INFORMATION TO USERS

This reproduction was made from a copy of a document sent to us for microfilming. While the most advanced technology has been used to photograph and reproduce this document, the quality of the reproduction is heavily dependent upon the quality of the material submitted.

The following explanation of techniques is provided to help clarify markings or notations which may appear on this reproduction.

1. The sign or "target" for pages apparently lacking from the document photographed is "Missing Page(s)". If it was possible to obtain the missing page(s) or section, they are spliced into the film along with adjacent pages. This may have necessitated cutting through an image and duplicating adjacent pages to assure complete continuity.

2. When an image on the film is obliterated with a round black mark, it is an indication of either blurred copy because of movement during exposure, duplicate copy, or copyrighted materials that should not have been filmed. For blurred pages, a good image of the page can be found in the adjacent frame. If copyrighted materials were deleted, a target note will appear listing the pages in the adjacent frame.

3. When a map, drawing or chart, etc., is part of the material being photographed, a definite method of "sectioning" the material has been followed. It is customary to begin filming at the upper left hand corner of a large sheet and to continue from left to right in equal sections with small overlaps. If necessary, sectioning is continued again—beginning below the first row and continuing on until complete.

4. For illustrations that cannot be satisfactorily reproduced by xerographic means, photographic prints can be purchased at additional cost and inserted into your xerographic copy. These prints are available upon request from the Dissertations Customer Services Department.

5. Some pages in any document may have indistinct print. In all cases the best available copy has been filmed.
Kumar, Dharmendra

A COMPUTATIONALLY EFFICIENT METHOD OF ANALYZING THE
PARAMETRIC SUBSTRUCTURES

The University of Arizona

M.S. 1985

University Microfilms International 300 N. Zeeb Road, Ann Arbor, MI 48106
PLEASE NOTE:

In all cases this material has been filmed in the best possible way from the available copy. Problems encountered with this document have been identified here with a check mark \( \checkmark \).

1. Glossy photographs or pages
2. Colored illustrations, paper or print
3. Photographs with dark background
4. Illustrations are poor copy
5. Pages with black marks, not original copy
6. Print shows through as there is text on both sides of page
7. Indistinct, broken or small print on several pages \( \checkmark \)
8. Print exceeds margin requirements
9. Tightly bound copy with print lost in spine
10. Computer printout pages with indistinct print
11. Page(s) _________ lacking when material received, and not available from school or author.
12. Page(s) _________ seem to be missing in numbering only as text follows.
13. Two pages numbered _________. Text follows.
14. Curling and wrinkled pages
15. Other

University Microfilms International
A COMPUTATIONALLY EFFICIENT METHOD OF ANALYZING THE PARAMETRIC SUBSTRUCTURES

by

Dharmendra Kumar

A Thesis Submitted to the Faculty of the
DEPARTMENT OF AEROSPACE AND MECHANICAL ENGINEERING
In Partial fulfillment of the requirements
For the Degree of
MASTER OF SCIENCE
WITH A MAJOR IN MECHANICAL ENGINEERING
In the Graduate College
THE UNIVERSITY OF ARIZONA

1985
STATEMENT OF AUTHOR

This thesis has been submitted in partial fulfillment of requirements for an advanced degree at the University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgement of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate college when in his or her judgement the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: Shamendra Kumar

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

Dr. H. A. Kamel
Professor of Mechanical Engineering

Date
7/9/85
dedicated to my parents and
all Gabdoos of my family
ACKNOWLEDGMENTS

I wish to express my sincere appreciation to Prof. H. A. Kamel for his valuable advice and guidance that I received throughout the research and writing of this thesis. I am also thankful to Dr. B. R. Simon and Dr. N. G. R. Iyengar for a careful review of this thesis.

I would like to extend my sincere thanks to my colleagues, Mr. R. R. Nagulpally, Mr. A. V. Mobley, Mr. Bob Campbell, Mr. David W. Colla, Mr. Sungmin Kim, Mr. Hamid Movahedi Lankarani, Mr. David A. Gubbels, Mr. Andy Kissil, Mr. Jorge Ambrosio, Dr. Manuel Pereira, Dr. Ramesh Kolar and Mr. Prasad Rao Errabelli for their contribution to this research.

I would also like to express my heart-felt appreciation to Ruth Makepeace, Carol Ann Calderon, Susan Lewis and Linda Darrell for their kind cooperation and encouragement during the critical phases of this research.

Finally, much of the credit goes to my eldest brother, Dr. Shailendra Kumar and my eldest sister-in-law, Dr. Sudhanshu Kumar who have given their fullest support, encouragement and understanding to this effort.

I would like to thank the GIFTS group at the University of Arizona and CASA/GIFTS Inc. for funding this research.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF ILLUSTRATIONS</td>
<td>vii</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>viii</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>ix</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1. DISCUSSION OF SOME OF THE BASIC REQUIREMENTS FOR A WELL</td>
<td></td>
</tr>
<tr>
<td>BEHAVED FINITE ELEMENT MODEL</td>
<td>4</td>
</tr>
<tr>
<td>Rigid Body Motion Criterion</td>
<td>5</td>
</tr>
<tr>
<td>Interelement Compatibility Criterion</td>
<td>5</td>
</tr>
<tr>
<td>2. INTERPOLATION SCHEMES</td>
<td>8</td>
</tr>
<tr>
<td>Polynomial Interpolation</td>
<td>9</td>
</tr>
<tr>
<td>Linear Interpolation</td>
<td>10</td>
</tr>
<tr>
<td>Quadratic Interpolation</td>
<td>12</td>
</tr>
<tr>
<td>Cubic Interpolation</td>
<td>12</td>
</tr>
<tr>
<td>Bilinear Interpolation</td>
<td>13</td>
</tr>
<tr>
<td>Biquadratic Interpolation</td>
<td>13</td>
</tr>
<tr>
<td>Polynomial Regression using the Method of Least Squares</td>
<td>15</td>
</tr>
<tr>
<td>3. STIFFNESS MATRIX RECONDITIONING SCHEMES</td>
<td>19</td>
</tr>
<tr>
<td>Scheme 1</td>
<td>20</td>
</tr>
<tr>
<td>Scheme 2</td>
<td>22</td>
</tr>
<tr>
<td>A Mathematical Comparison of the Two Schemes</td>
<td>24</td>
</tr>
<tr>
<td>4. DEFLECTION AND STRESS COMPUTATION METHODS</td>
<td>26</td>
</tr>
<tr>
<td>Deflection Computations</td>
<td>26</td>
</tr>
<tr>
<td>Dependent Node Deflections in Constrained Substructures</td>
<td>33</td>
</tr>
<tr>
<td>Rigid Connection</td>
<td>33</td>
</tr>
<tr>
<td>Linear Boundary (Linear Interpolation)</td>
<td>36</td>
</tr>
<tr>
<td>Cubic Interpolation</td>
<td>36</td>
</tr>
<tr>
<td>Stress Computations</td>
<td>37</td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS – Continued

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5. RESULTS AND DISCUSSION</td>
<td>41</td>
</tr>
<tr>
<td>Part 1</td>
<td>41</td>
</tr>
<tr>
<td>Comparative Study of Various Orders of Interpolation</td>
<td>41</td>
</tr>
<tr>
<td>Effect of Scaling of Geometric Parameters</td>
<td>46</td>
</tr>
<tr>
<td>Comparative Performance Study of Different Point Selection schemes</td>
<td>49</td>
</tr>
<tr>
<td>Effect of Stretching the Range of the Interpolation Parameters</td>
<td>49</td>
</tr>
<tr>
<td>Part 2</td>
<td>52</td>
</tr>
<tr>
<td>Part 3</td>
<td>66</td>
</tr>
<tr>
<td>CONCLUSIONS AND PROPOSED RESEARCH</td>
<td>69</td>
</tr>
<tr>
<td>APPENDIX (a): GIFTS analysis procedure for generating reduced substructure matrix</td>
<td>71</td>
</tr>
<tr>
<td>APPENDIX (b): Analysis procedure for computing stress coefficients using GIFTS</td>
<td>72</td>
</tr>
<tr>
<td>APPENDIX (c): GIFTS analysis procedure for computing main model deflections and stress computations for substructure elements</td>
<td>73</td>
</tr>
<tr>
<td>LIST OF REFERENCES</td>
<td>74</td>
</tr>
</tbody>
</table>
## LIST OF ILLUSTRATIONS

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>A constrained substructure interface with a beam element</td>
<td>5</td>
</tr>
<tr>
<td>2.1</td>
<td>Linear Interpolation</td>
<td>11</td>
</tr>
<tr>
<td>2.2</td>
<td>Quadratic Interpolation</td>
<td>11</td>
</tr>
<tr>
<td>2.3</td>
<td>Cubic Interpolation</td>
<td>11</td>
</tr>
<tr>
<td>2.4</td>
<td>Bilinear Interpolation</td>
<td>14</td>
</tr>
<tr>
<td>2.5</td>
<td>Biquadratic Interpolation</td>
<td>14</td>
</tr>
<tr>
<td>4.1</td>
<td>Flow Chart for Program SOLVE</td>
<td>28</td>
</tr>
<tr>
<td>4.2</td>
<td>Flow Chart for Program MODD</td>
<td>29</td>
</tr>
<tr>
<td>4.3</td>
<td>Removal of rigid body displacements</td>
<td>31</td>
</tr>
<tr>
<td>4.4</td>
<td>Flow Chart for Program INTP</td>
<td>34</td>
</tr>
<tr>
<td>4.5</td>
<td>Rigid Connection</td>
<td>35</td>
</tr>
<tr>
<td>4.6</td>
<td>Linear Boundary (Linear Interpolation)</td>
<td>35</td>
</tr>
<tr>
<td>4.7</td>
<td>Cubic Interpolation</td>
<td>35</td>
</tr>
<tr>
<td>4.8</td>
<td>Flow Chart of Program STRE</td>
<td>38</td>
</tr>
<tr>
<td>4.9</td>
<td>Flow Chart for Program RESUL</td>
<td>40</td>
</tr>
<tr>
<td>5.1a</td>
<td>Test Example – 1</td>
<td>43</td>
</tr>
<tr>
<td>5.1b</td>
<td>Test Example – 2</td>
<td>44</td>
</tr>
<tr>
<td>5.2</td>
<td>A Comparative Performance of One-dimensional Interpolations</td>
<td>45</td>
</tr>
<tr>
<td>5.3</td>
<td>A Comparative Performance of 2-dimensional Interpolations</td>
<td>47</td>
</tr>
<tr>
<td>5.4</td>
<td>Effect of Scaling of Geometric Parameters</td>
<td>48</td>
</tr>
<tr>
<td>5.5</td>
<td>A Comparative Performance of Point Selection Schemes</td>
<td>50</td>
</tr>
<tr>
<td>5.6</td>
<td>Effect of Stretching the Range of Interpolation</td>
<td>51</td>
</tr>
<tr>
<td>5.7a</td>
<td>Constrained Substructure Family – C₁</td>
<td>53</td>
</tr>
<tr>
<td>5.7b</td>
<td>Constrained Substructure Family – C₂</td>
<td>54</td>
</tr>
<tr>
<td>5.7c</td>
<td>Constrained Substructure Family – C₃</td>
<td>55</td>
</tr>
<tr>
<td>5.7d</td>
<td>Constrained Substructure Family – C₄</td>
<td>56</td>
</tr>
<tr>
<td>5.7e</td>
<td>Constrained Substructure Family – C₅</td>
<td>57</td>
</tr>
<tr>
<td>5.7f</td>
<td>Constrained Substructure Family – C₆</td>
<td>58</td>
</tr>
<tr>
<td>5.7g</td>
<td>Constrained Substructure Family – C₇</td>
<td>59</td>
</tr>
<tr>
<td>5.7h</td>
<td>Constrained Substructure Family – C₈</td>
<td>60</td>
</tr>
<tr>
<td>5.7i</td>
<td>Constrained Substructure Family – C₉</td>
<td>61</td>
</tr>
<tr>
<td>5.7j</td>
<td>Constrained Substructure Family – C₁₀</td>
<td>62</td>
</tr>
<tr>
<td>5.8</td>
<td>Main Model – Web Frame of a Ship</td>
<td>63</td>
</tr>
<tr>
<td>5.9</td>
<td>Changed Geometry of the Main Model</td>
<td>67</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Table for Deflection Comparison</td>
<td>64</td>
</tr>
<tr>
<td>5.2 Table for Stress Comparison</td>
<td>65</td>
</tr>
<tr>
<td>5.3 Table for CPU Time Comparison</td>
<td>68</td>
</tr>
</tbody>
</table>
ABSTRACT

The research effort presents a method by which to establish a library of pre-computed substructures and to use them to assemble complex structures more quickly.

