AN INVESTIGATION INTO ACCELERATION TECHNIQUES FOR SUBSPACE ITERATION

by

Syed Amir Shah, B.Sc., M.Sc.

Thesis submitted to the University of Sheffield
for the degree of Doctor of Philosophy
in the Faculty of Engineering

UNIVERSITY OF SHEFFIELD
Department of Mechanical Engineering

December 1982

For my Parents

"If knowledge is infinite,

then herein lies a finite piece of infinity"

SUMMARY

The mathematical statement of the eigenproblem is deceptively simple and, although the basic theory has been well established for a long time, obtaining an accurate solution still remains far from trivial. The eigenproblem arises in many branches of science. In this study, however, it is considered only in the context of dynamic and buckling analysis.

The genesis of the eigenproblem in dynamic and buckling analysis is considered and a brief survey of popular solution techniques is presented. A most powerful solution technique, namely subspace iteration, which forms the kernal of this study is discussed in some detail.

Various ideas which may accelerate the subspace iteration method are investigated theoretically. These ideas are subsequently converted into algorithms, which are implemented in the form of FORTRAN computer programmes.

The validity and accuracy of the results obtained is tested against known solutions with a satisfactory outcome. The various modifications are then presented with a menu of problems for comparison purposes. This process identifies the 'best' modification and also yields new ideas and insights.

The subsequent investigations lead to the conception of the 'hybrid technique', which employs the best modification in conjunction with the original subspace iteration. The convergence rate and solution time of the hybrid technique compare favourably with those of the original subspace iteration. In fact, for the problem considered, the hybrid technique is always superior to the original subspace iteration.

ACKNOWLEDGEMENTS

The author wishes to thank his supervisor, Dr. C. Patterson, Senior Lecturer in the Department of Mechanical Engineering at the University of Sheffield, for his excellent guidance, expert advice and continuous encouragement throughout this research.

Dr. J. L. Wearing, Lecturer in the Department of Mechanical Engineering at the University of Sheffield, and the members of the Finite and Boundary Element Research Groups are to be thanked for their humorous company and general discussion.

Thanks are also due to the Computing Services staff for their assistance, in particular to Peter Froggatt who was always willing to advise and assist in computing matters.

The financial support of the Science Research Council is gratefully acknowledged.

Finally, the author wishes to thank Pauline Windle for patiently and expertly typing this thesis.

CONTENTS

		Page No.
	Summary	(i)
	Acknowledgements	(ii)
	Contents	(iii)
	List of Figures	(iv)
	List of Tables	(vii)
	Notation	(ix)
CHAPTER 1	INTRODUCTION	1
CHAPTER 2	THE LARGE GENERALISED EIGENPROBLEM	5
2.1	Introduction	5
2.2	Dynamic analysis	5
2.3	Solution techniques	9
	2.3.1 Vector iteration methods	9
	2.3.2 Transformation methods	10
	2.3.3 Characteristic polynomial iteration techniques	12
	2.3.4 Sturm sequence property	13
	2.3.5 Rayleigh-Ritz analysis	13
	2.3.6 Static condensation	16
	2.3.7 Component mode synthesis	16
	2.3.8 Discussion	17
2.4	Buckling analysis	19
CHAPTER 3	SUBSPACE ITERATION	20
3.1	Introduction	20
3.2	Algorithm	20
3.3	Generalised Jacobi iterations	22
3.4	Initial iteration vectors	25
3.5	Numerical considerations	27
	3.5.1 Dimension of the subspace	27
	3.5.2 Convergence	27
	3.5.3 Check calculations	28
	3.5.4 Shifting	30
3.6	Operation Counts	

		Page No.
CHAPTER 4	MODIFICATIONS TO SUBSPACE ITERATION	33
4.1	Introduction.	33
4.2	Method 1	34
	4.2.1 Theroretical considerations	34
	4.2.2 Algorithm	37
	4.2.3 Numerical considerations	39
	4.2.3.1 Numerical stability	39
	4.2.3.2 Convergence	40
	4.2.4 Implementation	40
	4.2.5 Operation counts	41
	4.2.6 Variation of the basic method	42
	4.2.6.1 Algorithm	42
	4.2.6.2 Implementation	43
	4.2.6.3 Convergence	44
	4.2.6.4 Numerical stability	45
	4.2.6.5 Operation counts	46
4.3	Method 2	46
	4.3.1 Theoretical considerations	46
	4.3.2 Algorithm	51
	4.3.3 Numerical considerations	53
	4.3.3.1 Numerical stability	53
	4.3.3.2 Convergence	54
	4.3.4 Implementation	56
	4.3.5 Operation counts	57
CHAPTER 5	NUMERICAL INVESTIGATIONS	58
5.1	Introduction	58
5.2	Philosophical considerations	60
5.3	Comparison with known solutions	63
	5.3.1 Beams	63
	5.3.2 Plates	64
	5.3.3 Discussion	65
5.4	The available options	67

		Page No.
5.5	General comparison of modifications	68
	5.5.1 Beams	69
	5.5.2 Plates	72
	5.5.3 Discussion	72
5.6	Detailed comparison of modifications	78
	5.6.1 Beams	79
	5.6.2 Plates	79
	5.6.3 Rig	83
	5.6.4 Discussion	83
5.7	Varying NIIT	90
5.8	Varying NROOT	93
5.9	Hybrid Technique	97
5.10	Neglecting Gram-Schmidt Ortho-normalisation	114
CHAPTER 6	CONCLUDING REMARKS	116
6.1	Conclusion	116
6.2	Ideas for further research	118
APPENDIX A	CARDANO'S RULE FOR SOLVING CUBIC EQUATIONS	122
REFERENCES		124

LIST OF FIGURES

Figure No.	Title	Page No.
5.1	Cantilevered beam	63
5.2	Cantilevered plate	65
5.3	Cantilevered plane frame	69
5.4	Off-shore structure (Rig)	84

LIST OF TABLES

Table No.	Title	Page No.
3.1	Operation Counts for Subspace iteration solution	32
4.1	Operation Counts for Method 1	41
4.2	Operation Counts for Method 2 and Method 2A	57
5.1	Comparison with known solution for a cantilevered beam	64
5.2	Comparison with known solution for a cantilevered plate	65
5.3	General comparison of the SSP option and modifications for a cantilevered beam	70
5.4	General comparison of the SSP option and modifications for a cantilevered plane frame	71
5.5	Solution characteristic of problem 4	73
5.6	Solution characteristic of problem 13	74
5.7	General comparison of the SSP option and modifications for a cantilevered plate	75
5.8	Solution characteristic of problem 17	76
5.9	Comparison of t_e, t_m and t_I for problem 4	80
5.10	Comparison of t_e, t_m and t_I for problem 13	81
5.11	Comparison of t_e, t_m and t_I for problem 17	82

Table No.	Title	Page No.
5.12	Comparison of t_e, t_m and t_I for problem 18	85
5.13	Comparison of t_e, t_m and t_I for problem 19	86
5.14	Comparison of t_e, t_m and t_I for problem 20	87
5.15	Comparison of the XXSS1 and XXSS2 options	92
5.16	Comparison of the SSP and XXSS options for NROOT = 8	95
5.17	Comparison of the SSP and XXSS options for NROOT = 12	95
5.18	Comparison of the SSP, XXSSH2 and XXSSH1 options for NROOT = 5	100
5.19	Comparison of the SSP, XXSSH2 and XXSSH1 options for NROOT = 8	101
5.20	Comparison of the SSP, XXSSH2 and XXSSH1 options for NROOT = 12	102
5.21	Values of t_e after each iteration for the XXSSH1 and XXSSH2 options	105
5.22	Comparison of the SSP, SXXH1 and SXXH2 options for NROOT = 5	108
5.23	Comparison of the SSP, SXXH1 and SXXH2 options for NROOT = 8	109
5.24	Comparison of the SSP, SXXH1 and SXXH2 options for NROOT = 12	110
5.25	Values of t_e after each iteration for the SXXH1 and SXXH2 options	111
5.26	Comparison of the SSP option and the hybrid techniques for NMOD = 1, 2 and 3	113

NOTATION

All the symbols used in the text are defined as they appear. However, for convenient reference, a list of the principal symbols is presented below. The necessity of requiring additional symbols in a minor context has, on occasion, led to non-uniqueness. However, in such cases, explanatory text is available in order to avoid confusion.

Matrices are denoted by square brackets, e.g. [K], and column vectors by an underscore, e.g. $\underline{v}$. The inverse and transpose of [K] are denoted by $[K]^{-1}$ and $[K]^{T}$ respectively. Differentiation with respect to time is denoted by dots, e.g. $\frac{df}{dt} = \dot{f}$, $\frac{d^2f}{dt^2} = \ddot{f}$, etc.

a	Length
A	Cross-sectional area
b	Width
[C]	Damping matrix
[D]	Diagonal matrix
E	Young's modulus
E_n	Space of dimension, n
h	Thickness
I	Moment of inertia
[I]	Identity matrix
k_{ij}	(i,j)th element of K
[K]	Stiffness matrix
$[\bar{K}]$	Projected (reduced) stiffness matrix
$[K]_G$	Non-linear strain stiffness matrix

$\bar{\ell}, \ell'$	Eigenvalue ratio
[L]	Lower triangular matrix
m_k	Half bandwidth of K
m_{ij}	(i,j)th element of M
[M]	Mass matrix
$[\bar{M}]$	Projected (reduced) mass matrix
N,n	Dimension of full space
NIIT,ℓ	Number of $[K]^{-1}$ [M] operations for iteration (Method 1) Power to which eigenproblem is raised (Method 1A)
NITE	Iteration number
$NITE_c$	Number of iterations for convergence
NMOD	Number of iterations in which the modification is applied
NROOT,r	Number of required eigenvalues
q	Dimension of subspace
[Q]	Eigenvector matrix of projected eigenproblem
$\underline{R}$	Load vector
RTOL	Relative convergence tolerance
$\bar{s}, s_\ell, s_r$	Shift
t	General time variable
t_c	Time taken for convergence
t_I	Time taken for an iteration
t_m	Time taken for the modification
t_e	Time taken for solving the projected eigenproblem
TOL	Convergence tolerance

$\underline{v}$	Eigenvector
[V]	Eigenvector matrix
$\underline{u}, \underline{x}, \underline{y}, \underline{z}$	Vectors as specified
[U], [X], [Y], [Z]	Matrices as specified
α	Optimising factor (Methods 2 and 2A)
δ_{ij}	Kroneker delta
ν	Poisson's ratio
ρ	Density
λ , p	Eigenvalue
[Λ], [P]	Eigenvalue matrix
ω	Angular frequency

CHAPTER 1

INTRODUCTION

The solution of the eigenproblem has been a constant source of interest for the mathematicians and a source of annoyance for practical scientists and engineers. The latter being more interested in the results and not the mechanics of obtaining them. The formulation of the eigenproblem is deceptively simple and the background theory has been known for years yet the challange of obtaining an accurate solution is not easily overcome. The accurate solution of the eigenproblem is an ideal illustration of bridging the gap between classical mathematics and numerical analysis.

The generalised eigenproblem arises in many fields of scientific study. However, this work is only concerned with the solution in the context of dynamic and buckling analysis. Clearly, the discipline in which the eigenproblem is posed, governs the properties of the operators in the formulation.

The ability to carry out structural analysis has improved dramatically in recent years. This has led to a better understanding of the strengths and weaknesses of structures. Consequently, the undertaking of ambitious engineering projects has become possible, since in this age of ever increasing costs an optimal use of materials is required. The 'finite element method' used in conjunction with high speed computers makes possible the dynamic, buckling and earthquake analysis of large and complicated structures[1,2].

The solution of the eigenproblem is also of importance in the design of dynamic components. The analysis of such a component will yield the natural frequency spectrum. It can then be ensured that the working frequency is not in the neighbourhood of a natural frequency. Consider the following problem: the lowest natural frequency of a structure is required to fall within a prescribed range. The structure is then analysed and, if the condition is not satisfied, the system matrices are varied until it is satisfied. In practice, the mass of a structure is far easier to vary than the stiffness.

Generally, in the solution of the eigenproblem, only the few lowest eigenpairs are required. Therefore, the solution methods, which take no advantage of the special properties of the operators or which solve for the complete eigensystem instead of the required few eigenpairs, are inefficient and uneconomical.

When the order of the operators in the eigenproblem is large and only the few lowest eigenpairs are required, approximate solution techniques[3] are employed. Recently, a most powerful numerical tool has been provided for the finite element analyst, namely "the subspace iteration method" for the solution of the large generalised eigenproblem[4,5]. The object of the work is to obtain a practical algorithm by modifying the subspace iteration algorithm and may be stated as follows:

(i) Test the original algorithm

(ii) Look for possible modifications in the original algorithm

(iii) Justify the modifications theoretically

(iv) Programme the modified algorithms

(v) Test the modified algorithms

(vi) Compare the results with the original algorithm.

The following is a general outline of the thesis. The chapter immediately following this introduction gives the mathematical statement of the eigenproblem. A discussion of how the eigenproblem arises in dynamic and buckling analysis is presented and solution methods prior to the subspace iteration method are discussed critically.

The basic groundwork for this study is laid out in Chapter 3, where the original subspace iteration algorithm is presented. The detailed discussion here is a necessary requirement for the subsequent chapter.

In Chapter 4, the theoretical aspects of the modifications to the subspace iteration algorithm are discussed. Two basic modifications are considered along with a few variations on these. The numerical aspects, e.g. stability and convergence of each modification are discussed. The various difficulties arising in the implementation of the modified algorithms are mentioned.

Chapter 5 initially contains a comparison of the results obtained from the original subspace algorithm and known solutions. Subsequently, general and detailed comparisons between the various modifications available are made. The discussion of the results obtained from these lead to the formulation and testing of the 'hybrid

technique. Finally, the results from the hybrid technique are presented and discussed.

The tolerances used in the original algorithm and the modifications are the same. This allows the results obtained and the times used in the solution to be compared directly.

Chapter 6 contains the overall conclusions and implications drawn from this study along with possible avenues of further research.

CHAPTER 2

THE LARGE GENERALISED EIGENPROBLEM

2.1 Introduction

In this Chapter, a discussion of how and where the eigenproblem arises in dynamic and buckling analysis is presented. The properties of the operators are observed and the solution techniques prior to subspace iteration method are discussed.

2.2 Dynamic analysis

For a finite element system in equilibrium the equations of motion may be written as

$$[M]\ddot{\underline{y}} + [C]\dot{\underline{y}} + [K]\underline{y} = \underline{R} \quad .. \quad .. \quad .. \quad (2.1)$$

where [M],[C] and [K] are the mass, damping and stiffness matrices of order n respectively, and $\underline{R}$ is the load vector. The displacement, velocity and the acceleration of the finite element assemblage are denoted by $\underline{y}$, $\dot{\underline{y}}$ and $\ddot{\underline{y}}$ respectively. Clearly, if the inertia, $[M]\ddot{\underline{y}}$, and the damping, $[C]\dot{\underline{y}}$ are neglected, equation (2.1) reduces to

$$[K]\underline{y} = \underline{R} \quad .. \quad .. \quad .. \quad .. \quad (2.2)$$

which is the equation to be solved in a static analysis. Thus, dynamic analysis involves carrying out static analysis at time, t, and taking into account the inertia and damping forces.

For a n-degree-of-freedom system, the equations of motion are a set of n coupled equations which must be solved simultaneously. It is, however, possible to express this set of equations in terms of a different coordinate system

for which they become uncoupled, by employing a linear transformation. An uncoupled system of equations is, in fact, a set of independent equations each of which resembles in structure the equation of motion of a single-degree-of-freedom system. The advantage of uncoupling the equations is that an uncoupled system of equations is considerably easier to solve than a coupled system of equations[6].

If the normal modes of vibration for a multi-degree system are used as generalised coordinates, the equations of undamped motion become uncoupled. This approach requires the solution of

$$[M]\ddot{\underline{y}} + [K]\underline{y} = 0 \qquad .. \quad .. \quad .. \quad (2.3)$$

which is the expression obtained from equation (2.1) when the damping forces are neglected. This is the free vibration problem and has solutions of the form

$$\underline{y} = \underline{v} \sin \omega (t - t_o) \qquad .. \quad .. \quad .. \quad (2.4)$$

where $\underline{v}$ is a vector of order, n, t and t_o are the time variable and constant respectively, and ω represents the frequency of vibration in rad/sec of the vector $\underline{v}$.

Substituting equation (2.4) into equation (2.3) leads to the mathematical statement of the generalised eigenproblem, namely

$$[K]\underline{v} = \omega^2 [M]\underline{v} \qquad .. \quad .. \quad .. \quad (2.5)$$

The solution of this yields the eigenpairs $(\underline{v}_1, \omega_1^2)$, $(\underline{v}_2, \omega_2^2)$.. where the eigenvectors are [M] -orthonormalised such that

$$\underline{v}_i [M] \underline{v}_j = \begin{cases} 1, i=j \\ 0, i \neq j \end{cases} \qquad .. \quad .. \quad .. \quad (2.6)$$

and

$$0 \leq \omega_1^2 \leq \omega_2^2 \leq \ldots\ldots\ldots \leq \quad .. \quad .. \quad .. \quad (2.7)$$

The ith mode shape is given by $\underline{v}_i$ and ω_i^2 is the corresponding frequency of vibration.

Consider now the following definitions.

$$\left.\begin{aligned} [V] &= (\underline{v}_1, \underline{v}_2 \ldots\ldots\ldots \underline{v}_n) \\ [\Lambda] &= \text{diag } (\omega_i^2),\ i = 1, \ldots. n) \end{aligned}\right\} \quad .. \quad (2.8)$$

The n solution of equation (2.5) may now be written as

$$[K][V] = [M][V][\Lambda] \quad .. \quad .. \quad .. \quad .. \quad .. \quad (2.9)$$

Thus, equations (2.6) and (2.9) yield

$$\left.\begin{aligned} [V]^T[M][V] &= [I] \\ [V]^T[K][V] &= [\Lambda] \end{aligned}\right\} \quad .. \quad .. \quad .. \quad .. \quad .. \quad (2.10)$$

where [I] is the identity matrix. Hence, a suitable linear transformation to uncouple the equations is

$$\underline{y}(t) = [V]\underline{x}(t) \quad .. \quad .. \quad .. \quad .. \quad .. \quad (2.11)$$

Substituting from equations (2.10) and (2.11) in equation (2.1) leads to

$$(\ddot{\underline{x}})(t) + [V]^T[C][V]\dot{\underline{x}}(t) + [\Lambda]\underline{x}(t) = [V]^T\underline{R}(t) \quad .. \quad (2.12)$$

The initial conditions on $\underline{x}(t)$ are obtained from equation (2.11), i.e. at time, t = 0,

$$\underline{x}_o = \underline{v}^T[M]\underline{y}_o \quad , \quad \dot{\underline{x}}_o = \underline{v}^T[M]\dot{\underline{y}}_o \quad .. \quad .. \quad (2.13)$$

It is clearly seen from equation (2.12) that if the damping term is neglected, the finite element equilibrium equations are uncoupled.

Generally, the damping matrix cannot be derived explicitly and the damping effects are only included approximately. It is thus reasonable to construct a damping matrix which will include all the required effects, i.e. the overall energy dissipation during the system

response and also allow an effective solution of the equilibrium equations.

In practice, $[V]^T[C][V]$ is assumed to be diagonal, i.e. damping is proportional. Thus, the total damping in the structure is the sum of individual dampings in each mode. Therefore,

$$\underline{v}_i^T[C]\underline{v}_j = \begin{cases} 2\omega_i e_i , & i = j \\ 0 & i \neq j \end{cases} \quad .. \quad .. \quad (2.14)$$

where e_i is the modal damping parameter. Hence, the eigenvectors $\underline{v}_i$ are also [C] - orthogonal and under the transformation given in equation (2.11), equation (2.1) reduces to a system of uncoupled equations.

The main computational effort in dynamic analysis is in the solution of the eigenproblem. If the system matrices are large, it becomes prohibitively expensive to solve for all the eigenpairs. However, experience has shown that generally only the few lowest eigenpairs are required and that the higher eigenpairs may be neglected. The number of eigenpairs sought depends upon the structure, loading conditions and the required accuracy of the analysis. If the required eigenpairs can be obtained with a justifiable amount of effort then the analysis of large structures becomes feasible.

Consider now the operators [K] and [M] from equation (2.5), an important property of these is bandedness, i.e.

$$k_{ij} = 0 , \; j > i + m_k \quad .. \quad .. \quad .. \quad (2.15)$$

where $(2m_k + 1)$ is the bandwidth of [K]. Also, at least one

of the operators [K] and [M] is positive definite. In practice, [K] can always be taken positive definite since by shifting[5] rigid body modes may be removed. In finite element analysis, if a consistent mass formulation is used, then [M] conforms to equation (2.15) and is also positive definite with $m_m = m_k$; here m_m denotes the half-bandwidth of [M]. However, often good accuracy can be obtained by using a lumped mass formulation. In this case [M] is a diagonal, non-negative definite matrix.

2.3 Solution techniques

Since the order of the system matrices is large, the exact solution of the eigenproblem using conventional techniques becomes generally impossible. This is due to the fact that the solution requires more high speed core storage than is available in a reasonable size computer. Thus the use of conventional solution techniques is generally not possible for large systems. Consequently, approximate solution techniques have been developed.

In the following sub-sections, brief, critical summaries of the frequently used conventional and approximate solution techniques is presented.

2.3.1 Vector iteration methods

Vector iteration methods[7-10] are very effective in solving the eigenproblem. However, the eigenvectors are only calculated one at a time. Also, to preserve numerical stability, the eigenvectors must be calculated to high precision. Vector iteration techniques are effectively employed in subspace iteration.

2.3.2. Transformation methods

These methods[7-23] generally suffer from one or more of the following:

(i) Transformation of the eigenproblem to standard form is required. This is not always possible.

(ii) All the eigenpairs are calculated instead of the required few.

(iii) No advantage is taken of the bandedness of the operators.

Consider now briefly a few of the frequently used transformation methods.

