A Fast Variant of the Gaussian Reduction Algorithm

Michael Kaib*
FB Mathematik, Universität Frankfurt, 60054 Frankfurt, Germany

January 1994

Abstract

We propose a fast variant of the Gaussian algorithm for the reduction of two–dimensional lattices for the l_1-, l_2- and $l_\infty-$norm. The algorithm runs in at most $O(n\,\mathcal{M}(B)\,\log B)$ bit operations for the $l_\infty-$norm and in $O(n\,\log n\,\mathcal{M}(B)\,\log B)$ bit operations for the l_1- and l_2-norm on input vectors $a, b \in \mathbb{Z}^n$ with norm at most 2^B where $\mathcal{M}(B)$ is a time bound for B-bit integer multiplication. This generalizes Schönhages monotone Algorithm [Sch91] to the centered case and to various norms.

1 Introduction

The Gaussian algorithm computes a reduced basis for a lattice of rank 2, or, in other terms, a reduced binary quadratic form out of a set of equivalent positive forms [Ga1801]. This algorithm is a natural generalisation of the centered Euclidean algorithm to dimension 2. Lehmer [Le38] gave a fast gcd–algorithm for integers by performing most of the arithmetic on the leading bits. Schönhage [Sch71] modified this method to achieve an asymptotically low bit complexity. Recently Schönhage [Sch91] extended this techniques to a fast reduction algorithm for binary quadratic forms. Yap [Ya92] published an outline of a similar result in the language of lattice basis reduction. This research considered *monotone* reduction algorithms reducing to the smallest nonnegative integers. However the Gaussian algorithm is more efficient with *centered* reduction steps [Va91, Da93]. Our basic idea is, that the core of the Gaussian algorithm operates stable until the approximation error exceeds $\frac{1}{12}$ of the norm of the actual vector. This is valid for arbitrary norms. Recently Kaib and Schnorr gave a sharp worst case analysis for the number of interations of the Gaussian algorithm for arbitrary norms [KS93]. We use their explicit formulas for the transformation matrices in the centered algorithm to bound the approximation errors.

*e-mail: kaib@cs.uni-frankfurt.de

2 Preliminaries

Let $a, b \in \mathbb{R}^n$ be a basis of the lattice $\mathbb{Z}a + \mathbb{Z}b$. We denote by $\| \, . \, \|$ an l_p-norm for $p \in \{1, 2, \infty\}$ on $\mathbb{R}^n$ and by succ $: \mathbb{R}^n \times \mathbb{R}^n \longrightarrow \mathbb{R}^n$ a *Gaussian reduction function*, i.e. a function satisfying

1. $\text{succ}\,(a,b) \;=\; \varepsilon(b - \mu a)$ for some $\mu \in \mathbb{Z},\ \varepsilon = \pm 1\,,$
2. $\| \,\text{succ}\,(a,b)\, \| \;=\; \min_{\mu \in \mathbb{Z}} \;\| \, b - \mu a \, \|\,,$
3. $\| \,\text{succ}\,(a,b) - a\, \| \;\leq\; \| \,\text{succ}\,(a,b) + a\, \| \quad .$

The Gaussian algorithm iterates the reduction function on the input basis:

REPEAT $(a,b) \;:=\; (\text{succ}\,(a,b)\,, a)$ **UNTIL** $\| \, b \, \| \;\leq\; \| \, a - b \, \| \quad .$

We call a single application of the Gaussian reduction function $(a,b) \;:=\; (\text{succ}\,(a,b)\,, a)$ a *Gaussian reduction step* (Gaußstep). Defining a lattice basis (a,b) as

reduced if $\quad \| \, a \, \|\,,\ \| \, b \, \| \;\leq\; \| \, a - b \, \| \;\leq\; \| \, a + b \, \|$ $\quad$ and

well–ordered if $\quad \| \, a \, \| \;\leq\; \| \, a - b \, \| \;<\; \| \, b \, \|\,,$

we obtain the following:

Proposition 1.

