MATHEMATICS OF COMPUTATION, VOLUME 27, NUMBER 124, OCTOBER 1973

An Algorithm for the Exact Reduction of a Matrix to Frobenius Form Using Modular Arithmetic. I

By Jo Ann Howell*

Abstract. This paper is in two parts. Part I contains a description of the Danilewski algorithm for reducing a matrix to Frobenius form using rational arithmetic. This algorithm is modified for use over the field of integers modulo p. The modified algorithm yields exact integral factors of the characteristic polynomial. A description of the single-modulus algorithm is given. Part II contains a description of the multiple-modulus algorithm. Since different moduli may yield different factorizations, an algorithm is given for determining which factorizations are not correct factorizations over the integers of the characteristic polynomial.

A. The Danilewski Method

1. Introduction. It is well known that the Danilewski method (Danilewski [1937]) for reducing a matrix A to Frobenius form

$$F = \begin{bmatrix} F_1 & & & \\ & F_2 & * & \\ & & \ddots & \\ & O & & F_l \end{bmatrix} \tag{1.1}$$

where each diagonal block has the form

$$F_i = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & b_{r_i}^{(i)} \\ 1 & 0 & 0 & \cdots & 0 & b_{r_i-1}^{(i)} \\ 0 & 1 & 0 & \cdots & 0 & b_{r_i-2}^{(i)} \\ \cdot & \cdot & \cdot & \cdots & \cdot & \cdot \\ 0 & 0 & 0 & \cdots & 1 & b_1^{(i)} \end{bmatrix} \tag{1.2}$$

is numerically unstable (Frank [1958]). Several attempts have been made to reduce the inaccuracies by using multiple-precision arithmetic and pivoting for size (Chartres [1964]), (Hansen [1963]). These variations yield a Frobenius form much more accurately than previously reported. However, it has been shown that because of the

Received August 22, 1972.

AMS (MOS) subject classifications (1970). Primary 15A21, 68A10; Secondary 10A10, 12C05, 15A36.

Key words and phrases. Modular arithmetic, residue arithmetic, modulus, Frobenius form, Danilewski method, characteristic polynomial, similarity transformation, prime number, Chinese Remainder Theorem.

* Work on this paper was supported in part by NSF Grant GP-23655 and by U. S. Army Research Office (Durham) Grant DA-ARO(D)-31-124-72-G34 at the University of Texas at Austin.

ill-condition of the Frobenius form of a matrix, the Danilewski method and its variations usually prove unsatisfactory for determining eigenvalues (Wilkinson [1965, pp. 405–411]). Even small errors in the diagonal blocks, F_i, may lead to catastrophic errors in the eigenvalues.

Owing to the fact that a Frobenius form** of a matrix does give us the characteristic polynomial (or a factorization of it over the integers) and some information on the derogatory nature of the matrix, this condensed form is still of some interest to us. For matrices arising from damped mechanical or electrical systems it is common for the Frobenius form to be well-conditioned (Wilkinson [1965, p. 482]).

It is for these reasons that we describe here a modification of the Danilewski method with which we can reduce a matrix to Frobenius form exactly, that is, compute the F_i exactly without the use of multiple-precision arithmetic or pivoting for size. This algorithm is described briefly by Slotnick [1963, pp. 4-42–4-46]. Since this modification uses modular (or residue) arithmetic, it is applicable only to integral matrices. This restriction is not serious, however, since fixed-word-length computers store only rational numbers which can be scaled to integer form. We observe that if the Frobenius form of the matrix A is

$$F_A = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & x_n \\ 1 & 0 & 0 & \cdots & 0 & x_{n-1} \\ 0 & 1 & 0 & \cdots & 0 & x_{n-2} \\ \cdot & \cdot & \cdot & \cdots & \cdot & \cdot \\ 0 & 0 & 0 & \cdots & 1 & x_1 \end{bmatrix},$$

then the Frobenius form of the scaled matrix $k \cdot A$ is given by

$$F_{kA} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & k^n x_n \\ 1 & 0 & 0 & \cdots & 0 & k^{n-1} x_{n-1} \\ 0 & 1 & 0 & \cdots & 0 & k^{n-2} x_{n-2} \\ \cdot & \cdot & \cdot & \cdots & \cdot & \cdot \\ 0 & 0 & 0 & \cdots & 1 & k x_1 \end{bmatrix}.$$

Hence, given F_{kA} and k, we can compute F_A.

The residue arithmetic algorithm which is analogous to the Danilewski method can be performed using either single-modulus or multiple-modulus residue arithmetic, and examples are given for both cases.

We take advantage of the fact that the integers modulo a prime form a finite field, GF(p). Thus, all the theorems relative to matrices and polynomials over a field can be utilized in describing the algorithm. Using the modular arithmetic algorithm, we reduce a matrix to Frobenius form by means of similarity transformations over GF(p). The Frobenius form obtained using modular arithmetic is congruent modulo p to the Frobenius form obtained using rational arithmetic. Thus, from the blocks along the diagonal of the Frobenius form we obtain (using the Chinese Remainder Theorem) the exact multiple-precision coefficients of the characteristic polynomial or of its factors over the integers.

** The nonuniqueness of this form (and hence of the factors) is discussed below.

A related algorithm for obtaining the characteristic polynomial of a matrix is described by McClellan [1971]. However, no comparisons are made here.

This paper begins with a description of the reduction over a field $\mathfrak{F}$ to Frobenius form. In Chapter B, the single-modulus algorithm for the reduction over GF(p) of a matrix to Frobenius form is described, and examples are given. The multiple-modulus algorithm is described in Part II, Chapter C. It is shown that different moduli may yield different factorizations of the characteristic polynomial and in Section 8 a theorem is proved which gives an algorithm for determining which factorizations are not correct factorizations over the integers. Bounds are given in Chapter D for the number of moduli required to guarantee that the coefficients can be reconstructed using the Chinese Remainder Theorem. Examples which illustrate the algorithm are given in Section 12, and numerical results from a computer program are in Section 13.

If $p(\lambda)$ is a polynomial in λ with coefficients over a field $\mathfrak{F}$, we denote this by $p(\lambda) \in \mathfrak{F}(\lambda)$. Also, let $\mathfrak{M}(\mathfrak{F})$ denote the set of matrices with elements over $\mathfrak{F}$, and $\mathfrak{M}(\mathfrak{F}(\lambda))$ denote the set of matrices with elements over $\mathfrak{F}(\lambda)$.

2. The Similarity Transformations Over a Field, $\mathfrak{F}$. Using the Danilewski method we transform an $n \times n$ matrix $A \in \mathfrak{M}(\mathfrak{F})$ by means of similarity transformations over $\mathfrak{F}$, into a matrix $F \in \mathfrak{M}(\mathfrak{F})$, which is in the form (1.1). The elements in the last column of the F_i are coefficients of the characteristic polynomial for F_i,

$$f_i(\lambda) = (-1)^{r_i}[\lambda^{r_i} - b_1^{(i)}\lambda^{r_i-1} - \cdots - b_{r_i-2}^{(i)}\lambda^2 - b_{r_i-1}^{(i)}\lambda - b_{r_i}^{(i)}], \tag{2.1}$$

where $f_i(\lambda) \in \mathfrak{F}(\lambda)$. Thus, when $l = 1$, the characteristic polynomial for F is

$$f(\lambda) = f_1(\lambda) = f_l(\lambda). \tag{2.2}$$

Since A and F are similar over $\mathfrak{F}$, then $f(\lambda)$ is also the characteristic polynomial for A. Thus, by using the Danilewski method, we can compute the characteristic polynomial (or a factorization of it) for the matrix A.

