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 11932, 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 \(O(n^3 \log n)\) 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 \(b_1, \ldots, b_n\) of dimension \(n = km\) is partitioned into \(m\) segments \(b_{k+1}, \ldots, b_{k+1}\) 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
2. **LLL-reduction of lattice bases.**

Let $\mathbb{R}^d$ be the $d$-dimensional real vector space with the Euclidean inner product $\langle \cdot, \cdot \rangle$ and the Euclidean norm, called the length, $\|y\| = \langle y, y \rangle^{1/2}$. An integer lattice $L \subset \mathbb{Z}^d$ is an additive subgroup of $\mathbb{Z}^d$. 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 $L = \left\{ t_1 b_1 + \ldots + t_n b_n \mid t_1, \ldots, t_n \in \mathbb{Z} \right\}$. Let $L(b_1, \ldots, b_n)$ denote the lattice with basis $b_1, \ldots, b_n$.

With an ordered lattice basis $b_1, \ldots, b_n \in \mathbb{Z}^d$ we associate the Gram-Schmidt orthogonalization $\hat{b}_1, \ldots, \hat{b}_n \in \mathbb{R}^d$ which can be computed together with the Gram-Schmidt coefficients $\mu_{j,i} = \langle b_j, \hat{b}_i \rangle / \langle \hat{b}_i, \hat{b}_i \rangle$ by the recursion
\[ \tilde{b}_1 = b_1, \quad \tilde{b}_j = b_j - \sum_{i=1}^{j-1} \mu_{j,i} \tilde{b}_i \quad \text{for } j = 2, \ldots, n. \]

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

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

1. \( |\mu_{j,i}| \leq 1/2 \quad \text{for } 1 \leq i < j \leq n, \)
2. \( \delta \|b_i\|^2 \leq \mu_{i+1,i}^2 \|\tilde{b}_i\|^2 + \|\tilde{b}_{i+1}\|^2 \quad \text{for } i = 1, \ldots, n - 1. \)

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 \( b_1, \ldots, b_n \) is good if the values \( \|b_i\| \) are good approximations to the successive minima. The \( i \)-th successive minimum \( \lambda_i \) 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 \( \delta \in [1/4, 1] \) and \( \alpha := 1/(\delta - 1/4) \) yields

**Theorem 1.** A basis \( b_1, \ldots, b_n \) of lattice \( L \) that is LLL-reduced with \( \delta \) satisfies:

1. \( \|b_i\|^2 \leq \alpha^{n-1} \lambda_i^2 \quad \text{and} \quad \|b_i\|^2 \leq \alpha^{n-1} (\|\tilde{b}_i\|)^2 \quad \text{for } i = 1, \ldots, n, \)
2. \( \|\tilde{b}_i\|^2 \leq \alpha^{-2} (\det L)^{1/2} \quad \text{and} \quad \|\tilde{b}_i\|^2 \geq \alpha^{-1} (\det L)^{1/2} \).

Consider the QR-factorization \( B = QR \) of the basis matrix \( B = [b_1, \ldots, b_n] \in \mathbb{Z}^{d \times n} \), where \( Q \in \mathbb{R}^{d \times d} \) is an orthogonal matrix and \( R = [r_{i,j}] = [r_1, \ldots, r_n] \in \mathbb{R}^{d \times n} \) is an upper triangular matrix, \( r_{i,i} = 0 \) for \( i > j \). We have \( \mu_{j,i} = r_{i,j}/r_{i,i} \) and \( |b_i| = \|\tilde{b}_i\| \). We present the core of the LLL-reduction algorithm using the coefficients \( r_{i,j} \) of the matrix \( R \). The vector \( r_i \) is the orthogonal transform of \( b_i \).

**LLL**

**INPUT** \( b_1, \ldots, b_n \in \mathbb{Z}^d, \delta \)

**OUTPUT** \( b_1, \ldots, b_n \) LLL-reduced basis

1. \( l := 1 \) (\( l \) is the stage)
2. while \( l \leq n \) do

   compute \( r_l = (r_{1,l}, \ldots, r_{l,l}, 0, \ldots, 0)^T \),
   size-reduce \( b_l \) against \( b_{l-1}, \ldots, b_1 \),
   if \( l \neq 1 \) and \( \delta r_{l-1,l-1} r_{l-1,l} > r_{l-1,l}^2 + r_{l,l}^2 \)
   then swap \( b_{l-1}, b_l \), \( l := l - 1 \) else \( l := l + 1 \).

