Segment LLL-Reduction with Floating Point Orthogonalization.

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

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

Abstract. We associate with an integer lattice basis a *scaled basis* that has orthogonal vectors of nearly equal length. The orthogonal vectors or the QR-factorization of a scaled basis can be accurately computed up to dimension 2^{16} by Householder reflexions in floating point arithmetic (fpa) with 53 precision bits.

We develop a highly practical fpa-variant the new *segment* LLL-*reduction* of Koy and Schnorr [KS01]. The LLL-steps are guided in this algorithm by the Gram-Schmidt coefficients of an associated scaled basis. The new reduction algorithm is much faster than previous codes for LLL-reduction and performs well beyond dimension 1000.

Keywords. LLL-reduction, Householder reflexion, floating point arithmetic, stability, scaled basis, segment LLL-reduction, local LLL-reduction.

1 Introduction.

Practical algorithms for LLL-reduction compute the orthogonal vectors of a basis in floating point arithmetic (fpa). The corresponding fpa-errors must be small otherwise the size-reduction included in the orthogonalization gets unstable. Up to dimension 250 Gram-Schmidt orthogonalization of an LLL-reduced basis can be done in fpa with some correction steps [SE91]. Householder orthogonalization has better stability, as was shown in [RS96]. In practice it yields up to dimension 350 a sufficient QR-factorization of the basis matrix — but this process gets unstable in dimension 400. [S88] presents a method for correcting the Gram-Schmidt coefficients using Schulz's method of matrix inversion. This method is provably stable but requires $O(n + \log_2 M)$ precision bits, where M bounds the Euclidean length of the basis vectors. It is useless for fpa with 53 precision bits.

In this paper we associate with an integer lattice basis a *scaled basis* that has orthogonal vectors of nearly equal length. The orthogonal vectors or the QR-factorization of a scaled basis can be accurately computed up to dimension 2^{16} by Householder reflexions in fpa with 53 precision bits. We present an algorithm **HRS** that efficiently generates an associated scaled basis. We present a fpa-variant of LLL-reduction where LLL-steps are guided by the orthogonal vectors of an associated scaled basis. In particular, size-reduction is done against the scaled basis. The weaker size-reduction does in practice not degrade the quality of the reduced basis.

We present a *fpa*-variant of segment LLL-reduction, a novel concept proposed in [KS01]. The algorithm **segment sLLL** performs scaled segment LLL-reduction, so that all LLL-steps are guided by the orthogonaliation of an associated scaled basis. The algorithm **segment sLLL** is a very efficient reduction algorithm. Its efficiency comes from local LLL-reduction of two consecutive segments B_{l-1}, B_l that is done by reducing the local matrix R_l in *fpa*. To make this local LLL-reduction possible in the limits of *fpa* it is necessary to bound the integer transformation matrix of the local LLL-reduction. For this we carefully prepare the local matrix R_l of B_{l-1}, B_l prior to local LLL-reduction.

In practice, **segment sLLL** is much faster than previous codes for LLL-reduction. It performs well beyond dimension 1000 and provides lattice bases that are in practice of similar quality as LLL-reduced bases. For dimension 1000 and basis vectors with 400 bit integer coordinates, the new LLL-reduction takes only about 10 hours on a 800 MHz PC. It is the first code for LLL-reduction that performs well beyond dimension 350.

In this paper we focus on practical aspects related to *fpa*. A rigorous analysis of segment LLL-reduction in the model of integer arithmetic is in the companion paper [KS01]. For general background on *fpa* and Householder transformation see [LH95].

2 Stability Properties of Householder Orthogonalization.

For an introduction of LLL-reduced lattice bases and of our notation on lattices we refer to Section 2 of the companion paper [KS01]. For easy reference we recall Definition 1 and Theorem 1 from [KS01]. We let $\delta \in]\frac{1}{4}, 1]$ and $\alpha = 1/(\delta - \frac{1}{4})$.

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

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

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

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

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

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

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

Consider the process of orthogonalization of a basis matrix $B = [b_1, ..., b_n]$. In ideal arithmetic we get the QR-factorization $B = QR$ by a sequence of Householder transformations

$$C_1 := B, \ C_{j+1} := Q_j C_j \text{ for } j = 1, \ldots, n,$$

where $Q_j \in \mathbf{R}^{d \times d}$ is an orthogonal matrix — an Householder reflexion — that produces zeros in positions $j+1$ through d of column j of $Q_j C_j$. Thus $C_{j+1} \in \mathbf{R}^{d \times n}$ is upper triangular in the first j columns. Finally, $R = C_{n+1}$, $Q = Q_n \cdots Q_1$.

In actual computation, however, we use floating point operations

$$\bar{C}_1 = fl(B), \ \bar{C}_{j+1} := fl(\bar{Q}_j \bar{C}_j) \text{ for } j = 1, \ldots, n.$$

We assume the standard fpa-model of Wilkinson, see [LH95, p. 85] for details. Let $0 < \eta \ll 1$ be the relative precision — each floating point operation induces a normalized relative error bounded in magnitude by η. [1] In this model, it has been shown [LH 95, p. 87, formula (15.38)] that

$$\|\bar{C}_{j+1} - Q_j \cdots Q_1 B\|_F \leq (6d - 3j + 40) j\eta \|B\|_F + O(\eta^2), \tag{1}$$

where $\|A\|_F = (\sum_{i,j} a_{i,j}^2)^{\frac{1}{2}}$ denotes the Frobenius norm of the matrix $A = [a_{i,j}]$. In the following we neglect the low order term $O(\eta^2)$. Thus, in actual computation we get $\bar{R} = [\bar{r}_{i,j}] = \bar{C}_{n+1}$ satisfying for $n \geq 14$.

$$\|\bar{R} - R\|_F \leq 6dn\eta \|B\|_F. \tag{2}$$

It follows that the approximate Gram-Schmidt coefficients $\bar{\mu}_{j,i} = \bar{r}_{i,j}/\bar{r}_{i,i}$ satisfy for $i < j$

$$|\bar{\mu}_{j,i} - \mu_{j,i}| \leq 6dj\eta \|b_1, \ldots, b_j\|_F / (|r_{i,i}|(1 - \varepsilon)) \tag{3}$$

provided that $|\bar{r}_{i,i} - r_{i,i}| \leq \varepsilon |r_{i,i}|$. Therefore, we get from inequality (2) the

