape of a beam and, after Gram-Schmidt orthogonalization, the second vector has become the classic second bending normal mode shape (shown in Figs. 1 to 2(b)).

The eigenvalues of the reduced Krylov subspace were calculated next. The first five natural frequencies from this reduced system are shown in Table 1. For comparison, Table 1 also includes the frequencies of a reduced system where the component was formed using traditional normal modes. This component was also formed with a fixed interface, but thirteen normal modes were used for numerical convenience. The full, or "exact," finite element eigenvalue solution is also shown. In the case of the Krylov vectors, no modal selection of any kind was used. For the case of the normal modal component, modal selection by truncation was used. The superior accuracy of the normal modal component, in the fourth bending mode, does not necessarily represent a limitation of the block-Krylov method, but instead demonstrates the need for a Krylov modal selection criteria.

In addition to the fixed interface example, two free interface examples were created. Both were based on the same 10 element beam described above, but with different boundary conditions. The first model was free-fixed, with the component interface being the free boundary condition, and hence it had no rigid-body modes. Equations (24) to (40) define the formulation of this free interface component. Table 2 shows the first five system natural frequencies of this model compared to the natural frequencies of the full finite element model. The other model consisted of the 10 element beam with free-free boundary conditions and rigid body modes. Equations (41) to (48) define the formulation of this component. Table 3 shows the first four elastic frequencies of this model compared to the frequencies of the full finite element model. In the two free interface cases, there is no comparison with the normal component mode synthesis. This is because standard free-interface MSC/NASTRAN routine does not use the "Rubin-MacNeal" method, and so a direct comparison was not performed.
Conclusions

Using boundary flexibility matrices to initialise the block-Krylov recurrence algorithm provides an efficient and simple method for generating static Ritz vectors. Static Ritz vectors so generated accurately represent the dynamics of a substructure. Because this methodology does not require the solution of the component eigenvalue problem, the component can be formed with a significant decrease in computational effort. Although the issues of vector dependency and modal selection require further investigation, the block-Krylov Component Mode Synthesis method is a promising alternative in dynamic substructuring.

References

### TABLE 1. COMPARISON BETWEEN THE CANTILEVERED BEAM FREQUENCIES USING FIXED INTERFACE COMPONENT NORMAL MODES, KRYLOV MODES, AND FINITE ELEMENTS

<table>
<thead>
<tr>
<th>System frequencies</th>
<th>From component normal modes, ( \Delta% )</th>
<th>From component Krylov modes, ( \Delta% )</th>
</tr>
</thead>
<tbody>
<tr>
<td>full finite element</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1st Bend</td>
<td>0.5464 Hz</td>
<td>0.5464 Hz</td>
</tr>
<tr>
<td>2nd Bend</td>
<td>3.415 Hz</td>
<td>3.415 Hz</td>
</tr>
<tr>
<td>3rd Bend</td>
<td>9.522 Hz</td>
<td>9.522 Hz</td>
</tr>
<tr>
<td>4th Bend</td>
<td>18.57 Hz</td>
<td>18.57 Hz</td>
</tr>
<tr>
<td>1st Tors</td>
<td>27.58 Hz</td>
<td>27.58 Hz</td>
</tr>
<tr>
<td></td>
<td>From component component normal modes, ( \Delta% )</td>
<td>From component Krylov modes, ( \Delta% )</td>
</tr>
<tr>
<td></td>
<td>0.000 Hz</td>
<td>0.000 Hz</td>
</tr>
<tr>
<td></td>
<td>3.474 Hz</td>
<td>3.474 Hz</td>
</tr>
<tr>
<td></td>
<td>9.549 Hz</td>
<td>9.549 Hz</td>
</tr>
<tr>
<td></td>
<td>18.65 Hz</td>
<td>18.65 Hz</td>
</tr>
<tr>
<td></td>
<td>30.70 Hz</td>
<td>30.70 Hz</td>
</tr>
</tbody>
</table>

### TABLE 2. COMPARISON BETWEEN THE CANTILEVERED BEAM FREQUENCIES USING FREE INTERFACE COMPONENT NORMAL MODES, KRYLOV MODES, AND FINITE ELEMENTS

