Journal of Computational and Applied Mathematics 18 (1987) 279-287
North-Holland

Error-free computer solution of certain systems of linear equations

Stephen F. CHANG
Department of Mathematics and Computer Science, California State College, Bakersfield, CA 93311-1099, U.S.A

William J. KENNEDY
Department of Statistics, Iowa State University, Ames, IA 5001 I, U.S.A.

Received 18 October 1985
Revised 29 May 1986

Abstract: In this article we propose a procedure which generates the exact solution for the system $\boldsymbol{Ax} = \boldsymbol{b}$, where $\boldsymbol{A}$ is an integral nonsingular matrix and b is an integral vector, by improving the initial floating-point approximation to the solution. This procedure, based on an easily programmed method proposed by Aberth [1], first computes the approximate floating-point solution $\boldsymbol{x}^*$ by using an available linear equation solving algorithm. Then it extracts the exact solution $\boldsymbol{x}$ from $\boldsymbol{x}^*$ if the error in the approximation $\boldsymbol{x}^*$ is sufficiently small. An a posteriori upper bound for the error of $\boldsymbol{x}^*$ is derived when Gaussian Elimination with partial pivoting is used. Also, a computable upper bound for $|\det(\boldsymbol{A})|$, which is an alternative to using Hadamard's inequality, is obtained as a byproduct of the Gaussian Elimination process.

Keywords: Computable error bounds, linear systems, exact solution.

1. Introduction

Most numerical methods described in the numerical mathematics literature assume use of the real number system. Computer oriented algorithms which implement these methods employ floating-point arithmetic, provided by the hardware and software of the computer system, to simulate real number arithmetic. Users of such software are forced to accept floating-point approximations to solution of given problems, even in view of the fact that the floating-point number system being used usually consists of a small subset of the rational numbers and problems having rational solution might be solved without error.

Consider the problem of solving a system of linear equations Ax = $\boldsymbol{b}$. In most cases the data A and b are given as rational numbers, and often these numbers all lie in a short range about zero. When the number are entered as floating-point values into the computer, the problem is still one involving only rational numbers. Since this problem has solution vector $\boldsymbol{x}$ having rational number components, one might hope to find these rational numbers in the floating-point system or select values in that system which are as close as possible to the exact solution of the problem defined inside the computer. Using floating-point arithmetic operations to carry out classical numerical methods is not likely to produce these desired results. The purpose of the present paper is to consider a numerical method which can often be used to obtain the exact solution to a system of linear equations. Production of such a solution silences all questions about the probable amount of error in the computed solution. Such questions often arise in connection with solutions obtained using only floating-point arithmetic.

Borosh and Fraenkel [3], Newman [11], Howell and Gregory [9], and Cabay and Lam [4,5] have all described methods for obtaining the exact solution to $Ax = \boldsymbol{b}$ using residue arithmetic. Adegbeyeni and Krishnamurthy [2] also have suggested a method using finite segment p-adic number arithmetic to compute the exact solution. These procedures do produce an exact solution in most cases, however, a large amount of computer memory and considerable execution time are needed when a digital computer is used. With today's sophisticated computer systems, a moderately good approximation to the solution of $Ax = \boldsymbol{b}$ can often be obtained using high precision floating-point arithmetic and production of such an approximation requires less memory space and execution time than exact methods.

In this paper we propose a procedure which generates the exact solution for the system $Ax = \boldsymbol{b}$ by improving the initial floating-point approximation. This procedure, which is based on an easily programmed method [1], first computes the approximate floating-point solution $\boldsymbol{x}^*$ by using an available linear equation solving algorithm. Then it extracts the exact solution $\boldsymbol{x}$ from $\boldsymbol{x}^*$ if the error in the approximation $\boldsymbol{x}^*$ is sufficiently small. This procedure requires that the relationship $\epsilon^* < 1/2 \det(A)^2$, where $\epsilon^* = \|\boldsymbol{x} - \boldsymbol{x}^*\|_\infty$ is the maximum error in the approximation, be satisfied in order to guarantee that the extraction is correct.