Lemma 1. *Inequality* (3) *holds provided that*

$$6dj\eta \|b_1, \ldots, b_j\|_F \leq \varepsilon |r_{i,i}|. \tag{4}$$

Instability of Householder QR-factorization for dimension 400. Inequalities (1), (2), (3) are rather sharp. To combine Householder transformation and size-reduction we need accurate coefficients $\mu_{j,i}$. Condition (4) characterizes the stability of Householder transformation — stability requires that $6dn\eta \|B\|_F < \min_i |r_{i,i}|$. On the other hand, Theorem 2 shows that random LLL-reduced bases on the average satisfy $\|b_1\| \approx 1.1^{n-1} \|\widehat{b}_n\|$. For dimension $n = 400$ and $\eta = 2^{-53}$ this yields $6dn\eta \|b_1\|_F \approx 6dn\eta 1.1^{n-1} \|\widehat{b}_n\| \approx 4 \cdot 10^6 \|\widehat{b}_n\|$. Inequality (4) is grossly violated. Therefore, Householder transformation of LLL-reduced bases is necessarily unstable in dimension 400 for fpa with 53 precision bits.

Theorem 2. *Consider a random LLL-reduced basis with random coefficients* $\mu_{i+1,i} \in_R [-\frac{1}{2}, \frac{1}{2}]$ *for* $i = 1, ..., n-1$, *and let the inequalities* **2.** *of Definition* 1 *be tight. Then* $\|b_1\| \approx 1.1^{n-1} \|\widehat{b}_n\|$ *holds on the average.*

[1] standard double length, wired fpa has 53 precision bits, $\eta = 2^{-53}$.

Proof. While $(\delta - \mu_{i+1,i}^2)\|\widehat{b}_i\|^2 \le \|\widehat{b}_{i+1}\|^2$ holds for LLL-reduced bases, the converse $(\delta-\varepsilon^2)\|\widehat{b}_i\|^2 \ge \|\widehat{b}_{i+1}\|^2$ holds provided that $\delta\|\widehat{b}_i\|^2 = \|\widehat{b}_{i+1}\|^2 + \mu_{i+1,i}^2\|\widehat{b}_i\|^2$ and $|\mu_{i+1,i}| \le \varepsilon$. The inequality $|\mu_{i+1,i}| \le \varepsilon$ holds with probability 2ε for random $\mu_{i+1,i} \in_R [-\frac{1}{2}, \frac{1}{2}]$. As $\int_{-\frac{1}{2}}^{\frac{1}{2}} 2\varepsilon^2 d\varepsilon = \frac{1}{6}$ and $\sqrt{\delta - \frac{1}{6}} \approx 1.1^{-1}$ holds for $\delta \approx 1$ we see that $\|b_1\| \approx 1.1^{n-1}\|\widehat{b}_n\|$ holds on the average. □

3 The Scaled Basis Matrix.

Scaling is a well known method for improving the stability of *fpa*. We associate with an integer lattice basis $b_1, ..., b_n$ a scaled basis $b_1^s, ..., b_n^s$ that has orthogonal vectors of nearly equal length.

Definition 2. *Let $b_1, \dots, b_n$ be a basis of an integer lattice L. We call $b_1^s, ..., b_n^s \in L$ an associated* scaled basis *with* scaling factors $2^{e_1}, ..., 2^{e_n}$ — $e_1, ..., e_n \in \mathbf{N}$ — *if $b_1^s, ..., b_n^s$ form a size-reduced basis of a sublattice of L satisfying*

$$\|b_1^s\| \le 2^{e_i}\|\widehat{b}_i\| = \|\widehat{b}_i^s\| < 2\|b_1^s\| \quad \text{for } i = 1, ..., n. \tag{5}$$

We show in Theorem 3 that a given scaled basis yields accurate Gram-Schmidt coefficients by Householder reflexions in *fpa*. In Section 4 we show how to produce an associated scaled basis efficiently in *fpa*. In Sections 5 and 7 we use an associated scaled basis to guide LLL-reduction and segment LLL-reduction.

We can easily transform a basis $b_1, ..., b_n$ into an associated scaled basis $b_1^s, ..., b_n^s$ using exact arithmetic — first scale then size-reduce:

1. *scaling.* $e_1 := \lfloor \log_2(\max_j \|\widehat{b}_j\|/\|b_1\|) \rfloor$, $b_1^s := 2^{e_1}\, b_1$,

for $i = 2, ..., n$ **do** $e_i := \max(0, \lceil \log_2 \|b_1^s\|/\|\widehat{b}_i\| \rceil)$, $b_i^s := 2^{e_i} b_i$

2. *size-reduction.* **for** $j = 2, ..., n$ **for** $i = j-1, ..., 1$ **do** $b_j^s := b_j^s - \lceil \frac{\langle b_j^s, \widehat{b}_i^s \rangle}{\|\widehat{b}_i^s\|^2} \rfloor\, b_i^s$. [2]

New Notation. For the remaining of the paper we let the column vector $\mathbf{r}_\nu$ of the R-matrix be related to the scaled vector b_ν^s rather than to the original vector b_ν.

The size of scaled Gram-Schmidt coefficients. The coefficients $\mu_{\nu,j}^s$ of an arbitrary, unscaled vector $b_\nu = \widehat{b}_\nu + \sum_{j=1}^{\nu-1} \mu_{\nu,j}^s \widehat{b}_j^s$ are uniformly bounded:

Corollary 1. $|\mu_{\nu,j}^s| = \frac{|\langle b_\nu, \widehat{b}_j^s \rangle|}{\|\widehat{b}_j^s\|^2} \le \frac{\|b_\nu\|}{\|\widehat{b}_j^s\|} \overset{(5)}{\le} 2 \cdot \frac{\|b_\nu\|}{\|b_1^s\|}$.

In contrast, LLL-reduced bases $b_1, \dots, b_{\nu-1}$ satisfy $b_\nu = \sum_{j=1}^{\nu} \mu_{\nu,j}\widehat{b}_j$ with $|\mu_{\nu,j}|^2 \le \frac{\|b_\nu\|^2}{\|b_1\|^2}\, \alpha^{j-1}$. The coefficient $\mu_{\nu,\nu-1}$ that enters first into size-reduction of b_ν tends to be very large due to the factor $\alpha^{\nu-1}$. A small relative error of $\mu_{\nu,\nu-1}$ confuses the size-reduction of b_ν.

