7

ORNL-2652
Physics and Mathematics
TID-4500 (14th ed.)

MATHEMATICS PANEL

PROGRESS REPORT

FOR

PERIOD MARCH 1, 1957 TO AUGUST 31, 1958

[redacted]

OAK RIDGE NATIONAL LABORATORY

operated by

UNION CARBIDE CORPORATION

for the

U.S. ATOMIC ENERGY COMMISSION

Printed in USA. Price $2.25 Available from the

Office of Technical Services
U. S. Department of Commerce
Washington 25, D. C.

ORNL-2652
Physics and Mathematics
TID-4500 (14th ed.)

Contract No. W-7405-eng-26

MATHEMATICS PANEL

PROGRESS REPORT

for

Period March 1, 1957 to August 31, 1958

A. S. Householder, Chief

DATE ISSUED

MAR 9 1959

OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee
operated by
UNION CARBIDE CORPORATION
for the
U. S ATOMIC ENERGY COMMISSION

CONTENTS

MATHEMATICS PANEL PROGRESS REPORT

NUMERICAL ANALYSIS, PROGRAMING, AND BASIC MATHEMATICS

THE BOLTZMANN EQUATION

K. Fan J. J. Andrews

A function $f_n(x,y)$ has been proposed by Kofink[1] as the nth approximation to the solution of a form of the Boltzmann equation. Since Kofink's proofs are incomplete, the proposed approximation has been examined in detail and proofs have been supplied. In particular, the definition of the functions f_n presupposes certain properties of the zeros of auxiliary functions G_{2n}. These properties are established and, in fact, sharpened by showing, among other things, that the positive zeros λ_i of G_{2n} and those, μ_i, of the Legendre polynomial P_{2n}, satisfy $\lambda_1 > \mu_1 > \lambda_2 > \ldots > \lambda_n > \mu_n$. It is shown that Kofink's expressions for coefficients α_k occurring in the definition of the f_n are valid only when the parameters a and c appearing in the Boltzmann equation satisfy inequalities $0 < c < 1$, $0 < a < 3(1 - c)$, whereas somewhat simpler expressions for α_0 and α_1 are valid when $0 < c < 1$, $a = 0$, or when $c = 0$, $0 < a < 3$.

NORMED LINEAR SPACES AND MATRICES

K. Fan

In a joint paper with Glicksberg,[2] several types of rotundity of the spheres in normed linear spaces are studied. As a departure from notions of uniform convexity, and local uniform convexity, found in the literature, a number of related properties are defined and examined in terms of their mutual relations and various consequences.

The theory of nonnegative matrices is known to have important applications to the criticality theory of reactors. A topological method related to the Brouwer fixed-point theorem has been applied in the proof of certain known properties of nonnegative matrices, and this method has revealed several new properties of such matrices.

Some of the properties of nonnegative matrices have been generalized in the following way: Continuous functions f_i and g_i $(1 \leq i \leq n)$ are defined over an $(n-1)$-dimensional simplex S, where each f_i is convex on S and nonpositive on the ith face of S, $\max_i f_i(x) > 0$ for all $x \in S$, each g_i is concave and positive on S. An equilibrium point $\hat{x}$ is one for which there exists a real λ, called an equilibrium value, such that $g_i(\hat{x}) = \lambda f_i(\hat{x})$ for all i. It can be shown that there is a unique equilibrium value λ and a unique equilibrium point $\hat{x}$. The

---

[1] W. Kofink, *Studies of the Spherical Harmonics Method in Neutron Transport Theory. I. The Relation Between* P_{L-} *and Gauss Quadrature Solutions of the Milne's Problem*, ORNL-2334 (May 27, 1957); *II. Behavior of the Solution of the Milne Problem with Anisotropic Scattering for* L, ORNL-2358 (July 1957).

[2] K. Fan and I. Glicksberg, *Proc. Natl. Acad. Sci. U.S.* 41, 947 (1955).

value λ is characterized by certain minimax and maximin relations, known to hold in the linear case, and further natural generalizations are found to carry through.

Closely related to nonnegative matrices are the so-called M matrices, whose inverses are nonnegative. More precisely, the diagonal elements of an M matrix are positive, along with all principal minors, and the off-diagonal elements are nonpositive. If α, β, and γ are subsets of the indices, and $A(\alpha)$ denotes the principal minor of a matrix A formed by those rows and columns whose indices lie in α, with $A(\alpha) = 1$ when α is the null set, define

$$\Phi(A;\alpha,\beta,\gamma) = \frac{A(\alpha\cap\beta)\; A(\alpha\cap\gamma)\; A(\beta\cap\gamma)\; A(\alpha\cup\beta\cup\gamma)}{A(\alpha)\; A(\beta)\; A(\gamma)\; A(\alpha\cap\beta\cap\gamma)} \;.$$

It can be shown that if A and B are both M matrices, and $A \leqq B$, then

$$\Phi(A;\alpha,\beta,\gamma) \leqq \Phi(B;\alpha,\beta,\gamma) \leqq 1 \;,$$

and the theorem can be generalized to functions similar to Φ but defined over a larger number of sets. Also, Szász's inequalities for positive definite Hermitian matrices are shown to be valid for M matrices and their inverses. The kth term of these inequalities is the product of all k-rowed principal minors raised to a power necessary for homogeneity.

MATRIX NORMS

A. S. Householder

Because of their many applications in numerical analysis, a systematic study of matrix norms has been under way for some time.[3–5] First to be considered was their application to questions relating to the convergence of iterative methods[3,4] and to the localization of eigenvalues. Next, they were used to obtain rigorous bounds for computational errors in the solution of the systems of difference equations used to approximate certain differential equations.[5]

Similar bounds have been obtained for still other partial difference systems, and the same methods turn out to yield one of the factors required for the estimation of truncation error. In fact, if A is the matrix of the system in which each unknown is the value of the dependent variable at a lattice point, then in many cases certain norms $\|A^{-1}\|$ can be calculated explicitly, or at least bounded, as a function of the number of lattice points, and this quantity enters as a factor into both types of error: truncation error and the error generated by the computation itself. These bounds are intrinsic to the matrix A and hence to the particular system of algebraic equations being solved, and make no reference to any particular method of solving. They represent limits of achievable accuracy in solving, by whatever method, the system under consideration.

[3] A. S. Householder, *On Norms of Vectors and Matrices*, ORNL-1756 (Aug. 26, 1954).

[4] A. S. Householder, *On the Convergence of Matrix Iterations*, ORNL-1883 (May 18, 1955).

[5] A. S. Householder, *Generated Error in the Solution of Certain Partial Difference Equations*, ORNL-2230 (Dec. 12, 1956).

For any system $Ax = b$, and in terms of any preassigned norm, the product

$$\gamma(A) = ||A|| \; ||A^{-1}||$$

has been defined to be the "condition" of the system referred to that particular norm. For whatever matrix and whatever norm, $\gamma \geqq 1$, but a large γ signifies in general that when any particular routine is applied for solving the system or inverting the matrix, then the norm of the error can be expected to be correspondingly large. Most direct methods of solving a system require the formation of a sequence of matrices $L_1, L_2, \ldots$, each of some simple form, and such that the product $\ldots L_2 L_1 A = R$ is itself of some simple form. It is important that none of the transforming matrices L_i have a large value of γ, since, if it does, there is danger that the transformed system will be poorly conditioned. These considerations provide certain explicit guide rules for selecting the matrices L_i, the rules being consistent with accepted rules of thumb while being more explicit and more rigorously grounded.

The above remarks indicate some of the applications of matrix norms, but in the course of making these and other applications it became necessary to develop the theory more fully than is done in the literature. When a suitable postulational definition is used, it can be shown that any norm is associated in a certain way with a certain convex body in the complex Euclidean n-space, and conversely. Thus a complete geometric characterization is available. It is readily shown that for any matrix A, a norm of A cannot be less than the spectral radius (the maximal modulus of the eigenvalues), but that there exist norms whose value for a particular A is as close as may be desired to the spectral radius. In fact, for a large class of matrices this minimum can be achieved. Any such norm is called "minimal" for the given matrix, and the associated convex body is "minimizing." It is easily shown that the spectral radius has certain convexity properties over the class of all matrices for which a given convex body is minimizing, and the question arises, then, of otherwise characterizing these matrices. This has been done for all convex bodies of two particular classes. One is the class of all ellipsoids, and the matrices represent a certain generalization of normal matrices. The other class is of convex bodies that are, in a sense, generalized polyhedra, and the matrices turn out to be closely related to stochastic matrices. They are, in fact, except for a diagonal matrix multiplier, all similar to nonnegative matrices, such as are applied by Birkhoff and Varga[6] in their recent discussion of criticality theory.

LINEAR EQUATIONS AND THE INVERSION OF MATRICES

A. S. Householder

A large fraction of all scientific computation reduces, implicitly or explicitly, to the solution of systems of linear algebraic equations or the inversion of matrices. This is an

[6]G. Birkhoff and R. S. Varga, *J. Soc. Ind. Appl. Math.* 6, 354–77 (1958).

elementary problem when the system is small, but for large systems it is easily possible to lose all significance through the buildup of rounding errors. Many methods for solving the problem have been published, some obviously identical in principle, some quite different, at least in appearance. Some appear to be well adapted to systems of special form only; some seem to be general. It is important to have a diversified library of routines for solving systems of equations, but it is essential to make an intelligent selection of the methods to be coded. Unfortunately, the literature is extensive and chaotic, and except for counting of operations, there is little besides intuition and experiment to guide the selection.

A fairly exhaustive, critical survey of known methods has been under way. Some of the results of the survey have been published or are being published (see "Publications," this report). Also, a program for revising and adding to the library of matrix routines is under way, both for the Oracle and for the IBM-704 and IBM-709. The following account summarizes a few of the conclusions to date.

Methods based upon infinite iterations will probably decline in popularity, even for solving the extremely large, sparse systems which arise in the solution of partial differential equations. Generally speaking, the rate of convergence declines as the size of the system (the fineness of the mesh) increases. However, for high precision it may be desirable to first apply a direct method and then improve by a few iterations. Also, there is always the possibility that new and more rapidly convergent methods will be discovered, especially for systems of special form.

On the other hand, n-step iterations, such as the method of conjugate gradients, may be advantageous, especially for systems for which some approximate solution may be available in advance. These methods represent a certain compromise between the direct methods and the infinite iterations, in that, like the latter methods, they provide a sequence of approximations that approach the true solution (apart from rounding errors), whereas they are closed and yield the true solution (apart from rounding errors) after a finite number of steps.

All known direct methods can be classified as "analytic" and "synthetic," although the classification represents the point of view rather than the basic formulas employed. One of the synthetic methods is the method of "modification," based upon an identity due to Woodbury, and it turns out to include as a special case the method of "tearing," or "diakoptics," advocated by G. Kron, although the fact is by no means clear from Kron's writings. It appears that the method can be spectacularly successful for particular sparse systems, but it is not clear that it could be useful in general.

Another synthetic method is the escalator method, in which the inverse of a matrix of order n is obtained in terms of the inverse of a submatrix. It is easy to see that such a method could be very useful in particular cases. As a general method, however, it has nothing to recommend it. Moreover, it has the serious drawback at present that no way is known for appraising the stability of the method a priori (its sensitivity to rounding errors), and there is even reason to suspect that an a priori appraisal cannot be made.

The analytic methods include the classical methods of elimination, and all reduce to the following: A sequence of matrices, $L_1, L_2, \ldots$, is formed such that the product

$$\ldots L_2^{-1} L_1^{-1} A = R$$

is of some easily invertible form. Naturally, the matrices L_i must themselves be easily invertible. Also, operation upon A^T instead of A is carried out if this seems more convenient.

The following choices are known to be possible: each L_i is orthogonal, and R is upper triangular; each L_i is lower triangular, and R is orthogonal by rows; each L_i is lower triangular, and R is upper triangular; each L_i differs in one column from the identity, and R is diagonal and is possibly the identity. The same process can be applied to the matrix CA, where C is any nonsingular matrix, possibly A^T when A is not positive definite, or possibly some known approximation to A^{-1}.

In simple elimination each L_i is lower triangular, and R is upper triangular. The stability of this method as applied to a positive definite matrix A was established in a well-known paper by von Neumann and Goldstine, and for other matrices they recommend premultiplication by A^T. This was suggested because they failed to establish stability otherwise, although the number of operations required is approximately doubled.

It appears, however, that when A is not positive definite the same degree of stability can be secured by taking the matrices L_i to be orthogonal. A method of this type has already been coded and used on the Oracle, but with the matrices L_i as plane rotations. Since $n(n-1)/2$ rotations are required in general, this method is rather costly in terms of the operational count. It turns out, however, that orthogonal matrices of somewhat more general character can be employed. These are of the form $I - 2ww^T$, where w is a unit vector, and, at most, $n - 1$ of these are required. This permits a substantial reduction in the number of operations and is considerably more economical than the method of forming $A^T A$ and eliminating. The reduced operational count should not only save time but also reduce the sources of rounding errors. An auxiliary advantage of the orthogonal transformations, in general, lies in the fact that if the columns of the matrix are prescaled to unit norms or less, these norms remain invariant and no rescaling is required up to the point of inverting R. This is especially important for a fixed-point machine.

THE EIGENVALUE PROBLEM

A. S. Householder

The method for computing the eigenvalues of a symmetric matrix in use on the Oracle (the Givens method[7]) was first developed at ORNL and is now in wide use in computing

[7] W. Givens, *Numerical Computation of the Characteristic Values of a Real Symmetric Matrix*, ORNL-1574 (Feb. 19, 1954).

laboratories both here and abroad. A detailed error analysis has provided rigorous error bounds, and from all points of view the method can probably be regarded as unsurpassed.

The situation is rather different with respect to nonsymmetric matrices, where the problem is considerably more complicated. A survey of methods known for this problem led to some rather striking and unexpected conclusions. In most methods for the explicit expansion of the characteristic polynomial, the derivation (in more or less disguised form) is from a method of Krylov's,[8] which makes use of the fact that the coefficients satisfy a system of equations with the matrix made up of the columns

$$v,\ Av,\ A^2 v,\ \ldots\ ,$$

where v is an almost arbitrary vector. Such a system, however, will be more or less ill-conditioned according to the choice of v, and no way is known for ascertaining in advance how well or how poorly conditioned the system may be. This is the case for symmetric as well as for nonsymmetric matrices. On the other hand, the Givens method starts out, though in a disguised manner, to generate the equations for the coefficients of the polynomial, but never solves them completely. It proceeds only to the point of replacing the original matrix by a similar one of much simpler form, and the stability of the method seems to be due to the fact that the characteristic polynomial is never required explicitly.

The methods of the class considered can all be described as follows: If the iterated vectors v_i can be made columns of a nonsingular matrix V, then the coefficients of the characteristic polynomial are elements of a vector f which satisfies

$$Vf = A^n v\ ,$$

and all methods amount to methods of reducing the matrix V to some simpler form, possibly triangular, possibly even diagonal. Hence they require the formation of a matrix L such that $V = LR$, where R is triangular or diagonal. In that event it turns out that $L^{-1}AL$ is then subtriangular, possibly even tridiagonal, and the characteristic polynomial can be obtained from it by a simple recursion. But to do so amounts ultimately to the formation of $R^{-1}L^{-1} = V^{-1}$ and its multiplication by $A^n v$; hence if V is ill-conditioned the errors can be great. On the other hand there is evidence that if L is orthogonal and one works directly with the determinant of $L^{-1}AL - \lambda I$, the computations may be stable. This is precisely the method of Givens when A is symmetric. For the nonsymmetric case he has also suggested the formation of the orthogonal matrix L as a product of plane rotations. However, here, as in the case of matrix inversion, operations can be saved by selecting matrices of the form $I - 2ww^T$, $w^T w = 1$.

---

[8]A. N. Krylov, *Izvest. Akad. Nauk S.S.S.R. Otdel. m.e.n.* 1931, 491–539.

CHARACTERISTIC VECTORS OF SYMMETRIC MATRICES

A. C. Downing D. J. Wehe

Experience at ORNL and elsewhere has revealed difficulties in obtaining accurate characteristic vectors for symmetric matrices. This appears to be especially true for those of tridiagonal form. There is no known error analysis for any of the many methods proposed in the literature.

An experimental comparison of the Givens method,[9] presently used at ORNL, with that of Wilkinson[10] has been started. Each of these can be considered as employing one cycle of an inverse power iteration but using different initial estimates.

GENERAL PROGRAMING ACTIVITIES

An important part of the programing effort of a computer installation is the preparation of general-purpose programs which are not directly related to any specified problem, but which are capable of improving programing efficiency, and therefore the usefulness of a computer, or which are related to a wide variety of problems to which they may be applied by a suitable choice of parameters. This section is devoted to a description of the programs of the first kind, on which work has been done during the period of the report.

Programed Arithmetic. – The Oracle is a fixed-point machine without provision for automatic floating-point operation. Therefore such operations must be programed. Interpretive routines which interpret machine orders for (40,40) and (28,12) floating-point numbers have been available for some time. A similar routine for (40,40,40) floating-point complex numbers has been completed and debugged but has not yet been added to the Compiler library. An (80,40) double-precision floating-point routine has been written but has not been completely debugged. The three-address (8,32) floating-point system has been extended to include additional operations, resulting in the super "B" code. Several sets of programed arithmetic have been added to the Compiler library; they are listed under "Special Codes."

Oracle Compiler.[11] – Most programing is not done in the raw language of a computer, but in a language which is translated by a compiler and translator into machine code. Compiling programs include simple assemblers, floating-address and symbolic assemblers, and, more recently, algebraic translators.

In the past, several subroutine locators and compilers had been used to some extent in Oracle coding. Early in the period of this report, the present Oracle Compiler, planned and written by M. E. LaVerne of the Neutron Physics Division with the cooperation of

---

[9] R. H. Dykaar and C. D. LaBudde, *The NYU Matrix Codes*, NYO-6484 (March 15, 1956).

[10] J. H. Wilkinson, *Computer J.* 1, 90 (1958).

[11] *Math. Semiann. Prog. Rep. Feb. 28, 1957*, ORNL-2283, p 9–10.

members of that division and the Mathematics Panel, was completed and placed in general service. That it has proved useful is shown by the fact that almost all current coding for the Oracle is being done in its language, which is similar in appearance to machine code but differs in important respects. An earlier version of the Compiler was planned and written by R. R. Bate, USA, while he was associated with ORNL.

A programmers' manual for the Oracle Compiler[12] has been prepared by the Mathematics Panel from working papers used in the preparation of the Compiler. The manual contains information on the use of the Compiler and a glossary of programing terminology that includes terms peculiar to Oracle Compiler programing.

Subroutine Library. – Since considerable time, effort, and skill go into the preparation of a program for a computer, the library of prepared programs forms an important adjunct of any computer installation. In the case of the Oracle, this library, excluding engineering test routines, may be divided into three main categories: complete self-contained programs, service routines, and items in the Compiler library. Additions to these categories are listed under "Special Codes."

In order for the subroutines in the library to have value, the programmer must have detailed information concerning each program. In order to assure that reports on such routines meet certain minimal standards, a subroutine committee, consisting of A. C. Downing, N. M. Dismuke, R. R. Coveyou, and A. A. Grau, was formed to supervise the issuance of such reports. In general, reports for subroutines are written by the contributor of the program. In order for a report to be acceptable, the program must have been tested with reasonable thoroughness, and the report must have good literary form and must contain a comprehensive and complete description of the function and use of the program. The reports for mathematical codes contain a discussion of the methods employed and an analysis of error. Flow charts and the code in compiler language are usually included. These reports have been issued to personnel on the Oracle distribution list, and are revised and reissued as required. Reports are now available for practically all items in the Compiler library.

Reports of some older subroutines have been rewritten and reissued. Obsolete library codes have been discarded.

Conversion Routines. – The new input-output system with which the Oracle was equipped in June 1957 permitted a revision of the input and output conversion codes used by the Oracle. First, the input or output information can be expressed in a notation much nearer the usual algebraic notation, in many cases the same notation; second, output can be obtained by a single character punch or record order, which permits the development of routines where the entire binary-to-decimal conversion process is done in the free computing time available between punches or recordings.

[12] M. E. LaVerne *et al.*, *Programmers Manual for the Oracle Computer*, ORNL CF-57-8-92 (Aug. 23, 1957).

COMPILER REVISION

J. Harrison

In 1957, the preparation of a compiler which would meet more satisfactorily the needs of Oracle programmers was initiated. New features comprise an optional typewriter edit of the private library, seven additional shorthand items, optional format for the final program edit, optional arrangement of the final program, which will eliminate transfers from the end of one item to the beginning of another, and conversion of (40,40) floating-point numbers. Included will be a more extensive error monitoring system with correction facilities. Key words will be required only through the last nonzero one. Absolute addresses will be used when correcting items. Items from the private library, the public library, or paper tapes may be corrected and stored on the private library tape, or used only for the present compiling, or both. An item may be corrected and placed in the private library, and it may be simultaneously modified with different corrections and stored for use in the present compiling. If words are added or deleted from an item, the addresses will be changed accordingly. There will also be a correction check feature which compares the word to be corrected with the word the programmer expects to correct.

At present the nearly completed first segment provides for monitoring the private library, loading paper-tape items and correction tapes, computing length of alphameric items, checking for key-word errors, correcting items, and adjusting addresses of corrected items.

Plans for the revised Compiler provide for an unlimited number of shorthand items and automatic counting of all key words.

Compiler Program. – One new device written for the Compiler is a correlator for code and storage addresses. As implied, this scheme manufactures a list for every item containing the relative address of each word, along with the memory address of that word. Extra words are distinguished by adding 100, 200, or 300 to the relative address. This list is needed for correcting items, adjusting addresses, and changing required cross-reference addresses.

Another feature is a method for measuring the length of items. The item is loaded without a transfer of control into the memory where pseudo subroutine entries have previously been generated. These entries transfer back to the code, with the address of the first word following the item in the accumulator, so that the length of the item may be calculated.

The Compiler employs numerous subroutines such as the versatile Duplex Do-All (00 E58), which operates on one or two lists simultaneously, and the List Edit (00 FFF), which searches a list for a portion of a given word.

A drive-sorter subroutine has been written which picks an item to be corrected off the proper drive. Other subroutines convert numbers to alphameric, modify item and block numbers for outputting, and move lists up or down a required number of places.

