AN ENHANCED JACOBI METHOD FOR LATTICE-REDUCTION-AIDED MIMO DETECTION

Zhaofei Tian and Sanzheng Qiao

McMaster University
Department of Computing and Software
Hamilton, Ontario, L8S 4K1, Canada

ABSTRACT

Lattice reduction aided decoding has been successfully used for signal detection in multiinput and multioutput (MIMO) systems and many other wireless communication applications. In this paper, we propose a novel enhanced Jacobi (short as EJacobi) method for lattice basis reduction. To assess the performance of the new EJacobi method, we compared it with the LLL algorithm, a widely used algorithm in wireless communications. Our experimental results show that the EJacobi method is more efficient and produces better results measured by both orthogonality defect and condition number than the LLL algorithm.

Index Terms— MIMO, Wireless communication, Signal processing algorithms, Least squares approximation, Lattices

1. INTRODUCTION

Recently, lattice reduction aided decoding has been successfully used in many signal processing applications, such as Global Positioning System (GPS) [1, 2], multiinput multioutput (MIMO) system [3], frequency estimation, and particularly data detection and precoding in wireless communication systems. See [4] for more details about lattice basis reduction with applications in wireless communications.

Consider an $m \times n$ MIMO system of n transmit antennas and m receive antennas. The relation between an $n \times 1$ transmitted signal $\mathbf{x}$ and an $m \times 1$ received signal $\mathbf{y}$ is modelled by

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}, \tag{1.1}$$

where $\mathbf{A}$ and $\mathbf{n}$ represent the channel matrix and the additional noise, respectively. The matrix $\mathbf{A}$ is complex in a full-rank flat-fading MIMO system, but it can be transformed into a real matrix of double size straightforwardly [4]. Hence, in this paper, we assume $\mathbf{A}$ is real. The optimum maximum likelihood (ML) decoding selects $\mathbf{x}_{ML}$ that is a solution for the following minimization integer least squares problem:

$$\mathbf{x}_{ML} = \arg\min_{\mathbf{x}\in\mathcal{A}} ||\mathbf{y} - \mathbf{A}\mathbf{x}||_2, \tag{1.2}$$

where $\mathcal{A}$ denotes the finite set of real-valued modulation alphabet being used. The complexity of solving (1.2) grows exponentially corresponding to the number of antennas [1, 5]. Hence ML decoding is not feasible for large number of transmit antennas. To reduce the decoding cost, many approximate algorithms have been introduced to achieve high performance with low complexity, such as zero-forcing (ZF) decoding and successive interference cancellation (SIC) decoding [3]. The performance of those decoding strategies heavily depends on the orthogonality of the columns of $\mathbf{A}$. Lattice reduction algorithms can improve the orthogonality of the columns of the channel matrix $\mathbf{A}$.

Suppose $\mathbf{A}$ is an $m \times n$ ($m \geq n$) real matrix of full column rank, a *lattice* generated by $\mathbf{A}$ is defined as $L(\mathbf{A}) = \{\mathbf{A}\mathbf{z} \mid \mathbf{z} \in \mathbb{Z}^n\}$, where $\mathbb{Z}^n$ is the set of all integer n-vectors. The columns of $\mathbf{A}$ and n are respectively called the *basis* and the *dimension* of the lattice. A lattice of dimension at least 2 has infinitely many bases [6]. Any two bases $\mathbf{A}$ and $\mathbf{A}'$ for the same lattice are related by a *unimodular* matrix $\mathbf{Z}$, i.e., $\mathbf{Z}$ is an integer matrix and $|\det(\mathbf{Z})| = 1$, such that $\mathbf{A} = \mathbf{A}'\mathbf{Z}$. For a given basis, the lattice reduction algorithms are aimed to find a *reduced* basis with relatively shorter and more orthogonal vectors. There are several notions of reduced basis, such as the *Minkowski reduced basis* [7, 8] and the *HKZ reduced basis* [9], both need exponential time to be computed. Another category of reduced basis can be found in polynomial time, such as the *Schnorr reduced basis* [10] and the widely used *LLL reduced basis* [11]. The latter is also used in many number theory applications in addition to wireless communications, for example in cryptosystems [12].

