Segment LLL-Reduction of Lattice Bases.

Henrik Koy[1] and Claus Peter Schnorr[2]

[1] Deutsche Bank AG, Frankfurt am Main, henrik.koy@db.com
[2] Fachbereiche Mathematik und Informatik, Universität Frankfurt, PSF 111932, D-60054 Frankfurt am Main, Germany. schnorr@cs.uni-frankfurt.de

Abstract. We present an efficient variant of LLL-reduction of lattice bases in the sense of Lenstra, Lenstra, Lovász. We organize LLL-reduction in segments of size k. Local LLL-reduction of segments is done using local coordinates of dimension k.

We introduce *segment* LLL-*reduced bases*, a variant of LLL-reduced bases achieving a slightly weaker notion of reducedness, but speeding up the reduction time of lattices of dimension n by a factor n. We also introduce a variant of LLL-reduction using *iterated segments*. The resulting reduction algorithm runs in $O(n^3 \log_2 n)$ arithmetic steps for integer lattices of dimension n with basis vectors of length 2^n.

Keywords. LLL-reduction, shortest lattice vector, segments, iterated segments, local coordinates, local LLL-reduction, divide and conquer.

1 Introduction.

The famous algorithm for LLL-reduction of lattice bases of Lenstra, Lenstra, Lovász [LLL82] is a basic technique for solving important problems in algorithmic number theory, integer optimization, diophantine approximation and cryptography. Of the many possible applications we refer to a few recent ones [BN00, Bo00, Co98,NS00]. Present codes for LLL-reduction merely perform for lattices up to dimension 350, our new contributions lift this barrier beyond dimension 1000. In this paper we present a theoretic reduction algorithm together with a rigorous analysis counting the number of arithmetic steps on integers of bounded length. In the companion paper [KS01] we introduce an orthogonalization via scaling using floating point arithmetic that is stable in very high dimension. The resulting segment LLL-reduction with floating point orthogonalization is a highly practical reduction algorithm that is in practice much faster than previous codes for LLL-reduction. In practice it finds a lattice basis that is as good as a truly LLL-reduced basis.

In this paper we propose the concept of segment LLL-reduction in which a basis $b_1, \dots, b_n$ of dimension $n = km$ is partitioned into m segments $b_{kl+1}, \dots, b_{kl+k}$ of k consecutive basis vectors. Segment LLL-reduction is designed to do most of the LLL-exchanges within the segments using local coordinates of dimension

k. There is a double advantage. Firstly, local LLL-vector operations cost merely $O(k)$ arithmetic steps, whereas global vector operations require $O(n)$ steps. Secondly, as local operations are for lattices of small dimension they can be done in single precision whereas global operations require multi-precision steps.

First we introduce k-segment reduced bases, a variant of LLL-reduced bases that is designed to minimize the global overhead associated to the local LLL-reductions. Segment LLL-reduction saves a factor n in the running time compared to standard LLL-reduction of lattices of dimension n. Using *iterated segments* we present an even faster theoretic reduction algorithm that runs in $O(n^3 \log_2 n)$ arithmetic steps for integer lattices of dimension n with basis vectors of length $O(2^n)$. In this paper we analyse a theoretic version of the novel reduction algorithms in the model of integer arithmetic. In the companion paper [KS01] we we propose a pracrical implementation using floating point orthogonalization. Our present code reduces lattice bases of dimension 1000 consisting of integers of bit length 400 in 10 hours on a 800 MHz PC. Even for dimension $n < 100$ the new code is in practice much faster than previous codes. We did not yet implement reduction using iterated segments. The use of iterated segments should further speed up the reduction in high dimensions.

Previously, Schönhage [Sc84] has used local coordinates to speed-up LLL-reduction. His concept of semi-reduction approximates the length of the shortest lattice vector up to a factor 2^n whereas we get close to a factor $(4/3)^{n/2}$. We use the [Sc84] analysis of the size of integers occuring during the reduction.

Future work. We expect that the practical reduction algorithm of [KS01] will greatly improve the present codes for finding very short lattice vectors, in particular for lattices of high dimension. LLL-reduction is the basis for the more powerful reduction algorithms of Kannan [K84] and of block reduction of Schnorr [S87,S94]. Combining the segment LLL-reduction with the techniques of [S87, S94, SH95] will speed up the search for very short basis vectors. The novel concept of segment LLL-reduction and the block reduction of [S87, S94] are based on similar structures and can be easily combined. Moreover, the search for very short lattice vectors can be enhanced by the new concept of primal-dual segment reduction proposed by Koy [K01].

2 LLL-reduction of lattice bases.

Let $\mathbf{R}^d$ be the d–dimensional real vector space with the Euclidean inner product $\langle \,.\,, \,.\,\rangle$ and the Euclidean norm, called the length, $\|y\| = \langle y, y\rangle^{1/2}$. An integer *lattice* $L \subset \mathbf{Z}^d$ is an additive subgroup of $\mathbf{Z}^d$. Its *dimension* is the dimension of the minimal linear subspace that contains L. Every lattice L of dimension n has a *basis*, i.e. a set $b_1, \ldots, b_n$ of linearly independent vectors satisfying: $L = \{t_1 b_1 + \ldots + t_n b_n \,|\, t_1, \ldots, t_n \in \mathbf{Z}\}$. Let $L(b_1, \ldots, b_n)$ denote the lattice with basis $b_1, \ldots, b_n$.

With an ordered lattice basis $b_1, \ldots, b_n \in \mathbf{Z}^d$ we associate the *Gram–Schmidt orthogonalization* $\widehat{b}_1, \ldots, \widehat{b}_n \in \mathbf{R}^d$ which can be computed together with the Gram–Schmidt coefficients $\mu_{j,i} = \langle b_j, \widehat{b}_i\rangle \,/\, \langle \widehat{b}_i, \widehat{b}_i\rangle$ by the recursion

$$\widehat{b}_1 = b_1, \ \widehat{b}_j = b_j - \sum_{i=1}^{j-1} \mu_{j,i} \widehat{b}_i \qquad \text{for } j = 2, \ldots, n.$$

We have $\mu_{j,j} = 1$ and $\mu_{j,i} = 0$ for $i > j$. From the above equations we have the Gram–Schmidt decomposition $[b_1, \ldots, b_n] = [\widehat{b}_1, \ldots, \widehat{b}_n] \, [\mu_{j,i}]^{\top}_{1 \le i,j \le n}$, where $[b_1, \ldots, b_n]$ denotes the matrix with column vectors $b_1, \ldots, b_n$ and $[\mu_{j,i}]^{\top}$ is the transpose of the matrix $[\mu_{j,i}]$. The *determinant* of lattice $L(b_1, ..., b_n)$ is defined $\det L = \det([b_1, ..., b_n][b_1, ..., b_n]^{\top})^{1/2}$.