[2] Let $\lceil r \rfloor = \lceil r - \frac{1}{2} \rceil$ be the nearest integer to the real number r.

Corollary 2. *The basis $b_1^s, \dots, b_n^s$ satisfies $\|b_j^s\| \le \sqrt{j+3}\,\|b_1^s\|$ for $j = 1, \dots, n$.*

Proof.
$$\begin{aligned} \|b_j^s\|^2 &= \|\widehat{b}_j^s\|^2 + \textstyle\sum_{i=1}^{j-1} (\mu_{j,i}^s)^2 \|\widehat{b}_i^s\|^2 \\ &\le \|\widehat{b}_j^s\|^2 + (j-1)/4 \max_{i<j} (\|\widehat{b}_i^s\|)^2 \\ &\overset{(5)}{\le} 4\|b_1^s\|^2 + (j-1)\,\|b_1^s\|^2 = (j+3)\|\widehat{b}_1^s\|^2. \end{aligned}$$
□

Next we study for a given scaled basis the accuracy of the approximate coefficients $\bar{\mu}_{j,i}^s$, computed in fpa by Householder reflexions.

Theorem 3. *The approximate $\bar{\mu}_{j,i}^s$ of $b_1^s, \dots, b_n^s$ satisfy $|\bar{\mu}_{j,i}^s - \mu_{j,i}^s| \le \varepsilon/(1-\varepsilon)$ for $\varepsilon = 6dj^2\eta$.*

Proof. By scaling and size-reduction we have for $j \ne 2$: [3]

$$\|b_j^s\|^2 \le \tfrac{1}{4} \textstyle\sum_{i=1}^{j-1} \|\widehat{b}_i^s\|^2 + \|\widehat{b}_j^s\|^2 \le \frac{j+3}{4} \max_{i\le j} \|\widehat{b}_i^s\|^2 \overset{(5)}{\le} j \min_{i\le j} \|\widehat{b}_i^s\|^2.$$

Hence $\|b_1^s, \dots, b_j^s\|_F = (\sum_{i=1}^{j} \|b_i^s\|^2)^{1/2} \le \sqrt{j}\max_{i\le j}\|b_i^s\| \le j\,\|\widehat{b}_i^s\|$, and thus

$$\|b_1^s, \dots, b_j^s\|_F / \|\widehat{b}_i^s\| \le j \qquad \text{for } i = 1, \dots, j. \tag{6}$$

Hence, Inequality (4) holds for $\varepsilon = 6dj^2\eta$. By Lemma 1 Inequality (3) holds for that ε and thus $|\bar{\mu}_{j,i}^s - \mu_{j,i}^s| \overset{(3)}{\le} 6dj\eta\,\|b_1^s, \dots, b_j^s\|_F / (|r_{i,i}|(1-\varepsilon)) \overset{(6)}{\le} 6dj^2\eta/(1-\varepsilon) = \varepsilon/(1-\varepsilon)$. □

Stability up to dimension 2^{16}. Consider Theorem 3 in the case $j \le n = d = 2^{16}$ and $\eta = 2^{-53}$. Then we have $\varepsilon \le 6n^3\eta \le 0.19$ and $\varepsilon/(1-\varepsilon) \le 0.24$. Therefore, Householder reflexions yield up to dimension 2^{16} sufficiently accurate Gram-Schmidt coefficients for the basis $b_1^s, \dots, b_n^s$.

4 Orthogonalization via Scaling and Size-Reduction.

Suppose we are given $b_1^s, \dots, b_{\nu-1}^s, b_\nu$ and we want to produce a scaled vector b_ν^s. At that point the $\mu_{j,i}^s$ of $b_1^s, \dots, b_{\nu-1}^s$ are given with high accuracy. We iteratively transform b_ν into b_ν^s using better and better approximations of the $\mu_{\nu,i}^s$. The procedure **HRS** (Householder, Reduction, Scaling) iterates the following steps

1. the first $\nu - 1$ Householder transformations $b_\nu \mapsto \bar{Q}_{\nu-1} \cdots \bar{Q}_1 b_\nu$,
2. size-reduction of b_ν against $b_1^s, \dots, b_{\nu-1}^s$,
3. scaling of b_ν to b_ν^s.

Steps **1. 2.** must be iterated as the size of b_ν is by Inequality (2) crucial for the accuracy of Householder transformation. As the scaling increases the size of b_ν we scale in stages, repeating **1. 2.** after each stage of scaling. Let

[3] In the following we neglect the exception $j = 2$.

$Q_j = I_d - 2v_j v_j^\top \|v_j\|^{-2}$, where $I_d \in \mathbf{Z}^{d\times d}$ is the identity matrix and $v_j \in \mathbf{R}^d$ is the Householder vector associated with Q_j. Note that $x \mapsto Q_j x$ reflects x at the hyperplane that is orthogonal to v_j: $Q_j v_j = -v_j, \;\; Q_j u = u \;\;$ for $\;\; u \perp v_j$. **HRS** is given for input the Householder vectors $v_1, ..., v_{\nu-1} \in \mathbf{R}^d$, the first $\nu - 1$ columns $\bar{\mathbf{r}}_1, ..., \bar{\mathbf{r}}_{\nu-1}$ of the matrix $\bar{C}_\nu = \bar{Q}_{\nu-1} \cdots \bar{Q}_1 \bar{C}_1$ and the computed scaling exponents $\bar{e}_1, ..., \bar{e}_{\nu-1}$. The operations on $b_1^s, ..., b_{\nu-1}^s$ are in exact integer arithmetic, the other operations are in fpa. Taking fpa-errors into account we relaxe size-reduction of b_ν^s to the relaxed condition $|\mu_{\nu,j}^s| \le 0.52$.