1. *Any basis obtained by a Gaussian reduction step is either well–ordered or reduced.*
2. *The algorithm terminates after finitely many iterations.*

For an abstract norm it might be hard to compute the Gaussian reduction function. However, the computation is easy in the interior steps where $(\text{succ}\,(a,b), b)$ is well–ordered. In this case we can compute succ (a,b) in $O(n\mathcal{M}(B))$ bit operations where $a, b \in \mathbb{Z}^n$ with $\max\,(\| \, a \, \|, \| \, b \, \|) \;\leq\; 2^B$ for the l_1-, l_2- and $l_\infty-$norm. For simplicity we restrict the norm to be one of those l_p-norms. Nevertheless the results of this paper hold for any l_p-norm for which we can compute succ (a,b) in $O(n\mathcal{M}(B))$ bit operations.

Kaib and Schnorr recently proved a sharp worst case bound for the number of iterations of the Gaussian algorithm for any norm. They observed that the transformation matrices defined by $(\text{succ}\,(a,b), a) = (a,b)M^{-1}$ are of the form

$$M = \begin{bmatrix} 0 & \varepsilon \\ 1 & \mu \end{bmatrix} \in SL_2(\mathbb{Z}) \quad \text{where either } \varepsilon = 1,\ \mu \geq 2 \text{ or } \varepsilon = -1,\ \mu \geq 3\ .$$

We call such a matrix M a *stepmatrix* and we call a product of stepmatrices a *reduction matrix*. Note that the Gaussian reduction step is performed by the *inverse* of the stepmatrix. The central point in the proof of the sharp bound on the number of iterations is that the following Lemma holds for any norm:

Lemma 2. *Let (α, β) be well–ordered, M a stepmatrix and $(a, b) = (\alpha, \beta)\, M$. Then (a, b) is well–ordered and $(\alpha, \beta) = (\mathrm{succ}\,(a, b), a)$.*

Proof. See [KS93], Lemma 9. □

The goal of this work is to achieve an asymptotically low bit complexity by using approximate arithmetic. We will show that the Gaussian algorithm operates stable until the approximation error exceeds some threshold. Therefore we need a property of the bases that preserves well–orderness if the approximation error is bounded. We define

Definition 3. *A lattice basis (a, b) is called*

1. strictly well–ordered (swo) *if*
   - $\frac{5}{4} \parallel a \parallel \;\leq\; \parallel a - b \parallel \;<\; \parallel b \parallel \;-\; \frac{1}{2} \parallel a \parallel$ *and*
   - $2 \parallel a \parallel \;\leq\; \parallel b \parallel$,
2. σ–minimal *for some threshold $\sigma \in \mathbb{R}$ if (a, b) is strictly well–ordered with $\parallel a \parallel \geq \sigma$ and either*
   - $\parallel \mathrm{succ}\,(a, b) \parallel \;<\; \sigma$ *or*
   - $(\mathrm{succ}\,(a, b), a)$ *is not strictly well–ordered*

where σ-minimality is the appropriate porperty for the output of the fast algorithm: The approximation error is controlled by the threshold σ and we will prove that all bases in the core of the Gaussian algorithm are strictly well–ordered.

Lemma 4. *Let (a, b), (b, c) be well–ordered bases, $a = \mathrm{succ}\,(b, c) = \varepsilon(c - \mu b)$. Let $\Delta < 1$ satisfy $\parallel \mathrm{succ}\,(a, b) \parallel \;\leq\; \Delta \parallel b \parallel$. Then*

1. $\parallel c \parallel \;-\; \parallel c - b \parallel \;\geq\; \parallel b \parallel \;-\; \parallel a \parallel$
2. $\parallel c - b \parallel \;-\; \parallel b \parallel \;\geq\; \frac{1-\Delta}{2} \parallel b \parallel$.

