ANL-7213
Mathematics and
Computers (TID-4500)
AEC Research and
Development Report

ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

MULTISTEP INTEGER-PRESERVING GAUSSIAN ELIMINATION

by

Erwin H. Bareiss

Applied Mathematics Division

May 1966

Operated by The University of Chicago
under
Contract W-31-109-eng-38
with the
U. S. Atomic Energy Commission

TABLE OF CONTENTS

LIST OF FIGURES

MULTISTEP INTEGER-PRESERVING GAUSSIAN ELIMINATION

by

Erwin H. Bareiss

ABSTRACT

A method is developed which permits integer-preserving elimination in systems of linear equations, $AX = B$, such that (a) the magnitude of the coefficients in the transformed matrices is minimized, and (b) the computational efficiency is considerably increased in comparison with the corresponding ordinary (single-step) Gaussian elimination. The algorithms presented can also be used for the efficient evaluation of determinants and their leading minors. Explicit algorithms and flow charts are given for the two-step method. The method should also prove superior to the widely used fraction-producing Gaussian elimination when A is nearly singular.

I. EVALUATION OF DETERMINANTS

Let A be a square matrix of order n with elements a_{ij}, whose determinant is

$$|A| = \begin{vmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{vmatrix}. \tag{1}$$

The matrix A will be reduced to triangular form by integer-preserving Gaussian elimination. Assume for convenience of explanation that pivot searching with consequent interchange of rows is not considered. Then we define

$$|A^{(0)}| = |A| \qquad (a_{ij}^{(0)} = a_{ij}), \tag{2}$$

and

$$|A^{(1)}| = \frac{1}{\left[a_{11}^{(0)}\right]^{n-1}} \begin{vmatrix} a_{11}^{(0)} & a_{12}^{(0)} & a_{13}^{(0)} & \cdots & a_{1n}^{(0)} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2n}^{(1)} \\ 0 & a_{32}^{(1)} & a_{33}^{(1)} & \cdots & a_{3n}^{(1)} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & a_{n2}^{(1)} & a_{n3}^{(1)} & \cdots & a_{nn}^{(1)} \end{vmatrix} \tag{3}$$

$$|A^{(1)}| = \frac{1}{\left[a_{11}^{(0)}\right]^{n-2}} \begin{vmatrix} a_{22}^{(1)} & a_{23}^{(1)} & \dots & a_{2n}^{(1)} \\ a_{32}^{(1)} & a_{33}^{(1)} & \dots & a_{3n}^{(1)} \\ \dots & \dots & \dots & \dots \\ a_{n2}^{(1)} & a_{n3}^{(1)} & \dots & a_{nn}^{(1)} \end{vmatrix}, \qquad (3) \text{ Contd.}$$

where

$$a_{ij}^{(1)} = \begin{vmatrix} a_{11}^{(0)} & a_{1j}^{(0)} \\ a_{i1}^{(0)} & a_{ij}^{(0)} \end{vmatrix}. \qquad (4)$$

Continuing in an obvious way, we have

$$|A^{(k)}| = \frac{1}{\left[a_{11}^{(0)}\right]^{n-1} \dots \left[a_{kk}^{(k-1)}\right]^{n-k}} \begin{vmatrix} a_{11}^{(0)} & a_{12}^{(0)} & \dots & a_{1k}^{(0)} & a_{1,k+1}^{(0)} & \dots & a_{1n}^{(0)} \\ 0 & a_{22}^{(1)} & \dots & a_{2k}^{(1)} & a_{2,k+1}^{(1)} & \dots & a_{2n}^{(1)} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & a_{k+1,k+1}^{(k)} & \dots & a_{k+1,n}^{(k)} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & a_{n,k+1}^{(k)} & \dots & a_{nn}^{(k)} \end{vmatrix}$$

$$= \frac{1}{\left[a_{11}^{(0)}\right]^{n-2} \dots \left[a_{kk}^{(k-1)}\right]^{n-k-1}} \begin{vmatrix} a_{k+1,k+1}^{(k)} & \dots & a_{k+1,n}^{(k)} \\ \dots & \dots & \dots \\ a_{n,k+1}^{(k)} & \dots & a_{nn}^{(k)} \end{vmatrix}, \qquad (5)$$

where

$$a_{ij}^{(k)} = \begin{vmatrix} a_{kk}^{(k-1)} & a_{kj}^{(k-1)} \\ a_{ik}^{(k-1)} & a_{ij}^{(k-1)} \end{vmatrix}, \qquad (6)$$

and finally

$$|A^{(n-1)}| = \frac{1}{\left[a_{11}^{(0)}\right]^{n-1} \dots \left[a_{n-1,n-1}^{(n-2)}\right]^{1}} \begin{vmatrix} a_{11}^{(0)} & \dots & a_{1,n-1}^{(0)} & a_{1n}^{(0)} \\ 0 & \ddots & a_{n-1,n-1}^{(n-2)} & a_{n-1,n}^{(n-2)} \\ 0 & & 0 & a_{nn}^{(n-1)} \end{vmatrix}$$

$$= \frac{1}{\left[a_{11}^{(0)}\right]^{n-2} \dots \left[a_{n-2,n-2}^{(n-3)}\right]} a_{nn}^{(n-1)}. \qquad (7)$$

We note that $|A| = |A^{(0)}| = \ldots = |A^{(k)}| = \ldots = |A^{(n-1)}|$ is an integer when all $a_{ij}^{(0)}$ are integers. The same is true for $a_{11}^{(0)}, \ldots, a_{kk}^{(k-1)}, \ldots, a_{n-2,n-2}^{(n-3)}$, and the determinants

$$\begin{vmatrix} a_{k+1,k+1}^{(k)} & \ldots & a_{k+1,n}^{(k)} \\ \ldots & \ldots & \ldots \\ a_{n,k+1}^{(k)} & \ldots & a_{nn}^{(k)} \end{vmatrix} \quad (k = 1,\ 2,\ \ldots,\ n-1). \tag{8}$$

Thus by (5) the determinants (8) are divisible by

$$\left\{ \left[a_{11}^{(0)}\right]^{n-2} \ldots \left[a_{kk}^{(k-1)}\right]^{n-k-1} \right\}.$$

In particular, for $n = 3$, we have

$$|A^{(0)}| = |A^{(1)}| = |A^{(2)}|,$$

or explicitly

$$\begin{vmatrix} a_{11}^{(0)} & a_{12}^{(0)} & a_{13}^{(0)} \\ a_{21}^{(0)} & a_{22}^{(0)} & a_{23}^{(0)} \\ a_{31}^{(0)} & a_{32}^{(0)} & a_{33}^{(0)} \end{vmatrix} = \frac{1}{a_{11}^{(0)}} \begin{vmatrix} a_{22}^{(1)} & a_{23}^{(1)} \\ a_{32}^{(1)} & a_{33}^{(1)} \end{vmatrix} = \frac{a_{33}^{(2)}}{a_{11}^{(0)}}. \tag{9}$$

Because this relationship holds for every determinant of order 3, it is natural to replace (6) by the integer-preserving transformation

$$a_{ij}^{(k)} = \frac{1}{a_{k-1,k-1}^{(k-2)}} \begin{vmatrix} a_{kk}^{(k-1)} & a_{kj}^{(k-1)} \\ a_{ik}^{(k-1)} & a_{ij}^{(k-1)} \end{vmatrix}. \tag{10}$$

