A Jacobi Method for Lattice Basis Reduction

Sanzheng Qiao
Department of Computing and Software
McMaster University
Hamilton, Ontario, L8S 4K1, Canada
Email: qiao@mcmaster.ca

***Abstract*—Lattice reduction aided decoding has been successfully used in wireless communications. In this paper, we propose a Jacobi method for lattice basis reduction. Jacobi method is attractive, because it is inherently parallel. Thus high performance can be achieved by exploiting multiprocessor and/or multicore architectures. We also present our experimental results on the convergence of our method and the comparison with the LLL algorithm, a lattice basis reduction method widely used in wireless communication applications.**

I. Introduction

Lattice reduction has been successfully used in signal processing applications, such as global positioning system (GPS), frequency estimation, and particularly data detection and precoding in wireless communications. In this paper, we present a novel Jacobi method for lattice basis reduction. In this section, we briefly introduce the data detection in a multiinput multioutput (MIMO) system and lattices and bases. The details of lattice reduction with applications in wireless communications can be found in [21].

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

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

where $\mathbf{A}$ is an $m \times n$ matrix representing the channel matrix, and $\mathbf{n}$ is an $m \times 1$ vector representing the additive noise vector. In a full-rank flat-fading MIMO system, $\mathbf{A}$ is a complex matrix, however, it is straightforward to transform the complex problem into a real one, see [21] for example. So, 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 problem as the transmit signal:

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

where $\mathcal{A}$ denotes the finite set of real-valued modulation alphabet being used. Assume that the constellation $\mathcal{A}$ is of lattice type, such as PAM or QAM, then upon scaling and shifting the above problem can be transformed into an integer least squares problem. The complexity of solving integer least squares problems grows exponentially with the number of transmit antennas [1], [7], [8]. So ML decoding is not feasible for large number of transmit antennas or fast fading situation where the received signal changes rapidly. To reduce the detection cost, many approximate algorithms with low-complexity have been proposed, such as zero-forcing (ZF) decoding and successive interference cancellation (SIC) decoding [2], [18], [22]. The performance of an approximate detector is highly related to the structure of $\mathbf{A}$. It is well known that the closer to being orthogonal the column vectors of $\mathbf{A}$ are, the lower BER the approximate detector has [18], [20]. A lattice basis reduction algorithm can improve the orthogonality of the columns of $\mathbf{A}$, thus improve the performance of an approximate detector.

Suppose that $\mathbf{A}$ is an m-by-n, $m \geq n$, real matrix of full column rank, then a *lattice* generated by $\mathbf{A}$ is defined by the set:

$$L(\mathbf{A}) = \{\mathbf{A}\mathbf{z} \quad | \quad \mathbf{z} \in \mathbb{Z}^n\},$$

where $\mathbb{Z}^n$ denotes the set of integer n-vectors. The columns of $\mathbf{A}$ form a *basis* for the lattice $L(\mathbf{B})$, and the value of n is called the *dimension* of $L(\mathbf{A})$. When $n \geq 2$, the lattice $L(\mathbf{A})$ can have infinitely many different basis matrices other than $\mathbf{A}$. Two basis matrices $\mathbf{A}$ and $\mathbf{A}'$ generate a same lattice, if and only if $\mathbf{A}' = \mathbf{A}\mathbf{Z}$, where $\mathbf{Z}$, called a unimodular matrix, is an integer matrix with $|\det(\mathbf{Z})| = 1$. Thus the inverse of a unimodular matrix is an integer matrix. Given a lattice basis matrix $\mathbf{A}$, a lattice basis reduction algorithm finds a basis for the lattice $L(\mathbf{A})$ consisting of relatively short and more orthogonal vectors. In other words, a lattice reduction algorithm produces a unimodular matrix $\mathbf{Z}$ such that $\mathbf{A}\mathbf{Z}$ is reduced. In addition to wireless communications, lattice reduction plays an important role in many fields of mathematics and computer science [3], [5], [10], [19], particularly in communications [1], [4], [21] and cryptology [9], [17].

In this paper, we present a Jacobi method for lattice basis reduction. Jacobi method is attractive, because it is inherently parallel [6]. Parallel lattice basis reduction algorithms are useful. By exploiting multiprocessor/multicore architectures, they can make it possible to solve large size problems occurring in cryptography, also they can improve performance, which is essential in real-time applications such as wireless communications. We have compared our method with the LLL algorithm, which is widely used in wireless communications because it is the only method that produces reasonably good results in reasonable time. Our experimental results show that our Jacobi method computes lattice bases with better orthogonality in less time than the LLL algorithm.

The rest of the paper is organized as follows. In Section II, we describe the Lagrange's algorithm for computing reduced bases for lattices of dimension two. A row-cyclic version of

our Jacobi method is presented in Section III. Finally, we show our experimental results in Section IV.

