Improving Convergence and Practicality of Slide-type Reductions*

Jianwei Li** and Michael Walter***

Abstract. The best lattice reduction algorithm known in theory for approximating the Shortest Vector Problem (SVP) over lattices is the slide reduction algorithm (STOC '08 & CRYPTO '20). In this paper, we first improve the running time analysis of computing slide-reduced bases based on potential functions. This analysis applies to a generic slide reduction algorithm that includes (natural variants of) slide reduction and block-Rankin reduction (ANTS '14). We then present a rigorous dynamic analysis of generic slide reduction using techniques originally applied to a variant of BKZ (CRYPTO '11). This provides guarantees on the quality of the current lattice basis during execution. This dynamic analysis not only implies sharper convergence for these algorithms to find a short nonzero vector (rather than a fully reduced basis), but also allows to heuristically model/trace the practical behaviour of slide reduction. Interestingly, this dynamic analysis inspires us to introduce a new slide reduction variant with better time/quality trade-offs. This is confirmed by both our experiments and simulation, which also show that our variant is competitive with state-of-the-art reduction algorithms. To the best of our knowledge, this work is the first attempt of improving the practical performance of slide reduction beyond speeding up the SVP oracle.

Keywords: Lattice Reduction · Slide Reduction · (H)SVP · Dynamical Systems · Gaussian Heuristic.

1 Introduction

Lattices are discrete subgroups of $\mathbb{R}^m$. A lattice $\mathcal{L}$ in $\mathbb{R}^m$ is represented as a set of all integer linear combinations of n linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ in $\mathbb{R}^m$: $\mathcal{L} = \{\sum_{i=1}^n x_i \cdot \mathbf{b}_i,\ x_i \in \mathbb{Z}\}$. The matrix $\mathbf{B} := (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ forms a *basis* of $\mathcal{L}$, and the integer n is the *rank* of $\mathcal{L}$.

A central lattice problem is the *shortest vector problem* (SVP): given a basis of a lattice $\mathcal{L}$ (endowed with the Euclidean norm), SVP is to find a shortest nonzero vector in $\mathcal{L}$. There are two relaxations with factor $\delta \geq 1$:

- The δ-approximate variant of SVP (δ-SVP) is to find a non-zero vector $\mathbf{v}$ in $\mathcal{L}$ such that $\|\mathbf{v}\| \leq \delta \cdot \lambda_1(\mathcal{L})$, where $\lambda_1(\mathcal{L}) := \min_{\mathbf{x} \in \mathcal{L}_{\neq \mathbf{0}}} \|\mathbf{x}\|$ denotes the length of the shortest nonzero vector in $\mathcal{L}$.
- *δ-Hermite SVP* (δ-HSVP) is to find a non-zero vector $\mathbf{v}$ in $\mathcal{L}$ such that $\|\mathbf{v}\| \leq \delta \cdot \mathrm{vol}(\mathcal{L})^{1/n}$, where $\mathrm{vol}(\mathcal{L})$ denotes the *volume* of $\mathcal{L}$.

Solving δ-SVP is hard for any $\delta \leq n^{c/\log\log n}$ with some constant $c > 0$ under reasonable complexity-theoretic assumptions [Ajt98, CN98, Mic01, Kho05, HR07, Mic12]. Lovász [Lov86] showed that any δ-HSVP solver in rank n can be used to efficiently solve δ^2-SVP in rank n. For random lattices $\mathcal{L}$ of rank n, the classical *Gaussian heuristic* claims $\lambda_1(\mathcal{L}) \approx \sqrt{\frac{n}{2\pi e}} \cdot \mathrm{vol}(\mathcal{L})^{1/n}$: hence, any δ-HSVP solver in rank n for $\delta \geq \sqrt{n}$ can possibly be used to solve $(\delta/\sqrt{n})$-SVP in the same rank in practice (see [GN08b, §3.2]). The output quality of a δ-HSVP solver in rank n is typically assessed with the so-called *root Hermite factor* (RHF) $\delta^{1/(n-1)}$ (see, e.g., [ABF+20, ABLR21]).

Many cryptographic primitives base their security on the worst-case hardness of δ-SVP or related lattice problems [Ajt96, Reg05, GPV08, Pei09]. Security estimates of these constructions depend on solving δ-HSVP, typically for $\delta = \mathrm{poly}(n)$ [GN08b, ADPS16].

The standard approach for solving δ-(H)SVP is *lattice reduction*, which finds reduced bases consisting of reasonably short and relatively orthogonal vectors. Lattice reduction has numerous applications in mathematics and computer science, and especially is a key tool in cryptanalysis. So, it is interesting and important to understand its potential and its limitations.

---

* This work was accepted by Journal: *Information and Computation*. A preliminary version of this work was published in *Proceedings of PKC '21* as [Wal21].

** Inria and DIENS, PSL. Email: jianwei.li@inria.fr. This work was partly supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 885394).

*** Zama, France. Email: michael.walter@zama.ai. This work was supported by the European Research Council, ERC consolidator grant (682815 - TOCNeT). Part of this work was done while the second author was visiting the Simons Institute for the Theory of Computing at the University of California, Berkeley, for the Spring 2020 semester on "Lattices: Algorithms, Complexity, and Cryptography".

Blockwise lattice reduction. The most widely used reduction algorithm is the simplest one: LLL [LLL82], which can solve both SVP and HSVP in polynomial time within exponential approximation factors. Such large approximation factors are not good enough for all applications, especially in cryptanalysis. This has led to the development of stronger blockwise reduction algorithms [Sch87, SE94, SH95, GHGKN06, GN08a, MW16, ALNS20, ABF+20, ABLR21], which generalize LLL using a special subroutine parameterized by an additional input parameter – the *block size* k – which controls time/quality trade-offs: the larger k is, the more reduced the output basis, but the running time grows at least exponentially with k.

The execution of blockwise reduction consists of a sequence of tours: each *tour* modifies the basis using a subroutine which solves SVP (or other hard lattice problems) in a lattice of rank $\leq k$. After each tour, all the basis vectors have been examined, and may have been modified.

In practice, BKZ [Sch87, SE94]– the simplest generalization of LLL – provides the best time/quality trade-offs for approximating (H)SVP. Most work in lattice reduction has focused on BKZ, see e.g. [GN08b, HPS11, CN11, Wal15, AWHT16, ADH+19, ABF+20, ABLR21, LN20]. Other blockwise reduction algorithms are known, like slide reduction [GN08a, ALNS20] and DBKZ [MW16], all of which have better proven trade-offs (than BKZ) for approximating HSVP. In particular, slide reduction provides the best proven trade-offs for approximating SVP.

Slide reduction. The focus of this work is slide-type reductions (including slide reduction [GN08a, ALNS20] and, to some degree, its generalization to block-Rankin reduction [LN14]). When it was originally introduced by [GN08a], slide reduction restricts the block size k to divide the lattice rank n. This restriction was lifted in the slide reduction algorithm [ALNS20, Alg. 3]. At a high level, the algorithm [ALNS20, Alg. 3] simply repeats polynomially many slide tours (until no more progress is made) to output a so-called *slide-reduced* basis: given a basis $\mathbf{B}$ for an n-rank lattice where $n = pk + q \geq 2k$ with $0 \leq q < k$, each *slide tour* uses an SVP oracle in rank k to modify the p disjoint (projected) blocks $\mathbf{B}_{[1,k+q]}, \mathbf{B}_{[k+q+1,2k+q]}, \ldots, \mathbf{B}_{[(p-1)k+q+1,pk+q]}$ of the basis (possibly in parallel), then to modify (in a *dual* way) the disjoint blocks $\mathbf{B}_{[2,k+q+1]}, \mathbf{B}_{[k+q+2,2k+q+1]}, \ldots, \mathbf{B}_{[(p-2)k+q+2,(p-1)k+q+1]}$ (possibly in parallel). This results in "primal" and "dual" blocks that overlap by 1 index (or $(k-1)$ indices by duality when $q = 0$).

Its higher-dimensional generalization to block-Rankin reduction [LN14] works similarly, but it solves a more general problem and uses a more general tool. It approximates the *densest sublattice problem* (DSP) [DM13], by relying on an oracle that solves the r-DSP in rank k. (SVP corresponds to 1-DSP.) In this variant, the dual blocks are shifted by r resulting in overlaps of size r (or $(k-r)$ by duality).

Initial experimental evaluations of slide reduction [GN08a, GN08b] found it to be not competitive in practice with BKZ. To the best of our knowledge, so far there has been no research into practical variants of slide reduction and block-Rankin reduction. This is despite the fact that it offers some trivial parallelization on the disjoint blocks. This is not true for other reduction algorithms and in light of the fact that modern SVP solvers are hard to parallelize, could give slide reduction [GN08a, ALNS20] a considerable advantage in practice.

Potential analyses vs. dynamic analyses. Assuming an oracle in fixed block size, the running time of a blockwise reduction algorithm is dominated by the number of tours (or oracle calls).

The classical approach for bounding the number of tours mimics the analysis of LLL [LLL82], based on an always-decreasing potential function. It was also used to analyse slide reduction [GN08a, ALNS20] and block Rankin reduction [LN14]. However, this type of analysis can only upper bound well (say,) $\|\mathbf{b}_1\|$ for the final reduced basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ output by the algorithm, not for the current basis during execution. Furthermore, it seems hard to find a suitable potential function for analysing BKZ and for a long time it was open if the number of oracle calls in BKZ may be polynomially bounded.

Hanrot, Pujol and Stehlé introduced at CRYPTO '11 [HPS11] the use of discrete dynamical systems to analyse a certain variant BKZ' of BKZ and show that one can put a polynomial bound on the number of oracle calls for BKZ' while preserving its output quality. This dynamic analysis provides guarantees on the quality of the current basis during execution. Recently, [HPS11]'s method was adapted and improved in [LN20] to (the genuine) BKZ [Sch87, SE94], together with slightly sharper bounds for both the output quality and the running time. In particular, [LN20] claimed a sufficient number of tours for BKZ independent of the input basis when being LLL-reduced. Interestingly, the time bound claimed in [LN20] for BKZ was stronger than the one proven for slide reduction (and block-Rankin reduction) using the potential function. [HPS11]'s method was also used to analyse the DBKZ algorithm [MW16], and the analysis was further simplified and completed by Neumaier [Neu17]. Hanrot *et al.* made two suggestions:

– **Suggestion 1:** "*It may be possible to obtain a similar bound for the slide reduction algorithm [GN08a] by adapting (their) analysis*" [HPS11, p. 449].

– **Suggestion 2:** "*Parts of the analysis might prove useful to devise reduction algorithms with improved practical time/quality trade-offs*" [HPS11, p. 451].

Our results. This work aims to improve convergence and practicality of slide-type reductions.

First, we present a generic slide reduction (GSR) algorithm in Sect. 3, which captures (natural variants of) slide reduction and block-Rankin reduction. Intuitively, GSR generalises the slide reduction algorithm [ALNS20, Alg. 3] by replacing the SVP-oracle in rank k with a generic (h, r)-reduction oracle in rank k. Here, we say that a projected block (say,) $\mathbf{B}_{[1,k]}$ is (h, r)-*reduced* if $\mathrm{vol}(\mathbf{B}_{[1,r]}) \leq h^{(k-r)r} \cdot \mathrm{vol}(\mathbf{B}_{[1,k]})^{r/k}$: For instance, any r-DSP solver, or even LLL or BKZ can do (h, r)-reductions (w.r.t. different h). The motivation behind introducing GSR is to provide universal time/quality analyses of slide-type reductions, and to reveal more practical variants.

By slightly adapting the potential function used in [ALNS20], we prove that given an LLL-reduced basis $\mathbf{B}_0$ of a lattice $\mathcal{L}$ of rank n where $n = pk + q \geq 2k$ with $0 \leq q < k$ and $k > r \geq 1$, one can compute a so-called generalized slide reduced basis within at most $O\left(\frac{n^3}{k\varepsilon}\right)$ GSR tours for a slack factor $\varepsilon \in (0, 1]$. This improves the previous number of tours $O\left(\frac{n^2 \log \|\mathbf{B}_0\|}{k\varepsilon}\right)$ claimed for slide reduction [GN08a, ALNS20] and block Rankin reduction [LN14]. This is because the term $\log \|\mathbf{B}_0\|$ might be arbitrarily large: e.g., $\mathbf{D} = \mathrm{Diag}(\frac{1}{n^{n^6}}, 1, \ldots, 1, n^{n^6}) \in \mathbb{R}^{n \times n}$ is LLL-reduced, but $\log \|\mathbf{D}\| = n^6 \log n$.

In particular, with $O\left(\frac{n^3}{k\varepsilon}\right)$ tours, [ALNS20, Alg. 3] can output a slide-reduced basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$, which approximates (H)SVP as follows:

$$\|\mathbf{b}_1\| \leq ((1+\varepsilon)^2 \gamma_k)^{\frac{n-1}{2(k-1)}} \mathrm{vol}(\mathcal{L})^{1/n} \quad \text{and} \quad \|\mathbf{b}_1\| \leq (1+\varepsilon)((1+\varepsilon)^2 \gamma_k)^{\frac{n-k}{k-1}} \lambda_1(\mathcal{L}).$$

Here, γ_k denotes as usual Hermite's constant.

Second, we adapt [HPS11]'s dynamic analysis for BKZ' to our GSR in Sect. 4 and deduce a sharper polynomial bound on the number of oracle calls for GSR while preserving the similar output quality: with $O\left(\frac{n^2 \ln \frac{n}{\varepsilon}}{(k-r)r}\right)$ GSR tours on an LLL-reduced input basis, one can output a basis $\mathbf{B}$ with small $\mathrm{vol}(\mathbf{B}_{[1,i]})$ for all indices $i \in \{r, k+q, \ldots, (p-1)k+q\}$ (rather than a fully reduced basis). For instance, this implies that with $O\left(\frac{n^3 \ln \frac{n}{\varepsilon}}{k^2}\right)$ oracle calls, [GN08a, Alg. 1] can output a basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ such that

$$\|\mathbf{b}_1\| \leq (1+\varepsilon) \gamma_k^{\frac{n-1}{2(k-1)}} \mathrm{vol}(\mathcal{L})^{1/n}.$$

This not only confirms [HPS11]'s **Suggestion 1**, but also provides a better dependence of time/quality trade-offs on ε than that achieved by the potential analyses.

Similar improvements for the block-Rankin reduction algorithm [LN14] are notable if one is interested in computing a basis with small $\mathrm{vol}(\mathbf{B}_{[1,r]})$ (see Cor. 1 vs. Cor. 3).

Third, we introduce in Sect. 5 a cheaper analogue of the block-Rankin-reduction algorithm by replacing its expensive r-DSP oracle in rank k with any HKZ solver in rank k. We call this particular instantiation of GSR *HKZ-slide reduction*, which is inspired by the following three observations:

– **Observation 1:** It was observed in [ADH+19] that modern SVP solvers do not only find the shortest vectors, but also approximately HKZ-reduce the head of the basis essentially for free.
– **Observation 2:** HKZ-reduction approximates r-DSP quite well, especially for small $r = O(1)$ (guaranteed by [HS07, Lem. 3]).
– **Observation 3:** Experiments suggest that part of BKZ's efficiency in practice seems to be the fact that the blocks overlap maximally.

Compared with slide reduction, increasing the overlap in HKZ-slide reduction decreases the number of tours at the cost of slightly increasing the length of the short vector found. All of our arguments (Cor. 5.2 and Sect. 5.1), experimental evidence (Sect. 5.2) and heuristic simulation (Sect. 6.2) demonstrate that this time/quality trade-off can be very favorable in practice. This somewhat confirms [HPS11]'s **Suggestion 2**. A well chosen overlap yields a variant of slide reduction that we consider competitive with the state-of-the-art BKZ [ADH+19]. When interpreting this result, it should be kept in mind that we did not explore all options to fine-tune the oracle to our algorithm and that BKZ has received considerable research effort to arrive at the performance level it is at now. This is not the case for slide reduction.

Finally, thanks to the dynamic analysis of GSR, we propose in Sect. 6 a simple and efficient simulator to heuristically model/trace the execution of (HKZ-)slide reduction with high block size. It can estimate both the output quality and the number of tours, once given parameters (k, r) for HKZ-slide reduction. This further highlights the usefulness of [HPS11]'s dynamical systems technique.

In summary, we believe that our work is a step towards better understanding and improving the practical performance of slide reduction beyond speeding up the SVP oracle.