The matrix F is obtained after a *finite* number of similarity transformations of the form

$$A_{k+1} = J_k^{-1} A_k J_k \qquad (k = 0, 1, 2, \cdots, M) \tag{2.3}$$

where $A_0 = A$ and J_k, $A_k \in \mathfrak{M}(\mathfrak{F})$. It is recommended by both Hansen [1963] and Wilkinson [1965, p. 409] that the computation be broken into two stages. During the first stage, the matrix A is reduced to Hessenberg form. Wilkinson has shown that for this step of the algorithm, single-precision arithmetic is usually sufficient. In the second stage, the Hessenberg matrix is further reduced to Frobenius form. It is during this stage that we generally need to work in higher precision arithmetic. (See, for example, Chartres [1964].)

Since stability considerations are of no concern in the modified algorithm which uses modular arithmetic (all arithmetic is exact), and since we wish to discuss the similarities and differences between the algorithm over the rational number field Q and the algorithm over GF(p), we shall assume that the computation is *not* broken into two stages.*** We now discuss the similarity transformations.

*** Further advantages in combining the two stages are mentioned below.

We shall assume here that the algorithm is carried out using *elementary* similarity transformations, since analogous transformations will be used in carrying out the modified algorithm. These transformations consist of one of the following three types of operations:

(a) Interchange of rows i and j (or columns i and j);

(b) Multiplication of row i (or column i) by a nonzero constant K;

(c) Addition to the ith row of an arbitrary multiple K of row j (and the analogous operation on columns).

Thus the J_k can be assumed to be products of *elementary matrices*, which are obtained from the identity matrix by performing one of the above operations on it. We denote these elementary matrices by E_{ij}, $E_i(K)$, and $E_{ij}(K)$, respectively.

In order to save arithmetic, we first reduce the matrix A_0 to the form

$$D = \begin{bmatrix} D_1 & & & * \\ & D_2 & & \\ & & \ddots & \\ & O & & D_l \end{bmatrix} \tag{2.4}$$

where each diagonal block D_i is of the form

$$D_i = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & y_{r_i}^{(i)} \\ d_{21}^{(i)} & 0 & 0 & \cdots & 0 & y_{r_i-1}^{(i)} \\ 0 & d_{32}^{(i)} & 0 & \cdots & 0 & y_{r_i-2}^{(i)} \\ \cdot & \cdot & \cdot & \cdots & \cdot & \cdot \\ 0 & 0 & 0 & \cdots & d_{r_i,r_i-1}^{(i)} & y_1^{(i)} \end{bmatrix}. \tag{2.5}$$

The coefficients $b_j^{(i)}$ of the characteristic polynomial of D_i,

$$f_i(\lambda) = (-1)^{r_i}[\lambda^{r_i} - b_1^{(i)}\lambda^{r_i-1} - \cdots - b_{r_i-2}^{(i)}\lambda^2 - b_{r_i-1}^{(i)}\lambda - b_{r_i}^{(i)}],$$

are thus given by

$$b_1^{(i)} = y_1^{(i)} \tag{2.6}$$

and

$$b_j^{(i)} = y_j^{(i)} \prod_{k=r_i-j+1}^{r_i-1} d_{k+1,k}^{(i)} \qquad (j = 2, \cdots, r_i). \tag{2.7}$$

(These products usually must be computed using double-precision arithmetic to reduce error.) Hence, replacing the $y_j^{(i)}$ by the $b_j^{(i)}$ and replacing the $d_{j+1,j}^{(i)}$ by unity completes the computation of the F_i of the form (1.1).

The reduction of A_0 to the form (2.4) requires at most $n - 1$ transformations of the form (2.3). Each transformation changes a matrix A_k into a matrix A_{k+1} in which there is an additional column with zeros everywhere except at the pivotal position (the $(j + 1, j)$ position). The columns must be annihilated from left to right in order not to destroy zeros produced by previous transformations. We note that since each transformation produces zeros in *all* the elements in one column except the first subdiagonal element (the pivotal element), then the inverse transformation modifies

only the elements in a *single* column. We now describe the transformations which produce the columns of zeros.

After the first transformation we have

$$A_1 = J_0^{-1} A_0 J_0 = \left[\begin{array}{c|ccc} 0 & x & \cdots & x \\ x & x & \cdots & x \\ \hline 0 & & & \\ \vdots & & * & \\ 0 & & & \end{array}\right], \tag{2.8}$$

where

$$J_0^{-1} = \left[\begin{array}{cc|c} 1 & u_{11} & \\ 0 & 1 & \\ \hline 0 & u_{31} & \\ \vdots & \vdots & I_{n-2} \\ 0 & u_{n1} & \end{array}\right], \tag{2.9}$$

and

$$u_{i1} = -a_{i1}^{(0)} \cdot a_{21}^{(0)-1} \qquad (i = 1, 3, 4, \cdots, n). \tag{2.10}$$

The case in which a pivotal element is equal to zero is discussed in Section 3.

The $(j+1)$st transformation produces

$$A_{j+1} = J_j^{-1} A_j J_j = J_j^{-1} \cdots J_0^{-1} A_0 J_0 \cdots J_j = [F' \mid *], \tag{2.11}$$

where F' is an $n \times (j+1)$ submatrix with zeros everywhere except on the first subdiagonal, and where

$$J_j^{-1} = \left[\begin{array}{c|c|c} & u_{1,j+1} & \\ I_{j+1} & \vdots & \\ & u_{j+1,j+1} & \\ \hline 0 \cdots 0 & 1 & 0 \cdots 0 \\ \hline & u_{j+3,j+1} & \\ & \vdots & \\ \multicolumn{1}{c|}{\raisebox{1.5ex}{0}} & u_{n,j+1} & \raisebox{1.5ex}{I_{n-j-2}} \end{array}\right] \tag{2.12}$$

and

$$u_{i,j+1} = -a_{i,j+1}^{(j)} \cdot a_{j+2,j+1}^{(j)-1} \qquad (i = 1, \cdots, j+1, j+3, \cdots, n). \tag{2.13}$$

Finally, at the $(n-1)$st step, if no pivots are equal to zero (if $l = 1$ in (2.4)), we have a matrix in the form (2.4):

$$A_{n-1} = J_{n-2}^{-1} A_{n-2} J_{n-2} = J_{n-2}^{-1} \cdots J_0^{-1} A_0 J_0 \cdots J_{n-2}$$

$$= \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & x \\ x & 0 & 0 & \cdots & 0 & x \\ 0 & x & 0 & \cdots & 0 & x \\ \cdot & \cdot & \cdot & \cdots & \cdot & \cdot \\ 0 & 0 & 0 & \cdots & 0 & x \\ 0 & 0 & 0 & \cdots & x & x \end{bmatrix}, \tag{2.14}$$

where

$$J_{n-2}^{-1} = \left[\begin{array}{ccc|c} & & & u_{1,n-1} \\ & I_{n-1} & & \vdots \\ & & & u_{n-1,n-1} \\ \hline 0 & \cdots & 0 & 1 \end{array}\right] \tag{2.15}$$

and

$$u_{i,n-1} = -a_{i,n-1}^{(n-2)} \cdot a_{n,n-1}^{(n-2)-1} \qquad (i = 1, \cdots, n-1). \tag{2.16}$$