Letting $a_{00}^{(-1)} = 1$, we have now, instead of (5), $|A^{(0)}| = |A|$,

$$|A^{(k)}| = \frac{1}{a_{11}^{(0)} \ldots a_{k-1,k-1}^{(k-2)} \left[a_{kk}^{(k-1)}\right]^{n-k}} \begin{vmatrix} a_{11}^{(0)} & & \ldots & a_{1,k+1}^{(0)} & \ldots & a_{1,n}^{(0)} \\ 0 & a_{22}^{(1)} & \ldots & a_{2,k+1}^{(1)} & \ldots & a_{2,n}^{(1)} \\ & & \cdot & a_{k+1,k+1}^{(k)} & \ldots & a_{k+1,n}^{(k)} \\ & & & \ldots & & \ldots \\ 0 & & & a_{n,k+1}^{(k)} & \ldots & a_{n,n}^{(k)} \end{vmatrix} \tag{11}$$

$$|A^{(k)}| = \frac{1}{\left[a_{kk}^{(k-1)}\right]^{n-k-1}} \begin{vmatrix} a_{k+1,k+1}^{(k)} & \cdots & a_{k+1,n}^{(k)} \\ \cdots & & \cdots \\ a_{n,k+1}^{(k)} & \cdots & a_{nn}^{(k)} \end{vmatrix}, \qquad \text{(11) Contd.}$$

and, instead of (7),

$$|A^{(n-1)}| = a_{nn}^{(n-1)}. \qquad (12)$$

Since (12) is true for any n and any choice of the $a_{ij}^{(0)}$, it follows that we have for $n = k + 1$

$$a_{ij}^{(k)} = \begin{vmatrix} a_{11}^{(0)} & a_{12}^{(0)} & \cdots & a_{1k}^{(0)} & a_{1j}^{(0)} \\ a_{21}^{(0)} & a_{22}^{(0)} & \cdots & a_{2k}^{(0)} & a_{2j}^{(0)} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{k1}^{(0)} & a_{k2}^{(0)} & \cdots & a_{kk}^{(0)} & a_{kj}^{(0)} \\ a_{i1}^{(0)} & a_{i2}^{(0)} & \cdots & a_{ik}^{(0)} & a_{ij}^{(0)} \end{vmatrix}. \qquad (13)$$

In particular $a_{kk}^{(k-1)}$ $(k = 1, \ldots, n)$ are the main principal minors of A, with $a_{nn}^{(n-1)} = |A|$. Thus (10) can be used as a simple, integer-preserving, recurrence formula to calculate determinants and to investigate the signature of A. Its application among others is in stability theory (Hurwitz Criterion).

Because the right side of (13) is a determinant, we can apply (11) to $a_{ij}^{(k)}$ to obtain the representation

$$a_{ij}^{(k)} = \frac{1}{\left[a_{\ell\ell}^{(\ell-1)}\right]^{k-\ell}} \begin{vmatrix} a_{\ell+1,\ell+1}^{(\ell)} & \cdots & a_{\ell+1,k}^{(\ell)} & a_{\ell+1,j}^{(\ell)} \\ \cdots & \cdots & \cdots & \cdots \\ a_{k,\ell+1}^{(\ell)} & \cdots & a_{kk}^{(\ell)} & a_{kj}^{(\ell)} \\ a_{i,\ell+1}^{(\ell)} & \cdots & a_{ik}^{(\ell)} & a_{ij}^{(\ell)} \end{vmatrix}. \qquad (14)$$

For $\ell = 0$, (14) reduces to (13); for $\ell = k - 1$, (14) reduces to (10); and, of course, for $\ell = k$, (14) is simply the identity $a_{ij}^{(k)} = a_{ij}^{(k)}$.

We have shown that the transformations (10) yield successively, as diagonal elements, the leading principal minors and thus lead gradually to the calculation of the determinant of A. Since these minors are the smallest numbers (in absolute value) that can reasonably be expected from a general integer-preserving transformation, there is little incentive to search for formulas that yield smaller $a_{ij}^{(k)}$ than those given by (10). Instead, a useful question, which can be answered affirmatively, is: Can the $a_{ij}^{(k)}$ be computed more efficiently than by the recurrence formula (10)?

Once all $a_{ij}^{(\ell)}$'s are known, we can determine any $a_{ij}^{(k)}$ $(k > \ell)$ by (14). If we calculate the elements of a row, which means that i will be fixed, the determinant in (14) can be expanded by the last column. We see then that the cofactors of $a_{kj}^{(\ell)}$ are common to each element of the row and therefore must be calculated but once for each row. Indeed, the cofactor of $a_{ij}^{(\ell)}$ is even independent of both i and j. After the cofactors are determined, there will be only $(k - \ell + 1)$ multiplications necessary to advance from $a_{ij}^{(\ell)}$ to $a_{ij}^{(k)}$. If we choose to calculate the new elements of a column instead of a row, which means that j will be fixed instead of i, the determinant in (14) can be expanded by the last row and conclusions corresponding to those above can be reached also.

Furthermore, *all the cofactors are divisible by* $\left[a_{\ell\ell}^{(\ell-1)}\right]^{k-\ell-1}$. This is obvious for the cofactor of $a_{ij}^{(\ell)}$ because it follows from (14) for $a_{kk}^{(k-1)}$. For the rest of the cofactors it is sufficient for proof to note that since interchanging rows or columns does not affect the absolute value of a determinant, the matrix A could have been arranged so that any one of the border elements $a_{im}^{(\ell)}$ or $a_{mj}^{(\ell)}$ $(\ell + 1 \le m \le k)$ takes the place of the present corner element $a_{ij}^{(\ell)}$.

Thus, $a_{ij}^{(k)}$ could be calculated by an integer-preserving recurrence formula of the form

$$a_{ij}^{(k)} = \left[c_{kk}^{(\ell)} a_{ij}^{(\ell)} + \sum_{m=\ell+1}^{k} c_{mj}^{(\ell)} a_{mj}^{(\ell)}\right] \Big/ a_{\ell\ell}^{(\ell-1)} \qquad (15)$$

or

$$a_{ij}^{(k)} = \left[c_{kk}^{(\ell)} a_{ij}^{(\ell)} + \sum_{m=\ell+1}^{k} c_{im}^{(\ell)} a_{im}^{(\ell)}\right] \Big/ a_{\ell\ell}^{(\ell-1)}, \qquad (16)$$

where the $c_{mj}^{(\ell)}$, $c_{im}^{(\ell)}$ are the divided cofactors discussed above. The last two formulas have the advantage of keeping the absolute value of the numerator as small as can reasonably be expected in general. This statement means that matrices exist such that dividend and divisor in (15) and (16) are relatively prime.

From the multitude of transformations given by (14) we restrict ourselves in what follows to $\ell = k - 1$ and $\ell = k - 2$.

Note can be taken of the following further property, which follows from (13) or (14): If A is symmetric, then $a_{ij}^{(k)} = a_{ji}^{(k)}$ for $i,j > k$.

II. INTEGER-PRESERVING TRANSFORMATIONS FOR THE EXACT SOLUTION OF SYSTEMS OF LINEAR EQUATIONS

Let a linear system of equations be given by

$$AX = B, \tag{1}$$

where

$$A = (a_{ij}) = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \cdots & \cdots & \cdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}, \tag{2}$$

$$B = (a_{ij}) = \begin{pmatrix} a_{1,n+1} & \cdots & a_{1m} \\ \cdots & \cdots & \cdots \\ a_{n,n+1} & \cdots & a_{nm} \end{pmatrix}, \tag{3}$$

and

$$X = \begin{pmatrix} x_{11} & \cdots & x_{1,m-n} \\ \vdots & & \vdots \\ x_{n1} & \cdots & x_{n,m-n} \end{pmatrix}. \tag{4}$$

To solve (1), A shall be reduced

a. to triangular form with subsequent back substitution,

b. to diagonal form

such that the elements of the reduced system are integers, provided the elements a_{ij} of

$$A^{(0)} = A \oplus B \quad (A \text{ augmented by } B) \tag{5}$$

are integers.

