CONTINUANT MATRICES IN NUMERICAL ANALYSIS

A REVIEW

by

John M. Bennett

School of Physics, University of Sydney,

Sydney, N.S.W.

1. INTRODUCTION

"A matrix, of order nxn, which has zero elements everywhere except in the principal diagonal, the super-diagonal directly above it, and the sub-diagonal directly below, is called a continuant matrix, and its determinant a continuant." 1 The name continuant dervives from a similarity between properties of continuant matrices and those of continued fractions, to which reference will be made in Section 16 of this article.

An example of a continuant matrix is the 4x4 matrix

$$\begin{bmatrix} a_1 & b_1 & & \\ c_1 & a_2 & b_2 & \\ & c_2 & a_3 & b_3 \\ & & c_3 & a_4 \end{bmatrix}$$

zero elements being omitted.

Continuant matrices arise when, for example, differential equations are expressed in finite difference form. Also many numerical processes can be described in terms of the properties of continuant matrices. It is the purpose of this review to give examples of how this can be done.

2. SOLUTION OF LINEAR SIMULTANEOUS EQUATIONS (1)

A simple application of the properties of continuant matrices leads to a method which is sometimes convenient in solving sets of simultaneous linear equations with a continuant matrix on the L.H.S. If the n equations are represented by

$$B.x = b,$$

then the assumption of a value for x_1, the first element of x, enables us to generate a solution x^1 to the equations

$$B.x^1 = b + e^1,$$

where e^1 is a vector with all elements zero except the last one, e^1_n.

Next we choose another value for x_1 and generate a solution to the equations

$$B.x^2 = e^2,$$

where e^2 is a vector with all elements zero except for the last one, e^2_n.

The solution of the equations $A.x = b$ is then

$$x^1 - (e_n^1/e_n^2)x^2,$$

for this will satisfy the original equations.

<u>Example.</u> If

$$B = \begin{bmatrix} 2 & 1 & & \\ 1 & 3 & 1 & \\ & 1 & 3 & 1 \\ & & 1 & 1 \end{bmatrix}, \qquad b = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

placing $x_1^1 = 1$ results in

$$x^1 = \begin{bmatrix} 1 \\ -1 \\ 2 \\ -5 \end{bmatrix}, \qquad e_n^1 = 8.$$

Similarly, placing $x_1^2 = 1$ results in

$$x^2 = \begin{bmatrix} 1 \\ -2 \\ 5 \\ -13 \end{bmatrix}, \qquad e_4^2 = -21.$$

Thus, $x = x^1 - (8/21)x^2$

$$= 1/21 \begin{bmatrix} 13 \\ -5 \\ 2 \\ 1 \end{bmatrix}.$$

If we are working with an automatic computer and storage space is at a premium, we need not store the whole of x^1 and x^2 : it suffices to store the first elements only. Then we have

$$x_1 = x_1^1 - (e_n^1/e_n^2)x_1^2,$$

and the remaining elements of x can be obtained from the equations.

The general recursive relationship employed to generate successive elements of x is:

$$B_{r,r-1}x_{r-1} + B_{rr}x_r + B_{r,r+1}x_{r+1} = b_r \quad .$$

(where $B_{0,1} = B_{n,n+1} = 0$)

i.e.,

$$x_{r+1} = (1/B_{r,r+1}) \left[b_r - B_{rr}x_r - B_{r,r-1}x_{r-1} \right].$$

Rounding off errors in the computation arise from two sources. One is that the coefficients B_{rs}, b_s may be rounded off. The second source is that a rounding off error is committed whenever the products $B_{rr}x_r$ and $B_{r,r-1}x_{r-1}$ are evaluated: this is usually outside the accuracy to which we are working. Garwick [2] shows how we can correct for the errors of the first type, and gives numerous examples of the techniques just described.

This technique [3] can be extended to the case in which there are several super-diagonals and sub-diagonals. In this case, e_n^1 is a vector corresponding to a starting vector x_1^1, and e_n^2 a matrix consisting of a number of vectors, each obtained with a different set of initial vectors which together form a matrix x_1^2. The solution is:

$$x_1 = x_1^1 - x_1^2 . e_n^{2I} . e_n^1 .$$

3. SOLUTION OF LINEAR SIMULTANEOUS EQUATIONS (II) †

The continuant

$$B = \begin{vmatrix} a_1 & b_1 & & & \\ 1 & a_2 & b_2 & & \\ & 1 & a_3 & \ddots & \\ & & \ddots & \ddots & b_{n-1} \\ & & & 1 & a_n \end{vmatrix}$$

can be evaluated with the use of recurrence relation:

$$\bar{q}_0 = 1, \quad \bar{q}_1 = a_1,$$

$$\vdots$$

$$\bar{q}_r = a_r \bar{q}_{r-1} - b_{r-1} \bar{q}_{r-2},$$

where $\bar{q}_r$ is the determinant of the matrix with diagonal elements

$a_1, \ldots, a_r$. Thus

$$B = \bar{q}_n .$$

Similarly, the co-factor of a_1 may be obtained from the recurrence relation:

$$\bar{p}_0 = 1, \quad \bar{p}_1 = a_2,$$

$$\bar{p}_r = a_{r+1}\bar{p}_{r-1} - b_r\bar{p}_{r-2} ,$$

and $\bar{p}_{n-1}$ is the co-factor of a_1.

Thus the leading term of B^I, the inverse of B, is $\bar{p}_{n-1}/\bar{q}_n$, and other elements of B^I can be obtained from the relation:

$$B^I . B = B . B^I = I \text{ (the unit matrix).}$$

If we wish, this technique can be extended to solve the equations:

$$B . x = b .$$

† The bar is used to distinguish $\bar{q}_r$ from q_r used later.

We need evaluate only the first row of B^I - say B^I_1 - whence the first element of x, x_1, is available as $B^I_1 .b$. Subsequent elements of x are then available by sustitution in the original equations.

<u>Example.</u> If we take the example in the previous section,

$$\bar{q}_n = 21, \bar{p}_{n-1} = 13,$$

whence $$B^I_{11} = 13/21$$

and as $B^I.B = I$, B^I_1 can be generated element by element.

It is:

$$1/21 \begin{bmatrix} 13 & -5 & 2 & 1 \end{bmatrix}$$

and

$$x_1(=B^I_1.b) \text{ is } 13/21.$$

(Of course, we need have computed only B^I_{11} in this example, as the first element of b is the only non-zero element.) The other elements of x are available by back substitution.

4. SOLUTION OF LINEAR SIMULTANEOUS EQUATIONS (III)

The matrix B can be expressed [4] as the product of an upper unit triangular matrix E and a lower triangular matrix Q, the former having non-zero elements on the diagonal and super-diagonal only, and the latter on the diagonal and sub-diagonal only. In fact, the sub-diagonal elements in the latter case can be chosen to be unity.

Suppose we call the super-diagonal terms of E

$$e_1, e_2, \dots e_{n-1}$$

and the diagonal terms of Q

$$q_1, q_2, \dots q_n .$$

As B = Q.E, we have

$$q_1 = a_1, \quad e_1 = b_1/q_1,$$

$$q_2 = a_2 - e_1, \quad e_2 = b_2/q_2,$$

$$\vdots$$

$$q_r = a_r - e_{r-1}, \quad e_r = b_r/q_r.$$

If

$$A.x = b, \text{ then}$$

$$Q.E.x = b,$$

$$E.x = Q^I.b = c \text{ and}$$

$$x = E^I.c .$$

c can be also generated element by element as

$$Q.c = b.$$

Thus (if $\bar{b}_i$ is chosen to represent an element of b, to differentiate it from the b_i which are super-diagonal elements of B),

$$c_1 = \bar{b}_1/q_1,$$

$$c_2 = (\bar{b}_2 - c_1)/q_2,$$

$$c_r = (\bar{b}_r - c_{r-1})/q_r.$$

As E is triangular, x can be obtained by back-substitution.
Thus:

$$x_n = c_n$$

$$x_{n-1} = c_{n-1} \; x_n e_{n-1}$$

$$\vdots$$

$$x_{r-1} = c_{r-1} \; x_r e_{r-2}.$$

<u>Example.</u> The example in Section 2 yields:

$$Q = \begin{bmatrix} 2 & & & \\ 1 & 5/2 & & \\ & 1 & 13/5 & \\ & & 1 & 21/13 \end{bmatrix}, \quad E = \begin{bmatrix} 1 & 1/2 & & \\ & 1 & 2/5 & \\ & & 1 & 5/13 \\ & & & 1 \end{bmatrix},$$

$$c = \begin{bmatrix} 1/2 \\ -1/5 \\ 1/13 \\ -1/21 \end{bmatrix},$$ whence x is obtained by back-substitution.