Technical overview. We first sketch [HPS11]'s dynamic analysis for BKZ'. [HPS11] used the profile function $\mathcal{P}(\mathbf{B}) := (\log \mathrm{vol}(\mathbf{B}_{[1,1]}), \log \mathrm{vol}(\mathbf{B}_{[1,2]})^{1/2}, \ldots, \log \mathrm{vol}(\mathbf{B}_{[1,n]})^{1/n})^{\mathrm{T}}$ to study the quality of the current basis $\mathbf{B}$ at the end of each tour, rather than just at the end of the algorithm.

Let $\mathbf{B}_\ell$ denote the basis at the end of the ℓ-th tour, and let $\mathbf{B}_0$ be the input basis. [HPS11] built a matrix $M \in \mathbb{R}^{n\times n}$ and a vector $\mathbf{v} \in \mathbb{R}^n$ such that component-wise:

$$\mathcal{P}(\mathbf{B}_\ell) \leq M \cdot \mathcal{P}(\mathbf{B}_{\ell-1}) + \mathbf{v}. \tag{1}$$

[HPS11] then analyzes a discrete-time dynamical system $\mathbf{x} \leftarrow \mathbf{x}M + \mathbf{v}$. Its fixed point(s) and speed of convergence encode information on the output quality and runtime of BKZ', respectively. More specifically, if the dynamical system $\mathbf{x} \leftarrow M \cdot \mathbf{x} + \mathbf{v}$ has a fixed point $\mathbf{y} \in \mathbb{R}^n$, then Eq. (1) can be rewritten as

$$\mathcal{P}(\mathbf{B}_\ell) - \mathbf{y} \leq M \cdot (\mathcal{P}(\mathbf{B}_{\ell-1}) - \mathbf{y}).$$

If all the entries of M are ≥ 0, this implies $\mathcal{P}(\mathbf{B}_\ell) - \mathbf{y} \leq M^\ell \cdot (\mathcal{P}(\mathbf{B}_0) - \mathbf{y})$. [HPS11] proved that M has spectral norm $\|M\| < 1$. Then $M^\ell \cdot (\mathcal{P}(\mathbf{B}_0) - \mathbf{y})$ converges to zero with an explicit vectorial upper bound: for any sufficiently large ℓ, we roughly have $\mathcal{P}(\mathbf{B}_\ell) \leq \mathbf{y}$, i.e., the quality of $\mathbf{B}_\ell$ is guaranteed and upper bounded by the fixed point $\mathbf{y}$.

In order to make [HPS11]'s dynamic analysis work for GSR, we make the following three tweaks:

- We study the quantity $\log \frac{\mathrm{vol}(\mathbf{B}_{[1,j]})}{\mathrm{vol}(\mathbf{B})^{j/n}}$ for pivot indices $j \in \{ik + q : i = 1, \ldots, p-1\}$ (rather than for all indices $j \in \{1, \ldots, n\}$). Here, the input lattice to GSR has rank $n = pk + q$ w.r.t. block size k.
- To ensure the convergence of our dynamical system, we multiply each $\log \frac{\mathrm{vol}(\mathbf{B}_{[1,j]})}{\mathrm{vol}(\mathbf{B})^{j/n}}$ by calibration factor $\frac{1}{(n-j)j}$, which eventually results in our profile function $\mathcal{G}(\mathbf{B})$:

$$\mathcal{G}(\mathbf{B}) = (\mathcal{G}_1(\mathbf{B}), \mathcal{G}_2(\mathbf{B}), \ldots, \mathcal{G}_{p-1}(\mathbf{B}))^{\mathrm{T}} \in \mathbb{R}^{p-1} \quad \text{with each} \quad \mathcal{G}_i(\mathbf{B}) = \log \left(\frac{\mathrm{vol}(\mathbf{B}_{[1,ik+q]})}{\mathrm{vol}(\mathbf{B})^{(ik+q)/n}} \right)^{\frac{1}{(ik+q)(n-ik-q)}}.$$

  We deduce that our dynamical system for GSR is $\mathbf{x} \leftarrow \mathbf{A}\mathbf{x} + \mathbf{b}$ with unique fixed point $\overline{\mathbf{y}} \in \mathbb{R}^{p-1}$, where $\mathbf{A} \in \mathbb{R}^{(p-1)\times(p-1)}$ is a tridiagonal matrix and $\mathbf{b} \in \mathbb{R}^{p-1}$ is a vector.
- Instead of using the spectral norm, it is more convenient for us to analyse the row sum norm $\|A\|_\infty < 1$, which implies the convergence of our dynamical system.

As a result, similarly to [HPS11]'s analysis, we roughly have $\mathcal{G}(\mathbf{B}_\ell) \leq \overline{\mathbf{y}}$ for any sufficiently large ℓ. In particular, we can upper bound $\frac{\mathrm{vol}((\mathbf{B}_\ell)_{[1,k+q]})}{\mathrm{vol}(\mathbf{B}_\ell)^{(k+q)/n}}$ w.r.t. the fixed point $\overline{\mathbf{y}}$. One more reduction on $(\mathbf{B}_\ell)_{[1,k+q]}$ allows us to measure the quality of (say,) its first basis vector.

2 Preliminaries

We denote column vectors $\mathbf{x} \in \mathbb{R}^m$ by bold lower-case letters. Matrices $\mathbf{B} \in \mathbb{R}^{m\times n}$ are denoted by bold upper-case letters, and we often think of a matrix as a list of column vectors. We often implicitly assume that $m \geq n$ and that a basis matrix $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n) \in \mathbb{R}^{m\times n}$ has (column) rank n, and write $\mathcal{L}(\mathbf{B}) := \{z_1\mathbf{b}_1 + \cdots + z_n\mathbf{b}_n \ : \ z_i \in \mathbb{Z}\}$ for the lattice generated by $\mathbf{B}$ and $\|\mathbf{B}\| = \max\{\|\mathbf{b}_1\|, \ldots, \|\mathbf{b}_n\|\}$ for the maximum norm of a column.

For an $n \times n$ matrix M, we write $M \geq 0$ if all the entries of M are ≥ 0. For two vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$, we write $\mathbf{u} \leq \mathbf{v}$ if the inequalities hold componentwise. We will use the key elementary property: if $\mathbf{u} \leq \mathbf{v}$ and $M \geq 0$, then $M\mathbf{u} \leq M\mathbf{v}$.

We use the notations $\log(\cdot) := \log_2(\cdot)$, $\ln(\cdot) := \log_e(\cdot)$ and $[a, b]_{\mathbb{Z}} := [a, b] \bigcap \mathbb{Z}$ for any integers a and b.

2.1 Lattices

Let $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n) \in \mathbb{R}^{m\times n}$ be a basis of a lattice $\mathcal{L}$ and $1 \leq r < n$.

GSO. Lattice algorithms often use the orthogonal projections $\pi_i : \mathbb{R}^m \mapsto \mathrm{span}(\mathbf{b}_1, \ldots, \mathbf{b}_{i-1})^\perp$ for $i = 1, \ldots, n$. The *Gram–Schmidt orthogonalisation* (GSO) of $\mathbf{B}$ is $\mathbf{B}^* = (\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*)$, where the Gram–Schmidt vector $\mathbf{b}_i^*$ is $\pi_i(\mathbf{b}_i)$. Then $\mathbf{b}_1^* = \mathbf{b}_1$ and $\mathbf{b}_i^* = \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{i,j} \cdot \mathbf{b}_j^*$ for $i = 2, \ldots, n$, where $\mu_{i,j} = \frac{\langle \mathbf{b}_i, \mathbf{b}_j^* \rangle}{\langle \mathbf{b}_j^*, \mathbf{b}_j^* \rangle}$. The projected block $(\pi_i(\mathbf{b}_i), \pi_i(\mathbf{b}_{i+1}), \ldots, \pi_i(\mathbf{b}_j))$ is denoted by $\mathbf{B}_{[i,j]}$. Then the volume of the parallelepiped generated by $\mathbf{B}_{[i,j]}$ is $\mathrm{vol}(\mathbf{B}_{[i,j]}) = \prod_{\ell=i}^{j} \|\mathbf{b}_\ell^*\|$. In particular, $\mathbf{B}_{[1,j]} = (\mathbf{b}_1, \ldots, \mathbf{b}_j)$ and $\mathrm{vol}(\mathcal{L}) = \mathrm{vol}(\mathbf{B}) = \prod_{\ell=1}^{n} \|\mathbf{b}_\ell^*\|$. We will use the basic fact: $\mathrm{vol}(\mathbf{B}_{[i,z]}) = \mathrm{vol}(\mathbf{B}_{[i,j]}) \cdot \mathrm{vol}(\mathbf{B}_{[j+1,z]})$ for any integers $1 \leq i \leq j < z \leq n$.

Hermite's constant. *Hermite's constant* of dimension n is the maximum $\gamma_n = \max\left(\lambda_1(\Lambda)/\mathrm{vol}(\Lambda)^{1/n}\right)^2$ over all n-rank lattices Λ, where $\lambda_1(\Lambda) = \min_{\mathbf{v}\in\Lambda\setminus\{\mathbf{0}\}}\|\mathbf{v}\|$ is the *first minimum* of Λ. The exact value of γ_n is known for $1 \le n \le 8$ and $n = 24$. It asymptotically satisfies [CS98, MH98]: $\frac{n}{2\pi e} + \frac{\log(\pi n)}{2\pi e} \le \gamma_n \le \frac{1.744n}{2\pi e} + o(n)$.

Rankin's constant. We will use the following generalization of the first minimum $\lambda_1(\mathcal{L})$:

$$m_r(\mathcal{L}) := \min_{\substack{\mathbf{x}_1,\ldots,\mathbf{x}_r\in\mathcal{L}\\ \mathrm{vol}(\mathbf{x}_1,\ldots,\mathbf{x}_r)\neq 0}} \mathrm{vol}(\mathbf{x}_1,\ldots,\mathbf{x}_r).$$

Rankin's constant [Ran53] is $\gamma_{n,r} = \max\left(m_r(\Lambda)/\mathrm{vol}(\Lambda)^{r/n}\right)^2$ over all n-rank lattices Λ. Clearly, $\gamma_{n,1} = \gamma_n$.

DSP. The *r-densest sublattice problem* of $\mathcal{L}$ (r-DSP or (r,n)-DSP) is to find r linearly independent vectors $\mathbf{v}_1,\ldots,\mathbf{v}_r$ in $\mathcal{L}$ such that $\mathrm{vol}(\mathbf{v}_1,\ldots,\mathbf{v}_r) = m_r(\mathcal{L})$ [DM13]. Dadush and Micciancio [DM13] showed that r-DSP is at least as hard as SVP and how to solve (r,n)-DSP exactly with $r^{O(r\cdot n)}$ time and $2^n\mathrm{poly}(n)$ space.

Duality. The *dual basis* of $\mathbf{B}$ is $\mathbf{B}^\times := \mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}$. The *reversed dual basis* of $\mathbf{B}$ is $\mathbf{B}^{-s} := R_m\mathbf{B}^\times R_n$ [GHGN06], where $R_n = (r_{i,j}) \in \mathbb{Z}^{n\times n}$ is the reversed identity matrix: $r_{i,j} = 1$ if $i + j = n + 1$ and $r_{i,j} = 0$ otherwise. In lattice reduction, it is more convenient to consider $\mathbf{B}^{-s}$ than to consider $\mathbf{B}^\times$ [GN08a, LN14]. For instance, if the GSO of $\mathbf{B}^{-s}$ is $(\mathbf{d}_1^*,\ldots,\mathbf{d}_n^*)$, then $\|\mathbf{b}_i^*\|\cdot\|\mathbf{d}_{n-i+1}^*\| = 1$ for $i = 1,\ldots,n$ [Reg04, Claim 7].

Lattice reduction. $\mathbf{B}$ is *size-reduced* if $|\mu_{i,j}| \le \frac{1}{2}$ for all $1 \le j < i \le n$. $\mathbf{B}$ is *LLL-reduced* [LLL82] if it is size-reduced and every 2-rank projected block $\mathbf{B}_{[i,i+1]}$ satisfies Lovász's condition: $\frac{3}{4}\cdot\|\mathbf{b}_i^*\|^2 \le \|\mu_{i+1,i}\cdot\mathbf{b}_i^* + \mathbf{b}_{i+1}^*\|^2$ for $1 \le i < n$. In practice, the parameter $\frac{3}{4}$ can be replaced with any constant in the interval $(\frac{1}{4}, 1)$.

$\mathbf{B}$ is *SVP-reduced* if $\|\mathbf{b}_1\| = \lambda_1(\mathcal{L})$. There are two relaxations with $\delta \ge 1$: $\mathbf{B}$ is *δ-SVP-reduced* if $\|\mathbf{b}_1\| \le \delta\cdot\lambda_1(\mathcal{L})$; $\mathbf{B}$ is *δ-HSVP-reduced* if $\|\mathbf{b}_1\| \le \delta\cdot\mathrm{vol}(\mathcal{L})^{1/n}$.

$\mathbf{B}$ is *δ-DSVP-reduced* [GN08a] (where D stands for dual) if the reversed dual basis $\mathbf{B}^{-s}$ is δ-SVP-reduced. Similarly, we say that $\mathbf{B}$ is *δ-DHSVP-reduced* if $\mathbf{B}^{-s}$ is δ-HSVP-reduced, that is, $\mathrm{vol}(\mathbf{B})^{1/n} \le \delta\cdot\|\mathbf{b}_n^*\|$.

$\mathbf{B}$ is *HKZ-reduced* if it is size-reduced and $\mathbf{B}_{[i,n]}$ is SVP-reduced for $i = 1,\ldots,n$.

$\mathbf{B}$ is *k-BKZ-reduced* [Sch87] if it is size-reduced and $\mathbf{B}_{[i,\min\{i+k-1,n\}]}$ is SVP-reduced for $i = 1,\ldots,n$.

Gaussian heuristic. If $\mathcal{L}$ is a random lattice (cf. [CN11, §2] for the definition), the classical *Gaussian heuristic* claims $\lambda_1(\mathcal{L}) \approx \mathrm{GH}(n)\cdot\mathrm{vol}(\mathcal{L})^{1/n}$. Here, $\mathrm{GH}(n)$ denotes the radius of the unit-volume n-dimensional ball: $\mathrm{GH}(n) = \frac{\Gamma(n/2+1)^{1/n}}{\sqrt{\pi}} \approx \sqrt{\frac{n}{2\pi e}}\cdot(\pi n)^{\frac{1}{2n}}$. It is known that $\mathrm{GH}(n) < \sqrt{\gamma_n} \le \sqrt{2}\cdot\mathrm{GH}(n)$ if $n > 24$ [Bli14].

Recently, Li and Nguyen [LN20, §4.1] established the following weak version of the Gaussian heuristic: for any $\xi \in (0, \sqrt{2}-1]$, there exists $N > 0$ such that if Λ is a random lattice of rank $n \ge N$, then with high probability it satisfies

$$\lambda_1(\Lambda) \le (1+\xi)\,\mathrm{GH}(n)\cdot\mathrm{vol}(\Lambda)^{1/n}.$$

2.2 The ALNS slide reduction algorithm for $n \ge 2k$

We recall slide reduction and its algorithm for lattices with any rank $n \ge 2k$ presented in [ALNS20]. It uses the notion below: A basis $\mathbf{C}$ of rank d is *δ-twin-reduced* [ALNS20] if $\mathbf{C}_{[1,d-1]}$ is *δ-HSVP-reduced* and $\mathbf{C}_{[2,d]}$ is *δ-DHSVP-reduced*.

Definition 1 ([ALNS20, Def. 2]). *Let n, k, p, q be integers s.t. $n = pk + q \ge 2k$ with $0 \le q < k$ and $k \ge 2$, and let $\delta \ge 1$. A basis $\mathbf{B} \in \mathbb{R}^{m\times n}$ is* (δ, k)-slide-reduced *if it is size-reduced and satisfies the conditions below:*

1. *Twin condition: The block $\mathbf{B}_{[1,k+q+1]}$ is $(\delta^2\gamma_k)^{\frac{k+q-1}{2(k-1)}}$-twin-reduced.*
2. *Primal conditions: for all $i \in [1, p-1]_{\mathbb{Z}}$, the block $\mathbf{B}_{[ik+q+1,(i+1)k+q]}$ is δ-SVP-reduced.*
3. *Dual conditions: for all $i \in [1, p-2]_{\mathbb{Z}}$, the block $\mathbf{B}_{[ik+q+2,(i+1)k+q+1]}$ is δ-DSVP-reduced.*[1]

If $q = 0$, it is essentially the original slide reduction (see [GN08a, Def. 1]). If $q > 0$, we can use any δ-SVP oracle in rank k to efficiently realize the twin condition, thanks to the DBKZ algorithm [MW16] (see Alg. 4 revisited in App. A). By combining Alg. 4 with the original slide reduction algorithm [GN08a, Alg. 1], Alg. 1 computes slide-reduced bases by repeating polynomially many slide tours until no more progress is made. Here, a *slide tour* refers to a single execution of Steps 2-12 and uses a δ-SVP oracle in rank k.