(a) The Jacobi method

This method [7-13] was proposed over a century ago for the solution of the standard eigenproblem. Clearly, a detrimental feature of this method is the necessary transformation to the standard eigenproblem. Another undesirable feature is the calculation of all the eigenpairs; apart from this, it is a particularly simple and stable method.

The method basically consists of carrying out a finite number of congruance transformations to diagonalise [K]. This diagonal then contains the eigenvalues and the eigenvectors are the columns of the product of the transformation matrices.

The Jacobi method has been further developed[5] to take into account the generalised eigenproblem. The generalised Jacobi method has been effectively implemented for solving the reduced generalised eigenproblem generated by subspace iteration, see section (3.3).

(b) The Householder - QR - Inverse iteration (HQRI) method

As in (a), the HQRI[7-10, 19-23] method requires initially a transformation to the standard eigenproblem. Once this has been achieved, the method proceeds as follows.

(i) Householder transformations to reduce [K] to tridiagonal form.

(ii) QR iterations to obtain the eigenvalues.

(iii) Inverse iteration to obtain the eigenvectors of the tridiagonal matrix. These are then transformed to obtain the eigenvectors of [K].

Note that all the eigenvalues are calculated. However, only the required eigenvectors need to be calculated.

(c) The Lanczos method

A particularly elegant method from a mathematical viewpoint is that due to Lanczos[7-10, 14-18]. It transforms an arbitrary matrix [A] of order n into tridiagonal form. The idea basically is to employ two arbitrary but not orthogonal vectors in generating two sets of bi-orthogonal vectors using [A]. The conclusion of these sequences leads to a tridiagonal matrix which is similar to [A]. The eigenvalues of the resulting tridiagonal matrix may be obtained in a variety of ways. Note that if [A] is symmetric, only one bi-orthogonal set need be generated.

This method was neglected for many years after it was first proposed[14] due to several numerical instabilities associated with it. Despite the simplicity of the method in exact arithmetic, it was found to be wanting in finite arithmetic where round-off errors caused problems.

Recently, however, this method has gained favour due to the following:

(i) The reinterpretation of the nature of numerical instabilities[16, 17].

(ii) Procedures to minimise the effect of numerical instabilities[7, 15].

Thus, the Lanczos method is considered as a possible candidate for the solution of the large generalised eigenproblem.

There still remains the problem of equal or closely clustered eigenvalues; in this case the eigenpair is not accurately approximated. In general, the accuracy of an eigenpair approximation depends largely upon the spreading of the eigenvalue spectrum. The accuracy of the predicted eigenpair depends also upon the starting vector. If this is orthogonal to some required eigenvector than the corresponding eigenvalue is missed in the eigensolution.

If the truncated Lanczos method is used then the accuracy of the solution improves[18] as the number of generated vectors is increased.

2.3.3 Characteristic polynomial iteration techniques

It should be noted that polynomial iteration techniques[7-10, 24] yield only the eigenvalues, the eigenvectors have to be calculated separately. Explicit polynomial iteration requires the expansion of a determinant which, for a matrix of order n, means about n! operations and is, therefore, impractical. Implicit polynomial iteration requires several triangular factorisations[5] for each eigenvalue, each triangular factorisation of [K] say, consists of about $\frac{1}{2}nm_K$ operations. Therefore, the time

taken for triangular factorisation increases rapidly with the order of the operators. This technique is useful, however, when used in conjunction with inverse iteration.

2.3.4 Sturm sequence property

This technique[7-10, 25-27] suffers from the same defects that were mentioned with regard to implicit polynomial iteration. When used in conjunction with other techniques, this is a very useful property.

2.3.5 Rayleigh-Ritz analysis

The Rayleigh-Ritz[5,7-10,28] approach is the basis of a lot of approximate solution techniques. The two methods discussed subsequently can be shown to be Ritz analyses. For a general discussion, consider the following

$$[A]\underline{v} = \lambda[B]\underline{v} \quad .. \quad .. \quad .. \quad .. \quad (2.16)$$

The operators [A] and [B] are assumed to be positive definite and defined in an n-dimensional space E_n. The Rayleigh minimum principal states that

$$\lambda_1 = \min \rho(\underline{v}) \quad .. \quad .. \quad .. \quad .. \quad (2.17)$$

where $\rho(\underline{v})$ is the Rayleigh quotient and the minimum is taken over all possible vectors $\underline{v}$

$$\rho(\underline{v}) = \frac{\underline{v}^T[A]\underline{v}}{\underline{v}^T[B]\underline{v}} \quad .. \quad .. \quad .. \quad .. \quad (2.18)$$

Define now a set of vectors, $\bar{\underline{v}}$, which are linear combinations of the Ritz basis vectors, $\underline{u}_i$, i = 1, ... , q and are given by

$$\bar{\underline{v}} = \sum_{i=1}^{q} x_i \underline{u}_i \qquad .. \quad .. \quad .. \quad .. \quad (2.19)$$

where the x_i are the Ritz coordinates. Clearly, the $\bar{\underline{v}}$ must be in the subspace, E_q, spanned by the Ritz basis vectors. The vectors, $\underline{u}_i$ must be linearly independent so that the subspace, E_q, has dimension q, also we have that E_q is contained in E_n.

Substituting equation (2.19) in equation (2.18) gives

$$\rho(\bar{\underline{v}}) = \frac{\sum_{1}^{q}\sum_{1}^{q} x_i x_j \, a_{ij}}{\sum_{1}^{q}\sum_{1}^{q} x_i x_j \, b_{ij}} = \frac{\bar{a}}{\bar{b}} \qquad .. \quad .. \quad .. \quad (2.20)$$

where

$$\left.\begin{aligned} a_{ij} &= \underline{u}_i^T [A] \underline{u}_j \\ b_{ij} &= \underline{u}_i^T [B] \underline{u}_j \end{aligned}\right\} \qquad .. \quad .. \quad .. \quad (2.21)$$

The necessary condition for a minimum of $\rho(\bar{\underline{v}})$ is $\partial\rho(\bar{\underline{v}})/\partial x_i = 0$, $i = 1, \ldots\ldots q$. This yields

$$[\bar{A}]\underline{x} = \rho[\bar{B}]\underline{x} \qquad .. \quad .. \quad .. \quad .. \quad (2.22)$$

where $[\bar{A}]$ and [B] are symmetric matrices of order q with typical elements defined in equation (2.21) and $\underline{x}$ is the vector containing the Ritz coordinates. The solution of equation (2.22) yields the eigenvalues ρ_i, $i = 1, \ldots\ldots, q$ and the corresponding eigenvectors, $\underline{x}_i$, $i = 1, \ldots\ldots, q$. These $\underline{x}_i$ are then used in equation (2.19) to obtain $\bar{\underline{v}}_i$, $i = 1, \ldots\ldots, q$. The ρ_i obtained, are upper bounds to the λ_i, i.e.

$$\lambda_1 \leq \rho_1,\ \lambda_2 \leq \rho_2,\ \ldots\ldots\ \lambda_q \leq \rho_q$$

The first inequality follows from the fact that E_q is contained in E_n, the second inequality may be proved as follows:

$$\lambda_2 = \min \rho(\underline{v}) \qquad .. \quad .. \quad .. \quad (2.23)$$

The minimum is taken over all vectors $\underline{v}$ in E_n that satisfy the orthogonality condition

$$\underline{v}^T[B]\underline{v}_1 = 0 \qquad .. \quad .. \quad .. \quad (2.24)$$

Also from the Rayleigh-Ritz analysis,

$$\rho_2 = \min \rho(\bar{\underline{v}}) \qquad .. \quad .. \quad .. \quad (2.25)$$

the minimum here is taken over all possible vectors, $\bar{\underline{v}}$ in E_q that satisfy the orthogonality condition,

$$\bar{\underline{v}}^T[B]\bar{\underline{v}}_1 = 0 \qquad .. \quad .. \quad .. \quad (2.26)$$

Consider now an auxiliary problem,

$$\bar{\rho}_2 = \min \rho(\bar{\underline{v}}) \qquad .. \quad .. \quad .. \quad (2.27)$$

where the minimum is taken over all vectors $\underline{v}$ which satisfy the following condition

$$\bar{\underline{v}}^T[B]\underline{v}_1 = 0 \qquad .. \quad .. \quad .. \quad (2.28)$$

Now $\lambda_2 \leq \bar{\rho}_2$ since E_q is contained in E_n, also $\bar{\rho}_2 \leq \rho_2$ since the most severe constraint on $\bar{\underline{v}}$ in equation (2.28) is $\bar{\underline{v}}_1$, therefore

$$\lambda_2 \leq \bar{\rho}_2 \leq \rho_2$$

The third and subsequent inequalities may be proved similarly. Note that in the evaluation of ρ_i and $\bar{\underline{v}}_i$, $\rho(\underline{v})$ has to be minimised with the orthogonality condition,

$\underline{v}^T[B]\underline{v}_j = 0$, $j = 1, \ldots, i-1$. This indicates that less accuracy is obtained in the approximation of the higher eigenvalues since $(i - 1)$ constraint equations have to be satisfied in the evaluation of ρ_i.

2.3.6 Static condensation

In static condensation[5,29-31] those degrees of freedom which are not required to appear in the global finite element assemblage are eliminated. The assumption that the mass of the structure can be lumped at some specific degrees of freedom without having much effect on the eigenpairs of interest is inherent to this approach. Once the mass lumping has been carried out, then by static condensation the original n degree-of-freedom becomes a n_m degree-of-freedom problem, where n_m are the allowed mass degrees of freedom. Typically, the ratio of the n_m to n is between $\frac{1}{2}$ and $\frac{1}{10}$. It should be noted, however, that n_m must be significantly larger than the required number of eigenpairs in order to keep an adequate mass distribution in the system. Clearly, the accuracy of the solution obtained depends upon the engineering judgment of the analyst.

2.3.7 Component mode synthesis

This method[5,32-35] is appropriate if a large complex structure is to be analysed. Such a structure is partitioned into substructures and each substructure is analysed separately. Once the preliminary analyses has been carried out, the mode shape characteristic of each component is know. These are then combined to

estimate the mode shape characteristic of the complete structure.

2.3.8 Discussion

Note that both Static condensation, section (2.3.6), and Component mode synthesis, section (2.3.7), can be understood as variations of Ritz analyses[5].

Consider now the Rayleigh-Ritz analysis, section (2.3.5), in practical dynamic analysis, the Ritz basis vectors are calculated from the static solution, i.e. solution of equation (2.2). This is done by specifying q load vectors in the matrix R so that,

$$[K][U] = [R] \quad .. \quad .. \quad .. \quad .. \quad (2.29)$$

where [U] is a n x q matrix containing the Ritz basis vectors, $[U] = [\underline{u}_1, \underline{u}_2, \ldots\ldots, \underline{u}_q]$. The next step in the analysis is the evaluation of the projections of [K] and [M] onto the subspace, E_q, which is spanned by the vectors, $\underline{u}_i$, i = 1,...., q.

$$[\bar{K}] = [U]^T[K][U] \quad .. \quad .. \quad .. \quad (2.30)$$

$$[\bar{M}] = [U]^T[M][U] \quad .. \quad .. \quad .. \quad (2.31)$$

This leads to the reduced eigenproblem

$$[\bar{K}]\underline{x} = p[\bar{M}]\underline{x} \quad .. \quad .. \quad .. \quad .. \quad (2.32)$$

The solution of which can be written as

$$[\bar{K}][X] = [M][X][P] \quad .. \quad .. \quad .. \quad (2.33)$$

where [P] is a diagonal matrix containing the eigenvalue approximation p_i and the columns of the matrix [X] are the [M]-orthogonal eigenvectors of equation (2.32). The approximations to the eigenvectors of the original problem are then given by ,

$$[V] = [U][X] \quad .. \quad .. \quad .. \quad .. \quad (2.34)$$

In the discussion of the Rayleigh-Ritz analysis the operators [A] and [B] were assumed to be positive definite. This is always true for [K] since, as mentioned previously, a shift can be introduced to obtain a shifted [K] which satisfies this conditions. The case where [M] is non-negative definite can be dealt with by ensuring that the Ritz basis vectors lie in the subspace corresponding to the finite eigenvectors. The error in the eigenvalue approximations depends upon the Ritz basis vectors chosen. Thus, good results are only obtained if these basis vectors span a subspace which is close to the least dominant q-dimensional subspace of [K] and [M].

The paramount problem in a Ritz analysis is the selection of 'good' basis vectors. The repetition of the analysis with a larger set of basis vectors is not necessarily a check on the first analysis. Since a large discrepancy in the results of the two analyses only indicates that either one or both analyses are giving inaccurate eigenpair approximations. Such a situation necessarily commits the problem to further analyses.

In a Ritz-type analysis, two major points of uncertainty exist. Firstly, the accuracy of the approximations to the required eigenpairs is not known. Secondly, the possibility that an approximation to an eigenpair may have been missed altogether is present. The uncertainty due to these points leads to a large number of repetitions of the analysis, involving high cost, without, however, removing all uncertainty. The question then arises,

whether it would have been more efficient to solve the full eigenproblem, provided a solution is possible.

2.4 Buckling analysis

One other generalised eigenproblem merits brief discussion. Consider the equations governing the bifurcation buckling of a structure.

$$[K]\underline{v} = \lambda[K]_G\underline{v} \quad .. \quad .. \quad .. \quad (2.35)$$

where [K] is the linear strain, stiffness matrix and $[K]_G$ is the non-linear strain (geometric)[36] stiffness matrix. The buckling load and the corresponding buckling mode are given by λ and $\underline{v}$ respectively. $[K]_G$ is a banded matrix with the same bandwidth as [K] and is, in general, indefinite. In this case, the problem to be solved is

$$[K]_G\underline{v} = \bar{\lambda}[K]\underline{v} \quad .. \quad .. \quad .. \quad .. \quad (2.36)$$

where $\bar{\lambda} = \frac{1}{\lambda}$ and may be positive or negative.

Equation (2.36) is solved for the largest value of $\bar{\lambda}$ which corresponds to the smallest buckling load. In practical analyses, it is desirable to find the lowest 'few' buckling loads since, if they are very close, preventing the lowest buckling mode becomes inconsequential.

Note that here static condensation cannot be used. However, a Ritz analysis is applicable.

CHAPTER 3

SUBSPACE ITERATION

3.1 Introduction

In this Chapter, a subspace iteration algorithm is presented[3]. In the past, a number of subspace iteration algorithms have been proposed by various authors[37-41]. A primary advantage of subspace iteration over other methods is that high precision in the calculation of eigenvalues and eigenvectors is not required to preserve numerical stability.

The subspace iteration method is largely based on various techniques mentioned earlier, namely simultaneous vector iteration, Sturm sequence property and the Rayleigh-Ritz analysis. It uses primarily vector inverse iteration with a few triangular factorisations. The projections of the operators [K] and [M] onto a subspace are formed and the reduced eigenproblem is solved by the generalised Jacobi method.

The main difficulty encountered in this method is the selection of the initial subspace. The scheme used is a simple one. However, experience has shown that it is successful.

3.2 Algorithm

The basic idea is iterating simultaneously with a number of vectors. The number of iterations to convergence depends primarily upon how rich the individual initial iteration vectors are in their corresponding final eigenvectors.

Suppose the initial iteration vectors, $[X]_1$ span the r-dimensional least dominant subspace but are not eigenvectors. An effective iteration scheme would, in this case, find the eigenvectors in a single step.

Let $[X]_k$ be the iteration vectors after (k - 1) iterations which span the subspace E_k. The aim is to find the subspace E_{k+1}, the basis vectors of which, $[X]_{k+1}$ are a better approximation to the eigenvectors than $[X]_k$.

The solution algorithm proceeds as follows:
let [Y] be the initial iteration vectors, then iterating from E_k to E_{k+1} for k = 1,2 .....

$$[K][X]_{k+1} = [Y]_k \quad .. \quad .. \quad .. \quad .. \quad (3.1)$$

$$[\bar{Y}]_{k+1} = [M][X]_{k+1} \quad .. \quad .. \quad .. \quad (3.2)$$

Obtain the projections of [K] and [M] onto E_{k+1}

$$[\bar{K}] = [X]_{k+1}^T [K][X]_{k+1} \quad .. \quad .. \quad .. \quad (3.3)$$

$$[\bar{M}] = [X]_{k+1}^T [M][X]_{k+1} \quad .. \quad .. \quad .. \quad (3.4)$$

Solve the projected eigenproblem

$$[\bar{K}][Q]_{k+1} = [\bar{M}][Q]_{k+1}[\Lambda]_{k+1} \quad .. \quad .. \quad (3.5)$$

where $[Q]_{k+1}$ and $[\Lambda]_{k+1}$ are the eigenvectors and eigenvalues respectively of the projected problem.
Calculate an improved approximation to the eigenvectors.

$$[Y]_{k+1} = [\bar{Y}]_{k+1}[Q]_{k+1} \quad .. \quad .. \quad .. \quad (3.6)$$

Then provided the vectors in $[Y]_1$ are not orthogonal to one of the required eigenvectors,

$$[\Lambda]_{k+1} \rightarrow [\Lambda] \text{ and } [Y]_{k+1} \rightarrow [V] \text{ as } k \rightarrow \infty.$$

Note that the projected eigenproblem is solved in the r-dimensional least dominant subspace.

The convergence analysis given by Rutishauser[22] is applicable to the above algorithm, although he uses a different subspace iteration. The convergence is asymptotic and the convergence rate of the ith iteration vector to an eigenvector can be shown to be λ_i/λ_{r+1}, where the iteration is performed with r vectors. This asymptotic convergence rate indicates that the eigenvectors corresponding to the lowest eigenvalues converge fastest. A higher convergence rate can be obtained by using q iteration vectors when r eigenvectors are required with, $q > r$. Provided $\lambda_r < \lambda_{r+1}$, multiple eigenvalues do not have a detrimental effect on the convergence rate.

Consider equations (3.3) - (3.5), these represent a Ritz analysis as described in section (2.2), with $[X]_{k+1}$ as the Ritz basis vectors. As with all Ritz type analysis, $[\Lambda]_{k+1}$ contains upperbounds to the eigenvalues, $\lambda_1, \ldots, \lambda_r$ and the lower eigenvalues will be approximated best.

3.3 Generalised Jacobi iterations

For a general discussion, consider the symmetric operators [A] and [B] of order q with [B] positive definite. The problem to be solved is:

$$[A]\underline{v} = p[B]\underline{v} \quad .. \quad .. \quad .. \quad .. \quad (3.7)$$

Note that in this case there are no restrictions on the eigenvalues, p_i, which may be positive, negative or zero.

The method is best demonstrated by a general example; for ease of calculation consider the case n = 2.

$$[A] = \begin{bmatrix} a_{ii} & a_{ij} \\ a_{ji} & a_{jj} \end{bmatrix}; \quad [B] = \begin{bmatrix} b_{ii} & b_{ij} \\ b_{ji} & b_{jj} \end{bmatrix}$$

The aim is to obtain the two [B]-orthonormal vectors which also diagonalise [A]. The directions of these vectors are determined completely by the columns in [V], where

$$V = \begin{bmatrix} 1 & s \\ t & 1 \end{bmatrix}$$

The condition governing the choice of s and t is that a_{ij} and b_{ij} shall be zero simultaneously. Hence, by forming $[V]^T[A][V]$ and $[V]^T[B][V]$ two equations for s and t are obtained.

$$s\, a_{ii} + (1 + st)\, a_{ij} + t\, a_{jj} = 0 \qquad .. \qquad (3.8)$$

$$s\, b_{ii} + (1 + st)\, b_{ij} + t\, b_{jj} = 0 \qquad .. \qquad (3.9)$$

Equations (3.8) and (3.9) are linear equations which in the general case are solved by defining the following

$$\left.\begin{aligned} \bar{a}_{jj} &= a_{jj}\, b_{ij} - b_{jj}\, a_{ij} \\ \bar{a}_{ii} &= a_{ii}\, b_{ij} - b_{ii}\, a_{ij} \\ a &= a_{ii}\, b_{jj} - a_{jj}\, b_{ii} \\ s &= \frac{\bar{a}_{jj}}{x} \qquad t = -\frac{\bar{a}_{ii}}{x} \end{aligned}\right\} \qquad .. \quad .. \qquad (3.10)$$

The value of x required to calculate s and t is then obtained by solving

$$x^2 - ax - \bar{a}_{jj}\, \bar{a}_{ii} = 0 \qquad .. \quad .. \quad .. \qquad (3.11)$$

The solution of which is given by

$$x = \frac{a}{2} \pm \sqrt{\frac{a^2 + 4\,\bar{a}_{jj}a_{ii}}{4}} \quad .. \quad .. \quad .. \quad .. \quad (3.12)$$

where the absolutely larger value of x is employed.

It is generally known that this congruence transformation is possible[4] provided one of the operators is positive definite. Therefore, when considering [K] and [M], this effectively allows [M] to be non-negative definite.

Physically, the aim in the diagonalisation is to reduce the coupling between the degrees of freedom i and j. The coupling factors are,

$$\sqrt{a^2{}_{ij}/a_{ii}a_{jj}} \; ; \; \sqrt{b^2{}_{ij}/b_{ii}/b_{ji}} \quad .. \quad .. \quad .. \quad (3.13)$$

For efficiency, first the most significant and then the smaller couplings are annihilated. The method proceeds as follows:

(i) Initiatise the threshold for sweep, ℓ.

(ii) Calculate the coupling factors for all (i,j) with $i < j$, and apply a transformation if either of the factors is larger than the current threshold.

(iii) Obtain current eigenvalue estimates.

(iv) Check for convergence of eigenvalues, if convergence has not occurred start a new sweep.

The tolerance used on the eigenvalues is 10^{-8}, consequently the threshold used in step (i) is $10^{-2\ell}$, so that the coupling factors can be expected to be smaller than 10^{-8} after about four iterations. This is quite

important since, even if the eigenvalues have converged, convergence is not accepted until the coupling factors are also smaller than 10^{-8}.

