This amounts to correcting n_2' by the term

$$-\frac{\epsilon_3 M^2}{n_3}.$$

Obviously if l_2' had been corrected instead of n_2' the correction would be

$$-\frac{\epsilon_3 M^2}{l_3}.$$

To make the correction small, the largest of l_3, m_3, n_3 is made to appear in the denominator. This also prevents division by zero, which might occur if n_2 were corrected under all circumstances.

After the direction of r_2 has been established perpendicular to r_3, its magnitude must be adjusted. The procedure is to alter all components proportionately as for r_3 above. Clearly this does not destroy orthogonality. l_1, m_1, n_1 can now be computed from equation (3)

$$l_1 = \frac{m_2 n_3 - m_3 n_2}{M}, \text{ etc.}$$

6. Conclusions. The equations of motion of a rotating rigid body can be formulated directly in terms of the direction cosines. The direction cosines can be determined as described above in a manner which is thought to be ideally suited to digital computation. No trigonometric functions of the Eulerian angles appear, obviating the necessity of referring to a function table or to a series expansion to obtain the solutions of the equations of motion. In addition, all quantities are finite for all orientations of the rigid body.

This work was done in connection with a study performed by the University of Pennsylvania, Moore School of Electrical Engineering, under Contract N6Onr 24913 sponsored by the Office of Naval Research, Special Devices Center, Port Washington, New York.

H. J. Gray, Jr.
M. Rubinoff
H. Sohon

University of Pennsylvania
Moore School of Electrical Engineering

[1] Leigh Page, *Introduction to Theoretical Physics*, Second Edition—Fourteenth Printing, D. Van Nostrand Company, Inc., New York, N. Y., Introduction, Chap. I, II; 1951.

A Solution of Simultaneous Linear Equations and Matrix Inversion with High Speed Computing Devices

In solving several systems of simultaneous linear equations, in which the coefficients of the unknowns are the same in all the systems but in which the constant terms vary from system to system, it is advantageous to invert the matrix formed by the coefficients of the unknowns. This inverted matrix, common to all the systems, is then multiplied by the vector formed by each set of constant terms, and a set of solutions is thus obtained for each of the above systems,[1] or simultaneous solutions of the systems obtained at the time of inversion. The inversion of a matrix, especially if it is of a high order, is a laborious and time-consuming process. From time to time new methods of inversion have been developed and modified to suit the computation equipment available until it seems that a saturation point in simplicity of application, accuracy of results, and speed of computation has been reached, and further improvements will have to come from the development of the computing machines themselves or from their application. For information on these topics the reader is referred to items 2–12 of the bibliography.

The following is a modification of an algebraic treatment of the matrix by the direct method which will reduce the labor and time necessary for inversion. This method has time and labor saving characteristics applicable to computations with low and high memory computing devices, making the inversion of high order matrices less forbidding.

In addition to the customary definitions of inversion, pivotal row, leading element, etc., which are here taken for granted, we will define prepivotal and transpivotal rows as the rows respectively preceding and following the pivotal row. Consider also the original matrix as extended to the right by means of the identity matrix and a check column (other augmentations of the original matrix may be included to suit the needed results[6] and a check row may be added for ease in locating the source of spurious results).

In a simplified method[11] of inversion the elements of this extended matrix are operated upon by certain reduction operations and an intermediate matrix is obtained, the elements of which, on a second application of the reduction formulas, yield another intermediate matrix, and so on until, after the nth reduction, the inverted matrix is obtained. The reduction operations usually consist of a quotient operation applied to the elements of the pivotal row and a binomial operation applied to the elements of the other rows. This reduction cycle has a definite advantage in its simplicity, but it involves computations with the elements of the transpivotal rows in all the previous reductions, in preparation of values for the time when these transpivotal rows become pivotal or prepivotal. Also, with automatic computing machines, all the elements of the identity matrix are involved in the calculations even at the time when some of them do not contribute to the final result.

The method proposed in this article computes partial reductions (on some of the rows only) on pivotal, prepivotal, and the first transpivotal rows, until the memory capacity of the computing equipment is reached, at which point a complete reduction (on all the rows of the matrix) is computed, repeating this cycle, with the last obtained matrix, until final reduction (inversion) is reached; all this while, at a given time only the relevant contributing part of the identity matrix is used. It is to be noted that the saving in the identity matrix is independent of the transpivotal row economy and may be used with other inversion methods as well.

Three types of computing operations are used in each matrix reduction.