Basically, the method involves computing an approximate stiffness matrix for a substructure and a stress interpolation matrix for the selected elements, for given parameters governing the geometry of a substructure.

The approximate stiffness matrix is expressed as a function of the controlling geometric parameters. The interpolated stiffness matrix is reconditioned to satisfy equilibrium conditions by including the missing rigid body motions.

To establish the stress interpolation matrix, the stress coefficients for the selected elements are computed in terms of unit displacement of degrees of freedom of the master nodes. A polynomial regression is performed using the method of least squares to obtain the stress interpolation matrix over the parametric space.

Several studies were performed to analyze the behavior of approximate stiffness matrices. The Web Frame of a ship was analyzed to show the accuracy and efficiency of the proposed method compared to the regular finite element method. The results are encouraging and show good agreement with the values obtained using the regular finite element.
INTRODUCTION

In the field of Structural Optimization, the problems involved in the real world are usually very complex. A sizable number of iterations are required in the optimization process before an acceptable combination of design parameters are obtained. The Finite Element Method has been utilized extensively to handle such complex problems. Many general purpose finite element programs, such as, NASTRAN, ANSYS, GIFTS, MARC etc., are available which are capable of modeling and analyzing such complex problems with reliable accuracies. The analysis time has been greatly reduced due to the efficient use of today's sophisticated computer systems by these general purpose programs. However, even with such versatile finite element programs and highly sophisticated computer systems, the problems in Structural Optimization still require a significant amount of time and money.

The study presents a method by which it is possible to change the design parameters of a structure and quickly re-analyze the structure without the need for a complete re-analysis.

In this method, a library of pre-analyzed substructures is established. The library contains families of pre-analyzed substructure stiffness matrices where each family has the same topology. The actual shape is controlled by certain geometric parameters, such as length of sides, plate thicknesses, radius of fillets etc. These substructures can be used to model the structure which is to be analyzed with structural optimization in mind.

Basically the method involves computing an approximate stiffness matrix and a selective stress interpolation matrix for selected elements for the given geometric parameters of a substructure.
Although the stiffness matrix behaves in general in a complex manner as a function of the geometric parameters, it is possible to approximate its dependence on the governing parameters. The method used to approximate the stiffness matrix is to compute it for specific values of the control parameters. For any values, other than those for which the stiffness has been computed, interpolation is used. Here the interpolation concept of the Finite Element Method can be generalized to handle several variables (multidimensional). Naturally, the more the variables and the higher the order of interpolation, the more stiffness matrices are computed. This will require a large amount of disk storage. But, since the technology is moving towards high capacity archival read only memories this should pose no problem. For validation purposes, only two variables are kept in the given test examples and up to 3rd order interpolation is used to interpolate for the desired stiffness matrices. A similar treatment is possible for computing the mass matrix; however, this has not been dealt with here.

The stiffness matrix, when computed for specific values of the control parameters is in equilibrium. For the interpolated stiffness, however, no such claim can be made. As a matter of fact, in all test examples the interpolated stiffness matrices did not fully satisfy the equilibrium conditions. All of them would produce noticeable residual moments when subjected to a rotational rigid body motion. To remove these residual moments two formulations were exercised. In the first formulation, the residual forces are expressed in terms of basis vectors obtained using the Gram-Schmidt orthogonalization procedure. A correction matrix, which represents the out of equilibrium force components, is obtained by congruent transformation of the basis vector matrix on an appropriate coefficient matrix. The correction matrix is further subtracted from the interpolated matrix to obtain a modified stiffness
matrix which satisfies the equilibrium conditions. However, it requires the inversion of the interpolated stiffness matrix which may be singular if one or more rigid body motions are present initially. The second formulation explicitly includes the missing rigid body modes by using a hypermatrix scheme given by Cantin[1]. In addition, it only requires the inversion of an $n \times n$ matrix where $n$ is the number of missing rigid body modes.

In order to compute the stress interpolation matrix, a unit displacement is given at each degree of freedom associated with the master nodes of the substructure separately, and the stress coefficients are computed for selected elements, where high stresses are expected. An approximate least square polynomial fit is utilized to interpolate for the values of the desired stresses in the selected elements as a function of the control parameters.

To validate the formulation, several test examples were investigated. The results obtained are encouraging and show good agreement with the values obtained using the regular finite element. To select a class of problems that can be handled with the proposed method, the web frame of a ship was analyzed completely. The results obtained are compared against the regular Finite Element solutions and show good agreement from a practical design point of view. In addition, a comparative study of the CPU (Central Processing Unit) time taken by the proposed method and the regular finite element method for the web frame shows the flexibility and efficiency of the proposed method relative to the regular finite element method.

To generate substructure stiffness matrices, a general purpose finite element program GIFTS (Graphics oriented Interactive Finite element Total Software) was used. The GIFTS Program was also used to analyze the web frame model of a ship. The MATPAK Program was used for all matrix computations required in obtaining the approximate stiffness matrices.
CHAPTER 1

DISCUSSION OF SOME OF THE BASIC REQUIREMENTS FOR A WELL BEHAVED FINITE ELEMENT MODEL

In the proposed formulation, after interpolating the desired stiffness matrix from the sample matrices computed for specific combinations of control parameters, a check must be performed to verify whether the interpolated stiffness matrix satisfies all of its essential properties. In this chapter a brief discussion illustrates some of these properties for a well behaved finite element model. Such considerations are applicable, with minor extensions, to substructures and constrained substructures.

In the 1960's, when the Finite Element Method was gaining recognition in the field of Structural Mechanics, there were fairly strict requirements for a well behaved finite element model. For example, Cantin¹ listed the following set of properties which a displacement function – used to construct the stiffness matrix of a finite element – should possess.

(1) Infinitesimal rigid body motions should be accurately represented. If this requirement is not met the conditions of equilibrium are not satisfied.

(2) The displacement function should contain all the lower order polynomial terms of a complete set of functions. The behavior functions, satisfying this requirement, ensure monotonic convergence by mesh size reduction.

(3) A minimum degree of interelement continuity must be maintained between adjacent elements. This minimum degree of compatibility must ensure the continuity, required for the existence of the highest order derivative, present in the variational function representing the total potential energy of the system. A behavior function, satisfying this requirement, ensures convergence to an ‘exact’ result by mesh size reduction.
**Rigid Body Motion Criterion**

The first requirement dealing with the representation of the rigid body motion without any strains and stresses, is widely accepted as a rule for choosing an appropriate displacement function in a well behaved finite element. In the proposed formulation, the interpolated stiffness matrix is tested against this requirement. The interpolated stiffness matrix does not satisfy the rigid body motion criterion completely. As a result, the stiffness matrix is further reconditioned using the formulation given in chapter 3.

**Interelement Compatibility Criterion**

The second and third requirements, stated before, essentially state the interelement compatibility condition. In a substructure that interfaces with an element, the compatibility requirement is satisfied by employing appropriate boundary constraints consistent with the interface element. For example, a substructure boundary, interfaced with a beam element, must deform as a rigid line (see fig(1.1)) because the cross-section of a beam remains a plane under deformation. Various types of boundary interpolation, used to describe the boundary behavior of a substructure, are described in chapter 4.

![Substructure Beam](image)

**Fig(1.1)** A constrained substructure interface with a beam element
In the proposed method, the substructure stiffness matrix is approximated using interpolation and is further reconditioned to satisfy the equilibrium conditions which might disturb the boundary behavior of a constrained substructure. Hence the interelement compatibility criterion might not be fully satisfied by the interpolated stiffness matrices. However, since all the sample matrices of a family are generated with same boundary behavior, the interpolated stiffness matrix is expected to behave close to the sample stiffness matrices.

In addition, the interelement compatibility condition, have gone through extensive research and experimentation by many scholars in the field since then. Many views have been presented regarding the relaxation of the absolute satisfaction of the requirement. The following discussion is made to support the relaxation in the absolute satisfaction of the interelement compatibility condition.

The full mathematical basis for the interelement compatibility criterion has been illustrated by many researchers\cite{2,3,4}. However, they also have mentioned difficulties in the construction of behavior functions satisfying the interelement continuity requirements. In addition, even if such behavior functions are possible to construct, the resulting element is often too expensive to compute, e.g., curved beam, shell elements.

Another requirement, which should be met by behavior functions, is that the assumed function should include representation of constant values of pertinent stresses or strains. The requirement was first stated by Bazeley et al. in 1965\cite{5}. This condition ensures the convergence to the 'exact' solution with the mesh refinement because after a considerable refinement the element will be so small that a constant strain or stress condition will exist. Obviously if the requirement is not satisfied then the solution may converge to incorrect values.
Zienkiewicz\textsuperscript{[6]} points out that the discontinuity of displacements would cause infinite strains at the interface. However, if in the limit, as the size of the subdivision decreases continuity is restored, then the formulation already obtained will still tend to the correct answer. This condition is always reached if the constant strain condition automatically ensures displacement continuity.

To test such a continuity condition Irons\textsuperscript{[7]} introduced the Patch Test. In this test nodal displacements corresponding to any state of constant strain are imposed on an arbitrary patch of elements. If nodal equilibrium is simultaneously achieved without the imposition of external nodal forces with a state of constant stress then clearly no external work has been lost through interelement discontinuities. An element which passes such a Patch Test will converge. Indeed many nonconforming elements have been formulated which show superior performance to conforming elements. In such elements, however, it is not possible to predict the bounds on the functional or its solution. The validity of the relaxation of properties 2 and 3, which require the interelement compatible displacement function for monotonic convergence, have been clearly demonstrated by many researchers\textsuperscript{[5],[6],[7]}. 
CHAPTER 2

INTERPOLATION SCHEMES

In the proposed method, sample stiffness matrices are computed using the regular finite element approach for all the required sample cases for a certain order of interpolation. Each geometric control parameter is treated as one parameter of an interpolation polynomial. The order of interpolation for a specific parameter depends on the nature of the dependence of the stiffness matrix upon that specific parameter. For example, in a 2-D membrane substructure the stiffness matrix is a scalar function of the thickness of the membrane for which a linear interpolation is sufficient for exact representation. On the other hand, the stiffness matrix behaves as a higher order inverse function to the length and width of the sides of a membrane. Consequently a higher order polynomial interpolation is required to obtain a reasonably close approximation to the actual value of a stiffness matrix while interpolating for the length or width parameters of a membrane. However, the behavior of such an inverse function, as is observed in the case of the length or the width parameters of a membrane, cannot be represented exactly using the polynomial interpolation. A compromise has to be made for the order of interpolation for such parameters which have a complex influence on the stiffness matrix, because the higher the order of interpolation the more is the number of required sample stiffness matrices.