In this paper, we present an enhanced Jacobi method, the EJacobi method, for lattice basis reduction based on the *Jacobi* method [13] introduced by S. Qiao. Our experimental results indicate that the EJacobi method performs better than the LLL algorithm in both output quality and time consumption. The rest of the paper is organized as follows. In section 2, we review the Lagrange algorithm and the Jacobi method. Both of them can be regarded as fundamental algorithms for our new lattice reduction algorithm presented in section 3. In section 4, we demonstrate the experimental results of the comparison between the EJacobi method and the LLL algorithm. Finally, the paper is concluded in section 5.

2. JACOBI METHOD

In this section, we recall the Lagrange reduction algorithm [14], focusing on its one single iteration. The Lagrange iteration is a part of our EJacobi method presented in the section 3.3. We also recall the Jacobi method [13, 15] in this section.

2.1. Lagrange Algorithm

We call a two dimensional basis matrix $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2]$ *Lagrange-reduced* (*L-reduced*), if

$$||\mathbf{a}_1||_2 \leq ||\mathbf{a}_2||_2 \quad \text{and} \quad |\mathbf{a}_1^T \mathbf{a}_2| \leq \frac{1}{2}||\mathbf{a}_1||_2^2. \tag{2.1}$$

An L-reduced basis is Minkowski reduced [12, 14]. Denote θ the angle between $\mathbf{a}_1$ and $\mathbf{a}_2$, then $|\cos(\theta)| = |\mathbf{a}_1^T \mathbf{a}_2|/(||\mathbf{a}_1||_2||\mathbf{a}_2||_2) \leq |\mathbf{a}_1^T \mathbf{a}_2|/||\mathbf{a}_1||_2^2 \leq \frac{1}{2}$, implying that $\theta \in [\frac{\pi}{3}, \frac{2\pi}{3}]$ [13]. Thus, we may say that $\mathbf{a}_1$ and $\mathbf{a}_2$ are close to being orthogonal to each other.

The Lagrange algorithm is a polynomial time iterative method for computing an L-reduced basis. It deduces the length of one of the two vectors in each iteration. We now apply the Lagrange algorithm to a pair of columns $\mathbf{a}_i$ and $\mathbf{a}_j$, $i < j$, of a lattice basis matrix $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n]$. Let $\mathbf{G} = [g_{ij}] = \mathbf{A}^T\mathbf{A}$ be the *Gram Matrix*, then $g_{ij} = \mathbf{a}_i^T \mathbf{a}_j$ and $g_{jj} = ||\mathbf{a}_j||_2^2$. Given the Gram matrix $\mathbf{G}$ and $i < j$, Procedure LAGRANGEIT$(\mathbf{G}, i, j)$ produces a unimodular matrix $\mathbf{Z}$ [13] so that $\mathbf{AZ}$ performs one iteration of the Lagrange algorithm on $\mathbf{a}_i$ and $\mathbf{a}_j$, where $\mathbf{a}_i$ is updated by $\mathbf{a}_i \leftarrow \mathbf{a}_i - q\mathbf{a}_j$ and the length $\|\mathbf{a}_i\|_2$ is reduced, hence $\mathbf{a}_i$ and $\mathbf{a}_j$ get closer to being orthogonal to each other.

Procedure LagrangeIT($\mathbf{G}, i, j$)

Input : The Gram matrix $\mathbf{G}$ and a pair (i, j) of indices
Output: A unimodular matrix $\mathbf{Z}$, s.t. $\mathbf{AZ}$ performs one iteration of the Lagrange algorithm on $\mathbf{a}_i$ and $\mathbf{a}_j$

$q = \lfloor \frac{g_{ij}}{g_{jj}} \rceil$; `// Nearest integer rounding`
Set $\mathbf{Z} = I_n$ except $z_{ji} = -q$;

Having Procedure LAGRANGEIT$(\mathbf{G}, i, j)$, Algorithm 1 produces a unimodular matrix $\mathbf{Z}_{ij}$, such that the ith and jth columns of $\mathbf{AZ}_{ij}$ form an L-reduced basis for the lattice $L([\mathbf{a}_i\ \mathbf{a}_j])$.

2.2. Jacobi Method for Lattice Basis Reduction

In 1846, C. Jacobi originally proposed a method for solving eigenvalue problems of real symmetric matrices [16, 17]. S. Qiao introduced a Jacobi method for lattice basis reduction in 2012 [13], which embeds the Lagrange algorithm to reduce every pair of basis vectors in an n dimensional lattice and produces a reduced basis defined below.