This Lemma implies that in a run of the Gaussian algorithm all except the last three bases are strictly well–ordered:

Corollary 5. *Let (b_0, b_1), (b_1, b_2), (b_2, b_3) be successive well–ordered bases with $b_i = \mathrm{succ}\,(b_{i+1}, b_{i+2})$. Then (b_2, b_3) is strictly well–ordered.*

Proof. Since (b_0, b_1) is well–ordered we have $\parallel b_1 \parallel \;\leq\; \frac{1}{2} \parallel b_2 \parallel$ and thus

$$\parallel b_3 \parallel \;-\; \parallel b_3 - b_2 \parallel \;\geq\; \parallel b_2 \parallel \;-\; \parallel b_1 \parallel \;\geq\; \frac{1}{2} \parallel b_2 \parallel .$$

On the other hand $\parallel b_0 \parallel \;<\; \parallel b_1 \parallel \;\leq\; \parallel b_2 \parallel$ and hence, with $\Delta = \frac{1}{2}$ the Lemma implies

$$\parallel b_3 - b_2 \parallel \;-\; \parallel b_2 \parallel \;\geq\; \frac{1}{4} \parallel b_2 \parallel . \;\square$$

A more careful analysis shows: If (a,b) is swo and $(\mathrm{succ}\,(a,b),a)$ is not swo, we have

$$\| \mathrm{succ}\,(a,b) \| \;\leq\; 2\lambda_2(L_{a,b}).$$

This inequality in particular holds for a 0-minimal basis of an integer lattice.

Proof of Lemma 4. Ad 1: Let $\rho := \| b \| / \| a \|$. We distinguish two cases:

$\underline{\varepsilon = 1}$:

$$\begin{aligned} \| c - b \| &= \| (\mu-1)b + a \| \\ &\leq (\mu-1) \| b \| + \frac{1}{\rho} \| b \| \\ &\leq (\mu - 1 + \frac{1}{\rho})\frac{1}{\mu} \| c \| \\ &= (1 - \frac{\rho-1}{\rho\mu}) \| c \| \end{aligned}$$

Hence

$$\| c \| - \| c - b \| \geq \frac{\rho-1}{\rho\mu} \| c \| \geq \frac{\rho-1}{\rho\mu}\mu \| b \| = \| b \| - \| a \| .$$

$\underline{\varepsilon = -1}$: Let $\eta = \mu - 1 - \frac{1}{\rho}$ and $G(\xi) = (1-\xi)\eta\rho a + \xi\eta b$.

We have $\| G(0) \| = \eta\rho \| a \| = \eta \| b \| = \| G(1) \|$. Hence $\eta\rho \| a \| \leq \| G(\frac{\mu-1}{\eta}) \| = \| c - b \|$. Let $H(\xi) = (1-\xi)(-\eta\rho a) + \xi(c-b)$. We have $\| H(0) \| = \eta\rho \| a \| \leq \| c - b \| = \| H(1) \|$. Hence $\| c - b \| \leq \| H(\frac{\mu\rho}{\mu\rho-1}) \| = (1 - \frac{\rho-1}{\mu\rho-1}) \| c \|$. Hence

$$\| c \| - \| c - b \| \geq \frac{\rho-1}{\mu\rho-1} \| c \| \geq \frac{\rho-1}{\mu\rho-1}(\mu-1) \| b \| \geq \| b \| - \| a \| .$$

Ad 2: Let $x := \mathrm{succ}\,(a,b) = \tilde{\varepsilon}(b - \tilde{\mu}a)$. We use the line $G(\xi) = (1-\xi)\lambda b + \xi c$ where $\lambda = \frac{(2\tilde{\mu}+1)/\Delta - 1}{\tilde{\mu}/\Delta}$. Lemma 11 of [Ka91] yields $G(1) = \| c \| \geq \lambda \| b \| = G(0)$. Hence

$$\frac{\lambda}{\lambda - 1} \| c - b \| = \| G(\tfrac{\lambda}{\lambda-1}) \| \geq \| G(0) \| = \lambda \| b \|$$

and finally

$$\| c - b \| - \| b \| \geq (\lambda - 2) \| b \| \geq \frac{1-\Delta}{2} \| b \| .$$

