Pergamon

Mathl. Comput. Modelling Vol. 19, No. 12, pp. 95–105, 1994

0895-7177/94 $7.00 + 0.00

0895-7177(94)00084-0

One-Modulus Residue Arithmetic Algorithm to Solve Linear Equations Exactly

M. MORHÁČ
Institute of Physics, Slovak Academy of Sciences
Bratislava, Slovakia

(Received December 1992; accepted October 1993)

Abstract—The paper presents an error-free algorithm to solve linear equations using the residue arithmetic. Simultaneously with solving linear equation system, the exact value of determinant of the system matrix is also calculated. The algorithm removes roundoff errors and according to this kind of errors ensures stability of the solution. It is suitable for implementation for computers with possibility of vector operations.

Keywords—Modular arithmetic, Ill-conditioned linear system, Roundoff errors, Iterative error-free algorithm.

1. INTRODUCTION

At present, a lot of methods, direct and iterative, to solve linear equation systems are known. Individual methods more or less fulfill the condition of precision of a found solution compared to the real solution. The most critical are an ill-conditioned linear equation systems. As an example of an ill-conditioned linear equation system, the Hilbert systems with matrix elements

$$a_{i,j} = \frac{1}{i+j-1}; \qquad i,j \in \langle 1, N \rangle,$$

is often given. For ill-conditioned systems, one can answer only with difficulties the question concerning the confidence of the solution. Inaccuracy of the solution is raised by roundoff or truncation errors given by finite length of computer word.

The algorithms getting the benefit of a residue class arithmetic [1–4] make it possible to solve linear equation systems exactly. They use multiple prime moduli and are well suited for parallel calculation algorithms and for multiple processor systems. For a sequential way of calculation, the presented algorithm is more effective.

On the other hand, there exist the error-free algorithms to solve special systems of linear equations, e.g., one and multidimensional deconvolution [5,6], which use only one modulus. However, they are based upon the assumption that the determinant of the system matrix is known beforehand.

The given method presents an exact iterative way of calculation of general linear equation systems, including the calculation of the determinant of a system matrix using one prime modulus. It is suitable for implementation for conventional one-processor computers or to computers with vector operations.

2. ITERATIVE METHOD TO SOLVE LINEAR EQUATION SYSTEM EXACTLY

The set of linear equations is given by

$$\bar{y} = A\bar{x}, \tag{1}$$

where we suppose the elements of vector $\bar{y}$ and matrix A to be integers. Then the solution of regular set (1) can be expressed as

$$\bar{x} = \frac{1}{D}\left[M^0\bar{x}_0 + M^1\bar{x}_1 + \cdots + M^m\bar{x}_m\right], \tag{2}$$

where M is the prime modulus, D is the determinant of matrix A and m is the finite integer. Let us suppose, for the moment, that the value of determinant D is known. From these facts, it follows that $\bar{x}_0,\ \bar{x}_1, \ldots, \bar{x}_m$ are integer vectors. Substituting (2) into (1) gives

$$\bar{y} = \frac{1}{D}\left[M^0\,A\bar{x}_0 + M^1\,A\bar{x}_1 + \cdots + M^m\,A\bar{x}_m\right]. \tag{3}$$

Then

$$\frac{D\bar{y}}{M^{m+1}} = \frac{1}{M^{m+1}}\,A\,\bar{x}_0 + \frac{1}{M^m}\,A\,\bar{x}_1 + \cdots + \frac{1}{M}\,A\,\bar{x}_m, \tag{4}$$

or

$$\frac{1}{M}\left[\ldots\frac{1}{M}\left[\frac{1}{M}(\bar{y}\,D - A\,\bar{x}_0) - A\,\bar{x}_1\right] - \cdots A\,\bar{x}_m\right] = \bar{0}. \tag{5}$$

Let us define

$$\bar{y}_1 = \frac{1}{M}(\bar{y}D - A\bar{x}_0), \tag{6}$$

and

$$\bar{y}_{j+1} = \frac{1}{M}(\bar{y}_j - A\bar{x}_j), \tag{7}$$

where $j = 1, 2, \ldots, m$. If $\bar{y}_{j+1}$ equals the zero vector, the calculation is finished.

Let us further suppose that the inverse matrix A^{-1}, in the sense of residue arithmetic, is also known. For A^{-1} it holds that

$$A\;A^{-1}(\mathrm{mod}\;M) = E, \tag{8}$$

where E is the unit matrix. Then the solution of the set (1) modulo M can be expressed as

$$\bar{x}\,(\mathrm{mod}\;M) = \bar{x}_M = \frac{D}{D}\,\bar{x}_M\,(\mathrm{mod}\;M) = \frac{D_M}{D}\,\bar{x}_M\,(\mathrm{mod}\;M), \tag{9}$$

where

$$D = k\,M + D_M, \tag{10}$$

and k is the integer. Comparing (2) to (9), it is obvious that

$$\bar{x}_0 = D_M\bar{x}_M\,(\mathrm{mod}\;M). \tag{11}$$

From (2), one can also obtain

$$\bar{x}_0 = \bar{x}D - M(\bar{x}_1 + M(\bar{x}_2 + \cdots M\bar{x}_m)\ldots). \tag{12}$$

