A Method of Computing Eigenvectors and Eigenvalues on an Analog Computer

By Lucien Neustadt*

1. Introduction. Many papers have been written in recent years describing methods for finding the eigenvalues and eigenvectors of an arbitrary matrix. Most of these methods apply to digital computation, and little attention has been paid to methods applicable to analog computers. One such method, which has been used successfully in the past, is described in [2]. The analog technique described in the present paper has the advantage of yielding both eigenvectors and eigenvalues of a real symmetric, or complex Hermitian matrix, and of not requiring a trial and error procedure. This is accomplished at the expense of additional computing equipment.

2. Mathematical Formulation. Consider the real, symmetric, $n \times n$ matrix $A = (a_{ij})$ with $a_{ij} = a_{ji}$. We shall state, without proof, the following properties of the matrix A [3], [4]:

(1) There is an orthonormal set of n real eigenvectors $\mathbf{e}^1, \mathbf{e}^2, \cdots, \mathbf{e}^n$ of the matrix A; i.e., there exist n numbers $\lambda_1, \lambda_2, \cdots, \lambda_n$, (the eigenvalues) with $A\mathbf{e}^i = \lambda_i \mathbf{e}^i$ and $i = 1, 2, \cdots, n$; also

$$\mathbf{e}^i \cdot \mathbf{e}^j = \delta_i{}^j.$$

(2) The eigenvalues are all real.

Suppose that the eigenvalues are arranged in descending order such that $\lambda_1 \geqq \lambda_2 \geqq \cdots \geqq \lambda_n$; then for an arbitrary real, non-zero vector

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}:$$

$$(3) \qquad \lambda_1 = \sup_{\mathbf{x}} \frac{\sum_{i,j=1}^{n} a_{ij} x_i x_j}{\sum_{i=1}^{n} x_i^2} \equiv \sup_{\mathbf{x}} \frac{A\mathbf{x} \cdot \mathbf{x}}{\mathbf{x} \cdot \mathbf{x}} = \sup_{\mathbf{x} \cdot \mathbf{x} = 1} A\mathbf{x} \cdot \mathbf{x}$$

The sup is attained at an eigenvector $\mathbf{e}^1$, belonging to $\lambda_1 \cdot \mathbf{e}^i$ and λ_i for $i > 1$ can be obtained from the same formula if the following added restriction is made:

$$\mathbf{x} \cdot \mathbf{e}^j = 0 \qquad\qquad j = 1, 2, \cdots, i - 1.$$

$\mathbf{e}^n$ and λ_n can also be computed from formula (3) if the sup is replaced by inf.

3. Solution Procedure. The technique is based on the analog computer method for solving problems in linear programming [5]. This method may be readily adapted to problems in nonlinear programming, i.e. to the problem of finding the extreme values of a nonlinear function subject to nonlinear restrictions. The problem of finding eigenvalues and eigenvectors is just such a problem, namely that of locating the extreme values listed above.

Let $\mathbf{x}$ be the radius vector to the point $x = (x_1, \cdots, x_n)$ in n space. Let x_1, $x_2, \cdots, x_n$ be functions of time. As described in the mathematical formulation, $\mathbf{e}^1$ may be found as follows:

Let the point x move on the unit sphere until it reaches a steady state corresponding to a maximum of the function $A\mathbf{x} \cdot \mathbf{x}$. Then $\mathbf{e}^1$ equals the steady state of $\mathbf{x}$.

The vector $\mathbf{e}^k$ is found in the same way, with the additional restrictions on $\mathbf{x}$ that $\mathbf{x} \cdot \mathbf{e}^1 = \mathbf{x} \cdot \mathbf{e}^2 = \cdots = \mathbf{x} \cdot \mathbf{e}^{k-1} = 0$.

Also, $\mathbf{e}^n$ is found in the same manner as $\mathbf{e}^1$, except that the steady state now corresponds to a minimum of $A\mathbf{x} \cdot \mathbf{x}$.