□

A strictly well–ordered basis remains well–ordered if the approximation error is bounded reasonably:

Lemma 6. *Let (a,b) be strictly well–ordered and $\max\{\| \Delta_a \|, \| \Delta_b \|\} \leq \frac{1}{12} \| a \|$. Then $(a + \Delta_a,\ b + \Delta_b)$ is well–ordered.*

Proof. We immediately see:

$$\| b+\Delta_b \| \ - \ \| a+\Delta_a-(b+\Delta_b) \| \ \geq \ \frac{1}{2} \| a \| \ - \ 2 \| \Delta_a \| \ - \ \| \Delta_b \| \ > \ 0$$

$$\| a+\Delta_a-(b+\Delta_b) \| \ - \ \| a+\Delta_a \| \ \geq \ \frac{1}{4} \| a \| \ - \ 2 \| \Delta_a \| \ - \ \| \Delta_b \| \ \geq \ 0 \ .$$

□

The preceding considerations lead to the following:

Central stability observation. *Denote $b_0, b_1, b_2, \ldots$, where $b_i := \mathrm{succ}\,(b_{i+1}, b_{i+2})$, the tail of the vectors generated by a run of the Gaussian algorithm. Let (b_0, b_1) be reduced, (b_i, b_{i+1}) swo for $i \geq t$ but (b_{t-1}, b_t) not swo. From Corollary 5 we know that $t \leq 3$. Let φ be a linear mapping with $\| \varphi(b_i) - b_i \| \ \leq \ \frac{1}{12} \| b_i \|$ for $i = t, \ t+1$. Let $\bar{b}_i := \varphi(b_i)$ for $i \geq t$ and $\bar{b}_i := \mathrm{succ}\,(\bar{b}_{i+1}, \bar{b}_{i+2})$ for $i < t$.*

Then $(\bar{b}_i, \bar{b}_{i+1})$ is well–ordered for $i \geq t$ and $(\bar{b}_{t+2}, \bar{b}_{t+3})$ is swo. Furthermore the Gaussian algorithm on $(\bar{b}_t, \bar{b}_{t+1})$ terminates after at most 3 Gaußsteps.

3 The fast algorithm

As a consequence of the central stability consideration, throughout the fast algorithm we will care that

- if a basis is well–ordered but not swo, we cancel (at most 2) preceding Gaußsteps to obtain a swo basis,
- if in $(a, b) := (a + \Delta_a, \ b + \Delta_b)$ some precision bits are ingored ore recovered their norm satisfies $\max(\| \Delta_a \|, \| \Delta_b \|) \ \leq \ \frac{1}{12} \| a \|$.

For the first point we need to protocol the transformation matrices $M_1, \ldots, M_t$ for all Gaußsteps. The fast algorithm performs two recursive calls of low accuracy. The full reduction Matrix $M = M_t \cdot \ldots \cdot M_1$ is protocolled for the recovering of the full accuracy. In our notation, the procedures performing and cancelling Gaußsteps are:

Gaußstep: $t := t+1, \quad M_t := \mathrm{Stepmatrix}(\alpha, \beta), \quad (\alpha, \beta) := (\alpha, \beta) M_t^{-1}, \quad M := M_t M.$

Backstep: $(\alpha, \beta) := (\alpha, \beta) M_t \, , \quad M := M_t^{-1} M, \quad t := t-1.$

For approximative arithmetic on the leading bits of vectors we need to generalize some type of "next ineger" function to vectors. Let $\lfloor . \rceil : \mathbb{R}^n \longrightarrow \mathbb{Z}^n$ be a function that minimizes $\| x - \lfloor x \rceil \|$. We denote the worst approximation by

$$\Gamma \ := \ \sup_{x \in \mathbb{R}^n} \ \inf_{\omega \in \mathbb{Z}^n} \ \| x - \omega \|$$

For l_p-norms we have $\Gamma = \frac{1}{2} n^{1/p}$ by taking the next integer in every coordinate. In the fast algorithm Γ will appear in form of the technical constant $z \ = \ \lceil \frac{1}{2} \log_2 \Gamma / \tau - 1 \rceil \ \approx \ \lceil \frac{1}{2p} \log_2 n + 1.31 \rceil$ where $\tau = \sqrt{\frac{13}{12}} - 1$.