ORACLE BINARY INTERNAL TRANSLATOR (ORBIT)

J. J. Andrews J. Harrison A. A. Grau

The Oracle Compiler, though it has demonstrated its utility, is limited in the following ways:

1. Coding involves much repetitive labor that is capable of automation.

2. It is rarely possible for the proposer of a problem to code it, because he is generally not conversant with the highly technical coding language employed.

3. Considerably more time is required to code a problem than to flow-chart it, whereas the coding time could be much less. In addition, much time is taken in tracking down errors in machine code that have little to do with the intrinsic logic of the problem.

These and similar considerations have promoted a wide interest in methods of automatic programing. Among these methods, the ones receiving most attention are algebraic coding systems. Leading examples are FORTRAN (formula translation) and IT (internal translation).

Since an automatic programing system for the Oracle would clearly be an important asset, consultation was held by members of the Panel with J. W. Carr III of the University of Michigan in November 1957. As a result, a decision was reached to construct a translator for the IT language, first used by A. J. Perlis of the Carnegie Institute of Technology. This language is in use on a number of machines, and the method of translation, consisting in the processing of successive pairs of characters, is both simple and capable of easy extension to more elaborate schemes.

The IT language is independent of machine language and facilitates the programing of mathematical problems directly from flow charts. The instructions of the IT language are statements of several types. These correspond in a natural way with the processes found in flow charts. They include the following:

1. substitution statement – essentially a formula where the result of performing the operations in the right member is to be substituted for the variable on the left;
2. iteration statement – necessary in the writing of loops;
3. linkage statement – since the statements are normally executed in the order in which they are written, a linkage statement is used to interrupt this sequence; the interruption may be conditional, in which case it depends on whether a stated equality or inequality holds;
4. input statement;
5. output statement;
6. halt statement;
7. extension statement – permits a call for the execution of a subroutine before the resumption of the normal sequence of statements.

In the ORBIT version of IT, a "continue" statement has been included to permit a shortening of the final machine code in some cases.

Sets of flow charts for IT translators for other machines were consulted; among these a set obtained from Ohio State University for the IBM-650 proved very useful. The original flow charts for ORBIT required extensive revision when it was found that memory requirements were excessive. The program is at present about 90% complete; many short programs have been successfully translated into compiler language. Because of the memory limitations, a revision is needed which will provide memory space to accommodate other desirable features. The flow charts for this revision have been written.

A programmers' manual is in preparation and will be in final form at the time the system is ready for release. It is planned to inaugurate the new programing system with lectures, where the features of the system will be explained.

ORBIT SUBROUTINES

M. Feliciano G. C. Caldwell E. L. Cooper A. A. Grau

The ORBIT system consists of the main translator program and a series of compiler items which are incorporated into the subject programs as needed in the process of compilation. They may be classified into (1) input and output, (2) arithmetic and special functions, and (3) service items.

The input and output subroutines are activated by the corresponding statements in IT language and require a definite alphanumeric format.

The arithmetic subroutines were established to equip ORBIT with the necessary tools to permit the use of (8,32) floating-point operation. Consideration was also given to the use of binary integers in exponentiation and in conversion to and from (8,32) floating-point numbers. The special functions include some of the elementary functions. At present the question of the most appropriate choice of more functions to be used by ORBIT is under close scrutiny.

The service subroutines include the error stop, which directs the machine to type the operation and operands yielding an inadmissible result. The statement linkage item, as the name implies, provides for the execution, in machine language, of linkage statements appearing in the program written in IT language.

COMPILER LIBRARY LINKAGE

N. M. Dismuke

Early in the development of ORBIT it seemed reasonable to attempt to include a large portion of the Compiler library in the extension list for the translator. However, a review of the linkage employed by the library routines made clear the necessity for maintaining separate libraries for programmer and ORBIT coding.

The Compiler library is composed of more than 300 routines, of which possibly half would be useful ORBIT extensions. A fine enough classification to enable the ORBIT program to conveniently code the linkage would require almost as many linkage types as routines to be classified. A coarser classification of linkage is possible if its purpose is to convey information to programmers. For the latter purpose about half the routines fall into a dozen linkage categories.

SYMBOLIC ASSEMBLY PROGRAM

J. J. Andrews A. A. Grau

As an intermediate step between the present Oracle Compiler and ORBIT, a symbolic assembly program for the Oracle was planned. A substantial part of the code has been written and tested. This program would permit programing for the Oracle by use of symbolic addresses and mnemonic symbols for operations and pseudo operations. The floating addresses are desirable because of the decrease in bookkeeping that their use effects, and the use of mnemonics is advantageous to many programmers, especially new ones.

Because interest shifted to the possibility of constructing a translator for the IT language without such an intermediate program, work on the symbolic program was suspended to permit work on the translator.

CURVE-PLOTTER PRINT-OR-GRAPH OUTPUT SYSTEM (POGO)

A. H. Culkowski C. T. Fike A. A. Grau

POGO consists of various routines which enable a programmer to code for alphameric curve-plotter output and yet change to any other type of output with ease and with no additional coding. Output via typewriter will be equivalent to curve-plotter output in virtually every respect. One frame of single-spaced curve-plotter output will be tantamount to exactly one page of double-spaced typewriter output or a half page of single-spaced output, if the margins are set for a width of 51 characters.

Basically, if curve-plotter output is wanted but it is not desirable to wait for the return of curve-plotter prints while debugging, the debugging may be carried on via console typewriter or off-line printer. On the other hand, even if extensive reports via typewriter are the primary aim of a code, a quick curve-plotter print in advance of extensive typing may prove of value.

There are three basic POGO orders which are interpretive "type" orders for the curve-plotter and which simulate machine output orders via paper tape.

To facilitate changing from one mode of output to another, there is the POGO output changer. It is not a closed subroutine, but may be compiled along with the master program by a suitable reference to it, or it may be obtained in Servi-seer form at the console.

The POGO carriage positioner is an aid in determining the x and y coordinates of the first line of a frame or the number of carriage returns down and the number of spaces over from the left-hand margin for the first line of typewriter output. Use of this routine allows for synchronization of curve-plotter frames and printed pages.

At present two decimal output subroutines (08B and 08D) have been rewritten in POGO form and are available as items 28B and 28D. It is anticipated that other output subroutines of frequent utility will be rewritten in the POGO convention.

Figures 1 and 2 illustrate the type of compatible output possible with POGO coding.

-2-

THE CURVE-PLOTTER PRINT-OR-GRAPH OUTPUT SYSTEM

(POGO)

Arline Culkowski

C. T. Fike

A. A. Grau

General Description and Purpose. The curve-plotter print-or-graph output (POGO) system of coding is designed to make programmed alphameric curve-plotter output interchangeable with print output via typewriter with very little modification. One frame of single-spaced curve-plotter output will be equivalent to exactly one page of double-spaced typewriter output if the margins are set for a width of 51 characters, or a half-page of single-spaced typewriter output.

Use of the POGO Output Changer Item 280, will make the necessary changes to convert from curve-plotter output to the equivalent in paper tape, magnetic tape, or console typewriter.

POGO coding will be desirable where it is convenient to have a flexible output of this kind. If, basically, curve-plotter output is wanted but it is not desirable to wait for the return of curve-plotter prints while debugging, the debugging may be carried on via console typewriter or off-line printer. On the other hand even if extensive reports via typewriter are the primary aim of a code, a quick curve-plotter print in advance of extensive typing may prove of value.

Basic POGO Output Orders. There are three basic POGO output orders which are interpretive "type" orders for the curve plotter, each of which simulates a machine output order via paper tape. In the POGO system all output must be programmed using these orders which are a particular type of subroutine entry, and are written as follows:

1. Interpretive 84 order: w-2: ff000 00284
   w-1: 84aaa 43 w
2. Interpretive 88 order: w-2: ff000 00288
   w-1: 88aaa 43 w
3. Interpretive 8c order: w-2: ff000 0028c
   w-1: 8caaa 43 w

POGO Output Changer Item 280. The POGO Output Changer is a convenient means of changing the mode of output of a program coded to use POGO output. It is not a closed subroutine and may be compiled along with the master program by a suitable reference to it. It is also available in serviseered form for use at the console.

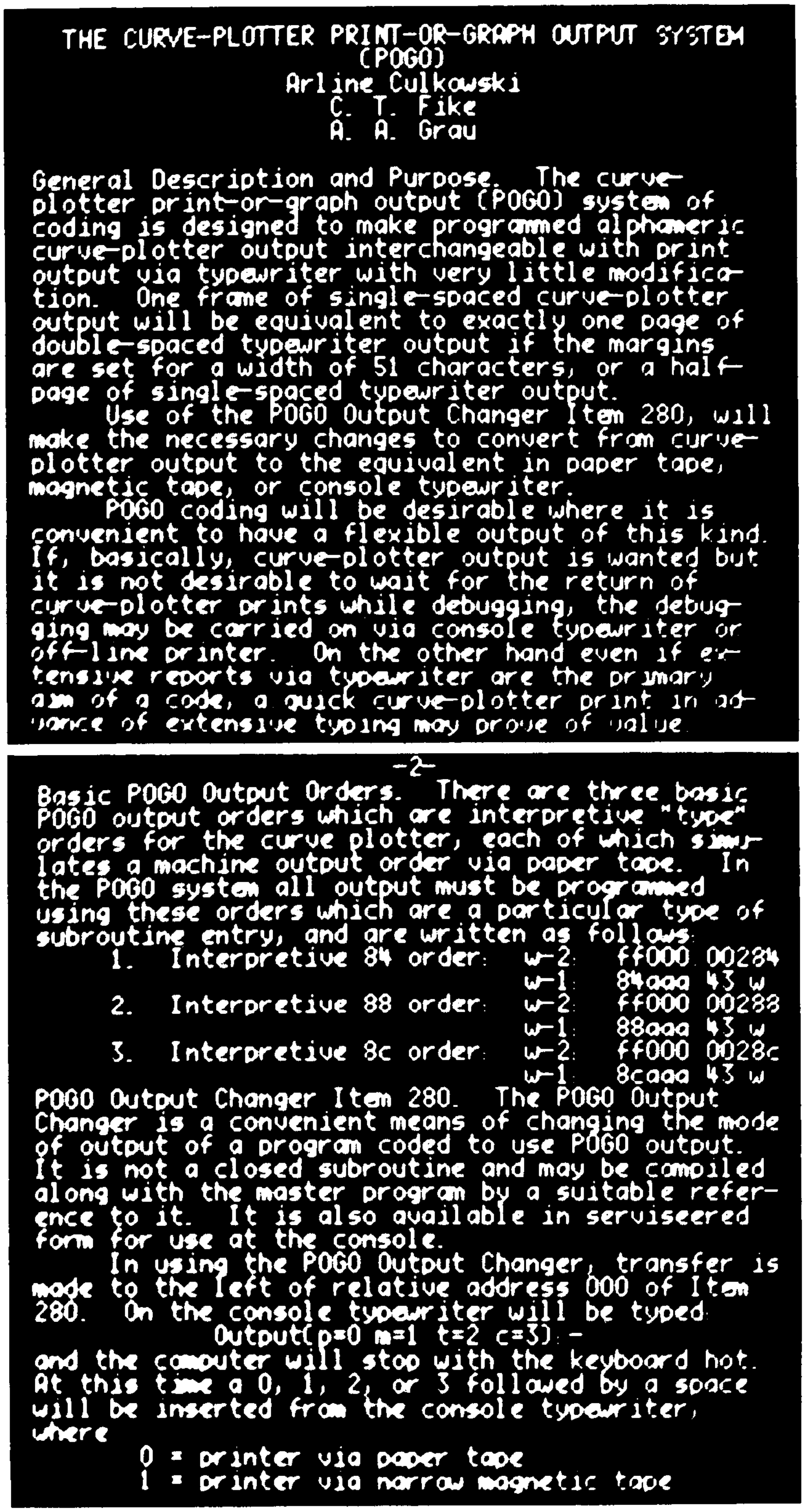
In using the POGO Output Changer, transfer is made to the left of relative address 000 of Item 280. On the console typewriter will be typed:

Output(p=0 m=1 t=2 c=3):-

and the computer will stop with the keyboard hot. At this time a 0, 1, 2, or 3 followed by a space will be inserted from the console typewriter, where:

0 = printer via paper tape
1 = printer via narrow magnetic tape

Fig. 1. Example of POGO Output by Peripheral Printer. (Reduced 38%).



THE CURVE-PLOTTER PRINT-OR-GRAPH OUTPUT SYSTEM
(POGO)
Arline Culkowski
C. T. Fike
A. A. Grau

General Description and Purpose. The curve-
plotter print-or-graph output (POGO) system of
coding is designed to make programmed alphameric
curve-plotter output interchangeable with print
output via typewriter with very little modifica-
tion. One frame of single-spaced curve-plotter
output will be equivalent to exactly one page of
double-spaced typewriter output if the margins
are set for a width of 51 characters, or a half-
page of single-spaced typewriter output.
Use of the POGO Output Changer Item 280, will
make the necessary changes to convert from curve-
plotter output to the equivalent in paper tape,
magnetic tape, or console typewriter.
POGO coding will be desirable where it is
convenient to have a flexible output of this kind.
If, basically, curve-plotter output is wanted but
it is not desirable to wait for the return of
curve-plotter prints while debugging, the debug-
ging may be carried on via console typewriter or
off-line printer. On the other hand even if ex-
tensive reports via typewriter are the primary
aim of a code, a quick curve-plotter print in ad-
vance of extensive typing may prove of value.

-2-

Basic POGO Output Orders. There are three basic
POGO output orders which are interpretive "type"
orders for the curve plotter, each of which simu-
lates a machine output order via paper tape. In
the POGO system all output must be programmed
using these orders which are a particular type of
subroutine entry, and are written as follows:

1. Interpretive 84 order: w-2: ff000 00284
   w-1: 84aaa 43 w
2. Interpretive 88 order: w-2: ff000 00288
   w-1: 88aaa 43 w
3. Interpretive 8c order: w-2: ff000 0028c
   w-1: 8caaa 43 w

POGO Output Changer Item 280. The POGO Output
Changer is a convenient means of changing the mode
of output of a program coded to use POGO output.
It is not a closed subroutine and may be compiled
along with the master program by a suitable refer-
ence to it. It is also available in serviseered
form for use at the console.
In using the POGO Output Changer, transfer is
made to the left of relative address 000 of Item
280. On the console typewriter will be typed:
Output(p=0 m=1 t=2 c=3): -
and the computer will stop with the keyboard hot.
At this time a 0, 1, 2, or 3 followed by a space
will be inserted from the console typewriter,
where
0 = printer via paper tape
1 = printer via narrow magnetic tape

Fig. 2. Example of POGO Output by Curve Plotter.

DESIGN OF FIXED-POINT ITERATIONS

A. C. Downing

In the application of the methods of functional iteration to the solution of nonlinear problems, the accuracy of a theoretically convergent iteration depends ultimately upon the rounding errors committed during the final cycle of the iteration. In designing the computational algorisms for performing such iterations, the usefulness of an important property of fixed-point digital computers has been generally overlooked. This property is the ability to perform exact addition modulo one or two, even though overflow occurs.

The recursion relation for a functional iteration can always be written in the form

$$f_{\nu+1} = f_\nu + \rho(f_\nu)g(f_\nu) \ ,$$

where the residual $\rho(f_\nu)$ tends to zero as ν increases without bound. The rounding errors in $\rho(f_\nu)g(f_\nu)$ may be minimized in a large class of problems by computing

$$\left\{ \left[2^k \rho(f_\nu) \right] g(f_\nu) \right\} 2^{-k} \ ,$$

provided it is known a priori that $|\rho(f_\nu)| < 2^{-k}$. The previous iteration always provides a lower bound for k. In this manner it is possible to continue the iteration until the interval of uncertainty of the final f_ν is minimal. For second-order iterative procedures in a binary computer with β places to the right of the binary point, one can therefore always obtain

$$|f_\nu - f(\text{exact})| \leqq 2^{-\beta-1} + O(2^{-2\beta}) \ .$$

Each of the applications mentioned below is a second-order iteration.

The calculation of $2^k \rho(f_\nu)$ may be carried out by single-precision fixed-point methods whenever $\rho(f_\nu)$ is a sum of terms, each of which is at most quadratic. In most computers, repeated division will yield a double-length quotient for a single-precision divisor. Such terms are readily included in the sums defining the residual $\rho(f_\nu)$.

As a simple example, consider the calculation of the cube root of a, using Newton's method. The algorism may be carried out in the form

$$x_{\nu+1} = x_\nu - \left[\left(x_\nu^2 2^{+k} - \frac{a}{x_\nu} 2^{+k} \right) \frac{1}{3x_\nu} \right] 2^{-k} \ .$$

Here $$\rho(x_\nu) = x_\nu^2 - \frac{a}{x_\nu} \ , \qquad g(x_\nu) = \frac{1}{3x_\nu} \ .$$

It has been shown that this technique can be used to improve an approximate inverse of a matrix or a set of approximate characteristic vectors of a matrix. In each of these problems the correction term $\rho(f_\nu)g(f_\nu)$ involves the product of three matrices. Hence the rounding errors would ordinarily be of the order $n^2 2^{-\beta-1}$, where n is the order of the matrix.

A detailed analysis has been made of the application of this technique to the calculation of the cosine and sine of an angle of a right triangle whose two sides, a and b, are given. This is precisely the problem of computing the plane rotations used in transforming matrices to simpler forms. An algorism has been developed which obtains approximate solutions γ_ν and σ_ν for the equations

$$\gamma y - \sigma x = 0 \;,$$
$$\gamma^2 + \sigma^2 = 1 \;,$$

with minimal intervals of uncertainty for both γ and σ. This problem has previously been studied by Goldstine, Murray, and von Neumann;[13] by Givens;[14] and by Householder.[15]

[13]H. H. Goldstine, F. J. Murray, and J. von Neumann, *J. Assoc. Computing Mach.* 6(1), 59 (1959).

[14]J. W. Givens, *Numerical Computation of the Characteristic Values of a Real Symmetric Matrix*, ORNL-1574, p 23 (Feb. 19, 1954).

[15]A. S. Householder, *The Generation Error in Digital Computation*, ORNL-1983, p 43 (Oct. 11, 1955).

BIOMETRICS AND STATISTICS

SEX RATIO IN HUMANS AS A FUNCTION OF PARENTAL AGES AND BIRTH ORDER

A. W. Kimball G. J. Atta E. Leach

A weighted multiple linear regression model was fitted to U.S. birth data giving simultaneously the parental ages, birth order, and sex of the offspring for the year 1955. This work was done in collaboration with E. Novitski of the Biology Division. Proportion of males was taken as the dependent variable, and a second-degree polynomial in the three independent variables was fitted to the data, which include over four million births. With the Oracle it was feasible to compute a large number of regressions, each involving a different subset of the terms in the polynomial.

It was shown that variation in secondary sex ratio is independent of the mother's age but is dependent on both birth order and paternal age. It was further shown that the relationships are not linear and cannot be represented adequately without the inclusion of a component that reflects an interaction between birth order and age of the father.

A LIKELIHOOD RATIO TEST FOR A HYPOTHESIS IN A NEUROSPORA EXPERIMENT

A. W. Kimball M. A. Kastenbaum E. Leach

In the experiment, involving two biochemical mutants of *Neurospora crassa*, a test was desired for a hypothesis related to the structure of the chromosome. The problem is discussed here only in its statistical form.

Random samples are taken on eight different variables yielding the following data:

$$(x_i, y_i, z_i) \ , \qquad i = 1, \ldots, m \ ,$$

$$(u_j, v_j, w_j) \ , \qquad j = 1, \ldots, n \ ,$$

$$(r_k, s_k) \ , \qquad k = 1, \ldots, p \ .$$

The eight variables are mutually independent and have Poisson distributions with means

$$E(x_i) = \alpha_i \ , \qquad E(y_i) = \beta_i \ , \qquad E(z_i) = \gamma_i \ ,$$

$$E(u_j) = \delta_j \ , \qquad E(v_j) = \epsilon_j \ , \qquad E(w_j) = \theta_j \ ,$$

$$E(r_k) = \lambda_k \ , \qquad E(s_k) = \mu_k \ .$$

One hypothesis, referred to as the alternative hypothesis, is stated as follows:

$$\beta_i = a\alpha_i \eta \ , \qquad \gamma_i = b\alpha_i \rho \ , \qquad \text{for all } i \ ,$$

$$\epsilon_j = a\delta_j \pi \ , \qquad \theta_j = b\delta_j \sigma \ , \qquad \text{for all } j \ ,$$

$$\mu_k = c\lambda_k \phi \ , \qquad \text{for all } k \ ,$$

where a, b, and c are known constants and η, π, ϕ, ρ, and σ are unknown parameters. The second hypothesis, referred to as the null hypothesis, includes the alternative hypothesis together with the linear restrictions

$$\rho + \phi = \sigma \ , \qquad \pi + \phi = \eta \ .$$

The problem is to find a test for the null hypothesis against the alternative hypothesis.