5. LATENT ROOTS AND VECTORS (I)

The latent roots of B can be computed by evaluating the determinant

$$\left| \lambda I - B \right|$$

for trial values of λ, using the recursive relations discussed in Section 3. If B is an nxn matrix, then the evaluation of the determinant for n+1 trial values of λ will define the polynomial expansion of the determinant (i.e. the characteristic equation) exactly. [5] Alternatively, if one specific root is required, some approach such as the rule of false position can be adopted to improve an approximation to it.

B may have several off-diagonal terms, say r, in which case it becomes necessary to assume r values $x_1, \ldots, x_r$ for each value of λ to get the process of Section 2 started. Similarly, there will be r equations unsatisfied at the end of the process. If r linearly independent sets of starting vectors are chosen, the matrix formed by the column vectors of the resulting errors must have zero determinant for a correct choice of λ [6]. The values of this determinant for trial λ's can be used in any of a number of approximation processes which will result in the choice of λ for which the determinant has zero value.

As

$$(\lambda I - B). \operatorname{Adj}(\lambda I - B)$$

$$= \operatorname{Adj}(\lambda I - B).(\lambda I - B) = \left| \lambda I - B \right| I,$$

where $Adj(\lambda I - B)$ is the adjoint matrix of $(\lambda I - B)$, for any latent root λ_s, any one of the rows of $Adj(\lambda_s I - B)$ is a latent row of $(\lambda I - B)$ and any one of the columns of $Adj(\lambda_s I - B)$ is a latent column.

The leading term in the principal diagonal of $Adj(\lambda_s I - B)$ (i.e. the co-factor of $(\lambda_s - a_1)$ in $(\lambda_s I - B)$) can be computed as described in Section 3, and other terms deduced from it by use of the relation

$$Adj(\lambda_s I-B).(\lambda_s I-B) = (\lambda_s I-B).Adj(\lambda_s I-B) = 0.$$

6. GENERATION OF POLYMONIALS ORTHOGONAL TO SUMMATION

Suppose A is a diagonal matrix such that $A_{ii} = \lambda_i$. We often wish to generate a set of polynomials which are orthogonal to summation. Thus, if these polynomials are $P_r(\lambda)$, then

$$\sum_{i=1}^{n} P_r(\lambda_i)\, P_s(\lambda_i)\, w_i^2 = 0 \quad (r \neq s),$$

where w_i^2 are the weighting factors. If we know $w_i, P_r(\lambda_i)$, for all r and i, then any function $f(\lambda)$ defined at the λ_i can be expressed with their help in the form of a polynomial of degree r, $F_r(\lambda)$, which is such that

$$\sum_{i=1}^{n} w_i^2 \left[f(\lambda_i) - F_r(\lambda_i) \right]^2$$

is a minimum for all polynomials of this degree [7]. See Section 7.

Suppose we seek

$$A.R. = R.B.$$

where

$$R_{i1} = w_i, \qquad B = \begin{bmatrix} a_1 & b_1 & & \\ 1 & a_2 & \ddots & \\ & \ddots & \ddots & b_{n-1} \\ & & 1 & a_n \end{bmatrix}$$

and $R^T.R$ is a diagonal matrix.

Thus, if columns R are $r^1, r^2, \ldots$

$$A.r^i = b_{i-1} r^{i-1} + a_i r^i + r^{i+1}$$

i.e. $$r^{i+1} = (A - a_i I) r^i - b_{i-1} r^{i-1}.$$

Moreover

$$a_i = r^{iT}.A.r^i / r^{iT}.r^i$$

$$\begin{aligned} b_{i-1} &= r^{(i-1)T}.A.r^i / r^{(i-1)T}.r^{(i-1)}, \\ &= r^{iT}.A.r^{(i-1)} / r^{(i-1)T}.r^{(i-1)T}, \\ &= r^{iT}.r^i / r^{(i-1)T}.r^{(i-1)}. \end{aligned}$$

Then

$$A.R = R.B,$$
$$R^T.A.R = \Sigma \; .B$$

where

$$R^T.R = \Sigma \; .$$

Thus

$$\Sigma^{-1}.R^T.A.R = B.$$

Thus B has the same latent roots as A, i.e. λ_i, and the polynomials

$$P_0(x) = 1$$
$$P_1(\lambda) = (\lambda - a_1)$$
$$P_2(\lambda) = (\lambda - a_2)P_1(\lambda) - b_1 \, P_0(\lambda)$$
$$P_r(\lambda) = (\lambda - a_r) \, P_{r-1}(\lambda) - b_{r-1} \, P_{r-1}(\lambda)$$
$$\vdots$$
$$P_n(\lambda) = (\lambda - a_n) \, P_{n-1}(\lambda) - b_{n-1} \, P_{n-1}(\lambda)$$
$$= 0$$

are obtained by expanding $\left|\lambda I - B\right|$, beginning at the top corner, as shown in Section 3.

Moreover, as the recurrence relation obeyed by the $P_r(\lambda)$ is the same as that obeyed by the r^r, we have

$$r_i^r = P_{r-1}(\lambda_i) \; w_i,$$

i.e.

$$r^{rT}.r^s = \sum_{i=1}^{n} w_i^2 \, P_{r-1}(\lambda_i) \, P_{s-1}(\lambda_i),$$

and the manner of generation of the r^r is such that $r^{rT}.r^s = 0$ if $r = s$. Thus the $P_r(\lambda)$ are orthogonal.

Example. Suppose we wish the polynomials which are orthogonal to summation with weights unity at points 0, 1, 3.

$$A = \begin{bmatrix} 0 & & \\ & 1 & \\ & & 3 \end{bmatrix}, \qquad r^1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix},$$

$$a_1 = 4/3, \qquad r^2 = \begin{bmatrix} -4/3 \\ -1/3 \\ 5/3 \end{bmatrix},$$

$$b_1 = 42/27, \qquad a_2 = 76/42,$$

$$r^3 = \begin{bmatrix} 6/7 \\ -9/7 \\ 3/7 \end{bmatrix}, \qquad a_3 = 6/7, \quad b_2 = 27/49.$$

I.e.,

$$\begin{bmatrix} 0 & & \\ & 1 & \\ & & 3 \end{bmatrix} \cdot \begin{bmatrix} 1 & -4/3 & 6/7 \\ 1 & -1/3 & -9/7 \\ 1 & 5/3 & 3/7 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & -4/3 & 6/7 \\ 1 & -1/3 & -9/7 \\ 1 & 5/3 & 3/7 \end{bmatrix} \cdot \begin{bmatrix} 4/3 & 42/27 & \\ 1 & 76/42 & 27/49 \\ & 1 & 6/7 \end{bmatrix} .$$

As a check,

$$r^4 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} .$$

Thus we have

$$P_0(\lambda) = 1, \; P_1(\lambda) = \lambda - 4/3, \; P_2(\lambda) = \lambda^2 - 22\lambda/7 + 6/7$$

$$P_3(\lambda) = \lambda^3 - 4\lambda^2 + 3\lambda.$$

7. CURVE FITTING

Suppose we wish to fit a polynomial of a given degree to a set of observational data $f(\lambda_i)$ at n points $\lambda_1, \ldots \lambda_n$ so that

$$\sum_{i=1}^{n} w_i^2 \left[f(\lambda_i) - \sum_{r=0}^{k} a_r x^r \right]^2$$

is a minimum. If $w_i f(\lambda_i)$ are arranged as a vector f, and we have a set of vectors r^i computed as described in the previous section, we can use the orthogonality property of the polynomials P_0, P_1 etc. to deduce a representation:

$$f(\lambda) = \sum_{i=0}^{k} a_i P_i(\lambda)$$

which can be converted back to the power series representation. Thus, as

$$r_i^r = P_{(r-1)}(\lambda_i) r_i^1,$$

$$r^{(r+1)T} . f = a_r \sum_{i=1}^{n} w_i^2 \; P_r^2(\lambda_i),$$

as

$$\sum_{i=1}^{n} w_i^2 \; P_r(\lambda_i) \; P_s(\lambda_i) = 0 \quad (r \neq s),$$

$$= a_r r^{(r+1)T} . r^{(r+1)}.$$

Thus

$$a_r = r^{(r+1)T} . f / r^{(r+1)T} . r^{(r+1)}.$$

Example. Suppose we wish the straight line of best fit to the data shown below, with weighting factors as shown.