Supressing backward rescaling. Upon entry of $\mathbf{HRS}(\nu)$, the scaled vectors $b_1^s, ..., b_{\nu-1}^s$ are given while $b_\nu^s, ..., b_n^s$ are unknown. At this stage the scaling factors $2^{\bar{e}_1}, ..., 2^{\bar{e}_{\nu-1}}$ correspond to the subbasis $b_1, ..., b_{\nu-1}$. If $\|\widehat{b}_\nu\| > \|\widehat{b}_1^s\|$ we would need to rescale b_i^s by increasing $\bar{e}_i := \bar{e}_i + \bar{e}$ and $b_i^s := 2^{\bar{e}}\, b_i^s$ for $\bar{e} := \lfloor \log_2(\|\widehat{b}_\nu\| / \|\widehat{b}_1^s\|\,) \rfloor$ and $i = 1, ..., \nu - 1$. We suppress this backward rescaling. It is sufficient to store $\bar{e}$ and to do all subsequent size-reductions against $2^{\bar{e}}\, b_i^s$ rather than against b_i^s, i.e., we replace subsequent reduction steps $\;b := b - \lceil \frac{\langle b, \widehat{b}_i^s\rangle}{\langle \widehat{b}_i^s, \widehat{b}_i^s\rangle} \rfloor b_i^s\;$ by the steps $b := b - 2^{\bar{e}} \lceil \frac{2^{-\bar{e}}\langle b, \widehat{b}_i^s\rangle}{\langle \widehat{b}_i^s, \widehat{b}_i^s\rangle} \rfloor b_i^s$. For simplicity, the program $\mathbf{HRS}(\nu)$ does not include the steps required in case that $\|\widehat{b}_\nu\| > \|\widehat{b}_1^s\|$.

$\mathbf{HRS}(\nu)$ *(Householder transformation, Reduction and Scaling of* b_ν*)*

INPUT $\quad b_1^s, \ldots, b_{\nu-1}^s \in \mathbf{Z}^d,\; \bar{\mathbf{r}}_1, ..., \bar{\mathbf{r}}_{\nu-1},\; v_1, \ldots, v_{\nu-1} \in \mathbf{Q}^d,\; \bar{e}_1, ..., \bar{e}_{\nu-1} \in \mathbf{Z}$

OUTPUT $\;b_\nu^s$ (size-reduced against $b_1^s, ..., b_{\nu-1}^s$), $\;v_\nu, \bar{\mathbf{r}}_\nu, \bar{e}_\nu$

1. $\bar{e}_\nu := 0,\; b_\nu^s := b_\nu$

2. $\bar{\mathbf{r}}_\nu := \bar{Q}_{\nu-1} \cdots \bar{Q}_1 \cdot b_\nu^s$

3. *size-reduce* b_ν^s *against* $b_1^s, ..., b_{\nu-1}^s$
  for $i = \nu - 1, \ldots, 1$ **do**
    $\bar{\mu}_{\nu,i}^s := \bar{r}_{i,\nu} \,/\, \bar{r}_{i,i}$
    if $|\bar{\mu}_{\nu,i}^s| \ge 0.51$ **then** $b_\nu^s := b_\nu^s - \lceil \bar{\mu}_{\nu,i}^s \rfloor b_i^s,\; \bar{\mathbf{r}}_\nu := \bar{\mathbf{r}}_\nu - \lceil \bar{\mu}_{\nu,i}^s \rfloor \bar{\mathbf{r}}_i$

4. if $\exists i : |\bar{\mu}_{\nu,i}^s| \ge 2^{10}$ **then go to 2.**

5. $\tau := (\,\sum_{i=\nu}^d \bar{r}_{i,\nu}^2\,)^{\frac{1}{2}},\;\; \widetilde{e} := \min(\lceil \log_2(\|b_1^s\|/\tau) \rceil, 30)$

6. if $\tilde{e} > 0$ **then** $\bar{e}_\nu := \bar{e}_\nu + \widetilde{e},\;\; b_\nu^s := 2^{\widetilde{e}} b_\nu^s$ **go to 2.**

7. $\bar{\mathbf{r}}_\nu := \bar{Q}_{\nu-1} \cdots \bar{Q}_1 b_\nu^s,\;\; \sigma := \mathrm{sig}(\bar{r}_{\nu,\nu}), \quad \tau := (\,\sum_{i=\nu}^d \bar{r}_{i,\nu}^2\,)^{\frac{1}{2}}$
  $v_\nu := (0, \ldots, 0, \bar{r}_{\nu,\nu} + \sigma\tau,\; \bar{r}_{\nu+1,\nu}, \ldots, \bar{r}_{d,\nu})^\top$
  $\bar{\mathbf{r}}_\nu := (\bar{r}_{1,\nu}, \ldots, \bar{r}_{\nu-1,\nu}, -\sigma\tau, 0, \ldots, 0)^\top \qquad$ *end*

How **HRS** *works.* According to (5) the input vectors $b_1^s, ..., b_{\nu-1}^s$ satisfy $|r_{1,1}| \le |r_{i,i}| < 2|r_{1,1}|$ for $i = 1, ..., \nu - 1$. Let $6d\nu^2\eta < \frac{1}{2}$ so that Theorem 3 holds for $\varepsilon = \frac{1}{2}$. Then, by (2) and (6) we have that $|\bar{r}_{i,i} - r_{i,i}| \le 6dj^2\eta |r_{i,i}|$ for $i = 1, ..., j$.

Reducing long $\|b_\nu\|$. By (2) the relative error of $\bar{r}_{i,\nu}/\|b_\nu\|$ in step **2.** is at most $6d\nu\eta$. The subsequent size-reduction reduces the large $\bar{r}_{i,\nu}$ — in the equation

$\|b_\nu\|^2 = \sum_{i=1}^{d} r_{i,\nu}^2$ — to less than $|\bar{r}_{i,i}|$ in absolute value. Steps **2.** **3.** **4.** decrease the $(\sum_{i=1}^{\nu-1} \bar{r}_{i,\nu}^2)^{\frac{1}{2}}$–part until the inequality $(\sum_{i=1}^{\nu-1} \bar{r}_{i,\nu}^2)^{\frac{1}{2}} \leq \|b_1^s, ..., b_{\nu-1}^s\|_F$ holds.

Upon entry of step **3.** the coefficients $\mu_{\nu,i}^s$ are uniformly bounded as $|\mu_{\nu,j}^s| \leq 2 \cdot \frac{\|b_\nu\|}{\|b_1^s\|}$, see Corollary 1. Size-reduction of Step **3.** can temporarily increase the coefficients $|\mu_{\nu,j}^s|$ up to a factor $(3/2)^{\nu-j-1}$ in worst case. This could possibly make the size-reduction of step **3.** unstable in worst case. No such instability has been reported in practice, see Figure 1.

