$$\mathbf{A}\mathbf{x}_j = \lambda_j \mathbf{x}_j; \qquad \lambda_j = -m\omega_j^2. \tag{5.57}$$

Because the set of equations expressed by 5.57 is homogeneous, no solution exists unless the determinate $|\mathbf{A} - \lambda| = 0$. Hence only the discrete values of λ_j (or ω_j), corresponding to the roots of this polynomial, are possible. These discrete values are called the eigenvalues of the system.

5.3.2 Finding the Eigenvalues of a Matrix

Expansion of the determinant

$$|\mathbf{A} - \lambda|$$

gives a polynomial in λ, called the secular polynomial, and equating this polynomial to zero gives the secular equation for the eigenvalues. Since our basic system is symmetric and real (Hermitian would do as well) the eigenvalues are real. For each of these eigenvalues, λ_i, there is associated a vector which is the solution, the eigenvector $\mathbf{x}_i$.

Consider the following two equations, assuming two distinct eigenvalues;

$$\mathbf{A}\mathbf{x}_a = \lambda_a \mathbf{x}_a, \quad \text{and} \quad \mathbf{A}\mathbf{x}_b = \lambda_b \mathbf{x}_b$$

Multiplying the first equation by the transpose of $\mathbf{x}_b$ $[\equiv \tilde{\mathbf{x}}_b]$, the second by $\tilde{\mathbf{x}}_a$ and subtracting, we have

$$(\tilde{\mathbf{x}}_b \mathbf{A}\mathbf{x}_a) - (\tilde{\mathbf{x}}_a \mathbf{A}\mathbf{x}_b) = \lambda_a(\tilde{\mathbf{x}}_b \mathbf{x}_a) - \lambda_b(\tilde{\mathbf{x}}_a \mathbf{x}_b),$$

Since A is symmetric,

$$\begin{aligned}(\tilde{\mathbf{x}}_b \mathbf{A}\mathbf{x}_a) &= \sum x_b^i A_{ij} x_a^j &&= \sum x_b^i A_{ji} x_a^j \\ &= \sum x_a^j A_{ji} x_b^i &&= \sum x_a^i A_{ij} x_b^j \ . \\ &= (\tilde{\mathbf{x}}_a \mathbf{A}\mathbf{x}_b)\end{aligned}$$

Also, since $(\tilde{\mathbf{x}}_b \mathbf{x}_a) = (\tilde{\mathbf{x}}_a \mathbf{x}_b)$ we find

$$(\lambda_a - \lambda_b)(\tilde{\mathbf{x}}_b \mathbf{x}_a) = 0$$

so that eigenvectors corresponding to distinct eigenvalues are orthogonal. Assuming that we have the case of no degenerate eigenvalues, there must exist n independent eigenvectors which span the space (by which we mean that they constitute a complete set of all possible vectors in the n-dimensional space). Hence we can always write an arbitrary vector, $\mathbf{y}$, as an expansion in the eigenvectors of A

$$\mathbf{y} = \sum_{i=1}^{n} \beta_i \mathbf{x}_i.$$

The Power Method for Finding Eigenvalues

Suppose we multiply a vector $\mathbf{y}$ chosen at random by the matrix A. Using the expansion of $\mathbf{y}$ in the eigenvectors of A given just above, we have

$$\mathbf{A}\mathbf{y} = \sum_{i=1}^{n} \beta_i \mathbf{A}\mathbf{x}_i = \sum_{i=1}^{n} \beta_i \lambda_i \mathbf{x}_i.$$

After multiplying k times by A we find

$$\mathbf{A}^k\mathbf{y} = \sum \beta_i \lambda_i^k \mathbf{x}_i.$$

We see that each multiplication increases the coefficient of $\mathbf{x}_i$ by the factor λ_i. Thus the vector becomes "richer" in the eigenvector with the largest (magnitude) eigenvalue with each multiplication. If we define a sequence of vectors $\mathbf{y}_k$, starting with some arbitrary vector $\mathbf{y}_1$ such that:

$$\mathbf{y}_k \equiv \mathbf{A}\mathbf{y}_{k-1}.$$

$\mathbf{y}_k$ will converge to the eigenvector with the largest eigenvalue and $\frac{|\mathbf{y}_{k+1}|}{|\mathbf{y}_k|}$ will converge to the magnitude of the largest eigenvalue. This can be a slow process since the enhancement of of the eigenvector corresponding to the largest eigenvalue over the next largest is only λ_n/λ_{n-1} and the ratio of the two largest eigenvalues may be close to one.