In order to find $\mathbf{e}^1$, we write a differential equation for the vector $\mathbf{x}$ such that the steady state solution of the equation is $\mathbf{e}^1$.

Denote $A\mathbf{x} \cdot \mathbf{x}$ by $f(x) = f(x_1, x_2, \cdots, x_n)$.

If we set $\dot{\mathbf{x}} = \operatorname{grad} f$, the point x will move in such a way as to increase f. To insure a steady state with $\mathbf{x} \cdot \mathbf{x} = 1$, modify this equation to read

$$(1) \qquad\qquad \dot{\mathbf{x}} = \operatorname{grad} f + k\epsilon\mathbf{x}$$

where

$$\epsilon = \begin{cases} 1 \text{ if } \mathbf{x} \cdot \mathbf{x} \leqq 1 \\ -1 \text{ if } \mathbf{x} \cdot \mathbf{x} > 1 \end{cases}$$

and k is a positive constant, chosen such that

$$k > 2 \max (|\lambda_1|, |\lambda_n|).$$

The point x will move as follows: Assume $x(0) = 0$. The origin is a point of unstable equilibrium. Once x moves slightly away from zero it will continue to move in a direction which is given by the vector sum of $\operatorname{grad} f$ and $k\mathbf{x}$. Because k is large enough, x will move toward the boundary of the unit sphere. Once it breaks through, again because of the choice of k, it will immediately re-enter. Having re-entered, it will immediately break through again, and so on indefinitely. The point x will then oscillate back and forth across the surface of the sphere with a frequency depending on the rapidity of the switching arrangement which determines the sign of ϵ. Superimposed on this oscillation is a tangential motion produced by the tangential component of $\operatorname{grad} f$. The tangential motion will continue until a point is reached where $\operatorname{grad} f$ has only a normal component. There $\operatorname{grad} f(x) = a\mathbf{x}$ for some constant a. It is easily verified that in general, $\operatorname{grad} f(x) = 2A\mathbf{x}$, so that this point corresponds to an eigenvector. Furthermore, at such a point, $f(x)$, where x is restricted to lie on the unit sphere, has an extreme value. If $\mathbf{x} \neq \mathbf{e}^1$, the solution is unstable, since any motion of x in the direction of $\mathbf{e}^1$ will be self-reinforcing. A stable steady state solution is reached only when $\mathbf{x} = \mathbf{e}^1$.

The eigenvector $\mathbf{e}^n$, belonging to the smallest eigenvalue λ_n, is the steady state of the solution to the equation

(2) $$\dot{\mathbf{x}} = -\operatorname{grad} f + k\epsilon\mathbf{x}.$$

If equation (1) is modified, the steady state of $\mathbf{x}$ can be made to correspond to the eigenvector $\mathbf{e}^2$. Consider the differential equation:

(3) $$\dot{\mathbf{x}} = \operatorname{grad} f + k\epsilon\mathbf{x} + k_1\epsilon_1 \operatorname{grad} f_1 \;;$$

where (a) ϵ and k are defined as above.

(b) $$f_1(x) = \mathbf{x}\cdot\mathbf{e}^1,$$

(c) $$\epsilon_1 = \begin{cases} 1 & \text{if} \quad f_1(x) \leqq 0 \\ -1 & \text{if} \quad f_1(x) > 0, \quad \text{and} \end{cases}$$

(d) k_1 is a large enough positive constant. It is sufficient that $k_1 > 2k$.

The added term will insure that the steady state of x yield a radius vector $\mathbf{x}$ orthogonal to $\mathbf{e}^1$. Since the maximum condition is also satisfied, this steady state corresponds to the eigenvector $\mathbf{e}^2$.

The other eigenvectors can be obtained by modifying equations 2 or 3 by the addition of similar terms.

In principle the eigenvalues can be obtained directly from the eigenvectors. Given the eigenvector $\mathbf{e}^i$, one may compute $A\mathbf{e}^i$. The ratio of the components of $A\mathbf{e}^i$ to the components of $\mathbf{e}^i$ equals the eigenvalue λ_i. Since $\mathbf{e}^i$ is obtained only approximately, this ratio will in general not be constant.