Neither ϵ^* nor $\det(A)$ will be known a-priori, however, an a-posteriori upper bound for ϵ^*, $\hat{\epsilon}^*$, and an upper bound for $|\det(A)|$, Q^*, will be defined and used. Moreover: these two estimates can be used to verify the relationship $\epsilon^* < 1/2 \det(A)^2$ because $\hat{\epsilon}^* < 1/2 Q^{*2}$ implies that $\epsilon^* < 1/2 \det(A)^2$. Hence, we can use the criterion $\hat{\epsilon}^* < 1/2\hat{Q}^*$ to determine whether the extraction is correct. In Section 2, an upper bound for ϵ^* will be derived when Gaussian Elimination with partial pivoting is used to solve the linear system $Ax = b$. An upper bound for $|\det(A)|$, which is an alternative to using Hadamard's inequality, can be obtained as a byproduct of Gaussian Elimination process. This will be done in Section 3 using the notation developed in Section 2. In Sections 4 and 5, the extracting of $\boldsymbol{x}$ from $\boldsymbol{x}^*$ will be described and three numerical examples given.

2. A posteriori error bound for computed solution x^* of $Ax = b$

Let $Ax = b$ be a system of linear equations, where A is a nonsingular $n \times n$ matrix, b is a vector with n elements and $\boldsymbol{x}$ is the required solution. When floating-point arithmetic is used to solve this system, errors in the solution stem from two sources. One is the so-called inherent errors and the other is the so-called abbreviation errors [12]. In this research, we are only concerned with abbreviation errors and will construct an explicit error bound for the computed solution x^* when Gaussian Elimination with partial pivoting is used. The maximum norm will be used to measure the size of vectors and matrices, that is

$$\|\boldsymbol{x}\|_\infty = \max_{1 \leqslant i \leqslant n} |x_i|, \quad \text{if } \boldsymbol{x} = (x_1, x_2, \ldots, x_n)',$$

and

$$\| A \|_\infty = \max_{1 \leqslant i \leqslant n} \sum_{j=1}^{n} | a_{ij} | \quad \text{if } A = (a_{ij}).$$

Also, given any two floating-point numbers x and y, we denote the result of floating-point addition, subtraction, multiplication and division, respectively, by $fl(x+y)$, $fl(x-y)$, $fl(x \cdot y)$ and $f(x/y)$.

When partial pivoting Gaussian Elimination is used to solve the linear system Ax = $\boldsymbol{b}$, the computed solution $\boldsymbol{x}^*$ exactly satisfies a perturbed equation

$$(A + \delta A)\boldsymbol{x}^* = \boldsymbol{b}, \tag{2.1}$$

where δA is a matrix whose elements are about the size of round off errors in the elements of A [7]. Based on (2.1), an explicit error bound for $\boldsymbol{x}^*$ will be derived by using the computed results. As we know, the first and largest step in Gaussian Elimination is the decomposition of A into the product of two triangular matrices L and U. We assume that A is initially given with its rows scaled and ordered in such a way that no row interchanges are needed. In practice this is not always the case, but row interchanges are irrelevant to the error analysis. The decomposition consists of computing a sequence of matrices as follows,

$$A = \hat{A}^{(1)} \to \hat{A}^{(2)} \to \cdots \to \hat{A}^{(n)} = U \tag{2.2}$$

where $\hat{A}^{(k)} = (\hat{a}_{ij}^{(k)})$, $k = 1, 2, \ldots, n$, and

$$\hat{a}_{ij}^{(k)} = \begin{cases} a_{ij}^{(k)} + \sum_{l=1}^{k-1} \epsilon_{ij}^{(l)} & \text{for } k = 2, 3, \ldots, n, \\ & \quad i = k, k+1, \ldots, n, \quad j = k, \ k+1, \ldots, n; \\ \hat{a}_{ij}^{(k-1)} & \text{for } k = 1, 2, \ldots, n, \ 1 \leqslant i \leqslant k, \ 1 \leqslant j \leqslant n; \\ 0 & \text{otherwise}; \end{cases} \tag{2.3}$$

$$\epsilon_{ij}^{(l)} = 0 \ \text{for } l = 1, 2, \ldots, k, \quad 1 \leqslant i \leqslant l$$

($\hat{a}_{ij}^{(k)}$ are computed values, $a_{ij}^{(k)}$ are exact values, and $\epsilon_{ij}^{(k)}$ are rounding errors).

