

# THE EIGENVALUE PROBLEM

*Eigenvalues And Eigenvectors Are Properties Of The Equations That Simulate The Behavior Of A Real Structure*

## D.1 INTRODUCTION

The classical mathematical eigenvalue problem is defined as the solution of the following equation:

$$\mathbf{A}\mathbf{v}_n = \lambda_n \mathbf{v}_n \quad n = 1, \dots, N \quad (\text{D.1})$$

The  $N$  by  $N$   $\mathbf{A}$  matrix is real and symmetric; however, it may be singular and have zero eigenvalues  $\lambda_n$ . A typical eigenvector  $\mathbf{v}_n$  has the following orthogonality properties:

$$\begin{aligned} \mathbf{v}_n^T \mathbf{v}_n = 1 \quad \text{and} \quad \mathbf{v}_n^T \mathbf{v}_m = 0 \quad \text{if } n \neq m, \text{ therefore} \\ \mathbf{v}_n^T \mathbf{A}\mathbf{v}_n = \lambda_n \quad \text{and} \quad \mathbf{v}_n^T \mathbf{A}\mathbf{v}_m = 0 \quad \text{if } n \neq m \end{aligned} \quad (\text{D.2})$$

If all eigenvectors  $\mathbf{V}$  are considered the problem can be written as

$$\mathbf{A}\mathbf{V} = \mathbf{V}\mathbf{\Omega} \quad \text{or} \quad \mathbf{V}^T \mathbf{A}\mathbf{V} = \mathbf{\Omega} \quad (\text{D.3})$$

There are many different numerical methods to solve Equation (D.3) for eigenvectors  $\mathbf{V}$  and the diagonal matrix of eigenvalues  $\mathbf{\Omega}$ . In structural analysis, in general, it is only necessary to solve for the exact eigenvalues of small systems. Therefore, the most reliable and robust will be selected since the computational time will always be relatively small. For the determination of the dynamic mode shapes

and frequencies of large structural systems subspace iteration or Load Dependent Ritz, LDR, vectors are the most efficient approaches.

## D.2 THE JACOBI METHOD

One of the oldest and most general approaches for the solution of the classical eigenvalue problem is the Jacobi method that was first presented in 1846. This is a simple iterative algorithm in which the eigenvectors are calculated from the following series of matrix multiplications:

$$\mathbf{V} = \mathbf{T}^{(0)}\mathbf{T}^{(1)} \dots \mathbf{T}^{(k)} \dots \mathbf{T}^{(n-1)}\mathbf{T}^{(n)} \quad (\text{D.4})$$

The starting transformation matrix  $\mathbf{T}^{(0)}$  is set to a unit matrix. The iterative orthogonal transformation matrix  $\mathbf{T}^{(k)}$ , with four nonzero terms in the  $i$  and  $j$  rows and columns, is of the following orthogonal form:

$$\mathbf{T}^{(k)} = \begin{bmatrix} - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \\ - & - & T_{ii} & - & - & T_{ij} & - & - \\ - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \\ - & - & T_{ji} & - & - & T_{jj} & - & - \\ - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - \end{bmatrix} \quad (\text{D.5})$$

The four nonzero terms are functions of an unknown rotation angle  $\theta$  and are defined by

$$T_{ii} = T_{jj} = \cos \theta \quad \text{and} \quad T_{ji} = -T_{ij} = \sin \theta \quad (\text{D.6})$$

Therefore,  $\mathbf{T}^{(k)T}\mathbf{T}^{(k)} = \mathbf{I}$  which is independent of the angle  $\theta$ . The typical iteration involves the following matrix operation:

$$\mathbf{A}^{(k)} = \mathbf{T}^{(k)T}\mathbf{A}^{(k-1)}\mathbf{T}^{(k)} \quad (\text{D.7})$$

The angle is selected to force the terms  $i,j$  and  $j,i$  in the matrix  $\mathbf{A}^{(k)}$  to be zero. This is satisfied if the angle is calculated from

$$\tan 2\theta = \frac{2A_{ij}^{(k-1)}}{A_{ii}^{(k-1)} - A_{jj}^{(k-1)}} \quad (\text{D.8})$$

The classical Jacobi eigenvalue algorithm is summarized within the computer subroutine given in Table D.1.