The reasons for preferring this method over others for solving the projected eigenproblem are the following:

(a) There is no initial transformation to the standard form. This has two implications:

  (i) If the matrices are ill-conditioned, then the solution of the standard eigenproblem with ill-conditioned matrices is avoided.

  (ii) If the off-diagonal terms in [K] and [M] are already small or only a few non-zero off-diagonal terms are present, then zeroing a few or small terms will not change the diagonal terms drastically. Since the ratios of the diagonal terms are the eigenvalues, the eigenproblem is nearly solved.

(b) Advantage is taken of the fact that, as the number of iterations increases, $[X]_{k+1}$ tend to the eigenvectors and, therefore, $[\bar{K}]$ and $[\bar{M}]$ tend to diagonal form.

3.4 Initial Iteration Vectors

The most critical aspect of subspace iteration is the choice of the initial iteration vectors in $[Y]_1$. If the initial iteration vectors span the least dominant subspace, then convergence is almost immediate.

Consider the case when there are only r non-zero terms in the diagonal mass matrix. The initial iteration

vectors in this case are unit vectors with the +1 entries corresponding to the mass degrees of freedom. One subspace iteration in this case is, in fact, a static condensation analysis and convergence is immediate. Similarly, component mode synthesis and other related methods can all lead to good initial iteration vectors. Another case when convergence is immediate is when [K] and [M] are both diagonal matrices. The initial iteration vectors in this case are unit vectors with the +1 entries corresponding to those degrees of freedom where the smallest k_{ii}/m_{ii} ratios occur. The k_{ii} and m_{ii} being the diagonal elements of [K] and [M] respectively. The initial iteration vectors obtained by this method are effective because they are actually the eigenvectors corresponding to the smallest eigenvalues.

Considering the above discussion, the initial iteration vectors for a general problem are chosen as follows. The first column in $[Y]_1$ is the diagonal of [M]. This ensures that all the mass degrees of freedom are excited. The subsequent columns of $[Y]_1$ are unit vectors with the +1 entries corresponding to those degrees of freedom which have the smallest k_{ii}/m_{ii} ratios. Thus, apart from the first vector, all the other vectors are linearly independent and excite points of maximum mass and flexibility. Note that the unit entries in the second to last vector should not be clustered together too much.

It should be noted that, since subspace iteration may be interpreted as a repeated application of the Ritz analysis, all the characteristics of the Ritz analysis

pertain also to subspace iteration. Recall that in particular, good results are only obtained if the Ritz basis vectors span a subspace sufficiently close to the least dominant subspace. Thus, the choice of the initial iteration vectors is of paramount importance.

3.5 Numerical considerations

There are several numerical aspects which should be considered separately.

3.5.1 Dimension of the subspace

It was mentioned in section (3.2) that the ultimate rate of convergence of an iteration could be increased by increasing the dimension of the iterating subspace. Increasing the dimension of the iterating subspace, however, increases the cost of solutions. Therefore, an optimum for the dimensionality of the iterating subspace is required. Clearly, the number of iteration vectors q, need to be greater than the eigenvalue sought, r in order to allow for multiple roots and to obtain a better convergence rate. Experience has shown that a reasonable number of iteration vectors are given by[5],

$$q = \min \{2\ r\ ,\ 8 + r\} \qquad .. \quad .. \quad .. \quad (3.14)$$

This allows for multiple roots and the dimension of the subspace is large enough to expect monotonic convergence without employing an eccessive number of iteration vectors.

3.5.2 Convergence

The criteria used for convergence is:

$$\left| \frac{\lambda_i^{(k+1)} - \lambda_i^{k}}{\lambda_i^{(k+1)}} \right| \leq \text{TOL} \quad .. \quad .. \quad .. \quad (3.15)$$

where $\lambda_i^{(k+1)}$ and λ_i^{k} are the estimates for the ith eigenvalue after k and (k-1) iterations respectively, and TOL is the prescribed tolerance. Clearly, TOL is dependent upon the accuracy sought, type of problem being analysed and word length of the computer used.

The tolerance used in the case studies was 10^{-4}, which means that λ^{k} differs from $\lambda^{(k+1)}$ by less than 1% when convergence is accepted. The program stops once the required r eigenvalues have converged or the specified maximum number of iterations is reached. If the required r eigenvalues do not converge, then either the number of iteration vectors or the maximum number of iterations allowed must be increased.

It should be noted that equation (3.15) represents only a necessary condition for convergence. The necessary and sufficient conditions are satisfied if and only if the eigenvalue and the corresponding eigenvector estimates satisfy the eigenproblem.

3.5.3 Check calculations

Another important aspect of this solution procedure is the verification of the fact that the required eigenpairs have been calculated and none of the required eigenvalues have been missed. In this phase use is made of the sturm sequency property of the characteristic polynomial of the eigenproblem.

A triangular factorisation of $[K]' = ([K] - \bar{s}[M])$ is carried out, where $\bar{s}$ is just greater than the largest eigenvalue calculated, λ_r, i.e.

$$[K] - \bar{s}[M] = [L][D][L]^T \quad .. \quad .. \quad .. \quad .. \quad (3.16)$$

where [L] and $[L]^T$ are lower and upper unit triangular matrices respectively, and [D] is a diagonal matrix. The number of negative elements in [D] is equal to the number of eigenvalues smaller than $\bar{s}$, thus, in this case, there should be r negative terms in [D]. There remains the problem, however, of obtaining a meaningful value for $\bar{s}$. The fact that the λ_i are only approximations, should be taken into account here.

The criteria for obtaining a value for $\bar{s}$ should be less stringent than the eigenvalue convergence criteria since inaccuracy in the eigenvalue approximation should be bracketed by the bounds for the eigenvalue. Experience has shown that a reasonable estimate for the region within which the exact eigenvalue lies is given by[5],

$$0.99\ \lambda_i^{(k+1)} \leq \lambda_i \leq 1.01\ \lambda^{(k+1)} \quad .. \quad .. \quad .. \quad (3.17)$$

where $\lambda_i^{(k+1)}$ is the approximation of the ith eigenvalue after k iterations. Here, only those eigenvalues which have converged are used, thus bounds on all converged eigenvalues can be established. This leads to a reasonable estimate for $\bar{s}$ and consequently a realistic sturm sequence check may be performed.

The overall advantage of such checks is that if an eigenvalue approximation is missing, then the interval it is missing from can be identified.

3.5.4 Shifting

Since the lowest eigenvalues converge first, it would seem natural to use shifting as a method of speeding up convergence. However, the difficulty here is that if the shift is 'too large' then convergence to the lower eigenvalue is lost. Alternatively, if the shift is 'too small' then the increase in convergence is not significant. A good value for the shift may be obtained once the eigenvalue spectrum is known approximately, but then the additional triangular factorisation becomes uneconomical since subspace iteration is close to converging.

Shifting becomes important if a significant eigenvalue approximation has been missed. First the interval which contains the missing eigenvalue is identified, then the eigenproblem is shifted to

$$([K] - \bar{s}'[M])\underline{v} = \lambda[M]\underline{v} \qquad .. \quad .. \quad .. \quad (3.18)$$

where $\bar{s}'$ is the upper bound for the eigenvalue directly prior to the missing eigenvalue. The missing eigenvalue approximation may now be obtained by performing subspace iteration on the shifted eigenproblem, i.e. equation (3.18).

A similar approach can be used to obtain the eigenvalues in a given interval. The procedure in this case is to set the shift equal to the lower bound of the interval and perform subspace iteration on the shifted problem.

Triangular factorisation can be used here to determine the number of eigenvalues in the specified interval.

3.6 Operation counts

An operation is defined by a multiplication which is nearly always followed by an addition. There are two cases to be considered:

(i) Consistent Mass formulation, $m_k = m_m$

(ii) Lumped Mass formululation, $m_m = 0$

where m_k and m_m are the half-bandwidths of the stiffness and mass matrices respectively. The number of operations required in subspace iteration for the two cases mentioned are given in Table (3.1), a detailed discussion is given elsewhere[5].

Note that operation counts are very useful for comparison purposes. Also, an operation count for any modification will give an indication of how much extra work is required when that modification is employed.

In Table (3.1), $(\lambda_i', \underline{v}_i')$ is the final approximation of the ith eigenpair and the norm used is the Euchilean vector norm.

Operation	Calculation	Number of Operations	
		$m = m_k = m_m$	$m = m_k;\ m_m = 0$
Factorisation of [K]	$[K] = [L][D][L]^T$	$\frac{1}{2} nm^2 + \frac{3}{2} nm$	$\frac{1}{2} nm^2 + \frac{3}{2} nm$
Subspace Iteration	$[K][X]_{k+1} = [Y]_k$	$nq(2m + 1)$	$nq(2m + 1$
	$[\bar{Y}]_{k+1} = [M][X]_{k+1}$	$nq(2m + 1)$	nq
	$[\bar{K}] = [X]^T_{k+1}[Y]_k$	$\frac{1}{2} nq(q + 1)$	$\frac{1}{2} nq(q + 1)$
	$[\bar{M}] = [X]^T_{k+1}[\bar{Y}]_{k+1}$	$\frac{1}{2} nq(q + 1)$	$\frac{1}{2} nq(q + 1)$
	$[\bar{K}][Q]_{k+1} = [\bar{M}][Q]_{k+1}[\Lambda]_{k+1}$	of order	(q^3)
	$[Y]_{k+1} = [\bar{Y}]_{k+1}[Q]_{k+1}$	nq^2	nq^2
Sturm Sequence Check	$[K]' = ([K] - \bar{s}[M])$	$n(m + 1)$	n
	$[K]' = [L][D][L]^T$	$\frac{1}{2} nm^2 + \frac{3}{2} nm$	$\frac{1}{2} nm^2 + \frac{3}{2} nm$
Check Calculations	$\frac{\| [K]\underline{v}_i' - \lambda_i' [M]\underline{v}_i' \|}{\| [K]\underline{v}_i' \|}$	$2\ nm + 4\ n$	$5\ nm + 2\ n$

TABLE 3.1. OPERATION COUNTS FOR SUBSPACE ITERATION SOLUTION

CHAPTER 4

MODIFICATIONS TO SUBSPACE ITERATION

4.1 Introduction

In this chapter, various techniques which may accelerate the subspace iteration algorithm are considered. There exists a number of ideas[42-45] which may be employed to modify particular parts of the subspace iteration algorithm. It is not difficult to dream up modifications of high complexity which may converge in fewer iterations[42,43] but which, however, use more computer time due to the complexity of the modification. It is easy to lose perspective and become too involved with the modifications. Therefore, in order to design modifications which best optimise the algorithm, a clear overview of the prime objectives is necessary. This follows from the fact that the algorithm is a modular entity, with each module contributing to the total time taken for an iteration. Clearly, since these contributions cannot be of the same order, it would be prudent to neglect those steps which use relatively little time and concentrate on those which are the prime contributors to the total time.

Two methods of modifying the subspace iteration algorithm and the variations arising from these have been studied. The two basic methods are:

(1) Repeatedly operating upon the iterating subspace with the operators [K] and [M]

(2) Taking a linear combination of the previous two estimates of the subspace to form the current subspace.

These are referred to as Method 1 and Method 2 respectively throughout.

Consider now the objectives of the modifications:

(a) Faster solution without loss of accuracy

(b) Higher convergence rate

(c) Fewer iterations for convergence.

It is with the above objectives in mind that the various modifications have been designed. Here, the theory is presented and the primary numerical aspects of the modifications are discussed.

4.2 Method 1

4.2.1 Theoretical considerations

The calculation of a few lowest eigenvalues and the corresponding eigenvectors of the problem

$$[K]\underline{v}_i = \lambda_i [M]\underline{v}_i \quad .. \quad .. \quad .. \quad .. \quad (4.1)$$

is required. Where the [K] and [M] are the system matrices, λ_i and $\underline{v}_i$ are the eigenvalues and the corresponding eigenvectors respectively. Let q be the number of iteration vectors, with $q \leq n$, and $\underline{x}_i$ be the iteration vectors with $i = 1, \ldots, q$, then:

$$\underline{x}_i = \sum_{j=1}^{n} a_{ij}\underline{v}_j \quad .. \quad .. \quad .. \quad (4.2)$$

where the a_{ij} are constant coefficients and are defined by

$$a_{ij} = \begin{cases} \text{Ord}(1) , & i = j \\ \text{Ord}(e) & i \neq j \end{cases}$$

and e is small.

Consider now the implications of operating upon the $\underline{x}_i$ with the operators [K] and [M]. Two possibilities are available here, i.e. operating with $[M]^{-1}[K]$ or $[K]^{-1}[M]$, both cases are investigated separately.

(a) Suppose $\bar{\underline{y}}_i$ is formed in the following manner

$$\bar{\underline{y}}_i = [M]^{-1}[K]\underline{x}_i \quad .. \quad .. \quad .. \quad .. \quad (4.3)$$

substituting for $\underline{x}_i$ from equation (4.2)

$$\bar{\underline{y}}_i = [M]^{-1}[K] \sum_{j=1}^{n} a_{ij}\underline{v}_j \quad .. \quad .. \quad .. \quad (4.4)$$

Re-arranging equation (4.1) yields

$$([M]^{-1}[K])\underline{v}_i = \lambda_i \underline{v}_i \quad .. \quad .. \quad .. \quad (4.5)$$

Substituting from equation (4.5) into equation (4.4)

$$\bar{\underline{y}}_i = \sum_{j=1}^{n} a_{ij}\lambda_j \underline{v}_j \quad .. \quad .. \quad .. \quad .. \quad (4.6)$$

Let $\bar{f}_a$ be the normalising factor for $\bar{\underline{y}}_i$, also let

$$f_a = \bar{f}_a \lambda_1 \quad .. \quad .. \quad .. \quad .. \quad .. \quad (4.7)$$

Let $\underline{y}_i$ be the normalised iteration vectors, then substituting from equation (4.7) into equation (4.6) leads to

$$\underline{y}_i = f_a \sum_{j=1}^{n} a_{ij} \left(\frac{\lambda_j}{\lambda_1}\right) \underline{v}_j \quad .. \quad .. \quad .. \quad (4.8)$$

(b) In this case let $\bar{\underline{z}}_i$ be formed as follows

$$\bar{\underline{z}}_i = [K]^{-1}[M]\underline{x}_i \quad .. \quad .. \quad .. \quad .. \quad (4.9)$$

Substituting for $\underline{x}_i$ from equation (4.2)

$$\bar{\underline{z}}_i = [K]^{-1}[M] \sum_{j=1}^{n} a_{ij}\underline{v}_j \quad .. \quad .. \quad .. \quad (4.10)$$

Re-arranging equation (4.1) yields

$$\frac{1}{\lambda_i}\underline{v}_i = ([K]^{-1}[M])\underline{v}_i \quad .. \quad .. \quad .. \quad (4.11)$$

Substituting from equation (4.11) into equation (4.10) gives

$$\bar{\underline{z}}_i = \sum_{j=1}^{n} a_{ij}\frac{1}{\lambda_j}\underline{v}_j \quad .. \quad .. \quad .. \quad (4.12)$$

Let $\bar{f}_b$ be the normalising factor for $\bar{\underline{z}}_i$, also let

$$f_b = \frac{\bar{f}_b}{\lambda_1} \quad .. \quad .. \quad .. \quad .. \quad .. \quad (4.13)$$

Let $\underline{z}_i$ be the normalised iteration vectors, then substituting from equation (4.13) into equation (4.12) leads to

$$\underline{z}_i = f_b \sum_{j=1}^{n} a_{ij}\left(\frac{\lambda_1}{\lambda_j}\right)\underline{v}_j \quad .. \quad .. \quad .. \quad (4.14)$$

Let $\lambda_j/\lambda_1 = \bar{\ell}_j$ in the following discussion, then clearly $\lambda_1/\lambda_j = 1/\bar{\ell}_j$. In both (a) and (b), f_a and f_b are formed from the approrpriate normalising factors in order to obtain $\bar{\ell}_j$ and $1/\bar{\ell}_j$ in equations (4.8) and (4.14) respectively. The presence of $\bar{\ell}_j$ and $1/\bar{\ell}_j$ clarifies the convergence behaviour of $\underline{y}_i$ and $\underline{z}_i$ respectively.

Consider case (a), the minimum value of $\bar{\ell}_j$ is clearly unity and since $\lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \ldots\ldots \leq \lambda_n$, the

values of $\bar{\ell}_j$ form an increasing sequence. Therefore, it follows that the higher modes are enhanced in the vectors [Y], in fact as $j \to n$ the higher modes become progressively more dominant in the vectors [Y]. Note that the columns of [Y] are the iteration vectors $\underline{y}_i$. The effect of operating in this manner upon the iteration vectors is an enhancement of the higher modes and thus a divergence away from the least dominant subspace. This is obvious from the fact that $\underline{y}_i$ converges linearly to $\underline{v}_i$ provided $\underline{y}_i$ is rich in $\underline{v}_i$.

Consider now case (b), here, $1/\bar{\ell}_j$ forms a decreasing sequence and, therefore, the higher modes are suppressed. This suppression becomes stronger as $j \to n$, i.e. $1/\bar{\ell}_j$ gets smaller. It follows from equation (4.14) that the vectors in [Z] will be rich in the lower modes while the higher modes become progressively less significant. Hence, in this case, the overall effect is a convergence towards the least dominant subspace. In fact, if this operation is carried out repeatedly, the effect is a convergence to the least dominant vector, i.e. all the iteration vectors converge to the lowest mode.

From the above discussion, it is clear that case (b) is appropriate since the lowest eigenvalue and the corresponding eigenvectors are required.

4.2.2 Algorithm

The subspace spanned by the vectors in $[Y]_k$ is operated upon by $[K]^{-1}[M]$, ℓ times where ℓ is obtained

through experience. Suppose $[Y]_1 = [M][X]_1$ and [X] is given, then the algorithm proceeds as follows:

$$[K][X]_{k+1} = [Y]_k \quad .. \quad .. \quad .. \quad .. \quad (4.15)$$

$$[Y]_{k+1} = [M][X]_{k+1} \quad .. \quad .. \quad .. \quad (4.16)$$

$$[K][X]_{k+2} = [Y]_{k+1} \quad .. \quad .. \quad .. \quad (4.17)$$

$$[Y]_{k+2} = [M][X]_{k+2} \quad .. \quad .. \quad .. \quad (4.18)$$

$$[K][X]_{k+\ell} = [Y]_{k+\ell-1} \quad .. \quad .. \quad .. \quad (4.19)$$

$$[Y]_{k+\ell} = [M][X]_{k+\ell} \quad .. \quad .. \quad .. \quad (4.20)$$

Now the algorithm proceeds as in the original algorithm with the vectors in $[Y]_{k+\ell}$ spanning the initial subspace.
Back substitution and forward reduction to obtain $[X]_{k+\ell+1}$

$$[K][X]_{k+\ell+1} = [Y]_{k+\ell} \quad .. \quad .. \quad .. \quad (4.21)$$

Matrix multiplication to obtain $[\bar{Y}]_{k+\ell+1}$

$$[\bar{Y}]_{k+\ell+1} = [M][X]_{k+\ell+1} \quad .. \quad .. \quad .. \quad (4.22)$$

form the projected operators.

$$[\bar{K}] = [X]^T_{k+\ell+1} [Y]_{k+\ell} \quad .. \quad .. \quad .. \quad (4.23)$$

$$[\bar{M}] = [X]^T_{k+\ell+1} [\bar{Y}]_{k+\ell+1} \quad .. \quad .. \quad .. \quad (4.24)$$

Solve the projected eigenproblem

$$[\bar{K}][Q] = [\bar{M}][Q][\Lambda] \quad .. \quad .. \quad .. \quad (4.25)$$

where [Q] and [Λ] are as defined in the previous chapter.

Vector multiplication to form $[Y]_{k+\ell+1}$

$$[Y]_{k+\ell+1} = [\bar{Y}]_{k+\ell+1}[Q] \quad .. \quad .. \quad .. \quad (4.26)$$

Now $[Y]_{k+\ell+1}$ replaces $[Y]_k$ in the next iteration.

4.2.3 Numerical considerations

4.2.3.1 Numerical stability

It is seen from equation (4.14) that, if the iteration vectors are operated upon by $[K]^{-1}[M]$, the higher modes are suppressed and thus the lower modes are dominant in the resulting vectors. This effect becomes more and more pronounced as the iteration vectors are repeatedly operated upon by $[K]^{-1}[M]$. The overall effect is to make the iteration vector more and more parallel to each other and thereby making them a poorer basis for the subspace. In order to counteract this and preserve numerical stability, Gram-Schmidt orthogonalisation and normalisation with respect to [M] are employed.

It is difficult to decide how often the numerical stability preserving procedure should be applied. Clearly, the number of $[K]^{-1}[M]$ operations after which the numerical stabilities become irreversible due to finite precision arithmetic in the computer are problem and machine dependent. Therefore, it appears prudent to apply Gram-Schmidt orthogonalisation and normalisation after each operation of $[K]^{-1}[M]$. Although this takes into account the worse case, it is inefficient if the numerical instabilities in the

iteration vectors do not occur after each operation of $[K]^{-1}[M]$.

4.2.3.2 Convergence

Here there are two convergence rates to consider, i.e.

(i) The convergence rate due to the $[K]^{-1}[M]$ operations

(ii) The convergence rate due to the original subspace iteration algorithm.

The remarks made on the convergence of subspace iteration in section (3.5.2) are directly applicable for (ii). This follows from the fact that the $[Y]_{k+\ell}$ in equation (4.2.1) are the initial iteration vectors for subspace iteration. Thus, the modification is applied at the level of obtaining a better initial subspace.

Consider now the convergence rate from (i). This clearly depends upon the number of $[K]^{-1}[M]$ operations. This is obvious from equation (4.14) since ℓ, $[K]^{-1}[M]$ operations will lead to a factor of $(1/\bar{\ell})^{\ell}$ in the equation. Therefore, although the convergence rate is asymptotic, it may be increased by operating upon the iteration vectors by $[K]^{-1}[M]$.

4.2.4 Implementation

The modification to the subspace iteration algorithm has been programmed in modular form. Equations (4.15) to (4.20) are programmed in a subroutine which may be inserted into any working subspace iteration programme.

