Journal of Research of the National Bureau of Standards Vol. 49, No. 1, July 1952 Research Paper 2341

Solution of Systems of Linear Equations by Minimized Iterations[1]

Cornelius Lanczos

A simple algorithm is described which is well adapted to the effective solution of large systems of linear algebraic equations by a succession of well-convergent approximations.

1. Introduction

In an earlier publication [14][2] a method was described which generated the eigenvalues and eigenvectors of a matrix by a successive algorithm based on minimizations by least squares.[3] The advantage of this method consists in the fact that the successive iterations are constantly employed with maximum efficiency which guarantees fastest convergence for a given number of iterations. Moreover, with the proper care the accumulation of rounding errors can be avoided. The resulting high precision is of great advantage if the separation of closely bunched eigenvalues and eigenvectors is demanded [16].

It was pointed out in [14, p. 256] that the inversion of a matrix, and thus the solution of simultaneous systems of linear equations, is contained in the general procedure as a special case. However, in view of the great importance associated with the solution of large systems of linear equations, this problem deserved more than passing attention. It is the purpose of the present discussion to adopt the general principles of the previous investigation to the specific demands that arise if we are not interested in the complete analysis of a matrix but only in the more special problem of obtaining the solution of a given set of linear equations

$$Ay=b_0 \tag{1}$$

with a given matrix A and a given right side b_0. This is actually equivalent to the evaluation of one eigenvector only, of a symmetric, positive definite matrix. It is clear that this will require considerably less detailed analysis than the problem of constructing the *entire* set of eigenvectors and eigenvalues associated with an arbitrary matrix.

2. The Double Set of Vectors Associated With the Method of Minimized Iterations

The previous investigation [14] started out with an algorithm (see p. 261) which generated a double set of polynomials, later on denoted by $p_t(x)$ and $q_t(x)$ (see p. 274). Then a second algorithm was introduced, called "minimized iterations", which avoided the numerical difficulties of the first algorithm (see p. 287) and had, in addition, theoretically valuable properties for the solution of differential and integral equations (p. 272).

In this second algorithm, however, only *one-half* of the previous polynomials were represented, corresponding to the $p_t(x)$ polynomials whose coefficients appeared in the *full* columns of the original algorithm [14, (60)]. The polynomials $q_t(x)$, associated with the *half columns* of [14, (60)] did not come into evidence in the later procedure.

The vectors b_t, generated by minimized iterations, correspond to the polynomials $p_t(x)$ in the sense

$$b_k=p_k(A)b_0. \tag{2}$$

We should expect that the vectors generated by $q_k(A)b_0$ might also have some significance. We will see that this is actually the case. It is of considerable advantage to translate the *entire* scheme [14, (60)] into the language of minimized iterations, without omitting the half columns. We thus get a *double* set of vectors, instead of the single set considered before.

The additional work thus involved is not superfluous because the second set of polynomials can be put to good use. Moreover, the two sets of polynomials belong logically together and complement each other in a natural fashion. From the practical standpoint of adapting the resultant algorithm to the demands of large scale electronic computers, we gain in the simplicity of coding. The recurrence relations which exist between the polynomials $p_t(x)$, $q_t(x)$ are simpler in structure than the recurrence relation obtained by eliminating the second set of polynomials.

We want to simplify and systematize our notations. The vector obtained by letting the polynomial $p_k(A)$ operate on the original vector b_0, shall be called p_k:

$$p_k=p_k(A)b_0. \tag{3}$$

We thus distinguish between p_k as a *vector* and $p_k(A)$ as a *polynomial operator*. Hence the notation p_k will take the place of the previous b_k. Correspondingly we denote the associated second set of vectors by q_k:

$$q_k=q_k(A)b_0. \tag{4}$$

[1] The preparation of this paper was sponsored (in part) by the Office of Naval Research.

[2] Figures in brackets indicate the literature references at the end of this paper.

[3] The present paper is a natural sequel to the previous publication and depends on the previous findings. The reader's familiarity with the earlier development is assumed throughout this paper; the symbolism of the present paper is in harmony with that used before, in particular the notation pq, if applied to vectors, shall mean the scalar product of these two vectors.

Both of these vector sets have invariant significance. The vectors $p_k(A)b_0$ can be characterized as the solution of the following minimum problem. Form the polynomial

$$\begin{aligned} p_k &= [A^k - (a_0 + a_1 A + \ldots + a_{k-1}A^{k-1})]b_0 \\ p_k^* &= [A^{*k} - (a_0 + a_1 A^* + \ldots + a_{k-1}A^{*k-1})]b_0^* \end{aligned} \tag{5}$$

determining the coefficients a_i by the condition that the square of the length of p_k, that is, the invariant $p_k p_k^*$ shall become a minimum.

The vectors $q_k(A)b_0$ can be characterized as the solution of the following minimum problem. Form the polynomial

$$\begin{aligned} \bar{q}_k &= [1 - (\bar{a}_1 A + \bar{a}_2 A^2 + \ldots + \bar{a}_k A^k)]b_0 \\ \bar{q}_k^* &= [1 - (\bar{a}_1 A^* + \bar{a}_2 A^{*2} + \ldots + \bar{a}_k A^{*k})]b_0^* \end{aligned} \tag{6}$$

determining the coefficients $\bar{a}_i$ by the condition that the square of the length of $\bar{q}_k$, that is, the invariant $\bar{q}_k \bar{q}_k^*$, shall become a minimum.

In the case (5) the highest coefficient of the polynomial is normalized to 1, and in the case (6) the lowest coefficient is unity.[4]

After the minimization we shall normalize, for the sake of convenience, the largest coefficient of $\bar{q}_k$ once more to 1; hence we define

$$q_k = -\frac{1}{\bar{a}_k}\bar{q}_k. \tag{7}$$

While the vectors p_k and p_k^* form a biorthogonal set of vectors [14, p. 266], this cannot be said of the vectors q_k. However, the vectors q_k are of particular importance for the solution of sets of linear equations. If we form the ratio

$$\bar{y}_{k-1} = \frac{q_k(A) - q_k(0)}{-q_k(0)A}\, b_0, \tag{8}$$

we have obtained a solution of the equation

$$A\bar{y}_{k-1} - b_0 = -\bar{q}_k. \tag{9}$$

Hence we see that if the vectors q_k are at our disposal, we can at every step of our algorithm obtain an *optimum solution of smallest residual*. Indeed, the vector $\bar{q}_k$ was defined by the condition that it shall have the smallest length among all the linear combinations which can be formed with the help of the successive iterates

$$b^m = Ab^{m-1} = A^m b_0 \tag{10}$$

up to the order k.

The alternate solution

$$y_{k-1} = \frac{p_k(A) - p_k(0)}{-p_k(0)A}\, b_0 \tag{11}$$

gives a larger residual for the same k, except if we proceed to the very end of the process, $k=n$, in which case the residual vanishes for both y_k and $\bar{y}_k$ and both coincide with the exact solution y:

$$y_{n-1} = \bar{y}_{n-1} = y. \tag{12}$$

3. The Complete Algorithm for Minimized Iterations

We will now proceed to the exposition of the completed algorithm which does not omit one-half of the basic algorithm [14, (60)] but translates the *entire* algorithm into the frame of reference of minimized iterations.

The algorithm [14, (60)], generated a double set of polynomials, mutually interlocked by the following recurrence relations:

$$\begin{aligned} p_{k+1}(x) &= \rho_k p_k(x) + x q_k(x) \\ q_{k+1}(x) &= \sigma_k q_k(x) + p_{k+1}(x). \end{aligned} \tag{13}$$

Elimination of the $q_k(x)$ leads to the three-term recurrence relation for the $p_k(x)$ alone:

$$p_{k+1}(x) = (x - \alpha_k)p_k - \beta_{k-1}p_{k-1}(x) \tag{14}$$

with[5]

$$\begin{aligned} \alpha_k &= -(\rho_k + \sigma_{k-1}) \\ \beta_k &= \rho_k \sigma_k. \end{aligned} \tag{15}$$

On the other hand, elimination of the $p_i(x)$ leads to the three-term recurrence relation of the $q_i(x)$ alone:

$$q_{k+1}(x) = (x - \bar{\alpha}_k)q_k(x) - \bar{\beta}_{k-1}q_{k-1}(x) \tag{16}$$

with

$$\begin{aligned} \bar{\alpha}_k &= -(\rho_k + \sigma_k) \\ \bar{\beta}_k &= \rho_{k+1}\sigma_k. \end{aligned} \tag{17}$$

Replacing x by A, A^* and letting these polynomials operate on b_0, b_0^*, we obtain the following relations between the vectors p_k and q_k:

$$\begin{aligned} p_{k+1} &= \rho_k p_k + q_k' & p_{k+1}^* &= \rho_k p_k^* + q_k^{*\prime} \\ q_{k+1} &= \sigma_k q_k + p_{k+1} & q_{k+1}^* &= \sigma_k q_k^* + p_{k+1}^*. \end{aligned} \tag{18}$$

The notation "prime" refers to the multiplication by the matrix A:

$$\begin{aligned} q_k' &= Aq_k \\ q_k^{*\prime} &= A^* q_k^*. \end{aligned} \tag{19}$$

[4] The definition of the vectors p_k and p_k^* reveals the following remarkable property of this vector set. Let b_0 remain unchanged but the matrix A be changed to $A - \lambda I$, where λ is arbitrary. The vectors p_k, p_k^* *remain invariant* with respect to this transformation. The same cannot be said of the vectors q_k, q_k^*.

[5] The negative signs in (14) and (16) are chosen because for symmetric and positive definite matrices an important prediction can be made concerning the signs of the fundamental scalars. The original algorithm which introduces the h_i and h_i' coefficients reveals [14, p. 262] that both of these coefficients arise from a minimization process and both of them have the significance of the square of a length. In the case of symmetric (or Hermitian) and positive definite matrices the metric is real and the square of a length necessarily positive. Hence the h_i and h_i' are *all positive*, the ρ_i, σ_i all negative. This makes the α_i and β_i (and likewise the $\bar{\alpha}_i$, $\bar{\beta}_i$) always *positive* for such matrices.

Now the biorthogonality of the vectors p_i gives, if we multiply the upper left equation (18) by p_k^*

$$\rho_k = -\frac{p_k^* q_k'}{p_k^* p_k} = -\frac{p_k q_k^{*\prime}}{p_k p_k^*}. \tag{20}$$

Moreover, the same equation shows the orthogonality of q_k' to all p_k^*, except $m=k$ and $k+1$. In particular

$$(q_k' p_{k-1}^*) = (q_k^{*\prime} p_{k-1}) = 0. \tag{21}$$

Now we prime the second equation and multiply on both sides by p_k^*. This gives:

$$\sigma_k = -\frac{p_k^* p_{k+1}'}{p_k^* q_k'} = -\frac{p_{k+1}^* p_{k+1}}{p_k^* q_k'}. \tag{22}$$

We introduce the scalars h_i and h_i' by putting

$$\begin{aligned} h_k &= p_k^* p_k \\ h_k' &= p_k^* q_k' = p_k q_k^{*\prime} \end{aligned} \tag{23}$$

and obtain:

$$\begin{aligned} \rho_k &= -\frac{h_k'}{h_k} \\ \sigma_k &= -\frac{h_{k+1}}{h_k'}. \end{aligned} \tag{24}$$

This completely translates the "progressive algorithm" into the language of minimized iterations. The h_i numbers are identical with the h_i of the scheme [14], (60) (p. 263), corresponding to the full columns 0, 1, 2, . . . , while the h_i' give the h-numbers of the half columns 0.5, 1.5, 2.5, ...[6]

A remarkable relation between the ρ_i and the determinant of the matrix A can be obtained if in the first equation of (13) we substitute $x=0$:

$$p_{k+1}(0) = \rho_k p_k(0). \tag{25}$$

Hence

$$p_k(0) = \rho_0 \rho_1 \ . \ . \ . \ \rho_{k-1} \tag{26}$$

and

$$p_n(0) = \rho_0 \rho_1 \ . \ . \ . \ \rho_{n-1}. \tag{27}$$

Since $p_n(A)\ b_0 = 0$ yields the characteristic equation of the matrix A, $(-1)^n p_n(0)$ must be the determinant associated with the matrix A. The determinant of A is thus obtained as the product of all the ρ_i, multiplied by $(-1)^n$:

$$|A| = (-1)^n \rho_0 \rho_1 \ . \ . \ . \ \rho_{n-1}. \tag{28}$$

In the following sketch of the general work scheme we will restrict ourselves to the particularly important case of *symmetric* matrices. This suffices for the purpose of solving linear equations that can always be symmetrized, by transforming the originally given set

$$Gy = g \tag{29}$$

into

$$Ay = b_0, \tag{30}$$

where

$$A = G^* G \tag{31}$$

$$b_0 = G^* g. \tag{32}$$

The matrix A is now symmetric and positive definite. In this case the general scheme is reduced to one half of its original size, since

$$\begin{aligned} A &= A^* \\ b_0 &= b_0^*. \end{aligned} \tag{33}$$

We need not distinguish between p_i and p_i^*, q_i, and q_i^*, since our reference system is orthogonal and the adjoint vector coincides with the vector itself.