A. Reduction of A to Triangular Form

1. *Division-free Algorithms*

The simplest reduction algorithm is given by Eq. (1.6).* The recurrence formulas are

---

*Equation (1.6) means Eq. (6) of Section I.

$$a_{ij}^{(0)} = a_{ij}$$

$$a_{ij}^{(k)} = \begin{vmatrix} a_{kk}^{(k-1)} & a_{kj}^{(k-1)} \\ a_{ik}^{(k-1)} & a_{ij}^{(k-1)} \end{vmatrix} \tag{6}$$

$(k = 1, 2, \ldots, n - 1) \quad (i = k + 1, \ldots, n)$

$(j = k + 1, \ldots, n, n + 1, \ldots, m)$.

The advantage of this formula is the absence of any division operations. The disadvantage lies in large absolute integers $a_{ij}^{(k)}$.

The next-simplest division-free transformation is given by Eq. (1.14), if the divisor is disregarded and $\ell = k - 2$. The result is

$$a_{ij}^{(k)} = \begin{vmatrix} a_{k-1,k-1}^{(k-2)} & a_{k-1,k}^{(k-2)} & a_{k-1,j}^{(k-2)} \\ a_{k,k-1}^{(k-2)} & a_{kk}^{(k-2)} & a_{kj}^{(k-2)} \\ a_{i,k-1}^{(k-2)} & a_{ik}^{(k-2)} & a_{ij}^{(k-2)} \end{vmatrix}. \tag{7}$$

It is also instructive to obtain (7) directly from (6) instead of from (1.14) by applying (6) twice as follows:

$$\begin{aligned} a_{ij}^{(k)} &= \begin{vmatrix} a_{kk}^{(k-1)} & a_{kj}^{(k-1)} \\ a_{ik}^{(k-1)} & a_{ij}^{(k-1)} \end{vmatrix} \\ &= \left(a_{k-1,k-1}^{(k-2)} a_{kk}^{(k-2)} - a_{k-1,k}^{(k-2)} a_{k,k-1}^{(k-2)}\right)\left(a_{k-1,k-1}^{(k-2)} a_{ij}^{(k-2)} - a_{k-1,j}^{(k-2)} a_{i,k-1}^{(k-2)}\right) \\ &\quad - \left(a_{k-1,k-1}^{(k-2)} a_{ik}^{(k-2)} - a_{k-1,k}^{(k-2)} a_{i,k-1}^{(k-2)}\right)\left(a_{k-1,k-1}^{(k-2)} a_{kj}^{(k-2)} - a_{k-1,j}^{(k-2)} a_{k,k-1}^{(k-2)}\right) \\ &= \left(a_{k-1,k-1}^{(k-2)} a_{kk}^{(k-2)} - a_{k-1,k}^{(k-2)} a_{k,k-1}^{(k-2)}\right) a_{k-1,k-1}^{(k-2)} a_{ij}^{(k-2)} \\ &\quad - \left(a_{k-1,k-1}^{(k-2)} a_{ik}^{(k-2)} - a_{k-1,k}^{(k-2)} a_{i,k-1}^{(k-2)}\right) a_{k-1,k-1}^{(k-2)} a_{kj}^{(k-2)} \\ &\quad - a_{k-1,k-1}^{(k-2)} a_{kk}^{(k-2)} a_{k-1,j}^{(k-2)} a_{i,k-1}^{(k-2)} + \left[a_{k-1,k}^{(k-2)} a_{k,k-1}^{(k-2)} a_{k-1,j}^{(k-2)} a_{i,k-1}^{(k-2)}\right] \\ &\quad + a_{k-1,k-1}^{(k-2)} a_{ik}^{(k-2)} a_{k-1,j}^{(k-2)} a_{k,k-1}^{(k-2)} - \left[a_{k-1,k}^{(k-2)} a_{i,k-1}^{(k-2)} a_{k-1,j}^{(k-2)} a_{k,k-1}^{(k-2)}\right]. \end{aligned}$$

The two products indicated by brackets [ ] cancel. The remaining terms have the common factor $a_{k-1,k-1}^{(k-2)}$. It then follows easily that for (6)

$$a_{ij}^{(k)} = a_{k-1,k-1}^{(k-2)} \begin{vmatrix} a_{k-1,k-1}^{(k-2)} & a_{k-1,k}^{(k-2)} & a_{k-1,j}^{(k-2)} \\ a_{k,k-1}^{(k-2)} & a_{kk}^{(k-2)} & a_{kj}^{(k-2)} \\ a_{i,k-1}^{(k-2)} & a_{ik}^{(k-2)} & a_{ij}^{(k-2)} \end{vmatrix} .$$

Disregarding the factor $a_{k-1,k-1}^{(k-2)}$ in this equation yields (7). Therefore, the coefficients $a_{ij}^{(k)}$ of (7) are smaller by a factor $a_{k-1,k-1}^{(k-2)}$ and, in addition, can be obtained from $a_{ij}^{(k-2)}$ more efficiently than those of (6) because two terms cancel and need not be calculated. This fact also implies greater numerical stability if the a_{ij} are not integers.

To save space in the fast memory of an electronic computer, the recursion formulas should be arranged in such a way that overwriting is possible. Because (7) implies that two iteration steps are taken at once, care must be taken to sequence the calculations properly. The recursion algorithm, transforming one row at a time, is given below by Eqs. (8); and the proper sequencing of the calculation is determined by the flow chart shown in Fig. 1.

$$\left.\begin{aligned}
&a_{ij}^{(0)} = a_{ij}; \qquad c_0^{(k-2)} = \begin{vmatrix} a_{k-1,k-1}^{(k-2)} & a_{k-1,k}^{(k-2)} \\ a_{k,k-1}^{(k-2)} & a_{kk}^{(k-2)} \end{vmatrix}; \\
&c_{i1}^{(k-2)} = - \begin{vmatrix} a_{k-1,k-1}^{(k-2)} & a_{k-1,k}^{(k-2)} \\ a_{i,k-1}^{(k-2)} & a_{ik}^{(k-2)} \end{vmatrix}; \qquad c_{i2}^{(k-2)} = \begin{vmatrix} a_{k,k-1}^{(k-2)} & a_{kk}^{(k-2)} \\ a_{i,k-1}^{(k-2)} & a_{ik}^{(k-2)} \end{vmatrix}; \\
&a_{ij}^{(k)} = a_{ij}^{(k-2)} c_0^{(k-2)} + a_{kj}^{(k-2)} c_{i1}^{(k-2)} + a_{k-1,j}^{(k-2)} c_{i2}^{(k-2)}, \\
&\qquad \textit{for } i = k+1, \ldots, n; \quad j = k+1, \ldots, m; \\
&a_{k\ell}^{(k-1)} = \begin{vmatrix} a_{k-1,k-1}^{(k-2)} & a_{k-1,\ell}^{(k-2)} \\ a_{k,k-1}^{(k-2)} & a_{k\ell}^{(k-2)} \end{vmatrix} = a_{k\ell}^{(k)}, \quad \textit{for } \ell = k, \ldots, m.
\end{aligned}\right\} \qquad (8)$$

