EXTRACTION OF EIGEN-PAIRS FROM BEAM STRUCTURES USING AN
EXACT ELEMENT BASED ON A CONTINUUM FORMULATION AND
THE FINITE ELEMENT METHOD

by

J. Jara-Almonte

Dissertation submitted to the Faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Mechanical Engineering

APPROVED:

L. D. Mitchell, Chairman

C. Beattie

N. S. Eiss

R. H. Fries

C. E. Knight

R. G. Leonard

October, 1985

Blacksburg, Virginia
EXTRACTION OF EIGEN-PAIRS FROM BEAM STRUCTURES USING AN EXACT ELEMENT BASED ON A CONTINUUM FORMULATION AND THE FINITE ELEMENT METHOD

by

James Jara-Almonte

Committee Chairman: Larry D. Mitchell
Mechanical Engineering

(ABSTRACT)

Studies of numerical methods to decouple structure and fluid interaction have reported the need for more precise approximations of higher structure eigenvalues and eigenvectors than are currently available from standard finite elements. The purpose of this study is to investigate hybrid finite element models composed of standard finite elements and exact-elements for the prediction of higher structure eigenvalues and eigenvectors.

An exact beam-element dynamic-stiffness formulation is presented for a plane Timoshenko beam with rotatory inertia. This formulation is based on a converted continuum transfer matrix and is incorporated into a typical finite element program for eigenvalue/vector problems. Hybrid models using the exact-beam element generate transcendental, nonlinear eigenvalue problems. An eigenvalue extraction technique for this problem is also implemented. Also presented is a post-processing capability to reconstruct the mode shape each of exact element at as many discrete locations along the element as desired.

The resulting code has advantages over both the standard transfer matrix method and the standard finite element method. The advantage
over the transfer matrix method is that complicated structures may be
modeled with the converted continuum transfer matrix without having to
use branching techniques. The advantage over the finite element method
is that fewer degrees of freedom are necessary to obtain good
approximations for the higher eigenvalues. The reduction is achieved
because the incorporation of an exact-beam-element is tantamount to the
dynamic condensation of an infinity of degrees of freedom.

Numerical examples are used to illustrate the advantages of this
method. First, the eigenvalues of a fixed-fixed beam are found with
purely finite element models, purely exact-element models, and a closed-
form solution. Comparisons show that purely exact-element models give,
for all practical purposes, the same eigenvalues as a closed-form
solution. Next, a Portal Arch and a Verdeel Truss structure are modeled
with hybrid models, purely finite element, and purely exact-element
models. The hybrid models do provide precise higher eigenvalues with
fewer degrees of freedom than the purely finite element models. The
purely exact-element models were the most economical for obtaining
higher structure eigenvalues. The hybrid models were more costly than
the purely exact-element models, but not as costly as the purely finite
element models.
ACKNOWLEDGEMENTS

First and foremost I would like to thank my wife

for her support and encouragement throughout the pursuit of this
degree. I must also thank my son , who with his toothless
smile, taught me to place things in their proper perspective, and assign
them their proper priority. I am also indebted to my advisor Larry D.
Mitchell for his patience throughout this study and the several drafts
of this dissertation.

There are others, too numerous to list, people who have assisted me
these past three years with their help and/or guidance. I feel that
there are a few who I must mention. First, Dr. Chris Beattie who guided
me in the way of handling poles. Second, Drs. M.P. Kamat, and W.L.
Hallauer for sharing their computer codes with me. Third, Mr. E. Sitler
and Ms. Z. Terry who got paperwork through the university bureaucracy.
Fourth, Dr. C. E. Knight for his comments throughout this work.

Last, but not least I would like to thank the Union Carbide
Corporation for their financial support during my graduate education.
It was an honor for me to become a Union Carbide Fellow during my M. S.
work. Without their continued financial support it may have been
impossible to pursue this degree immediately after my M. S..
# TABLE OF CONTENT

<table>
<thead>
<tr>
<th>Section</th>
<th>page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>ii</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>iv</td>
</tr>
<tr>
<td>List of Figures</td>
<td>v</td>
</tr>
<tr>
<td>List of Tables</td>
<td>vi</td>
</tr>
<tr>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Literature Review</td>
<td>5</td>
</tr>
<tr>
<td>A. The Standard Finite Element Eigenvalue Problem</td>
<td>5</td>
</tr>
<tr>
<td>B. The Exact Method (Dynamic Stiffness Method)</td>
<td>10</td>
</tr>
<tr>
<td>C. The Finite Element-Transfer Matrix Method</td>
<td>14</td>
</tr>
<tr>
<td>Theory</td>
<td>20</td>
</tr>
<tr>
<td>A. The Transfer Matrix Method</td>
<td>20</td>
</tr>
<tr>
<td>B. The FEM Beam Element</td>
<td>26</td>
</tr>
<tr>
<td>C. The Increment-Falsi-Bisection Method</td>
<td>29</td>
</tr>
<tr>
<td>Program Description</td>
<td>31</td>
</tr>
<tr>
<td>Numerical Examples</td>
<td>43</td>
</tr>
<tr>
<td>A. Fixed-Fixed Beam</td>
<td>43</td>
</tr>
<tr>
<td>B. Portal Arch</td>
<td>54</td>
</tr>
<tr>
<td>C. Verdeel Truss</td>
<td>75</td>
</tr>
<tr>
<td>Conclusions</td>
<td>99</td>
</tr>
<tr>
<td>Future Work</td>
<td>104</td>
</tr>
<tr>
<td>References</td>
<td>108</td>
</tr>
<tr>
<td>Appendix A - Dynamic Pointers in DSTAP</td>
<td>113</td>
</tr>
<tr>
<td>Appendix B - Program DSTAP</td>
<td>119</td>
</tr>
<tr>
<td>Appendix C - Instructions for Using and Modifying DSTAP</td>
<td>160</td>
</tr>
<tr>
<td>Appendix D - Sample Runs using Program DSTAP</td>
<td>167</td>
</tr>
<tr>
<td>Vita</td>
<td>178</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Content</th>
<th>page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Schematic and Notation for a Plane Beam in the TMM</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>Typical Periodic Structure that May Be analyzed with FETM</td>
<td>17</td>
</tr>
<tr>
<td>3</td>
<td>Positive Sign Conventions in the FEM and TMM</td>
<td>24</td>
</tr>
<tr>
<td>4</td>
<td>Massless Elastic Beam with Lumped Mass Transfer Matrix</td>
<td>27</td>
</tr>
<tr>
<td>5</td>
<td>Flowchart of Original STAP Program</td>
<td>32</td>
</tr>
<tr>
<td>6</td>
<td>Flowchart of Dynamic Portion of Program DSTAP</td>
<td>34</td>
</tr>
<tr>
<td>7</td>
<td>Schematic of the Beam for Example A</td>
<td>44</td>
</tr>
<tr>
<td>8</td>
<td>Models Used in Example A</td>
<td>51</td>
</tr>
<tr>
<td>9</td>
<td>Schematic of the Portal Arch</td>
<td>55</td>
</tr>
<tr>
<td>10</td>
<td>Mode Shapes for the Portal Arch</td>
<td>58</td>
</tr>
<tr>
<td>11</td>
<td>Number of Acceptable Eigenvalues versus Degrees of Freedom</td>
<td>68</td>
</tr>
<tr>
<td>12</td>
<td>CPU seconds per Acceptable Eigenvalue versus Model Size</td>
<td>73</td>
</tr>
<tr>
<td>13</td>
<td>Schematic of the Structure for Example C</td>
<td>76</td>
</tr>
<tr>
<td>14</td>
<td>Verdeel Truss Mode Shapes</td>
<td>80</td>
</tr>
<tr>
<td>15</td>
<td>Proposed 6&lt;sup&gt;th&lt;/sup&gt; and 7&lt;sup&gt;th&lt;/sup&gt; Modes (After Ref. [31])</td>
<td>84</td>
</tr>
<tr>
<td>16</td>
<td>Acceptable Eigenvalues versus Model Size for the Verdeel Truss</td>
<td>91</td>
</tr>
<tr>
<td>17</td>
<td>Execution Time versus Model Size for the Verdeel Truss</td>
<td>96</td>
</tr>
<tr>
<td>D-1</td>
<td>DSTAP Input for the Two-Exact-Element Model</td>
<td>168</td>
</tr>
<tr>
<td>D-2</td>
<td>DSTAP Output for the Two-Exact-Element Model</td>
<td>169</td>
</tr>
<tr>
<td>D-3</td>
<td>DSTAP Input for the Four—Exact-Element Model</td>
<td>173</td>
</tr>
<tr>
<td>D-4</td>
<td>DSTAP Output for the Four—Exact-Element Model</td>
<td>174</td>
</tr>
</tbody>
</table>
### LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Content</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Transfer Matrices for a Timoshenko Beam with Rotatory Inertia and Axial Deformation.</td>
<td>22</td>
</tr>
<tr>
<td>2</td>
<td>First 20 Roots of Eq. (27) Obtained in Quad-precision With a VAX BASIC Program.</td>
<td>46</td>
</tr>
<tr>
<td>3</td>
<td>Comparison Between Eigenvalues from Close-Form Solution and an Exact Element Model.</td>
<td>48</td>
</tr>
<tr>
<td>4</td>
<td>Comparison Between DSTAP and SUPERB Eigenvalue Extractions.</td>
<td>53</td>
</tr>
<tr>
<td>5</td>
<td>Portal Arch Exact Model Eigenvalues, With and Without Shear Deformation and Rotatory Inertia.</td>
<td>57</td>
</tr>
<tr>
<td>6</td>
<td>Portal Arch Finite Element Model Eigenvalues.</td>
<td>62</td>
</tr>
<tr>
<td>7</td>
<td>Portal Arch Combined Model Eigenvalues.</td>
<td>70</td>
</tr>
<tr>
<td>8</td>
<td>Exact Model Eigenvalues, With and Without Shear Deformation and Rotatory Inertia.</td>
<td>78</td>
</tr>
<tr>
<td>9</td>
<td>DSTAP FEM Eigenvalues using Subroutine EIGENR.</td>
<td>86</td>
</tr>
<tr>
<td>10</td>
<td>DSTAP FEM Eigenvalues using Subroutine EIGZF.</td>
<td>87</td>
</tr>
<tr>
<td>11</td>
<td>SUPERB FEM Eigenvalues.</td>
<td>92</td>
</tr>
<tr>
<td>12</td>
<td>DSTAP combined Model Eigenvalues.</td>
<td>93</td>
</tr>
<tr>
<td>A-2</td>
<td>Distribution of the Dynamic Storage Area (A-array) During The Exact Beam Element Matrix Assembly Process.</td>
<td>114</td>
</tr>
<tr>
<td>A-3</td>
<td>Distribution of the Dynamic Storage Area (A-array) During An Eigenvalue Extraction using EIGENR.</td>
<td>115</td>
</tr>
<tr>
<td>A-4</td>
<td>Distribution of the Dynamic Storage Area (A-array) During The Search for Poles.</td>
<td>115</td>
</tr>
<tr>
<td>A-5</td>
<td>Distribution of the Dynamic Storage Area (A-array) During The Execution of Subroutine BFMTHD.</td>
<td>116</td>
</tr>
<tr>
<td>A-6</td>
<td>Distribution of the Dynamic Storage Area (A-array) During The Post-Processing of Eigenvectors with Exact Elements.</td>
<td>117</td>
</tr>
</tbody>
</table>
INTRODUCTION

The finite element method (FEM)\(^+\) has become a commonly accepted problem solving tool in many different fields of study. The basic premise behind this method is the ability to approximate a continuous solution by a set of piecewise continuous functions over the same domain. Hence, the large domain is broken down to a set of small regions called elements and the governing differential equations of the entire system are solved over each element. Theoretically, as the elements become infinitesimally small, the solution obtained with the finite element method will converge on the solution obtained by solving the differential equations of the continuous system.

The idea of a finite element solution, the approximation of the continuous solution by piecewise continuous functions, was first postulated by Courant [1]* in 1943. He presented the solution to a continuum problem by using continuous functions defined only over finite triangular regions. The union of all the triangular regions represented the solution to the continuum problem. At the time, Courant's work could not be applied directly to engineering applications. The solution of the large system of simultaneous equations needed to model engineering problems was intractable by hand calculations.

It was not until 1956 that the finite element method came into the realm of applicable mathematics, thanks to second-generation computers.

\(^+\) Acronyms in parentheses will be used in the remainder of this dissertation to represent the phrase preceding the parentheses.

* Numbers in square brackets refer to references at the end of this dissertation.
The paper by Turner, et al [2] is considered the cornerstone in the development of the finite element method. This work presented two elements for the analysis of aircraft structures, a truss element and an inplane load plate element. However, the nature of the method presented by Turner, et al is such that several boundary value problems can be solved:

a) equilibrium problems (static, time-independent problems),
b) eigenvalue problems, and
c) transient or propagation problems (time dependent).

The work to be presented in the next sections deals with the second class of problems, the eigenvalue problems.

Since the publication of the Turner et al paper, the number of applications for the FEM has steadily increased through the years, as evidenced by the hundreds of studies published each year either on the FEM or having used the FEM. Recent work has tried to couple the effects of fluids on structures [3-5]. This coupling created a set of nonlinear simultaneous equations. One way to linearize the set of equations is to use modal analysis on the fluid and structure. For this linearization to provide acceptable results, fairly precise higher modes of the structure are required. Unfortunately, the standard eigenextraction algorithms in the FEM are not as effective in extracting higher eigenvalues as they are in extracting lower eigenvalues [6].

The objective of this work was to explore an enhancement to the FEM that would allow a more accurate extraction of higher eigenvalues and eigenvectors. The FEM is an entrenched analytical tool in structural analysis. Therefore, an enhancement to the FEM would allow practicing engineers to use it without much retraining. The enhancement to be
studied is the incorporation of a so-called "exact" element to the FEM library. Thus, the enhancement is in reality a hybrid method. Work leading up to this hybrid method is presented in the Literature Review section.

An "exact" element arises whenever the boundary value problem is solved exactly over the element domain, and not by approximating functions as proposed by Courant. In particular, the element studied here is a beam. The development of an "exact" beam element may be obtained directly from continuum mechanics equations. However, in this study the equations for the "exact" beam element are obtained from transforming a continuum transfer matrix representation of a beam into an equivalent dynamic stiffness matrix. This route was chosen because the transfer matrix method is also commonly used by practicing engineers. Furthermore, extensive libraries of continuum transfer matrices have been developed through the years. Practically anyone of these continuum transfer matrices may be converted to a dynamical stiffness matrix and added to the library of a FEM program. The transformation technique employed is presented in the Theory section, along with the FEM beam element matrices.

The addition of an "exact" element to a FEM model makes the resulting eigenvalue problem nonlinear. Therefore, the models using "exact" elements require a special eigenvalue extraction if simplifying assumptions are to be avoided. A rather crude eigenvalue extraction method is presented in the Program Description section. Although this extraction method is inefficient, it is still possible to obtain the higher eigenvalues at a reduced computational cost, for the examples presented here.
Three examples will be presented in the Numerical Examples section, a fixed-fixed beam, a Portal Arch and a Verdeel Truss. In the fixed-fixed beam example, the accuracy of the exact element is established empirically by comparing it with a closed-form solution. Also in this example, the finite element beam formulation used is compared against a commercial code. In the second and third examples, the Portal Arch and Verdeel Truss, the structures are first modeled with purely FEM beam elements and purely exact elements, to establish how many finite element beams are required to produce acceptable eigenvalues. Then, the structures are modeled with a combination of exact and finite element beams. The results from these hybrid models are compared against the purely finite element results and the exact element results to establish the accuracy and usefulness of the hybrid model.
LITERATURE REVIEW

This section is a summary of the recent work leading up to the proposed enhancement of the FEM eigenvalue solution. This enhancement is the incorporation of a continuum transfer matrix into a standard FEM set of equations. Therefore, recent work in the standard FEM eigenvalue formulation, the exact displacement eigenvalue problem, and the finite element-transfer matrix method is reviewed. Work in each one of these areas is presented below, preceded by a brief introduction to each area.

A. The Standard Finite Element Eigenvalue Problem:

The standard FEM eigenvalue problem in structural analysis is expressed as \([K]d = \omega^2[M]d\), where, \([K]\) and \([M]\) are, respectively, the global stiffness and mass matrices, \([d]\) is the nodal displacement vector, and \(\omega\) is the circular frequency. The mass and stiffness matrices are assembled from element matrices, which are dependent on the element formulation (discretization scheme) chosen by the analyst. The common way to formulate these matrices is through either variational methods, such as the Rayleigh-Ritz method; through weighted residual methods, such as the Galerkin method; or the least squares methods [6-11]. The variational formulation yields a functional that characterizes the solution as being stationary at the eigenvalues. For example, if Rayleigh's quotient is chosen as the functional, then the stationary values of Rayleigh's quotient:

\[
R = \frac{d^T[K]d}{d^T[M]d} \tag{1}
\]
would be in the vicinity of the natural frequencies of the continuous structure approximated by $[K]$ and $[M]$.

Recent work in the FEM has tried to take into account the effect of fluids upon structures and vice-versa [3-5]. This type of problem generates nonlinear equations. Rather than solving complicated nonlinear equations Sung and Nefske [3] linearized the problem through modal analyses of the structure and the fluid independently. For this method to provide accurate results, fairly precise higher modes of the structure are required. Sung and Nefske found that the extraction of higher eigenvalues and eigenvectors in large structures was a difficulty for their linearization scheme. Using NASTRAN [12], they were unable to obtain sufficiently accurate eigenvalues beyond 110 Hz to provide satisfactory results from their linearization scheme.

Strang and Fix [13] have quantified the deviation between the eigenvalues of a large FEM discretization ($\lambda_i$) from those of the continuous system ($\lambda_i$). Let the continuous system eigenvalue problem be expressed by,

$$L \mathbf{u} = \lambda \mathbf{m} \mathbf{u}, \quad \mathbf{u} \in \mathbf{S}$$

(2)

where, $L$ is a linear, homogeneous, self-adjoint differential operator of order $2p$, $\mathbf{u}$ is the displacement function, $\mathbf{m}$ is the mass per unit length, and $\mathbf{S}$ is the domain over which Eq. (2) applies. The boundary constraints have not been included, but it is understood that they are needed for the solution of Eq. (2). In a variational solution to Eq. (2) the admissible or energy functions would belong to a space $\mathbf{K}^p$; that is, they would need to be at least $p$ times differentiable. If the continuous system described by Eq. (2) is discretized by a FEM model with a
large number of degrees of freedom, \( n \), and with interpolation function of order \( k \), then Strang and Fix have proven that,

\[
\lambda_i - \lambda_i = c \left( \frac{1}{n} \right)^{(k-p)}, \quad i=1, 2, 3, \ldots
\]  

(3)

where, \( c \) is a real constant. Thus, theoretically the eigenvalues of a FEM model should approximate the actual eigenvalues as \( n \) goes to infinity. Although it is reassuring to know that FEM models are guaranteed to converge on the true eigenvalues of a continuous system, it is not feasible nor desirable to have extremely large models. Large FEM models require more storage space, more computation time, and have increased computational truncation errors.

Since it became obvious that large models were less desirable, much work has been done to reduce the number of degrees of freedom of FEM models. One of the first, and presently more popular, techniques [14, 15] to reduce degrees of freedom is known in the static structural analysis area as "substructuring", "partitioning", or "nodal condensation". This technique was introduced by Kron [16, 17] in the late 1950's to condense out nodes in electrical networks. Kron's Diakoptics technique was adapted to static structural analysis in the early 1960's [18-21], and used to develop so-called "mixed methods" [22, 23]. In these methods, the substructure may be analyzed by the force method, while the rest of the structure is analyzed by the displacement method. Later, in 1965, Guyan [24] and Irons [25] extended the technique to dynamic structural analysis. This technique is commonly referred to as "dynamic substructuring", and "dynamic condensation".

Suppose there is a structure of \( (m+s) \) degrees of freedom. If there is a distinct substructure of \( s \) degrees of freedom then the standard FEM
The eigenvalue problem can be rewritten as,

\[
\begin{bmatrix}
K_{mm} & K_{ms} \\
K_{ms}^T & K_{ss}
\end{bmatrix}
\begin{bmatrix}
d_m \\
d_s
\end{bmatrix} = \omega^2 \begin{bmatrix}
M_{mm} & M_{ms} \\
M_{ms}^T & M_{ss}
\end{bmatrix}
\begin{bmatrix}
d_m \\
d_s
\end{bmatrix}
\]  

(4)

where, \(K_{mm}, M_{mm}\) are the master stiffness and mass matrices,
\(K_{ss}, M_{ss}\) are the slave stiffness and mass matrices,
\(K_{ms}, M_{ms}\) are the coupling mass and stiffness matrices,
\(d_m\) is the vector of master degrees of freedom, and
\(d_s\) is the vector of slave degrees of freedom.

If the second set of equations is solved for \(d_s\), then,

\[
d_s = -(K_{ss} - \omega^2 M_{ss})^{-1}(K_{ms} - \omega^2 M_{ms})^T d_m
\]  

(5)

provided \((K_{ss} - \omega^2 M_{ss})\) is nonsingular. When Eq. (5) is substituted back into the first set of equations, in Eq. (4), the result is,

\[
[(K_{mm} - \omega^2 M_{mm}) - (K_{ms} - \omega^2 M_{ms})(K_{ss} - \omega^2 M_{ss})^{-1}(K_{ms} - \omega^2 M_{ms})^T]d_m = 0
\]  

(6)

The order of the eigenvalue problem in Eq. (6) is described by the number of master degrees of freedom, \(m\). The remaining \(s\) degrees of freedom have been eliminated from the eigenvalue problem. These \(s\) degrees of freedom, however, can be reconstructed from Eq. (5). Because the reconstruction of the \(s\) degrees of freedom depends on the \(m\) degrees of freedom, the former are usually called the "slaves" and the latter the "masters".

The reduction of the size of the matrices in Eq. (6) is accompanied by a more complicated eigenvalue extraction, since the reduced system of equations does not resemble an algebraic eigenvalue problem. The second matrix in Eq. (6) contains frequency dependent terms which are off the main diagonal. For this reason several methods have been presented in the literature to facilitate the solution of the eigenvalue problem in
Eq. (6), by eliminating these off-diagonal, frequency-dependent terms.

In his original paper, Irons [25] argued that if the slaves were chosen such that they were the ones with the smallest inertia forces then the terms \( \omega^2 M_{ss} \) could be set to zero. Furthermore, if the FEM discretization generates a diagonal mass matrix \( (M_{ms} = 0) \), then Eq. (6) can be rewritten as,

\[
[K_{mm} - K_{ms}K_{ss}^{-1}K_{ms}^T] - \omega^2 M_{mm} \quad \ddot{d}_m = 0
\]

This reduction of Eq. (6) is commonly used in practice as evidenced by Rao [11] and commercial codes such as ANSYS [14] and SUPERB [15]. This technique is usually referred to as the "Guyan Reduction" or "Static Condensation". This second name arises from the fact that Eq. (7) can also be obtained if \( \omega \) is set to zero for the slaves. Equations (7) can also be obtained if the slave degrees of freedom are considered massless. In this case, Meirovitch [6] points out that Eq. (6) is in reality a fourth order polynomial in \( \omega \), whereas Eq. (7) is only a second order polynomial. Thus, a static condensation is tantamount to discarding the higher order terms in \( \omega \). This is possible as long as the coefficients of \( \omega^4 \) are small. They are only small whenever the slaves are chosen from areas of the structure with large stiffness and low mass.

To improve the accuracy of the dynamic substructuring some attempts [26, 27] have been made to take into account the higher order terms in \( \omega \). The increased accuracy of the approximations made to solve Eq. (6) were offset by the increased computational effort. Thus, it proved more efficient to establish guidelines for choosing master and slave degrees
of freedom. Henshell and Ong [28] devised an automatic selection of the masters and Thomas [29] developed an approximate error bound for the eigenvalues based on these selected masters.

B. The Exact Method (Dynamic Stiffness Method):

Dynamic (dynamical) stiffness matrices are matrices whose elements contain transcendental functions. These matrices arise in structural analysis whenever a structure is modeled by continuous elements using the displacement method. Continuous elements are often called the exact elements since the boundary value problem is solved exactly over each element without a-priori assumptions of the shape functions (interpolation functions) and without the use of admissible functions. This is why the library of exact elements is limited to trusses, rods in torsion, beams, and simple plate geometries [30].

The free-vibration problem in the exact method is expressed as,

\[ [K(\omega)] \mathbf{d} = \mathbf{0} \]  

(8)

where, \([K(\omega)]\) is the dynamic stiffness matrix whose elements are functions of the circular frequency \(\omega\), and \(\mathbf{d}\) is the displacement vector of the juncture points at which elements meet. Theoretically, Eq. (8) is the limit of Eq. (6) when the number of slave degrees of freedom \(s\) tends toward infinity. In Eq. (3) it was seen that the eigenvalues of the discretized model will converge on the eigenvalues of the continuous system as \(n\) goes to infinity. It follows then that if \(s\) is allowed to go to infinity then Eqs. (6) and (8) will give the same eigenvalues. For this reason, Eq. (8) may be viewed as a substructure technique in which an infinity of degrees of freedom have been condensed out.

time an exact element is used.

The eigenvalue problem in Eq. (8) is a nonlinear eigenvalue problem. This precludes the use of algorithms developed for the linear eigenvalue problem, such as those incorporated into commercial FEM codes [12, 14, 15]. Therefore, it is desirable to linearize Eq. (8) in some way while still retaining the essence of the eigenvalue problem. Several linearization attempts have been studied [31-34]. Richards and Leung [34] proved that a linearization of the form,

\[(BA - \omega^2PA)d = 0\]  \hspace{1cm} (9)

where, \(BA = [K(\omega_A)] + \omega_A^2PA\),

\[PA = \frac{1}{2\omega} \left[ \frac{\partial[K(\omega)]}{\partial\omega} \right] \]

\(\omega = \omega_A\)

\(\omega_A\) is an arbitrary frequency,

will generate positive definite matrices \(BA\) and \(PA\) as long as the system being modeled is constrained against rigid-body motion. Positive definite matrices are desirable since there are efficient algorithms for the extraction of eigenvalues from a system of positive definite matrices [6]. In Eq. (9), the trial circular frequency \(\omega_A\) will be equal to \(\omega\) only at an eigenvalue of Eq. (8). Hopper, Simpson and Williams [32] converted Eq. (9) to a form better suited for computer implementation by evaluating the partial derivative in Eq. (9) at two trial frequencies \(\omega_1\) and \(\omega_2\). This linearization gave excellent results by providing bounds on all the eigenvalues below the first pole of Eq. (8), as long as \(\omega_1\) and \(\omega_2\) are also below the first pole. Unfortunately, Hopper et al were unable to establish any bound criteria above the first pole of Eq. (8).
A pole of Eq. (8) is a nonzero frequency at which Eq. (8) is satisfied and \( \det[K(\omega)] \) is either indeterminate or nonzero. Since \( \det[K(\omega)] \neq 0 \), Eq. (8) can only be satisfied if \( d=0 \); that is, when the juncture points of the continuous element model are stationary. This does not imply that the structure is stationary, it means the connection points are fixed while the rest of the structure vibrates. In other word, the juncture points become the vibration nodes (not nodes in the FEM sense) of the mode shape. Since these poles satisfy Eq. (8) they are in reality system eigenvalues. They will not, however, be called eigenvalues here, since the equivalent linear eigenvalue problem is not satisfied at these frequencies. The nontrivial solution of a linear eigenvalue problem of positive definite matrices must have \( \det[K - \omega^2M] = 0 \), and \( d \neq 0 \), at an eigenvalue.

In the FEM dynamic substructuring technique the slave degrees of freedom correspond to a physical substructure. The eigenvalue problem for this substructure alone is \( (K_{ss} - \omega^2M_{ss})d_s=0 \), where the terms have been defined in Eq. (4). Naturally, a substructure eigenvalue will make \( (K_{ss} - \omega^2M_{ss}) \) singular. In the exact displacement formulation, the poles of Eq. (8) correspond to the frequencies which make \( (K_{ss} - \omega^2M_{ss}) \) singular. For this reason, the poles of an exact displacement formulation are also referred to as substructure eigenvalues.

Another way of solving Eq. (8) is through frequency scanning methods. References [8-11] outline the frequency search method for the linear eigenvalue problem \( [K-\omega^2M]d=0 \). They recommend that the eigenvalues be counted using the Sturm Sequence Theorem, since there is the possibility of missing eigenvalues in a frequency scan. Wittrick and
Williams [35, 36] have modified the Sturm Sequence theorem for a nonlinear eigenvalue problem to take into account the infinity of eigenvalues contained in Eq. (8). Their formula can determine the number of structure eigenvalues below an arbitrary frequency \( \omega_A \),

\[
J(\omega_A) = J_0(\omega_A) + s[K(\omega_A)]
\]

(10)

where, \( J(\omega_A) \) is the number of structure eigenvalues below the chosen \( \omega_A \), \( J_0(\omega_A) \) is the number of poles lying below \( \omega_A \), \( s[K(\omega_A)] \) is the sign count of the dynamic stiffness matrix in Eq. (8). The sign count is the number of negative terms along the main diagonal of \( [K(\omega_A)] \) after it has been upper-triangularized, without pivoting. This counting technique has been used in several studies [37-39], where a system of equations similar to Eq. (8) had to be solved.

Equation 10 has some interesting consequences. For example, suppose a cantilever beam is to be analyzed with one exact beam element for in-plane vibration. Such a model has 3 active degrees of freedom at the free end. They are the longitudinal displacement, the transverse displacement, and the rotation about the neutral axis. Thus, the order of \( [K(\omega)] \) is 3, and at most there will be 3 eigenvalues below the first pole of \( \det[K(\omega)] \). Supposing that there are 3 eigenvalues in this range, then between two poles there can be at most 1 eigenvalue. This latter conclusion, that poles must sandwich eigenvalues, is easily verified by considering the characteristic equations for the eigenvalues and poles. According to Refs. [40, 41], the eigenvalues of a simple cantilever beam are given by the roots of,

\[
\cos(\beta \ell) \cosh(\beta \ell) = -1
\]

(a)

where, \( \beta^* = \omega^2 \mu/(EI) \).
μ is the mass per unit length,
E is the modulus of elasticity,
I is the cross-section second area moment of inertia.

The poles of \( \text{det}[K(\omega)] \) are given by the roots of,

\[
\cos(\beta \ell) \cosh(\beta \ell) = 1
\]

Since the \( \cosh \) function is never less than 1, then between two roots of Eq. (a), eigenvalues, there must be a root of Eq. (b), a pole of \( \text{det}[K(\omega)] \).

C. The Finite Element-Transfer Matrix Method:

The finite element-transfer matrix (FETM) method is a hybrid method as its name implies. The method uses both the FEM and the Transfer Matrix Method (TMM). The FEM eigenvalue problem was presented above, and the TMM method is presented below.

The general theory of the TMM can be found in Ref. [42]. In the TMM certain variables of interest are defined at different stations in the system. For example, in structural analysis, the system may be a plane beam, as shown in Fig. 1. A beam usually has four variables of interest: deflection of the neutral axis, slope of the neutral axis, internal moment, and internal shear. Naturally, the stations would be the points along the neutral axis of the beam. These variables are placed in a vector which is called the state vector. The TMM then seeks to relate the state vectors at two stations by solving the boundary value problem between the two stations. This leads to the general equation,

\[
z_i = [TM] z_{i-1}
\]
Figure 1. Schematic and Notation for a Plane Beam in the TMM.
where, \([\text{TM}]\) is the transfer matrix,

\[\mathbf{z}_i\] is the state vector at station \(i\), see Fig. 1,

\[\mathbf{z}_i\] is the state vector at station \(i-1\).

This equation is obviously simple to apply between stations. The difficulty in Eq. (11) is the derivation of the transfer matrix. However, through the years, extensive transfer matrix libraries have been developed, Refs. [42, 43].

A system normally contains many stations. To obtain the solution, Eq. (11) must be applied between all the stations, following a certain path. Thus, any solution using the TMM is inherently path dependent. If the path is a smooth, continuous path, along which the number of state variables is constant, then the TMM may be used with ease. However, if there are forks in the path, called "branching", or if the number of state variables changes along the path, then Eq. (11) becomes cumbersome to use. Pestel and Leckie [42] say that although these type of problems can be handled with modifications to Eq. (11), they recommend avoiding the TMM in favor of a more general analysis method, such as the FEM.

In the analysis of long chain-like structures, the FEM has been used extensively. A sample chain-like structure is shown in Fig. 2. As a chain-like structure gets longer, the number of degrees of freedom increases. This causes higher order matrices in a FEM model, creating more complex and expensive solutions [44, 45]. One way to reduce the size of the matrices is to couple the FEM and TMM techniques. This coupling leads to the FETM technique [44-52].

In the FETM a transfer matrix representing the periodic substruc-
Substructure analyzed with FEM

Figure 2. Typical Periodic Structure that May Be Analyzed with the FETM
ture is found with the aid of an FEM analysis. For example, the peri-
odic substructure in Fig. 2 is the substructure in the square box. The
FEM can then be used to develop the transfer matrix for this substruc-
ture. This transfer matrix is necessarily a lumped-parameter transfer
matrix because of the discretization imposed by the FEM. Once the
transfer matrix for the periodic structure is known, Eq. (11) is used to
march from station to station. As the structure gets longer more sta-
tions are added.

The enhancement to the FEM studied here is the incorporation of an
element matrix derived from a continuum transfer matrix. A continuum
transfer matrix is obtained by solving the boundary value problems
exactly between two stations [42]. Therefore, no discretization assump-
tions are made regarding the material property or mass distribution.
This is in contrast to the lumped-parameter transfer matrices, which as
their name implies, lump properties to facilitate the solution of the
boundary value problem.

Of the cited work in the FETM area only Nagamatsu and Nagaike [52]
considered coupling the two techniques in a dynamic stiffness formula-
tion. The standard FEM eigenvalue problem may be written as ([K]−
ω^2[M])d = 0. The matrix in parentheses is commonly called the dynamic
stiffness matrix. Nagamatsu and Nagaike termed this matrix the
"impedance" matrix of the structure. In their work, they converted the
transfer matrix of a beam with neither shear deformation nor rotatory
inertia into an impedance formulation. They then coupled this to the
master degrees of freedom of a finite element model, obtained with
static substructuring. Next, they extracted the lower eigenvalues and
the associated mode shape for the master degrees of freedom. They reported good agreement between computed and experimental results for their technique, although not as good as for a pure TMM eigenvalue extraction.

The enhancement proposed here is similar to Nagamatsu and Nagaike's work because the enhancement is a transfer matrix conversion added to a FEM set of equations. But, the work here differs in the type of transfer matrix used, the purpose of the addition to the FEM, and the type of eigenvector extracted. The transfer matrix to be used here will include rotatory inertia and shear deformation. Nagamatsu and Nagaike's work sought the lower eigenvalues of a structure, whereas the purpose here is to improve the prediction of midrange and higher eigenvalues of a structure. The proposed method will not only find the master eigenvalues and eigenvectors but it will also reconstruct the eigenvector of every continuum element with any degree of precision required. This new technique will be presented in the theory section that follows.
THEORY

The FEM enhancement studied here is the addition of a continuous, exact element to the FEM. There are several ways to formulate continuous elements, as seen in the studies [33, 37, 57]. By definition, these elements satisfy the boundary value problem exactly and generate a transcendental, nonlinear eigenvalue problem. The eigenvalue extraction must, therefore, be carried out with different techniques than for lumped parameter or approximate models. The associated eigenvectors must also be obtained with different techniques [31]. Some of these techniques only approximate the eigenvectors, arguing that the eigenvector is no more than a visual aid.

For the fluid-structure linearization techniques that require accurate eigenvectors, approximations may prove unacceptable. Moreover, the entire ability of experimentally verifying theoretical FEM results depends on verifying the similarity of the experimental and theoretical mode shapes. The field of experimental modal analysis also depends upon accurate mode shapes to carry out structural modifications [54]. Mode shapes from experimental and theoretical approaches can be compared to help pin-point areas of weakness in the theoretical dynamic FEM model. For these reasons, it is considered necessary to obtain the total structure eigenvectors. The eigenvector extraction technique presented here uses the TMM to reconstruct the exact eigenvector description.

A. The Transfer Matrix Method:

The basic concept and definitions in the TMM which led to Eq. (11) were presented in the previous section. This equation summarized the TMM for all
fields of study which use this method. In structural analysis, the state vectors in Eq. (11) are usually made up of an even number of entries. Half of the entries are "displacements" and the remaining are the internal "forces" associated with the displacements. Thus, Eq. (11) may be rewritten as,

\[
\begin{bmatrix}
\mathbf{d}
\end{bmatrix}_i = [\mathbf{T}] \begin{bmatrix}
\mathbf{d}
\end{bmatrix}_{i-1}
\]

where \( \mathbf{d} \) represents the displacements and \( \mathbf{f} \) are the internal forces.

The transfer matrices for structural analysis are quite numerous [42, 43]. Many of these transfer matrices are exact solutions of the differential equations between two stations. These transfer matrices are referred to as "continuum transfer matrices", since the domain of the system between two stations is not discretized. The derivation of continuum transfer matrix for a Timoshenko beam with rotatory inertia is presented in Ref. [42] (section 5.1), and summarized in Table 1.

By definition, the interpolation functions in the FEM and the admissible functions of a Rayleigh–Ritz method are approximations to the solutions of a boundary value problem. That is, the set of solutions that may be obtained with the FEM or a variational method are only approximations of the solution obtained with continuum transfer matrices. Therefore, with a continuum transfer matrix and Eq. (12), once the state vector at one end of an exact element is known, then the displacements and internal loads along the element may be extracted with almost any degree of precision. This makes a continuum transfer matrix an ideal candidate for an exact element to be incorporated into a finite element code. The transformation of a typical catalog transfer matrix into a dynamical stiffness form follows. References will be made to Table 1 since this beam formulation will be used in later sections.
Table 1. Transfer Matrices for a Timoshenko Beam with Rotatory Inertia and Axial Deformation.

<table>
<thead>
<tr>
<th>Positive Sign Convention</th>
<th>Axial Vibration</th>
<th>Bending Vibration</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_{i-1}$</td>
<td>$u_i$</td>
<td>$w_{i-1}$</td>
</tr>
<tr>
<td>$N_{i-1}$</td>
<td>$N_i$</td>
<td>$w_i$</td>
</tr>
</tbody>
</table>

**Transfer Matrix**

$$
\begin{align*}
\cos \Omega & = \frac{l}{EA} \sin \frac{\Omega}{D} \\
-\mu \omega \sin \frac{\Omega}{D} & = \frac{E}{EA} \cos \Omega
\end{align*}
$$

**Variables used in Transfer Matrices**

- $E$: Modulus of Elasticity
- $G$: Modulus of Rigidity
- $A$: Cross section area
- $A_s$: Equivalent shear area
- $I$: Area moment of inertia
- $i$: Radius of gyration
- $\mu$: Length of beam
- $\mu_s$: Mass per unit length

$$
\Omega = \omega \sqrt{\frac{p}{E}}
$$

$$
\begin{align*}
\frac{1}{Z} & = \frac{l}{E I} \\
\alpha & = \frac{\mu \omega^2}{E I} \\
\beta & = \frac{\mu \omega^2}{G A_s} \\
\gamma & = \frac{\mu \omega^2}{E I} \\
\delta & = \frac{\mu \omega^2}{E I} \\
\lambda & = \frac{\lambda_1^2 + \lambda_2^2}{\lambda_1^2 + \lambda_2^2}
\end{align*}
$$

$$
C_0 = (\lambda_2^2 \cos \lambda_1 + \lambda_1^2 \cos \lambda_2) A
$$

$$
C_1 = (\lambda_2^2 \sin \lambda_1 + \lambda_1^2 \sin \lambda_2) A
$$

$$
C_2 = (\lambda_1^2 \cos \lambda_1 - \lambda_2^2 \cos \lambda_2) A
$$

$$
C_3 = (\frac{1}{\lambda_1} \sinh \lambda_1 - \frac{1}{\lambda_2} \sinh \lambda_2) A
$$

$$
C_4 = (\frac{1}{\lambda_1} \sin \lambda_1 + \frac{1}{\lambda_2} \sin \lambda_2) A
$$

$$
C_5 = (\frac{1}{\lambda_1} \sin \lambda_1 + \frac{1}{\lambda_2} \sin \lambda_2) A
$$
Partitioning the \([\text{TMM}]\) in Eq. (12) into,

\[
[\text{TMM}] = \begin{bmatrix}
U_1 & U_2 \\
U_3 & U_4
\end{bmatrix}
\]

Eq. (12) may be rewritten as,

\[
\begin{bmatrix}
d_i \\
f_i
\end{bmatrix} = \begin{bmatrix}
U_1 & U_2 \\
U_3 & U_4
\end{bmatrix} \begin{bmatrix}
d_{i-1} \\
f_{i-1}
\end{bmatrix}
\]

Solving the first set of equations for \(f_{i-1}\),

\[
f_{i-1} = U_2^{-1}d_i - U_2^{-1}U_1d_{i-1}
\]  

(14)

and substituting into the second set of equation to solve for \(f_i\),

\[
f_i = (U_3 - U_4U_2^{-1}U_1)d_{i-1} + U_4U_2^{-1}d_i
\]  

(15)

Eqs. (14) and (15) can be assembled into:

\[
\begin{bmatrix}
-U_2^{-1}U_1 & U_2^{-1} \\
-U_3U_4U_2^{-1}U_1 & U_4U_2^{-1}
\end{bmatrix} \begin{bmatrix}
d_{i-1} \\
d_i
\end{bmatrix} = \begin{bmatrix}
f_{i-1} \\
f_i
\end{bmatrix}
\]

(16)

Equation (16) resembles the classical static stiffness formulation encountered in static FEM problems. Equation (16) must be incorporated into a finite element code with care since the sign conventions between the TMM and the FEM generally do not agree. For the beam element studied here, the differences in sign convention are summarized in Fig. 3. In this figure \(q_i\) is the \(i\)th degree of freedom in the local coordinate system of a finite element beam element. The beam transfer matrix presented in Table 1 was converted into the stiffness matrix by using Eq. (16). The resulting stiffness matrix was algebraically complex to evaluate because of the many trigonometric and hyperbolic terms. The number of operations necessary to evaluate the converted stiffness matrix was greater than those required to evaluate and convert it via Eq. (16). For this reason the latter approach was implemented.

In Eq. (16), all four submatrices contain the submatrix \(U_2^{-1}\). Should the matrix \(U_2\) be singular then every term in the matrix in Eq. (16) will be
Figure 3. Positive Sign Conventions in the FEM and the TMM.
indeterminate because \( \det(U_2) = 0 \) and \( U_2^{-1} = [\text{Cof}(U_2)^T] / \det(U_2) \), \( U_2 \neq 0 \). It is convenient at this point to ask when will the submatrix \( U_2 \) become singular? Perhaps the easiest way to answer this question is by performing a free-vibration analysis of a structure described by Eq. (16). Suppose that a structure has fixed-fixed boundary conditions (i.e. \( d_{i-1} = d_i = 0 \)), and suppose that the structure is vibrating harmonically, then from Eq. (13),

\[
\begin{pmatrix}
0 \\
\mathbf{f}_1
\end{pmatrix} = \begin{bmatrix}
U_1 & U_2 \\
U_3 & U_4
\end{bmatrix} \begin{pmatrix}
0 \\
\mathbf{f}_{i-1}
\end{pmatrix}
\]

To avoid the trivial solution of Eq. (17), \( \mathbf{O} = [\mathbf{T} \mathbf{M}] \mathbf{O} \), then \( U_2 \mathbf{f}_{i-1} = \mathbf{O} \), without \( \mathbf{f}_{i-1} = \mathbf{O} \). This can only be satisfied if \( U_2 \) is a singular matrix, that is, \( \det(U_2) = 0 \). Since all the terms of \( U_2 \) for the beam studied here are functions of frequency, \( \det(U_2) \) is a continuous function of frequency. Setting this function equal to zero gives the characteristic equation, \( \det(U_2) = 0 \), the roots of this equation are the natural frequencies of the fixed-fixed beam problem. In other words, the terms of the dynamical stiffness matrix derived from a transfer matrix will become undefined whenever the circular frequencies selected are evaluated at the natural frequencies of the exact elements with fixed-fixed boundary conditions.

This result was expected based on the discussion of poles of a dynamical matrix, given in the previous section. In a frequency-search eigenvalue extraction, the determinant of a dynamical stiffness matrix will "blow up" as the trial frequency approaches a pole. Poles were shown to occur at the natural frequencies of each of the structure exact elements with fixed-fixed boundary conditions. At each pole, the \( U_2 \) matrix of the associated exact element becomes singular. This introduces terms into the structure dynamic matrix that tend to infinity. These terms that tend to infinity cause the
interminant of the dynamical stiffness matrix to become undefined. Obviously, it is necessary to avoid evaluating the determinant of the structure matrix at a pole.

