rately. This should speed the convergence and should eliminate most of the effect of imprecision in the scaling factor.

JAMES N. SNYDER

University of Illinois
Urbana, Illinois

1. D. J. WHEELER & J. P. NASH, "Digital and Analogue Computers and Computing Methods." Symposium at the 18th Applied Mechanics Division Conference of the American Society of Mechanical Engineers, University of Minnesota, June 18–20, 1953.
2. D. J. WHEELER, *The Automatic Linear Equation Solver*, University of Illinois Computer Library Routine No. 51.
3. J. N. SNYDER, *The Complete Linear Equation Solver*, University of Illinois Computer Library Routine No. 100.

The Use of Iterative Methods for finding the Latent Roots and Vectors of Matrices

In a recent note in *MTAC* E. BODEWIG [1] presented what he claimed was "a practical refutation of the iteration method for the algebraic eigenproblem." In my opinion this note gave an entirely misleading impression of the value of the iterative method. Moreover an example was chosen as the basis of this refutation which so far from serving the purpose for which it was used, is in fact quite well suited to the iterative method provided it is used in a flexible manner. The iterative method, supplemented by a number of simple devices for accelerating convergence, has been used very effectively on the Pilot ACE to find the latent roots and vectors of a very large number of matrices, symmetric and unsymmetric, real and complex, up to orders as high as 60. Very high accuracy has been achieved even when many or all of the latent vectors of a matrix have been wanted. The details of the method used have been described in a recent paper [2] by WILKINSON, but since that paper was written a magnetic drum store has been added to the Pilot ACE and the speed of iteration has thereby been considerably increased particularly for the larger matrices. The addition of the drum has also led to modifications in the details of the programme which have made it much more satisfactory to use. For this reason a general description of the programme is given below. The note includes an assessment of the value of iteration and concludes with the results achieved with it on Bodewig's example.

An understanding of the iterative programme will be aided by a description of the one or two facilities provided on the Pilot ACE, which are employed therein. There is a register, called the input register, which stores one of the standard words of 32 binary digits, into which a number may be inserted manually by means of 32 keys. The number thus inserted is displayed on a set of 32 lights and this number may be changed at any time during computation by the operator. The input register is addressable in the same way as all the other storage registers and the machine has access in 32 microseconds to the number held there, but it cannot send a word *to* the input register. There is a second register, the output register, the contents of which are also displayed on a set of 32 lights. The machine may send a number to the output register in 32 microseconds, but it cannot read the number stored in it. The machine is also equipped with a monitoring device on which are displayed the contents of 32 consecutive storage registers. The

monitor may be made to display different groups of 32 words by the appropriate setting of a number of keys. The 32 words displayed on the monitor appear on 32 rows in the form shown in fig. 1 where the first three lines only are shown with words of 10 binary digits instead of 32 for convenience.

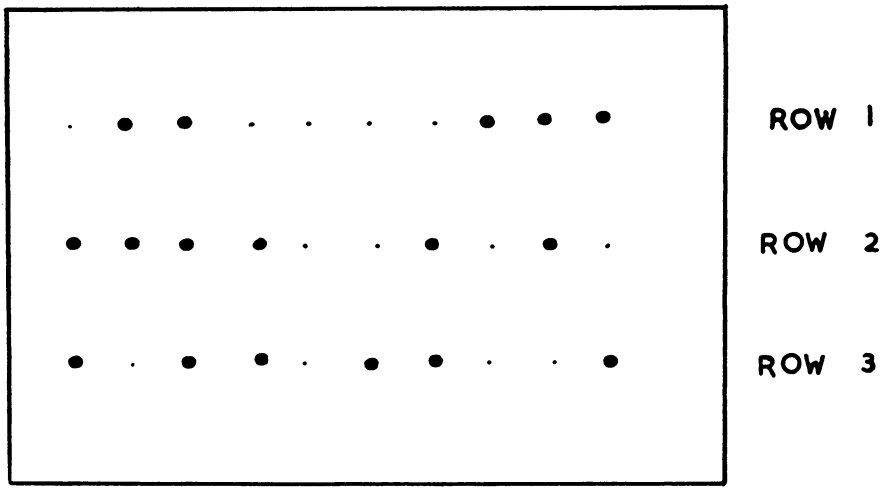

FIG. 1.

