JOURNAL OF COMPUTER AND SYSTEM SCIENCES: 5, 17–25 (1971)

The Concept of Effective Method Applied to Computational Problems of Linear Algebra

OLIVER ABERTH

Department of Mathematics, Texas A & M University, College Station, Texas 77843

Received April 28, 1969

A classification of computational problems is proposed which may have applications in numerical analysis. The classification utilizes the concept of effective method, which has been employed in treating decidability questions within the field of computable numbers. A problem is effectively soluble or effectively insoluble according as there is or there is not an effective method of solution. Roughly speaking, effectively insoluble computational problems are those whose general solution is restricted by an intrinsic and unavoidable computational difficulty. Some standard problems of linear algebra are analyzed to determine their type.

INTRODUCTION

Given a square matrix A with real entries, the computation of the eigenvalues of A is, more or less, a routine matter, and many different methods of calculation are known. On the other hand, the computation of the Jordan normal form of A is a problematical undertaking, since when A has multiple eigenvalues, the structure of off-diagonal 1's of the normal form will be completely altered by small perturbations of the matrix elements of A. Still, one may ask whether the Jordan normal form computation can always be carried through for the case where the entries of A are known to arbitrarily high accuracy.

This particular question and others like it can be given a precise answer by making use of a constructive analysis developed in the past two decades by a number of Russian mathematicians, the earliest workers being Markov, Ceitin, Zaslavskii, and Sanin [4, 6, 10, 12, 13]. The analysis is restricted to the field of computable numbers, where a real number is called computable if there is an algorithm for obtaining arbitrarily precise rational approximations. The functions and sequences of the analysis also are defined in terms of algorithms, and in order to make clear what constitutes an "algorithm", there must be employed some explicit constructive concept, as, e.g., that of Turing machines or of recursive functions. In the articles [1, 2], we gave

571/5/1-2

an exposition of the analysis where a formal concept of "program" is the central constructive device.

An interesting feature of the analysis is that it permits a classification of computational problems that may prove useful in numerical analysis. In computations with real numbers, we can deal with only the computable numbers, so the restriction of the analysis to this subfield of the reals is no disadvantage here.

The classification we have in mind makes use of the concept of "effective method," often employed in decidability investigations, and in order to present this idea in our notation, a brief summary of terminology may be appropriate here: A rational-valued function of rational or integer variables is called "programmable" if it can be realized as a program (see Def. 2 of [1]). A number a is "computable" if there is a programmable function $\alpha(\epsilon)$ of the positive rational variable ϵ satisfying the inequality $|a - \alpha(\epsilon)| \leqslant \epsilon$. To each program P is assigned a unique positive integer N_P, the "descriptive integer of the program P." The integers N_P are a kind of "Gödel numbering" for the programs P. In a sense, they represent an "encoding" of the programs P, and are used chiefly for the formal convenience of dealing with integers rather than the actual programs. Details about a program P that are constructively available by knowing P, are constructively available by knowing N_P. Thus, in the case of $\alpha(\epsilon)$ just mentioned, if $\alpha(\epsilon)$ is defined by a program P with descriptive integer N_P, there is a programmable function $U(n, r)$, n an integer, r a rational number, such that $\alpha(\epsilon) = U(N_P, \epsilon)$ [1, Th. 2].

Given a computational problem, let the input computable numbers, functions, or sequences be defined by programs whose descriptive integers, in some assigned order, are $N_1, N_2, \ldots, N_k$. There is an effective method of solving the problem if there is a programmable function $F(N_1, N_2, \ldots, N_k)$ of the integer variables N_i which defines the computational result. F may have additional variables besides the ones shown in accordance with the type of result that must be defined. If the result is merely an indication of which of several alternative statements is true, then this can be done by requiring F to assume one of an appropriate number of integer values. When the computational result is a (computable) number, the argument ϵ is added to F, ϵ the bound on the error of the rational approximation given by F. Similarly, when the computational result is a finite or infinite sequence of numbers, two additional arguments n and ϵ are required, with ϵ as before and n an integer index to indicate which particular number F is defining.

As illustrations, consider the following two problems:

A. Given the numbers a, b, decide whether or not $a = b$.

B. Given the numbers a, b, find their maximum value.

If a, b are defined by the programmable approximation functions $\alpha(\epsilon)$, $\beta(\epsilon)$, with N_1, N_2 the descriptive integers of the corresponding programs, then there is an effective method of solving problem A if there is a programmable function $F(N_1, N_2)$ which equals 1 if $a = b$ and 2 if $a \neq b$. There is an effective method of solving

problem B if there is a programmable function $F(N_1, N_2; \epsilon)$ which gives a rational approximation to the maximum, with error bound ϵ.