An excellent approximation to the eigenvalue, knowing an approximate eigenvector $\mathbf{e}^i$, is given by

$$\frac{A\mathbf{e}^i\cdot\mathbf{e}^i}{\mathbf{e}^i\cdot\mathbf{e}^i}$$ [4].

4. The Computer Setup. In order to find the eigenvector corresponding to the largest (or smallest) eigenvalue of A the following amount of computing equipment is necessary:

n integrating amplifiers, whose outputs are $x_1, x_2, \cdots, x_n$.

n inverting amplifiers, whose outputs are $-x_1, -x_2, \cdots, -x_n$.

n^2 scale factor potentiometers, corresponding to n potentiometers for each component of grad f.

One "switch" to compute ϵ. A high gain amplifier with two diodes and two voltage sources can be·used for the switch.

Multiplying equipment to compute $x_1^2, x_2^2, \cdots, x_n^2$, as well as $\epsilon x_1, \epsilon x_2, \cdots, \epsilon x_n$. For this purpose n multiplying servos positioned by $x_1, \cdots, x_n$ may be used.

One inverting amplifier to compute $-\epsilon$.

In order to find the eigenvector corresponding to λ_2 (or λ_{n-1}) the following additional equipment is necessary:

A switch to compute ϵ_1. This can again be a high gain amplifier.

n potentiometers to compute $f_1(x)$.

n potentiometers to compute the components of grad f_1.

One inverter to compute $-\epsilon_1$.

For every additional pair of eigenvectors more of the same equipment must be used.

The eigenvalues can also be found with the computer. Having computed the approximate unit eigenvector $\mathbf{e}^i$, λ_i can be approximated by $A\mathbf{e}^i \cdot \mathbf{e}^i$. But the steady state of $\mathbf{x}$ is $\mathbf{e}^i$, and $A\mathbf{x} = \frac{1}{2}\,\text{grad}\,f(x)$, so that in the steady state $2\lambda_i = (\text{grad}\,f) \cdot \mathbf{x}$. Since the components of grad f are available from the computation of $\dot{\mathbf{x}}$, λ_i can be obtained by performing the n multiplications necessary to form the dot product. The amount of equipment necessary for this is:

n summing amplifiers, whose outputs are the components of grad f.

n inverting amplifiers, whose outputs are the negative of the components of grad f.

One summing amplifier to form the dot product. Multiplying equipment to compute the terms of the dot product $\mathbf{x} \cdot (\text{grad}\,f)$. If multiplying servos were used above, one additional potentiometer per shaft is required.

Two precautions must be taken in order to allow for the dynamic limitations of the multipliers. First, the problem may have to be slowed down in order that the multipliers remain within their rate limits. Second, the constant k may have to be adjusted carefully, in order to insure stability of the computing loop.

Once an $n \times n$ matrix problem has been programmed, this programming—and the associated patching—can be used for any other $n \times n$ matrix, with only minor changes. Since the x_i are restrained to lie between 1 and -1, the scaling need not be changed. Only the potentiometer settings and the sign of the variable being fed into the potentiometers must be adjusted.

As an illustration, consider the following 3×3 matrix:

$$A = \begin{bmatrix} 3 & 5 & 2 \\ 5 & 0 & -3 \\ 2 & -3 & 1 \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$f(x) = A\mathbf{x} \cdot \mathbf{x} = 3x_1^2 + 10x_1x_2 + 4x_1x_3 - 6x_2x_3 + x_3^2$$

$$\text{grad}\,f = \begin{bmatrix} 6x_1 & + & 10x_2 & + & 4x_3 \\ 10x_1 & - & 6x_3 & & \\ 4x_1 & - & 6x_2 & + & 2x_3 \end{bmatrix}$$

Let $k = 20$.