If $\hat{m}_{ik} = fl(\hat{a}_{ik}^{(k)} / \hat{a}_{kk}^{(k)})$, for $i \geqslant k+1$ denote multipliers at each step, we define

$$L^{(k)} = \begin{vmatrix} 0 & \ldots & 0 & \ldots & 0 \\ 0 & & 0 & & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & & \hat{m}_{k+1,k} & & 0 \\ & & \hat{m}_{k+2,k} & & \\ \vdots & & \vdots & & \vdots \\ 0 & \ldots & \hat{m}_{n,k} & \ldots & 0 \end{vmatrix} \tag{2.4}$$

so that

$$\hat{A}^{(k+1)} = \hat{A}^{(k)} - L^{(k)}\hat{A}^{(k)} + \epsilon^{(k)} \tag{2.5}$$

where $E^{(k)} = (\epsilon_{ij}^{(k)})$ for $k = 2, 3, \ldots, n$.

Since the matrix $L^{(k)}\hat{A}^{(k)}$ depends upon only the k th row of $\hat{A}^{(k)}$, and this row is equal to the kth row of $\hat{A}^{(n)}$, we have

$$LU = A + E, \tag{2.6}$$

where

$$L = L^{(1)} + L^{(2)} + \cdots L^{(n-1)} + I$$

and

$$E = E^{(1)} + E^{(2)} + \cdots + E^{(n)} = (\epsilon_{ij}). \tag{2.7}$$

Using the above rationale, Forsythe and Moler [7], derived the upper bounds for the size of E and δA, which are

$$\| E \|_\infty \leqslant n^2 \cdot \max_{i,j,k} \left| \hat{a}_{ij}^{(k)} \right| \cdot u, \tag{2.8}$$

and

$$\| \delta A \|_\infty \leqslant 1.01(n^3 + 3n^2) \cdot \max_{i,j,k} \left| \hat{a}_{ij}^{(k)} \right| \cdot u, \tag{2.9}$$

where u is the unit of round off error.

Suppose x is the exact solution of $Ax = b$. Then $x = A^{-1}b$ and $x - x^* = A^{-1}\delta A x^*$, and therefore

$$\begin{aligned} \| x - x^* \|_\infty &\leqslant \| A^{-1} \|_\infty \| \delta A \|_\infty \| x^* \|_\infty \\ &\leqslant 1.01 \cdot (n^3 + 3n^2) \cdot \| A^{-1} \|_\infty \cdot \max_{i,j,k} \left| \hat{a}_{ij}^{(k)} \right| \cdot \| x^* \|_\infty \cdot u. \end{aligned} \tag{2.10}$$

Since $\max_{i,j,k} |\hat{a}_{ij}^{(k)}|$ can be obtained during the forward course of Gaussian. Elimination, all the values on the right hand side of (2.10) are computable except $\| A^{-1} \|_\infty$. If a bound for $\| A^{-1} \|_\infty$ can be found, then a computable upper bound for $\| x - x^* \|_\infty$ will result. We will now derive a computable upper bound for $\| A^{-1} \|_\infty$ by using an approximation to A^{-1}.

Let X be an approximation of A^{-1}. In this case X is chosen as a computed result of A^{-1} using floating-point arithmetic. If $R = I_n - XA$ then

$$A^{-1} = (I_n - R)^{-1} X, \tag{2.11}$$

and

$$\| A^{-1} \|_\infty \leqslant (1 - \| R \|_\infty)^{-1} \| X \|_\infty \tag{2.12}$$

provided that $\| R \|_\infty < 1$. According to (2.12) the $\| X \|_\infty$ is computable, therefore, the task of finding a computable upper bound for $\| A^{-1} \|_\infty$ is replaced by the task of finding a computable upper bound for $\| R \|_\infty$. To derive the upper bound for $\| R \|_\infty$, the following two lemmas, proved by Yamamoto [13], are needed. We shall denote the computed approximation to the value a by $\hat{a}$, and let $\nu[\,]$ be the function such that $\nu[A] = \nu[(a_{ij})] = (|a_{ij}|)$. Let $\theta_n = nu$, where u is the unit of round off error.