It is simple to compute the poles before performing a frequency-search eigenvalue extraction. Blevins [40] has tabulated the natural frequencies of fixed-fixed beams. These published eigenvalues do not include the effects of shear deformation nor rotatory inertia. Nevertheless, these eigenvalues can be used as upper bounds for the eigenvalues of a beam element which includes both of these effects. Once a pole is located, it can be determined whether the determinant of the structure dynamic matrix will change signs as a pole is crossed. If there is an even number of like exact elements then the determinant will not change sign, whereas, if there is a odd number of identical exact elements then the determinant will change sign.

In a frequency-search, eigenvalue extraction of a lumped-parameter model, there are no poles. As an illustration of this a lumped-parameter transfer matrix for a beam is shown in Fig. 4. It is easily verified that $U_2$ for this transfer matrix does not contain frequency dependent terms. That is, $\det(U_2) = \text{constant} = -\xi^4/[12(EI)^2]$. $U_2$ can be singular only if $\xi = 0$ or if the product $EI$ is equal to infinity. Both of these conditions are outside the realm of practical structural analysis.

The dynamical stiffness matrix transformed from a transfer matrix was incorporated into a finite element code. A finite element code was written into which the above exact element could be incorporated. The FEM beam element used in this code follows, while the actual implementation of the code is presented in the next section.
Figure 4. Massless Elastic Beam with Lumped Mass Transfer Matrix.
B. The FEM Beam Element:

Of all the structural analysis elements in use, the beam is one of the simplest to develop [10, 11], because energy formulations are not required. The finite element presented here is assumed to have uncoupled bending and axial deformations. The development of the element matrices in local coordinates corresponding to the local displacement vector \( \{q_1, q_2, q_3, q_4, q_5, q_6\}^T \) (see Fig. 3) is found in Ref. [11]. The element stiffness matrix in local coordinates for a linear, plane beam element is,

\[
[K_{\text{element}}] = \frac{EI}{l^3} \begin{bmatrix}
A\frac{l^2}{I} & 0 & 0 & -A\frac{l^2}{I} & 0 & 0 \\
0 & 12 & 6l & 0 & -12 & 6l \\
0 & 6l & 4l^2 & 0 & -6l & 2l^2 \\
-A\frac{l^2}{I} & 0 & 0 & A\frac{l^2}{I} & 0 & 0 \\
0 & -12 & -6l & 0 & 12 & -6l \\
0 & 6l & 2l^2 & 0 & -6l & 4l^2
\end{bmatrix}
\] (18)

where, \( E \) is the modulus of elasticity, 
\( l \) is the length of the beam element, and 
\( I \) is the area moment of inertia.

The mass matrix in local coordinates is,

\[
[M_{\text{element}}] = \frac{EI}{l^3} \begin{bmatrix}
l/3 & 0 & 0 & l/6 & 0 & 0 \\
0 & 13/35l & 11/210 l^2 & 0 & 9/70 l & -13/420 l^2 \\
0 & 11/210 l^2 & 1/105 l^3 & 0 & 13/420 l^2 & -1/140 l^3 \\
l/6 & 0 & 0 & l/3 & 0 & 0 \\
0 & 9/70 l & 13/420 l^2 & 0 & 13/35 l & -11/210 l^2 \\
0 & -13/420 l^2 & -1/140 l^3 & 0 & -11/210 l^2 & 1/105 l^3
\end{bmatrix}
\] (19)
where, $A$ is the cross section area and $\rho$ is the mass density of the material.

The transformation from local into global coordinates is accomplished by the use of coordinate transformation matrix $[g]$,

$$
[g] = \begin{bmatrix}
\cos(a) & \sin(a) & 0 & 0 & 0 & 0 \\
-\sin(a) & \cos(a) & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & \cos(a) & \sin(a) & 0 \\
0 & 0 & 0 & -\sin(a) & \cos(a) & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
$$

(20)

where, $\alpha$ is the orientation angle of the beam before any deformation has occurred. Then with Eq. (20) the element stiffness and mass matrices in global coordinates can be defined,

$$
[K^G_{\text{element}}] = [g]^T[K_{\text{element}}][g],
$$

(21)

$$
[M^G_{\text{element}}] = [g]^T[M_{\text{element}}][g].
$$

(22)

The global mass and stiffness matrices described by Eqs. (21) and (22) were incorporated into a program along with the exact beam element formulation. The development of the program is presented in the next section. What follows in this section outlines the eigenvalue extraction technique implemented into the computer code.

C. The Incremental-Falsi-Bisection Method:

The Incremental, Falsi, and Bisection methods are basic root finding methods that can be found in most numerical analysis textbooks such, as [55]. All three are used in the eigenvalue extraction. First, the Incremental method is used to locate a frequency interval containing a root. This search used a frequency step supplied by the user of the program. Having found such an interval, the Falsi technique is used to home in on the root, to a desired tolerance. The Falsi technique is a first order polynomial interpolation of the determinant. Once the polynomial is found, the root of the polynomial is
used as an approximation to the zero of the determinant. In certain rare circumstances the Falsi technique has computed new approximations to the root which lie outside the interval containing the root [56]. When this occurs, the Falsi technique usually misses the root in the interval. To avoid this, the Falsi iteration implemented checks every time a new approximation is calculated to determine whether or not it lies inside the interval. If the new approximation lies outside the interval, the approximation is discarded and the extraction proceeds using the Bisection method.

A more thorough explanation of details related to the eigenvalues extraction is presented in the Description of Program DSTAP section. It is more appropriate to present the eigenvalue extraction technique in that section since many of the compromises and choices made were directly related to how the extraction was to be implemented.
DESCRIPTION OF PROGRAM DSTAP

A dynamic structural analysis program (DSTAP) was developed to study the two elements presented in the Theory section. This program was the result of extensive modifications and additions to the Structural Analysis Program (STAP). Program STAP is a small, instructional-type, finite element program that can be found in Ref. [9]. It was implemented on an IBM 370 at Virginia Polytechnic Institute and State University, by personnel of the Engineering Science and Mechanics Department [58]. This implementation was moved to a VAX cluster, running VMS 4.1, before the two elements were added. The VAX cluster consisted of two 11/780 processors with 8 megabytes of memory. Off-line storage was provided by RP07 and RA81 disks.

As presented in Ref. [9], program STAP is written in FORTRAN 66. It has a one element library, a truss element, and is set up to perform static analysis. The procedure for solving static problems may be seen in the flowchart presented in Fig. 5, taken from Ref. [9]. A detailed explanation of each function block in the flowchart may be found in Ref. [9], and will not repeated here.

In STAP, the set of equations arising from static problems is solved using a nodal front solver and Gauss elimination. The nodal front solver retains the same numbering scheme for the degrees of freedom as supplied by the user. That is, the numbering of the system of unknowns is not optimized for an in-core solution, as is done in some commercial codes. This same numbering scheme for the unknown degrees of freedom was retained in DSTAP. Program STAP also uses an efficient matrix storage scheme in which only the nonzero terms of the global matrices lying on or above the main diagonal are
START

Read nodal point data and establish equation numbers

Read and store the load data for all load cases

Read and store element data over all element groups

Read element group data and assemble global stiffness matrix for all element groups

FOR EACH LOAD CASE

Read load vector and calculate nodal point displacements

Read element group data and calculate element stresses over all element groups

END

Figure 5. Flowchart of program STAP and Static Portion of Program DSTAP.
actually stored. This same storage scheme was adapted for program DSTAP. This technique for storing model matrices is oftentimes referred to as "Skyline" storage.

The last feature of STAP incorporated into DSTAP is, perhaps, the prime cause of frustration among novice programmers. This feature deals with the way in which model data is stored in-core. All array data needed in an analysis is stored in one large, one-dimensional array called the A-array. Since several arrays and matrices are stored in the A-array at once, pointers are set up to locate initial word of each array or matrix. This partitioning scheme is known as "dynamic storage". There are two advantages to using dynamic storage. First, the code can be easily modified for different size problems by changing the dimension of the A-array in the main program. Second, in-core memory can be efficiently reused as arrays are no longer needed by changing the pointers to the A-array. A set of tables explaining the allocation of pointers to the A-array during different phases of the program DSTAP are attached in Appendix A.

The flowchart of program DSTAP is presented in Fig. 6. Although it is not shown in Fig. 6, program DSTAP retains the static analysis capabilities of STAP. In the case of a static analysis program DSTAP functions in the same manner as program STAP (Fig. 5). In the flowchart of program DSTAP (Fig. 6) a subroutine name appears next to each function block. This indicates that a major portion of the operations described in said block are done in that subroutine. For the interested reader, the subroutines named in the flowchart are included in Appendix B (except subroutine INPUT which is in Ref. [9]), along with other supporting subroutines. The set of subroutines in Appendix B constitute the additions and modifications to program STAP which
Start

1. Read nodal point data. Establish equation numbers
   INPUT
2. Read, generate, and store element data. Loop over 2 element groups
   FEBEAM
   CBEAM
3. Assemble FEM \([K]\) and \([M]\) matrices
   FEASEM
4. Solve the linear eigenvalue problem
   EIGZ
   EXPR
5. Set-up constant values for evaluation of \([K(\omega)]\). Set-up dynamic storage.
   CPTVAL
6. Extract all poles of \([K(\omega)]\) in frequency range
   POLES
7. Use a frequency scan to find all eigenvalues of \([K(\omega) + K - \omega^2M]\)
   BFMTHD
   Loop over the entire frequency range of interest
   Plot eigenvectors?
   Yes
   Compute associated eigenvector
   SOLVEC
   Plot eigenvectors?
   No
   Stop
   IEIGV
   Loop over all the eigenvalues found in the frequency range
   Plot eigenvectors?
   Yes
   Compute initial state vector. March down the beam
   PLOTB
   PLOT
   Display eigenvectors
   EXPVEC
   Stop

Figure 6. Flowchart of program DSTAP.
were necessary to implement the two elements described in the Theory section and the eigenvalue extraction outlined there. Program DSTAP consists of these subroutines and subroutines from program STAP [9], the PLOT10 library [59], and the IMSL library [60].

The remainder of this section will describe in detail the function blocks in the flowchart of program DSTAP, Fig. 6. These have been numbered to facilitate any references to a particular block. The number assigned to each block appears to the left of the block in bold numbers, while the names of the associated subroutines appear to the right of the block. The reader may wish to refer to the particular subroutine(s) in Appendix B, although this will not be necessary.

The first block in Fig. 6 describes the process of inputting the model nodal points. This task is performed in subroutine INPUT which is taken directly from program STAP; thus, it is not included in Appendix B. The nodal data consists of the coordinates and degrees of freedom of each node point. Provisions are included for generating nodes automatically along straight lines. This nodal generation technique is described in Ref. [9].

The second block in Fig. 6 summarizes the element generation process. The exact beam element information is read in through subroutine CBEAM. The pointers needed for the arrays in subroutine CBEAM are computed in subroutine TBEAM. Thus, the sole purpose of CBEAM is place the user supplied information into the storage locations allotted by TBEAM. Some automatic element generation capabilities are supplied in CBEAM, as described in Appendix C which contains the instruction for using program DSTAP. The FEM beam element formulation was placed in subroutine FEBEAM, and the computation of the pointers for dynamic storage were also placed in subroutine TBEAM.
Unlike CBEAM, FEBEAM is called twice, (1) to read in the element data, (2) to generate the FEM beam element stiffness and consistent mass matrices. These element matrices are assembled into the structure FEM matrices using the subroutine FEASEM, as shown in the third function block of Fig. 6.

A model made up of only finite elements gives a linear eigenvalue problem. There are many algorithms available for solving this problem. As shown in the fourth function block of Fig. 6, two algorithms are provided in DSTAP, one in subroutine EIGENR and another in subroutine EIGZF. The modified QR algorithm in EIGENR uses the global matrix properties of the finite element formulation to optimize the solution time; that is, it requires real, symmetric, positive definite matrices. This modified QR algorithm was developed by a graduate student of the Aerospace and Ocean Engineering Department at Virginia Polytechnic Institute and State University [61]. Unfortunately, no documentation is available for this routine.

In certain cases the FEM matrices become non-positive definite due to the finite arithmetic performed by the computer. In these cases it is convenient to have an extraction routine that does not require the matrices to be positive definite. This is why EIGZF was also added to STAP. EIGZF is a routine available in the IMSL library of subroutines [60]. EIGZF only requires that the matrices be real.

Whenever an exact beam element is incorporated into a structure model the resultant eigenvalue problem is a transcendental eigenvalue problem. In this case a frequency-scan is performed on the structure dynamic stiffness matrix as was outlined in the Theory section. This search is embodied in the first loop of Fig. 6. The kernel of this search resides in subroutine BFMTHD
which contains a rather crude algorithm that combines the Incremental Search, Regula-Falsi, and Bisection algorithms. The Incremental Search seeks a zero crossing of the determinant of the dynamic structure matrix,

\[ \det([K(\omega)] + [K] - \omega^2 [M]) = 0, \]  

(23)

where, \([K(\omega)]\) is the dynamic stiffness matrix assembled from beam exact elements, 
\([K]\) is the global stiffness matrix assembled from beam finite elements, and 
\([M]\) is the assembled mass matrix obtained from the consistent beam finite element mass matrices.

Once a zero crossing has been detected, the Regula-Falsi algorithm, in subroutine FALSI, is invoked. Should the new approximation to a root in subroutine FALSI lie outside the interval containing the root, then FALSI calls subroutine HALF, which contains a Bisection algorithm. This rare occurrence was observed once during the development of program DSTAP. However, not enough information was saved about that model to be able to reproduce the results.

It should be evident from the discussion above that Eq. (23) will be evaluated at many different trial frequencies. If possible the trial frequencies should avoid the poles of \(\det[K(\omega)]\) because this may cause an overflow condition. Thus, before subroutine BFMTHD is actually called, the poles of \(\det[K(\omega)]\) are found and the dynamic memory is rearranged for an effective evaluation of \([K(\omega)]\). This is done in subroutines CPTVAL and POLES, as shown in the 5th and 6th function blocks of Fig. 6. CPTVAL first computes all the values independent of \(\omega\) needed to evaluate \([K(\omega)]\). These constants are stored in dynamic memory, rather than off-line, to speed up the evaluation of \([K(\omega)]\). Prior to rearranging the A-array, CPTVAL calls subroutine POLES to compute the poles of \(\det[K(\omega)]\). These poles are also stored in-core, as will be explained below.
In subroutine POLES, the 6th function block in Fig. 6, the poles of det[K(ω)] are found by approximating them with the eigenvalues of an exact, fixed-fixed beam element, neglecting shear deformation and rotatory inertia. The approximating eigenvalues have been tabulated by Ref. [40]. Since the approximating eigenvalues correspond to a more rigid element, these eigenvalues are upper bounds to the poles of the exact beam element with both shear deformation and rotatory inertia. The lower bound for each pole can either be searched for, by backing off the upper bound, or by arbitrarily choosing a known lower bound, such as a previous pole. It was found more convenient to search for a pole by backing off the upper bound. Once the upper and lower bounds have been established, subroutine FALSI finds the pole with an accuracy of 0.1 rad/sec. The poles are extracted up to and including the first pole beyond the upper frequency specified by the user. This assures that the last pole in the frequency range of interest will not be missed, regardless of how close it is to the upper frequency.

Having computed and stored the poles in the frequency range of interest, subroutine CPTVAL rearranges the A-array storage in preparation for BFMTHD, the 7th block in Fig. 6. With the structure poles stored in-core, it is simple to determine whether a sign change of Eq. (23) is indicative of an eigenvalue or not. Let f(ω)=det([K(ω)]+[K]-ω²[M]), so that an eigenvalue ω makes f(ω)=0. Suppose that the last evaluation of f(ω) occurred at frequency ωₐ, and that the current frequency being tested is ωᵦ. If there are an odd number of poles between ωₐ and ωᵦ then the Wittrick-Williams theorem establishes that the f(ω) will change signs. This means that if f(ωₐ)*f(ωᵦ) > 0 then an eigenvalue has been trapped, whereas if f(ωₐ)*f(ωᵦ) < 0 then there is no eigenvalue in the interval [ωₐ, ωᵦ]. This may seem contrary to the
typical criteria to trap roots of functions. However, it is easy to recast
the bound criteria in a more traditional fashion by defining \( g(\omega) = -1^Pf(\omega) \),
where \( p \) is the number of poles lying between \( \omega_b \) and the low end of the frequency
range specified by the user. For this reason, \( p \) is not necessarily equal to
\( J_0(\omega) \), since \( J_0(\omega) \) is the number of all the poles below \( \omega \). This definition
of \( g(\omega) \) has the advantage that whenever \( g(\omega_a) \times g(\omega_b) < 0 \), then an eigenvalue
has been trapped. This definition of \( g(\omega) \) was placed in function FCN, which
is used by subroutines BFMTHD, FALSI, and HALF to extract the structure
eigenvalues.

An additional feature added to the function FCN is a logarithmic
compression of \( g(\omega) \). Usually the terms along the main diagonal of structure
stiffness matrices are quite large in magnitude. For the numerical examples
to be presented in the next section, these terms were in the order of \( 10^5 \)
(lbf/in). The highest possible order of the determinant of a matrix may be
estimated by raising the largest term in the matrix to the order of the matrix.
Thus, for a 3 by 3 matrix with terms in the order of \( 10^5 \) the determinant could
be in the order of \( 10^{15} \). The double precision representation of real
variables in a VAX 11/780 allows for number of up to \( 10^{36} \). Therefore, if the
the matrix in the example was an 8 by 8 matrix the determinant could be in the
order of \( 10^{40} \), exceeding the capacity of the VAX representation and causing an
overflow error. For this reason, instead of returning \( g(\omega) = -1^Pf(\omega) \) function
FCN actually returns,

\[
g(\omega) = \begin{cases} 
-1^Pf(\omega), & \text{if } |f(\omega)| < e \\
-1^P\text{sgn}(f(\omega)) \ln(|f(\omega)|), & \text{if } |f(\omega)| > e
\end{cases}
\]

where the \( \text{sgn} \) function is 1 if \( f(\omega) \) positive or -1 if \( f(\omega) \) is negative, the \( \ln \)
function is the Napierian logarithm, and \( e \) is the base of the Napierian
logarithms. It is obvious that the function returned by FCN will have a slight discontinuity whenever \( f(\omega) = e \). Although it is unlikely, if this discontinuity should cause any difficulties for subroutine FALSI, then the Bisection algorithm in subroutine HALF would be called.

Once an eigenvalue has been found with the desired accuracy, the eigenvector is backed out using subroutine SOLVEC, the 8th function block in Fig. 6. SOLVEC uses an full pivoting technique to find the eigenvector \( \mathbf{x} \). In full pivoting, the last diagonal term will be the term with the smallest magnitude. If there are \( n \) equations in the eigenvalue problem, then after upper triangularizing \( \mathbf{A} = ([K(\omega)] + [K] - \omega^2[M]) \) at an eigenvalue \( \omega \) with full pivoting, \( a_{n,n} = 0 \). Therefore, to find the eigenvector \( \mathbf{x} \), the last entry in \( \mathbf{x} \) is arbitrarily set to 1. The remaining \( (n-1) \) entries of \( \mathbf{x} \) are found by backsolving the system in the upper-triangular matrix \( \mathbf{A} \). Once \( \mathbf{x} \) is known, it is normalized so the largest entry in \( \mathbf{x} \) is equal to 1, and the resulting vector is written to file IEIGV for post-processing.

The procedure described above for BFMTHD and SOLVEC is repeated for each eigenvalue found in the frequency range of interest, as shown in Fig. 6. Once the entire frequency range has been scanned, tape IEIGV will contain all the mode shapes and eigenvalues in the frequency range. These mode shapes will only contain the displacements at the juncture points of the elements. For example, if a cantilever beam is modeled with one exact beam element, and \( n \) mode shapes are extracted, it would be rather difficult to see any evident difference from mode shape to mode shape. The traditional vibration nodes would be missing from these mode shapes because the known displacements would be at the fixed-end (zero) and at the free end. As presented in the Literature Review section several studies in the exact stiffness field have
attempted to reconstruct the actual mode shapes by approximations of
different types. These approximations are usually argued to be sufficient
based on the premise that mode shapes are no more than a visual aid for the
analyst [31]. However, recent work refutes this assumption and,
furthermore, this work claims that it is not necessary to approximate the mode
shapes at all, as explained below.

In the post-processing phase of program DSTAP is comprised of the last
two function blocks of Fig. 6. The purpose of the part of the program is plot
the structure eigenvectors on the terminal screen. This section of the
program is optional and the user must turn on the plotting flag for this
section to execute. The plotting flag is described in Appendix C. Prior to
displaying the mode shape on the screen, DSTAP must compute the eigenvector of
each continuous element, since only the end displacements are known. This is
accomplished by resorting to Eqs. (11) through (14). In Eq. (11) the
transfer matrix technique was outlined as,

$$ z_i = [TM]z_{i-1} $$  \hspace{1cm} (24)

where, $z_{i-1} = \{d_{i-1} \ f_{i-1}\}^T$. If the initial state vector at one end of the
continuous beam element is obtained, then it is possible to define a transfer
matrix over a fraction of the beam length and use it to compute state vectors
along the continuous element. It is also possible to define a transfer
matrix over the entire length of the beam element such that,

$$ \left\{ \begin{array}{c} d_J \\
 f_J \end{array} \right\} = \left[ \begin{array}{cc}
 U_1 & U_2 \\
 U_3 & U_4 \end{array} \right] \left\{ \begin{array}{c} d_I \\
 f_I \end{array} \right\} $$  \hspace{1cm} (25)

where, the subscript $I$ refers to the $I^{th}$ end of the beam, and the subscript $J$
refers to the $J^{th}$ end of the beam. Both $d_I$ and $d_J$ are known from the
eigenvector $x$, which was stored on tape. With these displacements the
initial state vector \( z_I \) may be found with the use of Eq. (14),

\[
f_I = U_2^{-1}(d_J - U_1 d_I)
\]  

(26)

Note, \( U_2 \) in Eq. (26) is nonsingular, because if it were singular then the frequency at which the eigenvector \( x \) was computed would be a pole of \( \text{det}[K(\omega)] \). This is not possible in program DSTAP; as frequencies that correspond to poles are ignored in the eigen-extraction process. Suppose that a structure eigenvalue is also pole. Although it is possible that \( g(\omega) \) will be bounded at such an eigenvalue, it is unlikely that \( g(\omega) = 0 \). Thus, \( x = 0 \) at such an eigenvalue, and \( f_I = 0 \), which would make the eigenvector reconstruction impossible using Eq. 24. Therefore, the eigenvector reconstruction is performed for only those structure eigenvalues which are not poles of \( \text{det}[K(\omega)] \). With the initial state vector known, subroutine PLOTB loops over all continuous elements to obtain the state vectors along each one. This data is then scaled so the largest eigenvector entry is ten percent of the largest structure dimension. The scaled vector is finally plotted on the terminal screen for the user to see.

The evaluation of the continuous element eigenvectors at frequencies above 12,000 rad/sec incurred severe round-off error when double precision variables were used. At these relatively high structure frequencies the computed state vector displacements at the \( J^{th} \) end of the beam were not close to the \( d_J \) from eigenvector \( x \). In order to reduce truncation error, the computation described in Eq. (24) was performed in REAL*16, commonly known as quad-precision. The added precision in the REAL*16 operations permitted the extraction of all the eigenvectors attempted. This included the extraction of eigenvectors up to the 50,000 rad/sec region.
NUMERICAL EXAMPLES

In this section, three different structures will be analyzed with the proposed enhancement to the finite element eigenvalue problem. The first example is a fixed-fixed beam, which will be used to benchmark both of the elements added to DSTAP against known solutions. After establishing the accuracy of these elements, the second example compares the FEM beam element against the exact beam element, by analyzing a Portal Arch in DSTAP. Finally, in the third example, another structure is examined to test the conclusions arrived at in the second example and to compare the results against a commercial FEM code.

A. Fixed-Fixed Beam:

The first example presented here is the analysis of a beam with fixed-fixed boundary conditions as shown in Fig. 7. This is a rather simple eigenvalue problem which should make the comparisons among the different models clearer. The first comparison to be presented is between the eigenvalues of a closed-form solution and those of a model using the so called exact elements. The second comparison is between the eigenvalues extracted from a finite element model in DSTAP and those from a finite element model in SUPERB [15].

The closed-form solution for a Timoshenko beam with no shear deformation is widely available in the literature [40, 41]. In this solution, the axial vibration is usually uncoupled from the transverse vibration. Hence, the axial and transverse eigenvalues are found independently. The exact displacement formulation, which led to the exact element, also uncouples
Material Properties:

- \( E = 30.0 \text{ Mpsi} \) (207 GPa)
- \( \rho = 0.282 \text{ lb}_f/\text{in}^3 \) (7.81 mg/mm\(^3\))
- \( G = 11.6 \text{ Mpsi} \) (79.9 GPa)

**Figure 7. Schematic of the Beam for Example A.**
these two types of vibration. To determine the accuracy of the exact beam element it is appropriate to compare the closed-form solution eigenvalues with those from an exact beam element model.

The closed-form solution to the transverse vibration of a beam with fixed-fixed end conditions may be found in [40, 41]. The associated characteristic equation is,

$$1 - \cosh(\beta_i)\cos(\beta_i) = 0$$

where,

$$\beta_i^4 = \mu \omega_i^2 L^4/(EI),$$

and,

- $\mu$ is the mass per unit length,
- $L$ is the length of the beam,
- $E$ is the modulus of elasticity,
- $I$ is the moment of inertia, and
- $\omega_i$ is the $i^{th}$ eigenvalue.

The values of the above constants for this example appear in Fig. 7.

As the solution of Eq. (28) for the eigenvalues $\beta_i$ is difficult to accomplish by hand calculations, the first twenty positive, nonzero roots of Eq. (27) were found numerically. A small program using a combination Incremental-Bisection method was written to extract these roots employing the quad-precision capability of VAX BASIC. This allows a mantissa of about 33 digits throughout all operations. The roots obtained with this program are presented in Table 2, along with the residuals of the computation. A residual is defined by letting $f(\theta) = 1 - \cosh(\theta)\cos(\theta)$, then at a computed root $\beta_i$ the residual is $f(\beta_i)$, which usually is nonzero. The entries in this table have been included with all the digits available from the VAX BASIC program.

With the first twenty roots of Eq. (27) known, it is simple to compute the first twenty bending eigenvalues with Eq. (28). In addition to the bending
Table 2. The First 20 Roots of Eq. (27) Obtained in Quad-Precision With a VAX BASIC Program.

<table>
<thead>
<tr>
<th>Root No.</th>
<th>Root</th>
<th>Residual</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.730040744863799773156642913813</td>
<td>-0.63164913776664798064E-0010</td>
</tr>
<tr>
<td>2</td>
<td>7.8532046240958379712537862360477</td>
<td>5.3339650106764214789E-0013</td>
</tr>
<tr>
<td>3</td>
<td>10.995607838001667744654327023774</td>
<td>9.4261355268224589479E-0011</td>
</tr>
<tr>
<td>4</td>
<td>14.1371654912574642448674921979546</td>
<td>4.6735385022183045213E-0011</td>
</tr>
<tr>
<td>5</td>
<td>17.2787596573994814419528154303407</td>
<td>-0.6163440376432062314E-0010</td>
</tr>
<tr>
<td>6</td>
<td>20.4203522456260610907878561004480</td>
<td>-0.5486606774958574472E-0010</td>
</tr>
<tr>
<td>7</td>
<td>23.5619449033863673677369845451315</td>
<td>-0.1150295493682681283E+0002</td>
</tr>
<tr>
<td>8</td>
<td>26.7035375569710820053358241555949</td>
<td>4.8904674011102316517E-0011</td>
</tr>
<tr>
<td>9</td>
<td>29.8451302105659538446662712459055</td>
<td>1.000000000000642321000E+0000</td>
</tr>
<tr>
<td>10</td>
<td>32.9867228641557153184559825180806</td>
<td>-0.6202721470963317770E-0010</td>
</tr>
<tr>
<td>11</td>
<td>36.1283155177455184072684077070566</td>
<td>-0.2682509567109112007E-0008</td>
</tr>
<tr>
<td>12</td>
<td>39.269908171335112200588401156328</td>
<td>-0.6980363534164094482E-0007</td>
</tr>
<tr>
<td>13</td>
<td>42.4115008249251044769164479371144</td>
<td>-0.9954152765342742558E-0006</td>
</tr>
<tr>
<td>14</td>
<td>45.5530934785148977145841736139700</td>
<td>4.6382038809404356563E-0005</td>
</tr>
<tr>
<td>15</td>
<td>48.6946861321046909530811677072448</td>
<td>-0.4886073873088245472E-0003</td>
</tr>
<tr>
<td>16</td>
<td>51.8362787856944841915423266820226</td>
<td>-0.1468249583513834485E-0001</td>
</tr>
<tr>
<td>17</td>
<td>54.9778714392842774300050341389070</td>
<td>3.6833796870830283231E-0001</td>
</tr>
<tr>
<td>18</td>
<td>58.1194640928740706684676754213425</td>
<td>-0.2677648280154856920E+0001</td>
</tr>
<tr>
<td>19</td>
<td>61.2610567464638369069303183581392</td>
<td>-0.1573031435787337315E+0003</td>
</tr>
<tr>
<td>20</td>
<td>64.4026494000536571453929612949359</td>
<td>1.5821998698373199700E+0003</td>
</tr>
</tbody>
</table>
eigenvalues, the axial vibration eigenvalues are also of interest. They will be the roots of the characteristic equation for the axial vibration of a fixed-fixed bar. From Ref. [41], that equation is

$$\sin(\omega \sqrt{\rho/E}) = 0$$  \hspace{1cm} (29)$$

where, \( \rho \) is the mass density of the material, and the remaining variables have been defined above. The positive, nonzero roots of this equation are, \( i\pi \), \( i=1,2,3, \ldots \). When the material and geometry values for this example are substituted into Eq. (29), the axial eigenvalues are given by the equation,

$$\omega_i = i 8,444.384175 \pi \text{ (rad/sec)}, i=1,2,3, \ldots$$  \hspace{1cm} (30)$$

The eigenvalues from Eqs. (28) and (30) were grouped and ordered in increasing magnitude. Once this list of eigenvalues was assembled, all the eigenvalues below 100,000 (rad/sec) were placed in Table 3. In this table, it is simple to determine which eigenvalues correspond to the axial vibrations of the beam because they have been left in the form of Eq. (30). That is, they are in the form of a constant times \( \pi \). Also, in this table, the bending eigenvalues have been reported with all the digits obtained from the VAX BASIC program used to evaluate them.

The two right columns of Table 3 contain the eigenvalues obtained with two exact-element models of the fixed-fixed beam in DSTAP. A two-element and a four-element models were prepared and analyzed with DSTAP. DSTAP was then requested to find all the eigenvalues in the frequency range from 1 rad/sec to 100,000 rad/sec. The step size for the incremental search part of the analysis was set at 400 rad/sec. The eigenvalues obtained by DSTAP have been reported with all the digits available from DSTAP. The input file supplied to DSTAP for the two-element model is presented in Appendix D, along with the corresponding DSTAP output. The Wittrick-Williams counter,
Table 3. Comparison Between Eigenvalues from Closed-Form Solution and Two Exact Element Models.

<table>
<thead>
<tr>
<th>Eig. No.</th>
<th>Eigenvalue from Characteristic Equation, see Table 2 (rad/sec)</th>
<th>Eigenvalues from DSTAP 2 Element Model (rad/sec)</th>
<th>Exact Element Models 4 Element Model (rad/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>568.114522040818126544595429219319</td>
<td>568.11452200995041</td>
<td>568.11452200995051</td>
</tr>
<tr>
<td>2</td>
<td>1566.02955644479324375556254257020</td>
<td>1566.0295563631167</td>
<td>1566.02955636311660</td>
</tr>
<tr>
<td>3</td>
<td>3070.044082687976292095954260308490</td>
<td>3070.04408270259415</td>
<td>3070.04408270259376</td>
</tr>
<tr>
<td>4</td>
<td>5074.938819855466481806042984155820</td>
<td>5074.93881958207260</td>
<td>5074.93881958207805</td>
</tr>
<tr>
<td>5</td>
<td>7581.083056944703460106143735837360</td>
<td>7581.08305653635637</td>
<td>7581.08305653631123</td>
</tr>
<tr>
<td>6</td>
<td>10588.4547685305197131341265640655000</td>
<td>10588.45476796008461</td>
<td>10588.45476796011121</td>
</tr>
<tr>
<td>7</td>
<td>14097.055170692215967979412239110300</td>
<td>14097.05516832095850</td>
<td>14097.05516832227977</td>
</tr>
<tr>
<td>8</td>
<td>18106.88419675103585034124229410050</td>
<td>18106.88419377764967</td>
<td>18106.88419379174047</td>
</tr>
<tr>
<td>9</td>
<td>22617.941851052668384564001178261500</td>
<td>22617.94184767444040</td>
<td>22617.94184767131458</td>
</tr>
<tr>
<td>10</td>
<td>8444.384175π a</td>
<td>26528.81529027919487</td>
<td>26528.81529027919532</td>
</tr>
<tr>
<td>11</td>
<td>27630.228133576398457278810826060200</td>
<td>27630.22812977982403</td>
<td>27630.22812963752040</td>
</tr>
<tr>
<td>12</td>
<td>33143.743044329232800284664465696600</td>
<td>33143.74303937236709</td>
<td>33143.74303986000541</td>
</tr>
<tr>
<td>13</td>
<td>39158.48658331108784449669456345500</td>
<td>39158.48657111666944</td>
<td>39158.4865782463820</td>
</tr>
<tr>
<td>14</td>
<td>45674.4587505219817611253266516600</td>
<td>45674.45873240057063</td>
<td>45674.4587449132977</td>
</tr>
<tr>
<td>15</td>
<td>52691.659545961913690792777793125800</td>
<td>52691.65953725586132</td>
<td>52691.6595397324772</td>
</tr>
<tr>
<td>16</td>
<td>16888.76835 π b</td>
<td>53057.630580583906</td>
<td>53057.630580583984</td>
</tr>
<tr>
<td>17</td>
<td>60210.089969630883674399023636859400</td>
<td>60210.08919613237413</td>
<td>60210.0896276889118</td>
</tr>
<tr>
<td>18</td>
<td>68229.747021528891709510796405643100</td>
<td>68229.74224404596498</td>
<td>68229.74701399454716</td>
</tr>
<tr>
<td>19</td>
<td>76750.633701655937769395321134119200</td>
<td>76750.62564726539495</td>
<td>76750.6336934289524</td>
</tr>
<tr>
<td>20</td>
<td>25333.15253 π c</td>
<td>79586.44587083758597</td>
<td>79586.44587083758597</td>
</tr>
<tr>
<td>21</td>
<td>85772.74901001201935048573162061300</td>
<td>85772.84231593319964</td>
<td>85772.7490010338587</td>
</tr>
<tr>
<td>22</td>
<td>95296.092946597141425470731841294000</td>
<td>95296.257983896483</td>
<td>95296.09293674484616</td>
</tr>
</tbody>
</table>

Notes:  
\(^a\) 8444.384175π = 26528.81528826990696439594387787...  
\(^b\) 16888.76835π = 53057.63057659381932879188775574...  
\(^c\) 25333.15253π = 79586.44586480972089318783163362...
presented in Eq. (10), is also demonstrated in this Appendix by performing a second analysis of the two-element model. In the reanalysis, DSTAP is requested to find the eigenvalues between 20,000 and 30,000 rad/sec. The sequential number of the eigenvalues in this frequency range is obtained by letting the counter determine how many structure eigenvalues lie below 20,000 rad/sec.

Comparing the columns in Table 3, it may be concluded that the exact displacement method does indeed find the exact eigenvalues. The values in the left column are for practical purposes the eigenvalues of the structure. They are based on the characteristic equation of the boundary value problem and they were computed in quad-precision, with a 33 digit mantissa. The middle column of Table 3 are the eigenvalues of the two-exact-element model in DSTAP. Comparing these eigenvalues with the exact eigenvalues, it can be verified that the DSTAP eigenvalues match to 9 significant digits with the correct eigenvalues for the eigenvalues below 60,000 rad/sec. Above 60,000 rad/sec, the DSTAP eigenvalues start deviating very slightly from the correct eigenvalues. By the 22\textsuperscript{nd} eigenvalue, the last eigenvalue in the range of interest, the DSTAP eigenvalue matches to 5 significant digits with the correct eigenvalues. The worst deviation from the correct eigenvalues occurs at this eigenvalue. This deviation is 0.165 rad/sec, which is a relative error of 0.000 173%. This is well within what is considered an acceptable result for practical applications. Thus, the two-exact-element (3 degrees of freedom) model is accurate enough to obtain the exact eigenvalues.

As a verification of the above conclusion, a four-exact-element (9 degrees of freedom) model was also run in DSTAP. The eigenvalues of this
model are presented in the third column of Table 3. Comparing these eigenvalues with the correct eigenvalues, it is evident that they match to at least 9 significant digits for the entire frequency range [1., 100,000.] rad/sec. Also, if the 10\textsuperscript{th} significant digit of the DSTAP eigenvalues is obtained by rounding, then the 10 significant digits match exactly. Thus, the conclusion that the exact-element models obtain the exact eigenvalues is more strongly reaffirmed by the DSTAP eigenvalues of the four-exact-element model.

In the following examples the frequency ranges of interest will lie below 50,000 rad/sec. The eigenvalues reported in these examples will be given with at most 9 significant digits. Thus, the eigenvalues obtained from the exact displacement method will be considered the correct eigenvalues. Eigenvalues from other models will be compared to these eigenvalues to determine whether or not they are acceptable. The definition of an acceptable eigenvalue will be presented in the Portal Arch example.

Having established the applicability of the exact beam element, the second comparison in this example seeks to determine how well the finite element added to DSTAP approximates the results obtained by a commercial code. It was not possible to add the exact beam element formulation to a commercial finite element code. Therefore, a standard finite element beam formulation was added to DSTAP. Before using this element in case studies, it is appropriate to compare its results with the results of a commercial code. The code chosen for this comparison was General Electric's SUPERB finite element program [15].

The three geometries presented in Fig. 8 were modeled with FEM beam elements in DSTAP and in SUPERB. The results from these analyses are
Figure 8. Models Used in Example A.
presented in Table 4. The table is divided into three wide columns, one for each geometry of Fig. 8. Each one of these wide columns is divided into three smaller columns. From left to right these columns are: (1) the set of results obtained from a DSTAP run, (2) the set of eigenvalues from a SUPERB run, and (3) the relative difference between the analyses. For example, the central wide column contains the results that correspond to the geometry in Fig. 8-b. This model had 4 elements and 9 degrees of freedom in both DSTAP and SUPERB. It took 0.92 of a CPU second to solve this problem in DSTAP, while it took 15.76 CPU seconds to solve the same model in SUPERB. The largest relative difference in this model between the eigenvalues computed with the two codes occurred in the sixth eigenvalue, where DSTAP computed it as 15,808.2841 rad/sec and SUPERB computed it as 15,772.68 rad/sec, a relative difference of 0.225%.

The relative difference column in Table 4 is not the same as the relative error of the computed eigenvalues. The relative error may be obtained by using the closed-form solution eigenvalues which have been reproduced in Table 4, truncated to 4 decimal places. For example, sixth eigenvalue from the closed-form solution column is 10,588.4547 rad/sec. Thus, the relative error that both DSTAP and SUPERB had for the sixth eigenvalue from the 4-FEM-element model was approximately 49%.

The main reason for computing the relative difference between the two codes is to demonstrate that the two codes give comparable results for the FEM beam element formulations. The largest relative difference between the two codes occurred in the third model, sixteenth eigenvalue, where the relative difference was approximately 1%. The available SUPERB documentation was rather ambiguous concerning both the plane-beam element formulation and the
Table 4. Comparison Between FEM models in DSTAP and SUPERB Eigenvalues (rad/sec).

<table>
<thead>
<tr>
<th>Program: CLOSED-</th>
<th>DSTAP</th>
<th>SUPERB</th>
<th>DSTAP</th>
<th>SUPERB</th>
<th>DSTAP</th>
<th>SUPERB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routine: FORM SOL.</td>
<td>EIGENR</td>
<td>JACOBI</td>
<td>EIGENR</td>
<td>JACOBI</td>
<td>EIGENR</td>
<td>JACOBI</td>
</tr>
<tr>
<td># Elements (^2)</td>
<td>2</td>
<td>2</td>
<td>Rel.</td>
<td>4</td>
<td>4</td>
<td>Rel.</td>
</tr>
<tr>
<td># dof (^3)</td>
<td>3</td>
<td>3</td>
<td>Diff.</td>
<td>9</td>
<td>9</td>
<td>Diff.</td>
</tr>
<tr>
<td>CPU secs. (^4)</td>
<td>0.7600</td>
<td>15.1300</td>
<td>(%)</td>
<td>0.9200</td>
<td>15.7600</td>
<td>(%)</td>
</tr>
</tbody>
</table>

| | E 1 | 568.1145 | 577.3233 | 577.2927 | 0.0052 | 568.8685 | 568.8356 | 0.005 | 568.1624 | 568.1319 | 0.005 |
| | i 2 | 1566.0295 | 2081.5687 | 2081.5038 | 0.0247 | 1580.5149 | 1580.5182 | 0.022 | 1567.0189 | 1566.712 | 0.019 |
| | g 3 | 3070.0440 | 29252.2048 | 29251.9975 | 0.0007 | 3135.6118 | 3134.1785 | 0.045 | 3077.3190 | 3075.933 | 0.004 |
| | e 4 | 5074.9388 | | | | 5932.3102 | 5927.3056 | 0.084 | 5106.5973 | 5102.637 | 0.077 |
| | n 5 | 7581.0830 | 9811.1566 | 9797.3708 | 0.140 | | | | 7680.4152 | 7671.140 | 0.120 |
| | v 6 | 10588.4547 | 15808.2841 | 15772.6800 | 0.225 | | | | 10827.0682 | 10808.33 | 0.173 |
| | a 7 | 14097.0551 | 27214.8261 | 27214.9888 | 0.000 | | | | 14403.8973 | 14369.64 | 0.237 |
| | l 8 | 18106.6841 | 58504.4097 | 58504.6233 | 0.000 | | | | 20331.8919 | 20268.29 | 0.312 |
| | u 9 | 22617.9418 | 95071.9966 | 95070.8769 | 0.001 | | | | 25956.9975 | 25854.673 | 0.392 |
| | e 10 | 26528.8152 | | | | | | | 26699.5893 | 26699.76 | 0.000 |
| | s 11 | 27630.2281 | 33148.3963 | 32984.83 | 0.493 | | | | 33148.3963 | 32984.83 | 0.493 |
| | 12 | 33143.7430 | | | | | | | 42068.6960 | 41810.20 | 0.614 |
| | 13 | 39158.4865 | | | | | | | 52925.8963 | 52522.40 | 0.762 |
| | 14 | 45674.4587 | | | | | | | 54429.6523 | 54429.97 | 0.000 |
| | 15 | 52691.6595 | | | | | | | 65279.4975 | 64666.54 | 0.938 |
| | 16 | 53057.6305 | | | | | | | 76624.1348 | 75768.93 | 1.116 |
| | 17 | 60210.0889 | | | | | | | 84227.5928 | 84226.09 | 0.001 |
| | 18 | 68229.7470 | | | | | | | 117008.8194 | 117011.7 | -0.002 |
| | 19 | 76750.6336 | | | | | | | 153002.1231 | 153001.8 | 0.000 |
| | 20 | 79586.4458 | | | | | | | 190143.9932 | 190141.7 | 0.001 |
| | 21 | 85772.7490 | | | | | | | 221254.4869 | 221256.0 | 0.000 |

Notes:
1. This column contains the exact eigenvalues presented in Table 3
2. This row contains the number of elements in the model
3. This row contains the number of degrees of freedom
4. This row contains the number of CPU seconds the model took to run
SUPERB version of the Modified Jacobi technique for extracting eigenvalues. Therefore, there was no easy way to implement an element in DSTAP that would give the same results as SUPERB.

Based on this example, it can be concluded that the exact displacement beam element arrives at almost the same eigenvalues as if the characteristic equation had been solved. For practical purposes, the exact-element models gave the correct eigenvalues. The fixed-fixed beam was modeled with only two elements, three degrees of freedom. These three degrees of freedom were enough to compute all the eigenvalues below 60,000 rad/sec correctly to the ninth or tenth digits. Between 60,000 and 100,000 rad/sec, these three degrees of freedom were enough to compute all the eigenvalues to the fifth significant digit. The FEM models tested here were not able to match this feat. The best FEM result obtained was for the first eigenvalue of the 21-degrees-of-freedom model which matched the correct eigenvalues to 4 digits. The second conclusion arrived at here was that the beam finite element along with the EIGENR routine incorporated into DSTAP gave approximately the same results as SUPERB. SUPERB took about an order of a magnitude longer to carry out the computation. This is probably because the overhead in SUPERB is necessarily much greater than in DSTAP.