The LLL-algorithm locally reduces the \( 2 \times 2 \)-diagonal submatrices of \( R \) by successively decreasing the length of the first column vector in a \( 2 \times 2 \)-matrix.
While the coefficients $r_{i,j}$ of the matrix $R$ are not rational, rational arithmetic must use the rational numbers $\mu_{j,i}, ||\bar{b}_i||^2$ satisfying $r_{i,j} = \mu_{j,i}||\bar{b}_i||, r_{i,i} = ||\bar{b}_i||$. 

LLL-time bound. Following [Sc81] we measure the size of the input in terms of

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

where $D_i = ||\bar{b}_i||^2 \cdot (\log ||\bar{b}_i||^2$ is the Gramian determinant of the sublattice with basis $b_1, \ldots, b_i$. The $M_{\text{Sc}}$-measure differentiates large and small lattice determinants. It relates to the [LLL82]-measure $M = \max_i ||b_i||^2$ by $M \leq M_{\text{Sc}} \leq M^n$.

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

3 Segment LLL-Reduction.

Segments and local coordinates. Let the basis $b_1, \ldots, b_n \in \mathbb{Z}^d$ have dimension $n = k \cdot m$ and the QR-factorization $[b_1, \ldots, b_n] = QR$. We partition the basis matrix $B$ into $m$ segments $B_l = [b_{k(l-1)+1}, \ldots, b_{kl}]$ for $l = 1, \ldots, m$. Local reduction of two consecutive segments uses the $2k \times 2k$-submatrix $R_l := \begin{bmatrix} r_{kl+1,kl+1} & r_{kl+1,kl+2} \\ r_{kl+2,kl+1} & r_{kl+2,kl+2} \end{bmatrix}$ of $R \in \mathbb{R}^{2k \times 2k}$, corresponding to two consecutive segments $B_{l-1}, B_l$. More precisely, local reduction uses the rational representation of $R_l$ by the coefficients $\mu_{kl+1,kl+1}$ and $||\bar{b}_{kl+1}||^2$. 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 $D(l) = \begin{bmatrix} r_{kl+1,kl+1} \\ r_{kl+2,kl+1} \\ \vdots \\ r_{kl+2,kl+2} \end{bmatrix}$ 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 $b_1, \ldots, b_n \in \mathbb{Z}^d, n = km$, $k$-segment LLL-reduced with $\delta \in \left[\frac{1}{4}, 1\right]$ if it is size-reduced and satisfies for $\alpha = 1/(\delta - \frac{1}{4})$:
1. \( \delta \|b_i\|^2 \leq \mu_{i+1}^2 \|b_i\|^2 + \|b_{i+1}\|^2 \) for \( i \neq 0 \mod k \),
2. \( D(l) \leq (\alpha/\delta)^k D(l+1) \) for \( l = 1, \ldots, m-1 \),
3. \( \delta^{k^2} \|b_k\|^2 \leq \alpha \|b_{k+1}\|^2 \) for \( l = 1, \ldots, m-1 \).

We use Inequality 3. to bound in Theorem 2 \( \|b_1\| \) by the successive minimum \( \lambda_1 \). Without Inequality 3, we bound in Theorem 5 \( \|b_1\| \) by \((\det L)^{\frac{1}{2}}\) as well as by \( \|\hat{b}_n\| \). The large exponent \( k^2 \) of \( \delta^{k^2} \) 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. \( \delta^{k^2+j-i} \|b_i\|^2 \leq \alpha^{j-i} \|b_j\|^2 \) for \( 1 \leq i < j \leq n \),
2. \( \delta^{n-1} \|b_1\|^2 \leq \alpha^{n-1} \|\hat{b}_n\|^2 \) and \( \delta^{k^2+j-1} \|b_1\|^2 \leq \alpha^{j-1} \|\hat{b}_j\|^2 \) 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 \( D(l) = \|\hat{b}_{k(l-1)+1}\|^2 \cdots \|\hat{b}_{kl}\|^2 \), there exists \( s \), \( 1 \leq s \leq k \) such that
\[
\|\hat{b}_{k(l-1)+s}\|^2 \leq (\alpha/\delta)^{k^2(l'-l)} \|\hat{b}_{k(l'-1)+s}\|^2.
\]
Inequality 1. follows by combining the latter inequality — at each end \( k(l-1)+s \) and \( k(l'-1)+s \) — with some inequalities \( \|\hat{b}_c\|^2 \leq \alpha \|\hat{b}_{c+1}\|^2 \), 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 \( \delta^{n-1} \|b_1\|^2 \leq \alpha^{n-1} \|\hat{b}_n\|^2 \). If \( i = 1 \) the pair \( \{i, k(l-1)+s\} \) is in one segment and thus \( \delta^{k^2+j-1} \|b_1\|^2 \leq \alpha^{j-1} \|\hat{b}_j\|^2 \).

