УДК 621

B. R. Sastry, K.N. Murty, V.V.S.S.S. Balaram
Aurora's Engineering College
(India, Bhongir, Nalgonda Dist.—508116 (AP), E-mail: nkanuri@hotmail.com)

General First Order Matrix Difference System — Existence and Uniqueness via New Lattice Based Cryptographic Construction

(Recommended by Prof. E. Dshalalow)

This paper is concerned with the existence and uniqueness of solutions to two-point boundary value problems associated with general first order matrix difference systems. Modified Gram — Schmidt process and modified QR-algorithm are presented to find the best least square solution of the system of equations. An efficient closest point search algorithm is presented to further improve the best least square solution. Modified encoding and decoding algorithms are presented in the process of finding shortest lattice vector.

Рассмотрено существование и единственность решений двухточечных граничных задач, связанных с обобщенными матричными разностными системами первого порядка. Для нахождения наилучшего решения системы уравнений методом наименьших квадратов использован модифицированный процесс Грама—Шмидта и модифицированный QR-алгоритм. Для дальнейшего улучшения решения наименьших квадратов представлен эффективный алгоритм поиска ближайшей точки. В процессе нахождения кратчайшего вектора решетки получены модифицированные алгоритмы кодирования и декодирования.

K e y w o r d s: two-point boundary value problem, difference system, existence and uniqueness.

1. Introduction. Difference equations play a crucial role in understanding discrete phenomena of nature. The theory of difference equations is a lot richer than the corresponding theory of differential equations. For example, a simple difference equation resulting from a first order differential equation may have a phenomenon often called appearance of «Ghost» solutions or the existence of chaotic orbits that can only happen for higher order differential equations. Consequently, the theory of difference equations is interesting in itself and assumes great importance in solving real world problems. The application of the theory of difference equations is already extended to various fields such as cryptology, numerical analysis, finite element techniques, computer science and controllability. All these reasons inspired us to consider the general first order matrix difference system of the form

$$T(n+1) = AT(n)B + F(n), \tag{1.1}$$

where A, B and F are all square matrices of order s and, whose elements are real, defined on $N_{n_0}^{+} = \{n_0, n_0 \pm 1, n_0 \pm 2, ..., n_0 \pm k, ...\}$, where $k \in N^{+}$ and $n_0 \in N$, N being the set of integers.

In communication engineering, cryptology played a crucial rule to enhance security. In the year 1992, the National Bureau of Standards (NBS), now the National Institute of Standards and Technologies (NIST) initiated a program to protect computer and communication data. The intricacies of relating key variations or key variables is of special importance. In communication theory lattices are used for modulation and quantization [1]. A comprehensive survey of closest point search methods for lattices without a regular structure are presented in a recent paper by E. Agrell et. al [2]. They also presented existing search strategies in a unified framework and highlighted differences between them. The closest point problem mainly deals with the problem of finding a given lattice Λ and a given input point $x \in R^n$, a vector $\hat{y} \in \Lambda$ such that

$$\|x - \hat{y}\| \le \|x - y\|, \text{ for all } y \in \Lambda,$$

where $\|\cdot\|$ denotes the Euclidean norm. In channel coding the closest point problem is often referred to as decoding and this is the terminology used by many authors in recent years. If a lattice is used as a code for Gaussian channel, the maximum likelihood of decoding in the demodulator is a closest point search. Analogously, if a lattice is used as a codebook for vector quantization and the mean square error criterion is used then the encoding of each input vector is also closest point search. The method for solving the closest point problem, in fact depends on the structure of the lattice. Intuitively the more structure a lattice has, the faster can the closest point be found. A common approach to the general closest point problem is to identify a certain critical point region in R^n within which the optimal lattice point lie, and then investigate all lattice points in the region and thereby reducing the size dynamically. For a comprehensive review on closest point search, we refer to an excellent survey made by E. Agrell et al [2].

This paper is organized as follows. In section 2, we present the general solution of the homogeneous matrix difference system

$$T(n+1) = AT(n)B \tag{1.2}$$

in terms of two fundamental matrix solutions of $T(n+1) = AT(n)$ and $T(n+1) = B^{*}T(n)$ and then develop a particular solution of (1.1) by using variation of parameters formula. Section 3, presents a criteria for the existence and uniqueness of solution to two-point boundary value problem

$$\begin{aligned} T(n+1) &= AT(n)B + F(n), \\ MT(n_0) + NT(n_f) &= \alpha, \end{aligned} \tag{1.3}$$

where n_0, $n_f \in N_{n_0}$, $n_0 < n_f$, M, N and α are given constant square matrices of order s. In section 4, we present a generalized inverse concept and the method of residual to find the best least square solution of the system of equations $Ax = b$. Modified Gram—Schmidt process is presented in section 5. Closest point search algorithm is presented in the last section. Section 5 also presents MINLS algorithm and then the best least square solution using modified QR algorithm.