The actual construction of the symmetrized matrix A is a very "expensive" operation, since it is equivalent to n matrix multiplications of the type Ab. Actually, we never need the matrix A itself but only A operating on a certain vector b. By the associative law $(G^*G)b = G^*(Gb)$. Hence the operation Ab is equivalent to the performing of the two successive matrix multiplications $b^{(1)} = Gb$ and $b^{(2)} = G^* b^{(1)}$. This requires $2n^2$ multiplications, compared with $\frac{1}{2}n^2(n+1)$ multiplications required for constructing G^*G.

Every cycle in the following iteration scheme consists of the construction of three vectors, viz., p_i, q_i, q_i'. The third is merely the matrix A applied to q_i. Hence the problem is reduced to the construction of the vectors p_i and q_i. In the following symbolic work scheme (34) the sequence of operations is indicated by going from row to row, and in each row from the left to the right:

$p_0 = b_0$	$h_0 = p_0^2$	
$q_0 = b_0$	———	
$q_0' = A q_0$	$h_0' = p_0 q_0'$	$\rho_0 = -\frac{h_0'}{h_0}$
———		
$p_1 = \rho_0 p_0 + q_0'$	$h_1 = p_1^2$	$\sigma_0 = -\frac{h_1}{h_0'}$
$q_1 = \sigma_0 q_0 + p_1$	———	
$q_1' = A q_1$	$h_1' = p_1 q_1'$	$\rho_1 = -\frac{h_1'}{h_1}$
———		
. . .	. . .	. . .

(34)

This scheme is characterized by great uniformity and is well suited to coding for large scale machines. The generation of each new pair of p_i, q_i vectors occurs constantly by the same scheme and involves

[6] The same algorithm shows another remarkable property of the q_i vectors. These vectors do not form an orthogonal set because the polynomials $q_i(A)$ have the property to give orthogonality only if they operate on $\sqrt{A}\ b_0$ rather than b_0 itself. But then by the associative law $[q_i(A)\sqrt{A}b_0]\ [q_k(A)\sqrt{A}b_0]^* = 0$ implies $[q_i(A)b_0]\ [q_k(A)Ab_0]^* = 0$, which gives $q_k q_i' = 0 \quad (i \neq k)$. This means that in the following work scheme the first rows (the p vectors) form an orthogonal set, but in addition *the second and third rows form likewise a mutually orthogonal* (biorthogonal) *set.*

for both vectors uniformly the immediately preceding vector and the penultimate vector (we skip the vector between). For example, p_1 is obtained as a combination of q_0' and b_0 (we skip q_0), whereas q_1 is obtained as a combination of p_1 and q_0 (we skip q_0'). The immediate predecessor is merely added, whereas the earlier predecessor is always multiplied by the negative ratio of the last two h-numbers (h_0' and h_0 in the case of p_1, h_1 and h_0' in the case of q_1).

It may help the coder to have a geometric picture of the scheme as a whole—such as the scheme that might profitably be used by a desk computer. In such an arrangement the ρ_i and σ_i factors should be placed in *front* of the respective rows that they multiply. Hence we keep a column free in front of the vector scheme and write down ρ_0, immediately in front of p_0; σ_0 in front of q_0, and so on. Moreover, it is of advantage to carry an extra column at the end of the vector scheme which makes the vectors $n+1$-dimensional instead of n-dimensional. This extra column does not participate in the formation of the h_i and h_i', but otherwise we operate with it exactly as with the other columns. The element that completes q_i' is always put equal to zero. The first two vectors p_0 and q_0 are completed by 1.

This "surplus" column provides two important scalars, namely, $p_k(0)$ and $q_k(0)$. The last row gives p_n, which is the null vector. The "surplus" element $p_n(0)$ associated with p_n terminates the algorithm, and gives the determinant of A, multiplied by $(-1)^n$.

The following numerical example is intentionally simple, since the aim is to display the operations rather than the numerical details. For the same reason the fractions encountered are not changed into decimals but left in fractional form.

$$\begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix} y = \begin{matrix} 1 \\ 1 \\ 1 \\ 0 \end{matrix}$$

ρ_0: $-\frac{2}{1}$	p_0 :	1	1	1	0	$p_0(0) =$	1	h_0 : 3		
σ_0: $-\frac{1}{2}$	q_0 :	1	1	1	0	$q_0(0) =$	1		η_0:	$\frac{1}{2}$
	q_0' :	1	0	1	−1	0		h_0' : 2		
ρ_1: $-\frac{3}{2}\frac{1}{6}$	p_1 :	$\frac{1}{2}$	$-\frac{3}{2}$	$\frac{1}{2}$	−1	$p_1(0) =$	$-\frac{3}{2}$	h_1 : $\frac{5}{2}$		
σ_1: $-\frac{2}{5}$	q_1 :	$-\frac{1}{2}$	$-\frac{5}{2}$	$-\frac{1}{2}$	−1	$q_1(0) =$	$-\frac{5}{2}$		η_1:	$-\frac{5}{7}$
	q_1' :	$\frac{1}{2}$	−2	$\frac{3}{2}$	$-\frac{3}{2}$	0		h_1': $\frac{7}{2}$		
ρ_2: $-\frac{1}{7}\frac{0}{0}$	p_2 :	$-\frac{1}{5}$	$-\frac{5}{5}$	$\frac{4}{5}$	$\frac{3}{5}$	$p_2(0) =$	$\frac{7}{7}$	h_2 : $\frac{7}{5}$		
σ_2: $-\frac{1}{14}$	q_2 :	0	0	0	1	$q_2(0) =$	2		η_2:	$\frac{1}{2}$
	q_2' :	0	−1	1	1	0		h_2' : 2		
ρ_3: $-\frac{5}{2}$	p_3 :	$\frac{2}{7}$	$-\frac{1}{7}$	$-\frac{1}{7}$	$\frac{1}{7}$	$p_3(0) =$	−2	h_3 : $\frac{1}{7}$		
	q_3 :	$\frac{2}{7}$	$-\frac{1}{7}$	$-\frac{3}{14}$	$\frac{1}{14}$	$q_3(0) =$	$-\frac{1}{7}\frac{0}{0}$		η_3:	$-\frac{1}{5}$
	q_3' :	$\frac{5}{7}$	$-\frac{5}{14}$	$-\frac{5}{14}$	$\frac{5}{14}$	0		h_3' : $\frac{5}{14}$		
	p_4 :	0	0	0	0	$p_4(0) =$	5			

The scheme comes automatically to a halt whenever the first p_i vanishes in all its components. If the vector b_0 has no "blind spots" in the direction of any of the principal axes, then the scheme will continue until $k=n$, and the first p_i that vanishes will be p_n. This is p_4 in our example, since $n=4$. The element in the bracketed column associated with p_4 is 5. Hence the determinant of the given system is established as 5.

Numerical checks. The algorithm provides the following powerful checks for the numerical calculations:

(a) The dot-product of any two different p-vectors is zero.

(b) The dot-product of any q-vector with any q'-vector except its own pair, is zero.

(c) Within each cycle the scalar h_i' can be obtained in two different ways: $h_i' = p_i q_i' = q_i q_i'$.

If we are interested in finding the characteristic equation of the matrix, we proceed in identical fash-

ion with the only difference that we put in the bracketed column opposite to q_i' not zero but the algebraic quantity λ times the element immediately above it. In our example, if we write the successive vertical elements of each cycle horizontally, the bracketed column becomes:

Cycle 0: 1, 1, λ

1: $-\frac{2}{3}+\lambda$, $-\frac{3}{2}+\lambda$, $-\frac{3}{2}\lambda+\lambda^2$

2: $\frac{7}{5}-\frac{18}{5}\lambda+\lambda^2$, $2-4\lambda+\lambda^2$, $2\lambda-4\lambda^2+\lambda^3$

3: $-2+\frac{50}{7}\lambda-\frac{38}{7}\lambda^2+\lambda^3$, $-\frac{15}{7}+\frac{52}{7}\lambda-\frac{11}{2}\lambda^2+\lambda^3$, $-\frac{15}{7}\lambda+\frac{52}{7}\lambda^2-\frac{11}{2}\lambda^3+\lambda^4$

4: $5-20\lambda+21\lambda^2-8\lambda^3+\lambda^4$.

The last polynomial is the characteristic polynomial whose roots give the eigenvalues λ_i of the matrix. The significance of the last column η_i will be explained in the next chapter.

4. Solution of the Linear System by the q-Expansion

So far we have constructed the two vector sets p_i and q_i, which characterize the method of minimized iterations. Our aim is, however, to obtain the solution y of the given linear set. For this purpose we assume that the vector y is expanded into the q_i-vectors:

$$y=\sum_{i=0}^{n-1}\eta_i q_i. \tag{35}$$

We now form the equation

$$Ay=b_0=p_0 \tag{36}$$

for the right side of (35). Making use of the first equation of the fundamental recurrence relation (18), we obtain the following recurrence set for the coefficients η_i of the expansion (35):

$$\begin{aligned} -\rho_0\eta_0&=1\\ -\rho_1\eta_1+\eta_0&=0\\ -\rho_{i+1}\eta_{i+1}+\eta_i&=0. \end{aligned} \tag{37}$$

Hence

$$\eta_{i+1}=\frac{\eta_i}{\rho_{i+1}} \tag{38}$$

starting with

$$\eta_0=-\frac{1}{\rho_0}. \tag{39}$$

In solved form

$$\eta_i=-\frac{1}{\rho_0\rho_1\cdots\rho_i}=-\frac{1}{p_{i+1}(0)}. \tag{40}$$

The vector equation (35), if translated into matrix language, has the following significance. Write the η_i as a column vector and multiply this column with the successive columns of the matrix Q, formed out of the middle vectors q_i of the iteration scheme (34). For this reason the numerical scheme (34) is augmented by a last column, composed of the successive η_i, and written down in the corresponding rows of the vectors q_i. We find in our numerical scheme the element

$$\eta_0=-\frac{1}{\rho_0}=\frac{3}{2}$$

in the row q_0, the element

$$\eta_1=\frac{\eta_0}{\rho_1}=\frac{3}{2}:\frac{21}{10}=-\frac{5}{7}$$

in the row q_1, and so on. Multiplication of this column by the successive columns of the q_i yields the successive components of the solution y:

$$y=\frac{9}{5},\frac{13}{5},\frac{12}{5},\frac{6}{5}.$$

Substitution into the original equation shows that this is indeed the correct solution.

If we do not carry the bracketed surplus column of our scheme, then it is convenient to generate the η_i in succession on the basis of the recursion (38), writing each η_i in line with the vector q_i. If the bracketed column is at our disposal, then we merely take the negative reciprocal of the first bracketed element in each cycle and transfer it to the q_i immediately preceding it. For example the first element of cycle 1 in the bracketed column is $-\frac{2}{3}$, the negative reciprocal is $\frac{3}{2}$, which is transferred to the middle line of the previous cycle. Then $\frac{7}{5}$ is transferred as $-\frac{5}{7}$ to the middle line of the previous cycle, and so on, until all the first elements of the bracketed column are exhausted, the last $\eta_i=\eta_{n-1}$ being the reciprocal of the determinant $|A|$. The sign of the η_i always alternates between $+$ and $-$.

The objection may be raised that the vectors p_i and q_i have no invariant significance in relation to the matrix A. They depend on b_0 and thus, while we did get the solution or the given linear set, yet the matrix inversion gives much more because it is immediately applicable to *any* given right side b_0.

