Available online at www.isr-publications.com/jnsa
J. Nonlinear Sci. Appl., 12 (2019), 720–727
Research Article

ISSN: 2008-1898

Journal of Nonlinear Sciences and Applications

Journal Homepage: www.isr-publications.com/jnsa

Three-Point boundary value problems associated with first order matrix difference system-existence and uniqueness via shortest and closest Lattice vector methods

Kasi Viswanadh V. Kanuri[a,*], K. N. Murty[b]

[a]3669 Leatherwood, Dr. Frisco, TX 75033, USA.

[b]Department of Applied Mathematics, Andhra University, Waltair (A.P), 530017, India.

Abstract

In this paper, we shall be concerned with the existence and uniqueness of solution to three- point boundary value problems associated with a system of first order matrix difference system. Shortest and Closest Lattice vector methods are used as a tool to obtain the best least square solution of the three-point boundary value problem when the characteristic matrix D is rectangular. An efficient decode algorithm is presented to find the shortest and closest vector and prove that this vector is the best least square solution of the three-point boundary value problem.

Keywords: Matrix difference system, fundamental matrix, closest and shortest vector methods, decode algorithms.

2010 MSC: 34B15, 93B05, 93B15.

1. Introduction

In this paper, we shall be concerned with the existence and uniqueness of solutions to first order linear system of difference equation

$$y_{n+1} = A(n)\, y_n + b_n, \tag{1.1}$$

where A is an (s× s) matrix whose elements $a_{ij}(n)$ are real or complex functions defined on N_{n0}^{+} and $y_n \in R^s\,(\text{or}C^s)$ with components that are functions defined on the same set N_{n0}^{+}. The corresponding homogeneous linear difference equation is

$$y_{n+1} = A(n)\, y_n. \tag{1.2}$$

When the initial vector $y(n_0) = y_0$ is given, both (1.1) and (1.2) determine the solution uniquely on the set N_{n0}^{+} as can be easily seen by induction. Let $e_1, e_2, \ldots, e_s$ be the standard base vectors in R_s and

*Corresponding author
Email addresses: vis.kanuri@gmail.com (Kasi Viswanadh V. Kanuri), nkanuri@hotmail.com (K. N. Murty)

doi: 10.22436/jnsa.012.11.03

Received: 2019-04-19 *Revised:* 2019-05-14 *Accepted:* 2019-05-28

$y(n, n_0, e_i)$, $i = 1, 2, \ldots, s$ be the s linearly independent solutions having e_i as initial base vectors and let S be the solution space of (1.2). It may be noted that any element of S can be expressed as a linear combination of the set of n solutions of $y(n, n_0, e_i)$, $i = 1, 2, \ldots, s$ i.e., if Z_n is any solution then

$$Z_n = \sum_{i=1}^{n} c_i y(n, n_0, e_i), \ i = 1, 2, \ldots, k.$$

We define K functions $f_i(n)$ on $N_{n_0}^+$ as

$$K(n) = \begin{bmatrix} f_1(n) & f_2(n) & \cdots & f_k(n) \\ f_1(n+1) & f_2(n+1) & \cdots & f_k(n+1) \\ \vdots & \vdots & \ddots & \vdots \\ f_1(n+k-1) & f_2(n+k-1) & \cdots & f_k(n+k-1) \end{bmatrix}.$$

If $f_i(n), i = 1, 2, \ldots, k$ are solutions of $y_{n+1} = A(n)y_n$, then $\det K(n) \neq 0$ for all $n \geqslant n_0$, provided that $\det(K(n_0)) \neq 0$. If y_n is any solution of (1.1) and $\bar{y}_n$ is a particular solution of (1.1), then $y_n - \bar{y}_n$ is a solution of (1.2) and any solution of (1.2) is of the form $\sum_{i=1}^{s} \alpha_i y(n, n_0, \alpha_i)$, where $\alpha_1, \alpha_2, \ldots, \alpha_s$ are arbitrary constants and hence

$$y_n = y_n + \sum_{i=1}^{k} \alpha_i y(n, n_0, \alpha_i).$$

To establish our main result, we use the qualitative properties of difference systems as developed in [9]. For linear system of differential equations we refer to [13].