To complete the reduction to (1.1) we carry out the operations described in (2.6) and (2.7). This corresponds to performing a similarity transformation on the D_i, producing

$$F_i = P_i^{-1} D_i P_i, \tag{2.17}$$

where

$$P_i^{-1} = \begin{bmatrix} 1 & & & & \\ & (d_{21}^{(i)})^{-1} & & & \\ & & (d_{21}^{(i)} d_{32}^{(i)})^{-1} & & \\ & & & \ddots & \\ & & & & (\prod_{k=1}^{r_i-1} d_{k+1,k}^{(i)})^{-1} \end{bmatrix}. \tag{2.18}$$

3. The Vanishing Pivot. Clearly, if a pivotal element $a_{j+1,j}^{(j-1)}$ is small in magnitude with respect to other elements in the column, then we can expect excessive roundoff errors to occur. Thus, by searching through the elements a_{ij} $(i = j + 2, \cdots, n)$ for the element of largest magnitude, say a_{kj}, then interchanging rows k and $j + 1$, we can pivot a relatively large element into the $(j + 1, j)$ position before annihilating column j. This corresponds to premultiplying A_{j-1} by the elementary matrix $E_{k,j+1}$. Then, to complete the similarity transformation, A_{j-1} must be postmultiplied by $E_{k,j+1}$. This interchanges columns k and $j + 1$.

We should point out that the reason the search for a nonzero pivot is *not* made among the elements $a_{ij}^{(j-1)}$ $(i = 1, \cdots, j)$ is that pre and postmultiplying A_{j-1} by E_{ij} $(i < j + 1)$ destroys zeros produced by previous transformations. Thus, the

pivotal element may be small with respect to the elements a_{ij} $(i = 1, \cdots, j)$, and hence $u_{ij} = -a_{ij}^{(j-1)} \cdot a_{j+1,j}^{(j-1)-1}$ $(i = 1, \cdots, j)$ may be quite large. This is the cause of the numerical instability in the Danilewski method.

If a pivot vanishes and no nonzero pivot can be found among the elements a_{ij} $(i = j + 2, \cdots, n)$, then we partition the matrix A_{j-1} into blocks, as follows, and apply the algorithm to the $(n - j) \times (n - j)$ submatrix H_1:

$$\begin{bmatrix} D_1 & * \\ O & H_1 \end{bmatrix},$$

where D_1 is a $j \times j$ submatrix which is in Frobenius form except for the elements on the first subdiagonal (not yet reduced to unity), and whose characteristic polynomial is $f_1(\lambda)$. If partitioning occurs when applying the algorithm to H_1, we obtain

$$\begin{bmatrix} D_1 & & * \\ & D_2 & \\ O & & H_2 \end{bmatrix},$$

where the characteristic polynomial of D_2 is $f_2(\lambda)$. Proceeding in this manner, we obtain the block triangular matrix (2.4). If the pivotal element is not zero, but less than some threshold value ϵ, say $2^{-t}\ ||A||_E$, where t is the number of bits in the mantissa of the floating-point computer word, then we can replace the pivot by zero and partition the matrix as described above. We note that if A is derogatory, then this partitioning into a block triangular matrix must occur.

Other descriptions of the Danilewski method and its variations are given in Wilkinson [1965, pp. 405–407], Householder and Bauer [1959], Householder [1964, pp. 156–158], and Wayland [1945].

B. The Modified Danilewski Method

4. Introduction. Let a be an integer and p a prime. Then $|a|_p$ denotes the unique integer which is in the interval $[-[p/2], [p/2]]$ and which is congruent to a modulo p. We say that $|a|_p$ is the *residue of a modulo p*. This residue is also called the *symmetric residue* since it is in an interval which is symmetric about zero. Similarly, we can apply the notation $|\cdot|_p$ to matrices or polynomials to indicate that the elements of a matrix or the coefficients of a polynomial are reduced modulo p and lie in the symmetric interval. The inverse modulo p of an integer a or of a matrix A, when the inverses exist, is denoted by $a^{-1}(p)$ or $A^{-1}(p)$, respectively.

In this section we describe an algorithm which uses similarity transformations over GF(p) to reduce an $n \times n$ integral matrix A to Frobenius form and obtain exact integral factors of the characteristic polynomial of A. The algorithm is based on the fact that if A has as its characteristic polynomial

$$f(\lambda) = \det(A - \lambda I) = f_1(\lambda) \cdots f_l(\lambda), \tag{4.1}$$

where $f_i(\lambda)$ is an integral polynomial given by

$$f_i(\lambda) = (-1)^{r_i'}(\lambda^{r_i'} - b_1^{(i)}\lambda^{r_i'-1} - \cdots - b_{r_i'-1}^{(i)}\lambda - b_{r_i'}^{(i)}), \tag{4.2}$$

then we define the *characteristic polynomial modulo p* of A to be

$$|f(\lambda)|_p = |\det(A - \lambda I)|_p, \tag{4.3}$$

where $|f(\lambda)|_p \in \mathrm{GF}(p)(\lambda)$.

Thus, if we compute a bound † β, where

$$\beta \geqq \max_{i,j} |b_j^{(i)}| \tag{4.4}$$

and if we choose p so that ‡

$$p \geqq 2 \cdot \beta, \tag{4.5}$$

and finally if we compute the $f_i(\lambda)$ using modular arithmetic, then

$$|f_i(\lambda)|_p = f_i(\lambda) \tag{4.6}$$

and

$$|f(\lambda)|_p = ||f_1(\lambda)|_p \cdots |f_l(\lambda)|_p|_p = |f_1(\lambda) \cdots f_l(\lambda)|_p. \tag{4.7}$$

Our objective, then, is to reduce a matrix A to the form

$$F^{(p)} = \begin{bmatrix} F_1^{(p)} & & & * \\ & F_2^{(p)} & & \\ & & \ddots & \\ & O & & \ddots \\ & & & F_{l'}^{(p)} \end{bmatrix}, \tag{4.8}$$

using similarity transformations over GF(p), where $l' = l$ and

$$F_i^{(p)} = |F_i|_p \qquad (i = 1, \cdots, l). \tag{4.9}$$

We now describe a method for computing the $F_i^{(p)}$ and, hence, also $|f(\lambda)|_p$, using modular arithmetic.