The Inverse Power Method

We can speed up the process if we note that

$$\mathbf{A}\mathbf{x}_i = \lambda_i \mathbf{x}_i \quad \text{implies} \quad \mathbf{A}^{-1}\mathbf{x}_i = \lambda_i^{-1}\mathbf{x}_i \tag{5.58}$$

so that if we apply the power method to the inverse of A, we will find the *smallest* eigenvalue and its corresponding eigenvector. This procedure implies a sequence of vectors,

$$\mathbf{y}_{k+1} = \mathbf{A}^{-1}\mathbf{y}_k$$

which we can write as

$$\mathbf{A}\mathbf{y}_{k+1} = \mathbf{y}_k.$$

Thus we use $\mathbf{y}_k$ as the input to a set of linear equations and *solve* for $\mathbf{y}_{k+1}$. Since matrix multiplication on a vector is an n^2 procedure and Gaussian elimination is n^3 it would seem to be a much slower method for large matrices. But we are solving with the same matrix each time and thus only need to make the LU decomposition once. Thus the inverse power approach involves n^3 operations only in the first step and n^2 on each succeeding step.

The inverse power method gains considerably in importance when we realize that we can shift the eigenvalues by a constant.

$$\mathbf{A}'\mathbf{x} = (\mathbf{A} - q\mathbf{I})\mathbf{x} = (\lambda - q)\mathbf{x} \tag{5.59}$$

If we can guess a value, q, close to the eigenvalue desired then we can reduce the eigenvalue of the modified system to near zero, causing the convergence to accelerate considerably. This observation places a premium on being able to find good approximations to the eigenvalue system.

Once we have found the first eigenvalue we can continue to find the next (and the rest) by the following procedure. When we chose (at random) the starting value for y it was hoped that it contained a significant amount of the eigenvector corresponding to the smallest eigenvalue. If the coefficient of this eigenvector, β_0, was small it could be built up by the multiplication process (and by round-off error if nothing else). However, until this buildup occurs we must expect the vectors $\mathbf{y}_k$ to be a combination of the other eigenvectors. We can use this to our advantage (once we have found the first eigenvector) by removing the eigenvector corresponding to the first eigenvalue found from the trial vector y. Since

$$\mathbf{y} = \sum \beta_i \mathbf{x}_i$$

then (assuming that $\mathbf{x}_i$ is normalized)

$$\beta_i = (\tilde{\mathbf{x}}_i \mathbf{y})$$

so that

$$\mathbf{y}' \equiv \mathbf{y} - (\tilde{\mathbf{x}}_0 \mathbf{y})\mathbf{x}_0$$

is orthogonal to the first eigenvector ($\mathbf{x}_0$ found. Starting with this new y, one may expect to find the next smallest eigenvalue by the inverse power method. Of course aiming the value of q to the next indicated eigenvalue enhances convergence of the method as well.

To help us obtain a better extraction of the eigenvalues from the approximate eigenvectors that we have generated, consider the Rayleigh-Ritz ratio;

$$R = \frac{(\tilde{\mathbf{y}}\mathbf{A}\mathbf{y})}{(\tilde{\mathbf{y}}\mathbf{y})}. \tag{5.60}$$

If we suppose that we have a vector y ($\mathbf{y} = \sum \beta_i \mathbf{x}_i$) whose largest component is a given eigenvector, say $\mathbf{x}_0$, then write

$$R = \frac{\sum \lambda_i \beta_i^2}{\sum \beta_i^2}.$$

If we believe that β_0 is dominant in the sums then this ratio gives a good estimate of λ_0. Applying this estimate to $\mathbf{A}^{-1}$ in the inverse power procedure should provide a good approximation for the next value of q.