Lemma 2.1. *Let* $K = (k_{ij}) = \nu[I_n - XA] = \nu[R]$ *and*

$$F = (f_{ij}) = 1.01\nu[X]\nu[A] + 1.01n^{-1} \begin{bmatrix} \hat{k}_{11} & & 0 \\ & \ddots & \\ 0 & & \hat{k}_{nn} \end{bmatrix}.$$

Then,

$$\hat{K} = K + \delta K \quad with \quad \nu[\delta K] \leqslant \theta_n F. \tag{2.13}$$

Lemma *2.2. Let*

$$k_i = \sum_{j=1}^{n} k_{ij}, \qquad \hat{k}_i = fl(\hat{k}_{i1}, + \ldots + \hat{k}_{in})$$

and

$$\boldsymbol{f} = (f_1, f_2, \ldots, f_n)' \quad where\ f_i = \sum_{j=1}^{n} f_{ij} + 1.01 \sum_{j=1}^{n} \hat{k}_{ij}.$$

Then,

$$\hat{\boldsymbol{k}} = \boldsymbol{k} + \delta \boldsymbol{k} \quad with \quad \nu[\delta \boldsymbol{k}] \leqslant \theta_n \boldsymbol{f},$$

where

$$\boldsymbol{k} = (k_1, k_2, \ldots, k_n)', \qquad \hat{\boldsymbol{k}} = (\hat{k}_1, \hat{k}_2, \ldots, \hat{k}_n)'. \tag{2.14}$$

Employing these two lemmas, we have

$$\| \boldsymbol{f} \|_\infty \leqslant 1.01\left(\| X \|_\infty \cdot \| A \|_\infty + n^{-1} \max_i \hat{k}_{ii} + 1.01 \max_i \hat{k}_i \right) \tag{2.15}$$

and

$$\| R \|_\infty \leqslant \max_i \hat{k}_i + \theta_n \| \boldsymbol{f} \|_\infty. \tag{2.16}$$

If we let

$$\hat{X}_\infty = \max_i fl(|x_{i1}| + |x_{i2}| + \cdots + |x_{in}|),$$

$$\hat{A}_\infty = \max_i fl(|a_{i1}| + |a_{i2}| + \cdots + |a_{in}|), \qquad \hat{\boldsymbol{k}}_\infty = \max_i \hat{k}_i,$$

then

$$\| \boldsymbol{f} \|_\infty < 1.01\left[(1 - 1.01\theta_{n-1})^{-2} \hat{X}_\infty \hat{A}_\infty + n^{-1} \max_i \hat{k}_{ii} + (1 - 1.01\theta_{n-1})^{-1} \max_i \hat{k}_i \right]$$
$$= \hat{\boldsymbol{f}}_\infty \tag{2.17}$$

and

$$\| R \|_\infty < \hat{\boldsymbol{k}}_\infty + \theta_n \hat{\boldsymbol{f}}_\infty. \tag{2.18}$$

Hence, if $\hat{\boldsymbol{k}}_\infty + \theta_n \hat{\boldsymbol{f}}_\infty < 1$, we have $\| R \|_\infty < 1$ and from (2.12)

$$\| A^{-1} \|_\infty \leqslant \left[1 - (\hat{\boldsymbol{k}}_\infty + \theta_n \hat{\boldsymbol{f}}_\infty)\right]^{-1} (1 - 1.01\theta_{n-1})^{-1} \hat{X}_\infty = \hat{A}_\infty^{-1}. \tag{2.19}$$

Thus, we have the principle result of this section:

Theorem 2.1. *Let x * be the computed solution of the linear system Ax = b by using partial pivoting Gaussian Elimination. If* $\hat{\boldsymbol{k}}_\infty + \theta_n \hat{\boldsymbol{f}}_\infty < 1$, *then* $\| A^{-1} \|_\infty \leqslant \hat{A}_\infty^{-1}$ *and, according to* (2.10),

$$\| \boldsymbol{x} - \boldsymbol{x}^* \|_\infty < 1.01(n^3 + 3n^2) \cdot \hat{A}_\infty^{-1} \cdot \max_{i,j,k} \left| \hat{a}_{ij}^{(k)} \right| \cdot \| \boldsymbol{x}^* \|_\infty \cdot u = \hat{\epsilon}^*, \tag{2.20}$$

where x is the exact solution of Ax = b.

This is a computable upper bound for $\|x-x^*\|_\infty$, and is an a posteriori bound because it involves knowing the actual computer result.

3. An upper bound for $|\det(A)|$

The Hadamard Inequality

$$|\det(A)| \leqslant \prod_{i=1}^{n}\left(\sum_{j=1}^{n} a_{ij}^2\right)^{1/2}$$

is often used to determine the upper bound for $|\det(A)|$, however, in most cases Hadamard's bound is quite conservative. An alternative upper bound, assuming $|\det(A)|$ is obtained as a byproduct using Gaussian Elimination with partial pivoting, is derived below.