5. The Similarity Transformations over GF(p). If we can reduce $|A_p|$ to the form

$$C = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & x_n \\ 1 & 0 & 0 & \cdots & 0 & x_{n-1} \\ 0 & 1 & 0 & \cdots & 0 & x_{n-2} \\ \cdot & \cdot & \cdot & \cdots & \cdot & \cdot \\ 0 & 0 & 0 & \cdots & 0 & x_2 \\ 0 & 0 & 0 & \cdots & 1 & x_1 \end{bmatrix}, \tag{5.1}$$

where $C = |S^{-1}(p)\ |A|_p\ S|_p$, then A and C are similar over GF(p), and the characteristic polynomial modulo p of C, and hence of $|A|_p$ is

$$|f(\lambda)|_p = |(-1)^n(\lambda^n - x_1\lambda^{n-1} - \cdots - x_{n-1}\lambda - x_n)|_p. \tag{5.2}$$

† Methods for computing β will be discussed in Section 11.

‡ The 2 is necessary because we are using the symmetric residue system.

Therefore, since the elements of C are reduced modulo p, we have, from (4.2),

$$x_i = |b_i^{(1)}|_p. \tag{5.3}$$

This assumes that no pivots vanish in reducing $|A|_p$ to the form (5.1).

We now consider the problem of reducing $|A|_p$ to the form (5.1). This is accomplished in a finite number of similarity transformations over GF(p), or *similarity transformations modulo p*, of the form

$$A_{k+1}^{(p)} = |A_{k+1}|_p = |J_k^{(p)-1}(p) A_k J_k^{(p)}|_p, \tag{5.4}$$

where

$$A_0^{(p)} = [a_{ij}^{(0)(p)}] = |A|_p \tag{5.5}$$

and

$$A_k^{(p)} = [a_{ij}^{(k)(p)}]. \tag{5.6}$$

Thus, $[a_{ij}^{(k)(p)}]$ is the matrix obtained at the kth step of the reduction of $|A|_p$ to the form (5.1) using modulo p arithmetic. In general $[a_{ij}^{(k)(p)}] \neq [a_{ij}^{(k)}]$. However, in order to simplify the notation in the discussion which follows, we omit the superscript p on the elements $a_{ij}^{(k)}$ of $A_k^{(p)}$ where it is clear that we are discussing the modular arithmetic algorithm and not the algorithm which uses rational arithmetic.

The transformations in (5.4) consist of the following three types of operations which are called *elementary operations modulo p*:

(a) Interchange of rows i and j (or columns i and j);

(b) Multiplication of row i (or column i) by a nonzero constant k, where $(k, p) = 1$, followed by reduction modulo p;

(c) Addition to the ith row of an arbitrary multiple k, of row j, followed by reduction modulo p (or the analogous operation on columns).

An *elementary matrix modulo p* is a matrix in $\mathfrak{M}(\mathrm{GF}(p))$ obtained from the identity matrix by performing one of the above operations on it. We shall denote matrices of these kinds by $|E_{ij}|_p$, $|E_i(K)|_p$, and $|E_{ij}(K)|_p$, respectively. Thus, we shall let the $J_k^{(p)}$ be products of elementary matrices modulo p.

We first reduce the matrix $A_0^{(p)}$ to the form

$$D^{(p)} = \begin{bmatrix} D_1^{(p)} & & & * \\ & D_2^{(p)} & & \\ & & \ddots & \\ & O & & D_{l'}^{(p)} \end{bmatrix} \tag{5.7}$$

where each diagonal block $D_i^{(p)}$ is in Frobenius form except for the subdiagonal elements which are nonzero but not yet reduced to unity. The reduction of $A_0^{(p)}$ to this form requires at most $n - 1$ steps of the form

$$A_{k+1}^{(p)} = |J_k^{(p)-1}(p) A_k^{(p)} J_k^{(p)}|_p, \tag{5.8}$$

where $J_k^{(p)}$ is a product of elementary matrices modulo p and $A_0^{(p)} = |[a_{ij}^{(0)}]|_p = |A|_p$. Each transformation changes a matrix $A_k^{(p)}$ into a matrix $A_{k+1}^{(p)}$ in which there is an additional column with zeros everywhere except at the pivotal position.

At the $(j+1)$st step of the reduction (if $a_{i+2,i+1}^{(j)} \not\equiv 0 \pmod{p}$), the similarity transformation over GF(p) produces

$$\begin{aligned} A_{i+1}^{(p)} &= |J_i^{(p)-1}(p) A_i^{(p)} J_i^{(p)}|_p \\ &= |J_i^{(p)-1}(p) \cdots J_0^{(p)-1}(p) A_0^{(p)} J_0^{(p)} \cdots J_i^{(p)}|_p \\ &= [F'^{(p)} \mid *], \end{aligned} \tag{5.9}$$

where $F'^{(p)}$ is an $n \times (j+1)$ submatrix with zeros everywhere except on the first subdiagonal, and where

$$J_i^{(p)-1}(p) = \left[\begin{array}{c|c|c} & \mu_{1,i+1}^{(p)} & \\ I_{i+1} & \vdots & \mathbf{O} \\ & \mu_{i+1,i+1}^{(p)} & \\ \hline 0 \cdots 0 & 1 & 0 \cdots 0 \\ \hline & \mu_{i+3,i+1}^{(p)} & \\ \mathbf{O} & \vdots & I_{n-i-2} \\ & \mu_{n,i+1}^{(p)} & \end{array}\right] \tag{5.10}$$

and

$$\mu_{i,i+1}^{(p)} = |-a_{i,i+1}^{(j)} \cdot a_{i+2,i+1}^{(j)-1}(p)|_p \qquad (i = 1, \cdots, j+1, j+3, \cdots, n). \tag{5.11}$$

(The case in which a pivot vanishes is discussed in Section 6.)

Finally, at the $(n-1)$st step, if no pivots are congruent to zero (if $l' = 1$ in (5.7)), we have a matrix in the form (5.7):

$$\begin{aligned} A_{n-1}^{(p)} &= |J_{n-2}^{(p)-1}(p) A_{n-2}^{(p)} J_{n-2}^{(p)}|_p \\ &= |J_{n-2}^{(p)-1}(p) \cdots J_0^{(p)-1}(p) A_0^{(p)} J_0^{(p)} \cdots J_{n-2}^{(p)}|_p \\ &= \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & x \\ x & 0 & 0 & \cdots & 0 & x \\ 0 & x & 0 & \cdots & 0 & x \\ \cdot & \cdot & \cdot & \cdots & \cdot & \cdot \\ 0 & 0 & 0 & \cdots & 0 & x \\ 0 & 0 & 0 & \cdots & x & x \end{bmatrix}, \end{aligned} \tag{5.12}$$

where

$$J_{n-2}^{(p)-1}(p) = \left[\begin{array}{c|c} & \mu_{1,n-1}^{(p)} \\ I_{n-1} & \vdots \\ & \mu_{n-1,n-1}^{(p)} \\ \hline 0 \cdots 0 & 1 \end{array}\right] \tag{5.13}$$

and

$$\mu_{i,n-1}^{(p)} = |-a_{i,n-1}^{(n-2)} \cdot a_{n,n-1}^{(n-2)-1}(p)|_p \qquad (i = 1, \cdots, n-1). \tag{5.14}$$

