# Progress on LLL and Lattice Reduction

Claus Peter Schnorr

**Abstract** We surview variants and extensions of the LLL-algorithm of LENSTRA, LENSTRA LOVÁSZ, extensions to quadratic indefinite forms and to faster and stronger reduction algorithms. The LLL-algorithm with Householder orthogonalisation in floating-point arithmetic is very efficient and highly accurate. We surview approximations of the shortest lattice vector by feasible lattice reduction, in particular by block reduction, primal-dual reduction and random sampling reduction. Segment reduction performs LLL-reduction in high dimension mostly working with a few local coordinates.

**Key words:** LLL-reduction, Householder orthogonalisation, floating-point arithmetic, block reduction, segment reduction, primal-dual reduction, sampling reduction, reduction of indefinite quadratic forms.

## 1 Introduction

A *lattice basis* of dimension $n$ consists of $n$ linearly independent real vectors $\mathbf{b}_1, ..., \mathbf{b}_n \in \mathbb{R}^m$. The basis $B = [\mathbf{b}_1, ..., \mathbf{b}_n]$ generates the *lattice* $\mathcal{L}$ consisting of all integer linear combinations of the basis vectors. Lattice reduction transforms a given basis of $\mathcal{L}$ into a basis with short and nearly orthogonal vectors.

*Unifying two traditions.* The LLL-algorithm of LENSTRA, LENSTRA LOVÁSZ [47] provides in polynomial time a reduced basis of proven quality. Its inventors focused on Gram-Schmidt orthogonalisation (GSO) in exact integer arithmetic which is only affordable for small dimension $n$. With floating-point arithmetic (*fpa*) available it is faster to orthogonalise in *fpa*. In numerical mathematics the orthogonalisation of

Claus Peter Schnorr
Fachbereich Informatik und Mathematik, Universität Frankfurt, PSF 111932, D-60054 Frankfurt am Main, Germany. schnorr@cs.uni-frankfurt.de – http: www.mi.informatik.uni-frankfurt.de

a lattice basis $B$ is usually done by $QR$-factorization $B = QR$ using Householder reflections [76]. The LLL-community has neglected this approach as it requires square roots and does not allow exact integer arithmetic. We are going to unify these two separate traditions.

Practical experience with the GSO in *fpa* led to the LLL-algorithm of [66] implemented by EUCHNER that tackles the most obvious problems in limiting *fpa*-errors by heuristic methods. It is easy to see that the heuristics works and short lattice vectors are found. The LLL of [66] with some additional accuracy measures performs well in double *fpa* up to dimension 250. But how to proceed in higher dimensions ? [65] gives a proven LLL in approximate rational arithmetic. It uses an iterative method by Schulz to recover accuracy and is not adapted to *fpa*. Its time bound in bit operations is a polynomial of degree 7, resp., 6+$\varepsilon$ using school-, resp. FFT-multiplication while the degree is 9, resp., $7 + \varepsilon$ for the original LLL.

In 1996 RÖSSNER implemented in his thesis a continued fraction algorithm applying [31] and Householder orthogonalization (HO) instead of GSO. This algorithm turned out to be very stable in high dimension [63]. Our experience has well confirmed the higher accuracy obtained with HO, e.g., by attacks of *May* and *Koy* in 1999 on NTRU- and GGH-cryptosystems. The accuracy and stability of the $QR$-factorization of a basis $B$ with HO has been well analysed for backwards accuracy [26], [34]. These results do not directly provide bounds for worst case forward errors. As fully proven *fpa*-error bounds are in practice far to pessimistic we will use strong heuristics.

**Outline.** Section 2 presents the LLL in ideal arithmetic and discusses various extensions of the LLL by deep insertions that are more powerful than LLL swaps. Section 3 extends the LLL to arbitrary quadratic forms, indefinite forms included.

Section 4 presents $\mathbf{LLL}_H$, an LLL with HO together with an analysis of forward errors. Our heuristics assumes that small error vectors have a negligible impact on the vector length as correct and error vectors are in high dimension very unlikely near parallel. It is important to weaken size-reduction under *fpa* such that the reduction becomes independent of *fpa*-errors. This also prevents infinite cycling of the algorithm. Fortunately, the weakened size-reduction has a negligible impact on the quality of the reduced basis. $\mathbf{LLL}_H$ of [70] and the $L^2$ of [56] are adapted to *fpa*, they improve the time bounds of the theoretical LLL of [65] and are well analysed. $L^2$ provides provable correctness, it is quadratic in the bit length of the basis. However it looses about half of the accuracy compared to $\mathbf{LLL}_H$ and it takes more time.

Sections 5–7 survey practical improvements of the LLL that strengthen LLL-reduction to find shorter lattice vectors and better approximations of the shortest lattice vector. We revisit block reduction from [64] extend it to KOY's primal-dual reduction and combine it with random sampling reduction of [69].

Sections 8, 9 survey recent results of [70], they speed up $\mathbf{LLL}_H$ for large dimension $n$ to LLL-type segment reduction that goes back to an idea of SCHÖNHAGE [71]. This reduces the time bound of $\mathbf{LLL}_H$ to a polynomial of degree 6, resp., 5+$\varepsilon$ and preserves the quality of the reduced bases. Iterating this method by iterated subseg-

ments performs LLL-reduction in $O(n^{3+\varepsilon})$ arithmetic steps using large integers and *fpa*-numbers. It is still open how to turn this into a practical algorithm.

For general background on lattices and lattice reduction see [13], [16], [52], [54], [59]. Here is a small selection of applications in number theory [8], [47], [14], [72] computational theory [11], [27], [31], [45], [49], [53] cryptography [1], [10],[9], [17], [54], [58], [61] and complexity theory [2], [12], [20], [25],[36], [54]. Standard program packages are LIDIA, Magma and NTL.

**Notation, GSO and GNF.** Let $\mathbb{R}^m$ denote the real vector space of dimension $m$ with *inner product* $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^t\mathbf{y}$. A vector $\mathbf{b} \in \mathbb{R}^m$ has *length* $\|\mathbf{b}\| = \langle \mathbf{b}, \mathbf{b} \rangle^{\frac{1}{2}}$. A sequence of linearly independent vectors $\mathbf{b}_1, ..., \mathbf{b}_n \in \mathbb{R}^m$ is a *basis*, written as matrix $B = [\mathbf{b}_1, ..., \mathbf{b}_n] \in \mathbb{R}^{m \times n}$ with columns $\mathbf{b}_i$. The basis $B$ generates the lattice

$$\mathcal{L} = \mathcal{L}(B) = \{B\mathbf{x} \,|\, \mathbf{x} \in \mathbb{Z}^n\} = \textstyle\sum_{i=1}^{n} \mathbf{b}_i\mathbb{Z} \subset \mathbb{R}^m.$$

It has *dimension* $\dim \mathcal{L} = n$. Let $\mathbf{q}_i$ denote the orthogonal projection of $\mathbf{b}_i$ in $\mathrm{span}(\mathbf{b}_1, ..., \mathbf{b}_{i-1})^{\perp}$, $\mathbf{q}_1 = \mathbf{b}_1$. The *orthogonal vectors* $\mathbf{q}_1, ..., \mathbf{q}_n \in \mathbb{R}^m$ and the *Gram-Schmidt coefficients* $\mu_{j,i}$, $1 \le i, j \le n$ of the basis $\mathbf{b}_1, ..., \mathbf{b}_n$ satisfy for $j = 1, ..., n$:

$$\mathbf{b}_j = \textstyle\sum_{i=1}^{j} \mu_{j,i}\mathbf{q}_i, \qquad \mu_{j,j} = 1, \qquad \mu_{j,i} = 0 \;\text{ for }\; i > j,$$

$$\mu_{j,i} = \langle \mathbf{b}_j, \mathbf{q}_i \rangle / \langle \mathbf{q}_i, \mathbf{q}_i \rangle, \qquad \langle \mathbf{q}_j, \mathbf{q}_i \rangle = 0 \text{ for } j \neq i.$$

The basis $B \in \mathbb{R}^{m \times n}$ has a unique $QR$-decomposition $B = QR$, where $Q \in \mathbb{R}^{m \times n}$ is *isometric* (i.e., $Q$ preserves the inner product, $\langle \mathbf{x}, \mathbf{y} \rangle = \langle Q\mathbf{x}, Q\mathbf{y} \rangle$, $Q$ can be extended to an orthogonal matrix $Q' \in \mathbb{R}^{m \times m}$) and $R = [r_{i,j}] \in \mathbb{R}^{n \times n}$ is upper-triangular ($r_{i,j} = 0$ for $i > j$) with positive diagonal entries $r_{1,1}, ..., r_{n,n} > 0$. Hence $Q = [\, \mathbf{q}_1/\|\mathbf{q}_1\|, ...., \mathbf{q}_n/\|\mathbf{q}_n\| \,]$, $\mu_{j,i} = r_{i,j}/r_{i,i}$, $\|\mathbf{q}_i\| = r_{i,i}$ and $\|\mathbf{b}_i\|^2 = \sum_{j=1}^{i} r_{j,i}^2$. Two bases $B = QR$, $B' = Q'R'$ are *isometric* iff $R = R'$, or equivalently iff $B^tB = B'^tB'$. We call $R$ the *geometric normal form* (GNF) of the basis, $\mathrm{GNF}(B) := R$. The GNF is preserved under isometric transforms $Q$, i.e., $\mathrm{GNF}(QB) = \mathrm{GNF}(B)$.

**The successive minima.** The $j$-th successive minimum $\lambda_j(\mathcal{L})$ of a lattice $\mathcal{L}$, $1 \le j \le \dim \mathcal{L}$, is the minimal real number $\rho$ for which there exist $j$ linearly independent lattice vectors of length $\le \rho$ ; $\lambda_1$ is the length of the shortest nonzero lattice vector.

**Further notation.** $\mathrm{GL}_n(\mathbb{Z}) = \{T \in \mathbb{Z}^{n \times n} \mid \det T = \pm 1\}$,
$R^{-t} = (R^{-1})^t = (R^t)^{-1}$ is the inverse transpose of $R \in \mathbb{R}^{n \times n}$,
$d_i = \det([\mathbf{b}_1, ..., \mathbf{b}_i]^t[\mathbf{b}_1, ..., \mathbf{b}_i])$, $d_0 = 1$, $\det \mathcal{L}(B) = \det(B^tB)^{1/2} = d_n^{1/2}$,
$\pi_i : \mathbb{R}^m \to \mathrm{span}(\mathbf{b}_1, ..., \mathbf{b}_{i-1})^{\perp}$ is the orthogonal projection, $\mathbf{q}_i = \pi_i(\mathbf{b}_i)$,
$\gamma_n = \max_{\mathcal{L}} \lambda_1^2(\mathcal{L}) / \det \mathcal{L}^{2/n}$ over all lattices $\mathcal{L}$ of $\dim \mathcal{L} = n$ is the HERMITE constant,
$U_k = \begin{bmatrix} & & 1 \\ & \cdot & \\ 1 & & \end{bmatrix} \in \mathbb{Z}^{k \times k}$; $B := BU_n$ / $B := U_mB$ reverses the order of columns/rows of $B \in \mathbb{R}^{m \times n}$, we write matrices $A = [a_{i,j}]$ with capital letters $A$ and small letter entries $a_{i,j}$, $I_n \in \mathbb{Z}^{n \times n}$ denotes the unit matrix, let $\varepsilon \in \mathbb{R}$, $0 \le \varepsilon \approx 0$.

**Details of floating point arithmetic.** We use the *fpa* model of WILKINSON [76]. A *fpa* number with $t = 2t' + 1$ *precision bits* is of the form $\pm 2^e \sum_{i=-t'}^{t'} b_i 2^i$, where $b_i \in \{0, 1\}$ and $e \in \mathbb{Z}$. It has bit length $t + s + 2$ for $|e| < 2^s$, two signs included. We denote the set of these numbers by $\mathbb{FL}_t$. Standard double length *fpa* has $t = 53$ precision bits, $t + s + 2 = 64$. Let $fl : \mathbb{R} \supset [-2^{2^s}, 2^{2^s}] \ni r \mapsto \mathbb{FL}_t$ approximate real numbers by *fpa* numbers. A step $c := a \circ b$ for $a, b, c \in \mathbb{R}$ and a binary operation $\circ \in \{+, -, \cdot, /\}$ translates under *fpa* into $\bar{a} := fl(a)$, $\bar{b} := fl(b)$, $\bar{c} := fl(\bar{a} \circ \bar{b})$, resp. into $\bar{a} := fl(\circ(\bar{a}))$ for unary operations $\circ \in \{\lceil\ \rfloor, \sqrt{}\}$. Each *fpa* operation induces a normalized relative error bounded in magnitude by $2^{-t}$: $|fl(\bar{a} \circ \bar{b}) - \bar{a} \circ \bar{b}|/|\bar{a} \circ \bar{b}| \le 2^{-t}$. If $|\bar{a} \circ \bar{b}| > 2^{2^s}$ or $|\bar{a} \circ \bar{b}| < 2^{-2^s}$ then $fl(\bar{a} \circ \bar{b})$ is undefined due to an *overflow*, resp. *underflow*.

Usually one requires that $2^s \le t^2$ and thus $s \le 2 \log_2 t$, for brevity we identify the bit length of *fpa*-numbers with $t$, neglecting the minor $(s+2)$-part. We use approximate vectors $\bar{\mathbf{h}}_l$, $\bar{\mathbf{r}}_l \in \mathbb{FL}_t^m$ for HO under *fpa* and exact basis vectors $\mathbf{b}_l \in \mathbb{Z}^m$.

## 2 LLL-Reduction and Deep Insertions

We describe reduction of the basis $B = QR$ in terms of the GNF $R = [r_{i,j}] \in \mathbb{R}^{n \times n}$.

*Standard reductions.* A lattice basis $B = QR \in \mathbb{R}^{m \times n}$ is *size-reduced* (for $\varepsilon$) if
$$|r_{i,j}|/r_{i,i} \le \tfrac{1}{2} + \varepsilon \quad \text{for all } j > i. \qquad (\textit{occasionally we neglect } \varepsilon)$$

$B = QR$ is LLL-*reduced* (or an LLL-*basis*) for $\delta \in (\eta^2, 1]$, $\eta = \frac{1}{2} + \varepsilon$, if $B$ is size-reduced and
$$\delta r_{i,i}^2 \le r_{i,i+1}^2 + r_{i+1,i+1}^2 \quad \text{for } i = 1, ..., n-1.$$

A basis $B = QR$ is HKZ-*reduced* (or an HKZ-*basis*) if $B$ is size-reduced, and each diagonal coefficient $r_{i,i}$ of the GNF $R = [r_{i,j}] \in \mathbb{R}^{n \times n}$ is minimal under all transforms in $\mathrm{GL}_n(\mathbb{Z})$ that preserve $\mathbf{b}_1, ..., \mathbf{b}_{i-1}$.

LLL-bases satisfy $r_{i,i}^2 \le \alpha\, r_{i+1,i+1}^2$ for $\alpha := 1/(\delta - \eta^2)$. This yields Theorem 1. A.K. LENSTRA, H.W. LENSTRA, JR. AND L. LOVÁSZ [47] introduced LLL-bases focusing on $\delta = 3/4$, $\epsilon = 0$ and $\alpha = 2$.

HKZ-bases are due to HERMITE [33] and KORKINE-ZOLOTAREFF [38]. LLL / HKZ-bases $B = QR$ are preserved under isometry. LLL / HKZ-reducedness is a property of the GNF $R$.

**Theorem 1. [47]** *An LLL-basis $B \in \mathbb{R}^{m \times n}$ of lattice $\mathcal{L}$ satisfies for $\alpha = 1/(\delta - \eta^2)$*

**1.** $\|\mathbf{b}_1\|^2 \le \alpha^{\frac{n-1}{2}} (\det \mathcal{L})^{2/n}$, **2.** $\|\mathbf{b}_1\|^2 \le \alpha^{n-1} \lambda_1^2$

**3.** $\|\mathbf{b}_i\|^2 \le \alpha^{i-1} r_{i,i}^2$, **4.** $\alpha^{-i+1} \le \|\mathbf{b}_i\|^2 \lambda_i^{-2} \le \alpha^{n-1}$ *for $i = 1, ..., n$.*

If an LLL-basis satisfies the stronger inequalities $r_{i,i}^2 \le \bar{\alpha} r_{i+1,i+1}^2$ for all $i$ and some $1 < \bar{\alpha} < \alpha$, then the inequalities **1-4** hold with $\alpha^{n-1}$ replaced by $\bar{\alpha}^{n-1}$ and with $\alpha^{i-1}$ in **3, 4** replaced by $\bar{\alpha}^{i-1}/(4\bar{\alpha} - 4)$.

Sections 5–9 survey variants of LLL- and HKZ-bases that either allow faster reduction for large dimension $n$ or provide a rather short vector $\mathbf{b}_1$. For these bases we either modify clause 1 or clause 2 of Theorem 1. Either clause is sufficient due to an observation of Lovász [49] pp.24 ff, that the following problems are polynomial time equivalent for all lattices $\mathcal{L}(B)$ given $B$:

**1.** Find $\mathbf{b} \in \mathcal{L}$, $\mathbf{b} \neq \mathbf{0}$ with $\|\mathbf{b}\| \leq n^{O(1)}\lambda_1(\mathcal{L})$.

**2.** Find $\mathbf{b} \in \mathcal{L}$, $\mathbf{b} \neq \mathbf{0}$ with $\|\mathbf{b}\| \leq n^{O(1)}(\det \mathcal{L})^{1/n}$.

**Theorem 2.** **[48]** *An HKZ-basis $B \in \mathbb{R}^{m\times n}$ of lattice $\mathcal{L}$ satisfies*

$$4/(i+3) \leq \|\mathbf{b}_i\|^2\lambda_i^{-2} \leq (i+3)/4 \quad \text{for } i = 1, ..., n.$$

The algorithms for HKZ-reduction [35], [21], exhaustively enumerate, for various $l$, all lattice vectors $\mathbf{b}$ such that $\|\pi_l(\mathbf{b})\| \leq \|\pi_l(\mathbf{b}_l)\|$ for the current $\mathbf{b}_l$. Their theoretical analysis has been stepwise improved [35], [32] to a proven $mn^{\frac{n}{2e}+o(n)}$ time bound for HKZ-reduction [28]. The enumeration **ENUM** of [66], [67] is particularly fast in practice, see [1] for a survey and [28], [60] for heuristic and experimental results.

**Alg. 1: LLL** *in ideal arithmetic*

INPUT $\mathbf{b}_1, ..., \mathbf{b}_n \in \mathbb{Z}^m$ a basis with $M_0 = \max\{\|\mathbf{b}_1\|, ..., \|\mathbf{b}_n\|\}$, $\delta$ with $\frac{1}{4} < \delta < 1$

1. $l := 1$, # *at stage* $l$ $\mathbf{b}_1, ..., \mathbf{b}_{\max(l-1,1)}$ *is an LLL-basis with given GNF.*

2. WHILE $l \leq n$ DO

2.1 #*compute* $\mathbf{r}_l = col(l, R)$:
FOR $i = 1, ..., l-1$ DO $[\ r_{i,l} := (\langle \mathbf{b}_i, \mathbf{b}_l\rangle - \sum_{k=1}^{i-1} r_{k,i}r_{k,l})/r_{i,i}\ ]$,
$r_{l,l} := |\ \|\mathbf{b}_l\|^2 - \sum_{k=1}^{l-1} r_{k,l}^2\ |^{1/2}$, $\mathbf{r}_l := (r_{1,l}, ..., r_{l,l}, 0, ...0)^t \in \mathbb{R}^n$.

2.2 #*size-reduce* $\mathbf{b}_l$ *and* $\mathbf{r}_l$, $\lceil r \rfloor = \lceil r - \frac{1}{2} \rceil$ *denotes the nearest integer to* $r \in \mathbb{R}$:
FOR $i = l-1, ..., 1$ DO $\mathbf{b}_l := \mathbf{b}_l - \lceil r_{i,l}/r_{i,i} \rfloor \mathbf{b}_i$, $\mathbf{r}_l := \mathbf{r}_l - \lceil r_{i,l}/r_{i,i} \rfloor \mathbf{r}_i$.