Algorithm $FG(a,b,m)$**.**
Input: Well–ordered basis $(a,b)\,,\ m \in \mathbb{N}$.

1. $M := I\,,\ t := 0,\ (\alpha,\beta)\ :=\ (a,b)$
   IF $\parallel a \parallel\ \leq\ 2^{m+z+1}$ **THEN GOTO** 8 .
2. Choose d minimal with $\parallel b \parallel\ \leq\ 2^{m+d}\,,\quad m' := \min(m,d)$,
   $z = \lceil \frac{1}{2p} \log_2 n + 1.31 \rceil\,,\quad h := m' + \lfloor \frac{d+z}{2} \rfloor\,,\quad h' := \lceil m' + z \rceil$
   IF $m \leq d$ **THEN GOTO** 4.
3. $k := m - d + 1,\quad (\alpha,\beta)\ :=\ (\lfloor 2^{-k}\, a \rceil,\ \lfloor 2^{-k}\, b \rceil)\,,$
   IF (α,β) is not swo **THEN** $[\,(\alpha,\beta)\ :=\ (a,b)\,,$ **GOTO** 8 $]\,.$
4. **IF** $\parallel \alpha \parallel\ \geq\ 2^h$ **THEN** $(\alpha,\beta,t,M,M_1,\ldots,M_t)\ :=\ FG(\alpha,\beta,h)\,.$
5. **WHILE** $\parallel \beta \parallel\ >\ 2^h$ and $\parallel \alpha \parallel\ \geq\ 2^{h'}$ and (α,β) swo **DO** Gaußstep
   IF $\parallel \alpha \parallel\ <2^{h'}$ or (α,β) not swo **THEN** [Backstep, **GOTO** 7],
6. $(\alpha,\beta,t',M',M_{t+1},\ldots,M_{t+t'})\ :=\ FG(\alpha,\beta,h')\,,\quad M\ :=\ M'\,M\,,\ t := t + t'\,.$
7. **IF** $m > d$ **THEN** $(\alpha,\beta)\ :=\ (a,b)\,M^{-1}$
8. **WHILE** (α,β) swo and $\parallel a \parallel\ \geq\ 2^m$ **DO** Gaußstep
   WHILE (α,β) not swo **DO** Backstep

Output: $(\alpha,\beta,t,M,M_1,\ldots,M_t)$ where
(α,β) is a 2^m–minimal basis, M is the reduction Matrix $(\alpha,\beta)\,M\ =\ (a,b)$ and
$M_1,\ldots,M_t$ are the stepmatrices for the performed Gaußsteps satisfying $M = M_t \cdot \ldots \cdot M_1\,.$

It is easy to check the output conditions recursively. The general idea for the time bound is that the integer d chosen in Step 2 is a measure for the *descent* of the algorithm. The recursive calls in Steps 4 and 6 have descent less than $d/2$. We state:

Theorem 7. *Algorithm FG terminates after at most* $O(n\,(1+\log\, n^{1/p})\ \mathcal{M}(B)\,\log B)$ *bit operations on an input basis* $a,b \in \mathbb{Z}^n$ *with* $\parallel a \parallel, \parallel b \parallel\ \leq\ 2^B$ *where* $\mathcal{M}(B)$ *denotes a bit complexity bound for* B*-bit integer multiplication.*

The proof will be given in the next section. We remark that for the l_2-norm we can always run algorithm FG with $n = 2$, since we can perform all operations by use of the Gram–matrix $(a,b)^\top(a,b)$ instead of the vectors $a,b \in \mathbb{Z}^n$. The initial and final transformation requires $O(n\,\mathcal{M}(B))$ bit operations. For reduction of bases in other l_p-norms it is helpful in many cases to perform an initial pre–reduction of the basis in the l_2-norm.