Algorithm 1: Lagrange2($\mathbf{G}, i, j$)

Input : The Gram matrix $\mathbf{G}$ and a pair of indices (i, j)
Output: Updated $\mathbf{G}$ and a unimodular matrix $\mathbf{Z}_{ij}$, s.t. the ith and jth columns of $\mathbf{AZ}_{ij}$ form an L-reduced basis

$\mathbf{Z}_{ij} = I_n$;
if $g_{ii} < g_{jj}$ **then**
  Swap the ith and jth columns of $\mathbf{G}$;
  Swap the ith and jth rows of $\mathbf{G}$;
  Swap the ith and jth columns of $\mathbf{Z}_{ij}$;
repeat
  $\mathbf{Z} =$ LAGRANGEIT$(\mathbf{G}, i, j)$;
  $\mathbf{G} \leftarrow \mathbf{Z}^T\mathbf{G}\mathbf{Z}$;
  $\mathbf{Z}_{ij} \leftarrow \mathbf{Z}_{ij}\mathbf{Z}$;
  Swap the ith and jth columns of $\mathbf{G}$;
  Swap the ith and jth rows of $\mathbf{G}$;
  Swap the ith and jth columns of $\mathbf{Z}_{ij}$;
until $g_{ii} \leq g_{jj}$;

Definition 2.1 (Reduced)**.** A basis matrix $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n]$ is *reduced*, if :

$$||\mathbf{a}_i||_2 \leq ||\mathbf{a}_j||_2 \quad \text{(for all } 1 \leq i < j \leq n); \tag{2.2a}$$

$$|\mathbf{a}_i^T \mathbf{a}_j| \leq \frac{1}{2}||\mathbf{a}_i||_2^2 \quad \text{(for all } 1 \leq i < j \leq n). \tag{2.2b}$$

We can see that in an n dimensional reduced basis defined in Definition 2.1, each pair of basis vectors is L-reduced. Algorithm 2, the Jacobi method, computes a reduced basis by reducing every pair of vectors using a `while` loop until all pairs are L-reduced.

Algorithm 2: Jacobi method

Input : A basis matrix $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n]$
Output: A unimodular matrix $\mathbf{Z}$, s. t. $\mathbf{AZ}$ is a reduced basis defined by 2.1

$\mathbf{Z} = I_n$, $\mathbf{G} = \mathbf{A}^T\mathbf{A}$;
while *not all pairs* $(\mathbf{a}_i, \mathbf{a}_j)$ *satisfy* (2.1) **do**
  for $i = 1$ **to** $n - 1$ **do**
    for $j = i + 1$ **to** n **do**
      $[\mathbf{G}, \mathbf{Z}_{ij}] =$ LAGRANGE2$(\mathbf{G}, i, j)$;
      $\mathbf{Z} \leftarrow \mathbf{Z}\mathbf{Z}_{ij}$;

3. AN ENHANCED JACOBI METHOD

From Definition 2.1, we expect that the columns of the basis matrix computed by Algorithm 2 are closer to being orthogonal than the original basis matrix. However, our experiments show that it is not effective on reducing the lengths of basis vectors or improving the condition number of a basis matrix.

For example, consider the following matrix [18],

$$\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3] = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ 0 & 0 & \frac{1}{2} \end{bmatrix}.$$

It can be verified that $\mathbf{A}$ satisfies the Definition 2.1 and its condition number $\mathcal{K}(\mathbf{A}) \approx 4.7387$. The condition number can be improved by reducing the size $||\mathbf{a}_3||_2$. Specifically, let $\mathbf{a} = \mathbf{a}_3 - \mathbf{a}_1 + \mathbf{a}_2 = [0, 0, \frac{1}{2}]^T$, we then have $||\mathbf{a}||_2 < ||\mathbf{a}_i||_2$ $(i = 1, 2, 3)$. Set $\mathbf{a}_3 = \mathbf{a}$, then the condition number of the new $\mathbf{A}$ is improved to 2.4495. In this section, we propose an enhancement of Algorithm 2 by integrating size reduction into the algorithm. Specifically, a technique called partial size reduction is introduced into Algorithm 2.