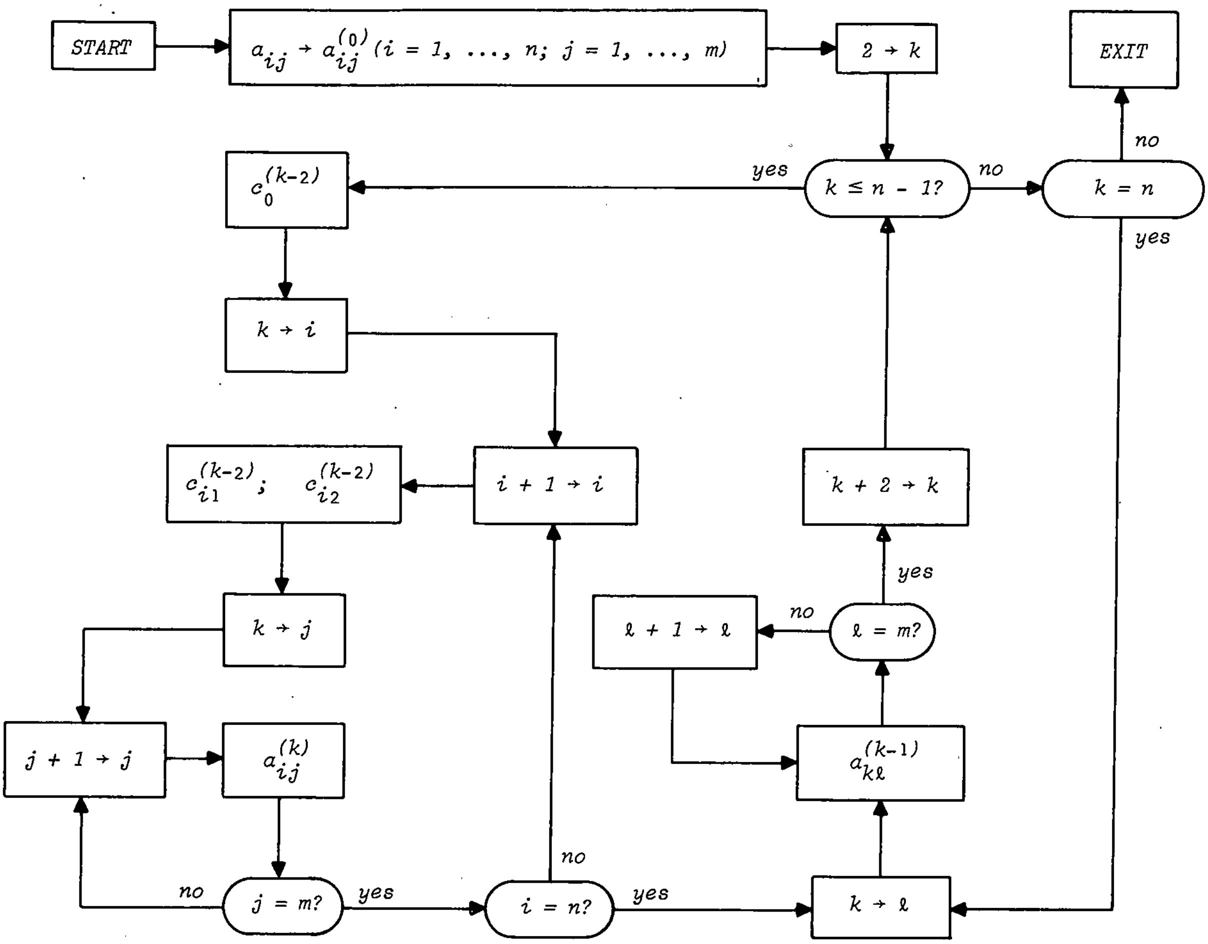

Fig. 1. Flow Chart for the Algorithms (8), (11), and (12)

It is worthwhile to visualize the effect of the transformations (8) on the matrix $A^{(k-2)}$. The equation for $a_{ij}^{(k)}$, when formally extended to $j = k - 1$ and $j = k$, reduces the elements $a_{ij}^{(k-2)} (i > k)$ to zero for two columns, and leaves the elements $a_{kj}^{(k-2)}$ unchanged. Once the elements $a_{ij}^{(k)} (i > k)$ have been determined, the element $a_{k,k-1}^{(k-2)}$ is transformed to zero by the formula for $a_{k\ell}^{(k-1)}$. This sequence in calculating the transformation is, of course, required only because the elements $a_{k-1,j}^{(k-2)}$, $a_{kj}^{(k-2)}$ are needed to calculate $a_{ij}^{(k)} (i > k)$, and space requirements are minimized in the fast memory by overwriting and avoiding unnecessary working storage.

By replacing the third-order determinant by a fourth-order determinant, one can develop a simultaneous, three-step, division-free, elimination algorithm similar to the two-step algorithm (8), and so on. Each of these algorithms will produce smaller integers in the final triangular matrix $A^{(n-1)}$ than the previous algorithms.

2. *Fraction-free Algorithms*

The direct use of (1.14) also yields integer-preserving transformations, but requires divisions in each step. Letting $\ell = k - 1$ in (1.14) yields (1.10), and the algorithm corresponding to (6) is as follows:

$$\left.\begin{aligned} &a_{00}^{(-1)} = 1, \quad a_{ij}^{(0)} = a_{ij}; \\ &b_{ij}^{(k)} = \begin{vmatrix} a_{kk}^{(k-1)} & a_{kj}^{(k-1)} \\ a_{ik}^{(k-1)} & a_{ij}^{(k-1)} \end{vmatrix}; \\ &a_{ij}^{(k)} = b_{ij}^{(k)} \Big/ a_{k-1,k-1}^{(k-2)}; \end{aligned}\right\} \tag{9}$$

$$(k = 1, \ldots, n - 1; \quad i = k + 1, \ldots, n; \quad j = k + 1, \ldots, m).$$

Letting $\ell = k - 2$ in (1.14) yields

$$a_{ij}^{(k)} = \frac{1}{\left[a_{k-2,k-2}^{(k-3)}\right]^2} \begin{vmatrix} a_{k-1,k-1}^{(k-2)} & a_{k-1,k}^{(k-2)} & a_{k-1,j}^{(k-2)} \\ a_{k,k-1}^{(k-2)} & a_{kk}^{(k-2)} & a_{kj}^{(k-2)} \\ a_{i,k-1}^{(k-2)} & a_{ik}^{(k-2)} & a_{ij}^{(k-2)} \end{vmatrix}. \tag{10}$$

According to Section I, the minors of order two are divisible by $a_{k-2,k-2}^{(k-3)}$. Thus, we have the following two algorithms (restricting ourselves, as before, to row-by-row transformations only). The first alternative is as follows:

$$\left.\begin{aligned} &a_{00}^{(-1)} = 1, \quad a_{ij}^{(0)} = a_{ij}; \\ &b_{ij}^{(k)} = a_{ij}^{(k-2)} c_{0}^{(k-2)} + a_{kj}^{(k-2)} c_{i1}^{(k-2)} + a_{k-1,j}^{(k-2)} c_{i2}^{(k-2)}; \\ &a_{ij}^{(k)} = b_{ij}^{(k)} \Big/ \left[a_{k-2,k-2}^{(k-3)}\right]^2; \\ &b_{k\ell}^{(k-1)} = a_{k-1,k-1}^{(k-2)} a_{k\ell}^{(k-2)} - a_{k,k-1}^{(k-2)} a_{k-1,\ell}^{(k-2)}; \\ &a_{k\ell}^{(k-1)} = a_{k\ell}^{(k)} = b_{k\ell}^{(k-1)} \Big/ a_{k-2,k-2}^{(k-3)}. \end{aligned}\right\} \tag{11}$$

In this algorithm, the $c_0^{(k-2)}$, $c_{i1}^{(k-2)}$, $c_{i2}^{(k-2)}$ are computed as in (8). Also, the range for i, j, k, ℓ is the same, and the sequence of computation is prescribed by the flow chart shown in Fig. 1. In writing down (11), we have emphasized that divisions must be carried out as the last arithmetic operation in determining $a_{ij}^{(k)}$, $a_{k\ell}^{(k)}$, to preserve fraction-free (i.e., integer) arithmetic.