Definition 1. *An ordered basis* $b_1, \ldots, b_n \in \mathbf{Z}^d$ *of the lattice* L *is* LLL-reduced *with* $\delta \in]\frac{1}{4}, 1]$ *if it has properties* **1.** **2.**:

1. $|\mu_{j,i}| \le 1/2$ *for* $1 \le i < j \le n$,

2. $\delta \, \|\widehat{b}_i\|^2 \le \mu_{i+1,i}^2 \, \|\widehat{b}_i\|^2 + \|\widehat{b}_{i+1}\|^2$ *for* $i = 1, \ldots, n-1$.

LLL-reduced bases have been introduced by A.K. Lenstra, H.W. Lenstra, Jr. and L. Lovász [LLL82] who focused on $\delta = 3/4$. A basis with property **1.** is called *size-reduced*. A basis $b_1, ..., b_n$ is good if the values $\|b_i\|$ are good approximations to the successive minima. The i–th *successive minimum* λ_i of a lattice L, relative to the Euclidean norm, is the smallest real number r such that there are i linearly independent vectors in L of length at most r. Extending [LLL82] to arbitrary $\delta \in]\frac{1}{4}, 1]$ and $\alpha := 1/(\delta - \frac{1}{4})$ yields

Theorem 1. *A basis* $b_1, \ldots, b_n$ *of lattice* L *that is* LLL*–reduced with* δ *satisfies*:

1. $\|b_i\|^2 \le \alpha^{n-1} \lambda_i^2$ *and* $\|b_1\|^2 \le \alpha^{i-1} \|\widehat{b}_i\|^2$ *for* $i = 1, \ldots, n$,

2. $\|b_1\|^2 \le \alpha^{\frac{n-1}{2}} (\det L)^{\frac{2}{n}}$ *and* $\|\widehat{b}_n\|^2 \ge \alpha^{-\frac{n-1}{2}} (\det L)^{\frac{2}{n}}$.

Consider the QR-factorization $B = QR$ of the basis matrix $B = [b_1, \ldots, b_n] \in \mathbf{Z}^{d \times n}$, where $Q \in \mathbf{R}^{d \times d}$ is an orthogonal matrix and $R = [r_{i,j}] = [\mathbf{r}_1, ..., \mathbf{r}_n] \in \mathbf{R}^{d \times n}$ is an upper triangular matrix, $r_{i,j} = 0$ for $i > j$. We have $\mu_{j,i} = r_{i,j}/r_{i,i}$ and $|r_{i,i}| = \|\hat{b}_i\|$. We present the core of the LLL-reduction algorithm using the coefficients $r_{i,j}$ of the matrix R. The vector $\mathbf{r}_l$ is the orthogonal transform of b_l.

LLL

INPUT $b_1, \ldots, b_n \in \mathbf{Z}^d, \ \delta$

OUTPUT $b_1, \ldots, b_n$ LLL-reduced basis

1. $l := 1$ (l *is the stage*)

2. **while** $l \le n$ **do**

compute $\mathbf{r}_l = (r_{1,l}, ..., r_{l,l}, 0, ..., 0)^{\top}$,
size-reduce b_l against $b_{l-1}, ..., b_1$.
if $l \ne 1$ **and** $\delta r_{l-1,l-1}^2 > r_{l-1,l}^2 + r_{l,l}^2$
then swap b_{l-1}, b_l, $l := l - 1$ **else** $l := l + 1$. *end*

The LLL-algorithm locally reduces the 2×2-diagonal submatrices of R by successively decreasing the length of the first column vector in a 2×2-matrix.

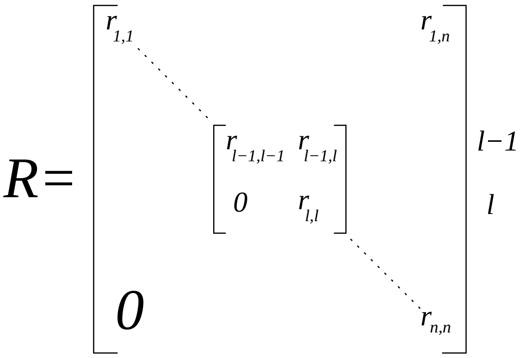

Fig. 1. The 2×2-diagonal submatrices of R

While the coefficients $r_{i,j}$ of the matrix R are not rational, rational arithmetic must use the rational numbers $\mu_{j,i}$, $\|\widehat{b}_i\|^2$ satisfying $r_{i,j} = \mu_{j,i}\|\widehat{b}_i\|$, $r_{i,i} = \|\widehat{b}_i\|$.
LLL-time bound. Following [Sc84] we measure the size of the input in terms of

$$M_{Sc} =_{\text{def}} \max_{i=1,\dots,n}(2^n, \|b_i\|^2, D_i),$$

where $D_i = \|\widehat{b}_1\|^2 \cdots \|\widehat{b}_i\|^2$ is the *Gramian determinant* of the sublattice with basis $b_1, ..., b_i$. The M_{Sc}-measure differentiates large and small lattice determinants. It relates to the [LLL82]-measure $M = \max_i \|b_i\|^2$ by $M \le M_{Sc} \le M^n$.

By the Lovász volume argument, the number of LLL-exchanges — swaps of b_{l-1}, b_l — is at most $(n-1)\log_{1/\delta} M_{Sc}$. Size-reduction of b_l requires $O(dl)$ arithmetic steps. **LLL** performs at most $O(n^2 d \log_{1/\delta} M_{Sc})$ arithmetic steps. These steps operate on rational integers where numerator and denominator have at most $O(\log_2 M_{Sc})$ bits, see [LLL82,Sc84].

3 Segment LLL-Reduction.

Segments and local coordinates. Let the basis $b_1, \dots, b_n \in \mathbf{Z}^d$ have dimension $n = k \cdot m$ and the QR-factorization $[b_1, \dots, b_n] = QR$. We partition the basis matrix B into m *segments* $B_l = [b_{k(l-1)+1}, \dots, b_{kl}]$ for $l = 1, ..., m$. Local reduction of two consecutive segments uses the $2k \times 2k$-submatrix $R_l := [r_{kl+i,kl+j}]_{-k<i,j\le k} \in \mathbf{R}^{2k\times 2k}$ of $R \in \mathbf{R}^{d\times n}$, corresponding to two consecutive segments B_{l-1}, B_l. More precisely, local reduction uses the rational representation of R_l by the coefficients $\mu_{kl+i,kl+j}$ and $\|\widehat{b}_{kl+i}\|^2$. We want to do most of the LLL-exchanges and the corresponding size-reduction in local coordinates of some R_l. Extra global transformations are required after local LLL-reduction. The novel concept of *k-segment reduced bases* intends to minimize these global costs. We let $D(l) = \|\widehat{b}_{k(l-1)+1}\|^2 \cdots \|\widehat{b}_{kl}\|^2$ denote the *local Gramian determinant* of segment B_l. We have that $D_{kl} = D(1) \cdots D(l)$.