Substituting (12) to (6), it follows

$$\bar{y}_1 = A\bar{x}_1 + M(A\bar{x}_2 + \cdots M\,A\bar{x}_m), \tag{13}$$

or

$$\bar{y}_1\,(\mathrm{mod}\;M) = A\bar{x}_1. \tag{14}$$

Similarly, substituting (12) into (7), it holds

$$\bar{y}_{j+1}\,(\mathrm{mod}\;M) = A\,\bar{x}_{j+1}. \tag{15}$$

Based on the above given formulas, one can define the algorithm of error-free solution of system (1) using one modulus residue arithmetic. So far, we suppose that the determinant D of the set (1) and inverse matrix (in the sense of residue arithmetic) A^{-1} are known. The algorithms of their calculation are dealt with in Chapters 3 and 4 of this paper.

Algorithm A

a. Let $\bar{y}m_0 = \bar{y}$.

b. Calculate vector $\bar{x}_M$

$$\bar{x}_M = A^{-1}\overline{ym}_0 \,(\text{mod } M). \tag{16}$$

c. According to (11) it holds

$$\bar{x}_0 = D_M \bar{x}_M \,(\text{mod } M). \tag{16a}$$

d. Let $\bar{x} = \bar{x}_0$.

e. Calculate vectors

$$\bar{y}'_0 = A\,\bar{x}_0, \tag{16b}$$

$$\bar{y}_1 = \frac{1}{M}\,(\overline{ym}_0 \cdot D - \bar{y}'_0), \tag{16c}$$

$$\overline{ym}_1 = \bar{y}_1 \,(\text{mod } M). \tag{16d}$$

If $\bar{y}_1$ equals zero vector the calculation is finished, if not continue in f.

f. $j = 1$.

g. Calculate

$$\bar{x}_j = A^{-1}\overline{ym}_j \,(\text{mod } M). \tag{16e}$$

h. For $i = 0, 1, \ldots, N-1$ (N is the size of vectors $\bar{x}, \bar{y}$) calculate

$$x(i) = x(i) + x_j(i)M^j. \tag{16f}$$

i. Calculate vectors

$$\bar{y}'_j = A\,\bar{x}_j, \tag{16g}$$

$$\bar{y}_{j+1} = \frac{\bar{y}_j - \bar{y}'_j}{M}, \tag{16h}$$

$$\overline{ym}_{j+1} = \bar{y}_{j+1} \,(\text{mod } M). \tag{16i}$$

j. If for all $i = 0, 1, \ldots, N-1$, $\bar{y}_{j+1}(i) = 0$, finish the calculation. If not, increment j and repeat the algorithm from point g on.

The resulting solution of system (1) is simply calculated as

$$\bar{x} = \frac{\bar{x}}{D}.$$

Assuming the positive numbers to be represented in the range $\langle 0, (M-1)/2\rangle$ and negative numbers in the range $\langle (M-1)/2+1, M-1\rangle$, it is necessary, in relations (16a), (16d), (16e), (16i) to carry out conversion of negative numbers from modulo representation by subtraction of modulus M.

3. CALCULATION OF EXACT VALUE OF DETERMINANT OF LINEAR EQUATION SYSTEM MATRIX

Let us denote the matrix A of system (1) as $A_{N,N}$, where the indices determine dimensions of matrix. Let us further divide the matrix $A_{N,N}$ to submatrices in the form

$$A_{N,N} = \begin{bmatrix} A_{N-1,N-1} & A_{N-1,1} \\ A_{1,N-1} & a(N-1,N-1) \end{bmatrix}, \tag{16j}$$

where

$$A_{1,N-1} = [a(N-1,0),\ a(N-1,1), \ldots, a(N-1,N-2)], \tag{16k}$$

and

$$A_{N-1,1} = [a(0,N-1), a(1,N-1), \ldots, a(N-2,N-1)]^{\top}. \tag{16l}$$

Let us also suppose that the determinant D_{N-1} of the matrix $A_{N-1,N-1}$ is nonzero. The solution of the set

$$A_{N-1,N-1} \cdot \bar{k}_{N-1} = A_{N-1,1}, \tag{16m}$$

can always be found using the algorithm given in Chapter 2. It can be formally expressed as

$$\bar{k}_{N-1} = A^{-1}_{N-1,N-1} \cdot A_{N-1,1}. \tag{16n}$$

Subtracting the linear combination of the first $N-1$ columns of matrix A

$$\begin{bmatrix} a(0,0), & \ldots, & a(0,N-2) \\ \vdots & \vdots & \vdots \\ a(N-1,0), & \ldots, & a(N-1,N-2) \end{bmatrix} \cdot \bar{k}_{N-1}, \tag{17}$$

from its last column, the value of determinant is not changed. According to relations (16m), (16n), one obtains the matrix A as

$$\begin{bmatrix} A_{N-1,N-1} & \vdots & 0 \\ \ldots\ldots\ldots & \vdots & \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ A_{1,N-1} & \vdots & a(N-1,N-1) - \sum_{i=0}^{N-2} a(N-1,i) \cdot k_{N-1}(i) \end{bmatrix}. \tag{18}$$