λ_i	$w_i f(\lambda_i)$	w_i
0	3	1
1	2	1
3	0	1

The r^r were deduced in the example given in the last section. Thus

$$a_0 = 5/3,$$

$$a_1 = -1,$$

i.e. $$5/3\ P_0(\lambda) - P_1(\lambda) = 3 - \lambda,$$

which is, of course, an exact result.

8. LATENT ROOTS AND VECTORS

The approach of Section 6 may be used to obtain the latent roots and vectors of a matrix.[8] and [9]. For ease in discussion, suppose we assume that A is symmetric and without repeated roots, and U is a matrix such that

$$U^T.A.U = D, \qquad U^T.U = I,$$

where here D is a diagonal matrix such that

$$D_{ii} = \lambda_i.$$

Then

$$U^T.A.U.R. = R.B.$$

I.e., if

$$U.R = S,$$

then

$$A.S = S.B,$$

and the algorithm of the previous section holds with

$$s^i \ (=U\ .r^i)$$

substituted for r^i throughout. The assumption is made that s^1 is chosen so that it has components in the direction of all the latent vectors (i.e. all $w_i \neq 0$).

Example. If

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 3 & 6 \end{bmatrix}, \quad s^1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix},$$

the transformation is:

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 3 & 6 \end{bmatrix} . \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -2 \\ 0 & -1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -2 \\ 0 & 1 & 2 \end{bmatrix} . \begin{bmatrix} 1 & 2 & . \\ 1 & 7 & 4 \\ . & 1 & 1 \end{bmatrix} .$$

As

$$D.R = R.B,$$

the rows of R are the row vectors of B.

The row corresponding to λ_i, which we shall call v_i, is

$$\left[w_i \; P_0(\lambda_i), \; w_i \; P_1(\lambda_i), \; \ldots\ldots, \; w_i \; P_{n-1}(\lambda_i) \right]$$

and the w_i can be dropped. This is a latent row vector of B: in the co-ordinate system of A, this vector is

$$v_i . S^I.$$

where

$$S^I = \Sigma^I . S^T,$$

Σ being the diagonal matrix with

$$\Sigma_{ii} = s^{iT} . s^i .$$

The latent columns can be found by similar procedure. An alternative approach which depends on the expansion of the continuant from the bottom right hand corner, i.e., beginning at $(\lambda - a_n)$, is described by Lanczos 8 .

9. AN ALTERNATIVE TRANSFORMATION

The process just described gives rise to computational difficulties in practice if any b_i becomes small, which arises if two nearly equal roots are present, or if a starting vector has been chosen which is nearly deficient in a component in the direction of one of the eigenvectors (i.e. one or more of the w_i is very small). In order to circumvent this difficulty, and to minimise the effect of accumulated round-off error, one possibility is to "re-orthogonalise" successive s^i to ensure that they are orthogonal to previous s^j. Thus, if

$$s^{iT} . s^j = \epsilon_{ij}, \quad \text{then replacing } s^i \text{ by}$$

$$t^i = s^i - \epsilon_{ij} \; s^j$$

will produce a vector such that

$$t^{iT} . s^j = 0.$$

Moreover, for any other s^j, say s^k, the scalar product is unaltered. I.e.,

$$t^{iT} . s^k = s^{iT} . s^k$$

as

$$s^{jT} . s^k = 0.$$

This process requires storing all values of s^i, and considerably increases computing time. It also has the effect of introducing non-zero terms above the super-diagonal of B.

A transformation which appears to be free of this difficulty, and which is applicable to symmetric matrices, is the following.

If we seek S such that

$$A.S. = S.B,$$

let

$$S^T.S = \Sigma$$

Then

$$A.S.\Sigma^{-1/2} = S.\Sigma^{-1/2}\ \Sigma^{1/2}.B.\Sigma^{-1/2},$$

or if

$$S.\Sigma^{-1/2} = T\ (\text{i.e. } T^T.T = I),\quad \Sigma^{1/2}.B.\Sigma^{-1/2} = C,$$

then

$$A.T. = T.C.$$

The symmetric matrix C has the same latent roots as B and proves to be of the form:

$$\begin{bmatrix} a_1 & b_1^{1/2} & & \\ b_1^{1/2} & a_2 & \ddots & \\ & \ddots & \ddots & b_{n-1}^{1/2} \\ & & b_{n-1}^{1/2} & a_n \end{bmatrix}$$

Now consider the transformation which is the basis of the Jacobi process[15] for the reduction of a matrix to diagonal form:

$$T^{(1T}.A.T^{(1},$$

where

$$T^{(1}_{ii} = T^{(1}_{jj} = \cos\theta,$$

$$T^{(1}_{ji} = -T^{(1}_{ij} = \sin\theta,\quad T^{(1}_{kl} = 0(k\neq 1, \neq i, \neq j),$$

$$T^{(1}_{kk} = 1\ (k\neq i, \neq j).$$

Suppose we choose $i = 3$, $j = 2$ initially. Then

$$(T^{(1T}.A.T^{(1})_{31} = A_{31}\cos\theta - A_{21}\sin\theta,$$

and, if we choose

$$\operatorname{Tan}\theta = A_{31}/A_{21},$$

$$(T^{(1T}.A.T^{(1})_{31} = 0.$$

Symmetry will be preserved and elements in the second and third rows and columns only will be affected. If we next choose $i = 4$, $j = 2$ to reduce element (4,1) to zero, element (3,1) will remain unaffected. Thus we can reduce to zero in turn elements (3,1), (4,1),...,(n,1); (4,2), ...,(n,2); (5,3), etc. (and terms symmetrically placed with respect to the diagonal). If successive reductions are $T^{(1}$, $T^{(2}$ etc., there will be (n-1)(n-1)/2 of these in all, the resulting matrix will be in the form C and

$$T^{(1}.T^{(2}.T^{(3}. \ldots\ldots .T^{(n-1)(n-2)/2)} = T.$$

The method appears to have all the advantages of stability which are possessed by the Jacobi process, and many of the computational devices used in programming the usual Jacobi process can also be used here. Even in the case of repeated roots, this process does not lead to loss of accuracy, although in this case some $b_i^{1/2}$ will vanish.

As in this process A_{11} is not affected, the values of a_i and b_i produced will be the same as if we had carried out the process first described with

$$s_i^1 = 1 \quad (i=1)$$
$$= 0 \quad (i \neq 1).$$

Example. In the example of the previous section, there is one transformation only. This is given by:

$$\text{Tan } \theta = 1, \quad \text{i.e. } \cos\theta = \sin\theta = 2^{-1/2}.$$

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2^{-1/2} & -2^{-1/2} \\ 0 & +2^{-1/2} & 2^{-1/2} \end{bmatrix}$$

$$\text{and } T^T.A.T = \begin{bmatrix} 1 & 2^{1/2} & . \\ 2^{1/2} & 7 & 2 \\ . & 2 & 1 \end{bmatrix}$$

which checks with the previous results.

10. AN ALTERNATIVE TRANSFORMATION (II)

We seek S such that

$$A.S = S.B.$$

If we write B = Q.E, Q and E being defined as in Section 4, we have:

$$A.S = S.Q.E,$$

i.e.

$$A.S.Q. = S.Q.E.Q.$$

If we write

$$A.Z. = S.Q. \quad \text{i.e., } S = Z.E.,$$

we have

$$A.Z. = Z.E.Q\ ,$$

i.e.

$$A.Z. = S.Q.$$

This gives rise to the algorithm:

$$A.z^r = q_r s^r + s^{r+1},$$

i.e.

$$s^{r+1} = A.z^r - q_r s^r,$$

and

$$s^{r+1} = e_r z^r + z^{r+1},$$

i.e.

$$z^{r+1} = s^{r+1} - e_r z^r.$$

Initially, $e_0 = 0$ and $z^1 = s^1$.

