Slide Reduction, Revisited—Filling the Gaps in Lattice SVP Approximation

Jianwei Li
ISG, RHUL, UK

London-ish Lattice Coding & Crypto Meetings
20 Nov 2019

Note

This talk is based on the following paper, but with some unessential update:

- Divesh Aggarwal; Jianwei Li; Phong Q. Nguyen; Noah Stephens-Davidowitz
- Slide Reduction, Revisited—Filling the Gaps in SVP Approximation.
- https://arxiv.org/abs/1908.03724
- It absorbs some ideas from discussions with coauthors.

Note

This talk is based on the following paper, but with some unessential update:

- Divesh Aggarwal; Jianwei Li; Phong Q. Nguyen; Noah Stephens-Davidowitz
- Slide Reduction, Revisited—Filling the Gaps in SVP Approximation.
- https://arxiv.org/abs/1908.03724
- It absorbs some ideas from discussions with coauthors.

Note

This talk is based on the following paper, but with some unessential update:

- Divesh Aggarwal; Jianwei Li; Phong Q. Nguyen; Noah Stephens-Davidowitz
- Slide Reduction, Revisited—Filling the Gaps in SVP Approximation.
- https://arxiv.org/abs/1908.03724
- It absorbs some ideas from discussions with coauthors.

Note

This talk is based on the following paper, but with some unessential update:

- Divesh Aggarwal; Jianwei Li; Phong Q. Nguyen; Noah Stephens-Davidowitz
- Slide Reduction, Revisited—Filling the Gaps in SVP Approximation.
- https://arxiv.org/abs/1908.03724
- It absorbs some ideas from discussions with coauthors.

Note

This talk is based on the following paper, but with some unessential update:

- Divesh Aggarwal; Jianwei Li; Phong Q. Nguyen; Noah Stephens-Davidowitz
- Slide Reduction, Revisited—Filling the Gaps in SVP Approximation.
- https://arxiv.org/abs/1908.03724
- It absorbs some ideas from discussions with coauthors.

Outline

1. Background on lattice reduction
2. Our results
3. Our technical ideas and argument
4. Conclusion and open problems

The most important lattice problem is the shortest vector problem (SVP)

- Given a basis of a lattice L, SVP is to find a shortest nonzero vector $\mathbf{v}$ in L, i.e., $\|\mathbf{v}\| = \min_{\mathbf{x} \in L_{\neq \mathbf{0}}} \|\mathbf{x}\| \triangleq \lambda_1(L)$.
- SVP is NP-hard under randomized reductions.

Two natural relaxations

- f-approximate SVP (f-SVP): Given a basis of a lattice L, find a non-zero lattice vector $\mathbf{v} \in L$ s.t. $\|\mathbf{v}\| \le f \cdot \lambda_1(L)$.
- f-Hermite SVP (f-HSVP): Given a basis B of a lattice L, find a non-zero lattice vector $\mathbf{v} \in L$ s.t. $\|\mathbf{v}\| \le f \cdot \mathrm{vol}(L)^{1/n}$, where $\mathrm{vol}(L) := \sqrt{\det(B^T B)}$ is the covolume of the lattice.

The most important lattice problem is the shortest vector problem (SVP)

- Given a basis of a lattice L, SVP is to find a shortest nonzero vector $\mathbf{v}$ in L, i.e., $\|\mathbf{v}\| = \min_{\mathbf{x} \in L_{\neq \mathbf{0}}} \|\mathbf{x}\| \triangleq \lambda_1(L)$.
- SVP is NP-hard under randomized reductions.

Two natural relaxations

- f-approximate SVP (f-SVP): Given a basis of a lattice L, find a non-zero lattice vector $\mathbf{v} \in L$ s.t. $\|\mathbf{v}\| \leq f \cdot \lambda_1(L)$.
- f-Hermite SVP (f-HSVP): Given a basis B of a lattice L, find a non-zero lattice vector $\mathbf{v} \in L$ s.t. $\|\mathbf{v}\| \leq f \cdot \mathrm{vol}(L)^{1/n}$, where $\mathrm{vol}(L) := \sqrt{\det(B^T B)}$ is the covolume of the lattice.

The most important lattice problem is the shortest vector problem (SVP)

- Given a basis of a lattice L, SVP is to find a shortest nonzero vector $\mathbf{v}$ in L, i.e., $\|\mathbf{v}\| = \min_{\mathbf{x} \in L_{\neq \mathbf{0}}} \|\mathbf{x}\| \triangleq \lambda_1(L)$.
- SVP is NP-hard under randomized reductions.

Two natural relaxations

- f-approximate SVP (f-SVP): Given a basis of a lattice L, find a non-zero lattice vector $\mathbf{v} \in L$ s.t. $\|\mathbf{v}\| \leq f \cdot \lambda_1(L)$.
- f-Hermite SVP (f-HSVP): Given a basis B of a lattice L, find a non-zero lattice vector $\mathbf{v} \in L$ s.t. $\|\mathbf{v}\| \leq f \cdot \mathrm{vol}(L)^{1/n}$, where $\mathrm{vol}(L) := \sqrt{\det(B^T B)}$ is the covolume of the lattice.

The most important lattice problem is the shortest vector problem (SVP)

- Given a basis of a lattice L, SVP is to find a shortest nonzero vector $\mathbf{v}$ in L, i.e., $\|\mathbf{v}\| = \min_{\mathbf{x} \in L_{\neq \mathbf{0}}} \|\mathbf{x}\| \triangleq \lambda_1(L)$.
- SVP is NP-hard under randomized reductions.

Two natural relaxations

- f-approximate SVP (f-SVP): Given a basis of a lattice L, find a non-zero lattice vector $\mathbf{v} \in L$ s.t. $\|\mathbf{v}\| \leq f \cdot \lambda_1(L)$.
- f-Hermite SVP (f-HSVP): Given a basis B of a lattice L, find a non-zero lattice vector $\mathbf{v} \in L$ s.t. $\|\mathbf{v}\| \leq f \cdot \mathrm{vol}(L)^{1/n}$, where $\mathrm{vol}(L) := \sqrt{\det(B^T B)}$ is the covolume of the lattice.

The most important lattice problem is the shortest vector problem (SVP)

- Given a basis of a lattice L, SVP is to find a shortest nonzero vector $\mathbf{v}$ in L, i.e., $\|\mathbf{v}\| = \min_{\mathbf{x}\in L_{\neq \mathbf{0}}} \|\mathbf{x}\| \triangleq \lambda_1(L)$.
- SVP is NP-hard under randomized reductions.

Two natural relaxations

- f-approximate SVP (f-SVP): Given a basis of a lattice L, find a non-zero lattice vector $\mathbf{v} \in L$ s.t. $\|\mathbf{v}\| \leq f \cdot \lambda_1(L)$.
- f-Hermite SVP (f-HSVP): Given a basis B of a lattice L, find a non-zero lattice vector $\mathbf{v} \in L$ s.t. $\|\mathbf{v}\| \leq f \cdot \mathrm{vol}(L)^{1/n}$, where $\mathrm{vol}(L) := \sqrt{\det(B^T B)}$ is the covolume of the lattice.

The most important lattice problem is the shortest vector problem (SVP)

- Given a basis of a lattice L, SVP is to find a shortest nonzero vector $\mathbf{v}$ in L, i.e., $\|\mathbf{v}\| = \min_{\mathbf{x}\in L_{\neq \mathbf{0}}} \|\mathbf{x}\| \triangleq \lambda_1(L)$.
- SVP is NP-hard under randomized reductions.

Two natural relaxations

- f-approximate SVP (f-SVP): Given a basis of a lattice L, find a non-zero lattice vector $\mathbf{v} \in L$ s.t. $\|\mathbf{v}\| \leq f \cdot \lambda_1(L)$.
- f-Hermite SVP (f-HSVP): Given a basis B of a lattice L, find a non-zero lattice vector $\mathbf{v} \in L$ s.t. $\|\mathbf{v}\| \leq f \cdot \mathrm{vol}(L)^{1/n}$, where $\mathrm{vol}(L) := \sqrt{\det(B^T B)}$ is the covolume of the lattice.

Lattice reduction

Goal

Find interesting bases, such as bases consisting of reasonably short and almost orthogonal vectors.

Importance

It is the classical approach for solving f-(H)SVP:

- Finding good reduced bases has proved invaluable in many fields of computer science and mathematics.
- Notably in cryptology, its importance is growing as lattice-based cryptography becomes the most popular candidate for post-quantum cryptography.
- ....

Lattice reduction

Goal

Find interesting bases, such as bases consisting of reasonably short and almost orthogonal vectors.

Importance

It is the classical approach for solving f-(H)SVP:

- Finding good reduced bases has proved invaluable in many fields of computer science and mathematics.
- Notably in cryptology, its importance is growing as lattice-based cryptography becomes the most popular candidate for post-quantum cryptography.
- . . . .

Lattice reduction

Goal

Find interesting bases, such as bases consisting of reasonably short and almost orthogonal vectors.

Importance

It is the classical approach for solving f-(H)SVP:

- Finding good reduced bases has proved invaluable in many fields of computer science and mathematics.
- Notably in cryptology, its importance is growing as lattice-based cryptography becomes the most popular candidate for post-quantum cryptography.
- . . . .

Lattice reduction

Goal

Find interesting bases, such as bases consisting of reasonably short and almost orthogonal vectors.

Importance

It is the classical approach for solving f-(H)SVP:

- Finding good reduced bases has proved invaluable in many fields of computer science and mathematics.
- Notably in cryptology, its importance is growing as lattice-based cryptography becomes the most popular candidate for post-quantum cryptography.
- . . . .

Lattice reduction

Goal

Find interesting bases, such as bases consisting of reasonably short and almost orthogonal vectors.

Importance

It is the classical approach for solving f-(H)SVP:

- Finding good reduced bases has proved invaluable in many fields of computer science and mathematics.
- Notably in cryptology, its importance is growing as lattice-based cryptography becomes the most popular candidate for post-quantum cryptography.
- . . . .