2.3 IF $l > 1$ and $\delta\, r_{l-1,l-1}^2 > r_{l-1,l}^2 + r_{l,l}^2$
THEN swap $\mathbf{b}_{l-1}, \mathbf{b}_l$, $l := l-1$ ELSE $l := l+1$.

3. OUTPUT LLL-basis $B = [\mathbf{b}_1, ..., \mathbf{b}_n]$, $R = [\mathbf{r}_1, ..., \mathbf{r}_n]$ for $\delta$.

*Comments.* **LLL** performs simultaneous column operations on $R$ and $B$, it swaps columns $\mathbf{r}_{l-1}, \mathbf{r}_l$ and $\mathbf{b}_{l-1}, \mathbf{b}_l$ if this shortens the length of the first column of the submatrix $\begin{bmatrix} r_{l-1,l-1} & r_{l-1,l} \\ 0 & r_{l,l} \end{bmatrix}$ of $R$ by the factor $\sqrt{\delta}$. To enable a swap the entry $r_{l-1,l}$ is first reduced to $|r_{l-1,l}| \leq \frac{1}{2}\,|r_{l-1,l-1}|$ by transforming $\mathbf{r}_l := \mathbf{r}_l - \lceil r_{l-1,l}/r_{l-1,l-1} \rfloor \mathbf{r}_{l-1}$. At stage $l$ we get $\mathbf{r}_l = \mathrm{col}(l, R)$ of $R = \mathrm{GNF}(B)$, and we have $\mathbf{r}_{l-1}$ from a previous stage. The equation $\mathrm{GNF}([\mathbf{b}_1, ...\mathbf{b}_l]) = [\mathbf{r}_1, ...\mathbf{r}_l]$ is preserved during simultaneous size-reduction of $\mathbf{r}_l$ and $\mathbf{b}_l$.

Each swap in step 2.3 decreases $\mathcal{D}^{(1)} := \prod_{i=1}^{n-1} d_i$ by the factor $\delta$. As initially $\mathcal{D}^{(1)} \leq M_0^{n^2}$ and $\mathcal{D}^{(1)}$ remains integer **LLL** performs $\leq n^2 \log_{1/\delta} M_0$ rounds, denoting $M_0 =$

$\max\{\|\mathbf{b}_1\|, ..., \|\mathbf{b}_n\|\}$ for the input basis. Each round performs $O(nm)$ arithmetic steps, **LLL** runs in $O(n^3 m \log_{1/\delta} M_0)$ arithmetic steps.

*Time bound in exact rational/integer arithmetic.* The rationals $r_{l,l}^2$, $\mu_{l,i} = r_{i,l}/r_{i,i}$ can easily be obtained within **LLL** in exact rational/integer arithmetic. Moreover, the integer $\mu_{j,i} d_i$ has bit length $O(n \log_2 M_0)$ throughout **LLL**-computations. This yields

**Theorem 3. LLL** *runs in* $O(n^5 m(\log_{1/\delta} M_0)^3)$*, resp.* $O(n^{4+\varepsilon} m(\log_{1/\delta} M_0)^{2+\varepsilon})$ *bit operations under school-, resp. FFT-multiplication.*

The degree of this polynomial time bound (in $n, m, \log M_0$) is 9 / $7+\varepsilon$ under school-/FFT-multiplication. The degree reduces to 7 / $6+\varepsilon$ by computing the GNF, resp., the GSO in floating-point arithmetic (*fpa*). This is done by $\mathbf{LLL}_H$ [70] of section 4 and the $L^2$ of [56], both use *fpa* numbers of bit length $O(n + \log_2 M_0)$. LLL-type segment reduction **SLLL** of [70] in section 9 further decreases the time bound degree to 6 / $5+\varepsilon$. The factor $m$ in the time bounds can be reduced to $m$ by performing the reduction under random projections [6]. The theoretical, less practical algorithms of [65], [73] approach the time bound of $\mathbf{LLL}_H$ by approximate rational, resp. very long integer arithmetic.

*A canonical order of the input basis vectors.* While the **LLL** depends heavily on the order of the basis vectors this order can be made nearly canonical by swapping before step 2.3 $\mathbf{b}_l$ and $\mathbf{b}_j$ for some $j \geq l$ that minimizes the value $r_{l-1,l}^2 + r_{l,l}^2$ resulting from the swap. This facilitates a subsequent swap of $\mathbf{b}_{l-1}, \mathbf{b}_l$ by step 2.3. Moreover, this tends to decrease the number of rounds of the **LLL** and anticipates subsequent deep insertions into the LLL-basis $\mathbf{b}_1, ...., \mathbf{b}_{l-1}$ via the following new step 2.3.

*New step* 2.3, *deep insertion at* $(j,l)$ [SE91], (the old step 2.3 is restricted to *depth* $l-j=1$):

`IF` $\exists j,\ 0 < j < l$ such that $\delta r_{j,j}^2 > \sum_{i=j}^{l} r_{i,l}^2$
`THEN` for the smallest such $j$ do $(\mathbf{b}_j, ..., \mathbf{b}_l) := (\mathbf{b}_l, \mathbf{b}_j, \mathbf{b}_{j+1}, ..., \mathbf{b}_{l-1})$, $l := j$ `ELSE` $l := l+1$.

**LLL** *with deep insertions* is efficient in practice, and runs with some restrictions in polynomial time. Deep insertion at $(j,l)$ decreases $r_{j,j}$ and $d_j = r_{j,j}^2 \cdots r_{n,n}^2$ and provides $r_{j+k,j+k}^{new}$ close to $r_{j+k-1,j+k-1}$ for $j+k < l$. This is because $\sum_{i=j}^{l} r_{i,l}^2 < \delta r_{j,j}^2$ and the $\langle \pi_j(\mathbf{b}_l), \mathbf{b}_{j+k-1} \rangle$ are quite small.

Deep insertion at $(j,l)$ can be strengthened by decreasing $\|\pi_j(\mathbf{b}_l)\|$ through additional, more elaborate size-reduction. Before testing $\delta r_{j,j}^2 > \sum_{i=j}^{l} r_{i,l}^2 = \|\pi_j(\mathbf{r}_l)\|^2$ shorten $\pi_j(\mathbf{b}_l)$ as follows

2.2.b *additional size-reduction of* $\pi_j(\mathbf{b}_l)$ :
`WHILE` $\exists h : j \leq h < l$ such that $\mu'_{j,h} := \sum_{i=j}^{h} r_{i,l} r_{i,h} / \sum_{i=j}^{h} r_{i,h}^2$ satisfies $\delta |\mu'_{j,h}| \geq \frac{1}{2}$
`DO` $\mathbf{b}_l := \mathbf{b}_l - \lceil \mu'_{j,h} \rfloor \mathbf{b}_h$, $\mathbf{r}_l := \mathbf{r}_l - \lceil \mu'_{j,h} \rfloor \mathbf{r}_h$.

Obviously, deep insertion with additional size-reduction remains polynomial time per round. This makes the improved deep insertion quite attractive and more efficient than the algorithms of sections 5–7. In addition one can perform deep insertion of

$\mathbf{b}_l$ and of several combinations of $\mathbf{b}_l$ with other lattice vectors. This leads to random sampling reduction of [69] to be studied in section 7.

*Decreasing $\alpha$ by deep insertion of depth $\leq 2$.* The bound $\|\mathbf{b}_1\| \leq \alpha^{\frac{n-1}{2}}(\det \mathcal{L})^{\frac{2}{n}}$ of Theorem 1 is sharp for the LLL-GNF $R = [r_{i,j}] \in \mathbb{R}^{n\times n}$: $r_{i,i} = \alpha^{(-i+1)/2}$, $r_{i,i+1} = \frac{1}{2}r_{i,i}$, $r_{i,j} = 0$ for $j \geq i+2$. (Note that deep insertion at $(1,n)$ results in $r_{1,1} = \lambda_1$ right away.) While this GNF satisfies $r_{i,i}^2/r_{i+1,i+1}^2 = \alpha$ the maximum of $r_{i,i}^2/r_{i+2,i+2}^2$ under LLL-reduction with deep insertion of depth $\leq 2$ for $\delta = 1$ is $\frac{3}{2}$, $\frac{3}{2} < \alpha^2 = \frac{16}{9}$, since the HKZ-reduced GNF of the critical lattice of dimension 3 satisfies $r_{1,1}^2/r_{3,3}^2 = \frac{3}{2}$. This shows that deep insertion of depth $\leq 2$ decreases $\alpha$ in clauses **1**, **2** of Theorem 1 from $\frac{4}{3}$ to $(\frac{3}{2})^{1/2}$.

*LLL-reduction of bases $B$ with large entries.* Similar to LEHMER's version of EUCLID's algorithm for large numbers [37] section 4.5.2, p. 342, most of the LLL-work can be done in single precision arithmetic:

Pick a random integer $\rho' \in_R [1, 2^{43}]$ and truncate $B = [b_{i,j}]$ into the matrix $B'$ with entries $b'_{i,j} := \lceil b_{i,j}\rho'/2^\tau \rfloor$ such that $|b_{i,j}\rho'/2^\tau - b'_{i,j}| \leq 1/2$ and $|b'_{i,j}| \leq \rho' + 1/2$, where $\tau := \max_{i,j}\lceil \log_2 |b_{i,j}| \rceil \gg 53$.

LLL-reduce $B'$ into $B'T'$ working mostly in single precision, e.g. with 53 precision bits, and transform $B$ into $BT'$. Iterate this process as long as it shortens $B$.

Note that $(2^\tau/\rho')B'$ and $(2^\tau/\rho')^n \det B'$ well approximate $B$ and $\det B$. In particular, $|\det B| \ll (2^\tau/\rho')^n$ implies that $\det B' = 0$, and thus LLL-reduction of $B'$ produces zero vectors of $B'T'$ and highly shortened vectors of $BT'$.

## 3 LLL-Reduction of Quadratic and Indefinite Forms

We present extensions of the LLL-algorithm to other domains than $\mathbb{Z}$ and general quadratic forms.

*Rational, algebraic and real numbers.* It is essential for the LLL-algorithm that the input basis $B \in \mathcal{R}^{m\times n}$ has entries in an euclidean domain $\mathcal{R}$ with efficient, exact arithmetic such as $\mathcal{R} = \mathbb{Z}$. Using rational arithmetic the LLL-algorithm directly extends from $\mathbb{Z}$ to the field of rational numbers $\mathbb{Q}$ and to rings $\mathcal{R}$ of algebraic numbers. However, the bit length of the numbers occuring within the reduction requires further care. If the rational basis matrix $B$ has an integer multiple $\mathrm{d}B \in \mathbb{Z}^{m\times n}$, $\mathrm{d} \in \mathbb{Z}$ of moderate size then the LLL-algorithm applied to the integer matrix $\mathrm{d}B \in \mathbb{Z}^{m\times n}$ yields an LLL-basis of $\mathcal{L}(\mathrm{d}B) \in \mathbb{Z}^m$ which is the d-multiple of an LLL-basis of the lattice $\mathcal{L}(B)$.

*Gram-matrices, symmetric matrices, quadratic forms.* The LLL-algorithm directly extends to real bases $B \in \mathbb{R}^{m\times n}$ of rank $n$ that have an integer Gram-matrix $B^tB \in$

$\mathbb{Z}^{n\times n}$. It transforms $A := B^tB$ into $A' = T^tAT$ such that the GNF $R'$ of $A' = R'^tR'$ is LLL-reduced.

We identify symmetric matrices $A = A^t = [a_{i,j}]_{1\le i,j\le n} \in \mathbb{R}^{n\times n}$ with $n$-ary quadratic forms $\mathbf{x}^tA\mathbf{x} \in \mathbb{R}[x_1,...,x_n]$. The forms $A, A'$ are *equivalent* if $A' = T^tAT$ holds for some $T \in \mathrm{GL}_n(\mathbb{Z})$.

The form $A \in \mathbb{R}^{n\times n}$ with $\det(A) \neq 0$ is *indefinite* if $\mathbf{x}^tA\mathbf{x}$ takes positive and negative values, otherwise $A$ is either *positive* or *negative* (definite). The form $A$ is *regular* if $\det(A) \neq 0$. We call $A = A^t = [a_{i,j}]$ *strongly regular* (s.r.) if $\det([a_{i,j}]_{1\le i,j\le \ell}) \neq 0$ for $\ell = 1,...,n$. Let $D_{\boldsymbol{\sigma}} \in \{0,\pm 1\}^{n\times n}$ denote the diagonal matrix with diagonal $\boldsymbol{\sigma} = (\sigma_1,...,\sigma_n) \in \{\pm 1\}^n$. An easy proof shows

**Proposition 1.** *Every s.r. form $A = A^t \in \mathbb{R}^{n\times n}$ has a unique decomposition $A = R^tD_{\boldsymbol{\sigma}}R \in \mathbb{R}^{n\times n}$ with a GNF $R \in \mathbb{R}^{n\times n}$ and a diagonal matrix $D_{\boldsymbol{\sigma}}$ with diagonal $\boldsymbol{\sigma} \in \{\pm 1\}^n$.*

We call $R$ the geometric normal form (GNF) of $A$, $R = \mathrm{GNF}(A)$. The *signature* $\#\{i \,|\, \sigma_i = 1\}$ is invariant under equivalence. The form $A = R^tD_{\boldsymbol{\sigma}}R$ is positive, resp., negative if all entries $\sigma_i$ of $\boldsymbol{\sigma}$ are $+1$, resp., $-1$. The form is indefinite if $\boldsymbol{\sigma}$ has $-1$ and $+1$ entries.

**Definition 1.** A s.r. form $A = RD_{\boldsymbol{\sigma}}R$ is an LLL-form if its GNF $R$ is LLL-reduced.

The LLL-form $A = [a_{i,j}] = B^tB$ of an LLL-basis $B = (\mathbf{b}_1,...,\mathbf{b}_n)$ satisfies the classical bound $a_{1,1}^2 = \|\mathbf{b}_1\|^2 \le \alpha^{\frac{n-1}{2}}|\det(A)|^{\frac{2}{n}}$ of Theorem 1.

If $B \in \mathbb{R}^{m\times n}$ generates a lattice $\mathcal{L} = \{B\mathbf{x} \,|\, \mathbf{x} \in \mathbb{Z}\}$ of dimension $n' \le n$ the LLL-algorithm transforms the input basis $B$ into an output basis $[\mathbf{0},...,\mathbf{0},\mathbf{b}'_1,...,\mathbf{b}'_{n'}] \in \mathbb{R}^{m\times n}$ such that $[\mathbf{b}'_1,...,\mathbf{b}'_{n'}]$ is an LLL-basis. This generalizes to

**Corollary 1.** *An adjusted LLL transforms an input form $A \in \mathbb{Z}^{n\times n}$ of rank $n'$ into an equivalent regular form $A' \in \mathbb{Z}^{n'\times n'}$ with $\det(A') \neq 0$.*

LLL-reduction easily extends to s.r. forms $A \in \mathbb{Z}^{n\times n}$: compute $R = \mathrm{GNF}(A)$, LLL-reduce $R$ into $RT$ and transform $A$ into the equivalent form $A' = T^tAT$. By Lemma 1 the LLL-reduction of non s.r. indefinite forms reduces to LLL-reduction in dimension $n-2$, see also Simon [72].

**Lemma 1.** *An adjusted LLL-algorithm transforms a non s.r. input form $A \in \mathbb{Z}^{n\times n}$ in polynomial time into an equivalent $A' = [a'_{i,j}]$ such that $a'_{1,i} = a'_{2,j} = 0$ for all $i \neq 2, j \ge 3$ and $0 \le a'_{2,2} \le 2a'_{1,2}$. Such $A'$ is a direct sum $\begin{bmatrix} 0 & a'_{1,2} \\ a'_{1,2} & a'_{2,2} \end{bmatrix} \oplus [a'_{i,j}]_{3\le i,j\le n}$. Moreover $a'_{1,2} = 1$ if $\det(A) \neq 0$ is square-free.*

**Proof.** If $\det([a_{i,j}]_{1\le i,j\le \ell}) = 0$ for some $\ell \le n$ then the LLL-algorithm achieves in polynomial time that $a_{1,1} = 0$. The corresponding $A$ can be transformed into $A'$ such that $a'_{1,2} = \gcd(a_{1,2},...,a_{1,n})$, $a'_{1,3} = \cdots = a'_{1,n} = 0 = a'_{2,3} = \cdots = a'_{2,n}$. Moreover $a'_{2,2}$ can be reduced modulo $2a'_{1,2}$. Doing all transforms symmetrically on rows and

columns the transformed $A' = T^t AT$ is symmetric $a'_{1,2} = a'_{2,1}$, and thus $(a'_{1,2})^2$ divides $\det(A)$. If $\det(A)$ is square-free then $|a'_{1,2}| = 1$ and $a'_{1,2} = 1$ is easy to achieve. □

Lemma 1 shows that the LLL-algorithm can be adjusted to satisfy

**Theorem 4.** *An adjusted LLL-algorithm transforms a given form $A \in \mathbb{Z}^{n\times n}$ in polynomial time into a direct sum $\oplus_{i=1}^{k} A^{(i)}$ of an LLL-form $A^{(k)}$ and binary forms $A^{(i)} = \begin{bmatrix} 0 & a_i \\ a_i & b_i \end{bmatrix}$ for $i = 1, ..., k-1$, where $0 \le b_i < 2a_i$ and $a_i = 1$ if $\det(A) \neq 0$ is square-free. If $A$ is positive definite and $\det A \neq 0$ then $k = 1$.*

LLL-reduction of indefinite forms is used in cryptographic schemes based hard problems of indefinite forms. In particular, the equivalence problem is NP-hard for ternary indefinite forms [29]. [30] presents public key identification and signatures based on the equivalence problem of quadratic forms.

## 4 LLL Using Householder Orthogonalization (HO)

In practice $\mathrm{GNF}(B) = R$ is computed in *fpa* within **LLL** whereas $B$ is transformed in exact integer arithmetic. For better accuracy we replace steps 2.1, 2.2 of **LLL** by the procedure $\texttt{TriCol}_l$ below, denoting the resulting LLL-algorithm by $\mathbf{LLL}_H$. $\texttt{TriCol}_l$ performs HO via *reflections*, these isometric transforms preserve the length of vectors and of error vectors. $\mathbf{LLL}_H$ improves the accuracy of the LLL of [66]. All subsequent reduction algorithms are based on $\mathbf{LLL}_H$. We streamline the $\mathbf{LLL}_H$-analysis of [70].

*Computing the GNF of $B$.* Numerical algorithms for computing the GNF $R = [r_{i,j}] \in \mathbb{R}^{n\times n}$ of $B = QR$ have been well analyzed, see [34] chapter 19. It is known for quite some time that HO is very stable. Wilkinson showed that the computation of a Householder vector and the transform of a given matrix by a reflection are both normwise stable in the sense that the computed Householder vector is very close to the exact one and the computed transform is the update of a tiny normwise perturbation of the original matrix. Wilkinson also showed that the $QR$ factorization algorithmn is normwise backwards stable [76] pp.153–162, p. 236. For a componentwise and normwise error analysis see [34] ch. 19.3.

To simplify the analysis let all basis vectors $\mathbf{b}_j$ start with $n$ zero entries $\mathbf{b}_j = (0^n, \mathbf{b}'_j) \in 0^n\mathbb{Z}^{m-n}$. The bases $\mathbf{b}'_1, ..., \mathbf{b}'_n$ and $\mathbf{b}_1, ..., \mathbf{b}_n$ have the same GNF. The padding with initial zeros increases `TriCol`'s number of steps by the factor $\frac{4}{3}$, it is not required in practice. We compute an orthogonal matrix $Q' \in \mathbb{R}^{m\times m}$ that extends $Q \in \mathbb{R}^{m\times n}$ by $m-n$ columns and a matrix $R' \in \mathbb{R}^{m\times n}$ that extends $R \in \mathbb{R}^{n\times n}$ by final zero rows. In ideal arithmetic we get $R'$ by a sequence of Householder reflections $Q_j \in \mathbb{R}^{m\times m}$ as

$$\begin{aligned} R'_0 &:= B, & R'_j &:= Q_j R'_{j-1} \quad \text{for } j = 1, ..., n, \\ R' &:= R'_n, & Q' &:= Q_1 \cdots Q_n = Q_1^t \cdots Q_n^t, \end{aligned}$$

where $Q_j := I_m - 2\|\mathbf{h}_j\|^{-2}\mathbf{h}_j\mathbf{h}_j^t$ is orthogonal and symmetric and $\mathbf{h}_j \in \mathbb{R}^m$.