II. Lagrange's Algorithm

Lagrange's algorithm [11] computes a reduced basis for a lattice of dimension two. A two-dimensional lattice $L(\mathbf{A})$ generated by $\mathbf{A} = [\mathbf{a}_1\ \mathbf{a}_2]$ is said to be *Lagrange-reduced* (or *L-reduced*) if

$$\|\mathbf{a}_1\|_2 \le \|\mathbf{a}_2\|_2 \quad \text{and} \quad |\mathbf{a}_1^{\mathrm{T}}\mathbf{a}_2| \le \|\mathbf{a}_1\|_2^2/2. \tag{1}$$

Intuitively, if θ denotes the angle between $\mathbf{a}_1$ and $\mathbf{a}_2$, then the condition (1) means that

$$\pi/3 \le \theta \le 2\pi/3,$$

since

$$|\cos\theta| = |\mathbf{a}_1^{\mathrm{T}}\mathbf{a}_2|/(\|\mathbf{a}_1\|_2\|\mathbf{a}_2\|_2) \le |\mathbf{a}_1^{\mathrm{T}}\mathbf{a}_2|/\|\mathbf{a}_1\|_2^2 \le 1/2.$$

For any two-dimensional lattice, an L-reduced basis always exists and is optimal in the sense that it consists of shortest possible basis vectors [19].

The Lagrange's algorithm can be viewed as a generalization of the following centered variant of Euclid's algorithm for computing the greatest common divisor (gcd) of a pair of integers a and b.

Algorithm 1 (Euclid): Given two integers a and b, this algorithm overwrites a with their gcd.

1. if $|a| < |b|$
2. swap a and b;
3. endif
4. while $b \neq 0$
5. $q = \lfloor a/b \rceil$;
6. $r = a - qb$;
7. $a = b$;
8. $b = r$;
9. endwhile.

Written in matrix form, the three lines 6–8 can be replaced by

$$\begin{bmatrix} a \\ b \end{bmatrix} \leftarrow \begin{bmatrix} 0 & 1 \\ 1 & -q \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}.$$

The Lagrange's algorithm is analogous to the above algorithm.

Algorithm 2 (Lagrange): Given a basis $\{\mathbf{a}_i\ \mathbf{a}_j\}$ for a two-dimensional lattice, this algorithm overwrites the basis with an L-reduced basis and computes a two-by-two unimodular matrix $\mathbf{Z}_{ij}$ so that the columns of $[\mathbf{a}_i\ \mathbf{a}_j]\mathbf{Z}_{ij}$ form an L-reduced basis.

$\mathbf{Z}_{ij} = \mathbf{I}_2$;
if $\|\mathbf{a}_i\|_2 < \|\mathbf{a}_j\|_2$
  swap $\mathbf{a}_i$ and $\mathbf{a}_j$;
  swap the columns of $\mathbf{Z}_{ij}$;
endif
repeat
  $q = \lfloor \mathbf{a}_i^{\mathrm{T}}\mathbf{a}_j/\|\mathbf{a}_j\|_2^2 \rceil$;
  $\mathbf{Z} = \begin{bmatrix} 0 & 1 \\ 1 & -q \end{bmatrix}$;
  $[\mathbf{a}_i\ \mathbf{a}_j] \leftarrow [\mathbf{a}_i\ \mathbf{a}_j]\mathbf{Z}$;
  $\mathbf{Z}_{ij} \leftarrow \mathbf{Z}_{ij}\mathbf{Z}$;
until $\|\mathbf{a}_i\|_2 \le \|\mathbf{a}_j\|_2$.

The matrix $\mathbf{Z}_{ij}$ can be viewed as the product of a permutation and a Gauss transformation [6] (elementary matrix):

$$\begin{bmatrix} 0 & 1 \\ 1 & -q \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & -q \\ 0 & 1 \end{bmatrix}.$$

Let a lattice generator matrix $\mathbf{A} = [\mathbf{a}_1\ \mathbf{a}_2\ \ldots\ \mathbf{a}_n]$, we consider $\mathbf{B} = [b_{ij}] = \mathbf{A}^{\mathrm{T}}\mathbf{A}$. Noting that $b_{ii} = \|\mathbf{a}_i\|_2^2$ and $b_{ij} = \mathbf{a}_i^{\mathrm{T}}\mathbf{a}_j$, we have the following squared version of Algorithm 2.

Algorithm 3 (Lagrange2($\mathbf{B}$*, i, j)):* Given $\mathbf{B} = \mathbf{A}^{\mathrm{T}}\mathbf{A}$, where $\mathbf{A}$ is a lattice generator matrix, this algorithm computes a unimodular matrix $\mathbf{Z}_{ij}$ such that the ith and jth columns of $\mathbf{A}\mathbf{Z}_{ij}$ are L-reduced and updates $\mathbf{B}$ accordingly.