2. Solution of the non-homogeneous system. In this section, we present the general solution of the homogeneous system (1.2) in terms of two fundamental matrix solutions and then develop variation of parameters formula for the non-homogeneous difference system (1.1). We shall denote $\Phi(n, n_0)$ and $\overline{\Psi}(n, n_0)$ as the fundamental matrix solutions of $T(n+1) = AT(n)$ and $T(n+1) = B^* T(n)$ respectively. With this notation, the proof of the following lemma is immediate.

L e m m a 2. 1. $\Phi(n, n_0)$ is a fundamental matrix solution of $T(n+1) = AT(n)$ if, and only if $\Phi^*(n, n_0)$ is a fundamental matrix solutions of $T(n+1) = T(n) A^*$ ($*$ refers to the transpose of the complex conjugate matrix).

Theorem 2.1. Let $\Phi(n, n_0)$ and $\overline{\Psi}(n, n_0)$ be two fundamental matrix solution of $T(n+1) = AT(n)$ and $T(n+1) = B^* T(n)$ respectively. Then any solution $T(n)$ of (1.2) is of the form $T(n) = \Phi(n, n_0) C \psi^*(n, n_0)$ where C is a constant square matrix of order s.

P r o o f. We seek a solution $T(n)$ of (1.2) in the form $T(n) = \Phi^*(n, n_0) K(n)$, where $K(n)$ is a square matrix of order s whose elements are defined on $N_{n_0}^+$. Then

$$\Phi(n+1, n_0) K(n+1) = A\Phi(n, n_0) K(n) B \Leftrightarrow A\Phi(n, n_0) K(n+1) =$$
$$= A\Phi(n, n_0) K(n) B \Leftrightarrow K(n+1) = K(n) B \Leftrightarrow K^*(n+1) = B^* K^*(n).$$

Since $\bar{\psi}(n, n_0)$ is a fundamental matrix solution of $T(n+1) = B^* T(n)$, it follows that there exists an $(s \times s)$ constant matrix C^* such that $K^*(n) = \bar{\psi}(n, n_0) C^*$ and hence $T(n) = \Phi(n, n_0) K(n) = \Phi(n, n_0) C \bar{\psi}^*(n, n_0)$.

Theorem 2.2. Any solution $T(n)$ of (1.1) is of the form $T(n) = \Phi(n, n_0) \times$ $\times C \bar{\psi}^*(n, n_0) + \overline{\boldsymbol{T}}(n)$, where $\overline{\boldsymbol{T}}(n)$ is a particular solution of (1.1).

P r o o f. It can easily be verified that $T(n)$ defined by $T(n) = \Phi(n, n_0) \times$ $\times C \bar{\psi}^*(n, n_0) + \overline{\boldsymbol{T}}(n)$ is a solution of (1.1). Now, to prove that every solution of (1.1) is of this form, let $T(n)$ be any solution of (1.1) and $\overline{\boldsymbol{T}}(n)$ be a particular solution of (1.1). Then $T(n) - \overline{\boldsymbol{T}}(n)$ is a solution of (1.2). Hence by theorem 2.1, $T(n) - \overline{\boldsymbol{T}}(n) = \Phi(n, n_0) C \bar{\psi}^*(n, n_0)$.

Therefore $T(n) = \overline{\boldsymbol{T}}(n) + \Phi(n, n_0) C \bar{\psi}^*(n, n_0)$.

Theorem 2.3. A particular solution $\overline{\boldsymbol{T}}(n)$ of (1.1) is given by

$$\overline{\boldsymbol{T}}(n) = \sum_{j=n_0}^{n-1} \Phi(n, j+1) F(J) \bar{\psi}^*(n, j+1).$$

P r o o f . Any solution of the homogeneous system (1.2) is of the form $T(n) = \Phi(n, n_0) C \bar{\psi}^*(n, n_0)$. Such a solution cannot be a solution of (1.1) unless $F(n) \equiv 0$. Let C be a function of n defined on $N_{n_0}^+$ and seek a particular solution of (1.1) in the form $\overline{\boldsymbol{T}}(n) = \Phi(n, n_0) C(n) \bar{\psi}^*(n, n_0)$.