Now the remarkable fact holds that actually our p_i, q_i, although generated with the help of some specific b_0, nevertheless, include the solution of a linear set with any given right side c. The right side of the equations (37) is 1, 0, 0, . . . only because the vector b_0, analyzed in the reference system of the p_i, has these components. Since, however, the p_i form an *orthogonal* set of vectors,

we can immediately analyze any given c in this frame of reference. The components of c in this system become

$$\frac{cp_0}{h_0}, \frac{cp_1}{h_1} \cdots \frac{cp_{n-1}}{h_{n-1}}$$

generally

$$\mu_i = \frac{cp_i}{h_i} \tag{41}$$

and these are the quantities that in the general case appear on the right side of (37):

$$\begin{aligned} \rho_0\eta_0 &= -\mu_0 \\ \rho_1\eta_1 - \eta_0 &= -\mu_1 \\ &\cdot \\ &\cdot \\ &\cdot \\ \rho_{i+1}\eta_{i+1} - \eta_i &= -\mu_{i+1}. \end{aligned} \tag{42}$$

This set is again readily solvable by recursions. Then after obtaining the vector η, we obtain y once more by (39).

Example. In our numerical example let us replace the right side by

$$c = 0, 0, 0, 1.$$

The dot-products of this c with the vectors p_i, divided by h_i become:

$$\mu = 0, -\frac{3}{5}, \frac{3}{7}, 1$$

and the step by step solution of (42) gives:

$$\eta = 0, -\frac{2}{7}, \frac{1}{2}, \frac{1}{5}.$$

Multiplying again by the q-vectors we obtain

$$y = \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5},$$

which is the correct solution.

$$Ay = c \tag{43}$$

with *any* given right side c is obtainable if we first construct the p_i, q_i vectors with the help of some definite b_0, which can be arbitrary except for the fact that it shall have no blind spots in the direction of any of the principal axes. If b_0 is deficient in the direction of m axes of A, then the iteration scheme will come to an end after $n-m$ iterations. This will necessarily happen if the matrix A has multiple eigenvalues, no matter how b_0 was chosen. Let a certain λ_i have the multiplicity μ. Then there is a μ-dimensional subspace in which the direction of the principal axes is undetermined. Let us project b_0 into this subspace. We get a definite vector which may be chosen as one of the principal axes. Then b_0 is still deficient in the other $\mu-1$ possible orthogonal axes.

From this viewpoint the premature termination of our scheme can always be conceived as a consequence of the deficiency of b_0, no matter whether that deficiency originates in the accidental degeneracy of b_0, or in the degeneracy of the matrix A. Whenever this situation is encountered, we do not obtain a full solution of the equation (43). Yet we have obtained a preliminary $y^{(1)}$ which solves the equation at least in all the nondeficient directions. If we then form $Ay^{(1)} - c = c^{(1)}$, this $c^{(1)}$ will contain only dimensions which before did not come into evidence. We can now repeat the scheme (34) once more, using $c^{(1)}$ as the b_0 of the new scheme; we obtain a new set of p_i, q_i vectors which can be added to the previous set. Assuming that $c^{(1)}$ does not bring in newer deficiencies relative to the previously omitted subspace, we will now have a complete set of p_i, q_i vectors which include the entire space. If some dimensions are still omitted, the procedure can be continued, until all n-dimensions of the vector space are exhausted.

The outstanding feature of the recurrence relations (37) and (42) is the fact that they are *two-term* relations. This has the following remarkable consequence. We have pointed out before that we can consider the successive stages of our iteration process as a succession of *approximations*. At every step of the process we can form the ratios (11) or (8) and thus obtain approximations y_k and $\bar{y}_k$ which come nearer and nearer to the true solution as the residual diminishes. Now the set (42) shows that this successive approximation process does not need constant readjustments as we go from k to $k+1$. *The previous approximation remains unchanged*, we merely add one more vector, namely, $\eta_{k+1}q_{k+1}$.

The expansion (35) into the q-vectors thus imitates the behavior of an orthogonal expansion whose coefficients remain unchanged as we gradually introduce more and more vectors of the function space until finally all dimensions are exhausted. This shows the superiority of the q_i-vectors for expansion purposes. If the vectors p_i are used, the relations involve three-term recurrences and we cannot solve the set by one single recursion, but need the proper linear combination of *two* recursions; this involves constant modification of the approximation previously obtained.

If we pursue our procedure as a sequence of successive approximations which may be terminated at any point where the residual has dropped down below a preassigned limit, it will be important to obtain not only the subsequent corrections, but also the remaining residual. This residual is directly available. The remaining residual, that is, right side minus left side of the linear system after substituting the kth approximation y_k, is simply given by the quantity

$$r_{k+1} = -\eta_k p_{k+1}. \tag{44}$$

For example, if in our numerical scheme we stop with η_2, we obtain the approximation

$$y_2 = \tfrac{13}{7}, \tfrac{18}{7}, \tfrac{33}{14}, \tfrac{17}{14}.$$

The residual associated with this approximation is thus

$$r_3 = -\eta_2 p_3 = -\tfrac{1}{7}, \tfrac{1}{14}, \tfrac{1}{14}, -\tfrac{1}{14}$$

which can be verified by substitution.

By merely watching the last two columns of our scheme we can constantly keep track of the successive whittling down of the residual. The length of the remaining residual is obtained by multiplying the last η_i by the square root of the next following h_i (we skip h_i'). For example in our numerical problem the lengths of the successive residuals become:

$$|r_i|\colon \sqrt{3}=1.7321, \quad \tfrac{3}{2}\sqrt{\tfrac{5}{3}}=1.9365, \quad \tfrac{5}{7}\sqrt{\tfrac{7}{5}}=0.8452,$$
$$\tfrac{1}{2}\sqrt{\tfrac{7}{7}}=0.1890, \quad \tfrac{1}{5}\sqrt{0}=0.$$

The simple expression of the residual (44) is of great advantage if we decide to use our process in "blocks" rather than as a continuous succession of operations. The accumulation of rounding errors tends to destroy the orthogonality of the p_i more and more. If we do not want to take recourse to the lengthy process of constant reorthogonalization, we can break our operations in blocks as soon as we notice that the rounding errors have done too much damage to the orthogonality. In that case we evaluate the remaining residual and start the process independently over again. The accumulation of rounding errors is thus avoided, at the price of retarded convergence.

Now the expression (44) shows that very little adjustment is needed in order to change from the continuous technique to the block technique.

The residual of the last block serves as the initial vector of the new block. Now the residual of a block of $k+1$ cycles (the cycles being numbered as 0, 1, 2, . . ., k) is $-\eta_k p_{k+1}$. In the continuous flow of operations the next cycle would have started with p_{k+1}. The changing over to independent blocks merely requires that we should multiply this vector by the negative value of the preceding η_k, but this is equivalent to the division by $p_{k+1}(0)$ which can be found in the surplus column of the same p_{k+1} row.

Hence the change to the block technique merely requires that we should continue in the regular fashion up to the row

$$p_{k+1}, \quad p_{k+1}(0)$$

which terminates that block. The next block starts with

$$\frac{p_{k+1}}{p_{k+1}(0)}, \ 1,$$

and we repeat under it once more

$$\frac{p_{k+1}}{p_{k+1}(0)}, \ 1.$$

These are the p_0, q_0 of the new block, and now we continue with the scheme in the regular fashion, until the next block is exhausted, and so on.

The solution itself is obtained exactly as before, by transferring the $-1/p_{k+1}(0)$ to the row of the q_k and then adding up the contributions of all the q_k.

We see that the block technique does not require essentially more work than the continuous technique, except that the total number of cycles needed for a certain accuracy is increased, compared with the continuous technique constantly corrected by reorthogonalization.

If the right side b_0 is changed to some other given vector c, then special precaution is necessary due to the fact that we do not possess now a universal orthogonal reference system which includes the entire space but each block provides its own partial reference system. We determine for the first block the μ_i according to (41) and then obtain the η_i by the recursions (42). But coming to the second block we have to replace c by the new vector $c^{(2)} = c - \Sigma \mu_i p_i$ and repeat the process of obtaining the μ_i and the η_i for the new block with this new vector. Then we reduce similarly $c^{(2)}$ to $c^{(3)}$ for the next block and so on.

The duality of the vectors p_i, q_i is mirrored by the duality of the two kinds of approximate solutions y_k and $\bar{y}_k$, defined by (11) and (8). The recurrence relations (13) permit us to establish recurrence relations between these two sets of solutions. We perform the operations (11) and (8) in (13), replacing x by A, and let these polynomials operate on b_0. This gives:

$$\begin{aligned} p_{k+1}(0)y_k &= p_k(0)\rho_k y_{k-1} - q_k \\ q_{k+1}(0)\bar{y}_k &= q_k(0)\sigma_k \bar{y}_{k-1} - p_{k+1}(0)y_k. \end{aligned} \tag{45}$$

We can simplify these relations by introducing the proportional vectors

$$v_k = \frac{p_{k+1}(0)}{\rho_0\rho_1 \cdots \rho_k} y_k = y_k \tag{46}$$

since from (26), $p_{k+1}(0) = \rho_0\rho_1 \cdots \rho_k$

$$\bar{v}_k = \frac{q_{k+1}(0)}{\sigma_0\sigma_1 \cdots \sigma_k} \bar{y}_k. \tag{47}$$

Hence we obtain

$$y_{k+1} = y_k - \frac{q_{k+1}}{\rho_0\rho_1 \cdots \rho_{k+1}} \tag{48}$$

$$\bar{v}_{k+1} = \bar{v}_k - \frac{\rho_0\rho_1 \cdots \rho_{k+1}}{\sigma_0\sigma_1 \cdots \sigma_{k+1}} y_{k+1}. \tag{49}$$

The recurrences (48) and (49) start with

$$y_0 = -\frac{q_0}{\rho_0}$$

$$\bar{v}_0 = -\frac{\rho_0}{\sigma_0} y_0 = \frac{q_0}{\sigma_0}. \tag{50}$$

The recursion (48) expresses our previous solution (35), (37) in slightly different form. However, an additional approximation is now provided by the scheme (49) which generates the $\bar{v}_k$ by a process analogous to that in (48). The vectors $\bar{v}_k$ are of value if we want a solution of smallest residual, since this solution is $\bar{y}_k$ and not y_k. After obtaining $\bar{v}_k$ by the scheme (49), we can also obtain y_k by multiplying by the constant $\sigma_0\sigma_1 \ldots \sigma_k/q_{k+1}(0)$.

The residual associated with $\bar{y}_k$ is given by

$$\bar{r}_{k+1}=b_0-A\bar{y}_k=\frac{q_{k+1}}{q_{k+1}(0)}, \tag{51}$$

and this is the absolutely smallest residual obtainable by k iterations. In the previous numerical example the length of the residual associated with y_1 is 1.9365, which is larger than the original length 1.7321 of the vector b_0. The length of r_1 associated with the solution $\bar{y}_1$, on the other hand, is

$$|\bar{r}_1|=\frac{|q_1|}{|q_1(0)|}=\frac{2}{3}\frac{\sqrt{15}}{2}=1.2910,$$

which is smaller than the original length.

The result is different, however, if we investigate the error of the *solution*, that is, $|y-y_k|$, rather than the magnitude of the residual, which is $|A(y-y_k)|$. The solution y_k has the property to minimize $(y-y_k)A(y-y_k)$ while the solution $\bar{y}_k$ minimizes $[A(y-\bar{y}_k)]^2$. The first quantity is less biased compared with the direct error square $|y-y_k|^2$ than the second. Hence y_k yields a smaller error in the solution, although a larger error in the residual than $\bar{y}_k$. To illustrate; in the numerical example the length of the vector $y-y_1$ is 1.884, while the length of the vector $y-\bar{y}_1$ is 3.0768. For this reason the vector $\bar{y}_k$ will usually be of smaller significance than the vector y_k.

5. The Preliminary Purification of the Vector b_0

In principle we have obtained a method for the solution of sets of linear equations which is simple and logical in structure. Yet from the numerical standpoint we must not overlook the danger of the possible accumulation of rounding errors. The theoretically demanded orthogonality of the vector set p_i can be quickly lost if we do not watch out for rounding errors. Now we can effectively counteract the damaging influence of rounding errors by constantly orthogonalizing every new p_i to all the previously obtained p_i. We do that by a correction scheme described in the earlier paper [14, p. 271, (60)].

This constant orthogonalization, however, is a lengthy process which basically destroys the simplicity of the generation of every new p_i and q_i by using only *two* of the earlier vectors. In order to make the corrections, *all* the previous p_i have to be constantly employed.