The scalars e_r and q_r can be deduced from the fact that $Z^T.A.Z$ is diagonal. This may be proved as follows.

$$Z^T.A.Z = Z^T.S.Q$$

$$= E^{IT}.S^T.S.Q.$$

If we wish to show that $Z^T.A.Z$ is diagonal, it suffices to show that $S^T.S.Q$ has non-zero elements in the same positions as E^T. As $S^T.S(= \Sigma\ )$ is diagonal, this is clearly so. (As a corollary to this, $S^T.A.Z$ is a matrix with non-zero elements only on and immediately below its diagonal. Thus

$$s^{rT}.A.s^s = 0 \quad (r \neq s,\ s-1).)$$

If we use these relations, we see that

$$q_r = s^{rT}.A.z^r / s^{rT}.s^r$$

and

$$e_r = z^{rT}.A.s^{r+1} / z^{rT}.A.z^r.$$

The order of computation is as follows. Given $s^1 (= z^1)$, we compute in turn q_1, s^2, e_1, z^2, q_2 etc.

Successive vectors are given by

$$s^{r+1} = P_r(A).s^1$$

and

$$z^{r+1} = Q_r(A).s^1,$$

where $P_r(\lambda)$ is orthogonal to summation over the λ_i with weighting factors w_i^2, and $Q_r(\lambda)$ is orthogonal to summation over the λ_i with weighting factor $\lambda_i\ w_i^2$. The polynomials $P_r(\lambda)$ and $Q_r(\lambda)$ can be obtained with the same recursion formula as s^r and z^r.

Example. If we take

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 3 & 6 \end{bmatrix}, \quad s^1 = z^1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix},$$

$$q_1 = 1, s^2 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \quad e_1 = 2,$$

$$z^2 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} - 2 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -2 \\ 1 \\ 1 \end{bmatrix}, \quad q_2 = 5,$$

$$s^3 = \begin{bmatrix} 0 \\ 3 \\ 7 \end{bmatrix} - 5 \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -2 \\ 2 \end{bmatrix}, \quad e_2 = 4/5,$$

$$z^3 = \begin{bmatrix} 0 \\ -2 \\ 2 \end{bmatrix} -4/5 \begin{bmatrix} -2 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 8/5 \\ -14/5 \\ 6/5 \end{bmatrix}.$$

$q_3 = 1/5.$

Thus

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 3 & 6 \end{bmatrix} . \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -2 \\ 0 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -2 \\ 0 & 1 & 2 \end{bmatrix} . \begin{bmatrix} 1 & . & . \\ 1 & 5 & . \\ & 1 & 1/5 \end{bmatrix} . \begin{bmatrix} 1 & 2 & . \\ & 1 & 4/5. \\ & & 1 \end{bmatrix}$$

11. AN ITERATIVE ALGORITHM FOR LATENT ROOTS AND VECTORS[10]

If we write the S, B, Q, E obtained as just shown with a superscript $^{(0}$ to indicate that they are the values of these matrices at the first iteration, we have:

$$A.S^{(0} = S^{(0}.Q^{(0}.E^{(0}, \quad \text{i.e.}$$

$$A.Z^{(1} = Z^{(1}.E^{(0}.Q^{(0} \quad \text{where}$$

$$A.Z^{(1} = S^{(0}.Q^{(0}.$$

If we now choose $Q^{(1}$, $E^{(1}$, of the same form as $Q^{(0}, E^{(0}$, such that

$$Q^{(1}.E^{(1} = E^{(0}.Q^{(0},$$

we have

$$A.Z^{(1} = Z^{(1}.Q^{(1}.E^{(1}.$$

If we write $Q^{(1}.E^{(1} = B^{(1}$, and recall that $Z.^{(1T}A.Z^{(1}$ is diagonal, we have

$$A.^{1/2}A.Z^{(1} = A.^{1/2}Z^{(1}.B^{(1}.$$

On writing $A^{1/2}.Z^{(1}$ as $S^{(1}$ we have

$$A.S^{(1} = S^{(1}.B^{(1},$$

an equation which differs essentially from that connecting the initial iterates only in the value of the starting vector $s^{1(1}$. This leads to polynomials which are now orthogonal to summation with weighting factors $w_i^2\lambda_i$.

If

$$|\lambda_1| > |\lambda_2| > |\lambda_3| \text{ etc.,}$$

the first weighting factor will swamp the others after a number of iterations, and $b_1^{(n}$, hence $e_1^{(n}$, will converge to zero, and $a_1^{(n}$, hence $q_1^{(n}$, will converge to λ_1. By an extension of this reasoning, we can see that eventually all $e_r^{(n}$ will converge to zero, and all $a_r^{(n}$ will converge to the λ_r. The columns of $S^{(n}$ will converge to the latent vectors.

The basic algorithm is the one which connects $Q^{(r}$, $E^{(r}$, $Q^{(r+1}$ and $E^{(r+1}$. The equation

$$E^{(r}.Q^{(r} = Q^{(r+1}.E^{(r+1}$$

gives in detail:

$$q_i^{(r} + e_i^{(r} = q_i^{(r+1} + e_{i-1}^{(r+1},$$

i.e.

$$q_i^{(r+1} = q_i^{(r} + e_i^{(r} - e_{i-1}^{(r+1}$$

with

$$e_0^{(r+1} = 0 \text{ and } e_n^{(r+1} = 0, \text{ and}$$

$$q_{i+1}^{(r} \; e_i^{(r} = q_i^{(r+1} \; e_i^{(r+1},$$

i.e.

$$e_i^{(r+1} = q_{i+1}^{(r} \; e_i^{(r}/q_i^{(r+1}.$$

Thus we compute in turn $q_1^{(r+1}$, $e_1^{(r+1}$, $q_2^{(r+1}$, $e_2^{(r+1}$ etc.

A modification of this process can be used to reconstruct a polynomial from a knowledge of its moments.

Example. If we apply this to the example of the previous section, we obtain for the first five iterates:

n =	0	1	2	3	4
q_1 =	1	3	6.3333	7.6315	7.8414
e_1 =	2	3.3333	1.2982	0.2098	0.0276
q_2 =	5	2.4667	1.2333	1.0306	1.0039
e_2 =	0.8	0.0649	0.0071	0.0009	0.0001
q_3 =	0.2	0.1351	0.1280	0.1271	0.1270

Successive vectors $S^{(2r}$ may be obtained as follows.

As $S^{(2r+1} = Z^{(2r+2}.\ E^{(2r+1}$ (See Section 10),

$$A^{1/2}.Z^{(2r+2}.E^{(2r+1}$$

$$= S^{(2r+2}.E^{(2r+1} \quad \text{and}$$

$$A^{1/2}.S^{(r+1} = A.Z^{(r+1} = S^{(r}.Q^{(r}.$$

Thus

$$S^{(2r+2}.E^{(2r+1} = S^{(r}.Q^{(r},$$

which leads to a recursive relation for the $S^{(2r+2}$.

Rutishuiser[10] discusses means of speeding up convergence, and of obtain- the roots of largest magnitude only without having to compute all $e_i^{(r}$ and $q_i^{(r}$.

12. OBTAINING THE ROOTS OF $P_n(\lambda)$

If A is symmetric, the $P_r(\lambda)$ form a Sturm sequence[11], and so the difference between the number of changes of sign of the sequence $P_r(\lambda)$ when evaluated for two λ_i will give the number of roots between these two λ_i.

Moreover, as the $P_r(\lambda)$ are orthogonal polynomials [7]*, of all polynomials of the same degree and with unity as the coefficient of the leading term, the $P_r(\lambda)$ minimise

$$\sum_{i=1}^{n} w_r^2 \ P_r^2(\lambda_i).$$

As a consequence of this, as r increases, the roots of the $P_r(\lambda)$ approximate most rapidly to the λ_i with greatest weights. In fact, if we are interested in the greatest λ_i only, it is worth our while to pre-multiply

* p.160.

an arbitrary vector by A several times before we start: this increases the w_i associated with the largest λ_i.