Since $\overline{\boldsymbol{T}}(n)$ must satisfy (1.1) we have

$$\Phi(n+1, n_0) C(n+1) \bar{\psi}^*(n+1, n_0) = A\Phi(n, n_0) C(n) \bar{\psi}^*(n, n_0) B + F(n) \Leftrightarrow$$

$$\Leftrightarrow \Phi(n+1, n_0) \Delta C(n) \bar{\psi}^*(n+1, n_0) = F(n) \Leftrightarrow$$

$$\Leftrightarrow \Delta C(n) = \Phi(n_0, n+1) F(n) \bar{\psi}^*(n_0, n+1) \Leftrightarrow$$

$$\Leftrightarrow C(n) = \left(Cn_0 + \sum_{j=n_0}^{n-1} \Phi(n_0, j+1) F(j) \bar{\psi}^*(n_0, j+1) \right).$$

Thus

$$T(n) = \Phi(n, n_0) Cn_0 \bar{\psi}^*(n, n_0) + \Phi(n, n_0) \left(\sum_{j=n_0}^{n-1} \Phi(n_0, j+1) F(j) \bar{\psi}(n_0, j+1) \right) =$$

$$= \Phi(n, n_0) C \bar{\psi}^*(n, n_0) + \overline{T}(n).$$

3. Two-point boundary value problem. In this section, we consider the two-point boundary value problem associated with the non-homogeneous general matrix difference system (1.1), satisfying the boundary condition

$$MT(n_0) + NT(n_f) = \alpha, \tag{3.1}$$

where $n_0, n_f \in N_{n_0}^*$, $n_0 < n_f$. Substituting the general form of the solution given in (2.1) in the boundary condition matrix (3.1), we get $M\Phi(n_0, n_0) C_{n_0} \bar{\psi}^*(n_0, n_0) +$

$$+ M\Phi(n_f, j+1) F(j) \bar{\psi}^*(n_f, j+1) = \alpha - M \sum_{j-n_0}^{n-1} \Phi(n_f, j+1) F(j) \bar{\psi}^*(n_f, j+1).$$

The above equation is equivalent to

$$M_1 C_{n_0} N_1 + M_2 C_{n_0} N_2 = Y, \tag{3.2}$$

where

$$M_1 = M\Phi(n_0, n_0),\ N_1 = \bar{\psi}^*(n_0, n_0),\ M_2 = M\Phi(n_f, n_0),\ N_2 = \bar{\psi}^*(n_f, n_0),$$

and $Y = \alpha - M \sum_{j=n_0}^{n-1} \Phi(n_f, j+1) F(j) \bar{\psi}^*(n_f, j+1)$ are all known square matrices of order s. Note that N_1 and N_2 are in fact fundamental matrix solutions and hence invertible in the usual sense. For the analysis of C_{n_0}, we employ the following notion on kronecker product of matrices. Note that $\Phi(n_0, n_0) = \bar{\psi}(n_0, n_0) = I$.

If $A, B \in R^{s\times s}$ are two square matrices of orders s, then their Kronecker product or tensor product denoted by $(A \otimes B)$ is defined as

$$A \otimes B = a_{ij} B \text{ for all } i = 1,2,...,s;\ j = 1,2,...,s = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1s}B \\ a_{21}B & a_{22}B & \cdots & a_{2s}B \\ a_{s1}B & a_{s2}B & \cdots & a_{ss}B \end{bmatrix}$$

and is in $R^{s^2 \times s^2}$ With this one can easily verify that if

$$A = (M_1 \otimes N_1^*) + (M_2 \otimes N_2^*),$$

then (3.2) is equivalent to be

$$AC_{n_0} = y, \tag{3.3}$$

where A is an $(s^2 \times s^2)$ matrix and C_{n_0} and y are column matrices of order $s^2 \times 1$ corresponding to the square matrices C_{n_0} and Y. In fact by viewing (3.3) as a system of s^2 scalar equations for the elements of C_{n_0}, (3.3) is exactly the same set of equations written in a vector form.