Chronology from LLL to slide-reduction/DBKZ

LLL is the first polynomial time lattice reduction algorithm for approximating SVP/HSVP within exponential factors: [1]

- Intuition : A basis $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ is *LLL-reduced* if every 2-rank projected block $\mathbf{B}_{[i,i+1]}$ is almost SVP-reduced for $1 \leq i \leq n-1$.
- Main properties: If a basis $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of a lattice L is LLL-reduced, then

$$\begin{aligned} \|\mathbf{b}_1\| &\leq 2^{(n-1)/4} \cdot \mathrm{vol}(L)^{1/n}, \\ \|\mathbf{b}_1\| &\leq 2^{(n-1)/2} \cdot \lambda_1(L). \end{aligned}$$

[1] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. Math. Ann., 1982

Chronology from LLL to slide-reduction/DBKZ

LLL is the first polynomial time lattice reduction algorithm for approximating SVP/HSVP within exponential factors: [1]

- Intuition : A basis $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ is *LLL-reduced* if every 2-rank projected block $\mathbf{B}_{[i,i+1]}$ is almost SVP-reduced for $1 \leq i \leq n-1$.
- Main properties: If a basis $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of a lattice L is LLL-reduced, then

$$\begin{aligned} \|\mathbf{b}_1\| &\leq 2^{(n-1)/4} \cdot \mathrm{vol}(L)^{1/n}, \\ \|\mathbf{b}_1\| &\leq 2^{(n-1)/2} \cdot \lambda_1(L). \end{aligned}$$

[1] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. Math. Ann., 1982

Chronology from LLL to slide-reduction/DBKZ

Schnorr's blockwise generalizations of LLL:[2]

1. Semi block $2k$-reduction is the first lattice reduction algorithm for approximating SVP/HSVP within (subexponential) factors $k^{O(n/k)}$ using polynomial calls to exact SVP-oracle in rank k.
2. BKZ is the most popular blockwise lattice reduction.
   - Intuition : A basis $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of rank n is *k-BKZ-reduced* if every projected block $B_{[i,\min\{i+k-1,n\}]}$ of rank $\leq k$ is SVP-reduced for $i = 1, \cdots, n$.
   - Main properties: If a basis $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of a lattice L is k-BKZ-reduced, then

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

$$\|\mathbf{b}_1\| \leq \gamma_k^{\frac{n-1}{k-1}} \cdot \lambda_1(L).$$

   Here, γ_k is Hermite's constant.

[2] C. P. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. TCS, 1987

Chronology from LLL to slide-reduction/DBKZ

Schnorr's blockwise generalizations of LLL:[2]

1. Semi block $2k$-reduction is the first lattice reduction algorithm for approximating SVP/HSVP within (subexponential) factors $k^{O(n/k)}$ using polynomial calls to exact SVP-oracle in rank k.
2. BKZ is the most popular blockwise lattice reduction.
   - Intuition : A basis $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of rank n is *k-BKZ-reduced* if every projected block $B_{[i,\min\{i+k-1,n\}]}$ of rank $\leq k$ is SVP-reduced for $i = 1, \cdots, n$.
   - Main properties: If a basis $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of a lattice L is k-BKZ-reduced, then

$$\|\mathbf{b}_1\| \leq \gamma_k^{\frac{n-1}{2(k-1)}+\frac{1}{2}} \cdot \mathrm{vol}(L)^{1/n},$$
$$\|\mathbf{b}_1\| \leq \gamma_k^{\frac{n-1}{k-1}} \cdot \lambda_1(L).$$

Here, γ_k is Hermite's constant.

[2] C. P. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. TCS, 1987

Chronology from LLL to slide-reduction/DBKZ

Schnorr's blockwise generalizations of LLL:[2]

1. Semi block $2k$-reduction is the first lattice reduction algorithm for approximating SVP/HSVP within (subexponential) factors $k^{O(n/k)}$ using polynomial calls to exact SVP-oracle in rank k.
2. BKZ is the most popular blockwise lattice reduction.
   - Intuition : A basis $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of rank n is *k-BKZ-reduced* if every projected block $B_{[i,\min\{i+k-1,n\}]}$ of rank $\leq k$ is SVP-reduced for $i = 1, \cdots, n$.
   - Main properties: If a basis $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of a lattice L is k-BKZ-reduced, then

$$\begin{aligned} \|\mathbf{b}_1\| &\leq \gamma_k^{\frac{n-1}{2(k-1)}+\frac{1}{2}} \cdot \mathrm{vol}(L)^{1/n}, \\ \|\mathbf{b}_1\| &\leq \gamma_k^{\frac{n-1}{k-1}} \cdot \lambda_1(L). \end{aligned}$$

   Here, γ_k is Hermite's constant.

[2] C. P. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. TCS, 1987

Chronology from LLL to slide-reduction/DBKZ

Schnorr's blockwise generalizations of LLL:[2]

1. Semi block $2k$-reduction is the first lattice reduction algorithm for approximating SVP/HSVP within (subexponential) factors $k^{O(n/k)}$ using polynomial calls to exact SVP-oracle in rank k.
2. BKZ is the most popular blockwise lattice reduction.
   - Intuition : A basis $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of rank n is *k-BKZ-reduced* if every projected block $B_{[i,\min\{i+k-1,n\}]}$ of rank $\leq k$ is SVP-reduced for $i = 1, \cdots, n$.
   - Main properties: If a basis $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of a lattice L is k-BKZ-reduced, then

$$\begin{aligned} \|\mathbf{b}_1\| &\leq \gamma_k^{\frac{n-1}{2(k-1)}+\frac{1}{2}} \cdot \mathrm{vol}(L)^{1/n}, \\ \|\mathbf{b}_1\| &\leq \gamma_k^{\frac{n-1}{k-1}} \cdot \lambda_1(L). \end{aligned}$$

     Here, γ_k is Hermite's constant.

[2] C. P. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. TCS, 1987

Chronology from LLL to slide-reduction/DBKZ

Schnorr's blockwise generalizations of LLL: BKZ again!

- BKZ achieves the best time/quality trade-off in practice and is the most popular blockwise lattice reduction algorithm: E.g., the NTL/fpLLL/G6K libraries and the SVP challenge.
- No polynomial-time bound is known for BKZ: it is typically employed with early termination in practice.
- Long-standing open problem: Within polynomial calls to SVP-oracle, can the BKZ algorithm output an almost BKZ-reduced basis?

Chronology from LLL to slide-reduction/DBKZ

Schnorr's blockwise generalizations of LLL: BKZ again!

- BKZ achieves the best time/quality trade-off in practice and is the most popular blockwise lattice reduction algorithm: E.g., the NTL/fpLLL/G6K libraries and the SVP challenge.
- No polynomial-time bound is known for BKZ: it is typically employed with early termination in practice.
- Long-standing open problem: Within polynomial calls to SVP-oracle, can the BKZ algorithm output an almost BKZ-reduced basis?

Chronology from LLL to slide-reduction/DBKZ

Schnorr's blockwise generalizations of LLL: BKZ again!

- BKZ achieves the best time/quality trade-off in practice and is the most popular blockwise lattice reduction algorithm: E.g., the NTL/fpLLL/G6K libraries and the SVP challenge.
- No polynomial-time bound is known for BKZ: it is typically employed with early termination in practice.
- Long-standing open problem: Within polynomial calls to SVP-oracle, can the BKZ algorithm output an almost BKZ-reduced basis?

Chronology from LLL to slide-reduction/DBKZ

- No publication claims to solve this open problem on BKZ.

★ In theory, both GN-slide-reduction [3] and MW-DBKZ [4] can achieve almost the same guarantees on

$$\|\mathbf{b}_1\|/\mathrm{vol}(L)^{1/n} \text{ and } \|\mathbf{b}_1\|/\lambda_1(L)$$

as that of BKZ-reduced bases, with polynomial calls to SVP-oracle.

[3] N. Gama and P. Q. Nguyen. Finding short lattice vectors within Mordell's inequality. STOC 2008.

[4] D. Micciancio and M. Walter. Practical, predictable lattice basis reduction. EUROCRYPT 2016.

Chronology from LLL to slide-reduction/DBKZ

Slide-reduction is the previously best polynomial time lattice reduction algorithm for solving $n^{c\geq 1}$-SVP in theory: [5]

- Definition: A basis $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of rank n is (ε, k)-*slide-reduced* where $n = pk \geq 2k$ if
  - Primal conditions: each block $B_{[ik+1,ik+k]}$ is HKZ-reduced.
  - Dual conditions: each block $B_{[ik+2,ik+k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.
- Main properties: Let $n = pk \geq 2k$ be integers. With $\mathrm{poly}(\mathrm{size}(B_{\mathrm{input}}), 1/\varepsilon)$ calls to exact SVP-oracle, the slide-reduction algorithm outputs a (ε, k)-slide-reduced basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of the input lattice L:

$$\begin{aligned} \|\mathbf{b}_1\| &\leq ((1+\varepsilon)\gamma_k)^{\frac{n-1}{2(k-1)}} \cdot \mathrm{vol}(L)^{1/n}, \\ \|\mathbf{b}_1\| &\leq ((1+\varepsilon)\gamma_k)^{\frac{n-k}{k-1}} \cdot \lambda_1(L). \end{aligned}$$

[5] N. Gama and P. Q. Nguyen. Finding short lattice vectors within Mordell's inequality. STOC 2008.

Chronology from LLL to slide-reduction/DBKZ

Slide-reduction is the previously best polynomial time lattice reduction algorithm for solving $n^{c\geq 1}$-SVP in theory: [5]

- Definition: A basis $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of rank n is (ε, k)-*slide-reduced* where $n = pk \geq 2k$ if
  - Primal conditions: each block $B_{[ik+1,ik+k]}$ is HKZ-reduced.
  - Dual conditions: each block $B_{[ik+2,ik+k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.
- Main properties: Let $n = pk \geq 2k$ be integers. With $\mathrm{poly}(\mathrm{size}(B_{\mathrm{input}}), 1/\varepsilon)$ calls to exact SVP-oracle, the slide-reduction algorithm outputs a (ε, k)-slide-reduced basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of the input lattice L:

$$\begin{aligned} \|\mathbf{b}_1\| &\leq ((1+\varepsilon)\gamma_k)^{\frac{n-1}{2(k-1)}} \cdot \mathrm{vol}(L)^{1/n}, \\ \|\mathbf{b}_1\| &\leq ((1+\varepsilon)\gamma_k)^{\frac{n-k}{k-1}} \cdot \lambda_1(L). \end{aligned}$$

[5] N. Gama and P. Q. Nguyen. Finding short lattice vectors within Mordell's inequality. STOC 2008.

Chronology from LLL to slide-reduction/DBKZ

Slide-reduction is the previously best polynomial time lattice reduction algorithm for solving $n^{c \geq 1}$-SVP in theory: [5]

- Definition: A basis $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of rank n is (ε, k)-*slide-reduced* where $n = pk \geq 2k$ if
  - Primal conditions: each block $B_{[ik+1, ik+k]}$ is HKZ-reduced.
  - Dual conditions: each block $B_{[ik+2, ik+k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.
- Main properties: Let $n = pk \geq 2k$ be integers. With $\mathrm{poly}(\mathrm{size}(B_{\mathrm{input}}), 1/\varepsilon)$ calls to exact SVP-oracle, the slide-reduction algorithm outputs a (ε, k)-slide-reduced basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of the input lattice L:

$$\begin{aligned} \|\mathbf{b}_1\| &\leq ((1+\varepsilon)\gamma_k)^{\frac{n-1}{2(k-1)}} \cdot \mathrm{vol}(L)^{1/n}, \\ \|\mathbf{b}_1\| &\leq ((1+\varepsilon)\gamma_k)^{\frac{n-k}{k-1}} \cdot \lambda_1(L). \end{aligned}$$

[5] N. Gama and P. Q. Nguyen. Finding short lattice vectors within Mordell's inequality. STOC 2008.