This consideration indicates that it will be advisable not to overstress our algorithm to too great a length. If by some means fast convergence can be enforced, the scheme might terminate in much fewer than n steps. Even if theoretically speaking the last vector vanishes exactly only after n iterations, it is quite possible that it may drop *practically* below negligible bounds after a relatively few iterations.

We can predict in advance, under what conditions we may expect fast convergence. If we want the scheme to terminate after less than n steps, it is necessary and sufficient that the vector b_0 shall be deficient in the direction of certain axes. The more "blind spots" the vector b_0 has in the direction of various principal axes, the quicker will the scheme terminate.

In the practical sense it will not be necessary that b_0 shall be *exactly* deficient in certain axes. It will suffice if the components of b_0 in the direction of certain principal axes are *small*. Strong convergence in this sense means that we shall reduce the components of b_0 in as many axes as possible.

That such a "purification" of b_0 of many of its components is actually possible, is shown by the Sylvester-Cayley procedure by which the largest eigenvalue and associated eigenvector of a matrix may be obtained [8, p. 134]. In principle *any* linear set of equations is solvable by the Sylvester-Cayley procedure. Indeed, let us homogenize the linear system (29) by completing the matrix G by an $n+1$st column defined as $-g$, and an $n+1$st row defined as identically zero. Then the linear eq (29) can now be formulated in the homogeneous form

$$G_1y_1=0, \tag{52}$$

where

$$G_1=G_0-g$$

$$y_1=a(y,1), \tag{53}$$

where a is any nonzero constant.

We now consider (52) as the solution of the following least-square problem. Minimize

$$(G_1y_1)^2 \tag{54}$$

under the auxiliary condition

$$y_1^2=1. \tag{55}$$

The solution of this minimum problem is the principal axis problem

$$A_1y_1-\lambda y_1=0, \tag{56}$$

where

$$A_1=G_1^*G_1. \tag{57}$$

We are particularly interested in the principal axis associated with the smallest eigenvalue

$$\lambda=0. \tag{58}$$

Let us now assume that we somehow estimated the largest eigenvalue λ_M of the nonnegative matrix A_1. Then the matrix

$$A_2=\lambda_M I-A_1 \tag{59}$$

is a new $n+1$-dimensional nonnegative matrix whose *largest* eigenvalue

$$\lambda=\lambda_M \tag{60}$$

corresponds to the zero eigenvalue of A_1.

Now the Sylvester-Cayley asymptotic method consists in choosing an arbitrary trial vector b_0 which has to satisfy the one condition that it shall not be deficient in the direction of the eigenvector connected with the largest eigenvalue λ_M. We now form the sequence

$$b^0=b_0, \quad b^1=A_2b_0, \quad b^2=A_2b^1=A_2^2b_0, \ldots$$

and obtain asymptotically

$$y_1 \underset{m\to\infty}{\to} b^m. \tag{61}$$

This method is of great theoretical importance, even if it often converges too slowly to be useful numerically. A proper refinement of the method, however, will make it well adapted to our present aims.

For the purpose of making the Sylvester-Cayley procedure more effective, let us analyze the problem in the reference system of the principal axes of the matrix A_2. Let us first normalize the largest eigenvalue to 1 by dividing A_2 by $\lambda_{M'}$. We thus want to operate with the matrix

$$A_0=\frac{A_2}{\lambda_M} \tag{62}$$

whose eigenvalues lie between 0 and 1.

In the reference system of the principal axes the trial vector b_0 shall have the components

$$\beta_{1\,0},\ \beta_{2\,0},\ \ldots,\ \beta_{n\,0}\ \beta_{n+1\,0}, \tag{63}$$

assuming that the eigenvalues λ_i are arranged according to increasing order on magnitude. The operation $b^m=A_0^mb_0$ generates the vector

$$\beta_{1\,0}\lambda_1^m,\ \beta_{2\,0}\lambda_2^m,\ \ldots,\ \beta_{n+1\,0}\lambda_{n+1}^m. \tag{64}$$

Now

$$\lambda_{n+1}=1,\ \lambda_i<1 \quad (i=1,2,\ldots,n). \tag{65}$$

Hence, as n grows to infinity, we get in the limit

$$\lambda_i^m\to 0 \quad (i=1,2,\ldots,n), \tag{66}$$

while λ_{n+1}^m remains constantly equal to 1. We thus get in the limit the vector

$$0,0,\ldots,0,\ \beta_{n+1\,0}, \tag{67}$$

which differs only by a factor of proportionality from the vector

$$0,0,\ldots,0,1. \tag{68}$$

This, however, is the principal axis associated with the largest eigenvalue $\lambda_{n+1}=1$.

While this method works very well in obliterating the small eigenvalues, it becomes very inefficient for a λ_i, which is near to 1.

Taking our lead from the Hamilton-Cayley procedure we will now approach the problem from a more general viewpoint. We go back to our original matrix A and the given right side b_0. Instead of considering a mere power b^m, we will consider an arbitrary *polynomial* $P_m(A)$ operating on b_0. For the sake of convenience we will once more introduce the reference system of the principal axes and we will once more normalize the largest eigenvalue of A to 1 by introducing the new matrix

$$A_0=\frac{1}{\lambda_M}A, \tag{69}$$

where λ_M is the largest eigenvalue of A.

Our aim is to solve the equation

$$A_0y=b^0, \tag{70}$$

where we have put

$$b^0=\frac{1}{\lambda_M}b_0. \tag{71}$$

Instead of the exact solution we consider an approximation $\bar{y}$ obtained by letting some polynomial $P_m(A_0)$ operate on b^0. This leads to a residual vector

$$r_{m+1}=[1-A_0P_m(A_0)]b^0 \tag{72}$$

and our aim is to reduce r_{m+1} to a small quantity.

Instead of $P_m(x)$ let us consider the polynomial of one higher order

$$F_{m+1}(x)=1-xP_m(x). \tag{73}$$

Apart from the boundary condition

$$F(0)=1 \tag{74}$$

$F(x)$ may be chosen as an arbitrary polynomial of the order $m+1$.

At this point we want to extablish a definite measure ϵ_{m+1} for the closeness of our approximation. We define ϵ_{m+1} as the ratio of the length of the residual vector r_{m+1} to the length of the correct solution y:

$$\epsilon_{m+1}=\frac{|r_{m+1}|}{|y|}. \tag{75}$$

Let us now discuss our problem in the reference system of the principal axes. The components of y in this system shall be denoted by

$$y_{1\,0},\quad y_{2\,0},\ \ldots,\ y_{n\,0}. \tag{76}$$

Then the components of b^0 become

$$b^0=\lambda_1y_{1\,0},\qquad \lambda_2y_{2\,0},\ \ldots,\ \lambda_ny_{n\,0}, \tag{77}$$

while the components of the vector r become

$$F(\lambda_1)\lambda_1 y_{10},\ \ldots,\ F(\lambda_n)\lambda_n y_{n0}. \tag{78}$$

Now by definition:

$$\epsilon^2=\frac{\sum_{k=1}^{n}[\lambda_k F(\lambda_k)]^2 y_{k0}^2}{\sum_{k=1}^{n} y_{k0}^2}, \tag{79}$$

and the theorem of weighted means gives the estimation

$$\epsilon\geq\max|\lambda_k F(\lambda_k)|. \tag{80}$$

Hence our aim must be to choose the polynomial $F(x)$ in such a fashion that the maxima of $xF(x)$ shall remain uniformly small in the interval between 0 and 1, which covers the entire range of the λ_i.

We make our choice as follows. We introduce the Chebyshev polynomials $T_n(x)$,[7] normalized to the range 0 to 1; [13, p. 140]. These polynomials are defined by [19, p. 3]

$$T_n(x)=\cos n\theta \tag{81}$$

with

$$x=\frac{1-\cos\theta}{2}=\sin^2\frac{\theta}{2}. \tag{82}$$

We now put

$$F_{m+1}(x)=\frac{1-T_{m+2}(x)}{2(m+2)^2x}=\frac{\sin^2(m+2)\frac{\theta}{2}}{(m+2)^2\sin^2\frac{\theta}{2}} \tag{83}$$

and notice that the quantity

$$xF_{m+1}(x) \tag{84}$$

is bounded by

$$\frac{1}{(m+2)^2} \tag{85}$$

throughout the range $0\leq x\leq 1$. Hence

$$\epsilon_{m+1}\leq\frac{1}{(m+2)^2}. \tag{86}$$

Since we have made our choice $F_{m+1}(x)$, the corresponding approximate solution

$$w_m=\frac{1-F_{m+1}(A_0)}{A_0}\,b^0 \tag{87}$$

is uniquely determined. We introduce the polynomials

$$g_m(x)=\frac{(m+2)^2}{4}\cdot\frac{1-F_{m+1}(x)}{x}=\frac{T_{m+2}(x)+2(m+2)^2x-1}{8x^2}, \tag{88}$$

which have integer coefficients. For the sake of convenience we list the first five $g_m(x)$ polynomials:

$$\begin{aligned}
g_0(x)&=1\\
g_1(x)&=6-4x\\
g_2(x)&=20-32x+16x^2\\
g_3(x)&=50-140x+160x^2-64x^3\\
g_4(x)&=105-448x+864x^2-768x^3+256x^4\\
g_5(x)&=196-1176x+3360x^2-4928x^3+3584x^4-1024x^5.
\end{aligned} \tag{89}$$

This table is actually not needed for the generation of the successive vectors $g_m(A_0)b^0$. We can obtain these vectors much more elegantly and with smaller rounding errors by a simple recursion scheme. We start out with the recursion formula of the Chebyshev polynomials, normalized to the range 0 to 1:

$$T_{m+1}(x)=2(1-2x)\,T_m(x)-T_{m-1}(x), \tag{90}$$

and obtain for the polynomials $g_m(x)$ the following recursion relation:

$$g_{m+1}(x)=2(1-2x)g_m(x)-g_{m-1}(x)+(m+2)^2 \tag{91}$$

starting with

$$\begin{aligned}
g_0(x)&=1\\
g_1(x)&=2(1-2x)+4=6-4x.
\end{aligned} \tag{92}$$

In order to utilize this relation for the generation of the vectors $g_m(A_0)b^0$, we introduce the matrix

$$\begin{aligned}
B&=2I-4A_0\\
&=2I-\frac{4}{\lambda_M}A
\end{aligned} \tag{93}$$

and obtain the generating scheme

$$g_{m+1}=Bg_m-g_{m-1}+(m+2)^2b^0 \tag{94}$$

starting with

$$\begin{aligned}
g_0&=b^0\\
g_1&=Bg_0+4b^0.
\end{aligned} \tag{95}$$

The last term of (94) can be absorbed in the simpler recursion formula

$$g_{m+1}=\bar{B}\bar{g}_m-g_{m-1} \tag{96}$$

if we agree to operate again with a surplus column similar to that used in our previous numerical example (*cf.* the bracketed column of the numerical scheme of section 3). We extend the matrix B by an $n+1$st

[7] The use of the Chebyshev polynomials for the solution of linear systems has been suggested at various times [4]. The author is not aware that the specific method here recommended has been suggested before.

column for which we choose the given right side b^0:

$$\bar{B}=B,\ b^0. \tag{97}$$

Similarly, we extend the vectors g_m by a surplus element, defined as the integer $(m+2)^2$:

$$\bar{g}_m=g_m,\ (m+2)^2. \tag{98}$$

The surplus column of the vector scheme g_m can be filled out in advance by the squares of the integers, starting with 4, 9, 16, . . ., in contrast to the bracketed column of the previous scheme which was filled out as the scheme unfolded itself. The surplus column of the matrix B and the surplus elements of the vectors g_m participate solely in the formation of the product $\bar{B}\bar{g}_m$, but have no effect on the subtraction of g_{m-1}, which is subtracted *without* its surplus element.

The definition of the $g_m(x)$ polynomials shows that the approximate solution (87) is in the following relation to the vectors g_m just generated

$$w_m=\frac{4}{(m+2)^2}\,g_m. \tag{99}$$

Moreover, if we want to find the residual vector associated with the solution w_m, we have to form

$$\begin{aligned} r_{m+1}&=b^0-A_0w_m \\ &=\frac{1}{(m+2)^2}\,[(m+2)^2b_0-4A_0g_m] \\ &=\frac{1}{(m+2)^2}\,(\bar{B}\bar{g}_m-2g_m). \end{aligned} \tag{100}$$

The last equation allows the following interpretation. Let us assume that at a certain m we want to terminate our process. We will now want to know how much the remaining residual is. For this purpose we merely add one more iteration according to (96), then the quantities required in (100) are available with the only modification that instead of subtracting g_{m-1} we subtract $2g_m$. This vector, divided by $(m+2)^2$, gives the residual r_{m+1}.