The matrix A of which the latent roots are required is stored on the drum, and the present drum can deal with matrices up to order 60. Information may be transferred from the drum to the high speed store in blocks of 32 words at a time, each such transfer taking place in a time which, in this programme, is only 10% of the time taken to use that block of information. The machine starts with an arbitrary vector y_0 (the programme is arranged to read in a good initial guess if one is known) and from it forms two sequences of vectors y_s and z_s defined by the relations

$$z_s = (A - pI)y_s$$

$$y_{s+1} = z_s/(\text{element of } z_s \text{ of maximum modulus})$$

where p is the number set up on the input register. The normalisation of z_s at each stage is carried out in order to keep numbers within range and it has the effect that the largest element in each y_s is unity. Since p is read in at the beginning of each iteration only, it may be changed at any time during an iteration without affecting the value which is used for the remainder of that iteration. The vectors y_s will tend to the latent vector, x, corresponding to that value of λ (assumed real) which is either the algebraically largest latent root or the algebraically smallest, according to the value of p. The value of p may also be chosen to improve the rate of convergence. Although p is set up as a binary number this is not particularly inconvenient because there is no need to choose p with any great precision, so that most of the less significant digits are left as zero. The progress of the convergence may be studied by observing the successive approximations to λ which appear on the output register. Since the time taken to send a number to the output register is 32 microseconds this display does not add

appreciably to the time of an iteration, as would printing successive values of λ. As y_s tends to the latent vector, the largest element of z_s tends to $(\lambda - p)$ and the approximation to λ which is sent to the output register at the end of each iteration is found by adding p to this largest element. It is well known that for a symmetric matrix the RAYLEIGH value given by

$$\lambda = \frac{y_s' A y_s}{y_s' y_s}$$

has, in general, twice as many correct figures as the vector, y_s, from which it is calculated, but no attempt is made to produce such a succession of λ's, the convergence of which would precede that of the vectors. This would be undesirable in that the convergence of the λ's would not then give a true impression of the convergence of the vectors. The sequence of λ's chosen reflects fairly accurately the convergence of the vectors, but experience has shown that there is a distinct tendency for the λ's to converge before the vectors. The behaviour of the vector may be studied by observing the monitor which is set so that it displays the components of the vector z_s or 32 of the components if the matrix is of order greater than 32. The provision of this display does not add to the computing time and great use of it is made in connection with two devices for accelerating convergence.

At the beginning of the iteration process, all the digits of all the components of z_s will change with each iteration, but after a few iterations the more significant digits of z_s will become constant and this is easily observed by the operator. Since the values of p set up on the input register will always contain zeros in the lower digital positions, these positions are effectively free and may be used as further controls on the programme. The two least significant digits of the input register, which are called P1 and P2, are in fact used in the following manner. If at any time the P1 digit is set up on the input register then at the end of the current iteration, the machine takes the last three iterated vectors

$$y_{s-2}, \quad y_{s-1} \quad \text{and} \quad y_s$$

and from them forms the vector Y, the i^{th} component, Y^i, of which is given in terms of the i^{th} components of the three y vectors by the relation

$$Y^i = \frac{y^i{}_{s-2} y^i{}_s - (y^i{}_{s-1})^2}{y^i{}_{s-2} - 2y^i{}_{s-1} + y^i{}_s}.$$

This vector Y is referred to as the AITKEN vector derived from y_{s-2}, y_{s-1} and y_s. In general, as described by WILKINSON [2], the vector Y will be a much improved approximation to the latent vector. In order to be able to form the Aitken vector at any time, it is only necessary to store one extra vector. Suppose the successive vectors y_s are stored in the set of registers A and the vectors z_s in the set of registers B. Then on completing an iteration, if an application of the Aitken process is not required, y_s is transferred from A to a set of registers C and z_s is normalised to y_{s+1} which is then stored in A. If at the end of the next iteration an application of the Aitken process has been requested by the operator then the transfer from A to C does not take place and when z_{s+1}, which is in B, is normalised

to y_{s+2} this is written over z_{s+1} instead of in A. Thus we have y_s in C, y_{s+1} in A and y_{s+2} in B and this is sufficient for an application of the Aitken process. The Aitken process may be used as often or as little as the operator demands, though there must be at least three iterations between each application.