Chronology from LLL to slide-reduction/DBKZ

Slide-reduction is the previously best polynomial time lattice reduction algorithm for solving $n^{c\geq 1}$-SVP in theory: [5]

- Definition: A basis $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of rank n is (ε, k)-*slide-reduced* where $n = pk \geq 2k$ if
  - Primal conditions: each block $B_{[ik+1,ik+k]}$ is HKZ-reduced.
  - Dual conditions: each block $B_{[ik+2,ik+k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.
- Main properties: Let $n = pk \geq 2k$ be integers. With $\mathrm{poly}(\mathrm{size}(B_{\mathrm{input}}), 1/\varepsilon)$ calls to exact SVP-oracle, the slide-reduction algorithm outputs a (ε, k)-slide-reduced basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of the input lattice L:

$$\begin{aligned} \|\mathbf{b}_1\| &\leq ((1+\varepsilon)\gamma_k)^{\frac{n-1}{2(k-1)}} \cdot \mathrm{vol}(L)^{1/n}, \\ \|\mathbf{b}_1\| &\leq ((1+\varepsilon)\gamma_k)^{\frac{n-k}{k-1}} \cdot \lambda_1(L). \end{aligned}$$

[5] N. Gama and P. Q. Nguyen. Finding short lattice vectors within Mordell's inequality. STOC 2008.

Chronology from LLL to slide-reduction/DBKZ

DBKZ is the previously best polynomial time lattice reduction algorithm for solving $n^{c \geq \frac{1}{2}}$-HSVP in theory: [6]

- Let $n \geq k \geq 2$ be integers. With $\mathrm{poly}(\mathrm{size}(B_{\mathrm{input}}), 1/\varepsilon)$ calls to exact SVP-oracle, the DBKZ algorithm outputs a basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of the input lattice L s.t.

$$\begin{aligned} \|\mathbf{b}_1\| &\leq (1+\varepsilon)\gamma_k^{\frac{n-1}{2(k-1)}} \cdot \mathrm{vol}(L)^{1/n}, \\ \|\mathbf{b}_1\| &\leq (1+\varepsilon)^2\gamma_k^{\frac{n-1}{k-1}} \cdot \lambda_1(L). \end{aligned}$$

- It matches Mordell's inequality:

$$\gamma_n \leq \gamma_k^{(n-1)/(k-1)} \quad \text{for any} \quad 2 \leq k \leq n.$$

[6] D. Micciancio and M. Walter. Practical, predictable lattice basis reduction. EUROCRYPT 2016.

Chronology from LLL to slide-reduction/DBKZ

DBKZ is the previously best polynomial time lattice reduction algorithm for solving $n^{c\geq\frac{1}{2}}$-HSVP in theory: [6]

- Let $n \geq k \geq 2$ be integers. With $\mathrm{poly}(\mathrm{size}(B_{\mathrm{input}}), 1/\varepsilon)$ calls to exact SVP-oracle, the DBKZ algorithm outputs a basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of the input lattice L s.t.

$$\begin{aligned} \|\mathbf{b}_1\| &\leq (1+\varepsilon)\gamma_k^{\frac{n-1}{2(k-1)}} \cdot \mathrm{vol}(L)^{1/n}, \\ \|\mathbf{b}_1\| &\leq (1+\varepsilon)^2\gamma_k^{\frac{n-1}{k-1}} \cdot \lambda_1(L). \end{aligned}$$

- It matches Mordell's inequality:

$$\gamma_n \leq \gamma_k^{(n-1)/(k-1)} \quad \text{for any} \quad 2 \leq k \leq n.$$

[6] D. Micciancio and M. Walter. Practical, predictable lattice basis reduction. EUROCRYPT 2016.

Three questions on lattice reduction

Case 1: Approximating SVP with sublinear factors

- The security of many lattice-based cryptographic constructions is based on the worst-case hardness of n^c-SVP with constant $c \in [\frac{1}{2}, 1]$.
- Awkward: All known lattice reduction algorithm can only solve n^c-SVP for $c \geq 1$.
- Prior results: (Almost) exact SVP algorithms can trivially solve n^c-SVP with any constant $c \in [\frac{1}{2}, 1]$.

A natural question

Is there an non-trivial (lattice reduction) algorithm for approximating SVP with sublinear factors?

Three questions on lattice reduction

Case 1: Approximating SVP with sublinear factors

- The security of many lattice-based cryptographic constructions is based on the worst-case hardness of n^c-SVP with constant $c \in [\frac{1}{2}, 1]$.
- Awkward: All known lattice reduction algorithm can only solve n^c-SVP for $c \geq 1$.
- Prior results: (Almost) exact SVP algorithms can trivially solve n^c-SVP with any constant $c \in [\frac{1}{2}, 1]$.

A natural question

Is there an non-trivial (lattice reduction) algorithm for approximating SVP with sublinear factors?

Three questions on lattice reduction

Case 1: Approximating SVP with sublinear factors

- The security of many lattice-based cryptographic constructions is based on the worst-case hardness of n^c-SVP with constant $c \in [\frac{1}{2}, 1]$.
- Awkward: All known lattice reduction algorithm can only solve n^c-SVP for $c \geq 1$.
- Prior results: (Almost) exact SVP algorithms can trivially solve n^c-SVP with any constant $c \in [\frac{1}{2}, 1]$.

A natural question

Is there an non-trivial (lattice reduction) algorithm for approximating SVP with sublinear factors?

Three questions on lattice reduction

Case 1: Approximating SVP with sublinear factors

- The security of many lattice-based cryptographic constructions is based on the worst-case hardness of n^c-SVP with constant $c \in [\frac{1}{2}, 1]$.
- Awkward: All known lattice reduction algorithm can only solve n^c-SVP for $c \geq 1$.
- Prior results: (Almost) exact SVP algorithms can trivially solve n^c-SVP with any constant $c \in [\frac{1}{2}, 1]$.

A natural question

Is there an non-trivial (lattice reduction) algorithm for approximating SVP with sublinear factors?

Three questions on lattice reduction

Case 1: Approximating SVP with sublinear factors

- The security of many lattice-based cryptographic constructions is based on the worst-case hardness of n^c-SVP with constant $c \in [\frac{1}{2}, 1]$.
- Awkward: All known lattice reduction algorithm can only solve n^c-SVP for $c \geq 1$.
- Prior results: (Almost) exact SVP algorithms can trivially solve n^c-SVP with any constant $c \in [\frac{1}{2}, 1]$.

A natural question

Is there an non-trivial (lattice reduction) algorithm for approximating SVP with sublinear factors?

Three questions on lattice reduction

Case 2: Approximating SVP with polynomial factors

- The security of some lattice-based cryptographic constructions is based on the worst-case hardness of n^c-SVP with constant $c \geq 1$ including fractional constant, e.g., $n^{1.5}$-SVP for the cryptosystem in[a].
- Awkward: The previously best GN-slide reduction algorithm can non-trivially solve $n^{\lceil c \rceil}$-SVP or $n^{\lfloor c \rfloor}$-SVP rather than n^c-SVP for $c \geq 1$.

[a] O. Regev. New lattice-based cryptographic constructions. JACM 2004.

A natural question

Can we extend GN-sldie-reduction algorithm into the case that k might not divide n, so that it can directly solve n^c-SVP over any constant $c \in [1, O(1)]$?

Three questions on lattice reduction

Case 2: Approximating SVP with polynomial factors

- The security of some lattice-based cryptographic constructions is based on the worst-case hardness of n^c-SVP with constant $c \geq 1$ including fractional constant, e.g., $n^{1.5}$-SVP for the cryptosystem in[a].
- Awkward: The previously best GN-slide reduction algorithm can non-trivially solve $n^{\lceil c \rceil}$-SVP or $n^{\lfloor c \rfloor}$-SVP rather than n^c-SVP for $c \geq 1$.

[a] O. Regev. New lattice-based cryptographic constructions. JACM 2004.

A natural question

Can we extend GN-sldie-reduction algorithm into the case that k might not divide n, so that it can directly solve n^c-SVP over any constant $c \in [1, O(1)]$?

Three questions on lattice reduction

Case 2: Approximating SVP with polynomial factors

- The security of some lattice-based cryptographic constructions is based on the worst-case hardness of n^c-SVP with constant $c \geq 1$ including fractional constant, e.g., $n^{1.5}$-SVP for the cryptosystem in[a].
- Awkward: The previously best GN-slide reduction algorithm can non-trivially solve $n^{\lceil c \rceil}$-SVP or $n^{\lfloor c \rfloor}$-SVP rather than n^c-SVP for $c \geq 1$.

[a] O. Regev. New lattice-based cryptographic constructions. JACM 2004.

A natural question

Can we extend GN-sldie-reduction algorithm into the case that k might not divide n, so that it can directly solve n^c-SVP over any constant $c \in [1, O(1)]$?

Three questions on lattice reduction

Case 2: Approximating SVP with polynomial factors

- The security of some lattice-based cryptographic constructions is based on the worst-case hardness of n^c-SVP with constant $c \geq 1$ including fractional constant, e.g., $n^{1.5}$-SVP for the cryptosystem in[a].
- Awkward: The previously best GN-slide reduction algorithm can non-trivially solve $n^{\lceil c \rceil}$-SVP or $n^{\lfloor c \rfloor}$-SVP rather than n^c-SVP for $c \geq 1$.

[a] O. Regev. New lattice-based cryptographic constructions. JACM 2004.

A natural question

Can we extend GN-sldie-reduction algorithm into the case that k might not divide n, so that it can directly solve n^c-SVP over any constant $c \in [1, O(1)]$?

Three questions on lattice reduction

Disharmony

- Slide-reduction is the previously best polynomial time lattice reduction algorithm for solving $n^{c\geq 1}$-SVP in theory;
- DBKZ is the previously best polynomial time lattice reduction algorithm for solving $n^{c\geq \frac{1}{2}}$-HSVP in theory.

A natural question

Is there a single algorithm which is the best in theory for solving both $n^{c\geq 1}$-SVP and $n^{c\geq \frac{1}{2}}$-HSVP?

Three questions on lattice reduction

Disharmony

- Slide-reduction is the previously best polynomial time lattice reduction algorithm for solving $n^{c\geq 1}$-SVP in theory;
- DBKZ is the previously best polynomial time lattice reduction algorithm for solving $n^{c\geq \frac{1}{2}}$-HSVP in theory.