**Theorem 2.** Let \( b_1, \ldots, b_n \) be a basis that is \( k \)-segment LLL-reduced with \( \delta \). Then we have for \( i = 1, \ldots, n : \)
\[
\delta^{2k^2+n-1} \|b_i\|^2 \leq \alpha^{n-1} \lambda_i^2 \text{ and } \delta^{k^2+i-1} \|b_i\|^2 \leq \alpha^{i-1} \|b_i\|^2,
\]
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 \textbf{segment LLL} transforms a given basis into a \( k \)-segment reduced basis. It iterates local LLL-reduction of two segments \([B_{l-1}, B_{l}] = \{b_{k(l-k+1)}, \ldots, b_{k(k+1)}\} \) via

The procedure \textbf{loc-LLL}(\( l \)). Given the orthogonalization of a \( k \)-segment reduced basis \( b_1, \ldots, b_{k(k+1)} \) the procedure \textbf{loc-LLL}(\( l \)) computes the orthogonalization and
size-reduction of the segments $B_{i-1}, B_i$. In particular it provides the submatrix $R_t \in \mathbb{R}^{3k \times 2k}$ of $R \in \mathbb{R}^{d \times n}$ corresponding to the segments $B_{i-1}, B_i$. Thereafter it performs a local LLL-reduction of $R_t$ and stores the LLL-transformation in the matrix $H \in \mathbb{Z}^{2k \times 2k}$. Finally, it transforms $[B_{i-1}, B_i]$ into the locally reduced segments $[B_{i-1}, B_i]H$ and size-reduces $[B_{i-1}, B_i]$ globally.

$$R = \begin{bmatrix} \begin{array}{c|c} R_t & \hline \mathbf{0} & \end{array} \end{bmatrix}$$

**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 $R_t$. 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** $b_1, \ldots, b_n \in \mathbb{Z}^d$, $k, m, n = km$, $\delta$

**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

   \[
   \begin{array}{l}
   (D(l-1) \geq (\alpha/\delta)^k D(l) \text{ or } \delta \ll \|b_{k(l-1)}\|^2 > \alpha \|\hat{b}_{k(l-1)+1}\|^2 ) \\
   \text{then } l := l - 1 \text{ else } l := l + 1.
   \end{array}
   \]

end

The original LLL-algorithm — with $\delta$ replaced by $\delta^2$ — essentially coincides with the case $k = 1$ of **segment LLL**.  \(^1\)

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

6
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 $r_{kl-k+1}, \ldots , r_{kl+k} \in \mathbb{R}^d$ 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

$$D(l-1) > (\alpha/\delta)^2 D(l) \text{ or } \delta^{k^2} ||b_{kl}||^2 > \alpha ||b_{kl+1}||^2$$

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. $decr \leq 2^{m-1} \log_{1/\delta} M_{Sc} < 2^{m} \log_{1/\delta} M_{Sc}$.

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 $\delta^2$.

