

$P_{N-1}$  of the type

$$(k + 1)^0, \quad (P_{\text{even}}(1, 2, \dots, k)), \quad (k + 2)^1, \quad (k + 3)^1, \dots, (n)^1.$$

Thus  $b_N(k) = 1$  if  $k$  is odd and  $I_{N-1}(k + 1) = 1$ , i.e.,  $I_N(k + 1) = 0$ . By (8), we have (6).

Finally, for even  $k < n - 1$  and  $N$  an odd multiple of  $(k + 1)(k + 2) \cdots n$ , the mark  $k + 1$  will have made an odd number of passages through the smaller marks and will be at the left of an even permutation of the marks from 1 to  $k$ . Thus the permutation of the marks from 1 to  $k + 1$  is also even since  $k$  is even and so  $I_{N-1}(k + 2) = 0$  with  $k + 2$  at the extreme left. The permutation on the marks  $1, 2, \dots, k + 2$  is then odd, so that  $k + 3$  is at the right of these marks with  $I_{N-1}(k + 3) = 1$ , as are the marks larger than  $k + 3$ . The permutation  $P_{N-1}$  is of the type

$$(k + 2)^0, \quad (k + 1)^0, \quad (P_{\text{even}}(1, 2, \dots, k)), \quad (k + 3)^1, \quad (k + 4)^1, \dots, n^1.$$

Since  $I_{N-1}(k + 1) = 0$ ,  $I_N(k + 1) = 1$ , or by (7) we see that  $b_N(k) = 2$  for even  $k < n - 1$  and  $d_k$  odd as (5) states.

The following table illustrates the technique as applied to the generation of 24 permutations on 4 marks, reading downward rather than left to right:

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 4 | 4 | 1 | 1 | 1 | 3 | 3 | 4 | 4 | 3 | 3 | 2 | 2 | 2 | 4 | 4 | 2 | 2 |   |   |   |   |
| 2 | 2 | 4 | 1 | 1 | 4 | 3 | 3 | 1 | 1 | 4 | 3 | 3 | 4 | 2 | 2 | 3 | 3 | 4 | 2 | 2 | 4 | 1 | 1 |
| 3 | 4 | 2 | 2 | 3 | 3 | 4 | 2 | 2 | 4 | 1 | 1 | 2 | 2 | 4 | 1 | 1 | 4 | 3 | 3 | 1 | 1 | 4 | 3 |
| 4 | 3 | 3 | 3 | 2 | 2 | 2 | 4 | 4 | 2 | 2 | 1 | 1 | 1 | 4 | 4 | 1 | 1 | 1 | 3 | 3 | 3 | 4 | 4 |

The RAND Corporation  
Santa Monica, California

1. D. H. LEHMER, "Generation of permutations by transposition," *Math. Reviews*, v. 23, 1962, no. 1B, B553, p. 94.  
2. MARK B. WELLS, "Generation of permutations by transposition," *Math. Comp.*, v. 15, 1961, p. 192-195.

## Determinants by Means of Solutions of Linear Equations

By R. W. Preisendorfer, B. W. Roos & W. C. Sangren

**1. Introduction.** In this paper, we are considering the converse of the classical problem of "Solutions of Linear Equations by Means of Determinants." The present problem arises from the occasional need for the explicit value of a determinant associated with a system of equations that is being solved by a machine method which does not necessarily involve the classical Cramer method of solution using determinants. Suppose then that a particular program of computation for the solution of a system of linear equations does not explicitly evaluate the determinant of the matrix of the system, but that nevertheless the determinant for some reason is

---

Received August 17, 1961, revised October 29, 1962.

required. Furthermore, suppose that it is expedient at the moment to use, if possible, the actual program at hand to evaluate the required determinant, rather than to use one of several existing techniques specifically designed to perform this special task. The question then arises: Is there some way in which the determinant of a system of equations can be evaluated using an *arbitrary* method of solution of the system? Some reflection will show that the answer is yes. The details of the required analytic procedure spring into view if we phrase the question in the following analytic terms.

Let  $|A|$  be the determinant of an  $n \times n$  matrix  $A = (a_{ij})$ . Suppose that an associated system of linear equations is formed from  $A$ :

$$Ax = y,$$

where  $x = (x_1, \dots, x_n)$ ,  $y = (y_1, \dots, y_n)$ . Is there some way of choosing specific values of the components of  $y$  so that the resultant components of  $x$  can be used, via a simple algorithm, to obtain  $|A|$ ? The choice of  $y$  and the required algorithm turn out to be quite simple.

**2. The Algorithm.** Let  $A = (a_{ij})$  be a nonsingular  $n \times n$  matrix.

*Cycle 1, Step 1.* Form the system of  $n$  equations in  $n$  unknowns,

$$\sum_{j=1}^n a_{ij}x_j = y_i \quad i = 1, 2, \dots, n$$

*Cycle 1, Step 2.* Set  $y_1 = 1$ ,  $y_i = 0$ ,  $i = 2, \dots, n$ .