If [12] we have an approximation λ_k to a root of $P_r(\lambda)$, we may wish to use Newton's approximation

$$\lambda_{k+1} = \lambda_k - P_m(\lambda_k)/P_m'(\lambda_k)$$

to obtain a better approximation, λ_{k+1}.

$P_r'(\lambda)$ and $Q_r'(\lambda)$ may be obtained at the same time as $P_r(\lambda)$ and $Q_r(\lambda)$ by differentiating the algorithm given in Section 10.

Thus

$$P'_{r+1}(\lambda_k) = \lambda Q'_r(\lambda_k) + Q_r(\lambda_k) - q_{r+1} P'_r(\lambda_k)$$

and

$$Q'_{r+1}(\lambda_k) = P'_{r+1}(\lambda_k) - e_{r+1} Q'_r(\lambda_k).$$

13. GENERATION OF ORTHOGONAL POLYNOMIALS [12]

It has been shown [7]* that orthogonal polynomials $P_r(\lambda)$ which are such that

$$\int_a^b P_r(\lambda)\, P_s(\lambda)\, m(x)\, dx = 0 \quad (r \neq s)$$

and which are normalised so that the coefficient of the term of highest degree in λ is unity, are connected by the relation:

$$P_{r+1}(\lambda) = (\lambda - a_{r+1})\, P_r(\lambda) - b_r\, P_{r-1}(\lambda).$$

If we generate at the same time a set of polynomials $Q_r(\lambda)$ such that

$$\int_a^b \lambda Q_r(\lambda)\; Q_s(\lambda)\; m(\lambda)\; d\lambda = 0 \quad (r \neq s),$$

we can obtain a very simple algorithm which is based on analogy with the algorithm for generating s^r and z^r of Section 10. This is:

$$\lambda Q_r(\lambda) = q_{r+1} P_r(\lambda) + P_{r+1}(\lambda),$$

giving $P_{r+1}(\lambda)$ and

$$P_{r+1}(\lambda) = e_{r+1}\;\; Q_r(\lambda) \qquad Q_{r+1}(\lambda),$$

giving $Q_{r+1}(\lambda)$.

Here

$$q_{r+1} = \int_a^b \lambda P_r(\lambda) Q_r(\lambda) m(\lambda) d\lambda \Big/ \int_a^b P^2(\lambda) m(\lambda) d\lambda$$

* p.158

and,

$$e_{r+1} = \int_a^b \lambda Q_r(\lambda) P_{r+1}(\lambda) m(\lambda) d\lambda \Big/ \int_a^b \lambda Q_r^2 m(\lambda) d\lambda$$

with

$$P_0(\lambda) = Q_0(\lambda) = 1.$$

<u>Example.</u> Considers the polynomials obtained with

$$m(\lambda) = \lambda.$$

and limits of integration 0, 1.

$$P_0 = Q_0 = 1$$

$$q_1 = 2/3, \quad P_1 = \lambda - 2/3,$$

$$e_1 = 1/12, \quad Q_1 = \lambda - 3/4,$$

$$q_2 = 9/20, \quad P_2 = \lambda^2 - 6\lambda/5 + 3/10.$$

The roots of P_2 are

$$\lambda_1, \lambda_2 = 3/5 \pm 6/10,$$

i.e.

$$\lambda_1 = 0.3551$$
$$\lambda_2 = 0.8449$$

$$Q.E = B = \begin{bmatrix} 2/3 & 1/18 \\ 1 & 8/15 \end{bmatrix}.$$

I.e.,

$$C = \begin{bmatrix} 2/3 & 1/3\sqrt{2} \\ 1/3\sqrt{2} & 8/15 \end{bmatrix}.$$

The latent row vectors of C may be written down (if we use the device shown in Section 8) to give the relation:

$$\begin{bmatrix} 3/5 - \sqrt{6}/10 & . \\ . & 3/5 + \sqrt{6}/10 \end{bmatrix} . \begin{bmatrix} 1 & (-1/15 - \sqrt{6}/10) \\ 1 & (-1/15 + \sqrt{6}/10) \end{bmatrix}$$

$$= \begin{bmatrix} 1 & (-1/15 - \sqrt{6}/10) \\ 1 & (-1/15 + \sqrt{6}/10) \end{bmatrix} . \begin{bmatrix} 2/3 & 1/18 \\ 1 & 8/15 \end{bmatrix}.$$

If we wish to state this in the form

$$A.R = R.B \text{ where } R^T R \text{ is diagonal,}$$

we must multiply these vectors by constants w_1 and w_2. This yields

$$w_1^2 + w_2^2 = k$$ (if we wish the first vector to be of length $k^{1/2}$)

and

$$(1/15 - \sqrt{6}/10) w_1^2 + (-1/15 + \sqrt{6}/10) w_2^2 = 0.$$

I.e.,

$$w_1^2 = (-1/3\sqrt{6} + 1/2)k$$

$$w_2^2 = (1/3\sqrt{6} + 1/2)k.$$

14. GAUSSIAN INTEGRATION

If we wish to evaluate the integral

$$\int_a^b F(\lambda)\, m(\lambda)\, d\lambda$$

by replacing it by a sum

$$\sum_{i=1}^{n} k_i F(\lambda_i),$$

and if the k_i and λ_i are appropriately chosen, the relationship may be made exact if $F(\lambda)$ is a polynomial of degree 2n-1.

The λ_i turn out to be the roots of the polynomial $P_n(\lambda)$ which is one of a series of polynomials generated so that

$$\int_a^b P_i(\lambda)\, P_j(\lambda)\, m(\lambda) = 0, \quad (i \neq j)$$

and the constants k_i are the w_i^2 of the last section. k is chosen so that a correct value is given to

$$\int_a^b m(\lambda)\, d\lambda.$$

The use of the properties of continuants affords a simple approach to establishing many of the properties associated with the k_i and λ_i 14*.

<u>Example.</u> If we continue the example of the last section, we have evaluated w_1^2, w_2^2, λ_1, λ_2. k is shown to be 1/2 by placing F(x) = 1; i.e.

$$\int_0^1 \lambda \quad dx = k.$$

Thus, if

$$F(x) = x^3,$$

* ch. 7.

$$\int_0^1 xF(x)\,dx = 1/2\,(-1/3\sqrt{6} + 1/2)(3/5 - \sqrt{6}/10)^3$$

$$+ (+1^1/3\ \sqrt{6} + 1/2)(3/5 + \sqrt{6}/10)^3$$

$$= 1/5,$$

the exact result.

15. SOLUTION OF LINEAR SIMULTANEOUS EQUATIONS (IV)[9] AND [12].

Another property[7] * of orthogonal polynomials may be used to set up a convenient algorithm for solving sets of simultaneous linear equations. A number of variants of this method have been proposed, one of which will be described here.

In the form in which we shall use it, this property is the following **. The polynomial $R_r(\lambda)$ of degree n which has a value unity at $\lambda = 0$ and for which

$$\sum_{i=1}^{n} R_r^2(\lambda_i) w_i^2$$

is a minimum, is the polynomial $Q_r(\lambda)$ scaled so that $Q_r(0) = 1$. For this, it will be recalled,

$$\sum_{i=1}^{n} Q_r(\lambda_i)\, Q_s(\lambda_i)\, \lambda_i\, w_i^2 = 0.$$

Suppose for the moment that our matrix is symmetric (i.e. w_i^2 is positive, and the minimum property is valid). Then, suppose we are solving the equation

$$A.x = b$$

and we have a sequence of vectors x^i approximating to x. Let

$$b - A.x^i = r^i,$$

and the components of r^1 in the direction of the normal co-ordinates be w_1, w_2 etc.

Then, if $\overline{Q}_s(\lambda)$ is $Q_s(\lambda)$ normalised so that $\overline{Q}_s(0)$ is unity, the vectors

$$r^i = \overline{Q}_{i-1}(A)\, r^1$$

have the property that

$$r^{1T} . r^i$$

---

*p. 160

**There is unlikely to be confusion between the r^i of this section and that of previous sections.

is less than it would be if any other polynomial of the same degree with unity as the constant term, were employed.