Scaling up small $\|\widehat{b}_\nu\|$. Let $\|b_\nu\| \leq \|b_1^s, ..., b_{\nu-1}^s\|_F$ and $|r_{\nu,\nu}| \leq \eta\|b_\nu\|$. By (2) and (6) we have after step **2.** that $|\bar{r}_{\nu,\nu} - r_{\nu,\nu}| < 6d\nu\eta\|b_1^s, ..., b_{\nu-1}^s\|_F \leq 6d\nu^2\eta\|b_1^s\|$. This yields a scaling factor $\tilde{e}$ for step **6.** so that $2^{\tilde{e}} \geq (6d\nu^2\eta)^{-1} > 1$. Therefore, steps **2.** to **6.** increase $|\bar{r}_{\nu,\nu}|$, $\|b_\nu^s\|$ until $|\bar{r}_{1,1}| \leq |\bar{r}_{\nu,\nu}| \approx \|b_\nu^s\| < 2|\bar{r}_{1,1}|$.

Corollary 3. **HRS**(ν) *produces a scaled vector* b_ν^s *satisfying* $|\mu_{\nu,i}^s| \leq 0.52$ *for* $i = 1, ..., \nu - 1$ *provided that size-reduction of Step* **3.** *remains stable.*

Speeding up **HRS**. To speed up **HRS** replace the rounded values $\lceil \bar{\mu}_{\nu,i}^s \rfloor$ in step **3.** by single precision integers consisting of the leading bits of $\lceil \bar{\mu}_{\nu,i}^s \rfloor$. This way **HRS** becomes very fast.

The number of arithmetic steps. A matrix-vector multiplication $b_\nu \mapsto \bar{Q}_1 b_\nu = b_\nu - \frac{2 v_1 v_1^\top b_\nu}{\|v_1\|^2}$ requires $2d$ multiplications and one division — as usual we neglect the additions/subtractions. Thus, we get $\bar{\mathbf{r}}_\nu := \bar{Q}_{\nu-1} \cdots \bar{Q}_1 b_\nu$ using $2\nu d$ multiplications/divisions. Size-reduction of step **3.** requires νd multiplications in exact arithmetic. In total, one *round* of steps **2.** **3.** **4.** requires $2\nu d$ multiplications/divisions in *fpa* and νd exact multiplications of long integers.

The number of rounds. In practice each round of steps **2.** **3.** **4.** either decreases $(\sum_{i=1}^{\nu-1} r_{i,\nu}^2)^{\frac{1}{2}}$ by the factor $6d\nu^2\eta$ or increases $\|\widehat{b}_\nu^s\|$ by the scaling factor $2^{\tilde{e}_\nu} \geq (6d\nu^2\eta)^{-1} > 1$ or does a combination of both, see the explanations for **HRS**. In practice **HRS**(ν) requires $\log_2(\|b_\nu\|/\|\widehat{b}_\nu\|)/|\log_2(6d\nu^2\eta)|$ rounds.

Practical Performance of **HRS.** Consider a random public-key basis of the GGH-cryptosystem [GGH97] of dimension 400. Figure 1 shows how **HRS**(400) decreases the coefficients $|\mu_{400,j}|$ of the last basis vector b_{400}. Using 53 bit *fpa* each of five rounds decreases $|\mu_{400,j}|$ by a factor about $10^9 \approx 2^{30}$. After each round all coefficients $|\mu_{400,j}|$ are of nearly equal size because the preceding orthogonal vectors have been scaled to nearly equal length $\|\widehat{b}_j^s\|$. After each of the first 4 rounds however, $|\mu_{\nu,j}^s|$ increases by about a factor 10 as j decreases from 400 to 1. This is due to the temporary increase of $|\mu_{\nu,j}^s|$ by a factor $(3/2)^{\nu-j-1}$ in worst case. Figure 1 shows that this temporary increase has little effect in practice.

Using 106 bit *fpa* — instead of 53 bit *fpa* — the five rounds of **HRS**(400) reduce to two rounds, and the running time of **HRS** reduces accordingly. Software implemented *fpa* with 106 precision bits makes the reduction clearly faster than the standard *fpa* with 53 precision bits.

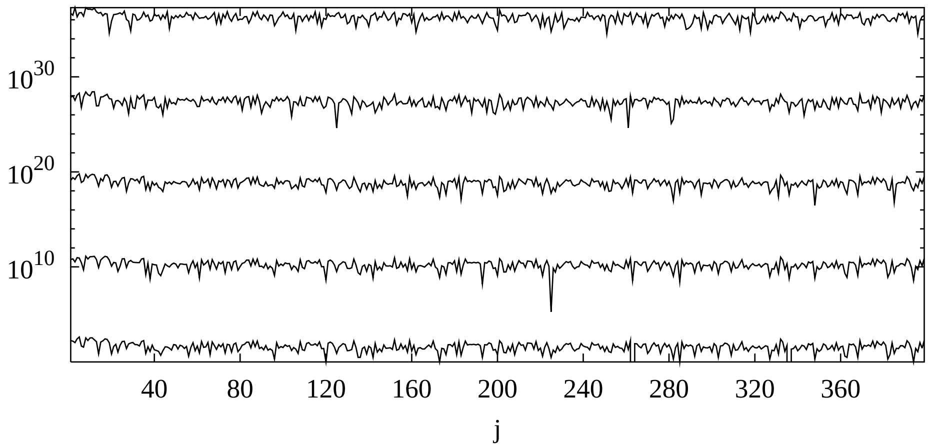

Fig. 1. Displayed are the values $|\mu_{400,j}|$ upon termination of each of five rounds. The $|\mu_{400,j}|$ are measured after step 2 of the next round — after new orthogonalization.

5 Scaled LLL-Reduction.

We introduce *scaled* LLL-*reduced* lattice bases, and we present an algorithm **scaled LLL** for scaled LLL-reduction. This algorithm is a useful prelude to the more complicated scaled segment LLL-reduction of Section 7.

Definition 3. *We call a lattice basis* $b_1, \ldots, b_n \in \mathbf{Z}^d$ scaled LLL-reduced *if it has properties* **1. 2.***:*

1. *There is an associated scaled basis* $b_1^s, ..., b_n^s$ *so that* $b_1, ..., b_n$ *is size-reduced against* $b_1^s, ..., b_n^s$*, i.e.,* $\left|\frac{\langle b_\nu, \widehat{b}_j^s\rangle}{\|\widehat{b}_j^s\|^2}\right| \leq 0.52$ *for* $1 \leq i < j \leq n$*,*

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

We call a basis with property **1.** *scaled-reduced.* Importantly, the inequalities of Theorem 1 still hold for scaled LLL-reduced lattice bases. The weaker size-reduction does not affect these inequalities. Scaled LLL-reduced bases are in practice as good as LLL-reduced bases.