5.3.3 Tridiagonal Symmetric Matrices

For a tridiagonal matrix it becomes practical to use the secular polynomial to solv for the eigenvalues. While the discussion of such a specialized matrix may seem t be too restrictive, the remainder of the methods discussed in this chapter for findin eigenvalues will depend on reducing an arbitrary real symmetric matrix to tridiagona form. The importance of the solution for the eigenvalues of the tridiagonal matrix i that it provides the basis for a number of more general methods.

Consider the secular polynomial provided by the determinant of the matrix:

$$f_n(\lambda) =$$

$$\begin{vmatrix}
a_1-\lambda & b_1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\
b_1 & a_2-\lambda & b_2 & 0 & \dots & 0 & 0 & 0 & 0 \\
0 & b_2 & a_3-\lambda & b_3 & \dots & 0 & 0 & 0 & 0 \\
0 & 0 & b_3 & a_4-\lambda & \dots & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & \dots & a_{n-3}-\lambda & b_{n-3} & 0 & 0 \\
0 & 0 & 0 & 0 & \dots & b_{n-3} & a_{n-2}-\lambda & b_{n-2} & 0 \\
0 & 0 & 0 & 0 & \dots & 0 & b_{n-2} & a_{n-1}-\lambda & b_{n-1} \\
0 & 0 & 0 & 0 & \dots & 0 & 0 & b_{n-1} & a_n-\lambda
\end{vmatrix} \tag{5.61}$$

An efficient manner for the evaluation of this determinant is that of expansion b minors. Let us expand using the last row.

$$f_n(\lambda) =$$

$$(a_n-\lambda)\begin{vmatrix}
a_1-\lambda & b_1 & 0 & 0 & \dots & 0 & 0 & 0 \\
b_1 & a_2-\lambda & b_2 & 0 & \dots & 0 & 0 & 0 \\
0 & b_2 & a_3-\lambda & b_3 & \dots & 0 & 0 & 0 \\
0 & 0 & b_3 & a_4-\lambda & \dots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & \dots & a_{n-3}-\lambda & b_{n-3} & 0 \\
0 & 0 & 0 & 0 & \dots & b_{n-3} & a_{n-2}-\lambda & b_{n-2} \\
0 & 0 & 0 & 0 & \dots & 0 & b_{n-2} & a_{n-1}-\lambda
\end{vmatrix}$$

$$-b_{n-1}\begin{vmatrix} a_1-\lambda & b_1 & 0 & 0 & \dots & 0 & 0 & 0 \\ b_1 & a_2-\lambda & b_2 & 0 & \dots & 0 & 0 & 0 \\ 0 & b_2 & a_3-\lambda & b_3 & \dots & 0 & 0 & 0 \\ 0 & 0 & b_3 & a_4-\lambda & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & a_{n-3}-\lambda & b_{n-3} & 0 \\ 0 & 0 & 0 & 0 & \dots & b_{n-3} & a_{n-2}-\lambda & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & b_{n-2} & b_{n-1} \end{vmatrix}$$

$$= (a_n - \lambda) f_{n-1} - b_{n-1}^2 f_{n-2}$$

Clearly the sequence of expansions can be continued for $n-1$ etc., until we reach the upper left hand corner of the matrix. There we have:

$$f_1(\lambda) = a_1 - \lambda$$

$$f_2(\lambda) = \begin{vmatrix} a_1-\lambda & b_1 \\ b_1 & a_2-\lambda \end{vmatrix} = (a_2-\lambda)(a_1-\lambda) - b_1^2 = (a_2-\lambda)f_1(\lambda) - b_1^2.$$

Hence we may use the recursion relation:

$$f_k(\lambda) = (a_k - \lambda) f_{k-1}(\lambda) - b_{k-1}^2 f_{k-2}(\lambda) \tag{5.62}$$

for all values of k if we set $f_0(\lambda) \equiv 1$. In this manner we can rapidly generate the secular determinant, $f_n(\lambda)$, for any value of λ through the relation 5.62. We can then use a numerical method for searching for the zeros of polynomials to find a zero of $f_n(\lambda)$ which will supply one of the eigenvalues. We can do better than this, however. We can specify a given eigenvalue in the sequence $\lambda_1 \geq \lambda_2 \geq \lambda_3 \dots \geq \lambda_n$.