From (18), apparently the determinant of matrix A is

$$\begin{aligned} D = D_N &= \left[a(N-1,N-1) - \sum_{i=0}^{N-2} a(N-1,i) \cdot k_{N-1}(i) \right] \det A_{N-1,N-1} \\ &= \left[a(N-1,N-1) - \sum_{i=0}^{N-2} a(N-1,i) \cdot k_{N-1}(i) \right] \cdot D_{N-1}. \end{aligned} \tag{19}$$

Applying above given procedure to determinants D_{N-1}, $D_{N-2}, \ldots$ results in final formula

$$D_N = \prod_{j=1}^{N-1} \left[a(j,j) - \sum_{i=0}^{j-1} a(j,i) \cdot k_j(i) \right], \tag{20}$$

where the coefficient vectors $\bar{k}_j$ are calculated by exact solution of sets

$$A_{j,j} \cdot \bar{k}_j = A_{j,i} \qquad j = 1,2,\ldots,N-1, \tag{21}$$

using the algorithm described in Chapter 2. We have assumed the determinants of matrices $A_{j,j}$ in (21) to be nonzero, i.e., the systems to be regular. If it is not the case, for some j, one must choose the matrix partition ensuring the regularity of (21). The eventual change of sign of the calculated determinant must be considered.

4. INVERSE MATRIX CALCULATION

In the basic algorithm of the error-free solution of linear equations in Part 2, as well as in relations (16), (21) in Part 3, the inverse matrix

$$A^{-1} \pmod{M} \quad \text{or} \quad A^{-1}_{j,j} \pmod{M}, \tag{22}$$

was supposed to be known. Now the question is how to calculate inverse matrices (22) in the most effective way. We could use classic Gauss' elimination method applied in residue arithmetic [1]. Because, according to the algorithm given by relations (20), (21), one needs the inverse matrix $A_j^{-1} \pmod M$ for each j this would be very time consuming. Consequently, the following algorithm can be applied to matrix A partition according to (16j). Similarly, as in (16j), we divide A of dimensions $(j+1) \times (j+1)$ to

$$A_{j+1,j+1} = \begin{bmatrix} A_{j,j}, & A_{j,1} \\ A_{1,j}, & A_{1,1} \end{bmatrix}. \tag{23}$$

The indices of the submatrices have again the meaning of dimensions. Let us denote

$$A_{j+1,j+1}^{-1} \pmod M = \begin{bmatrix} B_{j,j}, & B_{j,1} \\ B_{1,j}, & B_{1,1} \end{bmatrix}, \tag{24}$$

from which it follows

$$\begin{bmatrix} A_{j,j}, & A_{j,1} \\ A_{1,j}, & A_{1,1} \end{bmatrix} \begin{bmatrix} B_{j,j}, & B_{j,1} \\ B_{1,j}, & B_{1,1} \end{bmatrix} \pmod M = \begin{bmatrix} E_{j,j}, & 0_{j,1} \\ 0_{1,j}, & 1 \end{bmatrix}, \tag{25}$$

where $E_{j,j}$ denotes unit matrix and $0_{j,1}$, $0_{1,j}$ denote zero vectors. Multiplying the submatrices in (25), one gets

$$\begin{aligned} A_{j,j} \cdot B_{j,j} + A_{j,1} \cdot B_{1,j} &= E_{j,j}, \\ A_{1,j} \cdot B_{j,j} + A_{1,1} \cdot B_{1,j} &= \bar{0}_{1,j}, \end{aligned} \tag{26}$$

and

$$\begin{aligned} A_{j,j} \cdot B_{j,1} + A_{j,1} \cdot B_{1,1} &= \bar{0}_{j,1}, \\ A_{1,j} \cdot B_{j,1} + A_{1,1} \cdot B_{1,1} &= 1. \end{aligned} \tag{27}$$

Assume that we know $A_{j,j}^{-1} \pmod M$. Then the solution of systems (27), (26) is

$$B_{1,1} = \left[A_{1,1} - A_{1,j} \cdot A_{j,j}^{-1} \cdot A_{j,1}\right]^{-1} \pmod M, \tag{28}$$

$$B_{j,1} = -A_{j,j}^{-1} \cdot A_{j,1} \left[A_{1,1} - A_{1,j} \cdot A_{j,j}^{-1} \cdot A_{j,1}\right]^{-1} \pmod M, \tag{29}$$

$$B_{1,j} = -\left[A_{1,1} - A_{1,j} \cdot A_{j,j}^{-1} \cdot A_{j,1}\right]^{-1} A_{1,j} \cdot A_{j,j}^{-1} \pmod M, \tag{30}$$

$$B_{j,j} = A_{j,j}^{-1} + A_{j,j}^{-1} \cdot A_{j,1} \cdot \left[A_{1,1} - A_{1,j} \cdot A_{j,j}^{-1} \cdot A_{j,1}\right] \cdot A_{1,j} \cdot A_{j,j}^{-1} \pmod M, \tag{31}$$

or

$$B_{1,1} = \left[A_{1,1} - A_{1,j} \cdot A_{j,j}^{-1} \cdot A_{j,1}\right]^{-1} \pmod M, \tag{32}$$

$$B_{j,1} = -A_{j,j}^{-1} \cdot A_{j,1} \cdot B_{1,1} \pmod M, \tag{33}$$

$$B_{1,j} = -B_{1,1} \cdot A_{1,j} \cdot A_{j,j}^{-1} \pmod M, \tag{34}$$

$$B_{j,j} = A_{j,j}^{-1} + B_{j,1} \cdot B_{1,1}^{-1} \cdot B_{1,j} \pmod M. \tag{35}$$