Definition 2. *We call an ordered basis* $b_1, \dots, b_n \in \mathbf{Z}^d, n = km,$ k-segment LLL-reduced *with* $\delta \in]\frac{1}{4}, 1]$ *if it is size-reduced and satisfies for* $\alpha = 1/(\delta - \frac{1}{4})$:

1. $\delta\,\|\widehat{b}_i\|^2 \leq \mu_{i+1,i}^2\|\widehat{b}_i\|^2 + \|\widehat{b}_{i+1}\|^2$ *for* $i \neq 0 \mod k$,

2. $D(l) \leq (\alpha/\delta)^{k^2} D(l+1)$ *for* $l = 1, \ldots, m-1$,

3. $\delta^{k^2}\,\|\widehat{b}_{kl}\|^2 \leq \alpha\|\widehat{b}_{kl+1}\|^2$ *for* $l = 1, \ldots, m-1$.

We use Inequality **3.** to bound in Theorem 2 $\|b_i\|$ by the successive minimum λ_i. Without Inequality **3.** we bound in Theorem 5 $\|b_1\|$ by $(\det L)^{\frac{1}{n}}$ as well as by $\|\widehat{b}_n\|$. The large exponent k^2 of δ^{k^2} in **3.** is impractical for $k > \sqrt{n}$, we drop Inequality **3.** of the definition 2 in Section 4.

Lemma 1. *A k-segment* LLL*-reduced basis $b_1, \ldots, b_n$ satisfies*

1. $\delta^{2k^2+j-i}\,\|\widehat{b}_i\|^2 \leq \alpha^{j-i}\|\widehat{b}_j\|^2$ *for* $1 \leq i < j \leq n$,

2. $\delta^{n-1}\|b_1\|^2 \leq \alpha^{n-1}\|\widehat{b}_n\|^2$ *and* $\delta^{k^2+j-1}\,\|\widehat{b}_1\|^2 \leq \alpha^{j-1}\|\widehat{b}_j\|^2$ *for* $1 < j \leq n$.

Proof. The inequalities **2.** of Definition 2 imply for $l \leq l'$ that

$$D(l) \leq (\alpha/\delta)^{k^2(l'-l)} D(l').$$

As $D(l) = \|\widehat{b}_{k(l-1)+1}\|^2 \cdot \ldots \cdot \|\widehat{b}_{kl}\|^2$, there exists s, $1 \leq s \leq k$ such that

$$\|\widehat{b}_{k(l-1)+s}\|^2 \leq (\alpha/\delta)^{k(l'-l)}\|\widehat{b}_{k(l'-1)+s}\|^2.$$

Inequality **1.** follows by combining the latter inequality — at each end $k(l-1)+s$ and $k(l'-1)+s$ — with some inequalities $\|\widehat{b}_\nu\|^2 \leq \alpha\|\widehat{b}_{\nu+1}\|^2$, which hold within the segments, and possibly with an inequality **3.** of Definition 2 that bridges two consecutive segments. In particular, we can choose l, l' so that $i \leq k(l-1)+s \leq k(l'-1)+s \leq j$ and that each pair $\{i, k(l-1)+s\}$ and $\{j, k(l'-1)+s\}$ are indices either of the same or of two consecutive segments.

Inequalities **2.** If $i = 1, j = n$ we can choose $l = 1, l' = m$, and thus each pair $\{1, k(l-1)+s\}, \{n, k(l'-1)+s\}$ is in a single segment. Consequently, we have that $\delta^{n-1}\|b_1\|^2 \leq \alpha^{n-1}\|\widehat{b}_n\|^2$. If $i = 1$ the pair $\{i, k(l-1)+s\}$ is in one segment and thus $\delta^{k^2+j-1}\|b_1\|^2 \leq \alpha^{j-1}\|\widehat{b}_j\|^2$. □

Theorem 2. *Let $b_1, \ldots, b_n$ be a basis that is k-segment* LLL*-reduced with δ. Then we have for $i = 1, \ldots, n$:*

$$\delta^{2k^2+n-1}\,\|b_i\|^2 \leq \alpha^{n-1}\lambda_i^2 \quad \textit{and} \quad \delta^{k^2+i-1}\,\|b_1\|^2 \leq \alpha^{i-1}\|\widehat{b}_i\|^2,$$

where $\lambda_1 \leq \cdots \leq \lambda_n$ are the successive minima of the lattice.

The proof of Theorem 2 follows from Lemma 1 by standard arguments. Comparison of Theorems 1 and 2 shows that k-segment reduced bases are close to LLL-reduced bases.

Algorithm for segment LLL*-reduction.* The algorithm **segment LLL** transforms a given basis into a k-segment reduced basis. It iterates local LLL-reduction of two segments $[B_{l-1}, B_l] = [b_{kl-k+1}, \ldots, b_{kl+k}]$ via

The procedure **loc-LLL**(l). Given the orthogonalization of a k-segment reduced basis $b_1, \ldots, b_{kl-k}$ the procedure **loc-LLL**(l) computes the orthogonalization and size-reduction of the segments B_{l-1}, B_l. In particular it provides the submatrix $R_l \in \mathbf{R}^{2k\times 2k}$ of $R \in \mathbf{R}^{d\times n}$ corresponding to the segments B_{l-1}, B_l. Thereafter it performs a local LLL–reduction of R_l and stores the LLL-transformation in the matrix $H \in \mathbf{Z}^{2k\times 2k}$. Finally, it transforms $[B_{l-1}, B_l]$ into the locally reduced segments $[B_{l-1}, B_l]H$ and size-reduces $[B_{l-1}, B_l]$ globally.

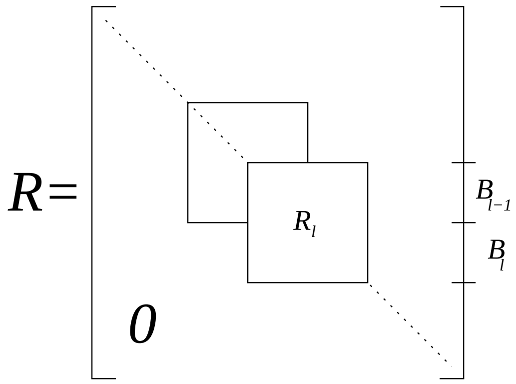

Fig. 2. Areas of subsequent local LLL-reductions.