3.1. Partial Size Reduction

The size reduced condition [4] is enforced by many lattice reduction algorithms, such as the LLL algorithm [19] and the Schnoor's algorithm [10, 20]. Let $\mathbf{A}$ be an n dimensional basis matrix of full-column rank and $\mathbf{A} = \mathbf{QR}$ be its QR decomposition [17], then $\mathbf{A}$ is called *size-reduced*, if the upper triangular matrix $\mathbf{R}$ satisfies

$$|r_{i,j}| \leq \frac{1}{2}|r_{i,i}| \quad \text{(for all } 1 \leq i < j \leq n\text{)}. \tag{3.1}$$

Now we introduce a notion of partial size reduction. A basis matrix $\mathbf{A}$ is said to be *partially size reduced* with respect to an index pair (i, j) $(i < j)$, if

$$|r_{k,j}| \leq \frac{1}{2}|r_{k,k}| \quad \text{(for } 1 \leq k \leq i\text{)}. \tag{3.2}$$

Thus, if $\mathbf{A}$ is partially size reduced with respect to $(i, i+1)$ for all i: $1 \leq i < n$, then $\mathbf{A}$ is size reduced.

Given a basis matrix $\mathbf{A}$ and an indix pair (i, j) $(i < j)$, Procedure PSIZEREDUCE$(\mathbf{R}, i, j)$ updates $\mathbf{R}$ and computes a unimodular matrix $\mathbf{Z}_{ij}$, so that $\mathbf{AZ}_{ij}$ is partially size reduced with respect to (i, j).

Procedure PSizeReduce($\mathbf{R}, i, j$)

Input : $\mathbf{R}$ and indices i, j
Output: Updated $\mathbf{R}$ and a unimodular matrix $\mathbf{Z}_{ij}$, s. t. $\mathbf{AZ}_{ij}$ is partially size reduced w.r.t. (i, j)

1 $\mathbf{Z}_{ij} = I_n$;
2 **for** $k \leftarrow i$ **downto** 1 **do**
3 **if** $|r_{kj}| > \frac{1}{2}|r_{kk}|$ **then**
4 $q = \lfloor \frac{r_{kj}}{r_{kk}} \rceil$;
5 Set $\mathbf{Z} = I_n$ except $z_{kj} = -q$;
6 $\mathbf{R} \leftarrow \mathbf{RZ}$, $\mathbf{Z}_{ij} \leftarrow \mathbf{Z}_{ij}\mathbf{Z}$;

3.2. Updating R

To integrate size reduction into the Jacobi method, it is necessary to update $\mathbf{R}$ matrix in the QR decomposition of A. There are two operations in the EJacobi method can destroy the upper triangular structure of $\mathbf{R}$: permutation of two columns and the application of the unimodular matrix $\mathbf{Z}$ produced by Procedure LagrangeIT. Suppose $i < j$, both operations create nonzero entries r_{ki}, $k = i+1, ..., j$. Procedure RESTORER$(\mathbf{R}, i, j)$ restores the upper triangular structure of $\mathbf{R}$ by eliminating those nonzero entries using the plane reflection [21].

Procedure RestoreR($\mathbf{R}, i, j$)

Input : $\mathbf{R}$ and indices i, j $(i < j)$
Output: Updated $\mathbf{R}$

1 **for** $k = j$ **downto** $i + 1$ **do**
2 Find a plane reflection $\mathbf{P}$ of order 2 to triangulize $\begin{bmatrix} r_{k-1,i} & r_{k-1,j} \\ r_{k,i} & r_{k,j} \end{bmatrix}$;
3 Set $\mathbf{U} = I_n$ except $u_{i,i} = u_{j,j} = 0, u_{k-1,i} = p_{1,1}, u_{k-1,j} = p_{1,2}, u_{k,i} = p_{2,1}, u_{k,j} = p_{2,2}$;
4 $\mathbf{R} \leftarrow \mathbf{UR}$;
5 **for** $k = i + 1$ **to** $j - 1$ **do**
6 Find a plane reflection $\mathbf{P}$ of order 2 to triangulize $\begin{bmatrix} r_{k,k} & r_{k,k+1} \\ r_{k+1,k} & r_{k+1,k+1} \end{bmatrix}$;
7 Set $\mathbf{U} = I_n$ except $u_{i,i} = u_{j,j} = 0, u_{k,k} = p_{1,1}, u_{k,k+1} = p_{1,2}, u_{k+1,k} = p_{2,1}, u_{k+1,k+1} = p_{2,2}$;
8 $\mathbf{R} \leftarrow \mathbf{UR}$;

In RESTORER$(\mathbf{R}, i, j)$, the first part, lines 1 to 4, eliminates the entries $r_{k,i}$, $k = i+1, ..., j$, which creates nonzero entries $r_{k+1,k}$, $k = i+1, ..., j-1$ in the subdiagonal. The second part, lines 5 to 8, eliminates those nonzero entries on the subdiagonal.