To see how the ordering of the eigenvalues comes about, consider the sequence of polynomials

$$f_0(\lambda),\ f_1(\lambda),\ f_2(\lambda),\ \dots,\ f_n(\lambda)$$

as a function of λ. From Eq. 5.62 we can see that if any two members of a consecutive pair $(k, k-1)$ have a common zero, then *all* polynomials in the sequence have the same zero. But this possibility is contradicted since $f_0(\lambda)$ has *no* zero. Thus we conclude that, as we vary λ, no two members of a consecutive pair can pass through zero at the same time (for the same value of λ).

Suppose that, as λ varies through a certain range, $f_{k-1}(\lambda)$ passes through zero at $\lambda = \lambda_0$. Since $f_{k-2}(\lambda)$ and $f_k(\lambda)$ do not change sign in this interval and $f_k(\lambda)$ and $f_{k-2}(\lambda)$ have opposite signs near a zero of $f_{k-1}(\lambda)$ (from Eq. 5.62), the number of agreements in signs between adjacent members of the sequence is unchanged. That is, if $f_{k-2}(\lambda_0) < 0$, $f_k(\lambda_0) > 0$ while $f_{k-1}(\lambda_0 - \epsilon) < 0$ and $f_{k-1}(\lambda_0 + \epsilon) > 0$ then there

is one sign difference in the sequence f_{k-2}, f_{k-1}, f_k both just above λ_0 and just below λ_0. Thus the total number of sign differences in the sequence can only be altered by a change in sign of one of the *end* members of the chain. But since $f_0(\lambda)$ has no sign changes, the number of sign differences in the full sequence is an indication of the position of λ among the zeros of $f_n(\lambda)$.

To get an absolute count on the eigenvalue positions, consider a quantity R greater than the absolute value of all of the eigenvalues. Then, for $\lambda = -R$ all $f_m(\lambda)$ will be positive and there will be no sign differences between consecutive members. For $\lambda = R$ each $f_m(\lambda)$ will alternate in sign and there will be n sign differences. Since each time λ is reduced so that $f_n(\lambda)$ passes through zero (which happens at each eigenvalue) one sign difference is removed from the sequence, the number of sign differences is equal to the number of eigenvalues less than the value of λ chosen.

For a practical algorithm choose R from the Gerschgorin disks*

$$R = \max(|b_i| + |a_i| + |b_{i-1}|); \quad i = 1, 2, \dots n$$

and initialize $\lambda_{min} = -R$ and $\lambda_{max} = R$. If we are searching for a specific eigenvalue, say λ_ℓ then:

1. set $\lambda = \frac{1}{2}(\lambda_{min} + \lambda_{max})$
2. compute m = the number of sign differences in the sequence $f_0(\lambda)$, $f_1(\lambda)$, ... $f_n(\lambda$
3. if $m < \ell$ we need to search higher in λ so set $\lambda_{min} = \lambda$; else set $\lambda_{max} = \lambda$
4. if $\lambda_{max} - \lambda_{min} < error$ quit; else go to 1.

To carry out this procedure we need a subroutine which implements the recursion given by Eq. 5.62 and returns just the number of sign differences in the sequence. The main routine can perform the search indicated above.

5.3.4 The Role of Orthogonal Matrices

Note that, for any matrix P which has an inverse,

$$\left|\mathrm{P}^{-1}\mathrm{AP} - \lambda\mathrm{I}\right| = \left|\mathrm{P}^{-1}(\mathrm{A} - \lambda\mathrm{I})\mathrm{P}\right| = \left|\mathrm{P}^{-1}\mathrm{P}(\mathrm{A} - \lambda\mathrm{I})\right| = \left|\mathrm{A} - \lambda\mathrm{I}\right|$$

so the matrix $\mathrm{P}^{-1}\mathrm{AP}$ has the same eigenvalues as A. We will shortly be interested in trying to transform the matrix A by $\mathrm{B} = \tilde{\mathrm{P}}\mathrm{AP}$ to uncover a more tractable system for finding eigenvalues.