A natural question

Is there a single algorithm which is the best in theory for solving both $n^{c\geq 1}$-SVP and $n^{c\geq \frac{1}{2}}$-HSVP?

Three questions on lattice reduction

Disharmony

- Slide-reduction is the previously best polynomial time lattice reduction algorithm for solving $n^{c\geq 1}$-SVP in theory;
- DBKZ is the previously best polynomial time lattice reduction algorithm for solving $n^{c\geq \frac{1}{2}}$-HSVP in theory.

A natural question

Is there a single algorithm which is the best in theory for solving both $n^{c\geq 1}$-SVP and $n^{c\geq \frac{1}{2}}$-HSVP?

Three questions on lattice reduction

Disharmony

- Slide-reduction is the previously best polynomial time lattice reduction algorithm for solving $n^{c \geq 1}$-SVP in theory;
- DBKZ is the previously best polynomial time lattice reduction algorithm for solving $n^{c \geq \frac{1}{2}}$-HSVP in theory.

A natural question

Is there a single algorithm which is the best in theory for solving both $n^{c \geq 1}$-SVP and $n^{c \geq \frac{1}{2}}$-HSVP?

1. Background on lattice reduction
2. **Our results**
3. Our technical ideas and argument
4. Conclusion and open problems

Our paper solves the three questions:

Q1 Is there an non-trivial (lattice reduction) algorithm for approximating SVP with sublinear factors?

Q2 Can we extend GN-slide-reduction algorithm into the case that k does not divide n exactly, so that it can directly solve n^c-SVP over any constant $c \in [1, O(1)]$?

Q3 Is there a single algorithm which is the best in theory for solving both $n^{c\geq 1}$-SVP and $n^{c\geq \frac{1}{2}}$-HSVP?

Our paper solves the three questions:

Q1 Is there an non-trivial (lattice reduction) algorithm for approximating SVP with sublinear factors?

Q2 Can we extend GN-slide-reduction algorithm into the case that k does not divide n exactly, so that it can directly solve n^c-SVP over any constant $c \in [1, O(1)]$?

Q3 Is there a single algorithm which is the best in theory for solving both $n^{c \geq 1}$-SVP and $n^{c \geq \frac{1}{2}}$-HSVP?

Our paper solves the three questions:

Q1 Is there an non-trivial (lattice reduction) algorithm for approximating SVP with sublinear factors?

Q2 Can we extend GN-slide-reduction algorithm into the case that k does not divide n exactly, so that it can directly solve n^c-SVP over any constant $c \in [1, O(1)]$?

Q3 Is there a single algorithm which is the best in theory for solving both $n^{c\geq 1}$-SVP and $n^{c\geq \frac{1}{2}}$-HSVP?

Our paper solves the three questions:

Q1 Is there an non-trivial (lattice reduction) algorithm for approximating SVP with sublinear factors?

Q2 Can we extend GN-slide-reduction algorithm into the case that k does not divide n exactly, so that it can directly solve n^c-SVP over any constant $c \in [1, O(1)]$?

Q3 Is there a single algorithm which is the best in theory for solving both $n^{c \geq 1}$-SVP and $n^{c \geq \frac{1}{2}}$-HSVP?

Our first result

Theorem (Approximating SVP with sublinear factor)

Let $2k > n \geq k \geq 2$ be integers and $\delta \geq 1$. There is an algorithm that with polynomial calls to δ-SVP-oracle in rank k, it outputs an nonzero vector $\mathbf{b}$ of the input lattice L s.t.

$$\|\mathbf{b}\| \leq O(\delta(\delta^2\gamma_k)^{\frac{n}{2k}}) \cdot \lambda_1(L).$$

★ This is the first non-trivial algorithm for approximating SVP with sublinear factors $n^{\frac{1}{2}} \leq f \leq n^{1-\varepsilon}$.

Corollary

For any constant $c \in (1/2, 1)$ and any factor $\delta \geq 1$, there is an efficient Cook-reduction from $O(\delta^{2c+1} n^c)$-SVP in rank n to δ-SVP in rank $k := \lceil \frac{n}{2c} \rceil$.

Our first result

Theorem (Approximating SVP with sublinear factor)

Let $2k > n \geq k \geq 2$ be integers and $\delta \geq 1$. There is an algorithm that with polynomial calls to δ-SVP-oracle in rank k, it outputs an nonzero vector $\mathbf{b}$ of the input lattice L s.t.

$$\|\mathbf{b}\| \leq O(\delta(\delta^2\gamma_k)^{\frac{n}{2k}}) \cdot \lambda_1(L).$$

★ This is the first non-trivial algorithm for approximating SVP with sublinear factors $n^{\frac{1}{2}} \leq f \leq n^{1-\varepsilon}$.

Corollary

For any constant $c \in (1/2, 1)$ and any factor $\delta \geq 1$, there is an efficient Cook-reduction from $O(\delta^{2c+1} n^c)$-SVP in rank n to δ-SVP in rank $k := \lceil \frac{n}{2c} \rceil$.

Our first result

Theorem (Approximating SVP with sublinear factor)

Let $2k > n \geq k \geq 2$ be integers and $\delta \geq 1$. There is an algorithm that with polynomial calls to δ-SVP-oracle in rank k, it outputs an nonzero vector $\mathbf{b}$ of the input lattice L s.t.

$$\|\mathbf{b}\| \leq O(\delta(\delta^2\gamma_k)^{\frac{n}{2k}}) \cdot \lambda_1(L).$$

★ This is the first non-trivial algorithm for approximating SVP with sublinear factors $n^{\frac{1}{2}} \leq f \leq n^{1-\varepsilon}$.

Corollary

For any constant $c \in (1/2, 1)$ and any factor $\delta \geq 1$, there is an efficient Cook-reduction from $O(\delta^{2c+1} n^c)$-SVP in rank n to δ-SVP in rank $k := \lceil \frac{n}{2c} \rceil$.

Our first result

Theorem (Approximating SVP with sublinear factor)

Let $2k > n \geq k \geq 2$ be integers and $\delta \geq 1$. There is an algorithm that with polynomial calls to δ-SVP-oracle in rank k, it outputs an nonzero vector $\mathbf{b}$ of the input lattice L s.t.

$$\|\mathbf{b}\| \leq O(\delta(\delta^2\gamma_k)^{\frac{n}{2k}}) \cdot \lambda_1(L).$$

★ This is the first non-trivial algorithm for approximating SVP with sublinear factors $n^{\frac{1}{2}} \leq f \leq n^{1-\varepsilon}$.

Corollary

For any constant $c \in (1/2, 1)$ and any factor $\delta \geq 1$, there is an efficient Cook-reduction from $O(\delta^{2c+1} n^c)$-SVP in rank n to δ-SVP in rank $k := \lceil \frac{n}{2c} \rceil$.

Our second result

Theorem (Approximating SVP with (at least) polynomial factor)

Let $n \geq 2k \geq 4$ be integers and $\delta \geq 1$. There is an algorithm that with $poly(\mathrm{size}(B_{input}), 1/\varepsilon)$ calls to δ-SVP-oracle in rank k, it outputs a basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of the input lattice L s.t.

$$\begin{aligned} \|\mathbf{b}_1\| &\leq ((1+\varepsilon)\delta^2\gamma_k)^{\frac{n-1}{2(k-1)}} \cdot \mathrm{vol}(L)^{1/n}, \\ \|\mathbf{b}_1\| &\leq ((1+\varepsilon)\delta^2\gamma_k)^{\frac{n-k}{k-1}} \cdot \lambda_1(L). \end{aligned}$$

Corollary

For any constant $c \geq 1$ and any factor $\delta \geq 1$, there is an efficient Cook-reduction from $O(\delta^{2c+1} n^c)$-SVP in rank n to δ-SVP in rank $k := \lfloor \frac{n}{c+1} \rfloor$.

Our second result

Theorem (Approximating SVP with (at least) polynomial factor)

Let $n \geq 2k \geq 4$ be integers and $\delta \geq 1$. There is an algorithm that with $poly(\mathrm{size}(B_{input}), 1/\varepsilon)$ calls to δ-SVP-oracle in rank k, it outputs a basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of the input lattice L s.t.

$$
\begin{aligned}
\|\mathbf{b}_1\| &\leq ((1+\varepsilon)\delta^2\gamma_k)^{\frac{n-1}{2(k-1)}} \cdot \mathrm{vol}(L)^{1/n}, \\
\|\mathbf{b}_1\| &\leq ((1+\varepsilon)\delta^2\gamma_k)^{\frac{n-k}{k-1}} \cdot \lambda_1(L).
\end{aligned}
$$

Corollary

For any constant $c \geq 1$ and any factor $\delta \geq 1$, there is an efficient Cook-reduction from $O(\delta^{2c+1} n^c)$-SVP in rank n to δ-SVP in rank $k := \lfloor \frac{n}{c+1} \rfloor$.

Our second result

Theorem (Approximating SVP with (at least) polynomial factor)

Let $n \geq 2k \geq 4$ be integers and $\delta \geq 1$. There is an algorithm that with $poly(\mathrm{size}(B_{input}), 1/\varepsilon)$ calls to δ-SVP-oracle in rank k, it outputs a basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of the input lattice L s.t.

$$\begin{aligned} \|\mathbf{b}_1\| &\leq ((1+\varepsilon)\delta^2\gamma_k)^{\frac{n-1}{2(k-1)}} \cdot \mathrm{vol}(L)^{1/n}, \\ \|\mathbf{b}_1\| &\leq ((1+\varepsilon)\delta^2\gamma_k)^{\frac{n-k}{k-1}} \cdot \lambda_1(L). \end{aligned}$$

Corollary

For any constant $c \geq 1$ and any factor $\delta \geq 1$, there is an efficient Cook-reduction from $O(\delta^{2c+1} n^c)$-SVP in rank n to δ-SVP in rank $k := \lfloor \frac{n}{c+1} \rfloor$.

Impact

- Our two algorithms provide currently the best polynomial-time lattice reduction algorithm:
  $\Rightarrow$ Achieve the best time/quality trade-off in theory.
  $\Rightarrow$ Formalize the common practice of approximating SVP in high rank with approx-SVP-oracle in low ranks.
- With well-chosen SVP-oracles in lower rank, our work implies the exponentially faster provable/heuristic algorithm for approximating SVP with factor $n^{1/2} \leq f \leq n^{O(1)}$:
  $\Rightarrow$ This is the regime most relevant for cryptography.

Impact

- Our two algorithms provide currently the best polynomial-time lattice reduction algorithm:
  $\Rightarrow$ Achieve the best time/quality trade-off in theory.
  $\Rightarrow$ Formalize the common practice of approximating SVP in high rank with approx-SVP-oracle in low ranks.
- With well-chosen SVP-oracles in lower rank, our work implies the exponentially faster provable/heuristic algorithm for approximating SVP with factor $n^{1/2} \leq f \leq n^{O(1)}$:
  $\Rightarrow$ This is the regime most relevant for cryptography.