Each execution of **loc-LLL**(l) induces a global overhead of $O(ndk)$ arithmetic steps for global size-reduction, orthogonalization and segment transformation via H. The efficiency relies on the fast local LLL-reduction of R_l. Here each LLL-exchange of two consecutive basis vectors costs merely $O(k^2)$ arithmetic steps, local size-reduction included. Compare this to the $O(nd)$ arithmetic steps for an LLL-exchange in global coordinates. Here is our segment LLL-reduction algorithm.

Segment LLL

INPUT $b_1, \ldots, b_n \in \mathbf{Z}^d, k, m, n = km,\ \delta$

OUTPUT $b_1, \ldots, b_n$ k-segment LLL-reduced basis

1. $l := 1$

2. **while** $l \leq m-1$ **do**

loc-LLL(l)

if $l \neq 1$ **and**

$(\ D(l-1) > (\alpha/\delta)^{k^2} D(l)$ **or** $\delta^{k^2} \|\widehat{b}_{k(l-1)}\|^2 > \alpha \|\widehat{b}_{k(l-1)+1}\|^2\ )$

then $l := l-1$ **else** $l := l+1$. *end*

The original LLL-algorithm — with δ replaced by δ^2 — essentially coincides with the case $k = 1$ of **segment LLL**. [1]

[1] The inequality $D(l-1) > (\alpha/\delta)^{k^2} D(l)$ holds for $k = 1$ if and only if $\delta\, r_{l-1,l-1}^2 > \alpha\, r_{l,l}^2$. Multiplying the latter inequality by $\alpha = 1/(\delta - \frac{1}{4})$ implies that $\delta^2 r_{l-1,l-1}^2 >$

Segment LLL proceeds like the original LLL-algorithm replacing vectors by segments of size k. Obviously, **segment LLL** results in a k-segment LLL-reduced basis.

Time analysis. The dominant work of **segment LLL** consists of the global overhead of the executions of **loc-LLL**(l). Initial and final global size-reduction of the segments B_{l-1}, B_l cost $O(ndk)$ arithmetic steps per execution. These costs also cover the initial computation of the column vectors $\mathbf{r}_{kl-k+1}, ..., \mathbf{r}_{kl+k} \in \mathbf{R}^d$ of the matrix R and the global transform of $[B_{l-1}, B_l]$ into $[B_{l-1}, B_l]H$. Note that the overhead $O(ndk)$ of **loc-LLL**(l) is linear in the segment size k. Next we show by the Lovász volume argument that the number of executions of **loc-LLL** decreases cubically in k.

We let *decr* denote the number of times that the condition

$$D(l-1) > (\alpha/\delta)^{k^2} D(l) \textbf{ or } \delta^{k^2} \|\widehat{b}_{kl}\|^2 > \alpha \|\widehat{b}_{kl+1}\|^2$$

holds and l is decreased for some l. The number of iterations of the while loop and the number of executions of **loc-LLL** is $m - 1 + 2 \cdot decr$.

Theorem 3. $decr \;\le\; 2\frac{m-1}{k^2} \log_{1/\delta} M_{Sc} \;<\; 2\frac{n}{k^3} \log_{1/\delta} M_{Sc}$.

Remarks. 1. For $k = 1, m = n$ the bound of Theorem 3 shows that there are at most $(n-1)\log_{1/\delta} M_{Sc}$ LLL-exchanges of two consecutive basis vectors during reduction. The factor 2 in Theorem 3 disappears as **segment LLL** with $k = 1$ corresponds to **LLL** using δ^2.
2. If the reduction reverses the order of the basis $b_1, ..., b_n$ we have $decr \ge \binom{m}{2} = \frac{m(m-1)}{2}$. Then, by Theorem 3 we must have that $\log_{1/\delta} M_{Sc} \ge nk/4$, $M_{Sc} \ge \delta^{-nk/4}$. Thus, the bound M_{Sc} must be rather large for interesting bases.

Proof. The Gramian determinant D_{kl} is the product $D_{kl} = D(1) \cdots D(l)$ of the first l local determinants. **loc-LLL**(l) performs a local LLL-reduction of two segments B_{l-1}, B_l, it merely changes $D_{k(l-1)}$, the Gramian determinant of $b_1, ..., b_{kl-k}$, and leaves $D_{kl'}$ for $l' \ne l-1$ unchanged.

Consider an execution of **loc-LLL**(l) performed after a decrease of l. We show that it decreases $D(l-1)$ by the factor $\delta^{k^2/2}$. First consider the case that $D(l-1) > (\alpha/\delta)^{k^2} D(l)$ holds upon entry of **loc-LLL**. As $D^{ter}(l-1) \le \alpha^{k^2} D^{ter}(l)$ holds upon termination, and since the product $D(l-1)D(l)$ does not change we have that

$$\begin{aligned} D^{ter}(l-1) &\le \alpha^{k^2} D^{ter}(l) = \alpha^{k^2} D(l-1)D(l)/D^{ter}(l-1) \\ &\le \delta^{k^2} D(l-1)^2/D^{ter}(l-1), \end{aligned}$$

and thus $D^{ter}(l-1) \le \delta^{k^2/2} D(l-1)$.

If **loc-LLL**(l) is performed in the case $\delta^{k^2} \|\widehat{b}_{kl}\|^2 > \alpha \|\widehat{b}_{kl+1}\|^2$ the previous argument shows again that $D^{ter}(l-1) \le \delta^{k^2/2} D(l-1)$. Hence, **loc-LLL**($l$) decreases $\mathbf{D} =_{\text{def}} \prod_{j=1}^{m-1} D_{jk}$ by the factor $\delta^{k^2/2}$. As $\mathbf{D}$ is a positive integer, $\mathbf{D} \le M_{Sc}^{m-1}$ this implies

---

$\frac{1}{4} r_{l-1,l-1}^2 + r_{l,l}^2$ which violates an Inequality **2.** of Definition 1 provided that δ is replaced by δ^2.

$$decr \leq \log_{1/\delta^{k^2/2}} M_{Sc}^{m-1} \leq 2\tfrac{m-1}{k^2} \log_{1/\delta} M_{Sc}. \qquad \square$$

Theorem 4. *For $k = \Theta(m) = \Theta(\sqrt{n})$* **segment LLL** *performs $O(nd \log_{1/\delta} M_{Sc})$ arithmetic steps using integers of bit length $O(\log_2 M_{Sc})$.*

Proof. Time bound. There are at most $(n \log_{1/\delta} M_{Sc})$ LLL-exchanges — each requiring $O(k^2)$ steps for local size-reduction. There are $decr \leq 2\frac{m}{k^2} \log_{1/\delta} M_{Sc}$ calls of **loc-LLL** — each requiring $O(ndk)$ arithmetic steps for global size-reduction, global orthogonalization and global transformation of two consecutive segments. The choice $k, m = \Theta(\sqrt{n})$ equalizes for $d = O(n)$ the theoretic time bounds $O(k^2 n \log_{1/\delta} M_{Sc})$ for the LLL-exchanges in local coordinates and $O(n^2 k \frac{n}{k^3} \log_{1/\delta} M_{Sc})$ for the global overhead.