We consider the three point boundary value problem

$$y(n+1) = A(n)\, y(n) + b(n), \tag{1.3}$$

$$My(n_0) + Ny(n_m) + Ry(n_f) = \alpha, \tag{1.4}$$

where b is an $(s \times 1)$ vector, M, N, and R are constant square matrices of order s, and α is an $s \times 1$ vector. Let $\varphi(n, n_0)$ be a fundamental matrix of $y(n+1) = A(n)y_n$. Then, if $y(n)$ is any solution of (1.3) and y_n is a particular solution of (1.3) then $y(n) - y(n)$ is a solution of (1.2) and any solution of (1.2) is of the form $\varphi(n, n_0)\, C$, where C is an $(s \times 1)$ vector. Thus

$$y(n) = y(n) + \varphi(n, n_0)\, C.$$

In the algorithm we discussed in Section 3, we assume that the boundary condition matrices M, N, and R are constant rectangular matrices of order $(r x s)$.

Theorem 1.1. *A particular solution* $y(n)$ *of* (1.3) *is given by*

$$y(n) = \sum_{j=n_0}^{n-1} \varphi(n, j+1)\, b_j.$$

Proof. Any solution of the homogeneous system (1.2) is of the form $y(n) = \varphi(n, n_0)\, C$, where C is a constant n-vector. Such a solution cannot be a solution of (1.3), unless $b(n) \equiv 0$. Let C be a function of n defined on $N_{n_0}^+$ and seek a particular solution of (1.3) in the form

$$y(n) = \varphi(n, n_0)\, C(n).$$

Then

$$\varphi(n+1, n_0)\, C(n+1) = A(n)\, \varphi(n, n_0)\, C(n) + b(n),$$

$$\varphi(n+1,n_0)\,[C(n+1)-C(n)] = b(n),$$
$$\varphi(n+1,n_0)\,\Delta C(n) = b(n),$$

or

$$\Delta C(n) = \varphi(n_0,n+1)\,b(n).$$

Therefore,

$$C(n) = C(n_0) + \sum_{j=n_0}^{n-1} \varphi(n_0,j+1)\,b(j).$$

Thus, any solution $y(n,n_0,C)$ can be written as

$$y(n,n_0,C(n)) = \varphi(n,n_0)\,C_{n_0} + \sum_{j=n_0}^{n-1} \varphi(n,n_0)\,\varphi(n_0,j+1)\,b_j = \varphi(n,n_0)\,C_{n_0} + \sum_{j=n_0}^{n-1} \varphi(n,j+1)\,b_j. \quad (1.5)$$

Note that $\varphi(n,n_0)\,\varphi(n_0,j+1) = \varphi(n,j+1)$. □

The above formula is known as the method of variation of constants formula. Note further that $\varphi^{-1}(n,n_0) = \varphi(n_0,n)$ and the fact that $|A(n)| \neq 0$ is not in fact necessary to derive variation parameters formula as given in [6]. If the components of $A(n)$ are continuous functions on $N_{n_0}^{+}$, then a fundamental matrix $\varphi(n,n_0)$ always exists.

2. Three-point boundary value problems

In this section, we consider the three-point boundary value problem associated with the non-homogeneous matrix difference system (1.3) satisfying the boundary conditions:

$$My(n_0) + Ny(n_m) + Ry(n_f) = \alpha, \quad (2.1)$$

where n_0, n_m and $n_f \in N_{n_0}^{+}$, $n_0 < n_m < n_f$. Substituting the general form of the solution given in (1.5) in the boundary condition matrix (2.1), we get

$$[M\varphi(n_0,n_0) + N\varphi(n_m,n_0) + R\varphi(n_f,n_0)]\,C_{n_0}$$
$$= \alpha - \left[M\sum_{j=n_0}^{n-1} \varphi(n_0,j+1)\,b_j + N\sum_{j=n_0}^{n-1} \varphi(n_m,j+1)\,b_j + R\sum_{j=n_0}^{n-1} \varphi(n_f,j+1)\,b_j\right].$$

We assume that the homogeneous three-point boundary value problem is in-compatible, and it ensures that the characteristics matrix D defined by

$$D = MI + N\varphi(n_m,n_0) + R\varphi(n_f,n_0)$$

is non-singular. Hence

$$C_{n_0} = D^{-1}\left[\alpha - \left(M\sum_{j=n_0}^{n-1} \varphi(n_0,j+1)\,b_j + N\sum_{j=n_0}^{n-1} \varphi(n_m,j+1)\,b_j + R\sum_{j=n_0}^{n-1} \varphi(n_f,j+1)\,b_j\right)\right].$$