3.3. An Enhanced Jacobi Method

Now we present our EJacobi method. Combining the size reduction condition (3.1) with the conditions in Definition 2.1, we have the following conditions for our EJacodi method:

$$||\mathbf{a}_i||_2 \leq ||\mathbf{a}_j||_2 \quad \text{(for all } 1 \leq i < j \leq n\text{)}, \tag{3.3a}$$

$$|\mathbf{a}_i^T \mathbf{a}_j| \leq \frac{1}{2}||\mathbf{a}_j||_2^2 \quad \text{(for all } 1 \leq i < j \leq n\text{)}, \tag{3.3b}$$

$$|r_{i,j}| \leq \frac{1}{2}|r_{i,i}| \quad \text{(for all } 1 \leq i < j \leq n\text{)}, \tag{3.3c}$$

where if (3.3b) and (3.3c) cannot be both satisfied, the condition (3.3c) is chosen. We call a basis matrix $\mathbf{A}$ E-reduced if it satisfies the above conditions.

Algorithm 3 computes an E-reduced basis. To avoid possible infinite loop caused by possible conflict between the conditions (3.3b) and (3.3c), we impose an upper bound m

for the `while` loop.

Algorithm 3: EJacobi method

Input : A basis matrix $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n]$ and an upper bound m

Output: A unimodular matrix $\mathbf{Z}$, s.t. $\mathbf{AZ}$ is E-reduced

$\mathbf{G} = \mathbf{A}^T\mathbf{A}$, $\mathbf{Z} = I_n$;
Get $\mathbf{R}$ from QR decomposition of $\mathbf{A}$;
while $\mathbf{AZ}$ *is not E-reduced and the number iterations is* $\leq m$ **do**
  for $i = 1$ **to** $n - 1$ **do**
    for $j = i + 1$ **to** n **do**
      if $|g_{ij}| > \frac{1}{2}g_{jj}$ **then**
        Set $\mathbf{Z}_{ij} = \text{LAGRANGEIT}(\mathbf{G}, i, j)$;
        $\mathbf{G} \leftarrow \mathbf{Z}_{ij}^T\mathbf{G}\mathbf{Z}_{ij}$;
        $\mathbf{R} \leftarrow \mathbf{R}\mathbf{Z}_{ij}$, $\mathbf{Z} \leftarrow \mathbf{Z}\mathbf{Z}_{ij}$;
        $\mathbf{R} = \text{RestoreR}(\mathbf{R}, i, j)$;
      if $\mathbf{A}$ *is not partially size reduced w.r.t.* (i, j) **then**
        $[\mathbf{R}, \mathbf{Z}_{ij}] = \text{PSIZEREDUCE}(\mathbf{R}, i, j)$;
        $\mathbf{G} \leftarrow \mathbf{Z}_{ij}^T\mathbf{G}\mathbf{Z}_{ij}$, $\mathbf{Z} \leftarrow \mathbf{Z}\mathbf{Z}_{ij}$;
      if $g_{ii} > g_{jj}$ **then**
        Swap the ith and jth columns of $\mathbf{G}$;
        Swap the ith and jth rows of $\mathbf{G}$;
        Swap the ith and jth columns of $\mathbf{R}$;
        Swap the ith and jth columns of $\mathbf{Z}$;
        $\mathbf{R} = \text{RestoreR}(\mathbf{R}, i, j)$;