Numerical example. The following illustrative example is chosen to demonstrate the operation of the method. Our matrix A is once more the matrix of the numerical example of section 3. The right side is chosen as $b_0=0,\ 0,\ 0,\ 4$.

Estimation of the largest eigenvalue λ_M. The largest eigenvalue of a matrix can be estimated by the method of Geršgorin [9], (cf. also [3] and [20]). Even if this estimation is not always very close, it gives a definite upper bound for λ_M by a very simple test. Such an estimate is what we need since an overestimation of λ_M merely makes the largest eigenvalue smaller than 1. The only thing we have to avoid is a λ_M larger than 1, because then we would overstep the region where the Chebyshev polynomials are bounded by unity in absolute value.

The method of Geršgorin, restricted for our case to the estimation of the largest eigenvalue, is based on the definition of the eigenvectors of a matrix A by the equations

$$\sum_{\alpha=1}^{n} a_{i\alpha}x_\alpha=\lambda x_i. \tag{101}$$

We consider only *one* equation of the given set, picking out that particular index i which belongs to the absolutely largest x_i. We now divide by x_i on both sides of the equation. Since $|x_\alpha/x_i|\leq 1$, we find at once

$$|\lambda|\leq\sum_{\alpha=1}^{n}|a_{i\alpha}|. \tag{102}$$

Hence the absolute value of our chosen λ is smaller than the sum of the elements of some row (or column). Now we can evaluate the sum of the absolute values of all the elements for *each* row (or column) and select the maximum of this sequence of m numbers. Then we know that for *any* λ_i the absolute value of λ_i cannot surpass this sum. We thus obtain the estimate

$$\lambda_M\leq S_M, \tag{103}$$

where S_M is the maximum among the sums of the absolute values of all the elements of the rows 1, 2, . . ., n.

It was pointed out before that the actual generation of the symmetrized matrix $A=G^*G$, which is a numerically heavy load, is not demanded since all our operations can be performed with the help of G and G^* alone. But then it becomes necessary to estimate the largest eigenvalue of A by utilizing G and G^* only, without generating the elements of A.

We assume the general case that G has arbitrary complex elements and conceive G as the sum of two Hermitian matrices G' and G'', defined by

$$\begin{aligned} G'&=\frac{1}{2}\,(G+\tilde{G}^*) \\ G''&=\frac{i}{-2}\,(G-\tilde{G}^*) \end{aligned} \tag{104}$$

(the symbol $\sim$ means conjugate complex). Then

$$\begin{aligned} G&=G'+iG'' \\ \tilde{G}^*&=G'-iG''. \end{aligned} \tag{105}$$

Hence

$$\tilde{G}^*G=(G')^2+(G'')^2+i(G'G''-G''G'). \tag{106}$$

Now the largest eigenvalue of a positive definite Hermitian matrix A can be defined as the largest possible length of any vector Ab_0, where $|b_0|=1$. In order to find this largest length, we let the eq (106) operate on b_0. We thus obtain the estimate

$$\lambda_M\leq\lambda_M'^2+\lambda_M''^2+\lambda_M'\lambda_M''+\lambda_M''\lambda_M',$$

or

$$\lambda_M\leq(\lambda_M'+\lambda_M'')^2, \tag{107}$$

where λ_M' is the largest eigenvalue of G' and λ_M'' the largest eigenvalue of G''. Since λ_M' and λ_M'' can be estimated by Geršgorin's theorem, we thus obtain an upper bound for λ_M, without using the elements of the least-squared matrix A.

In our simple numerical example the given matrix is already symmetric and positive definite. We can thus operate directly with A. The sums of the absolute values of each row are 3, 4, 4, 3. Hence we can choose $\lambda_M=4$ as a safe estimate of the largest eigenvalue.

We construct the matrix B according to (93), and extend it by the column $b_0=\frac{1}{4}b_0=0,\ 0,\ 0,\ 1$. We choose $m=5$, and continue the scheme by one more row to obtain the new residual. The factor $(m+2)^2$ is in our case 49. Hence the fifth row has to be multiplied by 4/49 in order to obtain the approximate solution w_5, while the sixth row s_6 has to be multiplied by 1/49 in order to get the residual r_6.

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad \begin{matrix} 0 \\ 0 \\ 0 \\ 1 \end{matrix}$$

g_0:	0	0	0	1	4
g_1:	0	0	1	4	9
g_2:	0	1	4	9	16
g_3:	1	4	9	16	25
g_4:	4	9	16	25	36
g_5:	8	16	25	36	49
s_6:	0	1	2	2	

The last row was obtained by multiplying the row 5 by the matrix B and then subtracting the row 5 (and not 4) twice.

We can test the residual estimate (86) on our scheme. According to this estimate $(m+2)^2\epsilon_{m+1}$ must become smaller than y_{m+1}. If the vector g_m, multiplied by $4/(m+2)^2$ is a fairly good approximation of y, then the length of the vector s_{m+1} cannot surpass $4/(m+2)^2$ of the length of g_m. In our case $(m=5)$: $|g_5|=46.34$, while $|s_6|=2.24$. Hence

$$\frac{|s_6|}{|g_5|}=0.048<\frac{4}{7^2}=0.081633. \tag{108}$$

If this test fails, it is an indication that our approximation is far from the correct solution, caused by the influence of the small eigenvalues, as we will show presently.

The approximation w_5 is obtained by multiplying row 5 by $\frac{4}{49}=0.081633$. This gives $w_5=0.65306$, 1.30613, 2.04082, 2.93879.

The correct solution is $y=0.8,\ 1.6,\ 2.4,\ 3.2$.

What did we accomplish with this algorithm? Let us analyze the situation in the reference system of the principal axes. Let us plot the eigenvalues λ_i, normalized to the range 0 to 1, along the abscissa, while we plot the components of the right side b^0, associated with a certain λ_i, as ordinates. In the language of physics we have a "line spectrum" since only certain definite "frequencies" λ_i, namely, the eigenvalues of A, are represented.

Whatever approximation scheme we may use, based on iterations, we will always obtain a preliminary solution y_{k+1}, which does not satisfy the equation exactly but generates a new right side in the form of a residual vector r_{k+1}. Hence quite generally, for any iterative solution we will have

$$y_k=P_k(A_0)b^0, \tag{109}$$

where $P_k(x)$ is some polynominal in x. Then the residual r_{k+1}, associated with this solution, becomes

$$r_{k+1}=[1-A_0P_k(A_0)]b^0=F_{k+1}(A_0)b^0. \tag{110}$$

This residual vector is then the new "right side" of the next approximation.

The result of our approximation can now be described as follows, if we view everything from the reference system of the principal axes. The original component $b_{i\,0}$, associated with the eigenvalue λ_i, became attenuated by the factor $\tau(\lambda_i)$ where the function $\tau(x)$ is defined by

$$\tau(x)=F_{k+1}(x). \tag{111}$$

In these discussions we have considered two kinds of approximations: the purification technique dealt with in the present section, and the method of minimized iterations, discussed before. Since the purification *precedes* the application of the algorithm technique given in section 3, let us call it algorithm I, while the algorithm of section 3 shall be called algorithm II. The attenuation obtained by these two kinds of algorithms is based on two very different principles. We discuss the algorithm I first.

Here we get according to (83):

$$\tau(x)=\frac{\sin^2(m+2)\frac{\theta}{2}}{(m+2)^2\sin^2\frac{\theta}{2}} \tag{112}$$

with

$$x=\sin^2\frac{\theta}{2}. \tag{113}$$

The attenuation thus obtained starts with 1 and falls off with $1/x$. The factor $\tau(x)$ cuts out effectively the higher frequencies but has little influence on the small frequencies (small λ_i). What we accomplish here is

that we put the spotlight on the small eigenvalues, while the large eigenvalues can be eliminated to any desired degree.

Actually this algorithm serves a double purpose. We limit the field of vision to a relatively narrow band of small eigenvalues. Aside from that, however, we can make the *focusing effect* of the process increasingly sharper. Let us limit ourselves to the case $m=5$, that is, to five iterations of the type described. We can now take the residual r^6 and repeat the process, thus obtaining a second "block" of five iterations The attenuation factor achieved as the result of the two blocks of iterations is the *square* of the previous $\tau(\lambda)$. Generally, if the process is repeated k times, the attenuation thus obtained is characterized by

$$\tau^{(k)}(\lambda)=[\tau(\lambda)]^k.$$

Figure 1 plots $\tau(\lambda)$, (for $m=5$) and the second, third, and fourth powers of $\tau(\lambda)$. If our matrix A contains a very small eigenvalue of the order of 0.0001 say, this very small eigenvalue will not be able to compete with the larger eigenvalues, except if the larger eigenvalues are blotted out *very strongly*. At first sight we might think that from the standpoint of such a small λ_i it makes no great difference how often we repeated the process since it will remain in the illuminated part of the spectrum for a practically unlimited time even if k is large. However, the situation is quite different if the algorithm I is conceived as a mere preparation to algorithm II. Then we are reconciled to the fact that our first efforts are unable to take out the contribution of that small eigenvalue. We leave that task to the second algorithm. But that second algorithm will operate much more satisfactorily if the large eigenvalues are eliminated with *great accuracy*. Hence the advantage of continuing the first algorithm to several blocks is not so much the increased accuracy

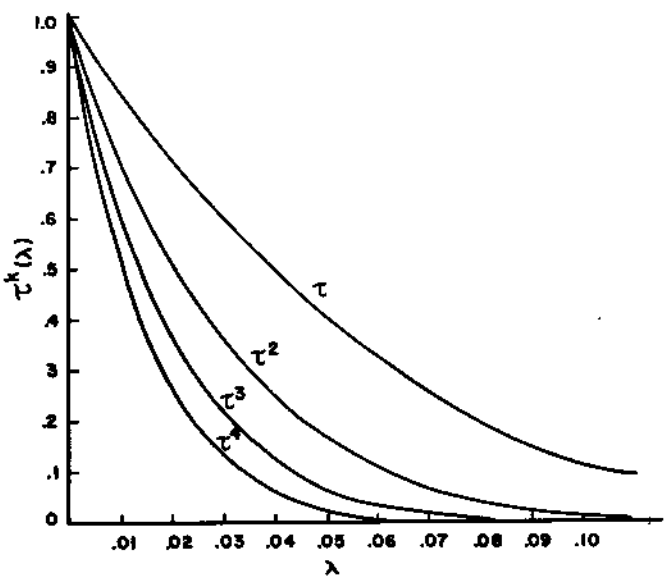

FIGURE 1. *Attenuation factors obtained by k blocks of algorithm I.*

of the solution as the proper preparation for the second process, which will then tackle the problem of small eigenvalues much more effectively. The field of vision is perhaps not much reduced. But the dim light that still spreads over the higher portion of the spectrum is more and more sharply eliminated.

The continuation of the g-algorithm to a second block can be achieved without any basic interruption of the operations. After obtaining the residual r_6, we transfer this row to $\overline{B}$ as an additional sixth column. The fifth column now remains inactive. Consequently, the squares 4, 9, 16, . . . are now moved over by one column. The resulting scheme, now extended to two blocks, and omitting the first five lines which have been obtained before, looks as follows:

	$\begin{pmatrix}0&1&0&0\\1&0&1&0\\0&1&0&1\\0&0&1&0\end{pmatrix}$				0	0
					0	1
					0	2
					1	2
g_5 :	. . .	. . .	. . .	. . .		. . .
$s_6=g_6^{(2)}$:	0	1	2	2		4
$g_1^{(2)}$:	1	6	11	10		9
$g_2^{(2)}$:	6	20	32	27		16
$g_3^{(2)}$:	19	48	68	54		25
$g_4^{(2)}$:	42	92	120	91		36
$g_5^{(2)}$:	73	150	187	138		49
$s_6^{(2)}=g_5^{(3)}$:	4	9	12	9		

The successive blocks can be generated continuously by one mechanized algorithm. If k blocks are generated, the approximation becomes

$$w^{(k)}=4\left(\frac{g_5^{(1)}}{49}+\frac{g_5^{(2)}}{49^2}+\frac{g_5^{(3)}}{49^3}+\ldots+\frac{g_5^{(k)}}{49^k}\right).$$

In our numerical example the two contributions and their sum becomes:

$\frac{4}{49}g_5^{(1)}$	=	0.65306	1.30613	2.04082	2.93879
$\frac{4}{49^2}g_5^{(2)}$	=	0.12162	0.24990	0.31154	0.22990
$w^{(2)}$	=	0.77468	1.55603	2.35236	3.16869
$(y$	=	0.8	1.6	2.4	3.2) .