The transform $R'_j \mapsto Q_j R'_{j-1}$ zeroes the entries in positions $j+1$ through $m$ of $\mathrm{col}(j, R'_{j-1})$, it *triangulates* $\mathbf{r} := (r_1, ..., r_m)^t := \mathrm{col}(j, R'_{j-1})$ so that $R'_j \in \mathbb{R}^{m\times n}$ is upper-triangular for the first $j$ columns. The reflection $Q_j$ reflects about the hyperplane $\mathrm{span}(\mathbf{h}_j)^\perp$:

$$Q_j\mathbf{h}_j = -\mathbf{h}_j, \qquad Q_j\mathbf{x} = \mathbf{x} \ \text{ for } \langle \mathbf{h}_j, \mathbf{x}\rangle = 0.$$

Note that $(r_j, ..., r_n)^t = 0^{n-j+1}$ due to $\mathbf{b}_1, ..., \mathbf{b}_j \in 0^n\mathbb{Z}^{m-n}$.
We set $r_{j,j} := (\sum_{i=j}^m r_i^2)^{1/2}$, $\mathbf{h}_j := (0^{j-1}, -r_{j,j}, 0^{n-j}, r_{n+1}, ..., r_m)^t$.

*Correctness of* $\mathbf{h}_j$. We have $2\langle \mathbf{h}_j, \mathbf{r}\rangle\|\mathbf{h}_j\|^{-2} = 1$ and $\|\mathbf{h}_j\|^2 = 2r_{j,j}^2$ and thus

$$Q_j\mathbf{r} = \mathbf{r} - \mathbf{h}_j = (r_1, ..., r_{j-1}, r_{j,j}, 0^{m-j})^t \in \mathbb{R}^m.$$

Hence $Q_j\mathbf{r}$ is correctly triangulated and the *Householder vector* $\mathbf{h}_j$ is well chosen.

`TriCol` $(\mathbf{b}_1, ..., \mathbf{b}_l, \mathbf{h}_1, ..., \mathbf{h}_{l-1}, \mathbf{r}_1, ..., \mathbf{r}_{l-1})$ (`TriCol`$_l$ for short)
\# `TriCol`$_l$ *computes* $\mathbf{h}_l$ *and* $\mathbf{r}_l := \mathrm{col}(l, R)$ *and size-reduces* $\mathbf{b}_l, \mathbf{r}_l$.
1. $\mathbf{r}_{0,l} := \mathbf{b}_l$, `FOR` $j = 1, ..., l-1$ `DO` $\mathbf{r}_{j,l} := \mathbf{r}_{j-1,l} - \langle \mathbf{h}_j, \mathbf{r}_{j-1,l}\rangle \mathbf{h}_j / r_{j,j}^2$.
2. $(r_1, ..., r_m)^t := \mathbf{r}_{l-1,l}$, $\zeta := \max_i |r_i|$, $r_{l,l} := \zeta(\sum_{i=n+1}^m (r_i/\zeta)^2)^{1/2}$,
   \# $\zeta$ *prevents under/overflow;* $r_i = 0$ *holds for* $i = l, ..., n$
3. $\mathbf{h}_l := (0^{l-1}, -r_{l,l}, 0^{n-l}, r_{n+1}, ..., r_m)^t$, \# *note that* $\|\mathbf{h}_l\|^2 = 2r_{l,l}^2$.
4. $\mathbf{r}_l := (r_1, ..., r_{l-1}, r_{l,l}, 0^{m-l})^t \in \mathbb{R}^m$.
5. \# *size-reduce* $\mathbf{b}_l$ *and* $\mathbf{r}_l$: `FOR` $i = l-1, ..., 1$ `DO`
   `IF` $|r_{i,l}/r_{i,i}| \le \frac{1}{2} + \varepsilon$
   `THEN` $\mu_i := 0$ `ELSE` $\mu_i := \lceil r_{i,l}/r_{i,i} \rfloor$, $\mathbf{b}_l := \mathbf{b}_l - \mu_i\mathbf{b}_i$, $\mathbf{r}_l := \mathbf{r}_l - \mu_i\mathbf{r}_i$.
6. `IF` $\sum_{i=1}^{l-1} |\mu_i| \neq 0$ `THEN` `GO TO 1` `ELSE output` $\mathbf{b}_l, \mathbf{r}_l, \mathbf{h}_l$.

`TriCol`$_l$ *under* fpa *with* $t$ *precision bits.* Zeroing $\mu_i$ in case $|r_{i,l}/r_{i,i}| <\le \frac{1}{2} + \varepsilon$ in step 5 cancels a size-reduction step and prevents cycling through steps 1-6. In `TriCol`$_l$'s last round size-reduction is void. Zeroing $\mu_i$, the use of $\zeta$ and the loop through steps 1– 6 are designed for *fpa*. The $\mu_i$ in steps 5, 6 consist of the leading $\Theta(t)$ bits of the $\mu_{l,i}$. `TriCol`$_l$ replaces in the SCHNORR-EUCHNER-LLL [66] classical Gram-Schmidt by an economic, modified Gram-Schmidt, where $\mathrm{col}(l, R)$ gets merely transformed by the $l-1$ actual reflections.

*Accurate floating point summation.* Compute the scalar product $\langle \mathbf{h}_j, \mathbf{r}_{j-1,l}\rangle = \langle \mathbf{x}, \mathbf{y}\rangle$ in step 1 by summing up positive and negative terms $x_iy_i$ separately, both in increasing order to $\sum_{>0}$ and $\sum_{<0}$. The increasing order minimizes the apriori forward error [34] chapter 4, see also [19]. If both $\sum_{>0}$ and $-\sum_{<0}$ are larger than $2^{-t/2}$ and nearly opposite, $|\sum_{>0} + \sum_{<0}| < 2^{-t/2}(\sum_{>0} - \sum_{<0})$, then compute $\langle \mathbf{x}, \mathbf{y}\rangle$ exactly. This provision is far more economic than that of [66] to compute $\langle \mathbf{x}, \mathbf{y}\rangle$ exactly if $|\langle \mathbf{x}, \mathbf{y}\rangle| < 2^{t/2}\|\mathbf{x}\|\|\mathbf{y}\|$. Prop. 2 and Thm. 5 do not require occasional exact computations of $\langle \mathbf{x}, \mathbf{y}\rangle$.

*Efficiency.* `TriCol`$_l$ performs $4ml + \frac{3}{2}l^2 + O(l)$ arithmetic steps and one sqrt per round, the $\frac{3}{2}l^2$ steps cover step 1. `TriCol`$_l$ is more economic than the full modified GSO of [34] as only the reflections of the current Householder vectors $\mathbf{h}_1, ..., \mathbf{h}_{l-1}$ are applied to $\mathbf{b}_l$. The contribution of step 1 to the overall costs is negligible for reasonably

long input bases since on average $l \leq n/2$ and there are no long integer steps. $\texttt{TriCol}_l$ performs at most $\log_2(2\|\mathbf{b}_l\|/2^t)$ rounds, each round shortens $\mathbf{b}_l$ by a factor $\leq 1/2^{t-1}$.

*fpa-Heuristics.* There is room for heuristics in speeding up the LLL since the correctness of the output can most likely be efficiently verified [75]. We want to catch the typical behaviour of *fpa*-errors knowing that worst case error bounds are to pessimistic. Note that error vectors $\mathbf{x} - \bar{\mathbf{x}}$ are in high dimension rarely near parallel to $\mathbf{x}$. Small errors $\|\mathbf{x} - \bar{\mathbf{x}}\| \leq \varepsilon\|\mathbf{x}\|$ with expected value $\mathbf{E}[\langle \mathbf{x}, \bar{\mathbf{x}} - \mathbf{x}\rangle] = 0$ satisfy $\mathbf{E}[\,\|\bar{\mathbf{x}}\|^2] = \mathbf{E}[\,\|\mathbf{x} + (\bar{x} - \mathbf{x})\|^2] = \mathbf{E}[\,\|\mathbf{x}\|^2 + \|\bar{\mathbf{x}} - \mathbf{x}\|^2] \leq (1 + \varepsilon^2)\mathbf{E}[\,\|\mathbf{x}\|^2]$. Hence the relative error $|\,\|\bar{\mathbf{x}}\| - \|\mathbf{x}\|\,|/\|\mathbf{x}\|$ is for $\varepsilon \ll 1$ on average smaller than $\varepsilon^2$ and can be neglected.

We let the projection $\pi'_n : \mathbb{R}^m \to \mathbb{R}^{m-n}$ remove the first $n$ coordinates so that $\pi'_n(\mathbf{b}_j) = \mathbf{b}'_j$ for $\mathbf{b}_j = (0^n, \mathbf{b}'_j)$. Recall that $\texttt{TriCol}_l$ computes in step 2 $\mathbf{r}_{l-1,l} = \prod_{j=1}^{l-1} Q_j \mathbf{b}_l$, as $\bar{\mathbf{r}}_{0,l} := \mathbf{b}_l$, $\bar{\mathbf{r}}_{j,l} := fl(\bar{Q}_j \bar{\mathbf{r}}_{j-1,l})$ for $1 \leq j < l$ and $\bar{Q}_j = I_m - \|\bar{\mathbf{h}}_j\|^{-2} \bar{\mathbf{h}}_j \bar{\mathbf{h}}_j^t$.

**Proposition 2. [fpa-Heur.]** $\texttt{TriCol}_l$ *applied to an LLL-basis* $\mathbf{b}_1, ..., \mathbf{b}_l \in 0^n\mathbb{Z}^{m-n}$ *approximates the GNF* $R = [r_{i,j}] = [\mathbf{r}_1, ..., \mathbf{r}_l]$ *such that for* $j = 0, ..., l-1$

1. $|\bar{r}_{j+1,l} - r_{j+1,l}| \leq \|\pi'_n(\bar{\mathbf{r}}_{j,l} - \mathbf{r}_{j,l})\| \leq 10\, m\, (\frac{3}{2})^{j-1} \max_{1 \leq i \leq l} r_{i,i}\, 2^{-t}$,
2. $\mathbf{E}[\|\pi'_n(\bar{\mathbf{r}}_{j,l} - \mathbf{r}_{j,l})\|] \leq 10\, m (\frac{5}{4})^{(j-1)/2} \max_{1 \leq i \leq l} r_{i,i}\, 2^{-t}$ *holds for random* fpa*-error vectors.*

LLL-bases $\mathbf{b}_1, ..., \mathbf{b}_l$ satisfy $r_{i,i}^2 \leq \alpha^{l-i} r_{l,l}^2$ and $\|\mathbf{b}_l\| \leq \alpha^{\frac{l-1}{2}} r_{l,l}$.Clause 1. of Prop. 2 shows for $j = i - 1$ that $\texttt{TriCol}_l$ achieves for $i = 1, ..., l$

$$|\bar{\mu}_{l,i} - \mu_{l,i}| \approx |\bar{r}_{i,l} - r_{i,l}|/r_{i,i} \leq 10\, m\, \alpha^{\frac{l-1}{2}} (\tfrac{3}{2})^{i-2} \tfrac{r_{l,l}}{r_{i,i}} 2^{-t}. \tag{1}$$

This bound can easily be adjusted to the case that merely $\mathbf{b}_1, ..., \mathbf{b}_{l-1}$ is LLL-reduced. Thus it covers for $i = l - 1$ the critical situation of swapping $\mathbf{b}_{l-1}, \mathbf{b}_l$ within $\mathbf{LLL}_H$. It guarantees correct swapping of $\mathbf{b}_{l-1}, \mathbf{b}_l$ if $2^t \geq 100\, m \sqrt{3}^{l-1}$, since $\frac{3}{2}\sqrt{\alpha} \approx \sqrt{3}$ holds for $\alpha \approx \frac{4}{3}$. On average the error bounds are much smaller for random *fpa*-error vectors. Clause 2. of Prop.2 reduces $\frac{3}{2}$ in (1) on average to $(\frac{5}{4})^{1/2} \approx 1.12$. Moreover, the constant $\alpha$ in (1) reduces considerably by stronger lattice reduction. Sections 3-7 show that $\alpha$ can be decreased by feasible lattice reduction to about 1.025. This reduces our correctness condition $2^t \geq 100\, m \sqrt{3}^{l-1}$, for LLL-swapping of $\mathbf{b}_{l-1}, \mathbf{b}_l$ to $2^t \geq 100\, m\, 1.132^{l-1}$.

**Proof of Prop. 2.** *Induction on j;* $j = 0$: Consider the last round of $\texttt{TriCol}_l$ where size-reduction in step 5 is void. So let $\mathbf{b}_l$ and $\mathbf{r}_l$ be size-reduced. We have that $\mathbf{r}_{0,l} = \mathbf{b}_l$, $\mathbf{r}_{1,l} = \mathbf{b}_l - \langle \mathbf{h}_1, \mathbf{b}_l \rangle \mathbf{h}_1 / r_{1,1}^2$, $\mathbf{h}_1 = (-r_{1,1}, 0^{n-1}, \mathbf{b}'_1)$, $r_{1,1} = \|\mathbf{b}_1\|$. A lengthy proof shows $\|\bar{\mathbf{r}}_{1,l} - \mathbf{r}_{1,l}\|/\|\mathbf{r}_{1,l}\| \leq (\frac{d}{2} + 3)2^{-t} + O(2^{-2t})$, see [44] pp. 84, 85, (15.21). We disregard all $O(2^{-2t})$-terms. The claim holds for $j = 0$ and arbitrary $l$ since the size-reduced $\mathbf{b}_l$ satisfies $\|\mathbf{b}_l\| = \|\mathbf{r}_{1,l}\| \leq (\sum_{1 \leq i \leq l} r_{i,i}^2)^{1/2}$. The constant factor 10 in the claim is a crude upper bound.

*Induction step* $j-1 \to j$: Clearly $\mathbf{r}_{j,l} = Q_j \mathbf{r}_{j-1,l}$, $\|\mathbf{h}_j\|^2 = 2r_{j,j}^2$, $\|\pi'_n(\mathbf{r}_{j-1,j})\| = r_{j,j}$, $\pi'_n(\mathbf{h}_j) = \pi'_n(\mathbf{r}_{j-1,j})$, the $j$-th entries of $\mathbf{h}_j, \mathbf{r}_{j,l}$ are $-r_{j,j}, r_{j,l}$. Hence

$$r_{j,l} = \langle \mathbf{h}_j, \pi'_n(\mathbf{r}_{j-1,l})\rangle / r_{j,j}, \tag{2}$$

$$\mathbf{r}_{j,l} - \mathbf{r}_{j-1,l} = -\langle \mathbf{h}_j, \pi_n(\mathbf{r}_{j-1,l})\rangle \, \mathbf{h}_j / r_{j,j}^2 = -r_{j,l} \, \mathbf{h}_j / r_{j,j},$$

$$\pi'_n(\mathbf{r}_{j,l}) = \pi'_n(\mathbf{r}_{j-1,l} - \tfrac{r_{j,l}}{r_{j,j}} \mathbf{r}_{j-1,j}). \tag{3}$$

Consider the part of $\|\pi'_n(\bar{\mathbf{r}}_{j,l} - \mathbf{r}_{j,l})\|$ induced via (5) from $|\bar{r}_{j,l} - r_{j,l}|$. We neglect *fpa*-errors from $\bar{\mathbf{r}}_{j-1,l} \mapsto fl(\bar{Q}_j \bar{\mathbf{r}}_{j-1,l}) = \bar{\mathbf{r}}_{j,l}$ as they are minor. (2) shows that

$$\bar{r}_{j,l} - r_{j,l} r_{j,j} = \langle \bar{\mathbf{h}}_j, \pi'_n(\bar{\mathbf{r}}_{j-1,l})\rangle - \langle \mathbf{h}_j, \pi'_n(\mathbf{r}_{j-1,l})\rangle$$
$$= \langle \bar{\mathbf{h}}_j - \mathbf{h}_j, \pi'_n(\mathbf{r}_{j-1,l})\rangle + \langle \bar{\mathbf{h}}_j, \pi'_n(\bar{\mathbf{r}}_{j-1,l} - \mathbf{r}_{j-1,l})\rangle.$$

The contribution of $\langle \bar{\mathbf{h}}_j, \pi'_n(\bar{\mathbf{r}}_{j-1,l} - \mathbf{r}_{j-1,l})\rangle$ via $\bar{r}_{j,l} - r_{j,l}$ to $\|\pi'_n(\bar{\mathbf{r}}_{j,l} - \mathbf{r}_{j,l})\|$ is part of $\bar{Q}_j \pi'_n(\mathbf{r}_{j-1,l} - \bar{\mathbf{r}}_{j-1,l})$. As the orthogonal $\bar{Q}_j$ preserves the length of error vectors that contribution is covered by $\|\pi'_n(\bar{\mathbf{r}}_{j-1,l} - \mathbf{r}_{j-1,l})\|$. We neglect the contribution of $\langle \bar{\mathbf{h}}_j - \mathbf{h}_j, \pi'_n(\mathbf{r}_{j-1,l})\rangle$ to $\|\pi'_n(\bar{\mathbf{r}}_{j,l} - \mathbf{r}_{j,l})\|$ assuming that the error $\bar{\mathbf{h}}_j - \mathbf{h}_j$ is random. Therefore, (3) shows up to minor errors that

$$\pi'_n(\bar{\mathbf{r}}_{j,l} - \mathbf{r}_{j,l}) \approx \pi'_n(\bar{\mathbf{r}}_{j-1,l} - \mathbf{r}_{j-1,l}) + \tfrac{r_{j,l}}{r_{j,j}} \pi'_n(\bar{\mathbf{r}}_{j-1,j} - \mathbf{r}_{j-1,j}). \tag{4}$$

Applying the induction hypothesis for $j-1$ and size-reducedness, $|r_{j,l}/r_{j,j}| \le \frac{1}{2}$, we get the second part of the induction claim:

$$\|\pi'_n(\bar{\mathbf{r}}_{j,l} - \mathbf{r}_{j,l})\| \le 10\, m (\tfrac{3}{2})^{j-2} (\tfrac{3}{2}) \max_{1 \le i \le l} r_{i,i} \, 2^{-t}.$$

The first part $|\bar{r}_{j+1,l} - r_{j+1,l}| \le \|\pi'_n(\bar{\mathbf{r}}_{j,l} - \mathbf{r}_{j,l}\|$ holds since $r_{j+1,l}$ is an entry of $\mathbf{r}_{j+1,l} = Q_j \mathbf{r}_{j,l}$. On the average the error bounds are much smaller since independent random error vectors are with high probability nearly orthogonal. Thus (4) shows

$$\mathbf{E}[\|\pi'_n(\bar{\mathbf{r}}_{j,l} - \mathbf{r}_{j,l})\|^2] \approx \mathbf{E}[\|\pi'_n(\bar{\mathbf{r}}_{j-1,l} - \mathbf{r}_{j-1,l})\|^2] + \mathbf{E}[\,|\tfrac{r_{j,l}}{r_{j,j}}|^2 \|\pi'_n(\bar{\mathbf{r}}_{j-1,j} - \mathbf{r}_{j-1,j})\|^2].$$

This proves by induction on $j$ clause 2 of Prop. 2. □

We set $\delta := 0.98$, $\delta_- := 0.97$, $\delta_+ := 0.99$, $\alpha := 1/0.73 < 1.37$, $\rho := \frac{3}{2}\sqrt{\alpha} \approx \sqrt{3}$, $\varepsilon := 0.01$ and $\alpha_\varepsilon := (1+\varepsilon^2\alpha)/(\frac{3}{4} - 4\varepsilon - \varepsilon^2/4 - (1+2\varepsilon)\varepsilon\alpha^{1/2}) < 1.44$.