To complete the reduction from the form (5.7), where

$$D_i^{(p)} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & y_{r_i'}^{\prime(i)} \\ d_{21}^{\prime(i)} & 0 & 0 & \cdots & 0 & y_{r_i'-1}^{\prime(i)} \\ 0 & d_{32}^{\prime(i)} & 0 & \cdots & 0 & y_{r_i'-2}^{\prime(i)} \\ \cdot & \cdot & \cdot & \cdots & \cdot & \cdot \\ 0 & 0 & 0 & \cdots & d_{r_i',r_i'-1}^{\prime(i)} & y_1^{\prime(i)} \end{bmatrix}, \tag{5.15}$$

to the form (4.8), where

$$F_i^{(p)} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & b_{r_i'}^{\prime(i)} \\ 1 & 0 & 0 & \cdots & 0 & b_{r_i'-1}^{\prime(i)} \\ 0 & 1 & 0 & \cdots & 0 & b_{r_i'-2}^{\prime(i)} \\ \cdot & \cdot & \cdot & \cdots & \cdot & \cdot \\ 0 & 0 & 0 & \cdots & 1 & b_1^{\prime(i)} \end{bmatrix} \tag{5.16}$$

we compute

$$|b_1^{\prime(i)}|_p = |y_1^{\prime(i)}|_p \tag{5.17}$$

and

$$|b_j^{\prime(i)}|_p = \left| y_j^{\prime(i)} \prod_{k=r_i'-j+1}^{r_i'-1} d_{k+1,k}^{\prime(i)} \right|_p, \qquad j = 2, \cdots, r_i'. \tag{5.18}$$

This corresponds to performing a similarity transformation modulo p on $D_i^{(p)}$, producing

$$F_i^{(p)} = |P_i^{(p)-1}(p) D_i^{(p)} P_i^{(p)}|_p, \tag{5.19}$$

where

$$P_i^{(p)-1}(p) = \begin{bmatrix} 1 & & & & \\ & (d_{21}^{\prime(i)})^{-1}(p) & & & \mathbf{O} \\ & & (d_{21}^{\prime(i)} d_{32}^{\prime(i)})^{-1}(p) & & \\ & \mathbf{O} & & \ddots & \\ & & & & (\prod_{k=1}^{r_i'-1} d_{k+1,k}^{\prime(i)})^{-1}(p) \end{bmatrix}. \tag{5.20}$$

The following is an algorithm for reducing a matrix to the form (5.7).

Algorithm I. *Reduction of a Matrix A to the Form* (5.7) (*Single-Modulus Algorithm*).

Input: An $n \times n$ matrix $|A|_p \in \mathfrak{M}(\mathrm{GF}(p))$.

Output: An $n \times n$ matrix $D^{(p)} \in \mathfrak{M}(\mathrm{GF}(p))$ in the form (5.7), $l', r_i'^{(p)}$ $(i = 1, \cdots, l')$.

(1) Set $i \leftarrow 2$, $j \leftarrow 1$, $ibl \leftarrow 1$, $l' \leftarrow 0$, $jp \leftarrow 0$.

(2) If $a_{i,j} \equiv 0 \pmod p$, go to (10).

(3) Set $ip \leftarrow a_{i,j}^{-1}(p)$.

(4) [Produce zeros in column j (premultiply by J_{i-1}).] For $k = ibl, \cdots, n$ $(k \neq i)$

and $t = i, \cdots, n$, set $a_{k,i} \leftarrow |a_{k,i} \cdot ip|_p$, $a_{k,t} \leftarrow |-a_{k,i} \cdot ip \cdot a_{i,t} + a_{k,t}|_p$.

(5) If $j > n - 2$, go to (7).

(6) [Postmultiply by J_{j-1}.] For $k = ibl, \cdots, n$, set $a_{k,j+1} \leftarrow |a_{k,j+1} + \sum_{t=j+2}^{n} a_{t,i} \cdot a_{k,t}|_p$.

(7) For $k = ibl + 1, \cdots, n$, set $a_{k,j+1} \leftarrow |a_{k,j+1} + a_{k-1,j} \cdot a_{k,k-1}|_p$.

(8) For $k = 1, \cdots, n$ $(k \neq i)$, set $a_{k,i} \leftarrow 0$.

(9) If $j = n - 1$, go to (18); otherwise go to (17).

(10) If $j = n - 1$, set $l' \leftarrow l' + 1$, $r_{l'}^{(p)} \leftarrow j + 1 - ibl$, $ibl \leftarrow j + 1$, and go to (18).

(11) [Look for a nonzero pivot.] For $k = j + 2, \cdots, n$, test to see if $a_{k,j} \not\equiv 0 \pmod{p}$. Let $ii = \min_{j+2 \leq k \leq n} k$ such that $a_{k,j} \not\equiv 0 \pmod{p}$ (if a nonzero element exists), and go to (13).

(12) [No nonzero element can be found in column j.] Go to (16).

(13) [Interchange rows ii and $j + 1$.] For $k = j, \cdots, n$, set temp $\leftarrow a_{ii,k}$, $a_{ii,k} \leftarrow a_{j+1,k}$, $a_{j+1,k} \leftarrow$ temp.

(14) [Interchange columns ii and $j + 1$.] For $k = ibl, \cdots, n$, set temp $\leftarrow a_{k,ii}$, $a_{k,ii} \leftarrow a_{k,j+1}$, $a_{k,j+1} \leftarrow$ temp.

(15) Set $jp \leftarrow jp + 1$, $\text{pivot}_{ip,1}^{(p)} \leftarrow j + 1$, $\text{pivot}_{ip,2}^{(p)} \leftarrow ii$, and go to (3).

(16) [Increment counter for number of blocks in $D^{(p)}$ and compute block size.] Set $l' \leftarrow l' + 1$, $r_{l'}^{(p)} \leftarrow j + 1 - ibl$, $ibl \leftarrow j + 1$.

(17) Set $j \leftarrow j + 1$, $i \leftarrow i + 1$, and go to (2).

(18) Set $l' \leftarrow l' + 1$, $r_{l'}^{(p)} \leftarrow j + 2 - ibl$, $D^{(p)} \leftarrow A$.

(19) *Exit.*

We note that this algorithm takes advantage of the fact that the superdiagonal elements not in one of the $D_i^{(p)}$ are of no interest. Hence, the transformations ignore these elements wherever possible.

The following is an algorithm for reducing a matrix $D^{(p)}$ in the form (5.7) to $F^{(p)}$ in the form (4.8).

Algorithm II. *Reduction of a Matrix $D^{(p)}$ to the Form* (4.8) (*Single-Modulus Algorithm*).

Input: An $n \times n$ matrix $D^{(p)} \in \mathfrak{M}(\text{GF}(p))$, l', $r_i'^{(p)}$ $(i = 1, \cdots, l')$.

Output: An $n \times n$ matrix $F^{(p)} \in \mathfrak{M}(\text{GF}(p))$.

(1) Set $i \leftarrow 0$, $j \leftarrow 0$.

(2) Set $j \leftarrow j + 1$, $i \leftarrow i + r_i'^{(p)}$.

(3) If $r_i'^{(p)} \not\equiv 1$, go to (5).

(4) If $i < n$, go to (2); otherwise set $F^{(p)} \leftarrow D^{(p)}$ and *exit*.