Under the alternative hypothesis, the likelihood is found to be

$$L_A(\alpha_i, \delta_j, \lambda_k, \eta, \rho, \pi, \sigma, \phi) = \prod_{i=1}^{m} \frac{(a\eta)^{y_i} (b\rho)^{z_i} \alpha_i^{x_i + y_i + z_i} e^{-\alpha_i(1 + a\eta + b\rho)}}{x_i! \, y_i! \, z_i!} \times$$

$$\times \prod_{j=1}^{n} \frac{(a\pi)^{v_j} (b\sigma)^{w_j} \delta_j^{u_j + v_j + w_j} e^{-\delta_i(1 + a\pi + b\sigma)}}{u_j! \, v_j! \, w_j!} \times$$

$$\times \prod_{k=1}^{p} \frac{(c\phi)^{s_k} \lambda_k^{r_k + s_k} e^{-\lambda_k(1 + c\phi)}}{r_k! \, s_k!} \ .$$

Application of the maximum likelihood method yields the following simple estimates:

$$\hat{\alpha}_i = \frac{X(x_i + y_i + z_i)}{X + Y + Z} \ , \qquad \hat{\delta}_j = \frac{U(u_j + v_j + w_j)}{U + V + W} \ , \qquad \hat{\lambda}_k = \frac{R(r_k + s_k)}{R + S} \ ,$$

$$\hat{\eta} = \frac{Y}{aX} \ , \qquad \hat{\rho} = \frac{Z}{bX} \ , \qquad \hat{\pi} = \frac{V}{aU} \ , \qquad \hat{\sigma} = \frac{W}{bU} \ , \qquad \hat{\phi} = \frac{S}{cR} \ ,$$

where the capital letters represent sums over appropriate subscripts of the corresponding lower-case letters. The likelihood under the null hypothesis, $L_N(\alpha_i, \delta_j, \lambda_k, \rho, \pi, \phi)$, is obtained by substituting $\rho + \phi$ for σ and $\pi + \phi$ for η in L_A. In this case the likelihood equations for α_i, δ_j, and λ_k yield the simple estimates

$$\alpha_i' = \frac{x_i + y_i + z_i}{1 + a(\pi' + \phi') + b\rho'} , \qquad \delta_j' = \frac{u_j + v_j + w_j}{1 + a\pi' + b(\rho' + \phi')} , \qquad \lambda_k' = \frac{r_k + s_k}{1 + c\phi'} .$$

Primes are used to denote estimates under the null hypothesis and carets to denote estimates under the alternative hypothesis. Unfortunately ρ', π', and ϕ' can only be obtained by solving the following equations iteratively:

$$\frac{Z}{\rho'} - \frac{b(X + Y + Z)}{1 + a(\pi' + \phi') + b\rho'} + \frac{W}{\rho' + \phi'} - \frac{b(U + V + W)}{1 + a\pi' + b(\rho' + \phi')} = 0 ,$$

$$\frac{Y}{\pi' + \phi'} - \frac{a(X + Y + Z)}{1 + a(\pi' + \phi') + b\rho'} + \frac{V}{\pi'} - \frac{a(U + V + W)}{1 + a\pi' + b(\rho' + \phi')} = 0 ,$$

$$\frac{Y}{\pi' + \phi'} - \frac{a(X + Y + Z)}{1 + a(\pi' + \phi') + b\rho'} + \frac{W}{\rho' + \phi'} - \frac{b(U + V + W)}{1 + a\pi' + b(\rho' + \phi')} + \frac{S}{\phi'} - \frac{R + S}{1 + c\phi'} = 0 .$$

Once estimates under both hypotheses have been obtained, the likelihood ratio, Λ, is computed. Since

$$\Lambda = \frac{L_N(\alpha_i', \delta_j', \lambda_k', \rho', \pi', \phi')}{L_A\left(\hat{\alpha}_i, \hat{\delta}_j, \hat{\lambda}_k, \hat{\eta}, \hat{\rho}, \hat{\pi}, \hat{\sigma}, \hat{\phi}\right)} ,$$

it may be shown that

$$\begin{aligned} \ln \Lambda = {} & Y \ln \left(\frac{\pi' + \phi'}{\hat{\eta}}\right) + Z \ln \left(\frac{\rho'}{\hat{\rho}}\right) + (X + Y + Z) \ln \left[\frac{1 + a\hat{\eta} + b\hat{\rho}}{1 + a(\pi' + \phi') + b\rho'}\right] + \\ & + V \ln \left(\frac{\pi'}{\hat{\pi}}\right) + W \ln \left(\frac{\rho' + \phi'}{\hat{\sigma}}\right) + (U + V + W) \ln \left[\frac{1 + a\hat{\pi} + b\hat{\sigma}}{1 + a\pi' + b(\rho' + \phi')}\right] + \\ & + S \ln \left(\frac{\phi'}{\hat{\phi}}\right) + (R + S) \ln \left(\frac{1 + c\hat{\phi}}{1 + c\phi'}\right) . \end{aligned}$$

Finally the required test can be made by noting that $-2 \ln \Lambda$ has asymptotically a χ^2 distribution with two degrees of freedom.

ESTIMATION FROM OBSERVATIONS HAVING SPECIFIED BUT UNASSIGNED EXPECTATIONS

A. W. Kimball E. Leach

This problem arose from an experiment performed by R. F. Kimball of the Biology Division. The experiment dealt with the postirradiation modification of mutagenesis in

Paramecium by streptomycin. In this report, however, the problem will be discussed in general statistical terms, since it must arise in many other experimental situations.

The basic model is the conventional multiple linear regression model with one dependent random variable and several independent fixed variables. It is assumed that N independent observations $(y_1, y_2, \ldots, y_N)$ are taken on a random variable Y with variance σ^2 but with expected values that may depend on different functions of $p + 1$ fixed (measured without error) variables $(t_{0j}, t_{1j}, \ldots, t_{pj}; j = 1, \ldots, N)$ observed simultaneously with the y_j. In particular, it is assumed that y_j may have one of m different expectations given by

$$E(y_j) = \eta_{1j} = \sum_{i=0}^{p} \alpha_i^{(1)} t_{ij} \ ,$$

$$E(y_j) = \eta_{2j} = \sum_{i=0}^{p} \alpha_i^{(2)} t_{ij} \ ,$$

$$\ldots \ ,$$

$$E(y_j) = \eta_{mj} = \sum_{i=0}^{p} \alpha_i^{(m)} t_{ij} \ ,$$

and that the correct expectation for each y_j is completely determined by the values of the fixed variables associated with it. The correct assignment of expectations is unknown and, as defined, is not determined by a stochastic process. The problem is to find estimates of the $\alpha_i^{(k)}$ $(k = 1, \ldots, m)$.

In a sense, the correct assignment of expectations is a nonquantitative parameter that must be estimated before estimates of the $\alpha_i^{(k)}$ can be obtained. Since standard least-squares theory does not provide such an estimate, the following modified approach was adopted. Let the N observations on Y be divided into m groups such that n_k fall into the kth group and such that $\sum_{k=1}^{m} n_k = N$. Suppose further that the n_k observations in the kth group are associated with η_{kj}. For each such assignment consider the least-squares estimates obtained in the usual manner by minimizing

$$V = \sum_{k=1}^{m} \left[\sum_{n_k} (y_j - \eta_{kj})^2 \right] \ , \tag{1}$$

where $\sum_{n_k}$ signifies summation over the y_j assigned to the kth group. Clearly, among the m^N possible assignments, the one which yields the smallest value of V would provide estimates that are optimum in a sense very closely related to the customary least-squares criterion.

Although it has been possible to define explicitly what is meant by the least-squares solution to this problem, the minimum cannot be obtained by the usual methods of calculus.

Apparently some iterative procedure is required, and one approach seems intuitively most likely to converge. Let the predicted values of y_j obtained by minimizing V (Eq. 1) be $\hat{\eta}_{kj}$ and consider the quantity

$$V^* = \sum_{k=1}^{m} \left[\sum_{j=1}^{N} \left(y_j - \hat{\eta}_{kj}\right)^2 \right] .$$

In some sense, V^* represents the grand total of deviations from regression summed over all observations for all expected values; V^* may be subdivided into two parts:

$$\begin{aligned} V^* &= \sum_{k=1}^{m} \left[\sum_{n_k} \left(y_j - \hat{\eta}_{kj}\right)^2 \right] + \sum_{k=1}^{m} \left[\sum_{\bar{n}_k} \left(y_j - \hat{\eta}_{kj}\right)^2 \right] \\ &= V_{\text{min}} + V_{\text{max}} , \end{aligned}$$

say, where as before $\sum_{n_k}$ signifies summation over the y_j assigned to the kth group and where $\sum_{\bar{n}_k}$ signifies summation over the y_j not assigned to the kth group. Of course V_{min} is the value of V (Eq. 1) at the minimum.

Suppose that preliminary information provides an initial assignment of the N observations to m groups and that V^* is calculated on the basis of this assignment. For this particular assignment V_{min} is at a minimum, but it is quite likely that another assignment, using the same estimates, would yield a smaller value for V_{min}. At this point it seems logical that the best "estimate" of the correct assignment would be obtained by assigning each y_j to the group for which $|y_j - \hat{\eta}_{kj}|$ is the smallest. In other words the predicted value of y_j would be computed for each of the m groups, and in the second iteration y_j would be assigned to the group that, on the basis of the first iteration, provides the best prediction. Note that this procedure, in general, reduces V_{min} and increases V_{max} as calculated from the estimates for the first iteration. Using the second assignment, the process is repeated and a third assignment is obtained.

At this point, the manner in which the iterations converge, if they do converge, must be considered. Let

$$G_s = (n_{1s}, n_{2s}, \ldots, n_{ms})$$

designate the sth assignment of N observations to m groups, and let

$$G^* = (n_1{}^*, n_2{}^*, \ldots, n_m{}^*)$$

designate the particular assignment which, among all possible and admissible assignments, yields the absolute minimum sum of squares. By definition this is the assignment that provides the least-squares solution. If the process converges after s iterations, it would be expected that the two assignments G_s and G_{s+1} would be identical and the same as G^*, or at worst that

$$G_s = G_{s+2} = G_{s+4} = \cdots \quad ,$$

$$G_{s+1} = G_{s+3} = G_{s+5} = \cdots \quad ;$$

in other words, the process might oscillate between the last two assignments. In practice, this would cause no difficulty since one of the two would provide a smaller residual sum of squares and this assignment would be taken as G^*.

Several questions remain unanswered with respect to the properties of this estimation procedure. No proof of convergence has been obtained, although the solution is well defined, and intuitively it would seem that convergence is likely provided that the initial assignment is close enough to G^*. The method was applied to two artificially constructed examples (based on the model for the *Paramecium* experiment) in which half the observations were purposely misclassified in the first assignment. Convergence to the correct assignment occurred after two iterations. No errors were appended to the observations in these examples, and it is possible that, in experiments with large experimental error, convergence would not take place. On the other hand, if the data are reliable and the model appropriate, convergence may be attained rapidly.

It is probable that estimates will be biased, and the magnitude of bias will undoubtedly depend on the degree by which the alternative expected values differ from one another. A practical difficulty with the procedure is that some assignments will be inadmissible in the sense that they will not permit estimation of all parameters. This can happen, for example, if, for any iteration, $n_k = 0$ and the expected value for the kth group contains a parameter not represented in any other expected value.

On the other hand it is reassuring to know that minimization of V (Eq. 1), when the correct assignment is known, provides the exact least-squares solution given by conventional theory. The standard procedure has been modified to provide an "estimate" of the correct assignment and logically it would seem that the best "estimate" is that assignment which makes V (Eq. 1) a minimum.

Work on this problem will continue, and it is hoped that more information can be obtained about the properties of the estimates.

APPROXIMATE LINEARIZATION OF THE INCOMPLETE BETA FUNCTION

A. W. Kimball E. Leach

In radiation mortality studies, the "target" theory predicts that, for some organisms, the probability M of death from a dose D of radiation is given by

$$M = \sum_{i=p}^{p+q-1} \binom{p+q-1}{i} x^i (1 - x)^{p+q-1-i} = I_x(p, q) \quad ,$$

where $x = 1 - e^{-kD}$; p, q, and k are unknown parameters; and $I_x(p, q)$ is Pearson's[1] notation for the incomplete beta function. In principle, the parameters may be estimated from observations on M and D by an iterative least-squares procedure. In practice the use of M as an expected value presents formidable difficulties.

To avoid these difficulties various attempts were made to find a transformation that would provide an approximate linear relationship between functions of the dependent and independent variables. To be of any value the approximation had to be satisfactory over almost the entire range of $I_x(p, q)$ and had to be valid for values of p and q which were unknown and for values of x which varied over an unknown range. None of the standard approximations satisfied all of these requirements.

In connection with a sample size problem, Scheffé and Tukey[2] give, as an approximate solution to the equation $I_x(p, q) = \alpha$,

$$(p + q - 1) = \frac{1}{4} \chi_\alpha^2 \left(\frac{1 + x}{1 - x}\right) + \frac{1}{2} (q - 1) \ ,$$

where χ_α^2 is the 100α per cent point on the χ^2 distribution with $2q$ degrees of freedom. They indicate that for $0.005 \leqq \alpha \leqq 0.10$ and $x \geqq 0.9$, the error in this solution is less than 0.1%. It also does well for larger values of α but becomes inaccurate as x is decreased. We have found that the approximation can be extended to smaller values of x if the first term is multiplied by

$$\left(\frac{21}{23} + \frac{1}{19x} + \frac{1}{3} e^{-5/2x}\right)^{-1} \ .$$

Wilson and Hilferty[3] give the following approximation to χ_α^2:

$$\chi_\alpha^2 \approx 2q \left(1 - \frac{1}{9q} + y_\alpha \sqrt{\frac{1}{9q}}\right)^3 \ ,$$

where y_α is the 100α per cent point on the standard normal distribution. By combining the modified Scheffé-Tukey approximation and the Wilson-Hilferty approximation, one obtains

$$y_\alpha = A + BX \ , \tag{1}$$

where

$$A = \frac{9q - 1}{3\sqrt{q}} \ ,$$

---

[1] K. Pearson, *Tables of the Incomplete Beta-Function*, Cambridge University Press, Cambridge, England, 1934.

[2] H. Scheffé and J. W. Tukey, *Ann. Math. Statistics* **15**, 217 (1944).

[3] E. B. Wilson and M. M. Hilferty, *Proc. Nat. Acad. Sci. U.S.* **17**, 684 (1931).

$$B = -3\sqrt{q}\left(1 + \frac{2p-1}{q}\right)^{1/3} ,$$

$$X = \left[\left(\frac{1-x}{1+x}\right)\left(\frac{21}{23} + \frac{1}{19x} + \frac{1}{3}\,e^{-5/2x}\right)\right]^{1/3} .$$

Thus y_α can be computed for any x, p, and q, and the approximation

$$I_x(p, q) = \int_{-\infty}^{y_\alpha} \frac{1}{\sqrt{2\pi}}\, e^{-t^2/2}\, dt$$

can be obtained from readily available tables of the normal distribution.

Computations over wide ranges of the arguments show that Eq. 1 is satisfactory for the regression problem provided $p \leqq 80$ and $q \leqq 41$. For other values of p and q, the normal approximation,

$$y_\alpha = \frac{p(1-x) - x(q-1) - 0.5}{[(p+q-1)\ x(1-x)]^{1/2}} ,$$

is preferable since it is more accurate in these ranges and also provides an approximate linearization. It should be noted that $y_\alpha + 5$ is a probit and existing aids to probit analysis may be employed in the fitting procedure. Also, if k is unknown, separate fits for different trial values of k will be required, and $\hat{k}$ will have to be determined by graphical minimization.

ESTIMATION IN MIXED SAMPLING FROM TWO POPULATIONS

A. W. Kimball

In a certain experiment observations come from one of two populations, but it is not known either before or after an observation has been taken which population has been sampled. The frequency functions for the two populations (A and B) are

$$\left.\begin{aligned} g_A(y) &= \frac{1}{\sqrt{2\pi}}\, e^{-(y-a)^2/2} \\ g_B(y) &= \frac{1}{\sqrt{2\pi}}\, e^{-(y-b)^2/2} \end{aligned}\right\} \qquad -\infty < y < \infty ,$$

and w is the probability that an observation drawn at random comes from population A. Thus the frequency function for y_i ($i = 1, \ldots, N$) is

$$f(y_i) = w\, g_A(y_i) + (1-w)\, g_B(y_i) , \qquad -\infty < y_i < \infty .$$

From the sample of N observations, estimates of a, b, and w are required.

The method of maximum likelihood would require that

$$\prod_{i=1}^{N} f(y_i)$$

be maximized with respect to the three parameters, and the resulting equations are transcendental and virtually intractable. Fortunately in this problem the method of moments provides simple estimates. With the use of standard formulas, it may be shown that

$$E(y) = wa + (1 - w)\, b \ ,$$

$$E(y^2) = w(1 + a^2) + (1 - w)\,(1 + b^2) \ ,$$

$$E(y^3) = w(3a + a^3) + (1 - w)\,(3b + b^3) \ .$$

Letting

$$\alpha_1 = \sum y_i/N \ , \qquad \alpha_2 = \sum (y_i - \alpha_1)^2/N \ , \qquad \alpha_3 = \sum (y_i - \alpha_1)^3/N$$

and equating the sample moments to their expectations, one obtains the equations

$$w(a - b) = \alpha_1 - b \ , \tag{1}$$

$$w(1 - w)\,(a - b)^2 = \alpha_2 - 1 \ , \tag{2}$$

$$w(1 - w)\,(1 - 2w)\,(a - b)^3 = \alpha_3 \ . \tag{3}$$

From Eq. 1

$$w = \frac{\alpha_1 - b}{a - b} \ . \tag{4}$$

On substituting Eq. 1 in Eqs. 2 and 3 and letting $u = \alpha_1 - b$, $v = a - \alpha_1$, Eqs. 2 and 3 are reduced to

$$uv = \alpha_2 - 1 \ ,$$

$$uv(v - u) = \alpha_3 \ .$$

The solution of these equations leads to the estimates

$$\hat{a} = \alpha_1 + \frac{1}{2(\alpha_2 - 1)} \left[\alpha_3 + \sqrt{\alpha_3^2 + 4(\alpha_2 - 1)^3}\right] \ ,$$

$$\hat{b} = \alpha_1 + \frac{1}{2(\alpha_2 - 1)} \left[\alpha_3 - \sqrt{\alpha_3^2 + 4(\alpha_2 - 1)^3}\right] \ ,$$

whereupon the estimate of w may be obtained from Eq. 4. Asymptotic variances of the estimates may be calculated if required.

IN-PILE SLURRY AUTOCLAVE EXPERIMENT

D. A. Gardiner

A modified factorial design of eight points was constructed at the request of E. Compere of the HRP. The intent of the experimentation was to explore the effect of reactor irradiation on corrosion by thoria slurries. The factors varied were the amount of radiation, the amount of thorium present, and the power level of fission in the presence of thoria. The experiment is described fully elsewhere.[4]

[4] E. L. Compere, *Notes on Proposed In-Pile Slurry Autoclave Experiments*, ORNL CF-57-11-27 (Nov. 6, 1957).

STUDY OF MASS TRANSFER PRESSURE BUILDUP IN A 500-kw RADIATOR

D. A. Gardiner

The purpose of the study was to examine and measure the effects of the radiator inlet temperature and of the temperature drop across the radiator on the measurement of friction factor for a 500-kw radiator. R. E. MacPherson, J. C. Amos, and L. H. Devlin of the Reactor Projects Division carried out the experiment.

The experimental design chosen was a second-order rotatable design to be performed in three sequential blocks, with observations taken every 72 hr. The experiment was curtailed at the completion of the second block because of equipment failures and other considerations. A planar response surface with a time covariate was estimated and reported.

An important by-product of this study was a demonstration of the computation of friction factor by the method of least squares. This computation entirely eliminates the necessity for making a separate observation to correct for bias in the pressure measuring device.

ANALYSIS OF AIR-MONITORING DATA COLLECTED IN THE OAK RIDGE AREA

D. A. Gardiner

This problem originated with K. E. Cowser of the Health Physics Division and F. A. Gifford of the Weather Bureau Office of the Oak Ridge Operations Office, AEC. It concerned the analysis of radioactivity measurements collected at air monitoring stations located at the waste pit area, at the X-10 site proper, and at the offsite stations surrounding Oak Ridge. The data used were for 25 weeks of the period January to July 1957.

An analysis of variance was performed on the data, and differences in the mean activity at the several sites were pointed out. A striking feature of the data as demonstrated by the analysis was the very large differences in means observed among the stations surrounding a waste pit.

MAXIMUM REMOVAL OF Sr^{90} AND Cs^{137} FROM PROCESS WASTES

D. A. Gardiner

A sequential second-order rotatable design was constructed for studying the response surfaces of per cent removal of Sr^{90} and Cs^{137} over the five-dimensional flat of excess lime, excess soda ash, amount of Conasauga shale per dose, amount of recycled sludge, and amount of $CaCl_2$ per dose. The problem originated with K. E. Cowser of the Health Physics Division.

The basic design was the vertices of a five-dimensional "tetrahedron" of radius $\sqrt{5}$ upon which was concentrically superposed the five-dimensional measure polytope of radius 2. The design was performed in two blocks in the event that only the first block, designed to measure first-order effects, was sufficient. Actually both blocks were used, and there was inconclusive evidence of any block effects.

With regard to the removal of Sr^{90}, only the amount of excess soda ash appeared to have a significant effect and that only linearly, but the apparent negative effect (although not significant) of excess lime seemed deserving of further study.

Both the linear and quadratic terms of the regression equation corresponding to the amount of clay per dose had the greatest effect on the per cent removal of Cs^{137}.

Both second-order equations were considered poor fits to the response surfaces. A new set of experiments was designed to test the difference between local Conasauga shale and imported Grundite as media for this waste disposal process. Other factors studied were the amount of clay per dose, size of the particles of clay (as measured by the size of mesh through which the particles would pass), fraction of stoichiometric requirement of lime, and excess soda ash. The design chosen was the four-dimensional simplex which was performed twice, once with Conasauga shale as the clay used and once with Grundite. The objects of these experiments were to compare the effectiveness of the two clays and to find the paths of steepest ascent in each to the maximum per cent removal of strontium and cesium.

Grundite was found to be significantly more effective. The removal of cesium appeared to be unaffected by the factors lime and soda ash, while the removal of strontium appeared to be independent of the size and amount of clay used. Accordingly, two-dimensional paths of steepest ascent were calculated: for cesium in the space of particle size of clay and amount of clay per dose; for strontium in the space of fraction of stoichiometric requirement for lime and excess soda ash. Exploration along these paths is in progress.

ESTIMATION OF CENTERS OF DETECTION OF ION CHAMBER AND PHOTOMULTIPLIER TUBE DETECTION INSTRUMENTS

D. A. Gardiner

This problem, supplied by W. Zobel of the Neutron Physics Division, was attacked by the "sliding sum of squares" technique.

A radioactive source of known strength was placed at a given distance from the geometrical center of the detection instrument. Several readings of the strength of the source at several distances from the source were obtained for each of three instruments.

Point estimates of the center of detection were obtained by use of the model

$$S_{ij} = \frac{C}{(Z_i + a)^2} ,$$

in which

S_{ij} = jth measurement when the instrument is positioned at the ith distance from the source,

Z_i = the ith distance from the source,

C = a constant,

a = the difference between the geometric center and the center of detection of the instrument.