Equation (1) becomes

$$\begin{cases} \dot{x}_1 = 6x_1 + 10x_2 + 4x_3 + 20\epsilon x_1 \\ \dot{x}_2 = 10x_1 - 6x_3 + 20\epsilon x_2 \\ \dot{x}_3 = 4x_1 - 6x_2 + 2x_3 + 20\epsilon x_3\,, \end{cases}$$

where

$$\epsilon = \text{sgn}\,[1 - (x_1^2 + x_2^2 + x_3^2)].$$

The programming for this example is shown in Figure 1. The problem has been slowed down by a factor of twenty.

The steady state of this system yields the eigenvector $\mathbf{e}^1$ corresponding to the largest eigenvalue λ_1.
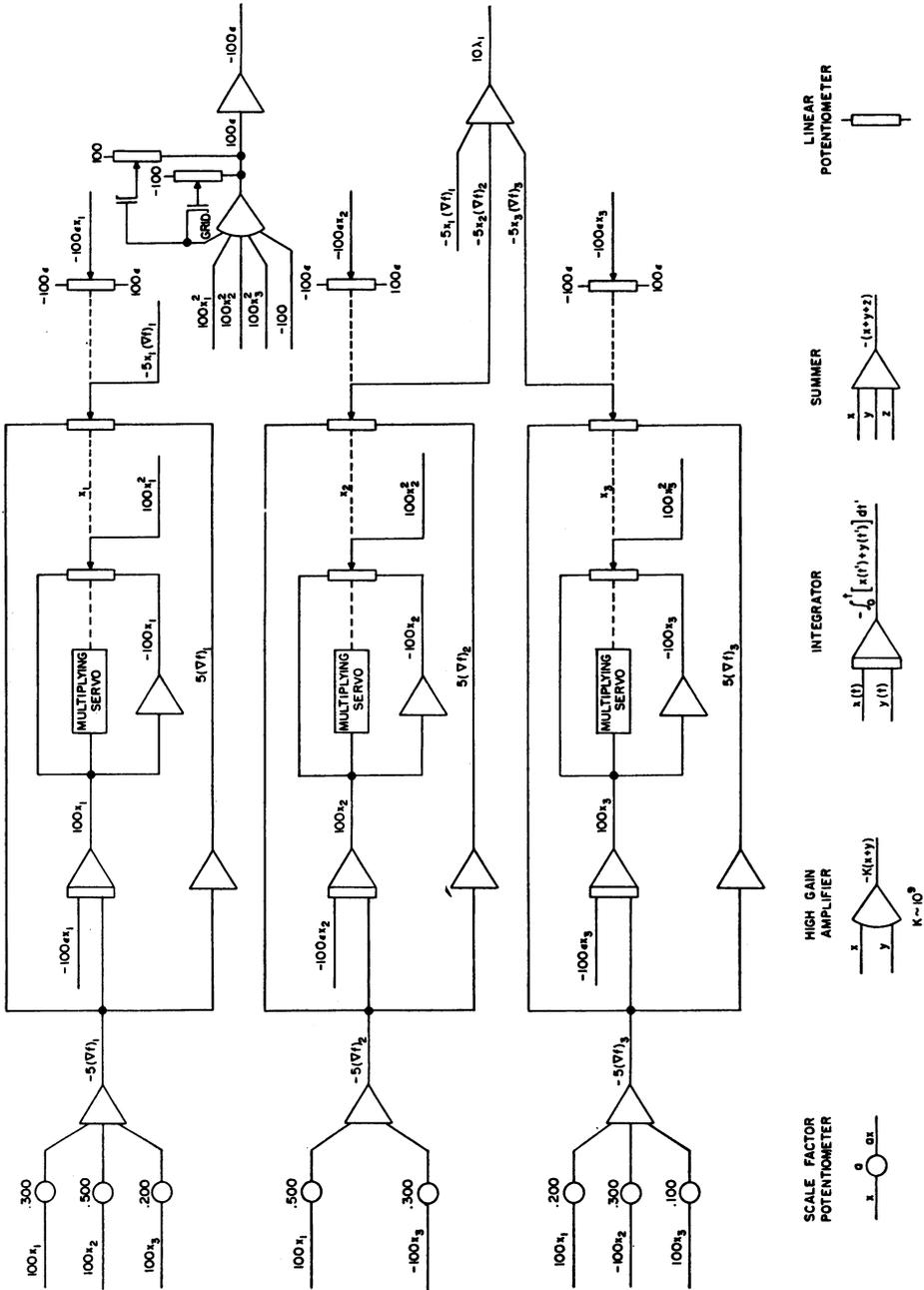
FIG. 1.—Program for typical problem.