For problem B, the existence of an effective method of solution is easily shown. An ϵ approximation to the maximum value of a, b is given by

$$\max(\alpha(\epsilon), \beta(\epsilon)) = (\alpha(\epsilon) + \beta(\epsilon))/2 + |\alpha(\epsilon) - \beta(\epsilon)|/2.$$

In terms of the programmable function U mentioned earlier, we may write $\alpha(\epsilon) = U(N_1, \epsilon)$, $\beta(\epsilon) = U(N_2, \epsilon)$, and the existence of the required programmable function F is evident [1, Th. 1 and Corollary].

For Problem A on the other hand, there is no effective method of solution. The following result was proved in [1] (Cor. 1 of Theorem 14):

$N1$.[1] For a fixed computable number b, there is no effective method of determining for computable numbers a whether or not $a = b$.

If, e.g., $b = 0$, there is no programmable function $F(N_1)$ which equals 1 or 2 according as a is or is not equal to 0, N_1 the descriptive integer of a program defining $\alpha(\epsilon)$. This implies that the more general programmable function required for problem A does not exist either.

This negative result for problem A may be elucidated by considering a variation on the problem. Suppose that instead of having the programs for $\alpha(\epsilon)$, $\beta(\epsilon)$ available via their descriptive integers N_1, N_2, we are supplied merely with any particular value of $\alpha(\epsilon)$ or $\beta(\epsilon)$ that we might require. Each function $\alpha(\epsilon)$ or $\beta(\epsilon)$ is available, so to speak, only as a "black box" with ϵ the input and the corresponding function value the output returned. For this arrangement, it is clear that there can be no general method of deciding whether or not $a = b$, since if the two numbers were actually equal, we could never be certain of this by knowledge of any finite number of values of $\alpha(\epsilon)$, $\beta(\epsilon)$. Now it is conceivable that if we had the details of the programs for $\alpha(\epsilon)$, $\beta(\epsilon)$, we might be able to resolve this problem. $N1$, however, shows that there is no general way of doing this. Note that there is a general method for solving problem B under a "black box" restriction for $\alpha(\epsilon)$, $\beta(\epsilon)$.

These particular results may be generalized. The work of Kreisel–Lacombe–Schoenfield [5] and Ceitin [4] both show that for a computational problem dependent only on initial computable numbers in finite or infinite intervals, where the result is either the designation of a computable number or a choice of alternatives, in any case a function of the initial numbers, there is an effective method of solution if and only if there is a general "black box" method of solution. This equivalence often is useful in understanding negative results. For instance, consider the following statement, needed later:

[1] We use the letter N for a statement denying the existence of an effective method, and the letter P for a statement affirming the existence of one.

$N1'$ For a fixed computable number b, there is no effective method of deciding, for computable numbers a, on one of the following relations R_1, R_2 as true:

$$R_1 : a \geqslant b$$
$$R_2 : a \leqslant b$$

If we specify that the relation R_1 must be chosen when $a = b$, then the choice of alternatives is a function of the initial number a. There, clearly, is no general "black box" solution; so by the equivalence, there is no effective method of solution. Even when complete freedom on the choice of R_1, R_2 is allowed, there still is no effective method of solution, and the technique of proof of Theorem 21 of [1] can be modified for this case.

To return to Problem A, suppose what we actually require is knowledge merely of whether or not a is "close" to b. A positive result is this direction is

$P1$. For any two computable numbers a, b, there is an effective method of deciding, for any prescribed positive integer n, on one of the following relations R_1, R_2 as true:

$$R_1 : |a - b| \leqslant \frac{1}{n}$$

$$R_2 : |a - b| > \frac{1}{2n}$$

Since R_1 is true if

$$\left|\alpha\left(\frac{1}{8n}\right) - \beta\left(\frac{1}{8n}\right)\right| \leqslant \frac{3}{4n}$$

and R_2 is true if

$$\left|\alpha\left(\frac{1}{8n}\right) - \beta\left(\frac{1}{8n}\right)\right| > \frac{3}{4n},$$

there is no difficulty in constructing the required programmable function $F(N_1, N_2; n)$.

A second positive formulation is

P_R1. For any two rational numbers a, b there is an effective method of deciding whether or not $a = b$.

Certainly, if a and b are the rational numbers p/q, p'/q', respectively, we can decide without difficulty whether or not $a = b$. However, even when a and b are known to be rational, if for a only the approximation function $\alpha(\epsilon)$ is available, an examination of the proof of $N1$ in [1] shows that it will still apply. For P_R1 then, and for all later P_R statements, the understanding is that an initial rational number is known in the usual form p/q, and, accordingly, we require that the corresponding descriptive integer define a specific approximation function, one identically equal to p/q.