1. Divisional or quotient operation, used on the elements of the pivotal row.
2. Binomial operation, used on the elements of the prepivotal rows.
3. Polynomial operation, used on the elements of the first transpivotal row for partial matrix reductions, and on all the transpivotal rows for complete matrix reductions.

Let the elements of the original extended matrix be x_{ij} (where i and j respectively represent the row and column of the particular element, i may assume values from 1 to n, and j values from 1 to $2n + 1$). Let the elements of the matrix developed after the first, second . . . and nth reductions be $a_{ij}, b_{ij} \cdots n_{ij}$, respectively. As usual the ith row is pivotal for the ith reduction. The formulas for the first five reductions are:

$$(1)\quad \begin{aligned} a_{1j} &= x_{1j}/x_{11} \\ a_{2j} &= x_{2j} - x_{21}a_{1j} \end{aligned} \qquad (j = 2(1)n + 1)$$

$$(2)\quad \begin{aligned} b_{1j} &= a_{1j} - a_{12}b_{2j} \\ b_{2j} &= a_{2j}/a_{22} \\ b_{3j} &= x_{3j} - x_{32}b_{2j} - x_{31}b_{1j} \end{aligned} \qquad (j = 3(1)n + 2)$$

$$(3)\quad \begin{aligned} c_{1j} &= b_{1j} - b_{13}c_{3j} \\ c_{2j} &= b_{2j} - b_{23}c_{3j} \\ c_{3j} &= b_{3j}/b_{33} \\ c_{4j} &= x_{4j} - x_{43}c_{3j} - x_{42}c_{2j} - x_{41}c_{1j} \end{aligned} \qquad (j = 4(1)n + 3)$$

$$(4)\quad \begin{aligned} d_{1j} &= c_{1j} - c_{14}d_{4j} \\ d_{2j} &= c_{2j} - c_{24}d_{4j} \\ d_{3j} &= c_{3j} - c_{34}d_{4j} \\ d_{4j} &= c_{4j}/c_{44} \\ d_{5j} &= x_{5j} - x_{54}d_{4j} - x_{53}d_{3j} - x_{52}d_{2j} - x_{51}d_{1j} \end{aligned} \qquad (j = 5(1)n + 4)$$

$$(5)\quad \begin{aligned} e_{1j} &= d_{1j} - d_{15}e_{5j} \\ e_{2j} &= d_{2j} - d_{25}e_{5j} \\ e_{3j} &= d_{3j} - d_{35}e_{5j} \\ e_{4j} &= d_{4j} - d_{45}e_{5j} \\ e_{5j} &= d_{5j}/d_{55} \\ e_{6j} &= x_{6j} - x_{65}e_{5j} - x_{64}e_{4j} - x_{63}e_{3j} - x_{62}e_{2j} - x_{61}e_{1j} \end{aligned} \qquad (j = 6(1)n + 5)$$

The check column should be included in the above values of j.

Thus, on the first reduction, the pivotal row is the first row, the elements of which are divided and the binomial operation is applied to the elements of the second row. On the second reduction, the second, first, and third rows are pivotal, prepivotal, and the first transpivotal, respectively. In the third reduction, the third row is pivotal, the first and the second rows are prepivotal, and the fourth row is the first transpivotal, etc.

With an unlimited, or sufficiently large, memory machine the process is continued until the nth reduction, when the nth row is pivotal, the rows less than n are prepivotal and there are no transpivotal rows. In applying the above formulas to the elements of the original (extended) and intermediate matrices, only columns of the identity matrix up to and including the unit element of the corresponding pivotal row need be included.

With limited memory machines the process is continued until the memory capacity is reached, at which point the quotient operation is applied to the elements of the pivotal row, the binomial operation is used on the elements of the prepivotal rows and the polynomial operation is computed for the elements of all the transpivotal rows. Thus, if in the above equations the fourth reduction had been complete, the general equation for the transpivotal row elements would have been:

$$d_{ij} = x_{ij} - x_{i4}d_{4j} - x_{i3}d_{3j} - x_{i2}d_{2j} - x_{i1}d_{1j} \qquad (i = 5(1)n)$$

With a minimum memory machine all the reductions are computed as usual with a quotient operation on the elements of the pivotal row and a binomial operation on the elements of all the other rows. Unlimited, limited, and minimum memory machines are left undefined except in their use to classify the three broad applications of the method as described above.

The polynomial expression for the transpivotal rows, which is easily obtained algebraically from previous binomial reductions by a rearrangement of terms and by a substitution of pivotal and prepivotal results of the same column, permits partial reductions of the matrix due to the fact that it depends only on the pivotal and prepivotal results of the same column and on the leading elements of the original extended matrix for the corresponding row.