<table>
<thead>
<tr>
<th>System frequencies</th>
<th>From component Krylov modes, ( \Delta% )</th>
</tr>
</thead>
<tbody>
<tr>
<td>full finite element</td>
<td></td>
</tr>
<tr>
<td>1st Bend</td>
<td>0.5464 Hz</td>
</tr>
<tr>
<td>2nd Bend</td>
<td>3.415 Hz</td>
</tr>
<tr>
<td>3rd Bend</td>
<td>9.522 Hz</td>
</tr>
<tr>
<td>4th Bend</td>
<td>18.57 Hz</td>
</tr>
<tr>
<td>1st Tors</td>
<td>27.58 Hz</td>
</tr>
<tr>
<td></td>
<td>From component component Krylov modes, ( \Delta% )</td>
</tr>
<tr>
<td></td>
<td>0.000 Hz</td>
</tr>
<tr>
<td></td>
<td>3.474 Hz</td>
</tr>
<tr>
<td></td>
<td>9.549 Hz</td>
</tr>
<tr>
<td></td>
<td>18.65 Hz</td>
</tr>
<tr>
<td></td>
<td>30.70 Hz</td>
</tr>
</tbody>
</table>

### TABLE 3. COMPARISON BETWEEN THE FREE-FREE BEAM FREQUENCIES USING FREE INTERFACE COMPONENT KRYLOV MODES, AND FINITE ELEMENTS

<table>
<thead>
<tr>
<th>System frequencies</th>
<th>From component Krylov modes, ( \Delta% )</th>
</tr>
</thead>
<tbody>
<tr>
<td>full finite element</td>
<td></td>
</tr>
<tr>
<td>Rigid Body (6)</td>
<td>0. Hz</td>
</tr>
<tr>
<td>1st Bend</td>
<td>3.474 Hz</td>
</tr>
<tr>
<td>2nd Bend</td>
<td>9.549 Hz</td>
</tr>
<tr>
<td>3rd Bend</td>
<td>18.65 Hz</td>
</tr>
<tr>
<td>4th Bend</td>
<td>30.70 Hz</td>
</tr>
<tr>
<td></td>
<td>From component component Krylov modes, ( \Delta% )</td>
</tr>
<tr>
<td></td>
<td>0.000 Hz</td>
</tr>
<tr>
<td></td>
<td>3.474 Hz</td>
</tr>
<tr>
<td></td>
<td>9.549 Hz</td>
</tr>
<tr>
<td></td>
<td>18.65 Hz</td>
</tr>
<tr>
<td></td>
<td>33.15 Hz</td>
</tr>
</tbody>
</table>
Figure 1.—Fixed interface Krylov mode, block one, vector one.

Figure 2.—Fixed interface Krylov mode, block one, vector two.

Figure 3.—Fixed interface Krylov mode, block two, vector one.

Figure 4.—Fixed interface Krylov mode, block one, vector two.
Appendix - MSC/NASTRAN DMAP Listing of the Boundary