B. Portal Arch:

The second example presented here is the Portal Arch, which is shown in Fig. 9. It is made of three identical beams, which have the same properties as the fixed-fixed beam in the previous example. The legs of the Portal Arch are built into the ground, so that the translational and rotational degrees of freedom are eliminated.
Material Properties:

\[ E = 30.0 \text{ Mpsi} \ (207. \text{ GPa}) \]
\[ G = 11.6 \text{ Mpsi} \ (79.9 \text{ GPa}) \]
\[ \rho = 0.282 \text{ lb}_f/\text{in}^3 \ (7.81 \text{ mgr/mm}^3) \]

Figure 7. Schematic of the Beam for Example A.
The Portal Arch was first modeled with 3 exact beam elements to determine the first 35 or so eigenvalues. In this model, one element was assigned to each side of the arch, giving a total of 6 active degrees of freedom. This model was then input to DSTAP with the shear deformation and rotatory inertia options on. The first 35 eigenvalues obtained this way are presented in Table 5. Next, the same model was analyzed without shear deformation and rotatory inertia effects. These eigenvalues also appear in Table 5, along with the absolute and relative difference between the computed eigenvalues. The relative difference column in Table 5 shows that the shear deformation and rotatory inertia effects are not significant in the first 35 eigenvalues. This was expected since the arch beams meet the classical definition of slender beams; that is, the length to largest cross section dimension is greater than 8. Therefore, the shear deformation effects may be neglected for lower circular frequencies.

The eigenvectors corresponding to the eigenvalues with shear and rotatory inertia are presented in Fig. 10. These eigenvectors were reconstructed in the post-processing option of DSTAP, using the technique described in the Program Implementation section. This catalogue of mode shapes for the Portal Arch is believed to be unique. A sample output for the mode shape reconstruction in DSTAP is presented in Appendix D. This output corresponds to the 10th eigenvalue. That is, DSTAP was requested to reanalyze the Portal Arch so that the only eigenvalue it could find was the 10th eigenvalue. Then, DSTAP was requested to reconstruct the mode shape using the TMM. The output shown in Appendix D includes most of the optional output, such as: the initial state vector for each reconstructed element, the transfer matrix used for each exact-element, the full mode shape (deflection,
Table 5. Portal Arch Exact-Element Model Eigenvalues, With and Without Shear Deformation and Rotatory Inertia.

<table>
<thead>
<tr>
<th>dof: 6</th>
<th>1.150E07</th>
<th>infinite</th>
<th>iy (in)=+0.0722</th>
<th>zero</th>
<th>Absolute Difference</th>
<th>Relative Difference (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU sec.: 117.1900</td>
<td>111.4100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>81.3588</td>
<td>81.3702</td>
<td>0.01136</td>
<td>0.01396</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>321.0140</td>
<td>321.1035</td>
<td>0.08943</td>
<td>0.02785</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>523.5551</td>
<td>523.8114</td>
<td>0.25626</td>
<td>0.04894</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>567.5300</td>
<td>567.8924</td>
<td>0.36233</td>
<td>0.06384</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1145.9509</td>
<td>1146.9407</td>
<td>0.98971</td>
<td>0.08636</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1399.3720</td>
<td>1401.0730</td>
<td>1.70092</td>
<td>0.12154</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1618.2454</td>
<td>1620.6311</td>
<td>2.38564</td>
<td>0.14742</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>2454.8651</td>
<td>2459.1925</td>
<td>4.32734</td>
<td>0.17627</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>2898.4833</td>
<td>2905.0732</td>
<td>6.58981</td>
<td>0.22735</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>3055.0334</td>
<td>3063.0854</td>
<td>8.05194</td>
<td>0.26356</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>4265.4817</td>
<td>4278.1797</td>
<td>12.69799</td>
<td>0.29769</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>4751.2740</td>
<td>4768.5267</td>
<td>17.25264</td>
<td>0.36311</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>5100.3959</td>
<td>5121.2102</td>
<td>20.81425</td>
<td>0.40809</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>6543.7767</td>
<td>6573.0268</td>
<td>29.25008</td>
<td>0.46499</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>7241.8762</td>
<td>7280.4695</td>
<td>38.59325</td>
<td>0.53291</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>7484.3794</td>
<td>7527.8916</td>
<td>43.51216</td>
<td>0.58137</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>9271.7240</td>
<td>9328.1874</td>
<td>56.46331</td>
<td>0.60898</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>10046.0964</td>
<td>10119.1890</td>
<td>73.09255</td>
<td>0.72757</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>10446.7596</td>
<td>10525.9006</td>
<td>79.14098</td>
<td>0.75756</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>11979.3648</td>
<td>12032.0228</td>
<td>52.65791</td>
<td>0.43957</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>12898.8176</td>
<td>12917.9393</td>
<td>19.12165</td>
<td>0.14824</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>13171.2416</td>
<td>13260.7219</td>
<td>89.48030</td>
<td>0.67936</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>13526.9674</td>
<td>13655.9250</td>
<td>128.95754</td>
<td>0.95333</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>14301.7357</td>
<td>14416.1023</td>
<td>114.36653</td>
<td>0.79966</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>16569.3844</td>
<td>16746.9062</td>
<td>177.52179</td>
<td>1.07138</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>17188.0366</td>
<td>17392.3481</td>
<td>204.31141</td>
<td>1.18068</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>18017.0837</td>
<td>18247.5027</td>
<td>230.41900</td>
<td>1.27889</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>20660.9769</td>
<td>20939.7906</td>
<td>278.81365</td>
<td>1.34946</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>21680.0371</td>
<td>22007.5501</td>
<td>327.51297</td>
<td>1.51066</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>21932.2404</td>
<td>22239.2230</td>
<td>306.98254</td>
<td>1.39968</td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>24791.8520</td>
<td>24999.2945</td>
<td>207.44250</td>
<td>0.83673</td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>25332.9633</td>
<td>25754.6906</td>
<td>421.72724</td>
<td>1.66473</td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>27091.5871</td>
<td>27608.7668</td>
<td>517.17967</td>
<td>1.90900</td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>27366.0790</td>
<td>27746.7669</td>
<td>380.68782</td>
<td>1.39109</td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>30399.8477</td>
<td>31009.4317</td>
<td>609.58397</td>
<td>2.00522</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

+i_y is the radius of Gyration

*This eigenvalue was obtained with a 6-exact-element model. There were many poles around this eigenvalue in the 3-exact-element model, which made the extraction difficult.
Figure 10. Modes Shapes for the Portal Arch, See Table 11.
Figure 10. Modes Shapes for the Portal Arch, See Table 11. (continued)
Figure 10. Modes Shapes for the Portal Arch, See Table 11. (continued)
slope, moment, and shear), and the scaled mode shape (deflection only) for plotting.

The arch in Fig. 10 was also modeled with finite elements to determine whether the first 30 or so eigenvalues could be extracted satisfactorily. The first 30 eigenvalues were chosen because the finite element in DSTAP does not have shear deformation and rotatory inertia capabilities. It was seen in Table 5 that these effects do not become significant factors until the 34th eigenvalue. Having established a desired number of eigenvalues, the definition of a satisfactory eigenvalue had to be established. The set of acceptable eigenvalues for this study was defined as all lower eigenvalues up to the first occurrence of a relative difference of more than 5 percent. The relative difference used was between the finite element model eigenvalues and the exact eigenvalues in the third column of Table 5.

In every finite element model developed for the Portal Arch, each segment was divided into the same number of equal-size elements. This element distribution was chosen to keep the comparisons independent of the modeling technique used in each model. Hence, the first model used one element to model each side of the arch. The next model used two elements to model each side, and so on. The results obtained from these models are tabulated in Table 6.

The left side of Table 6 contains thirty correct eigenvalues from Table 5, for comparison purposes. The right side of Table 6 contains the results from the different finite element models in DSTAP. Moving from left to right, the number of degrees of freedom per model increases. The results of each model occupy two columns in the table. The top of the first column contains the model parameters, such as number of elements, eigenvalue
Table 6. Portal Arch Finite Element Model Eigenvalues.

<table>
<thead>
<tr>
<th>Program</th>
<th>DSTAP</th>
<th>DSTAP</th>
<th>DSTAP</th>
<th>DSTAP</th>
<th>DSTAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routine</td>
<td>BFMTHD</td>
<td>EIGENR</td>
<td>EIGENR</td>
<td>EIGENR</td>
<td>EIGENR</td>
</tr>
<tr>
<td># of FE</td>
<td>0</td>
<td>3</td>
<td>6</td>
<td>9</td>
<td>12</td>
</tr>
<tr>
<td># of CE</td>
<td>3</td>
<td>0</td>
<td>Rel.</td>
<td>0</td>
<td>Rel.</td>
</tr>
<tr>
<td>dof</td>
<td>6</td>
<td>6</td>
<td>Diff.</td>
<td>15 Diff.</td>
<td>24 Diff.</td>
</tr>
<tr>
<td>CPU sec</td>
<td>0.9300 (%)</td>
<td>1.3400 (%)</td>
<td>2.5700 (%)</td>
<td>5.1000 (%)</td>
<td></td>
</tr>
</tbody>
</table>

<p>| 1 | 81.3702 | 80.4074 -1.18 | 81.9956 0.76 | 81.7642 0.48 | 81.5163 0.30 |
| 2 | 321.1035 | 384.3153 19.68 | 336.6700 4.84 | 330.5853 2.95 | 327.0545 1.85 |
| 3 | 553.9114 | 737.3769 40.77 | 575.6669 9.89 | 561.8787 7.26 | 549.3870 4.88 |
| 4 | 567.8924 | 924.2121 152.7 | 600.9932 5.82 | 593.2593 4.46 | 585.8758 3.16 |
| 5 | 1146.9407 | 11565.8657 908.4 | 1259.4344 9.80 | 1209.6473 5.46 | 1197.2259 4.38 |
| 7 | 1620.6311 | 2080.7392 28.39 | 1849.2565 14.10 | 1807.8869 11.55 |
| 8 | 2459.1925 | 3242.5577 31.85 | 2698.8713 9.74 | 2611.5018 6.19 |
| 9 | 2905.0732 | 4176.0193 43.74 | 3182.3908 9.54 | 3499.1263 20.44 |
| 10 | 3063.0854 | 11901.2435 | 3564.0379 15.35 | 3599.9348 17.52 |
| 11 | 4278.1797 | 12745.0689 | 4739.0393 10.77 | 4730.1071 10.56 |
| 12 | 4768.5267 | 21370.4003 | 5492.1177 15.17 | 5257.0530 10.24 |
| 13 | 5121.2102 | 37112.7949 | 6837.0365 33.50 | 5680.8305 10.92 |
| 14 | 6573.0268 | 37224.5293 | 8965.6496 36.40 | 6990.6763 6.35 |
| 15 | 7280.4695 | 45174.2353 | 9921.7519 36.27 | 7731.9373 6.20 |
| 16 | 7527.9816 | 13095.8040 73.96 | 8939.3861 18.75 |
| 17 | 9328.1874 | 13869.7972 48.68 | 10166.1388 9.98 |
| 18 | 10119.1890 | 22821.8220 125.5 | 10546.4189 4.22 |
| 19 | 10525.9006 | 40682.5523 286.4 | 13038.1753 23.86 |
| 20 | 12032.0228 | 40790.6590 239.0 | 14935.1844 24.12 |
| 21 | 12917.9393 | 48380.5309 274.5 | 15450.5719 19.60 |
| 22 | 13260.7219 | 64168.0204 383.8 | 18077.6504 36.32 |
| 23 | 13655.9250 | 64193.9858 370.0 | 20193.4744 47.87 |
| 24 | 14416.1023 | 74811.8918 418.9 | 23643.3254 64.00 |
| 25 | 16746.9062 | 42082.7056 151.2 |
| 26 | 17392.3481 | 42147.7442 142.3 |
| 27 | 18247.5027 | 49269.4254 170.0 |
| 28 | 20939.7906 | 70524.0782 236.7 |
| 29 | 22007.5501 | 70545.9382 220.5 |
| 30 | 22239.2230 | 77879.8456 250.1 |</p>
<table>
<thead>
<tr>
<th>Program</th>
<th>DSTAP</th>
<th>DSTAP</th>
<th>DSTAP</th>
<th>DSTAP</th>
<th>DSTAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routine</td>
<td>BFMTHD</td>
<td>EIGENR</td>
<td>EIGENR</td>
<td>EIGENR</td>
<td>EIGENR</td>
</tr>
<tr>
<td># of FE</td>
<td>0</td>
<td>15</td>
<td>21</td>
<td>27</td>
<td>33</td>
</tr>
<tr>
<td># of CE</td>
<td>3</td>
<td>0</td>
<td>Rel. 0</td>
<td>Rel. 0</td>
<td>Rel. 0</td>
</tr>
<tr>
<td>dof</td>
<td>6</td>
<td>42 Diff.</td>
<td>60 Diff.</td>
<td>78 Diff.</td>
<td>96 Diff.</td>
</tr>
<tr>
<td>CPU sec</td>
<td>9.3900 (%)</td>
<td>25.8500 (%)</td>
<td>55.6600 (%)</td>
<td>106.8700 (%)</td>
<td></td>
</tr>
</tbody>
</table>

1. 81.3702
2. 321.1035
3. 523.8114
4. 567.8924
5. 1146.9407
6. 1401.0730
7. 1520.6311
8. 2459.1925
9. 2905.0732
10. 3063.0854
11. 4278.1797
12. 4768.5267
13. 5121.2102
14. 6573.0268
15. 7280.4695
16. 7527.8916
17. 9328.1874
18. 10119.1890
19. 10525.9006
20. 12032.0228
21. 12917.4939
22. 13260.7219
23. 13655.9250
24. 14416.1023
25. 16746.9062
26. 17392.3481
27. 18247.5027
28. 20939.7906
29. 22007.5501
30. 22239.2230
Table 6. Portal Arch Finite Element Model Eigenvalues (continued)

<table>
<thead>
<tr>
<th>Program Routine</th>
<th>DSTAP BFMTHD</th>
<th>DSTAP EIGENR</th>
<th>DSTAP EIGENR</th>
<th>DSTAP EIGENR</th>
<th>DSTAP EIGENR</th>
</tr>
</thead>
<tbody>
<tr>
<td># of CE dof=6</td>
<td>0 Rel. 114</td>
<td>0 Rel. 132</td>
<td>0 Rel. 168</td>
<td>0 Rel. 204</td>
<td></td>
</tr>
<tr>
<td>CPU sec</td>
<td>176.1500 (%)</td>
<td>279.9700 (%)</td>
<td>584.4800 (%)</td>
<td>1032.0000 (%)</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>81.3702</td>
<td>81.3963</td>
<td>81.3898</td>
<td>81.3786</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>321.1035</td>
<td>321.7434</td>
<td>321.5857</td>
<td>321.3099</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>523.8114</td>
<td>526.7223</td>
<td>526.0073</td>
<td>524.7530</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>567.8924</td>
<td>570.0706</td>
<td>569.5381</td>
<td>568.9249</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1146.9407</td>
<td>1154.3070</td>
<td>1152.5424</td>
<td>1149.3769</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1401.0730</td>
<td>1427.402</td>
<td>1421.2381</td>
<td>1419.7538</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1620.6311</td>
<td>1647.5429</td>
<td>1641.0021</td>
<td>1633.4367</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>2459.1925</td>
<td>2497.0450</td>
<td>2488.5477</td>
<td>2478.1763</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>2905.0732</td>
<td>3022.2692</td>
<td>2994.2887</td>
<td>2961.4920</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>3063.0854</td>
<td>3158.3254</td>
<td>3135.0625</td>
<td>3108.2033</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>4278.1797</td>
<td>4374.4641</td>
<td>4354.3973</td>
<td>4328.7781</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>4768.5267</td>
<td>5095.9704</td>
<td>5019.5644</td>
<td>4928.3697</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>5121.2102</td>
<td>5419.3396</td>
<td>5347.7773</td>
<td>5263.7323</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>6573.0268</td>
<td>6774.4755</td>
<td>6738.0148</td>
<td>6687.8552</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>7280.4695</td>
<td>8012.5320</td>
<td>7851.0672</td>
<td>7649.6756</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>7527.8916</td>
<td>8180.5893</td>
<td>8025.3242</td>
<td>7839.8993</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>9328.1874</td>
<td>9649.2191</td>
<td>9589.5782</td>
<td>9511.1198</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>10119.1890</td>
<td>11494.0117</td>
<td>11214.0516</td>
<td>10840.6480</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>10525.9006</td>
<td>11841.2802</td>
<td>11561.9707</td>
<td>11194.6203</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>12032.0228</td>
<td>12207.0396</td>
<td>12166.5590</td>
<td>12121.1268</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>12917.9393</td>
<td>12937.0282</td>
<td>12929.1292</td>
<td>12929.3954</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>13260.7219</td>
<td>13964.4636</td>
<td>13961.0904</td>
<td>13916.1693</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>13655.9250</td>
<td>15911.9671</td>
<td>15515.3112</td>
<td>14921.0303</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>14416.1023</td>
<td>16222.5769</td>
<td>15788.9241</td>
<td>15218.6317</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>16746.9062</td>
<td>17714.6553</td>
<td>17559.0009</td>
<td>17358.2196</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>17392.3481</td>
<td>20739.1000</td>
<td>20247.4633</td>
<td>20400.2704</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>18247.5027</td>
<td>21620.9948</td>
<td>21036.0720</td>
<td>22009.6591</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>20939.7906</td>
<td>22081.5786</td>
<td>21909.9113</td>
<td>21714.9020</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>22007.5501</td>
<td>25258.3320</td>
<td>24916.5849</td>
<td>24204.2912</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>22239.2230</td>
<td>26197.5452</td>
<td>25844.4662</td>
<td>24996.6103</td>
<td></td>
</tr>
</tbody>
</table>

64
Table 6. Portal Arch Finite Element Model Eigenvalues (continued)

<table>
<thead>
<tr>
<th>Program Routine</th>
<th>DSTAP BFMTHD</th>
<th>DSTAP EIGENR</th>
<th>DSTAP EIGENR</th>
</tr>
</thead>
<tbody>
<tr>
<td># of CE</td>
<td>3 Rel. 0</td>
<td>6 Diff. 240</td>
<td>0 Rel.</td>
</tr>
<tr>
<td>CPU sec</td>
<td>1712.4700 (%)</td>
<td>3219.3600 (%)</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>81.3702</td>
<td>81.3763</td>
<td>81.3748</td>
</tr>
<tr>
<td>2</td>
<td>321.1035</td>
<td>321.2534</td>
<td>321.2172</td>
</tr>
<tr>
<td>3</td>
<td>523.8114</td>
<td>524.4956</td>
<td>524.3307</td>
</tr>
<tr>
<td>4</td>
<td>567.8924</td>
<td>568.4068</td>
<td>568.2829</td>
</tr>
<tr>
<td>5</td>
<td>1146.9407</td>
<td>1148.7160</td>
<td>1148.2906</td>
</tr>
<tr>
<td>6</td>
<td>1401.0730</td>
<td>1407.3852</td>
<td>1405.8657</td>
</tr>
<tr>
<td>7</td>
<td>1620.6311</td>
<td>1627.0196</td>
<td>1625.4826</td>
</tr>
<tr>
<td>8</td>
<td>2459.1925</td>
<td>2468.8961</td>
<td>2466.6046</td>
</tr>
<tr>
<td>9</td>
<td>2905.0732</td>
<td>2933.3501</td>
<td>2926.5687</td>
</tr>
<tr>
<td>10</td>
<td>3063.0854</td>
<td>3085.5221</td>
<td>3080.1083</td>
</tr>
<tr>
<td>11</td>
<td>4278.1797</td>
<td>4304.6812</td>
<td>4298.5464</td>
</tr>
<tr>
<td>12</td>
<td>4768.5267</td>
<td>4849.0175</td>
<td>4829.7706</td>
</tr>
<tr>
<td>13</td>
<td>5121.2102</td>
<td>5192.1148</td>
<td>5174.9888</td>
</tr>
<tr>
<td>14</td>
<td>6573.0268</td>
<td>6636.0668</td>
<td>6622.0646</td>
</tr>
<tr>
<td>15</td>
<td>7280.4695</td>
<td>7468.4420</td>
<td>7423.8097</td>
</tr>
<tr>
<td>16</td>
<td>7527.8916</td>
<td>7681.4723</td>
<td>7643.9638</td>
</tr>
<tr>
<td>17</td>
<td>9328.1874</td>
<td>9430.7726</td>
<td>9408.6387</td>
</tr>
<tr>
<td>18</td>
<td>10119.1890</td>
<td>10490.4252</td>
<td>10402.8015</td>
</tr>
<tr>
<td>19</td>
<td>10525.906</td>
<td>10860.6685</td>
<td>10779.4943</td>
</tr>
<tr>
<td>20</td>
<td>12032.0228</td>
<td>12080.7604</td>
<td>12070.2249</td>
</tr>
<tr>
<td>21</td>
<td>12917.9393</td>
<td>12923.8883</td>
<td>12923.3097</td>
</tr>
<tr>
<td>22</td>
<td>13260.7219</td>
<td>13626.2904</td>
<td>13557.0183</td>
</tr>
<tr>
<td>23</td>
<td>13555.9250</td>
<td>14317.8028</td>
<td>14162.5584</td>
</tr>
<tr>
<td>24</td>
<td>14416.1023</td>
<td>14752.8315</td>
<td>14656.8390</td>
</tr>
<tr>
<td>25</td>
<td>16746.9062</td>
<td>17136.4133</td>
<td>17065.7314</td>
</tr>
<tr>
<td>26</td>
<td>17392.3481</td>
<td>18466.0236</td>
<td>18217.9854</td>
</tr>
<tr>
<td>27</td>
<td>18247.5027</td>
<td>19156.5597</td>
<td>18925.4086</td>
</tr>
<tr>
<td>28</td>
<td>20939.7906</td>
<td>21501.6349</td>
<td>21422.0001</td>
</tr>
<tr>
<td>29</td>
<td>22007.5501</td>
<td>23255.7946</td>
<td>22992.4234</td>
</tr>
<tr>
<td>30</td>
<td>22239.2230</td>
<td>23706.5373</td>
<td>23326.1119</td>
</tr>
</tbody>
</table>
extraction routine, and execution time. For example, the first model in Table 6 has 3 finite elements, and it took 0.93 of a CPU second to find the eigenvalues, using EIGENR. The 6 degrees of freedom allowed the extraction of 6 eigenvalues. These eigenvalues appear below the model data, in the same column. The second column allotted to each model contains the relative difference between model eigenvalues and the exact eigenvalues, expressed as a percentage. For example, the 6 degree of freedom model only has one eigenvalue with a relative difference less than 5 per cent, the first one. The remaining eigenvalues for this model have relative differences greater than 5 per cent.

Note, the acceptable eigenvalue obtained for the 6-degrees-of-freedom model is questionable. Meirovitch [6] has shown that the FEM eigenvalue problem is a form of Rayleigh-Ritz approximation. This guarantees that the eigenvalues obtained with a FEM discretization will be upper bounds on the true eigenvalues; in other words, they should be greater than the exact eigenvalues. This is true for all the eigenvalues in Table 6, except for the first eigenvalue obtained with the 6-degree-of-freedom model. This eigenvalue is less than the exact eigenvalue, indicating that this is an unacceptable result. However, in practice, this eigenvalue would be considered acceptable; thus, this eigenvalue will be considered acceptable here.

Table 6 shows that indeed the finite element models in DSTAP are capable of finding the first 30 or so eigenvalues within the set tolerance. In Table 6, a heavy line separates the acceptable eigenvalues, above the line, from the rest. It is possible that some eigenvalues lying below the line may meet the specified tolerance of 5 per cent. This is because the line was drawn below
the first eigenvalue which had a relative difference greater than 5 per cent. Moving from left to right, this line moves downward, indicating that more acceptable eigenvalues are extracted by the models with more degrees of freedom. Thus, as the degrees of freedom increase, so do the number of acceptable eigenvalues.

The number of acceptable eigenvalues per model size was entered into a statistical package [62], which automatically tested the following 8 regression models,

1. $y = Ax$,
2. $y = A + Bx$,
3. $y = A \exp(x)$,
4. $y = 1/(A + Bx)$,
5. $y = A + B/x$,
6. $y = A + B \log(x)$,
7. $y = A x^B$, and
8. $y = x/(A + Bx)$,

where $x$ is the number of degrees of freedom, $y$ is the number of acceptable eigenvalues obtained with a model of $x$ degrees of freedom, $A$ and $B$ are regression constants. From these models the best fit for the finite element models was provided by the equation for a straight line (model 2),

$$y = -0.302 + (0.10298) x$$

(31)

A plot of this line may be seen in Fig. 11. This regression line may be used to approximate the number of correct eigenvalues knowing the total number of active degrees of freedom in a model. Since one of the question to be answered was "how many active nodes are needed per acceptable eigenvalue?", Eq. (31) is used in reverse form below. Being that Eq. (31) is a regression of $x$ on $y$, it is improper to use it for a $y$ on $x$ calculation. Nonetheless, this is done here because the answer sought is an approximate answer and the answer is not expected to vary with a proper regression. Thus, $(y + 0.302)/0.103$
Figure 11. Number of Acceptable Eigenvalues versus Degrees of Freedom for Portal Arch.
active degrees are freedom are needed for the first \( y \) eigenvalues to be acceptable, which is approximately equal to \( 10y + 3 \). Since there are 3 degrees of freedom per node, then approximately between 3 and 4 active nodes are needed per acceptable eigenvalue.

The regression line in Eq. (31) is also a verification of the proof, as presented by Strang and Fix [13], that a computed eigenvalue can be made sufficiently close to a true eigenvalue by increasing the number of degrees of freedom. Consequently, regardless of how many acceptable eigenvalues are desired, it should be possible to extract them with a sufficiently large model, provided the computer is able to handle the numerical difficulties associated with larger matrices. As presented earlier, it is generally not feasible to increase the size of the model arbitrarily.

In Fig. 11, there is a second curve that corresponds to the finite element—continuous element (FEC) models. The slope of this curve is much steeper than the one for finite elements alone, indicating that fewer finite elements are required to compute the same number of acceptable eigenvalues. The model used to generate this curve had both of the arch legs modeled with exact beam elements, while the top was modeled with finite elements. The number of degrees of freedom in these elements were increased by adding more finite elements to the top of the arch. The data for the FEC curve are presented in Table 7, which is organized similarly to Table 6. The final line in Fig. 11 is a vertical line at 6 dof. This line corresponds to the model made of purely exact beam elements, which was presented in Table 5. Note that no data points are presented for this last line because theoretically any number of eigenvalues can be obtained with the 6-degree-of-freedom model!

The results for the FEM models analyzed here indicate that at least for
Table 7. Portal Arch Combined Model (FEC) Eigenvalues.

<table>
<thead>
<tr>
<th>Program Routine</th>
<th>DSTAP BFMTHD</th>
<th>DSTAP BFMTHD</th>
<th>DSTAP BFMTHD</th>
<th>DSTAP BFMTHD</th>
<th>DSTAP BFMTHD</th>
</tr>
</thead>
<tbody>
<tr>
<td># of FE # of CE</td>
<td># of dof CPU sec</td>
<td># of dof CPU sec</td>
<td># of dof CPU sec</td>
<td># of dof CPU sec</td>
<td># of dof CPU sec</td>
</tr>
<tr>
<td>0 3</td>
<td>6</td>
<td>111.4100 84.5700 (%)</td>
<td>117.8800 (%)</td>
<td>151.3800 (%)</td>
<td>223.1700 (%)</td>
</tr>
<tr>
<td>1 3</td>
<td>6</td>
<td>81.3702</td>
<td>81.3704 0.00</td>
<td>81.3702 0.00</td>
<td>81.3702 0.00</td>
</tr>
<tr>
<td>2 3</td>
<td>6</td>
<td>321.1035 351.9085 9.59</td>
<td>322.4091 0.40</td>
<td>321.1924 0.02</td>
<td></td>
</tr>
<tr>
<td>3 3</td>
<td>6</td>
<td>523.8114 524.9729 0.22</td>
<td>524.4229 0.11</td>
<td>523.9031 0.01</td>
<td></td>
</tr>
<tr>
<td>4 3</td>
<td>6</td>
<td>567.8924 739.5138 30.22</td>
<td>570.9586 0.53</td>
<td>569.1431 0.04</td>
<td></td>
</tr>
<tr>
<td>5 3</td>
<td>6</td>
<td>1146.9407 1303.5349 13.65</td>
<td>1157.8771 0.95</td>
<td>1150.8222 0.33</td>
<td></td>
</tr>
<tr>
<td>6 3</td>
<td>6</td>
<td>1401.0730 1619.3473 15.57</td>
<td>1402.2725 0.08</td>
<td>1402.0894 0.02</td>
<td></td>
</tr>
<tr>
<td>7 3</td>
<td>6</td>
<td>1620.6311 2153.2601 32.86</td>
<td>1777.4224 9.67</td>
<td>1624.8925 0.26</td>
<td></td>
</tr>
<tr>
<td>8 3</td>
<td>6</td>
<td>2459.1925 3091.3132 25.70</td>
<td>2668.2318 8.50</td>
<td>2488.1906 1.17</td>
<td></td>
</tr>
<tr>
<td>9 3</td>
<td>6</td>
<td>2905.0732 3376.2028 16.21</td>
<td>2979.9512 2.57</td>
<td>2912.9787 0.27</td>
<td></td>
</tr>
<tr>
<td>10 3</td>
<td>6</td>
<td>3063.0854 5078.1218 65.78</td>
<td>4035.4435 31.74</td>
<td>3086.0232 0.74</td>
<td></td>
</tr>
<tr>
<td>11 3</td>
<td>6</td>
<td>4278.1797 5271.8786 23.22</td>
<td>4668.4929 9.12</td>
<td>4527.5265 5.82</td>
<td></td>
</tr>
<tr>
<td>12 3</td>
<td>6</td>
<td>4768.5267 7563.6879 58.61</td>
<td>5554.7787 16.48</td>
<td>4772.5707 0.08</td>
<td></td>
</tr>
<tr>
<td>13 3</td>
<td>6</td>
<td>5121.2102 7727.9307 50.90</td>
<td>6356.0937 24.11</td>
<td>5452.8584 6.47</td>
<td></td>
</tr>
<tr>
<td>14 3</td>
<td>6</td>
<td>6573.0268 10531.9099 60.22</td>
<td>7781.2465 18.38</td>
<td>6921.5856 5.30</td>
<td></td>
</tr>
<tr>
<td>15 3</td>
<td>6</td>
<td>7280.4695 10682.2848 46.72</td>
<td>8207.2945 12.73</td>
<td>7588.5255 4.23</td>
<td></td>
</tr>
<tr>
<td>16 3</td>
<td>6</td>
<td>7527.8918 12277.5529 63.09</td>
<td>10068.7764 40.92</td>
<td>8414.0948 11.77</td>
<td></td>
</tr>
<tr>
<td>17 3</td>
<td>6</td>
<td>9328.1874 12704.1274 36.19</td>
<td>10825.9317 16.05</td>
<td>10005.4207 7.26</td>
<td></td>
</tr>
<tr>
<td>18 3</td>
<td>6</td>
<td>10119.1890 14021.3793 38.56</td>
<td>12878.3789 27.26</td>
<td>10348.0909 2.26</td>
<td></td>
</tr>
<tr>
<td>19 3</td>
<td>6</td>
<td>10525.9006 14296.0815 35.81</td>
<td>12995.2982 23.46</td>
<td>11688.0257 12.75</td>
<td></td>
</tr>
<tr>
<td>20 3</td>
<td>6</td>
<td>12032.0228 17905.8595 48.81</td>
<td>14283.5372 18.71</td>
<td>12752.7386 5.98</td>
<td></td>
</tr>
<tr>
<td>21 3</td>
<td>6</td>
<td>12917.9393 18248.0941 41.26</td>
<td>14625.7616 13.22</td>
<td>13294.3528 2.91</td>
<td></td>
</tr>
<tr>
<td>22 3</td>
<td>6</td>
<td>13260.7219 22089.1691 66.57</td>
<td>18018.2383 35.87</td>
<td>13459.8030 1.50</td>
<td></td>
</tr>
<tr>
<td>23 3</td>
<td>6</td>
<td>13655.9250 22742.8013 66.54</td>
<td>18386.0679 34.62</td>
<td>15009.6078 9.91</td>
<td></td>
</tr>
<tr>
<td>24 3</td>
<td>6</td>
<td>14416.1023 25817.2154 79.08</td>
<td>22151.9892 53.66</td>
<td>16995.1017 17.88</td>
<td></td>
</tr>
<tr>
<td>25 3</td>
<td>6</td>
<td>16764.9062 27745.4060 65.67</td>
<td>22801.9465 36.15</td>
<td>17664.8516 5.48</td>
<td></td>
</tr>
<tr>
<td>26 3</td>
<td>6</td>
<td>17392.3481 29038.8182 66.96</td>
<td>25823.9772 48.47</td>
<td>20712.8523 19.09</td>
<td></td>
</tr>
<tr>
<td>27 3</td>
<td>6</td>
<td>18247.5027 33784.7939 85.14</td>
<td>27736.4626 52.00</td>
<td>21284.4572 16.64</td>
<td></td>
</tr>
<tr>
<td>28 3</td>
<td>6</td>
<td>20939.7906 29056.0230 38.75</td>
<td>25436.6690 21.47</td>
<td>24166.7261 15.41</td>
<td></td>
</tr>
<tr>
<td>29 3</td>
<td>6</td>
<td>22007.5501 33791.1110 53.54</td>
<td>27952.4897 27.01</td>
<td>24914.5367 13.20</td>
<td></td>
</tr>
<tr>
<td>30 3</td>
<td>6</td>
<td>22239.2230</td>
<td>28780.6913 29.41</td>
<td>25271.9573 13.63</td>
<td></td>
</tr>
</tbody>
</table>
Table 7. Portal Arch Combined Model (FEC) Eigenvalues (continued).

<table>
<thead>
<tr>
<th>Program</th>
<th>DSTAP</th>
<th>DSTAP</th>
<th>DSTAP</th>
<th>DSTAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routine</td>
<td>BFMTHD</td>
<td>BFMTHD</td>
<td>BFMTHD</td>
<td>BFMTHD</td>
</tr>
<tr>
<td># of FE</td>
<td>0</td>
<td>5</td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td># of CE</td>
<td>3</td>
<td>2</td>
<td>Rel. 2</td>
<td>Rel. 2</td>
</tr>
<tr>
<td>dof</td>
<td>6</td>
<td>18</td>
<td>Diff. 24</td>
<td>Diff. 33</td>
</tr>
<tr>
<td>CPU sec</td>
<td>111.4100</td>
<td>267.8800</td>
<td>(76)</td>
<td>459.2200</td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>1</td>
<td>81.3702</td>
<td>81.3702</td>
<td>0.00</td>
<td>81.3702</td>
</tr>
<tr>
<td>2</td>
<td>321.1035</td>
<td>321.1035</td>
<td>0.01</td>
<td>321.1058</td>
</tr>
<tr>
<td>3</td>
<td>523.8114</td>
<td>523.8114</td>
<td>0.00</td>
<td>523.8123</td>
</tr>
<tr>
<td>4</td>
<td>567.8924</td>
<td>567.8924</td>
<td>0.01</td>
<td>567.8932</td>
</tr>
<tr>
<td>5</td>
<td>1146.9407</td>
<td>1146.9407</td>
<td>0.14</td>
<td>1147.0480</td>
</tr>
<tr>
<td>6</td>
<td>1401.0730</td>
<td>1401.0730</td>
<td>0.01</td>
<td>1401.0892</td>
</tr>
<tr>
<td>7</td>
<td>1620.6311</td>
<td>1620.6311</td>
<td>0.11</td>
<td>1620.7522</td>
</tr>
<tr>
<td>8</td>
<td>2459.1925</td>
<td>2459.1925</td>
<td>0.55</td>
<td>2460.1436</td>
</tr>
<tr>
<td>9</td>
<td>2905.0732</td>
<td>2905.0732</td>
<td>0.09</td>
<td>2905.2610</td>
</tr>
<tr>
<td>10</td>
<td>3063.0854</td>
<td>3063.0854</td>
<td>0.46</td>
<td>3064.0750</td>
</tr>
<tr>
<td>11</td>
<td>4278.1797</td>
<td>4278.1797</td>
<td>1.48</td>
<td>4283.4373</td>
</tr>
<tr>
<td>12</td>
<td>4768.5267</td>
<td>4768.5267</td>
<td>0.25</td>
<td>4769.2575</td>
</tr>
<tr>
<td>13</td>
<td>5121.2102</td>
<td>5121.2102</td>
<td>0.67</td>
<td>5125.1104</td>
</tr>
<tr>
<td>14</td>
<td>6573.0268</td>
<td>6573.0268</td>
<td>4.96</td>
<td>6590.6318</td>
</tr>
<tr>
<td>15</td>
<td>7280.4695</td>
<td>7280.4695</td>
<td>0.10</td>
<td>7283.5353</td>
</tr>
<tr>
<td>16</td>
<td>7527.8916</td>
<td>7527.8916</td>
<td>6.54</td>
<td>7540.9813</td>
</tr>
<tr>
<td>17</td>
<td>9328.1874</td>
<td>9328.1874</td>
<td>5.42</td>
<td>9380.6951</td>
</tr>
<tr>
<td>18</td>
<td>10119.1890</td>
<td>10119.1890</td>
<td>0.79</td>
<td>10125.5586</td>
</tr>
<tr>
<td>19</td>
<td>10525.9006</td>
<td>10525.9006</td>
<td>6.55</td>
<td>10550.1468</td>
</tr>
<tr>
<td>20</td>
<td>12032.0228</td>
<td>12032.0228</td>
<td>4.67</td>
<td>12117.5350</td>
</tr>
<tr>
<td>21</td>
<td>12917.9339</td>
<td>12917.9339</td>
<td>1.72</td>
<td>12949.1544</td>
</tr>
<tr>
<td>22</td>
<td>13260.7219</td>
<td>13260.7219</td>
<td>0.79</td>
<td>13272.3267</td>
</tr>
<tr>
<td>23</td>
<td>13655.9250</td>
<td>13655.9250</td>
<td>2.98</td>
<td>13679.9656</td>
</tr>
<tr>
<td>24</td>
<td>14416.1023</td>
<td>14416.1023</td>
<td>14.31</td>
<td>14531.1908</td>
</tr>
<tr>
<td>25</td>
<td>16746.9062</td>
<td>16746.9062</td>
<td>4.36</td>
<td>16915.6607</td>
</tr>
<tr>
<td>26</td>
<td>17392.3481</td>
<td>17392.3481</td>
<td>5.67</td>
<td>17425.9243</td>
</tr>
<tr>
<td>27</td>
<td>18247.5027</td>
<td>18247.5027</td>
<td>17.15</td>
<td>18436.7089</td>
</tr>
<tr>
<td>28</td>
<td>20939.7906</td>
<td>20939.7906</td>
<td>3.44</td>
<td>21179.2150</td>
</tr>
<tr>
<td>29</td>
<td>22007.5501</td>
<td>22007.5501</td>
<td>11.67</td>
<td>22082.7241</td>
</tr>
<tr>
<td>30</td>
<td>22239.2230</td>
<td>22239.2230</td>
<td>13.14</td>
<td>22543.3575</td>
</tr>
</tbody>
</table>
beams the FEM is able to extract midrange eigenvalues successfully. These models do require a larger number of degrees of freedom than the exact-element models. Nevertheless, there does not seem to be a need for an enhancement. But, the structure presented here is a small, simple structure. In practice, the structures analyzed are large and complex. This makes the addition of degrees of freedom almost impossible. In fact most large structures need to be solved through some form of dynamic substructuring. Sung and Nefske [3] have reported difficulties extracting acceptable eigenvalues when substructuring large models, and Thomas [29] has proven that the errors due to dynamic substructuring increase for larger number of degrees of freedom or for higher frequencies. The small test models presented here may have actually helped the FEM eigenvalue extraction. But even for these small models, if the solution times for the different models presented here are compared, then the advantage of the proposed enhancement is quickly seen.

It is more inexpensive to extract higher eigenvalues with exact-beam-element models. The solution time, for each model, reported in Tables 6 and 7 were divided by the number of acceptable eigenvalues obtained with that model. The result is shown in Fig. 12, which clearly shows that the cost of extracting eigenvalues within the set tolerance increases for larger finite element models. The data was input into the statistical package [62], which tested the 8 regression models presented earlier. From these models the better fits were provided by the models: (1) \(-12.915 + 0.348x\) and (2) \(y=(0.00309)x(1.81562)\). The residual norms for the linear model was less than one percent lower than the residual norm for the second model. Since the data in Fig. 12 seems to be ascending in a parabolic fashion, the second regression model is chosen over the first regression model. Based on this
Figure 12. CPU seconds per Acceptable Eigenvalue versus Model Size for Portal Arch.
regression, the finite element models used should have few degrees of freedom to reduce the cost of the eigenvalue extraction. However, the only way that the first 30 acceptable eigenvalues of the Portal Arch can be obtained is by using at least 276 degrees of freedom. The cost of this extraction is 100.6 CPU sec per acceptable eigenvalue. On the other hand, the CPU cost per correct eigenvalue with a purely exact element model was only 3.18 CPU secs per correct eigenvalue. This leads to the obvious conclusion that, for the Portal Arch, whenever the finite element model has more than 46 degrees of freedom then it less expensive to use an exact model. This break-even point corresponds to roughly 4 acceptable eigenvalues. In other words, when 4 or less acceptable eigenvalues are needed then it is more economical to use a FEM model, but when more that 4 acceptable eigenvalues are needed then it is more economical to use an exact model.

It is also more economical to extract the higher eigenvalues with combined models. Figure 12 has a set of points toward the left of the curve, labeled FEC. These points correspond to the combined FEM-exact models. These points are too scattered to perform a reliable regression analysis. But, neglecting the FEC model with only 1 FEM beam element, the remaining FEC models seem to be scattered about the 25 CPU sec per acceptable eigenvalue line. Using this last value for a break-even analysis, whenever more than 142 degrees of freedom are required in a finite element model, then it is more economical to use the combined model presented here. This corresponds to 15 acceptable eigenvalues. Therefore, the break-even point for the FEC models used here is 15 acceptable eigenvalues.

The results presented above for the FEC models are only for the FEC models in which the legs of the Portal Arch are modeled with exact-beam
elements. Obviously, the results of this analysis would change if the Portal Arch is modeled with a different combination of FEM-elements and exact-elements. Thus, it is impossible to escape modeling considerations when studying the FEC models. This implies that depending on an analyst's choice of location and quantity of exact-elements the cost versus accuracy comparison may vary. It seems natural that since the eigenvector of the exact elements can be reconstructed with almost any degree of precision desired, the analyst should tend to place these exact elements in areas where much detail is required. That is, replace as many finite elements as possible with a single exact displacement element.

C. Verdeel Truss

The final example is presented in Fig. 13. In the literature, this structure is sometimes called a Verdeel Truss. This name is really a misnomer because the structure is made up of beams rather than trusses. The beams in this example have the same cross-section dimensions and material properties as the beams in the previous two examples. The first objective of this example is to verify the conclusions arrived at in the Portal Arch example. This leads to the following tasks, (1) determine whether finite element models can obtain a certain number of eigenvalues within the specified tolerance (2) ascertain whether the eigenvalue extraction can be improved by using a combination of FEM elements and exact elements. The second objective in this example is to compare solution efficiency with a commercial code.

The first model created for this example was the exact-beam model. This model was made up of 6 exact beam elements, with a total of 12 degrees of
Material Properties:

- $E = 30.0 \text{ Mpsi} (207. \text{ GPa})$
- $G = 11.6 \text{ Mpsi} (79.9 \text{ GPa})$
- $\rho = 0.282 \text{ lb}_f/\text{in}^3 (7.81 \text{ mg}/\text{mm}^3)$

Figure 13. Schematic of the Beam for Example C.
freedom. As in the previous example, this model was analyzed with DSTAP, both with and without the shear deformation and rotatory inertia effects. These results are presented in Table 8, along with the relative difference between the eigenvalues. According to the results in Table 8, the shear deformation and rotatory effects do not have a noticeable impact on the computed eigenvalues in the frequency range specified to DSTAP.

The reconstructed eigenvectors from the analysis with shear deformation and rotatory inertia are presented in Fig. 14. As was the case with the Portal Arch, this catalog of the first 48 mode shapes for the Verdeel Truss is believed to be unique. This catalogue further shows that the incorporation of the TMM into the eigenvector extraction is more desirable than the approximate methods for mode shape determination. For example, Hopper and Williams [31] tested their approximate mode extraction technique on a Verdeel Truss similar to the one for this example. With their approximation they obtained a possible 6th and 7th modes, as shown in Fig. 15. Comparing their approximated mode shapes with the catalog of mode shapes in Fig. 16, it is evident that their mode shapes resemble the 7th and 9th mode shapes more closely than the 6th and 7th. However, even the 7th and 9th modes do not exhibit the relatively stationary substructure as depicted in Fig. 15.

Having determined from Table 8 that shear deformation and rotatory inertia are minor factors in the first 48 eigenvalues, the number of desired acceptable eigenvalues was set at 35. This is an increase from the 30 acceptable eigenvalues sought in the Portal Arch example. The criteria for acceptable eigenvalues remains the same for this example; that is, all lower eigenvalues up to the first occurrence of a 5 per cent relative difference were are acceptable. The relative difference was computed for the FEM and
Table 8. Verdeel Truss Exact-Element Model Eigenvalues, With and Without Shear Deformation and Rotatory Inertia.