Impact 1: the fastest provable algorithm

- WLW algorithm solves δ-SVP in rank k with $2^{0.802k}$-time for some constant factor δ.[7]

★ By using WLW algorithm as SVP-oracle in lower rank, our work implies the exponentially faster provable algorithm for approximating SVP with factor $n^{1/2} \leq f \leq n^{O(1)}$.

Table: Provable algorithms for approximating SVP.

Approx-factor	Previous best		This work
Exact	2^n	[ADRS15]	—
$\Omega(1) \leq f \leq \sqrt{n}$	$2^{0.802n}$	[WLW15]	—
n^c for $c \in [\frac{1}{2}, 1)$	$2^{0.802n}$	[WLW15]	$2^{\frac{0.802n}{2c}}$
n^c for $c \geq 1$	$2^{\frac{n}{\lfloor c+1 \rfloor}}$	[GN08]+[ADRS15]	$2^{\frac{0.802n}{c+1}}$

[7] W. Wei, M. Liu, and X. Wang. Finding shortest latticevectors in the presence of gaps. CT-RSA 2015.

Impact 1: the fastest provable algorithm

- WLW algorithm solves δ-SVP in rank k with $2^{0.802k}$-time for some constant factor δ.[7]

★ By using WLW algorithm as SVP-oracle in lower rank, our work implies the exponentially faster provable algorithm for approximating SVP with factor $n^{1/2} \leq f \leq n^{O(1)}$.

Table: Provable algorithms for approximating SVP.

Approx-factor	Previous best		This work
Exact	2^n	[ADRS15]	—
$\Omega(1) \leq f \leq \sqrt{n}$	$2^{0.802n}$	[WLW15]	—
n^c for $c \in [\frac{1}{2}, 1)$	$2^{0.802n}$	[WLW15]	$2^{\frac{0.802n}{2c}}$
n^c for $c \geq 1$	$2^{\frac{n}{\lfloor c+1 \rfloor}}$	[GN08]+[ADRS15]	$2^{\frac{0.802n}{c+1}}$

[7] W. Wei, M. Liu, and X. Wang. Finding shortest latticevectors in the presence of gaps. CT-RSA 2015.

Impact 1: the fastest provable algorithm

★ By using WLW algorithm as SVP-oracle in lower rank, our work imply the exponentially faster provable algorithm for approximating SVP with factor $n^{1/2} \leq f \leq n^{O(1)}$.

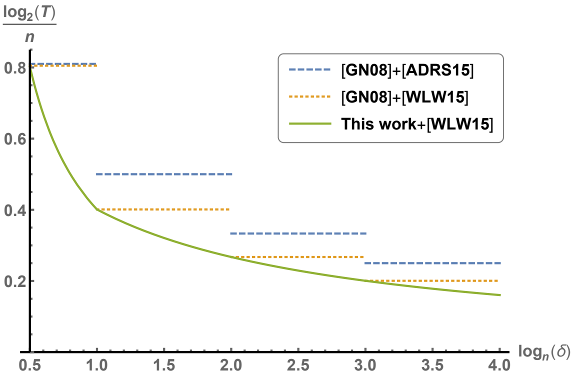

Figure: Runtime T as a function of approximation factor f for f-SVP. The y-axis is $\log_2(T)/n$, and the x-axis is $\log_n f$.

Impact 2: the fastest heuristic algorithm

- BDGL heuristic sieving algorithm solves SVP exactly in rank k with $2^{0.292k}$-time.[8]

★ By using BDGL algorithm as SVP-oracle in lower rank, our work imply the exponentially faster heuristic algorithm for approximating SVP with factor $n^{1/2} \leq f \leq n^{O(1)}$:

$\Rightarrow$ Security estimates of lattice-based cryptosystems.

Table: Heuristic algorithms for approximating SVP.

Approx-factor	Previous best		This work
$1 \leq f \leq \sqrt{n}$	$2^{0.292n}$	[BDGL16]	—
n^c for $c \in [\frac{1}{2}, 1)$	$2^{0.292n}$	[BDGL16]	$2^{\frac{0.292n}{2c}}$
n^c for $c \geq 1$	$2^{\frac{0.292n}{\lfloor c+1 \rfloor}}$	[GN08]+[BDGL16]	$2^{\frac{0.292n}{c+1}}$

[8] A. Becker, L. Ducas, N. Gama, and T. Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. SODA 2016.

Impact 2: the fastest heuristic algorithm

- BDGL heuristic sieving algorithm solves SVP exactly in rank k with $2^{0.292k}$-time.[8]

★ By using BDGL algorithm as SVP-oracle in lower rank, our work imply the exponentially faster heuristic algorithm for approximating SVP with factor $n^{1/2} \leq f \leq n^{O(1)}$:

$\Rightarrow$ Security estimates of lattice-based cryptosystems.

Table: Heuristic algorithms for approximating SVP.

Approx-factor	Previous best		This work
$1 \leq f \leq \sqrt{n}$	$2^{0.292n}$	[BDGL16]	—
n^c for $c \in [\frac{1}{2}, 1)$	$2^{0.292n}$	[BDGL16]	$2^{\frac{0.292n}{2c}}$
n^c for $c \geq 1$	$2^{\frac{0.292n}{\lfloor c+1 \rfloor}}$	[GN08]+[BDGL16]	$2^{\frac{0.292n}{c+1}}$

[8] A. Becker, L. Ducas, N. Gama, and T. Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. SODA 2016.

1. Background on lattice reduction
2. Our results
3. **Our technical ideas and argument**
4. Conclusion and open problems

Preliminaries

GSO

Given a basis $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$, define the orthogonal projection:

$$\pi_i : \mathrm{span}(\mathbf{b}_1, \ldots, \mathbf{b}_n) \mapsto \mathrm{span}(\mathbf{b}_1, \ldots, \mathbf{b}_{i-1})^{\perp}.$$

- The vectors $\mathbf{b}_i^* = \pi_i(\mathbf{b}_i)$ for $i = 1, \ldots, n$ are the Gram-Schmidt vectors of B.
- The projected block $B_{[i,j]} = (\pi_i(\mathbf{b}_i), \pi_i(\mathbf{b}_{i+1}), \ldots, \pi_i(\mathbf{b}_j))$.

Preliminaries

SVP reduction and its extensions

Let $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ be a basis of a lattice L and $1 \leq \delta \in \mathbb{R}$.

- B is *SVP-reduced* if $\|\mathbf{b}_1\| = \lambda_1(L)$.
- B is *f-SVP-reduced* if $\|\mathbf{b}_1\| \leq f \cdot \lambda_1(L)$.
- B is *f-DSVP-reduced* if $1/\|\mathbf{b}_n^*\| \leq f \cdot \lambda_1(\text{the dual lattice of } L)$.
- B is *f-HSVP-reduced* if $\|\mathbf{b}_1\| \leq f \cdot \mathrm{vol}(L)^{1/n}$.
- B is *f-DHSVP-reduced* if $\mathrm{vol}(L)^{1/n} \leq f \cdot \|\mathbf{b}_n^*\|$.
- B is *HKZ-reduced* if $B_{[i,n]}$ is SVP-reduced for all $i = 1, \ldots, n$.

- *Hermite's constant* γ_n in dimension n is the maximum

$$\gamma_n := \max \frac{\lambda_1(L)^2}{\mathrm{vol}(L)^{2/n}} \quad \text{over all n-rank lattices} L.$$

- Fact: Any δ-SVP-oracle in rank n is also a $\delta\sqrt{\gamma_n}$-(D)HSVP-oracle in rank n.

Preliminaries

SVP reduction and its extensions

Let $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ be a basis of a lattice L and $1 \leq \delta \in \mathbb{R}$.

- B is *SVP-reduced* if $\|\mathbf{b}_1\| = \lambda_1(L)$.
- B is *f-SVP-reduced* if $\|\mathbf{b}_1\| \leq f \cdot \lambda_1(L)$.
- B is *f-DSVP-reduced* if $1/\|\mathbf{b}_n^*\| \leq f \cdot \lambda_1(\text{the dual lattice of } L)$.
- B is *f-HSVP-reduced* if $\|\mathbf{b}_1\| \leq f \cdot \mathrm{vol}(L)^{1/n}$.
- B is *f-DHSVP-reduced* if $\mathrm{vol}(L)^{1/n} \leq f \cdot \|\mathbf{b}_n^*\|$.
- B is *HKZ-reduced* if $B_{[i,n]}$ is SVP-reduced for all $i = 1, \ldots, n$.

- *Hermite's constant* γ_n in dimension n is the maximum

$$\gamma_n := \max \frac{\lambda_1(L)^2}{\mathrm{vol}(L)^{2/n}} \quad \text{over all n-rank lattices} L.$$

- Fact: Any δ-SVP-oracle in rank n is also a $\delta\sqrt{\gamma_n}$-(D)HSVP-oracle in rank n.

Preliminaries

SVP reduction and its extensions

Let $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ be a basis of a lattice L and $1 \leq \delta \in \mathbb{R}$.

- B is *SVP-reduced* if $\|\mathbf{b}_1\| = \lambda_1(L)$.
- B is *f-SVP-reduced* if $\|\mathbf{b}_1\| \leq f \cdot \lambda_1(L)$.
- B is *f-DSVP-reduced* if $1/\|\mathbf{b}_n^*\| \leq f \cdot \lambda_1(\text{the dual lattice of } L)$.
- B is *f-HSVP-reduced* if $\|\mathbf{b}_1\| \leq f \cdot \mathrm{vol}(L)^{1/n}$.
- B is *f-DHSVP-reduced* if $\mathrm{vol}(L)^{1/n} \leq f \cdot \|\mathbf{b}_n^*\|$.
- B is *HKZ-reduced* if $B_{[i,n]}$ is SVP-reduced for all $i = 1, \ldots, n$.

- *Hermite's constant* γ_n in dimension n is the maximum

$$\gamma_n := \max \frac{\lambda_1(L)^2}{\mathrm{vol}(L)^{2/n}} \text{ over all n-rank lattices} L.$$

- Fact: Any δ-SVP-oracle in rank n is also a $\delta\sqrt{\gamma_n}$-(D)HSVP-oracle in rank n.

Preliminaries

SVP reduction and its extensions

Let $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ be a basis of a lattice L and $1 \leq \delta \in \mathbb{R}$.