Note that $\varphi(n_0,n_0) = I$. If D is non-invertible, then

$$DC = f,$$

where

$$f = D^{-1}\left[\alpha - \left(M\sum_{j=n_0}^{n-1} \varphi(n_0,j+1)\,b_j + N\sum_{j=n_0}^{n-1} \varphi(n_m,j+1)\,b_j + R\sum_{j=n_0}^{n-1} \varphi(n_f,j+1)\,b_j\right)\right].$$

We solve the system of equations by using closest and shortest Lattice vector problems only in the least square sense. We present some algorithms on this in the next section.

3. Algorithms for finding shortest and closest Lattice vector problem

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 et al. [7], popularly known as LLL-algorithm is used in basic reduction criteria, by factoring polynomials. Kannan [3, 4] has proposed an algorithm to find the shortest lattice vector in time $n^{o(n)}$, which was later improved by Schnorr [11] to $n^{\frac{n}{2}+o(n)}$. The LLL-reduction is often used in most cases, where as the Korkine-Zolotarev (KZ) [5] reduction is time consuming. For that reason we present the modified Gram-schimidt process of Rice [10] for the computation of the best least square solution of (2.1) in the general case. For that, we consider the general first order matrix system of the form,

$$DC = f, \tag{3.1}$$

where D is an $(m\times n)$ rectangular matrix and C is an n-vector (unknown) and f is a given $(m \times 1)$ vector. If the columns of D are linearly independent, then we multiply the system of equations (3.1) by D^T and get

$$D^T DC = D^T f.$$

We assume that $m < n$. In this case C is determined uniquely as

$$C = \left(D^T D\right)^{-1} D^T f.$$

On the other hand if the rows of D are linearly independent then, the transformation $C = D^T y$ gives

$$DD^T y = f \quad \text{or} \quad y = \left(DD^T\right)^{-1} f.$$

Hence $C = D^T y$ gives

$$C = D^T \left(DD^T\right)^{-1} f.$$

In both cases, we can determine the vector C uniquely. If D is an $(r \times s)$ rectangular matrix and C is a s-vector (unknown) and f is a given $(r \times 1)$ vector then the system of equations (3.1) can have either an infinite number of solutions or no solution. On the other hand, if D is singular and $R(D)$ and $N(D)$ represent, respectively, the range and the null space of D, then (3.1) will have a solution if $f \in R(D)$. In this case if C is any s-vector in $N(D)$ and C_0 is any solution of (3.1), then the vector $C + C_0$ will also be a solution of (3.1). Otherwise if $\notin R(D)$, the problem (3.1) will not have solution. If D is an $(r \times s)$ matrix, then for a given $D \in C^{r\times s}$ and $f \in C^r$, the linear system (3.1) is consistent if and only if $b \in R(D)$. Otherwise the residual vector

$$r = f - DC \tag{3.2}$$

is non-zero for all $C \in C^n$, and it may be desired to find an approximate solution of (3.1), by which we mean a vector C making the residual vector in (3.2) "Closest" or "shortest" to zero in some sense.

The next theorem shows that $\|DC - f\|$ is minimized by choosing $C = D^+ f$, where D^+ is such that

$$DD^+ D = D, \tag{3.3}$$

$$\left(DD^+\right)^* = DD^+. \tag{3.4}$$

Theorem 3.1. *Let* $D \in C^{r\times s}$ *and* $f \in C^r$. *Then* $\|DC - f\|$ *is smallest when* $C = D^+ f$, *where* D^+ *satisfies* (3.3) *and* (3.4). *Conversely, if* $D^T \in C^{s\times r}$ *has the property that for all f,* $\|DC - f\|$ *is smallest when* $C = D^+ f$, *then* D^+ *satisfies* (3.3) *and* (3.4).

Proof. The proof of the theorem is similar to proof of the Theorem 1 in [8]. □

If D^+ satisfies the conditions

$$D^+ DD^+ = D^+, \quad \left(D^+ D\right)^* = \left(D^+ D\right),$$

then a similar result holds as in Theorem 3.1.

Our main interest in the paper is to obtain the best least square solution of (3.1) in the non-invertible case. In such situations, one normally seeks solutions which satisfy the boundary conditions exactly and solves the differential equation in the best least square sense. We now present few algorithms for finding the shortest and closest lattice vector problems and refine these algorithms by using modified encoding and decoding algorithms. Note that, the boundary condition matrices in (1.4), namely M, N, and R, are constant rectangular matrices of order (rxs).