The P2 digit is used to modify the operation of the programme as follows. If during an iteration a P2 digit is set up on the input register, then the machine stores away the current y_s on the drum for use when the subdominant latent vector is being found. Suppose the initial vector y_0 is expressed in terms of the latent vectors $x_1, x_2, \cdots, x_n$ of A by the relation

$$y_0 = \alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_n x_n.$$

Then, after s iterations, y_s will be parallel to the vector

$$\alpha_1 \lambda_1^s x_1 + \alpha_2 \lambda_2^s x_2 + \cdots + \alpha_n \lambda_n^s x_n.$$

and after some iterations we will have in general

$$\alpha_1 \lambda_1^s \gg \alpha_2 \lambda_2^s \gg \alpha_3 \lambda_3^s \gg \cdots \gg \alpha_n \lambda_n^s.$$

This means that after a number of iterations the main impurity in y_s will consist of the x_2 component and y_s will be approximately

$$x_1 + \epsilon x_2.$$

When the iteration for the dominant vector x_1 is complete a matrix of order one less is formed by a process of root removal and it is shown by WILKINSON [2] that

$$(y_s - x_1)$$

where y_s is the vector which was stored on the drum, will, in general, be a very good initial approximation to the latent vector of the reduced matrix. The P2 digit will be used once only during the process of iterating for one vector, and usually the time that is chosen is when about half of the binary digits in the successive y_s are constant. Iteration for each latent vector is continued until all the digits of y_s are constant or the last few digits are going through a repeated cycle of values. This is most important because if we are to use successive root removals, the latent vectors must be as accurate as possible. The decision when iteration for a latent vector is complete is made by the operator and the programme is designed to accept a signal advising it when this is so. The process of reducing the matrix [2] then takes place and the machine begins to iterate with the reduced matrix for the next vector, determining its initial guess as described above.

The last two techniques described are invaluable and are sometimes so successful that, starting from the good initial guess, 3 iterations plus one application of the Aitken process are sufficient for complete convergence.

One or two comments on the role played by the operator might be made at this point. In the first place it should be stressed that at no time does the success of the programme depend critically on the operator's performance. A failure to produce the best value of p, to apply the Aitken process or to store away a vector at the optimum time merely means that one does not gain as much as one might

have done. It is undoubtedly true that an operator becomes rather more proficient with the programme after a little experience with using it but the operators employed on the Pilot ACE are in no sense of the word skilled mathematicians, and I think it is generally agreed that the programme is much more entertaining than any other used on ACE.

In my opinion the "algebraic eigenproblem" is not a problem for which there is a universal solution. There are, in practice, many different requirements. The matrix may be symmetric or unsymmetric and in the latter case, the roots, real or complex. All the roots may be wanted, several dominant roots, the few smallest roots or the roots, if any, in a given range. Again we may or may not want the vectors. Since all machines have stores of finite size often divided up into high speed and auxiliary sections, storage considerations often have a vitally important part to play.

All possible combinations of the above requirements have been met in using the Pilot ACE and the iterative process has proved to be about the most useful weapon at our disposal but it is certainly not the only method we use. For example if all the roots, but not the vectors, of a symmetric matrix are wanted the JACOBI process [3] is undoubtedly better than iteration unless the separation of roots is quite exceptionally good. If a few isolated roots of a symmetric matrix, but not the vectors, are wanted, then the method described by W. GIVENS [4] is almost certainly the most effective of known methods. The iteration method has been used mainly on unsymmetric matrices and most commonly where several of the dominant roots and the vectors have been wanted but it would be used equally on symmetric matrices where the requirement was the same.

As regards the separation of successive roots, Bodewig's statement that a separation of four to one is necessary is very wide of the mark. For some of the larger matrices, of orders 50 to 60, which we have received from aircraft firms, separations of one part in ten have been quite common and much worse separations have been dealt with satisfactorily using the above accelerating techniques. Among smaller matrices quite bad separations have been treated very successfully. Below are given the roots of three matrices A, B and C for which separation is bad. These three examples are chosen from a group of 30 unsymmetric matrices of orders 10 to 15 all of which gave separations comparable with those in the examples.