An interpolation scheme also assumes that a range is specified for each parameter within which all the practical cases are expected to lie. The range of parameters also dictates the locations at which sample cases are to be computed for
a certain order of interpolation. Amongst the different combinations of locations on the parameter's specified range, the selection of equally spaced points tends to minimize maximum possible errors for an arbitrary value of the parameter within the specified range. Some illustrative cases are also shown in chapter 5 in support of the above mentioned argument.

Several different orders of polynomial interpolation were tested to approximate the stiffness matrix and a comparative study of the accuracy of the results with different orders of interpolation has been performed. A general description of the basic formulation behind the different polynomial interpolations is discussed in this chapter.

**Polynomial Interpolation**[8]

The selection of the order of an interpolation polynomial is based on the following reasoning.

A complete $n^{th}$ order polynomial in $x$ can be written as

$$f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_n x^n \equiv P_n$$

(2.1)

where $P_n$ is the function to be approximated and $a_0, a_1, a_2, \ldots, a_n$ are $n+1$ unknown coefficients. Given $n+1$ paired values $(x_i, f(x_i)), i = 0, 1, 2, \ldots n$, perhaps the most obvious criterion for determining the coefficients of $P_n(x)$ requires that

$$P_n(x_i) = f(x_i), \quad i = 0, 1, 2 \ldots n$$

(2.2)

Thus the $n^{th}$ degree polynomial must reproduce $f(x)$ exactly for $n+1$ arguments $x = x_i$. This criterion seems pertinent since (from a fundamental theorem of Algebra) there is one and only one polynomial of degree $n$ or less which assumes specified values for $n + 1$ distinct arguments.
In the proposed method for interpolation purposes the interpolating parameter has been normalized and varies from $-1$ to $1$ over the specified range of the parameter. Some sample polynomial interpolations are given below.

**Linear Interpolation**

Linear interpolation requires two sample points [see fig(2.1)] at

$$f(x)|_{x=-1} = p$$

$$f(x)|_{x=1} = q$$

Here $p$ and $q$ are matrices, hence the interpolation applies to all the elements of the matrices.

$$f(x) = a_0 + a_1 x \quad (2.3)$$

Thus

$$p = a_0 - a_1 \quad (2.4a)$$

$$q = a_0 + a_1 \quad (2.4b)$$

or

$$a_0 = \frac{1}{2}(p + q) \quad (2.5a)$$

$$a_1 = \frac{1}{2}(q - p) \quad (2.5b)$$

hence

$$f(x) = \frac{1}{2}(p + q) + \frac{1}{2}(q - p)x \quad (2.6a)$$

$$f(x) = q \frac{(1 + x)}{2} + p \frac{(1 - x)}{2} \quad (2.6b)$$
Fig (2.1) Linear Interpolation

Fig (2.2) Quadratic Interpolation

Fig (2.3) Cubic Interpolation
Quadratic Interpolation

The polynomial used for quadratic interpolation is given by

\[ f(x) = a_0 + a_1x + a_2x^2 \]  \hspace{1cm} (2.7)

Three sample points are required for a quadratic interpolation (see fig(2.2)).

\[ f(x)|_{x=-1} = p \]
\[ f(x)|_{x=0} = q \]
\[ f(x)|_{x=1} = r \]

Substituting the above listed values of the function in the equation of the polynomial, solving for constants and substituting in equation (2.7) we get

\[ f(x) = p\left(\frac{-x}{2} + \frac{x^2}{2}\right) + q(1 - x^2) + r\left(\frac{x}{2} + \frac{x^2}{2}\right) \]  \hspace{1cm} (2.8)

Cubic Interpolation

For cubic interpolation four sample points are required (fig(2.3)). The cubic polynomial can be written as

\[ f(x) = a_0 + a_1x + a_2x^2 + a_3x^3 \]  \hspace{1cm} (2.9a)

sample values and their locations are as follows

\[ f(x)|_{x=-1} = p \]
\[ f(x)|_{x=-\frac{1}{3}} = q \]
\[ f(x)|_{x=\frac{1}{3}} = r \]
\[ f(x)|_{x=1} = s \]

Solving for coefficients and substituting their values in the polynomial expression we get

\[ f(x) = p\left(\frac{-1 + x + 9x^2 - 9x^3}{16}\right) + q\left(\frac{9(1 - 3x - x^2 + 3x^3)}{16}\right) + \\
\frac{9(1 + 3x - x^2 - 3x^3)}{16} + s\left(\frac{-1 - x + 9x^2 + 9x^3}{16}\right) \]  \hspace{1cm} (2.10)
The concept of polynomial interpolation can be easily generalized to n dimensional interpolation. Two 2-dimensional interpolation cases are discussed here in full detail.

**Bilinear Interpolation**

The interpolation polynomial used for the bilinear interpolation can be written in general as

\[ f(x, y) = (a_1 + a_2 x)(a_3 + a_4 y) \]  

(2.11)

four sample values are required to find four constants in the interpolation polynomial (fig(2.4)).

Locations and sample values are given as

\[
\begin{align*}
&f(x, y)|_{(-1,-1)} = p \\
&f(x, y)|_{(1,-1)} = q \\
&f(x, y)|_{(1, 1)} = r \\
&f(x, y)|_{(-1, 1)} = s
\end{align*}
\]

Solving for constants and substituting their values in the interpolation polynomial we get

\[ f(x, y) = p \frac{(1 - x)(1 - y)}{4} + q \frac{(1 + x)(1 - y)}{4} + r \frac{(1 + x)(1 + y)}{4} + s \frac{(1 - x)(1 + y)}{4} \]  

(2.12)

**Biquadratic Interpolation**

For biquadratic polynomial interpolation the polynomial can be given by

\[ f(x, y) = (a_1 + a_2 x + a_3 x^2)(a_4 + a_5 y + a_6 y^2) \]  

(2.13)
Fig (2.4) Bilinear Interpolation

Fig (2.5) Biquadratic Interpolation
To calculate the 9 coefficients of the interpolating polynomial, 9 sample values are required (fig 2.5). The sample values and their respective normalized coordinates are as follows:

\[ f(x, y)|(-1, -1) = p_1 \]
\[ f(x, y)|(0, -1) = p_2 \]
\[ f(x, y)|(1, -1) = p_3 \]
\[ f(x, y)|(-1, 0) = p_4 \]
\[ f(x, y)|(0, 0) = p_5 \]
\[ f(x, y)|(1, 0) = p_6 \]
\[ f(x, y)|(-1, 1) = p_7 \]
\[ f(x, y)|(0, 1) = p_8 \]
\[ f(x, y)|(1, 1) = p_9 \]

(2.14)

Solving for unknown coefficients and substituting their values in the interpolating polynomial we get:

\[ f(x, y) = p_1 \frac{xy}{4}[1-x][1-y] + p_2 \frac{y}{2}[-1+y][1-x^2] + p_3 \frac{xy}{4}[1+x][-1+y] + p_4 \frac{x}{2}[-1+x][1+y^2] + p_5 \frac{x^2}{2}[1-x][1+y^2] + p_6 \frac{x}{2}[1-x][1-y^2] + p_7 \frac{xy}{4}[-1+x][1+y] + p_8 \frac{y}{2}[1+y][1-x^2] + p_9 \frac{xy}{4}[1+x][1+y] \]

(2.15)

**Polynomial Regression Using the Method of Least Squares**  

The method used for the stress calculation in a specific element of a substructure involves calculating the stresses in the element for every combination of control parameters and in each possible straining mode based on the degree of
freedom of master nodes. To obtain the stresses for the desired geometry of the substructure, a polynomial regression is performed using the method of least squares. The following discussion highlights the method of least squares from the theoretical and programming points of view.

If given m pairs of values \((y_i, x_i), i = 0, 1 \ldots m\), the relationship between the two parameters is approximately given by the expression

\[ y = \alpha + \beta x \]  

(2.16)

with the constant variance of \(\sigma^2\), then in the small interval of extent \(\Delta y\), the probability of observing the value \(y_1\) is, from the normal distribution

\[ Pr(y_1) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y_1 - \alpha - \beta x_1)^2}{2\sigma^2} \Delta y} \]

\[ \approx k e^{-\frac{(y_1 - \alpha - \beta x_1)^2}{2\sigma^2}} \]

(2.17)

where \(k\) is a constant. Similar expressions hold true for \(Pr(y_2) \ldots Pr(y_n)\), where \(m\) is the number of data points or observations. The probability \(P\) of all these values of \(y\) occurring simultaneously is

\[ P = Pr(y_1)Pr(y_2) \ldots Pr(y_n) = k^m e^{-\sum_{i=1}^{m} \frac{(y_i - \alpha - \beta x_i)^2}{2\sigma^2}} \]

(2.18)

The regression will be best if \(P\) is maximum, which will occur when \(\alpha\) & \(\beta\) are chosen to minimize the sum of squares

\[ S = \sum_{i=1}^{m} (y_i - \alpha - \beta x_i)^2 \]

(2.19)

for \(S\) to be minimum

\[ \frac{\partial S}{\partial \alpha} = \frac{\partial S}{\partial \beta} = 0 \]

(2.20)
and $\alpha$ and $\beta$ will be replaced by their estimates $a$ and $b$. Hence

\begin{align*}
\sum (y - a - bx) &= 0 \\
\sum x(y - a - bx) &= 0
\end{align*}
(2.21)

or

\begin{align*}
\sum y &= ma + b \sum x \\
\sum xy &= a \sum x + b \sum x^2
\end{align*}
(2.22)

or in matrix form the set of simultaneous equations can be expressed as

\begin{equation}
\begin{bmatrix}
\sum m \\
\sum x \\
\sum x^2
\end{bmatrix}
\begin{bmatrix}
a \\
b
\end{bmatrix}
= 
\begin{bmatrix}
\sum y \\
\sum xy
\end{bmatrix}
(2.23)
\end{equation}

In general, the concept of least squares can be generalized for an $n^{th}$ order polynomial regression in $x$, such as

\begin{equation}
y = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_n x^n
(2.24)
\end{equation}

The resulting simultaneous equations can be written in matrix form as follows

\begin{equation}
\begin{bmatrix}
m \\
\sum x \\
\sum x^2 \\
\sum x^3 \\
\vdots \\
\sum x^{n+n}
\end{bmatrix}
\begin{bmatrix}
a_0 \\
a_1 \\
a_2 \\
\vdots \\
a_n
\end{bmatrix}
= 
\begin{bmatrix}
\sum y \\
\sum xy \\
\sum x^2 y \\
\vdots \\
\sum x^n y
\end{bmatrix}
(2.25)
\end{equation}

The concept also can be extended to the multivariable polynomial regression along the same lines. For example:

If the polynomial used for regression is given by

\begin{equation}
T = a_1 + a_2 x + a_3 y + a_4 xy + a_5 x^2 + a_6 y^2
(2.26)
\end{equation}
then if we express the polynomial as

\[ T = \sum_{i=1}^{n} a_i D_i \]  

(2.27)

where for the given case

\[ D_1 = 1 \]
\[ D_2 = x \]
\[ D_3 = y \]
\[ D_4 = xy \]
\[ D_5 = x^2 \]
\[ D_6 = y^2 \]

then

\[ C_{ij} = \sum D_i D_j \]  

(2.28)

\[ F_i = \sum T D_i \]  

(2.29)

Hence the coefficients of the polynomial are given by

\[ A = C^{-1} F \]  

(2.30)

where \( C \) is also a symmetric matrix.
CHAPTER 3

STIFFNESS MATRIX RECONDITIONING SCHEMES

In the proposed formulation, an interpolation is required on the sample matrices to obtain a stiffness matrix for the desired set of control parameters. A stiffness matrix satisfies equilibrium conditions when computed for the sample values of control parameters using the regular finite element method. In other words, the stiffness matrix represents all the rigid body motions without causing any residual forces in the substructure. However, once the interpolation is performed no such claim can be made. As a matter of fact, in all the test examples, the interpolated stiffness matrix shown residual forces when subjected to a rigid body rotation.