*Cycle 1, Step 3.* Solve the system for the resultant  $x_i$ 's and choose  $j_1$  so that  $x(n, j_1)$  is the first nonzero member of the solution  $x = (x_1, \dots, x_n)$ .

Cramer's rule states:

$$x(n, j_1) = \frac{(-1)^{j_1+1} |A(n-1, j_1)|}{|A|},$$

where  $A(n-1, j_1)$  is the matrix obtained by deleting the first row and the  $j_1$  column of  $A$ .

*Cycle 2, Step 1.* Repeat the first step, but apply it to the nonsingular  $(n-1) \times (n-1)$  matrix,  $A(n-1, j_1)$ .

*Cycle 2, Step 2.* Repeat the second step, but apply it to  $A(n-1, j_1)$ .

*Cycle 2, Step 3.* Solve the  $(n-1)$ -order system associated with  $A(n-1, j_1)$ . Let  $x(n-1, j_2)$  denote the first nonzero value of the solution set. Then

$$x(n-1, j_2) = \frac{(-1)^{j_2+1} |A(n-2, j_2)|}{|A(n-1, j_1)|}.$$

Here,  $A(n-2, j_2)$  denotes the  $(n-2) \times (n-2)$  matrix obtained by deleting the first row and  $j_2$  column of  $A(n-1, j_1)$ .

*Cycle 3.* Continue.

The result of the algorithm may be stated in the following theorem.

**THEOREM.** Let  $A = (a_{ij})$  be a nonsingular  $n \times n$  matrix, and let  $x(n-k, j_{k+1})$  be the first nonzero member of the solution of the  $(n-k)$ -order system obtained in the third step of the  $k$ th cycle of the algorithm. Then

$$|A| = \left[ \sum_{k=0}^{n-1} X(n - k, j_{k+1}) \right]^{-1}$$

where  $X(n - k, j_{k+1}) = (-1)^{j_{k+1}+1} x(n - k, j_{k+1})$ .

*Proof.* The third step in the  $k$ th cycle results in the value

$$x(n - k, j_{k+1}) = \frac{(-1)^{j_{k+1}+1} |A(n - k, j_{k+1})|}{|A(n - k, j_k)|}.$$

The next-to-last cycle in the method yields

$$x(2, j_{n-1}) = \frac{(-1)^{j_{n-1}+1} |A(1, j_{n-1})|}{|A(2, j_{n-2})|}.$$

The last cycle yields

$$x(1, j_n) = \frac{1}{|A(1, j_{n-1})|}.$$

If we now multiply the  $n$  solutions  $x(n - k, j_{k+1})$ , we obtain

$$\begin{aligned} \prod_{k=0}^{n-1} x(n - k, j_{k+1}) &= \frac{(-1)^{j_1+1} |A(n - 1, j_1)|}{|A|} \cdot \frac{(-1)^{j_2+1} |A(n - 2, j_2)|}{|A(n - 1, j_1)|} \\ &\quad \cdot \dots \cdot \frac{(-1)^{j_{n-1}+1} |A(1, j_{n-1})|}{|A(2, j_{n-2})|} \cdot \frac{1}{|A(1, j_{n-1})|} \\ &= \frac{(-1)^{\left[ \begin{smallmatrix} (n-1)+\sum_{k=1}^{n-1} j_k \end{smallmatrix} \right]}}{|A|} \end{aligned}$$

or

$$|A| = \frac{(-1)^{\left[ \begin{smallmatrix} (n-1)+\sum_{k=1}^{n-1} j_k \end{smallmatrix} \right]}}{\prod_{k=0}^{n-1} x(n - k, j_{k+1})} = \left[ \prod_{k=0}^{n-1} X(n - k, j_{k+1}) \right]^{-1}.$$

**3. Numerical Efficiency and Comparison with Other Methods.** In order to obtain some insight into the numerical economy of the algorithm, we summarize some well-known methods for the calculations of an  $n \times n$  determinant and compare the required number of operations for each case with the required number of operations for the method discussed.

**3.1. Expansion in Elements and Expansion in Minors [1].** The well-known method of the expansion in elements requires  $(n - 1)n!$  multiplications and  $n! - 1$  additions, while the expansion-in-minors method requires  $\sum_{j=1}^n n!/(n - j)!$  multiplications.

**3.2. Chio's Method [1].** In this method, the following steps are prescribed:

1. Divide an arbitrary row by one of its nonzero elements, e.g.,  $a_{ij}$ .
2. *a.* Multiply the  $i$ th row by  $a_{1j}$  and subtract this row from the first row.
- b.* Multiply the  $i$ th row by  $a_{2j}$  and subtract this row from the second row.
3. Etc.