Roots of Matrix A	Matrix B	Matrix C
0.2144582	0.2155673	0.1562830
0.4644432	0.4826537	0.3571287
0.6813217	0.7523862	0.9520832
0.9824573	1.0217583	1.2017653
1.2457623	1.5162837	1.8635762
1.7428765	2.0256132	1.8801257
2.1281317	2.6578326	2.4613258
2.6315722	4.1243587	2.9980751
3.2458126	4.5625830	3.5287125
5.4916793	5.5612571	4.0165324
5.5491920	5.6013757	4.5238175
5.5695580		

In matrix A, for example, the three dominant roots are separated by one part in 270 and one part in 100 respectively. It was immediately obvious that convergence for the dominant root was very slow so that an attempt was made to find the roots in the reverse order by an appropriate choice of p. A value of p was chosen which was appreciably greater than half the value of the initial approximations which had been sent to the output register when iterating with $p = 0$. The values of p which were used were not kept but it is probable that the root 0.2144582, which was found first, was found by iterating with $(A - pI)$ where p was between 3 and 3.5. The separation between the two dominant roots of $(A - 3I)$, for example, is about one part in ten and this with the accelerating process is quite adequate. The three largest roots were found last and by this time the reduced matrix was of order 3, so that iterations were taking place at the rate of 30 per second. Choosing p to be near the approximations to λ which appeared in the output register also gave quite good separation so that the largest three roots were found very quickly. For matrix C the two roots 1.8635762 and 1.8801257 were found last.

It is popularly believed that the eigenvectors obtained after several root removals tend to be fairly inaccurate. In this connection our experience on the Pilot ACE with several hundred matrices (literally several thousand if matrices of order less than 10 are included!) on which the iteration method has been used is most interesting. If we define the residuals corresponding to a latent root λ and a vector x as the components of

$$Ax - \lambda x$$

then in no example has any residual been greater than 20×2^{-29} (i.e., approximately 37×10^{-9}) where it is assumed that the elements of A are of order unity. Moreover more than 90% of all residuals are in fact below 5×2^{-29} and a set of latent vectors with all its residuals consisting of either zeros, $\pm 2^{-29}$, $\pm 2^{-28}$ is far from uncommon. The main reasons for the accuracy are that iteration for each vector is taken to the limit and scalar products are accumulated with all the digits produced by each multiplication (N.B., the elements of the matrix A and the vectors y_s are single length numbers). No other method which we have used has produced such consistently accurate vectors. Processes in which the latent roots are found from the characteristic equation have, in our experience proved to be deceptively satisfactory when dealing with small matrices. For larger order matrices it very easily happens that such precautions are necessary in forming the characteristic equation that what appears to have been a very fast and effective method for small matrices, proves unexpectedly tiresome for those of higher order. Determining the latent vectors accurately, particularly in the case of unsymmetric matrices is even more hazardous, and it is therefore very misleading to try to assess the relative efficiency of methods by counting the number of multiplications they require. Bodewig remarks that it is possible to find the latent roots and vectors to any accuracy from the characteristic equation but it should be remembered that the accuracy is in fact limited by that of the characteristic equation which has been calculated and that when roots are close

together a given accuracy in the characteristic equation does not produce the same accuracy in the roots.

It is interesting to see how the Pilot ACE programme for iteration dealt with the example given by Bodewig. The coefficients of the matrix were stored in the machine to a precision of 29 binary figures (or 8.7 decimals) for a reason which is associated with the details of the programme. Iteration was started with $p = 0$ and it was immediately obvious that convergence, which one would expect to be almost instantaneous for a matrix of order 4, was in fact proceeding comparatively slowly. As a trial value, to speed convergence, $p = 2$ was set up on the input register. Convergence then took place almost immediately so that no thought was given to finding a better value of p. All four roots were found in a few seconds iteration time and the roots and vectors, after normalising to agree with Bodewig's method, were as given below.