*See Ref 1, page 159 for a discussion of these circles which bound the eigenvalues of a matrix.

Matrices with orthogonal columns have a special place in the determination of eigenvalues. For these matrices remember that we have:

$$\sum_{k=1}^{n} P_{ki}P_{kj} = \delta_{ij}$$

so that $\tilde{\mathbf{P}} = \mathbf{P}^{-1}$. In the remainder of this chapter we shall assume that $\mathbf{P}$ denotes an orthogonal matrix.

The transformation $\mathbf{B} = \tilde{\mathbf{P}}\mathbf{A}\mathbf{P}$ may be regarded as a change of basis. To see this we write an arbitrary vector in the space of n-dimension column vectors as:

$$\mathbf{x} = x_1\begin{pmatrix}1\\0\\0\\\vdots\\0\end{pmatrix} + x_2\begin{pmatrix}0\\1\\0\\\vdots\\0\end{pmatrix}\ldots + x_2\begin{pmatrix}0\\0\\0\\\vdots\\1\end{pmatrix}\ldots$$

$$= x_1\mathbf{e}_1 + x_2\mathbf{e}_2\ldots + x_n\mathbf{e}_n\ldots$$

Here the $\mathbf{e}_i$ form the natural unit basis vectors.

We have also

$$\mathbf{P}\mathbf{x} = x_1\mathbf{P}_1 + x_2\mathbf{P}_2\ldots + x_n\mathbf{P}_n$$

where the $\mathbf{P}_i$ are the columns of $\mathbf{P}$ which may be taken to form an alternate orthonormal basis. If

$$\mathbf{B} = \tilde{\mathbf{P}}\mathbf{A}\mathbf{P}$$

(remember $\mathbf{B}$ has the same eigenvalues as $\mathbf{A}$) then the elements of the matrix $\mathbf{B}$ are:

$$b_{\alpha\beta} = (\tilde{\mathbf{e}}_\alpha\mathbf{B}\mathbf{e}_\beta) = (\tilde{\mathbf{e}}_\alpha\tilde{\mathbf{P}}\mathbf{A}\mathbf{P}\mathbf{e}_\beta)$$

$$= \sum_{i,j,k}\delta_{\alpha i}P_{ji}A_{jk}P_{k\ell}\delta_{\ell\beta} = \sum_{j,k}P_{j\alpha}A_{jk}P_{k\beta}$$

$$= (\tilde{\mathbf{P}}_\alpha\mathbf{A}\mathbf{P}_\beta).$$

Thus, for the purpose of determining the eigenvalues of $\mathbf{A}$, we may chose any orthogonal basis for the computation of its elements that we find convenient.

If we knew how to choose the columns of $\mathbf{P}$ to be the eigenvectors of $\mathbf{A}$ then:

$$b_{\alpha\beta} = (\tilde{\mathbf{x}}_\alpha\mathbf{A}\mathbf{x}_\beta) = \lambda_\beta(\tilde{\mathbf{x}}_\alpha\mathbf{x}_\beta) = \lambda_\alpha\delta_{\alpha\beta},$$

and $\mathbf{B}$ would be a diagonal matrix with elements equal to the eigenvalues. For this reason finding the eigenvalues of a matrix is sometimes referred to as "diagonalizing the matrix." While we don't know how to find such a matrix $\mathbf{P}$ directly, there are several methods known to "almost" diagonalize it, i.e. to bring it into the tridiagonal form discussed in the previous section.

5.3.5 The Householder Method

One might think that a general matrix could be brought to diagonal form by the Householder transformation studied in section 5.2.3 by acting on the right hand side of the matrix $\mathbf{A}$ to put all of the non-diagonal elements of the rows to zero and on the left hand side to set the non-diagonal elements of the columns to zero. A problem arrises with this attack, however. If one applies the transformation $\mathbf{P}^{(0)}\mathbf{A}\tilde{\mathbf{P}}^{(0)}$ to a symmetric matrix, while the right-most factor can be chosen to reduce the first row (except for the first element) to zero, the left-most factor of $\mathbf{P}^{(0)}$ will destroy those zeros. However, if we start with $\mathbf{P}^{(1)}$ on both sides we can reduce $n-2$ elements of the row to zero and the application of the second $\mathbf{P}^{(1)}$ (on the left) will zero $n-2$ elements in the first column, leaving the first row unchanged. Schematically we have:

$$\mathbf{P}^{(1)}\mathbf{A}\mathbf{P}^{(1)} = \begin{pmatrix} a_{00} & a_R & a_R & a_R & \ldots \\ a_L & a_{LR} & a_{LR} & a_{LR} & \ldots \\ a_L & a_{LR} & a_{LR} & a_{LR} & \ldots \\ a_L & a_{LR} & a_{LR} & a_{LR} & \ldots \\ \vdots & \vdots & \vdots & & \end{pmatrix} \tag{5.63}$$

where a_{00} means that the element is unchanged, a_L or a_R means that the element was changed by the left (or right) multiplication and a_{LR} means that it was changed by both. The matrix can, however, be reduced to tridiagonal form by successive applications of the $\mathbf{P}^{(r)}$.

5.3.6 The Lanczos Algorithm

This very powerful algorithm generates an orthogonal change of basis and the tridiagonal matrix in the same procedure. It starts out like other methods we have studied by choosing a vector $\mathbf{y}_1$ at random and operating on it with $\mathbf{A}$. We assume that $\mathbf{y}_1$ (and the other vectors in the sequence to be generated) are normalized. The procedure continues as follows.

The resulting vector from this first operation is resolved into a component along $\mathbf{y}_1$ and a direction orthogonal to it.

$$\mathbf{A}\mathbf{y}_1 = a_1\mathbf{y}_1 + b_1\mathbf{y}_2. \tag{5.64}$$

Since we require $\mathbf{y}_2$ to be orthogonal to $\mathbf{y}_1$, we can calculate a_1 from

$$(\tilde{\mathbf{y}}_1\mathbf{A}\mathbf{y}_1) = a_1(\tilde{\mathbf{y}}_1\mathbf{y}_1) + b_1(\tilde{\mathbf{y}}_1\mathbf{y}_2) = a_1.$$

We can find b_1 from

$$b_1\mathbf{y}_2 \equiv \mathbf{A}\mathbf{y}_1 - a_1\mathbf{y}_1. \tag{5.65}$$

The condition that $\mathbf{y}_2$ is normalized permits the determination of b_1 (to within a sign, which is all that is needed). Note that from Eq. 5.65 we have

$$b_1(\tilde{\mathbf{y}}_2\mathbf{y}_2) = b_1 = (\tilde{\mathbf{y}}_2\mathbf{A}\mathbf{y}_1). \tag{5.66}$$

Next, $\mathbf{y}_2$ is multiplied by A to give

$$\mathbf{A}\mathbf{y}_2 = b_1'\mathbf{y}_1 + a_2\mathbf{y}_2 + b_2\mathbf{y}_3, \tag{5.67}$$

where we require that $\mathbf{y}_3$ is orthogonal to both $\mathbf{y}_1$ and $\mathbf{y}_2$. But $b_1' = b_1$ since

$$b_1 = (\tilde{\mathbf{y}}_2\mathbf{A}\mathbf{y}_1) = (\tilde{\mathbf{y}}_1\mathbf{A}\mathbf{y}_2) = b_1'. \tag{5.68}$$

The quantity a_2 is also known since $a_2 = (\tilde{\mathbf{y}}_2\mathbf{A}\mathbf{y}_2)$ and we can find b_2 and $\mathbf{y}_3$ from

$$b_2\mathbf{y}_3 = \mathbf{A}\mathbf{y}_2 - b_1\mathbf{y}_1 - a_2\mathbf{y}_2. \tag{5.69}$$