2. If the reduction reverses the order of the basis $b_1, \ldots, b_n$ we have $decr \geq \binom{m}{2} = \frac{m(m-1)}{2}$. Then, by Theorem 3 we must have that $\log_{1/\delta} M_{Sc} \geq nk/4$, $M_{Sc} \geq \delta^{-nk/4}$. 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 $D_{i(l+1)}$, the Gramian determinant of $b_1, \ldots, b_{kl+k}$, and leaves $D_{kl'}$ for $l' \neq l-1$ 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 $D(l-1) > (\alpha/\delta)^2 D(l)$ holds upon entry of loc-LLL. As $D^{ter}(l-1) \leq \delta^{k^2} D^{ter}(l)$ holds upon termination, and since the product $D(l-1)D(l)$ does not change we have that

$$D^{ter}(l-1) \leq \alpha^{k^2} D^{ter}(l) = \alpha^{k^2} D(l-1)D(l)/D^{ter}(l-1) \leq \delta^{k^2} D(l-1)/D^{ter}(l-1),$$

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

If loc-LLL($l$) is performed in the case $\delta^{k^2} ||b_{kl}||^2 > \alpha ||b_{kl+1}||^2$ the previous argument shows again that $D^{ter}(l-1) \leq \delta^{k^2/2}D(l-1)$. Hence, lo...
\[ \text{decr} \leq \log_{1/\delta}^{1/2} M_{Sc}^{m-1} \leq \frac{2^{m-1}}{\delta^2} \log_2 M_{Sc}. \] 

**Theorem 4.** For \( k = \Theta(m) = \Theta(\sqrt{n}) \) segment LLL performs \( O(nd \log_1 M_{Sc}) \) arithmetic steps using integers of bit length \( O(\log_2 M_{Sc}) \).

**Proof.** Time bound. There are at most \( (n \log_1 M_{Sc}) \) LLL-exchanges — each requiring \( O(k^2) \) steps for local size-reduction. There are \( \text{decr} \leq 2^{m-1} \log_1 M_{Sc} \) calls of \( \text{loc-LLL} \) — each requiring \( O(ndk) \) arithmetic steps for global size-reduction, global orthogonalization and global transformation of two consecutive segments. The choice \( k, m = \Theta(\sqrt{n}) \) equilizes for \( d = O(n) \) the theoretical time bounds \( O(k^2 \log_1 M_{Sc}) \) for the LLL-exchanges in local coordinates and \( O(n^2k \log_1 M_{Sc}) \) for the global overhead.

We need that \( M_{Sc} \geq 2^n \), otherwise the \( O(nd \log_1 M_{Sc}) \) bound does not cover the \( O(n^2d) \) steps required for \( QR \)-factorization, the \( m-1 \) calls \( \text{loc-LLL}(l) \) for \( l = 1, \ldots, m-1 \) also require \( O(n^2d) \) steps for global size-reduction.

**Size of the integers.** We first show that the initial bound
\[ M_{Sc} = \max_{i=1, \ldots, n}(2^n, \|b_i\|^2, D_i) \]
can temporarily increase not more than by a factor \( 2^n \) 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 \( \|b_i\|^2 = D_i/D_{i+1} \) is a rational integer, \( M_{Sc}^{-1} \leq \|b_i\|^2 \leq M_{Sc} \) 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 \( b_i := b_i - [\mu_i, h]b_h \) for \( h = i-1, \ldots, 1 \). Assuming that \( b_1, \ldots, b_{i-1} \) is already LLL-reduced we have that \( |\mu_i, h| = \frac{\|b_i, h\|^2}{\|b_h\|^2} \leq \|b_i\|/\|b_h\| \leq \sqrt{M_{Sc}^{2n\alpha^2-1}} \).

We see that, during size-reduction of \( b_i \), the value \( \max(||b_1||^2, \ldots, ||b_i||^2) \) can temporarily increase not more than by a factor \( 2^n \) for \( \delta \geq \frac{3}{4} \).

Consider the coefficients of the matrix \( H \in \mathbb{Z}^{2k \times 2k} \) representing the local LLL-reduction of the segments \( B_{i-1}, B_i \) so that local LLL-reduction of \( B_{i-1}, B_i \) transforms \( [B_{i-1}, B_i] := [B_{i-1}, B_i]H \). We let \( b_i, b_j, \mu_{i, j} \) denote the values corresponding to the transformed segments \( [b_{ki-k+i+1}, \ldots, b_{ki+k}] = [B_{i-1}, B_i]b_i \). We let \( ||H||_1 \) denote the maximal \( ||\cdot||_1 \)-norm of the columns of \( H \).