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

$\lambda_1 = -8.02857835$ $\lambda_2 = +7.93290471$ $\lambda_3 = +5.66886437$ $\lambda_4 = -1.57319073$

x_1	x_2	x_3	x_4
+1.00000000	+1.00000000	+1.00000000	+1.00000000
+2.50146029	+0.37781815	+0.95700150	−0.90709211
−0.75773064	+1.38662122	−1.42046822	−0.37759122
−2.56421169	+0.34880573	+1.74331690	−0.38333124

The vectors agree, apart from the end figure of one or two components, with those given by Bodewig, except for the final component of x_4 where the two results differ by 0.00001886. It is easy to satisfy oneself that Bodewig's result is in error, and it is probable that his results were copied incorrectly. The only peculiarity displayed by Bodewig's example was that its two dominant roots were almost equal and opposite in value and this is not a difficult situation. Indeed even if the roots had been exactly equal and opposite in value there would have been no difficulty whatever. If the two dominant roots had had the values in the example but with the same sign the situation would have been just a little less satisfactory. Then the root -1.573 etc., would have been found first and the root 5.669 second. The matrix obtained after two reductions would have been of order two, so that iteration would have been extremely fast and a choice of p anywhere near 8.0 would have given almost immediate convergence. Such a choice of p would have been suggested by the approximations to λ obtained on the output register. Besides being a poor choice on which to base a refutation of the iteration process, Bodewig's example has other weaknesses which make it a rather poor one for estimating the relative efficiency of methods. In the first place it is of very low order so that it does not reveal the difficulties of preserving accuracy which arise with larger matrices and secondly, it has small integer coefficients. This

means, for example, that if we use a method in which we find the characteristic equation we will almost certainly obtain the latter exactly, which will not be true if the coefficients were numbers with several digits.

J. H. WILKINSON

National Physical Laboratory
Teddington, Middlesex
England

This note is published with the permission of the Director of the National Physical Laboratory.

1. E. BODEWIG, "A practical refutation of the iteration method for the algebraic eigenproblem," *MTAC*, v. 8, 1954, p. 237–239.
2. J. H. WILKINSON, "The calculation of the latent roots and vectors of matrices on the Pilot Model of the ACE," Cambridge Phil. Soc., *Proc.*, v. 50, 1954, p. 536–566.
3. C. G. J. JACOBI, "Über ein leichtes Verfahren, die in der Theorie der Säkularstörungen vorkommenden Gleichungen numerisch aufzulösen," *J. reine angew. Math.* 30, 1846, p. 51–94.
4. W. GIVENS, "Numerical computation of the characteristic values of a real symmetric matrix," Oak Ridge National Laboratory. ORNL1574.

TECHNICAL NOTES AND SHORT PAPERS

The Specific Heat Function for a Two-Dimensional Continuum

Numerical Values of

$$\frac{C_2}{C_\infty} = \frac{6}{x^2}\int_0^x \frac{\xi^2 d\xi}{e^\xi - 1} - \frac{2x}{e^x - 1}.$$

This function which appears in the theory of low-temperature specific heats of two-dimensional (layer) structures [1] was computed as follows:

(1) For $0 \leqslant x \leqslant 2.0$, the formula

$$\frac{C_2}{C_\infty} = 1 - \frac{x^2}{24} + \frac{x^4}{720} - \frac{x^6}{24{,}192} + \frac{x^8}{864{,}000} - \frac{x^{10}}{31{,}933{,}440} + \cdots$$

was used. The maximum error (using seven terms) is no greater than 0.5×10^{-5}.

(2) For $2.0 \leqslant x \leqslant 16.0$, the formula

$$\frac{C_2}{C_\infty} = \frac{14.424684}{x^2} - 6x\sum_{n=1}^{\infty} e^{-nx}\left(\frac{1}{nx} + \frac{2}{(nx)^2} + \frac{2}{(nx)^3}\right) - \frac{2x}{e^x - 1}$$

was used. The maximum error was approximately 2×10^{-6}.

The value $C_\infty = 3R = 5.9616$ cal mol^{-1} deg^{-1} was used throughout [2].

x	C_2/C_∞	C_2
0.0	1.00000	5.9616
0.1	0.99958	5.95911
0.2	.99833	5.95167
0.3	.99625	5.93925
0.4	.99333	5.92186
0.5	.98959	5.89955