In order to compute the eigenvector $\mathbf{e}^3$ corresponding to the smallest eigenvalue λ_3, equation 3 is used:

$$\begin{cases} \dot{x}_1 = -6x_1 - 10x_2 - 4x_3 + 20\epsilon x_1 \\ \dot{x}_2 = -10x_1 + 6x_3 + 20\epsilon x_2 \\ \dot{x}_3 = -4x_1 + 6x_2 - 2x_3 + 20\epsilon x_3 . \end{cases}$$

Finally, in order to compute the eigenvector $\mathbf{e}^2$ either of the two above systems of equations must be modified by the addition of an extra term. For example

$$\begin{cases} \dot{x}_1 = 6x_1 + 10x_2 + 4x_3 + 20\epsilon x_1 - \xi_1\epsilon_1 \\ \dot{x}_2 = 10x_1 - 6x_3 + 20\epsilon x_2 - \xi_2\epsilon_1 \\ \dot{x}_3 = 4x_1 - 6x_2 + 2x_3 + 20\epsilon x_3 - \xi_3\epsilon_1 , \end{cases}$$

where

$$\begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{bmatrix} = \mathbf{e}^1 \quad \text{and} \quad \epsilon_1 = \operatorname{sgn}(x_1\xi_1 + x_2\xi_2 + x_3\xi_3).$$

5. Results. The eigenvalues and eigenvectors of two matrices were computed at Project Cyclone, using REACs and Reeves multiplying servos. One of the matrices was the 3×3 listed above. The other was a 6×6 matrix, listed in reference 1 and reprinted below:

$$\begin{bmatrix} .06667 & .02634 & -.04640 & -.07368 & -.02131 & -.00431 \\ .02634 & .26841 & -.02243 & .15952 & -.05923 & -.12797 \\ -.04640 & -.02243 & .10932 & .05150 & -.04100 & .08558 \\ -.07368 & .15952 & .05150 & .25152 & -.01141 & -.07169 \\ -.02131 & -.05923 & -.04100 & -.01141 & .14403 & .01105 \\ -.00431 & -.12797 & .08558 & -.07169 & .01105 & .19450 \end{bmatrix}$$

A comparison between the computed and actual values of the eigenvector components are listed in Table 1.

The eigenvalues were not computed on the REAC, but a comparison may be made between $(A\mathbf{e}^i \cdot \mathbf{e}^i)/(\mathbf{e}^i \cdot \mathbf{e}^i)$ (where $\mathbf{e}^i$ is the computed eigenvector) and λ_i. This is shown in Table 2.

6. Additional Remarks. In the case of multiple eigenvalues, the eigenvectors are not, of course, unique. The method of this paper will then yield some set of orthonormal eigenvectors belonging to the eigenvalue. If two eigenvalues differ only slightly, the corresponding eigenvectors will be computed with less accuracy. But the corresponding eigenvalues will be obtained with no sacrifice in accuracy. In fact, the eigenvalues are relatively insensitive to errors in the eigenvectors. This is vividly demonstrated in Table 2, where the eigenvalues are seen to be correct to five or six significant figures.