Flexibility Method of Block-Krylov Component Synthesis

$ BOUNDARY FLEXIBILITY METHOD DMAP IMPLEMENTATION $ $ VERSION 1.0 - KELLY CARNEY $ $ IMPLEMENTED ABDALLAH'S WORK AS PUBLISHED $ $ NOTES ON USAGE $ $ - A DUMMY METHOD AND EIGR CARD MUST BE USED TO INITIATE RITZ VECTOR CALCULATION. $ $ - MODAL DEGREES OF FREEDOM MUST BE PROVIDED AND SPECIFIED ON SEQSET CARDS, AS WITH STANDARD NASTRAN SOLUTION $ $ - MIXED BOUNDARY CONDITIONS (B AND C SET POINTS) ARE NOT YET IMPLEMENTED. USE B OR C, NOT BOTH. $ $ - NOT COMPATIBLE WITH DYNAMIC REDUCTION, IF REQUESTED IT WILL NOT BE PERFORMED $ $ - AUTO-OMIT FEATURE HAS BEEN DISABLED. IF AUTOSPC IS ON, AUTO-OMIT IS NOT REQUIRED $ $ PARAMETERS $ $ PARAM,BFLEX,-1 - A NEGATIVE VALUE ON THIS PARAMETER TURNS ON THE BOUNDARY FLEXIBILITY METHOD. IF MULTIPLE SUPERELEMENTS ARE BEING PROCESSED, PLACE IN SUBCASE SECTION OF CASE Control $ $ PARAM,L,I - L IS THE NUMBER OF RITZ VECTOR DESIRED. IT IS SET BY THE NUMERICAL VALUE OF I $ $ PARAM,DIAI,1 - A POSITIVE VALUE CAUSES A LARGE AMOUNT OF MOSTLY INDESCRIPTABLE DATA TO BE PRINTED $ $ MATRICES $ $ INPUT - CMXX, CMMXX, GOAT, USET $ $ OUTPUT, FIXED INTERFACE - GOQ, $ $ OUTPUT,FREE INTERFACE - GOQ, REVISED MAA AND GOAT $ $ SETUP BOUNDARY FLEXIBILITY PARAMETER AND DISABLE DYNAMIC REDUCTION $ $ ALTER 736 $ TYPE PARM,I,Y,(BFLEX=-1) $ IF( BFLEX < 0 ) NODYNRED = -1 $ $ DISABLE AUTO-OMIT REDUCTION $ $ ALTER 737 $ IF( BFLEX < 0 ) NOARED = -1 $ $ MOVE TO BOUNDARY FLEXIBILITY SECTION IF REQUESTED $ $ ALTER 776 $ COND BOUNDFLX,BFLEX $ JUMP TO BOUNDARY FLEXIBILITY DMAP SECTION $ $ ALTER 783 $ JUMP ENDBF $ $ BEGIN BOUNDARY FLEXIBILITY METHOD SECTION $ $ LABEL BOUNDFLX $ $ CREATE AND SET INITIAL PARAMETERS $ $ TYPE PARM,I,N,(I=2) $ INC OF RITZ VEC IN FIRST SET $ $ TYPE PARM,I,N,(I=2) $ INC OF RITZ VEC IN ITH SET $ $ TYPE PARM,I,N,(I=2) $ INC OF R-SET $ $ TYPE PARM,I,N,(I=2) $ SIZE OF RITZ VEC SET
TYPE PARMS,N,(MAXRAT) $ MAXRAT FROM DECOMP OF CMKXX
TYPE PARMS,I,Y,(L-1) $ TOTAL NUMBER OF RITZ VEC
TYPE FILE,(L-1) $ DIAGONAL PRINTOUTS
FILE QP=APPEND,OVERWRITE $ ALLOW QP TO APPEND, OVERWRITE
FILE QL=APPEND $ ALLOW QL TO BE APPENDED TO
FILE QNEW=APPEND $ ALLOW QNEW TO BE APPENDED TO
FILE PRVE=OVERWRITE $ ALLOW PRVE TO BE OVERWRITTEN
FILE NPHIVR=APPEND $ ALLOW NPHIVR TO APPEND
FILE MA=OVERWRITE $ ALLOW MA TO BE OVERWRITTEN
FILE KA=OVERWRITE $ ALLOW KA TO BE OVERWRITTEN