Next we present an algorithm **scaled LLL** that transforms a lattice basis into a scaled LLL-reduced basis of the same lattice. **Scaled LLL** guides the LLL-steps on the original basis by the orthogonalization of an associated scaled basis. So we keep and update two versions of the basis. LLL-transformations operate on the original basis. Size-reduction is done against the scaled basis. This form of LLL-reduction is quite stable as **HRS** yields an accurate QR-factorization of the scaled basis.

The procedure **HRS'**. Local LLL-reduction of two basis vectors $b_{\nu-1}, b_\nu$ is guided by the orthogonal vector of a modified scaled vector b_ν^s — with a scaling factor

$2^{\bar{e}_\nu}$ that coincides with the scaling factor $2^{\bar{e}_{\nu-1}}$ of $b_{\nu-1}$. We get the modified b_ν^s by the following variant **HRS'** of **HRS**: Set in Step **1.** $\bar{e}_\nu := \bar{e}_{\nu-1}$ and skip Step **6.**. Moreover, we let **HRS'** perform in Step **3.** the same size-reduction steps $b_\nu^s := b_\nu^s - \lceil \bar{\mu}_{\nu,\nu-1}^s \rfloor b_{\nu-1}^s$, $b_\nu := b_\nu - \lceil \bar{\mu}_{\nu,\nu-1}^s \rfloor b_{\nu-1}$ on b_ν^s against $b_{\nu-1}^s$ and on b_ν against $b_{\nu-1}$. As a consequence we have

Lemma 2. *The reduction coefficients of b_ν against $b_{\nu-1}$ and of b_ν^s against $b_{\nu-1}^s$ coincide in* **HRS'**(ν)*, and thus $\mu_{\nu,\nu-1} = \mu_{\nu,\nu-1}^s$. The output vector b_ν of* **HRS'**(ν) *is size-reduced against $b_{\nu-1}$. The coefficients $r_{i,j}$ for $\nu - 1 \le i, j \le \nu$ associated with $b_{\nu-1}, b_\nu$ are proportional to the coefficients associated with $b_{\nu-1}^s, b_\nu^s$ with proportionality factor $2^{\bar{e}_\nu}$.*

Scaled LLL (*Algorithm for scaled* LLL*-reduction.*)
INPUT $b_1, \ldots, b_n \in \mathbf{Z}^d$, δ
OUTPUT $b_1, \ldots, b_n$ scaled LLL-reduced basis
1. $\nu := 1$
2. **while** $\nu \le n$ **do**
3. **if** $\nu = 1$ **then** **HRS**(1), $\nu := 2$
4. **HRS'**(ν)
5. **if** $\delta\, \bar{r}_{\nu-1,\nu-1}^2 > \bar{r}_{\nu-1,\nu}^2 + \bar{r}_{\nu,\nu}^2$
then swap $b_{\nu-1}, b_\nu$, $\nu := \nu - 1$
else **HRS**(ν), $\nu := \nu + 1$ *end*

Scaling does not affect LLL*-exchanges.* By Lemma 2, the output vector b_ν of **HRS'**(ν) is size-reduced against $b_{\nu-1}$, i.e., $|\mu_{\nu,\nu-1}| \le 0.52$. Moreover, the decision about swapping $b_{\nu-1}, b_\nu$ in Step **5.** is the same as for the original LLL-algorithm — except for fpa-errors.
fpa-errors do not affect the LLL*-exchanges.* We see from $\|b_\nu^s\|^2 = \sum_{i=1}^d r_{i,\nu}^2$ and (2) that the relative error of $\bar{r}_{\nu-1,\nu}/\|b_\nu^s\| \approx \bar{r}_{\nu-1,\nu}/|\bar{r}_{\nu,\nu}|$ and $|r_{\nu,\nu}|/\|b_\nu^s\|$ is at most $6d\nu\eta$. On the other hand, $|\bar{r}_{\nu-1,\nu-1}| \ge \|b_1^s\|$ holds by scaling. Therefore, the decision about swapping $b_{\nu-1}, b_\nu$ in Step **5.** is correct for $6d\nu\eta \ll 1$.

6 Segment LLL-Reduction.

We summarize the concept of segment LLL-reduced bases of the companion paper, for full coverage see [KS01].

Segments and local coordinates. Let the basis $b_1, \ldots, b_n \in \mathbf{Z}^d$ have dimension $n = k \cdot m$ and the QR-factorization $[b_1, \ldots, b_n] = QR$. We partition the basis-matrix B into m *segments* $B_l = [b_{k(l-1)+1}, \ldots, b_{kl}]$ for $l = 1, ..., m$. Local reduction of two consecutive segments uses the coefficients of the submatrix $R_l := [r_{kl+i,kl+j}]_{-k<i,j\le k} \in \mathbf{R}^{2k\times 2k}$ of $R \in \mathbf{R}^{d\times n}$, corresponding to two consecutive segments B_{l-1}, B_l. We want to do most of the LLL-exchanges

and the corresponding size-reduction in local coordinates of some R_l. Extra global transformations are required after local LLL-reduction. In order to minimize these global costs we introduce *k-segment reduced bases*. We let $D(l) = \|\widehat{b}_{k(l-1)+1}\|^2 \cdots \|\widehat{b}_{kl}\|^2$ denote the *local Gramian determinant* of segment B_l. We have that $D_{kl} = D(1) \cdots D(l)$.

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

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

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

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

Theorem 4. [KS01] *Let* $b_1, \ldots, b_n$ *be a basis that is k-segment* LLL*-reduced with* δ. *Then we have for* $i = 1, \ldots, n$:

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

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

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

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

Segment LLL

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

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

1. $l := 1$

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

loc-LLL(l)

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

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

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

Theorem 5. [KS01] *For* $k = \Theta(m) = \Theta(\sqrt{n})$ **segment LLL** *performs* $O(nd \log_{1/\delta} M)$ *arithmetic steps using integers of bit length* $O(\log_2 M)$.