Employing (2.3) and (2.7), we have

$$\hat{A}^{(n)} = A^{(n)} + E \tag{3.1}$$

where $\hat{A}^{(n)} = (\hat{a}_{ij})$, $A^{(n)} = (a_{ij})$, and $E = (\epsilon_{ij})$. Therefore,

$$a_{ii}^{(n)} = \hat{a}_{ii}^{(n)} - \epsilon_{ii} \quad \text{for } 2 \leqslant i \leqslant n$$

and

$$\left|a_{ii}^{(n)}\right| \leqslant \left|a_{ii}^{(n)}\right| + \|E\|_\infty \leqslant \left|\hat{a}_{ii}^{(n)}\right| + n^2 \cdot \max_{i,j,k}\left|\hat{a}_{ij}^{(k)}\right| \cdot u \quad \text{by (2.8)}$$

$$\leqslant \left|a_{ii}^{(n)}\right| \cdot \left[1 + \left(n^2 \cdot \max_{i,j,k}\left|\hat{a}_{ij}^{(k)}\right| / \min_{2\leqslant i\leqslant n}\left|\hat{a}_{ii}^{(n)}\right|\right) \cdot u\right]. \tag{3.2}$$

Furthermore,

$$fl\left\{\left|\hat{a}_{11}^{(n)}\right| \cdot \prod_{i=2}^{n}\left[\left|\hat{a}_{ii}^{(n)}\right|\left[1 + \left(n^2 \cdot \max_{i,j,k}\left|\hat{a}_{ij}^{(k)}\right| / \min_{2\leqslant i\leqslant n}\left|\hat{a}_{ii}^{(n)}\right|\right) \cdot u\right]\right]\right\}$$

$$\geqslant fl\left\{\prod_{i=1}^{n}\left|a_{ii}^{(n)}\right|\right\} \geqslant \prod_{i=1}^{n}\left|a_{ii}^{(n)}\right|(1-\theta_1)^{n-1} \geqslant |\det(A)|(1-\theta_1)^{n-1}. \tag{3.3}$$

Since $(1-\theta_1) > 0$

$$|\det(A)| < (1-\theta_1)^{1-n} fl\left\{\left[1 + \frac{n^2 \max_{i,j,k}\left|\hat{a}_{ij}^{(k)}\right|}{\min_{2\leqslant i\leqslant n}\left|\hat{a}_{ii}^{(n)}\right|} \cdot u\right] \cdot \prod_{i=1}^{n}\left|\hat{a}_{ii}^{(n)}\right|\right\}$$

$$< (1 - 1.01(n-1)u)^{-1} fl\left\{\left[1 + \frac{n^2 \max_{i,j,k}\left|\hat{a}_{ij}^{(k)}\right|}{\min_{2\leqslant i\leqslant n}\left|\hat{a}_{ii}^{(n)}\right|} \cdot u\right] \cdot \prod_{i=1}^{n}\left|\hat{a}_{ii}^{(n)}\right|\right\} = \hat{Q}^*. \tag{3.4}$$

Again, this is a computable upper bound for $|\det(A)|$ and in most cases this bound is smaller than Hadamard's bound.

4. Extracting the exact solution of the system of linear equations

For a given rational number $r = p/q$, let x be a floating-point approximation to r for which it is known that $|x - r| < \epsilon, q \leqslant Q$ and $\epsilon < 1/2Q^2$. Then the following process, developed and verified by Aberth [1], can be used to extract r from x.

Algorithm 1

Step 1. Record the sign of x.
Set $b_0 = |x|, p_{-2} = 0, p_{-1} = 1, q_{-2} = 1$ and $q_{-1} = 0$.

Step 2. Iterate
$a_k = [b_k]$ (i.e., the greatest integer $\leqslant b_k$),
$p_k = a_k p_{k-1} + p_{k-2}$,
$q_k = a_k q_{k-1} + q_{k-2}$,
$b_{k+1} = (b_k - a_k)^{-1}$,
for $k = 0, 1, 2, \ldots, K$ where K is the first instance of k that either $q_{k+1} > Q$ or b_{k+1} is undefined (i.e., $b_k = a_k$).

Step **3.** Let $r = (\text{sign } x) p_K/q_K$.