$ CHECK TO SEE IF BOTH B AND RC HAVE BEEN DEFINED $ IF(NORC>0 AND NOBSET>0) MESSAGE / 'FATAL ERROR - BOTH BSET AND CSET OR RSET DEGREES OF FREEDOM' / 'HAVE BEEN DEFINED, ONLY ONE TYPE IS CURRENTLY ALLOWED' / $ IF(NORC>0 AND NOBSET>0) JUMP RFERR $ $ PARTITION OUT SUPPORT DEGREES OF FREEDOM, IF REQUESTED $ VEC USET/VVRO/'V'/'R'/'COMP'/$ CREATE PARTN VECTOR R-SET IF(NORSET>0) THEN $ EXECUTE ONLY IF R-SET PARTN CMKXX,VVRO,,,KZZ/-I/////S,N,SING/S,N,NBRCHG/S,N,MAXRAT/$ DECOMP KZZ/LZZ,/1/////S,N,SING/S,N,NBRCHG/ S,N,MAXRAT/$ DECOMP STIFFN S $ IF(SING=1 OR NBRCHG>0 OR MAXRAT>1.ES) MESSAGE / '/FATAL ERROR - STIFFNESS SINGULAR SUPPORT DOPS MAY BE REQUIRED' / $ IF(SING=1 OR NBRCHG>0 OR MAXRAT>1.ES) JUMP RFERR $ SINGULAR MAT DIAGONAL KZZ/LZZ/'SQUARE'/'O.$ Z-SIZE IDENTITY MAT FBS LZZ,,/KINV/$ CALCULATE K**-1 $ FOR FIXED BC METHOD $ IF(NOBSET>0) THEN $ IF EITHER R OR C SET $ MPYAD KINV,CMKXX,KINV/$ CALCULATE K**-1*M MPYAD KINV,MAO,KINVMOA/$ CALCULATE K**-1*MOA MPYAD KINVMOA/Q18/$ CALCULATE INITIAL VEC UPARTN USET,Q18/Q1A,,,/'A'/'B'/'Q'/2 $ PARTN DOWN TO B-SIZE ENDIF $ $ THIS SECTION IS EXECUTED FOR FREE INTERFACE METHOD $ IF(NORC>0) THEN $ IF EITHER R OR C SET $ THIS SECTION IS EXECUTED FOR FREE INTERFACE METHOD, $ WITH RIGID BODY MODES $ IF(NORSET>0) THEN $ ONLY IF R SET $ CREATE AND ORTHO-NORMALIZE RIGID BODY TRANSFORMATION MATRIX $ VECPLOT, BGPDTS,EQEXINS,QSTMS,USER/PPhigR/GRDPTN/'G'/$ UPARTN USET,PHIGR/PHIVR,,,/'G'/'V'/'S'/I PARTN TO R-SIZE ENDIF $ $ $ ORTHOGONALIZE REMAINING RIGID BODY VECTORS $ DO WHILE (IR <= NORSET) $ MATMOD PHIVR,,,,/PHIVR,1/1/1 $ EXTRACT 1ST COLUMN SMPYAD PHIVR1,CMKXX,PHIVR1/NPH1///S///S///S $ DIAGONAL NPH1/SQUARE/'S'/'S' SCALE FACTOR MPYAD PHIVR1,NPH1/S/PHIVR $ NORMALIZE VECTOR $ $ ORTHOGONALIZE REMAINING RIGID BODY VECTORS $ DO WHILE (IR <= NORSET) $ MATMOD PHIVR,,,,/PHIVR,1/IR/1 $ EXTRACT IRTH COLUMN SMPYAD PHIVR1,CMXKXX,PHIVR1/NPH1///S///S///S $ DIAGONAL NPH1/SQUARE/'S'/'S' SCALE FACTOR MPYAD PHIVR1,NPH1/S/PHIVR $ NORMALIZE ORTHOGONAL IR VECTOR $