As is common in the standard potential analyses of lattice reduction algorithms [LLL82, GN08a, LN14], using the integrality of the input lattice and the integral potential $P(\mathbf{B}) := \prod_{i=1}^{p-1}\mathrm{vol}(\mathbf{B}_{[1,ik+q]})^2 \in \mathbb{Z}^+$, the authors showed the correctness and efficiency of Alg. 1:

[1] When $p = 2$, there are simply no dual conditions.

Algorithm 1 The slide-reduction algorithm for $n \geq 2k$ [ALNS20, Alg. 3]

Input: Block size $k \geq 2$, slack $\varepsilon > 0$, approximation factor $\delta \geq 1$, a basis $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n) \in \mathbb{Z}^{m \times n}$ of a lattice $\mathcal{L}$ of rank $n = pk + q \geq 2k$ for $0 \leq q < k$, and access to a δ-SVP oracle for lattices with rank k.
Output: A $((1+\varepsilon)\delta, k)$-slide-reduced basis of $\mathcal{L}(\mathbf{B})$.

while $\mathrm{vol}(\mathbf{B}_{[1,ik+q]})^2$ is modified by the loop for some $i \in [1, p-1]_{\mathbb{Z}}$ **do**
  $(1+\varepsilon)\eta$-HSVP-reduce $\mathbf{B}_{[1,k+q]}$ using Alg. 4 for $\eta := (\delta^2 \gamma_k)^{\frac{k+q-1}{2(k-1)}}$.
  for $i = 1$ **to** $p-1$ **do**
    δ-SVP-reduce $\mathbf{B}_{[ik+q+1,(i+1)k+q]}$.
  end for
  if $\mathbf{B}_{[2,k+q+1]}$ is not $(1+\varepsilon)\eta$-DHSVP-reduced, **then**
    $(1+\varepsilon)^{1/2}\eta$-DHSVP-reduce $\mathbf{B}_{[2,k+q+1]}$ using Alg. 4.
  end if
  for $i = 1$ **to** $p-2$ **do**
    Find a new basis $\mathbf{C} := (\mathbf{b}_1, \ldots, \mathbf{b}_{ik+q+1}, \mathbf{c}_{ik+q+2}, \ldots, \mathbf{c}_{(i+1)k+q+1}, \mathbf{b}_{ik+q+2}, \ldots, \mathbf{b}_n)$ of $\mathcal{L}$ by δ-DSVP-reducing $\mathbf{B}_{[ik+q+2,(i+1)k+q+1]}$.
    if $(1+\varepsilon)\|\mathbf{b}^*_{(i+1)k+q+1}\| < \|\mathbf{c}^*_{(i+1)k+q+1}\|$ **then** $\mathbf{B} \leftarrow \mathbf{C}$.
  end for
end while
return $\mathbf{B}$.

Theorem 1 ([ALNS20, Th. 8]). *For $\varepsilon \in (0,1]$, given as input a basis $\mathbf{B}_0 \in \mathbb{Z}^{m \times n}$ of a lattice $\mathcal{L}$ of rank $n = pk + q \geq 2k$ for $0 \leq q < k$, Alg. 1 makes at most $\left\lceil \frac{(p-1)(n+q)\cdot \log \|\mathbf{B}_0\|}{\log(1+\varepsilon)} \right\rceil$ $\left(\in O\left(\frac{n^2 \cdot \log \|\mathbf{B}_0\|}{k\varepsilon}\right)\right)$ slide tours, and outputs a $((1+\varepsilon)\delta, k)$-slide-reduced basis of $\mathcal{L}$.*

Remark 1. The potential $\mathcal{D}_{\mathbf{B}}$ used by Schnorr [Sch21] in the complexity analysis of the original slide reduction [GN08a] with blocksize $k = \frac{n}{p} \in \mathbb{Z}$ is indeed $\mathcal{D}_{\mathbf{B}} = \frac{\prod_{i=1}^{p-1} \mathrm{vol}(\mathbf{B}_{[1,ik]})^2}{\mathrm{vol}(\mathbf{B})^{p-1}}$. We mention that $\mathcal{D}_{\mathbf{B}} \leq 1$ (cf. [Sch21, Th. 6.8]) is not enough to guarantee slide-reducedness. For instance, the LLL-reduced basis $\mathbf{C} = \mathrm{Diag}(1, 1, 1, \sqrt{\frac{4}{3}}, 1, 1) \in \mathbb{R}^{6\times 6}$ satisfies $\mathcal{D}_{\mathbf{C}} \leq 1$ w.r.t. blocksize 3, but is not slide-reduced in the sense of [GN08a, Def. 1], because $\mathbf{C}_{[4,6]}$ is not SVP/HKZ-reduced.

3 Generic slide reduction

In this section, we present a generic slide reduction algorithm, of which all of [GN08a, Alg. 1], [LN14, Alg. 3] and Alg. 1 are only particular instantiations. We aim to provide universal time/quality analyses of slide-type reductions: it allows to better understand which properties of slide reduction [GN08a, ALNS20] are essential for approximating HSVP, and to modify slide reduction while preserving its most important properties.

3.1 Definition and properties

In order to present our generic slide reduction, we first generalize the HSVP reduction, DHSVP reduction and the twin reduction used in slide reduction [ALNS20]. These generalizations are straightforward:

Definition 2. *Let $\mathbf{B}$ be a basis of rank n where $n > r \geq 1$, and let $h \geq 1$.*
- $\mathbf{B}$ *is* (h,r)-reduced *if* $\mathrm{vol}(\mathbf{B}_{[1,r]}) \leq h^{(n-r)r} \cdot \mathrm{vol}(\mathbf{B})^{r/n}$*;*
- $\mathbf{B}$ *is* (h,r)-dual-reduced *if the reversed dual basis* $\mathbf{B}^{-s}$ *is* (h,r)*-reduced;*
- $\mathbf{B}$ *is* (h,r)-twin-reduced *if* $\mathbf{B}_{[1,n-r]}$ *is* (h,r)*-reduced and* $\mathbf{B}_{[r+1,n]}$ *is* (h,r)*-dual-reduced. (Here, we assume* $n > 2r$*.)*

A typical reduced basis $\mathbf{B}$ is usually (h,r)-reduced for some constant h and some appropriate indices r, both of which depend on the lattice reduction algorithm. For example, if $\mathbf{B}$ is LLL-reduced, it follows from [PT08, Eq. (3)] that for any index $r \in [1, n-1]_{\mathbb{Z}}$, we have $\mathrm{vol}(\mathbf{B}_{[1,r]}) \leq 2^{(n-r)r/4} \cdot \mathrm{vol}(\mathbf{B})^{r/n}$ and hence $\mathbf{B}$ is $(2^{1/4}, r)$-reduced.

Consider a projected block $\mathbf{B}_{[i,j]}$ of rank $d = j - i + 1 > r$. If $\mathbf{B}_{[i,j]}$ is (h,r)-reduced, this implies:

$$\mathrm{vol}(\mathbf{B}_{[i,j]})^{1/d} \leq h^r \cdot \mathrm{vol}(\mathbf{B}_{[i+r+1,j]})^{1/(d-r)}.$$

If $\mathbf{B}_{[i,j]}$ is (h,r)-dual-reduced, then $\mathrm{vol}(\mathbf{B}_{[i,j]})^{r/d} \leq h^{(d-r)r} \cdot \mathrm{vol}(\mathbf{B}_{[j-r+1,j]})$, which has two equivalent forms:

$$\mathrm{vol}(\mathbf{B}_{[i,j-r]})^{1/(d-r)} \leq h^d \cdot \mathrm{vol}(\mathbf{B}_{[j-r+1,j]})^{1/r} \text{ and } \mathrm{vol}(\mathbf{B}_{[i,j-r]}) \leq h^{(d-r)r} \cdot \mathrm{vol}(\mathbf{B}_{[i,j]})^{(d-r)/d}.$$

Definition 3 (Generic slide reduction). *Let n, k, p, q, r be integers such that $n = pk + q \geq 2k$ with $0 \leq q < k$ and $k > r \geq 1$, and let $h \geq 1$. A basis $\mathbf{B} \in \mathbb{R}^{m \times n}$ is* (h, k, r)-slide-reduced *if it is size-reduced and satisfies the following three sets of conditions.*

1. *Twin condition: The block $\mathbf{B}_{[1,k+q+r]}$ is (h, r)-twin-reduced.*
2. *Primal conditions: for all $i \in [1, p-1]_{\mathbb{Z}}$, the block $\mathbf{B}_{[ik+q+1,(i+1)k+q]}$ is (h, r)-reduced.*
3. *Dual conditions: for all $i \in [1, p-2]_{\mathbb{Z}}$, the block $\mathbf{B}_{[ik+q+r+1,(i+1)k+q+r]}$ is (h, r)-dual-reduced.*

Notice that any (δ, k)-slide-reduced basis in the sense of Def. 1 is $((\delta^2\gamma_k)^{\frac{1}{2(k-1)}}, k, 1)$-slide-reduced, while any $(1+\varepsilon, r)$-block-Rankin reduced basis with block size k in the sense of [LN14, Def. 3] is $(((1+\varepsilon)\gamma_{k,r})^{\frac{1}{2(k-r)r}}, k, r)$-slide-reduced.

Some main properties of generic slide reduction are illustrated below.

Theorem 2. *Let n, k, p, q, r be integers such that $n = pk + q \geq 2k$ with $0 \leq q < k$ and $k > r \geq 1$, and let $h \geq 1$. If $\mathbf{B}$ is a (h, k, r)-slide-reduced basis of an n-rank lattice $\mathcal{L}$, then*

$$\mathrm{vol}(\mathbf{B}_{[1,r]}) \leq h^{(n-r)r} \cdot \mathrm{vol}(\mathcal{L})^{r/n}, \tag{2}$$

$$\mathrm{vol}(\mathbf{B}_{[1,ik+q]}) \leq h^{(ik+q)(n-ik-q)} \cdot \mathrm{vol}(\mathcal{L})^{(ik+q)/n} \quad \text{for} \quad i = 1, \ldots, p-1. \tag{3}$$

We prove this theorem with the following simple fact:

Fact 3. *Let d, k, r be integers such that $d \geq k > r \geq 1$, and let $h \geq 1$. If $\mathbf{C}$ is a basis of rank $(d+k)$ such that $\mathbf{C}_{[1,d+r]}$ is (h, r)-twin-reduced and $\mathbf{C}_{[d+1,d+k]}$ is (h, r)-reduced, then*

$$\mathrm{vol}(\mathbf{C}_{[1,d]})^{1/d} \leq h^{d+k} \cdot \mathrm{vol}(\mathbf{C}_{[d+1,d+k]})^{1/k}.$$

Proof. Since $\mathbf{C}_{[1,d]}$ is (h, r)-reduced and $\mathbf{C}_{[r+1,d+r]}$ is (h, r)-dual-reduced, we have

$$\mathrm{vol}(\mathbf{C}_{[1,d]})^{1/d} \leq h^r \cdot \mathrm{vol}(\mathbf{C}_{[r+1,d]})^{1/(d-r)} \text{ and } \mathrm{vol}(\mathbf{C}_{[r+1,d]})^{1/(d-r)} \leq h^d \cdot \mathrm{vol}(\mathbf{C}_{[d+1,d+r]})^{1/r}.$$

By multiplying the above two inequalities together, this implies $\mathrm{vol}(\mathbf{C}_{[1,d]})^{r/d} \leq h^{(d+r)r} \cdot \mathrm{vol}(\mathbf{C}_{[d+1,d+r]})$.

The (h, r)-reducedness of $\mathbf{C}_{[d+1,d+k]}$ ensures $\mathrm{vol}(\mathbf{C}_{[d+1,d+r]}) \leq h^{(k-r)r} \cdot \mathrm{vol}(\mathbf{C}_{[d+1,d+k]})^{r/k}$. By multiplying the above two inequalities together, this proves Fact 3. □

Proof of Th. 2. Since $\mathbf{B}_{[1,k+q+r]}$ and each $\mathbf{B}_{[ik+q+1,(i+1)k+q+r]}$ are (h, r)-twin-reduced, the primal conditions and Fact 3 allow to deduce:

$$\begin{aligned}
\mathrm{vol}(\mathbf{B}_{[1,k+q]})^{1/(k+q)} &\leq h^{2k+q} \cdot \mathrm{vol}(\mathbf{B}_{[k+q+1,2k+q]})^{1/k}, \\
\mathrm{vol}(\mathbf{B}_{[ik+q+1,(i+1)k+q]})^{1/k} &\leq h^{2k} \cdot \mathrm{vol}(\mathbf{B}_{[(i+1)k+q+1,(i+2)k+q]})^{1/k} \quad \text{for} \quad i = 1, \ldots, p-2.
\end{aligned}$$

Now, let $i \in [1, p-1]_{\mathbb{Z}}$ be fixed. By combining the above inequalities, we have:

$$\begin{aligned}
\mathrm{vol}(\mathbf{B}_{[1,k+q]})^{1/(k+q)} &\leq h^{2ik+q} \cdot \mathrm{vol}(\mathbf{B}_{[ik+q+1,(i+1)k+q]})^{1/k}, \\
\mathrm{vol}(\mathbf{B}_{[gk+q+1,(g+1)k+q]})^{1/k} &\leq h^{2(i-g)k} \cdot \mathrm{vol}(\mathbf{B}_{[ik+q+1,(i+1)k+q]})^{1/k} \quad \text{for} \quad g = 1, , \ldots, i-1, \\
\mathrm{vol}(\mathbf{B}_{[ik+q+1,(i+1)k+q]})^{1/k} &\leq h^{2(j-i)k} \cdot \mathrm{vol}(\mathbf{B}_{[jk+q+1,(j+1)k+q]})^{1/k} \quad \text{for} \quad j = i, \ldots, p-1.
\end{aligned}$$

We make products of the above inequalities to deduce

$$\begin{aligned}
\mathrm{vol}(\mathbf{B}_{[1,ik+q]})^{1/(ik+q)} &\leq h^{(i+1)k+q} \cdot \mathrm{vol}(\mathbf{B}_{[ik+q+1,(i+1)k+q]})^{1/k}, \\
\mathrm{vol}(\mathbf{B}_{[ik+q+1,(i+1)k+q]})^{1/k} &\leq h^{(p-i-1)k} \cdot \mathrm{vol}(\mathbf{B}_{[ik+q+1,n]})^{1/((p-i)k)}.
\end{aligned}$$

This implies $\mathrm{vol}(\mathbf{B}_{[1,ik+q]})^{1/(ik+q)} \leq h^n \cdot \mathrm{vol}(\mathbf{B}_{[ik+q+1,n]})^{1/((p-i)k)}$, which is equivalent to Eq.(3).

Since $\mathbf{B}_{[1,k+q]}$ is (h, r)-reduced, we have $\mathrm{vol}(\mathbf{B}_{[1,r]}) \leq h^{(k+q-r)r} \cdot \mathrm{vol}(\mathbf{B}_{[1,k+q]})^{r/(k+q)}$. Combining it with Eq.(3) for $i = 1$, this proves Eq.(2) and hence Th. 2. □

3.2 A reduction algorithm

Our generic slide reduction algorithm is Alg. 2, which aims to provide a universal algorithmic framework of slide-type reductions.

Just like [GN08a, Alg. 1] and [LN14, Alg. 3], (h, r)-(dual-)reducing any projected block $\mathbf{B}_{[\cdot,\cdot]}$ does not change the input lattice, and does not modify the basis vectors outside the block. Many lattice reduction algorithms (such as LLL [LLL82], BKZ [SE94], and even slide-reduction itself) can be used to do (h, r)-(dual-)reductions (w.r.t. different h).

We remark that it is folklore to insert LLL-reductions in lattice reduction algorithms, because it cheaply guarantees that the final basis is "short" and that all intermediate entries during execution remain polynomially bounded (see, e.g., [LN14] for details). Since LLL never increases $\mathrm{vol}(\mathbf{B}_{[1,j]})$ for any $j \in [1, n]_{\mathbb{Z}}$ during execution, it can be checked that our analyses in both the proof of Th. 4 and Sect. 4 still work with LLL-reduction right after Step 5 or Step 11.