An interesting phenomenon of the interpolated stiffness matrix was observed while verifying the rigid body motion condition. It was noticed that the interpolated matrix did satisfy the zero residual force condition when subjected to rigid body translation; however, it did not satisfy the same condition when subjected to rigid body rotation for a two-dimensional membrane substructure case. A possible reason for such behavior of the interpolated stiffness matrix could be the fact that the same interpolation is performed for all elements of the matrix. In case of rigid body translation, the sum of all the horizontal or vertical forces is required to be zero to satisfy equilibrium. In an interpolation, all the elements of a sample matrix are scaled by a certain constant which does not affect the element ratios of that matrix. Consequently, the rigid body translation is perfectly represented by the interpolated matrix because all the sample matrices have this common characteristic. However, the rigid body rotation requires the sum of moments to be
zero. The magnitudes of the moments depend on the geometry of the substructure, hence different sample matrices computed for different geometric parameters possess different relationships amongst elements of the matrix resulting in the non-satisfaction of the condition of moment equilibrium. As a result, nothing can be predicted specifically about the resulting interelement relationship for the moment equilibrium in the interpolated stiffness matrix when sample stiffness matrices are linearly added to obtain the desired interpolated stiffness matrix. Consequently, the interpolated stiffness matrix results in residual forces, as observed in all the test examples, when subjected to a rigid body rotation.

To recondition the interpolated matrix to satisfy the equilibrium conditions, or the rigid body motion condition, two schemes were used. A detailed description of the two schemes with their relative advantages and disadvantages is given in this chapter.

**Scheme 1**

In the first scheme for reconditioning the interpolated stiffness matrix, \( K \) \((n \times n)\) is the interpolated stiffness matrix, \( T \) \((n \times m)\) is a set of nodal displacements corresponding to all possible rigid body motion. The transformation matrix \( T \) has as many columns as there are independent rigid body modes \((m)\). \( \psi \) \((n \times m)\) is a matrix in which each column gives the residual forces corresponding to the rigid body mode given by the respective column of the transformation matrix \( T \). Mathematically,

\[
K T = \psi = V C \tag{3.1}
\]

where \( V \) has the property

\[
V^T V = I
\]
and may be easily found using the *Gram-Schmidt orthogonalization* procedure applied to $\psi$. As a result, one may write

hence

$$T = K^{-1}VC$$

(3.2)

$$V = KTC^{-1}$$

(3.2a)

Here the $\psi$ matrix is represented in terms of the basis vector matrix $V$ obtained using the Gram-Schmidt orthogonalization procedure. Thus, a congruent transformation of the basis vector matrix on an appropriate coefficient matrix should give the correction matrix, representing the out of equilibrium force component. Hence, by setting

$$(K - V\Lambda V^T)T = 0$$

(3.3)

where $\Lambda$ is an unknown $(m \times m)$ matrix. To obtain $\Lambda$, equations (3.1) and (3.2) are substituted in equation (3.3)

$$\begin{align*}
(K - V\Lambda V^T)T &= \psi - V\Lambda V^T T = VC - V\Lambda V^T K^{-1}VC = 0 \\
\end{align*}$$

(3.4)

or

$$C - \Lambda(V^TK^{-1}V)C = 0$$

(3.5)

giving

$$\Rightarrow \Lambda = (V^TK^{-1}V)^{-1}$$

(3.6)

Hence the conditioned matrix is given by

$$K^* = K - V\Lambda V^T$$

(3.7)

where $\Lambda$ is given by the expression in equation (3.6)

To prove that $(K - V\Lambda V^T)T = 0$
Substituting for $A$ in equation (3.3), and using equation (3.2a)

$$[K - VAV^T]T = (K - V(V^TK^{-1}V)^{-1}V^T)T$$

$$= [K - KTC^{-1}(V^TK^{-1}V)^{-1}((C^{-1})^TT^TK)]T$$

$$= [K - KTC^{-1}((C^{-1})^TT^TKK^{-1}KTC^{-1})^{-1}((C^{-1})^TT^TK)]T$$

$$= [K - KTC^{-1}[C(T^TK)^{-1}C^{-1}]^{-1}((C^{-1})^TT^TK)]T$$

$$= [K - KT(T^TK)^{-1}T^TK]T$$

$$= [KT - KT]$$

$$= 0$$

q.e.d.

The scheme essentially involves

1. Gram-Schmidt orthogonalization to obtain the $V$ matrix.
2. Inversion of the interpolated $K$ matrix to compute $A$
3. Computation of the modified stiffness matrix $K^* = K - VA V^T$

A point to be noted here is that we need to calculate $K^{-1}$ to compute $A$ which might not be possible if the interpolated matrix does satisfy any one of the rigid body motion in which case it would be singular. As discussed in the introductory part of this chapter, the interpolated stiffness matrix does satisfy rigid body translation modes, which implies that the interpolated stiffness matrix is singular to begin with. Hence the scheme fails in such cases where the interpolated matrix is singular.

**Scheme 2**

This scheme of reconditioning the interpolated matrix was first proposed by Cantin$^{[1]}$ in 1970 for the explicit inclusion of missing rigid body modes in the
formulation of a shell element. In this scheme, the nodal displacements are expressed as a linear combination of displacements due to straining modes and displacements due to rigid body modes as following

\[
\{u\} = \{ U | T \} \begin{bmatrix} u_i \\ u^R_i \end{bmatrix} \tag{3.8}
\]

where \( U \) is a \( n \times n \) unit matrix, \( T \) is a \( n \times m \) transformation matrix relating the nodal coordinates to the \( m \) components representing rigid body motion.

\( u \) is the nodal degrees of freedom (coordinates).

\( u_i \) is the modified set of nodal coordinates.

\( u^R_i \) is rigid body movement vector.

The stiffness matrix is then expanded by the congruent transformation to add all the rigid body modes as follows

\[
\begin{bmatrix} U \\ T^T \end{bmatrix} [K] \begin{bmatrix} U \\ T \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \tag{3.9}
\]

where

\[
k_{11} = K, \quad k_{12} = k_{21}^T = KT, \quad k_{22} = T^T KT \tag{3.10}
\]

The matrix \( k_{22} \) is a \( 3 \times 3 \) matrix for a 2-dimensional case (a maximum size of \( 6 \times 6 \) occurs in the 3-dimensional case). Since rigid body modes should not introduce nodal forces, we set these forces equal to zero and write

\[
\begin{bmatrix} F \\ 0 \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \begin{bmatrix} u_i \\ u^R_i \end{bmatrix} \tag{3.11}
\]

Solving for \( u^R_i \) gives

\[
u^R_i = -[k_{22}]^{-1}[k_{21}]u_i \tag{3.12}
\]

The last expression requires inverting \( k_{22} \). This matrix, however, will be singular if the original stiffness matrix already contains any rigid body modes, in which
case some of the rows and columns will be either zero or linearly dependent on the others. To avoid this difficulty, one can remove those dependent rows and columns from the matrix and use the reduced matrix, which is non-singular. However, the corresponding columns in the \([k_{12}]\) matrix must be removed before constructing the triple product \([k_{12}] [k_{22}]^{-1} k_{21}\), which is required subsequently to obtain the modified stiffness matrix. Solving (3.12) and substituting in (3.11) gives

\[
[F] = ([k_{11}] - k_{12} [k_{22}]^{-1} k_{21}) u_i
\]  
\text{(3.13)}

or

\[
K^* = K - [k_{12}] [k_{22}]^{-1} k_{21}
\]  
\text{(3.14)}

which requires only the inversion of the reduced matrix \([k_{22}]\) and, as discussed above, in case of a singular matrix we can remove dependent or zero rows and columns and obtain a non-singular matrix which can be inverted. If we need to eliminate the \(i^{th}\) row and \(i^{th}\) column from the \([k_{22}]\) matrix then the \(i^{th}\) column of matrix \([k_{12}]\) and the \(i^{th}\) row of matrix \([k_{21}]\) should be removed simultaneously. This scheme has been used to recondition an interpolated stiffness matrix in the proposed formulation.

**A Mathematical Comparison of the Two Schemes**

According to the second scheme, proposed by Cantin\(^2\) to recondition the stiffness matrix, the modified stiffness matrix is given by

\[
K^* = [k_{11}] - [k_{12}] [k_{22}]^{-1} k_{21}
\]  
\text{(3.15)}

where

\[
\begin{align*}
k_{11} &= K \\
k_{12} &= KT \\
k_{21} &= T^T K \\
k_{22} &= T^T KT
\end{align*}
\]
substituting the above values in equation (3.15) gives

\[ K^* = K - [KT](T^TKT)^{-1}[KT]^T \]  (3.16)

Now using the relations according to the first scheme

\[ KT = VC; \quad T = K^{-1}VC \]

also

\[ T^TKT = (KT)^T T = (VC)^T T = C^T V^T K^{-1} VC \]

Hence

\[ K^* = K - VC[C^T V^T K^{-1} VC]^{-1} C^T V^T \]

but

\[ \Lambda^{-1} = V^T K^{-1} V \]

\[ K^* = K - VC[C^T \Lambda^{-1} C]^{-1} C^T V^T \]

\[ = K - V C C^{-1} \Lambda (C^T)^{-1} C^T V^T \]

\[ = K - V \Lambda V^T \]

as obtained with the first scheme. Hence the two schemes result in the same type of modification to the stiffness matrix. However, the second approach does not require the inversion of the illconditioned matrix \( K \) which was the only drawback of the first scheme.
DEFLECTION AND STRESS COMPUTATION METHODS

Deflection Computations

A regular finite element approach has been adopted to compute deflections of a complex structure comprised of substructures, stored in the parametric substructure library. Typically, the following steps are performed in the deflection analysis in the proposed formulation.

1. Model a structure using the substructures which belong to a family that is pre-analyzed and stored in the parametric substructure library.
2. Interpolate each family of required substructures with the selected control parameters.
   (a) Compute the normalized interpolation parameters corresponding to the given values of the control parameters.
   (b) Select the interpolation type and feed the normalized interpolation parameters to obtain coefficients for each sample stiffness matrix of that family.
   (c) Multiply all the coefficients with the corresponding stiffness matrices and add them algebraically to obtain the interpolated stiffness matrix.
3. Recondition each interpolated matrix using Scheme 2, given in chapter 3.
   (a) Form the transformation matrix \( T \) giving the nodal displacements for all the rigid body motions for the substructure.
(b) Compute \( k_{12} = [K]^*[T] \) \& \( k_{22} = T^T K T \), where \( K \) is the interpolated stiffness matrix.

(c) Check for linear dependencies in matrix \( k_{22} \) and remove linear dependent rows and columns from it. Also remove the corresponding columns from the \( k_{12} \) matrix.