7 Scaled Segment LLL-Reduction.

We combine the methods of Sections 4 and 6 to a stable algorithm for segment LLL-reduction. Size-reduction is done against an associated scaled basis. Segment LLL-reduction is guided by the orthogonal vectors of the scaled basis.

Definition 5. *We call a basis $b_1, \ldots, b_n \in \mathbf{Z}^d$ k-segment* sLLL-reduced*, if it is k-segment* LLL*-reduced except that $b_1, ..., b_n$ is size-reduced in a weaker sense — it is size-reduced against an associated basis $b_1^s, ..., b_n^s$ with the properties* **1. 2.**:

1. *The first orthogonal vectors $\widehat{b}^s_{kl-k+1}$ of segments are nearly equally long:*

$$\|b_1^s\| \leq 2^{e_{kl}} \|\widehat{b}_{kl-k+1}\| = \|\widehat{b}^s_{kl-k+1}\| < 2\|b_1^s\| \quad \text{for } l = 1, ..., m. \tag{6}$$

2. *There is a* uniform *scaling factor $2^{e_{kl}}$ for segment B_l so that the coefficients $r_{kl-k+i,kl-k+j}$, for $1 \leq i, j \leq k$ of the R-matrix corresponding to $b_1, ..., b_n$ and those corresponding to $b_1^s, ..., b_n^s$ are proportional with proportionality factor $2^{e_{kl}}$.*

The inequalities for k-segment LLL-reduced bases in Theorem 4 also hold for k-segment sLLL-reduced bases. The weaker size-reduction is not important. In practice k-segment sLLL-reduced bases are nearly as good as LLL-reduced bases.

We sketch a procedure **loc-sLLL**(l) — for local scaled LLL-reduction — which replaces **loc-LLL**(l) within **segment LLL**. The algorithm **segment sLLL** iterates local scaled LLL-reduction of two consecutive segments $[B_{l-1}, B_l] = [b_{kl-k+1}, ..., b_{kl+k}]$.

The procedure **loc-sLLL**(l). Its inputs are a lattice basis, uniform scaling factors $2^{\bar{e}_{k\ell}}$ satisfying $b^s_{k\ell-j} = 2^{\bar{e}_{k\ell}}\, b_{k\ell-j}$ for $j = 0, ..., k-1$, $\ell = 1, ..., l-1$. The Householder vectors $v_1, ..., v_{k(l-1)}$ of $b_1, ..., b_{k(l-1)}$ and the matrix $R \in \mathbf{R}^{d \times k(l-1)}$ for $[b_1^s, ..., b^s_{k(l-1)}]$ are also given. Local LLL-reduction of B_{l-1}, B_l is done in *fpa* via the local matrix $R_l \in \mathbf{R}^{2k \times 2k}$.

The procedure **loc-sLLL**(l)

- computes a uniform scaling factor $2^{\bar{e}_{kl}}$ for the two segments $[B_{l-1}, B_l]$
- computes the local matrix R_l of $2^{\bar{e}_{kl}}[B_{l-1}, B_l]$ via **HRS'**
- performs a local LLL-reduction on $[B_{l-1}, B_l]$ using R_l
- stores the transformation in the matrix $H \in \mathbf{Z}^{2k \times 2k}$.
- Upon termination it transforms $[B_{l-1}, B_l]$ into $[B_{l-1}, B_l]H$ in exact arithmetic.

Restarting **loc-sLLL**(l). Whenever $\|H\|_\infty$ surpasses the threshold 2^{15} the procedure **loc-sLLL**(l) is restarted with the transformed segments $[B_{l-1}, B_l]H$. This is necessary as the norm $\|H\|_\infty$ directly translates into additional *fpa*-errors. As the number of restarts is crucial for the running time we carefully prepare R_l as to prevent an early restart. We show below how to predict the size of the matrix $H \in \mathbf{Z}^{2k \times 2k}$ occuring in the subsequent local LLL-reduction of B_{l-1}, B_l. We slash R_l so that $\|H\|_\infty \leq 2^{15}$ holds on the average. The threshold 2^{15} is for wired 53-bit *fpa*, for 106-bit *fpa* we increase the threshold accordingly.

Computing the matrix R_l and $2^{\bar{e}_{kl}}$. First compute via **HRS** the uniform scaling factor $2^{\bar{e}_{kl}}$ and the first vector $\mathbf{r}_{kl-k+1}$ of B_{l-1} so that $\|b_1^s\| \leq 2^{e_{kl}}\|\widehat{b}_{kl-k+1}\| = \|\widehat{b}^s_{kl-k+1}\| < 2\|b_1^s\|$. With the same scaling factor $2^{\bar{e}_{kl}}$ compute $\mathbf{r}_h$ via **HRS'** for $h = kl - k + 2, ..., kl + k$. For local LLL-reduction we have to size-reduce b_h against $b_{kl-k+1}, \ldots b_{h-1}$. We generalize Step **3.** of **HRS'** accordingly: perform the same size-reduction steps $b_h^s := b_h^s - \lceil \bar{\mu}^s_{h,j} \rfloor b_j^s$, $b_h := b_h - \lceil \bar{\mu}^s_{h,j} \rfloor b_j$ on b_h^s against b_j^s and on b_h against b_j for $j = kl - k + 1, \ldots h - 1$. This implies that the local matrix R_l of segments B_{l-1}, B_l corresponding to $b_1, ..., b_n$ and the R_l corresponding to $b_1^s, ..., b_n^s$ are proportional with the factor $2^{\bar{e}_{kl}}$.

The heuristic for slashing R_l. We let $r^l_{i,j} = r_{kl-k+i,kl-k+j}$ denote the coefficients of the local matrix R_l. Large values $|r^l_{1,1}/r^l_{j,j}|$ have a double negative effect on the stability of **loc-sLLL**(l). By Lemma 1 the orthogonalization of b_j gets inaccurate. Moreover, $\|H\|_\infty$ will be large due to Lemma 2 [KS01]. A detailed argument shows that $\|H\|_\infty$ is expected to be less than 2^{15} if $\max_j |r^l_{1,1}/r^l_{j,j}| \leq 2^{15}$. This suggests to *slash* R_l so that $\max_j |r^l_{1,1}/r^l_{j,j}| \leq 2^{15}$.