Recall that $\mathbf{LLL}_H$ is obtained by replacing steps 2.1, .2.2 of **LLL** by $\texttt{TriCol}_l$; let $M = \max(d_1, ..., d_n, 2^n)$ for the input basis and $M_0 = \max(\|\mathbf{b}_1\|, ..., \|\mathbf{b}_n\|)$.

**Theorem 5. [Thm. 2 of [70] using fpa-Heur.]** $\mathbf{LLL}_H$ *transforms with* fpa *of precision* $2^t \ge 2^{10}\, m\, \rho^n$ *a basis* $\mathbf{b}_1, ..., \mathbf{b}_n \in \mathbb{Z}^m$ *into an approximate LLL-basis satisfying*

1. $|\mu_{j,i}| < \frac{1}{2} + \varepsilon \alpha_\varepsilon^{\frac{j-i}{2}} r_{j,j}/r_{i,i} + \varepsilon$ *for* $1 \le i < j \le n$,
2. $\delta_- r_{i,i}^2 \le \mu_{i+1,i}^2 r_{i,i}^2 + r_{i+1,i+1}^2$ *for* $i = 1, \ldots, n-1$.
3. *Clauses 1.-3. of Theorem 1 hold with* $\alpha$ *replaced by* $\alpha_\varepsilon$.

$\mathbf{LLL}_H$ *runs in* $O(n^2 m \log_{1/\delta} M)$ *arithmetic steps using* $n + \log_2 M_0$ *bit integers and* fpa *numbers.*

We call a basis *size-reduced under* fpa if it satisfies clause 1 of Theorem 5. The term $\varepsilon \alpha_\varepsilon^{\frac{j-i}{2}} r_{j,j}/r_{i,i} \ge \varepsilon$ covers the error $|\bar{\mu}_{j,i} - \mu_{j,i}|$ from (1).

In particular $\mathbf{LLL}_H$ runs for $M_0 = 2^{O(n)}$, $m = O(n)$ in $O(n^5)$ arithmetic steps and in $O(n^6)$ / $O(n^{5+\varepsilon})$ bit operations under school-/ FFT-multiplication.

Similar to $L^2$ of [56] $\mathbf{LLL}_H$ should be quadratic in $\log_2 M_0$ since size-reduction in step 5 of $\texttt{TriCol}_l$ is done in *fpa* using the $t$ most significant bits of $r_{i,l}$, $r_{l,l}$, $\mu_{l,i}$. Thus at least one argument of multiplication/division has bit length $t$. The maximal bit length of the $\mu_{l,i}$ decreases in practice by $\Theta(t)$ per round of $\texttt{TriCol}_l$, e.g. by about 30 for $t = 53$, see Fig. 1 [41]. In our experience $\mathbf{LLL}_H$ is most likely correct up to dimension $n = 350$ under *fpa* with $t = 53$ precision bits for *arbitrary* $M_0$, and not just for $t \geq \Omega(n + \log_2 M_0)$ as shown in Theorem 5. $\mathbf{LLL}_H$ computes minimal, near zero errors for the LLL-bases given in [56], where the *fpa*-LLL-code of NTL fails in dim. 55, NTL orthogonalizes the Gram-matrix by classical GSO and $\mathbf{LLL}_H$ by HO.

*Givens rotations* zero a single entry of $\mathrm{col}(j, R_{j-1})$ and provide a slightly better *fpa*-error bound than HO [34] chapter 18.5. Givens rotations have been used in parallel LLL-algorithms of Heckler, Thiele and Joux.

The $L^2$ of Nguyen, Stehlé [56] uses GSO in *fpa* for the Gram matrix $B^tB$. Theorem 2 of [56] essentially replaces $\sqrt{3}$ by 3 in our condition $2^t \geq 100\, m\sqrt{3}^{l-1}$ for correct LLL-swapping. $\mathbf{LLL}_H$ performs correct LLL-swaps in about twice the dimension compared to the $L^2$ of [56]. This is because replacing the basis $B$ by the Gram matrix $B^tB$ squares the matrix condition number $\mathcal{K}(B) = \|B\|\|B^{-1}\|$ which characterizes the sensitivity of $Q, R$ to small perturbations of $B$, see [34] chapter 18.8. As *fpa*-errors during the $QR$-factorization act similar to perturbations of $B$ the squaring $\mathcal{K}(B^tB) = O(\mathcal{K}(B)^2)$ essentially halves the accurate bits for $R, Q$ and the dimension for which $L^2$ is correct. The squaring also doubles the bit length of $B$ and more than doubles the running time. While [74] reports that $L^2$ is most likely correct with 53 precision bits up to dimension 170, $\mathbf{LLL}_H$ is most likely correct up to dimension 350. The strength of $L^2$ is its proven accuracy, moreover $L^2$ is quadratic in the bit length $\log_2 M_0$. While the proven accuracy bound of [56] is rather modest for 53 precision bits the Magma-code of $L^2$ uses intermediary heuristics and provides proven accuracy of the output via multiprecision *fpa* [74].

*Scaled LLL-reduction.* Scaling is a useful concept of numerical analysis for reducing *fpa*-errors. Scaled LLL-reduction of [41] associates with a given lattice basis a scaled basis of a sublattice of the given lattice. The scaled basis satisfies $\frac{1}{2} \leq |r_{1,1}^2/r_{j,j}^2| \leq 2$ for all $j$. Scaled LLL-reduction performs a relaxed size-reduction, reducing relative to an associated scaled basis. The relaxed size-reduction is very accurate, independent of *fpa*-errors, and its relaxation is negligible. Scaled LLL-reduction is useful in dimension $n > 350$ where $\mathbf{LLL}_H$ becomes inaccurate. That way we reduced in 2002 a lattice bases of dimension 1000 consisting of integers of bit length 400 in 10 hours on a 800 MHz PC.

*Comparison with [65] and the modular LLL of [73].* Theorem 7 improves the time bound and the accuracy of the theoretic method of [S88] which uses approximate rational arithmetic.

The modular LLL [73] performs $O(nm \log_{1/\delta} M)$ arithmetic steps on integers of bit length $\log_2(M_0 M)$ using standard matrix multiplication, where $M$ denotes $\max(d_1, ..., d_n, 2^n)$. If $M_0 = 2^{\Omega(n)}$, the LLL's of [65], [73] match asymptotically the bound for the number of bit operations of $\mathbf{LLL}_H$. Neither of these LLL's is quadratic in $M_0$. The practicability of $\mathbf{LLL}_H$ rests on the use of small integers of bit length $1.11\,n + \log_2 M_0$ whereas [73] uses long integers of bit length $\log_2(M_0 M) = O(n \log M_0)$.

## 5 Semi Block $2k$-Reduction Revisited

*Survey, background and perspectives of sections 5–7.* We survey feasible basis reduction algorithms that decrease $\alpha$ in clauses 1, 2 of Theorem 1 to $\bar{\alpha} < \alpha$ for $n \gg 2$. The factors $\alpha^{\frac{n-1}{2}}$, $\alpha^{n-1}$ of Theorem 1 decrease within polynomial time reduction to $2^{O((n \log\log n)^2/\log n)}$ [64] and combined with [4] to $2^{O(n \log\log n/\log n)}$. In this survey we focus on reductions of $\alpha$ achievable in feasible lattice reduction time. Some reductions are proven by heuristics to be feasible on the average.

For the rest of the paper let $\delta \approx 1$ so that $\alpha \approx 4/3$. LLL-bases approximate $\lambda_1$ up to a factor $\alpha^{\frac{n-1}{2}} \lesssim 1.155^n$. They approximate $\lambda_1$ much better for lattices of *high density* where $\lambda_1^2 \approx \gamma_n (\det \mathcal{L})^{2/n}$, namely up to a factor $\lesssim \alpha^{\frac{n-1}{4}}/\sqrt{\gamma_n} \lesssim 1.075^n$ due to part **1** of Theorem 1. Moreover, [57] reports that $\alpha$ decreases on average to about $1.02^4 \approx 1.08$ for the random lattices of [57].

The constant $\alpha$ can be further decreased within polynomial reduction time by blockwise basis reduction. We compare Schnorr's algorithm for semi block $2k$-reduction [64] and Koy's primal-dual reduction[39] with blocksize $2k$. Both algorithms perform HKZ-reductions in dimension $2k$ and have similar polynomial time bounds. They are feasible for $2k \leq 50$. Koy's algorithm guarantees within the same time bound under known proofs better approximations of the shortest lattice vector. Under reasonable heuristics both algorithms are equally strong and much better than proven in worst-case. We combine primal-dual reduction with Schnorr's random sampling reduction (RSR) to a highly parallel reduction algorithm that is on the average more efficient than previous algorithms. It reduces the approximation factor $(\frac{4}{3})^{n/2}$ guaranteed by the LLL-algorithm on average to $1.025^{n/2}$ using feasible lattice reduction.

Semi block $2k$-reduced bases of [64] satisfy by Theorem 6 the inequalities of Theorem 1 with $\alpha$ replaced by $(\beta_k/\delta)^{1/k}$, for a lattice constant $\beta_k$ such that $\lim_{k\to\infty} \beta_k^{1/k} = 1$. The best known bounds on $\beta_k$ are $k/12 < \beta_k < (1 + \frac{k}{2})^{2\ln 2 + 1/k}$ [23]. Primal-dual reduction (**Alg. 3**) replaces $\alpha$ by $(\alpha \gamma_{2k}^2)^{1/2k}$ (Theorem 7). The second bound outperforms the first, unless $\beta_k$ is close to its lower bound $k/12$. Primal-dual reduction for blocks of length 48 replaces $\alpha$ in Theorem 1 within feasible reduction time by $(\alpha \gamma_{48}^2)^{1/48} \approx 1.084$. The algorithms **Alg. 2, Alg. 3** for semi block $2k$ reduction and

primal-dual reduction are equally powerful in approximating $\lambda_1$ under the **GSA**-heuristic of [69]. They perform under **GSA** much better than proven in worst case.

Section 7 surveys some basis reduction algorithms that are effcient on average but not proven polynomial time. BKZ-reduction of [66] runs in practice for blocksize 10 in less than twice the LLL-time. The LLL with the *deep insertion* step of [66] seems to be polynomial time on the average and greatly improves the approximational power of the LLL. Based on experiments [57] reports that LLL with deep insertions decreases $\alpha$ for random lattices on average to $1.012^4 \approx 1.05 \approx \alpha^{1/6}$.

In section 7 we replace HKZ-reduction within primal-dual reduction by *random sampling reduction* (RSR) of [69], a parallel extension of the deep insertion step of [66]. RSR is nearly feasible up to blocksize $k = 80$. The new algorithm, *primal-dual RSR* (**Alg. 4**) replaces under the worst-case **GSA**-heuristics $\alpha$ in Theorem 1 by $(80/11)^{1/80} \approx 1.025$. **Alg. 4** is highly parallel and polynomial time on the average but not proven polynomial time.

For Table 1 we assume that the densest known lattice packings $P_{48p}, P_{48q}$ in dimension 48 [16] table 1.3, have nearly maximal density, then $\gamma_{48} \approx 6.01$. For the assumptions **GSA**, **RA** see section 6. **GSA** is a worst case heuristics in the sense that bases $B = QR$ having a large spread of the values $r^2_{i+1,i+1}/r^2_{i,i}$ are in general easier to reduce.

| | $\bar{\alpha}$ |
|---|---|
| 1. Semi block $2k$-reduction [64], $k = 24$ | |
| proven [23] | $(\beta_{24}/\delta)^{1/24} < 1.165$ |
| by heuristic, **GSA** | $\gamma_{47}^{1/47} \approx 1.039$ |
| 2. Primal-dual reduction, Koy 2004, $k = 48$ | |
| proven | $(\alpha\gamma_{48}^2)^{1/48} \approx 1.084$ |
| by heuristic, **GSA** | $\gamma_{48}^{1/47} \approx 1.039$ |
| 3. Primal-dual RSR, $k = 80$, under **GSA, RA** | 1.025 |
| 4. LLL on the average for random lattices, | |
| experimental [57]: | 1.08 |
| 5. LLL with deep insertion [66] on the average for random lattices, | |
| experimental [57], [7]: | $1.012^4 \approx 1.05$ |

**Table 1: Reductions $\bar{\alpha}$ of $\alpha \approx \frac{4}{3}$ in Thm. 1 under feasible lattice basis reduction for $n \gg 2$.**

*Notation.* For a basis $B = QR \in \mathbb{R}^{m\times n}$, $R = [r_{i,j}]_{1\le i,j\le n}$ with $n = hk$ let

$$R_\ell := [r_{i,j}]_{k\ell-k<i,j\le k\ell} \in \mathbb{R}^{k\times k} \text{ for } \ell \le h$$

$$R_{\ell,\ell+1} = [r_{i,j}]_{k\ell-k<i,j\le k\ell+k} = \begin{bmatrix} R_\ell & R'_\ell \\ O & R_{\ell+1} \end{bmatrix} \in \mathbb{R}^{2k\times 2k} \text{ for } \ell < h$$

denote the principal submatrices of the GNF $R$ corresponding to the segments $B_\ell = [\mathbf{b}_{k\ell-k+1}, ..., \mathbf{b}_{k\ell}]$ and $[B_\ell, B_{\ell+1}]$ of $B$. We denote

$$\mathcal{D}_\ell =_{def} (\det R_\ell)^2 = d_{k\ell}/d_{k\ell-k},$$
$$\mathcal{D}_k^{(1)} = \mathcal{D} =_{def} \prod_{\ell=1}^{h-1} d_{k\ell} = \prod_{\ell=1}^{h-1} \mathcal{D}_\ell^{h-\ell}.$$

*The lattice constant* $\beta_k$. Let $\beta_k =_{def} \max(\det R_1/\det R_2)^{1/k}$ maximized over all HKZ-reduced GNF's $R = R_{1,2} = \begin{bmatrix} R_1 & R_1' \\ O & R_2 \end{bmatrix} \in \mathbb{R}^{2k\times 2k}$.

Note that $\beta_1 = \max r_{1,1}^2/r_{2,2}^2$ over all GNF's $R = \begin{bmatrix} r_{1,1} & r_{1,2} \\ 0 & r_{2,2} \end{bmatrix} \in \mathbb{R}^{2\times 2}$ satisfying $|r_{1,2}| \le r_{1,1}/2$, $r_{1,1}^2 \le r_{1,2}^2 + r_{2,2}^2$ and thus $\beta_1 = \frac{4}{3} = \alpha$ holds for $\delta = 1$, $\eta = \frac{1}{2}$. Note that $\beta_k \le (1 + \frac{k}{2})^{2\ln 2 + 1/k}$ [23].

**Definition 2. [64]** A basis $B = QR \in \mathbb{R}^{m\times n}$, $n = hk$, is *semi block* $2k$*-reduced* for $\delta \in (\eta^2, 1]$ and $\alpha = 1/(\delta - \eta^2)$ if the *GNF* $R = [r_{i,j}]$ satisfies

1. $R_1, ..., R_h \subset R$ are HKZ-reduced,
2. $r_{k\ell,k\ell}^2 \le \alpha\, r_{k\ell+1,k\ell+1}^2$ for $\ell = 1, ..., h-1$,
3. $\delta^k\, \mathcal{D}_\ell \le \beta_k^k\, \mathcal{D}_{\ell+1}$ for $\ell = 1, ..., h-1$.

In [64] $\alpha$ in clause 2 has been set to 2 and $\delta^k$ in clause 3 has been set to $\frac{3}{4}$. For $k = 1$, clause 3 means that $r_{\ell,\ell}^2 \le \alpha r_{\ell+1,\ell+1}^2$. LLL-bases for $\delta$ are semi block $2k$-reduced for $k = 1$.

**Theorem 6. [64].** *A semi block* $2k$*-reduced basis* $B = QR \in \mathbb{R}^{m\times n}$, $n = hk$, *satisfies*

$$\|\mathbf{b}_1\|^2 \le \gamma_k (\beta_k/\delta)^{\frac{n/k-1}{2}} (\det \mathcal{L}(B))^{2/n}.$$

**Proof.** We have that $\|\mathbf{b}_1\|^2 \le \gamma_k \mathcal{D}_1^{1/k} = \gamma_k (\det R_1)^{2/k}$ since $R_1$ is HKZ-reduced. Clause 3 of Def. 3 shows $\mathcal{D}_\ell^{1/k} \le (\beta_k/\delta)\mathcal{D}_{\ell+1}^{1/k}$ and yields

$$\|\mathbf{b}_1\|^2 \le \gamma_k \mathcal{D}_1^{1/k} \le \gamma_k (\beta_k/\delta)^{\ell-1} \mathcal{D}_\ell^{1/k} \quad \text{for } \ell = 1, ..., h = n/k.$$

Multiplying these inequalities and taking $h$-th roots yields the claim. □

Moreover, $\|\mathbf{b}_1\|^2 \lambda_1^{-2} \le k^{\ln k + 2}\, (\beta_k/\delta)^{n/k-2}$ holds for $k \ge 3$ [64] Thm 3.1, Cor. 3.5. Ajtai [3] proved these bounds to be optimal up to a constant factor in the exponent. There exist semi block $2k$-reduced bases of arbitrary dimension $n$ satisfying $\|\mathbf{b}_1\|^2 \ge \gamma_k (\beta_k/\delta)^{\Omega(n/k)} (\det \mathcal{L}(B))^{2/n}$.

**Alg. 2** rephrases the algorithm of [64] without using the unknown constant $\beta_k$. For $k = 1$ **Alg. 2** essentially coincides with LLL-reduction.

The transforms $T$ of $R_{\ell,\ell+1}$ that perform LLL-, resp. HKZ-reduction are only transported to the basis $B$ if this decreases $\mathcal{D}_\ell$ by the factor $\delta$, resp. $\delta^{k/2}$.

**Alg. 2: Semi block $2k$-reduction**

INPUT basis $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n] \in \mathbb{Z}^{m \times n}$, $\delta \in [\eta^2, 1), \eta = \frac{1}{2} + \varepsilon$, $n = hk$.

OUTPUT semi block $2k$-reduced basis $B$.

1. LLL-reduce $B$, HKZ-reduce $[B_1, B_2] = [\mathbf{b}_1, ..., \mathbf{b}_{2k}]$, compute $R = [r_{i,j}] \in \mathbb{R}^{n \times n}$, $\ell := 2$.
2. HKZ-reduce $R_{\ell+1}$ into $R_{\ell+1} T'$ for some $T' \in \mathrm{GL}_k(\mathbb{Z})$, $B_{\ell+1} := B_{\ell+1} T'$,
   LLL-reduce $R_{\ell,\ell+1}$ into $R_{\ell,\ell+1} T$.
   If an LLL-swap bridging $R_\ell$ and $R_{\ell+1}$ occured THEN
   $[B_\ell, B_{\ell+1}] := [B_\ell, B_{\ell+1}]\, T$, $\ell := \max(\ell - 1, 1)$ GO TO 2
   HKZ-reduce $R_{\ell,\ell+1}$ into $R_{\ell,\ell+1} T$ for some $T \in \mathrm{GL}_{2k}(\mathbb{Z})$.
3. Compute $\mathcal{D}_\ell^{new} := (\det R_\ell^{new})^2$ for $\begin{bmatrix} R_\ell^{new} & R_\ell'^{new} \\ O & R_{\ell+1}^{new} \end{bmatrix} := \mathrm{GNF}(R_{\ell,\ell+1} T)$,
   IF $\mathcal{D}_\ell^{new} \leq \delta^{k/2}\, \mathcal{D}_\ell$ THEN $[B_\ell, B_{\ell+1}] := [B_\ell, B_{\ell+1}]\, T$, recomputr $R_\ell, R_{\ell+1}$, $\ell := \ell - 1$
   ELSE $\ell := \ell + 1$.
4. IF $\ell = 1$ THEN GO TO 1, IF $1 < \ell < h$ THEN GO TO 2 ELSE terminate.

*Correctness.* Induction over the rounds of the algorithm shows that the basis $\mathbf{b}_1, ..., \mathbf{b}_{k\ell}$ is always semi block $2k$-reduced for the current $\ell$. We show that clauses 2, 3 of Def. 2 hold, clause 1 obviously holds.