The preceding examples suggest a classification of computational problems. We may call a problem *effectively soluble* if there is an effective method of solution. These problems fall within the province of numerical analysis, and the various solution algorithms for any such particular problem may be compared one with another according to various criteria of practicality or efficiency.

A problem is *effectively insoluble* if there is no effective method of solution. These problems clearly possess an intrinsic difficulty that no amount of sophistication or subtlety can circumvent. Any conceivable system of solution which can be written down as a finite list of instructions either must fail to yield an answer in certain cases, or, if an answer is always forthcoming, must yield a wrong answer in certain cases. Of course, in any particular instance of an effectively insoluble problem, we may be able to correctly solve it, but this must be counted a fortuitous circumstance since our method of solution fails in other instances of the problem. As with problem A, there may be an acceptable reformulation of an effectively insoluble problem which avoids the intrinsic difficulties.

In this paper, we analyze some of the standard problems of linear algebra to determine their type. Although the vectors and matrices we deal with have computable (or complex computable) components rather than real (or complex) components, the various linear algebras are not particularly dissimilar, and the usual computational problems correspond. Considering a complex computable number as equivalent to an ordered pair of computable numbers, it is easy to adapt the concept of effective method to problems which have these numbers as inputs or results.

Since eigenvalues may be computed by finding the roots of polynomials, let us note here:

P2. For polynomials $P_n(x) = x^n + a_{n-1}x^{n-1} + \cdots + a_0$ of degree $n > 0$ with complex computable coefficients, the roots are complex computable numbers (Theorem 6 of [8]), and there is an effective method of obtaining them [9].

A related negative result is:

N2. For polynomials $P_n(x) = x^n + a_{n-1}x^{n-1} + \cdots + a_0$ of degree $n > 1$ and computable number coefficients, there is no effective method of determining whether or not there are multiple roots, and there is no effective method of determining the multiplicity of any individual root.

Take the case $n = 2$. If either of the above effective methods existed for quadratic polynomials, it could be employed on the polynomials $x^2 + (a + b)x + ab = (x + a)(x + b)$ and we then would obtain an effective method of deciding whether $a = b$, contradicting *N1*.

The Euclidean algorithm applied to $P_n(x)$ and $P_n'(x)$ does not enable us to decide whether there are multiple roots because of the difficulty of being certain of a zero remainder. For rational coefficients, of course there is no difficulty:

P_R2. For polynomials $P_n(x) = x^n + a_{n-1}x^{n-1} + \cdots + a_0$ with rational coefficients, there are effective methods of determining whether or not there are multiple roots, or the multiplicity of any individual root.

Vectors and Matrices

One of the simplest matrix computations, that of rank, is an effectively insoluble computational problem:

$N3$. For $m \times n$ matrices with computable number elements, there is no effective method of determining rank.

For instance, if for 2×2 matrices there were such an effective method of determining rank, then we could apply it to matrices $\left[\begin{smallmatrix} a & 0 \\ 0 & 0 \end{smallmatrix}\right]$ and obtain an effective method of determining whether the computable number a equals 0, contradicting $N1$.

For completeness, we list the positive result for rational elements:

P_R3. For $m \times n$ matrices with rational elements, there is an effective method of determining rank.

Let us define a vector, as usual, as a single column matrix. The norm of a vector X, $\| X \|$, we take as $(X^* X)^{1/2}$, where X^* designates the conjugate transpose of X. Then we may assert:

$N4$. For n-square matrices A, $n > 1$, with computable number elements and for which the rank is known to be less than n, there is no effective method of obtaining a solution vector X of norm 1 such that $AX = 0$.

Let us take the case of 2×2 matrices. For a given computable number a, set $b = | a | + a$, $c = | a | - a$, so that $bc = 0$. Then the matrix $\left[\begin{smallmatrix} b & c \\ b & c \end{smallmatrix}\right]$ always has a rank less than 2. If we know that $X = \left[\begin{smallmatrix} x_1 \\ x_2 \end{smallmatrix}\right]$ is a solution vector of norm 1, we will be able to find a component unequal to 0. If $x_1 \neq 0$, then from $bx_1 + cx_2 = 0$, we obtain $b^2 x_1 = 0$ and this entails $a \leqslant 0$. Similarly, if $x_2 \neq 0$, we must have $a \geqslant 0$. But then it is clear that an effective method for obtaining a solution vector X of norm 1 can be converted to an effective method for deciding whether $a \geqslant 0$ or $a \leqslant 0$, contradicting $N1'$.