SMPYAD PHIIRS,CMMXX,PHIIRS,,/NPHER/2///1///12
DIAGONAL NPHI/NPHIRS/SQUARE//-.S $$ SCALE FACTOR
MPYAD PHIIRS,NPHIRS,,/NPHER/ $ NORMALIZE VECTOR
AS
APPEND PHIIRS,,/NPHER/2 $$ ADD NEW VECTOR
$$
IR = IR + 1 $$ INCREMENT
ENDD $$ END DO LOOP
$$
FORM INERTIA RELIEF MATRIX, GE (EQNS 21,22,23 - ABDALLAH)
$$
MERGE, ,,KINV,VVRO,GC/-1/-6 $$ EXPAND KINV TO X-SIZE
DIAGONAL GC/IXX/SQUARE/0. $$ CREATE IDENTITY MAT
MPYAD CMMXX,NPHIVR,NPHIVR,IXX,IXX/AXX/-1/3///1/5 $$
$$
THIS SECTION IS EXECUTED IF FREE INTERFACE WITHOUT RIGID-BODY
$$
ELSE $$
EQUIV KINV,GE/ALWAYS $$ IF NO R-SET
ENDIF $$
$$
FOR FREE INTERFACE, RIGID BODY OR NOT
$$
VEC USET/VVCO/'V'/'C'/'COMP'/$ $ C-SET PARTN MATRIX
ADD VVRO,VVCO/VVRO///1/ $ R AND C-SET PARTN MAT
PARTN GE,VVCO//VVRG,-1/ $ PARTITION OUT COLS
MPYAD GE,CMMXX,./KINVM/$ $ CALCULATE GE
MPYAD KINVM,GVRC,./QISA/$ $ CALCULATE INITIAL VEC
ENDIF $$
$$
EXTRACT FIRST VECTOR FROM MATRIX AND NORMALIZE FOR FIXED OR FREE
$$
MATMOD QISA,,,,/QP5///1/ $ EXTRACT 1ST COLUMN
MPYAD QP5,CMMXX,QP5,,/NQ1///1/ $ $ SCALE FACTOR
MPYAD QP5,NQ15,./QP/$ $ NORMALIZE VECTOR
$$
COND PRINT1,DIAG $$
MATPRN KZE,CMMXX,GOAT,PHIGH,PHIVR/$
MATPRN NPHIVR,GC,AXX,GE,KINVM/$
PRTPARM //'/NOBSET'/1 $$
PRTPARM //'/NOTSET'//1 $$
PRTPARM //'/NORC'//1 $$
LABEL PRINT1 $$
$$
ORTHOGONALIZE FIRST SET OF VECTORS
$$
DO WHILE (11 <= NOTSET) $$
$$
MATMOD QISA,,,,/Q11I///1/ $ EXTRACT 11TH COLUMN
$$
ORTHOGONALIZE 11 VECTORS WITH PREVIOUS VECTORS
$$
MPYAD QP,CMMXX,Q11I,,/C11///1/ $ ORTHO SCALE FACTOR
MPYAD QP,C11,Q11I///-1/ $ ORTHOGONALIZE
$$
NORMALIZE ORTHOGONAL 11 VECTORS
$$
MPYAD Q11I5,CMMXX,Q11I5,,/NQ11///1/ $ SCALE FACTOR
MPYAD Q11I5,NQ115,./Q11I5/$ $ NORMALIZE VECTOR
$$
COND PRINT2,DIAG $$
PRTPARM //'/11///1 $$
MATPRN QP,Q11I,C11,Q11I5,./Q11I5/$
LABEL PRINT2 $$
$$
APPEND Q11I5,./QP/2 $$ ADD NEW VECTOR
NOTE IF NEW VECTOR IS NULL IT IS NOT APPENDED
$$
II = II + 1 $$ INCREMENT
ENDD $$ END DO LOOP
$$
INITIALIZE QL MATRIX
$$
COPY QP/QL/ALWAYS/1 $$ QL - SUM OF VECTORS