Any polynomial of degree r with unity as the constant term, say $R_r(A)$, can be expressed as:

$$R_r(A) = (I - k_r A)(I - k_{r-1} A)(\ldots\ldots)(I - k_1 A)$$

where k_s is the reciprocal of λ_s, a root of $R_r(\lambda)$.

Any one of the operations

$$r^{i+1} = (I - k_i A).r^i$$

represents an adjustment of the form (putting $\Delta r^i = r^{i+1} - r^i$):

$$\Delta r^i = - k_i A.r^i,$$

i.e., if we substitute for x^i and place

$$\Delta x^i = x^{i+1} - x^i \text{ we have}$$

$$\Delta x^i = k_i r^i.$$

Thus the r^i generated with the help of the polynomials $\bar{Q}_i(A)$ produce a smaller value of $r^{iT}.r^i$ at <u>any</u> stage than <u>any</u> other succession of iterations of the type

$$\Delta x^i = k_i r^i.$$

The previous approach must be modified because it is capable of producing only unit coefficients for the leading term of $\bar{Q}_{i_1}(\lambda)$. We therefore require a different algorithm to generate the $\bar{Q}_i(A).r^1$.

In fact, we are seeking to replace A by a suitably chosen matrix $B.G^{I}$ where G is a diagonal matrix of elements g_1, g_2 .... and B is the continuant matrix

$$\begin{bmatrix} a_1 & b_1 & & & \\ 1 & a_2 & b_2 & & \\ & 1 & \ddots & \ddots & \\ & & \ddots & \ddots & \ddots \\ & & & 1 & a_n \end{bmatrix}$$

The elements a_r, b_r, c_r and g_r are chosen such that the $\bar{Q}_i(\lambda)$ are given by successive expansion of the determinant:

$$\left| -B + \lambda G \right|$$

i.e.

$$\bar{Q}_0(\lambda) = 1$$

$$\bar{Q}_1(\lambda) = (g_1 \lambda - a_1)\bar{Q}_0(\lambda)$$

$$\bar{Q}_2(\lambda) = (g_2 \lambda - a_2)\bar{Q}_1(\lambda) - b_1 \bar{Q}_0(\lambda)$$

and, in general

$$\bar{Q}_r = (g_r\lambda - a_r)\bar{Q}_{r-1}(\lambda) - b_{r-1}\,\bar{Q}_{r-2}(\lambda).$$

Because

$$\bar{Q}_r(0) = 1,$$

$$a_r = -b_{r-1} - 1.$$

Expanding the relation

$$A.R = R.B.G^I,$$

we have:

$$A.r^i = \frac{b_{i-1}}{g_i}\; r^{i-1} + \frac{a_r}{g_i}\, r^i + 1/g_i\; r^{i+1},$$

i.e.

$$A.r^i = \frac{-b_{i-1}}{g_i}\;(r^i - r^{i-1}) + 1/g_i(r^{i+1} - r^i).$$

Thus, if we write:

$$\Delta r^i = r^{i+1} - r^i,$$

we have

$$\Delta r^i = b_{i-1}\Delta\; r^{i-1} + g_i\; A.r^i.$$

We must evaluate b_i, g_i. These are available by premultiplying the recurrence relation for $A.r^i$ by $(A\,r^i)^T$ and $(A.r^{i+1})^T$ and that for $A.r^{i+1}$ by $(A.r^i)^T$. If we use the property that

$r^{iT}.A.r^j = 0(i \neq j)$ (as $\bar{Q}_i(\lambda)$ is orthogonal to summation over the λ_i with weighting factors $\lambda_i\; w_i^2$) we obtain

$$(A.r^i)^T.(A.r^i) = \left[\frac{-b_{r-1} - 1}{g_i}\right](A.r^i)^T\,.r^i,$$

$$(A.r^{i+1})^T.(A.r^i) = 1/g_i(A.r^{i+1})^T.r^{i+1}$$

$$= (A.r^i)^T.(A.r^{i+1}) = b_i/g_{i+1}(A.r^i)^T.r$$

i.e.

$$b_{i-1}/g_i + 1/g_i = -\frac{(A.r^i)^T.(A.r^i)}{(A.r^i)^T.r^i}$$

and

$$\frac{b_i\,g_i}{g_{i+1}} = \frac{(A.r^{i+1})^T.r^{i+1}}{(A.r^i)^T.r^{i+1}}$$

The order of computation is:

$$r^1 \rightarrow g_i \rightarrow \Delta r^1 \rightarrow r^2 \rightarrow b_1/g_2 \rightarrow g_2 \rightarrow b_1 \rightarrow \Delta r^2 \quad \text{etc.}$$

At each step, $r^{iT}.r^i$ decreases, and gives a measure of the error.

It remains to obtain the x^i. If $\Delta x^i = x^{i+1} - x^i$, we have:

$$r^{i+1} = k - A.x^{i+1} = k - A.x^i - A.\Delta x^i$$
$$= r^i - A.\Delta x^i,$$

i.e.
$$\Delta r^i = -A.\Delta x^i,$$

i.e.
$$x^i = b_{i-1}\, x^{i-1} - g_i r^i.$$

Example. To clarify the process, this example is worked in normal co-ordinates. We could, of course, assume A to be replaced by $U^T.A.U$ where $U^T.U = I$, and b by $U^T.b$: in this case, x would be replaced by $U^T.x$.

$$A = \begin{bmatrix} -1 & & \\ & 1 & \\ & & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

$$r^1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad g_1 = 1/3, \Delta r^1 = \begin{bmatrix} 1/3 \\ -1/3 \\ -2/3 \end{bmatrix},$$

$$r^2 = \begin{bmatrix} 4/3 \\ 2/3 \\ 1/3 \end{bmatrix}, \quad b_1/g_2 = 5/3,\ g_2 = 15/11,\ b_1 = 75/33,$$

$$\Delta r^2 = \begin{bmatrix} -35/33 \\ 5/33 \\ -20/33 \end{bmatrix}, \quad r^3 = \begin{bmatrix} 3/11 \\ 9/11 \\ -3/11 \end{bmatrix}, \quad b_2/g_3 = -27/55,$$

$$g_3 = -11/10, \quad b_2 = 27/50,$$

$$\Delta r^3 = \begin{bmatrix} -3/11 \\ -9/11 \\ 3/11 \end{bmatrix}, \quad r^4 \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

$$\Delta x^1 = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}, \quad x^2 = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix},$$

$$\Delta x^2 = \begin{bmatrix} -35/33 \\ -5/33 \\ 10/33 \end{bmatrix}, \quad x^3 \begin{bmatrix} -8/11 \\ 2/11 \\ 7/11 \end{bmatrix},$$

$$\Delta x^3 = \begin{bmatrix} -3/11 \\ 9/11 \\ -3/22 \end{bmatrix}, \quad x^4 \begin{bmatrix} -1 \\ 1 \\ 1/2 \end{bmatrix}.$$

After a time, with larger matrices, because of rounding-off errors, $r^{iT}.A.r^j$ $(i \neq j)$ ceases to be zero. Rounding-off errors could be treated in the manner described in Section 9. On the other hand, we can accept the errors and either begin the algorithm again from a new r^1 every half-dozen or so iterations, or we can continue beyond r^{n+1}, which should be zero if no rounding errors have been incurred.

Another possibility recommended in the literature for positive definite matrices, is to use an approximation process which has the effect of removing r^1 components in the direction of latent vectors associated with all but the group of latent roots of smallest magnitude. These reduce all but a few -say k - of the w_i to zero. After this, k applications of the algorithm will reduce $r^{iT}.r^i$ to zero.

The advantages of this type of technique for solving linear equations, known as the technique of "Minimised Iteration", are that:

(i) it is a direct technique (i.e. were it not for rounding off errors, it would terminate after at most n steps;

(ii) it gives a reducing "least squared residue" (or some equivalent measure in the case of variants of the process described here) at each step; and

(iii) unlike other direct methods, it preserves the original matrix unaltered. This last feature enables economies in storage to be made as a result of the presence of a high percentage of zero elements in the matrix A, a frequent occurrence with matrices arising from ordinary and partial differential equations.