4. Shortest and closest Lattice vector problems

The shortest and closest vector problem consists of finding a vector $C \in Z^s \backslash 0$ minimizing $\|DC\|$, where vector $D \in Z^{r\times s}$ is given as input. The closest Lattice vector problem consists in finding a vector $C \in Z^s$ minimizing $\|DC - f\|$, where $D \in Z^{r\times s}$ and $f \in C^r$ are given as input. SVP and CVP have been a topic of discussion for more than a century. Among many investigations, the works of Hermite [2] Korkine and Zolotarev [5], minkowski [8] and Voronoi [15] are of some importance. However, the algorithmic study of lattice took off rather lately at the beginning of 1980. At that time, lattice happened to be a bottleneck in combinational optimization and in algorithmic number theory and earthshaking results were then obtained by Lenstra et al. [7] proposed the first polynomial time approximation algorithm for SVP [7] whereas Van Emde Boas [14] showed that the decisional variant of CVP is NP-hard. Shortly afterwards, Kannan [3, 4] obtained first SVP and CVP algorithms in a fruitful way. In mid 1990s public key cryptosystems from lattice reduction problems came into existence. The practicality of SVP solves attracted much attention than their CVP counterparts due to the their importance in cryptology applications.

The concept of public key cryptography algorithm has a long mathematical history right from 1976. Many of the public key encryption using cryptographic techniques are insecure. Some of the public key cryptography algorithms developed are even though secure but they are not practicable. One reason may be they have a very large key or the ciphertext is much larger than the plaintext. Only a very few algorithms are both secure and practical. The method we describe in this section here are more secure and more practicable and are infact based on closest point general lattice search algorithms. We first present an algorithm that computes a closest vector without any representation choice, but the computational aspect with which it reaches the required result varies significantly. We present a decode algorithm which is of practical importance and very effective in finding shortest Lattice vector. The main question we answer in this section is the following.

How can a given search problem be processed in order to make the efficient use of decode?

Definition 4.1. G is said to be a generator matrix if it has real entries and has linearly independent rows over R.

Let D be given $(r \times s)$ matrix with real entries and $C \in R^r$. By means of a linear integer transformation we first transform D into another matrix R_2, which in fact generates an identical Lattice and then rotate and reflect R_2 to generate a lower triangular matrix R_3, so that

$$\bigwedge (R_3) \backsimeq \wedge (R_2) = \wedge (D).$$

Next, we rotate and reflect the input vector x in the same way, so that the transformed input vector, say xz, is in same relation to (R_3) as x in related to $n(A)$.

All this can be regarded as a change of coordinate system. It may be noted that by the above transformation the input vector x also changes, so that the transformed input vector becomes x_2. By reversing the operation of rotation and reflection enables us to produce x, which is the lattice point closest to x in (D). We are now in a position to present the detailed algorithm to compute closest vector point and this closest vector is the best least square solution of the system of equations

$$DC = f.$$

Algorithm 4.2 (Closest vector point (D, C))**.**

Input: A lattice point $C \in \bigwedge(D)$ the closest to C.
Step 1: Let $R_2 = \omega D$, where ω is an $(m \times n)$ matrix with integer entries and $\det \omega = \pm 1$.
Step 2: Compute an $(m \times n)$ orthogonal matrix Q with orthonormal columns such that $R_2 = R_3 Q$, where R_3 is an $(m \times n)$ lower-triangular matrix with all the diagonal elements greater then 0.
Step 3: Let $H_3 R_2^{-1}$.
Step 4: Let $x_3 C Q^T$.
Step 5: Let $U_3 := \text{decode}(R_3, C_3)$.
Step 6: Return $C U_3 R_2$.

Note that step1 is a basic reduction. If no basic reduction is needed, we can take ω as a unit matrix. Further, the speed and numerical stability of the search can be improved significantly, if proper search is made. Step 2 implies rotation and reflection of R_2 into lower triangular form. The usual method adopted to achieve this is presented below as an algorithm.