If we perform tha ratio test (108) once more on the second block, we find

$$\frac{|s_6^{(2)}|}{|g_5^{(2)}|}=\frac{17.944}{286.08}=0.0627<0.0816.$$

Hence the inequality (86), multiplied by the factor 4, can still be verified. We can expect that, as we come to higher and higher blocks, the ratio test will eventually fail. The initial vectors of the successive blocks become more and more purified of the larger eigenvalues. As a consequence, the purification process, which leaves the very small eigenvalues untouched, becomes less and less effective. Eventually the polynomial $g_m(A)$ will operate on an initial vector $b_0^{(k)}$, which contains only small eigenvalues. We will then approach the extreme case

$$g_m(A)b_0^{(k)} = g_m(0) \cdot b_0^{(k)} = \frac{(m+2)^2[(m+2)^2-1]}{12} b_0^{(k)},$$

while s_{m+1} approaches $(m+2)^2 b_0^{(k)}$. The ratio test then gives

$$\frac{|s_{m+1}|}{|g_m|} = \frac{12}{(m+2)^2-1}$$

that is, 1/4, if $m=5$. This gives an upper bound for the ratio test, which cannot be surpassed, no matter how far the process is continued.

We now come to the analysis of the $\tau(\lambda)$-factor connected with algorithm II (see fig. 2). The principle by which this process gives good attenuation, is quite different from the previous one. Here we take heed of the specific nature of the matrix A and operate in a selective way. The polynomials $F_{m+1}(\lambda)$ of this process have the peculiarity that they attenuate due to the nearness of their zeros to those λ-values which are present in A. These polynomials take advantage of the fact that the spectrum to be attenuated is a line spectrum and not a continuous spectrum. They work efficiently in the neighborhood of the λ_i of the matrix but not for intermediate values. They are thus associated with the given specific matrix A and are of no use for other matrices. If we proceed to the polynomial of nth order $F_n(\lambda)$, the zeros of this polynomial hit all the λ_i exactly, and thus make the entire residual vanish.

This analysis explains the advantages and the disadvantages of the second algorithm. The advantage of the process is its great economy. The exact solution (apart from rounding errors) is obtainable in n iterations; this is the minimum number of steps for generating a polynomial that will have its zeros at the λ_i of the matrix A. If the number of components present in b^0 is smaller than n, then the order of $F_m(\lambda)$ is correspondingly lower and the solution is again obtained in the minimum number of steps.

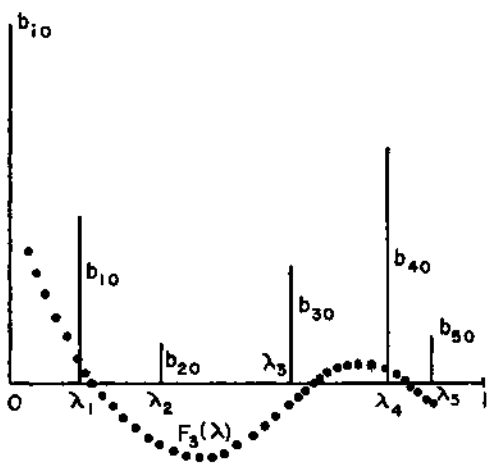

FIGURE 2. *Attenuation behavior of algorithm II.*

The price we have to pay is that the successive iterations of this process are more complicated than those of algorithm I. Instead of *one* new vector, a *pair* of vectors has to be generated. Moreover, the previous recurrence relation, based on the properties of the g-polynomials, had *fixed* coefficients, which needed no adjustments throughout the procedure. Here at every step a pair of scalars have to be evaluated which are needed for the generation of the new p, q vectors. The constants of the recurrence relations have to be readjusted at each new step of the process.

Another difficulty arises from the inevitable accumulation of rounding errors. If we want to maintain a long chain of interlocked operations, we have to counteract the effect of rounding errors. This can be done by constant reorthogonalization of the p vectors which, however, is a lengthly process. It is preferable not to correct for the rounding errors but avoid them by breaking the long algorithm into a sequence of shorter blocks. Then, however, we lose in convergence and the number of iterations has to be extended.

The two algorithms together complement each other. The first algorithm succeeds in purifying the given vector b^0 of all its large eigenvalues. The spectrum is thus effectively reduced which means that only a relatively small number of λ_i remain practically present in the final residual. This is now the point where the second algorithm takes over. Because of the small number of eigenvectors still present in b^0, a polynomial of low order will be sufficient for the final elimination of the residual. The process has thus good convergence and will be finished after a small number of iterations. The breaking up of the process into blocks will not be necessary since the rounding errors will have no time to accumulate to the point where they endanger the solution. The small extension of the spectrum tends to reduce the deorthogonalizing effect of the rounding errors, thus increasing the length of a block and preventing its premature termination. The opening of a second block will thus but seldom be required.

6. Iterative Solution of Nearly Singular Systems

In practical numerical work we frequently encounter nearly singular systems. We shall therefore discuss the relative merits of iterative schemes and other matrix inversion methods with respect to such systems.

We begin with the extreme case when the determinant of the matrix G and all its minors up to a certain order $n-\nu$ vanish *exactly*, thus reducing the

rank of the matrix to $n-\nu$. In this case the linear system (29) is generally not solvable, except if the right side satisfies certain compatibility conditions. The reduction of the rank from n to $n-\nu$ means that the left side of the system satisfies ν independent linear identities. The compatibility of the system, which is the necessary and sufficient condition for its solvability, demands that the same identities shall be satisfied by the given right sides.

If the compatibility conditions are actually satisfied and the system thus solvable, then another peculiarity arises. The solution is *not unique*. To any given solution an arbitrary linear combination of ν independent vectors may be added without disturbing the validity of the equations.

These theoretical conditions have to be translated into practical conditions if we want to analyze the numerical behavior of linear systems which are not exactly but nearly singular. We can base our analysis on the behavior of the eigenvalues and eigenvectors associated with the matrix G.

In the light of eigenvalues the lowering of the rank of the matrix G from n to $n-\nu$ means that the matrix G possesses ν vanishing eigenvalues. Such a matrix operates in an $n-\nu$-dimensional subspace only and blots out all the ν dimensions which are orthogonal to this subspace. Hence the linear set (29) can only be solvable if the right side g is free of all those dimensions which the matrix rejects. At the same time, the solution y may contain any vector which belongs totally to the rejected portion of the n-dimensional space, since the operation Gy extinguishes this vector and thus does not disturb the balance of the equation.

If the matrix G is not exactly but nearly singular in ν directions, this means that ν of the eigenvalues, although not exactly zero, are nevertheless very small compared with the other eigenvalues. We can associate such a matrix geometrically with a strongly skew-angular frame of reference which almost collapses into a lower dimensional space. In this interpretation we conceive the successive columns of G as n basic vectors

$$V_1, \quad V_2, \quad \ldots, \quad V_n, \tag{114}$$

which establish an n-dimensional set of axes. The linear system $Gx=g$ now assumes the following significance:

$$V_1x_1+V_2x_2+\ldots+V_nx_n=g. \tag{115}$$

This means that the given vector g shall be analyzed in the reference system of the base vectors V_i.

Now the skew-angular character of a frame of axes can be properly described by evaluating the volume included by these axes. This again is nothing but the determinant $|G|$ of the matrix G. The smaller the included volume, the more skew-angular is the system. However, this measure is adequate only if the various axes of our reference system are *properly scaled*. Otherwise even an orthogonal set of axes can have a very small determinant, caused not by the inclination of the axes, but by uneven scaling.

This uneven scaling can always be eliminated by the following linear transformation of the variables x_i:

$$x_i=\frac{|g|}{|V_i|}\,y_i. \tag{116}$$

Then the original equation (115) now appears in the following form

$$U_1y_1+U_2y_2+\ldots+U_ny_n=g_0, \tag{117}$$

where

$$U_i=\frac{V_i}{|V_i|}, \quad g_0=\frac{g}{|g|} \tag{118}$$

and thus

$$|U_i|=1, \quad |g_0|=1. \tag{119}$$

In matrix language the transformation (116) means that the columns of the matrix $G=(g_{\alpha i})$ are multiplied by [8]

$$\gamma_i=\frac{1}{\sqrt{\sum_{\alpha=1}^{n} g_{\alpha i}^2}} \tag{120}$$

and the right side by

$$\gamma_{n+1}=\frac{1}{\sqrt{\sum_{\alpha=1}^{n} g_{\alpha}^2}} \tag{121}$$

which transforms the vector x into

$$y_i=\frac{\gamma_i}{\gamma_{n+1}}\,x_i. \tag{122}$$

The consequence of this transformation on the symmetrized matrix A is that all the diagonal elements become 1, while all the nondiagonal elements range between ± 1. This is of great advantage from the viewpoint of numerical operations [15].

If the original matrix is already given as a positive definite, symmetric matrix A, then the scaling of the matrix is performed by the transformation

$$x_i=\frac{1}{\sqrt{a_{ii}}}\,\xi_i. \tag{123}$$

We multiply all the rows, and then all the columns by $1/\sqrt{a_{ii}}$, which makes the resulting diagonal elements once more equal to 1. Moreover, the vector g is transformed into the vector b by the transformation

$$b_i=\frac{g_i}{\sqrt{a_{ii}}}. \tag{124}$$

Finally, the length of this vector is normalized to 1 by putting

$$\xi_i=|b|y_i \tag{125}$$

$$b_0=\frac{b}{|b|}. \tag{126}$$

[8] The conditions (120) and (121) need not be met with any high degree of precision. The multipliers γ_i can be rounded off to two significant figures.

We now consider the vector equation (117). The smallness of the determinant $|G|$ associated with the rescaled system now actually measures the strongly skew-angular nature of our reference system. Nevertheless, the linear equation (117) can be considered as well adjusted if the right side g_0 falls inside the narrow space included by the basic vectors U_i. This condition is a natural counterpart of the compatibility conditions set up for the case that the vectors eventually collapse completely into a lower dimensional space. If the right side lies constantly inside the space included by the basic vectors, then it remains coplanar with those vectors even in the limit when the vectors do not include any finite volume any more. Practical compatibility includes thus the limiting case of theoretical compatibility. Let us examine, in what form this condition of "insidedness" comes into evidence in relation to the least-squared matrix A and its right side b_0. Let us project the vector b_0 on the principal axes of A. We obtain the components β_{i0}. Let us divide each one of these components by the eigenvalue λ_i associated with that axis. This gives the sequence

$$\frac{\beta_{10}}{\lambda_1}, \frac{\beta_{20}}{\lambda_2}, \ldots, \frac{\beta_{n0}}{\lambda_n}. \tag{127}$$

We pick out the absolutely largest of these numbers and consider

$$\mu = \max \left| \frac{\beta_{i0}}{\lambda_i} \right| \tag{128}$$

as the measure of the adjustment of the given system. No matter how small the determinant of A is, the linear equation $Ay = b$ can be considered as solvable practically if μ is a reasonably small number. The measure μ does not refer in any way to the condition of A itself. It measures the relation of the right side of the system to the left side. The meaning of a reasonably small μ is that the near identities which exist on the left side, lead to near identities also on the right side.

As a consequence of (117) we have

$$|\beta_{i0}| \leq \mu \lambda_i. \tag{129}$$

Let us collapse the given frame of axes more and more into a lower dimensional system, but keep μ bounded. Then in the limit a certain number ν of λ_i vanish. However, as a consequence of (129), the corresponding β_{i0} vanish too. This is exactly the compatibility requirement of a singular system. The measure μ is thus a reasonable measure of the adjustment of the given linear system.

If we are able to invert a matrix exactly, then the smallness or largeness of μ is of no importance. If, however, approximation techniques are employed, then it is natural to restrict ourselves to well adjusted systems whose μ is not too large. We cannot expect that *any* approximation procedure shall remain successful if μ becomes arbitrarily large, since in that case a minute change in the right side may cause a large error in the solution. For the same reason we can add at once that *physical* systems, whose right sides are given as the result of observations, must satisfy the condition of not too large μ, in order to allow any valid conclusions.