Point estimates of a were determined by an iterative least-squares technique and then added to the geometrical center of the instrument to obtain the center of detection. The sliding sum of squares technique was used to obtain 95% confidence intervals on a.

ANALYSIS OF SOME THERMOCOUPLE DRIFT DATA

D. A. Gardiner

This problem originated with J. F. Potts of the Instrumentation and Controls Division. Experiments with thermocouples had been performed and the data already assembled.

The physical description of the experiments is as follows. A copper block containing transverse holes or wells into which thermocouples are inserted is placed in an electric furnace. At irregular time intervals a "standard" thermocouple is inserted into an empty well and the differences between the readings from the standard and the test thermocouples are recorded. This difference, as it changes over time, is called "drift."

Several furnaces at several temperatures were used, and different types of thermocouples in different lengths of wells also were involved. However, each furnace at its temperature was treated as a single experiment.

The technique chosen for analyzing the data was the analysis of variance. Since the experiments had not been statistically designed, expected mean squares for each experiment had to be derived. The unexpected features of the data were that significant differences in drift were observed among the wells in a block and that these differences changed significantly over time.

Additional consultation with J. W. Reynolds and J. F. Potts dealt with revision of an Oracle program to process thermocouple drift data. The revised program will exhibit, among other things, the standard errors of the several estimates calculated from the information.

ANALYSIS OF ABSORPTION SPECTRA OF FUSED SALTS

D. A. Gardiner S. E. Atta

G. P. Smith, Jr., of the Metallurgy Division is responsible for this problem, which involves the visible and ultraviolet absorption spectra of fused salts as measured with a modified Cary model 11 MS spectrophotometer. The spectra are obtained as chart recordings (or in the near future as digital information from paper tape) of absorbance as a function of wavelength of light. The spectra consist of overlapping bands which arise from the absorption of energy by the electrons that give the light-absorbing particles their chemical properties.

The specific problem is to separate the lowest-energy electronic band from the absorption edge (i.e., the low-energy side) of the next most energetic band. The first model chosen was

$$Y = Y_1 + Y_2 \ , \tag{1}$$

where

$$Y_1 = k_1 e^{-c_1 (x-\mu_1)^2} \ ,$$

$$Y_2 = k_2 e^{-c_2 (x-\mu_2)^2} \ ,$$

and where x is wavelength, k_1 is the height of the low-energy band at its maximum, c_1 is an energy scale factor for the low-energy band, μ_1 is the energy at which the peak of the low-energy band is located, and k_2, c_2, and μ_2 are similarly defined for the second band. Since the second band is only partially observed, k_2, c_2, and μ_2 are very poorly estimated, and this model was discarded in favor of

$$Y = Y_1 + Y_2 \ , \tag{2}$$

where Y_1 is defined as before and

$$Y_2 = e^{c_2 x + k_2} \ ,$$

with c_2 an energy scale factor and e^{k_2} an absorption scale factor for the second energy band.

A special code was written to fit model 2 by either of two methods, known locally as the Deming method and the Garwood method, the choice of method being left to the user of the code. Both are iterative least-squares procedures, the difference being that the Deming method approximates a Taylor series expansion with planar terms only and the Garwood method with both linear and quadratic terms. Preliminary results indicate that both procedures converge to the same solution but that fewer iterations are required for the Deming method within the limits of significant digits.

The code prints out the observation pairs, the logarithm of absorbance, the estimates of Y_1, Y_2, and Y for each x, the estimated area of the low-energy band with its estimated 95% confidence interval, and the estimates of k_1, c_1, μ_1, k_2, and c_2 with their asymptotic 95% confidence intervals.

ASSISTANCE GIVEN THE PROCESS ANALYSIS DEPARTMENT OF THE ENGINEERING DIVISION, Y-12

D. A. Gardiner M. A. Kastenbaum

Several problems of a classified nature originated with W. L. Griffith and A. Christman of the Process Analysis Department of the Y-12 Engineering Division. Assistance was given them on topics such as analysis of variance, estimation of errors, the design of experiments, and probability.

ESTIMATION PROBLEM IN *DROSOPHILA MELANOGASTER*[5]

M. A. Kastenbaum

The problem involved the estimation of relative frequencies of four sperm types in the absence of some information concerning two of the sperm types. The maximum likelihood estimates of the relative proportions have already been reported.[5,6] The asymptotic variances and covariances of these estimates are the following:

$$\sigma^2(\hat{p}_1) = \frac{p_1(1-p_1)^2}{N_1N_2(1-p_1-p_3)}\left[N_2(1-p_3)^2 + N_1p_1p_3\right] ,$$

$$\sigma^2(\hat{p}_2) = \frac{p_2}{(1-p_1-p_3)} \times$$

$$\times \left\{\frac{(1-p_1)(1-p_3)(1-p_1-p_2-p_3)}{N_1(1-p_1)+N_2(1-p_3)} + \frac{p_2\left[N_1p_3(1-p_1)^2 + N_2p_1(1-p_3)^2\right]}{N_1N_2}\right\} ,$$

$$\sigma^2(\hat{p}_3) = \frac{p_3(1-p_3)^2}{N_1N_2(1-p_1-p_3)}\left[N_1(1-p_1)^2 + N_2p_1p_3\right] ,$$

$$\sigma(\hat{p}_1,\hat{p}_2) = -\frac{p_1p_2(1-p_1)}{N_1N_2(1-p_1-p_3)}\left[N_2(1-p_3)^2 - N_1p_3(1-p_1)\right] ,$$

$$\sigma(\hat{p}_1,\hat{p}_3) = -\frac{p_1p_3(1-p_1)(1-p_3)}{N_1N_2(1-p_1-p_3)}\left[N_1(1-p_1) + N_2(1-p_3)\right] ,$$

$$\sigma(\hat{p}_2,\hat{p}_3) = -\frac{p_2p_3(1-p_3)}{N_1N_2(1-p_1-p_3)}\left[N_1(1-p_1)^2 - N_2p_1(1-p_3)\right] .$$

---

[5]The details of this experiment and the preliminary results have been reported; see *Math. Semiann. Prog. Rep. Feb. 28, 1957*, ORNL-2283, p 16.

[6]M. A. Kastenbaum, *Biometrics* **14**, 223–28 (1958).

CONFIDENCE LIMITS ON THE ABSCISSA OF THE POINT OF INTERSECTION OF TWO FITTED LINEAR REGRESSIONS

M. A. Kastenbaum

Let $\hat{Y}_1 = a_1 + b_1 x$ and $\hat{Y}_2 = a_2 + b_2 x$ be two fitted linear regressions based on n_1 and n_2 observations, respectively, where the random variables Y_j ($j = 1,2$) upon which the observations are made are normally and independently distributed with means $E(Y_j) = \alpha_j + \beta_j x$ and common variance σ^2. The abscissa of the point of intersection of these two lines is estimated by

$$\hat{X}_I = \frac{a_2 - a_1}{b_1 - b_2} , \tag{1}$$

where $(a_2 - a_1)$ and $(b_1 - b_2)$ are distributed jointly as a bivariate normal distribution with means $(\alpha_2 - \alpha_1)$ and $(\beta_1 - \beta_2)$, and

$$\text{Var}\ (a_2 - a_1) = \sigma^2 \left(\frac{1}{n_1} + \frac{1}{n_2} + \frac{\bar{x}_1^2}{S_1} + \frac{\bar{x}_2^2}{S_2} \right) , \tag{2}$$

$$\text{Var}\ (b_1 - b_2) = \sigma^2 \left(\frac{1}{S_1} + \frac{1}{S_2} \right) , \tag{3}$$

$$\text{Cov}\ [(a_2 - a_1), (b_1 - b_2)] = \sigma^2 \left(\frac{\bar{x}_1}{S_1} + \frac{\bar{x}_2}{S_2} \right) , \tag{4}$$

where

$$S_j = \sum_{i=1}^{n_j} (x_{ij} - \bar{x}_j)^2 .$$

If we let

$$X_I = \frac{\alpha_2 - \alpha_1}{\beta_1 - \beta_2}$$

and if we estimate the quantities in Eqs. 2, 3, and 4 by replacing σ^2 with s^2, the pooled estimate of the error mean square, then according to a theorem by Fieller,[7] the confidence limits for X_I consist of those values for which

$$\left| (a_2 - a_1)^2 - t^2 s^2_{(a_2 - a_1)} \right| - 2X_I \left| (a_2 - a_1)(b_1 - b_2) - t^2 s_{(a_2-a_1)(b_1-b_2)} \right| + \\ + X_I^2 \left| (b_1 - b_2)^2 - t^2 s^2_{(b_1 - b_2)} \right| = 0 ,$$

[7]E. C. Fieller, *J. Roy. Statist. Soc.* **B7**, 1–53 (1940).

where t is the appropriate level of the Student distribution for $n_1 + n_2 - 4$ degrees of freedom.

TETRAD FREQUENCIES IN DROSOPHILA

M. A. Kastenbaum

Assume that the physical basis of crossing over (chromosome replication, for instance) starts at one end of the chromosome and proceeds to the other end. Assume further that there is a constant probability of an exchange, α, in every genetic unit of length. After an exchange has occurred, chromosome replication continues in the same direction, but the per-unit probability of an exchange is now β. After a second exchange, the probability reverts to α, after a third exchange, to β again, and so on. This simple system is disrupted somewhat by an interference factor which tends to prevent the occurrence of exchanges in adjacent or even near-adjacent units. The delay, which is a function of distance, is approximated, in this model, by a step function whose value is zero for i units after every exchange and is unity thereafter.

Let the probability of 0, 1, 2, and more than 2 exchanges be P_0, P_1, P_2, and P_3, respectively. Then for a chromosome N units long $P_0 = (1-\alpha)^N$. To obtain P_1, we consider separately all possible cases; that is, that the one required exchange may occur in any one of the N genetic units. Thus, for the first $(N-i)$ units we have the following:

Unit in Which Exchange Occurs	Probability
1	$(1-\alpha)^0\,\alpha(1-\beta)^{N-i-1}$
2	$(1-\alpha)^1\,\alpha(1-\beta)^{N-i-2}$
...	...
k	$(1-\alpha)^{k-1}\,\alpha(1-\beta)^{N-i-k}$
...	...
$N-i$	$(1-\alpha)^{N-i-1}\,\alpha(1-\beta)^0$

$$\text{Total: } \alpha \sum_{k=1}^{N-i} (1-\alpha)^{k-1}\,(1-\beta)^{N-i-k}$$

For the remaining i units, in which the probability of an exchange is zero by the definition of i, we have:

Unit	Probability
$N - i + 1$	$(1-\alpha)^{N-i}\alpha$
$N - i + 2$	$(1-\alpha)^{N-i+1}\alpha$
. . .	. . .
r	$(1-\alpha)^{N-i+r}\alpha$
. . .	. . .
N	$(1-\alpha)^{N-1}\alpha$
	Total: $\alpha \sum_{r=0}^{i-1} (1-\alpha)^{N-i+r}$

In total, then,

$$P_1 = \alpha \left[\sum_{k=1}^{N-i} (1-\alpha)^{k-1} (1-\beta)^{N-i-k} + \sum_{r=0}^{i-1} (1-\alpha)^{N-i+r} \right] .$$

By the same arguments,

$$P_2 = \alpha \sum_{s=1}^{N-2i-1} (1-\alpha)^{s-1} \left[\beta \sum_{k=1}^{N-2i-s} (1-\beta)^{k-1} (1-\alpha)^{N-2i-k-s} + \right.$$

$$\left. + \beta \sum_{r=0}^{i-1} (1-\beta)^{N-2i-s+r} \right] + \alpha\beta \sum_{t=0}^{i-1} (1-\alpha)^{N-2i-1+t} \sum_{x=0}^{i-t-1} (1-\beta)^x ,$$

and finally

$$P_3 = 1 - P_0 - P_1 - P_2 .$$

Evaluating the indicated summations, we find that

$$P_0 = (1-\alpha)^N ,$$

$$P_1 = \frac{\alpha(1-\beta)^{N-i} - \beta(1-\alpha)^{N-i}}{\alpha - \beta} - P_0 ,$$

$$P_2 = (1-\alpha)^{N-2i-1} \left[1 + \frac{(N-2i-1)\,\alpha\beta}{\beta-\alpha} \right] +$$

$$+ \frac{\alpha^2(1-\beta)}{(\beta-\alpha)^2} \left[(1-\beta)^{N-2i-1} - (1-\alpha)^{N-2i-1} \right] - (P_0 + P_1) \quad ,$$

$$P_3 = 1 - P_0 - P_1 - P_2 \quad .$$

A USEFUL PARTITION OF CHI SQUARE

M. A. Kastenbaum

Let n_{ij} be the observed frequency in the ith row and jth column of an $r \times s$ contingency table, and designate

$$R_i = \sum_{j=1}^{s} n_{ij} \quad , \qquad C_j = \sum_{i=1}^{r} n_{ij} \quad ,$$

and

$$N = \sum_{i=1}^{r} R_i = \sum_{j=1}^{s} C_j \quad .$$

Then on the hypothesis of independence of rows and columns, the expected frequency in the ijth cell is

$$E(n_{ij}) = \frac{R_i C_j}{N} \quad . \tag{1}$$

A test of this hypothesis is given by the statistic

$$X^2 = N \left(\sum_{i=1}^{r} \sum_{j=1}^{s} \frac{n_{ij}^2}{R_i C_j} - 1 \right) \quad , \tag{2}$$

which has a limiting chi-square distribution with $(r-1)(s-1)$ degrees of freedom. In many experimental situations the contingency table results from the selection of s r-celled multinomial samples of sizes $C_1, C_2, \ldots, C_s$. The null hypothesis

$$H_0\colon\ p_{ij} = p_i \qquad \text{for } i = 1, 2, \ldots, r; \text{ and } j = 1, 2, \ldots, s \tag{3}$$

is that the s samples have been drawn from the same multinomial population with parameters p_i, where

$$\sum_{i=1}^{r} p_i = 1 \quad .$$

A test of this hypothesis is provided by Eq. 2. If, however, there is reason to believe that the s samples were, in fact, drawn from two different multinomial populations in such a way, say, that the first $m < s$ samples were drawn from one population and the remaining $(s - m)$ samples from another population, then three separate hypotheses may be tested:

$$H_{01}: \; p_{i1} = p_{i2} = \ldots = p_{im} = p_i \quad ; \tag{4}$$

$$H_{02}: \; p_{i(m+1)} = p_{i(m+2)} = \ldots = p_{is} = p_i' \quad ; \tag{5}$$

$$H_{03}: \; p_i = p_i' \qquad \text{for all } i \quad . \tag{6}$$

A test of H_{01} with $(r - 1)(m - 1)$ degrees of freedom is given by

$$X^2_{(r-1)(m-1)} = N \sum_{i=1}^{r} \frac{1}{R_i} \left[\sum_{j=1}^{m} \frac{n_{ij}^2}{C_j} - \frac{\left(\sum_{j=1}^{m} n_{ij} \right)^2}{\sum_{j=1}^{m} C_j} \right] \quad ; \tag{4'}$$

a test of H_{02} with $(r - 1)(s - m - 1)$ degrees of freedom is given by

$$X^2_{(r-1)(s-m-1)} = N \sum_{i=1}^{r} \frac{1}{R_i} \left[\sum_{j=m+1}^{s} \frac{n_{ij}^2}{C_j} - \frac{\left(\sum_{j=m+1}^{s} n_{ij} \right)^2}{\sum_{j=m+1}^{s} C_j} \right] \quad ; \tag{5'}$$

and a test of H_{03} with $(r - 1)$ degrees of freedom is given by

$$X^2_{(r-1)} = N \sum_{i=1}^{r} \frac{1}{R_i} \left[\frac{\left(\sum_{j=1}^{m} n_{ij} \right)^2}{\sum_{j=1}^{m} C_j} + \frac{\left(\sum_{j=m+1}^{s} n_{ij} \right)^2}{\sum_{j=m+1}^{s} C_j} - \frac{R_i^2}{N} \right] \quad . \tag{6'}$$

Note that the sum of Eqs. 4′, 5′, and 6′ is identically equal to Eq. 2. Also note that

$$H_{01} \cap H_{02} \cap H_{03} \Longrightarrow H_0 \quad .$$

Illustration

In a stock of *Drosophila melanogaster* in which sperm of four different genotypes were produced, a geneticist wished to study the influence of irradiation on the relative frequencies with which the sperm types were represented in the subsequent generation. The variability from male to male, however, was so great as to completely obscure any influence of the irradiation on the composition of the sperm population.

After a genetic procedure involving approximately ten generations and designed to make the stock genetically homogeneous, it was found that five of the seven lines were still heterogeneous with respect to sperm types. The over-all heterogeneity at this point appeared to be due to the combination of data on males from two separate populations; that is to say, of the 23 samples of observed sperm, the distributions of the sperm types in 17 samples appeared to be homogeneous among themselves, but different from the remaining 6 samples. The number of offspring and their genetic classification are given in Table 1. To test the hypotheses related to these observations, the data were partitioned in such a way as to carry out the tests outlined above.

Table 1. Experimental Results

Sample No.	Number of Offspring			
	Genotypes			Total
	I	II	III + IV	
1	12	26	11	49
2	22	38	6	66
3	25	44	11	80
4	2	11	4	17
5	13	22	6	41
6	16	36	12	64
7	7	32	9	48
8	27	37	23	87
9	23	44	7	74
10	4	14	1	19
11	7	20	12	39
12	20	50	16	86
13	32	48	10	90
14	19	37	11	67
15	8	16	10	34
16	27	56	18	101
17	12	18	7	37
18	25	26	7	58
19	9	11	11	31
20	49	44	11	104
21	38	45	6	89
22	18	21	9	48
23	39	38	9	86
Total	454	734	227	1415

1. To test

$$H_{01}:\ p_{ij} = p_i \qquad i = 1, 2, 3;\ j = 1, 2, \ldots, 17\ ,$$

we use Eq. 4′, setting $m = 17$:

$$X^2_{32} = 1415 \left(\frac{3.245976}{454} + \frac{3.586132}{734} + \frac{3.832569}{227} \right) = 40.92038\ .$$

2. To test

$$H_{02}:\ p_{ij} = p_i' \qquad i = 1, 2, 3;\ j = 18, 19, \ldots, 23\ ,$$

we use Eq. 5′, setting $s = 23$:

$$X_{10}^2 = 1415 \left(\frac{0.972608}{454} + \frac{0.633155}{734} + \frac{2.192966}{227} \right) = 17.92239 \quad .$$

3. To test

$$H_{03}:\ p_i = p_i' \qquad i = 1, 2, 3\quad ,$$

we use Eq. 6′:

$$X_2^2 = 1415 \left(\frac{6.750696}{454} + \frac{3.228047}{734} + \frac{0.642455}{227} \right) = 31.26584 \quad .$$

4. To test

$$H_0:\ p_{ij} = p_i \qquad i = 1, 2, 3;\ j = 1, 2, \ldots, 23\ ,$$

we use Eq. 2 and find that

$$X_{44}^2 = 90.10861 \quad .$$

Test 4 shows that the 23 samples are heterogeneous. Tests 1, 2, and 3 show, respectively, that (1) the first 17 samples are homogeneous, (2) the last 6 samples are homogeneous, and (3) the first 17 samples and the last 6 samples are drawn from two different populations.

The results of these tests assured the geneticist that his genetic procedure had been partially successful and that a single genetic factor remained to be removed to achieve homogeneity.

ORACLE OPERATIONS AND SPECIAL CODES

Machine operations were on a three-shift, five-day-week schedule. The average good computing time was 405 hr per month. The operating ratio (good computing time ÷ scheduled computing time) was 94.8%.

The operations staff is composed of three machine operators, one input preparation operator, one subroutine librarian, four mechanics, two engineers, and one supervisor.

A new console with a faster paper-tape punch, narrow output magnetic-tape unit, monitoring typewriter, and provision for alphameric input and output was installed in May 1957.

The following routines and subroutines have been prepared during the period covered by this report.

GENERAL CHARACTERISTIC VALUE PROBLEM FOR SYMMETRIC MATRICES

A. C. Downing

The problem of finding the characteristic values and vectors for the product of two symmetric matrices has been one of continued interest in laboratory research on the structure of simple molecules. An integrated package of Oracle routines has been assembled for this calculation.

The numerical method is based upon Givens' procedure for finding the characteristic values and vectors of a single symmetric matrix. The major segments of the routine and their programmers are given in Table 2. The output segment includes the matrix output routine prepared by E. L. Cooper. The program for calculating the vectors of the tridiagonal form was modified by D. J. Wehe to provide a complete set of vectors for a multiple root.

Table 2. Segments of the Routine

Major Segment	Approximate Number of Instructions	Programmer
Matrix loading routine	256	Downing
Rotation to tridiagonal form*	418	Downing
Sturm sequence calculation of characteristic values*	379	Hildebrandt
Vectors of tridiagonal form*	400	Dismuke
Unrotation of vectors*	155	Dismuke
Matrix multiplication	202	Downing
Supervisory control	315	Downing
Output	1571	Downing

**Math. Semiann. Prog. Rep. Feb. 28, 1957,* ORNL-2283, p 6.

Theoretically, the matrix of vectors obtained in this calculation, call it V, should diagonalize both matrices. More precisely, if $ABV = V\Lambda$, where A is positive definite, then the vectors are normalized so that, if they were exact, it would be true that $V^TBV = \Lambda$ and $V^{-1}A(V^{-1})^T = I$. Both these matrix products are computed and may be obtained as an optional output. They provide a measure of the accuracy of the numerically computed vectors. Experience has shown that accurate vectors are not always obtained. This is mentioned elsewhere in this report (see "Characteristic Vectors of Symmetric Matrices").

APPROXIMATE MATRIX INVERSE AND SOLUTION OF LINEAR EQUATIONS

S. E. Atta

The program which will calculate the inverse of a square matrix of small order as well as obtain the solution of systems of linear equations has been rewritten.[1] The system of codes which are stored on a magnetic tape can be used as a complete program, or it can be used as a segment in a larger program. The input and output codes were written to take advantage of the new input-output equipment.

Past experience has shown that an essential part of a program for matrix problems is an evaluation of the errors in the approximate solutions due to propagation[2] and rounding. Several methods of appraising the accuracy of the approximate solutions are included in this program.