4 Proof of the time bound

Notation. Let $(\alpha_3,\beta_3) = (\lfloor 2^{-k}a \rceil, \lfloor 2^{-k}b \rceil) = (2^{-k}a - \Delta_a, 2^{-k}b - \Delta_b)$ denote the trunctated basis obtained from Step 3, let (α_6,β_6) denote the output basis of the recursive call in Step 6 and let (α,β) denote the full–bit basis obtained from Step 7. Let M be the

reduction Matrix satisfying $(\alpha_6, \beta_6) = (\alpha_3, \beta_3)M^{-1}$. Thus we write the recovering of the full bits in Step 7 as

$$\begin{aligned}(\alpha, \beta) &= (a, b)M^{-1} \\ &= 2^k \left[(\alpha_3, \beta_3) + (\Delta_a, \Delta_b)\right] M^{-1} \\ &= 2^k \left[(\alpha_6, \beta_6) + (\Delta_a, \Delta_b)M^{-1}\right] \\ &=: 2^k \left[(\alpha_6, \beta_6) + (\Delta_\alpha, \Delta_\beta)\right] \ .\end{aligned}$$

Bounds on M. To give upper bounds on the size of the error vectors $\Delta_\alpha, \Delta_\beta$ we use properties of the reduction Matrix M, derived from analysis of the generalized continuants for the centered euclidean algorithm. These properties have been proved by Kaib and Schnorr in [KS93]. The Matrix M has the form

$$M = \begin{bmatrix} \varepsilon p & \varepsilon q \\ r & s \end{bmatrix}$$

where $\varepsilon = \pm 1$ is the sign of the last reduction coefficient and p, q, r, s are the positive integral values of the continuant polynomials satistying $s \geq 2r, 2q \geq 4p \geq 0$ and $\varepsilon(ps - qr) = \det M = \pm 1$. We have

$$\| \beta_3 \| \ = \ \| \varepsilon q \alpha_6 + s\beta_6 \| \ \geq \ (s - \frac{q}{2}) \| \beta_6 \| \ \geq \ \frac{4}{3} \, s \| \beta_6 \| \ .$$

Other bounds. We have

$$(\Delta_\alpha, \Delta_\beta) \ = \ (\Delta_a, \Delta_b) \, M^{-1} \ = \ \pm \left(s\Delta_a - r\Delta_b, \varepsilon \left(q\Delta_a - p\Delta_b\right)\right)$$

where the latter inequalities imply $r + s \leq 2\frac{\| \beta_3 \|}{\| \beta_6 \|}$ and $p + q \leq \frac{\| \beta_3 \|}{\| \beta_6 \|}$. By definition of Γ we have $\| \Delta_a \|, \| \Delta_b \| \ \leq \ \Gamma$. Hence $\| \Delta_\alpha \| \ \leq \ 2\frac{\| \beta_3 \|}{\| \beta_6 \|} \Gamma$ and $\| \Delta_\beta \| \ \leq \ \frac{\| \beta_3 \|}{\| \beta_6 \|} \Gamma$. By Step 1 we have $z < d - 2$ and, by Step 2, $\Gamma \leq \tau 2^{2z+2}$ where $\tau = (\sqrt{\frac{13}{12}} - 1)$. Hence

$$\| \beta_3 \| \ \leq \ 2^{-k} \| b \| \ + \ \Gamma \ \leq \ 2^{2d-1} \ + \ \Gamma \ \leq \ (1 + \frac{\tau}{2}) \, 2^{2d-1}.$$

The $2^{h'}$-minimality of (α_6, β_6) implies $2^{h'} \ \leq \ \| \alpha_6 \| \ \leq \ \frac{1}{2} \| \beta_6 \|$. Using the latter inequalities it is easy to pove the final error bounds:

$$\| \Delta_\alpha \| \ \leq \ \frac{1}{12} \| \alpha_6 \| \quad \text{and} \quad \| \Delta_\beta \| \ \leq \ \frac{1}{24} \| \alpha_6 \| \ .$$