4. Closest point search in lattices. The problem of finding a shortest, non-zero lattice vector in a lattice of dimension s^2 is a landmark problem in complexity theory. Lenstra A. K., Lenstra H. W. and Lavasz L. [3] known as LLL – algorithm is used in basic reduction criteria. Kannan [4] has proposed an algorithm to find the shortest lattice vector in time $n^{0(n)}$, which was later improved by Helfrich [5] to $n^{n/2+0(n)}$. The LLL reduction is often used in most cases whereas the Korkine—Zotareff (KZ) reduction is time consuming. In this section, we present the modified Gram—Schimidt process of Rice [6] for the com-

putation of a best least square solution of (3.3) in the general case. For, we consider the general first order matrix system of equations

$$Ax = b, \tag{4.1}$$

where A is an $(m \times n)$ matrix and x is an n-vector (unknown) and b is a given $(m \times 1)$ vector. The problem is to find the existence of solutions of the system (4.1). If A is singular and if $R(A)$ and $N(A)$ represent respectively, the range and null spaces of A, then (4.1) will have solutions if $b \in R(A)$. In this case if x is any n-vector in $N(A)$ and $\bar{x}$ is any solution of (4.1) then the vector $x + \bar{x}$ will also be a solution. If $b \notin R(A)$, then the problem (4.1) will not have solutions.

If A is an $(m \times n)$ rectangular matrix, then for a given $A \in R^{m \times n}$ (or $C^{m \times n}$) and $b \in R^m$, the linear system (4.1) is consistent if, and only if $b \in R(A)$. Otherwise, the residual vector

$$R = b - Ax \tag{4.2}$$

is non-zero for all $x \in R^n$, and it may be desired to find an approximate solution of (4.1), by which we mean a vector x making the residual vector (4.2) «closest» to zero in some sense.

The following theorem shows that $\| Ax - b \|$ is minimized by choosing $x = A^+ b$, where A^+ is such that

$$AA^+A = A, \tag{4.3}$$

$$(AA^+)^* = AA^+. \tag{4.4}$$

Theorem 4.1. Let $A \in R^{m \times n}(C^{m \times n})$ and $b \in R^m(C^n)$, then $\| Ax - b \|$ is smallest when $x = A^+ b$ where A^+ satisfies (4.3) and (4.4). Conversely, if $A^+ \in R^{n \times m}$ has the properties, that for all b, $\| Ax - b \|$ is smallest when $x = A^+ b$, then A^+ satisfies (4.3) and (4.4).

P r o o f. We write $Ax - b = (Ax - P_{R(A)} b) + (P_{R(A)} b - b)$. Where $P_{R(A)}$ is the projection matrix on $R(A)$. Then

$$\|Ax - b\|^2 = \|Ax - P_{R(A)} b\|^2 + \|P_{R(A)} b - b\|^2. \tag{4.5}$$

Since $(Ax - P_{R(A)} b) \in R(A)$ and $-(I - P_{R(A)})\, x \in R(A)$; it follows that (4.5) assumes its minimum value if, and only if

$$Ax = P_{R(A)} b \tag{4.6}$$

which certainly holds, if $x = A^+ b$ for any A^+ satisfying (4.3) and (4.4). Hence $AA^+ = P_{R(A)}$. Conversely, if A^+ is such that for all b, $\| Ax - b \|$ is smallest when $x = A^+ b$, then by (4.6) we have $AA^+ b = P_{R(A)}^{x \vee x}$ and hence $AA^+ b = P_{R(A)}$. Thus A^+ satisfies (4.3) and (4.4). Hence the proof.

Suppose A^+ satisfies the following two conditions:

$$A^+ A\, A^+ = A^+, \tag{4.7}$$

$$(A^+ A)^* = A^+ A. \tag{4.8}$$

Then we have the following theorem.

Theorem 4.2. Let $A \in R^{m\times n}(C^{m\times n})$, $x \in R^m(C^m)$. If $Ax = b$ has a solution for x, the unique solution for which $\| x \|$ is smallest is given by $x = A^+ b$, where A^+ satisfies (4.7) and (4.8). Conversely, if $A^+ \in R^{n\times m}(C^{n\times m})$, is such that, whenever $Ax = b$ has solution, $x = A^+ b$ is the solution with minimum norm, then A^+ satisfies (4.7) and (4.8).

P r o o f . By Theorem 4.1, equation (4.1) has a unique solution say x_0 in $R(A^*)$. Now the general solution is given by

$$x = x_0 + y$$

for some $y \in N(A)$. Clearly $\|x\|^2 = \|x+0\|^2 + \|y\|^2$ proving that $\|x\| \geq \|x_0\|$ and equality holds only if $x = x_0$.