(d) Obtain the inverse of reduced matrix \( k_{22} \)

(e) Perform the congruent transformation \( k_{12}[k_{22}]^{-1}(k_{12})^T \) to obtain the correction matrix.

(f) Subtract the correction matrix from the interpolated stiffness matrix \( K \) to obtain the modified stiffness matrix.

\[
K^* = K - k_{12}[k_{22}]^{-1}(k_{12})^T
\]

(4) Assemble the stiffness of all the substructures and form a global stiffness matrix with applied boundary conditions.

(5) Solve for the unknown displacements due to applied forces.

To perform all the matrix operations involved in step 2 and 3, the program MATPAK was used. The modified matrices are then stacked in one large file which is used by the program SOLVE for the deflection calculations under the applied loads. A flow chart of the SOLVE program is given in fig(4.1).

The program SOLVE requires an input file giving the element connectivities, nodal coordinates, material and sizing properties and applied loads. It prepares an output file which contains element deflection information. This file is subsequently modified by the MODD program to obtain a consistant set of deflections for each substructure when the substructure is supported in a statically determinate manner. A flow chart of the program MODD is given in fig(4.2).
Open Input Output Files
- Input file for model description
- Output file for structure deflection and model description information
- A scratch file for storing global stiffness matrix.
- Output file for Element deflection.
- Input file for substructure stiffness matrices.

Read (from model description file)
- Node point data.
- Element data.
- Boundary Condition data.
- Applied Load data
Output read Information

Compute Band Width

Read (from Substructure Stiffness file)
- Stiffness for element IE

Do for all elements

Store element stiffness into Global Stiffness Matrix

Apply boundary conditions

Solve for unknown displacements

Output Deflections
- To output file
- To element deflection file

Echo applied force vector
Stiffness × Deflection

STOP

Fig(4.1) Flow Chart for Program SOLVE
Open input output files
- Input - Element deflection file
- output - Modified element deflection file

READ
- Element number and name
- Deflection of element nodes and their corresponding coordinates

Suppress $u$-translational rigid body motion
$$u_i = u_i - u_1 \quad i=1,2\ldots n$$

Suppress $v$-translational rigid body motion
$$v_i = v_i - v_2 \quad i=1,2\ldots n$$

for all substructure elements

Suppress $\theta$-rotational rigid body motion
$$\theta_i = \theta_i - \theta_1$$
$$u_i = u_i - (y_i - y_1)\theta_1$$
$$v_i = v_i + (x_i - x_1)\theta_1 \quad i = 1,2\ldots n$$

Output modified deflection vector

STOP

Fig(4.2) Flow Chart for Program MODD
The modified deflection set is obtained by removing all rigid body displacement components from the element deflection set obtained by the SOLVE program. It saves computing time in the stress computations subsequently. The stresses are computed and stored for specific elements of a substructure, due to a unit displacement at each one of the allowable degrees of freedom, associated only with the master nodes, acting separately. For example, in a four noded 2-dimensional substructure, there are 12 deflection components (3 at each master node); however, a similar state of strain or stress can be obtained by supporting the substructure in a statically determinate manner, which would have only 9 degrees of freedoms in the present case. Hence, essentially the stress computation for 3 displacement components can be saved for all the sample shapes of the substructure, by removing the rigid body displacement components from the original deflection set.

An example of removing the rigid body displacement components for a 2-dimensional substructure is discussed here in full details.

Assuming that each node has three degrees of freedoms; i.e., $u$, $v$, $\theta$. Thus, there are 12 deflection components to begin with (see fig(4.3)). The original deflection set can be expressed as follows:

$$
\mathbf{u} = [u_1 \ v_1 \ \theta_1 \ u_2 \ v_2 \ \theta_2 \ u_3 \ v_3 \ \theta_3 \ u_4 \ v_4 \ \theta_4]^T
$$

In order to support a 2-dimensional structure in a statically determinate manner we need to support 3 independent degrees of freedoms. Consequently, by making

$$u_1 = 0$$

(to suppress $u$-translation rigid body motion). All the rest of $u$ the displacements would become

$$u_i' = u_i - u_1, \quad i=2,3,4$$
nodal degrees of freedom $(u, v, \theta)$

Fig(4.3) Removal of rigid body displacements
Similarly we can make

$$\theta_1 = 0$$

(to suppress rigid body rotation).

Consequently, all the $\theta$ degree of freedoms would become

$$\theta'_i = \theta_i - \theta_1, \quad i=2,3,4$$

However, suppression of the $\theta$ degree of freedom at node 1 will also affect the $u$ & $v$ displacements at other nodes. The modification is given by the following set of formulae.

If $\theta_1$ is positive counterclockwise (right hand rule)

$$u'_i = u_i + (y_i - y_1)\theta_1$$
$$v'_i = v_i - (x_i - x_1)\theta_1$$

$i = 2,3,4$

Suppressing $v_2$ degree of freedom i.e.

$$v_2 = 0$$ (to suppress rigid body translation in $v$-direction)

the $v$ displacements at other nodes becomes

$$v'_i = v_i - v_2 \quad i=1,3,4$$

With the suppression of $u_1, \theta_1 \& v_2$ we have the equivalent modified set of displacements which has only 9 components as compared to 12 in the original deflection set. Hence

$$u' = [v'_1 \quad u'_2 \quad \theta'_2 \quad v'_3 \quad \theta'_3 \quad u'_4 \quad v'_4 \quad \theta'_4]$$
The concept of modifying the displacement components can be easily extended for 3-dimensional substructures along the same lines as shown for the 2-dimensional case here.

**Dependent Node Deflections in Constrained Substructures**

For stress computations, the method requires the computation of stress values in the desired substructure elements as a result of unit deformations along each degree of freedom associated with the external nodes, and taken separately. Consequently, there will be as many loading cases as there are allowable degrees of freedom at all external nodes. In a typical loading case, only one degree of freedom is given a unit displacement while all the rest of the degrees of freedoms remain suppressed. However, in case of constrained substructures the dependent nodes would also deflect according to the type of boundary constraint specified for the substructure boundaries involving master and dependent nodes. A program INTP was used to interpolate deflection for the following boundary constraints. A flow chart of the program is given in fig(4.4).

Three different types of boundary constraints are described here in full details.

**Rigid Connection**

where dependent nodes on a boundary are rigidly connected to one master node (see fig(4.5)).

\[
\begin{align*}
u_i &= u_0 - (y_i - y_0) \theta_{x_0} + (z_i - z_0) \theta_{y_0} \\
v_i &= v_0 + (x_i - x_0) \theta_{z_0} - (z_i - z_0) \theta_{x_0} \\
w_i &= w_0 + (y_i - y_0) \theta_{z_0} - (x_i - x_0) \theta_{y_0}
\end{align*}
\]
Open input output files

READ
- Dimension of space
- Number of degrees of freedom per node
- Line interpolation type
- Number of equally spaced points on the line
- Coordinates of the master nodes
- Deflections of the master nodes

For cubic interpolation obtain transformation matrix to get a local system having x - axis parallel to the line

Transform deflection to local system by premultiplying with Transformation Matrix

Branch off for appropriate line interpolation.

Perform line interpolation
Output interpolated deflections

STOP

Fig (4.4) Flow Chart For Program INTP
**Fig (4.5)** Rigid Connection

**Fig (4.6)** Linear Boundary (Linear Interpolation)

**Fig (4.7)** Cubic Interpolation
Linear Boundary (Linear Interpolation)

where the three displacement components $u, v \& w$ of a dependent node are interpolated linearly from the corresponding freedoms of two master points. The rotations about the two axes perpendicular to the boundary are adjusted to relieve bending due to displacements in the directions which are perpendicular to the boundary (see fig (4.6)).

\[
\begin{align*}
  u_i &= (1 - \alpha)u_1 + \alpha u_2 \\
  v_i &= (1 - \alpha)v_1 + \alpha v_2 \\
  w_i &= (1 - \alpha)w_1 + \alpha w_2 \\
  \theta_{zi} &= (1 - \alpha)\theta_{z1} + \alpha \theta_{z2} \\
  \theta_{yi} &= \frac{(w_1 - w_2)}{L} \\
  \theta_{xi} &= \frac{(v_1 - v_2)}{L} \\
  \alpha &= \frac{x_i}{L}
\end{align*}
\]

Cubic Interpolation

where displacements perpendicular to the line joining the master nodes are interpolated cubically from the corresponding displacements and appropriate rotations of the two master nodes. Only the displacements along the line joining the master nodes and the rotations around that line are interpolated linearly form the corresponding displacements and rotations alone. (See figure (4.7))
\[ u_i = (1 - \alpha)u_1 + \alpha u_2 \]
\[ v_i = (\alpha - 1)^2(2\alpha + 1)v_1 + \alpha(\alpha - 1)^2L\theta_{z1} + \alpha^2(3 - 2\alpha)v_2 - \alpha^2(1 - \alpha)L\theta_{z2} \]
\[ w_i = (\alpha - 1)^2(2\alpha + 1)w_1 - \alpha(\alpha - 1)^2L\theta_{y1} + \alpha^2(3 - 2\alpha)w_2 + \alpha^2(1 - \alpha)L\theta_{y2} \]
\[ \theta_{zi} = (1 - \alpha)\theta_{z1} + \alpha\theta_{z2} \]
\[ \theta_{yi} = \frac{6\alpha(1 - \alpha)}{L}w_1 + (3\alpha - 1)(\alpha - 1)\theta_{y1} + \frac{6\alpha(1 - \alpha)}{L}w_2 + \alpha(3\alpha - 2)\theta_{y2} \]
\[ \theta_{zi} = \frac{6\alpha(1 - \alpha)}{L}v_1 + (3\alpha - 1)(\alpha - 1)\theta_{z1} + \frac{6\alpha(1 - \alpha)}{L}v_2 + \alpha(3\alpha - 2)\theta_{z2} \]
\[ \alpha = \frac{x_i}{L} \]

**Stress Computations**

The stress computation approach is selective in nature. Only the critical elements of a substructure are analyzed to save computation time without loss of significant information. The critical elements, where high stresses are expected, are selected intuitively ahead of time. Stresses on these elements are calculated by deforming each degree of freedom associated with the external nodes of the substructure individually. To obtain stress coefficients for selected elements for all the sample cases, a program STRE was used, which extracts the desired stress values from the GIFTS data base. A flow chart of program STRE is given in fig(4.8). The stress coefficients for these elements are then stored in a file. A polynomial regression is performed using the method of least squares, discussed in chapter 2, to obtain a stress function over the parametric space for each one of the selected elements. The program RESUL was used to obtain the approximate stress function using the method of least squares and to calculate the stresses in the selected elements for the required geometry of the substructure.
Display an appropriate ERROR message if any

STOP

Fig(4.8) Flow Chart of Program STRE
A flow chart of the program RESUL is given in fig(4.9). Hence, the steps
involved in performing the stress analysis in the method, can be summarized as
follows:

1. Intuitive selection of the critical elements where high stresses are expected.
2. Computation of the stress coefficients of all the critical elements for all
different sets of sample control parameters due to a unit displacement at
each degree of freedom, associated with the master nodes.
3. Polynomial regression by using the method of least squares to obtain a
stress interpolation matrix over the parametric space for each one of the
selected elements of a substructure.
4. Computation of the stress coefficients for the desired configuration of the
substructure by substituting in the given values of the control parameters
in the stress function.
5. Computation of the required stress by multiplying the stress coefficients
obtained for the desired geometry by the modified element deflection set
obtained by removing rigid body displacement components.