Two subroutines[46] containing the in-core and the out-of-core versions of the modification are available. The programmes contain Gram-Schmidt orthogonalisation and normalisation with respect to [M]. The parameter, ℓ, i.e. the number of $[K]^{-1}[M]$ operations, is a variable in the input data. For details of the FORTRAN computer programme and how to use it, see departmental report[46].

4.2.5 Operation counts

Take an operation as defined in section (3.6). Let n be the order of the operators [K] and [M], m_k and m_m the half-bandwidths of [K] and [M] respectively and q the dimensionality of the subspace. The number of operations required for the modification are given in table (3.2).

Operation	Number of Operations	
	$m = m_k = m_m$	$m = m_k$; $m_n = 0$
Form $[X]_{k+1}$ in equation (4.15)	nq(2m+1)	nq(2m+1)
Form $[Y]_{k+1}$ in equation (4.16)	nq(2m+1)	nq
Gram-Schmidt Ortho-normalisation	3n(2q-1)	3n(2q-1)
Total for each operation of $[K]^{-1}[M]$	n(4mq+8q-3)	n(2mq+8q-3)

TABLE 4.1. OPERATION COUNT FOR METHOD 1

Thus, if the iteration vectors are operated upon ℓ times by $[K]^{-1}[M]$, then the appropriate total number of operations is multiplied by ℓ. Hence, the total number of

operations in comparison to the original subspace iteration algorithm is increased $\ell n(4mq + 8q - 3)$ or $n\ell(2mq + 8q - 3)$ per iteration depending upon whether $m = m_k = m_m$ or $m_m = 0$ respectively.

4.2.6 Variation of the Basic Method

The basic method consists of operating upon the iteration vectors with the operators [K] and [M]. Consider now a similar scheme using $[K]^2$ and $[M]^2$. Clearly, if the bandwidth of the system matrices is large, then forming $[K]^2$ and $[M]^2$ is expensive and time consuming. However, the actual squares of the system matrices are never required; consider the square of the eigenproblem given in equation (4.1).

$$[K]^2\underline{v}_i = \lambda^2[M]^2\underline{v}_i \quad .. \quad .. \quad .. \quad (4.27)$$

The problem in equations (4.1) and (4.27) have the same eigenvectors but the eigenvalues of equation (4.27) are the squares of the eigenvalues of equation (4.1). Note that this method is referred to as method 1A throughout.

4.2.6.1 Algorithm

The requirement of the squared operators is made unnecessary by employing double back-substitution and forward reduction followed by double matrix multiplication. As in section (4.2.2) let $[Y]_1 = [M][X]_1$ where $[X]_1$ is given, then the algorithm proceeds as follows:

Double back-substitution and forward reduction

$$[K][K][X]_{k+1} = [Y]_k \quad .. \quad .. \quad .. \quad (4.28)$$

Double matrix multiplication

$$[\bar{Y}]_{k+1} = [M][M][X]_{k+1} \quad .. \quad .. \quad (4.29)$$

From this point onwards the original subspace iteration algorithm is used with $[\bar{Y}]_{k+1}$ as the initial subspace.

4.2.6.2 Implementation

The implementation of the above algorithm presents no difficulty since $[K]^2$ and $[M]^2$ are never required. The following approach was adopted in writing the programmes

$$\left.\begin{aligned} [K][W]_k &= [Y]_k \\ [K][X]_{k+1} &= [W]_k \end{aligned}\right\} \quad .. \quad .. \quad (4.30)$$

$$\left.\begin{aligned} [W]_{k+1} &= [M][X]_{k+1} \\ [\bar{Y}]_{k+1} &= [M][W]_{k+1} \end{aligned}\right\} \quad .. \quad .. \quad (4.31)$$

Note that the above approach allows the consideration of

$$[K]\,{}^{\ell}\underline{v}_i = \lambda_i^{\ell}[M]\,{}^{\ell}\underline{v}_i \quad .. \quad .. \quad .. \quad (4.32)$$

without any difficulties. The only requirement is to add the appropriate number of steps to equations (4.30) and (4.31). The programmes have been written so that the number of steps to be taken in equations (4.30) and (4.31), i.e. the value of ℓ in equation (4.32), is an input parameter.

This algorithm has been programmed[46] in both in-core and out-of-core forms.

4.2.6.3 Convergence

Consider now the convergence of this approach. Let $\underline{x}_i$ be an iteration vector as defined by equation (4.2). Equation (4.27) may be rewritten as

$$\frac{1}{\lambda_i^2}\underline{v}_i = [K]^{-2}[M]^2\underline{v}_i \qquad .. \quad .. \quad .. \quad .. \quad (4.33)$$

Let $\underline{\bar{z}}_i$ be the iteration vector formed when $\underline{x}_i$ is operated upon by $[K]^{-2}[M]^2$, then

$$\underline{\bar{z}}_i = [K]^{-2}[M]^2\underline{x}_i \qquad .. \quad .. \quad .. \quad .. \quad (4.34)$$

Substituting for $\underline{x}_i$ and using equation (4.33) in equation (4.34) yields

$$\underline{\bar{z}}_i = \sum_{j=1}^{n} a_{ij} \frac{1}{\lambda_j^2}\underline{v}_j \qquad .. \quad .. \quad .. \quad .. \quad (4.35)$$

Let $\bar{f}_c$ be the normalising factor for $\underline{\bar{z}}_i$, also let

$$f_c = \frac{\bar{f}_c}{\lambda_1^2}$$

so that the normalised iteration vector $\underline{z}_i$ is given by

$$\underline{z}_i = f_c \sum_{j=1}^{n} a_{ij} \left(\frac{\lambda_1}{\lambda_j}\right)^2 \underline{v}_j \qquad .. \quad .. \quad .. \quad (4.36)$$

Note that the factor $(\lambda_1/\lambda_j)^2$ in equation (4.36) is equivalent to $\bar{\ell}^{-2}$ where $\bar{\ell}$ is the factor defined in section (4.2.1).

Recall now the discussion in section (4.2.3.2), a similar situation exists here with two convergence

rates to consider. The main point of interest is the convergence rate due to equations (4.28) and (4.29), since the remainder of the algorithm is the original subspace iteration algorithm. The convergence rate of interest is clearly related to the factors $1/\bar{\ell}_j$ since it is these that govern the presence of the eigenvectors in any iteration vector.

Recall that in section (4.2.3.2) it was stated that ℓ, $[K]^{-1}[M]$ operations would lead to a factor of $(1/\bar{\ell})^{\ell}$ in equation (4.14), which is analogous to equation (4.36). In the present case the iteration vectors have been operated upon $[K]^{-2}[M]^2$. It can be seen from equation (4.36) that this gives rise to a factor of $(1/\bar{\ell})^2$. Thus, it follows that this is equivalent to carrying out two $[K]^{-1}[M]$ operations.

Hence, the conclusion here is that raising the problem to the power ℓ has the same effect on the convergence rate as operating upon the iteration vectors ℓ times $[K]^{-1}[M]$.

4.2.6.4 Numerical stability

The discussion in section (4.2.3.1) is directly applicable here, all the points made there are relevant here. As in the basic method, Gram-Schmidt orthogonalisation and normalisation with respect to [M] are incorporated into the programmes.

An important point to note is that, although theoretically, methods 1 and 1A have the same overall effect the numerical stability requires further consideration

in this case. There is an intuitive feeling here that this case is more likely to suffer from numerical instability than method 1. The basis for this is that in method 1, a $[K]^{-1}$ operation is followed by an [M] operation which acts as a balancing device in numerical terms. However, in method 1A, a $[K]^{-1}$ operation is followed by another $[K]^{-1}$ operation before the two [M] operations. Hence, the numerical stress in method 1A appears to be far greater than in method 1. Due to this, the threat of numerical instability will be realised far quicker in method 1A than in method 1, as the power of the problem and the number of $[K]^{-1}[M]$ operations are increased respectively.

4.2.6.5 Operation counts

The remarks made in section (4.2.5) as directly applicable here. The number of operations due to this method is approximately the same as that given in section (4.2.5).

4.3 Method 2

4.3.1 Theoretical considerations

Let $[Y]_{k-1}$ and $[Y]_k$ be the iteration vectors after the (k-2) and (k-1) iterations respectively. Then the basic idea is to form Y_{k+1} as a linear combination of $[Y]_{k-1}$ and $[Y]_k$ in an optimum manner. Thus, the required expression is of the form

$$[Y]_{k+1} = [Y]_k + \alpha\, [Y]_{k-1} \qquad .. \quad .. \quad (4.37)$$

where α, the coefficient to be determined is the optimising factor.

The necessary requirements are some criteria for determining the optimising factor, α. Consider the following argument, a necessary condition for the iteration vectors to be eigenvectors is that they diagonalise the system matrices. Therefore, a reasonable criterion is to require $[Y]_{k+1}$ to zero the off-diagonal terms in the system matrices. The matrix considered is [K], the stiffness matrix, since the mass matrix, [M], may be a diagonal matrix, and the expression to be formed is

$$I_{ij} = \left(\underline{y}_{k+1}^{i}\right)^{T} [K] \left(\underline{y}_{k+1}^{j}\right) \quad .. \quad .. \quad .. \quad (4.38)$$

where the superscript indicates the ith and jth iteration vectors. The condition that $[Y]_{k+1}$ diagonalise [K] leads to the following expression

$$I_{ij} = K'_{ij} \; \delta_{ij} \quad .. \quad .. \quad .. \quad (4.39)$$

where δ_{ij} is the Kronecker delta defined by

$$\delta_{ij} = \begin{cases} 1 \; , \; i = j \\ 0 \; , \; i \neq j \end{cases}$$

and K'_{ij} is the appropriate diagonal term.

Substituting for Y_{k+1} from equation (4.37) into equation (4.38) gives

$$I_{ij} = \left(\underline{y}_{k}^{i} + \alpha \, \underline{y}_{k-1}^{i}\right)^{T} [K] \left(\underline{y}_{k}^{i} + \alpha \, \underline{y}_{k-1}^{i}\right) \quad .. \quad (4.40)$$

Considering only the upper or lower triangle since the system matrices are symmetric,

$$I_{ij} = \left(\underline{y}_k^i\right)^T [K] \left(\underline{y}_k^j\right) + \alpha \left(\underline{y}_k^i\right)^T [K] \left(\underline{y}_{k-1}^j\right) + \alpha^2 \left(\underline{y}_{k-1}^i\right)^T [K] \left(\underline{y}_{k-1}^j\right) \quad .. \quad .. \quad .. \quad (4.41)$$

which may be written as

$$I_{ij} = a_{ij} + \alpha b_{ij} + \alpha^2 c_{ij} \quad .. \quad .. \quad .. \quad (4.42)$$

Summing all such I_{ij} gives

$$\bar{I} = \sum_{\substack{i=1 \\ i>j}}^{n} I_{ij} \quad .. \quad .. \quad .. \quad .. \quad (4.43)$$

Substituting equation (4.42) into equation (4.43) yields

$$\bar{I} = \bar{a} + \alpha \bar{b} + \alpha^2 \bar{c} \quad .. \quad .. \quad .. \quad .. \quad (4.44)$$

where

$$\bar{a} = \sum a_{ij} \ , \quad \bar{b} = \sum b_{ij} \ , \quad \bar{c} = \sum c_{ij} \quad .. \quad .. \quad (4.45)$$

Consider equation (4.44), $\bar{I}$ is a function of α and clearly needs to be minimised in order to obtain the required value of α. This follows from the fact that the eigenvectors diagonalise the system matrices. There exists a problem however, concerning the coefficients $\bar{a}$, $\bar{b}$ and $\bar{c}$, namely that if the negative and positive weighting of the off-diagonal terms is similar, then summing the a_{ij}, b_{ij} and c_{ij} would lead to zero without the individual terms being zero. The following two strategies were considered in order to overcome this problem.

Case (a):

The idea here is to try and set up an analogue problem in which the moduli of a_{ij}, b_{ij} and c_{ij} are summed. Clearly, if the sum of the moduli is zero, then the individual terms must also be zero. Let

$$J_{ij} = \left|a_{ij}\right| + \alpha \left|b_{ij}\right| + \alpha^2 \left|c_{ij}\right| \quad .. \quad .. \quad .. \quad (4.46)$$

Summing all such J_{ij} gives

$$\bar{J} = \sum_{\substack{i=1 \\ i>j}}^{n} J_{ij} \quad .. \quad .. \quad .. \quad .. \quad .. \quad .. \quad (4.47)$$

Substituting equation (4.46) into equation (4.47) leads to

$$\bar{J} = A + \alpha B + \alpha^2 C \quad .. \quad .. \quad .. \quad .. \quad .. \quad (4.48)$$

where

$$\bar{A} = \sum\left|a_{ij}\right|, \; B = \sum\left|b_{ij}\right|, \; C = \sum\left|c_{ij}\right| \quad .. \quad .. \quad (4.49)$$

Clearly, in equation (4.48), A, B and C are greater than or equal to zero by construction. The requirement now is to minimise $\bar{J}$, the condition for that is

$$\frac{d\bar{J}}{d\alpha} = 0 \quad .. \quad .. \quad .. \quad .. \quad .. \quad .. \quad (4.50)$$

differentiating equation (4.48) with respect to α gives

$$\frac{d\bar{J}}{d\alpha} = B + 2\alpha C \quad .. \quad .. \quad .. \quad .. \quad .. \quad (4.51)$$

Equations (4.50) and (4.51) now yield the value of α which minimises $\bar{J}$, namely

$$\bar{\alpha} = -\frac{B}{2C} \quad .. \quad .. \quad .. \quad .. \quad .. \quad (4.52)$$

Now, in practice, the values $\pm\bar{\alpha}$ and zero are substituted for α in equation (4.44) and the optimising

factor is taken to be the value which minimises $\bar{I}$. The reason for trying three values for α in equation (4.44) is that, $\bar{\alpha}$ minimises $\bar{J}$, which is defined in terms of moduli. Therefore, by construction $\bar{\alpha}$ is always negative. However, the coefficient $\bar{b}$ in equation (4.44) may be positive or negative. Hence, in some cases, $-\bar{\alpha}$ will be applicable in equation (4.44). Finally, the value $\alpha = 0$ is employed if the parabola defined by equation (4.44) is symmetric about the $\bar{I}$ axis.

Case (b):

In this case, instead of moduli, the squares of the off-diagonal terms are considered. Thus, squaring equation (4.42) yields

$$(I_{ij})^2 = (a_{ij})^2 + 2\alpha(a_{ij}b_{ij}) + \alpha^2[2(a_{ij}c_{ij}) + (b_{ij})^2] + 2\alpha^3(b_{ij}c_{ij}) + \alpha^4(c_{ij})^2 \quad .. \quad (4.53)$$

Summing all such $(I_{ij})^2$ gives

$$I' = \sum_{\substack{i=1 \\ i>j}}^{n} (I_{ij})^2 \quad .. \quad .. \quad .. \quad .. \quad (4.54)$$

Substituting from equation (4.53) into equation (4.54) leads to

$$I' = A + \alpha B + \alpha^2 C + \alpha^3 D + \alpha^4 E \quad .. \quad .. \quad (4.55)$$

where

$$\left.\begin{aligned} A &= \sum(a_{ij})^2; \quad B = 2\sum(a_{ij}b_{ij})^2; \quad C = \sum 2\{(a_{ij}c_{ij}) + (b_{ij})^2\} \\ D &= 2\sum(b_{ij}c_{ij})^2; \quad E = \sum(c_{ij})^2 \end{aligned}\right\} \quad .. \quad .. \quad .. \quad .. \quad (4.56)$$

The required value of α is obtained by minimising equation (4.55). The condition for this is,

$$\frac{dI'}{dx} = 0 \quad .. \quad .. \quad .. \quad .. \quad .. \quad (4.57)$$

differentiating equation (4.55) with respect to α gives,

$$\frac{dI'}{dx} = B + 2\alpha C + 3\alpha^2 D + 4\alpha^3 E \quad .. \quad .. \quad (4.58)$$

Cardano's formula for the roots of a cubic equation was employed to solve equation (4.58), see Appendix A. Note that since the coefficients B, C, D and E in equation (4.58) are real, there will be at least one real root. The optimising factor is then taken to be the value which minimises I'. Note that cases (a) and (b) will be referred to as Method 2 and Method 2A, respectively, throughout.

4.3.2 Algorithm

The notation of section (4.2.2) is employed in this section. In this method two initial estimates of the iteration vectors are required. Therefore, the original subspace iteration algorithm is employed for the first two iterations. In the course of these iterations, three

vector arrays are stored, namely $[Y]_1$, the initial iteration vectors, $[Y]_2$, the iteration vectors after the first iteration and $[Y]_3$, the iteration vectors after the second iteration, then the subsequent iterations proceed as follows: For k = 3,4, ... form $[Y]'_{k+1}$ as an optional linear combination of $[Y]_k$ and $[Y]_{k-1}$,

$$[Y]'_{k+1} = [Y]_k + \alpha[Y]_{k-1} \quad .. \quad .. \quad .. \quad (4.59)$$

forward reduction and back substitution

$$[K][X]_{k+1} = [Y]'_{k+1} \quad .. \quad .. \quad .. \quad .. \quad (4.60)$$

Matrix multiplication

$$[\bar{Y}]_{k+1} = [M][X]_{k+1} \quad .. \quad .. \quad .. \quad .. \quad (4.61)$$

form the protections of [K] and [M] onto the subspace.

$$[\bar{K}] = [X]^T_{k+1}[Y]'_{k+1} \quad .. \quad .. \quad .. \quad .. \quad (4.62)$$

$$[\bar{M}] = [X]^T_{k+1}[\bar{Y}]_{k+1} \quad .. \quad .. \quad .. \quad .. \quad (4.63)$$

Solve the eigensystem of subspace operators,

$$[\bar{K}][Q] = [\bar{M}][Q][\Lambda] \quad .. \quad .. \quad .. \quad .. \quad (4.64)$$

Update the iteration vector estimate

$$[Y]_{k+1} = [\bar{Y}]_{k+1}[Q] \quad .. \quad .. \quad .. \quad .. \quad (4.65)$$

Update the previous generations of the iteration vectors for use in equation (4.59)

$$\left.\begin{aligned} [Y]_{k-1} &= [Y]_k \\ [Y]_k &= [Y]_{k+1} \end{aligned}\right\} \quad .. \quad .. \quad .. \quad .. \quad (4.66)$$

4.3.3 Numerical Considerations

4.3.3.1 Numerical Stability

This method becomes unstable when the $[Y]_k$ and $[Y]_{k-1}$ in equation (4.59) are very close to the least dominant subspace. The reason for this is that the ith columns in $[Y]_k$ and $[Y]_{k-1}$ are converging to the ith eigenvector and the ith column of $[Y]_{k+1}$ is formed by linearly combining the ith columns of $[Y]_k$ and $[Y]_{k-1}$. Now, when the ith columns of $[Y]_k$ and $[Y]_{k-1}$ are both reasonable approximations to the ith eigenvector, then any linear combination of these will also be a reasonable approximation to the eigenvector, irrespective of the value of α.

Thus, for a stable process $[Y]_k$ and $[Y]_{k-1}$ are required to differ significantly from each other. A consequence of this is that convergence is required to occur before $[Y]_k$ and $[Y]_{k-1}$ become too similar. Hence, the convergence criterion should be strict enough to obtain the required accuracy but not too strict so that $[Y]_k$ and $[Y]_{k-1}$ are not allowed to become too similar. Experience has shown that stability is lost when $|\alpha| \leq 0.05$. This clearly represents very little change in the iteration vectors from iteration to iteration. It also suggests that the convergence tolerance is too strict. In practice, best results were achieved when $|\alpha| \leq 0.35$. This verifies the intuitive feeling that once convergence has set-in and $[Y]_k$ and $[Y]_{k-1}$ are not too similar, then $|\alpha|$ is bounded by

unity. Thus, the iteration vectors in $[Y]_k$ are 'trimmed' by a small amount using the iteration vectors in $[Y]_{k-1}$.

If the convergence rate in practice is fast then this method is stable, i.e. there is no, or very little, similarity between the $[Y]_k$ and $[Y]_{k-1}$. However, if the convergence rate is slow, then this method should be used with care, since the degree of similarity between the $[Y]_k$ and $[Y]_{k-1}$ will be high. This clearly indicates that this method loses its potency as the iteration vectors tend to the eigenvectors. An important conclusion may be drawn here, namely that there exists a 'useful' range in which this method may be applied.

Note that the preceding discussion is only applicable provided subspace iteration is converging. The effects of this method are not clear if subspace iteration is not converging.

4.3.3.2 Convergence

As in section (4.2.3.2) it is necessary to take into account two convergence rates. This follows from the fact that the technique described here is applied at the level of forming the initial subspace iteration. The two rates of convergence to be considered are:

(i) The rate of convergence due to equation (4.37)

(ii) The rate of convergence due to the original subspace iteration algorithm.

The discussion in section (3.5.2) on the convergence of subspace iteration, is clearly relevant for (ii).

Consider now the convergence rate due to (i). This clearly depends upon the state of $[Y]_k$ and $[Y]_{k-1}$. A necessary requirement for equation (4.39) to yield a 'good' $[Y]_{k+1}$ is that $[Y]_k$ is a better estimate of the initial subspace than $[Y]_{k-1}$. The worst case occurs when $\alpha = 0$ in equation (4.37). In this case, $[Y]_{k+1}$ is not an improvement on $[Y]_k$. In all other cases, provided $[Y]_k$ is a better approximation of the initial subspace than $[Y]_{k-1}$, $[Y]_{k+1}$ will be an improvement on $[Y]_k$.

The degree of improvement depends upon how close $[Y]_k$ is to $[Y]_{k-1}$. As $[Y]_k$ gets closer to $[Y]_{k-1}$, the degree of improvement decreases. In other words, when the iteration vectors have nearly converged, the work done in calculating α offers little or no return in accelerating convergence. In fact, as was mentioned in section (4.3.3.1), a high degree of similarity between $[Y]_k$ and $[Y]_{k-1}$ leads to numerical suitability.