16. CONTINUED FRACTIONS

Continuants derive their name from a connection with continued fractions ', which will now be described.

The continued fraction '

$$1/a_1 - b_1/a_2 - b_2/a_3 - \ldots - b_{n-1}/a_n$$

may be expanded by taking successive convergents to the numerator, $\bar{p}_r$, and the denominator, $\bar{q}_r$. The recurrence relations which $\bar{p}_r$ and $\bar{q}_r$ obey are identical with those described in Section 3.

Thus, if the a_r and b_r correspond to the scalors in a continuant matrix, successive convergents to the numerator and to the denominator will correspond to the $\bar{p}_r$ and $\bar{q}_r$, which converge to cofactor of a_1 & B respectively. The final value of the continued fraction is:

$\bar{p}_{n-1}/\bar{q}_n$.

In practice, this is one of the most convenient methods of computing the value of a continued fraction.

The determinant $|\lambda I - B|$, which gives rise to the polynomials $P_r(\lambda)$ as successive terms in its expansion, then corresponds to the denominator of

$$F(\lambda) = \frac{1}{\lambda - a_1} - \frac{b_1}{\lambda - a_2} - \cdots - \frac{b_{n-1}}{\lambda - a_n}$$

$P_n(\lambda)$ is thus the final denominator.

Similarly, if $S_r(\lambda)$ are successive terms in the expansion of the cofactor of $(\lambda - a_1)$, we have:

$$F(\lambda) = S_{n-1}(\lambda)/P_n(\lambda).$$

Here the coefficients of λ^{n-1} in $S_{n-1}(\lambda)$ and λ^n in $P_n(\lambda)$ are unity.

If $F(\lambda)$ is expressed as:

$$\frac{\alpha_1^2}{\lambda-\lambda_1} + \frac{\alpha_2^2}{\lambda-\lambda_2} + \cdots + \frac{\alpha_n^2}{\lambda-\lambda_n},$$

then, as

$$P_n(\lambda) = \prod_{i=1}^{n} (\lambda-\lambda_i),$$

we have:

$$S_{n-1}(\lambda) = \sum_{i=1}^{n} \alpha_i^2 \prod_{j \neq i} (\lambda - \lambda_j).$$

This can be expressed as:

$$S_{n-1}(\lambda) = -\begin{vmatrix} \lambda-\lambda_1 & & & & \alpha_1 \\ & \lambda-\lambda_2 & & & \alpha_2 \\ & & \ddots & & \vdots \\ & & & \lambda-\lambda_n & \alpha_n \\ \alpha_1 & \alpha_2 & \cdots & \alpha_n & 0 \end{vmatrix}$$

and, as this is the cofactor of $(\lambda - a_1)$ in $|\lambda I - B|$, it turns out that the α_i are the w_i of our previous sections.

This correspondence is of use if we wish to convert a ratio of two polynomials of degree n-1 and n respectively to continued fraction form for convenience of computation.

<u>Example.</u> Suppose we wish to express

$$F(\lambda) = \frac{\lambda^2 - 11\lambda/3 + 3}{\lambda^3 - 6\lambda^2 + 11\lambda - 6}$$

as a continued fraction. $F(\lambda)$ may be expressed as:

$$\frac{1/6}{\lambda-1} + \frac{1/3}{\lambda-2} + \frac{1/2}{\lambda-3} \quad .$$

We wish B such that

$$A.S = S.B$$

where

$$s_1 = 1/6^{1/2} \begin{bmatrix} 1 \\ 2^{1/2} \\ 3^{1/2} \end{bmatrix} \quad .$$

Using the algorithm of Section 8, we have:

$$1/6^{1/2} \begin{bmatrix} 1 & & \\ & 2 & \\ & & 3 \end{bmatrix} \cdot \begin{bmatrix} 1 & -4/3 & 3/5 \\ 2^{1/2} & -2^{1/2}.1/3 & -2^{1/2}.3/5 \\ 3^{1/2} & 3^{1/2}.2/3 & 3^{1/2}.1/5 \end{bmatrix}$$

$$= 1/6^{1/2} \begin{bmatrix} 1 & -4/3 & 3/5 \\ 2^{1/2} & -2^{1/2}.1/3 & -2^{1/2}.3/5 \\ 3^{1/2} & 3^{1/2}.2/3 & 3^{1/2}.1/5 \end{bmatrix} \cdot \begin{bmatrix} 7/3 & 5/9 & \\ 1 & 28/15 & 9/25 \\ & 1 & 9/5 \end{bmatrix} ,$$

Thus

$$F(\lambda) = \frac{1}{\lambda - 7/3 \; - } \; \frac{5/9}{\lambda - 28/15 \; - } \; \frac{9/25}{\lambda - 9/5} \quad .$$

This may be expressed as:

$$- 1/6 \; \frac{\begin{vmatrix} \lambda-1 & & & 1 \\ & \lambda-2 & & 2^{1/2} \\ & & \lambda-3 & 3^{1/2} \\ 1 & 2^{1/2} & 3^{1/2} & 0 \end{vmatrix}}{P_3(\lambda)} \quad \text{or} \quad \frac{\begin{vmatrix} \lambda - 28/15 & -9/25 \\ -1 & \lambda - 9/5 \end{vmatrix}}{P_3(\lambda)} \quad .$$

17. MORE GENERAL MATRICES

The fact that many of the properties of orthogonal polynomials rely on positive weight factors restricts the use of some of the devices given here. However, we can always obtain a U and a V such that

$$A.U = U.B,$$

$$A^T.V = V.B,$$

and

$$v^{iT}.u^j = 0, \; i \neq j.$$

If we follow through similar algebra to that given in Section 6, we have

$$x^{i+1} = A.x^1 - a_i x^i - b_{i-1} x^{i-1} \quad ,$$

$$y^{i+1} = A^T.y^i - a_i y^i - b_{i-1} y^{i-1} \; ,$$

$$b_0 = 0, \; b_{i-1} = \frac{y^{iT}.x^i}{y^{(i-1)T}.x^{i-1}} \quad ,$$

$$a_i = \frac{y^{iT}.A.x^i}{y^{iT}.x^i} \quad .$$

With this as a basis, many of the techniques described in the preceding sections can be altered to cater for non-symmetric matrices.

We can always convert the equations

$$A.x = b,$$

to a positive definite set by replacing them with the set:

$$A^T.A.x = A^T.b.$$

It will not be necessary to carry out the matrix multiplication explicitly: it is, for example, merely necessary to replace r^i by $A^T.r^i$ in the algorithms of Section 15.

The operation $A^T.A$ has been shown to worsen the conditioning of a set of equations[16]. It is possible[17] to avoid this difficulty if we obtain the solutions to

$$A.x^i = b \text{ with residues } r^i \text{ and}$$

$$A^T.y^i = b \text{ with residues } s^i$$

simultaneously, extending, for example, the algorithm given in the previous section. If A has complex eigenvalues, w_i^2 will no longer be positive, and so we can no longer expect $s^{iT}.r^i$ to decrease at each step.

When two or more roots coincide*, the effect is that the processes described in Section 8 will terminate before the n-th step: this is because the minimum polynomial is of lower degree than the characteristic polynomial. If it is desired to complete the transformation to obtain

---

* This remark does not include the case of non-linear divisors, which must be handled by replacing the diagonal D of Section 8 by the Jacobi canonical form.

$$A.R = R.B,$$

another starting vector, orthogonal to all preceding ones, should be chosen. The resulting B for a 5x5 matrix with two repeated roots, for example, would appear as:

$$\begin{bmatrix} a_1 & b_1 & & & \\ 1 & a_2 & b_2 & & \\ & 1 & a_3 & 0 & \\ & & 0 & a_4 & b_4 \\ & & & 1 & a_5 \end{bmatrix} .$$

With near-equality, small b_i will result, and it is best to obtain C by the use of the Jacobi transformation described in Section 9. Otherwise it becomes necessary to re-orthogonalise vectors r^j obtained subsequent to the computation of the small b_i.