Algorithm 2 GSR: a generic slide reduction algorithm

Input: Block size $k > r \geq 1$, quality factor $h \geq 1$, a basis $\mathbf{B}$ of a lattice $\mathcal{L}$ of rank $n = pk + q \geq 2k$ for $0 \leq q < k$, and two slack factors $0 \leq \varepsilon_q, \varepsilon_k \leq 1$.
Output: A new basis of $\mathcal{L}$.

1: **repeat**
2: $((1+\varepsilon_q)^{1/((k+q-r)r)}h, r)$-reduce $\mathbf{B}_{[1,k+q]}$. //A GSR tour refers to a single execution of Steps 2-11.
3: **for** $i = 1$ **to** $p-1$ (possibly in parallel) **do**
4: (h, r)-reduce $\mathbf{B}_{[ik+q+1,(i+1)k+q]}$.
5: **end for**
6: **if** $\mathbf{B}_{[r+1,k+q+r]}$ is not $((1+\varepsilon_q)^{1/((k+q-r)r)}h, r)$-dual-reduced, **then**
7: $((1+\varepsilon_q)^{0.5/((k+q-r)r)}h, r)$-dual-reduce $\mathbf{B}_{[r+1,k+q+r]}$.
8: **end if**
9: **if** $\mathbf{B}_{[ik+q+r+1,(i+1)k+q+r]}$ is not $((1+\varepsilon_k)^{0.5/((k-r)r)}h, r)$-dual-reduced for some $i \in [1, p-2]_{\mathbb{Z}}$, **then**
10: (h, r)-dual-reduce $\mathbf{B}_{[ik+q+r+1,(i+1)k+q+r]}$.
11: **end if**
12: **until** no change occurs or termination is requested, and **return** $\mathbf{B}$.

By slightly adapting the standard integral potential used in [ALNS20], we show that within a sufficient number of GSR tours independent of the input LLL-reduced basis, Alg. 2 still outputs slide-reduced bases:

Theorem 4. *Let n, k, p, q, r be integers such that $n = pk + q \geq 2k$ with $0 \leq q < k$ and $1 \leq r < k$. Let $h \geq 1$ be the quality factor. Let $\varepsilon_q, \varepsilon_k$ be two slack factors such that $0 < \varepsilon_k \leq 1$, $\varepsilon_q = 0$ if $q = 0$ and $\varepsilon_q = \varepsilon_k$ otherwise. Given as input an LLL-reduced basis $\mathbf{B}_0 \in \mathbb{R}^{m \times n}$ of a lattice $\mathcal{L}$ of rank n, Alg. 2 makes at most $\left\lceil \frac{(p-1)n^2}{\log(1+\varepsilon_k)} \right\rceil$ $(\in O(\frac{n^3}{k\varepsilon_k}))$ GSR tours, and outputs a $((1+\varepsilon_k)^{1/((k-r)r)}h, k, r)$-slide-reduced basis of $\mathcal{L}$.*

Proof. First, notice that if Alg. 2 terminates, then its output must be $((1+\varepsilon_k)^{1/((k-r)r)}h, k, r)$-slide-reduced.

It remains to bound the number of GSR tours. Let $\mathbf{B} \in \mathbb{R}^{m \times n}$ denote the current basis during the execution of Alg. 2 . We consider a ratio potential of the form

$$\tau(\mathbf{B}) := \prod_{i=1}^{p-1} \frac{\mathrm{vol}(\mathbf{B}_{[1,ik+q]})^2}{m_{ik+q}(\mathcal{L})^2} \geq 1.$$

Since the input basis $\mathbf{B}_0$ is LLL-reduced, the initial potential satisfies $\log \tau(\mathbf{B}_0) \leq (p-1)n^2$ (see [PT08, Eq. (2)]). Every operation in Alg. 2 either preserves or significantly decreases $\tau(\mathbf{B})$. In particular, the potential is unaffected by the primal steps (i.e., Steps 2 and 4), which leave $\mathrm{vol}(\mathbf{B}_{[1,ik+q]})$ unchanged for all i. The dual steps (i.e., Steps 7 and 10) either leave $\mathrm{vol}(\mathbf{B}_{[1,ik+q]})$ for all i or decrease $\tau(\mathbf{B})$ by a multiplicative factor of at least $(1+\varepsilon_k)$: for instance, if Step 9 occurs for some index i, namely $\mathrm{vol}(\mathbf{B}_{[ik+q+r+1,(i+1)k+q]}) > \sqrt{1+\varepsilon_k}h^{(k-r)r} \cdot \mathrm{vol}(\mathbf{B}_{[ik+q+r+1,(i+1)k+q+r]})^{(k-r)/k}$, then Step 10 will update $\mathbf{b}_{ik+q+r+1}, \ldots, \mathbf{b}_{(i+1)k+q+r}$ such that the above inequality becomes $\mathrm{vol}(\mathbf{B}_{[ik+q+r+1,(i+1)k+q]}) \leq h^{(k-r)r} \cdot \mathrm{vol}(\mathbf{B}_{[ik+q+r+1,(i+1)k+q+r]})^{(k-r)/k}$.

Therefore, Alg. 2 updates $\mathrm{vol}(\mathbf{B}_{[1,ik+q]})$ for some i at most N times, where $N := \left\lceil \frac{\log \tau(\mathbf{B}_0)}{\log(1+\varepsilon_k)} \right\rceil \leq \left\lceil \frac{(p-1)n^2}{\log(1+\varepsilon_k)} \right\rceil$. This completes the proof. □

3.3 Implications

Revisiting [LN14, Alg. 3]. Our Alg. 2 with $(h, q, \varepsilon_k) = (\gamma_{k,r}^{\frac{1}{2(k-r)r}}, 0, \varepsilon)$ is exactly [LN14, Alg. 3]. The authors proved that given as input a basis $\mathbf{B}_0$, [LN14, Alg. 3] makes at most $\left\lceil \frac{(p-1)n \log \|\mathbf{B}_0\|}{\log(1+\varepsilon)} \right\rceil$ $(\in O(\frac{n^2 \log \|\mathbf{B}_0\|}{k\varepsilon}))$ tours for computing a $(1+\varepsilon, r)$-block-Rankin reduced basis with block size k (see [LN14, Def. 3]).

Th. 4 removes the dependence on the input LLL-reduced basis:

Corollary 1. *Let $n = pk \geq 2k$ and $\varepsilon \in (0, 1]$. Given as input an LLL-reduced basis $\mathbf{B}_0 \in \mathbb{R}^{m \times n}$ of a lattice $\mathcal{L}$ of rank n, [LN14, Alg. 3] makes at most $\left\lceil \frac{(p-1)n^2}{\log(1+\varepsilon)} \right\rceil$ $(\in O(\frac{n^3}{k\varepsilon}))$ tours, and outputs a $(1+\varepsilon, r)$-block-Rankin reduced basis with block size k of $\mathcal{L}$.*

Revisiting [ALNS20, Alg. 2]. The potential analysis used in Sect. 3.2 also works well for analyzing the ALNS slide reduction algorithm for $n = k+q \in [k+2, 2k]$ [ALNS20, Alg. 2]: under their notation and with the ratio potential $\tau(\mathbf{B}) = \frac{\mathrm{vol}(\mathbf{B}_{[1,q]})}{m_q(L(\mathbf{B}))} \geq 1$ (instead of their integral potential $P(\mathbf{B}) = \mathrm{vol}(\mathbf{B}_{[1,q]})^2 \in \mathbb{Z}^+$), if the input basis $\mathbf{B}_0$ is LLL-reduced, then [ALNS20, Alg. 2] makes at most $\left\lceil \frac{qk}{\log(1+\varepsilon)} + k - q \right\rceil$ $(\in O(\frac{qk}{\varepsilon}))$ calls to the SVP-oracle (rather than their claimed number $\left\lceil \frac{qk \log \|\mathbf{B}_0\|}{\log(1+\varepsilon)} \right\rceil$ $(\in O(\frac{qk \log \|\mathbf{B}_0\|}{\varepsilon}))$ and outputs a slide-reduced basis of the input lattice.

Revisiting [ALNS20, Alg. 3]. Our Alg. 2 with $(h, \varepsilon_q, \varepsilon_k) = ((\delta^2\gamma_k)^{\frac{1}{2(k-1)}}, \varepsilon, \varepsilon)$ is essentially [ALNS20, Alg. 3] (i.e. Alg. 1). If the input basis $\mathbf{B}_0$ is LLL-reduced, Th. 4 allows to remove the dependence on $\mathbf{B}_0$ from the claimed number of tours in Th. 1:

Corollary 2. *For $\varepsilon \in (0, 1]$, given as input an LLL-reduced basis $\mathbf{B}_0 \in \mathbb{R}^{m \times n}$ of a lattice $\mathcal{L}$ of rank $n = pk+q \geq 2k$ for $0 \leq q < k$, Alg. 1 makes at most $\left\lceil \frac{(p-1)n^2}{\log(1+\varepsilon)} \right\rceil$ $(\in O(\frac{n^3}{k\varepsilon}))$ slide tours, and outputs a $((1+\varepsilon)\delta, k)$-slide-reduced basis of $\mathcal{L}$.*

Remark 2. When Alg. 1 terminates, it outputs a $((1+\varepsilon)\delta, k)$-slide-reduced basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of the input lattice $\mathcal{L}$. By directly applying [ALNS20, Th. 7], it has the following properties:

$$\|\mathbf{b}_1\| \leq (((1+\varepsilon)\delta)^2\gamma_k)^{\frac{n-1}{2(k-1)}} \mathrm{vol}(\mathcal{L})^{1/n} , \tag{4}$$

and if $\lambda_1(\mathcal{L}(\mathbf{B}_{[1,k+q]})) > \lambda_1(\mathcal{L})$, then

$$\|\mathbf{b}_1\| \leq (1+\varepsilon)\delta(((1+\varepsilon)\delta)^2\gamma_k)^{\frac{n-k}{k-1}} \lambda_1(\mathcal{L}) . \tag{5}$$

As mentioned in Sect. 1, if $\mathcal{L}$ is a random lattice, it follows from the Gaussian heuristic that Property (4) might imply Property (5) in practice. This is why current security estimates in lattice-based cryptographic constructions actually rely on HSVP estimates [GN08b, ADPS16]. So, one may be interested to wonder whether Property (4) can be achieved with fewer slide tours. Besides, the potential analysis used in Sect. 3.2 can not analyse the current basis during the execution of Alg. 1. This is the topic of the next section.

4 Dynamic analysis of generic slide reduction

In this section, we adapt the discrete dynamical system analysis of Hanrot *et al.* [HPS11] to GSR (Alg. 2). This allows to claim Property (4) with fewer slide tours. Furthermore, we may use this analysis to model the practical behaviour of slide-reductions [GN08a, ALNS20]. Finally, this analysis provides guarantees on the quality of the current basis during execution (see Eq. (7) in Prop. 1). In Sect. 6, we develop Eq. (7) into a heuristic simulator, which can be used to predict the evolution of the RHF during execution.

The main result of this section is as follows:

Theorem 5. *Let n, k, p, q, r be integers such that $n = pk + q \geq 2k$ with $0 \leq q < k$ and $1 \leq r < k$. Let $h \geq 1$ be the quality factor. Let $\varepsilon_q, \varepsilon_k$ be two slack factors such that $\varepsilon_k = 0$, $\varepsilon_q = 0$ if $q = 0$ and $\varepsilon_q = \varepsilon$ otherwise, where $0 < \varepsilon \leq 0.4$. Given as input an LLL-reduced basis $\mathbf{B}_0$ of an n-rank lattice $\mathcal{L}$ in $\mathbb{R}^m$, if terminated after $\left\lceil \frac{n^2 \ln \frac{n-k-q}{4\log(1+\varepsilon)}}{8(k-r)r} \right\rceil$ $(\in O(\frac{n^2 \ln \frac{n}{\varepsilon}}{(k-r)r}))$ GSR tours, then Alg. 2 outputs a basis $\mathbf{B}$ of $\mathcal{L}$ such that*

$$\mathrm{vol}(\mathbf{B}_{[1,ik+q]}) \leq (1+\varepsilon)^{\frac{(ik+q)(p-i)}{p-1}} \overline{h}^{(ik+q)(n-ik-q)} \cdot \mathrm{vol}(\mathcal{L})^{(ik+q)/n} \quad \text{for} \quad i = 1, \ldots, p-1,$$

where $\overline{h} = (1+\varepsilon_q)^{1/((k+q-r)r)}h$. One more execution of Step 2 yields

$$\mathrm{vol}(\mathbf{B}_{[1,r]}) \leq (1+\varepsilon)^r \overline{h}^{(n-r)r} \cdot \mathrm{vol}(\mathcal{L})^{r/n}.$$

Sections 4.1-4.3 will be devoted to the proof of this theorem.

Revisiting [LN14, Alg. 3]. By [LN14, Th. 4.1], the output block-Rankin reduced basis $\mathbf{B}$ of [LN14, Alg. 3] satisfies $\mathrm{vol}(\mathbf{B}_{[1,r]}) \leq (\sqrt{1+\varepsilon}\gamma_{k,r})^{\frac{n-r}{2(k-r)}}\mathrm{vol}(\mathcal{L})^{\frac{r}{n}}$, which requires $O(\frac{n^2}{k}\cdot\frac{n}{\varepsilon})$ tours (by Cor. 1). It follows from Th. 5 with $(h,q) = (\gamma_{k,r}^{\frac{1}{2(k-r)r}}, 0)$ that with fewer tours, [LN14, Alg. 3] can obtain a basis whose bound on $\mathrm{vol}(\mathbf{B}_{[1,r]})$ is almost the same as that of full block-Rankin reduction:

Corollary 3. *Let n, k, p, r be integers such that $n = pk \geq 2k$ with $k > r \geq 1$, and let $0 < \varepsilon \leq 0.4$. Given as input an LLL-reduced basis $\mathbf{B}_0$ of an n-rank lattice $\mathcal{L}$ in $\mathbb{R}^m$, if terminated after $\left\lceil \frac{n^2 \ln \frac{n-k}{4\log(1+\varepsilon)}}{8(k-r)r} \right\rceil$ $(\in O(\frac{n^2}{(k-r)r}\cdot \ln\frac{n}{\varepsilon}))$ tours, then [LN14, Alg. 3] with one more execution to its Step 4 outputs a basis $\mathbf{B}$ of $\mathcal{L}$ such that*

$$\mathrm{vol}(\mathbf{B}_{[1,r]}) \leq (1+\varepsilon)^r \gamma_{k,r}^{\frac{n-r}{2(k-r)}} \cdot \mathrm{vol}(\mathcal{L})^{\frac{r}{n}}.$$

Revisiting [ALNS20, Alg. 3]. It follows from Th. 5 with $(h, r) = ((\delta^2\gamma_k)^{\frac{1}{2(k-1)}}, 1)$ that with fewer tours, Alg. 1 can still obtain a basis whose RHF is almost the same as that in Property (4) for full slide-reduction.

Notice that the dynamic analysis for Th. 5 is agnostic to the quality factor h. Under the weak version of the Gaussian heuristic (cf. Sect. 2.1), Th. 5 can also be used to model the practical behaviour of slide-reductions [GN08a, ALNS20]:

Corollary 4. *Let n, k, p, q be integers such that $n = pk + q \geq 2k$ with $0 \leq q < k$ and $k \geq 2$. Let $\delta \geq 1$ and $0 < \varepsilon \leq 0.4$. Let $\varepsilon_q = 0$ if $q = 0$ and $\varepsilon_q = \varepsilon$ otherwise. Given as input an LLL-reduced basis $\mathbf{B}_0$ of an n-rank lattice $\mathcal{L}$ in $\mathbb{R}^m$, if terminated after $\left\lceil \frac{n^2 \ln \frac{n-k-q}{4\log(1+\varepsilon)}}{8(k-1)} \right\rceil$ $(\in O(\frac{n^2\ln\frac{n}{\varepsilon}}{k}))$ slide tours, then Alg. 1 with one more execution to its Step 2 outputs a basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of $\mathcal{L}$ satisfying the following properties.*

1. *Worst-case: $\|\mathbf{b}_1\| \leq (1+\varepsilon)(1+\varepsilon_q)^{\frac{n-1}{k+q-1}}(\delta^2\gamma_k)^{\frac{n-1}{2(k-1)}}\mathrm{vol}(\mathcal{L})^{1/n}$.*
2. *Average-case: Suppose that every projected lattice Λ of rank k passed to (D)(H)SVP-reductions of Alg. 1 satisfies $\lambda_1(\Lambda) \leq (1+\xi)GH(k)\cdot\mathrm{vol}(\Lambda)^{1/k}$ for some $\xi \in (0, \sqrt{2}-1]$, we have*

$$\|\mathbf{b}_1\|/\mathrm{vol}(\mathcal{L})^{1/n} \leq (1+\varepsilon)(1+\varepsilon_q)^{\frac{n-1}{k+q-1}}((1+\xi)GH(k))^{\frac{n-1}{k-1}} \approx \left(\frac{k}{2\pi e}\right)^{\frac{n-1}{2(k-1)}}.$$

Remark 3.