(5) [Produce unity subdiagonal elements.] Set $k \leftarrow i - 1$, mult $\leftarrow d_{i,i-1}^{(p)}$, $d_{i,i-1}^{(p)} \leftarrow 1$.

(6) Set $d_{k,i}^{(p)} \leftarrow |\text{mult} \cdot d_{k,i}^{(p)}|_p$.

(7) If $k = i - r_i'^{(p)} + 1$, go to (4).

(8) Set mult $\leftarrow |\text{mult} \cdot d_{k,k-1}^{(p)}|_p$, $d_{k,k-1}^{(p)} \leftarrow 1$, $k \leftarrow k - 1$, go to (6).

In order to show that $l' = l$ and that $F_i^{(p)} = |F_i|_p$, $i = 1, \cdots, l$, we first need to discuss the vanishing of pivots (the case in which $l' \neq 1$).

6. The Vanishing Pivot. If a pivotal element $a_{j+1,j}^{(j-1)}$ is zero, then a search is made among the elements $a_{ij}^{(j-1)}$ $(i = j + 2, \cdots, n)$ for a nonzero element. If one is found, say a_{kj}, then *rows* k and $j + 1$ are interchanged. This corresponds to premultiplying $A_{j-1}^{(p)}$ by the elementary matrix $|E_{k,j+1}|_p$. Then, to complete the similarity transformation modulo p, $A_{j-1}^{(p)}$ must be postmultiplied by $|E_{k,j+1}|_p$. This interchanges *columns* k

and $j+1$. As in the real arithmetic algorithm, the reason the search for a nonzero pivot is *not* made among the elements $a_{ij}^{(j-1)}$ $(i = 1, \cdots, j)$ or $a_{j+1,k}^{(j-1)}$ $(k = j+1, \cdots, n)$ is that pre and postmultiplying $A_{j-1}^{(p)}$ by $|E_{i,j}|_p$ $(i < j+1)$ or by $|E_{j+1,k}|_p$ $(k < j)$ destroys zeros produced by previous transformations.

In case a nonzero pivot cannot be found, then we partition the matrix $A_{j-1}^{(p)}$ into blocks, as follows, and apply the algorithm to the $(n-j) \times (n-j)$ submatrix $H_1^{(p)}$:

$$\begin{bmatrix} D_1^{(p)} & * \\ O & H_1^{(p)} \end{bmatrix},$$

where $D_1^{(p)}$ is the form (5.15). If partitioning occurs when applying the algorithm to $H_1^{(p)}$, we obtain

$$\begin{bmatrix} D_1^{(p)} & * & \\ & D_2^{(p)} & \\ O & & H_2^{(p)} \end{bmatrix}$$

Proceeding in this manner, we obtain the block triangular matrix (5.7).

An example illustrates the vanishing of pivots in both the rational arithmetic algorithm and the modular arithmetic algorithm.

(6.1) *Example.* (a) *Rational Arithmetic Algorithm.* Let

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 7 & 1 & 0 & 1 \\ 5 & 0 & 0 & 0 \end{bmatrix}.$$

Since the first pivotal element is zero (the (2, 1) element), we must interchange rows 2 and 3 (and columns 2 and 3). If we let

$$J_0^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & \frac{-5}{7} & 0 & 1 \end{bmatrix},$$

we have

$$A_1 = J_0^{-1} I_{23} A_0 I_{23} J_0$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & \frac{-5}{7} & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 7 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & \frac{5}{7} & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 7 & \frac{5}{7} & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{-25}{49} & \frac{-5}{7} & \frac{-5}{7} \end{bmatrix}.$$

The second pivot also vanishes; so, we interchange rows 3 and 4 and columns 3 and 4. Thus,

$$A_2 = J_1^{-1} I_{34} A_1 I_{34} J_1$$

$$= \begin{bmatrix} 1 & 0 & \frac{49}{25} & 0 \\ 0 & 1 & \frac{7}{5} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 7 & \frac{5}{7} & 1 & 1 \\ 0 & \frac{-25}{49} & \frac{-5}{7} & \frac{-5}{7} \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & \frac{-49}{25} & 0 \\ 0 & 1 & \frac{-7}{5} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & \frac{-7}{5} & \frac{-7}{5} \\ 7 & 0 & \frac{-343}{25} & 0 \\ 0 & \frac{-25}{49} & 0 & \frac{-5}{7} \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Since the third pivot vanishes and there is no nonzero element to pivot into its place, we partition the matrix into blocks as shown. Thus,

$$D_1 = \begin{bmatrix} 0 & 0 & \frac{-7}{5} \\ 7 & 0 & \frac{-343}{25} \\ 0 & \frac{-25}{49} & 0 \end{bmatrix} \quad \text{and} \quad D_2 = [0].$$

Hence

$$F_1 = P_1^{-1} D_1 P_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{7} & 0 \\ 0 & 0 & \frac{-7}{25} \end{bmatrix} \begin{bmatrix} 0 & 0 & \frac{-7}{5} \\ 7 & 0 & \frac{-343}{25} \\ 0 & \frac{-25}{49} & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & \frac{-25}{7} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 5 \\ 1 & 0 & 7 \\ 0 & 1 & 0 \end{bmatrix}$$

and

$$F_2 = [0].$$

Therefore

$$\det(A - \lambda I) = (\lambda^3 - 7\lambda - 5)\lambda.$$

(b) *Modular Arithmetic Algorithm.* We let $p = 13$. Interchanging rows 2 and 3 to obtain a nonzero pivot, we obtain

$$A_1^{(13)} = | J_1^{(13)-1}(13) I_{23} A_0^{(13)} I_{23} J_1^{(13)} |_{13}$$

$$= \left\| \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 3 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 7 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -3 & 0 & 1 \end{bmatrix} \right\|_{13} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 7 & -3 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 4 & 3 & 3 \end{bmatrix}.$$

The second pivot vanishes, also. Thus

$$A_2^{(13)} = |J_2^{(13)-1}(13)I_{34}A_1^{(13)}I_{34}J_2|_{13}$$

$$= \left\| \begin{bmatrix} 1 & 0 & 3 & 0 \\ 0 & 1 & 4 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 7 & -3 & 1 & 1 \\ 0 & 4 & 3 & 3 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & -3 & 0 \\ 0 & 1 & -4 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \right\|_{13} = \left[\begin{array}{ccc|c} 0 & 0 & -4 & -4 \\ 7 & 0 & 5 & 0 \\ 0 & 4 & 0 & 3 \\ \hline 0 & 0 & 0 & 0 \end{array}\right].$$

Then,

$$D_1^{(13)} = \begin{bmatrix} 0 & 0 & -4 \\ 7 & 0 & 5 \\ 0 & 4 & 0 \end{bmatrix} \quad \text{and} \quad D_2^{(13)} = [0].$$

Hence,

$$F_1^{(13)} = |P_1^{(13)-1}(13)D_1^{(13)}P_1^{(13)}|_{13}$$

$$= \left\| \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 7 \end{bmatrix} \begin{bmatrix} 0 & 0 & -4 \\ 7 & 0 & 5 \\ 0 & 4 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 2 \end{bmatrix} \right\|_{13}$$

$$= \begin{bmatrix} 0 & 0 & 5 \\ 1 & 0 & -6 \\ 0 & 1 & 0 \end{bmatrix}$$

and

$$F_2^{(13)} = [0].$$

Therefore

$$|\det(A - \lambda I)|_{13} = |(\lambda^3 + 6\lambda - 5)\lambda|_{13}.$$

We see that, in this example, $l' = l = 2$ and $F_i^{(13)} = |F_i|_{13}$, for $i = 1, 2$.