The first three steps in the stress computation procedure are needed only
once. Once the least square polynomial fit is obtained for the stress of an element as
a function of geometric control parameter, then for subsequent iterations the method
would require only the value of geometrical parameters to obtain the desired stress
for that specific element of the substructure.
Open element stress coefficient file for all substructures for all requested stress values

Read stress coefficients for all sample cases from element stress coefficient file

Input required combination of control parameters

for all substructures

Perform a least square polynomial fit over the parametric space for stress distribution of a selected element for all requested stress values

Obtain the stress values for the given combination of control parameters due to unit displacements at all master node freedoms

Multiply obtained stress values with deflection vector to obtain desired stress value.

Output stresses

STOP

Fig(4.9) Flow Chart for Program RESUL
CHAPTER 5

RESULTS AND DISCUSSION

A number of test cases were analyzed with the proposed parametric substructure formulation. In the first part of this chapter the following performance studies are discussed.

1. Comparative performance of various orders of interpolation.
2. Effect of scaling all parameters on maximum percentage error.
3. Comparative performance of different point selection schemes.
4. Effect of the stretch in the interpolating parameter’s range on the maximum percentage errors.

The second part of the chapter discusses the comparison of deflection and stress values obtained with the proposed method relative to the regular finite element method’s respective values for the test example of the Web Frame of a ship.

In the last part of the chapter, CPU time performance of the proposed formulation is compared with that of the regular finite element method’s performance for the test example of the Web Frame of a ship.

Part 1

Comparative Study of Various Orders of Interpolation

Two illustrative examples were used for this purpose. The first test example is a rectangular beam modelled using four noded quadrilateral membrane elements. The beam behaviour is imposed by making the membrane structure a two noded substructure which has rigid boundary constraint on two opposite boundary lines.
(each one has one master node on it) and a cubic boundary constraint on one of the lines joining the two rigid lines (see fig(5.1a) for further details). The interpolation parameter is chosen to be $a/b$, since scaling the two parameters by the same amount does not affect the maximum percentage error in results within the corresponding scaled range of the interpolation parameter (see study-2). The following orders of interpolation were studied.

1. Linear Interpolation
2. Quadratic Interpolation
3. Cubic Interpolation

Master node (1) was kept fixed and a vertical force of 100 lbs was applied at node (2). The value of the $v$ deflection at node 2 is compared. The linear interpolation gives the least accurate results. The quadratic interpolation gives better results than the linear interpolation results. The cubic interpolation gives the best results amongst the three. These results can be interpreted easily as follows:

With increasing order of polynomial one is essentially considering more terms of the Taylor's series expansion of the function giving the relationship between the stiffness matrix and the interpolation geometric parameter. Also, the number of sample points are more for the higher order interpolation function. Hence, more accurate results are expected within the same range of the interpolation parameters if a higher order interpolation function is used. In this example, the stiffness matrix behaves as a higher order inverse function of the interpolation parameter, i.e., length of a side of a two dimensional substructure. A graph showing the comparison of the performance of the three interpolations is given in fig(5.2).

Another test example of a quarter of a plate with a hole in the center was analyzed to study two-dimensional interpolation performance. The boundary constraints applied on the substructure are:
$P = 100\text{lbs}$

$t = .1''$

$b = 1''$

R - Rigid Connections

C - Cubically Interpolated

• - master node

Fig(5.1a) Test Example - 1
P = 100lbs
\( t = .1'' \)
\( b = 1'' \)

R - Rigid Connections
C - Cubically Interpolated
• - master node

Fig(5.1b) Test Example - 2
Fig(5.2) A Comparative Performance of One-dimensional Interpolations
(1) Lines 1-2 and 4-5 were interpolated as a rigid line.

(2) Lines 2-3 and 3-4 were cubically interpolated.

The substructure was restricted to move in the y-direction at line 1-2 and in the x-direction at line 4-5 to simulate symmetry boundary conditions. Two concentrated loads of 100 lbs. each were applied at nodes 2 and 3 in the x-direction and the u deflection at node 2 is compared (fig(5.1b)).

The following types of interpolation were tested.

(1) Bilinear Interpolation

(2) Biquadratic Interpolation

Once again the better performance of the biquadratic interpolation can be interpreted in just the same way as the better performance of the cubic interpolation was interpreted in the case of one-dimensional interpolation. A table showing the comparative performance of the two interpolation types is shown in fig(5.3).

Effect of Scaling of Geometric Parameters

The effect of scaling all geometric parameters on the maximum percentage error in the results was observed on the two test examples described before, i.e., namely a thin beam and a quarter of a plate with a hole in the center. No significant effect was observed on the maximum percentage error in the results in either case. The two-dimensional membrane elements have the property that the stiffness matrix does not change by scaling all the geometry of the element by a constant. Hence, in the presented example cases, which utilize only substructures made of 2-dimensional membrane elements, the interpolation parameter is kept as a ratio of two sides of the substructure. A graph showing the effect of the scaling of the geometric parameters is given in fig(5.4).
<table>
<thead>
<tr>
<th>Computed Case</th>
<th>Error (Bilinear)</th>
<th>Error (Biquadratic)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>29.4%</td>
<td>0.0%</td>
</tr>
<tr>
<td>2</td>
<td>13.8%</td>
<td>0.3%</td>
</tr>
<tr>
<td>3</td>
<td>31.2%</td>
<td>4.4%</td>
</tr>
<tr>
<td>4</td>
<td>35.8%</td>
<td>4.8%</td>
</tr>
<tr>
<td>5</td>
<td>24.3%</td>
<td>3.1%</td>
</tr>
</tbody>
</table>

Fig (5.3) A Comparative Performance of 2-Dimensional Interpolations
Cubic Interpolation (Test Example - 1)

<table>
<thead>
<tr>
<th>b/10t</th>
<th>Percentage Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.874 %</td>
</tr>
<tr>
<td>2</td>
<td>1.874 %</td>
</tr>
<tr>
<td>5</td>
<td>1.874 %</td>
</tr>
<tr>
<td>10</td>
<td>1.874 %</td>
</tr>
<tr>
<td>20</td>
<td>1.874 %</td>
</tr>
<tr>
<td>50</td>
<td>1.874 %</td>
</tr>
<tr>
<td>100</td>
<td>1.874 %</td>
</tr>
</tbody>
</table>

Test Example - 2

<table>
<thead>
<tr>
<th>Case</th>
<th>% Error (Bilinear)</th>
<th>% Error (Biquadratic)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>r/t = 10</td>
<td>r/t = 20</td>
</tr>
<tr>
<td>1</td>
<td>29.4</td>
<td>29.4</td>
</tr>
<tr>
<td>2</td>
<td>13.8</td>
<td>13.8</td>
</tr>
<tr>
<td>3</td>
<td>31.2</td>
<td>31.2</td>
</tr>
<tr>
<td>4</td>
<td>35.8</td>
<td>35.8</td>
</tr>
<tr>
<td>5</td>
<td>24.3</td>
<td>24.3</td>
</tr>
</tbody>
</table>

Fig (5.4) Effect of Scaling of Geometric Parameters
Comparative Performance Study of Different Point Selection Schemes

For an \( n^{th} \) order interpolation, \( n + 1 \) sample values of the function are required to uniquely determine the polynomial. However, the selection of the sample parametric values within the specified range plays an important role in minimizing the maximum percentage error in the results. The study conducted here analyzes some of the point selection schemes which are shown in fig (5.5).

The equally spaced point selection tends to minimize the maximum percentage error in the results.

Effect of Stretching the Range of the Interpolation Parameters

To verify the effect of stretching the range of interpolation, test example 1 was analyzed with the following ranges of interpolation parameters

Case(1) \( a/b = 5 - 7 \)

Case(2) \( a/b = 4 - 8 \)

Case(3) \( a/b = 3 - 9 \)

Case(4) \( a/b = 2 - 10 \)

The computed case was for \( a/b = 5.5 \)

The study was performed with the linear, quadratic and cubic interpolations. In all the different interpolations, the maximum percentage error in the results tends to increase with the increase in the range of parameters. A graph showing the same effect is shown in fig(5.6).
Test Example – 1

- Sample Case
- Computed Case

<table>
<thead>
<tr>
<th>Scheme No.</th>
<th>Percentage Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.75%</td>
</tr>
<tr>
<td>2</td>
<td>7.39%</td>
</tr>
<tr>
<td>3</td>
<td>4.12%</td>
</tr>
<tr>
<td>4</td>
<td>4.60%</td>
</tr>
</tbody>
</table>

Fig (5.5) A Comparative Performance of Point Selection Schemes
Fig(5.6)  Effect of Stretching the Range of Interpolation
Part 2

To represent a class of problems that can be handled efficiently using the proposed method, the Web Frame of a ship was selected. A library of pre-analyzed substructures was used to model the Web Frame. The different types of substructures used are shown in figs(5.7a,b...j). The Web-Frame model is shown in fig(5.8). There are 10 different substructure types that were used to model the Web Frame. Only $u, v \& \theta$ degrees of freedoms were considered for every node. The loading pattern and boundary conditions are shown in fig(5.8). All the substructure stiffness matrices were generated using the GIFTS program. The modified interpolated stiffness matrices were obtained using the MATPAK program and the program SOLVE was used to solve the deflections.

For stress computations, certain elements were selected in each substructure for which stress values were computed. These elements for each substructure are shown shaded in figs(5.7a,b...j).

A comparison of deflection values obtained using the parametric substructure method and values obtained using the regular finite element method is shown in table(5.1). The percentage error is based on the maximum deflection in the structure.

A comparison of stresses is shown in table (5.2). Once again, the percentage error is based on maximum stress value.

The maximum percentage error in deflection is about 5% and in the stresses is around 5.5% which is acceptable from a practical design point of view. However, the major advantage of this formulation is highlighted in the next section where a comparison of the computing time taken by the two methods is made.
Interpolation Type For Approximating Stiffness Matrix – CUBIC Interpolation

Variable Parameter – a

<table>
<thead>
<tr>
<th>Boundary Interpolation Type</th>
<th>Line</th>
<th>Boundary Interpolation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2-3</td>
<td>Rigid</td>
</tr>
<tr>
<td></td>
<td>1-2</td>
<td>Cubic</td>
</tr>
<tr>
<td></td>
<td>1-4</td>
<td>Rigid</td>
</tr>
</tbody>
</table>

- Sample Cases
  - Computed Case

Fig (5.7a) Constrained Substructure Family – C1
Approximate Stiffness Matrix Interpolation—Biquadratic Interpolation

Variable Parameters—\( a,b \)

- Sample Cases
  - Computed Case

\[ \begin{align*}
  r &= 2' \\
  t &= .1'
\end{align*} \]

<table>
<thead>
<tr>
<th>Boundary Interpolation Type</th>
<th>Line</th>
<th>Boundary Interpolation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2-3</td>
<td>Rigid</td>
<td></td>
</tr>
<tr>
<td>5-6-7</td>
<td>Cubic</td>
<td></td>
</tr>
<tr>
<td>4-5</td>
<td>Rigid</td>
<td></td>
</tr>
<tr>
<td>7-8</td>
<td>Rigid</td>
<td></td>
</tr>
</tbody>
</table>

- element considered for stress computations
- master node

Fig (5.7b) Constrained Substructure Family—\( C_2 \)
Approximate Stiffness Matrix Interpolation – Biquadratic Interpolation