Lemma 8. *Algorithm FG performs at most*

- 2 *Gaußsteps in Step 5*
- $z + 3$ *Gaußsteps in Step 8*
- 2 *Backsteps in Step 8*

Proof for Step 5. Every Gaußstep decreases the norm of the vectors at least for a factor 2. Denote (α_5, β_5) the basis upon entry of Step 5 and ω_5 its successor vector. We have $\| \omega_5 \| < 2^h$ since (α_5, β_5) is 2^h–minimal. Hence $\| \beta \| \leq 2^h$ after at most two iterations. □

Proof for Step 8. We distinguish the cases that the algorithm enters Step 8 from Steps 1, 3 or 7. We my assume $m > d$ since otherwise Step 3 and Step 7 is omitted and the output of Step 6 is already 2^{m+z}–minimal.

Step 1: We have $\| \alpha \| < 2^{m+z+1}$. The norm decreases at least by a factor 2 in each iteration. Hence (α, β) is 2^m–minimal after at most $z+1$ iterations.

Step 3: (α_3, β_3) is not swo. We first prove that $\max\{\| \Delta_a \|, \| \Delta_b \|\} \leq \frac{1}{12} \| a \|$:

$$\frac{12\Gamma}{\| a \|} \leq 12\,\tau 2^{2z+2-m-z} < 12\,\tau 2^{z+d-m-z-1} \leq 3\tau < 1$$

(we used $d < m$). Hence (α_3, β_3) is well–ordered. The central stabylity consideration shows that in this case the algorithm takes at most 2 Gaußsteps on (a, b).

Step 7: Let $\omega_6 = \mathrm{succ}\,(\alpha_6, \beta_6) = \varepsilon(\beta_6 - \mu\alpha_6)$. Assume we are in the non–terminal case, i.e. the $2^{h'}$-minimality of (α_6, β_6) implies $\| \omega_6 \| < 2^{h'}$. Since (α_6, β_6) is swo and $\| \Delta_\alpha \| \leq \frac{1}{12} \| \alpha_6 \|$ the same μ, ε satisfy

$$\omega = \mathrm{succ}\,(\alpha, \beta) = \varepsilon(\beta - \mu\alpha) = 2^k[\omega_6 + \varepsilon(\Delta_\beta - \mu\Delta_\alpha)] \,.$$

We use

$$\mu = \frac{\| \beta_6 - \varepsilon\omega_6 \|}{\| \alpha_6 \|} \leq \frac{3}{2}\frac{\| \beta_6 \|}{\| \alpha_6 \|}$$

to bound the norm of the error tails:

$$\begin{aligned} \| \Delta_\beta \| + \mu \| \Delta_\alpha \| &\leq \frac{\| \beta_3 \|}{\| \beta_6 \|}\Gamma + \frac{3 \| \beta_6 \|}{2 \| \alpha_6 \|} 2 \frac{\| \beta_3 \|}{\| \beta_6 \|}\Gamma \\ &\leq \frac{7}{2}\,\Gamma\,\frac{\| \beta_3 \|}{\| \alpha_6 \|} \\ &\leq \frac{7}{2}\,\tau\,2^{2z+2}\,(1+\frac{\tau}{2})\,2^{2d-1}\,\cdot\,2^{-d-z} \\ &= \frac{7}{24}\,2^{d+z}\,, \end{aligned}$$

and hence

$$\| \omega \| \leq 2^k[2^{d+z} + \frac{7}{24}2^{d+z}] = \frac{31}{12}2^{m+z}\,.$$

Hence the algorithm takes at most $z + 1 + \log_2 \frac{31}{12}$ iterations in Step 8. □