For the second alternative, we divide the c's of (11) by $a_{k-2,k-2}^{(k-3)}$ before computing $a_{ij}^{(k)}$. This has the advantageous effect that the $b_{ij}^{(k)}$ of (11) will be replaced by smaller absolute integers. The net effect in computational efficiency is: one multiplication $\left(\text{to obtain } \left[a_{k-2,k-2}^{(k-3)}\right]^2\right)$ is saved, and one division per k-recursion and two divisions per i-recursion are added. If one is willing to accept this penalty in efficiency (which for large systems is relatively small), the following algorithm evolves. In each equation, the division, if any, should be the last arithmetic operation. We also take advantage of the fact that $a_{k-2,k-2}^{(k-3)} = a_{k-2,k-2}^{(k-2)}$.

$$\left.\begin{aligned}
a_{00}^{(0)} &= 1; \quad a_{ij}^{(0)} = a_{ij};\\
c_0^{(k-2)} &= \left(a_{k-1,k-1}^{(k-2)} a_{kk}^{(k-2)} - a_{k-1,k}^{(k-2)} a_{k,k-1}^{(k-2)}\right) \Big/ a_{k-2,k-2}^{(k-2)};\\
c_{i1}^{(k-2)} &= \left(a_{k-1,k}^{(k-2)} a_{i,k-1}^{(k-2)} - a_{k-1,k-1}^{(k-2)} a_{ik}^{(k-2)}\right) \Big/ a_{k-2,k-2}^{(k-2)};\\
c_{i2}^{(k-2)} &= \left(a_{k,k-1}^{(k-2)} a_{ik}^{(k-2)} - a_{kk}^{(k-2)} a_{i,k-1}^{(k-2)}\right) \Big/ a_{k-2,k-2}^{(k-2)};\\
a_{ij}^{(k)} &= \left(a_{ij}^{(k-2)} c_0^{(k-2)} + a_{kj}^{(k-2)} c_{i1}^{(k-2)} + a_{k-1,j}^{(k-2)} c_{i2}^{(k-2)}\right) \Big/ a_{k-2,k-2}^{(k-2)}\\
&\quad (\text{for } i = k+1, \ldots, n; \quad j = k+1, \ldots, m);\\
a_{k\ell}^{(k-1)} &= \left(a_{k-1,k-1}^{(k-2)} a_{k\ell}^{(k-2)} - a_{k-1,\ell}^{(k-2)} a_{k,k-1}^{(k-2)}\right) \Big/ a_{k-2,k-2}^{(k-2)} = a_{k\ell}^{(k)}\\
&\quad (\text{for } \ell = k, \ldots, m).
\end{aligned}\right\} \quad (12)$$

Again, the sequence of computation is prescribed by the flow chart in Fig. 1. As shown in Section I, the elements $a_{kk}^{(k-1)} \equiv a_{kk}^{(n-1)}$ $(k = 1, \ldots, n)$, obtained by (11) or (12), are the leading principal minors of A; in particular, $a_{nn}^{(n-1)} = |A|$.

In a similar manner, multistep elimination algorithms can be developed from (1.14).

After A has been reduced to triangular form, the system (1) can be solved for X by back substitution, using either rational arithmetic, or another suitable special algorithm. The net effect is, of course, the reduction of A to diagonal form.

B. Reduction of A to Diagonal Form

The extension of the one-step algorithms (6) and (9) to achieve reduction of A to diagonal form is simply accomplished by applying the transformation also to the elements of the rows 1 to $k - 1$. Corresponding to (9), we have the algorithm

$$\left.\begin{aligned} &a_{00}^{(-1)} = 1, \quad a_{ij}^{(0)} = a_{ij}; \\ &a_{ij}^{(k)} = \left(a_{kk}^{(k-1)} a_{ij}^{(k-1)} - a_{kj}^{(k-1)} a_{ik}^{(k-1)}\right) \Big/ a_{k-1,k-1}^{(k-1)} \\ &\qquad (i \neq k; \quad j = k+1, \ldots, m); \\ &a_{kj}^{(k)} = a_{kj}^{(k-1)}. \end{aligned}\right\} \tag{13}$$

We purposely omitted

$$a_{ii}^{(k)} = a_{kk}^{(k)} \quad (i = 1, \ldots, k - 1). \tag{14}$$

The algorithm (13) needs some explanation. To begin with, the last equation in (13) states that only row k remains unchanged; in particular, that $a_{kk}^{(k)} = a_{kk}^{(k-1)}$. Applying this identity to the divisor $a_{k-1,k-1}^{(k-2)}$ of (9), we have written for the divisor in (13) not $a_{k-1,k-1}^{(k-2)}$ but $a_{k-1,k-1}^{(k-1)}$. Assume now that (14) is true for $(k - 1)$. Because $a_{kj}^{(k-1)} = 0$ for $j < k$, it follows from the second line of (13) for $a_{ii}^{(k)}$ $(i < k)$, from (14) for $(k - 1)$ and from the last equation in (13) that

$$a_{ii}^{(k)} = \frac{a_{kk}^{(k-1)} a_{ii}^{(k-1)} - 0}{a_{k-1,k-1}^{(k-1)}} = \frac{a_{kk}^{(k-1)} a_{k-1,k-1}^{(k-1)}}{a_{k-1,k-1}^{(k-1)}} = a_{kk}^{(k-1)} = a_{kk}^{(k)}.$$

Since this equation is true for $k = 2$, (14) is true. Thus, when $k = n$, $A^{(n)} \oplus B^{(n)}$ should have the form

$$A^{(n)} \oplus B^{(n)} = \begin{matrix} a_{nn}^{(n)} \cdots 0 & a_{1,n+1}^{(n)} \cdots a_{1m}^{(n)} \\ \vdots \qquad \vdots & \vdots \qquad \vdots \\ 0 \cdots a_{nn}^{(n)} & a_{n,n+1}^{(n)} \cdots a_{nm}^{(n)} \end{matrix}, \tag{15}$$

where $a_{nn}^{(n)} = a_{nn}^{(n-1)} = |A|$. Then the solution to (1) is given by

$$x_{rs} = a_{r,n+s}^{(n)} \Big/ a_{nn}^{(n)}. \tag{16}$$

Note that the numerator and denominator in (16) have the same numerical values as if (1) were solved by *Cramer's* rule.

However, algorithm (13) as presented above yields not the matrix (15) in the fast memory but, after proper identification of the actual operations,

$$\begin{pmatrix} a_{11}^{(0)} & a_{12}^{(1)} & a_{13}^{(2)} & \cdots & a_{1n}^{(n-1)} & a_{1,n+1}^{(n)} & \cdots & a_{1m}^{(n)} \\ a_{21}^{(0)} & a_{22}^{(1)} & a_{23}^{(2)} & \cdots & a_{2n}^{(n-1)} & a_{2,n+1}^{(n)} & \cdots & a_{2m}^{(n)} \\ a_{31}^{(0)} & a_{32}^{(1)} & a_{33}^{(2)} & \cdots & a_{3n}^{(n-1)} & a_{3,n+1}^{(n)} & \cdots & a_{3m}^{(n)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & & \vdots \\ a_{n1}^{(0)} & a_{n2}^{(1)} & a_{n3}^{(2)} & & a_{nn}^{(n-1)} & a_{n,n+1}^{(n)} & \cdots & a_{nm}^{(n)} \end{pmatrix}, \tag{17}$$

and the solution of (1) is then given by

$$x_{rs} = a_{r,n+s}^{(n)} \Big/ a_{nn}^{(n-1)}, \tag{18}$$

which is identical to (16). The elements $a_{kk}^{(k-1)}$ in (17) are now the leading principal minors of A, as given by (1.13).