Variable Parameters – a,b

- Sample Cases
  - Computed Case

\[
\begin{array}{c|c|c|c|c|c|}
\text{b/r} & 3.5 & 3 & 2.5 & 1.5 & \\
\hline
\text{a/r} & 3.5 & 2.5 & 1.5 & t = .1' & r = 2'
\end{array}
\]

- Stress element
- Master node

<table>
<thead>
<tr>
<th>Boundary Interpolation Type</th>
<th>Line</th>
<th>Boundary Interpolation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2</td>
<td>Rigid</td>
<td></td>
</tr>
<tr>
<td>2-3</td>
<td>Cubic</td>
<td></td>
</tr>
<tr>
<td>3-4</td>
<td>Cubic</td>
<td></td>
</tr>
<tr>
<td>4-5</td>
<td>Rigid</td>
<td></td>
</tr>
</tbody>
</table>

Fig(5.7c) Constrained Substructure Family – C₃
Interpolation Type For Approximating Stiffness Matrix – CUBIC Interpolation

Variable Parameter – \( a \)

<table>
<thead>
<tr>
<th>Boundary Interpolation Type</th>
<th>Line</th>
<th>Boundary Interpolation</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-3</td>
<td></td>
<td>Rigid</td>
</tr>
<tr>
<td>1-2</td>
<td></td>
<td>Cubic</td>
</tr>
<tr>
<td>1-4</td>
<td></td>
<td>Rigid</td>
</tr>
</tbody>
</table>

- Sample Cases
  - Computed Case

\[ b = 4' \]
\[ t = .1' \]

\( \square \) element considered for stress computations

- master node

Fig(5.7d) Constrained Substructure Family – \( C_4 \)
Approximate Stiffness Matrix Interpolation - Biquadratic Interpolation

Variable Parameters - a, b

- Sample Cases
  - Computed Case

\[ r = 2' \]
\[ t = 1' \]

\[ \square \text{ stress element} \]
\[ \bullet \text{ master node} \]

<table>
<thead>
<tr>
<th>Line</th>
<th>Boundary Interpolation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2-3</td>
<td>Rigid</td>
</tr>
<tr>
<td>5-6-7</td>
<td>Cubic</td>
</tr>
<tr>
<td>4-5</td>
<td>Rigid</td>
</tr>
<tr>
<td>7-8</td>
<td>Rigid</td>
</tr>
</tbody>
</table>

Fig(5.7e) Constrained Substructure Family - \( C_5 \)
Approximate Stiffness Matrix Interpolation—Biquadratic Interpolation

Variable Parameters – a, b

• Sample Cases
  o Computed Case

\[ r = 2' \]
\[ t = .1' \]

stress element

• master node

<table>
<thead>
<tr>
<th>Line</th>
<th>Boundary Interpolation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2</td>
<td>Rigid</td>
</tr>
<tr>
<td>2-3</td>
<td>Cubic</td>
</tr>
<tr>
<td>3-4</td>
<td>Cubic</td>
</tr>
<tr>
<td>4-5</td>
<td>Rigid</td>
</tr>
</tbody>
</table>

Fig(5.7f) Constrained Substructure Family – C₆
Interpolation Type For Approximating Stiffness Matrix – CUBIC Interpolation

Variable Parameter – a

<table>
<thead>
<tr>
<th>Boundary Interpolation Type</th>
<th>Line</th>
<th>Boundary Interpolation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2–3</td>
<td>Rigid</td>
</tr>
<tr>
<td></td>
<td>1–2</td>
<td>Cubic</td>
</tr>
<tr>
<td></td>
<td>1–4</td>
<td>Rigid</td>
</tr>
</tbody>
</table>

- Sample Cases
  - Computed Case

\[ b = 4' \]
\[ t = .1' \]

\[ \square \] element considered for stress computations

- master node

Fig(5.7g) Constrained Substructure Family – C7
Approximate Stiffness Matrix Interpolation – Biquadratic Interpolation

Variable Parameter – a,b

<table>
<thead>
<tr>
<th>Line</th>
<th>Boundary Interpolation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2-3</td>
<td>Rigid</td>
</tr>
<tr>
<td>5-6-7</td>
<td>Cubic</td>
</tr>
<tr>
<td>4-5</td>
<td>Rigid</td>
</tr>
<tr>
<td>7-8</td>
<td>Rigid</td>
</tr>
</tbody>
</table>

- Sample Cases
  - Computed Case

\[ r = 2' \]
\[ t = .1' \]

\[ \square \] element considered for stress computations

- master node

**Fig(5.7h) Constrained Substructure Family – C_8**
Approximate Stiffness Matrix Interpolation – Biquadratic Interpolation

Variable Parameters – a, b

- Sample Cases
  - Computed Case

\[ \frac{b}{r} = 5 \quad \frac{3r^2}{2} \quad \frac{-a}{r} \]

\( r = 2' \)
\( t = .1' \)

- stress element
- master node

<table>
<thead>
<tr>
<th>Line</th>
<th>Boundary Interpolation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2-3</td>
<td>Rigid</td>
</tr>
<tr>
<td>5-6-7</td>
<td>Cubic</td>
</tr>
<tr>
<td>4-5</td>
<td>Rigid</td>
</tr>
<tr>
<td>7-8</td>
<td>Rigid</td>
</tr>
</tbody>
</table>

Fig (5.7i) Constrained Substructure Family – C9
Interpolation Type For Approximating Stiffness Matrix – CUBIC Interpolation

Variable Parameter – $a$

<table>
<thead>
<tr>
<th>Boundary Interpolation Type</th>
<th>Line</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2-3</td>
</tr>
<tr>
<td></td>
<td>1-2</td>
</tr>
<tr>
<td></td>
<td>1-4</td>
</tr>
</tbody>
</table>

- Sample Cases
  - Computed Case

$\mathbf{b = 4'}$

$\mathbf{t = .1'}$

\(\mathbf{\Box}\) element considered for stress computations

- master node

\textbf{Fig(5.7j) Constrained Substructure Family – C}_{10}
Fig(5.8) Main Model – Web Frame of a Ship
Table (5.1)
Table for Deflection Comparison

<table>
<thead>
<tr>
<th>Node</th>
<th>Dof</th>
<th>FE Value</th>
<th>PS Value</th>
<th>% Diff.*</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>v</td>
<td>-1.975E-03</td>
<td>-2.107E-03</td>
<td>4.43</td>
</tr>
<tr>
<td>2</td>
<td>v</td>
<td>-1.490E-03</td>
<td>-1.587E-03</td>
<td>3.18</td>
</tr>
<tr>
<td>2</td>
<td>$\theta$</td>
<td>9.688E-05</td>
<td>1.038E-04</td>
<td>2.28</td>
</tr>
<tr>
<td>3</td>
<td>u</td>
<td>4.038E-04</td>
<td>4.148E-04</td>
<td>0.36</td>
</tr>
<tr>
<td>3</td>
<td>v</td>
<td>-8.634E-04</td>
<td>-9.292E-04</td>
<td>2.16</td>
</tr>
<tr>
<td>4</td>
<td>u</td>
<td>5.394E-04</td>
<td>5.510E-04</td>
<td>0.38</td>
</tr>
<tr>
<td>5</td>
<td>u</td>
<td>-7.782E-04</td>
<td>8.038E-04</td>
<td>0.84</td>
</tr>
<tr>
<td>5</td>
<td>v</td>
<td>-1.358E-03</td>
<td>-1.448E-03</td>
<td>2.96</td>
</tr>
<tr>
<td>5</td>
<td>$\theta$</td>
<td>1.448E-04</td>
<td>1.517E-04</td>
<td>2.29</td>
</tr>
<tr>
<td>6</td>
<td>u</td>
<td>4.642E-04</td>
<td>4.684E-04</td>
<td>0.13</td>
</tr>
<tr>
<td>6</td>
<td>$\theta$</td>
<td>8.882E-04</td>
<td>9.163E-04</td>
<td>0.92</td>
</tr>
<tr>
<td>7</td>
<td>u</td>
<td>1.216E-03</td>
<td>1.247E-04</td>
<td>1.02</td>
</tr>
<tr>
<td>7</td>
<td>$\theta$</td>
<td>1.577E-03</td>
<td>1.661E-03</td>
<td>2.76</td>
</tr>
<tr>
<td>8</td>
<td>u</td>
<td>2.450E-03</td>
<td>2.580E-03</td>
<td>4.26</td>
</tr>
<tr>
<td>8</td>
<td>v</td>
<td>-1.333E-03</td>
<td>-1.232E-03</td>
<td>3.31</td>
</tr>
<tr>
<td>9</td>
<td>u</td>
<td>2.708E-03</td>
<td>2.842E-03</td>
<td>4.53</td>
</tr>
<tr>
<td>9</td>
<td>$\theta$</td>
<td>9.321E-04</td>
<td>1.052E-03</td>
<td>3.93</td>
</tr>
<tr>
<td>10</td>
<td>u</td>
<td>2.698E-03</td>
<td>2.846E-03</td>
<td>4.53</td>
</tr>
<tr>
<td>11</td>
<td>u</td>
<td>2.825E-03</td>
<td>2.977E-03</td>
<td>4.99</td>
</tr>
<tr>
<td>12</td>
<td>u</td>
<td>3.047E-03</td>
<td>3.187E-03</td>
<td>4.59</td>
</tr>
<tr>
<td>13</td>
<td>u</td>
<td>2.533E-03</td>
<td>2.665E-03</td>
<td>4.33</td>
</tr>
<tr>
<td>13</td>
<td>v</td>
<td>-6.770E-04</td>
<td>-7.190E-04</td>
<td>1.38</td>
</tr>
<tr>
<td>14</td>
<td>u</td>
<td>2.093E-03</td>
<td>2.145E-03</td>
<td>1.72</td>
</tr>
<tr>
<td>15</td>
<td>u</td>
<td>1.753E-03</td>
<td>1.768E-03</td>
<td>0.97</td>
</tr>
<tr>
<td>16</td>
<td>u</td>
<td>1.535E-03</td>
<td>1.566E-03</td>
<td>1.02</td>
</tr>
<tr>
<td>16</td>
<td>$\theta$</td>
<td>9.662E-06</td>
<td>-9.359E-06</td>
<td>0.00</td>
</tr>
<tr>
<td>17</td>
<td>u</td>
<td>1.115E-03</td>
<td>1.114E-03</td>
<td>1.10</td>
</tr>
<tr>
<td>17</td>
<td>v</td>
<td>-5.618E-04</td>
<td>-6.108E-04</td>
<td>1.61</td>
</tr>
<tr>
<td>18</td>
<td>u</td>
<td>1.084E-03</td>
<td>1.108E-03</td>
<td>0.79</td>
</tr>
<tr>
<td>18</td>
<td>v</td>
<td>-8.017E-04</td>
<td>-8.753E-04</td>
<td>2.42</td>
</tr>
<tr>
<td>19</td>
<td>u</td>
<td>4.342E-04</td>
<td>4.326E-04</td>
<td>0.05</td>
</tr>
</tbody>
</table>