*Clause 2* : LLL-reduction of $R_{\ell,\ell+1}$ in step 2 guarantees clause 2. In particular if an LLL-swap bridging $R_\ell$, $R_{\ell+1}$ occured the blocks $B_\ell$, $B_{\ell+1}$ get transformed.

*Clause 3* : After HKZ-reduction of $R_{\ell,\ell+1}$ we have that $\mathcal{D}_\ell^{new} \leq \beta_k^k\, \mathcal{D}_{\ell+1}^{new}$. Before increasing $\ell$ we have $\mathcal{D}_\ell^{new} > \delta^{k/2}\, \mathcal{D}_\ell$ and thus $\mathcal{D}_{\ell+1}^{new} < \delta^{-k/2}\, \mathcal{D}_{\ell+1}$ resulting in $\delta^{k/2}\, \mathcal{D}_\ell < \mathcal{D}_\ell^{new} \leq \beta_k^k\, \mathcal{D}_{\ell+1}^{new} < \delta^{-k/2} \beta_k^k \mathcal{D}_{\ell+1}$, and thus $\delta^k\, \mathcal{D}_\ell < \beta_k^k\, \mathcal{D}_{\ell+1}$. □

**Lemma 2.** *Semi block $2k$-reduction performs at most $h - 1 + 2n(h-1)\log_{1/\delta} M_0$ rounds, i.e., passes of step 2.*

**Proof.** Let $k \geq 2$. Semi block $2k$-reduction iteratively decreases $\mathcal{D}_\ell$ either by LLL-reduction or by HKZ-reduction of $R_{\ell,\ell+1}$ Each pass of steps 2, 3 either decreases $\mathcal{D}_\ell$ and $\mathcal{D} = \prod_{\ell=1}^{h-1} \mathcal{D}_\ell^{h-\ell}$ by the factor $\delta$, resp. $\delta^{k/2}$ or else increments $\ell$. Since initially $\mathcal{D} = \prod_{\ell=1}^{h-1} d_{k\ell} \leq M_0^{2k\binom{h}{2}} = M_0^{n(h-1)}$ the integer $\mathcal{D}$ can be decreased at most $2n(h-1)\log_{1/\delta} M_0$ times by the factor $\delta$. Hence there are at most $n(h-1)\log_{1/\delta} M_0$ passes of steps 2, 3 that decrease $\mathcal{D}$ by the factor $\delta$, resp. $\delta^{k/2}$ and at most $h - 1 + n(h-1)\log_{1/\delta} M_0$ passes of step 2 that do not change $\mathcal{D}$ but increment $\ell$ in step 3. □

The proof of [64] Theorem 3.2 shows that a HKZ-reduction of $R_{\ell,\ell+1}$ performs $O(n^2 k + k^4 \log M_0) + (2k)^{k+o(k)}$ arithmetic steps using integers of bit length $O(n \log M_0)$. Following [S87], semi block $2k$-reduction performs $O((n^4 + n^2 (2k)^{k+o(k)}) \log_{1/\delta} M_0)$ arithmetic steps.

## 6 Primal-Dual Reduction

Koy's primal-dual reduction [39] decreases $\mathcal{D}_\ell = (\det R_\ell)^2$ as follows. It maximizes $r_{k\ell,k\ell}$ over the GNF's of $R_\ell\, T_\ell$ and minimizes $r_{k\ell+1,k\ell+1}$ over the GNF's of $R_{\ell+1}\, T_{\ell+1}$ for all $T_\ell$, $T_{\ell+1} \in \mathrm{GL}_k(\mathbb{Z})$ and then swaps $\mathbf{b}_{k\ell}, \mathbf{b}_{k\ell+1}$ if this decreases $r_{k\ell,k\ell}$ and $\mathcal{D}_\ell$. Primal-dual reduction with double blocksize $2k$ replaces the constant $\beta_k/\delta$ in Theorem 6 by $\sqrt{\alpha}\,\gamma_{2k}$ which is better understood than $\beta_k/\delta$ since $\gamma_k = \Theta(k)$.

*Dual lattice and dual basis.* The *dual* of lattice $\mathcal{L} = \mathcal{L}(QR)$ is the lattice

$$\mathcal{L}^* = \{\mathbf{z} \in \mathrm{span}(\mathcal{L}) \,|\, \mathbf{z}^t\mathbf{y} \in \mathbb{Z} \text{ for all } \mathbf{y} \in \mathcal{L}\}.$$

$\mathcal{L}^* = \mathcal{L}(QR^{-t})$ holds because $(QR^{-t})^tQR = R^{-1}Q^tQR = R^{-1}R = I_n$. $QR^{-t} = B^{-t}$ holds for $m = n$.

$R^{-t}$ is a lower triangular matrix and $U_nR^{-t}U_n$ is upper-triangular with positive diagonal entries. Clearly $\mathcal{L}^* = \mathcal{L}(QR^{-t}U_n)$, the basis $QR^{-t}U_n$ has $QR$-decomposition $QR^{-t}U_n = (QU_n)(U_nR^{-t}U_n)$ because $QU_n$ is isometric and $U_nR^{-t}U_n$ is upper-triangular. $B^* := QR^{-t}U_n$ is the (reversed) *dual basis* of $B = QR$. Note that $(B^*)^* = B$. $B^*$ has the *dual GNF* $R^* := U_nR^{-t}U_n$. The (reversed) dual basis $B^* = [\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*]$ of $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ is characterized by

$$\langle \mathbf{b}_i^*, \mathbf{b}_{n-j+1} \rangle = \delta_{i,j} = \langle \mathbf{b}_i, \mathbf{b}_{n-j+1}^* \rangle,$$

where $\delta_{i,j} \in \{0, 1\}$ is 1 iff $i = j$. The dual basis $B^*$ satisfies $B^* = [\mathbf{b}_1^*, ..., \mathbf{b}_n^*] = B^{-t}$ for $m = n$. (The $\mathbf{b}_i^*$ denote the dual basis vectors and not the orthogonal vectors $\mathbf{q}_i = \pi_i(\mathbf{b}_i)$ as in [47]. The diagonal entries of $R = [r_{i,j}]$ and $R^* = [r_{i,j}^*]$ satisfy

$$r_{i,i} = 1/r_{n-i+1,n-i+1}^* \quad \text{for} \quad i = 1, \ldots, n. \tag{5}$$

HKZ-reduction of $R^*$ minimizes $r_{1,1}^* = \|\mathbf{b}_1^*\|$ and maximizes $r_{n,n} = 1/r_{1,1}^*$.

*Notation.* For a basis $B = QR \in \mathbb{R}^{m\times n}$ we let $\bar{r}_{k\ell,k\ell}$ for $k\ell \leq n$ denote the maximum of $\widetilde{r}_{k\ell,k\ell}$ over the GNF's $[\widetilde{r}_{i,j}]_{k\ell-k<i,j\leq k\ell} = \mathrm{GNF}(R_\ell T)$ for all $T \in \mathrm{GL}_k(\mathbb{Z})$. Shortly, $\bar{r}_{k\ell,k\ell}$ is the maximum of $r_{k\ell,k\ell}$ over the transforms of $R_\ell \subset R$. If $R_\ell^* = U_kR_\ell^{-t}U_k$ is HKZ-reduced then $r_{k\ell,k\ell} = \bar{r}_{k\ell,k\ell}$. We compute $\bar{r}_{k\ell,k\ell}$ by HKZ-reducing $R_\ell^*$ into $R_\ell^*\, T$, then $[\widetilde{r}_{i,j}] := \mathrm{GNF}(R_\ell U_k T^{-t})$ satisfies $\bar{r}_{k\ell,k\ell} = \widetilde{r}_{k\ell,k\ell}$.

**Definition 3.** A basis $B = QR \in \mathbb{R}^{m\times n}$, $n = hk$ is a *primal-dual basis* for $k$ and $\delta \in (\eta^2, 1]$, $\alpha = 1/(\delta - \frac{1}{4})$ if its *GNF* $R = [r_{i,j}]$ satisfies

1. $R_1, ..., R_h \subset R$ are HKZ-reduced,
2. $\bar{r}_{k\ell,k\ell}^2 \leq \alpha\, r_{k\ell+1,k\ell+1}^2$ for $\ell = 1, ..., h-1$.

We see from (5) that clause 2 of Def. 3 also holds for the dual $B^*$ of a primal-dual basis $B$. Therefore, such $B^*$ can be transformed into a primal-dual basis by HKZ-reducing $B_\ell^* = [\mathbf{b}_{k\ell+1}^*, ..., \mathbf{b}_{k\ell+k}^*]$ into $B_\ell^* T_\ell$ for $\ell = 1, ..., h$. Moreover, clauses 2 and 3 of Def. 1 are preserved under duality, they hold for the dual of a semi block $2k$-reduced basis. Theorem 7 replaces $\alpha$ in Theorem 1 by $(\alpha\gamma_k^2)^{1/k}$.

**Theorem 7. [39], [23].** *A primal-dual basis $B = QR \in \mathbb{R}^{m\times n}$, $n = hk$ of the lattice $\mathcal{L}$ satisfies* **1.** $\|\mathbf{b}_1\|^2 \le \gamma_k(\alpha\gamma_k^2)^{\frac{h-1}{2}}(\det\mathcal{L})^{2/n}$, **2.** $\|\mathbf{b}_1\|^2 \le (\alpha\gamma_k^2)^{h-1}\lambda_1^2$.

**Proof. 1.** The maximum $\bar{r}_{k\ell,k\ell}^2$ of $r_{k\ell,k\ell}^2$ over the $R_\ell T$ satisfies by clause 2 of Def. 3 $\bar{r}_{k\ell,k\ell}^2 \le \alpha\, r_{k\ell+1,k\ell+1}^2$.

Moreover we have $\mathcal{D}_\ell^{1/k} \le \gamma_k \bar{r}_{k\ell,k\ell}^2 = \gamma_k/\lambda_1^2(\mathcal{L}(R_\ell^*))$

since $\bar{r}_{k\ell,k\ell}^2$ is computed by HKZ-reduction of $R_\ell^*$, and

$$\lambda_1^2(\mathcal{L}(R_{\ell+1})) = r_{k\ell+1,k\ell+1}^2 \le \gamma_k \mathcal{D}_{\ell+1}^{1/k}$$

since $R_{\ell+1}$ is HKZ-reduced. Combining these inequalities we get

$$\mathcal{D}_\ell^{1/k} \le \gamma_k \bar{r}_{k\ell,k\ell}^2 \le \alpha\gamma_k r_{k\ell+1,k\ell+1}^2 \le \alpha\gamma_k^2 \mathcal{D}_{\ell+1}^{1/k}. \tag{6}$$

Since $R_1$ is HKZ-reduced this yields :

$$\|\mathbf{b}_1\|^2 \le \gamma_k \mathcal{D}_1^{1/k} \le \gamma_k(\alpha\gamma_k^2)^{\ell-1}\mathcal{D}_\ell^{1/k} \quad \text{for } \ell = 1, ..., h.$$

Multiplying these $h$ inequalities and taking $h$-th roots yields the claim.

**2.** Note that the inequality (6) also holds for the dual basis $B^*$, i.e., $(\mathcal{D}_\ell^*)^{1/k} \le \alpha\gamma_k^2(\mathcal{D}_{\ell+1}^*)^{1/k}$ holds for $\mathcal{D}_\ell^* = (\det R_\ell^*)^2 = \mathcal{D}_{h-\ell+1}^{-1}$. Hence the dual **1*.** of part **1.** of Thm 7 also holds:

$$\textbf{1*.} \quad \bar{r}_{n,n}^2 \ge \gamma_k^{-1}(\alpha\gamma_k^2)^{\frac{-h+1}{2}}(\det\mathcal{L})^{2/n}.$$

**1.** and 1.* yield $\|\mathbf{b}_1\|^2 \le \gamma_k^2(\alpha\gamma_k^2)^{h-1}\bar{r}_{n,n}^2$. By clause 2 of Def. 3 we get

$$\|\mathbf{b}_1\|^2 \le \gamma_k^2(\alpha\gamma_k^2)^{\ell-1}\bar{r}_{k\ell,k\ell}^2 \le (\alpha\gamma_k^2)^\ell r_{k\ell+1,k\ell+1}^2 \quad \text{for} \quad \ell = 0, ..., h-1.$$

Therefore $r_{k\ell+1,k\ell+1} \le \lambda_1$ yields the claim. In fact $r_{k\ell+1,k\ell+1} \le \|\pi_{k\ell+1}(\mathbf{b})\| \le \lambda_1$ holds if a shortest lattice vector $\mathbf{b} = \sum_{j=1}^n r_j\mathbf{b}_j \ne \mathbf{0}$ satisfies $k\ell < \mu \le k\ell + k$ for $\mu := \max\{j \mid r_j \ne 0\}$, because $R_{\ell+1}$ is HKZ-reduced. □

**Alg. 3: Koy's algorithm for primal-dual reduction**

INPUT basis $[\mathbf{b}_1, ..., \mathbf{b}_n] = B = QR \in \mathbb{Z}^{m\times n}$, $\delta \in ((\frac{1}{2}+\varepsilon)^2, 1)$, $n = hk$.

OUTPUT primal-dual reduced basis $B$ for $k, \delta$.

1. LLL-reduce $B$, HKZ-reduce $B_1 = [\mathbf{b}_1, ..., \mathbf{b}_k]$ and compute $R = \mathrm{GNF}(B)$, $\ell := 1$.
2. *#reduce $R_{\ell,\ell+1}$ by primal and dual HKZ-reduction of blocksize $k$:*
   HKZ-reduce $R_{\ell+1}$ into $R_{\ell+1}T'$, $B_{\ell+1} := B_{\ell+1}T'$.
   HKZ-reduce $R_\ell^* = U_k R_\ell^{-t} U_k$ into $R_\ell^* T_\ell$, set $\bar{T} := \begin{bmatrix} U_k T_\ell^{-t} & O \\ O & I_k \end{bmatrix}$.
   LLL-reduce $R_{\ell,\ell+1}\bar{T}$ into $R_{\ell,\ell+1}T$ with $\delta$.
3. IF an LLL-swap bridging $R_\ell$ and $R_{\ell+1}$ occured THEN
   $[B_\ell, B_{\ell+1}] := [B_\ell, B_{\ell+1}]\,T$, $\ell := \max(\ell-1, 1)$ ELSE $\ell := \ell+1$.
4. IF $\ell < h$ THEN GO TO 2 ELSE output $B$ and terminate.

*Correctness.* Induction over the rounds of the algorithm shows that the basis $\mathbf{b}_1, ..., \mathbf{b}_{k\ell}$ is always a primal-dual basis for the current $\ell$.

The algorithm deviates from semi block $2k$-reduction in the steps 2, 3. Step 2 maximizes $r_{k\ell,k\ell}$ within $R_\ell$ by HKZ-reduction of $R_\ell^*$ and minimizes $r_{k\ell+1,k\ell+1}$ within $R_{\ell+1}$ by HKZ-reduction of $R_{\ell+1}$, and then LLL-reduces $R_{\ell,\ell+1}$. If no LLL-swap bridging $R_\ell$ and $R_{\ell+1}$ occured in step 2 then clause 2 of Def. 2 was previously satisfied for $\ell$.

**Lemma 3.** *Primal-dual reduction performs at most* $h-1+2n(h-1)\log_{1/\delta} M_0$ *passes of step 2.*

**Proof.** Initially we have $\mathcal{D} = \prod_{\ell=1}^{h-1} d_{\ell k} \le M_0^{n(h-1)}$. Steps 2, 3 either decrease $\mathcal{D}_\ell$ by a factor $\delta$ or else increments $\ell$. Thus the proof of Lemma 2 applies. □

*The number of passes of step 2* is in worst case at most $h-1+2n(h-1)\log_{1/\delta} M_0$. The actual number of passes may be smaller but on the average it should be proportional to $h-1+2n(h-1)\log_{1/\delta} M_0$.

*The dual clause 2 of Def. 3* has been strengthened in [24] for arbitrary small $\varepsilon > 0$ to

$$2^+. \quad \bar{r}_{kl+1,kl+1} \le (1+\varepsilon) r_{kl+1,kl+1} \quad \text{for } l = 1, ..., h-1$$

denoting $\bar{r}_{kl+1,kl+1} := \max_T r'_{kl+1,kl+1}$ of $[r'_{i,j}] := \mathrm{GNF}([r_{i,j}]_{kl-k+2 \le i,j \le kl+1} T)$ over all $T \in \mathrm{GL}_k(\mathbb{Z})$.

This improves the inequalities of Theorem 7 to hold with $\alpha\gamma_k^2$ replaced by $\big((1+\varepsilon)\gamma_k\big)^{\frac{2k}{k-1}}$ [24]:

**1.** $\|\mathbf{b}_1\|^2 \le \gamma_k \big((1+\varepsilon)\gamma_k\big)^{\frac{n-k}{k-1}} (\det \mathcal{L})^{2/n}$, **2.** $\|\mathbf{b}_1\| \le \big((1+\varepsilon)\gamma_k\big)^{\frac{n-k}{k-1}} \lambda_1$.

In adjusting **Alg. 3** to the new clause $2^+$ it is crucial that increasing $r_{kl+1,kl+1}$ by a factor $1+\varepsilon$ to satisfy clause $2^+$ decreases $\mathcal{D}_l = \det R_l^2$ by the factor $(1+\varepsilon)^{-2}$ and preserves $\mathcal{D}_l \mathcal{D}_{l+1}$ and all $\mathcal{D}_i$ for $i \ne l, l+1$. The adjusted **Alg. 3** performs at most $h + 2nh \log_{1+\varepsilon} M_0$ HKZ-reductions in dimension $k$.

**Alg. 2** and **Alg. 3** (for blocksize $2k$) have by Lemma 2 and 3 the same time bound, both algorithms do HKZ-reductions in dimension $2k$.

*For double blocksize* $2k$ Theorem 7 shows

**1.** $\|\mathbf{b}_1\|^2 \le \gamma_{2k} (\alpha\gamma_{2k}^2)^{n/4k-1/2} (\det \mathcal{L})^{2/n}$, **2.** $\|\mathbf{b}_1\|^2 \le (\alpha\gamma_{2k}^2)^{n/2k-1} \lambda_1^2$.

These bounds are better than the bounds of Theorem 6 unless $\beta_k$ is close to the lower bound $\beta_k > k/12$ of [23] so that $\beta_k/\delta \le \sqrt{\alpha}\,\gamma_{2k}$. Due to the unknown values $\beta_k$ semi block $2k$-reduction can still be competitive.

*Theorems 6 and 7 under the* **GSA***-heuristics.* We associate the quotients $q_i := r_{i+1,i+1}^2 / r_{i,i}^2$ with the GNF $R = [r_{i,j}] \in \mathbb{R}^{n \times n}$. In practice the $q_i$ are all very close for typical reduced bases. For simplicity we assume the geometric series assumption (**GSA**) of [69]:

**GSA** $\qquad q =_{def} q_1 = q_2 = \cdots = q_{n-1}$.

**GSA** is a worst-case property. Bases that do not satisfy **GSA** are easier to reduce, see [69]. The worst case bases of Ajtai [3] also satisfy **GSA**.

We next show that the bounds of Theorem 6 improve under **GSA** to those of Theorem 7 while the bounds of Theorem 7 are mainly preserved under **GSA**.

Under **GSA** Theorem 1 holds with $\alpha$ replaced by $1/q$. Note that $(\det \mathcal{L})^{2/n} / r_{1,1}^2 = q^{\binom{n}{2}\frac{1}{n}} = q^{\frac{n-1}{2}}$ holds under **GSA** and thus $r_{1,1}^2 = q^{\frac{1-n}{2}} (\det \mathcal{L})^{2/n}$. Hence part **1.**

of Theorem 1 holds with $\alpha$ replaced by $1/q$. Part **2.** also holds due to the duality argument used in the proof of Theorem 7.

