INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING, *SERIES B*
Volume 5, Number 1-2, Pages 1–12

A PARALLEL JACOBI-TYPE LATTICE BASIS REDUCTION ALGORITHM

FILIP JEREMIC AND SANZHENG QIAO

Abstract. This paper describes a parallel Jacobi method for lattice basis reduction and a GPU implementation using CUDA. Our experiments have shown that the parallel implementation is more than fifty times as fast as the serial counterpart, which is twice as fast as the well-known LLL lattice reduction algorithm.

Key words. Lattice basis reduction, Jacobi method, GPU.

1. Introduction

Lattice basis reduction has been successfully used for many problems in integer programming, cryptography, number theory, and information theory [1]. In this paper we discuss a parallel version of the lattice basis reduction algorithm called the Jacobi method. The Jacobi method is very attractive as it is inherently parallel. We take advantage of this by utilizing the graphics processing unit (GPU) to capitalize on the algorithm's parallel nature. After introducing notations in Section 2, we will first describe a serial version of the Jacobi method for lattice reduction in Section 3, and later explore its parallel nature in Section 4. Moreover, in Section 5 we will discuss the tools and tricks used in our GPU implementation to achieve high runtime performance. Finally, in Section 6 we will present experimental results of our parallel implementation of the Jacobi method.

2. Preliminaries

In this section we cover some basic notations which we will use throughout the paper. Given a subspace W of $\mathbb{R}^n$ and a basis $\mathcal{B} = \{b_1, b_2, \ldots, b_m\}$ of n-dimensional vectors which span W, we define a *lattice* $\mathcal{L}$ of W generated by the basis $\mathcal{B}$ as the set of vectors:

$$\mathcal{L}(\mathcal{B}) = \left\{ \sum_{i=1}^{m} a_i b_i \;\middle|\; a_i \in \mathbb{Z} \right\}$$

Typically, we view a lattice basis $\mathcal{B}$ in matrix form, where the vectors in the basis form the columns of the matrix. In this context we say that the respective matrix $\mathcal{B}$ is a *generator* of the lattice $\mathcal{L}$. The value m in the above definition of a lattice is called the lattice *dimension*, or *rank*. A given lattice basis may generate proper subspace of the space it resides in. In such a case the generator matrix is rectangular with $m < n$. If on the other hand $m = n$, we say that the lattice is of *full rank*, and consequently the generator matrix will be an invertible square matrix.

Received by the editors January 1, 2014 and, in revised form, March 22, 2014.
2000 *Mathematics Subject Classification.* 65F30, 68W10.

When the lattice dimension $m \geq 2$, the lattice can have infinitely many distinct basis matrices. This is not surprising as the underlying vector space can also have infinitely many bases. For example,

$$\mathcal{B} = \begin{bmatrix} 2.0 & 2.7 \\ 0 & 0.7 \end{bmatrix} \quad \text{and} \quad \mathcal{B}' = \begin{bmatrix} -0.7 & 1.3 \\ -0.7 & -0.7 \end{bmatrix}$$

form two bases for the same lattice. The question arises as to how can we transform one basis matrix into another, and more importantly what makes one basis "better" than another? To answer the former question we introduce the notion of a lattice *determinant*, which is defined as the square root of the determinant of $B^{\mathrm{T}}B$, where B is the respective generator matrix, that is,

$$\det(\mathcal{L}(\mathcal{B})) = \sqrt{\det(\mathcal{B}^{\mathrm{T}}\mathcal{B})}.$$

The lattice determinant is an important numerical invariant as it is independent of the chosen lattice basis. Therefore, two generator matrices $\mathcal{B}$ and $\mathcal{B}'$ generate the same lattice $\mathcal{L}$ if and only if $\mathcal{B}' = \mathcal{B}Z$, where Z, called a unimodular matrix, is an integer matrix with $|\det Z| = 1$. Because the determinant of a unimodular matrix is of unit length, the inverse of a unimodular matrix is also an integer matrix. In the above example, the two generator matrices $\mathcal{B}$ and $\mathcal{B}'$ are related by

$$\mathcal{B}' = \begin{bmatrix} -0.7 & 1.3 \\ -0.7 & -0.7 \end{bmatrix} = \begin{bmatrix} 2.0 & 2.7 \\ 0 & 0.7 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ -1 & -1 \end{bmatrix} = \mathcal{B}Z$$

The answer to the latter question we posed is relative to the application problem at hand, however for many such problems a desirable property of a lattice basis is that it consists of relatively short and more orthogonal vectors. In this context, we say that such a basis is *reduced*. Thus given a lattice basis matrix $\mathcal{B}$, a lattice basis reduction algorithm produces a unimodular matrix Z, such that the basis $\mathcal{B}Z$ is reduced. In the above example, $\mathcal{B}'$ is reduced from $\mathcal{B}$. It consists of shorter and more orthogonal basis vectors than those of $\mathcal{B}$.

There are various notions of a reduced basis. In 1850, Hermite introduced the first notion of reduction for lattices of arbitrary dimensions, proposed an algorithm for computing such reduced bases, and proved its termination [2]. Hermite's algorithm is of theoretical significance, but its complexity is still unknown. Schnorr and Euchner [3] reconsidered this problem and developed a practical algorithm for constructing the Hermite reduced basis. In 1873, Korkine and Zolotareff [4] strengthened the definition of Hermite reduced basis. Their proposed notion of reduction is usually called the *HKZ reduced basis* [5], named after Hermite, Korkine and Zolotareff. In 1983, using induction, Kannan [6] presented the first algorithm for constructing the HKZ reduced bases. Helfrich [7], Kannan [8], and Banihashemi and Khandani [9] further refined Kannan's algorithm and improved the complexity analysis. Note that the methods based on Kannan's strategy are intended as theoretical tools, and the related papers usually focus on asymptotic complexity. Agrell et. al. [10] presented a practical algorithm and used it as a preprocessor for the integer least squares problems. In 1891, Minkowski [11] defined a new notion of reduction, which is stronger than the HKZ reduction. This definition is now known as the *Minkowski reduced basis*. Lenstra, Lenstra, and Lovász [12] developed the first polynomial-time lattice reduction algorithm, known as the LLL algorithm, named after the three authors. Their notion of reduced basis is actually a relaxation of the Hermite reduced basis [2]. The LLL algorithm has become the most important tool in public-key cryptanalysis [13] and integer least squares problems [10, 14]. Further