The expectation is that this method will converge faster than the original subspace iteration algorithm in the first few iterations. It would then be reasonable to expect a higher convergence rate than the original algorithm when the iteration vectors are 'far' from convergence, i.e. when the iteration vectors are not good approximations of the eigenvectors. The convergence rate will then approach that of the original algorithm as the iteration vectors approach the eigenvectors.

4.3.4 Implementation

The two cases considered in section (4.3.1) have been programmed in such a way that either may be inserted into a working subspace iteration programme. Subroutines for running an in-core or out-of-core solution are available.

The subspace iteration algorithm is such that, given $[Y]_k$, $[X]_{k+1}$ is obtained from the following expression:

$$[K][X]_{k+1} = [Y]_k \quad .. \quad .. \quad .. \quad .. \quad (4.67)$$

Consider now equations (4.38) - (4.41). It is clear that forming $[X]^T[K][X]$ type products is vital in this method. The required expressions are of the following form:

$$\left.\begin{array}{l} [X]_k^T[K][X]_k \\ [X]_k^T[K][X]_{k-1} \\ [X]_{k-1}^T[K][X]_{k-1} \end{array}\right\} \quad .. \quad .. \quad .. \quad (4.68)$$

Substituting from equation (4.67) into equation (4.68) leads to

$$\left.\begin{array}{l} [X]_k^T[Y]_{k-1} \\ [X]_k^T[Y]_{k-2} \\ [X]_{k-1}^T[Y]_{k-2} \end{array}\right\} \quad .. \quad .. \quad .. \quad (4.69)$$

Hence, by storing $[Y]_{k-2}$ and $[Y]_{k-1}$, the product in equation (4.67) need not be formed again. At the end of each iteration, the vectors in $[Y]_{k-2}$ and $[Y]_{k-1}$, are updated, as given by equation (4.66).

The approach described above has been adapted in the programmes[46]. In the out-of-core version of this programme, two working tapes are specified to store $[Y]_{k-2}$ and $[Y]_{k-1}$.

4.3.5 Operation counts

Let the definition of an operation be as given in section (3.6). Also let n be the order of the system matrices, [K] and [M], m_k and m_m the half-bandwidths of [K] and [M] respectively, and q the dimension of the iterating subspace. Consider the two cases described in section (4.3.1) separately.

Operations	Number of Operations	
	Case (a)	Case (b)
Operations due to equation (4.37)	nq	nq
Operations due to equation (4.69)	3n	6n
Total number of operations/iterations	n(3 + q)	n(6 + q)

TABLE 4.2. OPERATION COUNT FOR METHOD 2 AND METHOD 2A

Therefore, the increase in the number of operations per iteration relative to the original subspace iteration algorithm is approximately n(3 + q) and n(6 + q) for cases (a) and (b) respectively.

CHAPTER 5

NUMERICAL INVESTIGATION

5.1 Introduction

The ideas pursued theoretically in the previous chapter are investigated numerically in this chapter. The primary concern here is to validate the methodology and de-bug the coding.

Initially, a discussion of the course the numerical investigation may follow is presented. This is followed by an outline of the basic strategy. The results obtained by solving simple problems are then compared with known solutions. Two such cases are considered, namely problems modelled by using beam and plate elements. The insights gained from these comparisons are then discussed.

The purpose of this chapter is to give an indication of the efficiency of the modified algorithm when presented with a variety of problems. These problems are also solved using the original subspace iteration algorithm in order to gauge the relative merit of the modified algorithm.

In order to run the solution algorithms, the stiffness, [K] and mass, [M] matrices are required. These are generated by FINEL[2]. It was, therefore, necessary to couple[46] this general purpose finite element package to the solution algorithms.

Initially, a general comparison of the original and modified algorithms is carried out in two dimensional space. The problems considered are:

(i) A cantilevered beam, modelled by two noded beam elements with three degrees of freedom per node. This type of element was also used to model a plane frame.

(ii) A cantilevered plate of constant thickness, modelled by eight noded plate elements with three degrees of freedom per node.

The results of these comparisons are presented and discussed.

A detailed comparison of the original and modified algorithms is then carried out. Here, selected problems from the general comparison are used along with a three dimensional problem. This consists of modelling an off-shore structure using two noded beam elements with six degrees of freedom per node. The results of these comparisons are presented and discussed.

The insights gained and ideas generated from the aforementioned comparisons are put into practice. Selected problems from those solved previously are solved again with regard to the new ideas. The results obtained are presented and conclusions are drawn from them.

In all cases, the initial subspace was generated as described in section (3.4) and the convergence tolerance for the eigenvalues was, TOL = 10^{-4} throughout, see section (3.5.2). The times taken for the initial matrix factorisation, solution of the subspace eigensystem, t_e, and the modification, t_m, were all part of the iteration time, t_I. However, t_e and t_m were also timed separately.

All the work was carried out on the Sheffield University ICL 1906S computer with a maximum core space of 875K words and a 39 bit mantissa for reals. The frequencies and times are given in Hertz and CPU seconds respectively throughout.

5.2. Philosophical considerations

It is very difficult, if not impossible, to predict the course an investigation may take, since initial findings may lead to the pursuit of ideas previously not considered. Contrary to that, however, is the fact that it is similarly very difficult to carry out an investigation without an overall work plan, a general framework for the investigation, say. Such a framework is a necessary requirement in order to give coherence to the investigation. However, flexibility in the framework is of paramount importance so as not to discard interesting avenues of investigation by rigid devotion to the initial plan.

The primary aim of this investigation is to verify the methodology and the coding for the general case and not just to solve a large problem. The order of the problems considered could have been larger, but this would only have increased the solution time without changing the solution characteristic. This would also have caused inconvenience due to certain restrictions in the system.

A parameter which is very important in this investigation is NIIT[46]. This indicates the number of $[K]^{-1}[M]$ operations in method 1, see section (4.2.3.2) or

the power to which the eigenproblem is raised, see section (4.2.6). Another important parameter is NROOT[46]. This specifies the number of required eigenpairs and governs the size of the iterating subspace. A secondary aim of this investigation is a parameter survey concerning the two aforementioned parameters. Note, however, that, although it is preferable for a parameter survey to take into account the general case, it may not be practical. This is due to the resources available. For example, time, finance, facilities, etc.

Consider now the size of a test problem, the order of a 'real problem' will be large, clearly this is not the ideal type of problem for testing the methodology and coding. A desirable quality in a test problem is simplicity, so that each stage may be worked by hand if required. Another important point to consider is that carrying out a parameter survey, using a 'real problem' without first establishing some ground rules, is not practical. Such an exercise will be a long drawn process fraught with computing difficulties, expensive in terms of computer time and may be altogether impossible.

From the preceding discussion, it follows that trying to run before the art of walking has been mastered is rarely rewarding and always dangerous. Therefore, with regard to this investigation, it would appear prudent to begin at a very basic level and progressively increase the order of complexity as confidence in the methodology and coding increases.

A tentative framework for the investigation follows:

(i) Initially, choose a simple problem to verify the methodology and the coding. The simplest element available in finite element analysis is the two noded beam element with three degrees of freedom per node. Therefore, it is natural to use it in formulating a simple problem, namely a cantilevered beam.

(ii) Test the validity of the results obtained when the above problem is presented to the original algorithm by comparison with results obtained from FINEL[2] and simple beam theory[6].

(iii) Use a different element to test the original algorithm in order to rule out any element dependence. The element intended for use here is the eight noded plate bending element with three degrees of freedom per node.

(iv) The validity of the results obtained when the original algorithm is used may be tested by comparison with results obtained from FINEL[2] and an alternative solution[47].

(v) The modified algorithms are then presented with a number of problems using the two previously mentioned element types. The reliability of the modified algorithms may then be established by comparison with the original algorithm.

(vi) There remains then to test the reliability of the modified algorithms under adverse numerical conditions which may arise in a 'real problem'. It is, therefore, necessary to obtain a simple model of a 'real problem' without destroying the characteristics of the 'real problem'. Since the system matrices are generated by FINEL^2, a working programme, the characteristics of a 'real problem', such as bandedness, positive definiteness, etc., are assumed to be present.

(vii) The intention is to model and off-shore (Rig) structure using two noded beam elements with six degrees of freedom per node.

5.3 Comparison with known solutions

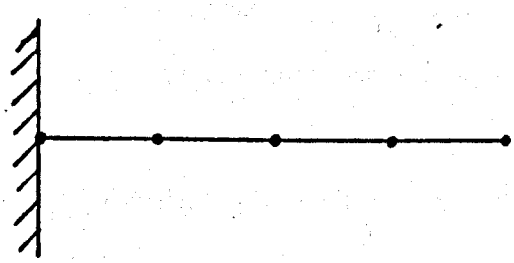
5.3.1 Beams

A comparison is made of the lowest five natural frequencies when:

(i) Calculated using the subspace, iteration algorithm.

(ii) Calculated using simple beam theory.

(iii) Calculated using FINEL^2.



Young's Modulus, E	$= 1.59 \times 10^{11}\ _{NM^{-2}}$
Moment of inertia, I	$= 7.854 \times 10^{-9}\ _{M^4}$
Density, ρ	$= 7.7 \times 10^{-3}\ _{KgM^{-3}}$
Poisson's ratio, ν	$= 0.3$
Cross-sectional area, A	$= 3.142 \times 10^{-4}\ _{M^2}$
length, a	$= 1$ M

FIG. 5.1. CANTILEVERED BEAM

The results obtained from (i), (ii) and (iii), are presented in Table 5.1.

Frequency Number	(i)	(ii)	(iii)
1	12.71	12.71	12.71
2	79.68	79.67	79.77
3	223.23	223.10	224.80
4	438.15	437.21	443.53
5	726.85	722.65	824.85

TABLE 5.1. COMPARISON WITH FINEL AND SIMPLE BEAM THEORY FOR A CANTILEVERED BEAM

The beam in figure 5.1 was modelled by four, two moded beam elements with three degrees of freedom per mode.

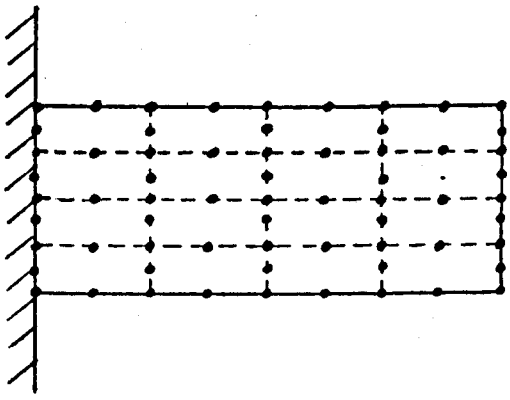
5.3.2 Plates

As in section (5.3.1) a comparison is made of the lowest five natural frequencies when:

(i) Calculated using the subspace iteration algorithm.

(ii) Calculated from an approximate formula derived from the Rayleigh-Ritz method assuming waveforms similar to those of beams[47].

(iii) Calculated using FINEL[2].



Young's Modulus, $E = 3 \times 10^7$ lb/in^2

Density, $\rho = 0.283$ lb/in^3

Poisson's ratio, $\nu = 0.3$

Length, a = 15 in

Width, b = 6 in

thickness, h = 0.236 in

FIG. 5.2. CANTILEVERED PLATE

The results obtained from (i), (ii) and (iii), are presented in Table 5.2.

Frequency Number	(i)	(ii)	(iii)
1	35.05	35.57	35.07
2	183.63	222.77	183.68
3	218.74	226.23	218.86
4	584.72	623.79	584.56
5	614.43	630.59	614.55

TABLE 5.2. COMPARISON WITH FINEL AND AN ALTERNATIVE SOLUTION FOR A CANTILEVERED PLATE

The plate in Fig. 5.2 was modelled by sixteen, eight noded plate elements with three degrees of freedom per node. The mesh is shown in Fig. 5.2.

5.3.3 Discussion

In sections (5.3.1) and (5.3.2), two independent problems have been presented to the original subspace

iteration algorithm. The results along with those obtained from FINEL and the known approximate solutions are given in Tables 5.1 and 5.2.

Consider first Table 5.1. Here, the discrepancy between (i) and (ii) is less than 0.6%. The discrepancy between (i) and (iii) is less than 1.3% for the first four natural frequencies. However, there is a discrepancy of approximately 12% in the fifth natural frequency. A possible explanation for this is that the FINEL solution routine employs an iterative process, which is terminated once the percentage change in the eigenvalue estimate satisfies some arbitrary criterion. Clearly, this is a necessary condition for convergence. However, the sufficiency part of the convergence criteria is not tested. This casts doubt on the integrity of the solution and thus, in this case, subspace iteration is an improvement on FINEL.

Consider now Table 5.2. Here the agreement between (i) and (iii) is satisfactory throughout. The disagreement between (i) and (iii) is less than 6.4% with the exception of the second natural frequency, where the discrepancy is large, however, this is the expected result[47].

Thus, the methodology and the coding are validated. This enables the original subspace iteration to be used as a 'bench mark' when compared with the modified algorithms. Note, however, that coding can rarely be purged completely of errors and even coding de-bugged to the highest standards can run into trouble if some infrequently occurring set of circumstances arise.

5.4 The available options

Consider now the available modifications: basically there are methods 1, 1A, 2 and 2A. In addition to these, a number of modifications may be constructed by combining the four basic modifications.

Clearly, since method 1A is a variation of method 1, combining these is of no interest. Similar remarks apply to method 2 and method 2A. Thus, a further four modifications are available if the basic modifications are combined. The following definitions[46] are employed throughout the thesis when referring to any particular algorithm.

The original algorithm ≡ SSP .. .. .. (5.1)

Method 1 ≡ XXSS
Method 1A ≡ XSSP
Method 2 ≡ ZSSP
Method 2A ≡ YSSP
.. .. .. (5.2)

Method 1 and Method 2 ≡ XXZSS
Method 1 and Method 2A ≡ XXYSS
Method 1A and Method 2 ≡ XZSS
Method 1A and Method 2A ≡ XYSS
(5.3)

Thus, a total of nine solution options are available in analysing any given data set. Note that combining the basic options presents no difficulty since each modification is contained in a subroutine[46] and may be inserted into the subspace iteration programme at the appropriate point.

The intuitive feeling here is that the options defined by equation (5.2) would be the closest rivals to

the original algorithm since, in the combinations, the extra workload per iteration is being increased to a high degree. Clearly, if the increase in the workload cannot be justified, then the offending option must be neglected since it is defeating the purpose of this work.

5.5 General comparison of modifications

The purpose of this general comparison is to discard those options which offer no advantage and concentrate on those which show the most potential. The following two sections are concerned with beam and plate type problems, respectively. In each of these, a table containing the order of the problem, N, number of iterations, N_c and time, t_c, for the convergence of the required eigenvalues are given for each option. This is followed by another table giving the solution characteristic in more detail. This table contains the relative tolerance, RTOL reached after each iteration for the lowest five eigenvalues. RTOL is defined as follows:

$$RTOL = \frac{\lambda^i - \lambda^{i-1}}{\lambda^i} \qquad .. \quad .. \quad .. \quad (5.4)$$

where λ^i is the eigenvalue approximation after the ith iteration and λ^o is taken as zero.

The problems selected for detailed observation are generally those of the highest order from the relevant problem types. Note that, in the beams section, two

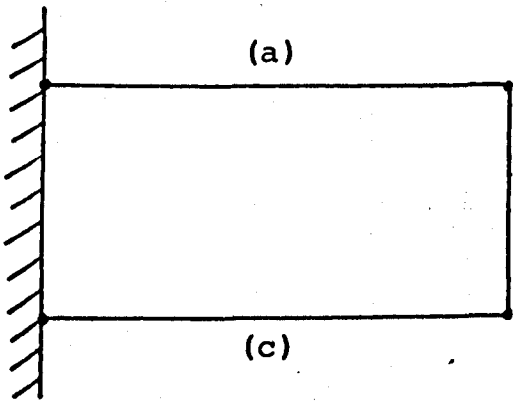
problems are considered in detail since both the cantilevered beam and plane frame are analysed in this section.

The number of eigenvalues required, NROOT = 5 and RTOL = 10^{-4} throughout in the following analyses unless stated otherwise.

5.5.1 Beams

A cantilevered beam was modelled using a varying number of two noded beam elements with three degrees of freedom per node. The results obtained are given in Table 5.3. In problems 1 and 2, NROOT was specified as 1 and 2 respectively, due to the small size of these problems. Note that, in problem 3, numerical difficulties were encountered whenever the XSSP option was used.

A cantilevered plane frame was modelled, see Fig. 5.3, using two noded beam elements with three degrees of freedom per node.



a = 1 m; b = 0.5 m. All other physical properties are as in the cantilevered beam case.

FIG. 5.3. CANTILEVERED PLANE FRAME

A number of problems were formulated by varying the number of elements used in modelling (a), (b) and (c). The results obtained are given in Table 5.4.

PROBLEM NUMBER	PROBLEM SIZE N	NUMBER OF ITERATIONS FOR LOWEST FIVE EIGENVALUES TO CONVERGE, $NITE_c$ TIME TAKEN FOR LOWEST FIVE EIGENVALUES TO CONVERGE, t_c								
		SSP	XXSS	XSSP	ZSSP	YSSP	XXZSS	XXYSS	XZSS	XYSS
1	3	3 0.019	2 0.021	2 0.020	3 0.021	3 0.023	2 0.021	2 0.019	2 0.020	2 0.021
2	6	3 0.030	2 0.083	2 0.087	3 0.079	3 0.080	2 0.082	2 0.083	2 0.086	2 0.085
3	12	2 0.492	2 0.561	-	2 0.492	2 0.492	2 0.560	2 0.559	-	-
4	24	3 0.928	2 0.975	2 1.071	3 1.007	3 1.009	2 0.975	2 0.975	2 1.071	2 1.070

TABLE 5.3. GENERAL COMPARISON OF THE SSP OPTIONS AND MODIFICATIONS FOR A CANTILEVERED BEAM

PROBLEM NUMBER	PROBLEM SIZE N	NUMBER OF ITERATIONS FOR LOWEST FIVE EIGENVALUES TO CONVERGE, $NITE_c$ TIME TAKEN FOR LOWEST FIVE EIGENVALUES TO CONVERGE, t_c								
		SSP	XXSS	XSSP	ZSSP	YSSP	XXZSS	XXYSS	XZSS	XYSS
5	6	3 0.069	2 0.082	2 0.088	3 0.073	3 0.075	2 0.080	2 0.080	2 0.086	2 0.086
6	12	3 0.617	2 0.541	4 1.537	3 0.658	3 0.665	2 0.540	2 0.540	4 1.614	4 1.615
7	15	3 0.704	2 0.640	3 1.218	3 0.754	3 0.755	2 0.641	2 0.639	3 1.267	3 1.265
8	24	3 0.944	2 0.955	2 1.101	3 1.023	3 1.023	2 0.955	2 0.955	2 1.101	2 1.101
9	27	3 0.981	2 1.052	2 1.167	3 1.069	3 1.070	2 1.053	2 1.053	2 1.169	2 1.169
10	33	4 1.521	2 1.320	3 2.120	4 1.736	4 1.733	2 1.321	2 1.321	3 2.224	3 2.226
11	48	4 1.902	2 1.786	2 1.898	4 2.204	4 2.208	2 1.788	2 1.788	2 1.901	2 1.900
12	51	4 1.988	2 1.906	3 2.949	4 2.307	4 2.314	2 1.909	2 1.910	3 3.110	3 3.111
13	57	4 2.128	2 2.114	2 2.188	4 2.485	4 2.488	2 2.117	2 2.118	2 2.190	2 2.189

TABLE 5.4. GENERAL COMPARISON OF THE SSP OPTION AND MODIFICATIONS FOR A PLANE FRAME

Consider now problems 4 and 13 in more detail: the relative tolerances reached after each iteration are given in Tables 5.5 and 5.6 respectively for the five lowest eigenvalues. All the available options were taken into account. However, it was found that the options defined by equation (5.3), the combinations, reproduced the results obtained by the XXSS and XSSP options. Thus, the XXSS and XSSP options were completely overwhelming the YSSP and ZSSP options. Note that NITE denotes the iteration number.

5.5.2 Plates

A cantilevered plate is modelled using eight node plate elements with three degrees of freedom per node. A number of problems are generated by varying the number of elements used in the model. The results obtained are given in Table 5.7.

Consider now problem 17 in more detail. Table 5.8 contains the relative tolerances reached after each iteration for the lowest five eigenvalues. As in section (5.5.1), it was found that the options defined by equation (5.3), the combinations reproduced the results obtained by the XXSS and XSSP options.