17
BEGIN DO LOOP OF SOLVING FOR SUBSEQUENT VECTORS

DO WHILE (I <= L)

CALCULATION OF NEW SET OF VECTORS

MPYAD KINVM,QP, / QIS/ (K**.1)*RITZ PREVIOUS PARAML QIS/ "TRAILER" / 1/ S, N, NOSET $ NO. COLUMNS IN QIS

MAKE NEW SET OF VECTOR ORTHOGONAL TO PREVIOUS SETS

SMYAD QL, CMXX, QIS, / CI/ S, / / I $ CALCULATE SCALE

MPYAD QL, CI, QIS / QISS $ ORTHOGONALIZE

EXTRACT FIRST VECTOR FROM NEW SET AND NORMALIZE

MATMOD QISS, / / I, QI1, / / I $ EXTRACT 1ST COLUMN

SMYAD QI1, CMXX, QI1, / / I $ DIAGONAL NQ1/ NQIS/ "SQUARE" / / S $ SCALE FACTOR

MPYAD QI1, NQIS, / QNEW $ NORMALIZE VECTOR

COND PRINT3, DIAG $ PRTPARM /// 'I' / 15 $ MATPRN QI, CI, QISS, QIJ1, QNEW $ LABEL PRINTS $ J = 2 $ RESET LOOP COUNTER

ORTHOGONALIZE NEW SET OF VECTORS WITH EACH OTHER

DO WHILE (J <= NOSET) $ MATMOD QISS, / / I, QI1, / / I $ EXTRACT JTH COLUMN

ORTHOGONALIZE JTH VECTOR WITH PREVIOUS VECTORS

SMYAD QNEW, CMXX, QI1, / / I $ ORTH SCALING

MPYAD QNEW, CI, QI1/ QISS $ ORTHOGONALIZE

NORMALIZE ORTHOGONAL JTH VECTOR

SMYAD QIJS, CMXX, QIJS, / / I $ DIAGONAL NQJS/ NQIS/ "SQUARE" / / S $ SCALE FACTOR

MPYAD QIJS, NQIS, / QISS $ NORMALIZE VECTOR

APPEND QIJS, / QNEW / 2 $ ADD NEW VECTOR

COND PRINT3, DIAG $ PRTPARM /// 'J' / 16 $ MATPRN QI1, CI, QIJS, QISS, QNEW $ LABEL PRINTS $ J = J + 1 $ INCREMENT

ENDDO $ END DO LOOP

APPEND NEW VECTOR TO VECTOR MATRIX AND COMPLETE LOOP

APPEND QNEW, / QL/ 2 $ ADD NEW VECTOR TO PREVIOUS COPY QNEW/ QP/ ALWAYS/ 1 $ RESET PREVIOUS VECTOR TO NEW

I = I + 1 $ INCREMENT

ENDDO $ END DO LOOP

FINISH PROCESSING MODE SHAPE MATRIX

IF (NORSET > 0) THEN $ IF R-SET, APPEND

APPEND NPHREV, QL, CMPHIXZ/ 1 $ RIGID BODY MODES
ELSE $ IN NO R-SET, JUST

EQUIV, QL, CMPHIXZ/ ALWAYS $ CONVERT TO NASTRAN NAME
ENDIF $ $ PERFORM Mode ORTHOGONALITY CHECK

SMYAD CMPHIXZ, CMXX, CMPHIXZ, / OCHKLL/ S / / I $ PHI-T * M * PHI

SMYAD CMPHIXZ, CMXX, CMPHIXZ, / KBAR/ S / / I $ PHI-T * K * PHI

18
TRANSFORMATION MATRIX CREATION

- BASED ON RUBIN-MACNEAL FOR FREE MODES (DOESN'T USE INREL)

ALTERNATIVE 807 $ TRANSFORM,BFLEX $ GO TO TRANSFORMATION DMAP
JUMP ETXTRANS $ JUMP AROUND IF NOT REQ.

LABEL TRANSF $ AVOID INREL MODULE

FREE BOUNDARY TRANSFORMATION

IF(NORC>0) THEN $ EXECUTE ONLY IF RC SET

CALCULATE SYSTEM FLEXIBILITY WHEN RITZ VECTOR HAVE BEEN ORTHOGONALIZED WRT THE MASS MATRIX (EQN. (15) ABDALLAH)

SMPYAD QL,CKMXX,QL,../GK/..///6 $ INTER CALC SOLVE GK1/GK1INV/3 $ INVERT INTER CALC
TRNSP QL/QLT $ TRANSPOSE QL
SMPYAD QL,GKINV,QLT,.../GK/3///6 $ CALC MODAL FLEX

CALCULATE RESIDUAL FLEXIBILITY MATRIX

NOTE: IF NO R-SET IS PRESENT THEN GE IS KINV

ADD GE,GK/GD/-1.0/0/ $ GD = GE - GK

PARTITION DOWN SQUARE FLEXIBILITY MATRIX TO C OR BOUNDARY DOFS

PARTN GD,VVRCO,/GCRCD,GORCD,../-1/6 $ GD = (GCRCD/GORCD)