**Table D.1 Subroutine to Solve the Symmetric Eigenvalue Problem**

<pre> SUBROUTINE JACOBI(A,V,NEQ,TL) IMPLICIT REAL*8 (A-H,O-Z) DIMENSION A(NEQ,NEQ),V(NEQ,NEQ) EIGENVALUE SOLUTION BY JACOBI METHOD - C WRITTEN BY ED WILSON DEC. 25, 1990 C A - MATRIX (ANY RANK) TO BE SOLVED --- C EIGENVALUES ON DIAGONAL C V - MATRIX OF EIGENVECTORS PRODUCED C TL- NUMBER OF SIGNIFICANT FIGURES C----- INITIALIZATION ----- ZERO = 0.0D0 SUM = ZERO TOL = DABS(TL) C----- SET INITIAL EIGENVECTORS ----- DO 200 I=1,NEQ DO 190 J=1,NEQ IF (TL.GT.ZERO) V(I,J) = ZERO 190 SUM = SUM + DABS(A(I,J)) IF (TL.GT.ZERO) V(I,I) = 1.0 200 CONTINUE C----- CHECK FOR TRIVIAL PROBLEM ----- IF (NEQ.EQ.1) RETURN IF (SUM.LE.ZERO) RETURN SUM = SUM/DFLOAT(NEQ*NEQ) C----- C----- REDUCE MATRIX TO DIAGONAL ----- C----- 400 SSUM = ZERO AMAX = ZERO DO 700 J=2,NEQ IH = J - 1 DO 700 I=1,IH C----- CHECK IF A(I,J) IS TO BE REDUCED ----- AA = DABS(A(I,J)) IF (AA.GT.AMAX) AMAX = AA SSUM = SSUM + AA IF (AA.LT.0.1*AMAX) GO TO 700 </pre>	<pre> C----- CALCULATE ROTATION ANGLE ----- AA=ATAN2(2.0*A(I,J),A(I,I)-(J,J))/2.0 SI = DSIN(AA) CO = DCOS(AA) C----- MODIFY "I" AND "J" COLUMNS ----- DO 500 K=1,NEQ TT = A(K,I) A(K,I) = CO*TT + SI*A(K,J) A(K,J) = -SI*TT + CO*A(K,J) TT = V(K,I) V(K,I) = CO*TT + SI*V(K,J) 500 V(K,J) = -SI*TT + CO*V(K,J) C----- MODIFY DIAGONAL TERMS ----- A(I,I) = CO*A(I,I) + SI*A(J,I) A(J,J) = -SI*A(I,J) + CO*A(J,J) A(I,J) = ZERO C----- MAKE "A" MATRIX SYMMETRICAL ----- DO 600 K=1,NEQ A(I,K) = A(K,I) A(J,K) = A(K,J) 600 CONTINUE C----- A(I,J) MADE ZERO BY ROTATION ----- 700 CONTINUE C----- CHECK FOR CONVERGENCE ----- IF(DABS(SSUM)/SUM .GT.TOL)GO TO 400 RETURN END </pre>
---	---

One notes that the subroutine for the solution of the symmetric eigenvalue problem by the classical Jacobi method does not contain a division by any number. Also, it can be proved that, after each iteration cycle, the absolute sum of the off-diagonal terms is always reduced. Hence, the method will always converge and yield an accurate solution for positive, zero or negative eigenvalues.

The Jacobi algorithm can be directly applied to all off-diagonal terms, in sequence, until all terms are reduced to a small number compared to the absolute value of all terms in the matrix. However, the subroutine presented uses a “threshold” approach in which it skips the relatively small off-diagonal terms and operates only on the large off-diagonal terms.

To reduce one off-diagonal term to zero requires approximately  $8N$  numerical operations. Clearly, one cannot precisely predict the total number of numerical operation because it is an iterative method; however, experience has indicated that the total number of numerical operations to obtain convergence is the order of  $10N^3$ . Assuming a modern (1998) personal computer can perform over 6,000,000 operations a second, it would require approximately one second of computer time to calculate the eigenvalues and eigenvectors of a full 100 by 100 matrix.