We need that $M_{Sc} \geq 2^n$, otherwise the $O(nd \log_{1/\delta} M_{Sc})$ bound does not cover the $O(n^2 d)$ steps required for QR-factorization, the $m-1$ calls **loc-LLL**(l) for $l = 1, ..., m-1$ also require $O(n^2 d)$ steps for global size-reduction.

Size of the integers. We first show that the initial bound

$$M_{Sc} = \max_{i=1,...,n}(2^n, \|b_i\|^2, D_i)$$

can temporarily increase not more than by a factor $2\,\alpha^{n-1}$ for $\delta \geq \frac{3}{4}$. Recall that the determinants D_i do not increase during LLL-reduction. In particular, we always have that $1 \leq D_i \leq M_{Sc}$, and $\|\widehat{b}_i\|^2 = D_i/D_{i+1}$ is a rational integer, $M_{Sc}^{-1} \leq \|\widehat{b}_i\|^2 \leq M_{Sc}$ with numerator and denominator bounded by M_{Sc}.

The length $\|b_i\|^2$ can only temporarily increase during size-reduction of b_i according to $b_i := b_i - \lceil \mu_{i,h} \rfloor b_h$ for $h = i-1, ..., 1$. Assuming that $b_1, ..., b_{i-1}$ is already LLL-reduced we have that $|\mu_{i,h}| = |\frac{\langle b_i, \widehat{b}_h \rangle}{\langle \widehat{b}_h, \widehat{b}_h \rangle}| \leq \|b_i\| / \|\widehat{b}_h\| \leq \sqrt{M_{Sc} \alpha^{i-1}}$. We see that, during size-reduction of b_i, the value $\max(\|b_1\|^2, ..., \|b_i\|^2)$ can temporarily increase not more than by a factor $2\,\alpha^{i-1}$ for $\delta \geq \frac{3}{4}$.

Consider the coefficients of the matrix $H \in \mathbf{Z}^{2k \times 2k}$ representing the local LLL-reduction of the segments B_{l-1}, B_l so that local LLL-reduction of B_{l-1}, B_l transforms $[B_{l-1}, B_l] := [B_{l-1}, B_l] H$. We let $b'_j, \widehat{b}'_j, \mu'_{j,i}$ denote the values corresponding to the transformed segments $[b'_{kl-k+1}, ..., b'_{kl+k}] = [B_{l-1}, B_l] H$. We let $\|H\|_1$ denote the maximal $\|\ \|_1$-norm of the columns of H.

Lemma 2. [Sc84, Inequality (3.3)] *We have that*

1. $H = ([\mu_{j,i}]^\top)^{-1} \; [\langle \widehat{b}_i, \widehat{b}'_j \rangle \|\widehat{b}_i\|^{-2}]_{kl-k<i,j\leq kl+k} \; [\mu'_{j,i}]^\top$,

2. $\|H\|_1 \leq (2k)^2 (\frac{3}{2})^{2k-1} M_{Sc} \leq M_{Sc}^2$.

Proof. Equality **1.** follows from the equations

$$[b_{kl-k+1}, ..., b_{kl+k}] = [\widehat{b}_{kl-k+1}, ..., \widehat{b}_{kl+k}][\mu_{j,i}]^\top,$$
$$[b'_{kl-k+1}, ..., b'_{kl+k}] = [\widehat{b}'_{kl-k+1}, ..., \widehat{b}'_{kl+k}][\mu'_{j,i}]^\top$$
$$= [\widehat{b}_{kl-k+1}, ..., \widehat{b}_{kl+k}][\mu_{j,i}]^\top H.$$

When starting the local LLL-reduction of R_l the segments are already size-reduced, i.e., $|\mu_{j,i}| \leq \frac{1}{2}$ for $kl - k < i < j \leq kl + k$. Then the coefficients $\nu_{j,i}$ of the inverse matrix $[\nu_{j,i}] = [\mu_{j,i}]^{-1}$ satisfy $|\nu_{j,i}| \leq (\frac{3}{2})^{|j-i|}$. Inequality **2.** follows from **1.** as $|\langle \widehat{b}_i, \widehat{b}'_j \rangle \|\widehat{b}_i\|^{-2}| \leq \|\widehat{b}'_j\| / \|\widehat{b}_i\| \leq M_{Sc}$ and $|\mu_{j,i}| \leq \frac{1}{2}$ for $i < j$. $\square$

Conclusion. All integers arising during the reduction are bounded in absolute value by $\max(M_{Sc}^2, 2\,\alpha^{n-1}M_{Sc}) \le M_{Sc}^2$ for $\delta \ge \frac{3}{4}$. The algorithm **segment LLL** improves the LLL-time bound — for the case $n = km$, $k = \Theta(\sqrt{n})$ — from $O(n^2 d \log_{1/\delta} M_{Sc})$ to $O(nd \log_{1/\delta} M_{Sc})$ arithmetic steps, saving a time factor n. □

Optimal segment size. For $k < m$ the dominant costs are for the global overhead of the executions of **loc-LLL** requiring $O(k^{-2}n^2 d \log_{1/\delta} M_{Sc})$ arithmetic steps. The LLL-exchanges require $O(k^2 n \log_{1/\delta} M_{Sc})$ arithmetic steps using local coordinates. The latter should become dominant for $k > m$. In practice, the crossover point of the two costs is for $k \gg m$. This is because the steps for the LLL-exchanges are mostly on small integers. In the [KS01] implemenation, these steps are even in floating point arithmetic. Large segment sizes as $k = n/4$ yield good results.