We will thus restrict ourselves to the solution of systems that can be considered as "well adjusted" in the sense of prescribing for μ a not too large upper bound. The length of our approximation procedure will depend on the magnitude of μ. If μ is too large, then we have to abandon the use of iteration techniques, or we have to employ the full technique of minimized iterations with all its precautions, continuing to the very end of n iterations.

Singular systems, however, show a second peculiarity, namely, the indeterminate character of the solution. Let us examine what the corresponding phenomenon is in the case of nearly singular, that is, strongly skew-angular systems. The corresponding phenomenon is that very small changes on the right side cause much larger changes in the solution. The danger exists solely in the direction of the small eigenvalues, and is caused by the fact that the component β_{i0} of the right side in the direction of the ith eigenvector has to be divided by λ_i in order to get y_{i0}.

This phenomenon is of considerable significance if we are interested in the solution of linear systems which arise from physical measurements. Let us assume that we know in advance from physical reasons that the given system is well adjusted, that is, that μ is reasonably small, compared with the accuracy of the measurements. Then an appearance of a large y_{i0} on account of dividing by a small λ_i must be caused by experimental errors and should be discarded. In such a situation the use of an iteration technique for finding the solution is superior to the exact solution. The exact solution, obtained by matrix inversion, would be of little help, since it would not separate the influence of the errors in the direction of the small λ_i. On the other hand, if we use the above advocated method of taking out first the contribution of the large eigenvalues by the g-polynomials, then we can actually separate the desirable part of the solution from the undesirable part. The first approximation, which leaves the small eigenvalues practically untouched, does not offer any difficulty and can stand as it is. Now we come to the second algorithm, which determines the contribution of the small eigenvalues. If in this successive approximation process a correction appears, the length of which is more than μ times the length of the remaining residual, we know that we should stop at this point, since this contribution comes from the errors of the data.

This analysis indicates that in the case of strongly skew-angular but well-adjusted physical systems the separation of the two algorithms has more than technical significance. It makes smoothing of the data possible by discarding large errors in the solution caused by small observational errors in the direction of the small eigenvectors.[9] The iteration technique gives in such a case a more adequate solution than the mathematically exact solution obtained by matrix

[9] The expression "small eigenvector" is used in the sense of "an eigenvector associated with a small eigenvalue."

inversion because it capitalizes on the sluggishness with which the small eigenvalues come into play. The smallest eigenvalues, which essentially test the compatibility of the system, appear last. Now the given system is such that this test of compatibility is not needed since we know in advance from physical considerations that the system is well adjusted. By omitting the contents of the last equations we take advantage of the good part of our measurements and reject the errors. While the uncertainty of the result is not completely eliminated by this procedure, it is nevertheless essentially reduced in magnitude.

7. Eigenvalue Analysis

The underlying principles of the two algorithms discussed in the previous sections can also be employed in the problem of finding the eigenvalues and eigenvectors of a matrix. The general p, q, p^*, q^* algorithm gives a complete analysis of the matrix, namely it gives all its eigenvalues and eigenvectors. If performed with the proper care, this method gives satisfactory results even when the eigenvalues are closely grouped [16].

However, in many situations we are not interested in the *complete* set of eigenvalues and eigenvectors. We would welcome a technique which puts the spotlight on a *few* eigenvectors only, or we might want to single out just *one* particular eigenvalue and its associated eigenvector, for example, the smallest one. The method now to be outlined should prove useful in connection with such problems.

The preliminary purification of b_0 served the purpose of increasing the convergence of the final algorithm by properly preparing the vector on which it operates. We were able to effectively eliminate all components of the original vector except those associated with the small eigenvalues.

After the purification, the spotlight is put on the small eigenvalues; we will therefore first obtain the small eigenvalues and the associated eigenvectors with great accuracy, in marked contrast to the Sylvester-Cayley asymptotic procedure which first obtains the absolutely *largest* eigenvalue and its associated eigenvector.

In "flutter" problems we are usually interested in the *smallest* eigenvalues of the given matrix. In order to apply the asymptotic power method, we first *invert* the matrix, thus transforming the smallest eigenvalues to the largest eigenvalues of the new matrix. If we possess a direct method for the evaluation of the smallest eigenvalues, we might dispense with the preliminary inversion of the matrix, thus saving a great deal in numerical effort.

However, our previous purification procedure, based on the properties of the Chebyshev polynomials, is strictly limited to nonnegative matrices and cannot be generalized to arbitrary complex eigenvalues, because the outstanding properties of the Chebyshev polynomials are not preserved in the complex range. We will now see that the general eigenvalue problem of an arbitrary complex matrix can always be formulated in such a way that it becomes transformed into the determination of the smallest eigenvalue and eigenvector of a nonnegative Hermitian matrix.

Let us first observe that all our previous procedures remain valid if we apply them to a nonnegative Hermitian matrix

$$A^*=\tilde{A} \tag{130}$$

where A^* is the transpose and $\tilde{A}$ is the conjugate of A. The quadratic form associated with a Hermitian matrix is still real.

We consider the solution of the linear equation

$$Gy=g \tag{131}$$

where the matrix G is a general matrix with complex elements; the vector g has likewise complex elements. We multiply on both sides by $\tilde{G}^*$ and obtain once more the standard form

$$Ay=b \tag{132}$$

with

$$A=\tilde{G}^*\,G \tag{133}$$

and

$$b=\tilde{G}^*g. \tag{134}$$

The matrix A defined by (133) is not only Hermitian but also nonnegative.

All the characteristic features of the previous algorithms remain the same. The largest eigenvalue λ_M can once more be estimated by Geršgorin's theorem. The g-algorithm carries over without any modification, although all the vectors involved have now complex elements.

The p, q algorithm can also be carried over with the only modification that the adjoint vectors p^*, q^* are now not identical with p, q but with $\tilde{p}$, $\tilde{q}$. Hence the basic scalars h_i and h_i' of the algorithm have to be defined as follows:

$$\begin{aligned} h_i&=\tilde{p}_i p_i \\ h_i'&=\tilde{p}_i q_i'=p_i\tilde{q}_i' \end{aligned} \tag{135}$$

We see from these relations that the h_i are again all positive; moreover, the h_i' are all real. Actually, the theory of the basic algorithm [14], section 6, allows a further conclusion. The significance of the h_i and h_i' within the framework of this algorithm reveals that for nonnegative Hermitian matrices not only the h_i but also the h_i' remain *positive*. Hence, in spite of the complex nature of the vector elements, the reality (and even positiveness) of the basic scalars remains preserved.

Let us now consider the eigenvalue problem connected with an arbitrary nonsymmetric and complex matrix K:

$$(K-\lambda I)\,y=0. \tag{136}$$

We put

$$G=K-\lambda I, \tag{137}$$

and write the equation

$$Gy=0 \tag{138}$$

in the "least square" form

$$\tilde{G}^*Gy=0. \tag{139}$$

This introduces the Hermitian matrix

$$A=\tilde{G}^*G=\tilde{K}^*K-(\tilde{\lambda}K+\lambda\tilde{K}^*)+\lambda\tilde{\lambda}I. \tag{140}$$

There is generally no predictable relation between the eigenvalues of an arbitrary matrix and its "least-square" form. Yet there is one exception, namely the eigenvalue zero. The eigenvalue zero of G carries over to the Hermitian matrix A. Let us now assume that we want to operate solely with the Hermitian matrix A and abandon the original matrix K completely. Then we can still obtain all the eigenvalues of K by determining all those values of λ in (140), which make the smallest eigenvalue of A equal to zero.

We now see how we can make good use of a method which discriminates in favor of the *small* eigenvalues. Such a method can be utilized to put the emphasis on one particular eigenvector, instead of an arbitrary mixture of eigenvectors.

Generally, if we start the p, q algorithm with some arbitrary b_0, b_0^* vector, we have no control over the sequence in which the successive eigenvectors and eigenvalues will be approximated. The particular eigenvector in question might appear quite late in the process. Let us assume, however, that we succeed in purifying the trial vector b_0, b_0^* of most of its components and emphasize strongly one particular eigenvector in which we are interested.

Such conditions actually arise if we possess a first approximation λ_0 to the desired eigenvalue λ. We can now form the Hermitian matrix (140) with this particular $\lambda=\lambda_0$ and let us assume that we can obtain its smallest eigenvector. If λ_0 were the correct value for λ, the smallest eigenvalue would be zero and the associated eigenvector the correct solution. Since λ_0 is only an approximation, we still get a good vector which has a strong component in the desired direction. This is enough for a good start of the algorithm II.

However, our work is only half done. Since the original matrix is not symmetric, we need the complete p, q, p^*, q^* process. That process starts with b_0 and the adjoint b_0^*. So far we have obtained b_0 only. In order to obtain a well-suited b_0^*, we proceed as follows. We consider the adjoint solution

$$(K^*-\lambda I)y^*=0, \tag{141}$$

which in "least-square" form leads to the new matrix

$$\bar{A}=\tilde{G}G^*=\tilde{K}K^*-(\tilde{\lambda}K^*+\lambda\tilde{K})+\lambda\tilde{\lambda}I. \tag{142}$$

The third part of this matrix is identical with the previous third part; the second part differs from the previous second part only in the change of i to $-i$. The first part, however, is an entirely independent new matrix, formed by multiplying the *rows* of K by its rows, while previously the *columns* were multiplied by columns.

The smallest eigenvector of this new Hermitian matrix $\bar{A}$ can now be introduced as a well-purified b_0^* which will strongly emphasize the desired eigenvector. Then two steps of the p, q algorithm will give an improved eigenvector and a much improved value for λ. This method resembles Newton's method of obtaining the root of an algebraic equation if a near root is given.

The problem is thus reduced to the problem of finding the smallest eigenvector of a Hermitian matrix. Our aim is to purify a trial vector b_0 of all its large eigenvalues, reducing it to a new vector in which the smallest eigenvector is strongly emphasized.

This was accomplished before in form of the residual of the previous g-process. There the attenuation obtained was characterized by the kth power of a certain function $\tau(x)$, if k blocks of the process were employed. As figure 1 illustrates, increasingly strong attenuations are obtainable even with a few blocks of five iterations. Since in our case the solution y is of no importance but only the residual, we can generate that residual immediately by utilizing the $F_{m+1}(x)$ polynomials. We multiply by $(m+2)^2$, in order to get integer coefficients. Hence we want to operate with the polynomials

$$f_{m+1}(x)=(m+2)^2F_{m+1}(x). \tag{143}$$

These polynomials once more satisfy a simple recurrence relation:

$$f_{m+1}(x)=2(1-2x)f_m(x)-f_{m-1}(x)+2, \tag{144}$$

which again leads to the previous algorithm

$$f_{m+1}=\bar{B}\bar{f}_m-f_{m-1} \tag{145}$$

with the only difference that the surplus column of the vectors f_m now remains 2 throughout the process:

$$\bar{f}_m=f_m,\ 2. \tag{146}$$

The matrix $\bar{B}$ is once more defined as before, see (93) and (97).

The termination of a block and changing over to the next block now occurs by the following simple procedure. We go on uninterruptedly with the recurrences, until the last vector f_{m+1} is reached. This vector is transferred to $\bar{B}$ as the new surplus column which will be in operation during the second block. Moreover, the last vector f_{m+1} becomes the initial vector $f_0^{(2)}$ of the second block. Then the algorithm starts over again until the new block is finished which occurs at $f_{m+1}^{(2)}$, and so on.