We can obtain an effectively soluble problem by either requiring more information, or by not insisting that the vector X be an exact solution.

$P4$. For n-square matrices A with computable number elements and for which the rank r is known and is less than n, there is an effective method of obtaining $n - r$ linearly independent solution vectors $X_1, X_2, \ldots, X_{n-r}$ of norm 1 such that $X = c_1 X_1 + \cdots + c_{n-r} X_{n-r}$ is a solution to $AX = 0$ for arbitrary computable number coefficients c_i.

If the rank r is 0, we may take for X_i, $i = 1, 2, \ldots n$, a vector with 1 as its i-th

component and all other components 0. If $r > 0$, we may test the determinants of all r-square submatrices A_r of A to decide whether $| \det A_r | \leqslant 1/k$ or $| \det A_r | > 1/2k$ (cf. $P1$), for k equal successively to 1, 2, 3,... . Eventually for some k we will locate a submatrix for which the second of the two inequalities above holds. Then since this submatrix has a nonzero determinant, we may obtain the vectors X_i in the usual way by the application of Cramer's rule.

$P4'$. For any n-square matrix A with computable number elements, for which the rank is known to be less than n, and for any prescribed positive integer m, there is an effective method of obtaining a vector X of norm 1 such that $\| AX \| \leqslant 1/m$.

Here we may proceed as follows: First, we test each element a_{ij} of A to decide whether $| a_{ij} | \leqslant 1/nm$ or $| a_{ij} | > 1/2nm$. If all the elements satisfy the first inequality, then we may take X as the vector with 1 for its first component and all other components 0. If at least one element a_{ij} satisfies the second inequality, then $1 \leqslant \text{rank } A \leqslant n - 1$. Then, taking the rank successively as 1, 2,..., $n - 1$ we proceed as described in the proof of $P4$. However, here we advance k only when all the rank computations are complete. Eventually, for some rank r we will locate an r-square submatrix with nonzero determinant and obtain a (supposed) set of solution vectors X_1 ,..., X_{n-r} with norm 1.

Testing $\| AX_1 \|$, we will be able to decide whether $\| AX_1 \| \leqslant 1/m$ or $\| AX_1 \| > 1/2m$. In the second case, we exclude all ranks less than or equal to r from further consideration and proceed in our search. Eventually, we must obtain a vector X of norm 1 for which $\| AX \| \leqslant 1/m$, although by $N4$ we can not, in general, be certain whether it is a solution vector.

Eigenvectors and the Diagonalization of Square Matrices

A commonly encountered problem in matrix calculus is the determination, corresponding to a given matrix A, of a transforming matrix U such that $U^{-1}AU = \Lambda$ is in some desired canonical form. In the case of the Jordan canonical form, the introductory problem of this paper, we have:

$N5$. For n-square matrices A, $n > 1$, with computable number elements, there is no effective method of determining the Jordan canonical form.

This is easy to show. For 2×2 matrices, if there were such an effective method, then since $\left[\begin{smallmatrix} 0 & a \\ 0 & 0 \end{smallmatrix}\right]$ has the canonical form $\left[\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right]$ or $\left[\begin{smallmatrix} 0 & 0 \\ 0 & 0 \end{smallmatrix}\right]$ according as $a \neq 0$ or $a = 0$, we would arrive at an effective method of determining whether $a = 0$, contradicting $N1$.

For matrices with rational elements, we have, of course, a positive result:

P_R5. For n-square matrices A with rational elements, there is an effective method of determining the Jordan canonical form and the transforming matrix U.

In the case of symmetric matrices with computable number elements, the Jordan form certainly can be determined since A is diagonal with the eigenvalues as the diagonal elements. These, being the roots of the characteristic polynomial, can be determined effectively by $P2$. However, even in this case we have the following negative result:

$N6$. For symmetric n-square matrices $A, n > 1$, with computable number elements, there is no effective method of determining an eigenvector of A with norm 1. (This implies that there is no effective method of determining an orthogonal matrix U such that $U^{-1}AU$ is diagonal.)

For $n = 2$, consider the matrices $\begin{bmatrix} c & b \\ b & c \end{bmatrix}$, where as in the proof of $N4$, $b = |a| + a$, $c = |a| - a$, and a may be any computable number. If $b = 0, c \neq 0$, the eigenvectors are $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$. If $b \neq 0$, $c = 0$, then the eigenvectors are $1/\sqrt{2}\begin{bmatrix} 1 \\ -1 \end{bmatrix}$ and $1/\sqrt{2}\begin{bmatrix} 1 \\ 1 \end{bmatrix}$. If we had an effective method of determining an eigenvector $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ with norm 1, then we could choose either 0 or 1 as a number unequal to $||x_1| - |x_2||$. In the first case, $b = 0$; in the second, $c = 0$. As in the proof of $N4$, we come to a contradiction of $N1'$.