4 Divide and Conquer Using Iterated Segments.

There is a natural way to iterate the concept of segments to subsegments, subsubsegments, etc. Let $n = k_1 \cdots k_s$ be a product of integers $k_1, ..., k_s \ge 2$, $s \le \log_2 n$. We consider segments of size n/k_s, subsegments of size $n/(k_s k_{s-1})$, subsubsegments of size $n/(k_s k_{s-1} k_{s-2})$ and so on. We denote $\mathbf{k}(\sigma) := (k_1, ..., k_\sigma)$, $\mathbf{k}_\sigma := k_1 \cdots k_\sigma$ for $\sigma = 1, ..., s$. There are $s-1$ levels of segments, we have $\mathbf{k}_\sigma$-segments $[b_{\mathbf{k}_\sigma(l-1)+1}, ..., b_{\mathbf{k}_\sigma l}]$ of size $\mathbf{k}_\sigma$ for $\sigma = 1, ..., s-1$. For $n = k_1 \cdots k_s = \mathbf{k}_s$ we let $D(l, \mathbf{k}(\sigma)) = \|\widehat{b}_{\mathbf{k}_\sigma(l-1)+1}\|^2 \cdots \|\widehat{b}_{\mathbf{k}_\sigma l}\|^2$ denote the *local determinant* of the l-th $\mathbf{k}_\sigma$-segment. Also, let $\mathbf{k}_0 := 1$, $D(l, \mathbf{k}(0)) := \|\widehat{b}_l\|^2$. For $n = 2^s$ it is natural to choose $k_1 = k_2 = ... = k_s = 2$.

Recall that Inequality **3.** of Definition 2 is impractical for segments of size $\mathbf{k}_\sigma$ greater than $\sqrt{n}$. We want to use large $\mathbf{k}_\sigma$-segments, where the exponent $\mathbf{k}_\sigma^2$ of $\delta^{\mathbf{k}_\sigma^2}$ in **3.** of Definition 2 gets impractical. Theorem 5 shows that the Inequalities **1.** and **2.** of Definition 2 — without Inequality **3.** — describe a sufficiently strong reduction. In the following we drop Inequality **3.** to render possible a divide and conquer approach.

Theorem 5. *Let the basis $b_1, \ldots, b_n, n = k \cdot m$ of lattice L satisfy the inequalities* **1.** *and* **2.** *of Definition 2. Then we have that*
1. $\|\hat{b}_{kl+1}\|^2 \le (\alpha/\delta)^{k(l'-l)-1}\|\hat{b}_{kl'}\|^2$ *for* $1 \le l < l' \le n$,
2. $\|b_1\|^2 \le (\alpha/\delta)^{\frac{n-1}{2}}(\det L)^{\frac{2}{n}}$
3. $\|\widehat{b}_n\|^2 \ge (\delta/\alpha)^{\frac{n-1}{2}}(\det L)^{\frac{2}{n}}$.

For comparison LLL-reduced bases $b_1, \ldots, b_n$ satisfy $\|b_1\|^2 \le \alpha^{\frac{n-1}{2}}(\det L)^{\frac{2}{n}}$. Inequality **2.** is a bit weaker than the inequality $\|b_1\|^2 \le \alpha^{n-1}\lambda_1^2$ of Theorem 1, but it is sufficient for most applications, in particular if δ is close to 1. The dual Inequality **3.** is useful in applying the method of [Co97] to find small integer solutions of polynomial equations.

Proof. Inequality **1.** follows from the inequalities $\|\hat{b}_{kl+1}\|^2 \le (\alpha/\delta)^{i-1}\|\hat{b}_{kl+i}\|^2$ for $1 \le i \le k$ that hold within the segments, and an inequality $\|\hat{b}_{k(l-1)+s}\|^2 \le (\alpha/\delta)^{k(l'-l)}\|b_{k(l'-1)+s}\|^2$ for some $1 \le s \le k$ that bridges the segments B_l and $B_{l'}$. The latter inequality holds as $D(l) = \|\widehat{b}_{k(l-1)+1}\|^2 \cdots \|\widehat{b}_{kl}\|^2$ satisfies $D(l) \le (\alpha/\delta)^{k^2(l'-l)} D(l')$.

To prove Inequality **2.** we note that $D(1) = \|\hat{b}_1\|^2 \cdots \|\hat{b}_k\|^2$ and $\|b_1\|^2 \le \alpha^{i-1}\|\hat{b}_i\|^2$ for $i = 1, ..., k$ imply that

$$\|b_1\|^2 \le \alpha^{\frac{k-1}{2}} D(1)^{\frac{1}{k}}.$$

Moreover, $D(1)\cdots D(m) = (\det L)^2$ and $D(1) \le (\alpha/\delta)^{k^2(i-1)} D(i)$ imply that

$$D(1) \le (\alpha/\delta)^{k^2\frac{m-1}{2}} (\det L)^{\frac{2}{m}}.$$

Combining the two inequalities yields Inequality **2.**

We get Inequality **3.** by applying Inequality **2.** to the *dual* basis $b_1^*, ..., b_n^*$ satisfying $\langle b_i^*, b_j\rangle = \delta_{i,j}$, $\|b_1^*\| = \|\widehat{b}_n\|^{-1}$ and $\det(L^*) = (\det L)^{-1}$. □

Definition 3. *A basis* $b_1, \ldots, b_n \in \mathbf{Z}^d$, $n = k_1 \cdots k_s = \mathbf{k}_s$ *is called* $\mathbf{k}(s)$-segment LLL-reduced *with* $\delta \in]\frac{1}{4}, 1]$ *if it is size-reduced and satisfies for* $\alpha := 1/(\delta - \frac{1}{4})$:

$$D(l, \mathbf{k}(\sigma)) \le (\alpha/\delta^\sigma)^{(\mathbf{k}_\sigma)^2} D(l+1, \mathbf{k}(\sigma)) \tag{1}$$

for $l \ne 0 \mod k_{\sigma+1}$, *for* $\sigma = 0, ..., s-1$, *and* $l = 1, ..., n/\mathbf{k}_\sigma - 1$.

The exponent σ of δ^σ is used for the time bound of Theorem 7. As δ can be chosen very close to 1 the factor $\delta^{-\sigma}$ is still close to 1.

Due to the restriction $l \ne 0 \mod k_{\sigma+1}$ the inequalities (1) only hold within $\mathbf{k}_{\sigma+1}$-segments, they cannot bridge distinct $\mathbf{k}_{\sigma+1}$-segments. The inequalities (1) get weaker and weaker as the size $\mathbf{k}_\sigma$ of the segments increases and $\delta^{\mathbf{k}_\sigma^2}$ decreases. For $\sigma = 0, k_0 = \mathbf{k}_0 = 1$, the inequalities (1) mean that $\|\widehat{b}_l\|^2 \le \alpha\|\widehat{b}_{l+1}\|^2$ for $l = 1, ..., n-1$, $l \ne 0 \mod k_1$ — slightly weakening Clause **1.** of Definition 2.