5.5.3 Discussion

An inspection of Tables 5.3 to 5.8 lead to the following conclusions:

(i) Tables 5.3, 5.4 and 5.7 verify the intuitive feeling that no advantage is gained by combining the basic modifications. In fact, in the

EIGEN-VALUE NUMBER	NITE	RELATIVE TOLERANCE, RTOL								
		SSP	XXSS	XSSP	ZSSP	YSSP	XXYSZ	XXYSS	XZSS	XYSS
1	1 2 3	1.0 5.6×10^{-6} 0.0	1.0 9.3×10^{-12}	1.0 5.6×10^{-11}	1.0 2.9×10^{-6} 0.0	1.0 2.86×10^{-8} 9.34×10^{-12}				
2	1 2 3	1.0 1.07×10^{-4} 0.0	1.0 3.8×10^{-11}	1.0 1.1×10^{-9}	1.0 1.1×10^{-4} 7.6×10^{-12}	1.0 1.07×10^{-4} 1.52×10^{-11}				
3	1 2 3	1.0 9.27×10^{-4} 4.16×10^{-9}	1.0 $3.87.10^{-11}$	1.0 2.1×10^{-7}	1.0 9.3×10^{-4} 4.2×10^{-9}	1.0 9.27×10^{-4} 4.19×10^{-9}	As in XXSS	As in XXSS	As in XSSP	As in XSSP
4	1 2 3	1.0 4.25×10^{-3} 3.35×10^{-7}	1.0 4.03×10^{-11}	1.0 3.7×10^{-7}	1.0 4.25×10^{-3} 3.35×10^{-9}	1.0 4.25×10^{-3} 3.35×10^{-7}				
5	1 2 3	1.0 1.23×10^{-2} 1.04×10^{-5}	1.0 1.23×10^{-8}	1.0 3.1×10^{-6}	1.0 1.23×10^{-2} 1.04×10^{-5}	1.0 1.23×10^{-2} 1.04×10^{-5}				

TABLE 5.5. SOLUTION CHARACTERISTIC OF PROBLEM 4

EIGEN-VALUE NUMBER	NITE	RELATIVE TOLERANCE, RTOL								
		SSP	XXSS	XSSP	ZSSP	YSSP	XXZSS	XXYSS	ZXSS	XYSS
1	1	1.0	1.0	1.0	1.0	1.0				
	2	6.8×10^{-4}	2.8×10^{-11}	2.9×10^{-10}	6.8×10^{-4}	6.8×10^{-4}				
	3	2.1×10^{-10}			2.1×10^{-10}	1.9×10^{-10}				
	4	2.8×10^{-11}			1.4×10^{-11}	2.8×10^{-11}				
2	1	1.0	1.0	1.0	1.0	1.0				
	2	5.45×10^{-3}	0.0	1.5×10^{-7}	5.45×10^{-3}	5.45×10^{-3}				
	3	7.6×10^{-7}			7.6×10^{-7}	7.6×10^{-7}				
	4	1.2×10^{-11}			7.0×10^{-11}	5.9×10^{-11}				
3	1	1.0	1.0	1.0	1.0	1.0	As in XXSS	As in XXSS	As in XSSP	As in XSSP
	2	4.1×10^{-2}	2.3×19^{-9}	7.3×10^{-8}	4.1×10^{-2}	4.5×10^{-2}				
	3	2.0×10^{-5}			2.0×10^{-5}	2.0×10^{-5}				
	4	2.2×10^{-9}			2.2×10^{-9}	2.2×10^{-9}				
4	1	1.0	1.0	1.0	1.0	1.0				
	2	5.3×10^{-2}	1.4×10^{-6}	6.6×10^{-6}	5.3×10^{-2}	5.3×10^{-2}				
	3	5.6×10^{-4}			5.6×10^{-4}	5.6×10^{-4}				
	4	1.4×10^{-6}			1.4×10^{-6}	1.4×10^{-6}				
5	1	1.0	1.0	1.0	1.0	1.0				
	2	7.6×10^{-2}	8.6×10^{-6}	6.5×10^{-5}	7.6×10^{-2}	7.6×10^{-2}				
	3	1.5×10^{-3}			1.5×10^{-3}	1.5×10^{-3}				
	4	8.6×10^{-6}			8.6×10^{-6}	8.6×10^{-6}				

TABLE 5.6. SOLUTION CHARACTERISTIC OF PROBLEM 13

PROBLEM NUMBER	PROBLEM SIZE N	NUMBER OF ITERATIONS FOR LOWEST FIVE EIGENVALUES TO CONVERGE, $NITE_c$ TIME TAKEN FOR LOWEST FIVE EIGENVALUES TO CONVERGE, t_c								
		SSP	XXSS	XSSP	ZSSP	YSSP	XXZSS	XXYSS	XZSS	XYSS
14	15	3 0.318	2 0.423	3 0.682	3 0.338	3 0.338	2 0.423	2 0.425	3 0.699	3 0.701
15	60	4 4.887	2 6.219	4 12.602	4 5.27	4 5.263	2 6.222	2 6.222	4 12.984	4 12.979
16	96	4 8.466	2 11.362	6 34.222	4 9.080	4 9.07	2 11.396	2 11.368	6 35.434	6 35.414
17	132	4 12.058	2 16.542	6 49.611	4 12.897	4 12.887	2 16.55	2 16.552	6 51.284	6 51.249
18	168	4 15.65	2 21.65	4 43.349	4 16.719	4 16.702	2 21.663	2 21.666	4 44.410	4 44.40

TABLE 5.7. GENERAL COMPARISON OF THE SSP OPTION AND MODIFICATIONS FOR A CANTILEVERED PLATE

EIGEN-VALUE NUMBER	NITE	RELATIVE TOLERANCE, RTOL								
		SSP	XXSS	XSSP	ZSSP	YSSP	XXZSS	XXYSS	ZXSS	XYSS
1	1	1.0	1.0	1.0	1.0	1.0				
	2	1.1×10^{-4}	2.0×10^{-11}	2.6×10^{-6}	1.1×10^{-4}	1.1×10^{-4}				
	3	0.0		1.4×10^{-6}	1.1×10^{-10}	3.9×10^{-11}				
	4	2.0×10^{-11}		3.7×10^{-7}	9.8×10^{-11}	2.9×10^{-11}				
	5			1.2×10^{-7}						
	6			4.0×10^{-8}						
2	1	1.0	1.0	1.0	1.0	1.0	As in XXSS	As in XXSS	As in XSSP	As in XSSP
	2	2.6×10^{-2}	7.9×10^{-10}	1.3×10^{-5}	2.6×10^{-2}	2.6×10^{-2}				
	3	4.8×10^{-6}		1.1×10^{-6}	4.8×10^{-6}	4.8×10^{-6}				
	4	7.9×10^{-10}		1.2×10^{-7}	9.2×10^{-10}	7.8×10^{-10}				
	5			1.3×10^{-8}						
	6			1.4×10^{-9}						
3	1	1.0	1.0	1.0	1.0	1.0				
	2	1.6×10^{-2}	1.2×10^{-6}	3.0×10^{-4}	1.6×10^{-2}	1.6×10^{-2}				
	3	7.5×10^{-7}		5.8×10^{-5}	7.5×10^{-7}	7.5×10^{-7}				
	4	4.8×10^{-11}		6.8×10^{-5}	1.5×10^{-10}	4.0×10^{-11}				
	5			2.7×10^{-5}						
	6			9.2×10^{-6}						
4	1	1.0	1.0	1.0	1.0	1.0				
	2	1.6×10^{-1}	9.4×10^{-6}	1.5×10^{-4}	1.6×10^{-1}	1.6×10^{-1}				
	3	8.0×10^{-4}		1.4×10^{-5}	8.0×10^{-4}	8.0×10^{-4}				
	4	9.2×10^{-6}		1.5×10^{-6}	9.2×10^{-6}	9.2×10^{-6}				
	5			1.6×10^{-7}						
	6			1.7×10^{-8}						
5	1	1.0	1.0	1.0	1.0	1.0				
	2	1.4×10^{-1}	5.7×10^{-6}	1.1×10^{-3}	1.4×10^{-1}	1.4×10^{-1}				
	3	5.9×10^{-4}		1.1×10^{-3}	5.9×10^{-4}	5.9×10^{-4}				
	4	5.6×10^{-6}		3.6×10^{-4}	5.6×10^{-6}	5.6×10^{-6}				
	5			1.0×10^{-4}						
	6			3.2×10^{-5}						

TABLE 5.8. SOLUTION CHARACTERISTIC OF PROBLEM 17

options defined by equation (5.3), the XXSS and XSSP part of the combination appears to completely overwhelm the ZSSP and YSSP options.

(ii) The ZSSP and YSSP options appear to be passive as far as changing the solution characteristic is concerned.

(iii) The XSSP option is behaving 'wildly' in numerical terms. In exact arithmetic, this should show complete agreement with the XXSS option, see section (4.2.6.3). However, in finite arithmetic, the XSSP option is more susceptible to numerical instability than the XXSS option, see section (4.2.6.4.). This numerical instability manifested itself most pointedly in the case of problem 3 and, as can be seen from Table 5.3, no results were obtained whenever the XSSP option was employed. The problem was then re-run with NROOT = 4 and the results obtained matched up with established results. Problem 4 was then re-run with NROOT = 10, i.e. with the same N/NROOT ratio which caused problem 3 to fail. In this case, problem 4 also failed; similar results were obtained from other such experiments. This indicates that increasing the size of the subspace to a significant percentage of the whole space causes pronounced numerical instabilities in the XSSP option.

(iv) The XXSS option appears to show the most potential. It can be seen from Tables 5.5, 5.6 and 5.8 that

the initial convergence rate is very fast when compared to that of the SSP option.

With regard to the above-mentioned points, it would appear prudent to discard those options which consist of combining the basic modifications, i.e. the options defined by equation (5.3). The ZSSP and YSSP options are retained in the hope that varying some parameters may cause these to prove themselves one way or the other. The XSSP option is also discarded herewith since it is only an unstable variation of the XXSS option.

5.6 Detailed comparison of modifications

In this section, the XXSS, ZSSP and YSSP options are compared with the original subspace iteration algorithm (SSP option) and with each other in detail. Each iteration step in the solution consists of the following:

(i) Modifying the iteration vectors (if the modified algorithm is employed).

(ii) Forming the reduced eigenproblem in the subspace.

(iii) Solving the reduced eigenproblem.

(iv) Obtaining the iteration vectors for the next iteration if convergence has not occurred.

Clearly, the time taken for (ii) and (iv) will be identical for a given set of parameters, but for (i) and (iii), it will vary according to the solution algorithm employed. The time taken for the modification, t_m is a necessary requirement for comparison purposes and also to give an indication of the increase in the time taken for each iteration, t_I due to the modification. Consider now

the time taken to solve the reduced eigenproblem, t_e. This is very important in appraising the condition of the subspace operators $[\bar{K}]$ and $[\bar{M}]$. Clearly, the degree of diagonality of $[\bar{K}]$ and $[\bar{M}]$ is indicated by the magnitude of t_e.

In the following sections, tables are presented in which t_e, t_m and t_I are tabulated after each iteration. Selected problems from those considered in section (5.5) are taken. The iteration number is specified by NITE throughout.

5.6.1 Beams

The cantilevered beam in problem 4 was modelled by eight, two noded beam elements with three degrees of freedom per node. The results obtained are given in Table 5.9.

The cantilevered portal frame (see Fig. 5.3) in problem 13 was modelled by two-noded beam elements with three degrees of freedom per node. A total of twenty elements was required, four for section (b) and eight each for (a) and (c). The results obtained are given in Table 5.10.

5.6.2 Plates

The cantilevered plate in problem 17 was modelled by twelve, eight noded plate elements with three degrees of freedom per node. The results obtained are given in Table 5.11.

NITE	SSP			XXSS			ZSSP			YSSP		
	t_e	t_m	t_I	t_e	t_m	t_I	t_e	t_m	t_I	t_e	t_m	t_I
1	0.281	-	0.445	0.141	0.234	0.537	0.282	0.0	0.447	0.281	0.0	0.447
2	0.096	-	0.258	0.042	0.233	0.438	0.096	0.0	0.257	0.096	0.0	0.257
3	0.065	-	0.227				0.066	0.077	0.303	0.067	0.077	0.305

TABLE 5.9. COMPARISON OF t_e, t_m and t_I FOR PROBLEM 4

NITE	SSP			XXSS			ZSSP			YSSP		
	t_e	t_m	t_I	t_e	t_m	t_I	t_e	t_m	t_I	t_e	t_m	t_I
1	0.269	-	0.654	0.178	0.563	1.121	0.269	0.0	0.659	0.268	0.0	0.659
2	0.160	-	0.538	0.052	0.562	0.993	0.160	0.0	0.538	0.159	0.0	0.538
3	0.109	-	0.487				0.109	0.178	0.665	0.109	0.178	0.665
4	0.071	-	0.449				0.070	0.174	0.623	0.070	0.177	0.626

TABLE 5.10. COMPARISON OF t_e, t_m and t_I FOR PROBLEM 13

NITE	SSP			XXSS			ZSSP			YSSP		
	t_e	t_m	t_I	t_e	t_m	t_I	t_e	t_m	t_I	t_e	t_m	t_I
1	0.272	-	3.139	0.176	5.288	8.328	0.273	0.0	3.151	0.272	0.0	3.151
2	0.150	-	3.013	0.062	5.288	8.214	0.150	0.0	3.013	0.150	0.0	3.012
3	0.102	-	2.963				0.102	0.409	3.373	0.101	0.408	3.372
4	0.080	-	2.943				0.079	0.418	3.360	0.081	0.408	3.352

TABLE 5.11. COMPARISON OF t_e, t_m and t_I FOR PROBLEM 17

In problem 18, the cantilevered plate of problem 17 was again modelled but with sixteen elements. The results obtained are presented in Table 5.12.

5.6.3 Rig

In problems 19 and 20, an off-shore structure (Rig), was analysed. The two problems differ only in the number of elements employed in the model. This is a full three-dimensional structure, see Fig. 5.4, and was modelled using two-noded beam elements with six degrees of freedom per node. All the degrees of freedom at the base of the four legs of the structure were assumed to be fixed. The results obtained from the analyses of problems 19 and 20 are given in Tables 5.13 and 5.14 respectively.

5.6.4 Discussion

The following statements may be made after inspecting Tables 5.9 to 5.14:

(i) The ZSSP and YSSP options still remain an enigma. The reason for persisting with these is that the modification here takes very little time. Therefore any improvement will be significant.

(ii) The XXSS option still shows the most potential. It always converges in the least number of iterations.

(iii) The t_m for the XXSS option becomes the major contributor to t_I as the order of the problem is increased.

Young's Modulus,
$E = 2.09 \times 10^{11}\ NM^{-2}$

Poisson's ratio,
$\nu = 0.3$

Density,
$\rho = 7{,}84 \times 10^{3}\ kg\ m^{-3}$

Legs

Cross-sectional area,
$A = 9.654 \times 10^{-4}\ m^2$

Moment of inertia,
$I = 5.7107 \times 10^{-7}\ m^4$

Platform and Strengthening Members,

Cross-sectional area,
$A = 2.879 \times 10^{-4}\ m^2$

Moment of inertia,
$I = 2.405 \times 10^{-8}\ m^4$

Note

(i) Circular cross-section used throughout

(ii) The bore of each leg is fully constrained.

(iii) The dimensions given are in metres throughout

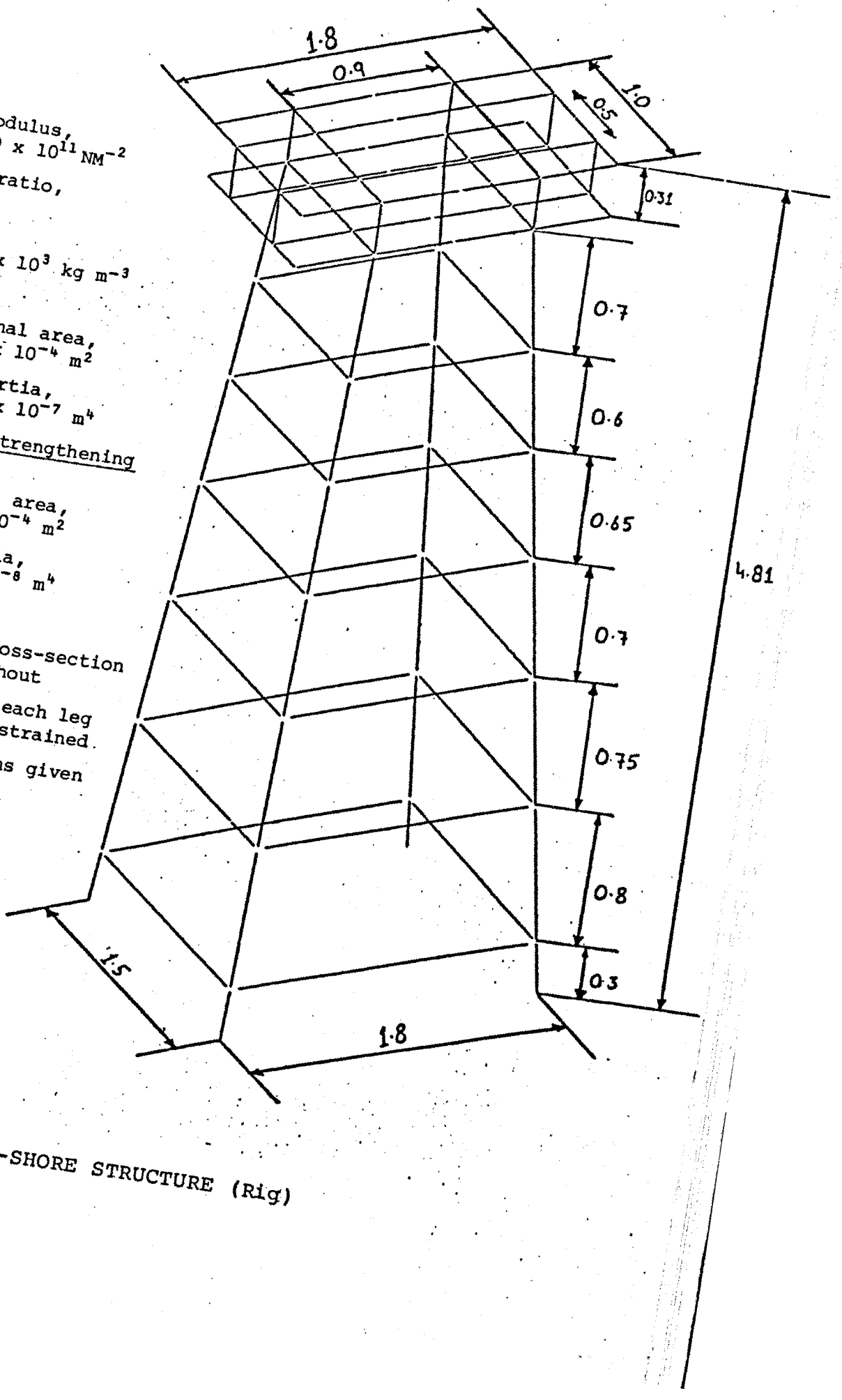


FIG. 5.4. OFF-SHORE STRUCTURE (Rig)

NITE	SSP			XXSS			ZSSP			YSSP		
	t_e	t_m	t_I	t_e	t_m	t_I	t_e	t_m	t_I	t_e	t_m	t_I
1	0.283	-	4.043	0.150	6.961	10.871	0.284	0.0	4.058	0.284	0.0	4.059
2	0.148	-	3.905	0.059	6.962	10.779	0.148	0.0	3.905	0.148	0.0	3.906
3	0.102	-	3.860				0.102	0.520	4.379	0.102	0.518	4.379
4	0.085	-	3.842				0.083	0.531	4.372	0.082	0.518	4.358

TABLE 5.12. COMPARISON OF t_e, t_m and t_I FOR PROBLEM 18

NITE	SSP			XXSS			ZSSP			YSSP		
	t_e	t_m	t_I	t_e	t_m	t_I	t_e	t_m	t_I	t_e	t_m	t_I
1	0.283	-	3.921	0.199	6.658	10.495	0.283	0.0	3.938	0.283	0.0	3.939
2	0.159	-	3.794	0.098	6.658	10.392	0.159	0.0	3.793	0.159	0.0	3.794
3	0.125	-	3.759	0.066	6.658	10.360	0.125	0.576	4.335	0.125	0.573	4.333
4	0.111	-	3.746				0.111	0.588	4.334	0.112	0.574	4.320
5	0.090	-	3.725				0.090	0.588	4.313	0.091	0.574	4.300

TABLE 5.13. COMPARISON OF t_e, t_m and t_I FOR PROBLEM 19

NITE	SSP			XXSS			ZSSP			YSSP		
	t_e	t_m	t_I	t_e	t_m	t_I	t_e	t_m	t_I	t_e	t_m	t_I
1	0.207	-	20.094	0.190	38.360	58.436	0.208	0.0	20.134	0.207	0.0	20.134
2	0.145	-	20.028	0.137	38.360	58.383	0.145	0.0	20.028	0.145	0.0	20.028
3	0.144	-	20.028	0.049	38.361	58.294	0.144	1.330	21.360	0.144	1.329	21.358
4	0.129	-	20.013	0.035	38.361	58.297	0.130	1.301	21.313	0.129	1.329	21.342
5	0.142	-	20.028				0.141	1.361	21.388	0.142	1.329	21.358
6	0.076	-	19.960				0.076	1.361	21.321	0.076	1.329	21.289
7	0.065	-	19.947				0.065	1.362	21.309	0.064	1.329	21.277
8	0.048	-	19.930				0.047	1.363	21.293	0.047	1.329	21.259

TABLE 5.14. COMPARISON OF t_e, t_m and t_I FOR PROBLEM 20

(iv) The SSP option is still the quickest with regard to the total iteration time, t_c.

It is clear from the above statements that the XXSS option appears to be the most promising approach. However, the modification here requires a lot of extra work. The major requirement from a successful modification is that it takes the least time to obtain a solution, comparable in accuracy to that obtained by the original subspace iteration algorithm. Note that, henceforth, the ZSSP and YSSP options will be neglected.

Consider now the XXSS option. It certainly fulfils the accuracy criterion. However, the extra workload in each iteration causes it to be slower than the SSP option. It is, therefore, necessary to:

(a) Try to decrease t_m.

(b) Seek an environment in which the modification thrives.

(c) Employ the modification selectively.

(d) Optimise the modification.

Consider now each of (a) to (d) in turn.

(a) The aim here is to decrease t_m. Most of the work in the XXSS option can be attributed to the $[K]^{-1}[M]$ operation. The following definition is now necessary. Let NIIT[46] be the number of $[K]^{-1}[M]$ operations carried out on the iteration vectors. Recall the parameter, ℓ, defined in section (4.2.2). In fact, $\ell \equiv$ NIIT. It is convenient to use ℓ with regard to the algorithm. However, NIIT is used throughout henceforth.

Clearly, varying NIIT is a possibility. This will affect t_m and the convergence characteristic of the solution.

(b) The size of the subspace is a major factor which governs the solution characteristic in all the options available. Recall that the size of the subspace is dependent upon the number of required eigenvalues, NROOT. Therefore, varying NROOT may provide a favourable change in the environment as far as the XXSS option is concerned.