- B is *SVP-reduced* if $\|\mathbf{b}_1\| = \lambda_1(L)$.
- B is *f-SVP-reduced* if $\|\mathbf{b}_1\| \leq f \cdot \lambda_1(L)$.
- B is *f-DSVP-reduced* if $1/\|\mathbf{b}_n^*\| \leq f \cdot \lambda_1(\text{the dual lattice of } L)$.
- B is *f-HSVP-reduced* if $\|\mathbf{b}_1\| \leq f \cdot \mathrm{vol}(L)^{1/n}$.
- B is *f-DHSVP-reduced* if $\mathrm{vol}(L)^{1/n} \leq f \cdot \|\mathbf{b}_n^*\|$.
- B is *HKZ-reduced* if $B_{[i,n]}$ is SVP-reduced for all $i = 1, \ldots, n$.

- *Hermite's constant* γ_n in dimension n is the maximum

$$\gamma_n := \max \frac{\lambda_1(L)^2}{\mathrm{vol}(L)^{2/n}} \text{ over all n-rank lattices} L.$$

- Fact: Any δ-SVP-oracle in rank n is also a $\delta\sqrt{\gamma_n}$-(D)HSVP-oracle in rank n.

Preliminaries

SVP reduction and its extensions

Let $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ be a basis of a lattice L and $1 \le \delta \in \mathbb{R}$.

- B is *SVP-reduced* if $\|\mathbf{b}_1\| = \lambda_1(L)$.
- B is *f-SVP-reduced* if $\|\mathbf{b}_1\| \le f \cdot \lambda_1(L)$.
- B is *f-DSVP-reduced* if $1/\|\mathbf{b}_n^*\| \le f \cdot \lambda_1(\text{the dual lattice of } L)$.
- B is *f-HSVP-reduced* if $\|\mathbf{b}_1\| \le f \cdot \mathrm{vol}(L)^{1/n}$.
- B is *f-DHSVP-reduced* if $\mathrm{vol}(L)^{1/n} \le f \cdot \|\mathbf{b}_n^*\|$.
- B is *HKZ-reduced* if $B_{[i,n]}$ is SVP-reduced for all $i = 1, \ldots, n$.

- *Hermite's constant* γ_n in dimension n is the maximum

$$\gamma_n := \max \frac{\lambda_1(L)^2}{\mathrm{vol}(L)^{2/n}} \quad \text{over all n-rank lattices} L.$$

- Fact: Any δ-SVP-oracle in rank n is also a $\delta\sqrt{\gamma_n}$-(D)HSVP-oracle in rank n.

Preliminaries

SVP reduction and its extensions

Let $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ be a basis of a lattice L and $1 \leq \delta \in \mathbb{R}$.

- B is *SVP-reduced* if $\|\mathbf{b}_1\| = \lambda_1(L)$.
- B is *f-SVP-reduced* if $\|\mathbf{b}_1\| \leq f \cdot \lambda_1(L)$.
- B is *f-DSVP-reduced* if $1/\|\mathbf{b}_n^*\| \leq f \cdot \lambda_1(\text{the dual lattice of } L)$.
- B is *f-HSVP-reduced* if $\|\mathbf{b}_1\| \leq f \cdot \mathrm{vol}(L)^{1/n}$.
- B is *f-DHSVP-reduced* if $\mathrm{vol}(L)^{1/n} \leq f \cdot \|\mathbf{b}_n^*\|$.
- B is *HKZ-reduced* if $B_{[i,n]}$ is SVP-reduced for all $i = 1, \ldots, n$.

- *Hermite's constant* γ_n in dimension n is the maximum

$$\gamma_n := \max \frac{\lambda_1(L)^2}{\mathrm{vol}(L)^{2/n}} \quad \text{over all n-rank lattices} L.$$

- Fact: Any δ-SVP-oracle in rank n is also a $\delta\sqrt{\gamma_n}$-(D)HSVP-oracle in rank n.

Preliminaries

SVP reduction and its extensions

Let $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ be a basis of a lattice L and $1 \leq \delta \in \mathbb{R}$.

- B is *SVP-reduced* if $\|\mathbf{b}_1\| = \lambda_1(L)$.
- B is *f-SVP-reduced* if $\|\mathbf{b}_1\| \leq f \cdot \lambda_1(L)$.
- B is *f-DSVP-reduced* if $1/\|\mathbf{b}_n^*\| \leq f \cdot \lambda_1(\text{the dual lattice of } L)$.
- B is *f-HSVP-reduced* if $\|\mathbf{b}_1\| \leq f \cdot \mathrm{vol}(L)^{1/n}$.
- B is *f-DHSVP-reduced* if $\mathrm{vol}(L)^{1/n} \leq f \cdot \|\mathbf{b}_n^*\|$.
- B is *HKZ-reduced* if $B_{[i,n]}$ is SVP-reduced for all $i = 1, \ldots, n$.

- *Hermite's constant* γ_n in dimension n is the maximum

$$\gamma_n := \max \frac{\lambda_1(L)^2}{\mathrm{vol}(L)^{2/n}} \quad \text{over all n-rank lattices} L.$$

- Fact: Any δ-SVP-oracle in rank n is also a $\delta\sqrt{\gamma_n}$-(D)HSVP-oracle in rank n.

Preliminaries

SVP reduction and its extensions

Let $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ be a basis of a lattice L and $1 \leq \delta \in \mathbb{R}$.

- B is *SVP-reduced* if $\|\mathbf{b}_1\| = \lambda_1(L)$.
- B is *f-SVP-reduced* if $\|\mathbf{b}_1\| \leq f \cdot \lambda_1(L)$.
- B is *f-DSVP-reduced* if $1/\|\mathbf{b}_n^*\| \leq f \cdot \lambda_1(\text{the dual lattice of } L)$.
- B is *f-HSVP-reduced* if $\|\mathbf{b}_1\| \leq f \cdot \mathrm{vol}(L)^{1/n}$.
- B is *f-DHSVP-reduced* if $\mathrm{vol}(L)^{1/n} \leq f \cdot \|\mathbf{b}_n^*\|$.
- B is *HKZ-reduced* if $B_{[i,n]}$ is SVP-reduced for all $i = 1, \ldots, n$.

- *Hermite's constant* γ_n in dimension n is the maximum

$$\gamma_n := \max \frac{\lambda_1(L)^2}{\mathrm{vol}(L)^{2/n}} \text{ over all n-rank lattices} L.$$

- Fact: Any δ-SVP-oracle in rank n is also a $\delta\sqrt{\gamma_n}$-(D)HSVP-oracle in rank n.

Preliminaries

SVP reduction and its extensions

Let $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ be a basis of a lattice L and $1 \leq \delta \in \mathbb{R}$.

- B is *SVP-reduced* if $\|\mathbf{b}_1\| = \lambda_1(L)$.
- B is *f-SVP-reduced* if $\|\mathbf{b}_1\| \leq f \cdot \lambda_1(L)$.
- B is *f-DSVP-reduced* if $1/\|\mathbf{b}_n^*\| \leq f \cdot \lambda_1(\text{the dual lattice of } L)$.
- B is *f-HSVP-reduced* if $\|\mathbf{b}_1\| \leq f \cdot \mathrm{vol}(L)^{1/n}$.
- B is *f-DHSVP-reduced* if $\mathrm{vol}(L)^{1/n} \leq f \cdot \|\mathbf{b}_n^*\|$.
- B is *HKZ-reduced* if $B_{[i,n]}$ is SVP-reduced for all $i = 1, \ldots, n$.

- *Hermite's constant* γ_n in dimension n is the maximum

$$\gamma_n := \max \frac{\lambda_1(L)^2}{\mathrm{vol}(L)^{2/n}} \quad \text{over all n-rank lattices} L.$$

- Fact: Any δ-SVP-oracle in rank n is also a $\delta\sqrt{\gamma_n}$-(D)HSVP-oracle in rank n.

Preliminaries

SVP reduction and its extensions

Let $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ be a basis of a lattice L and $1 \leq \delta \in \mathbb{R}$.

- B is *SVP-reduced* if $\|\mathbf{b}_1\| = \lambda_1(L)$.
- B is *f-SVP-reduced* if $\|\mathbf{b}_1\| \leq f \cdot \lambda_1(L)$.
- B is *f-DSVP-reduced* if $1/\|\mathbf{b}_n^*\| \leq f \cdot \lambda_1(\text{the dual lattice of } L)$.
- B is *f-HSVP-reduced* if $\|\mathbf{b}_1\| \leq f \cdot \mathrm{vol}(L)^{1/n}$.
- B is *f-DHSVP-reduced* if $\mathrm{vol}(L)^{1/n} \leq f \cdot \|\mathbf{b}_n^*\|$.
- B is *HKZ-reduced* if $B_{[i,n]}$ is SVP-reduced for all $i = 1, \ldots, n$.

- *Hermite's constant* γ_n in dimension n is the maximum

$$\gamma_n := \max \frac{\lambda_1(L)^2}{\mathrm{vol}(L)^{2/n}} \quad \text{over all n-rank lattices} L.$$

- Fact: Any δ-SVP-oracle in rank n is also a $\delta\sqrt{\gamma_n}$-(D)HSVP-oracle in rank n.

Preliminaries

Warning

- It is trivial to replace exact SVP-oracle with δ-SVP-oracle in our arguments.
- Argue the case $\delta = 1$.

Preliminaries

Warning

- It is trivial to replace exact SVP-oracle with δ-SVP-oracle in our arguments.
- Argue the case $\delta = 1$.

Preliminaries

Warning

- It is trivial to replace exact SVP-oracle with δ-SVP-oracle in our arguments.
- Argue the case $\delta = 1$.

Approximating SVP with sublinear factor

Given a lattice L of rank n and a SVP-oracle in rank k with $k \leq n \leq 2k-1$.

- Goal: Find a nonzero vector $\mathbf{b} \in L$ s.t.

$$\|\mathbf{b}\| \lesssim \gamma_k^{\frac{n}{2k}} \cdot \lambda_1(L).$$

- Idea: If finding a basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of L s.t. $\mathrm{vol}(\mathbf{b}_1, \ldots, \mathbf{b}_k)$ is small w.r.t. $\lambda_1(L)$, then

$$\lambda_1(L(\mathbf{b}_1, \ldots, \mathbf{b}_k)) \leq \sqrt{\gamma_k} \cdot \mathrm{vol}(\mathbf{b}_1, \ldots, \mathbf{b}_k)^{1/k}.$$

- Issue: How to efficiently find a basis whose first k basis vectors has small volume w.r.t $\lambda_1(L)$?

Approximating SVP with sublinear factor

Given a lattice L of rank n and a SVP-oracle in rank k with $k \le n \le 2k-1$.