Next, we implement the two-step algorithm (12) to yield a reduction of A to diagonal form. Assume diagonalization has been achieved up to $a_{k-2,k-2}^{(k-3)}$. Then the application of (10) to the $a_{ij}^{(k-2)}$'s of all rows except $i = k - 1$ and $i = k$ yields the matrix

$$\left[\begin{array}{c|cc|c} a_{k-2,k-2}^{(k-2)} & \vdots \; 0 & \vdots \; 0 & a_{k-2,k+1}^{(k)} \cdots \\ \hline \cdots \; 0 & a_{k-1,k-1}^{(k-2)} & a_{k-1,k}^{(k-2)} & a_{k-1,k+1}^{(k-2)} \cdots \\ \cdots \; 0 & a_{k,k-1}^{(k-2)} & a_{kk}^{(k-2)} & a_{k,k+1}^{(k-2)} \cdots \\ \hline & 0 & 0 & a_{k+1,k+1}^{(k)} \cdots \\ 0 & \vdots & \vdots & \vdots \end{array}\right] \tag{19}$$

The element $a_{k,k-1}^{(k-2)}$ in (19) is then transformed to zero by the last equation of (12) to yield

$$\left[\begin{array}{c|cc|cc} a_{k-2}^{(k-2)} & 0 & 0 & & \\ \hline 0 & a_{k-1,k-1}^{(k-2)} & a_{k-1,k}^{(k-2)} & a_{k-1,k+1}^{(k-2)} & \cdots \\ 0 & 0 & a_{kk}^{(k-1)} & a_{k,k+1}^{(k-1)} & \cdots \\ \hline & 0 & 0 & a_{k+1,k+1}^{(k)} & \end{array}\right] \tag{20}$$

It remains to transform $a_{k-1,k}^{(k-2)}$ to zero. We note that in (20),

$$a_{k-1,j}^{(k-2)} = a_{k-1,j}^{(k-1)}.$$

But then, (13) can be applied to calculate $a_{ij}^{(k)}$ for $i = k - 1$.

To terminate the iteration process, we will have to distinguish as before between n even and n odd and omit the appropriate algorithms which become unnecessary. The final matrix $A^{(n)} \oplus B^{(n)}$ again has the theoretical appearance (15); and the actual contents of the memory cells of the original $a_{ij}^{(0)}$ are given by (17). The solution of (1) is given by (18).

Thus we have the following algorithm, where the division, if any, should be the last arithmetic operation:

$$\left.\begin{aligned}
a_{00}^{(0)} &= 1, \quad a_{ij}^{(0)} = a_{ij}; \\
c_0^{(k-2)} &= \left(a_{k-1,k-1}^{(k-2)} a_{kk}^{(k-2)} - a_{k-1,k}^{(k-2)} a_{k,k-1}^{(k-2)}\right) \Big/ a_{k-2,k-2}^{(k-2)}; \\
c_{i1}^{(k-2)} &= \left(a_{k-1,k}^{(k-2)} a_{i,k-1}^{(k-2)} - a_{k-1,k-1}^{(k-2)} a_{ik}^{(k-2)}\right) \Big/ a_{k-2,k-2}^{(k-2)}; \\
c_{i2}^{(k-2)} &= \left(a_{k,k-1}^{(k-2)} a_{ik}^{(k-2)} - a_{kk}^{(k-2)} a_{i,k-1}^{(k-2)}\right) \Big/ a_{k-2,k-2}^{(k-2)}; \\
a_{ij}^{(k)} &= \left(a_{ij}^{(k-2)} c_0^{(k-2)} + a_{kj}^{(k-2)} c_{i1}^{(k-2)} + a_{k-1,j}^{(k-2)} c_{i2}^{(k-2)}\right) \Big/ a_{k-2,k-2}^{(k-2)} \\
&\qquad \text{for } (i \neq k,\ i \neq k-1; \quad j = k+1, \ldots, m); \\
a_{kp}^{(k-1)} &= \left(a_{k-1,k-1}^{(k-2)} a_{kp}^{(k-2)} - a_{k,k-1}^{(k-2)} a_{k-1,p}^{(k-2)}\right) \Big/ a_{k-2,k-2}^{(k-2)} = a_{kp}^{(k)} \\
&\qquad (p = k, \ldots, m);
\end{aligned}\right\} \tag{21}$$

$$\left.\begin{aligned} a_{k-1,q}^{(k)} &= \left(a_{kk}^{(k-1)} a_{k-1,q}^{(k-2)} - a_{kq}^{(k-1)} a_{k-1,k}^{(k-2)}\right) \Big/ a_{k-1,k-1}^{(k-2)} \\ & \qquad (q = k+1, \ldots, m); \\ &\text{and finally if } n \text{ is odd} \\ a_{ij}^{(n)} &= \left(a_{nn}^{(n-1)} a_{ij}^{(n-1)} - a_{nj}^{(n-1)} a_{in}^{(n-1)}\right) \Big/ a_{n-1,n-1}^{(n-1)} \\ & \qquad (i = 1, 2, \ldots, n-1,\ j = n+1, \ldots, m). \end{aligned}\right\} \quad (21) \text{ Contd.}$$

Again we have omitted $a_{ii}^{(k)} = a_{kk}^{(k)} (i = 1, \ldots, k - 1)$. The sequence of computation is prescribed by the flow chart shown in Fig. 2.

In a similar way, one can construct three-step elimination algorithms, and so on.

III. FRACTION-PRODUCING AND MULTIPLICATION-FREE ELIMINATION METHODS

For completeness and comparison, it should be recognized that one can improve the efficiency of the elimination by reducing diagonal elements to unity, but thereby sacrificing integer preservation:

$$a_{kj}^{(k)} = a_{kj}^{(k-1)} \Big/ a_{kk}^{(k-1)}; \quad a_{ij}^{(k)} = a_{ij}^{(k-1)} - a_{ik}^{(k-1)} a_{kj}^{(k)}. \tag{1}$$

Several equivalent techniques have been devised for the proper utilization of the Gaussian algorithm (1) to bring a square matrix into triangular form.* All use $(m - k)$ divisions and $(m - k)(n - k)$ multiplications and subtractions each to obtain $A^{(k)}$ from $A^{(k-1)}$.

The two-step algorithms of the previous sections can be reduced to

$$\left.\begin{aligned} a_{k-1,j}^{(k-1)} &= a_{k-1,j}^{(k-2)} \Big/ a_{k-1,k-1}^{(k-2)} && (j = k, \ldots, m); \\ a_{kj}^{(k-1)} &= a_{kj}^{(k-2)} - a_{k,k-1}^{(k-2)} a_{k-1,j}^{(k-1)} && (j = k, \ldots, m); \\ a_{kj}^{(k)} &= a_{kj}^{(k-1)} \Big/ a_{kk}^{(k-1)} && (j = k+1, \ldots, m); \\ a_{ij}^{(k)} &= a_{ij}^{(k-2)} - a_{kj}^{(k)} c_i^{(k)} - a_{k-1,j}^{(k-1)} a_{i,k-1}^{(k-2)} && (j = k+1, \ldots, m); \end{aligned}\right\} \quad (2)$$

where

$$c_i^{(k)} = a_{ik}^{(k-2)} - a_{k-1,k}^{(k-1)} a_{i,k-1}^{(k-2)} \qquad (i = k+1, \ldots, n).$$

*M. H. Doolittle (1878), T. Banachiewicz (1938), and P. D. Crout (1942).