In order to demonstrate the operation of this algorithm, we once more make use of the previous simple matrix of fourth order and choose once more

$m=5$. Two blocks of six iterations are used in accordance with our previous g-algorithm, but now generating directly the residuals. As trial vector we could use the vector 1, 1, 1, 1. However, in order not to capitalize unduly on the symmetry of our highly simplified matrix, the trial vector is chosen as 1, 1, 1, 0. The resulting work scheme looks as follows:

	0	1	0	0	1	5
	1	0	1	0	1	7
	0	1	0	1	1	6
	0	0	1	0	0	3
f_0 :	1	1	1	0	2	
f_1 :	3	4	3	1	2	
f_2 :	5	7	6	3	2	
f_3 :	6	9	9	5	2	
f_4 :	6	10	10	6	2	
f_5 :	6	9	9	5	2	
f_6 :	5	7	6	3		
$f_0^{(2)}$:	5	7	6	3		2
$f_1^{(2)}$:	17	25	22	12		2
$f_2^{(2)}$:	35	53	49	28		2
$f_3^{(2)}$:	46	73	71	43		2
$f_4^{(2)}$:	48	78	79	49		2
$f_5^{(2)}$:	42	68	68	42		2
$f_6^{(2)}$:	30	46	43	25		

For checking purposes we list the first six $f_m(x)$ polynomials:

$f_0(x)=1$

$f_1(x)=4-4x$

$f_2(x)=9-24x+16x^2$

$f_3(x)=16-80x+128x^2-64x^3$

$f_4(x)=25-200x+560x^2-640x^3+256x^4$

$f_5(x)=36-420x+1792x^2-3456x^3+3072x^4-1024x^5$

$f_6(x)=49-784x+4704x^2-13440x^3+19712x^4-14366x^5+4096x^6$

The last row of the scheme yields the vector that is strongly graded in favor of the small eigenvalues. In our numerical example the smallest eigenvalue of the given maxtrix A is known to be

$$2(1-\cos 36^\circ)=0.3819660.$$

The associated eigenvector has the components

$$1,\quad 2\cos 36^\circ,\quad 2\cos 36^\circ,\quad 1$$

$$-1,\quad 1.6180340,\quad 1.6180340,\quad 1.$$

If the length of this vector is normalized to 1, and the same is done with $f_6^{(2)}$, we obtain the following comparison:

$$\frac{f_6^{(2)}}{|f_6^{(2)}|}=.404888,\ .620828,\ .580340,\ .337407$$

$$\frac{u_1}{|u_1|}=.371748,\ .601501,\ .601501,\ .371748.$$

We notice that the approximation is not very close. However, our aim is merely to provide a good start to the second algorithm. If we perform two cycles, the cycles 0 and 1, of the p, q algorithm, we obtain the following basic scalars:

$$\rho_0=-0.38506375$$

$$\sigma_0=-0.0080299090$$

$$\rho_1=-1.37489569.$$

The first-order polynomial gives the solution $\lambda=-\rho_0=0.385064$.
This is already a close approximation of the correct λ, which is $\lambda=0.3819660$. The second-order polynomial gives the quadratic equation

$$\lambda^2-(\sigma_0+\rho_0+\rho_i)\lambda+\rho_0\sigma_1=0$$

$$\lambda^2-1.76798935\lambda+0.52942249=0$$

whose roots are $\lambda_1=0.38198259$, $\lambda_2=1.38600677$. The approximation to the true λ_1 is already remarkably close, the error being only 1.7 units in the fifth decimal place. Moreover, the second root is a very good first approximation to the next smallest characteristic value, which is $2(1-\cos 72^\circ)=1.3819660$.

In addition, the first two cycles allow a correction of the first principal axis, according to the formula

$$u_1=p_0+\frac{\lambda_1+\rho_0}{\rho_0\sigma_0}p_1.$$

This gives, if again the length is normalized to 1:

$$\frac{u_1}{|u_1|}=.3713944,\ .6025945,\ .6003686,\ .3721606.$$

The length of the error vector is $1.66\cdot10^{-3}$. A strong improvement compared with the error of $f_6^{(2)}$, which was $5.57\cdot10^{-2}$.

This example demonstrates that we have no difficulty in improving a given first approximation λ_0 of an eigenvalue; moreover, we obtain a good approximation to the eigenvector associated with that eigenvalue. Hence the problem is reduced to the question of obtaining a good first approximation of a certain desired λ. Usually it is the λ of smallest absolute value in which we are primarily interested.

We can now proceed as follows. For a first crude approximation we put $\lambda=0$ and apply the purification process to the Hermitian matrices A and $\bar{A}$. The two vectors thus obtained may be too crude to be useful as starting vectors of the p, q algorithm. It may be preferable to improve this approximation by a least squares method now to be explained. If we had the right y, we could obtain the right λ from the condition (136). Since we do not possess the right y, we can still obtain a preliminary λ by minimizing the square of the length, that is, $k\bar{k}$, of the vector $k=(K-\lambda I)y$. This gives one complex λ. Another complex $\lambda=\lambda^*$ is obtainable from the adjoint problem $k^*=(K^*-\lambda^* I)y^*$, again minimizing the square of the length of this vector. While for the correct λ the two values λ and λ^* should coincide, this is not necessarily true for the approximations. We now use the approximation λ as the λ_0 of the process above for obtaining b_0 and λ^* as the λ_0 for obtaining b_0^*.

If we have not been successful in our start and obtained too slow a convergence in the ensuing p, q process, we can at any point of the process speed up the convergence by applying the purification procedure again, but now using for λ_0 the absolutely smallest root of the last characteristic equation.

The following interesting problem offers itself. Let $\lambda=\lambda_0$ be a good approximation of an eigenvalue of the arbitrary matrix K. Then forming the Hermitian matrices (140) and (142) with this λ_0 and obtaining the smallest eigenvectors of these matrices, these vectors will have a strong component in the direction of the principal axis u, u^* of the matrix K, associated with that particular λ. The first cycle of the p, q algorithm will then bring us closer to the true value of λ, and two cycles will improve further and give a good correction to the vector u, u^*. But what can we say about the *second* root of the characteristic equation? Can we assume—in analogy with the behavior of symmetric matrices—that our initial vector is not only close but also well graded, that is, that the second root will be a good approximation of the λ that is nearest in the complex plane to the first λ? This question requires further discussion which cannot be given here.

In this section we have merely sketched a method for obtaining the eigenvalues of an arbitrary complex matrix. However, no extensive numerical experiments have been performed so far. The writer hopes to go into further details about the method at some future time.

8. Summary

The present investigation advocates a combination of two procedures for the solution of large scale linear systems of equations. The first procedure evaluates the contribution of the large eigenvalues, the second the contribution of the small eigenvalues. The first algorithm has the advantage that it operates with a constant routine which does not change throughout the process. The second algorithm is more lengthy and requires corrections to counteract the accumulation of rounding errors. Hence it is of advantage to cut down the length of this algorithm to a minimum; this is achieved by the application of the preceding algorithm.

The final work scheme can be systematized into three distinct phases:

(a) Rescaling of the columns of the given matrix G by normalizing the length of each column to approximately 1. This makes the diagonal elements of the associated Hermitian matrix A nearly equal to 1, and all the nondiagonal elements numerically less than 1.

(b) Purification of the given right side b_0 of all its components in the direction of the large eigenvectors of A; a two-block scheme of five iterations each eliminates practically 90 percent of the λ spectrum. An additional block of five iterations eliminates about 94 percent of the spectrum. In this algorithm every iteration generates one new vector, by a recurrence scheme which has fixed coefficients involving the last vector and its penultimate.

(c) The remaining components in the direction of the small eigenvalues are eliminated by an algorithm which is again based on recurrences. However, every cycle now requires the generation of a *pair* of vectors, called p and q, apart from the matrix multiplication applied to q. Thus every cycle consists of three vectors. The recurrence relations involve the generation of two scalars in each cycle. In absence of rounding errors the first vectors (called p_i) of every cycle form an orthogonal set of vectors, while the second and third vectors are biorthogonal to each other. In view of the deorthogonalizing effect of rounding errors we check from time to time the orthogonality of the vectors obtained and interrupt the scheme if the orthogonality is no longer sufficiently strong. We then form the residual and start an independent second block of approximations. The solution is obtained as a given linear combination of the q-vectors and can be generated along with the other vectors, by constantly adding one more correction.

This method is not recommended when the principal aim is the evaluation of the elements of the inverse matrix, because it depends primarily on considering the matrix together with the given right side as a unified system. It is true that the method of minimized iterations can be adapted to arbitrary right sides (which is equivalent to inverting a matrix). This is so in spite of the fact that the basic vectors are obtained with the aid of one *specific* right side. However, the convergence of the process changes greatly with the given right side. For an arbitrary right side we have to assume that the process does not end before n steps. This requires that we have to generate a complete set of basic vectors. But then con-

stant reorthogonalization is required which is a lengthy procedure. The simple successive orthogonalization of the columns of the matrix, which also gives the inverted matrix and does not require any matrix multiplication, is preferable for this purpose.

In a given problem the inverted matrix will not always be required. The number of right sides with which we have to operate may not be too large and thus we may prefer to repeat the algorithm for every right side, particularly if the number of iterations required for the given accuracy happens to be much less than n. For example, we may imagine the situation that a given 50×50 matrix is not too skew-angular, to the extent that the symmetrized matrix A has no eigenvalues below 0.1 of the maximum eigenvalue. In this case a simple recurrence routine of 10 iterations will give the solution with sufficient accuracy, while the inversion of the matrix may require a much more elaborate calculation. A further advantage arises in the case of strongly skew-angular but "well-adjusted" physical systems. Here it is of definite advantage to separate the contribution of the large from that of the small eigenvalues because we can thus ameliorate the damaging influence of observational errors. These errors are greatly magnified in the theoretically exact mathematical solution, while in the iteration procedure they come into evidence only in the latest phase of the calculations, and that phase can be discarded.

The literature on the iterative solution of linear equations is very extensive; (see [8] for the older literature, and [2] and [1] for the newer literature on the subject). During the last few years many iterative schemes have been investigated. Among those developed at the National Bureau of Standards the gradient method of Hestenes and its modifications [11, 17] deserve particular attention, together with the asymptotic acceleration technique of Forsythe and Motzkin [7]. There is also the Monte Carlo method of Forsythe and Leibler [6]. The latest publication of Hestenes [10] and of Stiefel [18] is closely related to the p, q algorithm of the present paper, although developed independently and from different considerations.

The present investigation is based on years of research concerning the behavior of linear systems, starting with the author's consulting work for the Physical Research Unit of the Boeing Airplane Company, and continued under the sponsorship of the National Bureau of Standards. The author is indebted to Miss Lillian Forthal for her excellent assistance in the extensive numerical experiments that accompanied the various phases of theoretical deductions. The author is likewise indebted to the administration of the Institute for Numerical Analysis and the Office of Naval Research for the generous support of his scientific activities.

9. References

[1] W. E. Arnoldi, Quart. Applied Math. **9,** 17 to 30 (1951).
[2] E. Bodewig, Koninkl. Nederland Akad. Wetenschap Proc. **50,** 930 to 941, 1104 to 1116, 1285 to 1295 (1947); **51,** 53 to 64, 211 to 219 (1948).
[3] A. Brauer, Duke J. **13,** 387 to 395 (1946).
[4] D. A. Flanders and G. Shortly, J. Applied Phys. **21,** 1326 to 1332 (1950).
[5] G. E. Forsythe, Classification and bibliography of methods of solving linear equations.
[6] G. E. Forsythe and R. A. Leibler, MTAC **4,** 127 to 129 (1950).
[7] G. E. Forsythe and T. S. Motzkin, On a gradient method of solving linear equations; multilithed outline at National Bureau of Standards, Los Angeles, Calif.
[8] R. A. Frazer, W. J. Duncan and A. R. Collar, Elementary Matrizes, (Cambridge University Press, 1938); (MacMillan, New York, N. Y. 1947).
[9] S. Geršgorin, Izvest. Akad. Nauk SSSR **7,** 672 to 675 (1931).
[10] M. R. Hestenes, Iterative methods for solving linear equations, NAML Report 52–9.
[11] M. R. Hestenes and M. L. Stein, The solution of linear equations by minimization.
[12] A. S. Householder, Am. Math. Monthly **57,** 453 to 459 (1950).
[13] C. Lanczos, J. Math. Phys. **17,** 123 to 199 (1938).
[14] C. Lanczos, J. Research NBS **45,** 255 to 282 (1950) RP 2133.
[15] J. v. Neumann and H. H. Goldstine, Bul. Am. Math. Soc. **53,** 1021 to 1099 (1947).
[16] J. B. Rosser, M. R. Hestenes, W. Karush, and C. Lanczos, J. Research NBS **47,** 291 (1951) RP2256.
[17] M. L. Stein, Gradient methods in the solution of systems of linear equations, NAML Report 52–7.
[18] E. Stiefel, Z. ang. Math. Phys. (Zürich, Techn. Hochschule) **3,** 1 to 33 (1952).
[19] G. Szegö, Orthogonal Polynomials (Am. Math. Soc., New York, N. Y. 1939).
[20] O. Taussky, Am. Math. Monthly **56,** 672 to 675 (1949).

Los Angeles, September 28, 1951.