- Goal: Find a nonzero vector $\mathbf{b} \in L$ s.t.

$$\|\mathbf{b}\| \lesssim \gamma_k^{\frac{n}{2k}} \cdot \lambda_1(L).$$

- Idea: If finding a basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of L s.t. $\mathrm{vol}(\mathbf{b}_1, \ldots, \mathbf{b}_k)$ is small w.r.t. $\lambda_1(L)$, then

$$\lambda_1(L(\mathbf{b}_1, \ldots, \mathbf{b}_k)) \le \sqrt{\gamma_k} \cdot \mathrm{vol}(\mathbf{b}_1, \ldots, \mathbf{b}_k)^{1/k}.$$

- Issue: How to efficiently find a basis whose first k basis vectors has small volume w.r.t $\lambda_1(L)$?

Approximating SVP with sublinear factor

Given a lattice L of rank n and a SVP-oracle in rank k with $k \le n \le 2k-1$.

- Goal: Find a nonzero vector $\mathbf{b} \in L$ s.t.

$$\|\mathbf{b}\| \lesssim \gamma_k^{\frac{n}{2k}} \cdot \lambda_1(L).$$

- Idea: If finding a basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of L s.t. $\mathrm{vol}(\mathbf{b}_1, \ldots, \mathbf{b}_k)$ is small w.r.t. $\lambda_1(L)$, then

$$\lambda_1(L(\mathbf{b}_1, \ldots, \mathbf{b}_k)) \le \sqrt{\gamma_k} \cdot \mathrm{vol}(\mathbf{b}_1, \ldots, \mathbf{b}_k)^{1/k}.$$

- Issue: How to efficiently find a basis whose first k basis vectors has small volume w.r.t $\lambda_1(L)$?

Approximating SVP with sublinear factor

Issue: How to efficiently find a basis whose first k basis vectors has small volume w.r.t $\lambda_1(L)$?

GN-slide-reduction in case $n = 2k$

- Definition: A basis $\mathbf{B}$ of rank $2k$ is (ε, k)-*slide-reduced* if
  - Primal conditions: $B_{[1,k]}$ and $B_{[k+1,2k]}$ are HKZ-reduced.
  - Dual condition: $B_{[2,k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.
- Observation: Let $(\mathbf{b}_1, \ldots, \mathbf{b}_{2k})$ be a (ε, k)-slide-reduced basis of a lattice L.
  - If $\lambda_1(L) = \lambda_1(L((\mathbf{b}_1, \ldots, \mathbf{b}_k))$, then $\|\mathbf{b}_1\| = \lambda_1(L)$;
  - If $\lambda_1(L) < \lambda_1(L((\mathbf{b}_1, \ldots, \mathbf{b}_k))$, then $\|\mathbf{b}^*_{k+1}\| \leq \lambda_1(L)$ implies: $\mathrm{vol}(\mathbf{b}_1, \ldots, \mathbf{b}_k)$ is small w.r.t. $\lambda_1(L)$.

Approximating SVP with sublinear factor

Issue: How to efficiently find a basis whose first k basis vectors has small volume w.r.t $\lambda_1(L)$?

GN-slide-reduction in case $n = 2k$

- Definition: A basis $\mathbf{B}$ of rank $2k$ is (ε, k)-*slide-reduced* if
  - Primal conditions: $B_{[1,k]}$ and $B_{[k+1,2k]}$ are HKZ-reduced.
  - Dual condition: $B_{[2,k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.
- Observation: Let $(\mathbf{b}_1, \ldots, \mathbf{b}_{2k})$ be a (ε, k)-slide-reduced basis of a lattice L.
  - If $\lambda_1(L) = \lambda_1(L((\mathbf{b}_1, \ldots, \mathbf{b}_k)))$, then $\|\mathbf{b}_1\| = \lambda_1(L)$;
  - If $\lambda_1(L) < \lambda_1(L((\mathbf{b}_1, \ldots, \mathbf{b}_k)))$, then $\|\mathbf{b}^*_{k+1}\| \le \lambda_1(L)$ implies: $\mathrm{vol}(\mathbf{b}_1, \ldots, \mathbf{b}_k)$ is small w.r.t. $\lambda_1(L)$.

Approximating SVP with sublinear factor

Issue: How to efficiently find a basis whose first k basis vectors has small volume w.r.t $\lambda_1(L)$?

GN-slide-reduction in case $n = 2k$

- Definition: A basis $\mathbf{B}$ of rank $2k$ is (ε, k)-*slide-reduced* if
  - Primal conditions: $B_{[1,k]}$ and $B_{[k+1,2k]}$ are HKZ-reduced.
  - Dual condition: $B_{[2,k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.
- Observation: Let $(\mathbf{b}_1, \ldots, \mathbf{b}_{2k})$ be a (ε, k)-slide-reduced basis of a lattice L.
  - If $\lambda_1(L) = \lambda_1(L((\mathbf{b}_1, \ldots, \mathbf{b}_k))$, then $\|\mathbf{b}_1\| = \lambda_1(L)$;
  - If $\lambda_1(L) < \lambda_1(L((\mathbf{b}_1, \ldots, \mathbf{b}_k))$, then $\|\mathbf{b}^*_{k+1}\| \le \lambda_1(L)$ implies: $\mathrm{vol}(\mathbf{b}_1, \ldots, \mathbf{b}_k)$ is small w.r.t. $\lambda_1(L)$.

Approximating SVP with sublinear factor

Issue: How to efficiently find a basis whose first k basis vectors has small volume w.r.t $\lambda_1(L)$?

GN-slide-reduction in case $n = 2k$

- Definition: A basis $\mathbf{B}$ of rank $2k$ is (ε, k)*-slide-reduced* if
  - Primal conditions: $B_{[1,k]}$ and $B_{[k+1,2k]}$ are HKZ-reduced.
  - Dual condition: $B_{[2,k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.
- Observation: Let $(\mathbf{b}_1, \ldots, \mathbf{b}_{2k})$ be a (ε, k)-slide-reduced basis of a lattice L.
  - If $\lambda_1(L) = \lambda_1(L((\mathbf{b}_1, \ldots, \mathbf{b}_k)))$, then $\|\mathbf{b}_1\| = \lambda_1(L)$;
  - If $\lambda_1(L) < \lambda_1(L((\mathbf{b}_1, \ldots, \mathbf{b}_k)))$, then $\|\mathbf{b}^*_{k+1}\| \leq \lambda_1(L)$ implies: $\mathrm{vol}(\mathbf{b}_1, \ldots, \mathbf{b}_k)$ is small w.r.t. $\lambda_1(L)$.

Approximating SVP with sublinear factor

Issue: How to efficiently find a basis whose first k basis vectors has small volume w.r.t $\lambda_1(L)$?

GN-slide-reduction in case $n = 2k$

- Definition: A basis $\mathbf{B}$ of rank $2k$ is (ε, k)*-slide-reduced* if
  - Primal conditions: $B_{[1,k]}$ and $B_{[k+1,2k]}$ are HKZ-reduced.
  - Dual condition: $B_{[2,k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.
- Observation: Let $(\mathbf{b}_1, \ldots, \mathbf{b}_{2k})$ be a (ε, k)-slide-reduced basis of a lattice L.
  - If $\lambda_1(L) = \lambda_1(L((\mathbf{b}_1, \ldots, \mathbf{b}_k))$, then $\|\mathbf{b}_1\| = \lambda_1(L)$;
  - If $\lambda_1(L) < \lambda_1(L((\mathbf{b}_1, \ldots, \mathbf{b}_k))$, then $\|\mathbf{b}^*_{k+1}\| \le \lambda_1(L)$ implies: $\mathrm{vol}(\mathbf{b}_1, \ldots, \mathbf{b}_k)$ is small w.r.t. $\lambda_1(L)$.

Approximating SVP with sublinear factor

Issue: How to efficiently find a basis whose first k basis vectors has small volume w.r.t $\lambda_1(L)$?

GN-slide-reduction in case $n = 2k$

- Definition: A basis $\mathbf{B}$ of rank $2k$ is (ε, k)*-slide-reduced* if
  - Primal conditions: $B_{[1,k]}$ and $B_{[k+1,2k]}$ are HKZ-reduced.
  - Dual condition: $B_{[2,k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.
- Observation: Let $(\mathbf{b}_1, \ldots, \mathbf{b}_{2k})$ be a (ε, k)-slide-reduced basis of a lattice L.
  - If $\lambda_1(L) = \lambda_1(L((\mathbf{b}_1, \ldots, \mathbf{b}_k))$, then $\|\mathbf{b}_1\| = \lambda_1(L)$;
  - If $\lambda_1(L) < \lambda_1(L((\mathbf{b}_1, \ldots, \mathbf{b}_k))$, then $\|\mathbf{b}^*_{k+1}\| \leq \lambda_1(L)$ implies: $\mathrm{vol}(\mathbf{b}_1, \ldots, \mathbf{b}_k)$ is small w.r.t. $\lambda_1(L)$.

Approximating SVP with sublinear factor

Issue: How to efficiently find a basis whose first k basis vectors has small volume w.r.t $\lambda_1(L)$?

GN-slide-reduction in case $n = 2k$

- Definition: A basis $\mathbf{B}$ of rank $2k$ is (ε, k)-*slide-reduced* if
  - Primal conditions: $B_{[1,k]}$ and $B_{[k+1,2k]}$ are HKZ-reduced.
  - Dual condition: $B_{[2,k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.
- Observation: Let $(\mathbf{b}_1, \ldots, \mathbf{b}_{2k})$ be a (ε, k)-slide-reduced basis of a lattice L.
  - If $\lambda_1(L) = \lambda_1(L((\mathbf{b}_1, \ldots, \mathbf{b}_k))$, then $\|\mathbf{b}_1\| = \lambda_1(L)$;
  - If $\lambda_1(L) < \lambda_1(L((\mathbf{b}_1, \ldots, \mathbf{b}_k))$, then $\|\mathbf{b}^*_{k+1}\| \leq \lambda_1(L)$ implies: $\mathrm{vol}(\mathbf{b}_1, \ldots, \mathbf{b}_k)$ is small w.r.t. $\lambda_1(L)$.

Approximating SVP with sublinear factor

GN-slide-reduction in case $n = 2k$: A basis B of rank $2k$ is (ε, k)-*slide-reduced* if

- Primal conditions: both $B_{[1,k]}$ and $B_{[k+1,2k]}$ are HKZ-reduced.
- Dual condition: $B_{[2,k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.

Our variant of slide-reduction

Let $n = k + q$ with $0 \le q \le k - 1$ and $k \ge 2$.

- Definition: A basis $\mathbf{B}$ of rank n is *k-slide-reduced* if
  - Primal conditions: for all $i = q+1, \ldots, k$, $B_{[i,n]}$ is SVP-reduced.
  - $B_{[1,q+1]}$ is $\sqrt{\gamma_{q+1}}$-DHSVP-reduced.
- Property: If $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ be a k-slide-reduced basis of a lattice L, then

$$\lambda_1(L(\mathbf{b}_1, \ldots, \mathbf{b}_k)) \le \sqrt{\gamma_k} \gamma_{q+1}^{\frac{q+1}{2k}} \lambda_1(L).$$

Approximating SVP with sublinear factor

GN-slide-reduction in case $n = 2k$: A basis B of rank $2k$ is (ε, k)*-slide-reduced* if

- Primal conditions: both $B_{[1,k]}$ and $B_{[k+1,2k]}$ are HKZ-reduced.
- Dual condition: $B_{[2,k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.