<table>
<thead>
<tr>
<th>Program Routine</th>
<th>DSTAP BFMTHD</th>
<th>DSTAP BFMTHD</th>
<th>G((\psi))= 1.1500E7 infinite</th>
<th>iy(in)= 0.0721 zero Absolute Relative</th>
</tr>
</thead>
<tbody>
<tr>
<td>dof CPU sec</td>
<td>12 260.4600</td>
<td>12 258.1900</td>
<td>(rad/sec) (%)</td>
<td>(rad/sec) (%)</td>
</tr>
<tr>
<td>1 107.1658</td>
<td>107.1966</td>
<td>0.03080</td>
<td>0.02874</td>
<td></td>
</tr>
<tr>
<td>2 377.2853</td>
<td>377.4589</td>
<td>0.17360</td>
<td>0.04601</td>
<td></td>
</tr>
<tr>
<td>3 397.0859</td>
<td>397.2549</td>
<td>0.16900</td>
<td>0.04256</td>
<td></td>
</tr>
<tr>
<td>4 475.4492</td>
<td>475.7334</td>
<td>0.28420</td>
<td>0.05977</td>
<td></td>
</tr>
<tr>
<td>5 1098.4206</td>
<td>1099.2899</td>
<td>0.86930</td>
<td>0.07914</td>
<td></td>
</tr>
<tr>
<td>6 1314.7449</td>
<td>1316.2433</td>
<td>1.49840</td>
<td>0.11396</td>
<td></td>
</tr>
<tr>
<td>7 1502.0990</td>
<td>1504.0376</td>
<td>1.93860</td>
<td>0.12905</td>
<td></td>
</tr>
<tr>
<td>8 1908.0520</td>
<td>1911.6293</td>
<td>3.57730</td>
<td>0.18748</td>
<td></td>
</tr>
<tr>
<td>9 2057.2227</td>
<td>2061.4500</td>
<td>4.22730</td>
<td>0.20548</td>
<td></td>
</tr>
<tr>
<td>10 2441.9228</td>
<td>2447.5039</td>
<td>5.58110</td>
<td>0.22855</td>
<td></td>
</tr>
<tr>
<td>11 2688.9983</td>
<td>2695.0238</td>
<td>6.02550</td>
<td>0.22407</td>
<td></td>
</tr>
<tr>
<td>12 2896.4220</td>
<td>2903.7459</td>
<td>7.32390</td>
<td>0.25286</td>
<td></td>
</tr>
<tr>
<td>13 4159.4145</td>
<td>4171.0937</td>
<td>11.67920</td>
<td>0.28078</td>
<td></td>
</tr>
<tr>
<td>14 4602.2060</td>
<td>4618.2581</td>
<td>16.05210</td>
<td>0.34879</td>
<td></td>
</tr>
<tr>
<td>15 4924.9877</td>
<td>4943.6005</td>
<td>18.61280</td>
<td>0.37792</td>
<td></td>
</tr>
<tr>
<td>16 5585.9694</td>
<td>5612.5382</td>
<td>26.56880</td>
<td>0.47563</td>
<td></td>
</tr>
<tr>
<td>17 5855.2977</td>
<td>5885.1186</td>
<td>29.82090</td>
<td>0.50929</td>
<td></td>
</tr>
<tr>
<td>18 6370.0693</td>
<td>6405.0077</td>
<td>34.93840</td>
<td>0.54847</td>
<td></td>
</tr>
<tr>
<td>19 6913.5078</td>
<td>6949.5716</td>
<td>36.06380</td>
<td>0.52164</td>
<td></td>
</tr>
<tr>
<td>20 7187.3031</td>
<td>7227.1974</td>
<td>39.89430</td>
<td>0.55506</td>
<td></td>
</tr>
<tr>
<td>21 9172.4259</td>
<td>9227.6363</td>
<td>55.21040</td>
<td>0.60191</td>
<td></td>
</tr>
<tr>
<td>22 9588.1739</td>
<td>9468.1561</td>
<td>60.39420</td>
<td>0.62988</td>
<td></td>
</tr>
<tr>
<td>23 10272.7812</td>
<td>10349.2749</td>
<td>76.49370</td>
<td>0.74462</td>
<td></td>
</tr>
<tr>
<td>24 11242.8782</td>
<td>11343.0263</td>
<td>100.14810</td>
<td>0.89076</td>
<td></td>
</tr>
<tr>
<td>25 11458.0602</td>
<td>11550.2063</td>
<td>92.14610</td>
<td>0.80420</td>
<td></td>
</tr>
<tr>
<td>26 11873.7397</td>
<td>11931.5463</td>
<td>57.80660</td>
<td>0.48684</td>
<td></td>
</tr>
<tr>
<td>27 12139.6204</td>
<td>12249.1092</td>
<td>109.48880</td>
<td>0.90191</td>
<td></td>
</tr>
<tr>
<td>28 12828.1377</td>
<td>12862.2577</td>
<td>34.12000</td>
<td>0.26597</td>
<td></td>
</tr>
<tr>
<td>29 13515.4265</td>
<td>13650.0872</td>
<td>134.66070</td>
<td>0.99634</td>
<td></td>
</tr>
<tr>
<td>30 14101.0285</td>
<td>14190.6086</td>
<td>89.58010</td>
<td>0.63527</td>
<td></td>
</tr>
<tr>
<td>31 16419.5340</td>
<td>16589.2026</td>
<td>169.66860</td>
<td>1.03333</td>
<td></td>
</tr>
<tr>
<td>32 16950.8632</td>
<td>17151.7868</td>
<td>200.92360</td>
<td>1.18532</td>
<td></td>
</tr>
<tr>
<td>33 17306.4635</td>
<td>17505.5446</td>
<td>199.08110</td>
<td>1.15032</td>
<td></td>
</tr>
<tr>
<td>34 18554.3350</td>
<td>18789.5466</td>
<td>235.21160</td>
<td>1.26769</td>
<td></td>
</tr>
<tr>
<td>35 18960.2803</td>
<td>19224.3217</td>
<td>264.04140</td>
<td>1.39260</td>
<td></td>
</tr>
<tr>
<td>36 19861.8067</td>
<td>20167.4736</td>
<td>305.66690</td>
<td>1.53896</td>
<td></td>
</tr>
<tr>
<td>37 20975.5233</td>
<td>21260.2766</td>
<td>284.75330</td>
<td>1.35755</td>
<td></td>
</tr>
<tr>
<td>38 21985.1230</td>
<td>22315.8535</td>
<td>330.73050</td>
<td>1.50433</td>
<td></td>
</tr>
<tr>
<td>39 23303.4247</td>
<td>23478.5106</td>
<td>175.08590</td>
<td>0.75133</td>
<td></td>
</tr>
</tbody>
</table>
Table 8. Verdeel Truss Exact-Element Model Eigenvalues, With and Without Shear Deformation and Rotatory Inertia (continued).

<table>
<thead>
<tr>
<th>Program</th>
<th>DSTAP</th>
<th>DSTAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routine</td>
<td>BFMTHD</td>
<td>BFMTHD</td>
</tr>
<tr>
<td>G(psi)=</td>
<td>1.1500E7</td>
<td>infinite</td>
</tr>
<tr>
<td>iy(in)=</td>
<td>0.0721</td>
<td>zero</td>
</tr>
<tr>
<td>dof</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>CPU sec</td>
<td>260.4600</td>
<td>258.1900</td>
</tr>
<tr>
<td>Absolute</td>
<td>Difference</td>
<td>Relative</td>
</tr>
<tr>
<td>(rad/sec)</td>
<td>(%)</td>
<td></td>
</tr>
</tbody>
</table>

| 40 | 24899.5450 | 25300.6025 | 401.05750 | 1.61070 |
| 41 | 25205.4059 | 25374.8936 | 169.48770 | 0.67242 |
| 42 | 26284.5061 | 26753.6289 | 469.12280 | 1.78478 |
| 43 | 27717.9006 | 28142.0960 | 424.19540 | 1.53040 |
| 44 | 27992.8458 | 28562.4642 | 569.61840 | 2.03487 |
| 45 | 29192.3830 | 29826.5806 | 634.19760 | 2.17247 |
| 46 | 30082.2357 | 30637.8194 | 555.58370 | 1.84688 |
| 47 | 31532.3081 | 32204.5491 | 672.24100 | 2.13191 |
| 48 | 31750.5550 | 32432.3189 | 681.76390 | 2.14725 |
Figure 14. Mode Shapes for the Verdeel Truss Model with Shear Deformation and Rotatory Inertia.
Figure 14. Mode Shapes for the Verdeel Truss Model with Shear Deformation and Rotatory Inertia (continued).
Figure 14. Mode Shapes for the Verdeel Truss Model with Shear Deformation and Rotatory Inertia (continued).
Figure 14. Mode Shapes for the Verdeel Truss Model with Shear Deformation and Rotatory Inertia (continued).
Figure 15. Proposed 6th and 7th Modes (After Ref. [31]).
FEC models by comparing the approximate eigenvalues to the exact eigenvalues.

Several finite element models were created for DSTAP. In each successive model, more degrees of freedom were included until approximately 35 acceptable eigenvalues were extracted. The results from these models are included in Tables 9 and 10. In both of these tables the eigenvalues from Table 8 are included for comparison. Table 9 contains the eigenvalues computed with subroutine EIGENR. This routine uses the positive definite property of the finite element matrices to optimize the eigenvalue extraction. If numerical round off makes the matrices nonpositive definite then EIGENR extracts wrong eigenvalues. For example, the first model in Table 9 had 18 degrees of freedom and gave a first eigenvalue of 107.8450 rad/sec. As the number of degrees of freedom in the successive models increased, the estimate of the first eigenvalue approached the exact eigenvalue, 107.1966 rad/sec, from above. When a model of 114 degrees of freedom was used the estimate of the first eigenvalue dropped to 75.01 rad/sec. Along with this unexpected result, EIGENR sent a "non-positive definite matrix" message to the terminal screen. Thus, the model with 85 degrees of freedom was the largest model solved satisfactorily with EIGENR.

Table 10 presents the finite element model eigenvalues computed using subroutine EIGZF, from the IMSL library [60]. All the models prepared for routine EIGENR were rerun using routine EIGZF. The numerical difficulty encountered in the solution with routine EIGENR manifested itself in EIGZF as complex conjugate pair of eigenvalues. The real component of this complex pair of eigenvalues was zero. This pair of eigenvalues was easily discarded because the matrices were known to be positive-definite at the start of the eigenvalue extraction, therefore, all eigenvalues are guaranteed to be real,
Table 9. DSTAP FEM Eigenvalues using Subroutine EIGENR.

<table>
<thead>
<tr>
<th>Program</th>
<th>DSTAP</th>
<th>DSTAP</th>
<th>DSTAP</th>
<th>DSTAP</th>
<th>DSTAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routine</td>
<td>BFMTHD</td>
<td>EIGENR</td>
<td>EIGENR</td>
<td>EIGENR</td>
<td>EIGENR</td>
</tr>
<tr>
<td># of FE</td>
<td>0</td>
<td>16</td>
<td>24</td>
<td>32</td>
<td>40</td>
</tr>
<tr>
<td># of CE</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
dof    | 12    | 42    | 66    | 90    | 114   |
| CPU sec | 258.1900 | 9.6000 | 32.8000 | 87.5500 | 181.1900 |

1. 107.1966   0.26  107.3378  0.13  107.2787  0.07  75.0090 -30.0
2. 377.4589   1.00  379.7971  0.61  126.3616 -66.5
3. 397.2549   0.43  398.1451  0.22  397.8727  0.13  241.3093 -39.2
4. 475.7334   0.16  476.1578  0.08  475.9911  0.05  368.1057 -22.6
5. 1099.2899  3.07  1118.4496 1.74  1111.0391 1.06  544.1080 -50.5
6. 1316.2433  1.10  1325.2077 0.61  1321.8825 0.42  719.4389 -45.3
7. 1504.0376  7.48  1548.4612 4.70  1538.0092 2.95  107.3378  0.42  719.4389 -45.3
8. 1911.6293  4.89  1960.5662 2.55  1960.5662 2.55  1512.7147 -20.8
9. 2061.4500  2.84  2120.0610 1.10  2120.0610 1.10  1775.1528 -13.8
10. 2447.5039 2.59  2511.1180 1.29  2511.1180 1.29  1995.6209 -18.4
11. 2695.0238 1.80  2721.7413 0.99  2721.7413 0.99  2178.8817 -19.1
12. 2903.7459 1.09  2928.6360 0.85  2928.6360 0.85  2287.0080 -21.2
13. 4171.0937 11.04 4331.7611 3.85  4294.4801 2.95  2595.0512 -37.7
14. 4618.2581 10.34 4676.9491 1.27  4676.9491 1.27  3435.0934 -25.6
15. 4943.6005 17.01 5363.6059 8.49  5363.6059 8.49  3623.2252 -26.7
16. 5612.5382 19.89 6015.3827 7.17  6015.3827 7.17  4572.7918 -18.5
17. 5885.1186 17.50 6317.9385 7.35  6317.9385 7.35  5386.8732 -8.46
18. 6405.0077 21.61 6938.3559 8.32  6938.3559 8.32  5738.4236 -10.4
19. 6949.5716 26.33 7120.1827 2.45  7120.1827 2.45  6155.0409 -11.4
20. 7227.1974 28.13 7365.4795 1.91  7365.4795 1.91  6466.8575 -10.5
21. 9227.6363 20.00 9563.6520 3.64  9563.6520 3.64  6550.2610 -28.9
22. 9648.5681 22.75 9888.7795 2.48  9888.7795 2.48  7035.5089 -27.0
23. 10349.2749 21.91 11204.5037 8.26 11204.5037 8.26  9067.5834 -12.3
24. 11343.0263 17.66 11869.7849 4.64 11869.7849 4.64 10182.5401 -10.2
25. 11550.2063 29.48 11975.8197 3.68 11975.8197 3.68 10673.3853 -7.59
26. 11931.5463 44.62 12180.5677 2.08 12180.5677 2.08 11672.7721 -2.16
27. 12249.1092 54.08 12770.5564 4.25 12770.5564 4.25 12291.9971 0.34
28. 12862.2577 16.99 12996.4916 1.04 12996.4916 1.04 12389.7607 -3.67
29. 13650.0872 13.19 13894.0462 1.78 13894.0462 1.78 12636.9612 -7.42
30. 14190.6086 12.38 14487.2679 2.09 14487.2679 2.09 12858.0201 -9.39
31. 15659.2026 6.03 15831.0488 11.70 15831.0488 11.70 13291.4433 -19.8
32. 17151.7868 15.31 18781.6279 9.50 18781.6279 9.50 13512.4330 -21.2
33. 17505.5446 14.46 20938.3109 19.60 20938.3109 19.60 15428.5609 -11.8
34. 18789.5466 18.90 21154.9665 12.58 21154.9665 12.58 16602.2458 -14.8
35. 19224.3217 256.9 22891.7210 16.27 22891.7210 16.27 19214.9138 13.99
36. 20167.4736 21.33 23554.1929 16.79 23554.1929 16.79 17307.3786 -14.1
Table 10. DSTAP FEM Eigenvalues using Subroutine EIGZF.

<table>
<thead>
<tr>
<th>Program</th>
<th>DSTAP</th>
<th>DSTAP</th>
<th>DSTAP</th>
<th>DSTAP</th>
<th>DSTAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routine</td>
<td>BFMTHD</td>
<td>IMSL-EIGZF</td>
<td>IMSL-EIGZF</td>
<td>EIGENR</td>
<td>IMSL-EIGZF</td>
</tr>
<tr>
<td># of FE</td>
<td>0</td>
<td>8</td>
<td>16</td>
<td>24</td>
<td>32</td>
</tr>
<tr>
<td># of CE</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>dof</td>
<td>12</td>
<td>18</td>
<td>42</td>
<td>66</td>
<td>90</td>
</tr>
<tr>
<td>CPU sec</td>
<td>258.1900</td>
<td>2.1100</td>
<td>14.4700</td>
<td>50.1300</td>
<td>123.1400</td>
</tr>
</tbody>
</table>

1 107.1966 107.8450 0.60 107.4856 0.26 107.3378 0.13 107.2787 0.07
2 377.4589 383.5772 1.62 384.9831 1.99 382.1451 0.22 379.7827 0.13
3 397.2549 403.9738 1.69 398.9907 0.43 398.1451 0.22 397.7827 0.13
4 475.7334 481.8348 1.28 476.5404 0.16 476.1578 0.08 475.9911 0.05
5 1099.2899 1216.4625 10.65 1133.0545 3.07 1118.4496 1.74 1111.0391 1.06
6 1316.2433 1572.5912 19.47 1330.7447 1.10 1325.2077 0.68 1321.8825 0.42
7 1504.0376 1881.1082 25.07 1616.6789 7.48 1574.8122 4.70 1548.4618 2.95
9 2061.4500 3206.8254 55.56 2120.0610 2.84 2144.6072 4.03 2123.2459 2.99
10 2447.5039 4568.6860 86.66 2511.1180 2.59 2581.4980 5.47 2550.3930 4.20
11 2695.0238 11825.9291 338.8 2743.6629 1.80 2730.1718 1.30 2721.7413 0.99
12 2903.7459 11959.7983 311.8 2935.6575 1.09 2928.6360 0.85 2921.3386 0.60
13 475.7334 481.8348 1.28 476.5404 0.16 476.1578 0.08 475.9911 0.05
14 5612.5382 35809.8707 538.0 6729.2361 19.9 5991.6522 6.75 6156.3127 7.17
15 5885.1186 41031.4551 597.2 6185.3330 17.5 6148.9476 4.48 6317.9385 7.35
16 6405.0077 59373.5434 826.9 6915.3330 21.6 6868.0172 4.41 6938.3559 8.32
17 6949.5716 7889.5899 21.61 7132.5681 2.63 7120.1827 2.45
18 7227.1974 9260.6072 28.13 7403.3654 2.43 7365.4795 1.91
19 9227.6363 11073.9428 20.00 10249.4232 11.07 9563.6520 3.64
20 9648.5681 11843.7890 22.75 10628.5455 10.15 9888.7795 2.48
21 10349.2749 12616.8334 21.91 12107.8912 16.99 11204.5037 8.26
22 11343.0283 13346.3152 17.66 12260.3361 8.08 11869.7849 4.64
23 11550.2063 14956.1695 29.48 12982.2507 12.39 11975.8197 3.68
24 11931.5463 17256.4841 44.62 13152.4545 10.23 12180.5677 2.08
25 12249.1092 18874.2703 54.08 13575.9913 10.83 12770.5564 4.25
26 12862.2577 20029.7633 55.72 14309.3630 11.25 12946.4916 1.04
27 13650.0872 23051.6905 68.87 15450.8103 13.19 13894.0642 1.78
28 14190.6086 29760.7079 109.7 15948.4922 12.38 14487.2679 2.09
29 16589.2026 37986.9133 128.9 17590.0374 6.03 18531.0488 11.70
30 17151.7868 38698.5476 125.6 19778.6478 15.31 18781.6279 9.50
31 17505.9446 44214.9083 152.5 20037.5294 14.46 20938.3109 19.60
32 18789.5466 59487.6167 216.5 22342.6451 18.90 21154.9665 12.58
33 19224.3217 68826.6892 256.9 22891.7210 19.07 21914.9138 13.99
Table 10. DSTAP FEM Eigenvalues using Subroutine EIGZF (continued).

<table>
<thead>
<tr>
<th>Program</th>
<th>DSTAP</th>
<th>DSTAP</th>
<th>DSTAP</th>
<th>DSTAP</th>
<th>DSTAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routine</td>
<td>BFMTHD</td>
<td>IMSL—EIGZF</td>
<td>IMSL—EIGZF</td>
<td>IMSL—EIGZF</td>
<td>IMSL—EIGZF</td>
</tr>
<tr>
<td># of FE</td>
<td>0</td>
<td>40</td>
<td>Rel. 48</td>
<td>56</td>
<td>Rel. 64</td>
</tr>
<tr>
<td># of CE</td>
<td>6</td>
<td>0</td>
<td>Diff 0</td>
<td>Diff 0</td>
<td>Diff 0</td>
</tr>
<tr>
<td>dof</td>
<td>12</td>
<td>114</td>
<td>(%)</td>
<td>138</td>
<td>(%)</td>
</tr>
<tr>
<td>CPU sec</td>
<td>258.1900</td>
<td>246.6900</td>
<td>417.1400</td>
<td>673.5600</td>
<td>991.1400</td>
</tr>
</tbody>
</table>

1. 107.1966 107.2499 0.04 107.2339 0.03 107.2241 0.02 107.2177 0.01
2. 377.4589 378.9914 0.40 378.5368 0.28 378.2562 0.21 378.0728 0.16
3. 397.2549 397.6011 0.08 397.4986 0.06 397.4354 0.04 397.3938 0.03
4. 475.7334 475.9046 0.03 475.8549 0.02 475.8238 0.01 475.8032 0.01
5. 1099.2899 1107.1220 0.71 1104.8520 0.50 1103.4322 0.37 1102.4896 0.29
6. 1316.2433 1320.0532 0.28 1318.9703 0.20 1318.2843 0.15 1317.8251 0.12
7. 1504.0376 1533.8831 1.98 1525.3036 1.41 1519.9021 1.05 1516.3036 0.81
8. 1911.6293 1945.2593 1.75 1935.8621 1.26 1929.8231 0.95 1925.7523 0.73
9. 2061.4500 2105.8834 2.15 2094.2203 1.58 2086.3904 1.20 2080.9797 0.94
10. 2447.5039 2522.0126 3.04 2502.5530 2.24 2489.4150 1.71 2480.3184 1.34
11. 2695.0238 2714.5940 0.72 2709.6173 0.54 2706.2075 0.41 2703.8210 0.32
12. 2903.7459 2916.3195 0.43 2913.0348 0.31 2910.8352 0.24 2909.3110 0.19
13. 4171.0937 4262.8783 2.20 4240.7399 1.66 4225.2360 1.29 4214.1522 1.03
14. 4618.2581 4660.9758 0.92 4650.3282 0.69 4643.0460 0.53 4637.9403 0.42
15. 4943.6005 5260.3633 6.40 5182.2843 4.82 5127.3687 3.71 5088.4923 2.93
16. 5612.5382 5923.5402 5.54 5847.7285 4.19 5793.4150 3.22 5754.8511 2.53
17. 5885.1186 6254.5292 5.27 6180.7491 5.02 6120.7806 4.00 6075.0129 3.22
18. 6405.0077 6879.5840 7.40 6788.3823 5.98 6710.9041 4.77 6651.2993 3.84
19. 6949.5716 7085.7846 1.96 7054.8500 1.50 7031.1003 1.17 7014.6141 0.92
20. 7227.1974 7327.7611 1.39 7301.7314 1.03 7284.1264 0.78 7271.9081 0.61
21. 9227.6363 9511.6938 3.07 9460.2752 2.52 9420.0138 2.08 9388.3639 1.74
22. 9648.5681 9845.0749 2.03 9798.5752 1.55 9765.8503 1.21 9741.9966 0.96
23. 10349.2749 11375.2562 9.91 11229.2676 8.50 11072.2645 6.98 10941.6468 5.72
24. 11343.0263 11884.3961 4.77 11840.3856 4.38 11811.6767 4.13 11790.6518 3.94
25. 11550.2063 12325.8188 6.71 12205.2446 5.67 12068.4152 4.31 11915.4551 3.16
26. 11931.5463 12744.4040 6.81 12734.2100 6.72 12606.9449 5.66 12475.6058 4.55
27. 12249.1092 12887.2144 5.20 12835.7679 4.78 12806.9739 4.55 12782.5319 4.35
28. 12862.2577 13508.9993 5.02 13499.7098 4.95 13361.4756 3.88 13219.6200 2.77
29. 13650.0872 13991.7059 2.50 13296.7789 2.58 13865.9530 1.58 13820.2671 1.24
30. 14190.6086 14476.3084 2.01 14425.3127 1.65 14382.0484 1.34 14384.7028 1.11
31. 16859.2026 17299.0867 4.27 17314.2871 4.37 17236.5947 3.90 17162.2227 3.45
32. 17151.7882 17557.0282 2.36 17554.8012 2.34 17475.5471 1.88 17411.0941 1.51
33. 17505.5446 18775.8067 7.25 19230.1172 9.85 19087.8445 9.03 18875.5749 7.82
34. 18789.5466 19535.7594 3.97 20483.0307 9.01 20640.6644 8.89 20286.0626 7.96
35. 19224.3217 19829.5485 3.14 20810.5726 8.25 20792.2777 8.15 20593.1358 7.12
Table 10. DSTAP FEM Eigenvalues using Subroutine EIGZF (continued).

<table>
<thead>
<tr>
<th>Program</th>
<th>DSTAP</th>
<th>DSTAP</th>
<th>DSTAP</th>
<th>DSTAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routine</td>
<td>BFMTHD</td>
<td>IMSL-EIGZF</td>
<td>IMSL-EIGZF</td>
<td>IMSL-EIGZF</td>
</tr>
<tr>
<td># of FE</td>
<td>0</td>
<td>72</td>
<td>Rel. 80</td>
<td>Rel. 120</td>
</tr>
<tr>
<td># of CE</td>
<td>6</td>
<td>0</td>
<td>Diff</td>
<td>0</td>
</tr>
<tr>
<td>dof</td>
<td>12</td>
<td>210 (%)</td>
<td>234 (%)</td>
<td>354 (%)</td>
</tr>
<tr>
<td>CPU sec</td>
<td>258.1900</td>
<td>1404.4500</td>
<td>1921.0700</td>
<td>7150.2700</td>
</tr>
<tr>
<td>1</td>
<td>107.1966</td>
<td>107.2133</td>
<td>0.01</td>
<td>107.2102</td>
</tr>
<tr>
<td>2</td>
<td>377.4589</td>
<td>377.9456</td>
<td>0.12</td>
<td>377.8541</td>
</tr>
<tr>
<td>3</td>
<td>397.2549</td>
<td>397.3651</td>
<td>0.02</td>
<td>397.3444</td>
</tr>
<tr>
<td>4</td>
<td>475.7334</td>
<td>475.7889</td>
<td>0.01</td>
<td>475.7785</td>
</tr>
<tr>
<td>5</td>
<td>1099.2899</td>
<td>1101.3592</td>
<td>0.23</td>
<td>1101.3592</td>
</tr>
<tr>
<td>6</td>
<td>1316.2433</td>
<td>1317.2704</td>
<td>0.09</td>
<td>1317.2704</td>
</tr>
<tr>
<td>7</td>
<td>2061.4500</td>
<td>2074.2766</td>
<td>0.75</td>
<td>2074.2766</td>
</tr>
<tr>
<td>8</td>
<td>2447.5039</td>
<td>2469.0504</td>
<td>1.07</td>
<td>2469.0504</td>
</tr>
<tr>
<td>9</td>
<td>2695.0238</td>
<td>2700.3842</td>
<td>0.26</td>
<td>2700.3842</td>
</tr>
<tr>
<td>10</td>
<td>2903.7459</td>
<td>2907.4515</td>
<td>0.15</td>
<td>2907.4515</td>
</tr>
<tr>
<td>11</td>
<td>4171.0937</td>
<td>4199.9529</td>
<td>0.69</td>
<td>4199.9529</td>
</tr>
<tr>
<td>12</td>
<td>4618.2581</td>
<td>4631.4265</td>
<td>0.28</td>
<td>4631.4265</td>
</tr>
<tr>
<td>13</td>
<td>5494.6005</td>
<td>5060.3578</td>
<td>2.36</td>
<td>5060.3578</td>
</tr>
<tr>
<td>14</td>
<td>5612.5382</td>
<td>5726.9767</td>
<td>2.03</td>
<td>5726.9767</td>
</tr>
<tr>
<td>15</td>
<td>5885.1186</td>
<td>6013.9269</td>
<td>2.18</td>
<td>6013.9269</td>
</tr>
<tr>
<td>16</td>
<td>6405.0077</td>
<td>6571.5050</td>
<td>2.59</td>
<td>6571.5050</td>
</tr>
<tr>
<td>17</td>
<td>6994.5716</td>
<td>7001.7803</td>
<td>0.75</td>
<td>7001.7803</td>
</tr>
<tr>
<td>18</td>
<td>7227.1974</td>
<td>7256.7089</td>
<td>0.40</td>
<td>7256.7089</td>
</tr>
<tr>
<td>19</td>
<td>9227.6363</td>
<td>9343.0929</td>
<td>1.25</td>
<td>9343.0929</td>
</tr>
<tr>
<td>20</td>
<td>9648.5681</td>
<td>9724.6783</td>
<td>1.67</td>
<td>9724.6783</td>
</tr>
<tr>
<td>21</td>
<td>10349.2749</td>
<td>10757.9269</td>
<td>3.94</td>
<td>10757.9269</td>
</tr>
<tr>
<td>22</td>
<td>11343.0263</td>
<td>11730.4435</td>
<td>3.41</td>
<td>11730.4435</td>
</tr>
<tr>
<td>23</td>
<td>11550.2063</td>
<td>11757.2097</td>
<td>1.79</td>
<td>11757.2097</td>
</tr>
<tr>
<td>24</td>
<td>11931.5463</td>
<td>12278.2165</td>
<td>2.90</td>
<td>12278.2165</td>
</tr>
<tr>
<td>25</td>
<td>12249.1092</td>
<td>12716.4597</td>
<td>3.81</td>
<td>12716.4597</td>
</tr>
<tr>
<td>26</td>
<td>12862.2577</td>
<td>13026.3630</td>
<td>1.27</td>
<td>13026.3630</td>
</tr>
<tr>
<td>27</td>
<td>13650.0872</td>
<td>13761.9863</td>
<td>0.81</td>
<td>13761.9863</td>
</tr>
<tr>
<td>28</td>
<td>14190.6086</td>
<td>14302.9126</td>
<td>0.79</td>
<td>14302.9126</td>
</tr>
<tr>
<td>29</td>
<td>16589.2026</td>
<td>17095.9577</td>
<td>3.05</td>
<td>17095.9577</td>
</tr>
<tr>
<td>30</td>
<td>17151.7868</td>
<td>17326.4639</td>
<td>1.01</td>
<td>17326.4639</td>
</tr>
<tr>
<td>31</td>
<td>17505.5446</td>
<td>18507.3169</td>
<td>5.72</td>
<td>18507.3169</td>
</tr>
<tr>
<td>32</td>
<td>18789.5466</td>
<td>20097.1826</td>
<td>6.04</td>
<td>20097.1826</td>
</tr>
<tr>
<td>33</td>
<td>19224.3217</td>
<td>20199.1740</td>
<td>5.07</td>
<td>20199.1740</td>
</tr>
</tbody>
</table>
positive, and nonzero. Eliminating these eigenvalues from the solution set, the eigenvalues computed with EIGZF exhibit the expected behavior. The higher eigenvalues can be extracted satisfactorily, provided the model has a sufficient number of degrees of freedom.

Figure 16 presents graphically the results from Table 10. The data points in this figure were used in a regression analysis. The analysis gave the curve fit,

$$y = (0.13095)x,$$  \hspace{1cm} (32)

where $x$ is again the number of degrees of freedom and $y$ is the number of acceptable eigenvalues. The number of necessary active nodes per acceptable eigenvalue may be estimated by using Eq. (32) in reverse fashion. Approximately 8 degrees of freedom are required for each acceptable eigenvalue, which corresponds to approximately 3 active nodes per acceptable eigenvalue. Figure 16 also contains two more curves, one for the results obtained with FEM models in SUPERB and another one for combined models in DSTAP. These curves are discussed below.

Some of the smaller models used with EIGENR and EIGZF were analyzed with the SUPERB finite element program [15]. The results from SUPERB are presented in Table 11 and shown graphically in Fig. 16. These results show that the FEM-beam element in SUPERB is better suited for extracting higher eigenvalues than the FEM-beam element added to DSTAP. The regression line for the SUPERB models is of the form,

$$y = -3.293 + (0.37024)x$$

The higher slope of this regression line indicates that SUPERB requires fewer degrees of freedom to extract the desired number of acceptable eigenvalues. This result is quite surprising in light of the fixed-fixed example, where the
Figure 16. Number of Acceptable Eigenvalues versus Degrees of Freedom for the Verdeel Truss.
Table 11. SUPERB FEM Eigenvalues.

<table>
<thead>
<tr>
<th>Program</th>
<th>DSTAP</th>
<th>SUPERB</th>
<th>SUPERB</th>
<th>SUPERB</th>
<th>SUPERB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routine</td>
<td>BFMTHD</td>
<td>JACOBI</td>
<td>JACOBI</td>
<td>JACOBI</td>
<td>JACOBI</td>
</tr>
<tr>
<td># of FE</td>
<td>0</td>
<td>16 Rel.</td>
<td>32 Rel.</td>
<td>40 Rel.</td>
<td>48 Rel.</td>
</tr>
<tr>
<td># of CE</td>
<td>6</td>
<td>0 Diff</td>
<td>0 Diff</td>
<td>0 Diff</td>
<td>0 Diff</td>
</tr>
<tr>
<td>dof</td>
<td>12</td>
<td>42 (%)</td>
<td>90 (%)</td>
<td>114 (%)</td>
<td>138 (%)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CPU sec</th>
<th>258.1900</th>
<th>48.2700</th>
<th>351.7500</th>
<th>700.7000</th>
<th>1347.9000</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>107.1966</td>
<td>107.1974</td>
<td>0.00</td>
<td>107.1974</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>377.4589</td>
<td>377.5189</td>
<td>0.01</td>
<td>377.4497</td>
<td>-0.00</td>
</tr>
<tr>
<td>3</td>
<td>397.2549</td>
<td>397.4931</td>
<td>0.05</td>
<td>397.2543</td>
<td>-0.00</td>
</tr>
<tr>
<td>4</td>
<td>475.7334</td>
<td>476.1523</td>
<td>0.08</td>
<td>475.7376</td>
<td>0.00</td>
</tr>
<tr>
<td>5</td>
<td>1099.2899</td>
<td>1104.0813</td>
<td>0.43</td>
<td>1099.2432</td>
<td>-0.00</td>
</tr>
<tr>
<td>6</td>
<td>1316.4233</td>
<td>1324.8639</td>
<td>0.64</td>
<td>1316.2543</td>
<td>-0.00</td>
</tr>
<tr>
<td>7</td>
<td>1504.0376</td>
<td>1516.6352</td>
<td>0.83</td>
<td>1504.5715</td>
<td>0.00</td>
</tr>
<tr>
<td>8</td>
<td>1911.6293</td>
<td>1934.6555</td>
<td>1.20</td>
<td>1911.8476</td>
<td>0.01</td>
</tr>
<tr>
<td>9</td>
<td>2061.4500</td>
<td>2091.2954</td>
<td>1.44</td>
<td>2061.8900</td>
<td>0.00</td>
</tr>
<tr>
<td>10</td>
<td>2447.5039</td>
<td>2480.1617</td>
<td>1.33</td>
<td>2448.0546</td>
<td>0.00</td>
</tr>
<tr>
<td>11</td>
<td>2695.0523</td>
<td>2729.9776</td>
<td>2.15</td>
<td>2698.9891</td>
<td>0.03</td>
</tr>
<tr>
<td>12</td>
<td>2903.7459</td>
<td>2965.6634</td>
<td>2.13</td>
<td>2908.7378</td>
<td>0.17</td>
</tr>
<tr>
<td>13</td>
<td>4171.0937</td>
<td>4642.5827</td>
<td>11.30</td>
<td>4175.7421</td>
<td>0.11</td>
</tr>
<tr>
<td>14</td>
<td>4618.2581</td>
<td>5247.2765</td>
<td>13.62</td>
<td>4625.1155</td>
<td>0.14</td>
</tr>
<tr>
<td>15</td>
<td>4943.6005</td>
<td>5627.7234</td>
<td>13.83</td>
<td>4952.2810</td>
<td>0.17</td>
</tr>
<tr>
<td>16</td>
<td>5612.5382</td>
<td>6788.3534</td>
<td>20.94</td>
<td>5624.7074</td>
<td>0.21</td>
</tr>
<tr>
<td>17</td>
<td>5885.1186</td>
<td>7168.4681</td>
<td>21.60</td>
<td>5901.2304</td>
<td>0.27</td>
</tr>
<tr>
<td>18</td>
<td>6405.0077</td>
<td>8145.5214</td>
<td>27.17</td>
<td>6423.3003</td>
<td>0.28</td>
</tr>
<tr>
<td>19</td>
<td>6949.5716</td>
<td>8753.1054</td>
<td>25.95</td>
<td>6976.2206</td>
<td>0.38</td>
</tr>
<tr>
<td>20</td>
<td>7227.1974</td>
<td>9206.7514</td>
<td>27.39</td>
<td>7260.8489</td>
<td>0.46</td>
</tr>
<tr>
<td>21</td>
<td>9227.6363</td>
<td>11538.4415</td>
<td>25.04</td>
<td>9291.5748</td>
<td>0.69</td>
</tr>
<tr>
<td>22</td>
<td>9648.5071</td>
<td>12161.1051</td>
<td>26.04</td>
<td>9700.8221</td>
<td>0.95</td>
</tr>
<tr>
<td>23</td>
<td>10349.2749</td>
<td>12612.2378</td>
<td>21.86</td>
<td>10438.8840</td>
<td>0.86</td>
</tr>
<tr>
<td>24</td>
<td>11343.0263</td>
<td>14437.5032</td>
<td>27.28</td>
<td>11584.3087</td>
<td>2.12</td>
</tr>
<tr>
<td>25</td>
<td>11550.2063</td>
<td>14716.4766</td>
<td>27.41</td>
<td>11801.0786</td>
<td>2.17</td>
</tr>
<tr>
<td>26</td>
<td>11931.5463</td>
<td>17780.1577</td>
<td>49.01</td>
<td>12110.8396</td>
<td>1.50</td>
</tr>
<tr>
<td>27</td>
<td>12249.1092</td>
<td>18023.9453</td>
<td>47.14</td>
<td>12452.0166</td>
<td>1.65</td>
</tr>
<tr>
<td>28</td>
<td>12862.2577</td>
<td>19940.3168</td>
<td>55.02</td>
<td>13011.8484</td>
<td>1.16</td>
</tr>
<tr>
<td>29</td>
<td>13650.0872</td>
<td>21357.8035</td>
<td>56.46</td>
<td>13949.2997</td>
<td>2.19</td>
</tr>
<tr>
<td>30</td>
<td>14190.6086</td>
<td>25166.5704</td>
<td>77.34</td>
<td>14384.7244</td>
<td>1.36</td>
</tr>
<tr>
<td>31</td>
<td>16559.2026</td>
<td>36228.1560</td>
<td>130.40</td>
<td>18291.6090</td>
<td>10.26</td>
</tr>
<tr>
<td>32</td>
<td>17151.7868</td>
<td>38571.8462</td>
<td>124.8</td>
<td>19043.0780</td>
<td>11.02</td>
</tr>
<tr>
<td>33</td>
<td>17505.5446</td>
<td>44082.8281</td>
<td>151.8</td>
<td>19231.5735</td>
<td>9.85</td>
</tr>
<tr>
<td>34</td>
<td>18789.5466</td>
<td>52500.4114</td>
<td>179.4</td>
<td>20843.8389</td>
<td>10.93</td>
</tr>
<tr>
<td>35</td>
<td>19224.3217</td>
<td>68926.5428</td>
<td>258.5</td>
<td>21649.3433</td>
<td>12.61</td>
</tr>
</tbody>
</table>

|         | 19666.3700 | 2.29     | 19510.5470 | 1.48     |
greatest deviation between DSTAP and SUPERB model eigenvalues was about one percent. Without appropriate documentation, it is difficult to determine why the results from DSTAP and SUPERB agreed closely in the small models of the fixed-fixed example but deviate in the larger models of the Verdeel Truss.

The third curve in Fig. 16 corresponds to the combined models run in DSTAP. These models used exact-beam elements for the legs of the Verdeel Truss and FEM-beam elements for the horizontal members. The number of degrees of freedom in these models was increased by adding more FEM-beam elements. The regression line for these results is,

\[ y = -6.8 + (0.66667)x \]

This line is steeper than either of the other two regression lines. Thus, it takes even fewer degrees of freedom to extract acceptable higher eigenvalues with combined models than with DSTAP or SUPERB finite element models. This result is promising since it was found above that the SUPERB FEM-beam formulation gave better results than the DSTAP FEM-beam formulation which was used in these combined models. Thus, even with the poorer FEM-beam element formulation employed in the FEC models, the exact-beam elements were able to improve the models sufficiently to give better model results than the SUPERB models. The tabulated results of the FEC models are presented in Table 12.

The solution times for the different techniques are compared in the same manner as in the previous example. That is, the total solution time for a model is divided by the number of acceptable eigenvalues obtained with that model. These ratios are presented in graphically in Fig. 17. The curves in this figure reflect the same trends as were seen in the Portal Arch example. The higher eigenvalues are extracted more economically with either a combined model or an exact model. For example, when there are more than 65 degrees of
Table 12. DSTAP Combined Model (FEC) Eigenvalues.

<table>
<thead>
<tr>
<th>Program</th>
<th>DSTAP</th>
<th>DSTAP</th>
<th>DSTAP</th>
<th>DSTAP</th>
<th>DSTAP</th>
<th>DSTAP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BFMTHD</td>
<td>BFMTHD</td>
<td>BFMTHD</td>
<td>BFMTHD</td>
<td>BFMTHD</td>
<td>BFMTHD</td>
</tr>
<tr>
<td># of FE</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td># of CE</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>dof</td>
<td>12</td>
<td>12 (%)</td>
<td>18 (%)</td>
<td>24 (%)</td>
<td>30 (%)</td>
<td></td>
</tr>
<tr>
<td>CPU sec</td>
<td>258.1900</td>
<td>227.5500</td>
<td>369.1000</td>
<td>664.6500</td>
<td>1058.4700</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>107.1966</td>
<td>107.1978</td>
<td>0.00</td>
<td>107.1966</td>
<td>0.00</td>
<td>107.1966</td>
</tr>
<tr>
<td>2</td>
<td>377.4589</td>
<td>377.6729</td>
<td>0.05</td>
<td>377.5641</td>
<td>0.28</td>
<td>377.4647</td>
</tr>
<tr>
<td>3</td>
<td>475.7334</td>
<td>830.6440</td>
<td>74.60</td>
<td>481.6326</td>
<td>1.24</td>
<td>477.0909</td>
</tr>
<tr>
<td>4</td>
<td>1099.2899</td>
<td>1273.7065</td>
<td>15.86</td>
<td>1189.7511</td>
<td>8.22</td>
<td>1109.9622</td>
</tr>
<tr>
<td>5</td>
<td>1316.2433</td>
<td>1931.4248</td>
<td>46.73</td>
<td>2081.3655</td>
<td>58.12</td>
<td>1337.3768</td>
</tr>
<tr>
<td>6</td>
<td>1504.0376</td>
<td>2291.7081</td>
<td>52.37</td>
<td>2193.4467</td>
<td>45.83</td>
<td>1512.5566</td>
</tr>
<tr>
<td>7</td>
<td>1911.6293</td>
<td>2451.3042</td>
<td>28.23</td>
<td>2520.3462</td>
<td>31.84</td>
<td>1923.2429</td>
</tr>
<tr>
<td>8</td>
<td>2061.4500</td>
<td>2848.6489</td>
<td>38.18</td>
<td>3678.1027</td>
<td>78.42</td>
<td>2078.0802</td>
</tr>
<tr>
<td>9</td>
<td>2447.5039</td>
<td>3597.7735</td>
<td>46.99</td>
<td>4675.5190</td>
<td>91.03</td>
<td>2450.8987</td>
</tr>
<tr>
<td>10</td>
<td>2695.0238</td>
<td>6197.7142</td>
<td>129.9</td>
<td>4893.6162</td>
<td>81.57</td>
<td>3044.0843</td>
</tr>
<tr>
<td>11</td>
<td>4171.0937</td>
<td>6610.4310</td>
<td>58.48</td>
<td>4527.8708</td>
<td>6.59</td>
<td>4450.9114</td>
</tr>
<tr>
<td>12</td>
<td>4618.2581</td>
<td>6872.8248</td>
<td>48.81</td>
<td>8758.7492</td>
<td>89.65</td>
<td>5117.7473</td>
</tr>
<tr>
<td>13</td>
<td>4943.6005</td>
<td>11762.6313</td>
<td>137.9</td>
<td>5634.7126</td>
<td>13.97</td>
<td>5186.5908</td>
</tr>
<tr>
<td>14</td>
<td>5612.5382</td>
<td>12045.9483</td>
<td>116.7</td>
<td>5897.1256</td>
<td>103.0</td>
<td>5958.1750</td>
</tr>
<tr>
<td>15</td>
<td>6405.0077</td>
<td>12976.6223</td>
<td>79.55</td>
<td>10247.7793</td>
<td>41.79</td>
<td>5917.2160</td>
</tr>
<tr>
<td>16</td>
<td>6949.5716</td>
<td>12286.7726</td>
<td>118.9</td>
<td>6775.9252</td>
<td>15.13</td>
<td>6046.6586</td>
</tr>
<tr>
<td>17</td>
<td>7227.1974</td>
<td>12286.7726</td>
<td>118.9</td>
<td>6775.9252</td>
<td>15.13</td>
<td>6046.6586</td>
</tr>
<tr>
<td>18</td>
<td>7293.7459</td>
<td>6293.0632</td>
<td>116.7</td>
<td>5897.1256</td>
<td>103.0</td>
<td>5958.1750</td>
</tr>
<tr>
<td>19</td>
<td>8171.0937</td>
<td>6610.4310</td>
<td>58.48</td>
<td>4527.8708</td>
<td>6.59</td>
<td>4450.9114</td>
</tr>
<tr>
<td>20</td>
<td>8618.2581</td>
<td>6872.8248</td>
<td>48.81</td>
<td>8758.7492</td>
<td>89.65</td>
<td>5117.7473</td>
</tr>
<tr>
<td>21</td>
<td>9493.6005</td>
<td>11762.6313</td>
<td>137.9</td>
<td>5634.7126</td>
<td>13.97</td>
<td>5186.5908</td>
</tr>
<tr>
<td>22</td>
<td>5612.5382</td>
<td>12045.9483</td>
<td>116.7</td>
<td>5897.1256</td>
<td>103.0</td>
<td>5958.1750</td>
</tr>
<tr>
<td>23</td>
<td>6405.0077</td>
<td>12976.6223</td>
<td>79.55</td>
<td>10247.7793</td>
<td>41.79</td>
<td>5917.2160</td>
</tr>
<tr>
<td>24</td>
<td>6949.5716</td>
<td>12286.7726</td>
<td>118.9</td>
<td>6775.9252</td>
<td>15.13</td>
<td>6046.6586</td>
</tr>
<tr>
<td>25</td>
<td>7227.1974</td>
<td>12286.7726</td>
<td>118.9</td>
<td>6775.9252</td>
<td>15.13</td>
<td>6046.6586</td>
</tr>
<tr>
<td>26</td>
<td>7293.7459</td>
<td>6293.0632</td>
<td>116.7</td>
<td>5897.1256</td>
<td>103.0</td>
<td>5958.1750</td>
</tr>
<tr>
<td>27</td>
<td>8171.0937</td>
<td>6610.4310</td>
<td>58.48</td>
<td>4527.8708</td>
<td>6.59</td>
<td>4450.9114</td>
</tr>
<tr>
<td>28</td>
<td>8618.2581</td>
<td>6872.8248</td>
<td>48.81</td>
<td>8758.7492</td>
<td>89.65</td>
<td>5117.7473</td>
</tr>
<tr>
<td>29</td>
<td>9493.6005</td>
<td>11762.6313</td>
<td>137.9</td>
<td>5634.7126</td>
<td>13.97</td>
<td>5186.5908</td>
</tr>
<tr>
<td>30</td>
<td>5612.5382</td>
<td>12045.9483</td>
<td>116.7</td>
<td>5897.1256</td>
<td>103.0</td>
<td>5958.1750</td>
</tr>
<tr>
<td>31</td>
<td>6405.0077</td>
<td>12976.6223</td>
<td>79.55</td>
<td>10247.7793</td>
<td>41.79</td>
<td>5917.2160</td>
</tr>
<tr>
<td>32</td>
<td>6949.5716</td>
<td>12286.7726</td>
<td>118.9</td>
<td>6775.9252</td>
<td>15.13</td>
<td>6046.6586</td>
</tr>
<tr>
<td>33</td>
<td>7227.1974</td>
<td>12286.7726</td>
<td>118.9</td>
<td>6775.9252</td>
<td>15.13</td>
<td>6046.6586</td>
</tr>
<tr>
<td>34</td>
<td>7293.7459</td>
<td>6293.0632</td>
<td>116.7</td>
<td>5897.1256</td>
<td>103.0</td>
<td>5958.1750</td>
</tr>
<tr>
<td>35</td>
<td>8171.0937</td>
<td>6610.4310</td>
<td>58.48</td>
<td>4527.8708</td>
<td>6.59</td>
<td>4450.9114</td>
</tr>
</tbody>
</table>
Table 12. DSTAP Combined Model (FEC) Eigenvalues (continued).