As with $N4$, we can convert the above effectively insoluble problem to an effectively soluble one by either requiring more information or by reducing our demands.

$P6$. For n-square matrics $A, n > 1$, with computable number elements, if we know that the roots of the characteristic polynomial are all distinct, then there is an effective method of determining a matrix U and diagonal matrix Λ such that $U^{-1}AU = \Lambda$. Alternatively, there is an effective method of determining the eigenvectors with norm 1.

Since $A - \lambda I$ has rank $n - 1$ for any eigenvalue λ, by $P4$ we may determine a corresponding eigenvector of norm 1. The eigenvectors may then be combined to form U. Although λ may be a complex computable number, this does not cause any difficulty since $P4$ and $P4'$ are clearly still valid if for "computable number" we read "complex computable number."

$P6'$. For n-square matrices A, $n > 1$, with computable number elements and for any fixed positive integer m, if λ is an eigenvalue of A there is an effective method of finding a vector X of norm 1 such that $||AX - \lambda X|| \leqslant 1/m$.

Here $P4'$ is applicable since rank $(A - \lambda I) < n$.

A generalization of $P6'$ that may be obtained without much difficulty [3, pp 195–196] is:

$P7$. For n-square matrices A, $n > 1$, with computable number elements, and for any positive integers m, there is an effective method of finding a unitary matrix U such that $U^{-1}AU$ has all the elements below the main diagonal not greater than $1/m$ in absolute value.

Note that there can be no effective method for finding a unitary matrix U such that $U^{-1}AU$ has all elements below the main diagonal 0, for then the first column of U defines an eigenvector of A with norm 1, and this contradicts $N6$.

Finally, we list an "approximate diagonalization" result.

$P8$. For n-square matrices A, $n > 1$, with computable number elements, and for any positive integer m, there is an effective method of finding a matrix U such that $U^{-1}AU$ has all the elements below or above the main diagonal not greater than $1/m$ in absolute value.

If U_1 is the unitary matrix of $P7$, then $U = U_1D$, where D is an appropriate diagonal matrix.

Acknowledgment

I am indebted to Professor G. R. Blakley for pointing out the computational difficulty in obtaining the Jordan normal form of a matrix, and thus suggesting the application of computable analysis to linear algebra.

References

1. O. Aberth, Analysis in the computable number field, *J. Assoc. Comput. Mach.* **15** (1968), 275–299.
2. O. Aberth, A chain of inclusion relations in computable analysis, *Proc. Amer. Math. Soc.* **22** (1969), 539–548.
3. R. Bellman, "Introduction to Matrix Analysis," McGraw-Hill, New York, 1960.
4. G. S. Ceitin, Algorithmic operators in constructive complete separable metric spaces, *Dokl. Akad. Nauk SSSR* **128** (1959), 49–52.
5. G. Kreisel, D. Lacombe, and J. R. Shoenfield, "Partial recursive functionals and effective operations," Proc. Colloq. at Amsterdam 1957, 290–297, North-Holland, Amsterdam, 1959.
6. A. A. Markov, On constructive functions, Trudy Mat. Inst. Steklov **52** (1958), 315–348.
7. J. Myhill and J. C. Shepherdson, Effective operations on partial recursive functions, Z. Math. Logik Grundlagen Math. **1** (1955), 310–315.
8. H. G. Rice, Recursive real numbers, *Proc. Amer. Math. Soc.* **5** (1954), 784–791.
9. P. C. Rosenbloom, An elementary constructive proof of the fundamental theorem of algebra, *Amer. Math. Monthly*. **52** (1945), 562–570.
10. N. A. Sanin, A constructive interpretation of mathematical judgments, *Trudy Mat. Inst. Steklov* **52** (1958), 226–311; *Amer. Math. Soc. Transl.* (2) **23** (1963), 109–189.
11. A. M. Turing, On computable numbers, with an application to the Entscheidungsproblem, *Proc. Lond. Math. Soc.* (2) **42** (1936–37), 230–265.
12. I. D. Zaslavskii, Some properties of constructive real numbers and constructive functions, *Trudy Mat. Inst. Steklov* **67** (1962), 385–457; *Amer. Math. Soc. Transl.* (2) **57** (1966), 1–84.
13. I. D. Zaslavskii and G. S. Ceitin, Singular coverings and properties of constructive functions connected with them, *Trudy Mat. Inst. Steklov* **67** (1962), 458–502.