A simple scheme has been devised for this matrix program which will indicate approximately the number of accurate digits not lost through subtraction due to "ill condition." The scheme is based upon the number of accurate digits in the matrix A. An extra vector b' is formed by summing each row of A. This is equivalent to setting the elements of the vector x equal to 1. The elements of b' have the same linear relation as the equations. The number of accurate digits in b' is forced to be equal to the number of digits in A. The system of equations $Ax = b'$ is solved simultaneously with the other systems. A comparison of the solutions of this system with 1.0 should indicate the accuracy to be expected in the approximate solutions of the other systems.

Another method used to evaluate the error was suggested by Householder.[3] It is a rigorous upper bound to the error in the approximate inverse. Let C represent a computed inverse of a square matrix A, and set

$$H = I - AC \ .$$

Then

$$||A^{-1} - C|| \leqq ||C|| \cdot \frac{||H||}{I - ||H||} \ ,$$

in terms of any norm. For this program the norm is taken to be

$$||H|| = \max j \left(\Sigma_i |H_{ij}| \right) < 1 \ .$$

[1] *Math. Semiann. Prog. Rep. June 30, 1956*, ORNL-2134, p 10.

[2] S. E. Atta, *J. Assoc. Computing Mach.* 4(1), 36 (1957).

[3] A. S. Householder, *J. Assoc. Computing Mach.* 5(3), 205 (1958).

COMPUTING THE MAXIMAL EIGENVALUE OF A POSITIVE MATRIX

C. T. Fike

A method devised by Brauer[4] for computing the maximal eigenvalue of a positive matrix has been coded for the Oracle, and its speed and accuracy have been checked on certain test matrices. It is iterative in character and depends on the fact that the maximal eigenvalue of a positive matrix lies in the interval determined by the maximal and minimal row sums of the matrix. Simple similarity transformations which reduce successively the length of this interval are applied to the matrix.

Although it appears to be exceedingly difficult to estimate sharply the error which can occur in the application of this code to a particular matrix, tests indicate that the maximal eigenvalue can generally be computed accurately. Even in tests requiring more than 200 iterations, at least 8, and usually more, significant figures of the maximal eigenvalue were always obtained.

Surprisingly, perhaps, the number of iterations required to reduce the difference between the maximal and minimal row sums of the test matrices to less than 2^{-31} was observed to be about 200 regardless of the order of the matrix. For matrices of order less than ten, 200 iterations require less than 1 min of computing time; $2\frac{1}{2}$ min was needed to obtain the maximal eigenvalues of the 20×20 segment of the Hilbert matrix accurate to eight significant figures.

All computations are performed in fixed-point arithmetic, and only the fast memory of the machine is used.

BIORTHOGONALIZATION METHOD OF LANCZOS

C. T. Fike

Lanczos' method of biorthogonalization and the quotient-difference (Q.D.) algorithm have been discussed widely. A good bibliography and a discussion of the relation of the two methods are contained in a recent article by Henrici.[5]

The method of Lanczos was selected to be coded in floating-point arithmetic for the Oracle to provide a program which could be coupled with the Q.D. algorithm to provide a means of computing eigenvalues of nonsymmetric matrices. Experience with the method at other computing centers indicated that the method would not be suitable for large matrices, since the choice of a starting vector seriously affects the accuracy of the computations. This was confirmed in subsequent tests.

---

[4] A. T. Brauer, *Duke Math. J.* 24, 367–78 (1957).

[5] P. Henrici, in *Further Contributions to the Solution of Simultaneous Linear Equations and the Determination of Eigenvalues*, National Bureau of Standards Applied Mathematics Series No. 49, p 23–46 (Jan. 15, 1958).

The code was completed and tested extensively, both alone and with the Q.D. algorithm. It was found that the method was poorly suited to anything but small matrices, a disappointing loss of accuracy being experienced even on problems of moderate size. The work of Wilkinson[6] indicates that reorthogonalizing after each step is, in most cases, a satisfactory remedy to this difficulty, which apparently arises in a large degree from loss of the orthogonality conditions among the iterate vectors. A revision of the code to include reorthogonalization is planned.

QUOTIENT-DIFFERENCE ALGORITHM

A. A. Grau

The quotient-difference algorithm, previously programed in fixed point,[7] was programed in the (40, 40) floating-point mode. It was designed to use the output from the code described above, which reduces a nonsymmetric matrix to tridiagonal form by using the biorthogonalization method of Lanczos.

Some experimental work was done with the combination, much of which indicated that some modification in the orthogonalization process was desirable. It was also tried on a problem involving nonsymmetric matrices arising in the Laboratory; the matrices involved, however, for the most part were very poorly conditioned.

INTERACTION OF ELECTROMAGNETIC WAVES WITH PLASMA

E. L. Cooper C. T. Fike A. A. Grau J. Neufeld[8]

Mathematically, this problem consisted in the determination, for a large number of individual cases, of the roots of a polynomial equation given in the form of the determinant of a ninth-order nonsymmetric matrix equal to zero. The application of basic principles from the theory of determinants and matrices reduced the problem analytically to the determination of the eigenvalues of a fourth-order nonsymmetric matrix.

A code was written to handle the problem in general which incorporated the quotient-difference code just described. Most of the matrices which resulted were, however, very poorly conditioned, and in these cases the process was not expected to give reliable results. In some cases, the matrices decomposed, and it was possible to handle them by a more direct method. Comparison of the results with those obtained from the general code gave better agreement than was anticipated, even for matrices with condition number of the order 10^{10}, though generally one of the four roots obtained by the general method was in gross error.

---

[6] J. H. Wilkinson, *Computer J.* 1, 148–52 (1958).

[7] *Math. Semiann. Prog. Rep. Feb. 28, 1957*, ORNL-2283, p 3.

[8] Health Physics Division.

Additional cases of this problem will probably be considered in the future. It is hoped that modifications of the method outlined will permit the reliable treatment of some of the cases of interest. Others, if necessary, can be treated more directly.

GENERALIZED LINEAR REGRESSION CODE I[9]

C. L. Gerberich A. H. Culkowski

This code will calculate, by the method of least squares, the coefficients a_j of the equation

$$Y = \sum_{j=0}^{n} a_j X_j \ ,$$

where

$$X_j = X_j(x_0, x_1, \ldots, x_{j-1}, y) \ ,$$

$$Y = Y(x_0, x_1, \ldots, x_{p-1}, y) \ ,$$

$$2 \leqq n \leqq 29 \text{ (hex) and } p \geqq 1 \ .$$

The $a_j X_j$ are the terms of the equation, and each term is composed of the product of an unknown coefficient a_j and a known function X_j of the variables $x_0, x_1, \ldots, x_{p-1}, y$. The term Y has the coefficient 1 and is considered the resultant or dependent-variable term.

The code is linear in that Y is a linear function of the coefficients. Nevertheless, the terms may be composed of exponentials, products, powers, basic trigonometric functions, divisions, logarithms, differences, sums, or combinations of these.

In addition to finding the coefficients, there is the option of obtaining $Y^T Y$, $(X^T X)^{-1}$, and the summation of the residuals squared, $(Y^T Y - AX^T Y)^2$, the latter quantity being useful in computing the standard estimate of fit.

GENERALIZED LINEAR REGRESSION CODE II

A. H. Culkowski N. M. Dismuke

This code, which is an elaboration of Code I designed and begun by C. L. Gerberich, will calculate, by the method of weighted least squares, r sets of coefficients $b_{j,r}$ of the equations

$$Y_r = \sum_{j=0}^{n} b_{j,r} X_j \ ,$$

where

$$X_j = X_j(x_0, x_1, \ldots, x_{p-1}, y_0, y_1, \ldots, y_{q-1}) \ ,$$

$$Y_r = Y_r(x_0, x_1, \ldots, x_{p-1}, y_0, y_1, \ldots, y_{q-1}) \ ,$$

$$2 \leqq n + r \leqq 29 \text{ (hex) and } p \geqq 1,\ q \geqq 1,\ r \geqq 1 \ .$$

[9] *Math. Semiann. Prog. Rep. June 30, 1955*, ORNL-1928, p 6; a report on the completed program is available from the Oracle subroutine library.

The $b_{j,r}X_j$ are the terms of the rth equation, and each term is composed of the product of an unknown coefficient $b_{j,r}$ and known functions X_j, made up of any or all of the variables $x_0, x_1, \ldots, x_{p-1}, y_0, y_1, \ldots, y_{q-1}$. Each Y_r is considered a resultant or dependent term, and each Y_r may be formed from the variables $x_0, x_1, \ldots, x_{p-1}, y_0, y_1, \ldots, y_{q-1}$.

The code is linear in that Y is a linear function of the coefficients. Nevertheless, the terms may be composed of exponentials, sums, products, differences, powers, basic trigonometric and hyperbolic functions, divisions, logarithms, or any combinations of these.

The input instructions to the code describe the functions to be formed and are written in modified Polish notation. This input and other pertinent information should be in alphameric form; however, the numerical input, which constitutes the variables, is to be given in packed decimal form.

The user may choose the output wanted from the following list of items (the first ten of which are now available) and may also designate whether it is to be in binary or packed decimal form:

1. **Means** (mu)

$$\bar{X}_j = \frac{\sum_{i=1}^{m} W_i X_{ij}}{\sum_{i=1}^{m} W_i}, \qquad j = 0, 1, 2, \ldots, n$$

2. **Standard deviations** (*sd*)

$$S_j = \frac{\sqrt{\sum_{i=1}^{m} W_i (X_{ij} - \bar{X}_j)}}{\sqrt{m-1}}, \qquad m = \text{the number of observations}$$

$$S_{yr} = \frac{\sqrt{\sum_{i=1}^{m} W_i (Y_{ir} - \bar{Y}_r)^2}}{\sqrt{m-1}}, \qquad 1 \leqq r \leqq 29 - n$$

3. **Standard deviation ratios** (*sr*)

$$sr = \frac{S_j}{S_{yr}}$$

4. **"Transpose" matrix or augmented matrix of the normal equations** (*t*)

$$t = (X,Y)^T WX,\ (X,Y)^T WY$$

5. **Matrix of sums of squares and sums of products of deviations from the means** (*spm*)

$$spm = X^T WX - (\Sigma W)\ \bar{X}\bar{X}^T$$

6. **Correlation matrix** (cm)

$$cm = (S) \left[X^T WX - (\Sigma W) \bar{X}\bar{X}^T \right] (S) \quad , \qquad S = \begin{pmatrix} \frac{1}{\sqrt{S_0}} & & & 0 \\ & \frac{1}{\sqrt{S_1}} & & \\ & & \ddots & \\ 0 & & & \frac{1}{\sqrt{S_j}} \end{pmatrix}$$

7. **Inverse of the normal matrix** (ti)

$$ti = (X^T WX)^{-1}$$

8. **Raw regression coefficients** (b)

$$b_r = (X^T WX)^{-1} X^T WY_r$$

9. **Standard regression coefficients** (b')

$$b'_r = b_{jr} \left(\frac{S_{yr}}{S_j} \right)$$

10. **The vector y computed** (yc)

$$yc_r = Xb_r$$

11. **Residuals** (res)

$$res = Y_r - Xb_r$$

12. **Weighted sum of residuals** (swr)

$$swr = W^T (Y_r - Xb_r)$$

13. **Weighted sum of residuals squared** (ssr)

$$ssr = (Y_r - Xb_r)\ W^T (Y_r - Xb_r)$$

14. **Multiple correlation coefficient** (mcc)

$$mcc = 1 - \frac{(Y_r - Xb_r)\, W^T (Y_r - Xb_r)}{(m-1)\ S_{yr}^2}$$

15. **Standard error squared or variance of deviations about regression** (msr)

$$msr = \frac{(Y_r - Xb_r)\ W^T (Y_r - Xb_r)}{m - n - 1}$$

16. Variances of parameter estimates (vb)

$$vb_{rj} = \frac{\sum_{i=1}^{m} W_i (Y_{ri} - Yc_{ri})^2}{m - n - 1} (X^T WX)^{-1}_{j,j}$$

17. Variance of Y_r for a single prediction (vy)

$$vy_{ir} = \left| 1 + (X_i - \bar{X} (X^T WX)^{-1} (X_i - \bar{X})^T \right| \frac{\sum_{i=1}^{m} W_i (Y_{ri} - Yc_r)^2}{m - n - 1}$$

18. Variance of Y_r for a mean prediction (vym)

$$vym_{ir} = \left| (X_i - \bar{X}) (X^T WX)^{-1} (X_i - \bar{X})^T \right| \frac{\sum_{i=1}^{m} W_i (Y_{ri} - Yc_{ri})^2}{m - n - 1}$$

DETERMINATION OF ATOMIC ENERGY LEVELS

M. Feliciano

Given a set of energy levels E_i and a set of observed transitions D_α between these energy levels, it is required to adjust the values of the energy levels to give the best fit to the observed transition energies. As many as 512 levels and about 10,000 transitions are anticipated.

Each transition energy is coupled with a weight depending upon experimental information. However, the former must also be associated with the pair of energy levels from which it was derived. It is supposed that the given set of energy levels is precise enough to afford the following correlation. All differences,

$$E_j - E_i = D_{ij} \quad , \qquad\qquad i < j \quad ,$$

being established, the observed transition levels are paired to the index ij for which

$$|D_{ij} - D_\alpha| < \epsilon \quad ,$$

where ϵ is a fixed positive quantity determined by the experimenter.

With the conventions that W is the diagonal matrix of the weights arranged according to index; y is the solution vector for energy levels ranging from E_0 to E_n; D is the vector whose elements are the transition energies arranged according to index; and M is the matrix

$$\begin{pmatrix} -1 & 1 & 0 & 0 & \dots & 0 & 0 \\ -1 & 0 & 1 & 0 & \dots & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot\ \ \cdot & \cdot & \cdot \\ 0 & -1 & 1 & 0 & \dots & 0 & 0 \\ 0 & -1 & 0 & 1 & \dots & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot\ \ \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & \dots & -1 & 1 \end{pmatrix} ;$$

then

$$WMy = WD \quad .$$

Hence

$$M^T WMy = M^T WD \quad .$$

Now the matrix $M^T WM$ is a symmetric matrix obeying the following:

$$a_{ii} = -\sum_{\substack{j=1 \\ i \neq j}}^{n} a_{ij} \quad ,$$

and a_{ij} $(i > j)$ is the weight affixed to the transition D_{ij}. This matrix is scanned for its largest element, which must be on its diagonal, and is compressed around it; that is, the row and column to which this element belongs are eliminated. Similar action is taken with the corresponding elements of y and D. This procedure has the effect of establishing $E_k = 0$, where k is the index of the row eliminated. Now let the matrix obtained by compressing $M^T WM$ be denoted by A and the corresponding vectors derived from y and D by x and b. Then $Ax = b$.

At this juncture it must be pointed out that the method for obtaining the desired result of the problem can be divided appropriately into three parts. The first part generates the matrix A and scales it by that power of 2 not less than twice its largest element. The second part, an intermediate step, establishes b scaled so that its norm is not greater than $\frac{1}{2}$. It generates a first approximation x_1 of x from the given set of energy levels, scales it analogously to b, and finds bounds for the proper values of the matrix A. The final part of the method uses the iterative method of conjugate gradients, which converges to x after n steps. The first two parts have been completed, and the last one is near the final stage.

ESTIMATION OF GENOTYPIC PROBABILITIES IN THE ABO BLOOD GROUP SYSTEM

G. J. Atta E. Novitsky[10]

Human blood can be classified into four phenotypic groups, A, B, Ab, and O, and into six genotypic groups, AA, AO, BB, BO, Ab, and OO. The inheritance of any one of the groups is controlled by the three allelomorphic genes A, B, and O, of which O is recessive to A and B. From data collected in Australia on the phenotypic blood groups of mother-child pairs it is desired to estimate the expected probabilities of the genotypes by the method of maximum likelihood.

In general, let there be m mother-child pairs. Let π_j $(j = 1, 2, \ldots, m)$ be the probability associated with the jth pair. Then

$$\sum_{j=1}^{m} \pi_j = 1 \ .$$

Define

$$\begin{aligned} p_1 &= \text{probability of genotype AA} \ , \\ p_2 &= \text{probability of genotype AO} \ , \\ p_3 &= \text{probability of genotype BB} \ , \\ p_4 &= \text{probability of genotype BO} \ , \\ p_5 &= \text{probability of genotype AB} \ , \\ p_6 &= \text{probability of genotype OO} \ , \end{aligned}$$

where

$$\sum_{i=1}^{6} p_i = 1 \ .$$

Let

$$\pi_j = \phi_j(p_1, p_2, \ldots, p_6) \ .$$

If f_j is the observed frequency in the jth pair, then the likelihood function is

$$\prod_{j=1}^{m} \pi_j^{f_j} \ ,$$

and its logarithm is given by

$$L = \sum_{j=1}^{m} f_j \log \pi_j \ .$$

[10]Biology Division.

The maximum likelihood estimates of the p_i are obtained from the solution of the system of equations

$$\frac{\partial L}{\partial p_i} = 0 \qquad (i = 1, 2, 3, \ldots, 5) \ .$$

Since these five equations are nonlinear, the solution cannot be obtained directly. Consequently Rao's method of scoring[11] was used. Thus the scores at p_i are

$$S_i = \sum_{j=1}^{m} \frac{f_j}{\pi_j} \cdot \frac{\partial \pi_j}{\partial p_i} \qquad (i = 1, 2, \ldots, 5) \ ,$$

and the (i,k) element of the information matrix is

$$I_{ik} = \sum_{j=1}^{m} \frac{1}{\pi_j} \cdot \frac{\partial \pi_j}{\partial_p i} \cdot \frac{\partial \pi_j}{\partial p_k} \qquad (i,\ k = 1, 2, \ldots, 5) \ .$$

Initial values of the p_i are chosen. The S_i and I_{ik} are evaluated at these values. Small additive connections, δp_i, are given by the solution of the simultaneous equations

$$Id = s \ ,$$

where

I is the information matrix,

d is the vector $\begin{pmatrix} \delta p_1 \\ \delta p_2 \\ \delta p_3 \\ \delta p_4 \\ \delta p_5 \end{pmatrix}$,

s is the vector $\begin{pmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \end{pmatrix}$.

The operation is repeated with corrected values each time until stable values of the p_i are

[11] C. R. Rao, *Advanced Statistical Methods in Biometric Research*, p 165–72, Wiley and Sons, New York, 1952.

obtained. Then p_6 is computed from the equation

$$p_6 = 1 - \sum_{i=1}^{5} p_i \; .$$

The asymptotic variance-covariance matrix is given by the inverse of the information matrix.

An Oracle program has been written to estimate the p_i, together with their variances, and to evaluate, by means of the chi-square test, the extent to which the data fit the model. Additional parameters were introduced in the model so that the hypotheses of (1) abnormal segregation in heterozygotes, either male or female, (2) viability effects, and (3) misclassification of some blood groups could be tested.

Preliminary results and discussion have been reported elsewhere.[12] Final results are still being evaluated and will be published in the open literature.

DISEASE-INCIDENCE ESTIMATION IN POPULATIONS SUBJECT TO MULTIPLE CAUSES OF DEATH

G. J. Atta D. A. Gardiner A. W. Kimball
E. Leach A. C. Upton[13]

A random sample of N animals from an infinite population is observed until all animals in the sample have died. At the time of death, the cause of death is recorded. The problem considered is the estimation of disease incidences in a population in which one or more diseases have been eliminated as causes of death. The estimation is performed with a sample from a population in which all causes of death are operating. An estimate of the incidence of the ith disease, $\hat{I}_i'$, and an approximate variance formula for $\hat{I}_i'$ have been derived.[14]

A random sampling experiment has been conducted in an attempt to learn something about the small-sample properties of the estimates. An Oracle program was written from which 1000 random samples were generated for a predetermined set of conditional probabilities. Each sample consisted of 400 animals, 10 time intervals, and 4 causes of death ($N = 400$, $m = 4$, $n = 10$). When disease 1 was eliminated, $\hat{I}_i'$ and $V(\hat{I}_i')$ were computed for each sample. The distribution of the estimates is shown in Table 3. Column 4 gives a measure of the bias of the estimates, column 5 the standard error of the bias, and column 6 Student's ratio. Columns 7 and 8 show a measure of the skewness and kurtosis of the sampling distributions, respectively. Examination of Table 3 indicates that the estimates

---

[12] E. Novitski and R. D. Owen, *Biol. Semiann. Prog. Rep. Aug. 15, 1958*, ORNL-2593, p 34.

[13] Biology Division.

[14] *Math. Semiann. Prog. Rep. June 30, 1956*, ORNL-2134, p 13; *Math. Semiann. Prog. Rep. Feb. 28, 1957*, ORNL-2283, p 11.

are unbiased and that the sampling distributions of $\hat{I}_i'$ approximate the normal distribution. Table 4 shows the distribution of the variances of the estimates. Note that in every case the approximate variance formula overestimates the sample variance.

The empirical study of these estimates will be continued. A final summary of the results will be submitted for separate publication.