3. Expand in minors according to the elements of the  $j$ th column. In order to complete the first step, we need to perform  $(n - 1)$  divisions,  $(n - 1)^2 + 1$  multiplications, and  $n^2$  additions. The second step requires  $(n - 2)$  divisions,  $(n - 2)^2 + 1$  multiplications, and  $(n - 1)^2$  additions, etc. In total, the following numbers of operations are required:

$$\begin{aligned} \sum_{j=1}^{n-1} (j^2 + 1) &= \frac{(n-1)(2n^2 - n + 1)}{6} && \text{multiplications} \\ \sum_{j=1}^{n-1} j &= \frac{n(n-1)}{2} && \text{divisions} \\ \sum_{j=1}^n j^2 &= \frac{n}{6} (n+1)(2n+1) && \text{additions.} \end{aligned}$$

**3.3. Diagonalization Method [2].** Let  $f(\lambda)$  be the characteristic equation of the matrix  $A$ :  $f(\lambda) = |A - \lambda I|$ . The value of the determinant  $|A|$  is found from the product of the eigenvalues. The following method for the estimation of the eigenvalues of a matrix is representative.

**3.4. Tridiagonal Method.** In this method, the matrix  $A$  is first transformed into a tridiagonal form by means of a similarity transformation. The estimate for the number of multiplications required in this operation is  $(4/3)n^3$ , and the number of additions required is  $(2/3)n^3$ . The additional effort to obtain the eigenvalues of the tridiagonal form is negligible, and hence the number of multiplications to obtain the value of the determinant is  $(4/3)n^3 + (n - 1)$  while the corresponding number of additions is  $(2/3)n^3$ .

**3.5. Determinants by Means of Solution of Linear Equations.** In our method we have to solve, successively, systems of linear equations of decreasing order. At the  $j$ th step, it is required to solve a system of  $j$  linear equations in  $j$  unknowns. If the number of multiplicative operations required for the particular auxiliary algorithm is  $J(j)$ , the total number of these operations will be

$$J = \sum_{j=1}^n J(j),$$

with similar expressions for divisions and additions.

The following estimates of the dominant term in the number of operations for the different possible methods of solving systems of  $n$  linear equations in  $n$  unknowns were obtained:

| <i>Method</i>  | <i>Multiplications</i> | <i>Additions</i> |
|--|------------------------|------------------|
| 1. Elimination   | $n^3/3$                | $n^3/3$          |
| 2. Seidel (one iteration)  | $n^2$                  | $n^2$            |
| 3. Relaxation (one iteration)  | $n^2$                  | $n^2$            |
| 4. Gradient method (one iteration)   | $2n^2$                 | $2n^2$           |
| 5. Conjugate gradient method, symmetric positive definite matrix (one iteration) | $n^2$                  | $n^2$            |
| 6. Conjugate gradient method, general matrix (one iteration)                     | $3n^2$                 | $3n^2$           |

In the absence of definite information about the problem to be considered, it is difficult to obtain an estimate of the total number of iterations required for the last five methods.

For our method, we obtain:

| <i>Auxiliary Method</i>  | <i>Multiplications</i> | <i>Additions</i> |
|--|------------------------|------------------|
| I. Elimination   | $n^4/12$               | $n^4/12$         |
| II. Seidel (one iteration)   | $n^3/3$                | $n^3/3$          |
| III. Relaxation (one iteration)  | $n^3/3$                | $n^3/3$          |
| IV. Gradient method (one iteration)  | $2n^3/3$               | $2n^3/3$         |
| V. Conjugate gradient method, symmetric positive definite matrix (one iteration) | $n^3/3$                | $n^3/3$          |
| VI. Conjugate gradient method, general matrix (one iteration)                    | $n^3$                  | $n^3$            |

**3.6. Numerical Comparison.** For a  $10 \times 10$  determinant, we obtain the following estimates for the number of operations required for the methods discussed:

| <i>Method</i>  | <i>Multiplications</i> | <i>Additions</i>   |
|--|------------------------|--------------------|
| 1. Expansion in elements                                       | $3.26 \times 10^7$     | $3.63 \times 10^6$ |
| 2. Expansion in minors   | $6.23 \times 10^6$     |                    |
| 3. Chio's method   | 330                    | 330                |
| 4. Tridiagonal method  | 1300                   | 660                |
| 5. Auxiliary method (by means of solution of linear equations) |                        |                    |
| I  | 1000                   | 1000               |
| II (one iteration)   | 330                    | 330                |
| III (one iteration)  | 330                    | 330                |
| IV (one iteration)   | 660                    | 660                |
| V (one iteration)  | 330                    | 330                |
| VI (one iteration)   | 1000                   | 1000               |

John Jay Hopkins Laboratory for Pure and Applied Science  
 General Atomic Division of General Dynamics Corporation  
 San Diego, California

1. K. S. KUNTZ, *Numerical Analysis*, McGraw-Hill, New York, 1957.  
 2. E. M. GRABBE, S. RAMO & D. E. WOOLDRIDGE, *Handbook of Automation, Computation and Control*, v. I, John Wiley and Sons, New York, 1958.

## Coefficients in Quadrature Formulas

By A. H. Stroud

The following result is well known (see, for example, Krylov [2], p. 104, or Szego [3], p. 48):

**THEOREM 1.** *If  $w(x)$  is nonnegative throughout the finite or infinite segment  $[a, b]$  and if the quadrature formula*

$$(1) \quad \int_a^b w(x)f(x) dx \simeq \sum_{i=1}^n A_i f(x_i)$$

---

Received October 29, 1962.