<table>
<thead>
<tr>
<th>Program</th>
<th>DSTAP</th>
<th>DSTAP</th>
<th>DSTAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routine</td>
<td>BFMTHD</td>
<td>BFMTHD</td>
<td>BFMTHD</td>
</tr>
<tr>
<td># of FE</td>
<td>0</td>
<td>10</td>
<td>Rel.</td>
</tr>
<tr>
<td># of CE</td>
<td>6</td>
<td>4</td>
<td>Diff</td>
</tr>
<tr>
<td>dof</td>
<td>12</td>
<td>36</td>
<td>(%)</td>
</tr>
<tr>
<td>CPU sec</td>
<td>258.1900</td>
<td>1729.6800</td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>107.1966</td>
<td>107.1966</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>377.4589</td>
<td>377.1646</td>
<td>-0.07</td>
</tr>
<tr>
<td>3</td>
<td>397.2549</td>
<td>397.3580</td>
<td>0.02</td>
</tr>
<tr>
<td>4</td>
<td>475.7334</td>
<td>475.9149</td>
<td>0.03</td>
</tr>
<tr>
<td>5</td>
<td>1099.2899</td>
<td>1100.8606</td>
<td>0.14</td>
</tr>
<tr>
<td>6</td>
<td>1316.2433</td>
<td>1319.6876</td>
<td>0.26</td>
</tr>
<tr>
<td>7</td>
<td>1504.0376</td>
<td>1504.9309</td>
<td>0.05</td>
</tr>
<tr>
<td>8</td>
<td>1911.6293</td>
<td>1915.2443</td>
<td>0.18</td>
</tr>
<tr>
<td>9</td>
<td>2061.4500</td>
<td>2063.2616</td>
<td>0.08</td>
</tr>
<tr>
<td>10</td>
<td>2447.5039</td>
<td>2448.8780</td>
<td>0.05</td>
</tr>
<tr>
<td>11</td>
<td>2695.0238</td>
<td>2718.8640</td>
<td>0.88</td>
</tr>
<tr>
<td>12</td>
<td>2903.7459</td>
<td>2935.6818</td>
<td>1.09</td>
</tr>
<tr>
<td>13</td>
<td>4171.0937</td>
<td>4238.3727</td>
<td>1.61</td>
</tr>
<tr>
<td>14</td>
<td>4618.2581</td>
<td>4716.7942</td>
<td>2.13</td>
</tr>
<tr>
<td>15</td>
<td>4943.6005</td>
<td>4983.1073</td>
<td>0.79</td>
</tr>
<tr>
<td>16</td>
<td>5612.5382</td>
<td>5648.0927</td>
<td>0.63</td>
</tr>
<tr>
<td>17</td>
<td>5885.1186</td>
<td>5924.3911</td>
<td>0.66</td>
</tr>
<tr>
<td>18</td>
<td>6405.0077</td>
<td>6409.2304</td>
<td>0.06</td>
</tr>
<tr>
<td>19</td>
<td>6949.5716</td>
<td>7703.0385</td>
<td>10.84</td>
</tr>
<tr>
<td>20</td>
<td>7227.1974</td>
<td>8112.0575</td>
<td>12.24</td>
</tr>
<tr>
<td>21</td>
<td>9227.6363</td>
<td>9778.0739</td>
<td>5.96</td>
</tr>
<tr>
<td>22</td>
<td>9648.5681</td>
<td>10576.6049</td>
<td>9.61</td>
</tr>
<tr>
<td>23</td>
<td>10349.2749</td>
<td>10716.8253</td>
<td>3.55</td>
</tr>
<tr>
<td>24</td>
<td>11343.0263</td>
<td>11687.6693</td>
<td>3.03</td>
</tr>
<tr>
<td>25</td>
<td>11550.2063</td>
<td>12253.8659</td>
<td>6.09</td>
</tr>
<tr>
<td>26</td>
<td>11931.5463</td>
<td>12367.4574</td>
<td>3.65</td>
</tr>
<tr>
<td>27</td>
<td>12249.1092</td>
<td>12764.6829</td>
<td>4.20</td>
</tr>
<tr>
<td>28</td>
<td>12862.2577</td>
<td>13552.1932</td>
<td>5.36</td>
</tr>
<tr>
<td>29</td>
<td>13650.0872</td>
<td>15985.6317</td>
<td>17.11</td>
</tr>
<tr>
<td>30</td>
<td>14190.6086</td>
<td>17183.8486</td>
<td>21.09</td>
</tr>
<tr>
<td>31</td>
<td>16589.2026</td>
<td>17709.1930</td>
<td>6.75</td>
</tr>
<tr>
<td>32</td>
<td>17151.7868</td>
<td>18900.0709</td>
<td>10.19</td>
</tr>
<tr>
<td>33</td>
<td>17505.5446</td>
<td>19667.0212</td>
<td>12.34</td>
</tr>
<tr>
<td>34</td>
<td>18789.5466</td>
<td>19812.3991</td>
<td>5.44</td>
</tr>
<tr>
<td>35</td>
<td>19224.3217</td>
<td>23436.8548</td>
<td>21.91</td>
</tr>
</tbody>
</table>
Figure 17. CPU seconds per Acceptable Eigenvalue versus Model Size for the Verdeel Truss.
freedom in a FEM model in DSTAP then it is more economical to use an exact-beam model in DSTAP. As this corresponds to 8 acceptable eigenvalues, the break-even point for the exact is 8 acceptable eigenvalues. Again, the combined model data shows too much scatter for a meaningful regression. The data for these models seem to be scattered mostly around 90 CPU sec per eigenvalue. Using this value, the break-even point for the combined models is about 36 eigenvalues.

The solution times for the SUPERB FEM models are lower than those from the DSTAP FEM and combined models. In fact, according to Fig. 17, the combined models could not give more economical solutions than SUPERB, at least for the range of eigenvalues sought. It is important to realize that the combined models were using the DSTAP FEM-beam element which was seen to be less effective than the SUPERB FEM-beam element for higher eigenvalues. Thus, this is an unfair comparison for the combined models. Furthermore, while SUPERB uses a sophisticated eigenvalue extraction technique, the DSTAP eigenvalue extraction for the FEC models is rather crude. The last reason why this is an unfair comparison for the FEC models is that DSTAP automatically finds the eigenvectors for the model degrees of freedom, and SUPERB was not asked to find eigenvectors.

The DSTAP exact-beam model solution was by far the least expensive for higher eigenvalues. It cost an average of 5.38 CPU seconds to extract an acceptable eigenvalue with an exact model in DSTAP, regardless of whether it was a high or low eigenvalue. As pointed out above, this model is more economical than the DSTAP FEM-beam element model if 8 or more acceptable eigenvalues are sought. The break-even point with the SUPERB solutions is 17 acceptable eigenvalues. This is quite remarkable considering that the same
rather crude eigenvalue extraction algorithm used for the FEC models is used for exact models in DSTAP. Moreover, DSTAP also finds the mode shapes automatically for the exact models, whereas SUPERB was not asked to find eigenvectors. However, in spite of these handicaps, the exact model in DSTAP gave the most economical way to compute the higher eigenvalues.
CONCLUSIONS

An exact beam-element formulation, based on a converted continuum transfer matrix was presented. This exact-element formulation was incorporated into a typical FEM code. In this setting, the exact-element formulation was shown to be equivalent to a dynamic condensation of an infinity of FEM degrees of freedom. Models using this exact element generate a transcendental, nonlinear eigenvalue problem, unsolvable with traditional linear eigenvalue problem extraction algorithms. A simple extraction technique was implemented into the hybrid finite element program DSTAP to study the effects of the exact-beam element on the approximation of higher structure eigenvalues. This extraction algorithm had to take into account the poles of the determinant of the dynamic stiffness matrix of each exact element. If the poles are not handled properly, there is the possibility of either predicting nonexistent eigenvalues or exceeding the computer representation of real numbers.

In the Numerical Examples section it was seen that an exact-element model is better suited for the extraction of higher structure eigenvalues/vectors than a standard FEM model. The first numerical example showed that an exact-element model is able to approximate the correct eigenvalues with an accuracy unmatched by an FEM model of the same size, or even models seven times larger. A fixed-fixed beam was modeled with only two equal-size-exact-beam elements, giving a total of 3 degrees of freedom. These 3 degrees of freedom lead to a 3 by 3 dynamic stiffness matrix, which was sufficient to approximate the 22 eigenvalues lying in the 1 to 100,000 rad/sec range. The computed eigenvalues matched to 9 significant digits with the
correct, closed-form-solution eigenvalues up to 60,000 rad/sec. For the extremely high structure frequencies between 60,000 and 100,000 rad/sec the accuracy of the exact-element model dropped slightly. By the 22nd eigenvalue, the last eigenvalue in the range, the computed eigenvalues matched to 5 significant digits with the correct eigenvalues, giving a relative error of less than two ten-thousandths of one percent. An FEM model of the same size was only able to approximate 3 eigenvalues and only matched the first eigenvalue of the closed-form solution to 1 digit.

The addition of this exact-beam-element into an FEM model improved the capabilities of the FEM models. The resulting models of combined FEM and exact-beam elements needed fewer degrees of freedom than an FEM model to calculate acceptable higher eigenvalue approximations. The reduction of model size is an advantage when higher eigenvalues in large structures are needed with precision. Also, in the upper frequency region, the models using exact-elements had lower execution times per acceptable eigenvalue. The lower execution time per acceptable eigenvalue was an unexpected benefit of the exact-element incorporation which resulted from the rather simple transcendental-eigenvalue-extraction algorithm used. Furthermore, since the addition of an exact-element was shown to be analogous to the dynamic condensation of an infinite number of degrees of freedom, the resulting eigenvectors are actually improved. That is, smaller models give better eigenvectors because the reconstruction of the condensed degrees of freedom can be accomplished with almost any precision required. The only limiting factor is the truncation error incurred by the finite arithmetic on the computer. For the frequencies above 12,000 rad/sec, the truncation error was overcome by using a larger word length.
In the second example, a Portal Arch was modeled with purely exact-beam-element models, purely FEM-beam-element models, and with combined FEM-exact-beam-element models. Of the three types of models, the exact-beam-element models were the best suited for the extraction of higher eigenvalues and had the lowest execution time per acceptable eigenvalue. A three-element model with 6 degrees of freedom was able to extract the 35 eigen-pairs lying in the frequency of interest, 1 to 33,000 rad/sec. The only difficulty encountered with this model occurred in the 33rd eigenvalue of the model with neither shear deformation nor rotatory inertia. This eigenpair was extracted by resorting to a 6-element model, thus, pushing the poles to higher frequencies. In other words, when it becomes difficult to find an eigenvalue due to the presence of poles, then by reducing the size of the beam elements, it is possible to change the location of the poles.

Of the two remaining types of models used to model the Portal Arch, the combined FEM-exact-beam-element models were better suited than the FEM-beam-element models for eigenvalue extraction. That is, for the same size model, the combined models gave better approximations for the higher eigenvalues. The combined models also gave lower solution times per acceptable eigenvalue than the purely FEM-element models. Naturally, the number of exact-elements as well as their location regulated the results for the combined models. These choices depend on the modeling techniques of the analyst. Thus, the combined-model results presented here are dependent on the author's choice of location and quantity of exact-elements. These choices were arbitrary because the optimum selection was beyond the scope of this study. Until such a study is conducted, the analyst using combined models must develop an engineering "feel" for the number of exact-elements and their location. The
rule of thumb used here was to replace as many FEM-elements as possible with a single exact-element.

In the third example, a Verdeel Truss structure was modeled with the same type of models as used in the previous example in program DSTAP and with standard FEM models in the commercial program SUPERB. The observations presented above were also applicable to the results from this example. The exact-element models were the best suited for extracting higher eigenvalues and had the lowest execution time per acceptable eigenvalue. The combined models were better suited than the purely FEM-element models in the upper frequency region. Finally, the hybrid research code presented here, DSTAP, was compared with a commercial code, SUPERB. These comparisons showed that the SUPERB-FEM models were better suited for higher eigenvalue extraction than the DSTAP-FEM models. Although the DSTAP-FEM-beam element was not as well formulated for higher frequencies as the SUPERB-FEM-beam element, the DSTAP combined models were better suited for higher eigenvalue/vector extractions than either the SUPERB-FEM or DSTAP-FEM models.

In summary, the addition of exact-beam elements to plane structure FEM-element models improves the approximations of higher structure eigenvalues. It will be up to the analyst to determine the choice of location and number of exact-elements to be added. From the results observed here, it is evident that if the structure to be modeled entirely of beams, then an exact-element model will be the better model. Moreover, if the structure to be modeled has non-beam components which can be found in transfer matrix catalogs, then it would be advantageous to convert these matrices to stiffness formulation and generate a new exact-element to be used with the exact-beam-element to model the structure.
Up to this point the advantages of using the exact element in DSTAP have been compared to the standard finite element method; however, DSTAP also has advantages over the standard transfer matrix method. By recasting the continuum transfer matrix into a general stiffness formulation, branching in the TMM can be easily accomplished. The user need not keep track of which path or branch is being followed nor in which direction they are being followed. As pointed out by Pestel and Leckie [42], this bookkeeping of paths, branches, and directions is rather cumbersome, and makes the TMM undesirable for complex geometries. Furthermore, by recasting a transfer matrix as a simple transformation into a stiffness matrix, the versatility of creating new elements with the TMM is preserved. That is, an analyst may create exact elements by combining several different continuum transfer matrices to describe a physical substructure. Then, the total transfer matrix may be entered into the code presented here to generate a new exact-element. Thus, although there are libraries of continuum transfer matrices for many applications, an analyst is not limited to these libraries, since he or she may easily create new exact-elements.
RECOMMENDATIONS FOR FUTURE WORK

Few investigations, if any, are ever completely finished. This investigation has shown that exact-displacement-beam elements are a feasible alternative to FEM-beam elements for the extraction of higher structure eigenvalues. The results presented here raise several unanswered questions. Some of these questions appear here as recommendations for future work.

A. Eigenvalue Sensitivity Study:

The dynamical stiffness of the beam exact displacement element contains the hyperbolic functions Sinh and Cosh, and the trigonometric functions Sin and Cos. The hyperbolic functions are known to grow very rapidly in magnitude as the frequency increases, whereas the magnitude of the trigonometric functions never exceeds unity. Several of the terms associated with the transverse deflection of an exact beam are made up of sums of trigonometric and hyperbolic functions. Therefore, there is the possibility that the contribution of the trigonometric functions will be lost in these terms at higher frequencies, due to machine truncation. A sensitivity study may lead to a better arrangement of the functions needed to evaluate the dynamic stiffness matrix. Ideally, the structure dynamic matrix might be decomposed into a product of two matrices, one matrix containing all the hyperbolic terms and another one containing the trigonometric terms. It may not be possible to obtain the ideal decomposition; however, determining which terms are more sensitive to frequency, geometry, and material parameters may lead to a better arrangement of the terms to reduce the truncation errors.
B. Eigenvector Sensitivity Study:

The acceptability of a solution to the eigenvalue problem was measured strictly on the basis of the computed eigenvalue in this study. This criteria does not guarantee that the corresponding eigenvectors will be likewise acceptable. Therefore, it should prove advantageous to study the sensitivity of both the purely FEM models and the combined models for eigenvector extractions. Qualitatively, the combined models should provide better eigenvector approximations. This study could quantify the error in eigenvector approximations and establish an acceptability criteria based on this error.

C. Eigenvalue Extraction Algorithm:

The eigenvalue extraction used here is a simple algorithm. Thus, there is much room for improvement. First, if the computer used is a sequential processor, the extraction algorithm can be easily changed such that the function $g(x)$ is curve fitted with an $n^{th}$ order polynomial. The zeroes of the polynomial can be used to determine intervals that contain a zero of $g(x)$. This investigation should determine a suitable size polynomial. Preferably, the selection of the correct size polynomial could be automated with an empirical formula so that it would depend on the problem size. Second, if the computer used is a parallel processor, then the evaluation of $g(x)$ can be made simultaneously at several frequencies. These values of $g(x)$ can then be used to establish intervals containing roots or the roots themselves, depending on the number of evaluations. Once the intervals are found the search can continue in several intervals at the same time.

D. Linearization of the Non-Linear Eigenvalue Problem:

For the same size matrices, the FEM matrices gave more economical solutions than the exact displacement method, but the exact displacement method
gave more accurate eigenvalues. Should the exact displacement method eigenvalue problem be approximated by a linear eigenvalue problem, then the solution would take advantage of both methods. As presented earlier, linearization attempts have been able to provide bounds to the true eigenvalue(s) provided both the linearization and eigenvalue(s) were below the first pole. Once a pole is crossed the theorems developed for the linearization technique were no longer valid.

The current linearization techniques have been developed for frequencies below the first pole. This study has shown that it is at high frequencies when there is a definite advantage to using exact elements. The high frequencies of interest may lie above the first pole, so current linearization techniques are not really useful. These linearization schemes have been developed for models made up entirely of exact elements. In the combined models, the contribution of the FEM-elements does not need to be linearized. Thus, this fact can be used as a starting point for the linearization scheme. The task then is to find the contribution from the exact elements to this linear eigenvalue problem.

E. Extension to Other Exact-Displacement Models:

As presented in the discussion of the Transfer Matrix Method, there are many other continuum elements in TMM catalogs. These transfer matrices can be recast as exact-displacement element matrices by the transformation presented here and added to DSTAP. Thus, further studies with other exact-displacement elements can be carried out to verify the conclusions presented here. In particular, since plates and shells are popular FEM elements, it would be advantageous to develop exact-displacement shell and plate elements. The results of such a study should provide some interesting results
since the FEM plate and shell elements are based on energy formulations, which may make these elements less stable than the FEM-beam element used in this study. The FEM-beam element may be formulated without resorting to energy formulations.
REFERENCES


APPENDIX A

DYNAMIC ARRAY POINTERS IN DSTAP

The tables in this appendix contain the dynamic storage allocation for the A-array stored in the unnamed common block of program DSTAP. An explanation of the basics behind dynamic storage and matrix skyline storage may be found in Ref. [9]. Thus, they are not reproduced here.

Each of the attached tables corresponds to a particular section of program DSTAP. This section can be determined by the title of the table. For example, Table A-1 is applicable only during the assembly of the [K] and [M] matrices due to finite elements, the second time subroutine FEBEAM is called.

Appendix B shows how the pointers and dynamic arrays are used in DSTAP. The pointers N1 through N30 are stored in common block DIM, while pointers N101 through N114 are stored in common block CONPTS.
Table A-1. Distribution of the Dynamic Storage Area (A-array) During the Finite Element Matrix Assembly Process

<table>
<thead>
<tr>
<th>A-array Pointers</th>
<th>Number of terms*</th>
<th>Array Name</th>
<th>Array Type</th>
<th>Array Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td>3(NUMNP)</td>
<td>ID</td>
<td>INTEGER</td>
<td>Equation numbers associated with each degree of freedom</td>
</tr>
<tr>
<td>N2</td>
<td>NEQ + 1</td>
<td>MAXA</td>
<td>INTEGER</td>
<td>Pointer to the main diagonal terms which are stored in skyline fashion</td>
</tr>
<tr>
<td>N3</td>
<td>NWK</td>
<td>K</td>
<td>REAL*8</td>
<td>FEM stiffness matrix, [K]</td>
</tr>
<tr>
<td>N4</td>
<td>NWK</td>
<td>M</td>
<td>REAL*8</td>
<td>FEM consistent mass matrix, [M]</td>
</tr>
<tr>
<td>N5 N101</td>
<td>NUMMAT</td>
<td>E</td>
<td>REAL*8</td>
<td>Modulus of elasticity</td>
</tr>
<tr>
<td>N102</td>
<td>NUMMAT</td>
<td>AREA</td>
<td>REAL*8</td>
<td>Cross section area</td>
</tr>
<tr>
<td>N103</td>
<td>NUMMAT</td>
<td>IXX</td>
<td>REAL*8</td>
<td>Area moment of inertia</td>
</tr>
<tr>
<td>N104</td>
<td>NUMMAT</td>
<td>RHO</td>
<td>REAL*8</td>
<td>Density of the material</td>
</tr>
<tr>
<td>N105</td>
<td>NUME</td>
<td>XYZ</td>
<td>REAL*8</td>
<td>Coordinates of the end-points of each beam finite element</td>
</tr>
<tr>
<td>N6</td>
<td>NLAST</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Definition of terms in column two:
- NEQ - Number of system equations
- NWK - Number of elements in upper skyline storage
- NUMNP - Number of nodal points
- NUMMAT - Number of different materials
- NUME - Number of finite elements

Table A-2. Distribution of the Dynamic Storage Area (A-array) During the Exact Beam Element Matrix Assembly Process

<table>
<thead>
<tr>
<th>A-array Pointers</th>
<th>Number of terms*</th>
<th>Array Name</th>
<th>Array Type</th>
<th>Array Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td>3(NUMNP)</td>
<td>ID</td>
<td>INTEGER</td>
<td>Equation numbers associated with each degree of freedom</td>
</tr>
<tr>
<td>N2</td>
<td>NEQ + 1</td>
<td>MAXA</td>
<td>INTEGER</td>
<td>Pointer to the main diagonal terms which are stored in skyline fashion</td>
</tr>
<tr>
<td>N3</td>
<td>NWK</td>
<td>K</td>
<td>REAL*8</td>
<td>FEM stiffness matrix, [K]</td>
</tr>
<tr>
<td>N4</td>
<td>NWK</td>
<td>M</td>
<td>REAL*8</td>
<td>FEM consistent mass matrix, [M]</td>
</tr>
<tr>
<td>N5 N8</td>
<td>NUMMAT</td>
<td>E</td>
<td>REAL*8</td>
<td>Modulus of elasticity</td>
</tr>
<tr>
<td>N6</td>
<td>NUMMAT</td>
<td>G</td>
<td>REAL*8</td>
<td>Shear modulus of rigidity</td>
</tr>
<tr>
<td>NARE</td>
<td>NUMMAT</td>
<td>AREA</td>
<td>REAL*8</td>
<td>Cross section area</td>
</tr>
<tr>
<td>NIXX</td>
<td>NUMMAT</td>
<td>IXX</td>
<td>REAL*8</td>
<td>Area moment of inertia</td>
</tr>
<tr>
<td>NRHO</td>
<td>NUMMAT</td>
<td>RHO</td>
<td>REAL*8</td>
<td>Density of the material</td>
</tr>
<tr>
<td>NLM</td>
<td>NUME</td>
<td>LM</td>
<td>INTEGER</td>
<td>Equation numbers associated with each beam finite element</td>
</tr>
<tr>
<td>NXYZ</td>
<td>NUME</td>
<td>XYZ</td>
<td>REAL*8</td>
<td>Coordinates of the end-points of each beam finite element</td>
</tr>
<tr>
<td>NMMAT</td>
<td>NUME</td>
<td>MATP</td>
<td>INTEGER</td>
<td>Material number of each element</td>
</tr>
<tr>
<td>N6</td>
<td>NLAST</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Definition of terms in column two:
- NEQ - Number of system equations
- NWK - Number of elements in upper skyline storage
- NUMNP - Number of nodal points
- NUMMAT - Number of different materials
- NUME - Number of finite elements
Table A-3. Distribution of the Dynamic Storage Area (A-array) During An Eigenvalue Extraction using EIGENR

<table>
<thead>
<tr>
<th>A-array Pointers</th>
<th>Number of Terms*</th>
<th>Array Name</th>
<th>Array Type</th>
<th>Array Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td>3(NUMNP)</td>
<td>ID</td>
<td>INTEGER</td>
<td>Equation numbers associated with each degree of freedom</td>
</tr>
<tr>
<td>N2</td>
<td>NEQ + 1</td>
<td>MAXA</td>
<td>INTEGER</td>
<td>Pointer to the main diagonal terms which are stored in skyline fashion</td>
</tr>
<tr>
<td>N3</td>
<td>NWK</td>
<td>K</td>
<td>REAL*8</td>
<td>FEM stiffness matrix, [K]</td>
</tr>
<tr>
<td>N4</td>
<td>NWK</td>
<td>M</td>
<td>REAL*8</td>
<td>FEM consistent mass matrix, [M]</td>
</tr>
<tr>
<td>N5</td>
<td>NEQ*NEQ</td>
<td>A</td>
<td>REAL*8</td>
<td>Square [K] matrix</td>
</tr>
<tr>
<td>N6</td>
<td>NEQ*NEQ</td>
<td>B</td>
<td>REAL*8</td>
<td>Square [M] matrix</td>
</tr>
<tr>
<td>N7</td>
<td>NEQ</td>
<td>EIG</td>
<td>REAL*8</td>
<td>Eigenvalues returned by EIGENR</td>
</tr>
<tr>
<td>N8</td>
<td>NEQ*NEQ</td>
<td>W</td>
<td>REAL*8</td>
<td>Work area and eigenvectors</td>
</tr>
</tbody>
</table>

*Definition of terms in column two:
- NEQ: Number of system equations
- NUMNP: Number of nodal points
- NWK: Number of matrix element in skyline storage

Table A-4. Distribution of the Dynamic Storage Area (A-array) During the Search for Poles

<table>
<thead>
<tr>
<th>A-array Pointers</th>
<th>Number of Terms*</th>
<th>Array Name</th>
<th>Array Type</th>
<th>Array Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td>3(NUMNP)</td>
<td>ID</td>
<td>INTEGER</td>
<td>Equation numbers associated with each degree of freedom</td>
</tr>
<tr>
<td>N2</td>
<td>NEQ + 1</td>
<td>MAXA</td>
<td>INTEGER</td>
<td>Pointer to the main diagonal terms which are stored in skyline fashion</td>
</tr>
<tr>
<td>N3</td>
<td>NWK</td>
<td>K</td>
<td>REAL*8</td>
<td>FEM stiffness matrix, [K]</td>
</tr>
<tr>
<td>N4</td>
<td>NWK</td>
<td>M</td>
<td>REAL*8</td>
<td>FEM consistent mass matrix, [M]</td>
</tr>
<tr>
<td>N5</td>
<td>NUMMAT</td>
<td>E</td>
<td>REAL*8</td>
<td>Modulus of elasticity</td>
</tr>
<tr>
<td>N6</td>
<td>NUMMAT</td>
<td>G</td>
<td>REAL*8</td>
<td>Shear modulus of rigidity</td>
</tr>
<tr>
<td>N7</td>
<td>NUMMAT</td>
<td>AREA</td>
<td>REAL*8</td>
<td>Cross section area</td>
</tr>
<tr>
<td>N8</td>
<td>NUMMAT</td>
<td>IX</td>
<td>REAL*8</td>
<td>Area moment of inertia</td>
</tr>
<tr>
<td>N9</td>
<td>NUMMAT</td>
<td>RHO</td>
<td>REAL*8</td>
<td>Density of the material</td>
</tr>
<tr>
<td>N10</td>
<td>(NCEL)</td>
<td>LM</td>
<td>INTEGER</td>
<td>Equation numbers associated with each continuous element</td>
</tr>
<tr>
<td>N11</td>
<td>(NCEL)</td>
<td>XY</td>
<td>REAL*8</td>
<td>Coordinates of the end-points of each continuous element</td>
</tr>
<tr>
<td>N12</td>
<td>MATP</td>
<td>INTEGER</td>
<td>Material number of each element</td>
<td></td>
</tr>
<tr>
<td>N13</td>
<td>(NCEL)</td>
<td>SN</td>
<td>REAL*8</td>
<td>Sine of the orientation angle for each continuous element</td>
</tr>
<tr>
<td>N14</td>
<td>(NCEL)</td>
<td>CS</td>
<td>REAL*8</td>
<td>Cosine of the orientation angle for each continuous element</td>
</tr>
<tr>
<td>N15</td>
<td>(NCEL)</td>
<td>S2</td>
<td>REAL*8</td>
<td>S2=SN*SN</td>
</tr>
<tr>
<td>N16</td>
<td>(NCEL)</td>
<td>C2</td>
<td>REAL*8</td>
<td>C2=CS*CS</td>
</tr>
<tr>
<td>N17</td>
<td>(NCEL)</td>
<td>SC</td>
<td>REAL*8</td>
<td>SC=CS*SN</td>
</tr>
<tr>
<td>N18</td>
<td>(NCEL)</td>
<td>XL</td>
<td>REAL*8</td>
<td>Length of each continuous element</td>
</tr>
</tbody>
</table>
Table A-4. Distribution of the Dynamic Storage Area (A-array) During the Search for Poles (continued).

<table>
<thead>
<tr>
<th>A-array Pointers</th>
<th>Number of Array terms*</th>
<th>Name</th>
<th>Array Type</th>
<th>Array Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>N107</td>
<td>NCEL</td>
<td>MU</td>
<td>REAL*8</td>
<td>Density per unit length</td>
</tr>
<tr>
<td>N108</td>
<td>NCEL</td>
<td>SF</td>
<td>REAL*8</td>
<td>Constant needed to compute [K(w)]</td>
</tr>
<tr>
<td>N109</td>
<td>NCEL</td>
<td>TF</td>
<td>REAL*8</td>
<td>Constant needed to compute [K(w)]</td>
</tr>
<tr>
<td>N110</td>
<td>NCEL</td>
<td>BF</td>
<td>REAL*8</td>
<td>Constant needed to compute [K(w)]</td>
</tr>
<tr>
<td>N111</td>
<td>NCEL</td>
<td>AXB</td>
<td>REAL*8</td>
<td>Constant needed to compute [K(w)]</td>
</tr>
<tr>
<td>N112</td>
<td>NCEL</td>
<td>AXOF</td>
<td>REAL*8</td>
<td>Constant needed to compute [K(w)]</td>
</tr>
<tr>
<td>N113</td>
<td>6(NCEL)</td>
<td>LM</td>
<td>INTEGER</td>
<td>Equation number associated with each element</td>
</tr>
<tr>
<td>N6, N114</td>
<td>(MTOT-N114)/3</td>
<td>IEQ</td>
<td>INTEGER</td>
<td>Work array</td>
</tr>
<tr>
<td>N7, N102</td>
<td>4(MTOT-N114)/9</td>
<td>IARRAY</td>
<td>REAL*8</td>
<td>Work array</td>
</tr>
<tr>
<td>N8, N106</td>
<td>(MTOT-N114)/9</td>
<td>POL</td>
<td>REAL*8</td>
<td>Poles of [K(w)]</td>
</tr>
</tbody>
</table>

* Definition of terms for the column two:

- NCEL: Number of exact elements
- NEQ: Number of system equations
- NPOL: Number of poles extracted
- NUMNP: Number of nodal points
- NUMMA: Number of different materials
- NWK: Number of terms stored in skyline fashion
- MTOT: Total amount of dynamic storage

Table A-5. Distribution of the Dynamic Storage Area (A-array) During the Execution of Subroutine BMPHED

<table>
<thead>
<tr>
<th>A-array Pointers</th>
<th>Number of Array terms*</th>
<th>Name</th>
<th>Array Type</th>
<th>Array Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td>3(NUMNP)</td>
<td>ID</td>
<td>INTEGER</td>
<td>Equation numbers associated with each degree of freedom</td>
</tr>
<tr>
<td>N2</td>
<td>NEQ + 1</td>
<td>MAXA</td>
<td>INTEGER</td>
<td>Pointer to the main diagonal terms which are stored in skyline fashion</td>
</tr>
<tr>
<td>N3</td>
<td>NWK</td>
<td>K</td>
<td>REAL*8</td>
<td>FEM stiffness matrix, [K]</td>
</tr>
<tr>
<td>N4</td>
<td>NWK</td>
<td>M</td>
<td>REAL*8</td>
<td>FEM consistent mass matrix, [M]</td>
</tr>
<tr>
<td>N5, N101</td>
<td>NCEL</td>
<td>SN</td>
<td>REAL*8</td>
<td>Sine of the orientation angle for each continuous element</td>
</tr>
<tr>
<td>N102</td>
<td>NCEL</td>
<td>CS</td>
<td>REAL*8</td>
<td>Cosine of the orientation angle for each continuous element</td>
</tr>
<tr>
<td>N103</td>
<td>NCEL</td>
<td>S2</td>
<td>REAL*8</td>
<td>S2=SNPSN</td>
</tr>
<tr>
<td>N104</td>
<td>NCEL</td>
<td>C2</td>
<td>REAL*8</td>
<td>C2=CPSCS</td>
</tr>
<tr>
<td>N105</td>
<td>NCEL</td>
<td>SC</td>
<td>REAL*8</td>
<td>SC=CPSCSN</td>
</tr>
<tr>
<td>N106</td>
<td>NCEL</td>
<td>XL</td>
<td>REAL*8</td>
<td>Length of each continuous element</td>
</tr>
<tr>
<td>N107</td>
<td>NCEL</td>
<td>MU</td>
<td>REAL*8</td>
<td>Density per unit length</td>
</tr>
<tr>
<td>N108</td>
<td>NCEL</td>
<td>SF</td>
<td>REAL*8</td>
<td>Constant needed to compute [K(w)]</td>
</tr>
<tr>
<td>N109</td>
<td>NCEL</td>
<td>TF</td>
<td>REAL*8</td>
<td>Constant needed to compute [K(w)]</td>
</tr>
<tr>
<td>N110</td>
<td>NCEL</td>
<td>BF</td>
<td>REAL*8</td>
<td>Constant needed to compute [K(w)]</td>
</tr>
<tr>
<td>N111</td>
<td>NCEL</td>
<td>AXB</td>
<td>REAL*8</td>
<td>Constant needed to compute [K(w)]</td>
</tr>
<tr>
<td>N112</td>
<td>NCEL</td>
<td>AXOF</td>
<td>REAL*8</td>
<td>Constant needed to compute [K(w)]</td>
</tr>
<tr>
<td>N113</td>
<td>5(NCEL)</td>
<td>LM</td>
<td>INTEGER</td>
<td>Equation number associated with each element</td>
</tr>
<tr>
<td>N6, N114</td>
<td>NPOL</td>
<td>MULT</td>
<td>INTEGER</td>
<td>Multiplicity of each pole</td>
</tr>
<tr>
<td>N7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table A-5. Distribution of the Dynamic Storage Area (A-array) During the Execution of Subroutine BMTRD (continued).

<table>
<thead>
<tr>
<th>Pointer</th>
<th>Number of terms</th>
<th>Array</th>
<th>Name</th>
<th>Array Type</th>
<th>Array Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>N7</td>
<td>NPOL</td>
<td>POL</td>
<td>REAL*8</td>
<td>Poles of $[K(\omega)]$</td>
<td></td>
</tr>
<tr>
<td>N8</td>
<td>NMK</td>
<td>REAL*8</td>
<td>$(K(\omega))$ and $(K(\omega))^t$ stored in skyline fashion</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N9</td>
<td>(NEQ)(NEQ)</td>
<td>REAL*8</td>
<td>$(K(\omega))^t$ converted to a square storage scheme</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N10</td>
<td>NEQ</td>
<td>JP</td>
<td>INTEGER</td>
<td>Columns of $(K(\omega))^t$ pivoted during upper-triangularization</td>
<td></td>
</tr>
<tr>
<td>N11</td>
<td>NEQ</td>
<td>EIGVEC</td>
<td>REAL*8</td>
<td>Eigenvector Extracted from the matrix $(K(\omega))^t$</td>
<td></td>
</tr>
<tr>
<td>N12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Definition of terms for the column two:
  - NCEL - Number of exact elements
  - NEQ - Number of system equations
  - NPOL - Number of poles extracted
  - NUMNP - Number of nodal points
  - NMK - Number of terms stored in skyline fashion

### Table A-6. Distribution of the Dynamic Storage Area (A-array) During the Post-Processing of Eigenvectors with Exact Elements

<table>
<thead>
<tr>
<th>Pointer</th>
<th>Number of terms</th>
<th>Array</th>
<th>Name</th>
<th>Array Type</th>
<th>Array Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td>3(NUMNP)</td>
<td>ID</td>
<td>INTEGER</td>
<td>Equation numbers associated with each degree of freedom</td>
<td></td>
</tr>
<tr>
<td>N2</td>
<td>NEQ + 1</td>
<td>MAXA</td>
<td>INTEGER</td>
<td>Pointer to the main diagonal terms which are stored in skyline fashion</td>
<td></td>
</tr>
<tr>
<td>N3</td>
<td>NMK</td>
<td>K</td>
<td>REAL*8</td>
<td>FEM stiffness matrix, $[K]$</td>
<td></td>
</tr>
<tr>
<td>N4</td>
<td>NMK</td>
<td>M</td>
<td>REAL*8</td>
<td>FEM consistent mass matrix, $[M]$</td>
<td></td>
</tr>
<tr>
<td>N5 N101</td>
<td>NCEL</td>
<td>SN</td>
<td>REAL*8</td>
<td>Sine of the orientation angle for each continuous element</td>
<td></td>
</tr>
<tr>
<td>N102</td>
<td>NCEL</td>
<td>CS</td>
<td>REAL*8</td>
<td>Cosine of the orientation angle for each continuous element</td>
<td></td>
</tr>
<tr>
<td>N103</td>
<td>NCEL</td>
<td>S2</td>
<td>REAL*8</td>
<td>$S2$ - SN*SN</td>
<td></td>
</tr>
<tr>
<td>N104</td>
<td>NCEL</td>
<td>C2</td>
<td>REAL*8</td>
<td>$C2$ - CS*CS</td>
<td></td>
</tr>
<tr>
<td>N105</td>
<td>NCEL</td>
<td>SC</td>
<td>REAL*8</td>
<td>$SC$ - CS*CS</td>
<td></td>
</tr>
<tr>
<td>N106</td>
<td>NCEL</td>
<td>XL</td>
<td>REAL*8</td>
<td>Length of each continuous element</td>
<td></td>
</tr>
<tr>
<td>N107</td>
<td>NCEL</td>
<td>MU</td>
<td>REAL*8</td>
<td>Density per unit length</td>
<td></td>
</tr>
<tr>
<td>N108</td>
<td>NCEL</td>
<td>SF</td>
<td>REAL*8</td>
<td>Constant needed to compute $[K(\omega)]$</td>
<td></td>
</tr>
<tr>
<td>N109</td>
<td>NCEL</td>
<td>TF</td>
<td>REAL*8</td>
<td>Constant needed to compute $[K(\omega)]$</td>
<td></td>
</tr>
<tr>
<td>N110</td>
<td>NCEL</td>
<td>BF</td>
<td>REAL*8</td>
<td>Constant needed to compute $[K(\omega)]$</td>
<td></td>
</tr>
<tr>
<td>N111</td>
<td>NCEL</td>
<td>AXB</td>
<td>REAL*8</td>
<td>Constant needed to compute $[K(\omega)]$</td>
<td></td>
</tr>
<tr>
<td>N112</td>
<td>NCEL</td>
<td>AOKF</td>
<td>REAL*8</td>
<td>Constant needed to compute $[K(\omega)]$</td>
<td></td>
</tr>
<tr>
<td>N113</td>
<td>6(NCEL)</td>
<td>LM</td>
<td>INTEGER</td>
<td>Equation number associated with each element</td>
<td></td>
</tr>
<tr>
<td>N114</td>
<td>NUMNP</td>
<td>X</td>
<td>REAL*8</td>
<td>$X$ coordinates of the model nodes</td>
<td></td>
</tr>
<tr>
<td>N7</td>
<td>NUMNP</td>
<td>X</td>
<td>REAL*8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A-array Pointers</td>
<td>Number of terms*</td>
<td>Array Name</td>
<td>Array Type</td>
<td>Array Content</td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>------------------</td>
<td>------------</td>
<td>------------</td>
<td>---------------</td>
<td></td>
</tr>
<tr>
<td>N7</td>
<td>NUMNP</td>
<td>Y</td>
<td>REAL*8</td>
<td>Y coordinates of the model nodes</td>
<td></td>
</tr>
<tr>
<td>N8</td>
<td>NFEL#2</td>
<td>FELM</td>
<td>INTEGER</td>
<td>FEM beam elements</td>
<td></td>
</tr>
<tr>
<td>N9</td>
<td>NCEL#2</td>
<td>CELM</td>
<td>INTEGER</td>
<td>Exact beam elements</td>
<td></td>
</tr>
<tr>
<td>N10</td>
<td>NEQ</td>
<td>XV</td>
<td>REAL*8</td>
<td>Eigenvector read in from tape</td>
<td></td>
</tr>
<tr>
<td>N11</td>
<td>MAXENT</td>
<td>X</td>
<td>REAL*8</td>
<td>X coordinate of plot point</td>
<td></td>
</tr>
<tr>
<td>N12</td>
<td>MAXENT</td>
<td>Y</td>
<td>REAL*8</td>
<td>Y coordinate of plot point</td>
<td></td>
</tr>
<tr>
<td>N13</td>
<td>MAXENT</td>
<td>DX</td>
<td>REAL*8</td>
<td>X displacement of plot point</td>
<td></td>
</tr>
<tr>
<td>N14</td>
<td>MAXENT</td>
<td>DY</td>
<td>REAL*8</td>
<td>Y displacement of plot point</td>
<td></td>
</tr>
</tbody>
</table>

* Definition of terms for the column two:

- NCEL - Number of exact elements
- NFEL - Number of FEM elements
- NEQ - Number of system equations
- NUMNP - Number of nodal points
- NMIN - Number of terms stored in skyline fashion
- MAXENT - Maximum number of plot entries:
  - MAXENT = (MTOT-N11)/5
- MTOT - Dimension of the A-array
Table B-1 contains a list of the modules necessary to generate the version of program DSTAP presented in the Program Description section of this dissertation. The first column of Table B-1 contains the module name. The second column contains the file in which the source code of the module may be found, and finally the third column contains the compiler and version number under which the module was compiled.