**Lemma 2.** [Sc:84, Inequality (3.3)] We have that
1. \( H = ([\mu_{i, j}]^T)^{-1} [\tilde{b}_i, \tilde{b}_j][\tilde{b}_i, \tilde{b}_j]^T, \)
2. \( ||H||_1 \leq (2k)^\frac{2}{3} (\frac{2k}{4})^{3n-1} M_{Sc} \leq M_{Sc}^2 \).

**Proof.** Equality 1. follows from the equations
\[ \begin{align*}
[b_{ki-k+i+1}, \ldots, b_{ki+k}] &= [\tilde{b}_{ki-k+i+1}, \ldots, \tilde{b}_{ki+k}][\mu_{i, j}]^T, \\
[b'_{ki-k+i+1}, \ldots, b'_{ki+k}] &= [\tilde{b}'_{ki-k+i+1}, \ldots, \tilde{b}'_{ki+k}][\mu'_{i, j}]^T \\
&= [\tilde{b}_{ki-k+i+1}, \ldots, \tilde{b}_{ki+k}][\mu_{i, j}]^T H.
\end{align*} \]

When starting the local LLL-reduction of \( \tilde{B}_k \) the segments are already size-reduced, i.e., \( |\mu_{j, i}| \leq \frac{1}{4} \) for \( kl - k < i < j < kl + k \). Then the coefficients \( \nu_{j, i} \) of the inverse matrix \( [\nu_{j, i}]^{-1} \) satisfy \( |\nu_{j, i}| \leq (\frac{2}{4})^{|j-i|} \). Inequality 2. follows from 1. as \( \|\tilde{b}_i, \tilde{b}_j\| \leq \|\tilde{b}_i\|/\|\tilde{b}_j\| \leq M_{Sc} \) and \( |\mu_{j, i}| \leq \frac{1}{4} \) for \( i < j \). \( \square \)
Conclusion. All integers arising during the reduction are bounded in absolute value by \( \max(M_3^2, 2 \alpha^{n-1} M_{SC}) \). 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] implementation, 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, sub-subsegments, etc. Let \( n = k_1 \cdots k_s \) be a product of integers \( k_1, \ldots, k_s \geq 2 \), \( s \leq \log_2 n \). We consider segments of size \( n/k_1 \), subsegments of size \( n/(k_1 k_2 \cdots k_{s-1}) \), sub-subsegments of size \( n/(k_1 k_2 \cdots k_{s-2}) \) and so on. We denote \( k(\sigma) := (k_1, \ldots, k_s) \), \( k_\sigma := k_1 \cdots k_\sigma \) for \( \sigma = 1, \ldots, s \). There are \( s - 1 \) levels of segments, we have \( k_\sigma \)-segments \( [b_{k_\sigma(\sigma-1)+1}, \ldots, b_{k_\sigma}] \) of size \( k_\sigma \) for \( \sigma = 1, \ldots, s - 1 \). For \( n = k_1 \cdots k_s = k \), we let \( D(l, k(\sigma)) = [||b_{k_\sigma(\sigma-1)+1}||^2 \cdots ||b_{k_\sigma}||^2 \) denote the local determinant of the \( l \)-th \( k_\sigma \)-segment. Also, let \( k_0 := 1 \), \( D(l, k(0)) := ||b_l||^2 \). For \( n = 2^s \) it is natural to choose \( k_1 = k_2 = \cdots = k_s = 2 \).

Recall that Inequality 3. of Definition 2 is impractical for segments of size \( k_\sigma \) greater than \( \sqrt{n} \). We want to use large \( k_\sigma \)-segments, where the exponent \( k_\sigma^2 \) of \( \delta k_\sigma^2 \) 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 \( b_1, \ldots, b_n = k \cdot m \) of lattice \( L \) satisfy the inequalities 1. and 2. of Definition 2. Then we have that

1. \( ||b_{k_\sigma(l+1)}||^2 \leq (\alpha/\delta)^{(l'-l)\frac{s-1}{2}} ||b_{k_\sigma(l)}||^2 \) for \( 1 \leq l < l' \leq n \),
2. \( ||b_1||^2 \leq (\alpha/\delta)^{\frac{s-1}{2}} (\det L)^{\frac{1}{s}} \),
3. \( ||b_{k_\sigma}||^2 \geq (\delta/\alpha)^{\frac{s-1}{2}} (\det L)^{\frac{1}{s}} \).