HKZ-bases $R_\ell$ satisfy under **GSA**

$$\|\mathbf{b}_1\|^2 = \lambda_1^2(\mathcal{L}(R_\ell)) \le \gamma_k \det \mathcal{L}(R_\ell)^{\frac{2}{k}} = \gamma_k \|\mathbf{b}_1\|^2 q^{\binom{k}{2}\frac{1}{k}} = \gamma_k \|\mathbf{b}_1\|^2 q^{\frac{k-1}{2}}.$$

This shows that $1/q \le \gamma_k^{\frac{2}{k-1}}$ holds under **GSA**. Replacing in Theorem 1 $\alpha$ by $1/q \le \gamma_k^{\frac{2}{k-1}}$ we get

**Corollary 2.** *Primal-dual bases of blocksize $k$ and $n = hk$ satisfy under* **GSA** *the bounds of Theorem 1 with $\alpha$ replaced by $\gamma^{\frac{2}{k-1}}$, in particular $\|\mathbf{b}_1\|^2 \le \gamma_k^{\frac{n-1}{k-1}} (\det \mathcal{L})^{2/n}$, $\|\mathbf{b}_1\|^2 \le \gamma_k^{2\frac{n-1}{k-1}} \lambda_1^2$.*

The bounds of Cor. 2 and those of Thm. 7 nearly coincide. Cor. 2 eliminates $\alpha$ from Thm. 7 and replaces $h = \frac{n}{k}$ by the slightly larger value $\frac{n-1}{k-1}$. Interestingly , the bounds of Cor. 2 coincide for $k = 2$ with clauses 1, 2 of Theorem 1 for LLL-bases with $\delta = 1, \alpha = \frac{4}{3}$ because $\gamma_2^2 = \frac{4}{3}$.

Similarly, $1/q \le \gamma_{2k}^{\frac{2}{2k-1}}$ holds under **GSA** for any HKZ-reduced GNF $R = R_{1,2} \in \mathbb{R}^{2k\times 2k}$. Hence

**Corollary 3.** *Semi block $2k$-reduced bases with $n = hk$ satisfy under* **GSA** *the inequalities of Theorem 1 with $\alpha$ replaced by $\gamma_{2k}^{\frac{2}{2k-1}}$, in particular $\|\mathbf{b}_1\|^2 \le \gamma_{2k}^{\frac{n-1}{2k-1}} (\det \mathcal{L})^{2/n}$, $\|\mathbf{b}_1\|^2 \le \gamma_{2k}^{2\frac{n-1}{2k-1}} \lambda_1^2$.*

Primal-dual bases of blocksize $2k$ and semi block $2k$-reduced bases are by Cor. 2, Cor. 3 nearly equally strong under **GSA**. This suggests that **Alg. 2** for semi block $2k$-reduction can be strengthened by working towards **GSA** (similar to **Alg. 4**) so that **GSA** approximately holds for the output basis.

*Practical bounds for $\|\mathbf{b}_1\|^2\lambda_1^{-2}$.*We compare the bounds of Thm. 6 for $2k = 48$ to those of Thm. 7 for double blocksize 48. Note that $\gamma_{24} = 4$ [15]. Assuming that the densest known lattice packings $P_{48p}, P_{48q}$ in dimension 48 [16] table 1.3, is nearly maximal we have that $\gamma_{48} \approx 6.01$. HKZ-reduction in dimension 48 is nearly feasible. Let $\delta = 0.99$.

*Semi block $2k$-reduction for $k = 24$.* Using $\beta_{24} \le 13^{2\ln 2+1/24}$ Theorem 6 proves $\|\mathbf{b}_1\|^2/(\det \mathcal{L})^{2/n} \le \gamma_{24}(\beta_{24}/\delta)^{n/48-1/2} < \gamma_{24}\, 1.165^{n/2}$. Moreover

$$\|\mathbf{b}_1\|^2/(\det \mathcal{L})^{2/n} \le \gamma_{48}^{\frac{1}{47}\frac{n-1}{2}}$$

holds under **GSA**. This replaces $\alpha$ in Theorem 1 by $\gamma_{48}^{1/47} < 1.039$.

*Primal-dual bases of blocksize* 48 satisfy by Theorem 7

$$\|\mathbf{b}_1\|^2/(\det \mathcal{L})^{2/n} < \gamma_{48}(\alpha\gamma_{48}^2)^{\frac{n/48-1}{2}} \lesssim 1.075^{n/2}/\sqrt{\alpha}.$$

This replaces $\alpha$ in Theorem 1 by $(\alpha\gamma_{48}^2)^{1/48} \approx 1.084$. Moreover,

$$\|\mathbf{b}_1\|^2/(\det \mathcal{L})^{2/n} \leq \gamma_{48}^{\frac{1}{47}\frac{n-1}{2}}$$

holds under **GSA** which replaces $\alpha$ in Theorem 1 by $\gamma_{48}^{1/47} < 1.039$.

While $\gamma_{48}$ is relatively large **Alg. 2** and **3** perform better when $\gamma_k$ is relatively small. This suggests to choose $k$ in practice clearly apart from multiples of 24 since $\gamma_k$ is relatively large for $k = 0 \mod 24$.

## 7 Primal-Dual Random Sampling Reduction

We replace in primal-dual reduction HKZ-reduction by random sampling reduction (RSR) of [69], a method that shortens the first basis vector. RSR extends the deep insertion step of [66] to a highly parallel algorithm. We use local RSR in a way to approximate the **GSA**-property which has been used in the analysis of [69]. Moreover, global RSR breaks the worst-case bases of [Aj03] against semi-block $k$-reduction and those of [24] against primal-dual reduction. These worst-case bases $B = QR$ satisfy $r_{i,j} = 0$ for $j \geq i + k$ which results in a very short last vector $\mathbf{b}_n$. Primal-dual RSR (**ALG. 4**) runs for $k = 80$ in expected feasible time under reasonable heuristics. It reduces $\alpha$ in Theorem 1 to less than $(80/11)^{1/80} \approx 1.025$ which so far is the smallest feasible reduction of $\alpha$ proven under heuristics.

*Notation.*We associate with a basis $B = QR = [\mathbf{b}_1, ...., \mathbf{b}_n] \in \mathbb{R}^{m\times n}$, $R = [r_{i,j}]_{1\leq i,j\leq n}$ the submatrix $R_{\nu,k} := [r_{i,j}]_{\nu<i,j\leq\nu+k} \subset R$ corresponding to $B_{\nu,k} := [\mathbf{b}_{\nu+1}, ..., \mathbf{b}_{\nu+k}] \subset B$. Let $T_{\mathbf{a},k} = \begin{bmatrix} a_1 & 1 & & \\ \vdots & 0 & \ddots & \\ & & \ddots & 1 \\ a_{k-1} & & & \\ 1 & 0 & \cdots & 0 \end{bmatrix}$.

We replace in **Alg. 4.** the HKZ-reductions of $R_\ell, R_\ell^*$ occuring in **Alg. 3** by RSR of suitable $R_{\nu,k}, R_{\nu,k}^*$.

*RSR of $R_{\nu,k}$.* Let $R_{\nu,k} = [r_{i,j}]_{\nu<i,j\leq\nu+k} \subset R = [\mathbf{r}_1, ..., \mathbf{r}_n]$. Enumerate in parallel all vectors $\mathbf{r} := \sum_{j=k/2+1}^{k} a_j\mathbf{r}_{\nu+j} \in \mathcal{L}(R)$ for $(a_{k/2+1}, \ldots, a_k) \in \mathbb{Z}^{k/2}, a_k = 1$ that satisfy $\|\pi_{\nu+k/2+1}(\mathbf{r})\| \leq \eta$ for some $\eta$ having about $(k/11)^{k/4}$ such vectors $\mathbf{r}$. Extend each sum to a short vector $\sum_{j=1}^{k} a_j\mathbf{r}_{\nu+j}$ using $n^2k^{o(k)}$ bit operations for minimization of $\|\sum_{j=1}^{k} a_j\mathbf{r}_{\nu+j}\|$. We can use the additional size-reduction step 2.3 b of section 2 and the Schnorr-Hörner heuristics [68] while full exhaustive search minimization, requiring $k^{k/4+o(k)} \log_2 M_0$ bit operations, is to expensive. RSR extends the SCHNORR-EUCHNER *deep insertion* step [66] of depth $k$. This step coincides with RSR for $a_1, ..., a_{k-1}$ set to zero. RSR tries the zero choice and about $(k/11)^{k/4}$ more instances $(a_1, ..., a_{k-1}) \in \mathbb{Z}^{k-1}$.

*Analysis of RSR on $R_{\nu,k}$.* Under the assumptions
**RA** $r_{\nu+j,\nu+j'} \in_R [-\frac{1}{2}, \frac{1}{2}]$ is random for $j' > j$,

**GSA** $r_{i+1,i+1}/r_{i,i} = q_\nu$ for all $i$, $\nu < i < \nu + k$,

[S03, Thm 1, Remark 2] shows that RSR of $R_{\nu,k}$ and $R^*_{\nu,k}$ succeeds in steps 3, 4 as long as $q_\nu < (11/k)^{1/k}$. Primal-dual RSR uses all indices $\nu = 1, ..., n-k$ in a uniform way, this helps to approximate the **GSA**-property.

**Theorem 8. [69] RA, GSA**. *Primal-dual RSR transforms a basis $B \in \mathbb{R}^{m\times n}$ of $\mathcal{L} = \mathcal{L}(B)$ such that*

**1.** $\|\mathbf{b}_1\|^2 \le (k/11)^{\frac{n-1}{2k}} (\det \mathcal{L})^{2/n}$, **2.** $\|\mathbf{b}_1\|^2 \le (k/11)^{\frac{n-1}{k}} \lambda_1^2$.

Theorem 8 replaces $\alpha$ in Theorem 1 by $(k/11)^{1/k}$ with $(80/11)^{1/80} \approx 1.025$ for $k = 80$.

**Alg. 4. Primal-dual RSR**

`INPUT` basis $B = QR \in \mathbb{Z}^{m\times n}$, $\delta \in [\eta^2, 1)$, $\eta = \frac{1}{2} + \varepsilon$, $k$

`OUTPUT` reduced basis $B$.

1. LLL-reduce $B$ with $\delta$ and compute the GNF $R$ of $B$, $\ell := 0$.
2. IF $\ell = 0 \mod \lfloor n/k \rfloor$ THEN [ BKZ-reduce $B$ into $BT$ with $\delta$ and blocksize 20, compute the new GNF $R$, $\ell := \ell + 1$. ] *# this approximates the* **GSA***-property.*
3. *Primal RSR-step.* Randomly select $0 \le \nu \le n-k$ that nearly maximizes $r_{\nu+1,\nu+1}/|\det R_{\nu,k}|^{1/k}$. Try to decrease $r_{\nu+1,\nu+1}$ by the factor $\delta$ through RSR of $R_{\nu,k} \subset R$, i.e., compute some $T_{\mathbf{a},k}$ in $\mathrm{GL}_k(\mathbb{Z})$ such that the GNF $\widetilde{R}_{\nu,k} = [\widetilde{r}_{i,j}]$ of $R_{\nu,k}T_{\mathbf{a},k}$ satisfies $\widetilde{r}_{\nu+1,\nu+1} \le \delta\, r_{\nu+1,\nu+1}$. Transform $B_{\nu,k} := B_{\nu,k}T_{\mathbf{a},k}$. Recompute $R_{\nu,k}$.
4. *Dual RSR-step.* Randomly select $0 \le \nu \le n-k$ that nearly minimizes $r_{\nu+k,\nu+k}/|\det R_{\nu,k}|^{1/k}$. Try to increase $r_{\nu+k,\nu+k}$ by the factor $1/\delta$ through RSR of the dual GNF $R^*_{\nu,k} = U_k R^{-t}_{\nu,k} U_k$, i.e., compute by RSR of $R^*_{\nu,k} = [r^*_{i,j}]$ some $T_{\mathbf{a},k} \in \mathrm{GL}_k(\mathbb{Z})$ such that the GNF $\widetilde{R}^*_{\nu,k} = [\widetilde{r}^*_{i,j}]$ of $R^*_{\nu,k}T_{\mathbf{a},k}$ satisfies $\widetilde{r}^*_{\nu+1,\nu+1} \le \delta r_{\nu+1,\nu+1}$. *# Hence the GNF $\widetilde{R}_{\nu,k} = [\widetilde{r}_{i,j}]$ of $R_{\nu,k}U_kT^{-t}_{\mathbf{a},k}$ satisfies $\widetilde{r}_{\nu+k,\nu+k} \ge r_{\nu+k,\nu+k}/\delta$.* Transform $B_{\nu,k} := B_{\nu,k}U_kT^{-t}_{\mathbf{a},\mathbf{k}}$ and recompute $R_{\nu,k}$.
5. *Global RSR-step.* Try to decrease $r_{1,1}$ by the factor$1/\delta$ through RSR of $R$, i.e., compute some $T_{\mathbf{a},n}$ in $\mathrm{GL}_n(\mathbb{Z})$ such that the GNF $\widetilde{R} = [\widetilde{r}_{i,j}]$ of $RT_{\mathbf{a},n}$ satisfies $\widetilde{r}_{1,1} \le \delta\, r_{1,1}$. Transform $B := BT_{\mathbf{a},n}$ and recompute $R$.
6. `IF` either of steps 3,4,5 succeeds, or $\ell \ne 0 \mod \lfloor n/k \rfloor$ `THEN GOTO 2` `ELSE` output $B$ and terminate.

*Primal-dual RSR time bound.* RSR succeeds under **RA, GSA** in steps 3 and 4 using $(k/11)^{k/4+o(k)}$ arithm. steps provided that $q_\nu < (11/k)^{1/k}$ [69] Thm 1, ff]. For **RA** see [57] Fig. 4, 5 (Randomness of $r_{i,i+1}$ is irrelevant, $r_{i,i+1}$ is in practice nearly random in $[-\frac{1}{2}, \frac{1}{2}]$ under the condition that $r^2_{i,i} \approx r^2_{i,i+1} + r^2_{i+1,i+1}$, and this improves by deep insertion.) **GSA** is a worst-case assumption, in practice **GSA** is approximately satisfied. Ludwig [51] analyses an approximate version of **GSA**.

On the average one round of **Alg. 4** decreases the integer $\mathcal{D}^{(1)} := \prod_{i=1}^{n-1} d_i$ by the factor $\delta^2$. This bounds the average number of rounds by about $\frac{1}{2}n^2 \log_{1/\delta} M_0$ since

initially $\mathcal{D}^{(1)} \leq M_0^{n(n-1)}$. In worst case however $\mathcal{D}^{(1)}$ can even increase per round and **Alg. 4** must not terminate.

*Compairing* **Alg. 3** *and* **Alg. 4** *for* $k = 80$. We assume that $\gamma_{80} \approx 4 \cdot 2^{40.14/40} \approx 8.02$, i.e., that the MORDELL-WEIL lattice $MW_n$ in dimension $n = 80$ has near maximal density, [16] table 1.3, similarly we assume $\gamma_{400} \approx 24$.

**Alg. 3** would under **GSA** reduce $\alpha$ in Theorem 1 for $k = 80$ to $\gamma_{80}^{1/79} \approx 1.027$ but **Alg. 3** does not work towards **GSA**. Moreover, primal-dual reduction with full HKZ-reduction is infeasible in dimension $k = 80$ requiring $80^{40+o(1)}$ steps, whereas RSR is nearly feasible.

**Alg. 4** for primal-dual RSR reduces $\alpha$ in Theorem 1 to $(80/11)^{1/80} \approx 1.025$ (by Theorem 8) and thus achieves $\|\mathbf{b}_1\|/(\det \mathcal{L})^{1/n} \lesssim 1.025^{\frac{n-1}{4}}$.
For lattices of high density, $\lambda_1 \approx \gamma_n \det(\mathcal{L})^{2/n}$, and $n = 400$, $k = 80$ this yields

$$\|\mathbf{b}_1\|/\lambda_1 \lesssim 1.025^{99.75}/\sqrt{\gamma_{400}} \lesssim 2.4.$$

If this bound further decreases on the average case this might endanger the NTRU schemes for parameters $N \leq 200$. The secret key is a sufficiently short lattice vector of a lattice of dimension $2N$. LUDWIG [51] reports on attacks to NTRU via RSR.

Doing HKZ-reduction via the sieve algorithm of [4] reduces all asymptotic time bounds at the expense of super-polynomial space. [62], [60] improve the AKS-algorithm and its analysis. The experimental comparison in [60] of AKS with the SCHNORR, EUCHNER version [66] of [64] shows that the SE-algorithm with BKZ for block size 20 outperforms the improved AKS for dimension $n \leq 50$.

## 8 Basic Segment LLL

Segment LLL uses an idea of SCHÖNHAGE [71] to do most of LLL-reduction locally in segments of low dimension $k$ using $k$ local coordinates. It guarantees that the determinants of segments do not decrease to fast, see Def. 5. Here we present the basic algorithm $\mathbf{SLLL}_0$. Theorem 12 bounds the number of local LLL-reductions within $\mathbf{SLLL}_0$. Lemma 4 and Corollary 5 bound the norm of and the *fpa*-errors induced by local LLL-transforms. The algorithm $\mathbf{SLLL}_0$ is faster by a factor $n$ in the number of arithmetic steps compared to $\mathbf{LLL}_H$ but uses longer integers and *fpa* numbers, a drawback that will be repaired by **SLLL**.

*Segments and local coordinates.* Let the basis $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n] \in \mathbb{Z}^{m \times n}$ have dimension $n = k\,h$ and GNF $R \in \mathbb{R}^{n \times n}$. We partition $B$ into $m$ *segments* $B_{l,k} = [\mathbf{b}_{lk-k+1}, \ldots, \mathbf{b}_{lk}]$ for $l = 1, ..., h$. Local LLL-reduction of two consecutive segments $B_{l,k}, B_{l+1,k}$ is done in local coordinates of the principal submatrix

$$R_{l,k} := [r_{lk+i,lk+j}]_{-k<i,j\leq k} \in \mathbb{R}^{2k \times 2k}$$

of $R$. Let $H = [\mathbf{h}_1, ..., \mathbf{h}_n] = [h_{i,j}] \in \mathbb{R}^{m \times n}$ be the lower triangular matrix of Householder vectors and $H_{l,k} = [h_{lk+i,lk+j}]_{-k<i,j\leq k} \subset H$ the submatrix for $R_{l,k}$. We control

the calls, and minimize the number, of local LLL-reductions of the $R_{l,k}$ by means of the *local squared determinant* of $B_{l,k}$

$$D_{l,k} =_{\text{def}} \|\mathbf{q}_{lk-k+1}\|^2 \cdots \|\mathbf{q}_{lk}\|^2.$$

We have that $d_{lk} = \|\mathbf{q}_1\|^2 \cdots \|\mathbf{q}_{lk}\|^2 = D_{1,k} \cdots D_{l,k}$. Moreover, we will use

$$\mathcal{D}^{(k)} =_{\text{def}} \prod_{l=1}^{h-1} d_{lk} = \prod_{l=1}^{h-1} D_{l,k}^{h-l},$$
$$M_{l,k} =_{\text{def}} \max_{lk-k<i\le j\le lk+k} \|\mathbf{q}_i\|/\|\mathbf{q}_j\|.$$

For the input basis $B = QR$ we denote $M_1 := \max_{1\le i\le j\le n} \|\mathbf{q_i}\|/\|\mathbf{q}_j\|$. $M_{l,k}$ is the $M_1$-value of $R_{l,k}$ when calling $\texttt{locLLL}(R_{l,k})$, obviously $M_{l,k} \le M_1$. Recall that $M = \max(d_1, ..., d_n, 2^n)$.

**Definition 4.** A basis $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{Z}^m$, $n = kh$, is an *SLLL$_0$-basis (or SLLL$_0$-reduced)* for given $k$, $\delta \ge \eta^2$, $\alpha = 1/(\delta - \frac{3}{4})$ if it is size-reduced and

1. $\delta \|\mathbf{q}_i\|^2 \le \mu_{i+1,i}^2 \|\mathbf{q}_i\|^2 + \|\mathbf{q}_{i+1}\|^2$ for $i \in [1, n-1] \setminus k\mathbb{Z}$,
2. $D_{l,k} \le (\alpha/\delta)^{k^2} D_{l+1,k}$ for $l = 1, \ldots, h-1$.