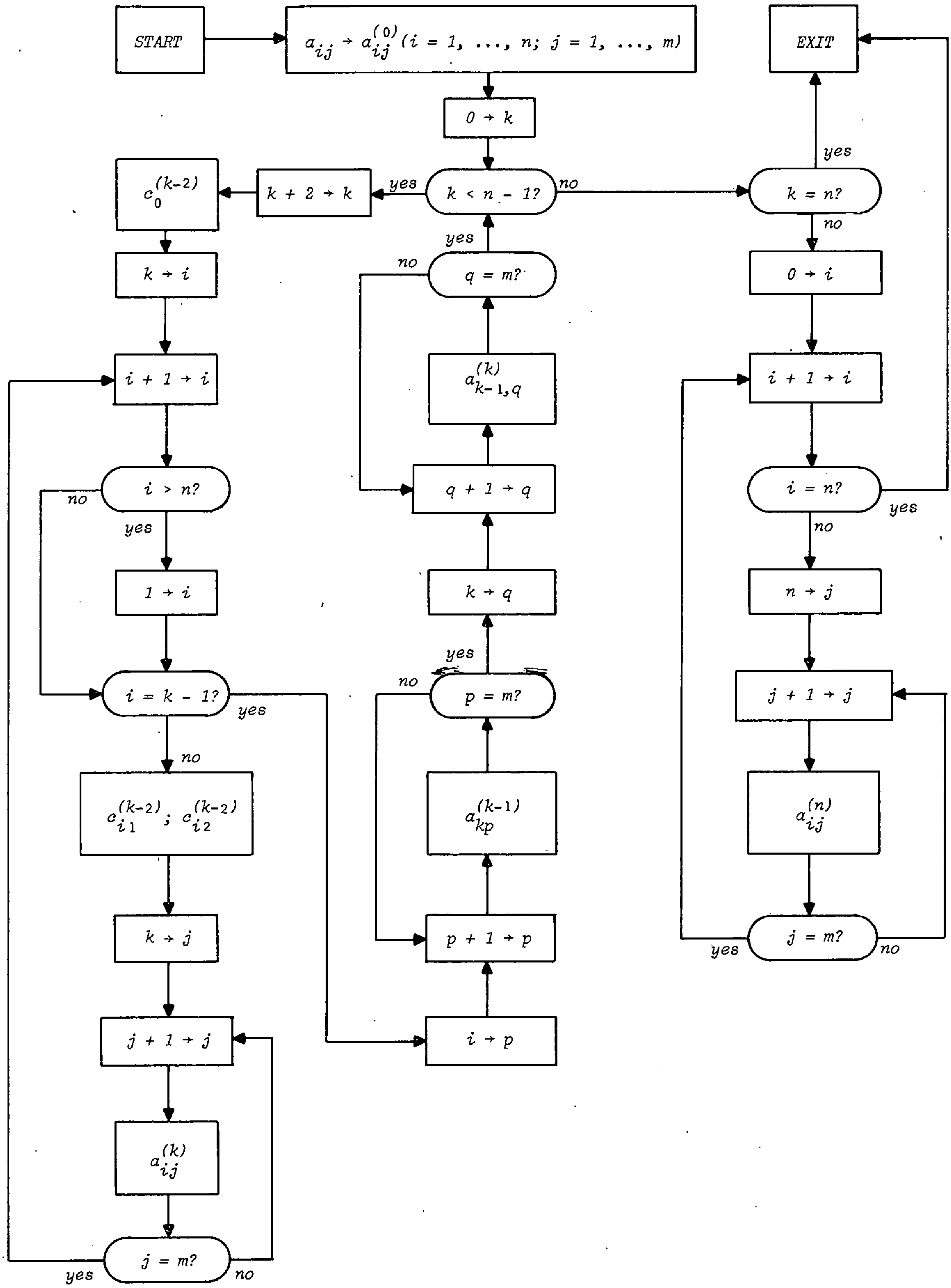

Fig. 2. Flow Chart for the Algorithm (21)

For the reduction of $A^{(k-2)}$ to $A^{(k)}$ by this arrangement, both (1) and (2) need the same number of arithmetic operations, namely $2(m - k) + 1$ divisions, and $2(m - k)(n - k) + (n - k) + (m - k) + 1$ multiplications and subtractions each. In (2), as against (1), only about half the number of new words need to be addressed to obtain a new element $a_{ij}^{(k)}$. But this advantage is balanced by the need for a more complicated algorithm (2). Thus, under the assumptions of this section, no significant advantage can be expected by using (2).

The following multiplication-free algorithm may also be of interest. We start by dividing each row by its first element and then subtract the first row from all other rows. This transformation makes all elements of the first column zero except a_{11}, which is unity. Then, we divide rows 2 to n by a_{i2} and subtract row 2 from all others below, and so on, to give

$$\left.\begin{aligned} a_{ij}^{(k-\frac{1}{2})} &= a_{ij}^{(k-1)} \big/ a_{ik}^{(k-1)} & (j > k); \\ a_{ij}^{(k)} &= a_{ij}^{(k-\frac{1}{2})} - a_{kj}^{(k-\frac{1}{2})} & (i,\ j > k); \end{aligned}\right\} \quad (3)$$

where

$$a_{kj}^{(k)} = a_{kj}^{(k-\frac{1}{2})},$$

of course. To transform $A^{(k-1)}$ into $A^{(k)}$, there are $(m - k)(n - k + 1)$ divisions and $(m - k)(n - k)$ subtractions but no multiplications necessary. For modern computers, which divide as fast, or nearly as fast, as they multiply, (3) is better suited, for the pivoting sweep and division sweep can be combined into a single sweep. The algorithm (3) can also be used to transform A into diagonal form with $a_{ii}^{(n)} = 1$ $(i = 1, \ldots, n)$ by changing $(i > k)$ into $(i \neq k)$.

IV. EFFICIENCY OF THE INTEGER-PRESERVING ALGORITHMS

To transform $A^{(k-1)}$ into $A^{(k)}$ in the process of reducing A to triangular form, the one-step integer-preserving methods need

$2(m - k)(n - k)$ multiplications,

$(m - k)(n - k)$ subtractions,

and, unless we choose the division-free algorithm (2.6),

$(m - k)(n - k)$ divisions.

To advance from $A^{(k-2)}$ to $A^{(k)}$, we need

$4(m - k)(n - k) + 2(n - k) + 2(m - k) + 2$ *multiplications,*

and

$2(m - k)(n - k) + (n - k) + (m - k) + 1$ *subtractions and divisions, if any.*

To advance from $A^{(k-2)}$ to $A^{(k)}$, the corresponding two-step method (2.12) uses

$3(m - k)(n - k) + 4(n - k) + 2(m - k + 1) + 2$ *multiplications,*

$2(m - k)(n - k) + 2(n - k) + (m - k + 1) + 1$ *additions or subtractions,*

and

$(m - k)(n - k) + 2(n - k) + (m - k + 1) + 1$ *divisions.*

Algorithm (2.11) uses $2(n - k) + 1$ divisions less and one multiplication more. Algorithm (2.8) uses no divisions at all.