Acting with A on $\mathbf{y}_3$:

$$\mathbf{A}\mathbf{y}_3 = \alpha\mathbf{y}_1 + b_2\mathbf{y}_2 + a_3\mathbf{y}_3 + b_3\mathbf{y}_4 \tag{5.70}$$

but $\alpha = (\tilde{\mathbf{y}}_1\mathbf{A}\mathbf{y}_3) = (\tilde{\mathbf{y}}_3\mathbf{A}\mathbf{y}_1) = 0$, from Eq. 5.64 since $\mathbf{y}_3$ was chosen to be orthogonal to $\mathbf{y}_1$ and $\mathbf{y}_2$. As we continue the process there will be only 3 vectors on the r.h.s. of the equation; the two most recently determined vectors and the new one to be calculated in the current step:

$$\mathbf{A}\mathbf{y}_k = b_{k-1}\mathbf{y}_{k-1} + a_k\mathbf{y}_k + b_{k+1}\mathbf{y}_{k+1}. \tag{5.71}$$

In this manner, a sequence of orthogonal vectors and a tridiagonal matrix can be built up.

$$\begin{array}{lccccccc}
\mathbf{A}\mathbf{y}_1 = & a_1\mathbf{y}_1 & +b_1\mathbf{y}_2 & +0 & +0 & \dots & +0 \\
\mathbf{A}\mathbf{y}_2 = & b_1\mathbf{y}_1 & +a_2\mathbf{y}_2 & +b_2\mathbf{y}_3 & +0 & \dots & +0 \\
\mathbf{A}\mathbf{y}_3 = & 0 & +b_2\mathbf{y}_2 & +a_3\mathbf{y}_3 & +b_3\mathbf{y}_4 & \dots & +0 \\
\mathbf{A}\mathbf{y}_4 = & 0 & +0 & +b_3\mathbf{y}_3 & +a_4\mathbf{y}_4 & \dots & +0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & & \vdots \\
\mathbf{A}\mathbf{y}_n = & 0 & +0 & +0 & +0 & \dots & +a_n\mathbf{y}_n
\end{array}$$

Thus we have changed the basis for the representation of A to the matrix composed of the column vectors $\mathbf{y}_k$. In this representation it is tridiagonal.

To make this process more explicit, consider the transformation by an orthogonal matrix composed of the y-vectors just determined,

$$\mathbf{P} = (\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \dots, \mathbf{y}_n) \tag{5.72}$$

where we have written P as a row vector of column vectors. Then the matrix, B, (with equivalent eigenvalues) will be

$$\mathbf{B} = \begin{pmatrix} \tilde{\mathbf{y}}_1 \\ \tilde{\mathbf{y}}_2 \\ \vdots \\ \tilde{\mathbf{y}}_n \end{pmatrix} \mathbf{A}(\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n) \tag{5.73}$$

$$= \begin{pmatrix} \tilde{\mathbf{y}}_1 \\ \tilde{\mathbf{y}}_2 \\ \vdots \\ \tilde{\mathbf{y}}_n \end{pmatrix} (a_1\mathbf{y}_1 + b_1\mathbf{y}_2, b_1\mathbf{y}_1 + a_2\mathbf{y}_2 + b_2\mathbf{y}_3, \dots)$$

$$\mathbf{B} = \begin{pmatrix} a_1 & b_1 & 0 & 0 & \dots \\ b_1 & a_2 & b_2 & 0 & \dots \\ 0 & b_2 & a_3 & b_3 & \dots \\ \vdots & \vdots & \vdots & \vdots & \end{pmatrix} \tag{5.74}$$

where we have considered $\tilde{\mathbf{P}}$ to be a column vector of row vectors. In practice one need only compute $a_i = (\tilde{\mathbf{y}}_i\mathbf{A}\mathbf{y}_i)$ and $b_i\mathbf{y}_{i+1} = \mathbf{A}\mathbf{y}_i - a_i\mathbf{y}_i - b_{i-1}\mathbf{y}_{i-1}$ to generate the tridiagonal matrix so that only five n-dimensional vectors need be kept in memory: $\mathbf{y}_i$, $\mathbf{A}\mathbf{y}_i$, $\mathbf{y}_{i-1}$, a_i and b_i. Note that only the absolute value of b_i is determined by the algorithm but only b_i^2 is needed by the algorithm presented in section 5.3.3. If the action of A on a vector can be computed, so that it does not have to be stored in memory, then eigenvalues of extremely large matrices may be found.