Result 4.3. Let D be an $(r \times s)$ given matrix with rank $p \leqslant \min\{r, s\}$. Then there exists a factorization $DP = QR$ with the following properties:

(i) P is an $(s \times s)$ permutation matrix with first P columns of DP form a basis of $\text{Im}(D)$;
(ii) Q is an $(r \times p)$ matrix with orthonormal columns and R is a $(p \times s)$ upper trapezoidal matrix of the form $R = [R_1, R_2]$, where R_1 is an non-singular $(p \times p)$ upper triangular matrix and R is a $(p \times s - p)$ matrix.

Result 4.4. Let D be an $(r \times s)$ with rank p. Write $D = [d_1, d_2, \ldots, d_n]$, where $d_j \in R^r$ and P be an $(s \times s)$ permutation matrix such that $DP = QR$, where Q is an $(r \times p)$ matrix with orthonormal columns and R is a $(p \times s)$ upper trapezoidal matrix of rank p. Further, $d_j \in R^r$ and the first p columns of AP are linearly independent and all the least square solutions of $DC = f$ are obtained by solving the consistent system

$$RP^T C = Q^* f.$$

Write $R = [R_1, R_2]$, where R_1 is $(p \times p)$ upper triangular. Then $C = P \begin{bmatrix} u \\ v \end{bmatrix}_{s-p}^{p}$, where $v \in R^{s-p}$ is arbitrary and $u = R_1^{-1}(Q^* f - R_2 v)$ are the least square solutions of $DC = f$. A basic least square solution is obtained by taking $v = 0$. To compute the least square solution of $DC = f$ we refer the algorithm presented in [8]. The modified QR-algorithm is also presented in [11]. In our context QR decomposition, R_2 gives both Q_T and R_3 with R_3 being equal to R_T. All the transformations can be thought of as a change of coordinate system. Measure first coordinate along v (the first row of R_2), the second in the rows spanned by v_1 and v_2, and the third in the rows spanned by v_1, v_2, and v_3 and so on. The generator matrix in this new coordinate system will be in general a square and lower triangular.

As an alternate to QR decomposition, R_3 can be obtained by cholesky decomposition and it states that one can find a real lower triangular matrix L such that $D = LL^T$. In our context, R_3 is equal to L and the rotation matrix is given by $Q = R_3^{-1^T} R_2$. Another way, to find QR decomposition is the decomposition of $D = LU$, where L is lower triangular and U is upper triangular and in our context $Q = R_3^T R_2$. All these algorithms can be found in [11].

In steps 4-6, the input vectors are processed. They are transformed into the coordinate system of R_2 decode, and transformed back again. One can also compute QR by house holders reflection. In fact reflect every vector $C \in R^r$ in the hyperplane span $\{v\}^+$ and is given by

$$H = I - \frac{2vv^T}{v^T v} = H^T,$$

where v is the house holder vector. However the modified QR algorithm is generally the recommended

one for calculating the least square solutions of the system of equations $DC = f$. In this paper we replace QR-algorithm by the modified QR algorithm given in [14]. Note that the decomposition $D = QR$ is unique, whatever technique we adopt and Q is orthonormal implies $\det(Q) = \pm 1$. The method works even if D is illconditioned. The methods, we presented in this paper are the most effective tools and gives the best least square solution of the three point boundary value problems associated with first order matrix difference system.

We now present the decode algorithms.

General algorithm for $DC = f$, where D is $(m \times n)$, C is $(n \times 1)$, and f is $(m \times 1)$ vector.

Algorithm 4.5 (algorithm decode (H , x))**.**

Input: an m-dimensional vector $u \in Z^m$ such that uH^{-1} is a Lattice point C that is closed to C;