In the relations (32) to (35), one needs to do $j^2 + j$ multiplications, i.e., entirely $4(j^2 + j)$ multiplications. This algorithm requires to invert only one integer (in the sense of residue arithmetic) in relation (32). The inversion can be realised using, e.g., Euclid's algorithm [7,8].

5. OPERATIONS WITH LONG INTEGERS

Because the intermediate results in steps (16b), (16c), (16g), (16h) have to retain the entire length of components of vectors $\bar{y}$ and $\bar{y}'$, we have to work (in these steps) with long integers. Also in the algorithm of exact calculation of matrix determinant given by (16n), (19) we have to retain entirely intermediate determinant.

Let us assume that the elements of matrix A and vector $\bar{y}$ are integers less that modulus M that can be contained in one computer word. We can store the vectors $\bar{y}'_0$, $\bar{y}_1$, $\bar{y}'_j$, $\bar{y}_j$ in (16b), (16c), (16g), (16h) or vector $\bar{k}_{N-1}$ in (16n) in matrix form denoted by letter D. It holds that

$$\sum_{j=0}^{N-1} D(i,j) \cdot M^j = \sum_{j=0}^{N-1} d_j(i) \cdot M^j = d(i), \qquad i = 0, 1, \ldots, N-1, \tag{36}$$

where $\bar{d}$ represents the above given vectors.

Then the operation $\overline{ym}_0 \cdot D$ in (16c) represents multiplication of long integers $D(i)$ by short (one-word) integers $ym_0(i)$, $i = 0, 1, \cdots, N-1$, adhering to all rules about overflows between vectors $\bar{d}_j$. Operation $A \cdot \bar{x}_0$ or $A \cdot \bar{x}_j$ in (16b), (16g) represents successive multiplication of columns (short integers) of matrix A by short integers of vectors $\bar{x}_0$ or $\bar{x}_j$ and their subtraction from long integers $\overline{ym}_0 \cdot D$ or $\bar{y}_j$.

Operation of division of vector $\bar{y}_1$ resp. $\bar{y}_{j+1}$ by modulus M in (16c) or (16g) represents according to (36) just the shift of columns of matrix one position to right.

Finally, relation (19) represents multiplication of long integer D_{N-1} by short integer k_{N-1} and subtraction of long integers $k_{N-1}(i) \cdot D_{N-1}$ multiplied by short integers $a(N-1, i)$. When using matrix form of storing the long integers all these operations can be done very simply.

6. FINAL ALGORITHM TO CALCULATE DETERMINANT AND EXACT SOLUTION OF LINEAR EQUATION SYSTEM

Based on conclusions from previous parts, we can now define final algorithm to calculate exact solution of linear equation system (1).

Algorithm B

1. Let $k = 1$.
2. Let

$$\begin{aligned} &A_{1,1} = a(0,0);\ A_{1,1}^{-1} = a(0,0)^{-1} \cdot (\mathrm{mod}\ M);\\ &D_1 = a(0,0);\ D_{1,M} = a(0,0);\ \bar{c}_1 = [a(0,1)];\\ &\bar{d}_1^{\mathsf{T}} = [a(1,0)]. \end{aligned}$$

3. Let

$$A = A_{k,k};\ A^{-1} = A_{k,k}^{-1};\ D = D_k;\ D_M = D_{k,M};\ \bar{y} = \bar{c}_k.$$

4. Carry out the algorithm A.
5. If $k = N$, finish the calculation. The resulting vector is stored in vector $\bar{x}$ and determinant in D. If $k \neq N$, continue in step 6.
6. Let $k = k + 1$.
7. According to (16j), (16k), (16l) set up the matrix

$$A_{k,k} = \begin{bmatrix} A_{k-1,k-1}, & \bar{c}_{k-1} \\ \bar{d}_{k-1}^{\mathsf{T}}, & a(k-1,k-1) \end{bmatrix}.$$

8. According to (16m), (18), (19) calculate the value of determinant in step k

$$D_k = a(k-1,k-1) \cdot D_{k-1} - \bar{d}_{k-1}^{\mathsf{T}} \cdot \bar{x}.$$

9. Calculate $D_{k,M} = D_k \cdot (\mathrm{mod}\ M)$.

10. According to (32), (33), (34), (35) calculate

$$e = \left[a(k-1,k-1) - \bar{d}_{k-1}^{\top} \cdot A_{k-1,k-1}^{-1} \cdot \bar{c}_{k-1}\right]^{-1} \cdot (\text{mod } M),$$
$$f = -A_{k-1,k-1}^{-1} \cdot \bar{c}_{k-1} \cdot e \cdot (\text{mod } M),$$
$$\bar{g}^{\top} = -e \cdot \bar{d}_{k-1}^{\top} \cdot A_{k-1,k-1}^{-1} \cdot (\text{mod } M),$$
$$H = A_{k-1,k-1}^{-1} + \bar{f} \cdot \bar{g}^{\top} \cdot e^{-1} \cdot (\text{mod } M).$$