For comparison LLL-reduced bases \( b_1, \ldots, b_n \) satisfy \( ||b_1||^2 \leq (\alpha^{n-1} \delta)^{\frac{1}{n}} \). Inequality 2. is a bit weaker than the inequality \( ||b_1||^2 \leq (\alpha^{n-1} \delta)^{\frac{1}{n}} \). 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 \( \|b_{k+1}^i\|^2 \leq (\alpha/\delta)^{i-1}\|b_{k+1}^i\|^2 \) for \( 1 \leq i \leq k \) that hold within the segments, and an inequality \( \|b_{k(i-1)+i}\|^2 \leq (\alpha/\delta)^{k(i'-i)}\|b_{k(i'-i)+1}\|^2 \) for some \( 1 \leq s \leq k \) that bridges the segments \( B_t \) and \( B_{t'} \). The latter inequality holds as \( D(l) = \|b_{k(i-1)+1}\|^2 \cdots \|b_{k+1}^i\|^2 \) satisfies \( D(l) \leq (\alpha/\delta)^{k(i'-i)} D(l'). \)

To prove Inequality 2, we note that \( D(1) = \|b_1\|^2 \cdots \|b_k\|^2 \) and \( \|b_1\|^2 \leq \alpha^{i-1}\|b_i\|^2 \) for \( i = 1, \ldots, k \) imply that
\[
\|b_1\|^2 \leq \alpha^{i-1} D(1) \frac{\alpha^{i-1}}{\delta}.
\]
Moreover, \( D(1) \cdots D(m) = (\det L)^2 \) and \( D(1) \leq (\alpha/\delta)^{k^2(i-1)} D(i) \) imply that
\[
D(1) \leq (\alpha/\delta)^{k^2 m^2} (\det L) \frac{\alpha^{i-1}}{\delta}.
\]
Combining the two inequalities yields Inequality 2.

We get Inequality 3. by applying Inequality 2. to the dual basis \( b_1', \ldots, b_n^* \) satisfying \( \langle b_l', b_j \rangle = b_{i,j}, \|b_l'\| = \|b_n^*\|^{-1} \) and \( \det(L^*) = (\det L)^{-1} \).

**Definition 3.** A basis \( b_1, \ldots, b_n \in \mathbb{Z}^d, n = k_1 \cdots k_s = k, \) is called \( \mathbf{k}(s) \)-segment LLL-reduced with \( \delta \in [1,\frac{2}{3}] \) if it is size-reduced and satisfies for \( \alpha = 1/(\delta - \frac{1}{3}) : \)
\[
D(l, \mathbf{k}(\sigma)) \leq (\alpha/\delta)^{(k_l^s)} D(l + 1, \mathbf{k}(\sigma))
\]
for \( l \neq 0 \mod k_{s+1}, \) for \( \sigma = 0, \ldots, s - 1, \) and \( l = 1, \ldots, n/k_{s+1} - 1. \)

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

Due to the restriction \( l \neq 0 \mod k_{s+1} \) the inequalities (1) only hold within \( k_{s+1} \)-segments, they cannot bridge distinct \( k_{s+1} \)-segments. The inequalities (1) get weaker and weaker as the size \( k_s \) of the segments increases and \( \delta^k \) decreases.

For \( s = 0, k_0 = \mathbf{k}_0 = 1, \) the inequalities (1) mean that \( \|b_l\|^2 \leq \alpha \|b_{l+1}\|^2 \) for \( l = 1, \ldots, n - 1, l \neq 0 \mod k_1 \) — 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 \( b_1, \ldots, b_n, n = k_1 \cdots k_s = k, \) satisfies \( \|b_1\|^2 \leq (\alpha/\delta^{s-1})^{m_i} (\det L) \frac{\alpha^{i-1}}{\delta} \) and \( \|b_n\|^2 \geq (\delta^{s-1}/\alpha)^{m_i} (\det L) \frac{\alpha^{i-1}}{\delta}. \)

Proof. We prove by induction on \( \sigma \) that
\[
\|b_1\|^2 \leq (\alpha/\delta^{s-1})^{m_i} D(1, \mathbf{k}(\sigma)) \frac{\alpha^{i-1}}{\delta}.
\]