Similarly, for a given integral system Ax = b, where A is $n \times n$ non-singular matrix, the above process can be used to extract the exact components of the solution x from its computed floating-point number x^*, provided $\epsilon^* = \|x - x^*\|_\infty < 1/2 \det(A)^2$. Both ϵ^* and det(A) will not be known a priori, however, they can be replaced by $\hat{\epsilon}^*$ and $\hat{Q}^*$, respectively, because $\hat{\epsilon}^* < 1/2\hat{Q}^{*2}$ implies that $\hat{\epsilon}^* \geqslant 1/2 \det(A)^2$. Therefore, if $\hat{\epsilon}^* < 1/2\hat{Q}^{*2}$, we can extract the exact solution x from x^* components by using Algorithm 1. This is obviously an alternative to residue arithmetic and finite segment p-adic number arithmetic which produces an exact solution for the system Ax = b whenever the rational problem can be scaled to integer within a 'reasonable' range of integer values.

5. Numerical examples

The following three examples demonstrate the extraction procedure. The program is coded in FORTRAN using double precision arithmetic and was run on a CDC Cyber 730 System.

Example 1.

$$A = \begin{bmatrix} 22 & 10 & 2 & 3 \\ 14 & 7 & 10 & 0 \\ -1 & 13 & -1 & -11 \\ 1 & 8 & 1 & -2 \end{bmatrix}, \quad b = \begin{bmatrix} 25 \\ 10 \\ 55 \\ 105 \end{bmatrix}.$$

$\hat{k}_\infty + \theta_4 \hat{f}_\infty$ is 2.5031 x 10^{-13}, hence the assumption of Theorem 2.1 is satisfied.

The solution using floating-point arithmetic is (showing 10 digits)

-9.862281355
18.53905674
1.829863669
17.64001473

Hadamard's bound is 60 831. The $\hat{Q}^*$ value is 10 856.00000001.

The computed $\hat{\epsilon}^*$ is 6.84 $\times 10^{-12}$ which is less than $1/2\hat{Q}^{*2}$ hence the exact solution can be extracted. In this case we find the solution to be

$$\begin{array}{l} -4655/472 \\ 50\,315/2714 \\ 19\,865/10\,856 \\ 47\,875/2714 \end{array}$$

Example 2.

$$A = \begin{bmatrix} 68.0 & 25.0 & 11.0 & -26.0 & 55.0 \\ 66.0 & -36.0 & -32.0 & -51.0 & 17.0 \\ 46.0 & 26.0 & 56.0 & -85.0 & 74.0 \\ 9.0 & 31.0 & 2.0 & -69.0 & -11.0 \\ -73.0 & 60.0 & 47.0 & -48.0 & -80.0 \end{bmatrix}, \quad b = \begin{bmatrix} 5 \\ 10 \\ 15 \\ 20 \\ 25 \end{bmatrix}.$$

$\hat{k}_\infty + \theta_5 \hat{f}_\infty$ is 2.38 x 10^{27}, hence the assumption in Theorem 2.1 is satisfied.

The solution using floating-point arithmetic is (showing 10 digits)

$$\begin{array}{r} 0.1084333957 \\ 0.0673963154 \\ 0.0743188777 \\ -0.2130862346 \\ -0.1893841555 \end{array}$$

Hadamard's bound is 1.43 $\times 10^{10}$. The $\hat{Q}^*$ value is 928648912.000000000000000003.

The $\hat{\epsilon}^*$ value is 2.92 $\times 10^{-27}$ which is clearly less than $1/2\hat{Q}^{*2}$ therefore the exact solution can be extracted. The exact solution is

$$\begin{array}{l} 100\,696\,555/928\,648\,912 \\ 62\,587\,515/928\,648\,912 \\ 69\,016\,145/928\,648\,912 \\ -49\,470\,575/232\,162\,228 \\ -87\,935\,695/464\,324\,456 \end{array}$$

Example 3.