*Size-reducedness* under *fpa* is defined by clause 1 of Theorem 5. Segment $B_{l,k}$ of an SLLL$_0$-basis is LLL-reduced in the sense that the $k \times k$-submatrix $[r_{lk+i,lk+j}]_{-k<i,j\le 0}$ $\subset R$ is LLL-reduced. Clause 1 does not bridge distinct segments since the $i \in k\mathbb{Z}$ are excepted. Clause 2 relaxes the inequality $D_{l,k} \le \alpha^{k^2} D_{l+1,k}$ of LLL-bases, and this allows to bound the number of local LLL-reductions, see Theorem 12.

We could have used two independent $\delta$-values for the two clauses of Def. 4. Theorem 9 shows that the first vector of an SLLL$_0$-basis of lattice $\mathcal{L}$ is almost as short relative to $(\det \mathcal{L})^{1/n}$ as for LLL-bases.

**Theorem 9. Thm. 3 of [70].** $\|\mathbf{b}_1\| \le (\alpha/\delta)^{\frac{n-1}{4}} (\det \mathcal{L})^{\frac{1}{n}}$ *holds for all SLLL$_0$-bases* $\mathbf{b}_1, ..., \mathbf{b}_n$ .

*The dual of Theorem 9.* Clause 2 of Def. 4 is preserved under duality. If it holds for a basis $\mathbf{b}_1, ..., \mathbf{b}_n$ it also holds for the dual basis $\mathbf{b}_1^*, ..., \mathbf{b}_n^*$ of the lattice $\mathcal{L}^*$. We have that $\|\mathbf{b}_1^*\| = \|\mathbf{q}_n\|^{-1}$ and $\det(\mathcal{L}^*) = (\det \mathcal{L})^{-1}$. Hence, Theorem 9 implies that every SLLL$_0$-basis satisfies $\|\mathbf{q}_n\| \ge (\delta/\alpha)^{\frac{n-1}{4}} (\det \mathcal{L})^{\frac{1}{n}}$.

*Local LLL-reduction.* The procedure $\texttt{locLLL}(R_{l,k})$of [S06] locally LLL-reduces $R_{l,k} \subset R$ given $H_{l,k} \subset H$. Initially it produces a copy $[\mathbf{b}'_1, ..., \mathbf{b}'_{2k}]$ of $R_{l,k}$. It LLL-reduces the local basis $[\mathbf{b}'_1, ..., \mathbf{b}'_{2k}]$ consisting of *fpa*-vectors. It updates and stores the local transform $T_{l,k} \in \mathbb{Z}^{2k\times 2k}$ so that $[\mathbf{b}'_1, ..., \mathbf{b}'_{2k}] = R_{l,k}T_{l,k}$ always holds for the current local basis $[\mathbf{b}'_1, ..., \mathbf{b}'_{2k}]$ and the initial $R_{l,k}$. E.g., it does $\text{col}(l', T_{l,k}) := \text{col}(l', T_{l,k}) - \mu\, \text{col}(i, T_{l,k})$ along with $\mathbf{b}'_{l'} := \mathbf{b}'_{l'} - \mu\, \mathbf{b}'_i$ within $\texttt{TriCol}_l$. It freshly computes $\mathbf{b}'_{l'}$ from the updated $T_{l,k}$. Using a correct $T_{l,k}$ this correction of $\mathbf{b}'_{l'}$ limits *fpa*-errors of the local basis, see Cor. 5. Local LLL-reduction of $R_{l,k}$ is done in local coordinates of dimension $2k$. A local LLL-swap merely requires $O(k^2)$ arithmetic steps, update of

$R_{l,k}$, local triangulation and size-reduction via $\texttt{TriCol}_l$ included, compared to $O(nm)$ arithmetic steps for an LLL-swap in global coordinates.

*$SLLL_0$-algorithm.* $\mathbf{SLLL}_0$ transforms a given basis into an $\text{SLLL}_0$-basis. It iterates $\texttt{locLLL}(R_{l,k})$ for submatrices $R_{l,k} \subset R$, followed by a global update that *transports* $T_{l,k}$ to $B$ and triangulates $B_{l,k}, B_{l,k+1}$ via $\texttt{TriSeg}_{l,k}$. *Transporting* $T_{l,k}$ to $B, R, T_{1,n/2}$ and so on means to multiply the submatrix consisting of $2k$ columns of $B, R, T_{1,n/2}$ corresponding to $R_{l,k}$ from the right by $T_{l,k}$.

$\mathbf{SLLL}_0$
INPUT $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{Z}^d$ (a basis with $M_0, M_1, M$), $k$, $m$, $\delta$
OUTPUT $\mathbf{b}_1, \ldots, \mathbf{b}_n$ $\text{SLLL}_0$-basis for $k$, $\delta$
WHILE $\exists\, l, 1 \le l < m$ such that either $D_{l,k} > (\alpha/\delta)^{k^2} D_{l+1,k}$
or $\texttt{TriSeg}_{l,k}$ has not yet been executed
DO for the minimal such $l$: $\texttt{TriSeg}_{l,k}$, $\texttt{locLLL}(R_{l,k})$
# *global update*: $[B_{l,k}, B_{l+1,k}] := [B_{l,k}, B_{l+1,k}]\, T_{l,k}$, $\texttt{TriSeg}_{l,k}$.

The procedure $\texttt{TriSeg}_{l,k}$ *triangulates* and size-reduces two adjacent segments $B_{l,k}, B_{l+1,k}$. Given $B_{l,k}, B_{l+1,k}$ and $\mathbf{h}_1, ..., \mathbf{h}_{lk-k}$, it computes $[\mathbf{r}_{lk-k+1}, ..., \mathbf{r}_{lk+k}] \subset R$ and $[\mathbf{h}_{lk-k+1}, ..., \mathbf{h}_{lk+k}] \subset H$.

$\texttt{TriSeg}_{l,k}$
1. FOR $l' = lk-k+1, ..., lk+k$ DO $\texttt{TriCol}_{l'}$ (including updates of $T_{l,k}$)
2. $D_{j,k} := \prod_{i=0}^{k-1} r^2_{kj-i,kj-i}$ for $j = l, l+1$.

*Correctness in ideal arithmetic.* All inequalities $D_{l,k} \le (\alpha/\delta)^{k^2} D_{l+1,k}$ hold upon termination of $\mathbf{SLLL}_0$. All segments $B_{l,k}$ are locally LLL-reduced and globally size-reduced and thus the terminal basis is $\text{SLLL}_0$-reduced.

*The number of rounds of* $\mathbf{SLLL}_0$. Let $\#_k$ denote the number of $\texttt{loclll}(R_{l,k})$-executions due to $D_{l,k} > (\alpha/\delta)^{k^2} D_{l+1,k}$ for all $l$. The first $\texttt{loclll}(R_{l,k})$-executions for each $l$ is possibly not counted in $\#_k$, this yields at most $n/k-1$ additional rounds.

$\#_k$ can be bounded by the Lovász volume argument.

**Theorem 10. Thm. 4 of [70].** $\#_k \le 2\, n\, k^{-3} \log_{1/\delta} M$.

All intermediate $M_{l,k}$-values within $\mathbf{SLLL}_0$ are bounded by the $M_1$-value of the input basis of $\mathbf{SLLL}_0$. Consider the local transform $T_{l,k} \in \mathbb{Z}^{2k \times 2k}$ within $\texttt{locLLL}(R_{l,k})$. Let $\|T_{l,k}\|_1$ denote the maximal $\|\ \|_1$-norm of the columns of $T_{l,k}$.

**Lemma 4. [70]** *Within* $\texttt{locLLL}(R_{l,k})$ *we have that* $\|T_{l,k}\|_1 \le 6k(\frac{3}{2})^{2k} M_{l,k}$.

Next consider $\texttt{locLLL}(R_{l,k})$ under *fpa* based on the iterative *fpa*-version of $\texttt{TriCol}_l$. Let $\|[r_{i,j}]\|_F = (\sum_{i,j} r^2_{i,j})^{1/2}$ denote the FROBENIUS *norm.* [S06] shows

**Corollary 4. [fpa-Heur.]**

1. *Within* $\texttt{locLLL}(R_{l,k})$ *the current* $R'_{l,k} := R_{l,k}T_{l,k}$ *and its approximation* $\bar{R}'_{l,k}$ *satisfy* $\|\bar{R}'_{l,k} - R'_{l,k}\|_F \leq \|\bar{R}_{l,k} - R_{l,k}\|_F 2^{2k} M_{l,k} + 7n\|R_{l,k}\|_F 2^{-t}$.
2. *Let* $\texttt{TriSeg}_{l,k}$ *and* $\texttt{locLLL}$ *use* fpa *with precision* $2^t \geq 2^{10} d\,\rho^n M_1^2$. *If* $\bar{R}_{l,k}$ *is computed by* $\texttt{TriSeg}_{l,k}$ *then* $\texttt{locLLL}(\bar{R}_{l,k})$ *computes a correct* $T_{l,k}$ *so that* $R_{l,k}T_{l,k}$ *is LLL-reduced with* $\delta_-$.

**Theorem 11. Thm. 5 of [70] using fpa-Heur.** *Let* $k = \Theta(\sqrt{n})$. *Given a basis with* $M_0, M_1, M$, $\mathbf{SLLL}_0$ *computes under* fpa *with precision* $2^t \geq 2^{10}\, m\, \rho^n M_1^2$ *an* $SLLL_0$*-basis for* $\delta_-$. *It runs in* $O(nm \log_{1/\delta} M)$ *arithmetic steps using* $2n + \log_2(M_0 M_1^2)$*-bit integers.*

$\mathbf{SLLL}_0$ saves a factor $n$ in the number of arithmetic steps compared to $\mathbf{LLL}_H$ but uses longer integers and *fpa* numbers. $\mathbf{SLLL}_0$ runs for $M_0 = 2^{O(n)}$, and thus for $M = 2^{O(n^2)}$, in $O(n^3 m)$ arithmetic steps using $O(n^2)$ bit integers. Algorithm **SLLL** of section 9 reduces the bit length $O(n^2)$ to $O(n)$.

## 9 Gradual SLLL Using Short *fpa*-Numbers

**SLLL** reduces the required precision and the bit length of integers and *fpa* numbers compared to $\mathbf{SLLL}_0$. This results from limilting the norm of local transforms to $O(2^n)$. Theorem 14 shows that **SLLL**-bases are as strong as LLL-bases. For input bases of length $2^{O(n)}$ and $d = O(n)$ **SLLL** performs $O(n^{5+o(1)})$ bit operations compared to $O(n^{6+o(1)})$ bit operations for $\mathbf{LLL}_H$, $\mathbf{SLLL}_0$ and the LLL-algorithms of [65], [73]. The advantage of **SLLL** is the use of small integers which is crucial in practice.

The use of small integers and short intermediate bases within **SLLL** rests on a gradual LLL-type reduction so that all local LLL-transforms $T_{l,2^\sigma}$ of $R_{l,2^\sigma}$ have norm $O(2^n)$. For this we work with segments of all sizes $2^\sigma$ and to perform LLL-reduction on $R_{l,2^\sigma}$ with a measured strength, i.e., SLLL-reduction according to Definition 6. If the submatrices $R_{2l,2^{\sigma-1}}, R_{2l+1,2^{\sigma-1}} \subset R_{l,2^\sigma}$ are already SLLL-reduced then $\texttt{locLLL}(R_{l,k})$ performs a transform $T_{l,2^\sigma}$ bounded as $\|T_{l,2^\sigma}\|_F = O(2^n)$. This is the core of *fpa*-correctness of **SLLL**.

*Comparison with* Schönhage*'s semi-reduction [71].* Semi-reduction also uses segments but proceeds without adjusting LLL-reduction according to Def. 5 and does not satisfy Theorems 11 and 12. While **SLLL** achieves length defect $\|\mathbf{b}_1\|/\lambda_1 \leq (\frac{4}{3}+\varepsilon)^{n/2}$ semi-reduction achieves merely $\|\mathbf{b}_1\|/\lambda_1 \leq 2^n$. **SLLL** is practical even for small $n$, all $O$-constants and $n_0$-values are small.

We let $n$ be a power of 2. We set $s := \lceil \frac{1}{2} \log_2 n \rceil$ so that $\sqrt{n} \leq 2^s < 2\sqrt{n}$.

**Definition 5.** A basis $\mathbf{b}_1, ..., \mathbf{b}_n \in \mathbb{R}^m$ is an *SLLL-basis (or SLLL-reduced)* for $\delta \geq \frac{1}{2}$ if it satisfies for $\sigma = 0, ..., s = \lceil \frac{1}{2} \log_2 n \rceil$ and all $l$, $1 \leq l < n/2^\sigma$*:*

$$D_{l,2^\sigma} \leq \alpha^{4^\sigma} \delta^{-n} D_{l+1,2^\sigma}.$$

If the inequalities of Def. 6 hold for a basis they also hold for the dual basis. Thus the dual of an SLLL-basis is again an SLLL-basis. To preserve SLLL-reducedness by duality we do not require SLLL-bases to be size-reduced.

The inequalities of Def. 6 for $\sigma = 0$ mean that $\|\mathbf{q}_l\|^2 \leq \alpha\delta^{-n}\|\mathbf{q}_{l+1}\|^2$ holds for all $l$. The inequalities of Def. 6 are merely required for $2^\sigma \leq 2\sqrt{n}$. Therefore, **SLLL** locally LLL-reduces $R_{l,2^\sigma}$ via $\texttt{locLLL}(R_{l,2^\sigma})$ merely for segment sizes $2^\sigma < 2\sqrt{n}$, where size-reduction of a vector requires $O(2^{2\sigma}) = O(n)$ arithmetic steps.

The inequalities of Def. 6 and $D_{l,k} \leq (\alpha/\delta)^{k^2} D_{l+1,k}$ of Def. 5 coincide for $k = 2^\sigma$ when setting $\delta := \delta_\sigma$ in Def. 6, and $\delta_\sigma := \delta^{n4^{-\sigma}}$ for the $\delta$ of Def. 6. Note that $\delta_\sigma$ can be arbitrarily small, e.g. $\delta_\sigma \ll \frac{1}{4}$, $\delta_\sigma$ decreases with $\sigma$. In particular for $2^\sigma = k \geq \sqrt{n}$ we have that $\alpha^{4^\sigma}\delta^{-n} \leq (\alpha/\delta)^{k^2}$ and thus the inequalities of Def. 6 are stronger than the ones of Def. 5. Thm 12 shows that the vectors of SLLL-bases approximate the successive minima in nearly the same way as for LLL-bases.

**Theorem 12. Thm. 6 of [70].** *Every size-reduced SLLL-basis satisfies*

**1.** $\lambda_j^2 \leq \alpha^{j-1}\delta^{-7n} r_{j,j}^2$ *for* $j = 1, \ldots, n$,
**2.** $\|\mathbf{b}_l\|^2 \leq \alpha^{j-1}\delta^{-7n} r_{j,j}^2$ *for* $l \leq j$,
**3.** $\|\mathbf{b}_j\|^2 \leq \alpha^{n-1}\delta^{-7n}\lambda_j^2$ *for* $j = 1, \ldots, n$.

$\texttt{LLLSeg}_{l,1}$
\# *Given* $R_{l,1}, \mathbf{b}_1, ..., \mathbf{b}_{l+1}, \mathbf{h}_1, ..., \mathbf{h}_l, \mathbf{r}_1, ..., \mathbf{r}_l$, $\texttt{LLLSeg}_{l,1}$ *LLL-reduces* $R_{l,1}$.
1. IF $r_{l,l}/r_{l+1,l+1} > 2^{n+1}$ THEN [ $R'_{l,1} := R_{l,1}$,
   $\mathrm{row}(2, R'_{l,1}) := \mathrm{row}(2, R'_{l,1})\, 2^{-n-1}\, r_{l,l}/r_{l+1,l+1}$ $\texttt{locLLL}(R'_{l,1})$,
   \# *global update*: $[\mathbf{b}_l, \mathbf{b}_{l+1}] := [\mathbf{b}_l, \mathbf{b}_{l+1}]\, T_{l,1}$, $\texttt{TriCol}_l$, $\texttt{TriCol}_{l+1}$ ]
2. $\texttt{locLLL}(R_{l,1})$.

**SLLL** uses the procedure $\texttt{LLLSeg}_{l,1}$ that breaks $\texttt{locLLL}(R_{l,1})$ up into parts, each with a bounded transform $\|T_{l,1}\|_1 \leq 9 \cdot 2^{n+1}$. This keeps intermediate bases of length $O(4^n M_0)$ and limits *fpa*-errors within $\texttt{LLLSeg}_{l,1}$. $\texttt{LLLSeg}_{l,1}$ LLL-reduces the basis $R_{l,1} = \begin{bmatrix} r_{l,l} & r_{l,l+1} \\ 0 & r_{l+1,l+1} \end{bmatrix} \subset R$ after dilating $\mathrm{row}(2, R_{l,1})$ so that $r_{l,l}/r_{l+1,l+1} \leq 2^{n+1}$. After the LLL-reduction of the dilated $R_{l,1}$ we undo the dilation, by transporting the local transform $T_{l,1} \in \mathbb{Z}^{2\times 2}$ to $B$. $\texttt{LLLSeg}_{l,1}$ includes global updates between local rounds.

$\texttt{LLLSeg}_{l,1}$ performs $O(nm)$ arithmetic steps [70] Le.3. An effectual step 1 decreases $\mathcal{D}^{(1)}$ by a factor $2^{-n/2}$ via a transform $T_{l,1}$ satisfying $\|T_{l,1}\|_1 \leq 9 \cdot 2^{n+1}$.

**SLLL**

INPUT $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{Z}^m$ (a basis with $M_0, M_1, M$), $\delta$, $\alpha$, $\varepsilon$

OUTPUT $\mathbf{b}_1, \ldots, \mathbf{b}_n$ size-reduced SLLL-basis for $\delta$, $\varepsilon$

1. $\texttt{TriCol}_1$, $\texttt{TriCol}_2$, $l' := 2$, $s := \lceil \frac{1}{2} \log_2 n \rceil$
   # $\texttt{TriCol}_{l'}$ *has always been executed for the current* $l'$
2. WHILE $\exists\, \sigma \le s,\ l,\ 2^\sigma(l+1) \le l'$ such that $D_{l,2^\sigma} > \alpha^{4^\sigma} \delta^{-n} D_{l+1,2^\sigma}$
   # *Note that* $r_{1,1}, ..., r_{l',l'}$ *and thus* $D_{l,2^\sigma}$, $D_{l+1,2^\sigma}$ *are given*
   DO for the minimal such $\sigma$ and the minimal $l$:
   IF $\sigma = 0$ THEN $\texttt{LLLSeg}_{l,1}$ ELSE $\texttt{locLLL}(R_{l,2^\sigma})$
   #*global update*: transport $T_{l,2^\sigma}$ to $B$, $\texttt{TriSeg}_{l,2^\sigma}$
3. IF $l' < n$ THEN [ $l' := l' + 1$, $\texttt{TriCol}_{l'}$, GOTO 2. ]

*Correctness in ideal arithmetic.* All inequalities $D_{l,2^\sigma} \le \alpha^{4^\sigma} \delta^{-n} D_{l+1,2^\sigma}$ hold upon termination of **SLLL**. As $\texttt{TriSeg}_{l,2^\sigma}$ results in size-reduced segments $B_{l,2^\sigma}$, $B_{l+1,2^\sigma}$ the terminal basis is size-reduced.

**Theorem 13. Thm. 7 of [70] using fpa-Heur.** *Given a basis with* $M_0, M$, **SLLL** *finds under* fpa *of precision* $t = 3n + O(\log m)$ *an SLLL-basis for* $\delta_-$. *It runs in* $O(nm \log_2 n \log_{1/\delta} M)$ *arithmetic steps using integers of bit length* $2\,n + \log_2 M_0$.

For $M_0 = 2^{O(n)}$ and $m = O(n)$ **SLLL** runs in $O(n^4 \log n)$ arithmetic steps, and in $O(n^{6+\varepsilon})$ / $O(n^{5+\varepsilon})$ bit operations under school-/FFT-multiplication.