11. Based on (24) set up inverse matrix in residue class M

$$A_{k,k}^{-1} \cdot (\text{mod } M) = \begin{bmatrix} H, & \bar{f} \\ \bar{g}^{\top}, & e \end{bmatrix}.$$

12. Set up vector

$$\bar{a}_k = [a(0,k),\, a(1,k), \cdots a(k-1,k)]^{\top}.$$

13. Set up vector

$$\bar{d}_k^{\top} = [a(k,0),\, a(k,1), \cdots, a(k,k-1)].$$

14. Return to point 3.

7. ALGORITHM ILLUSTRATION

Let us solve the system of linear equations

$$\begin{bmatrix} 5 & 2 & 0 \\ 1 & 3 & 6 \\ 2 & 1 & 4 \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix}, \tag{37}$$

that can be written in the form of extended matrix as

$$\begin{bmatrix} 5 & 2 & 0 & 3 \\ 1 & 3 & 6 & 2 \\ 2 & 1 & 4 & 1 \end{bmatrix}.$$

We shall use modulus $M = 7$. We proceed according to the algorithm given in Part 6 and Algorithm A given in Part 2. For the sake of illustration, we denote the steps exactly like in the given algorithms.

1. $k = 1$.
2. $A_{1,1} = [5]$; $A_{1,1}^{-1} = [3]$; $D_1 = 5$; $\bar{c}_1 = [2]$; $\bar{d}_1^{\top} = [1]$.
3. $A = [5]$; $A^{-1} = [3]$; $D = 5$; $\bar{y} = [2]$.
4. Algorithm A
   (a) $\overline{ym}_0 = [2]$.
   (b) $\bar{x}_M = A^{-1} \cdot \overline{ym}_0 \cdot (\text{mod } M) = [3] \cdot [2] \cdot (\text{mod } 7) = [6]$.
   (c) $\bar{x}_0 = D_M \cdot \bar{x}_M \cdot (\text{mod } M) = 5[6] \cdot (\text{mod } 7) = [2]$.
   (d) $\bar{x} = [2]$.
   (e) $\bar{y}_0' = A\,\bar{x}_0 = [5] \cdot [2] = [10]$.
   $\bar{y}_1 = \frac{1}{M}(\overline{ym}_0 \cdot D - \bar{y}_0') = \frac{1}{7}\,[[2]5 - [10]] = [0]$.
   Zero vector $\bar{y}_1$ indicates the end of Algorithm A.
5. $k \neq N \quad (1 \neq 3)$.
6. $k = 1 + 1 = 2$.
7. $A = \begin{bmatrix} 5, & 2 \\ 1, & 3 \end{bmatrix}$.
8. $D_2 = a(1,1)\,D_1 - \bar{d}_1^{\top} \cdot \bar{x} = [3]\,5 - [1] \cdot [2] = 13$.

9. $D_{2,M} = 13 \pmod 7 = 6.$
10. $e = \left[a(1,1) - \bar{d}_1^{\top} \cdot A_{1,1}^{-1} \cdot \bar{c}_1\right]^{-1} \cdot (\mathrm{mod}\ M) = [[3] - [1]\,[3]\,[2]]^{-1} \cdot (\mathrm{mod}\ 7) = 2$
    $f = -A_{1,1}^{-1} \cdot \bar{c}_1 \cdot e \ (\mathrm{mod}\ M) = -[3]\,[2]\,2 \ (\mathrm{mod}\ 7) = [2]$
    $\bar{g}^{\top} = -e\,\bar{d}_1^{\top} \cdot A_{1,1}^{-1} \cdot (\mathrm{mod}\ M) = -2\,[1]\,[3] \ (\mathrm{mod}\ 7) = [1]$
    $H = A_{1,1}^{-1} + \bar{f} \cdot \bar{g}^{\top} \cdot e^{-1} \cdot (\mathrm{mod}\ M) = [3] + [2]\,[1]\,2^{-1} \cdot (\mathrm{mod}\ M) = [4].$
11. $A_{2,2}^{-1} \cdot (\mathrm{mod}\ M) = \begin{bmatrix} A_{1,1}^{-1} & \bar{f} \\ \bar{g}^{\top} & e \end{bmatrix} = \begin{bmatrix} 4, & 2 \\ 1, & 2 \end{bmatrix}.$
12. $\bar{c}_2 = [0,6]^{\top}.$
13. $\bar{d}_2^{\top} = [2,1].$

We return to step 3. Now we are going to solve the system $\begin{bmatrix} 5, & 2, & 0 \\ 1, & 3, & 6 \end{bmatrix}.$