(c) Consider the t_e from Table 5.9 to 5.14, as the iteration number, NITE increases, t_e generally decreases. Note, however, this decrease is not monotonic. The decrease in t_e may be explained as follows. As NITE increases, the iteration vectors become better and better approximations to the eigenvectors. Consequently, the subspace operators $[\bar{K}]$ and $[\bar{M}]$ become more and more diagonal. Let the t_e for the SSP and XXSS options be t_e^s and t_e^x respectively. By comparing these, it can be seen that for a given value of NITE, t_e^x is always smaller than t_e^s. Thus, after a given NITE, the XXSS option has approximated the eigensystem more accurately than the SSP option. Another observation from Tables 5.9 to 5.14 is that the XXSS option achieves redundant accuracy, that is the lower eigenvalues are approximated far more accurately than required. The main point to note is that initially the convergence rate of the XXSS option is much faster than that of the SSP option.

However, as convergence is approached, the convergence rate decreases and there is no return for the extra work per iteration required by the modification. Clearly, therefore, the conclusion here is that the XXSS option should be employed for the 'first few iterations' and then revert to the XXSS option.

(d) It may be possible to optimise the subroutine containing the XXSS modification. Consider the Gram-Schmidt orthonormalisation process. If this can be neglected, then t_m may be decreased considerably. The requirement, therefore, is to discover how necessary the Gram-Schmidt process is for numerical stability.

The ideas outlined above are investigated in the following sections.

5.7 Varying NIIT

Recall now the parmeter NIIT, defined in section (5.6.4). In this section the effect of varying NIIT on the solution characteristics is considered. All the results presented in this chapter to date have been obtained for NIIT = 2. It is difficult to justify this choice of NIIT logically. However, intuitively the choice appeared reasonable at the outset of the investigation.

The original idea was to increase the convergence rate of the solution. Therefore, to easily observe this increase a high enough value for NIIT was required. Recall now the discussion in section (4.2.1). It was

observed that too large a value for NIIT would lead to convergence to the least dominant vector. Thus, all the modes except the lowest would be suppressed. Therefore, it was decided that NIIT = 2 should be employed, since NIIT = 1 may not lead to a significant change in the convergence rate.

To date, NIIT = 2 has been employed successfully. However, since NIIT has a significant effect on the iteration time, t_I, it is now necessary to investigate other possible values for NIIT.

The results obtained when NIIT takes the values 1 and 2 in the most promising modification, XXSS, are presented in Table 5.15. Note that NROOT = 5 throughout except in problems 1, 2 and 5, where it is 1, 2 and 2, respectively.

Henceforth, let XXSS1 and XXSS2 denote the XXSS option when NIIT takes the value 1 and 2, respectively. It can be seen from Table 5.15 that when the number of iterations for the solution to converge, $NITE_c$ are equal, XXSS1 is faster than XXSS2. In fact, in problem 20, XXSS1 is faster even when it requires more iterations to converge than XXSS2. However, it follows from the discussion in section (4.2.1), that increasing NIIT, 'focuses' the iteration vectors more and more to the least dominant vector. Thus, the higher the value of NIIT, the higher the convergence rate. Note also that both the XXSS1 and XXSS2 options are slower than the SSP option. Therefore, it appears that the successful

PROBLEM NUMBER	PROBLEM SIZE	XXSS2		XXSS1		PROBLEM TYPE
		t_c	$NITE_c$	t_c	$NITE_c$	
1	3	0.021	2	0.017	2	Beam
2	6	0.083	2	0.069	2	"
3	12	0.561	2	0.480	2	"
4	24	0.975	2	0.834	2	"
5	6	0.082	2	0.068	2	Frame
6	12	0.541	2	0.499	2	"
7	15	0.640	2	0.595	2	"
8	24	0.955	2	0.843	2	"
9	27	1.052	2	0.915	2	"
10	33	1.320	2	1.611	3	"
11	48	1.786	2	2.174	3	"
12	51	1.906	2	2.305	3	"
13	57	2.114	2	2.565	3	"
14	15	0.432	2	0.336	2	Plate
15	60	6.219	2	6.687	3	"
16	96	11.362	2	12.020	3	"
17	132	16.542	2	17.322	3	"
18	168	21.650	2	22.605	3	"
19	186	31.247	3	21.955	3	Rig
20	456	233.392	4	198.297	5	"

TABLE 5.15. COMPARISON OF THE XXSS1 and XXSS2 OPTIONS

modification will be one that incorporates the SSP option. In this case, as much benefit as possible must be derived from the 'first few iterations' while either the XXSS1 or XXSS2 options are being employed, before reverting to the SSP option. With regard to such a hybrid technique, it is clearly desirable to obtain a high convergence rate initially. Thus, in this respect, it appears that XXSS2 is preferable to XXSS1.

Consider now the case when NIIT > 2, although theoretically the convergence rate increases as NIIT is increased, in practice there are several detrimental features involved with employing NIIT > 2. The problems solved when NIIT takes the values 1 and 2 were presented to the XXSS option with NIIT = 3, XXSS3 say, in all cases numerical instabilities were encountered. These were in the main due to overflow and may be countered without difficulty by employing a scaling factor. However, the main point of concern was the magnitude of the time used for the modification, t_m. It was observed in section (5.6.4) that for NIIT = 2, as the order of the problem was increased, the contribution of t_m to t_I became progressively more significant. Clearly, this situation will be enhanced for NIIT > 2. Therefore, neglecting NIIT > 2 and pursuing NIIT = 2 was considered a reasonable course of action, thus henceforth, NIIT = 2 throughout.

5.8 Varying NROOT

The number of eigenvalues required in an analysis, NROOT is clearly an important parameter since it defines

the size of the iterating subspace, see section (3.5.1). In this section the behaviour of the solution characteristic is considered when NROOT is varied.

The results presented in this chapter to date, have been obtained for NROOT = 5. Numerical difficulties were encountered when the size of the iterating space, q, became a significant percentage of the size of the full space, N. Definite values defining a useful range for q/N ratio are not available since these will certainly be problem and size dependent. However, experience has shown that for the problem considered, the useful values of q/N could be as high as $\sim$ 2/3 for the smaller problems, but this tended to decrease as the order of the problem was increased. Therefore, only problems of the highest order will be considered. Thus, each problem may be solved for several values of NROOT. The values taken by NROOT are 8 and 12. The results obtained from the SSP and XXSS options are presented in Tables 5.16 and 5.17.

Notice that there is no result for problem 4 in Table 4.17, the q/N ratio here is too large and, due to numerical instabilities, the solution was terminated. However, the explanation that 'the q/N ratio is too large' is not entirely satisfactory and so the manifestation of these numerical instabilities must be investigated further. Consider, therefore, the basic concept upon which the modification is based, recall equation (4.14) from section (4.2.1):

PROBLEM NUMBER	N	SSP		XXSS		PROBLEM TYPE
		t_c	$NITE_c$	t_c	$NITE_c$	
4	24	3.278	4	2.326	2	Beam
13	57	5.399	4	4.697	2	Frame
16	96	16.019	4	19.578	2	Plate
17	132	27.361	5	42.021	3	"
18	168	34.887	5	54.743	3	"
19	186	34.523	5	53.415	3	Rig
20	456	334.100	10	382.821	4	"

TABLE 5.16. COMPARISON BETWEEN THE SSP AND XXSS OPTIONS WITH NROOT = 8

PROBLEM NUMBER	N	SSP		XXSS		PROBLEM TYPE
		t_c	$NITE_c$	t_c	$NITE_c$	
4	24	5.662	4	-	-	Beam
13	57	10.513	5	9.711	3	Frame
16	96	27.372	5	38.297	3	Plate
17	132	44.867	6	55.330	3	"
18	168	56.484	6	71.260	3	"
19	186	73.098	8	92.817	4	Rig
20	456	470.669	11	607.826	5	"

TABLE 5.17. COMPARISON BETWEEN THE SSP AND XXSS OPTIONS WITH NROOT = 12

$$\underline{z}_i = f_b \sum_{j=1}^{n} a_{ij} \left(\frac{\lambda_1}{\lambda_j}\right) \underline{v}_j \qquad .. \quad .. \quad .. \quad (5.7)$$

Clearly, the effect of the modification is to suppress the higher nodes and focus the iteration vectors, $\underline{z}_i$, towards the lowest node. Recall now the parameter $\bar{\ell}$ defined in section (4.2.1). This will henceforth be referred to as the eigenvalue ratio:

$$\bar{\ell}_j = \frac{\lambda_j}{\lambda_1} \qquad .. \quad .. \quad .. \quad .. \quad (5.8)$$

The 'richness' of the jth eigenvector in an iteration vector is obviously governed by the inverse of $\bar{\ell}_j$. Therefore, if $\bar{\ell}_j$ is 'very large' then the jth vector will effectively be missing from the iteration vectors. Suppose now that the least dominant subspace of interest has dimension q and the spectral range of the eigenvalues within this subspace is such that $\lambda_1 << \lambda_q$, then $\bar{\ell}_q$ will be large. Consequently, a few of the highest nodes within the subspace of interest may be completely lost. It is due mainly to this that numerical instabilities occur in the modification.

Therefore, the statement that 'the numerical instabilities occur because the q/N ratio becomes a significant percentage of unity' is still valid. However, now a deeper meaning is attached to it, namely that the increase in q/N leads to an increase in $\bar{\ell}$ which effects the 'richness' of eigenvectors in iteration vectors.

It is evident from Tables 5.16 and 5.17 that, although the XXSS option is converging in fewer iterations, $NITE_c$, the total time taken, t_c is still greater than that

for the SSP option. This becomes pointedly obvious as the order of the problem is increased. Therefore, since the object of this work is to obtain a practical algorithm, it appears that other avenues must be pursued.

5.9 Hybrid technique

In this section, a method consisting of the XXSS and SSP options is considered. Appropriately, it is referred to as the 'hybrid technique'. Prior to any numerical investigation, an important decision must be made, namely how and when should the two options be employed. Clearly, since the XXSS option has the highest initial convergence rate, it should be employed during the 'first few iterations'. As the solution approaches convergence, the convergence rate of the XXSS option decreases and no advantage is gained from the extra work due to the modification. Hence, a prudent course of action would be to employ the SSP option once the initial advantage has been gained from the XXSS option during the 'first few iterations'.

Although the approach outlined above is the logical way to proceed, a further question arises, namely what is meant by 'first few iterations'? An 'a priori' answer for this question is very difficult if not impossible to find. However, in the present case, an answer may be deduced. Consider the following discussion.

As the number of iterations, NITE increases, the eigenpairs are better approximated. Consequently, the

subspace operators $[\bar{K}]$ and $[\bar{M}]$ become progressively more diagonal and hence t_e decreases. Note, however, that this decrease in t_e is not always monotonic and fluctuations have been observed. It can be seen from Tables 5.9 to 5.14 that the approximate value of t_e at which convergence occurs is arrived at in considerably fewer iterations by the XXSS option in comparison to the SSP option. Let NMOD be the number of iterations employing the XXSS option in which t_e decreases quickly and let NSSP be the number of further iterations, employing the SSP option, required for convergence. Now, NMOD represents the 'first few iterations' and the problem of giving it a quantitative value still remains. However, by considering Tables 5.9 to 5.14, it is possible to choose a value for NMOD such that the time taken for an iteration, t_I for NMOD and NSSP iterations is less than the t_I when only the XXSS or the SSP options are employed. An inspection of Tables 5.9 to 5.14 shows that as the order of the problem, N increases, the time taken for the modification increases, therefore NMOD must be small enough to make use of the high convergence rate without becoming a burden on the final t_I. Another factor which plays an important part in the choice of NMOD is the total number of iterations required for convergence, $NITE_c^s$ when the SSP option is used. This is, in fact, the quantifying aspect of the preceding line of argument. Experience has shown that the contribution of t_m to t_I, can be as much as $\sim$ 70%, hence provided

$$\frac{NMOD}{NITE_c^s} \leq \frac{1}{3} \qquad .. \quad .. \quad .. \quad (5.6)$$

indications are that the t_c for the hybrid technique will be less than that for the SSP option.

It can be seen from Tables 5.9 to 5.14 that, for the problems considered, NMOD can take the vales 1 and 2 if equation (5.6) is to be satisfied. Note that equation (5.6) is only a crude indication by which a value for NMOD is obtained. Even this only applies to the type and size of problems considered here and is certainly not stated as a general rule.

The results obtained for NMOD = 1 and NMOD = 2, are given in Tables 5.18 to 5.20, three values of NROOT are used, namely 5, 8 and 12. For ease of comparison, the results due to the SSP option are reproduced from Tables 5.7, 5.16 and 5.17 for NROOT = 5, 8 and 12, respectively. Let XXSSH1 and XXSSH2 denote the hybrid technique with NMOD = 1 and 2, respectively.

The following statements may be made after inspecting Tables (5.18) and (5.19):

(i) The XXSSH1 option is always quicker than the SSP option.

(ii) The XXSSH2 option is always quickest for problem 20.

It appears that NMOD = 1 is a conservative choice and the XXSSH1 option will generally always converge faster then the SSP option, provided the SSP option does not converge in one or two iterations. It is observed that the XXSSH2 option is always slower than the XXSSH1 and SSP options with the exception of problem 20. The reason for this is

Problem No.	Order of Problem	SSP		XXSSH2		XXSSH1		Problem Type
		Time for Convergence	No. of Iterations for Convergence	Time for Convergence	No. of Iterations for Convergence	Time for Convergence	No. of Iterations for Convergence	
4	24	0.928	3	0.974	2	0.754	2	Beam
13	57	2.128	4	2.114	2	1.571	2	Frame
16	96	8.466	4	11.362	2	7.77	2	Plate
17	132	12.058	4	16.541	2	11.272	2	Plate
18	168	15.650	4	21.649	2	14.714	2	Plate
19	186	18.948	5	24.588	3	17.97	3	Rig
20	456	160.028	8	156.713	4	158.324	6	Rig

TABLE 5.18. COMPARISON OF SSP, XXSSH2 AND XXSSH1 OPTIONS FOR NROOT = 5

Problem No.	Order of Problem	SSP		XXSSH2		XXSSH1		Problem Type
		Time for Convergence	No. of Iterations for Convergence	Time for Convergence	No. of Iterations for Convergence	Time for Convergence	No. of Iterations for Convergence	
4	24	3.278	4	2.326	2	1.949	2	Beam
13	57	5.399	4	4.696	2	3.741	2	Frame
16	96	16.019	4	19.577	2	13.666	2	Plate
17	132	27.361	5	33.344	3	24.704	3	Plate
18	168	34.887	5	43.298	3	31.878	3	Plate
19	186	34.523	5	42.420	3	31.510	3	Rig
20	456	334.100	10	324.65	6	329.341	8	Rig

TABLE 5.19. COMPARISON OF THE SSP, XXSSH2 and XXSSH1 OPTIONS FOR NROOT = 8

Problem No.	Order of Problem	SSP		XXSSH2		XXSSH1		Problem Type
		Time for Convergence	No. of Iterations for Convergence	Time for Convergence	No. of Iterations for Convergence	Time for Convergence	No. of Iterations	
4	24	5.663	4	-	-	-	-	Beam
13	57	10.513	5	8.474	3	7.349	3	Frame
16	96	27.372	5	30.755	3	23.260	3	Plate
17	132	44.867	6	44.209	3	40.006	4	Plate
18	168	56.484	6	56.727	3	51.163	4	Plate
19	186	73.098	8	64.747	4	68.140	6	Rig
20	456	470.669	11	499.523	8	506.662	10	Rig

TABLE 5.20. COMPARISON OF THE SSP, XXSSH2 AND XXSSH1 OPTION FOR NROOT = 12

that the XXSSH1 and SSP options are converging in very few iterations and consequently giving the XXSSH2 option no chance to shine. This is verified by the fact that in problem 20 where the XXSSH1 and SSP options require more than a few iterations to converge, the XXSSH2 option emerges as the quickest.

Consider now Table 5.20, the following statements may be made:

(i) The XXSSH1 option is always quicker than the SSP option with the exception of problem 20.

(ii) The XXSSH2 option is comparable with the SSP option in problems 17 and 18.

(iii) The XXSSH2 option is the quickest in problem 19.

(iv) The SSP option is the quickest in problem 20.

With regard to the fact that NMOD = 1 was considered a safe choice, statement (i) was slightly disappointing, but perhaps to be expected. Statements (ii) and (iii) gave indications that the XXSSH2 option was about to come into its own, however statement (iv) was bitterly disappointing and unexpected.

It is now required to reconsider the application of the modification in the hybrid technique. The modification is applied at the level of the iterating subspace for the first NMOD iterations. Initially, this was thought to be a reasonable course of action. However, now it must be examined closely for any flaws in the reasoning.

Operating upon the initial iteration vectors is perhaps dubious since, although the $[K]^{-1}[M]$ operation enhances the lower modes and suppresses the higher modes, the vectors being operated upon are not eigenvector approximations. Therefore, it is possible to reach a situation in which the iteration vectors span the least dominant subspace but are not eigenvectors. This implies that, although the subspace has converged to the least dominant subspace, more iterations are required to convert the orthogonal basis of iteration vectors into a basis of eigenvectors.

Consider the discussion in section (5.6.4), particularly the monitoring of the time taken for solving the eigensystem of subspace operators, t_e. Recall that it was this which gave rise to the idea of the hybrid technique. Once again, the monitoring of t_e proves itself useful, the values of t_e in each iteration are presented in Table 5.21. Only problem 20 is considered since it requires more than a few iterations to converge. Note that NROOT takes the values 5,8 and 12 for both the XXSSH1 and XXSSH2 options.

Inspection of Table 5.21 reveals that the XXSSH1 option appears to suffer more than the XXSSH2 option from these fluctuations. This is clearly due to the fact that the enhancing and suppressing of the lower and higher modes, respectively, is not as severe in the XXSSH1 option as in the XXSSH2 option.

NITE	XXSSH1			XXSSH1		
	NROOT			NROOT		
	5	8	12	5	8	12
1	0.189	0.838	2.242	0.189	0.838	2.242
2	0.129	0.713	1.220	0.137	0.871	1.591
3	0.146	0.762	1.070	0.060	0.352	0.610
4	0.075	0.690	1.233	0.050	0.410	1.062
5	0.064	0.343	0.599		0.369	0.451
6	0.050	0.409	1.069		0.190	0.418
7		0.374	0.451			0.330
8		0.191	0.418			0.317
9			0.338			
10			0.311			

TABLE 5.21. VALUES OF t_e AFTER EACH ITERATION FOR THE XXSSH1 and XXSSH2 OPTIONS

It is now necessary to investigate phenomena which give rise to the fluctuations in the value of t_e. Thus, consider again the concept which forms the basis of the modification, namely equation (5.7), and the eigenvalue ratio, $\bar{\ell}_j$ defined in equation (5.8). The following definition is now necessary for convenience:

$$\ell_j' = \frac{1}{\bar{\ell}_j} \qquad .. \quad .. \quad .. \quad .. \quad (5.9)$$

Let the least dominant subspace of interest have dimension q, then if for some $j < q$, ℓ_j' becomes small, the components of the jth mode onwards will be correspondingly small in the iteration vectors. Suppose now that the r lowest eigenpairs are required and $j < r < q$, then components of the required eigenvectors are almost lost from the iteration vectors. However, due to finite arithmetic, these are not entirely lost but remain small and are preserved due to the orthogonalising process. A few iterations are required for such modes to emerge from dormancy, and it is this which causes the fluctuations in the values of t_e. This is explained by the fact that initially the subspace is close to convergence, then a 'missing' mode is pulled into the subspace necessitating the reorientation of the basis vectors. Thus, in the next iteration, the diagonality of $[\bar{K}]$ and $[\bar{M}]$ is degraded and consequently an increase in the t_e is observed. The requirement now is to suppress as much as possible the fluctuations in the sequence formed by the values of t_e.

Consider now the following idea: suppose the first iteration in the solution always employed the SSP option and the modification was applied from the second iteration onwards. This would have the effect of applying the modification at the level of the eigenvector approximations. Initially, this procedure was not employed due to the fact that it appeared to be a 'waste' of the first iteration. However, in the present circumstances, it is clearly justified.

Let SXXH1 and SXXH2 denote the hybrid techniques in which the first iteration employs the SSP option followed by NMOD = 1 and NMOD = 2, respectively. Tables 5.18 to 5.20 are reproduced for the SXXH1 and SXXH2 options in Tables 5.22 to 5.24. The expectation here is that the SXXH1 and SXXH2 options will be slower than the XXSSH1 and XXSSH2 options when only a few iterations are required. However, problem 20 will be the important test.

Observe that, as predicted, the SXXH1 and SXXH2 options are slower than the XXSSH1 and XXSSH2 options, respectively. Note, however, that for problem 20, both the SXXH1 and SXXH2 options are quicker than the SSP, XXSSH1 and XXSSH2 options. This is certainly encouraging as far as analysing 'real problems' is concerned.

The following Table 5.25 has been reproduced from Table 5.21 for the SXXH1 and SXXH2 options in order to illustrate the fact that the fluctuations in the values of t_e have been suppressed in comparison with the XXSSH1 and XXSSH2 options.