The files listed in Table B-1 are all on the VAX cluster at the Virginia Tech Computing Center. The subdirectory [JARAALJA.DISS] is on disk DUA3, while the SYS$COMMON and VPI$SYSROOT disks are system defined disks.

The modules appearing in bold in Table B-1 are presented following Table B-1. These subroutines are unique for DSTAP. The remaining subroutines in Table B-1 are subroutines that may be found in Ref. [9] (file STAP3.OLB), Tektronix PLOT10 library (file TCS.OLB), and the IMSL library (file IMSLIBD.OLB). For this reason these subroutines are not included here.
<table>
<thead>
<tr>
<th>Module Name</th>
<th>File containing module</th>
<th>Compiler - version</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSTAP</td>
<td>[JARAVLAJA. DIS3]DSTAP.OBJ;3</td>
<td>VAX FORTRAN V4.1-45</td>
</tr>
<tr>
<td>CPUSERC</td>
<td>[JARAVLAJA. DIS3]DSTAP.OBJ;3</td>
<td>VAX FORTRAN V4.1-45</td>
</tr>
<tr>
<td>PLOTB</td>
<td>[JARAVLAJA. DIS3]PLOTB.OBJ;53</td>
<td>VAX FORTRAN V4.1-45</td>
</tr>
<tr>
<td>INSTVC</td>
<td>[JARAVLAJA. DIS3]PLOTB.OBJ;53</td>
<td>VAX FORTRAN V4.1-45</td>
</tr>
<tr>
<td>PLOTG</td>
<td>[JARAVLAJA. DIS3]PLOTB.OBJ;53</td>
<td>VAX FORTRAN V4.1-45</td>
</tr>
<tr>
<td>MATHG</td>
<td>[JARAVLAJA. DIS3]PLOTB.OBJ;53</td>
<td>VAX FORTRAN V4.1-45</td>
</tr>
<tr>
<td>MATHPL</td>
<td>[JARAVLAJA. DIS3]PLOTB.OBJ;53</td>
<td>VAX FORTRAN V4.1-45</td>
</tr>
<tr>
<td>OTRANAM</td>
<td>[JARAVLAJA. DIS3]PLOTB.OBJ;53</td>
<td>VAX FORTRAN V4.1-45</td>
</tr>
<tr>
<td>ADDVEC</td>
<td>[JARAVLAJA. DIS3]PLOTB.OBJ;53</td>
<td>VAX FORTRAN V4.1-45</td>
</tr>
<tr>
<td>EXPVEC</td>
<td>[JARAVLAJA. DIS3]PLOTB.OBJ;53</td>
<td>VAX FORTRAN V4.1-45</td>
</tr>
<tr>
<td>PGOVR</td>
<td>[JARAVLAJA. DIS3]PLOTB.OBJ;53</td>
<td>VAX FORTRAN V4.1-45</td>
</tr>
<tr>
<td>QINSTVC</td>
<td>[JARAVLAJA. DIS3]PLOTB.OBJ;53</td>
<td>VAX FORTRAN V4.1-45</td>
</tr>
<tr>
<td>OTRANR</td>
<td>[JARAVLAJA. DIS3]PLOTB.OBJ;53</td>
<td>VAX FORTRAN V4.1-45</td>
</tr>
<tr>
<td>OTRAN2</td>
<td>[JARAVLAJA. DIS3]PLOTB.OBJ;53</td>
<td>VAX FORTRAN V4.1-45</td>
</tr>
<tr>
<td>FEASEN</td>
<td>[JARAVLAJA. DIS3]INXEGN.OBJ;3</td>
<td>VAX FORTRAN V3.3-145</td>
</tr>
<tr>
<td>GETCOT</td>
<td>[JARAVLAJA. DIS3]INXEGN.OBJ;3</td>
<td>VAX FORTRAN V3.3-145</td>
</tr>
<tr>
<td>EIGEN</td>
<td>[JARAVLAJA. DIS3]INXEGN.OBJ;3</td>
<td>VAX FORTRAN V3.3-145</td>
</tr>
<tr>
<td>CTVAL</td>
<td>[JARAVLAJA. DIS3]INXEGN.OBJ;3</td>
<td>VAX FORTRAN V3.3-145</td>
</tr>
<tr>
<td>CONDIN</td>
<td>[JARAVLAJA. DIS3]INXEGN.OBJ;3</td>
<td>VAX FORTRAN V3.3-145</td>
</tr>
<tr>
<td>COMUT</td>
<td>[JARAVLAJA. DIS3]INXEGN.OBJ;3</td>
<td>VAX FORTRAN V3.3-145</td>
</tr>
<tr>
<td>POLES</td>
<td>[JARAVLAJA. DIS3]INXEGN.OBJ;3</td>
<td>VAX FORTRAN V3.3-145</td>
</tr>
<tr>
<td>AIRPOL</td>
<td>[JARAVLAJA. DIS3]INXEGN.OBJ;3</td>
<td>VAX FORTRAN V3.3-145</td>
</tr>
<tr>
<td>DETU2</td>
<td>[JARAVLAJA. DIS3]INXEGN.OBJ;3</td>
<td>VAX FORTRAN V3.3-145</td>
</tr>
<tr>
<td>BPHTHD</td>
<td>[JARAVLAJA. DIS3]INXEGN.OBJ;3</td>
<td>VAX FORTRAN V3.3-145</td>
</tr>
<tr>
<td>RPT2</td>
<td>[JARAVLAJA. DIS3]INXEGN.OBJ;3</td>
<td>VAX FORTRAN V3.3-145</td>
</tr>
<tr>
<td>REPORT</td>
<td>[JARAVLAJA. DIS3]INXEGN.OBJ;3</td>
<td>VAX FORTRAN V3.3-145</td>
</tr>
<tr>
<td>WTHRLE</td>
<td>[JARAVLAJA. DIS3]INXEGN.OBJ;3</td>
<td>VAX FORTRAN V3.3-145</td>
</tr>
<tr>
<td>CHB9</td>
<td>[JARAVLAJA. DIS3]INXEGN.OBJ;3</td>
<td>VAX FORTRAN V3.3-145</td>
</tr>
<tr>
<td>CTNCR</td>
<td>[JARAVLAJA. DIS3]INXEGN.OBJ;3</td>
<td>VAX FORTRAN V3.3-145</td>
</tr>
<tr>
<td>FON</td>
<td>[JARAVLAJA. DIS3]INXEGN.OBJ;3</td>
<td>VAX FORTRAN V3.3-145</td>
</tr>
<tr>
<td>CHIOLE</td>
<td>[JARAVLAJA. DIS3]INXEGN.OBJ;3</td>
<td>VAX FORTRAN V3.3-145</td>
</tr>
<tr>
<td>DETERI</td>
<td>[JARAVLAJA. DIS3]INXEGN.OBJ;3</td>
<td>VAX FORTRAN V3.3-145</td>
</tr>
<tr>
<td>CYTDIA</td>
<td>[JARAVLAJA. DIS3]INXEGN.OBJ;3</td>
<td>VAX FORTRAN V3.3-145</td>
</tr>
<tr>
<td>UPDETF</td>
<td>[JARAVLAJA. DIS3]INXEGN.OBJ;3</td>
<td>VAX FORTRAN V3.3-145</td>
</tr>
<tr>
<td>SOLREC</td>
<td>[JARAVLAJA. DIS3]INXEGN.OBJ;3</td>
<td>VAX FORTRAN V3.3-145</td>
</tr>
<tr>
<td>ANP</td>
<td>[HSCDOOSAUDU3] [JARAVLAJA. DIS3]TCS.OLB;1</td>
<td>VAX V3.3-12</td>
</tr>
<tr>
<td>ANRMOE</td>
<td>[HSCDOOSAUDU3] [JARAVLAJA. DIS3]TCS.OLB;1</td>
<td>VAX V3.0-2</td>
</tr>
<tr>
<td>AGOUTST</td>
<td>[HSCDOOSAUDU3] [JARAVLAJA. DIS3]TCS.OLB;1</td>
<td>VAX V3.0-2</td>
</tr>
</tbody>
</table>
**DSTAP**

(Listen STAP)**

version 2.11

Modified version of STAP to include static and free-vibration of a beam element.

Programmer: Jim Jano-Alecone

Language: VAX FORTRAN

Location: Mechanical Engineering Dept.

V.P.I. & S.U.

Blackburg, Virginia

O.S.: VMS 4.1

---

**DISCLAIMER**

This program was developed for research purposes only. It has not been optimized for general use.

---

REAL*8 FREQ1,FREQ2,STEP,NSTP
CHARACTER*50 TIMED(10)
COMMON A(600000)
COMMON /SOL/ NUMP,NEO,NNK,NUMEST,MIDEST,MAKEST,MSK
COMMON /DIM/ N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
1,N16,N17,N18,N19,N20,N21,N22,N23,N24,N25,N26,N27,N28,N29,N30
COMMON /EL/ IND,INC(10),NUME,MOT,NEWST,LST,NUMN,NUMT,
COMMON /VAR/ NS,MODEX
COMMON /DEBUG/ ! Contains the debug flags 1can 0=off
1 ID1, ! The determinant is output to unit 12
2 ID2, ! Not assigned
3 ID3, ! Eigenvector in PLOTB to unit 17
4 ID4, ! Mode shapes of continuum elements are
5 ID5, ! output from EXPVEC to unit 13
6 ID6, ! Mode shapes of continuum elements are
7 ID7, ! State vector used for mode shapes are
* output from PLOTBC to unit 15
* State vector used for mode shapes are
* output from PLOTBC to unit 16
* Initial state vector computations output to unit 14
* COMMON /TAPES/ IELMNT, ILOAD, NSTIF, ID1, ID2, ID3, ID4, MODEX, IEIGV, IEIGV
COMMON /COMPAR/NFEL, NCEL, NVAL, IFP1, FREQ1, FREQ2, STEP, IWTM, NSTP
COMMON /SOIL/TIMEON, TPOL, TIMEON(10), NUM, TIMEIG(60)
COMMON /COMPTS/ N101,N102,N103,N104,N105,N106,N107,N108,N109,
1 N110,N111,N112,N113,N114
COMMON /PLTPAR/ MAXCH, NENTRY, IPLOTF

DIMENSION HED(20), IA(1)

DATA TIMED/
* The input of geometry ..........................................
* The input of loads .............................................
* Computation of global F.E.M. matrices .......................
* Triangularization of F.E.M. [K] matrix ....................
* Solution of load case(s) ....................................
* Eigen solution of a purely F.E.M. problem .................
* Set-up for a Continuum-F.E. problem ......................
* Pole extraction of continuum elements ...................
* 'Brute Force - False extraction' ..........................
* Post-processing of eigenvectors ..........................

EQUVALENCE (A(1),IA(1))

MTOT = 600000
ITWO = 2
ILOAD = 2
NSTIF = 3
IGEM = 183
IEIGV = 24
IDEB = 12
IIN = 25
IOUT = 26

C 200 TIMEON = CRUSEC(0.)
NUMEST = 0
MAKEST = 0
IFPLO = 0
READ (1,1000) HED,NUMP,NUMEG,NLCA,MODEX,IVB,IPLOTF,
1 ID1,ID2,ID3,ID4,ID5,ID6,ID7
IF (NUMP.EQ.0.) STOP 'SUCCESSFUL COMPLETION OF PROGRAM STAP'
WRITE (1,OUT,2000) HED,NUMP,NUMEG,NLCA,MODEX,IVB,IPLOTF
IF (IVB.EQ.0.) THEN
READ (1,1000) METHOD,FREQ1,FREQ2,STEP,IWTM,NSTP
WRITE (100,10000) METHOD,FREQ1,FREQ2,STEP,IWTM,NSTP
END IF
IF (ID1) WRITE (IOUT,3) 1, 'Determinant output sent to unit 12'
IF (ID3) WRITE (IOUT,3) 3, 'PLOTB Eigenvector sent to unit 17'
IF (ID4) WRITE (IOUT,3) 4, 'EXPVEC mode shapes to unit 13'
IF (ID6) WRITE (IOUT,3) 6, 'State vectors from PLOTB to unit 15'
IF (ID7) WRITE (IOUT,3) 7, 'State vector calculations-unit 14'

122
IF (163) WRITE (IOUT, 2003) 3, 'Determinant output sent to unit 12'
IF (164) WRITE (IOUT, 2003) 4, 'EXPEV mode shapes to unit 13'
IF (165) WRITE (IOUT, 2003) 5, 'State vectors from PLOT to unit 15'
IF (166) WRITE (IOUT, 2003) 6, 'Mode shapes from PLOT to unit 16'
IF (167) WRITE (IOUT, 2003) 7, 'State vector calculations-unit 14'
IF (168) WRITE (IOUT, 2003) 1, 'Determinant output sent to unit 12'
IF (169) WRITE (IOUT, 2003) 3, 'PLOT Eigenvector sent to unit 17'
IF (170) WRITE (IOUT, 2003) 4, 'EXPEV mode shapes to unit 13'
IF (171) WRITE (IOUT, 2003) 5, 'State vectors from PLOT to unit 15'
IF (172) WRITE (IOUT, 2003) 6, 'Mode shapes from PLOT to unit 16'
IF (173) WRITE (IOUT, 2003) 7, 'State vector calculations to unit 14'

N1 = 1
N2 = N1 + 3 * NUMNP
N3 = N2 + NUMNP * ITNO
N4 = N3 + NUMNP * ITNO
N5 = N4 + NUMNP * ITNO
IF (N5, GT, MTOT) CALL ERROR (N5 - MTOT, 1)
CALL INPUT (A(N1), A(N2), A(N3), A(N4), NUMNP, NEQ)
NEQ = NEQ + 1

CALL GET

CALL ADRC (A(N2), A(N5))
MM = MM + NEQ
N3 = N2 + NEQ + 1
N4 = N3 + ITNO
N5 = N4 + ITNO
WRITE (IOUT, 2005) NEQ, MM, NMK, N

IF (MODE.LE.1) GOTO 800

IF (IVB.EQ.0) THEN
N5 = N5 + MAXEST
IF (N5, GT, MTOT) CALL ERROR (N5 - MTOT, 4)
NLL = NMNP2 + NEQ
CALL CLEAR (A(N3), NNL)
IND = 2
CALL ASSEM (A(N5))
TIM(3) = CPUSEC (TIMEON)
CALL COLSOL (A(N3), A(N4), A(N2), NEQ, NMK, NEQ, 1, 1)
TIM(4) = CPUSEC (TIMEON)
IND = 3
REWORD ILOAD
DO L=1, NLCASE
CALL LOAD (A(N4), NEQ)
CALL COLSOL (A(N3), A(N4), A(N2), NEQ, NMK, NEQ, 1, 2)
WRITE (IOUT, 205) L
CALL WRITE (A(N4), A(N1), NEQ, NUMNP)
CALL STRESS (A(N5))
END DO
TIM(5) = CPUSEC (TIMEON)
DO 400 I=6, 10
TIM(1) = TIM(5)

400 CONTINUE

ELSE
IND = 2
IF (NEAS.EQ.0) THEN
CALL FEASEM (1)
TIM(2) = TIM(1)
TIM(3) = CPUSEC (TIMEON)
CALL EIGEN (METHOD)
TIM(4) = TIM(3)
TIM(5) = TIM(3)
TIM(6) = CPUSEC (TIMEON)
DO 490 I=1, 10
TIM(1) = TIM(6)

490 CONTINUE
ELSE
IND = 3
CALL FEASEM (0)
TIM(2) = TIM(1)
TIM(3) = CPUSEC (TIMEON)

END IF
END ELSE

! frequency search extraction

CALL BMTHD
TIM(9)=CPUSEC(TIMEON)

NT=NS+NUMNP*TN2
NB=NT+NUMNP*TN2
NC=N9+NFEL*#2
NC1=N9+NCCEL*2
N11=NC1+NEQ*TN2
MAXENT=(N101-N11)/5

STOP 'STOP - Not enough space for plotting eigenvects'
IF (NLOF) CALL PLOTB (A(N6),A(N7),A(N9),A(N10),A(N11),A(N12))

TIM(10)=CPUSEC(TIMEON)

END IF

800 TTT=TIM(1)
WRITE (IOUT,2030) HED
IF (NCCEL.EQ.0) GOTO 805
DO 820 I=KEIG,2,-1
802 TIMEIG(I)=TIMEIG(I)-TIMEIG(I-1)
TIMEIG(1)=TIMEIG(1)-TIMEIG(7)

805 DO 810 I=10,2,-1
TIM(I)=TIM(I)-TIM(I-1)
810 TTT = TTT + TIM(I)
TIM(7)=TIM(7)-TPOL
TIM(8)=TOPS
DO 820 I=1,10
IF (TIM(I).EQ.0) GOTO 820
WRITE (IOUT,2032) TIMEDTIM(1),TIM(I)
820 END DO
WRITE (IOUT,2038) TTT
IF (NCCEL.EQ.0) GOTO 999
WRITE (IOUT,2040)
DO I=1,KEIG
WRITE (IOUT,2042) I,TIMEIG(I)
END DO
999 CONTINUE
C
GO TO 200
C
FUNCTION CRUSEC

Programmer: Jim Jara-Almonte
Location: Mechanical Engineering Dept. V.P.I. & S.U.
Blackburg, Virginia
Language: VAX FORTRAN (V. 4.1-45)
Date: December 1984

DISCLAIMER: This function was developed for research purposes only. It has not been optimized for general use.

FUNCTION PURPOSE:
Evaluate the elapsed number of CPU seconds

REAL*4 FUNCTION CRUSEC (TIMEON)
INTEGER*2 LENGTH,TIMECODE
INTEGER*4 CPU TIME,LENADAR,ADDRESS

DEFINE SYMBOLIC VALUES USED IN THE GETJPI CALL
INCLUDE '(SIFDEF)

SET UP DATA STRUCTURE FOR PASSING INFO TO GETJPI
COMMON /JPILIST/ LENGTH,TIMECODE,ADDRESS,LENADAR
DATA LENGTH/4/
DATA TIMECODE/JPI$CPU TIME/
DATA LENADAR/0/
ADDRESS=ALLOC(CPUTIME)

INCLUDE THE PRECEDING LINES IN YOUR PROGRAM DECLARATION SECTION
WHENEVER YOU WANT A TIME STAMP, ISSUE THE COMMAND BELOW
CPUTIME IS RETURNED IN THE VARIABLE "CPUTIME"
CALL SYSlJPI(,,LENGTH,,)

CUMULATIVE CPU TIME IS GIVEN IN 10 MILISECOND TICS
CRUSEC = FLOAT(CPUTIME)/100. - TIMEON
RETURN
END

SUBROUTINE ELEMNT

Source: Numerical Methods in Finite Element Analysis
K. L. Bathe and E. L. Wilson
Prentice Hall, Inc.
Date: 1978

SUBROUTINE PURPOSE:
This subroutine calls the appropriate element

SUBROUTINE ELEMNT
COMMON /EL/ IND,NPAR(10),NUMEG,MTOT,NFIRST,NLAST,ITWO

NPARI=NPAR(1)
GO TO (1,2,3,4),NPARI
1 CALL TRUSS
RETURN
2 CALL TBEAM ! FEM-beam element
RETURN
3 CALL TBEAM ! Exact-Beam element
4 RETURN
END
SUBROUTINE TBEAM

SUBROUTINE TBEAM

Programmer: Jim Jara-Almonte
Location: Mechanical Engineering Dept. V.P.I. & S.U.
Blackburg, Virginia
Language: VAX FORTRAN (V. 3.5-62)
Date: Aug. 1984

DISCLAIMER: This subroutine was developed for research purposes only. It has not been optimized for general use.

SUBROUTINE PURPOSE:
This subroutine sets up dynamic storage allocation for both the exact and FEM beam elements.

COMMON /SOL/NUMNP,NE3,NK,NUMEST,MIDEST,MAXEST,MR
COMMON /DIM/ N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
COMMON /EL/, INDX, NPAR(10), NUME, MTOT, NFRST, NLAST, ITWO
COMMON /TAPES/, IELMT, ILOAD, NSTIF, IIN, IOUT, IDEB, IVEG, IEGOM, IEIGV
COMMON A(1)
INTEGER ELTP
C
EQUIVALENCE (NPAR(1), ELTP), (NPAR(2), NUME), (NPAR(3), NUMMAT)
IF (NUME.EQ.1) THEN
   MIDEST = 0
RETURN
ENDIF
IF (ELTP.EQ.3) GOTO 500
C
C FINITE ELEMENT METHOD BEAM ELEMENT
C
NFRST=N6
IF (INDX.EQ.1) NFRST=N5
N101=NFRST
NLAST=NUMMAT*ITWO
N102=N101 + NLAST
N103=N102 + NLAST
N104=N103 + NLAST
N105=N104 + NLAST
N106=N105 + NUME
NLAST=N106
C
IF (INDX.EQ.1) GO TO 100
IF (NLAST.GT.MTOT) CALL ERROR(NLAST-MTOT,3)
GO TO 200
C
100 IF (NLAST.GT.MTOT) CALL ERROR(NLAST-MTOT,4)
C
200 MIDEST=NLAST-NFRST
C
CALL FEBEAM (A(N1), A(N2), A(N3), A(N4), A(N5), A(N10), A(N101), A(N102), A(N103), A(N104), A(N105), A(N106))
RETURN
C
500 CONTINUE
C
CONTINUOUS BEAM ELEMENT
C
Set up the pointers for the continuum beam
C
and call the the CBEAM subroutine
C
A(NE) = A(NFRST)
A(NG) = G Modulus of Rigidity
A(NAREA) = AREA Cross-sectional area (assumed constant)
A(NIXX) = IXX Area moment of inertia
A(NRHO) = RHO Mass density
A(NLM) = [LM]
A(NXYZ) = [XYZ]
A(MATP) = [MATP]
A(NLAST)
C
C put these pointers in a if-then-else loop
C
NFRST=N6
IF (INDX.EQ.1) NFRST=N5
NE = NFRST
NLAST=NUMMAT*ITWO
NG = NE + NLAST
NAREA = NG + NLAST
N101 = NAREA + NLAST
N102 = N101 + NLAST
N104 = N102 + NLAST
N105 = N104 + NUME
NLAST = N106
C
IF (INDX.EQ.1) THEN
   IF (NLAST.GT.MTOT) CALL ERROR(NLAST-MTOT,3)
ELSE IF (NLAST.GT.MTOT) CALL ERROR(NLAST-MTOT,4)
ENDIF
C
RETURN
C**********************************************************************
C
C SUBROUTINE FEBEAM
C
C Programmer: Jim Jara-Almonte
C Location: Mechanical Engineering Dept. V.P.1 & S.U.
C Date: Blacksburg, Virginia O.S.: VMS 3.7
C
C DISCLAIMER: This subroutine was developed for research purposes only. It has not been optimized for general use.
C
C SUBROUTINE PURPOSE:
C This subroutine assembles the element mass and stiffness matrices, in global coordinates for the FEM beam element
C
C*****************************************************************************

SUBROUTINE FEBEAM(ID,X,Y,Z,U,MHT,E,AREA,IXX,RHO,SYT,LM,XYZ,MP)
IMPLICIT REAL*8(A-H,O-Z)
REAL*8 1XX
REAL A
COMMON /SOL/ NUMNP,NEO, NAW,NUMEST, MIDEST, MAXEST, MK
COMMON /DIM/ N1,N2,N3,N4,NS,NT,NB,N9,N10,N11,N12,N13,N14,N15
COMMON /ELE/ IND,MPAR(10),NUMEM,MOT1,FIRST,NILAST,ITWO
COMMON /VAR/ NG,MODEX
COMMON /TAPES/ ILMTN,ILOAD,NSTIF,IN,IOUT,IDEB,IVB,IGEOM,IEIGV
COMMON A(1)

DIMENSION X(1),Y(1),Z(1),ID(3),E(1),AREA(1),LM(6,1),
IXX(6,1),MATP(1,1),MHT(1)
DIMENSION S(21),ST(6),D(3),SYT(1),IXX(1),RHO(1)

EQUIVALENCE (MPAR(1),MPAR1),(MPAR(2),NUME),(MPAR(3),NUMMAT)
SQR(1)=DSQR1(X)
LMT=6

GO TO (300,610),IND

300 WRITE (IOUT,2000) NPARI,NUME ! Read material property sets
IF (NUMMAT.EQ.0) NUMMAT=1
WRITE (IOUT,2010) NUMMAT
WRITE (IOUT,2020)
DO 10 I=1,NUMMAT
C READ (IN,* ) N,E(N),AREA(N),IXX(N),RHO(N),SYT(N)
10 WRITE (IOUT,2030) N,E(N),AREA(N),IXX(N),SYT(N),RHO(N)

WRITE (IOUT,2040)

WRITE (IEGOM) NPARI,NUME
100 READ (IN,1020) M,II,JJ,MTP,YG
IF (NBI.EQ.0) XG=1
120 IF(M.EQ.N) GO TO 200
I=II
J=JJ
MTP=MTP
NY=M
200 XXY(I,N)=X(I)
XY(I,N)=X(I)
XY(I,J)=X(J)
XY(J,N)=X(J)
XY(J,J)=Z(J)
C WRITE (IEGOM) N,I,J ! Save the geometry
MATP(N)=MTP
DO 390 L=1,6
390 LM(L,N)=0
DO 400 L=1,3
LM(L,N)=ID(L,1)
LM(L+3,N)=ID(L,J)
400 CONTINUE
CALL COLHT (MTP,N,LM(1,N)) ! Update column heights
WRITE (IOUT,2050) N,I,J,MTP
3 FORMAT(215,FS.3)
IF (N.EQ.NUME) RETURN
N=N+1
I=I+KXX
J=J+KXX
IF(N.EQ.M) GO TO 100
GO TO 120

610 CONTINUE
C Initialize the pointer for the mass matrix if needed
IF (IVB.GT.0) N7 = N7 + NAW*ITWO
C
P=0.
DO 500 N=1,NUME
MTP=MTP
XLZ=0.
DO 505 L=1,3
O(L)=XY(L,N) - XYZ(L+3,N)
505 XLZ=XLZ+O(L)
XLZ=SORTXLZ
IF (XLZ.LT.1.E-27) THEN

WRITE (IOUT,2050) N,I,J,MTP
WRITE (6,6)
FORMAT (1X,6H10,1X,1H0,1X,1HYOU HAVE A ZERO LENGTH ELEMENT)
END IF
P=PI*XL*AREA(MTYPE)
Compute the element stiffness matrix in global coord.
S(20)=C
S(21)=S
S(21)=O(D(2)/XL)
S(21)=O(D(2)/XL)
C=S(20)*S(21)
S2=S(21)*S(21)
AE=E(MTYPE)/XL*AREA(MTYPE)
E12=2.*E(MTYPE)/XL*IXX(MTYPE)
E16=3.*E12/1XL
E12=2.*E16/1XL
S( 1) = AE*C2 + E112*S2
S( 2) = S(20)*S(21)+*(AE - E12)
S( 3) = -E16*S(21)
S( 4) = -S(1)
S( 5) = -S(2)
S( 6) = S(3)
S( 7) = AE*S2 + E112*C2
S( 8) = E16*S(20)
S( 9) = -S(2)
S(10) = -S(7)
S(11) = S(8)
S(12) = 2*E12
S(13) = -S(3)
S(14) = -S(6)
S(15) = E12
S(16) = S(1)
S(17) = S(2)
S(18) = -S(3)
S(19) = S(7)
C CALL AD08AN (A(N7),A(N2),S,LM(1,N),ND)
C E12=SIN
E12=Sin(angle)
E16=COS(angle)
S(20) = -S(8)
S(21) = S(12)
C CALL AD08AN (A(N3),A(N2),S,LM(1,N),ND)
C If a dynamic analysis is being performed then go ahead and
assemble the consistent mass matrix.
IF (IV8.LT.1) GOTO 500
C IVB = 0 static analysis
C IVB = 1 dynamic analysis
C E12 = SIN
C E16 = COS
AE = AREA(MTYPE)*RHO(MTYPE)*XL
S( 1) = AE*(C2/3. + 0.3714285714285714*P2)
S( 2) = AE*0.3095238095238095*P2
S( 3) = -AE*XLE12*0.0523809523809523
S( 4) = AE*(C2/6. + 0.1285714285714285*P2)
S( 5) = -S(2)
S( 6) = AE*XLE16*0.0523809523809523
S( 7) = AE*(0.3714285714285714*C2 + S2/3.)
S( 8) = AE*XLE16*0.0523809523809523
S( 9) = -S(2)
S(10) = AE*(0.1285714285714285*C2 + S2/5.)
S(11) = AE*XLE16*0.0309523809523809
S(12) = AE*XLE16*105.
S(13) = S(6)
S(14) = -S(11)
S(15) = AE*XLEL/XL/140.
S(16) = AE*(C2/3. + 0.3714285714285714*P2)
S(17) = S(2)
S(18) = -S(3)
S(19) = S(7)
S(20) = -S(8)
S(21) = S(12)
C CALL AD08AN (A(N7),A(N2),S,LM(1,N),ND)
C 500 CONTINUE
WRITE (JOUT,2045) P
RETURN
C 1000 FORMAT (15,5F10.0)
1010 FORMAT(2F10.0)
1020 FORMAT(5(5))
2000 FORMAT (36H ELEMENT DEFINITION,,
1 1 ELEMENT TYPE = . . . . . . . . . ( NPARI1 ) = ',15/,
2 2 ' EQ.2. FINITE ELEMENT METHOD',/,
3 3 ' B-E-A-M ELEMENTS',/,
5 5 ' NUMBER OF ELEMENTS . . . . . . ( NPARI2 ) = ',
6 6 '15/)
2010 FORMAT (42H MATERIAL DEFINITION //,
1 1 ' NUMBER OF DIFFERENT SETS OF MATERIAL',/,
2 2 ' AND CROSS-SECTIONAL CONSTANTS . ( NPARI3 ) = ',
SUBROUTINE CREAM

COMMON A(1)

C COMMON A(1) external arrays
DIGMIZATION X(1),Y(1),Z(1),ID(3,1),E(1),AREA(1),LM(6,1),
1XY2(6,1),MATP(1),U(1),MHT(1),G(6,1),IXX(1),RHO(1)

C local arrays
DIGMIZATION S(21),D(3),AK(2,2),BK(0:4),0:4)
C EQUIVALENCE (NPAPR1),NPAPR1),(NPAPR2),NUME),(NPAPR3),NUMMAT)

C END

C**** SAVE ELEMENT INFORMATION *****

C C

C COMMON /SOI/, NUMMP, NEQ, NSH, NUMEST, MIDEST, MAXEST, MK,
C COMMON /DIM/ N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
C COMMON /EL/ INO,NPNP(10),NUMEG,MIDGET,NFIRST,NLAST,ITRQ
C COMMON /VAR/ NG,MODEX
C COMMON /TAPES/ IELMAT, ILOAD, NSTIF, INI, IOUT, IDEB, IVB, IGEOM, IIEGV

C ***/
**SUBROUTINE FALS**

**SUBROUTINE FALS** (A,B,F,ROOT, TOL, ITLM, ITLMES)

**EXTERNAL**

**REAL** F, MIDPOINT

**COMMON** / CONPAR/ NFEI, NCEI, NPOL, IFPOL, F1, F2, STP DATA EPSILON, ONE/1.0-16.1/0/ **

**C**

check the tolerance:

IF (TOL.EQ.EPSILON) THEN
MESSAGE = 'FAL-Tolerance exceeds the machine epsilon.'
ITEM = 3
END IF

**C**

check the interval

IF (NOT (A.LT.B)) THEN
IF (A.GT.B) THEN
MIDPOINT = A
A = B
B = MIDPOINT
ELSE
IF (A.EQ.B) THEN
MESSAGE = 'FAL-The interval has zero length'
ITEM = 5
**C**

END

**C**

******************************************************************************

**DISCLAIMER:** This subroutine was developed for research purposes only. It has not been optimized for general use.

**SUBROUTINE PURPOSE:**

This subroutine finds an approximation to a zero of the function F provided the interval [A,B] contains a root. The value of TOL is the tolerance with which the root is to be approximated.

**EXPECTED DIFFICULTIES:**

It is possible the approximated root lies outside the interval [A,B]. In this case subroutine HALF is called.
RETURN
END IF
RETURN
C calculate the maximum number of iterations allowed
C this calculation is based on a pure bisection method.
MAXIT=JNINT((LOG10(G.0)-LOG10(TOL))/3.31029956)
C make sure that there is a root in the interval
FA=F(A)
FB=F(B)
IF (DSIGN(ONE,FA)*DSIGN(ONE,FB).LT.0.) THEN
  MESSAGE="FAL-There is no root in the specified interval"'
  ITEM=2
  RETURN
END IF
C begin the regula falsi method
I=1
C regula falsi iteration
C 1 OLDX=VN
C check to make sure that FA and FB are not zero
IF (FA.EQ.0.) THEN
  ROOT=A
  ITEM=6
  MESSAGE="FAL-Root was found exactly F(x)=0''
  RETURN
ELSE
  IF (FB.EQ.0.00) THEN
    ROOT=B
    ITEM=6
    MESSAGE="FAL-Root was found exactly F(x)=0''
    RETURN
  END IF
END IF
DENOM = 1.00 - FB/FA
IF (DENOM.EQ.0.00) THEN
  CALL HALF (A,B,F,ROOT,TOL,ITER,MESSAGE)
  RETURN
END IF
VN = A + (B - A)/DENOM
C make sure VN lies within A and B:
IF ((VN.LT.A).OR.(VN.GT.B)) THEN
  CALL HALF (A,B,F,ROOT,TOL,ITER,MESSAGE)
  RETURN
ENDIF
ENDIF
FN=F(N)
IF (IFPDL) RETURN
C see if iterations have been exceeded:
IF (I.LT.MAXIT) THEN
  IF (DSIGN(VN-OLDVN).LT.0.) THEN
    ITEM=1
    MESSAGE="FAL-The tolerance test was met with ROOT.'
    ROOT=VN
    RETURN
  END IF
C find the new bounds on the roots:
IF (DSIGN(ONE,FA)*DSIGN(ONE,FN).LT.0.) THEN
  S=VN
  FN=FA
ELSE
  A=VN
  FA=FN
ENDIF
I=I+1
GOTO 1
ELSE
  ITEM=2
  ROOT=VN
  MESSAGE="FAL-The number of iterations was exceeded.'
  RETURN
ENDIF
C******************************************************************************
C SUBROUTINE HALF

C******************************************************************************
C Programmer: Jim Jara-Almonte
C Location: Mechanical Engineering Dept. (V. 3.5-62)
C Date: May 1984
C
C DISCLAIMER: This subroutine was developed for research purposes only. It has not been optimized for general use.
C
C SUBROUTINE PURPOSE:
This subroutine finds an approximation to a zero of

function \( F \) provided the interval \([A,B]\) contains a root.

The value of TOL is the tolerance with which the root

is to be approximated

SUBROUTINE HALFW \((A,B,F,\text{ROOT},\text{TOL},\text{ITEM},\text{MESSAGE})\)

IMPLICIT REAL*8(A-H,O-Z)
CHARACTER*(*) \text{MESSAGE}
REAL*8 \text{MIDPOINT}, \text{ONE}
EXTERNAL \text{F}

\text{DATA EPSILON,ONE/0.10-16,1.00/}

C check the tolerance:
IF (TOL.LT.EPSILON) THEN
\text{MESSAGE}='HAL-Tolerance exceeds the machine epsilon.'
\text{ITEM}=3
\text{RETURN}
END IF

C check the interval
IF (.NOT.(A.LT.B)) THEN
\text{IF} (A.GT.B) THEN
\text{MIDPOINT}=A
\text{A}=B
\text{B}=\text{MIDPOINT}
\text{ELSE}
\text{IF} (A.EQ.B) THEN
\text{MESSAGE}='HAL-The interval has zero length'
\text{ITEM}=5
\text{RETURN}
END IF
\text{END IF}
\text{END IF}

C calculate the maximum number of iterations allowed
\text{MAXITR}=JDNINT((DLOG10(\(B-A\))-DLOG10(TOL))/0.301029996)

C make sure that there is a root in the interval
\text{FA}=F(A)
\text{FB}=F(B)
\text{IF} (SIGN(FA)*SIGN(FB).GT.0.) THEN
\text{MESSAGE}='HAL-There is no root in the specified interval'
\text{ITEM}=2
\text{RETURN}
\text{END IF}

C start the bisection method
**SUBROUTINE GETCOT**

* * *

**DISCLAIMER:** This subroutine was developed for research purposes only. It has not been optimized for general use.

**SUBROUTINE PURPOSE:**
Get the exact element data from tape.

* * *

**SUBROUTINE GETCOT (AA)**

COMMON /EL/ IND,NPAR(10),NUMEG,MTOT,NFIRST,NLAST,ITWO
COMMON /TAPES/ IELMT,ILOAD,NSTIF,INOUT,IDE,IBV,IGEOM,IEIGV
COMMON /SOL/ NUMNP,NEQ,NMK,NUMEST,MODEST,MAEST,MK
DIMENSION AA(I)
READ (IELMT) NUMEST,NPAR,(AA(I),I=1,NUMEST)
RETURN

*NON*X
CALL EIGENR (NEQ,NEQ,A(N5),A(N6),A(N7),A(N8))
WRITE (IOUT,1000)
DO I=N5,N8-2,2
CALL REPORT (A(I),-1,11111111)
END DO
RETURN

300 CONTINUE
NC=NEQ*NEQ*ITWO
N5=N5+NC ! A(N5)=GSS
N7=N7+NC ! A(N7)=Alfa array
N8=N8+NC ! A(N8)=Beta array
N9=N9+NC ! A(N9)=work array

IF (N10.GT.MTOL) STOP 'STOP - no space for calling EIGF in IML:'
CALL CVTSQR (A(N3),A(N5),A(N6),A(N7),A(N8),A(N9),A(N10),1,IER)
WRITE (IOUT,1010) IER
CALL RPRT2 (A(N7),A(N8),A(N9),NEQ)
RETURN

1000 FORMAT (1HI,10X,'EIGENVALUES BY EIGENR ROUTINE',/,'EX','Eig.',
110X,' Lambda ',14X,'Eigenvale',/,'EX',3HNo.)
1010 FORMAT (1HI,10X,'EIGENVALUES BY EIGZS ROUTINE',/,'2X','Returned fr'
1,2X,'EIGF in IML with IER=',I3)
END

**************************************************************************
SUBROUTINE CVTPVAL
**************************************************************************

C Programmer: Jim Jara-Almonte
C Location: Mechanical Engineering Dept. (V. 4.1-45)
C V.P.I & S.U.
C Blacksburg, Virginia O.S.: VMS 4.1
C Date: May 1985

C DISCLAIMER: This subroutine was developed for research purposes
C only. It has not been optimized for general use.
C
C SUBROUTINE PURPOSE:
C Prepare storage allocation and constants for eigenvalue/vector extraction of
C nonlinear eigenvalue problem. Constants include values independent of
C frequency and poles.
C
SUBROUTINE COPVAL

IMPLICIT REAL*8 (A-H,O-Z)
REAL*4 A,TIMEON,TPL,TIM,TIMEIG
COMMON A(1)
COMMON /EL/ IND,INPAR(10),NUMEG,MTOT,IF1ST,IXLST,ITWO
COMMON /IND/ N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
COMMON /SOL/ NUMMP,NEQ,NWK,NKST,MODEST,MXTST,MS
COMMON /TAPES/ IEMNT,IOIPD,STIF,INTP,IDEB,IEB,IEOM,IEIGV
COMMON /CONPAR/ NFL,NCPL,IPOL,FPOL,F1,F2,STP,INTM,WSTP
COMMON /CONPTS/ N101,N102,N103,N104,N105,N106,N107,N108,N109,
  N110,N111,N112,N113,N114
COMMON /SULTIM/ TIMEON,TPL,TIM(10),KEI,TIMEIG(60)

EQUIVALENCE (NUM,INPAR(2)),(NUMMAT,INPAR(3))

NFIRST=N5
IAUX=NUMMAT*ITWO

IF (NFIRST .NE. N5) NE = NFIRST
IAUX = NUMMAT*ITWO

CALL CONDIM (NLST,IAUX)
N114 = N113 + 6*NUME
N6 = N114

IF (N114 .GT. MTOT) STOP 'STOP - Not enough space to set-up values'

CALL COMPUT (A(N1),A(N2),A(N3),A(N4),A(N5),A(N6),A(N7),A(N8),A(N9),A(N10),
  A(N11),A(N12))

TPOL=CRUSEC(TIMEON)
IAUX2=(MTOT-N114)/3

NMAX=IAUX2/3
IAUX=IAUX1+IAUX2/3
IAUX2=IAUX1+IAUX2/3

CALL POLES (A(N1),A(N2),A(N3),A(N4),A(N5),A(N6),A(N7),A(N8),A(N9),A(N10),
  A(N11),A(N12))

TPOL=CRUSEC(TIMEON) - TPOL

IAUX=N114 - N113
NE=N113-1
NG=NUM-1
DO I=1,IAUX
  A(NE+1)=A(NG+1)
END DO

IAUX=N114 + ITWO*NPOL + NPOL - N101
NE=N5-1
NG=N101-1
DO I=1,IAUX
  A(NE+1)=A(NG+1)
END DO

IAUX=NUME*ITWO
CALL CONDIM (NS,IAUX)
N114= N113 + 6*NUME
N6 = N114
N7 = N6 + NPOL
N8 = N7 + NPOL*ITWO
N9 = N8 + NPOL*ITWO
N10 = N9 + NPOL*ITWO
N11 = N10 + NPOL - 1
N12 = N11 + NPOL - 1

IF (N12.GT.MTOT) STOP 'STOP - No space to solve Cont.-FE mats.'
RETURN
END
**SUBROUTINE COMPUT**

Programmer: Jim Jara-Almonte  
Location: Mechanical Engineering Dept.  
V.P.I. & S.U.  
Blackburg, Virginia  
Date: May 1985  
Language: VAX FORTRAN  
(V. 4.1-45)

**SUBROUTINE PURPOSE:**
Compute values independent of frequency.

**SUBROUTINE COMPUT**

```fortran
DO L=1,NCHEL
   MTYPE=MATP(L)
   XL2=0.0
   DO L=1,3
      D(L)=XYZ(L,1)-XYZ(L+3,1)
      XL2=XL2+D(L)**2
   END DO
   XL2=SQRT(XL2)
   CO(1)=D(1)/XL2
   SI(1)=D(2)/XL2
   SI(1)=SI(1)*SI(1)
   C2=CO(1)*CO(1)
   CSI=CO(1)*SI(1)
   MU(1)=RHOM(1)*MTYPE/AREA(MTYPE)
   BF(1)=MU(1)*XL2/2/MTYPE/IXX(MTYPE)
   BF1=BF(1)*PMTYPE/AREA(MTYPE)
   BF2=BF1/2/MTYPE
   IF (G(MTYPE,T)LT.0.0) THEN
      SF(1)=0.0
      TF(1)=0.0
      ELSE
      SF(1)=MUL1(1)*FXL2/MTYPE/AREA(MTYPE)
      TF(1)=SF(1)/XL2*TF1
   END IF
   AXB(1)=XL1**2/ETA(RHOM,MTYPE/MTYPE)
   AXXF(1)=BFX**2/XL(I)
   END DO
END
```

**DISCLAIMER:** This subroutine was developed for research purposes only. It has not been optimized for general use.

---

**SUBROUTINE POLES**

Programmer: Jim Jara-Almonte  
Location: Mechanical Engineering Dept.  
V.P.I. & S.U.  
Blackburg, Virginia  
Date: May 1985  
Language: VAX FORTRAN  
(V. 4.1-45)

**SUBROUTINE PURPOSE:**
Compute the poles.

**SUBROUTINE POLES**

```fortran
DO L=1,NCHEL
   MTYPE=MATP(L)
   XL2=0.0
   DO L=1,3
      D(L)=XYZ(L,1)-XYZ(L+3,1)
      XL2=XL2+D(L)**2
   END DO
   XL2=SQRT(XL2)
   CO(1)=D(1)/XL2
   SI(1)=D(2)/XL2
   SI(1)=SI(1)*SI(1)
   C2=CO(1)*CO(1)
   CSI=CO(1)*SI(1)
   MU(1)=RHOM(1)*MTYPE/AREA(MTYPE)
   BF(1)=MU(1)*XL2/2/MTYPE/IXX(MTYPE)
   BF1=BF(1)*PMTYPE/AREA(MTYPE)
   BF2=BF1/2/MTYPE
   IF (G(MTYPE,T)LT.0.0) THEN
      SF(1)=0.0
      TF(1)=0.0
      ELSE
      SF(1)=MUL1(1)*FXL2/MTYPE/AREA(MTYPE)
      TF(1)=SF(1)/XL2*TF1
   END IF
   AXB(1)=XL1**2/ETA(RHOM,MTYPE/MTYPE)
   AXXF(1)=BFX**2/XL(I)
   END DO
END
```

**DISCLAIMER:** This subroutine was developed for research purposes only. It has not been optimized for general use.

---

**SUBROUTINE TAPES**

Programmer: Jim Jara-Almonte  
Location: Mechanical Engineering Dept.  
V.P.I. & S.U.  
Blackburg, Virginia  
Date: May 1985  
Language: VAX FORTRAN  
(V. 4.1-45)

**SUBROUTINE PURPOSE:**
Compute the poles.

**SUBROUTINE TAPES**

```fortran
DO L=1,NCHEL
   MTYPE=MATP(L)
   XL2=0.0
   DO L=1,3
      D(L)=XYZ(L,1)-XYZ(L+3,1)
      XL2=XL2+D(L)**2
   END DO
   XL2=SQRT(XL2)
   CO(1)=D(1)/XL2
   SI(1)=D(2)/XL2
   SI(1)=SI(1)*SI(1)
   C2=CO(1)*CO(1)
   CSI=CO(1)*SI(1)
   MU(1)=RHOM(1)*MTYPE/AREA(MTYPE)
   BF(1)=MU(1)*XL2/2/MTYPE/IXX(MTYPE)
   BF1=BF(1)*PMTYPE/AREA(MTYPE)
   BF2=BF1/2/MTYPE
   IF (G(MTYPE,T)LT.0.0) THEN
      SF(1)=0.0
      TF(1)=0.0
      ELSE
      SF(1)=MUL1(1)*FXL2/MTYPE/AREA(MTYPE)
      TF(1)=SF(1)/XL2*TF1
   END IF
   AXB(1)=XL1**2/ETA(RHOM,MTYPE/MTYPE)
   AXXF(1)=BFX**2/XL(I)
   END DO
END
```

**DISCLAIMER:** This subroutine was developed for research purposes only. It has not been optimized for general use.
DIMENSION EQ(1), AREA(1), BF(1), SF(1), TF(1),
       F(1), MATP(1), L(1), D(1), SF(1), TF(1)
1, POL(1), MARRY(2,1), IEQ(1), LA(1)

DATA LA /22.3732854020597476, 61.6728228035839444,
1 120.90339309542410, 199.85448126889859, 298.5555352981764666,
1 217.3584337175864099, 555.165247630211639, 713.078918556849986,
3 890.73179728563347, 1088.12366531612428, 1305.255182149771741,
4 1541.125687736510775, 1798.7250422262243, 2075.0842354623918011,
5 2371.1724575041881, 2686.396596348231300, 3022.565347394650497,
6 3377.9121644267834, 3752.91707369328560, 4147.701297462305223/

IEQ(1)=1
DO I=2, NCEL
   IEQ(I)=I
   n this model.
   DO J=-1, 1, -1
      IF ((L(I).EQ.L(J)).AND.(MATP(I).EQ.MATP(J))) IEQ(I)=J
   END DO
END DO

NPOLO=0
WRITE (IOUT,100)

DO I=1, NCEL
   IF (IEQ(I).NE.1) THEN
      DO II=1, NPOLO
         IF (MARRY(I,II).EQ.IEQ(I)) MARRY(II,II)=MARRY(II,II)+1
      END DO
      GOTO 20
   END IF
   BF(I)=BF(I)
   SF(I)=SF(I)
   TF(I)=TF(I)
   X(2)=L(I)**4(1)
   IEL=MATP(I)
   ME=RHO(IEL)*AREA(IEL)
   BF=GSORT(1.0/BF(I))
   AF=GSORT(E(IEL)/RHO(IEL)/X(2))*3.14159265358979323846
   F=BF*PLA(N) + 1.00
   N=N + 1
   OLDF=F
   DO WHILE (OLDF.LT.F2)
      IF (SF(I).GT.0.00) THEN
         IF (DTU2(F)) THEN
            F=F + AF
            END IF
      END IF
      IF (SF(I).LT.0.00) THEN
         IF (OLDF.LT.F2) THEN
            IF (POL(1,1).GT.POL(1,1)) THEN
               IF (R1=DTU2(F)) THEN
                  F=F + AF
               END IF
            END IF
         END IF
      END IF
   END DO
20 DO
POL(K,J)=POL(K,J-1)
END IF
END DO
WRITE (IOUT,110) I,POL(1,I),POL(1,J),IARRAY(1)
END DO
WRITE (IOUT,120) TOL
END

N7=N6+NPOL
CALL AARRAY(AARRAY(N6),AARRAY(N7),IARRAY,POL)
RETURN

100 FORMAT(1H1,25X,'POLERES (Infinity wraps')/
*32X,'(radies/second)/'
128X,'H3,16X,HH,10X,4HPOLE,9X,SHUSER,13X,4HPOLE,8X,
212X,MULTIPLICITY,11X,3HNO.,9X,SHOUND,/
110 FORMAT(1X,36X,'Naturak frequencies of the fixed-fixed continuum'
1X,7X,'elements, computed with an accuracy of',8.3,4H,16H,/,..
25X,4H Based on the natural frequencies of similar elements',7X,TK,
3X,without shear deformation or rotatory inertia (Natural ',7X
4X,'frequency plus one')')
200 FORMAT(1X,'ERROR - Not enough space while computing pole:',I3)
900 WRITE (IOUT,200) NPOL
STOP 'STOP - Not enough space while computing poles'
END

******************************************************************************
**
** SUBROUTINE AARRAY
**
**
** Programmer: Jim Jarre-Almante
** Location: Mechanical Engineering Dept. 
** V.P.I & S.U.
** Blackburg, Virginia
** Date: May 1985
**
** DISCLAIMER: This subroutine was developed for research purposes 
** only. It has not been optimized for general use.
**
** SUBROUTINE PURPOSE:
** Arrange memory locations for poles
**
********************************************************************************
**
** FUNCTION DETU2
**
**
** Programmer: Jim Jarre-Almante
** Location: Mechanical Engineering Dept. 
** V.P.I & S.U.
** Blackburg, Virginia
** Date: May 1985
**
** DISCLAIMER: This function was developed for research purposes 
** only. It has not been optimized for general use.
**
** FUNCTION PURPOSE:
** Compute the determinant of submatrix U2.
**
********************************************************************************
**
** REAL*8 FUNCTION DETU2(W)
** IMPLICIT REAL*8 (A-H,0-Z)
** REAL*8 L1,L2
** REAL*16 QH,OS,QT,T,OC1,OC2,OC3,QQ
** COMM/DETU2/ BF,SE,TF
**
** W2=2W
** B4=BF*W2
** S=SF*W2
** T=TF*W2
** SPT(S+T)/2.0O
** SMT=S-T
** R1=CSRT(B4+SMT*SQRT(4.00))
** L1=CSRT(R1-SPT)
** L2=CSRT(R1+SPT)
** SHL1=CSIN(L1)/L1
** SIL2=CSIN(L2)/L2
** C1=L2*L2+SHL1+L1*L1*SIL2
SUBROUTINE BMTHD

* *

Programmer: Jim Janu-Dallace
Location: Mechanical Engineering Dept.
V. P. I. & S. U.
Blacksburg, Virginia
Date: May 1985

Language: VAX FORTRAN (V. 4.1-45)
OS: VM 4.1

DISCLAIMER: This subroutine was developed for research purposes only. It has not been optimized for general use.