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 \( \|b_1\|^2 \leq \alpha^{i-1}\|b_i\|^2 \) for \( i = 1, \ldots, k_1 \) and \( D(1, \mathbf{k}(1)) = \|b_1\|^2 \cdots \|b_{k_1}\|^2. \)

The induction hypothesis extends from \( \sigma \) to \( \sigma + 1 \) by the inequalities
\[
D(1, \mathbf{k}(\sigma)) \leq (\alpha/\delta)^{(k_l^s)} D(l, \mathbf{k}(\sigma))
\]
for \( l = 1, \ldots, k_{s+1} \) and the equation \( D(l, \mathbf{k}(\sigma + 1)) = \prod_{i=1}^{k_{s+1}} D(l, \mathbf{k}(\sigma)). \)
\textbf{k}_{\sigma}-\text{segment LLL-reduction.} The algorithm \textbf{k}_{\sigma}-\text{segment LLL} transforms a given basis into a \textbf{k}_{\sigma}-\text{segment LLL}-reduced basis. It iterates \textbf{k}_{\sigma-1}-\text{segment LLL} reduction of two \textbf{k}_{\sigma-1}-\text{segments} \quad [B_{l-1}, B_l] = [b_{k_{\sigma-1}(l-1)+1}, \ldots, b_{k_{\sigma-1}(l+1)}] via the procedure \textbf{k}_{\sigma-1}-\text{segment LLL}. For \( \sigma \geq 1 \) the procedure \textbf{k}_{\sigma}-\text{segment LLL}(l)

- computes the orthogonal transform of the \textbf{k}_{\sigma}-segments \([B_{l-1}, B_l]\) providing the local \( R \)-matrix \( R_l \in \mathbb{R}^{2k_{\sigma} \times 2k_{\sigma}} \),
- performs a local \textbf{k}_{\sigma-1}-\text{segment LLL}-reduction on \([B_{l-1}, B_l]\) by iterating \textbf{k}_{\sigma-1}-\text{segment LLL},
- stores the corresponding transformation matrix \( H_l \in \mathbb{Z}^{2k_{\sigma} \times 2k_{\sigma}} \),
- upon termination, it transports \( H_l \) to the matrix \( H_{l'} \) of the \textbf{k}_{\sigma+1}-\text{segment} \( B_{l'} \) that contains \( B_{l-1}, B_l \). The \( 2k_{\sigma} \) 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 \textbf{k}_{\sigma}-\text{segment LLL} for \( \sigma \geq 1 \) recursively calls \textbf{k}_{\sigma-1}-\text{segment LLL}, the procedure \textbf{k}_0-\text{segment LLL}(l) exchanges \( b_{l-1} \) and \( b_l \) in case that \( \|b_{l-1}\|^2 > \alpha\|b_l\|^2 \).

\textbf{k}_{\sigma}-\text{segment LLL}

INPUT \quad \( b_1, \ldots, b_n \in \mathbb{Z}^d, \quad n = k_1 \cdots k_s = k \), \( \delta \)

OUTPUT \quad \( b_1, \ldots, b_n \) \quad \( \text{\textbf{k}_{\sigma}-\text{segment LLL}-reduced basis} \)

1. \quad \( l := 1 \)

2. \quad \text{while} \( l \leq k_s - 1 \) \text{\quad do}

\textbf{k}_{\sigma-1}-\text{segment LLL}(l)

\quad \text{if} \quad l \neq 1 \quad \text{and} \quad D(l-1, k(s-1)) > (\alpha/\delta)^{(k_{\sigma-1})^2} D(l, k(s-1))

\quad \quad \text{then} \quad l := l - 1 \quad \text{else} \quad l := l + 1.

Theorem 7. \textit{Given a basis} \( b_1, \ldots, b_n \in \mathbb{Z}^d, \quad n = k_1 \cdots k_s = k \), \textit{the algorithm \textbf{k}_{\sigma}-\text{segment LLL} produces a \textbf{k}_{\sigma}-\text{segment LLL}-reduced basis. It performs at most} \( O(\alpha n^2 + d \sum_{\sigma=1}^s k_{\sigma}^2 \log_2 M_{\sigma}) \) \textit{arithmetic steps. If} \( \max_{\sigma} k_{\sigma} = O(1) \) \textit{and} \( \log_2 M_{\sigma} = O(n^2) \), \textit{the number of arithmetic steps is} \( O(n^2 d \log_2 n) \).