3. $A = \begin{bmatrix} 5, & 2 \\ 1, & 3 \end{bmatrix};\ A^{-1} = \begin{bmatrix} 4, & 2 \\ 1, & 2 \end{bmatrix};\ D = 13;\ D_M = 6;\ \bar{y} = \begin{bmatrix} 0 \\ 6 \end{bmatrix}.$
4. Algorithm A
    (a) $\overline{ym}_0 = [0,6]^{\top}.$
    (b) $\bar{x}_M = A^{-1} \cdot \overline{ym} \cdot (\mathrm{mod}\ M) = \begin{bmatrix} 4, & 2 \\ 1, & 2 \end{bmatrix} \begin{bmatrix} 0 \\ 6 \end{bmatrix} (\mathrm{mod}\ 7) = \begin{bmatrix} 5 \\ 5 \end{bmatrix}.$
    (c) $\bar{x}_0 = D_M \cdot \bar{x}_M \cdot (\mathrm{mod}\ M) = 6 \begin{bmatrix} 5 \\ 5 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}.$
    (d) $\bar{x} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}.$
    (e) $\bar{y}_0' = A \cdot \bar{x}_0 = \begin{bmatrix} 5, & 2 \\ 1, & 3 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 14 \\ 8 \end{bmatrix},$
    $\bar{y}_1 = \frac{1}{M}\left(\overline{ym}_0 \cdot D - \bar{y}_0'\right) = \frac{1}{7}\left[\begin{bmatrix} 0 \\ 6 \end{bmatrix} \cdot 13 - \begin{bmatrix} 14 \\ 8 \end{bmatrix}\right] = \begin{bmatrix} -2 \\ 10 \end{bmatrix},$
    $\overline{ym}_1 = [5, \quad 3]^{\top}.$
    (f) $j = 1.$
    (g) $\bar{x}_1 = A^{-1} \cdot \overline{ym}_1 \cdot (\mathrm{mod}\ M) = \begin{bmatrix} 4, & 2 \\ 1, & 2 \end{bmatrix} \begin{bmatrix} 5 \\ 3 \end{bmatrix} (\mathrm{mod}\ 7) = \begin{bmatrix} 5 \\ 4 \end{bmatrix} = \begin{bmatrix} -2 \\ -3 \end{bmatrix} (\mathrm{mod}\ 7).$
    (h) $\bar{x} = \bar{x} + \bar{x}_1 \cdot M = \begin{bmatrix} 2 \\ 2 \end{bmatrix} + \begin{bmatrix} -2 \\ -3 \end{bmatrix} \cdot 7 = \begin{bmatrix} -12 \\ -19 \end{bmatrix}.$
    (i) $\bar{y}_1' = A \cdot \bar{x}_1 = \begin{bmatrix} 5, & 2 \\ 1, & 3 \end{bmatrix} \begin{bmatrix} -2 \\ -3 \end{bmatrix} = \begin{bmatrix} -16 \\ -11 \end{bmatrix},$
    $\bar{y}_2 = \frac{\bar{y}_1 - \bar{y}_1'}{M} = \frac{1}{7}\left[\begin{bmatrix} -2 \\ 10 \end{bmatrix} - \begin{bmatrix} -16 \\ -11 \end{bmatrix}\right] = \begin{bmatrix} 2 \\ 3 \end{bmatrix},$
    $\overline{ym}_2 = \begin{bmatrix} 2 \\ 3 \end{bmatrix}.$
    (j) The vector $\bar{y}_2$ does not equal the zero vector, we increment j and return to the point g.
    (g) $\bar{x}_2 = A^{-1} \cdot \overline{ym}_2 \cdot (\mathrm{mod}\ M) = \begin{bmatrix} 4, & 2 \\ 1, & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix} (\mathrm{mod}\ 7) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$
    (h) $\bar{x} = \bar{x} + \bar{x}_2 \cdot M = \begin{bmatrix} -12 \\ -19 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot 49 = \begin{bmatrix} -12 \\ 30 \end{bmatrix}.$
    (i) $\bar{y}_2' = A \cdot \bar{x}_2 = \begin{bmatrix} 5, & 2 \\ 1, & 3 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \end{bmatrix},$
    $\bar{y}_3 = \frac{\bar{y}_2 - \bar{y}_2'}{M} = \frac{1}{7}\left[\begin{bmatrix} 2 \\ 3 \end{bmatrix} - \begin{bmatrix} 2 \\ 3 \end{bmatrix}\right] = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$
    $\overline{ym}_3 = [0,0]^{\top}.$
    (j) Zero vector $\bar{y}_3$ indicates the end of the Algorithm A.
5. $k \neq N$ $(2 \neq 3)$, i.e., we continue in point 6.
6. $k = 3$