For the fraction-producing algorithms of Section III, the algorithms (3.1) and (3.2) need

$2(m - k)(n - k) + (n - k) + (m - k) + 1$ *multiplications,*

$2(m - k)(n - k) + (n - k) + (m - k) + 1$ *subtractions,*

and

$2(m - k) + 1$ *divisions*

each to transform $A^{(k-2)}$ into $A^{(k)}$. Algorithm (3.3) needs no multiplications, the same number of subtractions, but

$2(m - k)(n - k) + (n - k) + 3(m - k) + 2$ *divisions.*

Thus, for large mn, the proportions of the number of multiplications in the one-step integer-preserving to the two-step integer-preserving to the fraction-producing elimination algorithms are about 4:3:2. A comparison of the number of divisions does not carry much weight, since they were introduced to obtain absolute smallest integers and are optional. One can postpone divisions until overflow forces a reduction in the magnitude of the integers.

The integer-preserving algorithms (2.12) and (2.21) can be used to devise an absolutely stable general elimination routine. Assume that through a preliminary transformation the elements a_{ij} became of roughly equal order of magnitude. Then the a_{ij} are truncated and the decimal point removed. The new elements are integers and designated by $a_{ij}^{(0)}$. The matrix $\left(a_{ij}^{(0)}\right)$ is then subjected to (2.12) or (2.21). We note that for noninteger a_{ij}'s, initial truncations can never be avoided on computers that work in the binary system, unless the a_{ij}'s are given as binary numbers, and then only if they can be represented accurately within a given word length. Because (2.12) and (2.21) yield in the general case the absolute smallest possible integers, the largest magnitude of any auxiliary number is of order $max\ det(a_{ij})$. This value can be used to estimate the maximum integer word length. The algorithms of Section III, in contrast, can never be reduced to a routine, free of rounding errors after $\left(a_{ij}^{(0)}\right)$ is given.

If floating-point arithmetic is used, and the a_{ij}'s are given as exact fractions with only a few significant figures relative to the total word length, (2.12) and (2.21) can be expected to yield more accurate solutions than (3.1) or (3.3).

We conclude with the following remark: Algorithm (2.12) was originally developed to provide for expansion of a determinant of general commutative elements (such as polynomials, or elements of an Abelian group, etc.). Its further usefulness in numerical application is most welcome.

V. REMARKS ON PIVOTING

In any single-step (i.e., ordinary) Gaussian-type elimination, pivoting becomes necessary when, in the course of computation, $a_{kk}^{(k-1)} = 0$ in (2.6), (2.9), (2.13), or (3.1).

In the two-step elimination methods, pivoting becomes necessary when $c_0^{(k-2)} = 0$ in (2.12) and (2.21). The fifth line in each of these equations shows that in this case the $a_{ij}^{(k-2)}$ would not participate in the transformation. Thus we have to interchange row k and/or $k - 1$ with rows $i > k$ of $A^{(k-2)}$ until we obtain a $c_0^{(k-2)} \neq 0$. If this is not possible, A is singular. Because single-step elimination is used in transforming row k, the element $a_{k-1,k-1}^{(k-2)}$ must also not be zero. Therefore, it is recommended to add a pivoting algorithm to (2.12) and (2.21). Of several possibilities, one may follow the flow chart given in Fig. 3.

If A is symmetric, it is recommended that corresponding rows and columns are interchanged simultaneously to preserve the symmetry of the transformed matrices $A^{(k)}$.

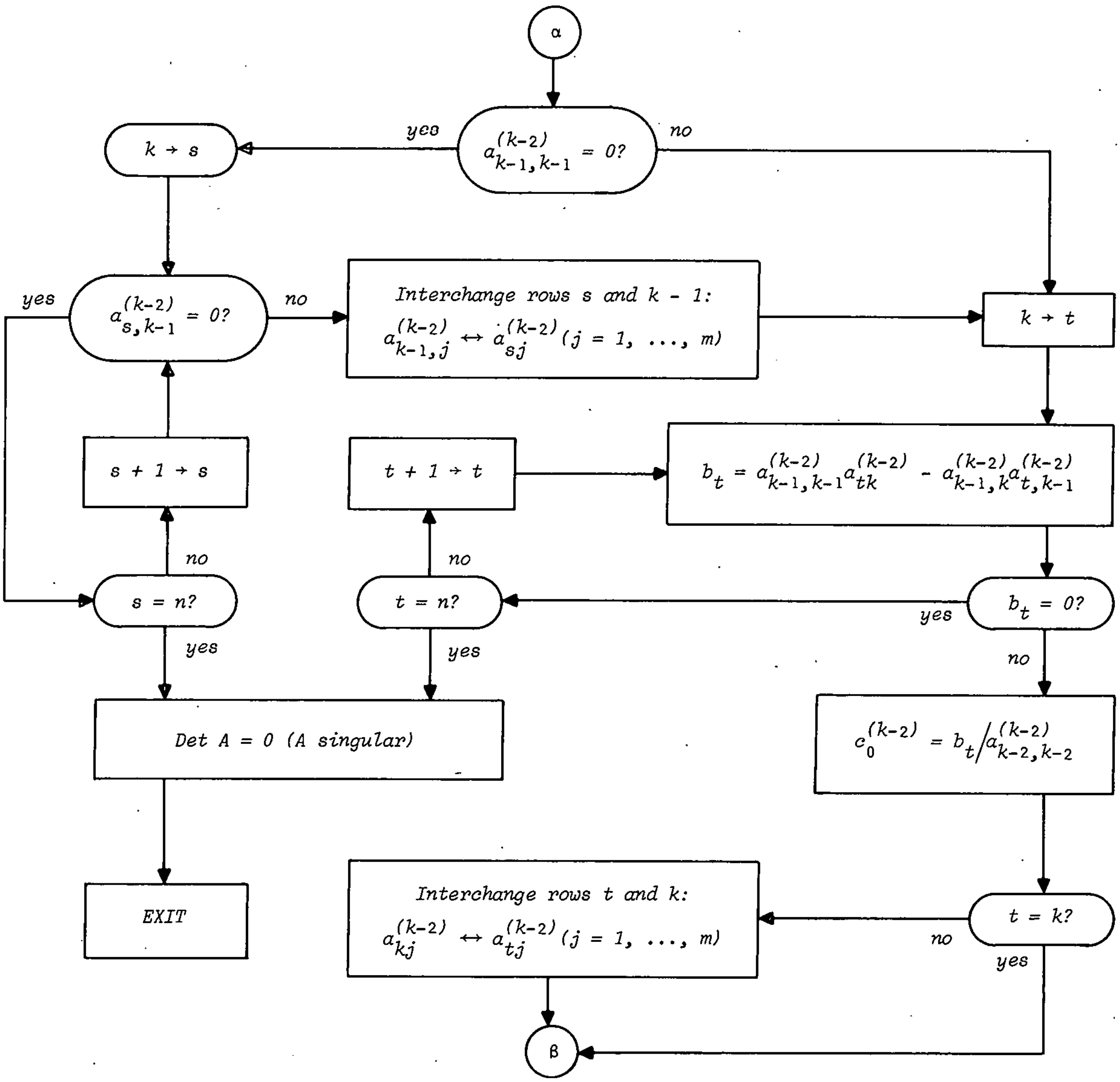

Fig. 3. Pivot-searching Subroutine to Algorithms (2.12) and (2.21), Replacing Box (α) → $c_0^{(k-2)}$ → (β) in Figs. 1 and 2

ACKNOWLEDGMENT

The author wishes to express his gratitude to Burton S. Garbow and William J. Cody for reading the manuscript and offering many suggestions for improving the text. Besides the careful reading of the manuscript, Burt Garbow has calculated several examples and is preparing general codes based on algorithms (2.12) and (2.21).

In going through Muir's five volumes of "The Theory of Determinants," the author could find no reference to a multistep approach in elimination methods as introduced here. Readers are invited to let the author know of any related publications of which they may have any knowledge.