For $n = k_1 \cdot k_2 = k \cdot m$, $s = 2$, Definition 3 recites clauses **1.** and **2.** of Definition 2 slightly weakening **1.** and dropping **3.**.

We next extend Lemma 3 to iterated segments.

Theorem 6. *A* $\mathbf{k}(s)$*-segment LLL-reduced basis* $b_1, \ldots, b_n$, $n = k_1 \cdots k_s = \mathbf{k}_s$ *satisfies* $\|b_1\|^2 \le (\alpha/\delta^{s-1})^{\frac{n-1}{2}} (\det L)^{\frac{2}{n}}$ *and* $\|\widehat{b}_n\|^2 \ge (\delta^{s-1}/\alpha)^{\frac{n-1}{2}} (\det L)^{\frac{2}{n}}$.

Proof. We prove by induction on σ that

$$\|b_1\|^2 \le (\alpha/\delta^{\sigma-1})^{\frac{\mathbf{k}_\sigma - 1}{2}}\ D(1, \mathbf{k}(\sigma))^{\frac{1}{\mathbf{k}_\sigma}}.$$

The first claim of the theorem follows from $D(1, \mathbf{k}(s)) = (\det L)^2$, $\mathbf{k}_s = n$ for $\sigma = s$. The second claim follows by duality.

The induction hypothesis holds for $\sigma = 1$ as we have $\|b_1\|^2 \le \alpha^{i-1}\|\hat{b}_i\|^2$ for $i = 1, \ldots, k_1$ and $D(1, \mathbf{k}(1)) = \|\hat{b}_1\|^2 \cdots \|\hat{b}_{k_1}\|^2$.

The induction hypothesis extends from σ to $\sigma + 1$ by the inequalities

$$D(1, \mathbf{k}(\sigma)) \le (\alpha/\delta^\sigma)^{(\mathbf{k}_\sigma)^2(l-1)}\ D(l, \mathbf{k}(\sigma)) \quad \text{for } l = 1, \ldots, k_{\sigma+1}$$

and the equation $D(1, \mathbf{k}(\sigma+1)) = \prod_{l=1}^{k_{\sigma+1}} D(l, \mathbf{k}(\sigma))$. □

$\mathbf{k}_s$**-segment LLL-reduction.** The algorithm $\mathbf{k}_s$**-segment LLL** transforms a given basis into a $\mathbf{k}_s$-segment LLL-reduced basis. It iterates $\mathbf{k}_{s-1}$-segment LLL-reduction of two $\mathbf{k}_{s-1}$-segments $[B_{l-1}, B_l] = [b_{\mathbf{k}_{s-1}(l-1)+1}, \dots, b_{\mathbf{k}_{s-1}(l+1)}]$ via the procedure $\mathbf{k}_{s-1}$**-segment LLL**. For $\sigma \geq 1$ the procedure $\mathbf{k}_\sigma$**-segment LLL**(l)

- computes the orthogonal transform of the $\mathbf{k}_\sigma$-segments $[B_{l-1}, B_l]$ providing the local R-matrix $R_l \in \mathbf{R}^{2\mathbf{k}_\sigma \times 2\mathbf{k}_\sigma}$,
- performs a local $\mathbf{k}_{\sigma-1}$-segment LLL-reduction on $[B_{l-1}, B_l]$ by iterating $\mathbf{k}_{\sigma-1}$**-segment LLL**,
- stores the corresponding transformation matrix $H_l \in \mathbf{Z}^{2\mathbf{k}_\sigma \times 2\mathbf{k}_\sigma}$,
- upon termination, it transports H_l to the matrix $H_{l'}$ of the $\mathbf{k}_{\sigma+1}$-segment $B_{l'}$ that contains B_{l-1}, B_l. The $2\mathbf{k}_\sigma$ columns of $H_{l'}$ corresponding to B_{l-1}, B_l are multiplied from the right by H_l. Thereafter, H_l is reset to the identity matrix.

While the procedure $\mathbf{k}_\sigma$**-segment LLL** for $\sigma \geq 1$ recursively calls $\mathbf{k}_{\sigma-1}$**-segment LLL** , the procedure $\mathbf{k}_0$**-segment LLL**(l) exchanges b_{l-1} and b_l in case that $\|\widehat{b}_{l-1}\|^2 > \alpha \|\widehat{b}_l\|^2$.

$\mathbf{k}_s$**-segment LLL**

INPUT $b_1, \dots, b_n \in \mathbf{Z}^d,\ n = k_1 \cdots k_s = \mathbf{k}_s,\ \delta$

OUTPUT $b_1, \dots, b_n$ $\mathbf{k}_s$-segment LLL-reduced basis

1. $l := 1$

2. **while** $l \leq k_s - 1$ **do**

$\mathbf{k}_{s-1}$**-segment LLL**(l)
if $l \neq 1$ **and** $D(l-1, \mathbf{k}(s-1)) > (\alpha/\delta^\sigma)^{(\mathbf{k}_{s-1})^2} D(l, \mathbf{k}(s-1))$
then $l := l - 1$ **else** $l := l + 1$. *end*

Theorem 7. *Given a basis* $b_1, \dots, b_n \in \mathbf{Z}^d$, $n = k_1 \cdots k_s = \mathbf{k}_s$ *the algorithm* $\mathbf{k}_s$**-segment LLL** *produces a* $\mathbf{k}_s$*-segment LLL-reduced basis. It performs at most* $O(dn^2 + d\sum_{\sigma=1}^s k_\sigma^2 \log_{1/\delta} M_{Sc})$ *arithmetic steps. If* $\max_\sigma k_\sigma = O(1)$ *and* $\log_{1/\delta} M_{Sc} = O(n^2)$*, the number of arithmetic steps is* $O(n^2 d \log_2 n)$*.*

Proof. Consider for $\sigma = 1, \dots, s-1$ the number of executions of $\mathbf{k}_\sigma$**-segment LLL** as sub$\cdots$subroutine of $\mathbf{k}_s$**-segment LLL**. The number of these executions is $n/\mathbf{k}_\sigma - 1 + 2 \cdot \mathrm{decr}(\mathbf{k}(\sigma))$, where $\mathrm{decr}(\mathbf{k}(\sigma))$ is the number of times that $\mathbf{k}_\sigma$**-segment LLL** is called as sub$\cdots$subroutine of $\mathbf{k}_s$**-segment LLL** due to a violated inequality (1), where $D(l-1, \mathbf{k}(\sigma)) > (\alpha/\delta)^{(\mathbf{k}_\sigma)^2} D(l, \mathbf{k}(\sigma))$ for some l.