Proof of the bit complexity bound. The descent d chosen in Step 2 is the crucial parameter for the time analysis. The descent d_6 of the recursive call in Step 6 is

$$d_6 \;= h - h' \;= \lfloor \frac{d-z}{2} \rfloor \; .$$

For $m > d$ the descent d_4 of the recursive call in Step 4 is

$$d_4 \;= \lceil \log_2 \; \| \; \beta_3 \; \| \; \rceil - h \;\leq\; 2d - 1 + \log_2(1 + \frac{\tau}{2}) - d - \lfloor \frac{d+z}{2} \rfloor \;<\; \frac{d}{2} \; ,$$

and for $m \leq d$

$$d_4 \;=\; m + d - (m + \lfloor \frac{d+z}{2} \rfloor) \;=\; \lfloor \frac{d-z}{2} \rfloor \; .$$

Hence the two recursive calls have less than half descent size. The bit complexity for the other operations is bounded $O(z \; n \; \mathcal{M}(B))$ in the following way: By Lemma 8 Algorithm FG performs at most $\lceil \frac{1}{2p} \log_2 n + 1.307 \rceil + 3$ Gaußsteps. Each Gaußstep takes $O(n \; \mathcal{M}(B))$ bit operations. Note that in Step 7 the tails can be updated very efficiently by computing

$$(\alpha, \beta) \;:=\; 2^k \, (\alpha, \beta) + (a - 2^k \, \lfloor 2^{-k} \, a \rceil, b - 2^k \, \lfloor 2^{-k} \, b \rceil) \, M^{-1} \, .$$

Denote $T(d)$ the number of bit operations required by a call of algorithm FG of descent d and input size $O(d)$. For the important case we have $m > d$ and thus $d < B < 2d$ holds. Hence the recursion yields

$$\begin{aligned} T(d) \;&\leq\; 2T(\frac{d}{2}) + O(z \; n \; \mathcal{M}(d)) \\ &=\; O(z \; n \; \mathcal{M}(d) \; \log \, d) \end{aligned}$$

□

Acknowledgment

I am grateful to Claus Schnorr for proposing the subject and for helpful comments and suggestions.

References

[Da93] H. Daudé: Des fractions continues a la réduction des réseaux: Analyse en moyenne. Thèse de doctorat, Université de Caen 1993.

[Ga1801] C.F. Gauss: Disquisitiones Arithmeticae. Leipzig 1801. German translation: Untersuchungen über die höhere Arithmetik. Springer, Berlin 1889. (reprint: Chelsea, New York, 1981.)

[Ka91] M. Kaib: The Gauß Lattice Basis Reduction Algorithm Succeeds With Any Norm. Proceedings of the FCT'91, Springer Lecture Notes on Computer Science, vol. 529 (1991), pp. 275-286.

[KS93] M. Kaib and C.P. Schnorr: The Generalized Gauß Reduction Algorithm. Technical Report, Universität Frankfurt (1993). To appear in J. Algorithms.

[Le38] D.H. Lehmer: AMM, vol. 45, (1938), pp. 227-233.

[LS92] L. Lovász, H. Scarf: The Generalized Basis Reduction Algorithm. Mathematics of Operations Research, vol. 17, No. 3 (1992), pp. 754-764.

[Sch71] A. Schönhage: Schnelle Berechnung von Kettenbruchentwicklungen. Acta Informatica 1, (1971), pp. 139-144.

[Sch91] A. Schönhage: Fast Reduction and Composition of Binary Quadratic Forms. In: Proc. ISSAC 1991, Ed. S.M. Watt, ACM 1991, pp. 128-133.

[Va91] B. Vallée: Gauss' Algorithm Revisited. Journal of Algorithms 12 (1991), pp. 556-572.

[VF90] B. Vallée, Ph. Flajolet: The Lattice Reduction Algorithm of Gauss: An Average Case Analysis. Proc. 31st IEEE Symposium on Foundations of Computer Science, 1990, pp. 830-842.

[Ya92] C.K. Yap: Fast Unimodular Reduction: Planar Integer Lattices. 33rd IEEE Symposium on Foundations of Computer Science (1992), pp. 437-446.