The method of this paper can be extended to complex Hermitian matrices. The theorems listed in the mathematical formulation are basically still true if

TABLE 1

Matrix 1

e¹		e²		e³	
computed	actual	computed	actual	computed	actual
.7933	.7930···	−.3189	−.3184···	−.5193	−.5194···
.6077	.6078···	.3562	.3555···	.7096	.7101···
−.0419	−.0415···	−.8794	−.8788···	.4762	.4754···

Matrix 2

e¹		e²		e³	
computed	actual	computed	actual	computed	actual
.0408	.0414··	−.3558	−.3554··	.3253	.3261··
−.6762	−.6763··	−.1227	−.1221··	.2627	.2639··
.0387	.0394··	.6012	.6014··	.2129	.2134··
−.5746	−.5745··	.5053	.5051··	−.3048	−.3044··
.1376	.1382··	−.0894	−.0889··	−.8014	−.8006··
.4361	.4361··	.4836	.4838··	.2104	.2116··

e⁴		e⁵		e⁶	
computed	actual	computed	actual	computed	actual
−.3742	−.3732··	.2977	.2989··	.7326	.7328··
−.5208	−.5213··	−.3411	−.3406··	−.2653	−.2650··
.0686	.0678··	−.6013	−.6014··	.4741	.4743··
.0214	.0222··	.5271	.5271··	.2095	.2093··
−.4721	−.4728··	−.2775	−.2772··	.1771	.1777··
−.6004	−.6002··	.2789	.2799··	−.3039	−.3040··

TABLE 2

	Matrix 1			Matrix 2	
	From Computed Eigenvector	Actual		From Computed Eigenvector	Actual
λ_1	6.727878	6.727881···	λ_1	.4982334	.4982344···
λ_2	2.938089	2.938091···	λ_2	.25946618	.25946632···
λ_3	−5.665964	−5.665972···	λ_3	.1759010	.1759013···
			λ_4	.0823481	.0823483···
			λ_5	.01581329	.01581310···
			λ_6	.00268708	.00268704···

one extends the definition of inner product to complex vectors by the formula $\mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^{n} x_i \bar{y}_i$ [3].

The quadratic form $A\mathbf{x} \cdot \mathbf{x} = \sum_{i,j=1}^{n} a_{ij} x_i \bar{x}_j$ is real, and the extremum property can be used in the same way to find the real and imaginary parts of the eigenvectors.

Even normal matrices can be handled if one separates them into their real and imaginary parts [3].

7. Conclusion. The eigenvalue problem for real symmetric, or Hermitian, matrices can be solved on an electronic analog computer by formulating it as an extremum problem. Both the eigenvectors and eigenvalues can be obtained. With care, three place accuracy can be obtained for the eigenvector. If the eigenvalue is computed by hand from the eigenvector, six place accuracy can be obtained for the eigenvalues.

Reeves Instrument Corporation,
Garden City, New York

1. OLGA TAUSSKY, Editor, "Contributions to the solution of linear equations and the determination of eigenvalues," NBS *Applied Math. Series* 39, 1954, p. 60.
2. LANDIS GEPHART, "Linear algebraic systems and the REAC," MTAC, v. 6, 1952, p. 190–203.
3. PAUL R. HALMOS, *Finite Dimensional Vector Spaces*, Princeton University Press, New Jersey, 1948, p. 124–134.
4. E. BODEWIG, *Matrix Calculus*, North Holland Publishing Company, Amsterdam, 1956, p. 54–56 and 245.
5. INSLEY PYNE, "Linear programming on an electronic analogue computer," A.I.E.E. *Transactions Annual*, Part 1, Communication and Electronics, 1956, p. 139–143.
6. CLARENCE L. JOHNSON, *Analog Computer Techniques*, McGraw-Hill, New York, 1956.
7. ABRAHAM MANY, "Improved electrical network for determining eigenvalues and eigenvectors of a real symmetric matrix," *Review of Scientific Instruments*, v. 21, 1950, p. 972–974.
8. V. ROGLA, "Analog machine for algebraic computations," presented at the International Analog Computation Meeting, Brussels, Belgium, 1955.