FORM FREE "CONSTRAINT" MODE MATRIX, WHICH IS THE LOWER LEFT PARTITION OF THE TRANSFORMATION MATRIX

SOLVE GCRCD,/GCCDINV/3 $ INVERT GCCD
MPYAD GCRCD,GCCDINV,/=GOATFRS/ $ GOAT = GCCD*GCCD**-1

FORM FREE "MODAL" MATRIX, WHICH IS THE LOWER RIGHT PARTITION OF THE TRANSFORMATION MATRIX

PARTN CMPHIXZ,VVRCO/QRL,QOL,../1/ $ CMPHIXZ = (QRL/QOL)
MPYAD GOATFRS,QRL,QOL/PHIVZ/-1/ $ PHIVZ=QOL-GOAT*QRL

PACK GOATFRS WITH ZEROS TO MAKE IT A-SIZE

THE LOWER LEFT PARTITION OF THE TRANSFORMATION MATRIX

VEC USET/VQRC/A'/Q''/COMP/ $ FORM PART VECTOR MERGE, "GOATFRS",VQRC,/GOATFREE/1/ $ MAKE A-SIZE

FORM REPLACEMENT BOUNDARY MASS AND STIFFNESS MATRIX, THIS TRANSFORMATION IS THE SIMILAR TO STANDARD CMS AND SOL 43,

EXCEPT FOR FREE, REVISED 'GOAT' MATRIX IS NOT EQUAL TO CONSTRAINT MASSES, SO STIFFNESS TRANSFORMATION IS SAME AS MASS

MPYAD MGG,GOATFREE,MOA/MAA1FR/ $ IDENTICAL TO $20
MPYAD MGG,GOATFREE,MAA1/MAA2FR/1/ $ INTERMEDIATE

MPYAD GOATFREE,MOA1FR,MAA2FR/MAA3///6 $ NEW MASS MATRIX

MPYAD KGG,GOATFREE,KA1/KA11FR/ $ STIFFNESS SECTION
MPYAD KGG,GOATFREE,KA11/KA12FR/1/ $ INTERMEDIATE

MPYAD GOATFREE,KA11FR,KA12FR/KA13///6 $ NEW KAA MATRIX

REPLACE PREVIOUS GOAT WITH FREE ONE CALCULATED HERE

EQUIV GOATFREE,GOAT/ALWAYS/ $ ENDIF $ $ MAKE PHIVZ Q-SIZE, THIS REPLACES INREL FOR FREE AND FIXED

BOUNDARY FLEXIBILITY METHODS

IF (NOSET > NOQSET) MESSAGE /
FATAL ERROR: NUMBER OF MODES EXCEEDS Q-SET DOF

IF (NOZSET > NOQSET) JUMP RFERR

MATGEN /SQ/6/NOQSET/NOQSET $ CREATE PARTITIONING MATRIX
MERGE PHIVZ,...,ZQ,/GOQ/1 $ TO MAKE PHIV Z-SIZE (GOQ)

COND PRINT5,DIAG $MATPRN CMPHIXZ,GK1,GK1INV,GK,GD/$
MATPRN GCRD,GORCD,GCCDINV,GOATFRS,QRCL/$
MATPRN QO1,PHIVZ,GOATFREE,MAA,KAA/$
LABEL PRINT6 $MATPRN $LABEL ETRANSF $SETUP INTERMEDIATE MODAL STIFFNESS MATRIX
ALTER 816,816 $UMERGE1 USET,KQQ,.../KLA1/'A'/"Q"/'T'/'0$ $CREATE OFF-DIAGONAL MODAL STIFFNESS TERMS
ALTER 818 $IF(NORC>0 AND BFLEX<0) THEN $MPYAD K00,GOAT,KOA/KOA1 $MPYAD G0A,KOA1/KQT/1 $MPYAD KQT/KAAQ $ADD KQT,KAAQ/KAAQ $ELSE $EQUIV KLA1,KLAA/ALWAYS $ENDIF $ALTER 859 $COND PRINT,DIAG $MATPRN GOQ,KLAA,MLAA/$LABEL PRINT$ $ENDALTER $
**Title and Subtitle:** Implementation of the Block-Krylov Boundary Flexibility Method of Component Synthesis

**Authors:** Kelly S. Carney, Ayman A. Abdallah, Arthur A. Hucklebridge

**Performing Organization Name(s) and Address(es):**
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135–3191

**Sponsoring/Monitoring Agency Name(s) and Address(es):**
National Aeronautics and Space Administration
Washington, D.C. 20546–0001

**Supplementary Notes:**

**Abstract:**
A method of dynamic substructuring is presented which utilizes a set of static Ritz vectors as a replacement for normal eigenvectors in component mode synthesis. This set of Ritz vectors is generated in a recurrence relationship, which has the form of a block-Krylov subspace. The initial seed to the recurrence algorithm is based on the boundary flexibility vectors of the component. This algorithm is not load-dependent, is applicable to both fixed and free-interface boundary components, and results in a general component model appropriate for any type of dynamic analysis. This methodology has been implemented in the MSC/NASTRAN normal modes solution sequence using DMAP. The accuracy is found to be comparable to that of component synthesis based upon normal modes. The block-Krylov recurrence algorithm is a series of static solutions and so requires significantly less computation than solving the normal eigenspace problem.