It is possible for different moduli to yield different factorizations of the characteristic polynomial. We illustrate this with an example. Since the factorizations obtained using the modular arithmetic algorithm are related to the vanishing of pivots and the choice of modulus, we exhibit a relationship between the pivots in the rational arithmetic algorithm and the pivots in the modular arithmetic algorithm.

(6.2) *Example.* Let A be the matrix used in the last example:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 7 & 1 & 0 & 1 \\ 5 & 0 & 0 & 0 \end{bmatrix}.$$

Choosing $p = 5$, we interchange rows 2 and 3 and columns 2 and 3, obtaining

$$A_1^{(5)} = |I_{23} A_0^{(5)} I_{23}|_5 = \left[\begin{array}{cc|c|c} 0 & 1 & 0 & 0 \\ 2 & 0 & 1 & 1 \\ \hline 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 \end{array}\right].$$

Thus,

$$D_1^{(5)} = \begin{bmatrix} 0 & 1 \\ 2 & 0 \end{bmatrix}, \quad D_2^{(5)} = [0] \quad \text{and} \quad D_3^{(5)} = [0].$$

Hence,

$$F_1^{(5)} = |P_1^{(5)-1}(5) D_1^{(5)} P_1^{(5)}|_5 = \left|\begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix}\begin{bmatrix} 0 & 1 \\ 2 & 0 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}\right|_5 = \begin{bmatrix} 0 & 2 \\ 1 & 0 \end{bmatrix},$$

and

$$F_2^{(5)} = F_3^{(5)} = [0].$$

We see that by using the modulus 5 we have obtained three blocks of orders 2, 1, and 1, as opposed to two blocks of orders 3 and 1 obtained using $p = 13$.

The following lemma exhibits the relationship between the vanishing of pivots in the rational arithmetic algorithm and in the modular arithmetic algorithm.

(6.3) LEMMA. *Let $A_k = [a_{tq}^{(k)}]$ and $A_k^{(p)} = [a_{tq}'^{(k)}]$ each be the kth matrix in a sequence of matrices obtained in the reduction of a matrix to Hessenberg form using rational arithmetic and arithmetic modulo p, respectively. Then, the statement $a_{k+2,k+1}^{(k)} = 0$ if and only if $a_{k+2,k+1}'^{(k)} = 0$ $(0 \leqq k \leqq j-1)$ implies that, for $i = j+2, \cdots, n$,*

(a) $$a_{i,j+1}^{(j)}\{(a_{j+1,j}^{(j-1)})^{b_1}(a_{j,j-1}^{(j-2)})^{b_2} \cdots (a_{21}^{(0)})^{b_r}\}$$

is an integer and

$$|a_{i,j+1}'^{(j)}(a_{j+1,j}'^{(j-1)})^{b_1} \cdots (a_{21}'^{(0)})^{b_r}|_p = |a_{i,j+1}^{(j)}(a_{j+1,j}^{(j-1)})^{b_1} \cdots (a_{21}^{(0)})^{b_r}|_p$$

for all j, $1 \leqq j \leqq n-2$, where

(i) $b_1 = 2$, $b_s = 1 + 2b_{s-1} + \sum_{k=1}^{s-2} b_k$ *and*

(ii) *the pivots $a_{j+1,j}^{(j-1)}, \cdots, a_{21}^{(0)}$ are the nonzero pivots obtained between step 1 and step j of the rational arithmetic algorithm, and r is the number of such pivots, and*

(b) $$a_{ti}^{(j)}\{(a_{j+1,j}^{(j-1)})^{c_1}(a_{j,j-1}^{(j-2)})^{c_2} \cdots (a_{21}^{(0)})^{c_r}\}$$

is an integer and

$$|a_{ti}'^{(j)}(a_{j+1,j}'^{(j-1)})^{c_1} \cdots (a_{21}'^{(0)})^{c_r}|_p = |a_{ti}^{(j)}(a_{j+1,j}^{(j-1)})^{c_1} \cdots (a_{21}^{(0)})^{c_r}|_p,$$

for $i = j+2, \cdots, n$, and $t = 1, \cdots, n$, where

(i) $c_1 = 1$, $c_s = b_{s-1} + c_{s-1}$, *and*

(ii) *the pivots $a_{j+1,j}^{(j-1)}, \cdots, a_{21}^{(0)}$ and r are defined as above.*

Proof. A lengthy proof is given in Howell [1972, Appendix A].

From the above lemma we have the following:

(6.4) THEOREM. *The statement* $a^{(k)}_{k+2,k+1} = 0$ *if and only if* $a'^{(k)}_{k+2,k+1} = 0$ $(0 \leqq k \leqq j-1)$ *implies that, for* $1 \leqq j \leqq n-2$, $a'^{(j)}_{j+2,j+1} = 0$ *if and only if either*

(a) $|a^{(j)}_{j+2,j+1}|_p = 0$ *or*

(b) $|a^{(j)}_{j+2,j+1}(a^{(j-1)}_{j+1,j})^{b_1}(a^{(j-2)}_{j,j-1})^{b_2} \cdots (a^{(0)}_{21})^{b_r}|_p = 0$,

where the b_i *and* $a^{(i-1)}_{i+1,i}$ *are described in Lemma* (6.3).

Proof. If either

$$|a^{(j)}_{j+2,j+1}|_p = 0 \quad \text{or} \quad |a^{(j)}_{j+2,j+1}(a^{(j-1)}_{j+1,j})^{b_1} \cdots (a^{(0)}_{21})^{b_r}|_p = 0,$$

then

$$|a'^{(j)}_{j+2,j+1}(a'^{(j-1)}_{j+1,j})^{b_1} \cdots (a'^{(0)}_{21})^{b_r}|_p = 0.$$

Since none of $(a'^{(j-1)}_{j+1,j})^{b_1}, \cdots, (a'^{(0)}_{21})^{b_r}$ is zero, we must have

$$a'^{(j)}_{j+2,j+1} = 0.$$

Conversely, if $a'^{(j)}_{j+2,j+1} = 0$, then either

$$|a^{(j)}_{j+2,j+1}|_p = 0 \quad \text{or} \quad |a^{(j)}_{j+2,j+1}(a^{(j-1)}_{j+1,j})^{b_1} \cdots (a^{(0)}_{21})^{b_r}|_p = 0.$$

This completes the proof of the theorem.