7. $A_{3,3} = \begin{bmatrix} A_{2,2} & \bar{c}_2 \\ \bar{d}_2^{\top} & a(2,2) \end{bmatrix} = \begin{bmatrix} 5, & 2, & 0 \\ 1, & 3, & 6 \\ 2, & 1, & 4 \end{bmatrix}$.
8. $D_3 = a(2,2) \cdot D - \bar{d}_2^{\top} \cdot \bar{x} = 4 \cdot 13 - [2,1] \cdot \begin{bmatrix} -12 \\ 30 \end{bmatrix} = 46$. We remind that $\bar{x}$ is the solution of Algorithm A.
9. $D_{3,M} = D_3 \cdot (\text{mod } M) = 46 \cdot (\text{mod } 7) = 4$
10. $e = \left[a(2,2) - \bar{d}_2^{\top} \cdot A_{2,2}^{-1} \cdot \bar{e}_2\right]^{-1} \cdot (\text{mod } M) = 4\left[-[2,1] \cdot \begin{bmatrix} 4, & 2 \\ 1, & 2 \end{bmatrix} \begin{bmatrix} 0 \\ 6 \end{bmatrix}\right]^{-1} (\text{mod } 7) = 5,$

$$\bar{f} = -A_{2,2}^{-1} \cdot \bar{c}_2 \cdot e \cdot (\text{mod } M) = -\begin{bmatrix} 4, & 2 \\ 1, & 2 \end{bmatrix} \begin{bmatrix} 0 \\ 6 \end{bmatrix} \cdot 5 \ (\text{mod } 7) = \begin{bmatrix} 3 \\ 3 \end{bmatrix},$$

$$\bar{g}^{\top} = -e \cdot \bar{d}_2^{\top} \cdot A_{2,2}^{-1} \cdot (\text{mod } M) = -[2,1] \begin{bmatrix} 4, & 2 \\ 1, & 2 \end{bmatrix} (\text{mod } 7) = [4,5],$$

$$H = A_{2,2}^{-1} + \bar{f} \cdot \bar{g}^{\top} \cdot e^{-1} \cdot (\text{mod } M) = \begin{bmatrix} 4, & 2 \\ 1, & 2 \end{bmatrix} + 3 \begin{bmatrix} 3 \\ 3 \end{bmatrix} [4,5] \, (\text{mod } 7) = \begin{bmatrix} 5, & 5 \\ 2, & 5 \end{bmatrix}.$$

11. We set up inverse matrix $A_{3,3}^{-1}$

$$A_{3,3}^{-1} = \begin{bmatrix} 5, & 5, & 3 \\ 2, & 5, & 3 \\ 4, & 5, & 5 \end{bmatrix}.$$

12. We set up the vector

$$\bar{c}_3 = [3, \quad 2, \quad 1]^{\top}.$$

13. It is impossible to set up the vector $\bar{d}_3^{\top}$ because the matrix contains only 3 rows. But we shall not need it in following computations.

We return to the step 3 where we are going to solve finally the system $\begin{bmatrix} 5, & 2, & 0, & 3 \\ 1, & 3, & 6, & 2 \\ 2, & 1, & 4, & 1 \end{bmatrix}$.

3. $A = \begin{bmatrix} 5, & 2, & 0 \\ 1, & 3, & 6 \\ 2, & 1, & 4 \end{bmatrix}$; $A^{-1} = \begin{bmatrix} 5, & 5, & 3 \\ 2, & 5, & 3 \\ 4, & 5, & 5 \end{bmatrix}$; $D = 46$; $D_M = 4$; $\bar{y} = \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix}$.
4. Algorithm A
   (a) $\overline{ym}_0 = [3,2,1]^{\top}$.
   (b) $\bar{x}_M = A^{-1} \cdot \overline{ym}_0 \cdot (\text{mod } M) = \begin{bmatrix} 5, & 5, & 3 \\ 2, & 5, & 3 \\ 4, & 5, & 5 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix} (\text{mod } M) = \begin{bmatrix} 0 \\ 5 \\ 6 \end{bmatrix}$.
   (c) $\bar{x}_0 = D_M \cdot \bar{x}_M \cdot (\text{mod } M) = 4 \begin{bmatrix} 0 \\ 5 \\ 6 \end{bmatrix} (\text{mod } 7) = \begin{bmatrix} 0 \\ 6 \\ 3 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \\ 3 \end{bmatrix} \cdot (\text{mod } 7)$.
   (d) $\bar{x} = \begin{bmatrix} 0 \\ -1 \\ 3 \end{bmatrix}$.
   (e) $\bar{y}_0' = A \cdot \bar{x}_0 = \begin{bmatrix} 5, & 2, & 0 \\ 1, & 3, & 6 \\ 2, & 1, & 4 \end{bmatrix} \begin{bmatrix} 0 \\ -1 \\ 3 \end{bmatrix} = \begin{bmatrix} -2 \\ 15 \\ 11 \end{bmatrix}$.
   (f) $j = 1$.
   (g) $\bar{x}_1 = A^{-1} \cdot \overline{ym}_1 \cdot (\text{mod } M) = \begin{bmatrix} 5, & 5, & 3 \\ 2, & 5, & 3 \\ 4, & 5, & 5 \end{bmatrix} \begin{bmatrix} 6 \\ 4 \\ 5 \end{bmatrix} \cdot (\text{mod } 7) = \begin{bmatrix} 2 \\ 5 \\ 6 \end{bmatrix} = \begin{bmatrix} 2 \\ -2 \\ -1 \end{bmatrix} \cdot (\text{mod } 7)$.
   (h) $\bar{x} = \bar{x} + \bar{x}_1 \cdot M^1 = \begin{bmatrix} 0 \\ -1 \\ 3 \end{bmatrix} + \begin{bmatrix} 2 \\ -2 \\ -1 \end{bmatrix} \cdot 7 = \begin{bmatrix} 14 \\ -15 \\ -4 \end{bmatrix}$.