It is to be noticed that in the above formulas we must distinguish between two types of leading elements: (a) derived leading elements for pivotal and prepivotal rows and taken from the previous matrix reduction, and (b) original leading elements for transpivotal rows and taken from the original extended matrix (or from an intermediate matrix obtained in a previous complete reduction). For a given transpivotal row, there are as many original leading elements as the number of the reduction. Varying degrees of continuity within the cycle itself may be obtained according to the storing capacity, programming facilities, available means for selection of properly coded rows and columns, and other characteristics of the available equipment.

Between the two extremes of no memory and unlimited memory there is an optimum program design which is approached by a judicious choice of continuity of operation using the above minimum memory requirements to meet the available memory capacity.

We have successfully and economically applied the above method to the inversion of 20 by 20 matrices using a full capacity 602-A.IBM calculating punch while using words of 12 digits throughout in order to take care of creeping decimal errors from the right and overflowing on the left. Three partial and one complete reductions per cycle were obtained.

With reduced numbers, when possible, greater savings may be achieved, but they are not advisable when handling high order matrices. Even when large integral values may be avoided by preliminary or intermediate treatment of the matrix, the creeping decimal error is ever present and may be met only with otherwise unnecessary decimals or with iterative methods.[8]

The time consumed by this method and the time by the method of all complete reductions were in the ratio of 1 to 3 when using the same equipment in both cases and with the same number handicap, by the same operators.

GABRIEL G. BEJARANO
BRUCE R. ROSENBLATT

California Research Corporation
Richmond, California

[1] A. C. AITKEN, *Determinants and Matrices*. Edinburgh, 5th ed., 1948.

[2] P. S. DWYER, *Linear Computations*. New York, 1951.

[3] H. HOTELLING, "Some new methods in matrix calculation," *Annals Math. Stat.*, v. 14, 1943, p. 1–34.

[4] P. D. CROUT, "A short method for evaluating determinants and solving systems of linear equations with real or complex coefficients," AIEE, *Trans.*, v. 60, 1941, p. 1235–1241.

[5] G. W. PETRIE III, "Matrix inversion and solution of simultaneous linear algebraic equations with the IBM type 604 Electronic Calculating Punch," IBM Computation Seminar, August 1951, *Proc.*, p. 105–111.

[6] J. CHANCELLOR, J. W. SHELDON, & G. L. TATUM, "The solution of simultaneous linear equations using the IBM Card-Programmed Electronic Calculator," IBM Industrial Computation Seminar, September 1950, *Proc.*, p. 57–61.

[7] W. D. BELL, "Punched card techniques for the solution of simultaneous equations and other matrix operations," IBM Scientific Computation Forum, 1948, *Proc.*, p. 28–31.

[8] J. VON NEUMANN & H. H. GOLDSTINE, "Numerical inverting of matrices of high order," Amer. Math. Soc., *Bull.*, v. 53, 1947, p. 1021–1099.

[9] A. OPLER, "Monte Carlo matrix calculation with punched card machines," *MTAC*, v. 5, 1951, p. 115–120.

[10] J. SHERMAN, "Computations of inverse matrices by means of IBM machines," IBM *Technical Newsletter*, no. 3, 1951, p. 22–25.

[11] J. LOWE, "Solution of simultaneous linear algebraic equations using the IBM type 604 Electronic Calculating Punch," IBM Computation Seminar, Dec. 1949, *Proc.*, p. 54–56.

[12] F. M. VERZUH, "The solution of simultaneous linear equations with the aid of the 602 Calculating Punch," *MTAC*, v. 3, p. 453–462.

Error Bounds on Approximate Solutions to Systems of Linear Algebraic Equations

1. The Abmatrix. Consider the real matrix $A = (a_{ij})$, $i = 1(1)m$, $j = 1(1)n$. Define $\alpha(A)$, read "the abmatrix of A," to be

$$\alpha(A) = (|a_{ij}|).$$

Thus, $\alpha(A)$ is the *matrix* having each element equal to the absolute value of the corresponding element of A. In particular, for a scalar s,

$$\alpha(s) = |s|.$$

If $A = (a_{ij})$ and $B = (b_{ij})$, $i = 1(1)m$, $j = 1(1)n$, define

$$\alpha(A) \leqslant \alpha(B)$$

to mean

$$|a_{ij}| \leqslant |b_{ij}| \qquad \text{for all } i, j.$$

Thus, if $\alpha(A) \leqslant \alpha(B)$ and $\alpha(B) \leqslant \alpha(C)$, then $\alpha(A) \leqslant \alpha(C)$.