$$A = \begin{bmatrix} -8.0 & -7.0 & 0.0 & 2.0 & -4.0 & -5.0 & 3.0 & -8.0 \\ 3.0 & -6.0 & -5.0 & 2.0 & -4.0 & 1.0 & -4.0 & 3.0 \\ 8.0 & 3.0 & -6.0 & -5.0 & -9.0 & -8.0 & -3.0 & 1.0 \\ -4.0 & -1.0 & -7.0 & 9.0 & 3.0 & 5.0 & -2.0 & -3.0 \\ -4.0 & -7.0 & 1.0 & 0.0 & -3.0 & 5.0 & 2.0 & 0.0 \\ -9.0 & 0.0 & 2.0 & -8.0 & -4.0 & 1.0 & -1.0 & 2.0 \\ -6.0 & -9.0 & 6.0 & 1.0 & 0.0 & 8.0 & 3.0 & -3.0 \\ 6.0 & -9.0 & 6.0 & 6.0 & -4.0 & -9.0 & -9.0 & -2.0 \end{bmatrix}, \quad b = \begin{bmatrix} 5.0 \\ 10.0 \\ 15.0 \\ 20.0 \\ 25.0 \\ 30.0 \\ 35.0 \\ 40.0 \end{bmatrix}$$

$\hat{k}_\infty + \theta_8 \hat{f}_\infty$ is 9.79 x 10^{-27}, hence the assumption in Theorem 2.1 is satisfied.

The solution using floating-point arithmetic is (showing 10 digits)

$$\begin{array}{r} -0.9986101595 \\ 4.679542683 \\ 4.523992104 \\ 4.097929425 \\ -10.08165004 \\ 5.611403234 \\ -4.535300125 \\ -2.862671881 \end{array}$$

Hadamard's bound is 1.56×10^9. The $\hat{Q}^*$ value is 22282414.00000000000000000008.

The $\hat{\epsilon}^*$ value is 1.24×10^{-24} which is clearly less than $1/2\hat{Q}^{*2}$ therefore the exact solution can be extracted. The exact solution is

$$\begin{array}{l} -22\,251\,445/22\,282\,414 \\ 104\,249\,225/22\,282\,414 \\ 100\,805\,465/22\,282\,414 \\ 45\,655\,880/11\,141\,207 \\ -112\,321\,750/11\,141207 \\ 8\,931\,115/1\,59\,601 \\ -101057\,435/22\,282\,414 \\ -2\,899\,420/1\,012\,837 \end{array}$$

References

[1] 0. Aberth, A method for exact computation with numbers, *J. Comput. Appl. Math. 4 (1978) 285-288.*

[2] E.O. Adegbeyeni and E.V. Krishnamurthy, Finite field computation technique for exact solution of systems of linear equations and interval linear programming problems, *Znternat. J. Systems* Sciences 8 (1977) 1181-1192.

[3] I. Borosh and A.S. Fraenkel, Exact solutions of linear equations with rational coefficients by congruence techniques, *Math. Comput. 20 (1966) 107-112.*

[4] S. Cabay and T.P.L. Lam, Congruence techniques for the exact solution of integer systems of linear equations, *ACM Trans. Math. Software 3 (1977) 386-397.*

[5] S. Cabay and T.P.L. Lam, Algorithm 522 ESOLVE, congruence techniques for the exact solution of integer systems of linear equations F4, *ACM Trans. Math. Software 3 (1977) 404-410.*

[6] S.F. Chang, Error-free computation in solution of linear systems and linear programming problems, Ph.D. dissertation, Library, Iowa State University, Ames.

[7] G.E. Forsythe and C.B. Moler, *Computer Solution of Linear Algebraic Systems* (Prentice-Hall, Englewood Cliffs, NJ, 1967).

[8] G.E. Forsythe, M.A. Malcolm and C.B. Moler, *Computer Methods for Mathematical Computations* (Prentice-Hall, Englewood Cliffs, NJ, 1977).

[9] J.A. Howell and R.T. Gregory, Solving linear equations using residue arithmetic-Algorithm II, *Nordisk Tidskrift for Znformationsbehandling 10 (1970) 23-37.*

[10] W.J., Jr. Kennedy and J.E. Gentle, *Statistical Computing* (Marcel Dekker, New York, 1980).

[11] M. Newman, Solving equation exactly, *J. Res. Nat. Bureau Standards-B. Math. and Math. Phys.* **71B** *(1967) 171-179.*

[12] F.W.J. Olver and J.H. Wilkinson, A posteriori error bounds for Gaussian elimination, *SIAM J. Numer. Anal. 2 (1982) 377-406.*

[13] T. Yamamoto, A posteriori componentwise error estimate for a computed solution of a system of linear equations, *Publ. RIMS, Kyoto University 18 (1982)* 101–117.