(i) $\bar{y}_1' = A \cdot \bar{x}_1 = \begin{bmatrix} 5, & 2, & 0 \\ 2, & 3, & 6 \\ 1, & 1, & 4 \end{bmatrix} \begin{bmatrix} 2 \\ -2 \\ -1 \end{bmatrix} = \begin{bmatrix} 6 \\ -10 \\ -2 \end{bmatrix},$

$$\bar{y}_2 = \frac{\bar{y}_1 - \bar{y}_1'}{M} = \frac{1}{7}\left[\begin{bmatrix} 20 \\ 11 \\ 5 \end{bmatrix} - \begin{bmatrix} 6 \\ -10 \\ -2 \end{bmatrix}\right] = \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix},$$

$\overline{ym}_2 = \bar{y}_2 \cdot (\mathrm{mod}\ M) = [2,3,1]^\top$.

Vector $\bar{y}_2$ does not equal the zero vector. We increment j and return to step g.

(g) $\bar{x}_2 = A^{-1} \cdot \overline{ym}_2 \cdot (\mathrm{mod}\ M) = \begin{bmatrix} 5, & 5, & 3 \\ 2, & 5, & 3 \\ 4, & 5, & 5 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix} \cdot (\mathrm{mod}\ 7) = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}.$

(h) $\bar{x} = \bar{x} + \bar{x}_2 \cdot M^2 = \begin{bmatrix} 14 \\ -15 \\ -4 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \cdot 7^2 = \begin{bmatrix} 14 \\ 34 \\ -4 \end{bmatrix}.$

(i) $\bar{y}_2' = A \cdot \bar{x}_2 = \begin{bmatrix} 5, & 2, & 0 \\ 1, & 3, & 6 \\ 2, & 1, & 4 \end{bmatrix},$

$$\bar{y}_3 = \frac{\bar{y}_2 - \bar{y}_2'}{M} = \frac{1}{7}\left[\begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix} - \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}\right] = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},$$

$\overline{ym}_3 = [0,0,0]^\top$.

(j) Zero vector $\bar{y}_3$ indicates the end of Algorithm A.

5. $k = N$ $(3 = 3)$ which indicates the end of calculation. The resulting solution is given by division the components of vector $\bar{x}$ by determinant D,i.e.,

$$\bar{x} = \frac{1}{46}\begin{bmatrix} 14 \\ 34 \\ -4 \end{bmatrix}.$$

Substituting this solution to (37), we can prove its correctness.

8. CONCLUSION

Exact algorithm to solve linear equation systems presented in the paper removes roundoff errors. These kinds of errors for ill-conditioned systems negatively influence or completely destroy the calculated solution.

The algorithm is based on residue arithmetic that does not introduce errors into solution and removes the necessity to watch overflows during calculation. In relations (16b), (16c), (16f), (16g) and (16h) of the Algorithm A and in the step 8 of the Algorithm B, long integer arithmetic must be used. This brings some inconvenience to algorithm implementations. However, keeping the way to store long integer vectors in form given by (36) facilitates to some extent these operations.

In the whole algorithm, there does not occur any operation of division or any other operation that could perform rounding off, and therefore, loss of information. The inversion in the sense of residue arithmetic is used in point 10 of the algorithm B and in calculation of scalar e. Because the used modulus is prime, this operation is defined for $a \in \langle 1, m-1 \rangle$. The problem can arise only if determinant D (mod M) $= 0$. Here another prime modulus must be chosen and whole calculation repeated. When choosing large value of M, the probability that such a situation occurs (it is dealt with in [1]) is rather small.

The algorithm is rather time consuming that is penalty for exact calculation. It is well suited for implementation in processors with vector operations or for hardware implementation in specialized processor to solve linear equation system. This could substantially speed up the calculation.

REFERENCES

1. M. Newman, Solving equations exactly, *National Bureau of Standards* **71B**, 171–179 (1967).
2. R.T. Gregory, E.V. Krishnamurthy, *Methods and Applications of Error-free Computation*, Springer-Verlag, New York, (1984).
3. M. Morháč, R. Lórencz, A modular system for solving linear equations exactly. I. Architecture and numerical algorithms, *Computers and Artifical Intelligence* **11** (4), 351–361 (1992).
4. R. Lórencz, M. Morháč, A modular system for solving linear equations exactly. II. Hardware realization, *Computers and Artificial Intelligence* **11** (5), 497–507 (1992).
5. M. Morháč, Precise deconvolution using the Fermat number transform, *Computers Math. Applic.* **12A** (1), 319–329 (1986).
6. M. Morháč, k-dimensional error-free deconvolution using the Fermat number transform, *Computers Math. Applic.* **18** (12), 1023–1032 (1989).
7. R.E. Blahut, *Fast Algorithms for Digital Signal Processing*, IBM Corporation, Owego, NY, (1985).
8. M. Morháč, System identification and deconvolution using the Fourier and the Fermat transforms, (in Slovak), Dissertation, Bratislava, (1983).