5. Modified Gram—Schmidt process. Let A be an $(m\times n)$ matrix of rank $p \leq \min\{m, n\}$. The algorithm discussed here depends upon the rank factorization of the form $AP = QR$, where P is an $(n\times m)$ permutation matrix such that the first P columns of AP are linearly independent, Q is an $(m\times p)$ matrix with orthonormal columns, and R is an upper-trapezoidal of rank p. We shall denote $\mathrm{Im}(A) = \{Ax \in R^m / x \in R^n\}$, the column space of A and $\ker(A) = \{x \in R^n / Ax = 0\}$. We further need the following results for constructing least square algorithm and the best least square algorithm [6].

Results 5.1. Let A be an $(m\times n)$ given matrix of rank p. Then there exists a factorization $AP = QR$ with the following properties:

(i) P is an $(n\times n)$ permutation matrix with the first p columns of AP form a basis of Im(A);

(ii) Q is an $(m\times p)$ matrix with orthonormal columns and R is a $(p\times n)$ upper trapezoidal matrix of the form $R = [R_1, R_2]$ where R_1 is a non-singular $(p\times p)$ upper triangular matrix and R_2 is a $p\times(n-p)$ matrix.

Result 5.2. Let A be an $(m\times n)$ matrix with rank p. Write $A = [a_1, a_2, ..., a_n]$, where $a_j \in R^m$ and let P be an $(n\times n)$ permutation matrix such that $AP = QR$ where Q is an $(m\times p)$ matrix with orthonormal columns and R is a $(p\times n)$ upper trapezoidal of rank p. Then the first p columns of AP are linearly independent and all the least square solutions of this system $Ax = b$ can be obtained by solving the consistent system $RP^T x = Q^* b$.

If we write $R=[R_1, R_2]$, R_1 is a $(p\times p)$ upper triangular, then

$$\bar{x} = P\begin{bmatrix} u \\ v \end{bmatrix}_{n-p}^{p},$$

where $v \in R^{n-p}$ is arbitrary and $u = R_1^{-1}(Q^* b - R_2 v)$ are the least square solutions of $Ax = b$. A basic least square solution is obtained by making $v = 0$.

Algorithms. Let A be an $(m\times n)$ matrix and $b \in R^m$ is given and let rank A be $p \le \min\{m, n\}$. The following is the algorithm to compute least square solution. We use the notation $a_{ij} := b_{ij}$ if a_{ij} becomes b_{ij} for all $i = 1, 2, ..., M$, and $j = 1, 2, ..., n$.

(i) A l g o r i t h m:

$q_{ij} := a_{ij}$, $i = 1, 2, \ldots, m$; $j = 1, 2, \ldots, n$
$r_{ij} := 0$, $i = 1, 2, \ldots, m$; $j = 1, 2, \ldots, n+1$
$s_j := j$, $j = 1, 2, \ldots, n$
$p := n$

for $k = 1, 2, \ldots, n$ $\delta_j = \sum_{i=1}^{m} |q_{ij}|^2, j = k, k+1, ..., n$

COMPUTE INDEX c, $k \le c \le n$ such that $\delta_c = \max\delta_j$ $1 \le j \le n$

IF $\delta_c = 0$, go to 30
30 $p := k - 1$ go to 40
interchange column k of Q with column C of Q
interchange column k of R with column C of R
interchange number δ_k with number δ_c,
interchange index S_k with index C
$r_{kk} := \sqrt{\delta_k}$
$q_k := q_k / r_{kk}$.
$r_{kj} := q_k^* q_j$, $j = k + 1, ..., n$
$q_j := q_j - r_{kj} q_k$, $j = k + 1, ..., n$
$r_k, n + 1 := q_k^* b$.
40 for $j = p + 1, ..., n$
$x_j := 0$.

Back solve the system of equations

$r_{11} x_1 + ... + r_{1p} x_p = r_{1,n+1}$
$r_{22} x_2 + ... + r_{2p} x_p = r_{2,n+2}$
.....
$r_{pp} x_p = r_{p,n+1}$

to determine $x_1, x_2, ..., x_p$.

For $j = n, n-1, ..., 1$

k: $=S_j$

if $k \neq j$, $x_k \leftrightarrow x_j$

$x = (x_1, x_2, ..., x_n)$ is a least square solution of $Ax = b$

(ii) A l g o r i t h m M I N L S

IF $p = n$ STOP

The least square solution already found is the minimal norm least square solution of $Ax = b$.

Else $v := n - p$

$b_j := x_j$, $j = 1, ..., n$

$x_j := 0$, $j = p+1, ..., n$

for $k = p + 1, ..., n$

$x_k := 1$.

Back solve the equation system $Rx = 0$, to determine $x_1, x_2, ..., x_p$

$j := k - p$

$aij := x_i$, $i = 1, 2, ..., n$

$x_k := 0$.