18. SPECIAL FORMS

Matrices arising from ordinary and partial differential equations usually conform in part if not in toto to a regular pattern which gives rise to some convenient properties. Typical of these is the pattern arising in solution of the equation

$$\frac{d^2y}{dx^2} = f(x).$$

If this equation is written in finite difference form, it gives rise to a continuant matrix B which is such that

$$B_{ii} = -2$$

$$B_{i+1,i} = B_{i,i+1} = 1.$$

Matrices of a general type of which this is an example, have been examined in some detail, and their properties can be deduced from the properties of a special type of continuant matrix.

This matrix is one of a class known as a "chain matrices".[18] In the case of a 5x5 matrix, a chain matrix W is one which can be written:

$$W = \sum_{r=0}^{4} a_r W(r)$$

where

$$W_{(0)} = I, \; W_{(1)} = \begin{bmatrix} \cdot & 1 & \cdot & \cdot & \cdot \\ 1 & \cdot & 1 & \cdot & \cdot \\ \cdot & 1 & \cdot & 1 & \cdot \\ \cdot & \cdot & 1 & \cdot & 1 \\ \cdot & \cdot & \cdot & 1 & \cdot \end{bmatrix},$$

$$W_{(2)} = \begin{bmatrix} \cdot & \cdot & 1 & \cdot & \cdot \\ \cdot & 1 & \cdot & 1 & \cdot \\ 1 & \cdot & 1 & \cdot & 1 \\ \cdot & 1 & \cdot & 1 & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot \end{bmatrix}, \quad W_{(3)} = \begin{bmatrix} \cdot & \cdot & \cdot & 1 & \cdot \\ \cdot & \cdot & 1 & \cdot & 1 \\ \cdot & 1 & \cdot & 1 & \cdot \\ 1 & \cdot & 1 & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot & \cdot \end{bmatrix},$$

$$W_{(4)} = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot & 1 & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & \cdot & \cdot \end{bmatrix}.$$

The relation

$$W_{(r+1)} = W_{(r)} \cdot W_{(1)} - W_{(r-1)}$$

can be shown to hold between these matrices, and so they can all diagonalised by the same transformation.

The matrix $W_{(1)}$ is a continuant, and so $\left| W_{(1)} - \lambda I \right|$ may be expanded by the usual recurrence relationship, leading to the polynomials:

$$-\lambda, \; \lambda^2 - 1, \; -\lambda^3 + 2\lambda, \text{ etc.}$$

These are Chebychef polynomials of the second type which can be written for $\lambda \leq 2$

$$U_n(\lambda) = (-1)^n \frac{\sin (n+1) \operatorname{acros} \lambda/2}{\sin \operatorname{acros} \lambda/2}$$

and for $\lambda \geq 2$,

$$U_n(\lambda) = (-1)^n \frac{\sinh (n+1) \operatorname{acrosh} \lambda/2}{\sinh \operatorname{acrosh} \lambda/2}.$$

The zeros of $U_n(\lambda)$ are the values of λ for which $(n+1) \operatorname{arcos} (\lambda/2) = s\pi$, i.e.

$$\lambda_s = 2 \cos s\pi/(n+1) \quad (s = 1, 2, \ldots, n).$$

From these roots, the roots of $W_{(2)}$, $W_{(3)}$ etc. can be deduced, and hence the roots of

$$\sum_{r=0}^{n-1} a_r W_{(r)}$$

are available.

The transforming matrix [20] U, where

$$U^T.W.U$$

is diagonal, is specified by

$$U_{ij} = (2/(n+1))^{1/2} \sin {ij}\pi/(n+1)$$

as can be checked by showing that

$$W_{(1)}.U = UD$$

where D is a diagonal matrix such that

$$D_{ii} = \lambda_i .$$

The ability to compute U and D directly can be used in solving the equations

$$W.x = b.$$

If we write

$$x = U.y, \quad c = U^T.b,$$

we have

$$U^T.W.U.y = D.y = c,$$

i.e., $y = D^I.c, \quad x = U.D^I.c.$

Thus we can solve the equations in about n^2 operations.

The elements a_i need not be scalars. They can be any commuting sub-matrices - i.e., any sub-matrices which can be transformed to diagonal form using the same transformation. This enables the process to be used with chain matrices arising from, for example, Laplace's equation in several dimensions expressed in Cartesian co-ordinates.

CONCLUSION

Many facets of numerical analysis have been shown to be expressible in corresponding matrix form, and continuants occur very frequently in these correspondences. A number of these have been given in outline here.

The author has found that considerable didactic and heuristic advantages result from expressing numerical techniques in equivalent continuant form where possible. When this is done, many results arrived at as a result of laborious algebra by other means become almost self-evident.

REFERENCES

1. Aitken, A.C.: "Determinants and Matrices", Oliver and Boyd, 1949.
2. Garwick, J.R.: Archiv for Mathematik og Naturvidenskab 52, 8, 1955.
3. Ritchie, C.C.: Proc. Seventh International Congress for Appl. Mech., 1948.
4. Fox, L. Huskey, H.D., & Wilkinson, J.H.: Quart. J. Mech. Appl. Maths., 1, 286, 1947.
5. Hicks, B.L.: J. Chem. Phys. 8, 569, 1940.
6. Myklestad, N.O.,: J. Aeronaut. Sci., 2, 2, 1944.
7. Erdelyi, A. et. al.: "The Bateman Manuscript Project," 2, McGraw Hill, 1953.
8. Lanczos, C.: J. Res. N.B.S., 45, 255, 1950.
9. Stiefel, E.: Set of four mimeographed lectures prepared for the U.S. National Bureau of Standards and the American University, 1955.
10. Rutishauser, H.: Z.A.M.P., 6, 387, 1955.
11. Turnbull, H.W.: "Theory of Equations," Oliver & Boyd, 1947.
12. Hestenes, M.R. and Stiefel, E.: J.Res. N.B.S., 49, 409, 1952.
13. Fishman, H.: M.T.A.C., 11, 1957.
14. Kopal, Z.: "Numerical Analysis," Chapman & Hall, 1955.
15. Householder, A.S.:"Principles of Numerical Analysis," McGraw Hill, 1953.
16. Taussky, O.: M.T.A.C., 4, 111, 1950.
17. Guest, J.: Aust. Jour. Phys., 8, 425, 1955.
18. Bergerhout, H.: Appl. Sci. Res. B, 4, 161, 1955.
19. Miller, J.C.P.: Proc. Roy. Soc. Edin., A62, 187, 1945.
20. Karlquist, O.: Sartrych ov Tellus, 4, 1952.

THE CALCULATION OF THE EIGENVICTORS OF CODIAGONAL MATRICES PRODUCED BY THE GIVENS AND LANCZOS PROCESSES

by

J.H. Wilkinson

National Physical Laboratory Teddington, Middlesex

1. INTRODUCTION

In both the Givens and the Lanczos methods for calculating the eigenvalues and eigenvectors of a matrix, a collineatory transformation is constructed which reduces the matrix to codiagonal form. Givens[1] has given a complete analysis of the problem of finding the eigenvalues and has described a very satisfactory practical procedure for evaluating them. No such analysis has been given for the eigenvectors, though Givens in an unpublished paper has described a procedure which, in his experience, has given accurate results. In this note an analysis of the problem is given and a method is described which has been used extensively for calculating the vectors on DEUCE.

2. STATEMENT OF THE PROBLEM

The codiagonal forms produced by the Givens and Lanczos processes will be denoted by C_1 and C_2 respectively where we have

$$C_1 = \begin{bmatrix} \alpha_1 & \beta_2 & & & & \\ \beta_2 & \alpha_2 & \beta_3 & & & \\ & \beta_3 & \alpha_3 & \beta_4 & & \\ & & \cdots & \cdots & & \\ & & & \cdots & \cdots & \\ & & & \beta_{n-1} & \alpha_{n-1} & \beta_n \\ & & & & \beta_n & \alpha_n \end{bmatrix} \qquad (1)$$

and

$$C_2 = \begin{bmatrix} \alpha_1 & \beta_2 & & & & \\ 1 & \alpha_2 & \beta_3 & & & \\ & 1 & \alpha_3 & \beta_4 & & \\ & & \cdots & \cdots & & \\ & & & \cdots & \cdots & \\ & & & 1 & \alpha_{n-1} & \beta_n \\ & & & & 1 & \alpha_n \end{bmatrix} \qquad (2)$$