Table 3. Distribution of the Estimates of I_i'

i	I_i'	$E\left(\hat{I}_i'\right)$	$E\left(\hat{I}_i'\right) - I_i'$	Standard Error	t	$\sqrt{\beta_1} = \mu_3/\mu_2^{3/2}$	$\beta_2 = \mu_4/\mu_2^2$
2	0.43357	0.43204	−0.00153	0.001118	1.37	0.0915	3.086
3	0.49978	0.50124	+0.00146	0.001228	1.19	0.0784	3.057
4	0.06665	0.06672	+0.00007	0.000683	1.10	0.2726	2.925

Table 4. Distribution of $V(I_i')$

i	Variance from Formula	Variance from 1000 Samples
2	0.0019073	0.0012499
3	0.0036176	0.0015081
4	0.0004999	0.0004660

ROOTS OF A DETERMINANTAL EQUATION

G. J. Atta A. Sauer[15]

The equation

$$\frac{k}{(B^2 L_f^2 + 1)\,[B^2 L_s^2 + (1 + \sigma)]} = 1$$

has two roots in terms of the unknown σ: $B_1^2(\sigma) > 0$ and $B_2^2(\sigma) < 0$. Similarly, the equation

$$\frac{k}{(B^2 L_f^2 + 1)\,(B^2 L_s^2 + 1)} = 1$$

has two roots: $B_3^2 > 0$ and $B_4^2 < 0$. If

$$\beta_1(o) = B_1^2(\sigma) \;, \qquad \beta_3 = B_3^2 \;,$$

$$\beta_2(o) = -B_2^2(\sigma) \;, \qquad \beta_4 = -B_4^2 \;,$$

[15] Director's Division.

then x and y are related by the equation

$$L_f \operatorname{ctnh} \frac{x}{L_f} = \frac{1-\alpha_3}{\alpha_3-\alpha_4} \cdot \frac{\tan \beta_3 y}{\beta_3} + \frac{1-\alpha_4}{\alpha_4-\alpha_3} \cdot \frac{\tan \beta_4 y}{\beta_4} ,$$

where

$$\alpha_i = \frac{1}{L_f^2 B_i^2 + 1} .$$

A program was written to solve for the real positive root $\sigma = \sigma(x)$ of the determinantal equation

$$\begin{vmatrix} \dfrac{\operatorname{ctn} x\beta_1(\sigma)}{\beta_1(\sigma)} & \dfrac{\operatorname{ctnh} x\beta_2(\sigma)}{\beta_2(\sigma)} & \dfrac{-\tan y\beta_3}{\beta_3} & \dfrac{-\tanh y\beta_4}{\beta_4} \\ -1 & 1 & 1 & 1 \\ \dfrac{\operatorname{ctn} x\beta_1(\sigma)}{\alpha_1\beta_1} & \dfrac{\operatorname{ctnh} x\beta_2(\sigma)}{\alpha_2\beta_2} & \dfrac{-\tan y\beta_3}{\alpha_3\beta_3} & \dfrac{-\tanh y\beta_4}{\alpha_4\beta_4} \\ -\alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 \end{vmatrix} = 0 .$$

A trial-and-error method was used.

ANALYSIS OF VARIANCE FOR RANDOMIZED BLOCK EXPERIMENTS

D. A. Gardiner

This routine will print out, via paper tape, the standard analysis of variance table, tables of means, and, where appropriate, tables of cell variances of a randomized block experiment, with or without repeat measurements.

The input tape is composed of the number of blocks (in hex), the number of treatments (in hex), the number of measurements per treatment per block (in hex), a carriage return, and the words representing the observations in decimal floating-point form. The words are punched block by block, with the same order of treatments followed in each block.

The output tape contains tabular stop and margin instructions, followed by the analysis of variance information. All tables printed are in standard form with decimal points properly aligned.

ORACLE PROGRAM FOR CALCULATING THE CHI-SQUARE TEST STATISTIC IN AN ORDINARY $R \times S$ CONTINGENCY TABLE

M. A. Kastenbaum

This program is designed to compute the simple chi-square test statistic for two-dimensional contingency tables varying in size from 2×2 to 256×256. Inasmuch as there is no iteration necessary to compute this test statistic, the calculations are extremely rapid. The use of a high-speed machine is justified only if the contingency tables are very large or if a large number of chi-square statistics are to be computed. For this latter problem, the program may be placed on "continuous," whereupon the Oracle will read each set of data in its turn, do the necessary calculations, punch out (1) the value of chi-square, (2) the degrees of freedom, (3) the expected cell frequencies, and (4) the deviations from expectation in each cell, and then continue to the next set of data.

ORACLE PROGRAM FOR TESTING THE HYPOTHESIS OF "NO-INTERACTION" IN AN $R \times S \times T$ CONTINGENCY TABLE

M. A. Kastenbaum

The primary difficulty in testing the hypothesis of "no-interaction" as given by Roy and Kastenbaum[16] for a three-way contingency table is the solution of the simultaneous systems of third-degree equations which arise from the "no-interaction" constraints. This problem may be handled now by the Oracle with a program which will take data from contingency tables varying in size from $2 \times 2 \times 2$ to $5 \times 5 \times 16$. The program will solve up to 240 simultaneous third-degree equations (of the type peculiar to the present problem), compute the chi-square test statistic, and punch out a solution to the equations, the expected cell frequencies, and the deviations from expectation in each of the cells of the table.

MAGNETIC FIELD CALCULATIONS FOR DCX

N. B. Alexander　　A. C. Downing　　B. S. Rose

It is well known that in the case of a single circular current the magnetic field is given in terms of elliptic functions. Formulas for the fields due to solenoids with cylindrical symmetry are apparently not generally available in the literature. Among the standard reference works on magnetic fields is the report by Snow.[17] He reduces each of the components of the field to a sum of four functions, each of which is an infinite series of Legendre polynomials.

Since sophisticated formulas convenient for digital computation do not seem to be available, the field expressions were developed as double integrals of elementary transcendental functions. A set of Oracle routines was written for computing both components of the field and the vector magnetic potential.

A separate Oracle program was written for determining the relative dimensions for a pair of coils with rectangular cross sections, a prescribed Fabry factor, and a fixed volume. This involved solving a pair of simultaneous transcendental equations.

---

[16] S. N. Roy and M. A. Kastenbaum, *Ann. Mathematical Statistics* 27(3), 749–57 (1956).

[17] C. Snow, *Magnetic Fields of Cylindrical Coils and Annular Coils*, National Bureau of Standards Applied Mathematics Series No. 38 (1953).

EXPONENTIAL FUNCTION

A. C. Downing

The fixed-point subroutine for evaluating the exponential series was rewritten for increased speed and accuracy. A precise error analysis was made and has been distributed as part of the preliminary report on this subroutine.

OVERLAP INTEGRALS

D. B. Grimes

This program evaluates the following integral by use of (8, 32) floating-point arithmetic and the Gaussian quadrature:

$$S = \tfrac{3}{4} N_a N_b \left(\frac{1}{2R} \right)^{m_a+m_b+1} \cdot \int_1^{\infty} \left[\int_{-1}^{+1} (\xi + n)^{m_a-2} (\xi - n)^{m_b-2} \cdot \right.$$

$$\left. \cdot\ (\xi^2 - 1)(1 - n^2)(\xi^2 - n^2)\, e^{-\rho(\xi + nt)}\, dn \right] d\xi \quad ,$$

where

$$\frac{1}{N_a^2} = \int_0^{\infty} r^{2m_a} e^{2u_a r/a_H}\, dr \quad ,$$

$$a_H = 0.529171 \quad ,$$

$$\rho = \frac{(u_a + u_b)\, R}{2a_H} \quad ,$$

$$t = \frac{u_a - u_b}{u_a + u_b} \quad .$$

INTEGRATION

D. B. Grimes

Floating-point arithmetic and Gaussian single integration were used to evaluate the following integral:

$$t = \int_0^x \frac{(\xi + C)\, d\xi}{C + \xi A e^{-b\xi}} \quad ,$$

$$x = [0\ (0.50)\ 10] \quad ,$$

$$x = [10\ (10)\ 100] \quad ,$$

$$x = [100\ (10)\ 1000] \quad .$$

CORNPONE A

D. B. Grimes

Cornpone, written by W. Kinney, has been set up to do essentially Eyewash (UNIVAC) reactor calculations on the Oracle for the Fused-Salt Reactor Group of the Reactor Projects Division. The Eyewash cross sections and group structure have been put on a permanent magnetic-tape library. Added to Kinney's edit is a curve-plotter edit of the absorptions in a region by element. Portions of the output from this code have been adapted as input for Sorghum, a reactor burnout code. Production problems have been running since March 1, 1958.

DIFFUSION OF CHROMIUM INTO INOR AND INCONEL

D. B. Grimes

In the course of experimental investigations of diffusion of chromium in nickel alloys being conducted by G. M. Watson of the Reactor Chemistry Division, it was hypothesized that the data could be correlated by assuming that the diffusion coefficient is dependent upon the concentration of the chromium. It was therefore of interest to solve the corresponding diffusion equation given below, by means of an Oracle program.

This program, written in fixed point, solved the following differential equation by means of a difference equation:

$$\frac{\partial}{\partial x} D \frac{\partial C}{\partial x} = \frac{\partial C}{\partial t} ,$$

where

$$D = D^0 \left(1 - \frac{BC}{C_\infty} \right) ,$$

$$B = \frac{AC_\infty}{D^0} .$$

Initial conditions were

$$C_{0,0} = 0 ,$$

$$C_{1,0} = 1 - 2^{-39} ,$$

$$C_{n,0} = 1 - 2^{-39} .$$

To test the accuracy of initial conditions, the following changes were made:

$$C_{0,0} = 1 - 2^{-39} ,$$

$$C_{n,0} = 1 - 2^{-39} ,$$

$$C_{0,t} = 0 .$$

After 90 days the maximum difference in C/C_∞ was 6.1%. After 180 days it was 3.2%, and after a period of one year it had decreased to 1.4%. The results of the calculation were assembled in the form of a table listing values of ratios of C/C_∞ as a function of time (t) and position (x), for the use of the Reactor Chemistry Division.

CALCULATION OF INTERACTION OF CHARGED COLLOIDAL PARTICLES

D. J. Wehe

A program was written which calculated the relative energy required for charged colloidal particles to approach each other. The energy was calculated as a function of temperature, valence, radius of the particle, concentration of particles, and distance between particles.

Fifteen hours of computing time was required, and 420 tables were calculated. The results will be published in an ORNL report.

INTERNAL-CONVERSION COEFFICIENTS

D. J. Wehe

With the use of a linear interpolation routine and a set of tables of the internal-conversion coefficients for eight molecular weights (Z) and ten energy levels (k), the coefficients were calculated for intermediate Z with each k. Nine tables of radical integrals $R_{-L}(m)$, $R_{L+1}(m)$, $R_L(e)$, $R_{-L-1}(e)$ were calculated.

The results were reported via magnetic tape in a form suitable for publication. Although the computational time was relatively small, approximately 48 hr was required to print the results. The tables, exactly as printed by the Oracle, were photocopied to provide plates for the book by Rose.[18]

SOLUTION OF SIMULTANEOUS DIFFERENTIAL EQUATIONS

D. J. Wehe

The system

$$\frac{dn_+}{dt} = A(1 - n_+^2) - Bn_+n_0 \quad ,$$

$$\frac{dn_0}{dt} = C(1 - n_0) - Dn_+n_0 \quad ,$$

with the conditions

$$n_+(0) = 0 \quad ,$$

$$n_0(0) = 1 \quad ,$$

was solved by a code written in FORTRAN for the IBM-704 computer at the K-25 plant. Use was made of a prepared subroutine to solve the system by the Runga-Kutta method.

[18] M. E. Rose, *Internal Conversion Coefficients*, Interscience, New York, 1958.

Because of the large range of the coefficients an extremely small interval (10^{-6}) was required, implying that many values of the functions $n_0(t)$, $n_+(t)$ would have to be calculated, with 15 to 20 min machine time required for each case. Therefore only a few cases were calculated, and the results were used as a test for those obtained on the analog computer.

DECAY OF RADIOACTIVE MATERIALS

N. B. Alexander B. S. Rose

The problem is to determine the amount of decay of certain radioactive materials at various time intervals. Functions such as the following were evaluated:

$$\frac{N_i}{N_1^0} = \lambda_1 \lambda_2 \ldots \lambda_{i-1} \sum_{j=1}^{i} \frac{e^{-\lambda_i t}}{\prod_{k \neq j} (\lambda_k - \lambda_j)} .$$

SPECIFIC HEATS OF ALLOYS

N. B. Alexander R. K. Benson E. C. Long

The problem is to calculate the specific heats of various alloys and the probable errors in such calculations.

GAMMA-RAY ABSORPTION

N. B. Alexander

In a study of the mechanism by which atomic holes are produced in solids exposed to gamma radiations, the cross sections for producing energetic electrons by the Compton effect, the photoelectric effect, and pair production were tabulated for a wide range of atomic weights. These cross sections were obtained by numerically integrating single, double, and triple integrals. One of the integrands was calculated by fitting a table of values by means of the linear regression code. Ninety per cent of the computation has been completed on this problem.

SUBROUTINE FOR SUBTABULATION

N. B. Alexander A. C. Downing

A subroutine was written for subtabulation of a function by introducing a specified number, $m - 1$, of intermediate values between consecutive pairs of given values. A table of the differences Δ^j of the given values is constructed, from which a new table of differences is calculated.

Let $f(x)$ be given at the equally spaced points $x_0, x_1, \ldots, x_n$ and let

$$u = \frac{x - x_0}{x_1 - x_0} .$$

Then Newton's expansion for $f(x)$ in terms of the old differences (Δ) is

$$f(x) = f(x_0) + \binom{u}{1}\Delta_0 + \binom{u}{2}\Delta_0^2 + \ldots$$

$$= f(x_0) + \sum_{k=0}^{\infty} \binom{u}{k}\Delta_0^k \quad .$$

In terms of the new differences (δ), it is

$$f(x) = f(x_0) + \sum_{k=0}^{\infty} \binom{nu}{k}\delta_0^k \quad ,$$

where

$$\delta^k = \sum_{j=k}^{\infty} a_j^k \Delta^j$$

is obtained recursively from the components of an auxiliary vector a^k. Let B be the lower triangular matrix:

$$\begin{pmatrix} 0 & 0 & 0 & \ldots & 0 & 0 \\ \binom{n^{-1}}{1} & 0 & 0 & \ldots & 0 & 0 \\ \binom{n^{-1}}{2} & \binom{n^{-1}}{1} & 0 & \ldots & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \binom{n^{-1}}{m-2} & \binom{n^{-1}}{m-3} & \binom{n^{-1}}{m-4} & \ldots & 0 & 0 \\ \binom{n^{-1}}{m-1} & \binom{n^{-1}}{m-2} & \binom{n^{-1}}{m-3} & \ldots & \binom{n^{-1}}{1} & 0 \end{pmatrix} .$$

Then a^k is defined by the conditions

$$a^0 = e_1 = \begin{pmatrix} 1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{pmatrix} , \qquad a^{k+1} = Ba^k \quad .$$

Hence

$$a^k = B^k e_1 \quad ,$$

and

$$\delta_0^k = \left(f(x_0),\ \Delta_0,\ \Delta_0^2,\ \ldots,\ \Delta_0^{m-1} \right)\ a^k \quad .$$

FUNCTION EVALUATOR SKELETON

A. C. Downing

The normal subroutine entry for the Oracle places two addresses in the accumulator: the address in the main code from which the subroutine was entered and the address of the first word of the subroutine. With this information available all subroutines of a given class may make use of a common slave routine which will perform the red-tape operations for all members of the class. Compiler item 00 088 was written for use by the function evaluation routines, specifically, those using a linkword or a Q-register word to specify $M(x)$ and $M[f(x)]$, that is, the storage locations of member x and the location where $f(x)$ is to be placed. The slave (item 00 088) fetches both of these addresses and brings x into the accumulator before transferring control back to the function-evaluating subroutine. After $f(x)$ is computed, item 00 088 stores it in $M[f(x)]$ and transfers control back to the proper point in the main routine. Thus the red-tape operations in the function evaluation subroutine are reduced to a minimum, namely, two transfers to item 00 088.

Provision is also made in item 00 088 for monitoring subroutine stops (e.g., for illegal values of x) on the console typewriter. The subroutine causing the stop is identified by its storage location, and the return address in the main code is also reported.

Item 00 088 is eight words in length.

THE OAK RIDGE RANDOM RESEARCH REACTOR ROUTINE (O5R): A GENERAL-PURPOSE MONTE CARLO CODE FOR THE IBM-704

H. P. Carter R. R. Coveyou[19] J. G. Sullivan[19]

A general-purpose Monte Carlo code[20] is being developed for the IBM-704. The immediate purpose of the code will be to facilitate the analysis of the problems which arise in the calculations and measurements of the age of neutrons in water.

SIMPSON'S RULE INTEGRATION SUBROUTINE

C. T. Fike

This is a closed subroutine which uses Simpson's rule for numerical quadrature. Computing is done entirely in fixed point, and certain simple scaling restrictions are made

---

[19]Neutron Physics Division.

[20]R. R. Coveyou, J. G. Sullivan, and H. P. Carter, *Neutron Phys. Ann. Prog. Rep. Sept. 1, 1958*, ORNL-2609, p 87.

therefore on the integral and integrand in addition to the customary conditions which must be met for the application of Simpson's rule.

FLUX DEPRESSION

B. S. Rose

The problem was to calculate the flux depression due to gold foil in various media. The mathematics of the problem reduced to the evaluation of a double integral.

ORACLE TIME CODE

D. J. Wehe

A code was written to minimize the bookkeeping required in the charging for the time used on the Oracle. From the cards which are stamped at the console each time a programmer uses the machine, the operator types a tape containing the problem number, the amount of time used, and the type of computing done, that is, whether the time was used for production or debugging or whether the time was lost because of a machine or power failure. With this information and a list of account numbers and problem numbers stored on magnetic tape, the program computes three reports: (1) a tabulation of production, debugging, and lost time, with a breakdown of the time used by the engineers and mechanics; (2) a listing of the divisions to which time has been charged and the amount of time used by each problem belonging to that division; (3) a financial report, which is sent to the accounting department at K-25, stating the charge to each account using the Oracle.

In addition to the time code itself, a series of codes is used to change or add new problems or accounts to the present list stored on magnetic tape.

From time to time, changes have been made in the calculations and format, but such changes have been relatively simple.

SUBROUTINES AND REPORTS[21]

D. J. Wehe

Reports of 19 subroutines already on the Compiler were rewritten in a form consistent with present usage and were published as preliminary reports.

Two routines were written: one to perform decimal addition and another to punch any portion of a word. Preliminary reports for these routines are available.

An (8,32) subroutine for arc sin x, written by the MIT practice students, was rewritten in compiler form, and the report was revised for general distribution.

Reports were written for two subroutines programed by H. Moran [error function and derivative, (40, 40) arc tan x].

[21] Available from the Oracle subroutine library.

COMPILER AND COMPILER LIBRARY

B. J. Osborne D. J. Wehe B. S. Rose

Codes have been written to make automatic the following changes to the Compiler:

1. scanning subroutines to determine slaves,
2. rearranging subroutines on the Compiler,
3. correcting subroutines on the Compiler,
4. writing over old subroutines and adding new subroutines,
5. getting a new directory from the Compiler with items in numerical order,
6. changing constants.

Three permanent constants were removed and all the alphameric symbols added. These are located from E00 through E3F. The Compiler has also been changed so that it will accept alphameric items.

A new and complete writeup for the Compiler was issued.[22]

A complete, dated edit has been made of the Compiler library and is available for reference. This listing is kept current, and anyone requesting a copy of an item may obtain a Thermo-Fax copy showing the date on which the edit was made.

A system has been introduced in which the author of a new routine receives an edit of his routine after it has been placed in the library. The paper-tape copy is dated and kept on file. As changes in a routine are requested, a dated edit of the altered routine is sent to the author or to the person requesting the change.

Routines which were obsolete due to changes in the machine and routines about which no information was available were removed from the library and filed.

CODE FOR CLASSIFICATION OF SUBROUTINES

B. J. Osborne D. J. Wehe N. B. Alexander N. M. Dismuke

A code has been completed for producing three classifications of subroutines in the Oracle subroutine library from a list stored on magnetic tape. The code also has provisions for deleting, adding, changing, and sorting as is necessary for updating this list.

Classification I is a listing of routines according to their mathematical and computational purposes. A routine may be included several times in this list. Classification II is a breakdown of routines by arithmetic or numerical representation. Classification III lists the routines in increasing numerical order of library number.

---

[22] M. E. LaVerne, R. R. Coveyou, and R. R. Bate, *Programmers Manual for the Oracle Compiler*, ORNL CF-57-8-92 (Aug. 23, 1957).

TRACER

B. J. Osborne D. J. Wehe

An automonitoring routine (the tracer) was completed but was too slow to be used effectively. It is now being rewritten, and the following new features are being added:

1. It will trace between address *aaa* and address *bbb*. This eliminates much output from subroutine loops.

2. When the (8, 32) floating-point routine is included in the program being traced, the format tape requires the beginning address, in which case it will give the following output for floating-point orders:

Address	Order	Results
aaa	*xxx* α *yyy zzz*	*C*(*zzz*)

3. More than one section of temporary storage, which will be kept in the memory at all times, can be specified by the programmer, thus speeding up the operation.

It is hoped that the new tracer will be able to keep at least half the program of the code being traced in the machine at one time. This will speed up the monitoring by a factor of 4.

CANNED PROGRAMS AND SERVICE PROGRAMS

B. J. Osborne

A new cataloging method was started in these groups. Canned programs were given a numbering system consisting of a three-digit number preceded by the letters "CP." These programs are now available in the tape preparation room.

Service programs were broken down into two groups: those needing input tapes, which were numbered SP 100, and those not needing input tapes, which were numbered SP 000. Both series are available at the computer console.

SERVICE PROGRAM TO RECONSTRUCT SEVERAL ITEMS INTO ONE COMPILER ITEM

N. B. Alexander

A need was felt for a service routine which would combine several items into one item for later use as one subroutine on the Compiler. This code uses the compiled program from the several items, the directory of the compiled version, and the items themselves in order to calculate the new key words, to reinsert the universal constants, to change any zero item into zero words at the end of the new item, and to check for length and references to routines outside itself.

SUPER "B"

N. B. Alexander E. C. Long

This subroutine simplifies the coding of many problems by setting up the entrances to other subroutines. Hence, once in this subroutine, the programmer could give a list of three-address orders that would make programing automatic. The subroutine has been revised in order to increase its usability.

LIST OF SUBROUTINES ON COMPILER

B. S. Rose

This service routine will produce an alphameric compiler directory listing the subroutines in numerical order with their associated magnetic-tape block number and listing the subroutines in the order in which they appear on the compiler tape.

INSERTION OF LINK-WORD INFORMATION INTO THE CLASSIFICATION DIRECTORY

B. S. Rose

This service routine will add the alphameric link-word information to the classified titles of the compiler subroutines which form the classification directory.

ORACLE ITEM EDIT

D. B. Grimes

There are two versions of this code available. The first edits the item from paper tape into a convenient form for checking and correcting. In addition to the key words, it punches an address followed by a word plus extra words without addresses. The second version edits items from private library tapes and gives a neater edit than the first one because it separates the order and address in each word.