For $i = n, n-1, ..., 1$.

$k : = S_i$

IF $K \neq i$, interchange a_{kj} and a_{ij} for $j = 1, 2, \ldots, v$.

(iii) C o m p u t a t i o n o f t h e p s e u d o i n v e r s e.

If A is an $(m \times n)$ matrix, then we can utilize algorithm MINLS to compute A^+, the pseudo inverse of A. Using MINLS m times, solve for a_i^+ the minimal norm least square solution of the problem $Ax = e_i$, where $e_i, 1 \leq i \leq m$ are the standard Euclidian basis for R^m. Then $A^+ = [a_1^+, a_2^+, ..., a_p^+]$.

The vector $x = (x_1, x_2, ..., x_p)$ computed from the above algorithm is the closet point search algorithm. This algorithm as detailed above is better than the Sehnorr norm — Euchner strategy algorithm [7] and that of [2]. We can apply this algorithm to find a unique solution of the boundary value problem (1.3). We need the following DECODE algorithm and the closet Point algorithm. Before we present these algorithms in the next section, we first present with suitable examples to find minimum least square solutions of equations $Ax = b$.

E x a m p l e 1. Consider the system of equations $Ax = b$, where

$$A=\begin{bmatrix}1 & 1 & 1\\ 2 & -1 & -1\\ 2 & -4 & 5\end{bmatrix},\ x=\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}\ \text{and}\ b=\begin{bmatrix}4\\ -1\\ 1\end{bmatrix}.$$

Using QR factorization, we find $AP = QR$, where

$$Q=\frac{1}{3}\begin{bmatrix}-1 & 2 & 2\\ -2 & 1 & -2\\ -2 & -2 & 1\end{bmatrix},\ R=3\begin{bmatrix}-1 & 1 & -1\\ 0 & 1 & -1\\ 0 & 0 & 1\end{bmatrix}.$$

Note that Q is orthonormal and R is upper trapezoidal. Using the algorithms given in sections 4 and 5, we find that the minimum least square solution is given by MINLS $x = [0.99999, 2.00001, 1.00000]^T$.

E x a m p l e 2. Consider the system of equations $Ax = b$, where

$$A=\begin{bmatrix}1 & 2 & 3\\ 1 & 5 & 6\\ 1 & 8 & 9\\ 1 & 11 & 12\end{bmatrix},\quad x=\begin{bmatrix}x1\\ x2\\ x3\end{bmatrix}\ \text{and}\ b=\begin{bmatrix}6\\ 13\\ 19\\ 24\end{bmatrix}.$$

This is an over determined system. Using the algorithms given in sections 4 and 5, we find that the minimum least square solution is given by MINLS $x = [1, 0.5, 1.5]^T$.

E x a m p l e 3. Consider the system of equations $Ax = b$, where

$$A=\begin{bmatrix}1 & 3 & 3 & 2\\ 2 & 6 & 9 & 5\\ 1 & 3 & -3 & 0\end{bmatrix},\ x=\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\end{bmatrix}\ \text{and}\ b=\begin{bmatrix}1.5\\ 6\\ 2.25\end{bmatrix}.$$

Note that the system is an underdetermined system and its minimum least square solution is MINLS $x = [-0.211009174, -0.6330275230, 0.963302752, 0.110091743]^T$.

6. Closest point search algorithm. The concept of public-key cryptography algorithm have a long mathematical history right from 1976. Many of the public-key cryptography were in secure. Of those still considered secure, many are not practicable. Either they have too large a key or the Cipher text is much larger than the plain text. Only a very few algorithms are both secure and practical. The method, we present here is more secure and more practical and is based on closest point general lattice search algorithm. This algorithm can be regarded

as a «front end» to Decode, where explicit pre processing and post processing is performed to allow generator matrices that are not lower triangular and are not even square matrices. We first present an algorithm that computes a closest vector without any representation choice, but the speed with which it reaches the required result varies significantly between two different representations. We then present a DECODE algorithm which is of practical importance. The main question we answer in this section is the following: How should a given search problem be preprocessed in order to make the most efficient use of DECODE ?

Definition 6.1. A matrix G is said to be a generator matrix if it has real entries and rows of G are linearly independent over R.

Firstly, we assume that a generator matrix A and an input vector x are given. Let A be an $(m\times n)$ matrix and $x \in R^m$. By means of a linear integer transformation, we first transform A into another matrix R_2 which generates an identical lattice and then rotate and reflect R_2 into a lower triangular matrix R_3, so that $\Lambda(R3) \approx \Lambda(R2) = \Lambda(A)$.