$\mathbf{Z}_{ij} = \mathbf{I}_n$;
if $b_{ii} < b_{jj}$
  swap the ith and jth rows of $\mathbf{B}$;
  swap the ith and jth column of $\mathbf{B}$;
  swap the columns of $\mathbf{Z}_{ij}$;
endif
repeat
  $q = \lfloor b_{ij}/b_{jj} \rceil$;
  set $\mathbf{Z}$ to the same as $\mathbf{I}_n$ except
  $z_{ii} = 0$, $z_{jj} = -q$, and $z_{ij} = z_{ji} = 1$;
  $\mathbf{B} \leftarrow \mathbf{Z}^{\mathrm{T}}\mathbf{B}\mathbf{Z}$;
  $\mathbf{Z}_{ij} \leftarrow \mathbf{Z}_{ij}\mathbf{Z}$;
until $b_{ii} \le b_{jj}$.

III. Jacobi Method

Applying Algorithm 3 for two-dimensional sublattice to all possible pairs of columns of $\mathbf{A}$ in row-by-row fashion, we present the cyclic-by-row version of the Jacobi method for lattice basis reduction.

Algorithm 4 (Jacobi): Given a lattice generator matrix $\mathbf{A}$, this algorithm computes a unimodular matrix $\mathbf{Z}$ such that the columns of $\mathbf{A}\mathbf{Z}$ form a reduced basis.

$\mathbf{Z}_{ij} = \mathbf{I}_n$; $\mathbf{B} = \mathbf{A}^{\mathrm{T}}\mathbf{A}$;
repeat
  for $i = 1$ to $n-1$
    for $j = i+1$ to n
      $\mathbf{Z}_{ij}$ = Lagrange2($\mathbf{B}$, i, j);
      $\mathbf{Z} \leftarrow \mathbf{Z}\mathbf{Z}_{ij}$;
    endfor
  endfor
until all pairs $(\mathbf{a}_i, \mathbf{a}_j)$ satisfy (1);

Due to the outer repeat-loop in the above algorithm, the repeat-loop in Lagrange2 can be removed to improve the efficiency.

TABLE I
MAXIMAL AND AVERAGE NUMBER OF SWEEPS TAKEN BY ALGORITHM 4 OUT OF TEN RANDOM MATRICES OF EACH SIZE.

size	maximal	average
50	8	6.4
100	8	6.5
200	8	6.2

TABLE II
ORTHOGONAL DEFECT $\delta(\mathbf{A})$ OF THE ORIGINAL MATRIX AND THOSE PRODUCED BY OUR JACOBI METHOD AND THE LLL ALGORITHM.

size	$\delta(\mathbf{A})$	Jacobi	LLL
50	3.0230	1.9959	2.0578
100	3.1560	2.0902	2.3705
200	3.2370	2.1638	2.4046

IV. EXPERIMENTAL RESULTS

We programed our algorithms in MATLAB. The orthogonality of the columns of a $m \times n$ lattice basis matrix $\mathbf{A}$ was measured by the orthogonality defect $\delta(\mathbf{A})$, defined by

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

It is also called linear independence number [14] or Hadamard ratio. From the Hadamard's inequality, $\delta(\mathbf{A}) \geq 1$, and the equality holds if and only if the columns $\mathbf{a}_j$ are orthogonal each other.

How fast does Algorithm 4 terminate? We experimented on various sizes of random matrices and found that it terminates in less than ten sweeps, where one sweep is the double for-loop, that is, the program sweeps through all pairs $(\mathbf{a}_i, \mathbf{a}_j)$ once. Table I shows the maximal number of sweeps and average number of sweeps out of 10 random matrices of each size.

The LLL algorithm [12], [13], [15] is widely used in lattice reduction aided decoding, because it practically produces reasonably good results with low complexity. A matrix form of the LLL algorithm can be found in [14].

We generated random matrices and compared the orthogonality of the reduced bases computed by our Jacobi method with those produced by the LLL algorithm. The parameter ω ($0.25 < \omega < 1.0$) in the LLL algorithm was set to 0.99. The larger the ω, the better orthogonality the LLL algorithm produces. The orthogonality defects of the bases computed by our Jacobi method were consistently smaller than those computed by the LLL algorithm. Table II lists our experimental results. Each figure is an average of ten random matrices of the same size.

We also compared the running times of our Jacobi method and the LLL algorithm. To be consistent with the comparison in the orthogonality, the parameter ω in the LLL algorithm was also set to 0.99, which means that the LLL algorithm required more time to produce better results. Table III shows that our method is significantly faster than the LLL algorithm.