SUBROUTINE PURPOSE:
Extract structure eigenvalues using an Incremental-False-Bisection technique and assuming that the structure eigenvalues are at least 1 rad/sec apart.

SUBROUTINE BMTHD
IMPLICIT REAL*8 (A-H,O-Z)
CHARACTER*40 MESS
REAL*4 TIME,TPOL,TIM,TIMEIG,TSTART
EXTERNAL FON
COMMON /DEBUG/ ID1,102,103,104,105,106,107
COMMON /TAPER/ IELMT,ILOAD,NSTIF,IIN,IOUT,IDEF,IBV,IGEOM,IEIGV
COMMON /CONPAR/ NFEL,NCOL,NNPOL,IPPOL,PMIN,PMAX,STEP,INW,NST
COMMON /SOLINF/ TIMEON,TPOL,TIM,TIME,NEIG,TIMEIG(NEIG)
NEIG=0
REWIND (1E19)
DF=STEP/3.00
F1=PMIN
10 DF=FCN(F1)
IF (IPPOL) THEN
F1=F1+1.00
IFPOL=0
GOTO 10
ELSE
IF (DF1.EQ.0.00) THEN
CALL DETERM (F1,1,1)
CALL REPORT (F1,ITER,MES)
IF (ID1) WRITE (12,1000)
TIMEIG(NEIG)=CPUSEC(TIMEON)
F1=F1+1.00
GOTO 10
END IF
END IF
GOTO 20
ELSE
IF (DF1.EQ.0.00) THEN
CALL DETERM (F1,1,1)
CALL REPORT (F1,ITER,MES)
IF (ID1) WRITE (12,1000)
TIMEIG(NEIG)=CPUSEC(TIMEON)
F1=F1+1.00
DF1=FCN(F1)
F2=F1+DF
GOTO 20
END IF
END IF

DO WHILE (F2.LT.PMAX)
20 DF2=FCN(F2)
IF (IPPOL) THEN
F2=F2+1.00
IFPOL=0
GOTO 20
ELSE
IF (DF2.EQ.0.00) THEN
CALL DETERM (F2,1,1)
CALL REPORT (F2,ITER,MES)
IF (ID1) WRITE (12,1000)
TIMEIG(NEIG)=CPUSEC(TIMEON)
F1=F2+1.00
DF1=FCN(F1)
F2=F1+DF
GOTO 20
END IF
END IF

IF (D1.EQ.0.00,DF1)**(D1.EQ.0.00,DF2).LT.0.00) THEN
A=F1
B=F2
END IF
CALL FALSI (A,B,FON,EIG,1.E-16,ITER,MES)

IF (FPOL) THEN
  FPOL=0
  GOTO 400
END IF

CALL DETERM (EIG,B,1)

CALL REPORT (EIG,ITER,MES)

IF (ID1) WRITE (12,1000)

TIME=(EIG)=CPUSC(1)(TIMEON)

F2=EIG+1.0000

DF=FPOL(F2)

END IF

400 F1=F2

DF1=DF

F2=F1+DF

900 END DO

RETURN

1000 FORMAT (1H1)
END

******************************************************************************

SUBROUTINE RPR2
******************************************************************************

C

C Programmed: Jim Jara-Almonte
C Location: Mechanical Engineering Dept. (V 4.1-45)
C Blackburg, Virginia
C Date: May 1985

C DISCLAIMER: This subroutine was developed for research purposes only. It has not been optimized for general use.

C SUBROUTINE PURPOSE:
C Report eigenvalues obtained from IMSL routine.

******************************************************************************

SUBROUTINE RPR2 (ALFA,BETA,WK,N)
COMPLEX*16 AUXING,ALFA(N)
REAL*8 AUX,BETA(N),WK(N)
COMMON /SOLINV/ TIMEON,FPOL,TIM(10),IROOT,TIMEIG(60)
COMMON /TAPES/ IELMTN,ICLOAD,NSTIF,IN,IOUT,IOEB,IVS,ESEM,EIVG

WRITE (IOUT,999) WK(1)

DO I=1,N

IF (BETA(I).EQ.0.) THEN

WK(I)=1.0D6
ELSE

WK(I)=REAL(ALFA(I))/BETA(I)
END IF
END DO

DO I=1,N

WORK(I)=IROOT(I)

IF (WK(I).LT.1.0D6) THEN

WRITE (IOUT,1010) IROOT
ELSE

IF (DIMAG(ALFA(I)).EQ.0.0.D0) THEN

WRITE (IOUT,1000) IROOT, WK(I), DSORT(DABS(WK(I)))
ELSE

WRITE (IOUT,1000) IROOT, WK(I), DSORT(DABS(WK(I)))
END IF
END IF
END DO

END

999 FORMAT (2X,'Returned with a performance index=',F10.2,/,6X,'1.E6',/,'10X','10X','14X','14X','14X','14X','14X','14X','14X')
1000 FORMAT (6X,I3,8X,I3,8X,8X,F12.5)
1020 FORMAT (6X,I3,20X,F12.5,'+','F12.5',')'
1010 FORMAT (6X,I3,12X,'infinite',13X,'infinite')
EN

C

C***************************************************************************

**

C***************************************************************************

C SUBROUTINE REPORT

C

C Program: Jim Jara-Almonte
C Location: Mechanical Engineering Dept.
C V.P.I & S.U. Blacksburg, Virginia
C Date: May 1985
C Language: VAX FORTRAN
C Version: V.4.1-45
C O.S.: VMS 4.1
C

C DISCLAIMER: This subroutine was developed for research purposes only. It has not been optimized for general use.
C

C SUBROUTINE PURPOSE:
C Report eigenvalues from non-IMSL extraction routines.
C

C***************************************************************************

C SUBROUTINE REPORT (EIG,ITER,MES)
C REAL*8 EIG,F1,F2,STP,WPST
C CHARACTER(*) MES
C COMMON /SOL,TIM/ TIMEON,TIM,TIME(10),IROOT,TIME(EIG)
C COMMON /COMPAR/ NTEL,NCOL,IPFOL,F1,F2,F3,STP,INTWSTP
C COMMON /TAPES/ IELMT,ILOAD,NSTIF,TIM,IOUT,IDEB,IV1,IGEN,IEIGV
C IROOT-IROOT+1
C END IF
C IF (ITER.GE.0) THEN
C IF (INTWSTP) THEN
C IF (.NOT.(IROOT-1).LT.1) WRITE (IOUT,2000)
C CALL WITWIL (EIGPSTP,NEIG)
C WRITE (IOUT,2020) NSTF,EIG,IPFOL
C ELSE
C IF (.NOT.(IROOT-1).LT.1) WRITE (IOUT,2010)
C WRITE (IOUT,2030) IROOT,EIG
C END IF
C ELSE
C WRITE (IOUT,1000) IROOT,EIG,DSORT(EIG)
C END IF
C RETURN
C
C 1000 FORMAT (6X,13,8X,E15.8,8X,F12.5)
C 2000 FORMAT (1H1,41X'ZPMTIO RESULTS'/
C 1' 'Wittrick Computed', 'Eigenvalue', 'Message',
C 2' 'Wittrick Computed', 'Eig', 'Message',
C 3' 'IROOT No.', )

END

C***************************************************************************

C***************************************************************************

C SUBROUTINE WITWIL (W,NEIG)
C IMPLICIT REAL*8 (A-H,O-Z)
C REAL*8 A
C COMMON (A)
C COMMON /DUM/ N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
C COMMON /COMPAR/ NTEL,NCOL,IPFOL,F1,F2,F3,STP,INTWSTP
C COMMON /SOL/ NUMNP,NUM,NUMP,NUMR,MIDEST,MIDNST,MIDXST,MM
C COMMON /TAPES/ IELMT,ILOAD,NSTIF,TIM,IOUT,IDEB,IV1,IGEN,IEIGV
C COMMON /CONSRT/ N101,N102,N103,N104,N105,N106,N107,N108,N109,
C 1 N110,N111,N112,N113,N114
C CALL CHKPOL (W,A(N6),A(N7),NTPOL)
C CALL CLEAR (A(N8),NWK)
C CALL CASSEM (W,A(N101),A(N102),A(N103),A(N104),A(N105),
C 1 A(N113))
C IF (IPFOL) THEN
C NSTEIG=1
C END IF
C RETURN
C
C 2010 FORMAT (1H1,41X'ZPMTIO RESULTS'/
C 1' 'Computed', 'Eig', 'Eigenvalue', 'Message',
C 2' 'Eig', 'Message',
C 3' 'No.', )
C 2020 FORMAT (1X,5X,12,4X,12,4X,F21.14,1X,A50)
C 2030 FORMAT (12X,12,4X,F21.14,1X,A50)

END
CALL COLS(N,N(K)), A(N,N(K),NEN)
CALL ONSON(A(N,N(1),NEN),NSON)
NSTEIG = NTPOL + NSON
RETURN
END

SUBROUTINE COLS

Programmer: Jim Jara-Almonte
Language: VAX FORTRAN
V.4.1-45
Location: Mechanical Engineering Dept.
V.P.I. & S.U.
Date: May 1985

SUBROUTINE COLS(A,MAXA,N)
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION A(1),MAXA(1)

40 DO 140 N1=1,N
41 K=MAXA(N)
42 KL=K+1
43 JU=MAXA(N+1)-1
44 NH=JU-KL
45 IF(KH)110,90,50
46 K=N-H
47 IC=0
48 KLT=KU
49 DO 80 J=1,KH
50 IC=IC+1
51 KLT=KLT-1
52 KI=MAXA(K)
53 ND=MAXA(N+1)-KI-1
54 IF(ND)80,80,60
55 K=MINO(IC,ND)
56 C=0.
57 DO 70 L=1,KK
58 C=C*(K1+L)*A(KLT+L)

100 END

SUBROUTINE ONSON

Programmer: Jim Jara-Almonte
Language: VAX FORTRAN
V.4.1-45
Location: Mechanical Engineering Dept.
V.P.I. & S.U.
Date: May 1985

SUBROUTINE ONSON(A,MAXA,NEQ,NSON)
REAL*8 A
DIMENSION A(1),MAXA(1)
NSON=0
DO 1=1,NSON
IF (A(MAXA(1)).LT.0.00) NSON=NSON+1
END DO
RETURN
END

FUNCTION FON

142
SUBROUTINE CHPUL

CALL DETERM(X,DET,0)

CALL CHPOL(X,NT,POL,NTPOL)

IF (NTPOL) THEN
FOC=DET
ELSE
FOC=DET
ENDIF

IF (ID) WRITE (12,100) X,FOC
RETURN

100 FORMAT (1X,F20.14,2X,E40.30)

SUBROUTINE CHPOL

CALL DETERM(W,DET,1)

IF (WTP) THEN
WTP=WTP+1
ELSE
RETURN
ENDIF

SUBROUTINE DETERM

CALL DETERM(W,DET,1)

IF (WTP) THEN
WTP=WTP+1
ELSE
RETURN
ENDIF

SUBROUTINE CHPUL

CALL DETERM(X,DET,0)

CALL CHPOL(X,NT,POL,NTPOL)

IF (NTPOL) THEN
FOC=DET
ELSE
FOC=DET
ENDIF

IF (ID) WRITE (12,100) X,FOC
RETURN

100 FORMAT (1X,F20.14,2X,E40.30)

SUBROUTINE CHPUL

CALL DETERM(X,DET,0)

CALL CHPOL(X,NT,POL,NTPOL)

IF (NTPOL) THEN
FOC=DET
ELSE
FOC=DET
ENDIF

IF (ID) WRITE (12,100) X,FOC
RETURN

100 FORMAT (1X,F20.14,2X,E40.30)
SUBROUTINE CASSEN

DIMENSION K(1), KA(3), KB(10), S(21), SN(1), C(1), S2(1), C2(1), CS(1),
1 LM(5, 1)

W2=NW
DO 1 = 1, NCEL
CALL AXIALK (I, W, A(N11), A(N12), KA)
IF (IFPOL) RETURN
CALL BENDK (I, N2, A(N108), A(N109), A(N110), A(N107), A(N106), KB)
IF (IFPOL) RETURN

S(1) = KA(1)*C2(1) + KB(1)*S2(1)
S(2) = CS(1)*(KA(1) - KB(1))
S(3) = KB(2)*PSN(1)
S(4) = KA(2)*C2(1) + KB(3)*S2(1)
S(5) = CS(1)*(KA(2) - KB(3))
S(6) = KB(4)*PSN(1)
S(7) = KA(1)*S2(1) + KB(1)*C2(1)
S(8) = KB(2)*IC(1)
S(9) = S(5)
S(10) = KB(3)*C2(1) + KA(2)*S2(1)
S(11) = KB(4)*IC(1)
S(12) = KB(5)
S(13) = KB(6)*PSN(1)
S(14) = KB(6)*IC(1)
S(15) = KB(7)
S(16) = S(1)
S(17) = S(2)
S(18) = -S(3)
S(19) = S(7)
S(20) = -S(8)
S(21) = S(12)

CALL ADDAN (A(N9), A(N2), S, LM(1, 1), S)
END DO

CALL INCFM (NMK, W2, A(NB), A(N3), A(N4))

RETURN
END

C SUBROUTINE CASSEN

C Assemble the structure dynamic stiffness matrix.

SUBROUTINE CASSEN (W, SN, C, S2, C2, CS, LM)

INTEGER I, J, K, L, M

REAL*8 A, K, KA, KB

COMMON /IOU/ NUMMP, NEK, NM, NUEST, MIDEST, MAXEST, MK
COMMON /IFPOL/ IFPOL, F1, F2, STP, WSNL, WSTP
COMMON /CPS/ N101, N102, N103, N104, N105, N106, N107, N108, N109,
1 N110, N111, N112, N113, N114
COMMON /SOL/ NUMMP, NEK, NM, NUMEST, MIDEST, MAXEST
COMMON /DIM/ N1, N2, N3, N4, N5, N6, N7, N8, N9, N10, N11, N12, N13, N14, N15

C SUBROUTINE AXIALK

C Assemble the structure dynamic stiffness matrix.
C
C DISCLAIMER: This subroutine was developed for research purposes
C only. It has not been optimized for general use.
C
C SUBROUTINE PURPOSE:
C Find the element axial stiffness matrix for an exact
C element.
C
SUBROUTINE AXIALK ( N, OMEG, AXB, AXF, K)
IMPLICIT REAL*8 (A-H, O-Z)
REAL*8 K, L
COMMON /CENPAR/ NFEI, NC, NPOI, IFPOI, F1, F2, S, T, INTHL, NOSTP
DIMENSION AXB(1), AXF(1), K(3)
C
B=AXB(N)**OMEG
SIN8=GSIN(B)
IF (DSABS(SIN8).LT.1.E-30) THEN
IFPOI=1
RETURN
END IF
FAC=AXF(N)**B/SIN8
K(1)=COS(B)
K(3)=K(1)
K(2)=1.D0
DO I=1,3
K(I)=K(I)*FAC
END DO
RETURN
END

SUBROUTINE BENDK

Programmer: Jim Jara-Almonte
Language: VAX FORTRAN
(V. 4.1-45)
Location: Mechanical Engineering Dept.
V. P. I. & S. U.
Blacksburg, Virginia
Date: May 1985

C
C DISCLAIMER: This subroutine was developed for research purposes
C only. It has not been optimized for general use.
C
C SUBROUTINE PURPOSE:
C Compute the element bending stiffness matrix for an
C
SUBROUTINE BENDK ( N, OMEG, AXB, AXF, K)
IMPLICIT REAL*8 (A-H, O-Z)
REAL*8 K, MU, L1, L2
COMMON /CENPAR/ NFEI, NC, NPOI, IFPOI, F1, F2, S, T, INTHL, NOSTP
DIMENSION SF(1), TF(1), BF(1), MU(1), XL(1), K(10)
A=8F(N)/A(N)/X(N)/X(N)
B=BF(N)**OMEG
T=TF(N)**OMEG
S=SF(N)**OMEG
SPT=S + T
SNT=S - T
RADI=DSQRT (8 + SMT**2)/4.D0
A1=RAD1 - SPT/2.D0
A2=A1 + SPT
L1=DSQRT (A1)
L2=DSQRT (A2)
CLI=2.0D0/RAD1
CH=CSS8(L1)
SN=CS8IN(L1)
CS=CSS8(L2)
SN=CS8IN(L2)
CO=(A2**CH + A1**CS)/CLI
C1=(A2/L1**SH + A1/L2**SN)/CLI
C2=(CH - CS)/CLI
C3=(SH/L1 - SN/L2)/CLI
C
U11=CO - 5*C2
U12=XL(N)**(C1 - SPT**C3)
U12=XL(N)**C3
U12=CO - T**C2
U21=A**C2
U21=AXF(N)**C2 - (S**C1 + (8 + S**5)**C3)
U21=AXF(N)**C2 - T**C3
DU2=U21**211 - U21**221
IF (DSABS(DU2).LT.1.E-30) THEN
IFPOI=1
RETURN
END IF
K(1)=U21**211 - U21**221
K(2)=U21**212 - U21**222
K(3)=U21**222
SUBROUTINE INCDEM

C Programmers: Jim Jara-Almonte
C Location: Mechanical Engineering Dept.
C Language: VAX FORTRAN (V. 4.1-45)
C O.S.: VMS 4.1
C
C DISCLAIMER: This subroutine was developed for research purposes only. It has not been optimized for general use.
C
C SUBROUTINE PURPOSE:
C Include finite element contributions.
C
SUBROUTINE INCDEM (NMWK,2,K,KFE,MFE)
REAL (K,KFE,MFE)
DIMENSION K(1),KFE(1),MFE(1)
DO I=1,NMK
   K(I)=K(I)+KFE(I)*KFE(I)
END DO
RETURN
END

SUBROUTINE CVTSQR

C Programmers: Jim Jara-Almonte
C Location: Mechanical Engineering Dept.
C Language: VAX FORTRAN (V. 4.1-45)
C O.S.: VMS 4.1
C
C DISCREATION: This subroutine was developed for research purposes only. It has not been optimized for general use.
C
C SUBROUTINE PURPOSE:
C Convert a skyline-stored matrix into a square matrix.
C
SUBROUTINE CVTSQR (A,AA,MAYA,NWK,N)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION A(1),AA(N,N),MAYA(1)

DO I=1,N
   KA=MAYA(I)
   KB=MAYA(I+1)-1
   KK=KB-KA
   AA(I,I)=A(KA)
   J=I-KK
   IF (JDST) THEN
      AA(J,J)=0.00
   END DO
   AA(J,I+1)=AA(I,J)
   J=I+1
   DO I=1,N
      AA(J,I)=AA(I,J)
   END DO
RETURN
END
C Date: May 1985

C DISCLAIMER: This function was developed for research purposes only. It has not been optimized for general use.

C FUNCTION PURPOSE:
This function computes the determinant of a dynamic stiffness matrix with logarithmic compression.

REAL*8 FUNCTION DETERM(N,K,JP)
REAL*8 K
REAL*16 DET,ABSDET
DIMENSION K(N,N),JP(N-1)

CALL UPPDET(N,IP,NPD,JP)

IF (NPD) THEN
   DET=1.00
   RETURN
END IF

DET=DETQDEXD(K(I,1))
END DO
IF (IP) DET=DET

ABSDET=QABS(DET)
IF (ABSDET.GT.2.18706129) THEN
   DET=DETEQ(EXP(DET))
ELSE
   DET=DETEQ(QLOG(ABSDET))
END IF
RETURN

C*************************************************************************

C SUBROUTINE UPPDET

C DISCLAIMER: This subroutine was developed for research purposes only. It has not been optimized for general use.

C SUBROUTINE PURPOSE:
Convert a skyline-stored matrix into a diagonal storage scheme.

SUBROUTINE UPPDET(A,AA,DD,MAXA,N,M,N)
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION A(1),AA(N,N),DD(1),MAXA(1)

DO I=1,N
   KA=MAXA(I)
   KB=MAXA(I+1)-1
   KD=KD-NC
   DO J=I-1,KD-1
      KA=KA
      AA(J,I)=KA
      END DO
   DO J=K,1,-1
      AA(J,I)=0.000
      END DO
   IA=0
   DO I=1,N
      IA=IA+1
      DO (J,IA)=AA(J,I)
      END DO
   RETURN
END

C*************************************************************************

C Programmer: Jim Jara-Almonte
C Location: Mechanical Engineering Dept.
C V.P.I & S.U.
C Blacksburg, Virginia
C Date: May 1985

C Language: VAX FORTRAN (V. 4.1-45)
C Date: May 1985

C

C DISCLAIMER: This subroutine was developed for research purposes only. It has not been optimized for general use.

C SUBROUTINE PURPOSE:

C Upper triangularize a matrix with full pivoting for a determinant calculation.

C******************************************************************************

SUBROUTINE UPPTR (N,IP,NPD,A,JP)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION A(N,N),JP(N-1)

NPD=0
IP=0

C This loop upper triangularizes matrix A:
DO 100 I=1,N-1
C Determine if pivoting is necessary:
C**II=0
JP(I)=0
AUX=0.00
DO J=1,N
DO K=1,N
IF (AUX.LT.DABS(A(J,K))) THEN
AUX=DABS(A(J,K))
II=J
JJ=K
END IF
END DO
END DO
IF (II.EQ.0) THEN
NPD=1
RETURN
END IF
END IF

C now exchange the rows or columns if pivoting necessary:
C**II.NE.1 THEN
IF (II.NE.1) THEN
I=IP+1
DO J=1,N
AUX=A(I,K)
A(I,K)=A(II,K)
A(II,K)=AUX
END DO
END IF

C******************************************************************************

SUBROUTINE SOLVEC

C Programmed: Jim Jara-Almonte
C Location: Mechanical Engineering Dept. V.4.1-45
C Date: May 1985
C

C DISCLAIMER: This subroutine was developed for research purposes only. It has not been optimized for general use.

C SUBROUTINE PURPOSE:

C Solve for the eigenvector.

C******************************************************************************

SUBROUTINE SOLVEC (EIG,N,JA,JP,X)
SUBROUTINE PLOTB(X,Y,FELM,Celm,XV,IO,MIN,HED,SJ,CO)

DIMENSION FELM(2,1),Celm(2,1),ID(3,1),X(1),Y(1),XV(1),HED(1)
DIMENSION S(1),CO(1),JS(15),Z(4),ZA(2),D(6),IPRESS(4)
DATA IPRESS / 'Pre',ss '<', 'Rctu', 'Rt>', '/

REWIND (IEGM)
REWIND (IEIGV)

READ (IGEM) NUMP, XKMIN, XMN, YM, XYNAX
READ (IGEM) YM, XYNAX - YM

FACTOR = DX
FACTOR = DY ! Find the global scaling
FACTR = 1.0 * FACTR
FACTR = 1.0 * FACTR

! Compute the actual max.
YM = YM - FACTR
YMN = YM - FACTR
XMAX = XMAX + FACTR
XMIN = XMIN + FACTR

READ (IGEM) (X(I), Y(I), I = 1, NUMP)
READ (IGEM) IELTYP, NUMP
IF (IELTYP = 2) THEN
    NUME = NUMF
    NUMF = 0
DO 12 I=1,NUMF
  READ (IGEOM) Iaux,FELM(1),Iaux,FELM(2),Iaux)
10 END DO
  READ (IGEOM) IELTP,NUMFC
  IF (IELTP,NE.3) STOP 'STOP - Did not find Cont. elements as expec
  led in PLOT'
12 DO 15 I=1,NUMFC
  READ (IGEOM) Iaux,Celm(1),Iaux,Celm(2),Iaux)
15 END DO
  CALL INITT (120)
  CALL DATE (TODAY)

DO IEIG=1,NEIG
  READ (IEIG) EIG,(XY(I),I=1,NEQ)
  IF (IEIG) WRITE (17,700) IEIG,EIG,JXY(I),I=1,NEQ)
  EIG2=EIG*IEIG
  CALL PGRAW
  CALL MOVARS (275,740)
  CALL TIME (HOUR)
  CALL ADJUST (52, MEDI
  CALL NEWLIN
  CALL AMNODE
  WRITE (6, 210) SGNL(EIG),TODAY,HOUR
  CALL TWINDO (300, 975, 50, 700)
  CALL WINDO (SGNL(XMIN),SGNL(XMAX),SGNL(YMIN),SGNL(YMAX))
  C
  Plot the undeformed shape:
  DO I=1,NUMFC
    CALL MOVEA (SGNL(X(FELM(1,I))),SGNL(Y(FELM(1)),I))
    CALL DASHA (SGNL(X(FELM(2,I))),SGNL(Y(FELM(2)),I)),34
  END DO
  DO I=1,NUMFC
    CALL MOVEA (SGNL(X(CELM(1),I))),SGNL(Y(CELM(1),I)))
    CALL DASHA (SGNL(X(CELM(2),I)),SGNL(Y(CELM(2)),I)),34
  END DO
  C
  Plot the eigenvector:
  IFENT=0
  C
  Plot the continuum mode shape:
  DO N=1,NUMFC
    I=CELMI(N)
    J=CELMI(2,N)
    K=0
    DO M=1,2
      
      IAU=CELMI(M,N)
      DO MM=1,3
        K=K+1
        NE=ID(MM,J)
        IF (NE.GT.0) THEN
          C(K)=XV(NE)
        ELSE
          C(K)=0.
        END IF
      END DO
      S=SI(N)
      C=CO(N)
      CALL QNSTVC (N,IEIG,D,S,C,Z,ZA)
      CALL ADDVEC (O,X(I),Y(J),D(1),D(2))
      CALL PLOTG (N,X(I),Y(J),C,S,IEIG,EIG,EIG2,
      A(N106),A(N111),A(N112),Z,ZA)
    END DO
    C
    Plot the finite element mode shape:
    DO N=1,NUMFC
      I=FELM(1,N)
      J=FELM(2,N)
      K=0
      DO M=1,2
        IM=IM+1
        NE=ID(MM,J)
        IF (NE.GT.0) THEN
          C(K)=XV(NE)*FACTR
        ELSE
          C(K)=0.0
        END IF
      END DO
      CALL ADDVEC (O,X(I),Y(J),D(1),D(2))
      CALL ADDVEC (0,X(J),Y(J),D(3),D(4))
    END DO
    
    END DO
    CALL EXPVEC (EIG,A(N11),A(N12),A(N13),A(N14),A(N15),
    A(N15),A(N16),A(N17),A(N18),A(N19),A(N20),A(N21),A(N22))
    CALL MOVARS (1,730)
    CALL ADJUST (16,IMPRESS)
    CALL TINPUT (IYACUC)
  END DO
  RETURN
**SUBROUTINE INSTVC**

Programmer: Jim Jara-Almonte
Location: Mechanical Engineering Dept.
Language: VAX FORTRAN (V. 4.1-45)
V.P.I. & S.U.
Blacksburg, Virginia
OS: VMS 4.1
Date: May 1985

Disclaimer: This subroutine was developed for research purposes only. It has not been optimized for general use.

SUBROUTINE PURPOSE:
Find the initial state vector at the beginning of an exact element.

SUBROUTINE INSTVC (IEL,EIG,D,SN,CO,ZA)
IMPLICIT REAL*8 (A-H,O-Z)
REAL*4 A
COMMON A(1)
COMMON /DEBUG/: ID1, ID2, ID3, ID4, ID5, ID6, ID7
DIMENSION U(4,4), U(2,2), D(6), Z(4), V(2), C(6)
DIMENSION U(2,2), U(2,2), U(2,2), T(2), R(2), V(2), C(2)

CALL QTRAMM (1.00, IEL, EIG, A(N108), A(N109), A(N110), A(N107), 1.00, A(N106), A(N111), A(N112), U, V, U, Z)

IF (ID7) WRITE (14,1000) IEL, EIG, (UA(J,1),I=1,4), J=1,4)
IF (ID7) WRITE (14,1010) (UA(J,1),I=1,2), J=1,2)

Get the bending initial state vector:
C(2) = C(5)

C(1) = C(4)

C(3) = C(6)

DO I=1,4,3
C(I) -O(I)CO + D(I+1)PSN
C(I+1)-O(I+1)CO - D(I)PSN
C(I+2) = D(I+2)

END DO

IF (ID7) WRITE (14,1020) ((C(I),I=1,C(I+1))), I=1,5,2)

U1(1,1)=UB1(1,1)
U1(1,2)=UB1(2,1)
U1(2,1)=UB2(1,2)
U2(1,1)=UB1(1,3)
U2(1,2)=UB1(1,4)
U2(2,1)=UB2(2,3)
U2(2,2)=UB2(2,4)

IF (ID7) THEN
WRITE (14,1030) 'U2', ((U2(J,1),I=1,2), J=1,2)
WRITE (14,1050) ((U2(J,1),I=1,2), J=1,2)
WRITE (14,1100) U2(1,1)=U2(1,1), U2(1,2)=U2(1,2), U2(2,1)=U2(2,1), U2(2,2)=U2(2,2),
1 Z(1)=C(2)
2 Z(2)=C(3)
3 Z(1)=C(2)
4 Z(3)=C(3)

END IF

V = Z(3) = U 2 [C(5)] - U 1 [C(3)]

CALL MATVEC (2, U1, C(2), T)
C(5)=C(5)-T(1)
C(6)=C(6)-T(2)

CALL MATVEC (2, U2, C(5), Z(3))
Z(1)=C(2)
Z(2)=C(3)

Find the axial initial state vector:
ZA(1)= C(1)
ZA(2)= C(4) - UA(1,1)*C(1))/UA(1,2)

Find the residual for the computed initial state vector:
CALL MATVEC (2, U1, Z(1), C(6))
CALL MATVEC (2, U2, Z(3), T)
R(2)=CV1(2)+I(2)-C(5)
RESNOM + DABS(R(1)) + DABS(R(2))
IF (IDT) THEN
WRITE (14,1060) ((I,Z(I)),I=1,4)
WRITE (14,1070) ((I,ZA(I)),I=1,2)
WRITE (14,1080) ((I,R(I)),I=1,2)
ENDIF
IF (RESNOM.LT.1.00) RETURN
MAXITR=10
IF (IDT) WRITE (14,1090) MAXITR
CV2(1)=CV1(1)-C(5)
CV2(2)=CV1(2)-C(5)
DO WHILE (RESNOM.GT.1.00)
DO 200 I=1,2
200 T(I)=R(I)-CV2(I)
CALL MATVEC (2,U21,T,Z(3))
CALL MATVEC (2,U22,Z(3),1)
R(I)=CV1(I)-T(I)-C(5)
R(2)=CV1(2)-T(2)-C(5)
RESNOM = DABS(R(1)) + DABS(R(2))
IF (IDT) WRITE (14,1060) ((I,Z(I)),I=1,4)
IF (IDT) WRITE (14,1080) ((I,R(I)),I=1,2)
MAXITR=MAXITR-1
IF (MAXITR.EQ.0) RETURN
END DO
RETURN

1000 FORMAT ('Initial state vector calculation for element',I3,' at 1.10.4, rads/sec. '/'Bending Transfer Matrix: ',(10X,4(5X,2(E18.10))),)
1010 FORMAT ('Axial transfer matrix: '/(10X,2(5X,E18.10))),
1020 FORMAT ('Local displacements: '/(15X,'C(I,I) = ',E20.14),1100)
1030 FORMAT ('Determinant of U2= ',E20.14),
1040 FORMAT ('U2 Inverse matrix: '/(10X,2(5X,E20.14))),
1050 FORMAT ('Initial bending state vector: '/(15X,'Z(I,I) = ',1E20.14)),
1060 FORMAT ('Initial axial state vector: '/(15X,'ZA(I,I) = ',1E20.14)),
1070 FORMAT ('The residual vector for this element is: '/(15X,1'R(I,I) = ',E20.14)),
1090 FORMAT ('ITERATE FOR INITIAL STATE VECTOR',20('"'),1,15X,'Maximum number of iterations = ',I3),
1100 FORMAT ('(U2 Inverse matrix: '/(10X,2(5X,E20.14))),

C
C SUBROUTINE PLOT

C Programmer: Jim Jara-Alcante
C Location: Mechanical Engineering Dept.
C V.P.I & S.U.
C Blacksburg, Virginia
C Date: May 1985
C
C DISCLAIMER: This subroutine was developed for research purposes only. It has not been optimized for general use.
C
C SUBROUTINE PURPOSE:
C Reconstruct eigenvector for a continuum beam element.
C
C************************************************************
SUBROUTINE PLOT (IEL,X1,Y1,Z1,EIG,EIG2,XL,AXB,AXXF,Z,1)
C IMPLICIT REAL*8 (A-N,O-Z)
REAL*4 A
COMMON A(I)
COMMON /DEBUG/ I01,I02,I03,1D4,1D5,1D6,1D7
COMMON /CONT5/ N101,N102,N103,N104,N105,N106,N107,N108,N109,
1 N110,N111,N112,N113,N114
DIMENSION X(1),AXB(I),AXXF(I),Z(1),ZA(1)
DIMENSION UB(4,4),UA(2,2),ZOLD(4),ZWED(4),ZAOLOD(2),ZANEW(2)
C
C Prepare the constants for plotting:
C NFIELD=56
DLXL=IELD/FLOAT(NFIELD)
FACT=1./FLOAT(NFIELD)
AXIAL=AXB(I)/EIG
AXXFAC=1./AXXF(I)
C
C Get the transfer matrix for a small field
CALL QT174M (FACTR,IEL,EIG,A(N108),A(N109),A(N110),A(N117),A(N118),A(N119),A(N110),A(N111),UB,UA,0,ZOLD)
1 IF (IDT) WRITE (15,1001) EIG,IEL
2 IF (IDT) WRITE (16,1030) IEL,EIG,UB(1,1),I=1,4,J=1,4
3 IF (IDT) WRITE (16,1040) (UB(I,J),I=1,2,J=1,2)
C DO 7 I=1,4
5 ZOLD(I)=Z(I)
6 DO 8 J=1,2
7 ZAOLD(I)=ZA(I)
C
C

512
IF (ID6) WRITE (16,1020) (ZOLD(III),III=1,4),(ZA(III),III=1,2)

CML=0.
DO I=1,NFIELD
CML=CML + DL
C
find the axial deformation:
CALL MATVEC (2,UA,ZOLD,ZANEW)
ZOLD(1)= ZANEW(1)
ZOLD(2)= ZANEW(2)
C
find the bending transverse deformation
CALL MATVEC (4,UB,ZOLD,ZANEW)
DO 11 III=1,4
11 ZOLD(III)=ZNEW(III)
C
IF (ID6) WRITE (16,1020) (ZOLD(III),III=1,4),
1
(ZA(III),III=1,2)
ML=ZNEW(1)
U= ZANEW(1)
C
now find the global displacement:
DELY=(ZML - WP*C)
DELX=(UML + WP*C)
XP=X + CML*C
YP=Y + CML*C
C
IF (ID6) WRITE (15,1010) CML,DELX,DELY,XP,YP
CALL ADDVEC (1,XP,YP,DELX,DELY)
END DO
RETURN

2AXIAL DEF.",1X,"AXIAL LD.",/3X,\"(N\")",14X,\"(PHI)\",11X,\"(M)\",14X,\"(V)\",14X,\"(U)\",13X,\"(N)\")
1010 FORMAT (1X,F15.5,4X,2X,E15.9,4X,2X,E15.9))
1020 FORMAT (1X,E15.9,2X))
1030 FORMAT ("1Transfer matrices used for element ",I3 ," at ",1,F10.4 ," rads/sec. ",/," Bending Transfer Matrix:",//,(10X,4(5X,2E18.10)))
1040 FORMAT (//,\" Axial transfer matrix:",(10X,2(5X,E18.10))))
**SUBROUTINE QTRAN**

**Programmer:** Jim Lara-Almonte

**Language:** VAX FORTRAN

**Location:** Mechanical Engineering Dept.

**V.P.I. & S.U.**

**Blacksburg, Virginia**

**Date:** May 1985

**DISCLAIMER:** This subroutine was developed for research purposes only. It has not been optimized for general use.

**SUBROUTINE PURPOSE:**

- Compute the transfer matrix of a continuum beam.

### Subroutine Code:

```fortran
REAL*8 A(N,N),B(M,NN),C(N,NN)
DO 10 I=1,N
  DO 10 J=1,M
    C(I,J)=0.DO
  DO 10 K=1,NN
  10 C(I,J)=C(I,J)+A(I,K)*B(K,J)
RETURN
END
```

**RAD1 = GORT(8 + SMT*SMT/4.00)**

- **A = RAD1 - SPT/2.00**
- **A2 = A1 + SPT**
- **L1 = GORT(A1)**
- **L2 = GORT(A2)**
- **CL1 = OEXTD(2.00)*RAD1**

- **CH = COS(L1)**
- **SH = SIN(L1)**
- **CS = COS(L2)**
- **SN = SIN(L2)**

- **C0 = (A2*A2)*CH + A1*CS)/CLI**
- **C1 = (A2/L1)*SH + A1/L2)*SN)/CLI**
- **C2 = (CH - CS)/CLI**
- **C3 = (SH/L1 - SN/L2)/CLI**

- **U211 = A*C2**
- **U212 = A*FL*FR(- SPC1 + (B + S))SC3**
- **U221 = A*FL*(C1 - T*C3)**
- **U222 = U211**

```fortran
UB(1,1) = ORLEQ(C0 - SPC2)
UB(1,2) = ORLEQ(FL*(C1 - SPT*C3))
UB(1,3) = ORLEQ(U211)
UB(1,4) = ORLEQ(U212)
UB(2,1) = ORLEQ(B/FL*C3)
UB(2,2) = ORLEQ(C0 - T*C3)
UB(2,3) = ORLEQ(U221)
UB(2,4) = UB(1,3)
UB(3,1) = ORLEQ(B/A*C3)
UB(3,2) = ORLEQ(FL/FA*(- SPC1 + (B + T)*C3))
UB(3,3) = UB(2,2)
UB(3,4) = UB(1,2)
UB(4,1) = ORLEQ(B/A*C3)
UB(4,2) = ORLEQ(B/FL*(C1 - SPC3))
UB(4,3) = ORLEQ(B/FL*C3)
UB(4,4) = UB(1,1)
```

**IF (U2FLG) THEN**

- **DEQ = U211*U211 - U212*U221**
- **U21(1,1) = ORLEQ(U222/DEQ)**
- **U21(1,2) = ORLEQ(U212/U221)**
- **U21(2,1) = ORLEQ(U221/U222)**
- **U21(2,2) = ORLEQ(U211/DEQ)**

**END IF**

**C find the axial transfer matrix:**
SUBROUTINE ADDVEC

IA(N1+IFENT)=IPF
A(N12+IFENT)=SNGL(X)
A(N13+IFENT)=SNGL(Y)
A(N14+IFENT)=SNGL(DX)
A(N15+IFENT)=SNGL(DY)
IFENT=IFENT+1
IF (IFENT .GE. MAXENT) STOP 'STOP-Exceeded storage during vector plotting'
RETURN
END

SUBROUTINE EXPVEC

REAL#X,X,DX,DY
REAL#A
COMMON A(I)
COMMON /PLTPAR/ MAXENT,IFENT
COMMON /DIM/ N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
DIMENSION IA(I)
EQUIVALENCE (A(I),IA(I))

IA(N1+IFENT)=IPF
A(N12+IFENT)=SNGL(X)
A(N13+IFENT)=SNGL(Y)
A(N14+IFENT)=SNGL(DX)
A(N15+IFENT)=SNGL(DY)
IFENT=IFENT+1
IF (IFENT .GE. MAXENT) STOP 'STOP-Exceeded storage during vector plotting'
RETURN
END

DISCLAIMER: This subroutine was developed for research purposes only. It has not been optimized for general use.