*SLLL-bases versus LLL-bases.* LLL-bases with $\delta$ satisfy the inequalities of Theorem 14 with $\delta$ replaced by 1. Thus $\|\mathbf{b}_j\|$ approximates $\lambda_j$ to within a factor $\alpha^{\frac{n-1}{2}}$ for LLL-bases, resp., within a factor $(\alpha/\delta^7)^{\frac{n-1}{2}}$ for SLLL-bases. But SLLL-bases for $\delta' = \delta^{1/8}$ are "better" than LLL-bases for $\delta$, in the sense that they guarantee a smaller length defect, because $\alpha'/\delta'^7 = \frac{1}{\delta'^8 - \delta'^7/4} = \frac{1}{\delta - \delta'^7/4} < \frac{1}{\delta - 1/4} = \alpha$.

*Dependence of time bounds on* $\delta$. The time bounds contain a factor $\log_{1/\delta} 2$,

$$\log_{1/\delta} 2 = \log_2(e)/\ln(1/\delta) \le \log_2(e) \tfrac{\delta}{1-\delta},$$

since $\ln(1/\delta) \ge 1/\delta - 1$. We see that replacing $\delta$ by $\sqrt{\delta}$ essentially halves $1 - \delta$ and doubles the SLLL-time bound. Hence, replacing $\delta$ by $\delta^{1/8}$ increases the **SLLL**-time bound at most by a factor 3. In practice, the LLL-time may increase slower than by the factor $\frac{\delta}{1-\delta}$ as $\delta$ approaches 1, see [41] Fig.3.

*Reducing a generator system.* There is an algorithm that, given a generator matrix $B \in \mathbb{Z}^{m \times n}$ of arbitrary rank $\le n$, transforms $B$ with the performance of **SLLL**, into an SLLL-basis for $\delta_-$ of the lattice generated by the columns of $B$.

**SLLL-Reduction via iterated subsegments. SLLL$^+$** of [70] is a variant of **SLLL** that extends LLL-operations stepwise to larger and larger submatrices $R_{l,2^\sigma} \subset R$ by transporting local transforms from level $\sigma - 1$ to level $\sigma$ recursively for $\sigma = 1, ..., s = \log_2 n$. Local LLL-reduction and the transport of local LLL-transforms is done by a local procedure $\texttt{locSLLL}(R_{l,2^\sigma})$ that recursively executes $\texttt{locSLLL}(R_{l',2^{\sigma-1}})$ for $l' =$

$2l-1, 2l, 2l+1$. **SLLL**$^+$ does not iterate the global procedure `TriSeg` but a faster local one.

**Definition 6.** A basis $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{Z}^m$ with $n = 2^s$ is an *SLLL*$^+$*-basis (or SLLL*$^+$*-reduced)* for $\delta$ if it satisfies for $\sigma = 0, ..., s = \log_2 n$

$$D_{l,2^\sigma} \leq (\alpha/\delta)^{4^\sigma} D_{l+1,2^\sigma} \quad \text{for } \textit{odd } l \in [1, n/2^\sigma[. \tag{7}$$

Unlike to Def. 5 and Def. 6 the inequalities (7) are not required for *even* $l$, this opens new efficiencies for SLLL$^+$-reduction. The inequalities (7) hold for each $\sigma$ and odd $l$ locally in double segments $[B_{l,2^\sigma}, B_{l+1,2^\sigma}]$, they do not bridge these pairwise disjoint double segments. For $\sigma = 0$ the inequalities (7) mean that $\|\mathbf{q}_l\|^2 \leq \alpha/\delta \, \|\mathbf{q}_{l+1}\|^2$ holds for odd $l$.

The inequalities (7) are preserved under duality. If $\mathbf{b}_1, ..., \mathbf{b}_n$ is an SLLL$^+$-basis then so is the dual basis $\mathbf{b}_1^*, ..., \mathbf{b}_n^*$. Theorem 16 extends Theorem 11 and shows that the first vector of an SLLL$^+$-basis is almost as short relative to $(\det \mathcal{L})^{\frac{2}{n}}$ as for LLL-bases.

**Theorem 14. Thm. 8 of [70].** *Every SLLL*$^+$*-basis* $\mathbf{b}_1, \ldots, \mathbf{b}_n$*, where* $n$ *is a power of 2 satisfies* **1.** $\|\mathbf{b}_1\| \leq (\alpha/\delta)^{\frac{n-1}{4}} (\det \mathcal{L})^{\frac{1}{n}}$ **2.** $\|\mathbf{q}_n\| \geq (\delta/\alpha)^{\frac{n-1}{4}} (\det \mathcal{L})^{\frac{1}{n}}$.

**Theorem 15. Thm. 9 of [70].** *In ideal arithmetic algorithm* **SLLL**$^+$ *of [S06] computes a size-reduced SLLL*$^+$*-basis for* $\delta$ *and runs in* $O(n^2 m + n \log_2 n \log_{1/\delta} M)$ *arithmetic steps.*

**SLLL**$^+$ requires $t = O(\log(M_0 M_1)) = O(n \log M_0)$ precision bits to cover the *fpa*-errors that get accumulated by the initial `TriSeg` and by iterating `locTri`. For $M_0 = 2^{O(n)}$ and $m = O(n)$ **SLLL**$^+$ saves a factor $n$ in the number of arithmetic steps compared to **SLLL** but requires $n$-times longer *fpa*-numbers.

**Acknowledgement.** I like to thank P. Nguyen and D. Stehlé for useful comments.

## References

1. *E. Agrell, T. Eriksson, A. Vardy and K. Zeger*, Closest Point Search in Lattices. *IEEE Trans. on Inform. Theory*, **48**(8), pp. 2201–2214, 2002.
2. *M. Ajtai*, The Shortest Vector Problem in $L_2$ is NP-hard for Randomized Reductions. In Proc. 30th STOC, ACM, pp. 10–19, 1998.
3. *M. Ajtai*, The Worst-case Behavior of Schnorr's Algorithm Approximating the Shortest Nonzero Vector in a Lattice. In Proc. 35th STOC, ACM, pp. 396–406, 2003.
4. *M. Ajtai, R. Kumar and D. Sivakumar*, A Sieve Algorithm for the Shortest Lattice Vector Problem. In Proc. 33th STOC, ACM, pp. 601–610, 2001.
5. *A. Akhavi*, Worst Case Complexity of the Optimal LLL. In Proc. of LATIN 2000, LNCS 1776, Springer-Verlag, Berlin NewYork, 2000.
6. *A. Akhavi and D.Stehlé*, Speeding-up Lattice Reduction with Random Projections. In Proc. of LATIN 2008, to be published in LNCS by Springer-Verlag, Berlin New York, 2008.

7. *W. Backes and S. Wetzel*, Heuristicsc on Lattice Reduction in Practice. *Journal of Experimental Algorithm*, ACM **7**(1), 2002.
8. *G. Bergman*, Notes on Ferguson and Forcade's Generalized Euclidean Algorithm. TR. Dep. of Mathematics, University of Berkeley, CA, 1980.
9. *D. Bleichenbacher and A. May*, New Attacks on RSA with Small Secret CRT-Exponents. In Proc. PKC 2006, LNCS 3958, Springer-Verlag, Berlin New York, pp. 1–13, 2006.
10. *J. Blömer and A. May*, New Partial Key Exposure Attacks on RSA. In Proc. Crypto 2003, LNCS 2729, Springer-Verlag, Berlin New York, pp. 27–43, 2003.
11. *J. Blömer and J.P. Seifert*, On the Complexity of Computing Short Linearly Independent Vectors and Short Bases in a Lattice. In Proc. 31th STOC, ACM, pp. 711–720, 1999.
12. *J. Cai*, The Complexity of some Lattice Problems. In Proc. Algorithmic- Number Theory, LNCS 1838, Springer-Verlag, Berlin New York, pp. 1–32, 2000.
13. *J.W.S. Cassels*, Rational Quadratic Forms. London Mathematical Society Monographs 13 , Academic Press Inc., Londen, 1978.
14. *H. Cohen*, A Course in Computational Number Theory, second edition, Springer-Verlag, Berlin New-York, 2001.
15. *H. Cohn and A. Kumar*, Optimality and Uniqueness of the Leech Lattice Among Lattices. arXiv:math.MG/04 03263v1 16 Mar 2004.
16. *J.H. Conway and N.J.A. Sloane*, Sphere Packings, Lattices and Groups. Third edition, Springer-Verlag, Berlin New York, 1998.
17. *D. Coppersmith*, Small Solutions to Polynomial Equations, and Low Exponent RSA Vulnerabilities. *J. Cryptology*, **10**, pp. 233–260, 1997.
18. *D. Coppersmith*, Finding Small Solutions to Small Degree Polynomials. In Proc. CaLC 2001, LNCS 2146, Springer-Verlag, Berlin New York, pp. 20–1, 2001.
19. *J.Demmel and Y. Hida*, Accurate floating point summation. TR University Berkeley, 2002, http://www.cs.berkeley.edu/ demmel/AccurateSummation.ps.
20. *P. van Emde Boas*, Another NP-Complete Partition Problem and the Complexity of Computing Short Vectors in a Lattice. Mathematics Department, University of Amsterdam, TR 81-04, 1981.
21. *U. Fincke and M. Pohst*, A Procedure for Determining Algebraic Integers of a Given Norm. In Proc. EUCAL, LNCS 162, Springer-Verlag, Berlin New York, pp. 194–202, 1983.
22. *C. F. Gauss*, Disquisitiones Arithmeticae. 1801; English transl., Yale Univ. Press, New Haven, Conn. 1966.
23. *N. Gama, N. How-Grave-Graham, H. Koy and P. Nguyen*, Rankin's Constant and Blockwise Lattice Reduction. In Proc. CRYPTO 2006, LNCS 4117, Springer-Verlag, Berlin New York, pp. 112–139, 2006.
24. *N. Gama and P. Q. Nguyen*, Finding short lattice vectors within Mordell's inequality. In Proc. of the 2008 ACM Symposium on Theory of Computing, pp. 207–216, 2008.
25. *O. Goldreich and S. Goldwasser*, On the Limits of Nonapproximability of Lattice Problems. *of Comp. and System Sciences*, **60**(3), pp. 540–563, 2000. Preliminary version in STOC '98.
26. *G. Golub and C. van Loan*, Matrix Computations. John Hopkins University Press, 1996.
27. *M. Grötschel, L. Lovász, and A. Schrijver*, Geometric Algorithms and Combinatorial Optimization, Springer-Verlag, Berlin New-York, 1988.
28. *G. Hanrot and D. Stehlé*, Improved Analysis of Kannan's Shortest Lattice Vector Algorithm. In Proc. CRYPTO 2007, LNCS 4622, Springer-Verlag, Berlin New York, pp. 170–186, 2007.
29. *H.J. Hartung and C.P. Schnorr*, Public Key Identification Based on the Equivalence of Quadratic Forms. In Proc. of Math. Found. of Comp. Sci., Aug. 26 –Aug. 31, Český Krumlov, Czech Republic, LNCS 4708, Springer-Verlag, Berlin New York, pp. 333–345, 2007.
30. *R.J. Hartung and C.P. Schnorr*, Identification and Signatures Based on NP-hard Problems of Indefinite Quadratic Forms. *J. of Math. Crypt.* **2**, pp. 327– 341, 2008. Preprint University Frankfurt, 2007,//www.mi.informatik.uni-frankfurt.de/research/papers.html.
31. *J. Håstad, B. Just, J.C. Lagarias und C.P. Schnorr*, Polynomial Time Algorithms for Finding Integer Relations Among Real Numbers, *SIAM Journal on Computing*, **18**, pp. 859–881, 1989.

32. *B. Helfrich*, Algorithms to Construct Minkowski Reduced and Hermite Reduced Bases. *Theor. Comput. Sci.* **41**, pp. 125–139, 1985.
33. *C. Hermite*, Extraits de lèttres de M. Ch. Hermiteà Jacobi sur differents objects de la théorie des nombres, deuxième lettre, *J. Reine Angewandte Mathematik*, **40**, pp. 279–290, 1850.
34. *N.J. Higham*, Accuracy and Stability of Numerical Algorithms. SIAM, Philadelphia, 2nd edition, 2002.
35. *R. Kannan*, Minkowski's Convex Body Theorem and Integer Programming. *Math. Oper. Res.*, **12**, pp. 415–440, 1987.
36. *S. Khot*, Hardness of Approximating the Shortest Vector Problem in Lattices. In Proc. FOCS, 2004.
37. *D.E. Knuth*, The Art of Computer Programming, Vol. 2, Addison -Wesley, , Boston, third edition, 1997.
38. *A. Korkine und G. Zolotareff*, Sur les Formes Quadratiques, *Mathematische Annalen*, **6**, pp. 366–389, 1873.
39. *H. Koy*, Primal/duale Segment-Reduktion von Gitterbasen, Lecture Universität Franfurt 2000, files from Mai 2004. //www.mi.informatik.uni-frankfurt.de/research/papers.html
40. *H. Koy and C.P. Schnorr*, Segment LLL-Reduction. In Proc. CaLC 2001, LNCS 2146, Springer-Verlag, Berlin New York, pp.67–80, 2001. //www.mi.informatik.uni-frankfurt.de/research/papers.html
41. *H. Koy and C.P. Schnorr*, Segment LLL-Reduction with Floating Point Orthogonalization. In Proc. CaLC 2001, LNCS 2146, Springer-Verlag, Berlin New York, pp. 81–96, 2001. //www.mi.informatik.uni-frankfurt.de/research/papers.html
42. *H. Koy and C.P. Schnorr*, Segment and Strong Segment LLL-Reduction of Lattice Bases. TR Universität Franfurt, April 2002, //www.mi.informatik.uni-frankfurt.de/research/papers.html
43. *J.L.Lagrange*, Recherches d'arithmétique. Nouveaux Mémoires de l'Académie de Berlin, 1773.
44. *C.L. Lawson and R.J. Hanson*, Solving Least Squares Problems. Siam, Philadelphia, 1995.
45. *H.W. Lenstra, Jr.*, Integer Programming With a Fixed Number of Variables. *Math. Oper. Res.* 8, pp. 538-548, 8, 1983.
46. LIDIA, a C++ Library for computational number theory. The LIDIA Group, http://www.informatik.tu-darmstadt.de/TI/LiDIA/
47. *A. K. Lenstra, H. W. Lenstra, Jr and L. Lovász*, Factoring Polynomials with Rational Coefficients. *Math. Ann.*, **261**, pp. 515–534, 1982.
48. *J.C. Lagarias, H.W. Lenstra, Jr and C.P. Schnorr*, Korkin-Zolotarev Bases and Successive Minima of a Lattice and its Reciprocal Lattice. *Combinatorica*, **10**, pp. 333–348, 1990.
49. *L. Lovász*, An Algorithmic Theory of Numbers, Graphs and Convexity, CBMS-NSF Regional Conference Series in Applied Mathematics, **50**, Siam Publications, Philadelphia, 1986.
50. *L. Lovász and H. Scarf*, The Generalized Basis Reduction Algorithm. *Math. of Oper. Research*, **17** (3), pp. 754–764, 1992.
51. *C. Ludwig*, Practical Lattice Basis Reduction. Dissertation, TU-Darmstadt, December 2005, http://elib.tu-darmstadt.de/diss/000640 and http://deposit.ddb.de/cgi-bin/dokserv?idn=978166493.
52. *J. Martinet*, Perfect Lattices in Euclidean Spaces. Springer-Verlag, Berlin New York, 2002.
53. *A. May*, Computing the RSA Secret Key is Deterministic Polynomial Time Equivalent to Factoring. In Proc. CRYPTO 2004, LNCS 3152, Springer-Verlag, Berlin New York, pp. 213–219, 2004.
54. *D. Micciancio and S. Goldwasser*, Complexity of Lattice Problems, A Cryptographic Perspective. Kluwer Academic Publishers, London, 2002.
55. *V. Shoup*, Number Theory Library. http: //www.shoup.net/ntl/
56. *P.Q. Nguyen and D. Stehlé*, Floating-Point LLL Revisited. In Proc. Eurocrypt'05, LNCS 3494, Springer-Verlag, Berlin New York, pp. 215–233, 2005.
57. *P. Nguyen and D. Stehlé*, LLL on the Average. In Proc. ANTS-VII, LNCS 4076, Springer-Verlag, Berlin New York, pp. 238–356, 2006.

58. *P.Q. Nguyen and J. Stern*, Lattice Reduction in Cryptology, An Update. Algorithmic Number Theory, LNCS 1838, Springer-Verlag, Berlin New York, pp. 85–112, 2000. full version http://www.di.ens.fr/pnguyen,stern/
59. *P.Q. Nguyen and J. Stern*, The Two Faces of Cryptology, In Proc. CaLC'01, LNCS 2139, Springer-Verlag, Berlin New York, pp. 260–274, 2001.
60. *P.Q. Nguyen and T. Vidick*, Sieve Algorithms for the Shortest Vector Problem are Practical. Preprint, 2007.
61. *A.M.Odlyzko*, The Rise and the Fall of Knapsack Cryptosystems, In Proc. of Cryptology and Computational Number Theory, vol. 42 of Proc. of Symposia in Applied Mathematics, pp. 75–88, 1989.
62. *O. Regev*, Lecture Notes of Lattices in Computer Science, taught at the Computer Science Tel Aviv University, 2004; available at http://www.cs.tau.il/ odedr.
63. *C. Rössner and C.P. Schnorr*, An Optimal, Stable Continued Fraction Algorithm for Arbitrary Dimension. In Proc. 5-th IPCO 1996, LNCS 1084, Springer, Berlin New York, pp. 31–43, 1996.
64. *C.P. Schnorr*, A Hierarchy of Polynomial Time Lattice Basis Reduction Algorithms. *Theoret. Comput. Sci.*, **53**, pp. 201–224, 1987.
65. *C.P. Schnorr*, A More Efficient Algorithm for Lattice Reduction. *J. of Algor.* **9**, 47–62, 1988.
66. *C.P. Schnorr and M. Euchner*, Lattice Basis Reduction and Solving Subset Sum Problems. In Proc. Fundamentals of Comput. Theory, LNCS 591, Springer-Verlag, Berlin New York, pp. 68–85, 1991. Complete paper in *Math. Programming Studies*, **66A**, 2, pp. 181–199, 1994.
67. *C.P. Schnorr*, Block Reduced Lattice Bases and Successive Minima. *Combin. Probab. and Comput.*, **3**, pp. 507-522, 1994.
68. *C.P. Schnorr and H.H. Hörner*, Attacking the Chor-Rivest cryptosystem by improved lattice reduction. In Proc. Eurocrypt 1995, LNCS 921, Springer-Verlag,Berlin New York, pp. 1–12, 1995.
69. *C.P. Schnorr*, Lattice Reduction by Random Sampling and Birthday Methods. In Proc. STACS 2003, Eds. H. Alt and M. Habib, LNCS 2607, Springer-Verlag, Berlin New York, pp. 145–156, 2003.
70. *C.P. Schnorr*, Fast LLL-type lattice reduction. Information and Computation, **204**, pp. 1–25, 2006. //www.mi.informatik.uni-frankfurt.de
71. *A. Schönhage*, Factorization of Univariate Integer Polynomials by Diophantine Approximation and Improved Lattice Basis Reduction Algorithm. In Proc. ICALP 1984, LNCS 172, Springer-Verlag, Berlin New York, pp. 436–447, 1984.
72. *D.Simon*, Solving Quadratic Equations Using Reduced Unimodular Quadratic Forms. *Math. of Comp.* **74** (251), pp. 1531–1543, 2005.
73. *A. Storjohann*, Faster Algorithms for Integer Lattice Basis Reduction. TR 249, Swiss Federal Institute of Technology, ETH-Zurich, July 1996. //www.inf.ethz.ch/research/publications/html.
74. *D. Stehlé*, Floating Point LLL: Theoretical and Practical Aspects. This Proceedings.
75. *G. Villard*, Certification of the QR Factor R, and of Lattice Basis Reducedness. LIP Research Report RR2007-03, ENS de Lyon, 2007.
76. *J.H. Wilkinson*, The Algebraic Eigenvalue Problem. Oxford University Press, 1965, reprinted 1999.