Formally, $\mathbf{k}_\sigma$**-segment LLL** is also executed after a decrease of l on some level σ' where $\sigma' > \sigma$. However, that execution induces no costs except for the case of a violated inequality (1) at level σ. The reason is that the $\mathbf{k}_s$-segments are already orthogonalized and size-reduced by previous calls of $\mathbf{k}_\sigma$**-segment LLL**.

Consider the product of the Gramian determinants

$$\mathbf{D}(\mathbf{k}(\sigma)) =_{\mathrm{def}} \prod_{l=1}^{n/\mathbf{k}_\sigma} D_{\mathbf{k}_\sigma l} = \prod_{l=1}^{n/\mathbf{k}_\sigma} \left(D(1, \mathbf{k}(\sigma) \cdots D(l, \mathbf{k}(\sigma)) \right).$$

We apply Theorem 3 to $\mathbf{k}_\sigma$-segments. Each execution of **$\mathbf{k}_\sigma$-segment LLL** — due to a violated inequality (1) — decreases $\mathbf{D}(\mathbf{k}(\sigma))$ by the factor $\delta^{(\mathbf{k}_\sigma)^2/2}$. $\mathbf{D}(\mathbf{k}(\sigma))$ is product of $n/\mathbf{k}_\sigma$ Gramian determinants. As initially $\mathbf{D}(\mathbf{k}(\sigma)) \leq M_{Sc}^{n/\mathbf{k}_\sigma}$, and upon termination $\mathbf{D}(\mathbf{k}(\sigma)) \geq 1$ we see that

$$\mathrm{decr}(\mathbf{k}(\sigma)) \leq \tfrac{2n}{(\mathbf{k}_\sigma)^3} \log_{1/\delta} M_{Sc}.$$

In total there are $n/\mathbf{k}_\sigma - 1 + \frac{2n}{(\mathbf{k}_\sigma)^3} \log_{1/\delta} M_{Sc}$ executions of **$\mathbf{k}_\sigma$-segment LLL** each inducing an overhead of $O(\mathbf{k}_\sigma \mathbf{k}_{\sigma+1}^2)$ arithmetic steps. This overhead includes: the orthogonal transform of the $\mathbf{k}_\sigma$-segments B_{l-1}, B_l providing the local R-matrix $R_l \in \mathbf{R}^{2\mathbf{k}_\sigma \times 2\mathbf{k}_\sigma}$ of $[B_{l-1}, B_l]$, the transport of H_l to the transformation matrix $H_{l'}$ of the next higher level and size-reduction of R_l against $R_{l'}$. The total overhead of all **$\mathbf{k}_\sigma$-segment LLL** executions is

$$O(n\mathbf{k}_{\sigma+1}^2 + nk_{\sigma+1}^2 \log_{1/\delta} M_{Sc}).$$

In the particular case $\sigma = s-1$, this overhead is $O(d\mathbf{k}_s^2 + dk_s^2 \log_{1/\delta} M_{Sc})$ as the global transforms are done directly on the basis vectors in $\mathbf{Z}^d$.

In the particular case $\sigma = 0$, the overhead covers a total of $O(n \log_{1/\delta} M_{Sc})$ LLL-exchanges of two consecutive vectors which each uses $O(k_1^2)$ arithmetic steps for a local exchange and a local size-reduction. We see that **$\mathbf{k}_s$-segment LLL** performs $O(n^2 d + d \sum_{\sigma=1}^{s} k_\sigma^2 \log_{1/\delta} M_{Sc})$ arithmetic steps providing the time bound of the theorem.

Conclusion. The time bound of the novel algorithm **$\mathbf{k}_s$-segment LLL** is comparable to that of classical matrix multiplication. The size of the integers occuring in the new reduction algorithm can be bounded by the method of Theorem 4. The algorithm uses integers of bit length $O(s \log_2 M_{Sc})$, where s is the number of levels σ. Each level can add $O(\log_2 M_{Sc})$ bits when transporting the local transformation H_l to the next higher level.

References

[BN00] *D. Bleichenbacher and P.Q. Nguyen*, Noisy Polynomial Interpolation and Noisy Chinese Remaindering, Proc. Eurocrypt'00, LNCS 1807, Springer-Verlag, pp. 53-69, 2000.

[Bo00] *D. Boneh*, Finding Smooth Integers in Small Intervals Using CRT Decoding, Proc. STOC'00, ACM Press, pp. 265-272, 2000.

[Ca00] *J. Cai*, The Complexity of some Lattice Problems, Proc. ANTS'00, LNCS 1838, Springer-Verlag, pp. 1-32, 2000.

[Co97] *D. Coppersmith*, Small Solutions to Polynomial Equations, and Low Exponent RSA Vulnerabilities, *J. Crypt.* **10**, pp. 233-260, 1997.

[K84] *R. Kannan*, Minkowski's Convex Body Theorewm and Integer Programming. Mathematical Operation Research, **12**, pp. 415–440, 1984.

[K01] *H. Koy*, Notes of a Lecture. Frankfurt 2001.

[KS01] *H. Koy and C.P. Schnorr*, LLL-Reduction with Floating Point Orthogonalization. This proceedings CaLC 2001.

[LLL82] *A. K. Lenstra, H. W. Lenstra and L. Lovász*, Factoring polynomials with rational coefficients, *Math. Ann.* **261**, pp. 515-534, 1982.

[NS00] *P.Q. Nguyen and J. Stern*, Lattice Reduction in Cryptology, An Update, Proc. ANTS'00, LNCS 1838, Springer-Verlag, pp. 85-112, 2000.

[S87] *C.P. Schnorr*, A hierarchy of polynomial time lattice basis reduction algorithms, *Theoretical Computer Science* **53**, pp. 201-224, 1987.

[S91] *C.P. Schnorr and M. Euchner*, Lattice Basis Reduction and Solving Subset Sum Problems. Proceedings FCT'91, LNCS 591, Springer-Verlag, pp. 68–85, 1991. The complete paper appeared in Mathematical Programming Studies, 66A, **2**, pp. 181–199, 1994.

[S94] *C.P. Schnorr*, Block Reduced Lattice Bases and Successive Minima, *Combinatorics, Probability and Computing* ,**3**, pp. 507-522, 1994.

[SH95] *C.P. Schnorr and H. Hörner*, Attacking the Chor-Rivest Cryptosystem by Improved Lattice Reduction. Proceedings Eurocrypt'95, LNCS 921, Springer-Verlag, pp. 1–12, 1995.

[Sc84] *A. Schönhage*, Factorization of univariate integer polynomials by diophantine approximation and improved lattice basis reduction algorithm, *Proc. 11-th Coll. Automata, Languages and Programming, Antwerpen 1984*, LNCS **172**, Springer-Verlag, pp. 436-447, 1984.