* Percentage difference is based on maximum deflection
<table>
<thead>
<tr>
<th>Main model name</th>
<th>Cosub name</th>
<th>Element no.</th>
<th>Stress Component</th>
<th>F.E value (lb/sqin)</th>
<th>P.S value (lb/sqin)</th>
<th>% diff.</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>C6</td>
<td>6</td>
<td>2</td>
<td>3.186E03</td>
<td>3.304E03</td>
<td>3.703</td>
</tr>
<tr>
<td>11</td>
<td>C6</td>
<td>18</td>
<td>2</td>
<td>2.557E03</td>
<td>2.617E03</td>
<td>1.883</td>
</tr>
<tr>
<td>11</td>
<td>C6</td>
<td>24</td>
<td>2</td>
<td>2.312E03</td>
<td>2.361E03</td>
<td>1.538</td>
</tr>
<tr>
<td>11</td>
<td>C6</td>
<td>36</td>
<td>2</td>
<td>1.303E03</td>
<td>1.432E03</td>
<td>4.048</td>
</tr>
<tr>
<td>13</td>
<td>C7</td>
<td>1</td>
<td>1</td>
<td>1.548E03</td>
<td>1.546E03</td>
<td>0.060</td>
</tr>
<tr>
<td>13</td>
<td>C7</td>
<td>10</td>
<td>1</td>
<td>1.355E03</td>
<td>1.343E03</td>
<td>0.370</td>
</tr>
<tr>
<td>13</td>
<td>C7</td>
<td>11</td>
<td>1</td>
<td>2.288E03</td>
<td>2.302E03</td>
<td>0.440</td>
</tr>
<tr>
<td>13</td>
<td>C7</td>
<td>20</td>
<td>1</td>
<td>1.170E03</td>
<td>1.178E03</td>
<td>0.250</td>
</tr>
<tr>
<td>14</td>
<td>C8</td>
<td>1</td>
<td>1</td>
<td>-1.945E03</td>
<td>-2.010E03</td>
<td>2.040</td>
</tr>
<tr>
<td>14</td>
<td>C8</td>
<td>3</td>
<td>1</td>
<td>-1.129E03</td>
<td>-1.201E03</td>
<td>2.250</td>
</tr>
<tr>
<td>14</td>
<td>C8</td>
<td>19</td>
<td>1</td>
<td>9.616E02</td>
<td>1.028E03</td>
<td>2.080</td>
</tr>
<tr>
<td>14</td>
<td>C8</td>
<td>22</td>
<td>1</td>
<td>1.269E03</td>
<td>1.215E03</td>
<td>1.690</td>
</tr>
<tr>
<td>14</td>
<td>C8</td>
<td>24</td>
<td>1</td>
<td>1.959E03</td>
<td>1.938E03</td>
<td>0.660</td>
</tr>
<tr>
<td>15</td>
<td>C7</td>
<td>1</td>
<td>1</td>
<td>1.552E03</td>
<td>1.547E03</td>
<td>0.345</td>
</tr>
<tr>
<td>15</td>
<td>C7</td>
<td>10</td>
<td>1</td>
<td>1.552E03</td>
<td>1.529E03</td>
<td>0.722</td>
</tr>
<tr>
<td>15</td>
<td>C7</td>
<td>11</td>
<td>1</td>
<td>2.156E03</td>
<td>2.178E03</td>
<td>0.690</td>
</tr>
<tr>
<td>15</td>
<td>C7</td>
<td>20</td>
<td>1</td>
<td>2.156E03</td>
<td>2.178E03</td>
<td>0.690</td>
</tr>
<tr>
<td>9</td>
<td>C9</td>
<td>3</td>
<td>1</td>
<td>1.149E03</td>
<td>1.239E03</td>
<td>2.820</td>
</tr>
<tr>
<td>9</td>
<td>C9</td>
<td>4</td>
<td>1</td>
<td>1.408E03</td>
<td>1.574E03</td>
<td>5.210</td>
</tr>
<tr>
<td>9</td>
<td>C9</td>
<td>21</td>
<td>1</td>
<td>1.763E03</td>
<td>1.262E03</td>
<td>3.100</td>
</tr>
<tr>
<td>9</td>
<td>C9</td>
<td>22</td>
<td>1</td>
<td>1.439E03</td>
<td>1.616E03</td>
<td>5.550</td>
</tr>
<tr>
<td>4</td>
<td>C3</td>
<td>6</td>
<td>2</td>
<td>1.453E03</td>
<td>1.349E03</td>
<td>3.200</td>
</tr>
<tr>
<td>4</td>
<td>C3</td>
<td>18</td>
<td>2</td>
<td>1.831E03</td>
<td>1.769E03</td>
<td>1.920</td>
</tr>
<tr>
<td>4</td>
<td>C3</td>
<td>24</td>
<td>2</td>
<td>1.997E03</td>
<td>1.943E03</td>
<td>1.695</td>
</tr>
<tr>
<td>4</td>
<td>C3</td>
<td>36</td>
<td>2</td>
<td>2.185E03</td>
<td>2.039E03</td>
<td>6.580</td>
</tr>
<tr>
<td>2</td>
<td>C2</td>
<td>1</td>
<td>1</td>
<td>1.062E03</td>
<td>1.128E03</td>
<td>2.071</td>
</tr>
<tr>
<td>2</td>
<td>C2</td>
<td>19</td>
<td>1</td>
<td>-1.771E03</td>
<td>-1.842E03</td>
<td>2.228</td>
</tr>
<tr>
<td>2</td>
<td>C2</td>
<td>21</td>
<td>1</td>
<td>-1.237E03</td>
<td>-1.297E03</td>
<td>1.880</td>
</tr>
<tr>
<td>1</td>
<td>C1</td>
<td>11</td>
<td>1</td>
<td>1.133E03</td>
<td>1.145E03</td>
<td>0.370</td>
</tr>
<tr>
<td>1</td>
<td>C1</td>
<td>20</td>
<td>1</td>
<td>1.133E03</td>
<td>1.146E03</td>
<td>0.370</td>
</tr>
<tr>
<td>1</td>
<td>C1</td>
<td>1</td>
<td>1</td>
<td>5.050E02</td>
<td>5.010E02</td>
<td>0.000</td>
</tr>
</tbody>
</table>

Percentage Error is based on maximum stress value.
In order to compare the efficiency of the proposed method relative to the regular finite element method, the control parameters in all the different substructures were changed and the model was analyzed again with the changed geometry (fig[5.9]). On the other hand the same model was analyzed using the regular finite element method with the help of the general purpose finite element program GIFTS (Graphics oriented Interactive Finite element Total Software). The comparison of CPU time taken by the two methods is given in table(5.3), which clearly shows a great saving in computing time when the proposed method was used relative to the time taken by the regular finite element method for analyzing the same model. The errors in maximum deflection and stress is within 6%.

The initial computing time to compute all the sample matrices is quite significant and is directly proportional to the order of interpolation and number of geometric variables. However, in design optimization one requires to change the geometric definitions several times before reaching an optimized combination of them. Consequently, the advantage of large savings in subsequent analysis quickly outweighs the disadvantage of significant initial computing time consumption for setting up sample matrices etc.

Hence the proposed method provides an excellent tool for design optimization of complex structural and mechanical systems and has sound bearings in the practical field wherever the updated design of certain parts of a complex system are analyzed from the design optimization point of view. The study performed validates basic aspects of the formulation; however, to generalize the concept to include several design variables as would be required in real world design optimization problems, extensive validation is needed, which is beyond the scope of this study.
All radii — 2'
Thickness — .1'

Fig(5.9) Changed Geometry of the Main Model
### Table (5.3)

Table for CPU Time Comparison

<table>
<thead>
<tr>
<th>Process</th>
<th>Finite Element Method (CPU)</th>
<th>Parametric Substruture Method (CPU)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Substructure Stiffness Generation</td>
<td>330.232</td>
<td>0.491</td>
</tr>
<tr>
<td>Computation of Deflections</td>
<td>11.122</td>
<td>0.090</td>
</tr>
<tr>
<td>Computation of Stresses</td>
<td>25.200</td>
<td>0.084</td>
</tr>
<tr>
<td>Total Analysis Computations</td>
<td>366.544</td>
<td>0.664</td>
</tr>
</tbody>
</table>
CONCLUSIONS AND PROPOSED RESEARCH

The proposed method provides a powerful tool for design optimization of complex structural and mechanical systems. The method could render potential advantages in applications where the updated designs of certain parts of a complex system are analyzed from the design optimization point of view.

The test examples validate the formulation for 2-dimensional structural systems made up of membrane elements. In addition, a maximum of two geometric control parameters were used in the test examples. The maximum percentage error was within 6%. A great saving in the computational cost and time was observed when the parametric sustructure method was used, compared to the regular finite element method.

However, quite often, several design parameters are involved in a real world design optimization problem. To generalize the method of parametric substructures, to include more geometric variable parameters, extensive checking with more variable geometric parameters and formalization of the method is required. Higher percentage errors in the results are expected with more design variables. A higher order interpolation for approximating the substructure stiffness matrix can render more accurate results; however, it causes large computational cost and time for setting up the library of pre-analyzed subsstructures. A careful study of the computational cost & time, and the accuracy of the results for different orders of interpolation is also needed.

In addition, the method can be utilized in the non-linear elasticity problems. The formulation used for approximating substructure stiffness matrix can as well be used to evaluate the tangential stiffness matrix in non-linear analysis.
The proposed formulation can render a large saving in the computational cost and time. However, to verify the accuracy of the results in the non-linear problems, an extensive validation of the formulation is required on a wide variety of non-linear problems before any computational advantage can be claimed. This would open up a new branch of solution techniques in the non-linear elasticity problem.
APPENDIX (a)

GIFTS analysis procedure for generating reduced substructure matrix.

- **BULKM, EDITM** - Substructure model generation
- **BULKF** - Automatic freedom generation

**Load modification**

- **EDITLB, BULKLB** - Load application (without boundary conditions)
- Bandwidth optimization

**Point modification**

- **DEFCS** - Definition of master and dependent points
- **ADSTIF, ELSTFD, STASD** - Stiffness computation
- **DECOMD** - Decomposition of equilibrium equations
- **REDCSD** - Stiffness and force reduction
APPENDIX (b)

Analysis procedure for computing stress coefficients using GIFTS.

<table>
<thead>
<tr>
<th>BULKM – Substructure model generation</th>
</tr>
</thead>
<tbody>
<tr>
<td>BULKF – Automatic model generation</td>
</tr>
<tr>
<td>EDITLB, BULKLB – Initial displacement application</td>
</tr>
<tr>
<td>OPTIM – Bandwidth optimization</td>
</tr>
<tr>
<td>ADSTIF, ELSTFD, STASD – Stiffness computations</td>
</tr>
<tr>
<td>DECOMD – Decomposition of equilibrium equations</td>
</tr>
<tr>
<td>STRESS – Stress Computation for ordinary finite elements</td>
</tr>
<tr>
<td>selected element information</td>
</tr>
<tr>
<td>STRE – Extraction of stresses for selected elements</td>
</tr>
</tbody>
</table>
APPENDIX (c)

GIFTS analysis procedure for computing main model deflections and stress computations for substructure elements.

- **EDITM** - Main model substructure generation (attach substructures)
- **BULKF** - Automatic freedom generation (ignores substructures)
- **BULKLB, EDITLB** - Generation of loads and boundary conditions (assembles substructure loads)
- **OPTIM** - Bandwidth optimization
- **ADSTIF, ELSTFD, STASD** - Stiffness computation (assembles substructures)
- **DECOMD** - Decomposition of equilibrium equations
- **DEFLD** - Displacement computation (substructure master node displacements)
- **STRESS** - Stress computation for ordinary finite elements
- **RESULT** - Result display
- **LOCALD** - Extraction of master displacements
  Computation of dependent and internal displacements
- **STRESS** - Stress Computation for ordinary finite elements
- **RESULT** - Result display

Main Analysis Procedure

Local Analysis Procedure
LIST OF REFERENCES


74