\textit{Proof.} Consider for \( \sigma = 1, \ldots, s - 1 \) the number of executions of \textbf{k}_{\sigma}-\text{segment LLL} as sub-subroutine of \textbf{k}_{\sigma}-\text{segment LLL}. The number of these executions is \( n/k_{\sigma} - 1 + 2 \cdot \text{dec}\!(k(\sigma)) \), where \( \text{dec}\!(k(\sigma)) \) is the number of times that \textbf{k}_{\sigma}-\text{segment LLL} is called as sub-subroutine of \textbf{k}_{\sigma}-\text{segment LLL} due to a violated inequality (1), where \( D(l-1, k(\sigma)) > (\alpha/\delta)^{(k_{\sigma})^2} D(l, k(\sigma)) \) for some \( l \).

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

Consider the product of the Gramian determinants

\( D(k(\sigma)) = \prod_{l=1}^{n/k_{\sigma}} D_{k_{\sigma}, l} = \prod_{l=1}^{n/k_{\sigma}} (D(1, k(\sigma)) \cdots D(l, k(\sigma))) \).
We apply Theorem 3 to $k_\sigma$-segments. Each execution of $k_\sigma$-segment LLL —
due to a violated inequality (1) — decreases $D(k(\sigma))$ by the factor $\delta'k_{\sigma}^{1/2}$.
$D(k(\sigma))$ is product of $n/k_\sigma$ Gramian determinants. As initially $D(k(\sigma)) \leq M_{Scn/k_{\sigma}}$, and upon termination $D(k(\sigma)) \geq 1$ we see that
$$\text{decr}(k(\sigma)) \leq \frac{2n}{(k_{\sigma})} \log_{1/\delta} M_{Sc}.$$
In total there are $n/k_\sigma - 1 + \frac{2n}{(k_{\sigma})} \log_{1/\delta} M_{Sc}$ executions of $k_\sigma$-segment LLL
each inducing an overhead of $O(k_{\sigma}k_{\sigma+1}^2)$ arithmetic steps. This overhead includes:
the orthogonal transform of the $k_\sigma$-segments $B_{t-1}$, $B_t$ providing the local
$R$-matrix $R_t \in \mathbb{R}^{2k_{\sigma} \times 2k_{\sigma}}$ of $[B_{t-1}, B_t]$, the transport of $H_t$ to the transformation
matrix $H_{\sigma}$ of the next higher level and size-reduction of $R_t$ against $R_{\sigma}$. The total
overhead of all $k_\sigma$-segment LLL executions is
$$O(nk_{\sigma+1}^2 + nk_{\sigma+1}^2 \log_{1/\delta} M_{Sc}).$$
In the particular case $\sigma = s - 1$, this overhead is $O(dk_{\sigma}^2 + dk_{\sigma}^2 \log_{1/\delta} M_{Sc})$ as
the global transforms are done directly on the basis vectors in $\mathbb{Z}^d$.
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 $O(k_0^2)$ arithmetic steps for
a local exchange and a local size-reduction. We see that $k_\sigma$-segment LLL
performs $O(n^2d + d \sum_{\sigma=1}^{s-1} k_{\sigma}^2 \log_{1/\delta} M_{Sc})$ arithmetic steps providing the time
bound of the theorem.

**Conclusion.** The time bound of the novel algorithm $k_\sigma$-segment LLL is
comparable to that of classical matrix multiplication. The size of the integers
occurring 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 $\sigma$. Each level can add $O(\log_2 M_{Sc})$ bits when transporting
the local transformation $H_t$ to the next higher level.

**References**

[BN00] D. Bleichenbacher and P. Q. Nguyen, Noisy Polynomial Interpolation and Noisy

[Bo00] D. Boneh, Finding Smooth Integers in Small Intervals Using CRT Decoding,

[Ca00] J. Cai, The Complexity of some Lattice Problems, Proc. ANTS’00, LNCS 1838,

[Co97] D. Coppersmith, Small Solutions to Polynomial Equations, and Low Exponent

[K84] R. Kannan, Minkowski’s Convex Body Theorem and Integer Programming,


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