1. The output basis in Cor. 4 might not be slide-reduced in the sense of Def. 1: hence, it might not satisfy Property (5) and the twin-reducednesses of $\mathbf{B}_{[1,ik+q+1]}$ for $i = 1, \ldots, p-1$.
2. The dependence of time/quality trade-offs on ε in Cor. 4.1 is better than that in Cor. 2, especially for $k = o(n)$. For instance, in order to claim $\|\mathbf{b}_1\| \leq (1+\varepsilon)^{\frac{n-1}{k-1}}\gamma_k^{\frac{n-1}{2(k-1)}}\mathrm{vol}(\mathcal{L})^{1/n}$ in the case $q = 0$, Cor. 2 requires at most $\left\lceil \frac{(n-k)n^2}{k\log(1+\varepsilon)} \right\rceil$ $(\in O(\frac{n^2}{k}\cdot\frac{n}{\varepsilon}))$ slide tours, while Cor. 4.1 requires at most $\left\lceil \frac{n^2 \ln \frac{(k-1)(n-k)}{4(n-1)\log(1+\varepsilon)}}{8(k-1)} \right\rceil$ $(\in O(\frac{n^2}{k}\cdot\ln\frac{n}{\varepsilon}))$ slide tours.
3. Our estimate in the average-case of Cor. 4.2 (ignoring both ε and ξ) matches well with experiments in Sect. 5.2 (see Fig. (1c)).

4.1 A dynamical system for GSR tours

We use the following profile function to do the dynamic analysis of GSR: given a basis $\mathbf{B}$ of rank $n = pk+q \geq 2k$ for $0 \leq q < k$, its bit profile is:

$$\mathcal{G}(\mathbf{B}) = (\mathcal{G}_1(\mathbf{B}), \mathcal{G}_2(\mathbf{B}), \ldots, \mathcal{G}_{p-1}(\mathbf{B}))^{\mathrm{T}} \in \mathbb{R}^{p-1} \quad \text{with each} \quad \mathcal{G}_i(\mathbf{B}) = \log\left(\frac{\mathrm{vol}(\mathbf{B}_{[1,ik+q]})}{\mathrm{vol}(\mathbf{B})^{(ik+q)/n}}\right)^{\frac{1}{(ik+q)(n-ik-q)}}.$$

In what follows, we also use the notations:

$$\beta = \frac{(k-r)r}{k^2}, \quad \theta = \frac{k+q-r}{k+q} \quad \text{and} \quad \omega_i = (ik+q)(n-ik-q) \quad \text{for} \quad i = 1, \ldots, p.$$

Proposition 1. *Let n, k, p, q, r be integers s.t. $n = pk + q \geq 2k$ with $0 \leq q < k$ and $1 \leq r < k$. Let $\varepsilon_q, \varepsilon_k$ be two slack factors such that $\varepsilon_k = 0$, $\varepsilon_q = 0$ if $q = 0$ and $\varepsilon_q \in (0, 1]$ otherwise. Let $h \geq 1$ be the quality factor in Alg. 2 and $\overline{h} = (1+\varepsilon_q)^{1/((k+q-r)r)} h$. If a GSR tour transforms a basis $\mathbf{B}$ of $\mathcal{L}$ into another basis $\mathbf{C}$ of $\mathcal{L}$, then*

$$\mathcal{G}(\mathbf{C}) \leq A \cdot \mathcal{G}(\mathbf{B}) + \mathbf{b}, \tag{6}$$

where the matrix $\mathbf{A} \in \mathbb{Q}^{(p-1)\times(p-1)}$ and vector $\mathbf{b} \in \mathbb{Q}^{p-1}$ are defined by

$$\mathbf{A} = \begin{pmatrix} \frac{r^2}{(k+q)^2} + \frac{(k-r)\theta}{k} & \frac{\theta \cdot r \cdot \omega_2}{k \cdot \omega_1} & & & & & \\ \frac{\beta\omega_1}{\omega_2} & 1-2\beta & \frac{\beta\omega_3}{\omega_2} & & & & \\ & \frac{\beta\omega_2}{\omega_3} & 1-2\beta & \frac{\beta\omega_4}{\omega_3} & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \ddots & \ddots & \ddots \\ & & & & \frac{\beta\omega_{p-3}}{\omega_{p-2}} & 1-2\beta & \frac{\beta\omega_{p-1}}{\omega_{p-2}} \\ & & & & & \frac{\beta\omega_{p-2}}{\omega_{p-1}} & 1-2\beta \end{pmatrix} \quad \text{and} \quad \mathbf{b} = (r \log \overline{h}) \cdot \begin{pmatrix} \frac{(2k+q)\theta}{\omega_1} \\ \frac{2(k-r)}{\omega_2} \\ \frac{2(k-r)}{\omega_3} \\ \vdots \\ \\ \vdots \\ \\ \vdots \\ \frac{2(k-r)}{\omega_{p-2}} \\ \frac{2(k-r)}{\omega_{p-1}} \end{pmatrix}.$$

Moreover, let $\mathbf{y} \in \mathbb{R}^n$ be a fixed point of the dynamical system $\mathbf{x} \leftarrow \mathbf{A}\mathbf{x} + \mathbf{b}$. If $\mathbf{B}_0$ is the input basis of $\mathcal{L}$ to GSR and $\mathbf{B}_\ell$ is the current basis at the end of the ℓ-th GSR tour, then

$$\mathcal{G}(\mathbf{B}_\ell) \leq \mathbf{y} + \mathbf{A}^\ell \cdot (\mathcal{G}(\mathbf{B}_0) - \mathbf{y}) \quad \text{for} \quad \forall \ell \in \mathbb{Z}^+. \tag{7}$$

Proof. Let $\mathbf{E}$ be the current basis right after Step 5 of GSR such that Steps 2-5 transform the basis $\mathbf{B}$ into $\mathbf{E}$ and then Steps 6-11 transform the basis $\mathbf{E}$ into $\mathbf{C}$. The main issue is to connect $\mathbf{B}$ with $\mathbf{C}$.

To do so, we first establish the relation between $\mathbf{B}$ and $\mathbf{E}$:

$$\mathrm{vol}(\mathbf{E}_{[1,r]}) \leq \overline{h}^{(k+q-r)r} \cdot \mathrm{vol}(\mathbf{B}_{[1,k+q]})^{r/(k+q)}, \tag{8}$$

$$\mathrm{vol}(\mathbf{E}_{[1,ik+q+r]}) \leq \overline{h}^{(k-r)r} \cdot \mathrm{vol}(\mathbf{B}_{[1,ik+q]})^{(k-r)/k} \cdot \mathrm{vol}(\mathbf{B}_{[1,(i+1)k+q]})^{r/k} \quad \text{for} \quad i = 1, \ldots, p-1. \tag{9}$$

Proof of Eq. (8) and Eq. (9). Eq. (8) is ensured by Step 2. For $i = 1, \ldots, p-1$, we have

$$\begin{aligned} \mathrm{vol}(\mathbf{E}_{[1,ik+q+r]}) &= \mathrm{vol}(\mathbf{B}_{[1,ik+q]}) \cdot \mathrm{vol}(\mathbf{E}_{[ik+q+1,ik+q+r]}) && \text{(Since } \mathrm{vol}(\mathbf{B}_{[1,ik+q]}) = \mathrm{vol}(\mathbf{E}_{[1,ik+q]})\text{)} \\ &\leq \overline{h}^{(k-r)r} \cdot \mathrm{vol}(\mathbf{B}_{[1,ik+q]}) \cdot \mathrm{vol}(\mathbf{E}_{[ik+q+1,(i+1)k+q]})^{r/k} && \text{(Due to Step 4)} \\ &= \overline{h}^{(k-r)r} \cdot \mathrm{vol}(\mathbf{B}_{[1,ik+q]}) \cdot \mathrm{vol}(\mathbf{B}_{[ik+q+1,(i+1)k+q]})^{r/k} \\ &= \overline{h}^{(k-r)r} \cdot \mathrm{vol}(\mathbf{B}_{[1,ik+q]})^{(k-r)/k} \cdot \mathrm{vol}(\mathbf{B}_{[1,(i+1)k+q]})^{r/k}. \end{aligned}$$

Thus, we proved Eq. (8) and Eq. (9). □

We next use Eq. (8) and Eq. (9) to establish the relation between $\mathbf{B}$ and $\mathbf{C}$:

$$\mathrm{vol}(\mathbf{C}_{[1,k+q]}) \leq \overline{h}^{(2k+q)\theta r} \cdot \mathrm{vol}(\mathbf{B}_{[1,k+q]})^{\frac{r^2}{(k+q)^2} + \frac{(k-r)\theta}{k}} \cdot \mathrm{vol}(\mathbf{B}_{[1,2k+q]})^{\frac{\theta \cdot r}{k}}, \tag{10}$$

$$\mathrm{vol}(\mathbf{C}_{[1,ik+q]}) \leq \overline{h}^{2(k-r)r} \cdot \mathrm{vol}(\mathbf{B}_{[1,(i-1)k+q]})^{\beta} \cdot \mathrm{vol}(\mathbf{B}_{[1,ik+q]})^{1-2\beta} \cdot \mathrm{vol}(\mathbf{B}_{[1,(i+1)k+q]})^{\beta} \quad \text{for} \quad i = 2, \ldots, p-1. \tag{11}$$

Proof of Eq. (10) and Eq. (11). Using the connection between $\mathbf{E}$ and $\mathbf{C}$, we first argue Eq. (10):

$$\begin{aligned} \mathrm{vol}(\mathbf{C}_{[1,k+q]}) &= \mathrm{vol}(\mathbf{E}_{[1,r]}) \cdot \mathrm{vol}(\mathbf{C}_{[r+1,k+q]}) && \text{(Since } \mathrm{vol}(\mathbf{E}_{[1,r]}) = \mathrm{vol}(\mathbf{C}_{[1,r]})\text{)} \\ &\leq \overline{h}^{(k+q-r)r} \cdot \mathrm{vol}(\mathbf{E}_{[1,r]}) \cdot \mathrm{vol}(\mathbf{C}_{[r+1,k+q+r]})^{\theta} && \text{(Due to Steps 6-8)} \\ &= \overline{h}^{(k+q-r)r} \cdot \mathrm{vol}(\mathbf{E}_{[1,r]}) \cdot \mathrm{vol}(\mathbf{E}_{[r+1,k+q+r]})^{\theta} && \text{(Since } \mathrm{vol}(\mathbf{E}_{[r+1,k+q+r]}) = \mathrm{vol}(\mathbf{C}_{[r+1,k+q+r]})\text{)} \\ &= \overline{h}^{(k+q-r)r} \cdot \mathrm{vol}(\mathbf{E}_{[1,r]})^{1-\theta} \cdot \mathrm{vol}(\mathbf{E}_{[1,k+q+r]})^{\theta} \\ &\leq \overline{h}^{(2k+q)\theta r} \cdot \mathrm{vol}(\mathbf{B}_{[1,k+q]})^{\frac{r^2}{(k+q)^2} + \frac{(k-r)\theta}{k}} \cdot \mathrm{vol}(\mathbf{B}_{[1,2k+q]})^{\frac{\theta \cdot r}{k}}. && \text{(Applying Eq. (8) and Eq. (9))} \end{aligned}$$

For $i = 2, \dots, p-1$, since $\mathrm{vol}(\mathbf{E}_{[1,(i-1)k+q+r]}) = \mathrm{vol}(\mathbf{C}_{[1,(i-1)k+q+r]})$, we have

$$\begin{aligned}
\mathrm{vol}(\mathbf{C}_{[1,ik+q]}) &= \mathrm{vol}(\mathbf{E}_{[1,(i-1)k+q+r]}) \cdot \mathrm{vol}(\mathbf{C}_{[(i-1)k+q+r+1,ik+q]}) \\
&\le \overline{h}^{(k-r)r} \cdot \mathrm{vol}(\mathbf{E}_{[1,(i-1)k+q+r]}) \cdot \mathrm{vol}(\mathbf{C}_{[(i-1)k+q+r+1,ik+q+r]})^{(k-r)/k} && \text{(Due to Steps 9-11)} \\
&= \overline{h}^{(k-r)r} \cdot \mathrm{vol}(\mathbf{E}_{[1,(i-1)k+q+r]}) \cdot \mathrm{vol}(\mathbf{E}_{[(i-1)k+q+r+1,ik+q+r]})^{(k-r)/k} \\
&= \overline{h}^{(k-r)r} \cdot \mathrm{vol}(\mathbf{E}_{[1,(i-1)k+q+r]})^{r/k} \cdot \mathrm{vol}(\mathbf{E}_{[1,ik+q+r]})^{(k-r)/k} \\
&\le \overline{h}^{2(k-r)r} \cdot \mathrm{vol}(\mathbf{B}_{[1,(i-1)k+q]})^{\beta} \cdot \mathrm{vol}(\mathbf{B}_{[1,ik+q]})^{1-2\beta} \cdot \mathrm{vol}(\mathbf{B}_{[1,(i+1)k+q]})^{\beta}. && \text{(By Eq. (9))}
\end{aligned}$$

Thus, we proved Eq. (10) and Eq. (11). □

With the notation $\mathcal{G}_i(\cdot)$, Eq. (10) and Eq. (11) are respectively equivalent to the inequalities below:

$$\begin{aligned}
\mathcal{G}_1(\mathbf{C}) &\le \left(\frac{r^2}{(k+q)^2} + \frac{(k-r)\theta}{k}\right)\mathcal{G}_1(\mathbf{B}) + \frac{\theta \cdot r \cdot \omega_2}{k \cdot \omega_1}\mathcal{G}_2(\mathbf{B}) + \frac{(2k+q)\theta r}{\omega_1}\log\overline{h}, \\
\mathcal{G}_i(\mathbf{C}) &\le \frac{\beta\omega_{i-1}}{\omega_i}\mathcal{G}_{i-1}(\mathbf{B}) + (1-2\beta)\mathcal{G}_i(\mathbf{B}) + \frac{\beta\omega_{i+1}}{\omega_i}\mathcal{G}_{i+1}(\mathbf{B}) + \frac{2(k-r)r}{\omega_i}\log\overline{h} \quad \text{for} \quad i = 2, \dots, p-1.
\end{aligned}$$

Their matrix form is exactly Eq. (6).

It remains to show Eq. (7), which is done by induction on ℓ. First, since $\mathbf{b} = \mathbf{y} - \mathbf{A}\mathbf{y}$, Eq. (6) implies Eq. (7) for $\ell = 1$. Assume that Eq. (7) holds for some $\ell \in \mathbb{Z}^+$.