4. EXPERIMENTAL RESULTS

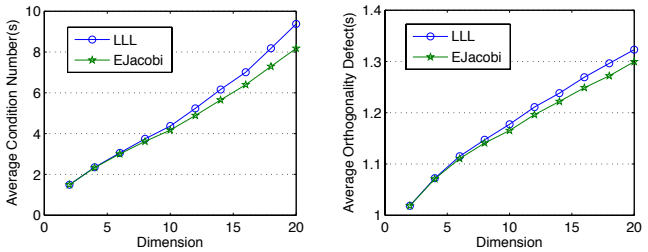
In this section, we compare our EJacobi method with the polynomial time LLL algorithm [11, 22], which is widely used in many lattice reduction aided decoding applications because of its extraordinary efficiency and high quality output in practice. It has been shown that the LLL-reduction-aided decoding can achieve the full diversity of a MIMO fading channels [23, 24]. We adopt the vector-operated version [25] to achieve higher efficiency, and set the parameter ω $(0 < \omega < 1)$ to 0.99 to get higher quality outputs.

The efficiency of the algorithms is measured by *CPU Time*. The quality of produced results is compared by two measurements, the *Condition Number* and the *Orthogonality Defect* $\delta(\mathbf{A})$, defined by

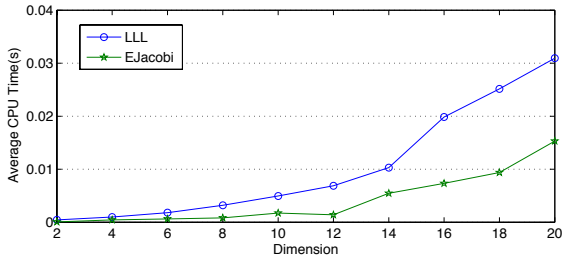
$$\delta^n(\mathbf{A}) = \frac{\Pi_j ||\mathbf{a}_j||_2}{\sqrt{\det(\mathbf{A}^T\mathbf{A})}},$$

also called *Hadamard Ratio* [6]. From the *Hadamard's Inequality*, $\delta(\mathbf{A}) \geq 1$, where the equality holds if and only if the columns $\mathbf{a}_j$ are orthogonal to each other. The closer $\delta(\mathbf{A})$ is to 1; the shorter the geometric mean of the lengths of the columns is; the more orthogonal the columns in $\mathbf{A}$ are, and hence the better the basis matrix $\mathbf{A}$ is.

The EJacobi method and the LLL algorithm are implemented in 64-bit version MATLAB 2012a running on a Dell computer with a 3.30GHz i3 Dual processor and 16GB memory. We compare the basis matrices of dimension up to 20.



(a) Condition Numbers (b) Orthogonality Defects



(c) Running Times

Fig. 1. Average condition numbers, orthogonality defects and CPU times (in seconds) between the LLL algorithm and the EJacobi method for random basis matrices of dimensions from 2 to 20.

For each dimension, we generated 1000 matrices with uniformly distributed random entries. The results shown in Fig. 1 are averages of 1000 matrices of same dimensions. Our experiments have shown that Algorithm 3 converges in five iterations of the `while` loop for dimensions under 20. Thus in our experiments, the maximum number m of iterations of the `while` loop is set to five. Fig. 1 (a) and (b) show that our EJacobi method produces basis matrices with smaller condition numbers and smaller orthogonality defects than the LLL algorithm. Fig. 1 (c) shows that our EJacobi method is about twices as fast as the LLL algorithm.

5. CONCLUSION

In this paper, we present a novel EJacobi method for lattice reduction aided decoding in MOMO systems. Our experimental results show that the algorithm is practically much faster, and produces better results, measured by both orthogonality defect and condition number, than the widely used LLL algorithm.

6. REFERENCES

[1] Babak Hassibi and Haris Vikalo, "On the sphere-decoding algorithm I. expected complexity," *IEEE Trans. Sig. Proc*, pp. 2806–2818, 2005.

[2] P. Xu, C. Shi, and J. Liu, "Integer estimation methods for GPS ambiguity resolution: an applications oriented review and improvement," *Survey Review*, vol. 44, pp. 59–71, Jan. 2012.

[3] Wa Ho Mow and Key Words, "Universal lattice decoding: Principle and recent advances," *Wireless Communications and Mobile Computing*, vol. 3, pp. 553–569, 2003.

[4] D. Wübben, D. Seethaler, J. Jalden, and G. Matz, "Lattice reduction: A survey with applications in wireless communications," *IEEE Signal Processing Magazine*, vol. 28, no. 3, pp. 70–91, May 2011.