OUTPUT CONVERSION AND EDIT SEGMENT FOR CP 00B AND CP 00C

E. L. Cooper N. M. Dismuke

A segment for putting results on paper tape or magnetic tape has been added to the fixed-point inversion and linear equation codes written earlier.[23] The codes are part of the canned-program library. Both programs solve s sets of k equations in k variables with the size limitations

$$\text{CP 00B:} \quad 0 \leqq k \leqq 127 \,, \qquad n = k + S \leqq 4095 \;;$$

$$\text{CP 00C:} \quad 128 \leqq k \leqq 2SS \,, \qquad n = k + S \leqq 4095 \;.$$

A preliminary report describing the codes has been published.

---

[23] *Math. Semiann. Prog. Rep. Dec. 31, 1955*, ORNL-2037, p 4; *June 30, 1956*, ORNL-2134, p 10; *Feb. 28, 1957*, ORNL-2283, p 9.

THREE-WORD INTERPRETIVE COMPLEX NUMBER SUBROUTINE

A. A. Grau

As a coding device, it is useful to write the orders for interpretive arithmetics used on the Oracle in a way that is mnemonically reminiscent of the corresponding fixed-point orders. Routines using this principle for (28, 12) and (40, 40) floating-point operation have been used on the Oracle for some time. In this convention the subroutine entry is followed by an arbitrary sequence of instructions which, although resembling machine instructions, are interpreted by the code as the corresponding floating-point operations. The regular A and Q registers are replaced by interpretive A and Q registers, each of which consists of reserved space in the memory. Exit is effected by the presence of a 1 bit in the breakpoint position of a transfer order that is executed.

A three-word interpretive complex number routine has been designed and constructed which handles complex numbers in the form

$$(x + yi) \cdot 2^z \quad .$$

The numbers x, y, z occupy successive memory cells; in the interpretive order, only the address x is mentioned. Here $-1 \leqq x < 1$ and $-1 \leqq y < 1$, while z is a binary integer. The number in question is normal if either x or y satisfies $-1 \leqq t < -\frac{1}{2}$ or $\frac{1}{2} \leqq t < 1$. Almost all standard arithmetic orders are interpreted by this routine.

The routine has been completed and tested but has not been added to the library.

FORMULA INTERPRETATION CODE

A. H. Culkowski A. A. Grau

This routine was written to translate mathematical expressions or equations from alphameric form to modified Polish pentahex notation. Once the expression is in the modified Polish notation, it can be evaluated in the Oracle by suitable codes.

ALPHAMERIC DECIMAL INPUT AND CONVERSION TO NORMALIZED (40,40) FLOATING POINT, ITEM 0DE

A. H. Culkowski A. A. Grau

This routine converts alphameric decimal numbers to normalized (40, 40) floating-point numbers. The numbers may be positive or negative, with or without a decimal, and have a power of 10 if desired. The number to be converted is limited to 11 digits and may contain commas, spaces, plus signs, and carriage returns, all such symbols being ignored by the code.

ALPHAMERIC NUMERICAL INPUT AND CONVERSION TO BINARY WITH DECIMAL SCALE FACTOR, ITEM 06F

A. H. Culkowski A. A. Grau

This subroutine converts alphameric numerical numbers to binary with a decimal scale factor. The numbers to be converted may be positive or negative, with or without decimal point, and may contain an exponent of 10 if desired. All punctuation symbols and plus signs, other than tabs, that may be included with the numbers are suppressed by the code.

A special code similar to ODE was devised for use in either ORSAP or ORBIT.

BI-HEX CHARACTER INTERPRETER AND PLOTTER, ITEM 0BC

A. H. Culkowski A. A. Grau

This subroutine will interpret and plot on the curve plotter any character of the Oracle bi-hex code to give curve-plotter, alphameric output similar to that obtained by the same bi-hex character on the off-line printer. Characters such as space, upper case, lower case, back space, tab, and carriage return are interpreted as they would be on a typewriter, and characters that would be typed are plotted.

KEY WORD LIST FOR CURVE PLOTTER, CONSISTENT WITH PERIPHERAL BI-HEX CODE, ITEM 0BD

A. H. Culkowski A. A. Grau

This item is a slave alphabet for item 0BC, which interprets all bi-hex characters for the curve plotter. It is a list of the key words which will describe the bi-hex characters they represent.

THE (8,32) LOCATOR FOR AN ISOLATED REAL ZERO OF AN ARBITRARY FUNCTION

N. B. Alexander

This subroutine will find one zero of an arbitrary function between two given points provided that the signs of the function at the two given points are opposite. The method employed is to halve the interval, calculate the value of the function at the half point, select a new interval of length half that of the previous one, and repeat the above three steps until a sufficiently small value is found for the function.

SOME ROUTINES FOR CONVERTING, EDITING, AND SCALING

E. L. Cooper

Binary-to-Decimal Conversion – One-Character Output. – A list of machine numbers, identically scaled, is put out on paper tape or narrow magnetic tape in alphameric form. Up to 15 digits per number can be reported. The code makes use of the single-character output order, recording on magnetic tape at maximum density.

Matrix and Eigenvector Conversion and Output Edit. – A rectangular matrix of digitalized elements is put out on paper tape or narrow magnetic tape as alphameric numbers with a matrix format. Eigenvalues and eigenvectors can be handled by changing a hex digit in one link word. This routine is available in both subroutine and canned program forms.

Calculation of Scale Factor and Power of 10 for Output Scaling. – Given the binary exponent b, this routine will determine the decimal exponent d and the scale factor $R = 2^{-b} \cdot 10^{d}$, where $\frac{1}{10} \leqq R < 1$; R is obtained from a double-precision number with rounding in the last stage. When a list of numbers, instead of one number only, is to be scaled by a common factor, R, this routine provides greater efficiency.

MATRIX NORM AND SCALING CODE

C. T. Fike

This is a self-contained code which will compute the sum of the squares of the elements of a matrix stored in fixed-point and scale the matrix down by a power of 2 so that the sum of the squares of the elements of the scaled matrix is less than unity. Partial sums are accumulated in multiple-precision arithmetic to prevent loss of accuracy, and the matrix is not scaled down any more than is necessary to make the sum of squares less than unity.

CURVE-PLOTTER MATRIX OUTPUT CODE

C. T. Fike

This is a self-contained code written to provide fast output on the curve plotter of a fixed-point matrix stored on magnetic tape. Conversion is made from binary to decimal, with provision for scaling the matrix up by a power of 2.

FLOATING-BINARY-POINT ARITHMETIC FOR REAL AND COMPLEX NUMBERS

S. E. Atta

Subroutines for floating-binary-point arithmetic[24] for both real and complex numbers were written in an attempt to accomplish the objectives of reducing to a minimum the time required for floating-point operations, using the regular single-address Oracle instructions for ease in coding as well as in debugging, and obtaining the maximum accuracy in single precision.

Separate subroutines for summation, multiplication, and division were written for each number field. Any of the regular Oracle instructions for summation or product of real numbers can be executed in floating point by using one of these subroutines. Regular Oracle instructions for summation or product can be executed in floating-binary-point arithmetic for complex numbers with the exception of the modulus (absolute value) instructions.

Two groups of codes are available for each number field. In one, two storage cells are required for each number (40,40), while in the other the mantissa and exponent of a number are packed in a single storage cell (8,32).

---

[24] *Math. Semiann. Prog. Rep. Dec. 31, 1954*, ORNL-1842, p 20; preliminary reports of Oracle subroutines, item numbers 00 0E4–00 0F1.

AUTOMONITORING FOR (8,32) ARITHMETIC

B. S. Rose

This service program is useful in debugging three-address codes. It will punch out the location of each order, what the order is, the two numbers being operated upon, and the result of the operation.

BINARY-TO-DECIMAL CONVERSIONS

J. Harrison

Four binary-to-decimal conversion codes have been designed to convert fixed-point integers, fractions, and numbers with decimal scaling. Each one is designed to take advantage of the free time between successive character punches when narrow magnetic tape is being used. Except for suppressing zeros following the exponent (for a decimal scale factor), the computing time for each character is below 900 μsec.

The integers and fractions are converted by means of the same loop. The conversion of an integer p to decimal form is reduced to the case of conversion of a fraction by noting that if the machine representation of p, $N = p \cdot 2^{-39}$, is multiplied by $2^{39}/10^k$, the number $P = p \cdot 10^{-k}$ is obtained whose decimal representation differs from that of p itself only in the position of the decimal point. Multiplication by $2^{39}/10^{11}$ is effected by shifting three places left and then multiplying by $m/8$ = 57F5FF85E6.

Items 00 08A and 00 08B convert and punch numbers of the form $N \cdot 10^P$ composed of a binary fixed-point part N and a decimal scale-factor part P. The reported number is composed of a decimal fraction and a decimal scale factor. Item 00 08A handles scale factors up to and including 1023_D, and item 00 08B handles scale factors up to and including 11 decimal places. Item 00 08C converts a starred binary integer to a decimal integer, and item 00 08D converts a binary fixed-point number to a decimal number.

THE n nth ROOTS OF A COMPLEX NUMBER

R. K. Benson

By use of the formula

$$(k)^{1/n} = e^{(\log_e k)/n}$$

for the nth root of a real number, this code provides the nth roots of complex numbers by expressing them first in polar form.

THREE-ADDRESS COMPLEX ARITHMETIC IN FIXED-POINT BINARY

R. K. Benson

This routine will add, subtract, multiply, and divide complex numbers and store a scale factor, if the answer is not digital, by shifting the two complex numbers being operated upon until they do not produce overflow.

THREE-ADDRESS CODING FOR (28,12) COMPLEX ARITHMETIC

R. K. Benson

This routine allows the programmer of a problem to code with three-address words which the routine interprets and sets up for the (28,12) complex arithmetic subroutine.

MISCELLANEOUS SHORT CODES

A. H. Culkowski

The following miscellaneous short codes were written during the period of this report:

1. code to manufacture integrand and limits for a double integral to be evaluated by use of Gaussian integral code;
2. HRE-III vessel analysis: heat generation data code, temperature data code, and stress data code;
3. variables needed for input in the generalized linear regression code;
4. first-, second-, or third-degree equation solver, given the variables and coefficients;
5. standard error of estimate for the generalized linear regression code.

SUBROUTINE FOR (8,32) FLOATING-POINT ARC-TANGENT

M. Feliciano

The arc-tangent of an (8,32) floating-point number $x = x_m \cdot 2^p$ is computed in one of the following manners.

1. If $-8 \leqq p \leqq 0$, then x is changed to fixed point and the arc-tangent is computed by the method of continued fractions.

2. If $-62 \leqq p \leqq -9$, then the arc-tangent is computed by using the first two terms of the series

$$\text{arc tan } x = x - \frac{1}{3}x^3 + \frac{1}{5}x^5 - \dots \ .$$

3. If $-128 \leqq p \leqq -63$, then the approximation arc tan $x = x$ is used.

4. If $p > 0$, the identity arc tan $1/x$ = arc cot x is used.

5. If $x = 0, +1, -1$, then the value of the arc-tangent is substituted without computation.

ORBIT TRANSLATOR PROGRAM

J. Harrison J. J. Andrews A. A. Grau

The purposes of the five segments contained in ORBIT are as follows:

Segment I loads into the memory the alphameric program and converts it to an internal bi-hex representation by adding 40(hex) to each upper-case character. This modification is necessary for proper interpretation of all available alphameric characters, including those in upper case and lower case.

Segment II places each ORBIT statement[25] on a separate block on drive 1, where it is taken and processed by segment III. Zero items for the variables are manufactured in this segment.

Segment III, the main segment, processes an ORBIT program and produces a finished object program ready for compiling. This segment was originally too large for the memory and had to be rewritten.

Segment IV forms key words for the object program and creates items containing constants and the statement number (SN) list.

Segment V reads in segment I of the Compiler, changes a few orders, supplies the basic item word, and transfers control to the Compiler.

ORBIT is designed to accommodate alphameric statements, each statement consisting of an equation, iteration, conditional or unconditional transfer, subroutine, or input or output order. Each statement is dissected into its alphameric characters by processing one pair of characters at a time. Every possible combination of permissible characters lies in a table of pairs which actually is the hub of ORBIT. This table contains over 160 pairs, with corresponding transfers to items producing the required machine orders.

ORBIT contains, in addition to the table of pairs, another interesting coding device which supplies addresses of transfers. Each time a numbered statement[25] is processed, the statement number together with a transfer to the object code address of the statement is placed in the SN list. When a reference to a statement not yet in the SN list appears, the SN alone is entered in the SN list and the object code transfers to the vacant address portion of the SN in question. When this particular SN is reached, its object code address is then placed in the proper position beside the SN. Thus the SN list links transfers and addresses in the object code.

Iterations are processed by storing the maximum, minimum, increment, variable, and return address and then creating the initialization, increase, and conditional transfer statements at the proper time. A variable transfer called the "tau switch" regulates the timing of these manufactured statements.

ORBIT's numerous subroutines include the advance instruction subroutine designed to insert additional orders between orders already produced and to change addresses accordingly. The COM subroutine inserts each order in a list, one order per word, and remembers the object code address of the present order. Subroutine IML programs a stall, if necessary, to assure that the next instruction will be a left-hand one when coding subroutine entries, transfers to the left, etc.

---

[25]Statements referred to by other statements or immediately following an iteration *must* be numbered. Numbering of other statements is optional.

After each statement has been processed, the instruction processor picks up the orders by two's from the above-mentioned list and places them in the final object code.

ORBIT uses two conversion codes: one which converts alphameric decimal numbers to binary and one which slaves the first to produce normalized (8,32) floating-point numbers.

COMPILER ITEMS AND SUBROUTINES

Sixty-five compiler items were added to the Compiler library by members of the Mathematics Panel during the period March 1, 1957, to September 1, 1958; the classification used here is one that is in standard use with the Compiler library. The data are given in Table 5.

Table 5. Items Added to Compiler Library

Item Number	Compiler Item	Contributor
	Programed Arithmetic	
Real Numbers		
00 0E8	(8,32) floating-point multiplication	S. E. Atta
00 0E9	(8,32) floating-point division	S. E. Atta
00 0EA	(8,32) floating-point addition	S. E. Atta
03 0A0	Super "B": simplified coding in (8,32)	N. B. Alexander, E. C. Long
03 205	(8,32) packed automonitoring floating-point arithmetic	B. S. Rose
00 0E4	(40,40) floating-point multiplication	S. E. Atta
00 0E5	(40,40) floating-point division	S. E. Atta
00 0E6	(40,40) floating-point addition	S. E. Atta
Complex Numbers		
00 0EF	(8,32) floating-point multiplicotion	S. E. Atta
00 0F0	(8,32) floating-point division	S. E. Atta
00 0F1	(8,32) floating-point addition	S. E. Atta
00 0EC	(40,40) floating-point multiplication	S. E. Atta
00 0ED	(40,40) floating-point division	S. E. Atta
00 0EE	(40,40) floating-point addition	S. E. Atta
Decimal		
01 0B7	Decimal oddition	D. J. Wehe
	Elementary Functions, Fixed Point	
Exponential and Logarithmic		
00 092	e^{-x} (series)	A. C. Downing
00 192	e^{-x} (series)	A. C. Downing
Roots and Powers		
00 00F	Square root (Newton)	A. C. Downing
00 10F	Square root (Newton)	A. C. Downing
	Elementary Functions, Floating Point	
Trigonometric		
00 091	(8,32) arc-tangent (series, trigonometric identities)	M. Feliciano

Table 5 (continued)

Item Number	Compiler Item	Contributor
	Polynomials and Special Functions	
00 0B3	Fixed-point locator for isolated zeros of arbitrary functions	A. C. Downing
00 0B4	(8,32) locator for isolated zeros of arbitrary functions	N. B. Alexander
00 088	Functional subroutine skeleton	A. C. Downing
	Operations on Functions and Solutions of Differential Equations	
Integration		
00 093	Simpson's rule integration	C. T. Fike
Others		
00 0B3	Fixed-point locator for isolated zeros of arbitrary functions	A. C. Downing
00 0B4	(8,32) locator for isolated zeros of arbitrary functions	A. C. Downing
	Interpolation and Approximations	
Table Lookup and Interpolation		
00 0F2	(8,32) interpolation (Newton, 03 005)	N. B. Alexander
00 0D9	Binary table lookup, one word	A. A. Grau
00 0DA	Binary table lookup, multiple word	A. A. Grau
Curve Fitting		
00 0F3	(8,32) least-square fitting for linear functions	E. C. Long
	Operations on Matrices and Vectors	
00 0BE	Matrix and eigenvector conversion and output code	E. L. Cooper
00 0D8	Computation of blocks required for vector and matrix storage	A. C. Downing
	Statistical Analysis and Probability	
00 071	(8,32) sum and sum of squares	D. A. Gardiner
	Input	
Alphameric		
00 06F	Alphanumerical input and conversion to binary floating point with decimal exponent	A. H. Culkowski
00 0DE	Alphameric input and conversion to (40,40) floating point	A. H. Culkowski
	Output	
(8,32) Floating Point		
00 095	(8,32) output conversion with choice of decimal places	D. A. Gardiner
00 098	(8,32) floating-point alphameric conversion and punch	T. W. Hildebrandt
00 336	(8,32) output, alphameric	E. C. Long
(40,40) Floating Point		
00 096	(40,40) floating-point alphameric conversion and punch	T. W. Hildebrandt
(28,12) Floating Point		
00 097	(28,12) floating-point alphameric conversion and punch	T. W. Hildebrandt

Table 5 (continued)

Item Number	Compiler Item	Contributor
	Output	
Fixed Point		
00 089	Binory-to-decimal conversion, one-character output	E. L. Cooper
00 08A	Fixed-point conversion with decimal scaling, binary to decimal, alphameric output, exponent less thon 1024	J. Harrison
00 08B	Fixed-point conversion with decimal scoling, binary to decimal, alphameric output	J. Harrison
00 08C	Integer conversion, binary to decimal, alphameric output	J. Harrison
00 08D	Fixed-point conversion, binary to decimal, alphameric output	J. Harrison
00 0BE	Matrix and eigenvector conversion and output code	E. L. Cooper
00 28D	Binary-to-decimal fixed-point conversion with interchangeable curve plotter, punch output	C. T. Fike
Alphameric		
00 0BE	Matrix and eigenvector conversion and output code	E. L. Cooper
00 095	(8,32) output conversion with choice of decimal places	D. A. Gardiner
00 098	(8,32) floating-point alphameric conversion and punch	T. W. Hildebrandt
00 336	(8,32) output, alphameric	E. C. Long
00 096	(40,40) floating-point alphameric conversion and punch	T. W. Hildebrandt
00 097	(28,12) floating-point alphameric conversion and punch	T. W. Hildebrandt
Others		
00 110	One-word typewriter output with zero suppression	C. T. Fike
00 0D6	Punch portion of word	D. J. Wehe
00 10E	Output changer; curve plotter or paper tape	A. A. Grau
	Conversions	
Fixed Point to Floating Point		
00 09A	Binary fraction to (40,40)	T. W. Hildebrandt
Floating Point to Floating Point		
00 0B9	(8,32) to (40,40)	E. C. Long
00 0BB	(40,40) to (8,32)	E. C. Long
Others		
00 0DB	Calculation of scale factor and power of 10 for output scaling	E. L. Cooper
00 023	(40,40) flooting-point binary power of 10 to floating-point binary power of 2	E. C. Long
00 024	(40,40) floating-point binary power of 2 to floating-point binary power of 10	E. C. Long
00 099	(40,40) floating-point binary power of 2 to floating-point binary power of 10	T. W. Hildebrandt
00 0BA	Scaleup of (40,40) floating-point number	J. H. Vander Sluis
	Curve Plotter	
00 28D	Binary-to-decimal fixed-point conversion with interchangeable curve plotter, punch output	C. T. Fike
00 0BC	Bi-hex character interpreter and/or plotter	A. H. Culkowski, A. A. Grau

Table 5 (continued)

Item Number	Compiler Item	Contributor
	Curve Plotter	
00 0BD	Peripheral bi-hex curve-plotter alphabet slaved by 00 0BC	A. H. Culkowski, A. A. Grau
00 10B	Curve-plotter coordinate setter	A. A. Grau, C. T. Fike
00 10E	Output changer; curve plotter or paper tape	A. A. Grau
00 284	Interpretive curve-plotter output, 84 order	A. A. Grau
00 288	Interpretive curve-plotter output, 88 order	C. T. Fike
00 28C	Interpretive curve-plotter output, 8C order	A. A. Grau
	Magnetic-Tape Codes	
00 0BE	Matrix ond eigenvector conversion and output code	E. L. Cooper
00 0D8	Computation blocks required for vector and matrix storage	A. C. Downing
	All Others	
00 08E	Subroutine stop monitor	A. C. Downing

ORBIT SUBROUTINES

The compiler item number and description of the subroutines that have been designed specifically for use with ORBIT are given in Table 6. All but item number 00 109 have been added to the Compiler library. The following auxiliary subroutines are used by many of the subroutines listed in Table 6:

F4	UNPACKER	M. Feliciano
F5	SCALER	M. Feliciano
F6	CODED STOP	M. Feliciano

Table 6. Orbit Subroutine Conventions

Q: an (8,32) number in the Q register
Q_x: a binary integer in the Q register
Acc: an (8,32) number in cell F00
Acc_x: a binary integer in cell F00
A_x: a binary integer in the A register
F01: an (8,32) number in cell F01
$F01_x$: a binary integer in cell F01

Item Number	Description	Contributor
00 0F7	$F01_x \to Q$	M. Feliciano
00 0F8	$Q \to A_x$	M. Feliciano
00 0F9	$-Q \to Q$	M. Feliciano
00 0FA	$\lvert Q \rvert \to Q$	M. Feliciano
00 0FB	$-\lvert Q \rvert \to Q$	M. Feliciano
00 0FC	$Q + Acc \to Q, Acc$	M. Feliciano
00 0FD	$Q \times Acc \to Q, Acc$	M. Feliciano
00 0FE	$Q \div Acc \to Q, Acc$	M. Feliciano
00 0FF	$\sqrt{Q} \to Q$	M. Feliciano
00 100	$\log_e Q \to Q$	M. Feliciano
00 101	$e^Q \to Q$	M. Feliciano
00 102	$Q^{Acc_x} \to Q, Acc$	M. Feliciano
00 103	$F01_x^{Acc_x} \to Q_x, Acc_x$	M. Feliciano
00 104	$\lvert Q \rvert^{Acc} \to Q, Acc$	M. Feliciano
00 107*	Statement linkage	A. A. Grau
00 108	Input	G. C. Caldwell
00 109	Output	E. L. Cooper
00 10A	$Acc \div Q \to Q, Acc$	M. Feliciano

*Used exclusively by ORBIT as needed.