The crucial fact $\mathbf{A} \ge 0$ allows us to deduce the following third row:

$$\begin{aligned}
\mathcal{G}(\mathbf{B}_{\ell+1}) &\le \mathbf{A} \cdot \mathcal{G}(\mathbf{B}_\ell) + \mathbf{b} && \text{(By Eq. (6))} \\
&= \mathbf{y} + \mathbf{A}(\mathcal{G}(\mathbf{B}_\ell) - \mathbf{y}) && \text{(Since } \mathbf{b} = \mathbf{y} - \mathbf{A}\mathbf{y}\text{)} \\
&\le \mathbf{y} + \mathbf{A}^{\ell+1}(\mathcal{G}(\mathbf{B}_0) - \mathbf{y}). && \text{(By the induction hypothesis)}
\end{aligned}$$

Thus, we proved Eq. (7). This completes the proof of Prop. 1. □

4.2 Properties of the dynamical system

Thanks to Eq. (6), the effect of a GSR tour on $\mathcal{G}(B)$ can be interpreted as the dynamical system

$$\mathbf{x} \leftarrow \mathbf{A}\mathbf{x} + \mathbf{b}. \tag{12}$$

Its fixed point(s) and speed of convergence encode information on the output quality and runtime of GSR, respectively, as illustrated by Eq. (7). The system (12) converges and has a unique fixed point:

Proposition 2. *Under the notation of Sect. 4.1, we have:*

1. *$\mathbf{A}$ has row sum norm: $\|\mathbf{A}\|_\infty \le 1 - \frac{8(k-r)r}{n^2}$.*
2. *The system $\mathbf{x} \leftarrow \mathbf{A}\mathbf{x} + \mathbf{b}$ has a unique fixed point: $\overline{\mathbf{y}} = (\log\overline{h}) \cdot (1, \dots, 1)^T \in \mathbb{R}^{p-1}$.*
3. *The tail term in Eq. (7) converges to zero: if c is a positive real s.t. $c \ge \max_{1 \le i < p} \mathcal{G}_i(\mathbf{B}_0)$, then*

$$\mathbf{A}^\ell \cdot (\mathcal{G}(\mathbf{B}_0) - \overline{\mathbf{y}}) \le c \cdot \|\mathbf{A}\|_\infty^\ell \cdot (1, \dots, 1)^T \quad \text{for any} \quad \ell \in \mathbb{Z}^+.$$

Proof. We first show Item 1. For $i = 2, \dots, p-2$, the sum of the i-th row of $\mathbf{A}$ is

$$S_i = 1 + \beta \cdot \left(\frac{\omega_{i-1} + \omega_{i+1}}{\omega_i} - 2\right) = 1 - \frac{2k^2\beta}{(ik+q)(n-ik-q)} \le 1 - \frac{8k^2\beta}{n^2} = 1 - \frac{8(k-r)r}{n^2}.$$

A direct calculation by hand shows: $\max\{S_1, S_{p-1}\} \le 1 - \frac{8(k-r)r}{n^2}$. This proves Item 1.

We now show Item 2. A direct calculation by hand shows:

$$\begin{aligned}
\log\overline{h} &= \left(\frac{r^2}{(k+q)^2} + \frac{(k-r)\theta}{k}\right)\log\overline{h} + \frac{\theta \cdot r \cdot \omega_2}{k \cdot \omega_1} \cdot \log\overline{h} + \frac{(2k+q)\theta r}{\omega_1}\log\overline{h}, \\
\log\overline{h} &= \frac{\beta\omega_{i-1}}{\omega_i} \cdot \log\overline{h} + (1-2\beta)\log\overline{h} + \frac{\beta\omega_{i+1}}{\omega_i} \cdot \log\overline{h} + \frac{2(k-r)r}{\omega_i}\log\overline{h} \quad \text{for} \quad i = 2, \dots, p-1.
\end{aligned}$$

Their matrix form is exactly $\overline{\mathbf{y}} = \mathbf{A}\overline{\mathbf{y}} + \mathbf{b}$.

To show the unicity, we assume that $\mathbf{y} \in \mathbb{Q}^{p-1}$ also satisfies $\mathbf{y} = \mathbf{A}\mathbf{y} + \mathbf{b}$. Then $\overline{\mathbf{y}} - \mathbf{y} = \mathbf{A}(\overline{\mathbf{y}} - \mathbf{y})$ implies $\|\overline{\mathbf{y}} - \mathbf{y}\|_\infty \leq \|\mathbf{A}\|_\infty \cdot \|\overline{\mathbf{y}} - \mathbf{y}\|_\infty$. Since $\|\mathbf{A}\|_\infty < 1$, we have $\|\overline{\mathbf{y}} - \mathbf{y}\|_\infty = 0$. Thus, $\overline{\mathbf{y}} = \mathbf{y}$. This proves Item 2.

It remains to show Item 3. The crucial fact $\mathbf{A} \geq 0$ allows us to deduce the following inequalities:

$$\begin{aligned}
\mathbf{A}^\ell \cdot (\mathcal{G}(\mathbf{B}_0) - \overline{\mathbf{y}}) &\leq \mathbf{A}^\ell \cdot \mathcal{G}(\mathbf{B}_0) && \text{(Since } \overline{\mathbf{y}} \geq 0\text{)} \\
&\leq c \cdot \mathbf{A}^\ell \cdot (1, \ldots, 1)^{\mathrm{T}} && \text{(Since } c \geq \max_{1 \leq i < p} \mathcal{G}_i(\mathbf{B}_0)\text{)} \\
&\leq c \cdot \|\mathbf{A}^\ell\|_\infty \cdot (1, \ldots, 1)^{\mathrm{T}} \\
&\leq c \cdot \|\mathbf{A}\|_\infty^\ell \cdot (1, \ldots, 1)^{\mathrm{T}}.
\end{aligned}$$

This proves Item 3 and completes the proof of Prop. 1. □

4.3 Proof of Theorem 5

We now show Th. 5 under the notation of Sections 4.1-4.2. Let ℓ be any integer such that $\ell \geq \frac{n^2 \ln \frac{n-k-q}{4 \log(1+\varepsilon)}}{8(k-r)r}$. Since $1 - x \leq e^{-x}$ for $0 \leq x \leq 1$, Prop. 2.1 implies

$$\frac{1}{4} \cdot \|\mathbf{A}\|_\infty^\ell \leq \frac{1}{4}\left(1 - \frac{8(k-r)r}{n^2}\right)^\ell \leq \frac{1}{4} e^{-\frac{8\ell(k-r)r}{n^2}} \leq \frac{\log(1+\varepsilon)}{n-k-q}.$$

Let $\mathbf{B}_\ell$ be the current basis at the end of the ℓ-th GSR tour. Since the input basis $\mathbf{B}_0$ is LLL-reduced, we have $\mathcal{G}(\mathbf{B}_0) \leq \frac{1}{4} \cdot (1, \ldots, 1)^{\mathrm{T}}$ (see [PT08, Eq. (3)]). Applying Prop. 2.2 to Eq. (7), we have:

$$\begin{aligned}
\mathcal{G}(\mathbf{B}_\ell) &\leq \overline{\mathbf{y}} + \mathbf{A}^\ell \cdot (\mathcal{G}(\mathbf{B}_0) - \overline{\mathbf{y}}) \\
&\leq \overline{\mathbf{y}} + \frac{1}{4} \cdot \|\mathbf{A}\|_\infty^\ell \cdot (1, \ldots, 1)^{\mathrm{T}} && \text{(By Prop. 2.3)} \\
&\leq \overline{\mathbf{y}} + \frac{\log(1+\varepsilon)}{n-k-q} \cdot (1, \ldots, 1)^{\mathrm{T}} \\
&= \left(\log\left((1+\varepsilon)^{1/(n-k-q)} \overline{h}\right)\right) \cdot (1, \ldots, 1)^{\mathrm{T}}.
\end{aligned}$$

By the definition of $\mathcal{G}(\cdot)$, this implies the desired upper bounds of $\mathrm{vol}(\mathbf{B}_{[1,ik+q]})/\mathrm{vol}(\mathcal{L})^{(ik+q)/n}$ in Th. 5. The remaining inequality is ensured by Step 2 of Alg. 2. This completes the proof of Th. 5. □

5 HKZ-slide reduction

Notice that solving (r,k)-DSP for larger $r \in [2, k-2]_\mathbb{Z}$ is much more expensive than solving SVP in rank k [DM13]. Both Th. 4 and Th. 5 suggest that one might replace the exact (r,k)-DSP solver in block-Rankin reduction [LN14, Alg. 3] with cheaper procedures for approximating (r,k)-DSP.

Based on the three observations illustrated in Sect. 1, we replace the exact (r,k)-DSP solver in [LN14, Alg. 3] with any *r-partial-HKZ* solver in rank k: An n-rank basis $\mathbf{B}$ is *r-partial-HKZ-reduced* for $r \in [1,n]_\mathbb{Z}$ if it is size-reduced and $\mathbf{B}_{[i,n]}$ is SVP-reduced for $i = 1, \ldots, r$. For $r = O(1)$, it is cheaper than full HKZ-reduction (defined as in Sect. 2) in practice.

If a projected block $\mathbf{B}_{[i,j]}$ of rank $k = j - i + 1 > r$ is r-partial-HKZ-reduced, Hanrot and Stehlé [HS07, Lem. 3] proved that it approximates (r,k)-DSP quite well especially for small $r = O(1)$:

$$\mathrm{vol}(\mathbf{B}_{[i,i+r-1]}) \leq \prod_{i=k-r+1}^{k} \gamma_i^{\frac{k-r}{2(i-1)}} \cdot \mathrm{vol}(\mathbf{B}_{[i,j]})^{r/k}.$$

We now present our cheaper analogue of block-Rankin-reduction [LN14]:

Definition 4 (HKZ-slide reduction). *Let n, k, p, r be integers such that $n = pk \geq 2k$ with $k > r \geq 1$. A basis $\mathbf{B} \in \mathbb{R}^{m \times n}$ is* (r,k)-HKZ-slide-reduced *if it is size-reduced and satisfies the following two sets of conditions.*

1. *Primal conditions: for all $i \in [1, p-1]_\mathbb{Z}$, the block $\mathbf{B}_{[ik+1,ik+k]}$ is r-partial-HKZ-reduced.*
2. *Dual conditions: for all $i \in [1, p-2]_\mathbb{Z}$, the reversed dual basis $(\mathbf{B}_{[ik+r+1,ik+k+r]})^{-s}$ of block $\mathbf{B}_{[ik+r+1,ik+k+r]}$ is r-partial-HKZ-reduced.*

HKZ-slide reduction can be viewed a higher-dimensional generalization of the original slide reduction [GN08a] (corr. to $r = 1$), and is also a particular instantiation of our generic slide reduction presented in Sect. 3. Therefore, we apply Th. 2 to deduce the following estimate for approximating HSVP:

Theorem 6. *Let n, k, p, r be integers such that $n = pk \geq 2k$ with $k > r \geq 1$. If $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n) \in \mathbb{R}^{m \times n}$ is a (r, k)-HKZ-slide-reduced basis of a lattice $\mathcal{L}$, then*

$$\|\mathbf{b}_1\| \leq \sqrt{\gamma_k} \prod_{i=k-r+1}^{k} \gamma_i^{\frac{n-k}{2(i-1)r}} \cdot \mathrm{vol}(\mathcal{L})^{1/n}.$$

Proof. Th. 2 with $(h, q) = (\prod_{i=k-r+1}^{k} \gamma_i^{\frac{1}{2(i-1)r}}, 0)$ implies: $\mathrm{vol}(\mathbf{B}_{[1,k]}) \leq \prod_{i=k-r+1}^{k} \gamma_i^{\frac{(n-k)k}{2(i-1)r}} \cdot \mathrm{vol}(\mathcal{L})^{k/n}$. Since $\mathbf{B}_{[1,k]}$ is r-partial-HKZ-reduced and hence SVP-reduced, we have $\|\mathbf{b}_1\| \leq \sqrt{\gamma_k} \cdot \mathrm{vol}(\mathbf{B}_{[1,k]})^{1/k}$. By combining the above two inequalities, Th. 6 follows. □

Applying Th. 4, it is not hard to deduce that given as input an LLL-reduced basis $\mathbf{B}_0 \in \mathbb{R}^{m \times n}$ of a lattice $\mathcal{L}$ of rank $n = pk \geq 2k$, Alg. 2 makes at most $\left\lceil \frac{(n-k)n^2}{k \log(1+\varepsilon)} \right\rceil$ GSR tours, and can output a (r, k)-HKZ-slide-reduced basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of $\mathcal{L}$ (up to a slack factor ε). In particular, we have:

$$\|\mathbf{b}_1\| \leq (1+\varepsilon)^{\frac{n-k}{(k-r)r}} \sqrt{\gamma_k} \prod_{i=k-r+1}^{k} \gamma_i^{\frac{n-k}{2(i-1)r}} \cdot \mathrm{vol}(\mathcal{L})^{1/n}. \tag{13}$$

It follows from Th. 5 that with fewer tours, Alg. 2 can still obtain a basis whose RHF is almost the same as that in Eq. (13) for full HKZ-slide-reduction. Under the weak version of the Gaussian heuristic (cf. Sect. 2.1), Th. 5 can also be used to model the practical behaviour of HKZ-slide-reduction:

Corollary 5. *Let n, k, p, r be integers such that $n = pk \geq 2k$ with $k > r \geq 1$. Let $0 < \varepsilon \leq 0.4$. Given as input an LLL-reduced basis $\mathbf{B}_0$ of an n-rank lattice $\mathcal{L}$ in $\mathbb{R}^m$ and access to an oracle for r-partial-HKZ-reducing any lattice of rank k, if terminated after $\left\lceil \frac{n^2 \ln \frac{n-k}{4 \log(1+\varepsilon)}}{8(k-r)r} \right\rceil$ $(\in O(\frac{n^2 \ln \frac{n}{\varepsilon}}{(k-r)r}))$ GSR tours, then Alg. 2 with one more execution to its Step 2 outputs a basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of $\mathcal{L}$ satisfying the following properties.*

1. *Worst-case: $\|\mathbf{b}_1\| \leq (1+\varepsilon)\sqrt{\gamma_k} \prod_{i=k-r+1}^{k} \gamma_i^{\frac{n-k}{2(i-1)r}} \cdot \mathrm{vol}(\mathcal{L})^{1/n}$.*
2. *Average-case: Suppose that every projected lattice Λ of rank $i \in [k-r+1, k]$ passed to r-partial-HKZ-reductions satisfies $\lambda_1(\Lambda) \leq (1+\xi)\,GH(i) \cdot \mathrm{vol}(\Lambda)^{1/i}$ for some $\xi \in (0, \sqrt{2}-1]$, we have*

$$\|\mathbf{b}_1\|/\mathrm{vol}(\mathcal{L})^{1/n} \leq (1+\varepsilon)(1+\xi)\,GH(k) \prod_{i=k-r+1}^{k} ((1+\xi)\,GH(i))^{\frac{n-k}{(i-1)r}} \approx \sqrt{\frac{k}{2\pi e}} \cdot \prod_{i=k-r+1}^{k} \left(\frac{i}{2\pi e}\right)^{\frac{n-k}{2(i-1)r}}.$$

Proof. Applying Th. 5 with $(h, q) = (\prod_{i=k-r+1}^{k} \gamma_i^{\frac{1}{2(i-1)r}}, 0)$ to the worst-case, we have

$$\mathrm{vol}(\mathbf{B}_{[1,k]}) \leq (1+\varepsilon)^k \prod_{i=k-r+1}^{k} \gamma_i^{\frac{(n-k)k}{2(i-1)r}} \cdot \mathrm{vol}(\mathcal{L})^{k/n}.$$

One more execution to Step 2 ensures that the first k-rank block $\mathbf{B}_{[1,k]}$ is SVP-reduced. Thus, $\|\mathbf{b}_1\| \leq \sqrt{\gamma_k} \cdot \mathrm{vol}(\mathbf{B}_{[1,k]})^{1/k}$. By combining the above two inequalities, this implies Item 1.

Thanks to the assumption, Item 2 follows from replacing each $\sqrt{\gamma_i}$ in the worst-case with $(1+\epsilon)\mathrm{GH}(i)$. This proves Cor. 5. □

5.1 Theoretical comparison

Hermite's constant γ_i satisfies Mordell's inequality [Mor44] and Newman's inequality [New63]: $\gamma_k^{(i-1)/(k-1)} \leq \gamma_i \leq \gamma_k^{k/i}$ for any integers $2 \leq i \leq k$. $\mathrm{GH}(i)$ satisfies similar inequalities (cf. [ABLR21, Fact 3] and [LN20, §4.2]): $\mathrm{GH}(k)^{(i-1)/(k-1)} \leq \mathrm{GH}(i) \leq \mathrm{GH}(k)^{k/i}$ for any integers $51 \leq i \leq k$. Thus, for $k \geq 50 + r$, the larger r is, the larger both $\prod_{i=k-r+1}^{k} \gamma_i^{\frac{1}{(i-1)r}}$ and $\prod_{i=k-r+1}^{k} \mathrm{GH}(i)^{\frac{1}{(i-1)r}}$:

$$\gamma_k^{\frac{1}{k-1}} \leq \prod_{i=k-r+1}^{k} \gamma_i^{\frac{1}{(i-1)r}} \leq \gamma_k^{\frac{1}{k-r}} \quad \text{and} \quad \mathrm{GH}(k)^{\frac{1}{k-1}} \leq \prod_{i=k-r+1}^{k} \mathrm{GH}(i)^{\frac{1}{(i-1)r}} \leq \mathrm{GH}(k)^{\frac{1}{k-r}}.$$

Therefore, both HSVP estimates in Cor. 5 would not outperform those for the original slide reduction [GN08a] with the same block size k (see Cor. 4 with $(\delta, q) = (1, 0)$).