[5] J. Jalden and B. Ottersten, "On the complexity of sphere decoding in digital communications," *Signal Processing, IEEE Transactions on*, vol. 53, no. 4, pp. 1474 – 1484, april 2005.

[6] J. Hoffstein, J.C. Pipher, and J.H. Silverman, *An introduction to mathematical cryptography*, Undergraduate texts in mathematics. Springer, 2008.

[7] H. Minkowski, "Discontinuity region for arithmetical equivalence," *J. reine Angew*, , no. 129, pp. 220–274, 1905.

[8] John L. Donaldson, "Minkowski reduction of integral matrices," *j-MATH-COMPUT*, vol. 33, no. 145, pp. 201–216, jan 1979.

[9] A. Korkine and G. Zolotareff, "Sur les formes quadratiques," *Mathematische Annalen*, vol. 6, pp. 366–389, 1873, 10.1007/BF01442795.

[10] C. P. Schnorr, "A hierarchy of polynomial time lattice basis reduction algorithms," *Theor. Comput. Sci.*, vol. 53, pp. 201–224, August 1987.

[11] A.K. Lenstra, Lenstra, and László Lovász, "Factoring polynomials with rational coefficients," *Math. Ann.*, vol. 261, pp. 515–534, 1982.

[12] Steven D. Galbraith, *Mathematics of public key cryptography.*, Cambridge: Cambridge University Press., 2012.

[13] Sanzheng Qiao, "A Jacobi method for lattice basis reduction," in *Proceedings of 2012 Spring World Congress on Engineering and Technology (SCET2012)*, Xi'an China, May 2012, Vol.2. IEEE, pp. 649–652.

[14] Phong Q. Nguyen and Damien Stehlé, "Low-dimensional lattice basis reduction revisited," *ACM Trans. Algorithms*, vol. 5, no. 4, pp. 46:1–46:48, Nov. 2009.

[15] Zhaofei Tian and Sanzheng Qiao, "A complexity analysis of a Jacobi method for lattice basis reduction," in *Proceedings of the Fifth International C* Conference on Computer Science and Software Engineering*, New York, NY, USA, 2012, C3S2E '12, pp. 53–60, ACM.

[16] C.J.G. Jacobi, "Über ein leichtes verfahren, die in der theorie der säkularstörungen vorkommenden gleichungen numerisch aufzulösen," *Journal für reine und angewandte Mathematik*, vol. 30, pp. 51–95, 1846.

[17] Gene H. Golub and Charles F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, 3rd edition, 1996.

[18] Igor A. Semaev, "A 3-dimensional lattice reduction algorithm," in *CaLC*, 2001, pp. 181–193.

[19] Phong Q. Nguyen and Brigitte Valle, *The LLL Algorithm: Survey and Applications*, Springer Publishing Company, Incorporated, 1st edition, 2009.

[20] Phong Nguyen, "Lattice reduction algorithms: Theory and practice," in *Advances in Cryptology - EUROCRYPT 2011*, Kenneth Paterson, Ed., vol. 6632 of *Lecture Notes in Computer Science*, pp. 2–6. Springer Berlin / Heidelberg, 2011.

[21] Franklin T. Luk and Sanzheng Qiao, "Conditioning properties of the LLL algorithm," in *Mathematics for Signal and Information Processing*. 2009, vol. 7444, pp. 7444–17, Proc. of SPIE.

[22] Sanzheng Qiao, "Integer least squares: sphere decoding and the LLL algorithm," in *Proceedings of the 2008 C3S2E conference*, New York, NY, USA, 2008, C3S2E '08, pp. 23–28, ACM.

[23] M. Taherzadeh, A. Mobasher, and A.K. Khandani, "LLL reduction achieves the receive diversity in MIMO decoding," *Information Theory, IEEE Transactions on*, vol. 53, no. 12, pp. 4801–4805, Dec. 2007.

[24] D. Wubben, R. Bohnke, V. Kuhn, and K.-D. Kammeyer, "Near-maximum-likelihood detection of MIMO systems using MMSE-based lattice reduction," in *Communications, 2004 IEEE International Conference on*, June 2004, vol. 2, pp. 798 – 802 Vol.2.

[25] Franklin T. Luk and Daniel M. Tracy, "An improved LLL algorithm," *Linear Algebra and its Applications*, vol. 428, no. 2-3, pp. 441 – 452, 2008.