Problem No.	Problem Size	SSP		SXXH2		SXXH1		Problem Type
		Time for Convergence	No. of Iterations for Convergence	Time for Convergence	No. of Iterations for Convergence	Time for Convergence	No. of Iterations for Convergence	
4	24	0.928	3	1.345	3	1.118	3	Beam
13	57	2.128	4	2.698	3	2.145	3	Frame
16	96	8.466	4	13.562	3	9.962	3	Plate
17	132	12.058	4	19.601	3	14.328	3	Plate
18	168	15.650	4	25.625	3	18.697	3	Plate
19	186	18.948	5	24.715	3	18.078	3	Rig
20	456	160.028	8	156.818	4	158.393	6	Rig

TABLE 5.22 COMPARISON OF THE SSP, SXXH1 AND SXXH2 OPTIONS FOR NROOT = 5

Problem No.	Problem Size	SSP		SXXH2		SXXH1		Problem Type
		Time for Convergence	No. of Iterations for Convergence	Time for Convergence	No. of Iterations for Convergence	Time for Convergence	No. of Iterations for Convergence	
4	24	3.278	4	3.318	3	2.908	3	Beam
13	57	5.399	4	6.087	3	5.110	3	Frame
16	96	16.019	4	23.991	3	18.053	3	Plate
17	132	27.361	5	34.087	3	25.393	3	Plate
18	168	34.887	5	44.150	3	32.691	3	Plate
19	186	34.523	5	42.946	3	31.915	3	Rig
20	456	334.100	10	324.601	6	329.381	8	Rig

TABLE 5.23. COMPARISON OF THE SSP, SXXH2 AND SXXH2 OPTIONS FOR NROOT = 8

Problem No.	Problem Size	SSP		SXXH2		SXXH1		Problem Type
		Time for Convergence	No. of Iterations for Convergence	Time for Convergence	No. of Iterations for Convergence	Time for Convergence	No. of Iterations for Convergence	
4	24	5.663	4	-	-	-	-	Beam
13	57	10.513	5	9.502	3	8.243	3	Frame
16	96	27.372	5	32.465	3	24.882	3	Plate
17	132	44.867	6	52.397	4	41.462	4	Plate
18	168	56.484	6	67.280	4	52.805	4	Plate
19	186	73.098	8	65.752	4	68.958	6	Rig
20	456	470.669	11	457.211	7	464.068	9	Rig

TABLE 5.24. COMPARISON OF THE SSP, SXXH1 AND SXXH2 OPTIONS FOR NROOT = 12

NITE	SXXH1			SXXH2		
	NROOT			NROOT		
	5	8	12	5	8	12
1	0.208	0.937	1.689	0.207	0.937	1.689
2	0.173	0.702	1.877	0.173	0.701	1.877
3	0.143	0.770	1.062	0.133	0.549	1.344
4	0.076	0.651	1.247	0.041	0.295	0.541
5	0.066	0.512	0.606		0.274	0.504
6	0.046	0.287	0.610		0.213	0.347
7		0.275	0.493			0.313
8		0.208	0.366			
9			0.302			

TABLE 5.25. VALUES OF t_e AFTER EACH ITERATION FOR THE SXXH1 AND SXXH2 OPTIONS

Although Table 5.25 shows that the fluctuations in the values of t_e have been suppressed to a certain extent, a comparison of Tables 5.18 to 5.20 with Tables 5.22 to 5.24 shows that this has only been of use in problem 20 with NROOT = 12.

Consider now a very interesting investigation; only problem 20 with NROOT = 12 is considered in the following discussion. Let XXSSH3 and SXXH3 denote the appropriate hybrid techniques with NMOD = 3. The idea was to observe the behaviour of the solution characteristic of the XXSSH3 and SXXH3 options and the results obtained were perhaps not unexpected in retrospect. For ease of comparison, the results for the SSP, XXSSH1, XXSSH2, SXXH1 and SXXH2 options are also reproduced in Table 5.26.

Although the hybrid techniques presented in this section have been successful, there remains a problem which requires attention, namely the choice of NMOD. The method used earlier in this section for deducing NMOD was applicable only because information about the solution characteristic was available. In order to utilise the technique to its full potential, an optimum value of NMOD is required and, at present, there appears to be no hard and fast rule for obtaining it. Note that even the crude indication given by equation (5.6) is not applicable without prior knowledge of the solution characteristic. Therefore, as the situation stands, analyst experience would appear to be the crucial factor.

OPTION	SSP	XXSSH1	XXSSH2	XXSSH3	SXXH1	SXXH2	SXXH3
t_c	470.669	506.662	499.523	493.005	464.068	457.211	451.492
$NITE_c$	11	10	8	6	9	7	5

TABLE 5.26. COMPARISON OF THE SSP OPTION AND THE HYBRID TECHNIQUES FOR NMOD = 1, 2 AND 3

5.10 Neglecting Gram-Schmidt Ortho-normalisation

The idea here is to investigate the possibility of decreasing the time required for the modification, t_m, by neglecting the Gram-Schmidt ortho-normalisation. All the problems considered in section (5.9) were presented to the XXSSG option, where XXSSG denotes the XXSS option without the Gram-Schmidt ortho-normalisation. In all cases, the algorithm failed due to ill-conditioning of the subspace operators $[\bar{K}]$ and $[\bar{M}]$, which led to numerical instability.

The ill-conditioning of $[\bar{K}]$ and $[\bar{M}]$ arises from the basic concept of the modification. Recall once again the governing equation of the modification, namely equation (5.7) and the eigenvalue ratio, ℓ', defined in equation (5.9). Now since

$$\lambda_1 \leq \lambda_2 \leq \ldots\ldots \leq \lambda_j \leq \ldots\ldots \leq \lambda_q$$

where q is the dimension of the iterating subspace, ℓ_j' forms a decreasing sequence as j is increased from 1 to q. Consequently, the suppression of the higher modes increases progressively as j increases; this leads to a deficiency of the higher modes in the iteration vectors. The effect of this is to progressively make the iteration vectors more and more parallel to the least dominant mode. It is due to this that ill-conditioning in $[\bar{K}]$ and $[\bar{M}]$ is encountered.

Suppose that the lowest r eigenpairs are required, with $r < q$ and that ℓ_j' is small for some $j < r$, then since the higher modes are suppressed according to ℓ_j', it

follows that the decision for retaining or discarding the Gram-Schmidt ortho-normalisation can only be made if prior information about the spread of the eigenvalues in the subspace of interest is available.

The modification may, therefore, be interpreted in the following manner: The $[K]^{-1}[M]$ operation filters out the higher modes indiscriminantly. This is followed by the Gram-Schmidt ortho-normalisation which reinstates some components of the higher modes, thus preserving numerical stability.

It is clear from the above discussion that Gram-Schmidt ortho-normalisation or some other device for preserving numerical stability is a necessary requirement if the modified algorithm is to be functional.

CHAPTER 6

CONCLUDING REMARKS

6.1 Conclusions

The aim of this study was to obtain a practical algorithm for the solution of the large generalised eigenproblem by modifying the subspace iteration algorithm[5]. In the context of the stated aim, the following conclusions may be drawn:

(i) Several ideas for modifying the subspace iteration algorithm were investigated theoretically.

(ii) The original subspace iteration algorithm was validated by comparison with know solutions and subsequently used as the fiducial reference.

(iii) The ideas investigated theoretically were converted into computationally efficient algorithms and programmed in subroutine form.

(iv) After prolonged comparisons the XXSS option emerged as the clear rival to the original subspace iteration algorithm.

(v) The XXSS option may be basically defined as a process which suppresses and enhances the higher and lower modes respectively in the iteration vectors.

(vi) The initial convergence rate of the XXSS option was considerably superior to that of the original subspace iteration algorithm. Consequently, it always converged in fewer iterations.

(vii) In terms of the time taken for convergence to occur, the original algorithm was always quicker than the XXSS option.

(viii) Utilisation of the superior initial convergence rate has led to the conception of the XXSSH and SXXH options.

(ix) Both the XXSSH and SXXH have higher initial convergence rates and converge in fewer iterations in comparison to the original subspace iteration algorithm.

(x) The time taken for convergence to occur by the modification is always less than that for the original subspace iteration algorithm provided the correct hybrid technique is employed.

(xi) The ratio of the lowest to highest eigenvalues in the subspace of interest emerges as the most important parameter with regard to explaining numerical instabilities.

(xii) In order to utilise the full potential of the hybrid technique, an optimum value for NMOD must be chosen.

(xiii) The experienced analyst will be able to employ the hybrid technique far more efficiently than the layman.

6.2 Ideas for further research

During the course of this study and particularly towards the end of it, various ideas have emerged which were either peripheral to the main theme of the study or were such that investigation could not be initiated due to lack of time. The author feels, however, that these should be pointed out in order to stimulate further study in this field. Some of the potentially promising ideas and interesting speculations are briefly outlined in the following:

(a) The idea of shifting has been proposed in the past[43,48] and implemented recently[49] with the original subspace iterations. However, consider the following; in the course of the study, the hybrid techniques described in section (5.9) showed a convergence rate far superior to that of the original subspace iteration. Utilising this superior convergence rate in conjunction with shifting would appear to be a logical step towards improving upon the hybrid technique.

Suppose the r lowest eigenpairs are required, then a check could be made after each iteration to obtain the number of eigenvalues which have satisfied the convergence criterion. Experience has shown that the lowest few eigenvalues converge fastest and in clusters,

a shift $\bar{s}_r$, beyond the last converged eigenvalue could then be employed. Thus, 'clusters' of eigenvalues may be obtained in between shifting, the advantage being the utilisation of the initial high convergence rate of the hybrid technique.

Notice, however, that there is also a disadvantage associated with this, namely, the fact that [K] must be factorised after each shift. Let q, ℓ and m denote the size of the iterating subspace, the number of $[K]^{-1}[M]$ operations per iterations and the half-bandwidth of [K] respectively. Then the number of operations required for a triangular factorisation of [K] are:

$$\frac{1}{2}\,nm^2 + \frac{3}{2}\,nm$$

and those for an iteration in which the modification is being employed are:

$$nq(4m + 2q + 3) + \ell(4nmq + 8nq - 3)$$

From these operation counts, it follows that if q is comparable to m, then the advantage is clear. It is the opinion of the author that the possibility of obtaining an efficient algorithm using the hybrid technique in conjunction with shifting is high. However, only detailed study can lend credence, or otherwise, to this idea.

(b) The following idea is again based upon shifting, but in the opposite direction. Recall that a small ℓ' implies a vicious suppression of the higher modes, some of which may be required. Suppose now that a shift, $\bar{s}_\ell$

to the left of the origin is employed, this would effectively increase ℓ' since

$$\frac{\lambda_1 + \bar{s}_\ell}{\lambda_q + \bar{s}_\ell} > \frac{\lambda_1}{\lambda_q}$$

In theory $\bar{s}_\ell$ may be made as large as required. However, the drawback here is that information about the eigenvalue spectrum is required prior to the analysis. The possibility exists, of course, of obtaining a rough approximation of the eigenvalue after a few iterations and then employing this idea.

Note that a major implication of this would be the possibility of neglecting Gram-Schmidt ortho-normalisation. Since, in this case, the severity of the suppression will be countered and, consequently, the components of the required higher modes need not be reinstated in the iteration vectors.

(c) Recall section (3.5.1). In the opinion of the author, this aspect of the analysis requires further study. It should be possible with systematic research to form a better methodology for choosing the dimension of the subspace than that given in section (3.5.1).

(d) An interesting idea is to generate the initial interation vectors for the hybrid technique using Lanczos method: This idea has been proposed and implemented with some success[18,43,49] for the original subspace iteration.

(e) It would be very interesting and instructive to carry out a systematic parameter survey involving N, q, NIIT, NROOT and NMOD. This could possibly lead to definite guidelines for choosing optimum values for the various parameters in a given problem.

APPENDIX A

CARDANO'S RULE FOR SOLVING CUBIC EQUATIONS

A real cubic equation, after dividing through by the coefficient of the leading term, may be written as:

$$x^3 + ax^2 + bx + c = 0 \quad .. \quad .. \quad .. \quad (A1)$$

where a, b and c are real constants.

By employing the transformation

$$x = y - \frac{a}{3} \quad .. \quad .. \quad .. \quad .. \quad (A2)$$

equation (A1) may be re-written as:

$$y^3 = Ay + B \quad .. \quad .. \quad .. \quad (A3)$$

where

$$\left.\begin{aligned} A &= 3\left(\frac{a}{3}\right)^2 - b \\ \text{and} \\ B &= -2\left(\frac{a}{3}\right)^3 + b\left(\frac{a}{3}\right) - c \end{aligned}\right\} \quad .. \quad .. \quad (A4)$$

now let

$$p = \frac{A}{3} \quad \text{and} \quad q = \frac{B}{2} \quad .. \quad .. \quad .. \quad .. \quad (A5)$$

Then the discriminant, D, is:

$$D = q^2 - p^3$$

Let y_i, i = 1,3 be the roots of equation (A3) in the following discussion. Three cases can arise according to whether D is positive, zero or negative.

(a) $D > 0$

In this case one root is read, y_1 say, where

$$y_1 = \{q + (q^2 - p^3)^{\frac{1}{2}}\}^{\frac{1}{3}} - \{q - (q^2 - p^3)^{\frac{1}{2}}\}^{\frac{1}{3}}$$

y_2 is imaginary and is given by

$$y_2 = 2p^{\frac{1}{2}} \cosh \{\tfrac{1}{3} \cosh^{-1}(q/p^{\frac{1}{2}})\}$$

and y_3 is the complex conjugate of y_2.

(b) $D = 0$

In this case there are two real distinct roots, one of these is repeated

$$y_1 = 2q^{\frac{1}{3}}$$

$$y_2 = y_3 = -q^{\frac{1}{3}}$$

(c) $D < 0$

In this case there are three real, distinct roots. Initially determine an angle, θ, from

$$\theta = \cos^{-1}(q/p^{\frac{3}{2}})$$

with $0^o < \theta < 180^o$

then

$$y_1 = 2p^{\frac{1}{2}} \cos (\tfrac{\theta}{3})$$

$$y_2 = 2p^{\frac{1}{2}} \cos (\tfrac{\theta}{3} + 120^o)$$

$$y_3 = 2p^{\frac{1}{2}} \cos (\tfrac{\theta}{3} + 240^o)$$

The roots of equation (A3) are then transformed into the roots of equation (A1) by employing equation (A2).

REFERENCES

1. CLOUGH, R. W., Analysis of structural vibrations and dynamic response. In R. H. Gallagher, Y. Yamada and J. T. Oden (Eds.), 'Recent advances in matrix methods of structural analysis and design', Proc. US-Japan Symp., University of Alabama Press, Huntsville, Alabama, 1971.

2. HUTCHINGS, D.,'FINEL User's guide', Aeronautics Department, Imperial College of Science and Technology, London, 1982.

3. WILSON, E. L., 'Earthquake analysis of reactor structures', A.S.M.E. 1st National Congress on Pressure vessels and piping components, San Francisco, May 1971.

4. BATHE, K. J., Ph.D. thesis, 'Solution methods for large generalised eigenvalue problems in structural engineering', Structural Eng. Lab., University of California, Berkeley, California, 1971.

5. BATHE, K. J. and WILSON, E. L., 'Numerical methods in finite element analysis', Prentice-Hall Inc., Englewood Cliffs, N.J., 1976.

6. TIMOSHENKO, S., YOUNG, D. H. and WEAVER (Jr.), W., 'Vibration problems in Engineering', John Wiley and Sons, Inc., 1974 (4th Edition).

7. WILKINSON, J. H., 'The algebraic eigenvalue problem', Clarendon Press, Oxford, 1965.

8. NOBLE, B., 'Applied linear algebra', Prentice-Hall, Inc., Englewood Cliffs, N.J., 1969.

9. LANCZOS, C., 'Applied analysis', Prentice-Hall, Inc., Englewood Cliffs, N.J., 1956.

10. HOUSEHOLDER, A. S., 'Principles of numerical analysis', Addison-Wesley, Inc., Reading, Mass., 1969.

11. JACOBI, C. G. J., 'Über ein leichtes verfahren die en der theorie der Säculärstörungen vorkommenden Gleichungen numerich aufzulösen', Crelle's Journal, Vol. 30, 1846, pp 51-94.

12. GOLDSTINE, H. H., MURRAY, F. J. and VON NEWMAN, J. 'The Jacobi method for real symmetric matrices', J. of the Assoc. of Computer Machines, Vol. 6, 1959, pp 59-96.

13. WILKINSON, J. H., 'Note on the quadratic convergence of the cyclic Jacobi method', Numer. Math., Vol. 4, 1962, pp 296-300.

14. LANCZOS, C., 'An iteration for the solution of the eigenvalue problem of linear differential and integral operators', J. Res. Nat. Bur. Standards, Vol. 45, 1950, pp 255-282.

15. CAUSEY, R. L. and GREGORY, R. T., 'On Lanczos' algorithm for tridiagonalising matrices', S.I.A.M. Review, Vol. 3, No. 4, Oct. 1961, pp 322-328.

16. PAIGE, C. C., 'Computational variants of the Lanczos method for the eigenproblem', IMA J. Numer. Anal., Vol. 10, No. 3, 1972, pp 373-381.

17. PAIGE, C. C., 'Accuracy and effectiveness of the Lanczos algorithm for the symmetric eigenproblem', Linear Algebra and Applications, Vol. 34, 1980, pp 235-258.

18. RAMSWAMY, S., 'On the effectiveness of the Lanczos method on the solution of large eigenvalue problems', J. of Sound and Vib., Vol. 73, 1980, pp 405-418.

19. FRIED, I., 'Optimal gradient minimisation scheme for finite element problems', J. of Sound and Vib., Vol. 20, No. 3, 1972, pp 333-342.

20. MARTIN, R. S. REINSCH, C. and WILKINSON, J. H., 'Householder's tridiagonalisation of a symmetric matrix', Numer. Math., Vol. 11, 1968, pp 181-195.

21. FRANCIS, J. G. F., 'The QR transformation, part I and II', Computer J., Vol. 4, 1961-1962, pp. 265-332.

22. PARLETT, B. N., 'Convergence of the QR algorithm', Numer. Math., Vol. 7, 1965, pp 187; Vol. 10, 1967, p 163.

23. PARLETT, B. N., 'Global convergence of the basic QR algorithm on Hessenberg matrices', Mathematical Computation, Vol. 22, 1968, p 803.

24. PETERS, G. and WILKINSON, J. H., 'Ax = λBx and the generalised eigenproblem', S.I.A.M. J. of Numer. anal., Vol. 7, 1970, pp 479-492.

25. BARTH, W., MARTIN, R. S. and WILKINSON, J. H., 'Calculation of the eigenvalue of a symmetric tridiagonal matrix by the method of bisection', Numer. Math., Vol. 9, 1967, pp 386-393.

26. GUPTA, K. K., 'Vibrations of frames and other structures with banded stiffness matrices', Int. J. for numer. meths. in engng.,Vol. 2, 1970, pp 221-228.

27. GUPTA, K. K., 'Solution of the eigenproblem by the Sturm sequence method', Int. J. for numer. meths. in engng.,Vol. 4, 1972, pp 379-404.

28. MEIROVITCH, L., 'Analytical methods in vibrations', Macmillan, New York, 1967.

29. UHRIG, R., 'Reduction of the number of unknowns in the displacement method applied to kinetic problems', J. of Sound and Vib., Vol. 4, No. 2, 1966, pp 149-155.

30. RAMSDEN, J. N. and STOKER, J. R., 'Mass condensation; a semi-automatic method for reducing the size of vibration problems', Int. J. of numer. meths. in engng., Vol. 1, 1969, pp 333-349.

31. GERADIN, M., 'Error bounds for eigenvalue analysis by elimination of variables', J. of Sound and Vib., Vol. 19, 1971, pp 111-132.

32. HURTY, W. C., 'Dynamic analysis of structural systems using component modes', A.I.A.A.J., Vol. 3, 1965, p 678.

33. GOLDMAN, L. R., 'Vibration analysis by dynamic partitioning', A.I.A.A. J., Vol. 7, No. 6, 1969.

34. HON, S., 'Review of model synthesis techniques and a new approach', Shock and Vib. Bull., Vol. 40, No. 4, 1969.

35. CRAIG, R. R. and BAMPTON, M. C. C., 'Coupling of sub-structures for dynamic analysis', A.I.A.A. J., Vol. 6, No. 7, 1968.

36. RUBINSTEIN, M. F. 'Structural systems - Statics, Dynamics and Stability', Prentice-Hall, Inc., Englewood Cliffs, N.J., 1970.

37. RUTISHAUSER, H., 'Computational aspects of F. L. Bauer's simultaneous iteration method', Numer. Math., Vol. 13, 1969, pp 4-13.

38. BAUER, F. L., 'Das Verfahren der Treppen-Iteration und verwandte verfahren zur lösung Algebraischer Eigenvertprobleme', Z.A.M.P., Vol. 8, 1957, pp 214-235.

39. JENNINGS, A., 'A direct iteration method of obtaining latent roots and vectors of a symmetric matrix', Proc. of the Cambridge Phil. Soc., Vol. 63, 1967, pp 755-765.

40. JENNINGS, A. and ORR, D. R. L., 'Application of the Simultaneous Iteration method to undamped vibration problems', Int. J. for numer. meths. in engng., Vol. 3, 1971, pp 13-24.

41. DONG, S. B., WOLF, J. A. and PETERSON, F. E., 'On a Direct-Iterative Eigensolution Technique', Int. J. for numer. meths. in engng., Vol. 4, 1972, pp 155-162.

42. YAMAMOTO, Y. and OHTSUBO, H., 'Subspace iteration accelerated by using Chebyshev polynomials with symmetric matrices', Int. J. of numer. meths. in engng., Vol. 10, Part 2, pp 935-944.

43. BATHE, K. J. and RAMASWAMY, S., 'An accelerated subspace iteration method', Computer Meths. in app. Mechanics and eng., Vol. 23, 1980, pp 313-331.

44. AKL, F. A., DILGER, W. H. and IRONS, B. M., 'Over relaxation and subspace iteration', Letter to ed. Int. J. of numer. meths. in engng., Vol. 14, No. 4, 1979, pp 629-630.

45. AKL, F. A., DILGER, W. H. and IRONS, B. M., 'Acceleration of subspace iteration', Int. J. of numer. meths. in engng., Vol. 18, No. 4, 1982, pp 583-591.

46. SHAH, S. A., 'Departmental Report No. 111', Dept. of Mech. Eng., University of Sheffield, 1982.

47. WARBURTON, G. B., 'The Vibration of rectangular plates', Proc. of the Inst. of Mech. Eng., Vol. 168, 1954, pp 371-383.

48. JENNINGS, A. and AGAR, T., 'Hybrid sturm sequence and simultaneous iteration methods', In Proc. Symp. Appl. Comput. Meths. Engng., University of Southern California, Los Angeles, California, 1977, pp 23-26.

49. WILSON, E. L. and ITOH, T., 'An eigensolution strategy for large systems', An Int. J. of Computers and Structures, Vol. 16, No. 1-4, 1983, pp 259-265.