When $r = O(1) > 1$ is sufficiently small w.r.t. k, (r, k)-HKZ-slide-reduction can achieve almost the same HSVP estimates as those for the original slide reduction, but with fewer tours. This is because:

$$\prod_{i=k-r+1}^{k} \gamma_i^{\frac{1}{(i-1)r}} \approx \gamma_k^{\frac{1}{k-1}}, \quad \prod_{i=k-r+1}^{k} \mathrm{GH}(i)^{\frac{1}{(i-1)r}} \approx \mathrm{GH}(k)^{\frac{1}{k-1}}, \text{ and } \frac{n^2 \ln \frac{n-k}{4\log(1+\varepsilon)}}{8(k-r)r} < \frac{n^2 \ln \frac{n-k}{4\log(1+\varepsilon)}}{8(k-1)}.$$

For instance, when $(n, k, r) = (170, 85, 10)$, $\mathrm{GH}(k)\prod_{i=k-r+1}^{k} \mathrm{GH}(i)^{\frac{n-k}{(i-1)r}} \approx 5.47965$ is close to $\mathrm{GH}(k)^{\frac{n-1}{k-1}} \approx 5.36823$, while $\frac{(k-r)r}{k-1} = 8.93$. This means that with well-chosen constant $r > 1$, (r, k)-HKZ-slide-reduction might provide better time/quality trade-offs in practice than the original slide reduction. This hypothesis is confirmed by both our experiments in Sect. 5.2 and simulation in Sect. 6.2.

5.2 Experiments

We implemented HKZ-slide reduction in the G6K framework of [ADH+19], which (among a lot of other things) provides an interface to an SVP algorithm based on sieving: their work shows that basic (called *naive* in [ADH+19]) BKZ based on sieving starts outperforming state-of-the-art enumeration-based methods for block sizes below 80, and more carefully tuned variants well below 65. The authors observed that, in fact, the output of this algorithm seems to approximate partial-HKZ reduction. For our implementation we treated the SVP algorithm of G6K as a r-partial-HKZ reduction oracle for arbitrary integer $r \leq 15$.

To confirm the hypothesis in Sect. 5.1, we tested (r, k)-HKZ-slide reduction for $r \in \{1, 5, 10, 15\}$ and $k \in \{60, 85\}$ on lattices from the lattice challenge [LR20]. To avoid issues with block-sizes not dividing the rank, we selected the rank as the largest integer multiple of k such that the algorithm did not run into numerical issues. For $k = 60$ and $k = 85$, this was $n = 180$ (i.e. $p = 3$ blocks) and $n = 170$ (i.e. $p = 2$ blocks), respectively. The results are shown in Figures (1a) and (1c). All datapoints are averaged (in both axes) over the same 10 lattices (challenge seeds 0 to 9), which were preprocessed using `fplll` [FPL19] with block size 45 (for $k = 60$) and 60 (for $k = 85$).[2]

Fig. (1a) demonstrates that for relatively small block sizes, (r, k)-HKZ-slide reduction actually behaves better than expected: not only does a larger r lead to a faster convergence (which is expected), all of the tested r also lead to better output quality. This can at least in part be explained by the relatively small block size and the corresponding approximation error of the Gaussian heuristic. This is somewhat supported by Fig. (1c), where at least the overlaps $r = 5$ and $r = 15$ behave as expected: faster convergence but poorer output quality. (Note though that the difference in output quality between overlaps 1 and 5 is minor.) However, the case of $r = 10$ seems to be a special case that behaves exceptionally well even for large block size. We cannot offer an explanation of this phenomenon beyond baseless speculation at this point and leave an in-depth investigation to future work.

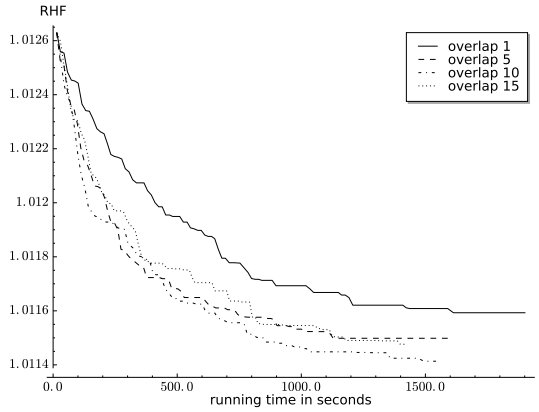
In summary, we believe that the results give sufficient evidence that the time/quality trade-off achieved by HKZ-slide reduction can indeed be very favourable when considering overlaps larger than 1 (i.e. beyond slide reduction).

HKZ-slide vs. BKZ. To put the results into context, we also compare HKZ-slide reduction with the BKZ variants implemented in G6K on the same lattices. For HKZ-slide reduction we chose $r = 10$ for the above reasons. We compared to three "standard" variants of BKZ:

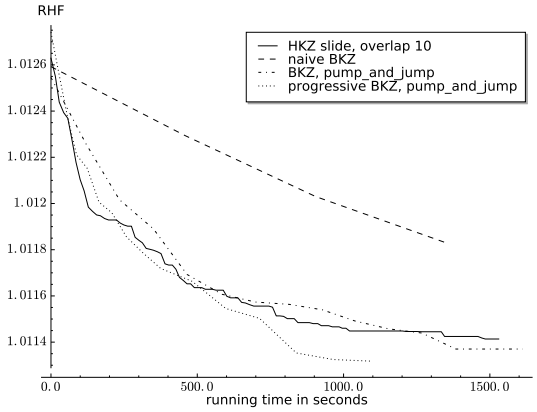
- Naive BKZ, which treats the SVP algorithm as a black box;
- The "Pump and Jump" (PnJ) variant, which recycles computation done during previous calls to the SVP algorithm to save cost in later calls;
- A progressive variant of the PnJ strategy, which starts with smaller block sizes and successively runs BKZ tours with increasing block sizes.

We left all parameters for the PnJ versions at their default. [ADH+19] reported that some fine-tuning can improve the PnJ variant further, but since our goal is only to demonstrate competetiveness of HKZ-slide reduction rather than a fine-grained comparison, we do not believe such fine-tuning is necessary here. Naive BKZ and the PnJ variant were called with the same block size (on the same bases as HKZ-slide reduction) and the number of tours was chosen such that the running time was roughly in the ballpark of the HKZ-slide reduction experiments. For progressive PnJ, we ran 1 tour of each block size starting from $k-10$ up to $k+5$, where k is the block size chosen for the other algorithms. This choice of block sizes is somewhat arbitrary, but was also made in an attempt to ensure that progressive PnJ has roughly the same running time as the other algorithms.

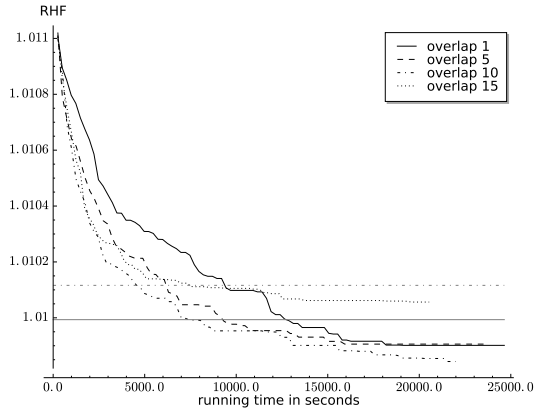
[2] G6K [ADH+19] calls `fplll` with default parameters to pre-reduce the lattice basis and stores it, so that future invocations do not need to redo that again.



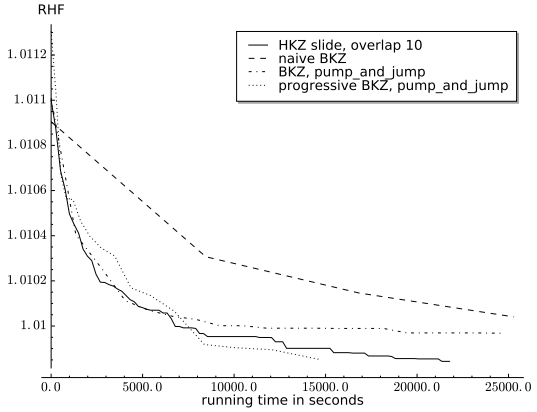
(a) HKZ-slide reduction on a lattice with rank 180 and block size 60



(b) Comparison of HKZ-slide reduction and BKZ on a lattice with rank 180 and block size 60



(c) HKZ-slide reduction on a lattice with rank 170 and block size 85. Solid and dotted horizontal lines correspond to expected upper bounds of RHFs in Cor. 4.2 and Cor. 5.2 (ignoring both ε and ξ) for overlap 1 and 10, respectively.



(d) Comparison of HKZ-slide reduction and BKZ on a lattice with rank 170 and block size 85

Fig. 1: Comparison of HKZ-slide-reduction with different overlaps and with various BKZ variants

The results are shown in Figures (1b) and (1d) respectively. They show that HKZ-slide reduction is able to clearly outperform the naive version of BKZ, but it also seems to be better than PnJ. However, while progressive PnJ is roughly on par with HKZ-slide reduction at the beginning, it seems to have the edge towards latter stages of the execution. We believe this is because just like naive BKZ, both our theoretical analyses and experiments show that much progress of HKZ-slide reduction is done at the beginning of the execution, and quickly the basis quality becomes excellent; the rest of the execution takes much longer, for a really minor quality improvement. At this stage, progressive BKZ switches to a block size $> k$, whereas our implementation of HKZ-slide reduction maintains the block size. This is a perfect example of the advantages that "progressiveness" adds to block reduction algorithms. But even without this, we consider HKZ-slide reduction at least competitive with progressive PnJ.

Caveats. We focused our attention in these experiments on the RHF that the different algorithms achieve in a given amount of time. This has been established as the main measure of output quality for lattice reduction, since they are usually used to find short vectors. When targeting a short vector, (HKZ-)slide reduction has the advantage that it is focusing on improving a set of pivot points distributed across the basis, while BKZ attempts to improve the entire basis. This seems to result in lower cost for slide reduction. But finding short vectors is not the only use case: often one is interested in a basis that is reduced according to a more global measure, e.g. one wants all basis vectors to be short or the GSO vectors should not drop off too quickly. In this case, BKZ seems to be the more natural choice.

Potential Improvements. We did not make any attempts to fine-tune the SVP oracle to HKZ-slide reduction and its parameters. The SVP-oracle itself has several parameters which potentially influence how well it performs as a r-partial-HKZ reduction oracle. We leave such a fine-tuning as interesting future work.

We also notice that applying BKZ/PnJ with increasing block sizes results in significant improvements. It stands to reason that including an element of "progressiveness" could significantly improve HKZ-slide reduction. However, the strength of HKZ-slide reduction of focusing its attention on pivot points instead of the entire basis could be a disadvantage here: it may not be as suitable as a preprocessing for other algorithms, possibly including itself. Still, finding an effective way of naturally progressing slide reduction might lead to improvements, but we believe simply increasing the block size is unlikely to be sufficient here. Finally, given the above observations, a natural approach seems to be to first use progressive BKZ/PnJ as a preprocessing and then run HKZ-slide reduction in the final step to find a short vector.

6 A simulator for predicting (HKZ-)slide reduction

We now present a simple and efficient simulator to predict the performances of (HKZ-)slide reduction with high block size k ($\geq 50 + r$) in high rank. It predicts the evolution of RHF with the number of tours. Our simulation is fairly consistent with experiments. We believe that our simulator can be used to explore the "optimal" parameters (k, r) in practice (dominating time/quality trade-offs), and predict approximately what can be achieved using much larger computational power than used in our experiments.

6.1 (HKZ-)slide simulator

Both Cor. 4 and Cor. 5 claim the worst-case number of tours for computing a basis of the desired quality. In practice, it may be of interest to estimate the actual number of tours more accurately, especially the impact of the parameter r.

The goal of our simulator is to predict the RHF depending on the number of tours during the execution of Alg. 2 for (HKZ-)slide-reduction. For simplicity, we just consider the case $n = pk$ (i.e., $q = 0$) in what follows.

For $\ell = 0, 1, \ldots$, let $\mathbf{B}_\ell = (\mathbf{b}_1^{(\ell)}, \ldots, \mathbf{b}_n^{(\ell)})$ be the current basis at the end of the ℓ-th tour of Alg. 2 and let $\mathrm{RHF}(\mathbf{B}_\ell)$ denote its RHF. Let $\mathbf{A} \in \mathbb{Q}^{(p-1)\times(p-1)}$ and $\overline{\mathbf{y}} \in \mathbb{R}^{p-1}$ be respectively the matrix and vector defined as in Prop. 1 and Prop. 2.2. Let $\underline{\mathbf{A}}_1$ denote the first row of $\mathbf{A}$. Then Eq. (7) implies: for $\ell \in \mathbb{Z}^+$, $\mathcal{G}(\mathbf{B}_\ell) \leq \overline{\mathbf{y}} + \mathbf{A}^\ell \cdot (\mathcal{G}(\mathbf{B}_0) - \overline{\mathbf{y}})$, in particular,

$$\mathcal{G}_1(\mathbf{B}_\ell) \leq \overline{y}_1 + \underline{\mathbf{A}}_1 \cdot (\mathbf{A}^{\ell-1}) \cdot (\mathcal{G}(\mathbf{B}_0) - \overline{\mathbf{y}}).$$

Step 2 in the $(\ell+1)$-th tour of Alg. 2 ensures: $\|\mathbf{b}_1^{(\ell+1)}\| \leq \sqrt{\gamma_k} \cdot \mathrm{vol}((\mathbf{B}_\ell)_{[1,k]})^{1/k}$, or equivalently,

$$\log \mathrm{RHF}(\mathbf{B}_{\ell+1}) \leq \frac{\log \gamma_k}{2(n-1)} + \frac{n-k}{n-1} \cdot \mathcal{G}_1(\mathbf{B}_\ell) \quad \text{for each} \quad \ell \geq 0.$$

We notice a loop invariant: $\|\mathbf{b}_1\|$ never increases during the execution of Alg. 2. This implies the following key relation between RHFs of two consecutive tours: for $\ell = 0, 1, \ldots$,

$$\log \mathrm{RHF}(\mathbf{B}_{\ell+1}) \leq \min\left\{\log \mathrm{RHF}(\mathbf{B}_\ell), \frac{\log \gamma_k}{2(n-1)} + \frac{n-k}{n-1}\left(\overline{y}_1 + \underline{\mathbf{A}}_1 \cdot (\mathbf{A}^{\ell-1}) \cdot (\mathcal{G}(\mathbf{B}_0) - \overline{\mathbf{y}})\right)\right\},$$

which is equivalent to the inequality below (with $h = \prod_{i=k-r+1}^{k} \gamma_i^{\frac{1}{2(i-1)r}}$)

$$\mathrm{RHF}(\mathbf{B}_{\ell+1}) \leq \min\left\{\mathrm{RHF}(\mathbf{B}_\ell), \gamma_k^{\frac{1}{2(n-1)}} \cdot h^{\frac{n-k}{n-1}} \cdot 2^{\frac{n-k}{n-1} \cdot \underline{\mathbf{A}}_1 \cdot (\mathbf{A}^{\ell-1}) \cdot \left(\mathcal{G}(\mathbf{B}_0) - (\log h) \cdot (1,\ldots,1)^{\mathrm{T}}\right)}\right\}.$$

Cor. 5.2 and the weak version of the Gaussian heuristic suggest that when every projected lattice Λ of rank $i \in [k-r+1, k]$ passed to r-partial-HKZ-reductions is "random", one might replace each Hermite's constant γ_i inside the above equation with its heuristic analog $\mathrm{GH}(i)^2$. This gives rise to a heuristic (HKZ-)slide simulator for predicting RHF especially with high block size, namely Alg. 3.