From the above theorem we see that if $a'^{(j)}_{j+2,j+1}$ is nonzero for some modulus p, then the same pivot must be nonzero in the rational arithmetic algorithm, provided previous pivots for the two algorithms vanished at the same point. Choosing a large p lessens the chance of having

$$a^{(j)}_{j+2,j+1}(a^{(j-1)}_{j+1,j})^{b_1} \cdots (a^{(0)}_{21})^{b_r} = C \cdot p,$$

where C is an integer not equal to zero. Thus, a large p will increase the probability that $a^{(j)}_{j+2,j+1} = 0$ if and only if $a'^{(j)}_{j+2,j+1} = 0$. Clearly, if we have $a^{(j)}_{j+2,j+1} = 0$ if and only if $a'^{(j)}_{j+2,j+1} = 0$, then

$$r'_i = r_i \tag{6.5}$$

and

$$l' = l. \tag{6.6}$$

Thus, we have shown that if p is a sufficiently large prime number, then the modular arithmetic algorithm produces blocks in Frobenius form which are the same order as corresponding blocks produced by the rational arithmetic algorithm.

We are now prepared to state a relationship between the elements of $F_i^{(p)}$ and the corresponding elements of F_i.

(6.7) LEMMA. *The statement that* $a'^{(k)}_{k+2,k+1} = 0$ *if and only if* $a^{(k)}_{k+2,k+1} = 0$ $(0 \leqq k \leqq j-1,\ 1 \leqq j \leqq n-2)$ *implies that*

$$a^{(j)}_{i,j+1}(a^{(j-1)}_{j+1,j})^{b_1}(a^{(j-2)}_{j,j-1})^{b_1+b_2} \cdots (a^{(1)}_{32})^{\sum_{k=1}^{r-1} b_k}(a^{(0)}_{21})^{\sum_{k=2}^{r} b_k}$$

is an integer, and

$$|a'^{(j)}_{i,j+1}(a'^{(j-1)}_{j+1,j})^{b_1}(a'^{(j-2)}_{j,j-1})^{b_1+b_2} \cdots (a'^{(1)}_{32})^{\sum_{k=1}^{r-1} b_k}(a'^{(0)}_{21})^{\sum_{k=2}^{r} b_k}|_p$$
$$= |a^{(j)}_{i,j+1}(a^{(j-1)}_{j+1,j})^{b_1}(a^{(j-2)}_{i,j-1})^{b_1+b_2} \cdots (a^{(1)}_{32})^{\sum_{k=1}^{r-1} b_k}(a^{(0)}_{21})^{\sum_{k=2}^{r} b_k}|_p$$

where the b_k, $a'^{(k-1)}_{k+1,k}$, *and* $a^{(k-1)}_{k+1,k}$ *are as described in Lemma* (6.3), $a^{(j)}_{i,j+1}$ *is in* F_m, $a'^{(j)}_{i,j+1}$ *is in* $F_m^{(p)}$, r *is the number of nonzero pivots between steps* 1 *and* j *and* $\sum_{k=1}^{m-1} r_k + 1 \leqq i \leqq n$.

Proof. See Howell [1972, Appendix B].

(6.8) THEOREM. *If* $a^{(j)}_{i,j+1}$ *is in* F_m *and* $a'^{(j)}_{i,j+1}$ *is in* $F^{(p)}_m$, *then*

$$|a'^{(j)}_{i,j+1}|_p = |a^{(j)}_{i,j+1}|_p,$$

where

$$\sum_{k=1}^{m-1} r_k + 1 \leqq i \leqq \sum_{k=1}^{m} r_k \quad \text{and} \quad j + 2 = \sum_{k=1}^{m} r_k.$$

Proof. Using Lemma (6.3) we can show that

$$|(a'^{(j-1)}_{j+1,j})^{b_1} \cdots (a'^{(0)}_{21})^{\sum_{k=2}^{r} b_k}|_p = |(a^{(j-1)}_{j+1,j})^{b_1} \cdots (a^{(0)}_{21})^{\sum_{k=2}^{r} b_k}|_p. \tag{6.9}$$

From Lemma (6.7),

$$|a'^{(j)}_{i,j+1}(a'^{(j-1)}_{j+1,j})^{b_1} \cdots (a'^{(0)}_{21})^{\sum_{k=2}^{r} b_k}|_p = |a^{(j)}_{i,j+1}(a^{(j-1)}_{j+1,j})^{b_1} \cdots (a^{(0)}_{21})^{\sum_{k=2}^{r} b_k}|_p. \tag{6.10}$$

Since the $a^{(j)}_{i,j+1}$ are coefficients of factors of the characteristic polynomial (which is primitive), then by Gauss' lemma, the $a^{(j)}_{i,j+1}$ must be integers. See, for example, Herstein [1964, pp. 120–121]. Therefore, we can multiply both sides of (6.10) by the inverse of (6.9), obtaining

$$|a'^{(j)}_{i,j+1}|_p = |a^{(j)}_{i,j+1}|_p.$$

This completes the proof of the theorem.

From this theorem we have the obvious result:

(6.11) COROLLARY. *For* $i = 1, \cdots, l$, $|F_i|_p = |F^{(p)}_i|_p$.

We have thus shown in this section that if p is a sufficiently large prime number, then $l = l'$, $r_i = r'_i$, and $|F_i|_p = |F^{(p)}_i|_p$.

We have shown in part I that if we choose p sufficiently large, we can carry out the Danilewski algorithm over the integers modulo p and obtain the Frobenius form which we would have obtained had we used exact rational arithmetic. Methods for selecting p are discussed in part II.

B. A. CHARTRES [1964], *Controlled Precision Calculations and the Danilewski Method*, Brown University, Division of Applied Mathematics Report.

A. DANILEWSKI [1937], "On a numerical solution of Vekua's equation," *Mat. Sb.*, v. 2, pp. 169–171. (Russian)

W. L. FRANK [1958], "Computing eigenvalues of complex matrices by determinant evaluation and by methods of Danilewski and Wielandt," *J. SIAM*, v. 6, pp. 378–392. MR **21** #2354.

E. R. HANSEN [1963], "On the Danilewski method," *J. ACM*, v. 10, pp. 102–109. MR **27** #5360.

I. N. HERSTEIN [1964], *Topics in Algebra*, Blaisdell, Waltham, Mass. MR **30** #2028.

A. S. HOUSEHOLDER [1964], *The Theory of Matrices in Numerical Analysis*, Blaisdell, New York. MR **30** #5475.

A. S. HOUSEHOLDER & F. L. BAUER [1959], "On certain methods for expanding the characteristic polynomial," *Numer. Math.*, v. 1, pp. 29–37. MR **20** #7387.

J. A. HOWELL [1972], *An Algorithm for the Exact Reduction of a Matrix to Frobenius Form Using Modular Arithmetic*, University of Texas at Austin Center for Numerical Analysis, Report CNA-39, Austin, Texas.

MICHAEL T. MCCLELLAN [1971], *The Exact Solution of Linear Equations with Polynomial Coefficients*, University of Wisconsin Computer Sciences Department, Technical Report #136, Madison, Wisconsin.

D. L. SLOTNICK [1963], *Modular Arithmetic Computing Techniques*, Westinghouse Electric Corporation, Technical Report ASD-TDR-63-280, Baltimore; Clearinghouse for Federal Scientific and Technical Information, Report No. AD410534, Springfield, Virginia 22151.

H. WAYLAND [1945], "Expansion of determinantal equations into polynomial form," *Quart. Appl. Math.*, v. 2, pp. 277–306. MR **6**, 218.

J. H. WILKINSON [1965], *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford. MR **32** #1894.