It is very essential to rotate and reflect the input vector x in the same way, so that the transformed input vector, say x_3, is in the same relation to $\Lambda(R3)$ as x is related to $\Lambda(A)$. All this can be regarded as a change of coordinate system. Note that by the above transformation the input vector x also changes, so that the transformed input vector becomes x_2. By reversing the operations of rotation and reflection enables us to produce $\overset{\Lambda}{x}$, which is the lattice point closest to x in $\Lambda(A)$. Following the above steps, we are now in a position to present the detailed algorithm as follows.

A l g o r i t h m. CLOSEST POINT (A, x).

Input: A lattice point $\overset{\Lambda}{x} \in \Lambda(A)$ the closest to x.

S t e p 1. Let $R_2 = WA$ where W is an $(m\times m)$ matrix with integer entries and $\det W = \pm 1$.

S t e p 2. Compute an $(m\times n)$ orthogonal matrix Q with orthonormal columns such that $R_2 = R_3\, Q$, where $R3$ is an $(m\times n)$ lower-triangular matrix with all diagonal elements positive.

S t e p 3. Let $H_3 := \boldsymbol{R}_2^{-1}$.

S t e p 4. Let $x_3 := \boldsymbol{xQ}^T$.

S t e p 5. Let $\overset{\Lambda}{u}_3$: = DECODE (R_3, x_3).

S t e p 6. Return $\overset{\Lambda}{x} := \overset{\Lambda}{u}_3 R_2$.

Step 1 is in fact a basic reduction. If no basic reduction is needed, we can take W as the unit matrix. Note that the speed and numerical stability of the search can be improved significantly if proper search is made. Step 2 implies ro-

tation and reflection of R_2 into lower triangular form. The usual method to achieve this is modified algorithm of QR presented in section 4. In our context QR decomposition of R_2 gives both Q^T and R_3 with R_3 being equal to R^T. All the transformation can be thought of a change of coordinate system. Measure first coordinate along v_1 (the first row of R_2) the second in the rows spanned by v_1 and v_2 and the third in the row spanned by v_1, v_2 and v_3 and so on. The generator matrix in this coordinate system will be in general square and lower triangular.

As an alternative to QR decomposition, R_3 can be obtained by Cholesky decomposition and it states that one can find a lower triangular matrix (real) L such that $A = LL^T$. In our context, R_3 is equal to L and the rotation matrix is given by $Q = R_3^{-1^T} R_2$. Another approach to find QR decomposition is decomposition of $A = LU$, where L is lower triangular and U is upper triangular and in our context $Q = \boldsymbol{R}_3^T R2$. If A is an $(m \times n)$ positive definite matrix, its cholesky decomposition is a factorization of A in the form $A = UU^T$ where U is an $(m \times m)$ — upper triangular matrix. In our context, R_3 is equal to U^T and the rotation matrix Q is given by $Q = \boldsymbol{R}_3^{-1} R2$. All these algorithms can be found in [8]. One can also compute QR by Householder's reflection and the Householder matrix H is symmetric and orthogonal. The Household reflection in fact reflect every vector $x \in \boldsymbol{R}^m$ in the hyperplane span $\{v\}^{\perp}$ and is given by $H = I - \dfrac{2vv^T}{v^T v} = \boldsymbol{H}^T$, where v is the Householder vector. However, the QR-method is the generally recommended method for calculating the least square solutions so far and in this paper, we replaced QR algorithm by the modified QR-algorithm and this method is the most effective tools in finding the least square solutions of the system of equations $Ax = b$. Note that the decomposition of $A = QR$ is unique, what ever technique we adopt and further in our modified QR-algorithm Q is orthonormal implies $\det(Q) = \pm 1$. For A is an ill-conditioned matrix, the method we presented to our belief is the most effective tool.

In steps 4—6 the input vectors are processed. They are transformed into the coordinate system of R_2 decode, and transformed back again.

We now present DECODE algorithms.

Algorithm DECODE (H, x)

Input: an $m \times m$ lower triangular matrix H with positive diagonal element, and an m-dimensional Vector $x \in \boldsymbol{R}^m$ to decode in the lattice $\Lambda(\boldsymbol{H}^{-1})$.

Output: an m-dimensional vector $\hat{u} \in z^m$ such that $\hat{u}\boldsymbol{H}^{-1}$ is a lattice point $\hat{x}$ that is close to x.