### D.3 CALCULATION OF 3D PRINCIPAL STRESSES

The calculation of the principal stresses for a three dimensional solid can be numerically evaluated from the stresses in the x-y-z system by the solution of a cubic equation. However, the definition of the directions of the principal stresses is not a simple procedure. An alternative approach to this problem is to write the basic stress transformation equation in terms of the unknown directions of the principal stresses in the 1-2-3 reference system. Or,

$$\begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} = \begin{bmatrix} V_{x1} & V_{y1} & V_{z1} \\ V_{x2} & V_{y2} & V_{z2} \\ V_{x3} & V_{y3} & V_{z3} \end{bmatrix} \begin{bmatrix} \sigma_x & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_y & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_z \end{bmatrix} \begin{bmatrix} V_{x1} & V_{x2} & V_{x3} \\ V_{y1} & V_{y2} & V_{y3} \\ V_{z1} & V_{z2} & V_{z3} \end{bmatrix} \quad (\text{D.9})$$

Or, in symbolic form

$$\Omega = \mathbf{V}^T \mathbf{S} \mathbf{V} \quad (\text{D.10})$$

in which  $\mathbf{V}$  is the standard direction cosine matrix. Since  $\mathbf{V} \mathbf{V}^T$  is a unit matrix Equation (D.3) can be written as the following eigenvalue problem:

$$\mathbf{S}\mathbf{V} = \mathbf{V}\mathbf{\Omega} \quad (\text{D.11})$$

where  $\mathbf{\Omega}$  is an unknown diagonal matrix of the principal stresses (eigenvalues) and  $\mathbf{V}$  is the unknown direction cosine matrix (eigenvectors) which uniquely define the directions of the principal stresses. In order to illustrate the practical application of the classical Jacobi method consider the following state of stress:

$$\mathbf{S} = \begin{bmatrix} \sigma_x & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_y & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_z \end{bmatrix} = \begin{bmatrix} 120 & -55 & -75 \\ -55 & -55 & 33 \\ -75 & 33 & -85 \end{bmatrix} \quad (\text{D.12})$$

The eigenvalues, principal stresses, and eigenvectors (direction cosines) are

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \end{bmatrix} = \begin{bmatrix} 162.54 \\ -68.40 \\ -114.14 \end{bmatrix} \text{ and } \mathbf{v} = \begin{bmatrix} .224 & .352 & .909 \\ -.308 & .910 & -.277 \\ .925 & .217 & -.312 \end{bmatrix} \quad (\text{D.13})$$

The solution of a 3 by 3 eigenvalue problem can be considered as a trivial numerical problem. Several hundred of these problems can be solved in one second of computer time.

#### D.4 SOLUTION OF THE GENERAL EIGENVALUE PROBLEM

The general eigenvalue problem is written as

$$\mathbf{A}\mathbf{V} = \mathbf{B}\mathbf{V}\mathbf{\Omega} \quad (\text{D.14})$$

where both  $\mathbf{A}$  and  $\mathbf{B}$  are symmetrical matrices. The first step is to calculate the eigenvectors  $\mathbf{V}_B$  of the  $\mathbf{B}$  matrix. We can now let the eigenvectors  $\mathbf{V}$  be a linear combination of the eigenvectors of the  $\mathbf{B}$  matrix. Or,

$$\mathbf{V} = \mathbf{V}_B \bar{\mathbf{V}} \quad (\text{D.15})$$

Substitution of Equation (D.15) into Equation (D.14) and the pre multiplication of both sides by  $\mathbf{V}_B^T$  yields

$$\mathbf{V}_B^T \mathbf{A} \mathbf{V}_B \bar{\mathbf{V}} = \mathbf{V}_B^T \mathbf{B} \mathbf{V}_B \bar{\mathbf{V}} \Omega \quad (\text{D.16})$$

If all eigenvalues of the  $\mathbf{B}$  matrix are nonzero the eigenvectors can be normalized so that  $\mathbf{V}_B^T \mathbf{B} \mathbf{V}_B = \mathbf{I}$ . Hence, Equation (D.16) can be written in the following classical form:

$$\bar{\mathbf{A}} \bar{\mathbf{V}} = \bar{\mathbf{V}} \Omega \quad (\text{D.17})$$

where  $\bar{\mathbf{A}} = \mathbf{V}_B^T \mathbf{A} \mathbf{V}_B$ . Therefore, the general eigenvalue problem can be solved by the application of the Jacobi algorithm to both matrices. If the  $\mathbf{B}$  matrix is diagonal the eigenvectors  $\mathbf{V}_B$  matrix will be diagonal with the diagonal terms equal to  $1/\sqrt{B_{nn}}$ . This is the case for a lumped mass matrix. Also, mass must be associated with all degrees-of-freedom and all eigenvectors and values must be calculated.

## D.5 SUMMARY

Only the Jacobi method has been presented in detail in this section. It is restricted to small full matrices in which all eigenvalues are required. For this problem the method is very robust and simple to program. For the dynamic modal analysis of large structural systems; or, for the stability analysis of structural systems, other more numerically efficient methods are recommended.