TABLE III
CPU TIMES, IN SECONDS, OF JACOBI METHOD AND THE LLL ALGORITHM. EACH FIGURE IS AN AVERAGE OF TEN RANDOM MATRICES OF THE SAME SIZE.

size	Jacobi	LLL
50	0.040	0.229
100	0.147	0.528
200	0.432	3.159

V. CONCLUSION

This paper presents a Jacobi method for lattice basis reduction. It is inherently parallel, so it can be developed into various parallel algorithm. Our experimental results show that the algorithm terminates in less than ten sweeps, runs much faster, and produces better results, measured by orthogonality defect, than the widely used LLL algorithm. Further speedup is expected from parallel implementations of our algorithm.

REFERENCES

[1] Erik Agrell, Thomas Eriksson, Alexander Vardy, and Kenneth Zeger. Closest point search in lattices. *IEEE Transactions on Information Theory*, vol. 48, no. 8, 2002, 2201–2214.

[2] L. Babai. On Lovász's lattice reduction and the nearest lattice point problem. *Combinatorica*, vol. 6, 1986, 1–13.

[3] J.W.S. Cassels. *An Introduction to the Geometry of Numbers, Second Printing*. Springer-Verlag, Berlin, Heidelberg, 1997.

[4] Y. H. Gan, C. Ling, and H. M. Mow. Complex lattice reduction algorithm for low-complexity full-diversity MIMO detection. *IEEE Transactions on Signal Processing*, vol. **57**, no. 7, 2009, 2701-2710.

[5] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Spriger, Berlin, 1993.

[6] G.H. Golub and C.F. Van Loan. *Matrix Computations, Third Edition*. The Johns Hopkins University Press, Baltimore, MD, 1996.

[7] B. Hassibi and H. Vikalo. On the sphere-decoding algorithm I: Expected complexity. *IEEE Trans. Signal Process.*, vol. 53, 2005, 2806–2818.

[8] J. Jaldén and B. Ottersen. On the complexity of sphere decoding in digital communications. *IEEE Trans. Signal Process.*, vol. 53, 2005, 1474–1484.

[9] A. Joux and J. Stern. Lattice reduction: A toolbox for the cryptanalyst. *Journal of Cryptology*, 11(**3**), 1998, 161–185.

[10] D. E. Knuth. *The Art of Computer Programming*. Vol. 2. 2nd ed. Reading, MA: Addison-Wesley, 1981.

[11] J. L. Lagrange. *Recherches d'arithmétique*. Nouveaux Mémoires de l'Académie de Berlin, 1773.

[12] A.K. Lenstra, H.W. Lenstra, Jr. and L. Lovász. Factorizing polynomials with rational coefficients. *Mathematicsche Annalen*, vol. 261, 1982, 515–534.

[13] F.T. Luk and S. Qiao. Numerical properties of the LLL algorithm. *Advance Signal Processing Algorithms, Architectures, and Implementations XVII, Proceedings of SPIE*, vol. 6697-3, 2007.

[14] F.T. Luk, S. Qiao, and W. Zhang. A lattice basis reduction algorithm. Technical Report 10-04, Institute for Computational Mathematics, Hong Kong Baptist University, April, 2010.

[15] F.T. Luk and D.M. Tracy. An improved LLL algorithm. *Linear Algebra and its Applications*, vol. 428, no. 2–3, 2008, 441–452.

[16] C. Ling. On the proximity factors of lattice reduction-aided decoding. *IEEE Transactions on Signal Processing*, vol. 59, no. 6, 2011, 2795–2808.

[17] D. Micciancio and S. Goldwasser. *Complexity of Lattice Problems: A Cryptographic Perspective*. Kluwer Internat. Ser. Engrg. Comput. Sci. 671. Kluwer Academic Publishers, Boston, MA, 2002.

[18] W. H. Mow. Universal lattice decoding: Principle and recent advances. *Wireless Commun. Mobile Comput.*, vol. 3, 2003, 553–569.

[19] The LLL Algorithm: Survey and Applications. *Information Security and Cryptography, Texts and Monographs*. Editors Phong Q. Nguyen and Brigitte Vallée. Springer Heidelberg Dordrecht London New York, 2010.

[20] D. Wübben, R. Böhnke, V. Kühn, and K.-D. Kammeyer. Near-maximum-likelihood detection of MIMO systems using MMSE-based lattice reduction. *Proc. 2004 Int. Conf. Communications (ICC04)*, June 2004, 798–802.

[21] D. Wübben, D. Seethaler, J. Jaldén, and G. Marz. Lattice reduction: a survey with applications in wireless communications. *IEEE Signal Processing Magazine*, vol. 28, no. 3, 2011, 70–91.

[22] P. W. Wolniansky, G. J. Foschini, G. D. Golden, and R. A. Valenzuela. V-BLAST: An architecture for realizing very high data rates over the rich-scattering wireless channel. In *IEEE Proceedings of ISSSE-98*, Pisa, Italy, 29. September, 1998.