SUBROUTINE PURPOSE:

Add vector points to plot buffer.

DISCLAIMER: This subroutine was developed for research purposes only. It has not been optimized for general use.

SUBROUTINE EXPVEC (E,IFF,X,Y,DX,DY,FACTR)
REAL#E
COMMON /DEBUG/ ID1,ID2,ID3,ID4,ID5,ID6,ID7
COMMON /PLTPAR/ MENT,IFENT
DIMENSION IFF(I),X(I),Y(I),DX(I),DY(I)

IF (IFF(I)) WRITE (13,1000)
FD=ABS(DX(I))
DO = 1,IFENT
IF (FD.LT.ABS(DX(I))) FD=ABS(DX(I))
IF (FD.LT.ABS(DY(I))) FD=ABS(DY(I))
END DO
FD=FACTR/FD
END IF
IF (IFF(I)) WRITE (13,1010) IFF(I),X(I),Y(I),DX(I),DY(I)
DX(I)=DX(I)*FD
DY(I)=DY(I)*FD
Y(I)=Y(I)+DY(I)
X(I)=X(I)+DX(I)
END IF
CALL MOVEA (X(I),Y(I))
ELSE
CALL MOVEA (X(I),Y(I))
END IF
END DO
RETURN

1000 FORMAT (I1H1,'MODE SHAPE VECTOR AT FREQ. ',F15.8,
**SUBROUTINE PGROW**

CALL NEWPAG
CALL MOVEAS (244, 1)
CALL DRWAAS (244, 779)
CALL DRWAAS (1023, 779)
CALL DRWAAS (244, 1)
CALL MOVEAS (270, 30)
CALL DRGREL (25.0)
CALL DRGREL (-10.5)
CALL MOVE (0, -10)
CALL MOVE (10.5)
CALL MOVE (0, 5)
CALL ANCH 'X'
CALL MOVEAS (270, 30)
CALL DRGREL (0.25)
CALL DRGREL (5, -10)
CALL MOVE (10, -10, 0)
CALL DRGREL (5, 10)
CALL BANKSP ('Y')
CALL MOVEAS (270, 30)
CALL LINET
CALL BANKSP
C CALL ANCHO ('Z')

**SUBROUTINE QINSTVC**

CALL TSEND
RETURN
END

**SUBROUTINE QINSTVC** (IEL, EIG, D, SN, CO, DJ, DZA)

IMPLICIT REAL*16 (A-H, O-Z)
REAL*4 A
REAL*8 EIG, D, SN, CO, DJ, DZA
COMMON A(1)
COMMON /DEBUG/ I01, I02, I03, I04, I05, I06, I07
COMMON /COPSY/ N01, N02, N03, N04, N05, N06, N07, N08, N09, N10, N11, N12, N13, N14
DIMENSION UB(4, 4), UA(2, 2), D(6), DJ(4), DZA(2), C(6), ZA(2)
DIMENSION U2(2, 2), U1(2, 2), T(2, 2), R(2), CV(2), CY(2)

CALL QTRANM2 (1, 0D0, IEL, EIG, A(N108), A(N109), A(N110), A(N110), A(N110), A(N110), A(N110), A(N110), A(N110), A(N110), A(N110), A(N110))

IF (ID) WRITE (14, 1000) IEL, EIG, ((UB(J, 1), I=1, 4), J=1, 4)
IF (ID) WRITE (14, 1010) ((UA(J, 1), I=1, 2), J=1, 2)

Get the bending initial state vector:

\[
\begin{align*}
C(1) &= C(2) + (C(3) + C(4)) \\
C(5) &= A(N108) + A(N110) + A(N112) + U2, U1, 0, 01
\end{align*}
\]

DO I=1, 4, 3

\[
C(1) = 2EXTD(D(I))*CO + D(I+1)*SN \quad U
\]
C(I+1)=C(I)+D(I)*SN
C(I+2)=C(I)+D(I)*SN
END DO
IF (IDT) WRITE (14,1020) ((I,C(I),I=1,1,C(I+1)),I=1,5,2)
U(I,1)=UB(I,1)
U(I,2)=UB(I,2)
U(I,3)=UB(I,3)
U(I,4)=UB(I,4)
U(I,5)=UB(I,5)
IF (IDT) THEN
WRITE (14,1030) 'U2:((U2(J,1),I=1,2),J=1,2)
WRITE (14,1050) ((U2(I,J),I=1,2),J=1,2)
WRITE (14,1100) U2(I,1)*U2(I,2)*U2(I,3),
U2(I,4)*U2(I,5)*U2(I,1),
U2(I,2)*U2(I,3)*U2(I,1),
U2(I,3)*U2(I,4)*U2(I,1),
U2(I,4)*U2(I,5)*U2(I,1),
END IF

CALL QMATVEC (2,U2,C(2),T)
C(5)=C(5)+T(1)
C(6)=C(6)+T(2)
CALL QMATVEC (2,U2,C(5),Z(3))
Z(I)=Z(2)
! Z(I)=W at I-1
Z(2)=Z(3)
! Z(3)=PHI at I-1

C Find the axial initial state vector:
ZA(I)=C(1)
! U at I-1
ZA(2)=C(4)-U(1,1)*C(1)+U(1,2)
! N at I-1

C Find the residual for the computed initial state vector:
CALL QMATVEC (2,U2,Z(1),G(1))
CALL QMATVEC (2,U2,Z(2),G(1))
R(I)=G(1)+T(1)-C(5)
R(2)=G(1)+T(2)-C(6)
RESOM=ABS(R(1)) + ABS(R(2))
IF (IDT) THEN
WRITE (14,1060) ((I,Z(I),I=1,4)
WRITE (14,1070) ((I,ZA(I),I=1,2
WRITE (14,1080) ((I,R(I),I=1,2

END IF
IF (RESOM.LT.1.00) GOTO 900
MAXITR=10
IF (IDT) WRITE (14,1090) MAXITR
CV1(1)=CV1(1)-C(5)
CV1(2)=CV1(2)-C(6)
DO WHILE (RESOM.GT.1.00)
CALL QMATVEC (2,U2,R,T)
DO 200 I=1,2
200 Z(I+2)=Z(I+2)-T(I)
IF (IDT) WRITE (14,1060) ((I,Z(I),I=1,4)
IF (IDT) WRITE (14,1080) ((I,R(I),I=1,2
MAXITR=MAXITR+1
IF (MAXITR.EQ.10) GOTO 900

END DO
900 DO 901 I=1,4
901 DZ(I)=OBSQ(Z(I))
DO 902 I=1,2
902 DZA(I)=OBSQ(ZA(I))
RETURN

1000 FORMAT ('Initial state vector calculation: I3,' at '1,F10.4, 'rads/sec.' //,' Banding Transfer Matrix:/(10X,4(SX,2E18.10))
1010 FORMAT('/','Axial transfer matrix:/(10X,2(SX,2E18.10))')
1020 FORMAT('/','Local displacements:/(15X,'C(','I1,')=',E20.14,
1100,'C(','I1,')=',E20.14)')
1030 FORMAT('/','1X,A2,' matrix:/(10X,2(SX,2E14.14))')
1040 FORMAT('/','1X,A2,' Determinant of U2: E20.14')
1050 FORMAT('/','U2 Inverse Matrix:/(10X,2(SX,2E14.14'))
1060 FORMAT('/','Initial banding state vector:/(15X, Z(',I1,')=',
1070,1E14 ')')
1080 FORMAT('/','Initial axial state vector:/(15X, ZA(',I1,')=',
1090,1E14 ')')
1090 FORMAT('/','The residual vector for this element is:/(15X,
1100,'1W',11,')=',E20.14')
1100 FORMAT('/','1X,20(1X),"ITERATE FOR INITIAL STATE VECTOR",20(1X)
1100,'15X, Maximum number of iterations=',13)

C *********************************************************************
C SUBROUTINE QTRANM2
C
C
C Programmed by: Jim Jara-Almonte
Language: VAX FORTRAN
Location: Mechanical Engineering Dept.
V.P.I & S.U.
Blackburg, Virginia
Date: May 1985

DISCLAIMER: This subroutine was developed for research purposes only. It has not been optimized for general use.

SUBROUTINE PURPOSE:
Find the transfer matrix for a continuum beam in Quad-Precision.

SUBROUTINE QTRANM2 (F,N,W,SF,TF,BF,MU,XLB,AXB,AXSF,UB,UA,UIU2FLG,
1 U21)
IMPLICIT REAL*16 (A-H,O-Z)
REAL*16 F,N,W,SF,TF,BF,MU,XLB,AXB,AXSF
REAL*16 K,L1,L2
DIMENSION SF(1),TF(1),BF(1),MU(1),XLB(1),AXB(1),AXSF(1)
DIMENSION UA(2,2),UB(4,4),U21(2,2)

OMEGA=QEXTD(W)
OMES2=OMES2*OMEN6
FL=QEXTD(L2(N))
CF2=QEXTD(FP)
CF =OMES2/CF2
B=BF(N)*CF/CF2
A=BF(N)/OMES2/QEXTD(MU(N))
T=QEXTD(TF(N))*CF
S=QEXTD(SF(N))*CF
SPT=S + T
SMT=S - T

RAD1=QSQRT(B + SMT*SMT/4.00)
A1=RAD1 - SPT/2.00
A2=A1 + SPT
L1=QSQRT(A1)
L2=QSQRT(A2)
CLI=QEXTD(2.00)*RAD1

CH=QCCOSH(L1)

SH=QSMH(L1)
CS=QCCOS(L2)
SN=QSMN(L2)
C0=(A2*CH + A1*CS)/CLI
C1=(A2/L1*SN + A1/L2*SN)/CLI
C2=(SN - CS)/CLI
C3=(SN/L1 - SN/L2)/CLI
U21 = A*2
U22 = A*FL/FP - S*F1 + (B + SMT)*F3)
U21 = A*FL*(C1 - T*F3)
U22 = U21
UB(1,1) = (C0 - S*F2)
UB(1,2) = (F2*(C1 - S*F3))
UB(1,3) = U21
UB(1,4) = U22
UB(2,1) = (B*F3)
UB(2,2) = (C0 - T*F2)
UB(2,3) = U21
UB(2,4) = U22
UB(3,1) = (B*F3)
UB(3,2) = (F2*(T*F1 + (B + T*F)*F3))
UB(3,3) = U21
UB(3,4) = U22
UB(4,1) = (B*F3)
UB(4,2) = (B*F3)
UB(4,3) = (B*F3)
UB(4,4) = U1(1,1)

IF (IUU2FLG) THEN ! U2 inverse is needed
DET=U211*U211 - U212*U221
U21(1,1) = (U222/DET)
U21(1,2) = (-U212/DET)
U21(2,1) = (-U211/DET)
U21(2,2) = (U211/DET)
END IF

C find the axial transfer matrix:
B = AXB(N)*OMES2*QEXTD(F)
FAC = QEXTD(F)/AXB(N)
UA(1,1) = (QCCOSH(B))
UA(2,2) = (U11)
UA(1,2) = (QSMH(B)*FAC/8)
UA(2,1) = (-B*QSMN(B)/FAC)
RETURN
**SUBROUTINE OMVEC**

<table>
<thead>
<tr>
<th>Subroutine Name:омвец</th>
<th>Language: Fortran (V. 4.1-45)</th>
<th>Date: May 1984</th>
</tr>
</thead>
</table>

**DISCLAIMER:** This subroutine was developed for research purposes only. It has not been optimized for general use.

**SUBROUTINE PURPOSE:** Multiplies a vector by a matrix.

**REAL** A(N,N), B(N), C(N)
**DO** 10 I = 1, N
      C(I) = C(I) + A(I,J) * B(J)
   10 **RETURN**
   **END**
APPENDIX C

Instructions for Using and Modifying DSTAP

As DSTAP was developed for research purposes only, it is not as user-friendly as other finite element codes. Therefore, the input of model data is somewhat more cumbersome than in production codes.

Program DSTAP expects to receive model data from the FORTRAN logical unit 25. The output from program DSTAP is sent to FORTRAN logical unit 26. The format of an input model is presented in subsection A below. An example of an input file (FORTRAN logical unit 25) may be found in Appendix D. The format of the input file assigned to logical FORTRAN unit 25 is by logical line numbers. Logical line numbers are different from physical line numbers. That is, logical line numbers should not be associated with the corresponding lines in FORTRAN unit 25. Several references will be made to Ref. [9], since the static capabilities of DSTAP are the same as STAP. Please see Ref. [9] for further details.

A. Format for an Input file:

Line 1. Problem Title -- Format (20A).

<table>
<thead>
<tr>
<th>Note</th>
<th>Columns</th>
<th>Variable</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>1-80</td>
<td>HED</td>
<td>Enter the title for the model to be entered. It is useful to make this as unique as possible.</td>
</tr>
<tr>
<td>Note</td>
<td>Columns</td>
<td>Variable</td>
<td>Entry</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>----------</td>
<td>-------</td>
</tr>
<tr>
<td>none</td>
<td>1-5</td>
<td>NUMNP</td>
<td>Total number of nodes in the model</td>
</tr>
<tr>
<td>1</td>
<td>6-10</td>
<td>NUMEG</td>
<td>Total number of element groups</td>
</tr>
<tr>
<td>2</td>
<td>11-15</td>
<td>NLCASE</td>
<td>Total number of load cases</td>
</tr>
<tr>
<td>none</td>
<td>16-20</td>
<td>MODEX</td>
<td>Execution mode:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 - Check data only</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 - Solve the modeled problem</td>
</tr>
<tr>
<td>3</td>
<td>21-25</td>
<td>IVB</td>
<td>Eigenvalue analysis flag:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 - Static analysis</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 - Eigenvalue analysis</td>
</tr>
<tr>
<td>4</td>
<td>26-30</td>
<td>IPlotF</td>
<td>Mode reconstruction flag:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 - do not plot eigenvectors</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 - reconstruct and plot eigenvectors</td>
</tr>
<tr>
<td>5,6</td>
<td>31-35</td>
<td>ID1</td>
<td>Determinant flag</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 - Do not save determinant</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 - Write determinant to unit 12</td>
</tr>
<tr>
<td>none</td>
<td>36-40</td>
<td>ID2</td>
<td>Unused</td>
</tr>
<tr>
<td>5</td>
<td>41-45</td>
<td>ID3</td>
<td>Eigenvector from PLOTB flag</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 - Vector is not saved</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 - Vector is written to unit 17</td>
</tr>
<tr>
<td>5</td>
<td>46-50</td>
<td>ID4</td>
<td>Scaled eigenvector flag (EXPVEC)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 - Vector is not saved</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 - Vector is written to unit 13</td>
</tr>
<tr>
<td>5,7</td>
<td>51-55</td>
<td>ID5</td>
<td>Initial state vector flag</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 - Vector is not saved</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 - Vector is written to unit 15</td>
</tr>
<tr>
<td>5,8</td>
<td>56-60</td>
<td>ID6</td>
<td>Mode shape flag (PLOTC)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 - Vector is not saved</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 - Vector is written to unit 16</td>
</tr>
<tr>
<td>5</td>
<td>61-65</td>
<td>ID7</td>
<td>Reconstruction state vector flag</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 - Vector is not saved</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 - Vector is written to unit 14</td>
</tr>
</tbody>
</table>

Notes:
1. DSTAP has been set-up so that this entry must be 2 whenever an exact element is used. The FEM-beam element group must be group 1 and the exact-beam element group 2.
2. This entry is ignored during eigenvalue analyses. In a static problem, this entry is used in the same manner as described by Ref. [9], page 219.
3. This entry must be present for eigenvalue problems.
4. This entry must be present for the reconstruction of eigenvectors. It is ignored in static problems.
5. ID1 through ID7 are debug flags, which were used during the debugging of DSTAP. Some of the information provided by the different flags is redundant.
6. When ID1 is set on, the function g(x), presented in the Program Description section is written to logical unit 12.
7. The computed initial state vector is written to logical unit 15. This information may used to obtain the internal reactions of the mode shape, at
8. The reconstructed mode shape of the exact-beam elements is written to unit 16. This information is the typical mode shape information obtained with a standard transfer matrix method program.

Line 3. Eigenvalue Problem Parameters -- Unformatted. This line is not used in a static analysis.

<table>
<thead>
<tr>
<th>Note</th>
<th>Type</th>
<th>Variable</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>INTEGER</td>
<td>METHOD</td>
<td>Linear eigenvalue extraction routine to use:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 - Subspace Iteration</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 - EIGENR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3 - IMSL EIGZF</td>
</tr>
<tr>
<td>2</td>
<td>REAL*8</td>
<td>FREQ1</td>
<td>Lower end of the frequency range of interest</td>
</tr>
<tr>
<td>3</td>
<td>REAL*8</td>
<td>STEP</td>
<td>Step size for Incremental search and step-off size from eigenvalues.</td>
</tr>
<tr>
<td></td>
<td>none</td>
<td>IWTWIL</td>
<td>Wittrick-Williams flag</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 - Counter off</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 - Counter on</td>
</tr>
<tr>
<td>4</td>
<td>REAL*8</td>
<td>WSTEP</td>
<td>Step-off frequency from a computed eigenvalue for the computation of the Wittrick-Williams counter.</td>
</tr>
</tbody>
</table>

Notes:
1. This is the extraction method to be used in purely FEM models. This is ignored whenever one or more exact elements are present. The Sub-Space iteration method is explained in Ref. [9] and was implemented by Ref. [58]. The EIGENR and EIGZF methods were described in the Program Description section of this dissertation.

2. FREQ1 and FREQ2 determine the frequency range of interest.
3. One third of this value is used as the interval size in the Incremental Search method and is used as a step-off after an eigenvalue is found.
4. At an eigenvalue, it is most likely that the dynamic stiffness matrix will be nonsingular. Thus, it is necessary to step-off from a computed eigenvalue to evaluate the Wittrick-Williams counter.

Line 4. Nodal Point Data - Format (4I5, 3F10.0, I5) This line is repeated as necessary.

<table>
<thead>
<tr>
<th>Note</th>
<th>Columns</th>
<th>Variable</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1-5</td>
<td>N</td>
<td>Node number</td>
</tr>
<tr>
<td>1</td>
<td>6-10</td>
<td>ID(1,N)</td>
<td>Boundary constraints - 0=free, 1=fixed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X - Translation boundary code</td>
</tr>
<tr>
<td>1</td>
<td>11-15</td>
<td>ID(2,N)</td>
<td>Boundary constraints - 0=free, 1=fixed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Y - Translation boundary code</td>
</tr>
</tbody>
</table>
163

1 16-20 ID(3,N) Boundary constraints - 0=free, 1=fixed
   Rotation boundary code
1 21-30 X(N) X-coordinate of node N
1 31-40 Y(N) Y-coordinate of node N
1 41-50 Z(N) Z-coordinate of node N
1 51-55 KN Node number increment for automatic node
generation

Notes:
1. See Ref. [9], page 218, for details.

Lines 5 and 5a. Load Information (Omit for Eigenvalue Problems)
   This line is repeated as necessary.

See Ref. [9], page 219, for this information. This section is only used in
static analysis. In this type of problem, DSTAP operates in the same manner
as STAP.

Line 6. FEM-beam Element group data - Format (3I5)
   This line must be present whenever an exact element is used.

<table>
<thead>
<tr>
<th>Note</th>
<th>Columns</th>
<th>Variable</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>1-5</td>
<td>NPAR(1)</td>
<td>Enter the number 2</td>
</tr>
</tbody>
</table>
| 1    | 6-10    | NPAR(2)  | Enter the number FEM-beam elements. This
        |         |          | entry may be set to zero for purely exact-
        |         |          | beam element models. |
| 2    | 11-15   | NPAR(3)  | Number of different set of material/cross-
        |         |          | sectional properties. May be set to zero
        |         |          | for purely exact-beam element models. |

Notes:
1. Set this entry to zero when no FEM-beam elements are present in the model.
2. This entry determines how many Lines 7 are present, see below.

Line 7. Material/Cross-Sectional Properties for FEM-Beam Elements -
   Unformatted
   This line is repeated as necessary.

<table>
<thead>
<tr>
<th>Note</th>
<th>Type</th>
<th>Variable</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>INTEGER</td>
<td>N</td>
<td>Number of property set.</td>
</tr>
<tr>
<td>none</td>
<td>REAL*8</td>
<td>E(N)</td>
<td>Modulus of elasticity.</td>
</tr>
<tr>
<td>none</td>
<td>REAL*8</td>
<td>AREA(N)</td>
<td>Cross-sectional area.</td>
</tr>
<tr>
<td>none</td>
<td>REAL*8</td>
<td>IXX(N)</td>
<td>Second area moment of inertia.</td>
</tr>
<tr>
<td>none</td>
<td>REAL*8</td>
<td>RHO(N)</td>
<td>Mass density.</td>
</tr>
</tbody>
</table>
| 2    | REAL*8 | SYT(N)   | Yield strength. Set to zero for this
        |         |          | version of DSTAP. |
Notes:
1. The property set number must be entered sequentially, starting at 1, for each element group. No set numbers may be skipped.
2. The version of DSTAP presented here does not use the Yield Strength of the FEM-beam elements. Please set it equal to zero (0.0D0).

Line 8. FEM-Beam Element Information - Format (5I5)
This line is repeated as necessary.

<table>
<thead>
<tr>
<th>Note</th>
<th>Columns</th>
<th>Variable</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1-5</td>
<td>M</td>
<td>FEM-Beam element number.</td>
</tr>
<tr>
<td>none</td>
<td>6-10</td>
<td>II</td>
<td>Ith node of the beam element.</td>
</tr>
<tr>
<td>none</td>
<td>11-12</td>
<td>JJ</td>
<td>Jth node of the beam element.</td>
</tr>
<tr>
<td>none</td>
<td>16-20</td>
<td>MTYPE</td>
<td>Material property set number. See Line 7 for details.</td>
</tr>
<tr>
<td>2</td>
<td>21-25</td>
<td>KG</td>
<td>Node generation Increment.</td>
</tr>
</tbody>
</table>

Notes:
1. The element numbers must be entered sequentially, starting with element number 1. The last occurrence of Line 8 must have M equal to NPAR(2) of Line 6.
2. Please refer to Ref. [9], page 222, for details of KG.

Line 9. Exact-Beam Element group data - Format (3I5)
Not used in a static analysis

<table>
<thead>
<tr>
<th>Note</th>
<th>Columns</th>
<th>Variable</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>1-5</td>
<td>NPAR(1)</td>
<td>Enter the number 3</td>
</tr>
<tr>
<td>1</td>
<td>6-10</td>
<td>NPAR(2)</td>
<td>Enter the number exact-beam elements. May be set to zero for purely FEM-beam element models.</td>
</tr>
<tr>
<td>2</td>
<td>11-15</td>
<td>NPAR(3)</td>
<td>Number of different set of material/cross-sectional properties. May be set to zero for purely FEM-beam element models.</td>
</tr>
</tbody>
</table>

Notes:
1. Set this entry to zero when no exact-beam elements are present in the model.
2. This entry determines how many Lines 10 are present, see below.

Line 10. Material/Cross-Sectional Properties for Exact-Beam Elements-Unformatted
Not used in a static analysis

<table>
<thead>
<tr>
<th>Note</th>
<th>Type</th>
<th>Variable</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>INTEGER</td>
<td>N</td>
<td>Number of property set.</td>
</tr>
<tr>
<td>none</td>
<td>REAL*8</td>
<td>E(N)</td>
<td>Modulus of elasticity.</td>
</tr>
<tr>
<td>2</td>
<td>REAL*8</td>
<td>G(N)</td>
<td>Modulus of rigidity. Enter a negative number for G(N)=infinity.</td>
</tr>
</tbody>
</table>
165

3 REAL*8 AREA(N) Cross-sectional area.
none REAL*8 IXX(N) Second area moment of inertia.
none REAL*8 RHO(N) Mass density.

Notes:
1. The property set number must be entered sequentially, starting at 1, for
each element group. No set numbers may be skipped.
2. Whenever the modulus of rigidity is entered as a negative number, DSTAP
assumes that the modulus of rigidity is infinite (neglects shear
deformation) and assumes that the rotatory inertia is zero.
3. This second area moment of inertia is taken about the neutral axis of
bending. For example, if the plane of bending lies on this page, then the
second area moment of inertia is taken about a line coming out of this page.

Line 11. Exact-Beam Element Information - Format (5I5)
Not used in a static analysis.
This line is repeated as necessary.

<table>
<thead>
<tr>
<th>Note</th>
<th>Columns</th>
<th>Variable</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1-5</td>
<td>M</td>
<td>Exact-Beam element number.</td>
</tr>
<tr>
<td>none</td>
<td>6-10</td>
<td>II</td>
<td>Ith node of the beam element.</td>
</tr>
<tr>
<td>none</td>
<td>11-12</td>
<td>JJ</td>
<td>Jth node of the beam element.</td>
</tr>
<tr>
<td>none</td>
<td>16-20</td>
<td>MTYPE</td>
<td>Material property set number. See Line 9 for details.</td>
</tr>
<tr>
<td>2</td>
<td>21-25</td>
<td>KG</td>
<td>Node generation Increment.</td>
</tr>
</tbody>
</table>

Notes:
1. The element numbers must be entered sequentially, starting with element
number 1. The last occurrence of Line 11 must have M equal to NPAR(2) of
Line 9.
2. Please refer to Ref. [9], page 222, for details of KG.

B. Running DSTAP:

Program DSTAP is located on the VAX cluster at the Virginia Tech
Computing Center, in subdirectory DUA:\JARAALJA.DISS. Thus, a global DCL
symbol DSTAP may be defined as:

```
DSTAP :== RUN DUA:\JARAALJA.DISS]DSTAP <RETURN>
```

to avoid typing in the subdirectory designation everytime DSTAP must be run.
Then, to actually run DSTAP the command: DSTAP $RETURN$ at command level will
invoke DSTAP.
C. Modifying DSTAP:

A command file LD.COM in the subdirectory DUA\1:\[JARAALJA.DISS] has been written to link all the necessary modules to generate program DSTAP. The list of modules needed to generate this version of DSTAP was presented in Appendix B. In summary, the main source code files are:

DUA3:\[JARAALJA.DISS]DSTAP.FOR  The main program and function CPUSEC.
DUA3:\[JARAALJA.DISS]STAP3.OLB  The routines from program STAP and the beam element routines.
DUA3:\[JARAALJA.DISS]NEWEIG.FOR  The routines for the solution of the nonlinear eigenvalue problem.
DUA3:\[JARAALJA.DISS]PLOTB2.FOR  The routines for reconstructing and plotting the eigenvectors.
IMSL-library  This commercial library contains the routines which allow the eigenvalue extraction using subprogram EIGZF. This library may not be modified.
Tektronix PLOT10 library  This commercial library contains the subroutines necessary to drive a Tektronix-type terminal in graphics mode. This library may not be modified.

Thus, after modifying and compiling the desired set of subroutines, a command @DUA3:\[JARAALJA.DISS]LD <RETURN> at command level will combine all the modules and create a new DSTAP.EXE file which will contain the modifications.
APPENDIX D

Sample Runs using Program DSTAP

The following pages present the input and output from the two exact-element models used in the first example of the Numerical Examples section. The input for the two-exact-element model of the fixed-fixed beam is presented in Fig. D-1. The corresponding DSTAP output is presented in Fig. D-2. Some of the lines in Fig. D-2 have been edited so that they fit in the prescribed space. None of the information from the output file has been changed in the editing process.

In the same manner, the input for the four-exact-element model is presented in Fig. D-3 and the corresponding output in Fig. D-4. Again, the output file has been edited to fit on the page; however, none of the information has been changed.
Figure D-1. DSTAP Input for the Two-Exact-Element Model.
Figure D-2. DSTAP output for the Two-Exact-Element Model.
ELEMENT GROUP DATA

ELEMENT DEFINITION

ELEMENT TYPE . . . . . . . . . . . . . . . . . . ( NPAR(1) ) = 3
EQ.3, CONTINUOUS BEAM ELEMENTS
NUMBER OF ELEMENTS . . . . . . . . . . . . . . . . . . ( NPAR(2) ) = 2

MATERIAL DEFINITION

NUMBER OF DIFFERENT SETS OF MATERIAL
AND CROSS-SECTIONAL CONSTANTS . . . . . ( NPAR(3) ) = 1

<table>
<thead>
<tr>
<th>SET</th>
<th>MODULUS OF ELASTICITY (E)</th>
<th>MODULUS OF RIGIDITY (G)</th>
<th>AREA</th>
<th>MOMENT OF INERTIA (I)</th>
<th>DENSITY (RHO)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.300000E+08</td>
<td>Infinite</td>
<td>0.125000E+00</td>
<td>0.051042E-03</td>
<td>0.730493E-03</td>
</tr>
</tbody>
</table>

ELEMENT INFORMATION /

<table>
<thead>
<tr>
<th>ELEMENT NUMBER</th>
<th>NODE I</th>
<th>NODE J</th>
<th>MATERIAL SET NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure D-2. DSTAP output for the Two-Exact-Element Model (continued).
### POLES (Infinity wraps)

<table>
<thead>
<tr>
<th>POLE No.</th>
<th>UPPER POLE</th>
<th>LOWER POLE</th>
<th>MULTIPLICITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2272.45809</td>
<td>2272.45809</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>6265.11822</td>
<td>6264.11822</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>12281.17647</td>
<td>12280.17647</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>20300.75528</td>
<td>20299.75528</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>30325.33223</td>
<td>30324.33223</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>42392.16107</td>
<td>42391.16107</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>53057.63058</td>
<td>53057.63058</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>56399.22068</td>
<td>56398.22068</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>72428.53678</td>
<td>72427.53678</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>90472.76740</td>
<td>90471.76740</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>110521.91253</td>
<td>110520.91253</td>
<td>2</td>
</tr>
</tbody>
</table>

* Natural frequencies of the fixed-fixed continuum elements, computed with an accuracy of .500E-01.

* Based on the natural frequencies of similar elements without shear deformation or rotatory inertia (Natural frequency plus one)

### BPATHD RESULTS

<table>
<thead>
<tr>
<th>WILLIAMS Eigen. (rads/sec)</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>568.11452200995041</td>
</tr>
<tr>
<td>2</td>
<td>1566.0295536311677</td>
</tr>
<tr>
<td>3</td>
<td>3070.0440720729415</td>
</tr>
<tr>
<td>4</td>
<td>5074.938819592707260</td>
</tr>
<tr>
<td>5</td>
<td>7581.0830565363537</td>
</tr>
<tr>
<td>6</td>
<td>10588.45476799008461</td>
</tr>
<tr>
<td>7</td>
<td>14097.05516932095050</td>
</tr>
<tr>
<td>8</td>
<td>18105.088419377764967</td>
</tr>
<tr>
<td>9</td>
<td>22617.94184764444040</td>
</tr>
<tr>
<td>10</td>
<td>26528.81529027919487</td>
</tr>
<tr>
<td>11</td>
<td>27630.2281297992403</td>
</tr>
<tr>
<td>12</td>
<td>33143.7430937236709</td>
</tr>
<tr>
<td>13</td>
<td>39158.4865711666944</td>
</tr>
<tr>
<td>14</td>
<td>45674.458734240057063</td>
</tr>
<tr>
<td>15</td>
<td>52594.65912725856132</td>
</tr>
<tr>
<td>16</td>
<td>53057.6305805335055</td>
</tr>
<tr>
<td>17</td>
<td>60210.0691623237143</td>
</tr>
<tr>
<td>18</td>
<td>68225.74224404569498</td>
</tr>
<tr>
<td>19</td>
<td>76750.62564726539455</td>
</tr>
<tr>
<td>20</td>
<td>79586.44970937658597</td>
</tr>
<tr>
<td>21</td>
<td>85772.8423159319964</td>
</tr>
<tr>
<td>22</td>
<td>95298.25798896946883</td>
</tr>
</tbody>
</table>

Figure D-2. DSTAP output for the Two-Exact-Element Model (continued).
SOLUTION TIME LOG IN SECONDS

FOR PROBLEM: FIXED-FIXED BEAM COMPARISON WITH 2 C.E.

<table>
<thead>
<tr>
<th>Description</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>The input of geometry</td>
<td>0.49</td>
</tr>
<tr>
<td>Computation of global F.E.M. matrices</td>
<td>0.23</td>
</tr>
<tr>
<td>Pole extraction of Continuum elements</td>
<td>0.05</td>
</tr>
<tr>
<td>Brute Force - False extraction</td>
<td>61.86</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>62.63</strong></td>
</tr>
</tbody>
</table>

ADDITIONAL TIMES OF INTEREST:

- Time to compute Eigenvalue number: 1. . . . . . . . . . . . . . . . . = 1.65
- Time to compute Eigenvalue number: 2. . . . . . . . . . . . . . . . . = 1.78
- Time to compute Eigenvalue number: 3. . . . . . . . . . . . . . . . . = 2.02
- Time to compute Eigenvalue number: 4. . . . . . . . . . . . . . . . . = 2.11
- Time to compute Eigenvalue number: 5. . . . . . . . . . . . . . . . . = 2.52
- Time to compute Eigenvalue number: 6. . . . . . . . . . . . . . . . . = 2.49
- Time to compute Eigenvalue number: 7. . . . . . . . . . . . . . . . . = 2.49
- Time to compute Eigenvalue number: 8. . . . . . . . . . . . . . . . . = 2.61
- Time to compute Eigenvalue number: 9. . . . . . . . . . . . . . . . . = 2.69
- Time to compute Eigenvalue number: 10. . . . . . . . . . . . . . . . = 3.03
- Time to compute Eigenvalue number: 11. . . . . . . . . . . . . . . . = 1.57
- Time to compute Eigenvalue number: 12. . . . . . . . . . . . . . . . = 3.05
- Time to compute Eigenvalue number: 13. . . . . . . . . . . . . . . . = 3.21
- Time to compute Eigenvalue number: 14. . . . . . . . . . . . . . . . = 3.51
- Time to compute Eigenvalue number: 15. . . . . . . . . . . . . . . . = 3.38
- Time to compute Eigenvalue number: 16. . . . . . . . . . . . . . . . = 1.98
- Time to compute Eigenvalue number: 17. . . . . . . . . . . . . . . . = 3.29
- Time to compute Eigenvalue number: 18. . . . . . . . . . . . . . . . = 3.75
- Time to compute Eigenvalue number: 19. . . . . . . . . . . . . . . . = 3.75
- Time to compute Eigenvalue number: 20. . . . . . . . . . . . . . . . = 2.75
- Time to compute Eigenvalue number: 21. . . . . . . . . . . . . . . . = 2.63
- Time to compute Eigenvalue number: 22. . . . . . . . . . . . . . . . = 4.00

Figure D-2. DSTAP output for the Two-Exact-Element Model (continued).
FIXED-FIXED BEAM COMPARISON WITH 4 C.E.

5  2  0  1  1
2  1.100000. 301. 1 10.
1  1  1  1  0.000  0.000  0.000  1  0.000
2  0  0  0  6.000  0.000  0.000  1  0.000
5  1  1  1  24.000  0.000  0.000  1  0.000
2  0  0
3  4  1
1  0.300E+08  -11.5E6  0.125E+00  6.5104166666667E-04  7.304034314207753E-4  0.000E+00
1  1  2  1  1
4  4  5  1  1

Figure D-3. DSTAP input for the Four-Exact-Element Model.
**CONTROL INFORMATION**

<table>
<thead>
<tr>
<th>NUMBER OF Nodal Points</th>
<th>(NUMP) = 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUMBER OF ELEMENT GROUPS</td>
<td>(NUMEG) = 2</td>
</tr>
<tr>
<td>NUMBER OF LOAD CASES</td>
<td>(NULCSE) = 0</td>
</tr>
<tr>
<td>SOLUTION MODE</td>
<td>(MODEX) = 1</td>
</tr>
<tr>
<td>VIBRATION ANALYSIS</td>
<td>(IVB) = 1</td>
</tr>
<tr>
<td>EIGENVECTOR RECONSTRUCTION FOR PLOTTING</td>
<td>(IPLOTF) = 0</td>
</tr>
<tr>
<td>METHOD USED IN EIGEN EXTRACTION</td>
<td>(METHOD) = 2</td>
</tr>
<tr>
<td>STARTING FREQUENCY</td>
<td>(FREQ1) = 0.1000000E+01</td>
</tr>
<tr>
<td>ENDING FREQUENCY</td>
<td>(FREQ2) = 0.1000000E+06</td>
</tr>
<tr>
<td>STEP SIZE FOR FREQUENCY SEARCH</td>
<td>(STEP) = 0.3010000E+03</td>
</tr>
<tr>
<td>WITTRICK-WILLIAMS CHECK</td>
<td>(IWTWL) = 1</td>
</tr>
<tr>
<td>WITTRICK-WILLIAMS STEP-OFF</td>
<td>(WTSTP) = 0.1000000E+02</td>
</tr>
</tbody>
</table>

**Nodal Point Data**

<table>
<thead>
<tr>
<th>NODE</th>
<th>BOUNDARY CONDITIONS</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.000</td>
<td>0.000</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6.000</td>
<td>0.000</td>
<td>0.0</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>24.000</td>
<td>0.000</td>
<td>0.0</td>
</tr>
</tbody>
</table>

**Fig. D-4.** DSTAP output for the Four-Exact-Element Model.
ELEMENT GROUP DATA

ELEMENT DEFINITION

ELEMENT TYPE ........................................ (NPAR(1)) = 3
eq3, CONTINUUM BEAM ELEMENTS

NUMBER OF ELEMENTS............................... (NPAR(2)) = 4

MATERIAL DEFINITION

NUMBER OF DIFFERENT SETS OF MATERIAL
AND CROSS-SECTIONAL CONSTANTS .......... (NPAR(3)) = 1

<table>
<thead>
<tr>
<th>SET NO.</th>
<th>MODULUS OF ELASTICITY</th>
<th>MODULUS OF RIGIDITY</th>
<th>AREA</th>
<th>MOMENT OF INERTIA</th>
<th>DENSITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.300000E+08</td>
<td>Infinite</td>
<td>0.125000E+00</td>
<td>0.851042E-03</td>
<td>0.730409E-03</td>
</tr>
</tbody>
</table>

ELEMENT INFORMATION /

<table>
<thead>
<tr>
<th>ELEMENT NUMBER-N</th>
<th>NODE I</th>
<th>NODE J</th>
<th>MATERIAL SET NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>5</td>
<td>1</td>
</tr>
</tbody>
</table>

TOTAL SYSTEM DATA

Figure D-4. DSTAP output for the Four-Exact-Element Model (continued).
**P.O.L.E.S (Infinity wraps)**  
(radans/second)

<table>
<thead>
<tr>
<th>POLE No.</th>
<th>UPPER BOUND</th>
<th>POLE</th>
<th>MULTIPLICITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9090.83233</td>
<td>9089.83233</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>25057.47288</td>
<td>25056.47288</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>49121.70588</td>
<td>49120.70588</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>81200.02111</td>
<td>81199.02111</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>121298.32890</td>
<td>121297.32890</td>
<td>4</td>
</tr>
</tbody>
</table>

* Natural frequencies of the fixed-fixed continuum elements, computed with an accuracy of 0.005E-01.

* Based on the natural frequencies of similar elements without shear deformation or rotatory inertia (Natural frequency plus one)

**BRTHO RESULTS**

<table>
<thead>
<tr>
<th>Wittrock Computed</th>
<th>Eigenvalue</th>
<th>Mes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Williams Counter No.</td>
<td>(rads/sec)</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>568.1145220095051</td>
<td>FAL-The tolerance test was me</td>
</tr>
<tr>
<td>2</td>
<td>1565.0296563611560</td>
<td>FAL-The tolerance test was me</td>
</tr>
<tr>
<td>3</td>
<td>3070.04408270259376</td>
<td>FAL-The tolerance test was me</td>
</tr>
<tr>
<td>4</td>
<td>5074.93861958207805</td>
<td>FAL-The tolerance test was me</td>
</tr>
<tr>
<td>5</td>
<td>7581.08305653631123</td>
<td>FAL-The tolerance test was me</td>
</tr>
<tr>
<td>6</td>
<td>10588.45476796011121</td>
<td>FAL-The tolerance test was me</td>
</tr>
<tr>
<td>7</td>
<td>14097.06516832279777</td>
<td>FAL-The tolerance test was me</td>
</tr>
<tr>
<td>8</td>
<td>18162.88419379174047</td>
<td>FAL-The tolerance test was me</td>
</tr>
<tr>
<td>9</td>
<td>22677.94184761731458</td>
<td>FAL-The tolerance test was me</td>
</tr>
<tr>
<td>10</td>
<td>26526.81529027919532</td>
<td>FAL-The tolerance test was me</td>
</tr>
<tr>
<td>11</td>
<td>27630.22812637520440</td>
<td>FAL-The tolerance test was me</td>
</tr>
<tr>
<td>12</td>
<td>33143.74303986000541</td>
<td>FAL-The tolerance test was me</td>
</tr>
<tr>
<td>13</td>
<td>39158.48657828463820</td>
<td>FAL-The tolerance test was me</td>
</tr>
<tr>
<td>14</td>
<td>45674.45874491329778</td>
<td>FAL-The tolerance test was me</td>
</tr>
<tr>
<td>15</td>
<td>52691.65953732477224</td>
<td>FAL-The tolerance test was me</td>
</tr>
<tr>
<td>16</td>
<td>53057.63058958389794</td>
<td>FAL-The tolerance test was me</td>
</tr>
<tr>
<td>17</td>
<td>60210.08896276889118</td>
<td>FAL-The tolerance test was me</td>
</tr>
<tr>
<td>18</td>
<td>68299.74701399454716</td>
<td>FAL-The tolerance test was me</td>
</tr>
<tr>
<td>19</td>
<td>76750.63369343295524</td>
<td>FAL-The tolerance test was me</td>
</tr>
<tr>
<td>20</td>
<td>79586.44587083758597</td>
<td>FAL-The tolerance test was me</td>
</tr>
<tr>
<td>21</td>
<td>85772.74900103338587</td>
<td>FAL-The tolerance test was me</td>
</tr>
<tr>
<td>22</td>
<td>95296.09293674486416</td>
<td>FAL-The tolerance test was me</td>
</tr>
</tbody>
</table>

Figure D-4. DSTAP output for the Four-Exact-Element Model (continued).
SOLUTION TIME LOG IN SECONDS

FOR PROBLEM: FIXED-FIXED BEAM COMPARISON WITH 4 C.E.
The input of geometry ........................................ = 0.48
Computation of global F.E.M. matrices ......................... = 0.25
Set-up for a Continuum-F.E. problem .......................... = 0.01
Pole extraction of Continuum elements ....................... = 0.04
Brute Force - Falsi extraction ............................... = 135.04
Post-processing of eigenvectors .............................. = 0.01

TOTAL = 135.83

ADDITIONAL TIMES OF INTEREST:

Time to compute Eigenvalue number: 1. ...................... = 3.79
Time to compute Eigenvalue number: 2. ...................... = 3.88
Time to compute Eigenvalue number: 3. ...................... = 4.08
Time to compute Eigenvalue number: 4. ...................... = 4.53
Time to compute Eigenvalue number: 5. ...................... = 4.90
Time to compute Eigenvalue number: 6. ...................... = 4.98
Time to compute Eigenvalue number: 7. ...................... = 5.24
Time to compute Eigenvalue number: 8. ...................... = 5.74
Time to compute Eigenvalue number: 9. ...................... = 6.15
Time to compute Eigenvalue number: 10. ..................... = 5.79
Time to compute Eigenvalue number: 11. ..................... = 3.96
Time to compute Eigenvalue number: 12. ..................... = 6.72
Time to compute Eigenvalue number: 13. ..................... = 6.99
Time to compute Eigenvalue number: 14. ..................... = 7.46
Time to compute Eigenvalue number: 15. ..................... = 7.75
Time to compute Eigenvalue number: 16. ..................... = 3.43
Time to compute Eigenvalue number: 17. ..................... = 8.03
Time to compute Eigenvalue number: 18. ..................... = 8.38
Time to compute Eigenvalue number: 19. ..................... = 8.72
Time to compute Eigenvalue number: 20. ..................... = 5.07
Time to compute Eigenvalue number: 21. ..................... = 7.27
Time to compute Eigenvalue number: 22. ..................... = 9.24

Figure D-4. DSTAP output for the Four-Exact-Element Model (continued).
The vita has been removed from the scanned document