1. m: = the Size of H \ * dimension* \
2. bestdist : = ∞ \ * current distance record *\
3. K : = m * dimension of the matrix cude examination*\

4. dis s_k := 0 * distance to examined layer *\
5. e_k: xH * $\overset{\Lambda}{u}m = \frac{xv}{11v^2}$ used to compute $\overset{\Lambda}{u}m^*$ *\1
6. u_k := e_{kk} * examined lattice point \
7. y: = $\frac{e_{kk} - u_k}{h_{kk}}$ * m= \$u_m - \overset{\Lambda}{u}_m$\ $\|v\perp\|$ is the orthonormal distance *\
8. Step k : = Sgn*(y) *off set to next layer in (15)*\
9. loop
10. new dis : = Dist $k + y^2$
11. if new dist < best dist then {
12. if $k \neq 1$ then {
13. $e_{k-1,i}$: = $e_{ki} - y_{hki}$ for i – 1, 2, ..., k – 1
14. k: = k – 1
15. dist k: New dist
16. u_k : = e_{kk}
17. y: = $\frac{e_{kk} - u_k}{h_{kk}}$
18. step k: = sgn*(y)
19. }else{
20. $\overset{\Lambda}{u}$: = u
21. best dist : = new dist
22. k:= k + 1
23. u_k : = u_k + step k
24. y : = $\frac{e_{kk} - u_k}{h_{kk}}$
25. Step k : = Step k – sgn* (step k)
26. }
27. }else {
28. if $k = m$ then return $\overset{\Lambda}{u}$ (and exit)
29. else {
30. k: = k +1 *move up*\
31. u_{kj} : = u_k+Step k *next layer*\
32. y : = $\frac{e_{kk} - u_k}{h_{kk}}$
33. go to step 25
34. }
35. }
36. go to <loop>

In the above algorithm m, k is the dimension of the sublayer structure that is currently being investigated. In case A is an ill-conditioned, the algorithm finds a k-dimensional layer, the distance to which is less than the currently smallest distance, this layer is expanded into $(k - 1)$ dimensional sub layer. Conversely, if the distance to the examined layer is greater than the lower distance the algorithm moves one step up in the hierarchy of layer. This is done in case 6. Case B is invoked when the algorithm has been successfully moved down all the way down to the zero dimensional layer without exceeding the lowest distance (that is, a lattice point). This lattice point is stored in the output, the lowest distance is updated, and the algorithm moves back up again, without restarting [2]. Note that in [2], the closest point search algorithm is based upon carefully selected preprocessing. Such a selection is not possible in each and every case. The methods we presented in section 4 and 5 will eliminate such careful selection and minimizes decoding time and at the same time reduce the complexity of the closest point search significantly.

Розглянуто існування та єдиність розв'язувань двоточечних граничних задач, зв'язаних з узагальненими матричними різницевими системами першого порядку. Для пошуку найкращого розв'язування системи рівнянь методом найменших квадратів використано модифікований процес Грама—Шмідта і модифікований QR-алгоритм. Для подальшого покращення розв'язування найменших квадратів наведено ефективний алгоритм пошуку найближчої точки. У процесі пошуку найкоротшого вектора решітки знайдено модифіковані алгоритми кодування та декодування.

1. *Blacke I. F.* «Lattices and Cryptography» in codes and systems /R. E Blahut and R.Kotter, eds. — Norwell : M. A Kluloev, 2002. — P. 317—332.
2. *Agrell E., Eriksson T., Vardy A., Zeger K.* Closen-point search in lattices//IEE Transactions on transformation theory. — 2002. — **148**, № 8. — P. 2201—2215.
3. *Lenstra A.K., Lenstra H.W, Lovasz L.* Factoring Polynomials with rational coefficients// Math.Ann. — 1982. — **261**. — P. 515—534.
4. *Kannan R.* Minkowski's convex body theorems and integer programming Mathematics of operations research. —1987. —**12**. — P. 415—440.
5. *Hel Frich B.* Algorithm to construct Minkowski reduced and Hermite reduces bases//Theoretical computer science. — 1985. — **41**. — P.125—139.
6. *Rice J.R.* Experiments of Gram—Schmidt orthogonalization Math. computers. — 1996.— **20**. — P. 325—328.
7. *Atkinson K.* An Introduction to Numerical Analysis, Second edition. — NY, Brisbane, Toronto: John Wiley and sons, 1987. — P. 209.
8. *Schnorr C. P.* A more efficient algorithm for Lattice based reduction//J. Algorithms. — 1988. — **9**. — P. 47—62.

Поступила 18.08.06