We make a remark on Step 3 of Alg. 3. In the real implementation of (HKZ-)slide-reduction, if the $(i+1)$-th tour does not update the first basis vector, then $\mathrm{RHF}(\mathbf{B}_{i+1}) = \mathrm{RHF}(\mathbf{B}_i)$; otherwise, the $(i+1)$-th tour makes the first basis vector shorter, i.e., $\mathrm{RHF}(\mathbf{B}_{i+1}) < \mathrm{RHF}(\mathbf{B}_i)$. However, in the context of simulation, if the prediction value of $\mathrm{RHF}(\mathbf{B}_i)$ is already less than the dynamical system prediction $f(i) := \mathrm{GH}(k)^{\frac{1}{n-1}} \cdot \widetilde{h}^{\frac{n-k}{n-1}} \cdot 2^{\frac{n-k}{n-1} \cdot \underline{\mathbf{A}}_1 \cdot (\mathbf{A}^{i-1}) \cdot \left(\mathcal{G}(\mathbf{B}_0) - (\log \widetilde{h}) \cdot (1,\ldots,1)^{\mathrm{T}}\right)}$, then the simulator predicts $\mathrm{RHF}(\mathbf{B}_{i+1}) = \mathrm{RHF}(\mathbf{B}_i)$; otherwise, the simulator predicts $\mathrm{RHF}(\mathbf{B}_{i+1}) = f(i)$. In brief, the minimum restriction at Step 3 of Alg. 3 ensures that the prediction value of $\mathrm{RHF}(\mathbf{B}_i)$ never increases during the execution of Alg. 3, just like that in the real implementation of (HKZ-)slide-reduction. For example, consider an input basis $\mathbf{B}_0$ which is already SVP-reduced and whose first vector is unusually short, the simulator without this minimum restriction at its Step 3 would predict $\mathrm{RHF}(\mathbf{B}_1) = f(0) > \mathrm{RHF}(\mathbf{B}_0)$, which contradicts the fact "$\mathrm{RHF}(\mathbf{B}_0) = \mathrm{RHF}(\mathbf{B}_1) = \cdots = \mathrm{RHF}(\mathbf{B}_i) = \cdots$" during the real execution of (HKZ-)slide-reduction.

Algorithm 3 A simulator of (HKZ-)slide reduction

Input: A block size $k > r \geq 1$ such that $k \geq 50 + r$, a number $\ell \geq 1$ of tours, and the bit profile $\mathcal{G}(\mathbf{B}_0)$ of an n-rank basis $\mathbf{B}_0$ where $n = pk \geq 2k$.

Output: A prediction for the RHF right after ℓ tours.

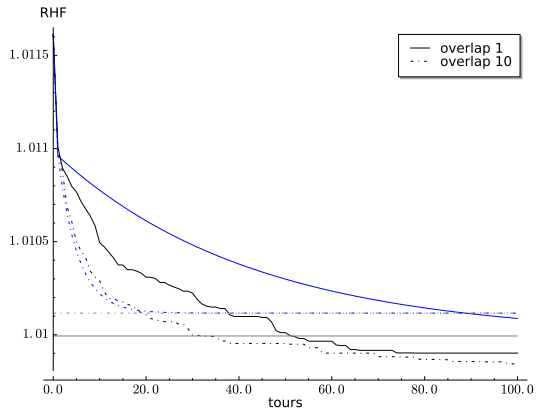
1: Compute the heuristic analog $\widetilde{h}$ of $h = \prod_{i=k-r+1}^{k} \gamma_i^{\frac{1}{2(i-1)r}}$: $\widetilde{h} \leftarrow \prod_{i=k-r+1}^{k} \mathrm{GH}(i)^{\frac{1}{(i-1)r}}$
2: **for** $i = 0$ to $\ell - 1$ **do**
3: $\quad \mathrm{RHF}(\mathbf{B}_{i+1}) \leftarrow \min\left\{\mathrm{RHF}(\mathbf{B}_i), \mathrm{GH}(k)^{\frac{1}{n-1}} \cdot \widetilde{h}^{\frac{n-k}{n-1}} \cdot 2^{\frac{n-k}{n-1} \cdot \underline{\mathbf{A}}_1 \cdot (\mathbf{A}^{i-1}) \cdot \left(\mathcal{G}(\mathbf{B}_0) - (\log \widetilde{h}) \cdot (1,\ldots,1)^{\mathrm{T}}\right)}\right\}$
4: **end for**
5: **return** $\mathrm{RHF}(\mathbf{B}_\ell)$.

6.2 Verification and comparison

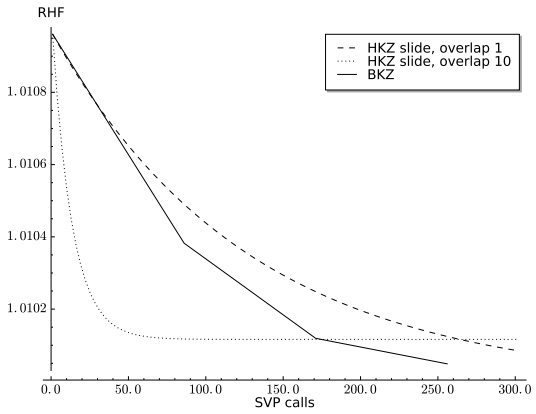
Consistency with experiments. To verify that the simulator predicts the output of the HKZ-slide algorithm accurately, we tested it against our experimental data from Sect. 5.2 for the larger set of experiments ($k = 85$). We disregarded the $k = 60$ parameterization here, since the Gaussian heuristic is less accurate in smaller ranks. (Note that we require the Gaussian heuristic to be fairly accurate starting from rank $(k - r + 1)$.) The results are shown in Fig. (2a).

There are small discrepancies between the predicted and the actual behavior of the algorithm, but given the small range on the y-axis, this can be viewed as a rough match. The discrepancies may be explained similarly as the "head concavity phenomenon" in BKZ [BSW18]: if, by chance, the SVP oracle finds a vector shorter than expected in the first block, this vector is never displaced during the execution of the algorithm (unless for an even shorter vector). We leave a further analysis along the lines of [BSW18] to future work.

Overall we believe that the results support the hypothesis that the simulator may be useful for the purposes we laid out in the preface of Sect. 6.



(a) Simulation vs. experimental results for rank 170 lattices with block size 85 and overlap 1 and 10. Black lines correspond to experimental data, blue lines to simulations on the same lattices, Solid and dotted horizontal lines correspond to expected upper bounds of RHFs in Cor. 4.2 and Cor. 5.2 (ignoring both ε and ξ) for overlap 1 and 10, respectively.



(b) (HKZ-)slide simulator vs. BKZ simulator on rank 170 lattices with block size 85. SVP calls in rank < 85 in BKZ are ignored for simplicity.

Fig. 2: Evaluation of HKZ-slide simulator

Comparison with BKZ. We give a comparison of Chen-Nguyen's BKZ simulator [CN11] and our (HKZ-)slide simulator on the lattices in our target set (cf. Sect. 5.2) in Fig. (2b). The simulation matches the behaviour observed in both theory (cf. Sect. 5.1) and the experiments: HKZ-slide reduction converges much faster to a small RHF than naive BKZ, but the achieved RHF is slightly worse for larger overlaps.

References

[ABF+20] M. R. Albrecht, S. Bai, P.-A. Fouque, P. Kirchner, D. Stehlé, and W. Wen. Faster enumeration-based lattice reduction: Root Hermite factor $k^{1/(2k)}$ in time $k^{k/8+o(k)}$. In *CRYPTO*, pages

186–212, 2020.

[ABLR21] M. R. Albrecht, S. Bai, J. Li, and J. Rowell. Lattice reduction with approximate enumeration oracles: Practical algorithms and concrete performance. In *CRYPTO*, pages 732–759, 2021.

[ADH+19] M. R. Albrecht, L. Ducas, G. Herold, E. Kirshanova, E. W. Postlethwaite, and M. Stevens. The general sieve kernel and new records in lattice reduction. In *EUROCRYPT*, pages 717–746, 2019.

[ADPS16] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe. Post-quantum key exchange—a new hope. In *USENIX*, pages 327–343, 2016.

[Ajt96] M. Ajtai. Generating hard instances of lattice problems. In *STOC*, pages 99–108, 1996.

[Ajt98] M. Ajtai. The shortest vector problem in L_2 is NP-hard for randomized reductions (extended abstract). In *STOC*, pages 10–19, 1998.

[ALNS20] D. Aggarwal, J. Li, P. Q. Nguyen, and N. Stephens-Davidowitz. Slide reduction, revisited — filling the gaps in SVP approximation. In *CRYPTO*, pages 274–295, 2020.

[AWHT16] Y. Aono, Y. Wang, T. Hayashi, and T. Takagi. Improved progressive BKZ algorithms and their precise cost estimation by sharp simulator. In *EUROCRYPT*, pages 789–819, 2016.

[Bli14] H. F. Blichfeldt. A new principle in the geometry of numbers, with some applications. *Transactions of the American Mathematical Society*, 15:227–235, 1914.

[BSW18] S. Bai, D. Stehlé, and W. Wen. Measuring, simulating and exploiting the head concavity phenomenon in BKZ. In *ASIACRYPT*, pages 369–404, 2018.

[CN98] J.-Y. Cai and A. Nerurkar. Approximating the SVP to within a factor $(1+1/\dim^{\varepsilon})$ is NP-hard under randomized reductions. In *CCC*, page 46, 1998.

[CN11] Y. Chen and P. Q. Nguyen. BKZ 2.0: Better lattice security estimates. In *ASIACRYPT*, pages 1–20, 2011.

[CS98] J. Conway and N. Sloane. *Sphere packings, lattices and groups*. Springer, third edition, 1998.

[DM13] D. Dadush and D. Micciancio. Algorithms for the densest sub-lattice problem. In *SODA*, pages 1103–1122, 2013.

[FPL19] FPLLL development team. FPLLL, a lattice reduction library. Available at https://github.com/fplll/fplll, 2019.

[GHGKN06] N. Gama, N. Howgrave-Graham, H. Koy, and P. Q. Nguyen. Rankin's constant and blockwise lattice reduction. In *CRYPTO*, pages 112–130, 2006.

[GHGN06] N. Gama, N. Howgrave-Graham, and P. Q. Nguyen. Symplectic lattice reduction and NTRU. In *EUROCRYPT*, pages 233–253, 2006.

[GN08a] N. Gama and P. Q. Nguyen. Finding short lattice vectors within Mordell's inequality. In *STOC*, pages 207–216, 2008.

[GN08b] N. Gama and P. Q. Nguyen. Predicting lattice reduction. In *EUROCRYPT*, pages 31–51, 2008.

[GPV08] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pages 197–206, 2008.

[HPS11] G. Hanrot, X. Pujol, and D. Stehlé. Analyzing blockwise lattice algorithms using dynamical systems. In *CRYPTO*, pages 447–464, 2011.

[HR07] I. Haviv and O. Regev. Tensor-based hardness of the shortest vector problem to within almost polynomial factors. In *STOC*, pages 469–477, 2007. Full version at *Theory of Computing 2012*.

[HS07] G. Hanrot and D. Stehlé. Improved analysis of Kannan's shortest lattice vector algorithm. In *CRYPTO*, pages 170–186, 2007.

[Kho05] S. Khot. Hardness of approximating the shortest vector problem in lattices. *Journal of the ACM*, 52(5):789–808, 2005. Preliminary version in *FOCS 2004*.

[LLL82] A. K. Lenstra, H. W. Lenstra Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:366–389, 1982.

[LN14] J. Li and P. Q. Nguyen. Approximating the densest sublattice from Rankin's inequality. *LMS J. Comput. Math.*, 17(Special Issue A):92–111, 2014. Contributed to ANTS-XI, 2014.

[LN20] J. Li and P. Q. Nguyen. A complete analysis of the BKZ lattice reduction algorithm. Available at https://eprint.iacr.org/2020/1237.pdf, 2020.

[Lov86] L. Lovász. *An algorithmic theory of numbers, graphs and convexity*, volume 50 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM, 1986.

[LR20] R. Lindner and M. Ruckert. TU Darmstadt lattice challenge. Available at http://www.latticechallenge.org/, 2020.

[MH98] J. Milnor and D. Husemoller. *Symmetrie Bilinear Forms*. Springer, third edition, 1998.

[Mic01] D. Micciancio. The shortest vector in a lattice is hard to approximate to within some constant. *SIAM J. Comput.*, 30(6):2008–2035, 2001. Preliminary version in *FOCS 1998*.

[Mic12] D. Micciancio. Inapproximability of the shortest vector problem: Toward a deterministic reduction. *Theory Comput.*, 8(1):487–512, 2012.
[Mor44] L. J. Mordell. Observation on the minimum of a positive quadratic form in eight variables. *Journal of the London Mathematical Society*, 19:3–6, 1944.
[MW16] D. Micciancio and M. Walter. Practical, predictable lattice basis reduction. In *EUROCRYPT*, pages 820–849, 2016.
[Neu17] A. Neumaier. Bounding basis reduction properties. *Des. Codes, Cryptogr.*, 84:237–259, 2017.
[New63] M. Newman. Bounds for cofactors and arithmetic minima of quadratic forms. *Journal of the London Mathematical Society*, 38:215–217, 1963.
[Pei09] C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In *STOC*, pages 333–342, 2009.
[PT08] G. Pataki and M. Tural. On sublattice determinants in reduced bases. Aailable at https://arxiv.org/pdf/0804.4014.pdf, 2008.
[Ran53] R. A. Rankin. On positive definite quadratic forms. *J. London Math. Soc.*, 28:309–314, 1953.
[Reg04] O. Regev. Lattices in computer science. Lecture 8: dual lattices. Available at http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.215.4052, 2004.
[Reg05] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, pages 84–93, 2005.
[Sch87] C. P. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theoretical Computer Science*, 53:201–224, 1987.
[Sch21] C. P. Schnorr. Fast factoring integers by SVP algorithms. Available at https://eprint.iacr.org/2021/232, 2021.
[SE94] C. P. Schnorr and M. Euchner. Lattice basis reduction: improved practical algorithms and solving subset sum problems. *Mathematical Programming*, 66:181–199, 1994.
[SH95] C. P. Schnorr and H. H. Hörner. Attacking the Chor-Rivest cryptosystem by improved lattice reduction. In *EUROCRYPT*, pages 1–12, 1995.
[Wal15] M. Walter. Lattice point enumeration on block reduced bases. In *ICITS*, pages 269–282, 2015.
[Wal21] M. Walter. The convergence of slide-type reductions. In *PKC*, pages 45–67, 2021. Available at https://eprint.iacr.org/2020/1409.

A The Micciancio-Walter DBKZ algorithm

We recall Micciancio and Walter's DBKZ algorithm [MW16] and slightly generalize DBKZ by allowing for the use of a δ-SVP-oracle (see Alg. 4).

Algorithm 4 The Micciancio-Walter DBKZ algorithm [MW16, Alg. 1]

Input: A block size $k \geq 2$, number of tours N, and a basis $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n) \in \mathbb{Z}^{m \times n}$.
Output: A new basis of $\mathcal{L}(\mathbf{B})$.
1: **for** $\ell = 1$ **to** N **do**
2: **for** $i = 1$ **to** $n - k$ **do** δ-SVP-reduce $\mathbf{B}_{[i,i+k-1]}$. //Steps 2-3 use a δ-SVP oracle in rank k.
3: **for** $j = n - k + 1$ **to** 1 **do** δ-DSVP-reduce $\mathbf{B}_{[j,j+k-1]}$.
4: **end for**
5: δ-SVP-reduce $\mathbf{B}_{[1,k]}$.
6: **return** $\mathbf{B}$.

Neumaier introduced in [Neu17] a simplification of [HPS11]'s method to analyse DBKZ. It allows to deduce the following properties:

Theorem 7 (See [LN20, App. A.1]). *For integers $n > k \geq 2$, a factor $1 \leq \delta < \left(\frac{3}{2}\right)^{(k-1)/4}$, an input LLL-reduced basis $\mathbf{B}_0 \in \mathbb{Z}^{m \times n}$ for a lattice $\mathcal{L} \subseteq \mathbb{Z}^m$, and $N := \left\lceil \left(\frac{n^2}{8k(k-1)} + \frac{1}{2}\right) \ln \frac{n}{4\log(1+\varepsilon)} \right\rceil \in O(\frac{n^2}{k^2} \log \frac{n}{4\varepsilon})$ for some $\varepsilon \in (0, 0.1)$, Alg. 4 outputs a basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of $\mathcal{L}$ such that*

$$\|\mathbf{b}_1\| \leq (1+\varepsilon) \cdot (\delta^2 \gamma_k)^{\frac{n-1}{2(k-1)}} \operatorname{vol}(\mathcal{L})^{1/n}$$

by making $N \cdot (2n - 2k + 1) + 1$ calls to the δ-SVP oracle for lattices with rank k.

We mention in passing that just like our (HKZ-)slide simulator (cf. Sect. 6.1), we can similarly develop a DBKZ simulator for predicting the RHF depending on the number of tours during the DBKZ execution, based on the dynamical system analysis in [MW16].