SERVICE PROGRAMS

During the period covered by this report the following service programs were added to the Oracle library:

SP 002	Magnetic-tape item edit	D. B. Grimes
SP 003	Five-to-seven-hole converter	A. H. Culkowski
SP 004	Postmortem punch with zero suppression	T. W. Hildebrandt
SP 005	Item check code	D. B. Grimes
SP 008	Paper-tape assembly	C. D. Griffies
SP 009	Reference check code	C. L. Gerberich
SP 00A	Alphameric hunk preparer	A. C. Downing
SP 00B	Output changer	A. A. Grau
SP 00C	Input-output check for magnetic tape	A. H. Culkowski
SP 00D	Reconstruct compiler items	N. B. Alexander
SP 00E	Reference to address, typewriter output	A. A. Grau
SP 102	Correction code for magnetic tape	A. H. Culkowski
SP 103	Compiled program edit	J. W. Reynolds
SP 104	Code edit routine	E. C. Long
SP 105	Tracer	B. J. Osborne
SP 106	(8,32) curve plotting routine	M. P. Lietzke
SP 107	Servi-seer	J. G. Sullivan
SP 108	Code editor	A. A. Grau

TRAINING

NUMERICAL ANALYSIS

A. S. Householder   A. A. Grau   N. M. Dismuke   K. Fan

Two series of lectures on topics in numerical analysis were given during 1958:

Series I. *The Approximation of Functions*

Polynomial Interpolation

1. Lagrange Polynomials; Divided Differences
2. Methods of Aitken and Neville; Several Variables
3. Inverse Interpolation; Remainders and Errors
4. Confluent Formulas
5. Optimum Interval Interpolation
6. Equal Interval Interpolation

Quadrature and Cubature; Numerical Differentiation

7. The Classical Formulas. I
8. Operator Methods. I
9. Operator Methods. II

10. The Classical Formulas. II
11. Gaussian and Related Formulas
12. Improper Integrals

Least Square Curve Fitting

13. Least Square Polynomials; Smoothing
14. Methods of Deming and Garwood

More General Methods

15. Interpolation with Arbitrary Functions
16. A General Remainder Theorem. I
17. A General Remainder Theorem. II

Continued Fractions

18. Elementary Properties

Series II. *Theory of Matrices and Determinants*

1. Determinants and Linear Dependence. I
2. Determinants and Linear Dependence. II
3. Characteristic Values and Vectors. I
4. Characteristic Values and Vectors. II
5. Jordan Canonical Form. I
6. Jordan Canonical Form. II
7. Givens Method for Eigenvalues and Vectors. I
8. Givens Method for Eigenvalues and Vectors. II
9. Methods for Solving Systems of Linear Equations
10. Theory of Systems of Linear Inequalities

STATISTICS

The following topics were presented at statistics section seminars during the first half of 1958:

D. A. Gardiner

Some Statistical Aspects of Life Testing

Elements of Accident Statistics

Life Testing from Exponential Populations

A New Look at Statistical Hypotheses

M. A. Kastenbaum

The "Aberrant" Observation in an Otherwise Valid Set of Data

Least Squares Estimation Using Order Statistics

Inverse, Multiple, and Sequential Sample Censuses

A. W. Kimball

Poisson Processes

Problem of Interval Estimation and Fiducial Distributions

The Estimation of Extraneous Variation in Binomial Populations

Estimates with Prescribed Variance Based on Two-Stage Sampling

H. L. Lucas, North Carolina State College

Experimental Design and Analysis with Nonlinear Models

M. J. R. Healy, Rothamsted Experimental Station, Harpenden, England

Experience with the Elliot 401 Computer at Rothamsted

N. F. Gjeddebaek, A/S Ferrosan, Copenhagen, Denmark

Efficiency of Grouping of Observations

CODING FOR DIGITAL COMPUTERS

E. C. Long

Three basic Oracle coding courses were presented to a total of 82 participants. One basic coding course for the IBM-704 stressing business application was presented. Thirty people from the three Oak Ridge Carbide plants attended this course.

ORACLE MANUAL FOR PROGRAMMERS

S. E. Atta

An introductory manual[1] to the Oracle for beginning programmers was issued. It includes many details, as well as simple examples to illustrate various points in principles and techniques. The material covered is divided into four chapters:

I. Introduction

II. A Description of the Oracle
   - A. General Characteristics
   - B. Description of the Seven Units

III. The Number Systems
   - A. Binary Numbers
   - B. Conversion of Decimal Numbers to Binary Numbers
   - C. Conversion of Binary Numbers to Decimal Numbers
   - D. Binary Number Representation in the Oracle
   - E. Negative Numbers
   - F. Hexadecimal Numbers

---

[1] S. E. Atta, *Oracle Manual for Programmers*, ORNL CF-57-6-68, 2d ed., rev. (Jan. 22, 1958).

IV. Basic Operations of the Oracle
   A. Instruction Word
   B. Arithmetical Operations of the Oracle
   C. Logical Operations of the Oracle
   D. Input-Output Operations
   E. Approximate Speeds of the Operations
   F. Summary of Operations

ORACLE APPLICATIONS PROGRAM

C. T. Fike

The following universities obtained Oracle time through the Oracle Applications Program, a joint project of the Laboratory and ORINS: The University of Tennessee, The University of Texas, Louisiana State University, Florida State University, Duke University, The University of Arkansas, Vanderbilt University, West Virginia University, Virginia Polytechnic Institute, Brooklyn Polytechnic Institute, and Michigan State University. Most of this work originated in the chemistry and physics departments of these universities, and occasionally in their psychology and statistics departments.

Recent increases in the number of computing machines to be found in universities have made the need for assistance of this type less urgent than before, and the program has been suspended since September 1, 1958.

A brief description of the various computing problems undertaken for these universities is given below:

1. Approximate solution of the generalized eigenvalue problem $Ax = \lambda Bx$, where each of A and B is a symmetric matrix and B is positive definite. Some of the codes written for this problem have become part of the Oracle library.

2. Approximate solution of two boundary value problems describing the phenomenon of adsorption on a plane electrode and on a dropping mercury electrode. A detailed description of this study has been published.[2]

3. Statistical factor analysis according to a method developed by E. E. Cureton of The University of Tennessee. The codes developed for this purpose make a principal axes decomposition of a modified correlation matrix and are of considerable flexibility. They are currently being applied to a problem originating in the Laboratory and have been used several times for university problems.

4. Tabulation of a function $F(t;a,b,c)$ defined by

$$F(t;a,b,c) = a \int_0^t \phi(t-\theta) \int_0^\infty \cos\,(b\phi\,\sqrt{\theta} - \psi)\, e^{-\phi^2 - c\phi\sqrt{\theta}}\, d\phi\, d\theta \quad ,$$

[2]P. Delahay and C. T. Fike, *J. Am. Chem. Soc.* 80, 2628–30 (1958).

which occurs as the solution of a boundary value problem arising in the study of heat transfer in radioactive waste.

5. Tabulation of elementary particle reactions for a prescribed set of elementary particles. A reaction was "possible" if certain quantum numbers were conserved. For these reactions, a table of threshold energies was also computed according to classical formulas. Results of these calculations are being incorporated in a master's thesis being written at Vanderbilt University.

6. Extension of the tables of the energy levels of an asymmetric rotor begun by King, Hainer, and Cross.[3] This required the computation of the eigenvalues of certain triple diagonal matrices;[3] the Sturm sequence method of Givens[4] was used for this purpose. Further extensions of the tables for the purpose of publication are being considered.

7. Inversion of an 81×81 symmetric matrix by the method of Givens.

8. Computational study of the distribution of eigenvalues of the matrix

$$\begin{pmatrix} 1 - \cos x \cos y \cos z & \mu \sin x \sin y \cos z & \mu \sin x \cos y \sin z \\ \mu \sin x \sin y \cos z & 1 - \cos x \cos y \cos z & \mu \cos x \sin y \sin z \\ \mu \sin x \cos y \sin z & \mu \cos x \sin y \sin z & 1 - \cos x \cos y \cos z \end{pmatrix},$$

where $0 \leqq \mu \leqq 1$, $0 \leqq x, y, z \leqq \pi$. A similar study of this type made on the AVIDAC has been described.[5] Results of this study are being incorporated in a dissertation being written at The University of North Carolina.

9. Least-squares refinement of the crystal structure parameters of quartz based on x-ray data. The codes used for this purpose were written for the Oracle by H. A. Levy and W. R. Busing of the Chemistry Division.

10. Study of problems relating to the design of a cyclotron magnet. Particle orbits in measured magnet fields, focusing properties of a magnet, and corrections to magnet structure were computed. The codes used for this purpose were written for work originating within the Laboratory.

---

[3]G. W. King, R. M. Hainer, and P. C. Cross, *J. Chem. Phys.* 11, 27–42 (1943).

[4]W. Givens, *Numerical Computation of the Characteristic Values of a Real Symmetric Matrix*, ORNL-1574 (Feb. 19, 1954).

[5]C. B. Walker, *Phys. Rev.* 103, 547–57 (1956).

PUBLICATIONS

C. W. Sheppard, M. Slater, and G. J. Atta, "A Theory for Heterogeneous Ion Chambers and Neutron Response of Victoreen Dosimeters," *Radiation Research* **7**, 342–56 (1957).

S. E. Atta, *Oracle Manual for Programmers*, ORNL CF-57-6-68, 2d ed., rev. (Jan. 22, 1958).

K. Fan and J. J. Andrews, *On Kofink's Approximate Solution of the Neutron Transport Equation*, ORNL-2485 (March 19, 1958).

K. Fan (with P. Davis), "Complete Sequences and Approximations in Normed Linear Spaces," *Duke Math. J.* **24**, 183–92 (1957).

K. Fan (with G. Pall), "Imbedding Conditions for Hermitian and Normal Matrices," *Can. J. Math.* **9**, 298–304 (1957).

K. Fan (with I. Glicksberg and A. J. Hoffman), "Systems of Inequalities Involving Convex Functions," *Am. Math. Soc. Proc.* **8**, 617–22 (1957).

K. Fan, "Existence Theorems and Extreme Solutions for Inequalities Concerning Convex Functions or Linear Transformations," *Math. Z.* **68**, 205–16 (1957).

K. Fan, "Topological Proofs for Certain Theorems on Matrices with Nonnegative Elements," *Monatsh. Math.* **62**, 219–37 (1958).

K. Fan, "Note on Circular Disks Containing the Eigenvalues of a Matrix," *Duke Math. J.* **25**, 441–45 (1958).

K. Fan (with I. Glicksberg), "Some Geometric Properties of the Spheres in a Normed Linear Space, "*Duke Math. J.* **25**, 553–68 (1958).

K. Fan, "On the Equilibrium Value of a System of Convex and Concave Functions," *Math. Z.* **70**, 271–80 (1958).

A. S. Householder, "Numerical Mathematics from the Viewpoint of Electronic Digital Computers," *Proceedings of the Darmstadt Conference*, abstracted in *Computers and Automation* **5**(4), 14–19 (1956).

A. S. Householder, "Glossaries and Malapropisms," *Computers and Automation* **6**(4), 34–35 (1957).

A. S. Householder, "A Survey of Some Closed Methods for Inverting Matrices," *J. Soc. Ind. Appl. Math.* **3**, 155–69 (1957).

A. C. Downing, A. A. Grau, E. C. Long, J. Harrison, and B. J. Osborne (with M. E. LaVerne, R. R. Bate, R. R. Coveyou, and J. G. Sullivan), *Programmers' Manual for the Oracle Compiler*, ORNL CF-57-8-92 (April 30, 1958).

A. C. Downing, "The Utilization of Digital Computers in Chemical, Physical and Metallurgical Research," p 24, Proceedings of Computer Applications Symposium, UCC Engineering Liaison Meeting, New York, October 14, 1957.

A. S. Householder (with W. C. Sangren), Sec III, Mathematics, p 3-1–3-148, *Nuclear Engineering Handbook*, McGraw-Hill, New York, 1958.

A. S. Householder, "A Class of Methods for Inverting Matrices," *J. Soc. Ind. Appl. Math.* **6**(2), 189–95 (1958).

A. S. Householder, "On Matrices with Nonnegative Elements," *Monatsh. Math.* **62**, 238–42 (1958).

A. S. Householder, "The Approximate Solution of Matrix Problems," *J. Assoc. Computing Mach.* **5**(3), 205–43 (1958).

M. A. Kastenbaum, "Estimation of Relative Frequencies of Four Sperm Types in *Drosophila melanogaster*," *Biometrics* **14**, 223–28 (1958).

L. Sandler and M. A. Kastenbaum, "A Note on the Frequency Distribution of Tetrads by Rank in *Drosophila melanogaster*," *Genetics* **43**(2), 215–22 (1958).

A. W. Kimball (with W. T. Burnett, Jr., and D. G. Doherty), "Chemical Protection Against Ionizing Radiation. I. Sampling Methods for Screening Compounds in Radiation Protection Studies with Mice," *Radiation Research* **7**, 1–12 (1957).

A. W. Kimball, "Errors of the Third Kind in Statistical Consulting," *J. Am. Stat. Assoc.* **52**, 133–42 (1957).

C. W. Sheppard, M. Slater, E. B. Darden, A. W. Kimball, G. J. Atta, C. W. Edington, and W. K. Baker, "Biological Effects of Fast Neutrons from an Internal Target Cyclotron: Physical Methods and Dominant Lethals in Drosophila," *Radiation Research* **6**, 173–87 (1937).

A. W. Kimball, "Evaluation of Data Relating Human Leukemia and Ionizing Radiation," *J. Natl. Cancer Inst.* **21**, 383–91 (1958).

A. C. Upton, F. F. Wolff, J. Furth, and A. W. Kimball, "A Comparison of the Induction of Myeloid and Lymphoid Leukemias in X-Irradiated RF Mice," *Cancer Research* **18**, 842–48 (1958).

LECTURES

J. J. Andrews, "Light Open Mappings in Euclidean Spaces," University of Tennessee, Knoxville, January 1958.

A. C. Downing, Panel Discussion on Matrices, Conference on Matrix Computations, Wayne University, Detroit, September 1957.

A. C. Downing, "The Utilization of Digital Computers in Chemical, Physical, and Metallurgical Research," Union Carbide Engineering Liaison Meeting, New York, October 1957.

A. C. Downing, "The Design of Fixed-Point Iterations," National Meeting of the Association for Computing Machinery, University of Illinois, Urbana, June 1958.

K. Fan, "Combinatorial Proofs of Some Topological Properties of the Spheres" (two lectures); "Systems of Linear Inequalities" (three lectures), University of Idaho, Moscow, July 1957.

K. Fan, "A New Closure Property in Banach Spaces and Its Application to Uniqueness Theorems and Approximations for Analytic Functions," Conference on Analytic Functions, Institute for Advanced Study, Princeton, September 1957.

K. Fan, "Some Geometric Properties of the Spheres in a Normed Linear Space," University of Tennessee, Knoxville, October 1957; University of Kentucky, Lexington, April 1958.

K. Fan, "Minimax Theorems and Systems of Linear Functional Inequalities," University of South Carolina, Columbia, January 1958.

K. Fan, "Inequalities for Eigenvalues," Alabama Polytechnic Institute, Auburn, February 1958.

K. Fan, "Topological Groups" (course), University of Tennessee, Knoxville, 1957–1958.

K. Fan, "Some Geometric Properties of the Spheres in Normed Linear Spaces"; "Combinatorial Proofs of Some Antipodal Point Theorems of Spheres," Florida State University, Tallahassee, April 1958; University of Florida, Gainesville, April 1958.

K. Fan, "Elementary Proofs of Some Topological Properties of the Spheres," University of Tennessee, Knoxville, May 1958.

D. A. Gardiner, "Experimental Designs for Exploring Response Surfaces," Milwaukee Section, American Chemical Society, Milwaukee, March 1957.

D. A. Gardiner, "Some Third Order Rotatable Designs," Mountain Lake Conference on Statistics, Mountain Lake, Virginia, August 1957.

D. A. Gardiner, "Experimental Designs for Industrial Research," Chemical Engineering Department Seminar, University of Tennessee, Knoxville, January 1958.

D. A. Gardiner, "Multiple Regression"; "Response Surface Methodology" (two lectures), Short Course in Statistical Methods for Research Workers in Industry and the Physical Sciences, North Carolina State College, Brevard, June 1958.

A. A. Grau, "Mathematical Systems Involving Ternary Operations Related to Boolean Algebras and Lattices," University of Tennessee, Knoxville, May 1957.

A. S. Householder, "A Survey of Some Closed Methods for Inverting Matrices" (two lectures), University of Wisconsin, Madison, March 1957; Boulder Chapter of Society of Industrial and Applied Mathematics, Boulder, Colorado, July 1957; Conference on Matrix Computations, Wayne University, Detroit, September 1957.

A. S. Householder, "Truncation Error in the Solution of Partial Differential Equations," Association for Computing Machinery, Houston, June 1957.

A. S. Householder, "The Approximate Solution of Matrix Problems" (ten lectures), University of Michigan, Ann Arbor, August 1957; two lectures, Argonne National Laboratory, Lemont, Illinois, April 1958.

A. S. Householder, "A Class of Methods for Inverting Matrices," Argonne National Laboratory, Lemont, Illinois, January 1958; American Mathematical Society, Cincinnati, January 1958.

A. S. Householder, "Norms of Vectors and Matrices," University of Tennessee, Knoxville, February 1958.

A. S. Householder, "Some Observations on the Method of Conjugate Gradients," Southeastern Section, Mathematical Association of America, Gainesville, Florida, March 1958.

A. S. Householder, "Minimal Matrix Norms," American Mathematical Society, Chicago, April 1958.

A. S. Householder, "Generated Error in Rotational Tridiagonalization," Association for Computing Machinery, University of Illinois, Urbana, June 1958.

A. S. Householder, "Convergence and Error in the Solution of Linear Systems," University of Michigan, Ann Arbor, June 1958.

A. S. Householder, "Computational Method in Linear Algebra: Survey and Critique," IBM Corporation, Endicott, New York, July 1958.

A. S. Householder, "Some Mathematical Problems in Matrix Computation" (three lectures), 7th Series of Earl Raymond Hedrick Lectures, Mathematical Association of America, Massachusetts Institute of Technology, Cambridge, August 1958.

M. A. Kastenbaum, "The Hypothesis of No-Interaction in Multiway Contingency Tables," Mountain Lake Conference on Statistics, Mountain Lake, Virginia, August 1957.

M. A. Kastenbaum, "Estimation of Sperm Frequencies in Drosophila," Joint Meeting of the Biometric Society (Eastern North American Region) and the Institute of Mathematical Statistics, Gatlinburg, Tennessee, April 1958.

A. W. Kimball, "Disease Incidence Estimation in Populations Subject to Multiple Causes of Death," 30th Session of the International Statistical Institute, Stockholm, Sweden, August 1957.

ORNL-2652
Physics and Mathematics
TID-4500 (14th ed.)

INTERNAL DISTRIBUTION

1. C. E. Center
2. Biology Library
3. Health Physics Library
4. Reactor Experimental Engineering Library
5-7. Central Research Library
8-14. Laboratory Records Department
15. Laboratory Records, ORNL RC
16. A. M. Weinberg
17. L. B. Emlet (K-25)
18. J. P. Murray (Y-12)
19. J. A. Swartout
20. R. R. Dickison
21. E. D. Shipley
22. M. L. Nelson
23. A. S. Householder
24. A. H. Snell
25. E. H. Taylor
26. A. Hollaender
27. W. H. Jordan
28. H. E. Seagren
29. M. T. Kelley
30. K. Z. Morgan
31. D. Phillips
32. C. E. Winters
33. E. O. Wollan
34. E. P. Blizard
35. R. S. Livingston
36. C. P. Keim
37. J. W. Givens
38. C. D. Susano
39. G. C. Williams
40. R. A. Charpie
41. J. Harrison
42. N. M. Dismuke
43. A. H. Culkowski
44. A. W. Kimball
45. M. R. Arnette
46. P. M. Reyling
47. A. C. Downing, Jr.
48. J. H. Vander Sluis
49. G. J. Atta
50. C. L. Gerberich
51. J. A. Lane
52. R. W. Johnson
53. M. J. Skinner
54. J. L. Gabbard
55. E. C. Long
56. M. A. Kastenbaum
57. J. P. Kelly (K-25)
58. S. E. Atta
59. G. W. Medlin
60. F. J. Witt
61. D. A. Gardiner
62. C. A. Preskitt
63. ORNL – Y-12 Technical Library, Document Reference Section

EXTERNAL DISTRIBUTION

64. A.E.S. Green, Florida State University
65. Division of Research and Development, AEC, ORO
66-676. Given distribution as shown in TID-4500 (14th ed.) under Physics and Mathematics category (75 copies – OTS)

Mathematics Panel progress reports previously issued in this series are as follows:

ORNL-345	December, January, February, 1948–1949
ORNL-408	Period Ending July 31, 1949
ORNL-516	Period Ending October 31, 1949
ORNL-634	Period Ending January 31, 1950
ORNL-726	Period Ending April 30, 1950
ORNL-818	Period Ending July 31, 1950
ORNL-888	Period Ending October 31, 1950
ORNL-979	Period Ending January 31, 1951
ORNL-1029	Period Ending April 30, 1951
ORNL-1091	Period Ending July 31, 1951
ORNL-1151	Period Ending October 31, 1951
ORNL-1232	Period Ending January 31, 1952
ORNL-1290	Period Ending April 30, 1952
ORNL-1360	Period Ending July 31, 1952
ORNL-1435	Period Ending October 31, 1952
ORNL-1498	Period Ending January 31, 1953
ORNL-1588	Period Ending June 30, 1953
ORNL-1662	Period Ending December 31, 1953
ORNL-1751	Period Ending June 30, 1954
ORNL-1842	Period Ending December 31, 1954
ORNL-1928	Period Ending June 30, 1955
ORNL-2037	Period Ending December 31, 1955
ORNL-2134	Period Ending June 30, 1956
ORNL-2283	Period Ending February 28, 1957