1. m the size of H, /∗ dimension∗ /;
2. bestdist $:= \infty$, /∗ current distance record∗ /;
3. Km, /∗ dimension of the matrix under examination∗ /;
4. $diss_k 0$, /∗ distance to examined layer∗ /;
5. $e_k : x_H$;
6. $u_k := e_{kk}$, /∗ examined lattice point∗ /;
7. $y := \frac{e_{kk}-u_k}{h_{kk}}$, /∗ $m = |u_m - u_m|.\|V^{\perp}\|$ is the orthonormal distance distance∗ /;
8. stepk $:= sgn^*(y)$, /∗ offset to next layer in(15)∗ /;
9. loop;
10. newdis $:=$ disK $+ y^2$;
11. if newdis $<$ bestdist then{;
12. if$k \neq 1$then{;
13. $e_{k-1,j} e_{ki} - y_{nki}$ for $i = 1, 2, \ldots, k-1$;
14. $kk - 1$;
15. distk : newdis;
16. $u_k e_{kk}$;
17. $y := \frac{e_{kk}-u_k}{h_{kk}}$;
18. stepk $:= sg_n^*(y)$;
19. }else{;
20. uu;
21. Best dist := newdis;
22. $kk + 1$;
23. $u_k := u_k +$ step k;
24. $y := \frac{e_{kk}-u_k}{h_{kk}}$;
25. step k := step k $- sg_n^*$ (step k);
26. };
27. }else{;
28. if $k = m$ then return to u (and exit);
29. else{;
30. $k = k + 1$, /∗ move u_p∗ /;
31. $u_{kj} = u_k +$ step k, /∗ next layer ∗ /;
32. $y := \frac{e_{kk}-u_k}{h_{kk}}$;
33. go to step25;
34. };
35. };
36. go to $<$ loop $>$

In the above algorithm m, k is the dimension of the sub layer structure that is currently being investigated. In case, if A is an ill-conditioned matrix, then the above algorithm finds a K-dimensional layer, the distance to which is less than the smallest distance, and thus layer is expanded to $(k-1)$ 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 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). The lattice point stores in the output, the lowest distance is updated, and the algorithm moves back upon carefully selected preprocessing. Note that such a selection is not possible in each and every case. The method we presented will eliminate such careful selection and minimizes decoding time and at the same time reduces the complexity of the closest vector point search significantly.

References

[1] D. Ding, G. Zhu, X. Wang, *A Genetic Algorithm for Searching Shortest Lattice Vector of SVP challenge*, J. Elect. Inform. Tech., **35** (2013), 1940–1945.

[2] C. Hermite, *Oeuvres de Charles Hermite (French) [Works of Charles Hermite. Volume 1]*, Edited and with a foreword by Emile Picard. Reprint of the 1905 original. Cambridge Library Collection. Cambridge University Press, Cambridge, 2009. 4

[3] R. Kannan, *Improved algorithms for integer programming and related lattice problems*, Proceedings of the 15th annual ACM symposium on Theory of computing, **1983** (1983), 193–206. 3, 4

[4] R. Kannan, *Minkowski's Convex Body Theorem and Integer Programming*, Math. Oper. Res., **12** (1987), 415–440. 3, 4

[5] A. Korkine, G. Zolotareff, *Sur les formes quadratiques*, Math. Ann. (French), **6** (1873), 336–389. 3, 4

[6] V. Lakshmikantham, D. Trigante, *Theory of Difference Equations: Numerical Methods and Applications*, Academic Press, Boston, (1988). 1

[7] A. K. Lenstra, H. W. Lenstra, L. Lovász, *Factoring Polynomials with Rational Coefficients*, Math. Ann., **261** (1982), 515–534. 3, 4

[8] H. Minkowski, *Geometrie der zahlen*, Band 40 Johnson Reprint Corp., New York-London, (1968). 3, 4, 4.4

[9] K. N. Murty, S. Andreou, K. V. K. Viswanadh, *Qualitative properties of general first order matrix difference systems*, Nonlinear Stud., **16** (2009), 359–369. 1

[10] J. R. Rice, *Experiments on Gram–Schmidt Orthogonalization*, Math. Comp., **20** (1966), 325–328. 3

[11] C.-P. Schnorr, M. Euchner, *Lattice Basis Reduction: Improved Practical Algorithms and Solving Subset Sum Problems*, Math. Programming, **66** (1994), 181–199. 3, 4.4, 4

[12] V. P. Sreedharan, *A note on the modified Gram–Schmidt process*, Int. J. Comput. Math., **24** (1988), 277–290.

[13] N. Swapna, S. Udaya Kumar, K. N. Murty, *Best Least Square Solution of Boundary Value Problems Associated with a System of First Order Matrix Differential Equation*, Electronic Modeling, **38** (2015), 1–14. 1

[14] P. Van Emde Boas, *Another NP–complete partition problem and the complexity of computing short vectors in a lattice*, Department of Mathmatics, University of Amsterdam (Tecnical Report), Amsterdam, (1981). 4, 4

[15] G. Voronoi, *Nouvelles applications des paramétres continus á la théorie des formes quadratiques. Deuxiéme mémoire. Recherches sur les parallélloédres primitifs*, J. für Math., **134** (1908), 198–287. 4