Slashing the matrix R_l. Slash all values $|r^l_{j,j}|$ — satisfying $|r^l_{1,1}/r^l_{j,j}| < 2^{-15}$ — to $r^l_{j,j} := |r^l_{1,1}|2^{-15}$. Moreover, set $r^l_{h,i} := 0$ for $h = j, ..., 2k$ and $h < i \leq 2k$.

$$R_l := \begin{bmatrix} r^l_{1,1} \cdots & r^l_{1,j-1} & r^l_{1,j} & \cdots & r^l_{1,k} \\ \ddots & \vdots & \vdots & & \vdots \\ 0 & r^l_{j-1,j-1} & r^l_{j-1,j} & \cdots & r^l_{j-1,k} \\ 0 \ \cdots & 0 & |r^l_{1,1}|2^{-15} & & 0 \\ \vdots & \vdots & & \ddots & \\ 0 \ \cdots & 0 & 0 & & |r^l_{1,1}|2^{-15} \end{bmatrix}$$

We locally LLL-reduce the slashed matrix R_l, and afterwards we transform $[B_{l-1}, B_l]$ into $[B_{l-1}, B_l]H$. If either $\|H\|_\infty > 2^{15}$ or if the slashing did effectively change R_l we restart the local reduction with the transformed segments $[B_{l-1}, B_l]H$. Note that a restart of **loc-sLLL**(l) also adjusts the uniform scaling factor $2^{\bar{e}_{kl}}$. This method of correction works very well for segment size $k \leq 100$.

Segment sLLL
INPUT $\quad b_1, \ldots, b_n \in \mathbf{Z}^d, k, m, n = km,\ \delta$
OUTPUT $\quad b_1, \ldots, b_n \quad k$-segment and scaled reduced basis
1. $l := 1$
2. **while** $l \leq m - 1$ **do**
  loc-sLLL(l)
  if $l \neq 1$ **and**
   ($D(l-1) > (\alpha/\delta)^{k^2} D(l)$ **or** $\delta^{k^2}\|\widehat{b}_{kl}\|^2 > \alpha\|\widehat{b}_{kl+1}\|^2$)
  then $l := l - 1$ **else** $l := l + 1$. *end*

8 Performance of Segment sLLL.

Consider a sample of [GGH]-bases of dimension $n = \nu \cdot 100$ for $\nu = 1, ..., 8$. The vectors of the input basis consist of integers with $n/2$ bits. The algorithm uses 106 bit *fpa* and $\delta = 0.99$ respectively $\delta = 0.999$. All tests have been performed on 350 MHz PC's, the code of **segment sLLL** uses the [NTL] computer algebra library. We present the segment size and the average bit length of the reduced basis vectors in Figure 2.

Lattice	$\delta = 0.99$		$\delta = 0.999$	
dimension	*segment size*	l^{av}	*segment size*	l^{av}
100	25	7.9	25	5.2
200	40	13.8	40	9.4
300	45	23.0	45	14.7
400	50	30.5	50	17.7
500	50	34.3	56	22.6
600	60	38.8	60	27.9
700	60	40.9	65	34.3
800	70	46.6	70	37.0

Fig. 2. **segment sLLL** reduced bases. l^{av} = average bit length of the output integers.

The running time increases considerably as δ approaches 1. The size of the reduced basis decreases accordingly. Figure 3 shows the time of **segment sLLL** for [GGH]-bases.

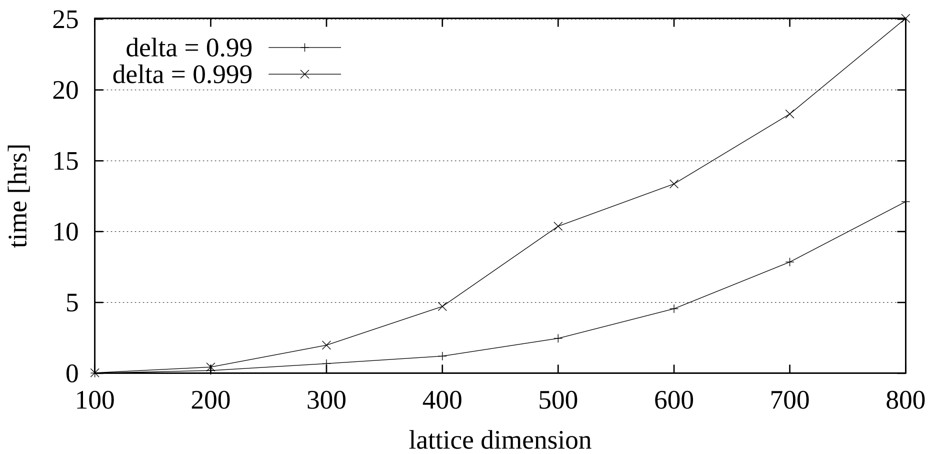

Fig. 3. Running time of **Segment sLLL** using $\delta = 0.99$ resp. $\delta = 0.999$.

Acknowledgement. We thank Bartol Filipović for his help in producing and measuring the code of the new reduction algorithm.

References

[GGH] *O. Goldreich, S. Goldwasser, and S. Halevi*, Public-key cryptosystems from lattice reduction problems. Proc. Crypto'97, LNCS 1294, Springer-Verlag, pp. 112–131, 1997.

[KS01] *H. Koy and C.P. Schnorr*, Segment LLL-Reduction of Lattice Bases. Proceedings CaLC 2001.

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

[LH95] *C.L. Lawson and R.J. Hanson*, Solving Least Square Problems, SIAM, Philadelphia, 1995.

[NTL] NTL homepage: `http://www.shoup.net/ntl/`, 2000.

[RS96] *C. Rössner and C.P. Schnorr*, An optimal stable continued fraction algorithm for arbitrary dimension. 5.-th IPCO, LNCS **1084**, pp. 31–43, Springer-Verlag, 1996.

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

[S88] *C.P. Schnorr*, A more efficient algorithm for lattice basis reduction, *J. Algorithms* **9**, pp. 47–62, 1988.

[SE91] *C.P. Schnorr and M. Euchner*, Lattice Basis Reduction: Improved Practical Algorithms and Solving Subset Sum Problems, *Proc. Fundamentals of Computation Theory '91*, L. Budach, ed., LNCS **529**, Springer-Verlag, pp. 68-85, 1991. (Complete paper in MATHEMATICAL PROGRAMMING STUDIES **66A**, No 2 , pp. 181–199, 1994.)

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