Our variant of slide-reduction

Let $n = k + q$ with $0 \leq q \leq k - 1$ and $k \geq 2$.

- Definition: A basis $\mathbf{B}$ of rank n is *k-slide-reduced* if
  - Primal conditions: for all $i = q + 1, \ldots, k$, [illegible] is SVP-reduced.
  - [illegible] is [illegible]-DHSVP-reduced.
- Property: If $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ be a k-slide-reduced basis of a lattice L, then

$$\lambda_1(L(\mathbf{b}_1, \ldots, \mathbf{b}_k)) \leq \sqrt{\gamma_k} \gamma_{q+1}^{\frac{q+1}{2k}} \lambda_1(L).$$

Approximating SVP with sublinear factor

GN-slide-reduction in case $n = 2k$: A basis B of rank $2k$ is (ε, k)-*slide-reduced* if

- Primal conditions: both $B_{[1,k]}$ and $B_{[k+1,2k]}$ are HKZ-reduced.
- Dual condition: $B_{[2,k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.

Our variant of slide-reduction

Let $n = k + q$ with $0 \le q \le k - 1$ and $k \ge 2$.

- Definition: A basis $\mathbf{B}$ of rank n is k*-slide-reduced* if
  - Primal conditions: for all $i = q+1, \ldots, k$, $B_{[i,n]}$ is SVP-reduced.
  - Dual condition: $B_{[1,q+1]}$ is $\sqrt{\gamma_{q+1}}$-DHSVP-reduced.
- Property: If $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ be a k-slide-reduced basis of a lattice L, then

$$\lambda_1(L(\mathbf{b}_1, \ldots, \mathbf{b}_k)) \le \sqrt{\gamma_k}\gamma_{q+1}^{\frac{q+1}{2k}}\lambda_1(L).$$

Approximating SVP with sublinear factor

GN-slide-reduction in case $n = 2k$: A basis B of rank $2k$ is (ε, k)-*slide-reduced* if

- Primal conditions: both $B_{[1,k]}$ and $B_{[k+1,2k]}$ are HKZ-reduced.
- Dual condition: $B_{[2,k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.

Our variant of slide-reduction

Let $n = k + q$ with $0 \leq q \leq k - 1$ and $k \geq 2$.

- Definition: A basis $\mathbf{B}$ of rank n is k-*slide-reduced* if
  - Primal conditions: for all $i = q+1, \ldots, k$, $B_{[i,n]}$ is SVP-reduced.
  - Dual condition: $B_{[1,q+1]}$ is $\sqrt{\gamma_{q+1}}$-DHSVP-reduced.
- Property: If $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ be a k-slide-reduced basis of a lattice L, then

$$\lambda_1(L(\mathbf{b}_1, \ldots, \mathbf{b}_k)) \leq \sqrt{\gamma_k}\gamma_{q+1}^{\frac{q+1}{2k}}\lambda_1(L).$$

Approximating SVP with sublinear factor

GN-slide-reduction in case $n = 2k$: A basis B of rank $2k$ is (ε, k)-*slide-reduced* if

- Primal conditions: both $B_{[1,k]}$ and $B_{[k+1,2k]}$ are HKZ-reduced.
- Dual condition: $B_{[2,k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.

Our variant of slide-reduction

Let $n = k + q$ with $0 \le q \le k-1$ and $k \ge 2$.

- Definition: A basis $\mathbf{B}$ of rank n is k-*slide-reduced* if
  - Primal conditions: for all $i = q+1, \ldots, k$, $B_{[i,n]}$ is SVP-reduced.
  - Dual condition: $B_{[1,q+1]}$ is $\sqrt{\gamma_{q+1}}$-DHSVP-reduced.
- Property: If $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ be a k-slide-reduced basis of a lattice L, then

$$\lambda_1(L(\mathbf{b}_1, \ldots, \mathbf{b}_k)) \le \sqrt{\gamma_k}\gamma_{q+1}^{\frac{q+1}{2k}}\lambda_1(L).$$

Approximating SVP with sublinear factor

GN-slide-reduction in case $n = 2k$: A basis B of rank $2k$ is (ε, k)-*slide-reduced* if

- Primal conditions: both $B_{[1,k]}$ and $B_{[k+1,2k]}$ are HKZ-reduced.
- Dual condition: $B_{[2,k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.

Our variant of slide-reduction

Let $n = k + q$ with $0 \leq q \leq k-1$ and $k \geq 2$.

- Definition: A basis $\mathbf{B}$ of rank n is k-*slide-reduced* if
  - Primal conditions: for all $i = q+1, \ldots, k$, $B_{[i,n]}$ is SVP-reduced.
  - Dual condition: $B_{[1,q+1]}$ is $\sqrt{\gamma_{q+1}}$-DHSVP-reduced.
- Property: If $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ be a k-slide-reduced basis of a lattice L, then

$$\lambda_1(L(\mathbf{b}_1, \ldots, \mathbf{b}_k)) \leq \sqrt{\gamma_k}\gamma_{q+1}^{\frac{q+1}{2k}}\lambda_1(L).$$

Approximating SVP with sublinear factor

GN-slide-reduction in case $n = 2k$: A basis B of rank $2k$ is (ε, k)-*slide-reduced* if

- Primal conditions: both $B_{[1,k]}$ and $B_{[k+1,2k]}$ are HKZ-reduced.
- Dual condition: $B_{[2,k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.

Our variant of slide-reduction

Let $n = k + q$ with $0 \le q \le k-1$ and $k \ge 2$.

- Definition: A basis $\mathbf{B}$ of rank n is k-*slide-reduced* if
  - Primal conditions: for all $i = q+1, \ldots, k$, $B_{[i,n]}$ is SVP-reduced.
  - Dual condition: $B_{[1,q+1]}$ is $\sqrt{\gamma_{q+1}}$-DHSVP-reduced.
- Property: If $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ be a k-slide-reduced basis of a lattice L, then

$$\lambda_1(L(\mathbf{b}_1, \ldots, \mathbf{b}_k)) \le \sqrt{\gamma_k} \gamma_{q+1}^{\frac{q+1}{2k}} \lambda_1(L).$$

Algorithm 1 Approximating SVP with sublinear factor

Input: Blocksize $k \geq 2$, termination factor $\varepsilon > 0$, a basis B of an integer lattice L of rank $n = k + q$ where $1 \leq q < k$, and an SVP-oracle in rank k.

Output: A nonzero vector of L.

while $\mathrm{vol}(\mathbf{B}_{[1,q]})$ is modified by the loop **do**
    SVP-reduce $B_{[q+1,n]}$
    if $B_{[1,q+1]}$ is not $\sqrt{(1+\varepsilon)\gamma_{q+1}}$-DHSVP-reduced **then**
        $\sqrt{\gamma_{q+1}}$-DHSVP-reduce $B_{[1,q+1]}$
end while
for $i = q+2$ to k **do** SVP-reduce $B_{[i,n]}$
SVP-reduce $B_{[1,k]}$
return The first basis vector.

★ Th: This algorithm terminates within $\mathrm{poly}(B_{\mathrm{input}}, 1/\varepsilon)$ calls to SVP-oracle in rank k, and outputs a nonzero vector **b** of L s.t.

$$\|\mathbf{b}\| \leq \sqrt{\gamma_k}\left((1+\varepsilon)\gamma_{q+1}\right)^{\frac{q+1}{2k}} \lambda_1(L).$$

Algorithm 2 Approximating SVP with sublinear factor

Input: Blocksize $k \geq 2$, termination factor $\varepsilon > 0$, a basis B of an integer lattice L of rank $n = k + q$ where $1 \leq q < k$, and an SVP-oracle in rank k.

Output: A nonzero vector of L.

1: **while** $\mathrm{vol}(\mathbf{B}_{[1,q]})$ is modified by the loop **do**

2: SVP-reduce $B_{[q+1,n]}$

3: **if** $B_{[1,q+1]}$ is not $\sqrt{(1+\varepsilon)\gamma_{q+1}}$-DHSVP-reduced **then** $\sqrt{\gamma_{q+1}}$-DHSVP-reduce $B_{[1,q+1]}$

4: **end while**

5: **for** $i = q+2$ to k **do** SVP-reduce $B_{[i,n]}$

6: SVP-reduce $B_{[1,k]}$

7: **return** The first basis vector.

★ Th: This algorithm terminates within $\mathrm{poly}(B_{\mathrm{input}}, 1/\varepsilon)$ calls to SVP-oracle in rank k, and outputs a nonzero vector $\mathbf{b}$ of L s.t.

$$\|\mathbf{b}\| \leq \sqrt{\gamma_k}\left((1+\varepsilon)\gamma_{q+1}\right)^{\frac{q+1}{2k}} \lambda_1(L).$$

Algorithm 3 Approximating SVP with sublinear factor

Input: Blocksize $k \geq 2$, termination factor $\varepsilon > 0$, a basis B of an integer lattice L of rank $n = k + q$ where $1 \leq q < k$, and an SVP-oracle in rank k.

Output: A nonzero vector of L.

1: **while** $\mathrm{vol}(\mathbf{B}_{[1,q]})$ is modified by the loop **do**

2: SVP-reduce $B_{[q+1,n]}$

3: **if** $B_{[1,q+1]}$ is not $\sqrt{(1+\varepsilon)\gamma_{q+1}}$-DHSVP-reduced **then** $\sqrt{\gamma_{q+1}}$-DHSVP-reduce $B_{[1,q+1]}$

4: **end while**

5: **for** $i = q+2$ to k **do** SVP-reduce $B_{[i,n]}$

6: SVP-reduce $B_{[1,k]}$

7: **return** The first basis vector.

★ Th: This algorithm terminates within $\mathrm{poly}(B_{\mathrm{input}}, 1/\varepsilon)$ calls to SVP-oracle in rank k, and outputs a nonzero vector $\mathbf{b}$ of L s.t.

$$\|\mathbf{b}\| \leq \sqrt{\gamma_k}\left((1+\varepsilon)\gamma_{q+1}\right)^{\frac{q+1}{2k}}\lambda_1(L).$$

Algorithm 4 Approximating SVP with sublinear factor

Input: Blocksize $k \geq 2$, termination factor $\varepsilon > 0$, a basis B of an integer lattice L of rank $n = k + q$ where $1 \leq q < k$, and an SVP-oracle in rank k.

Output: A nonzero vector of L.

1: **while** $\mathrm{vol}(\mathbf{B}_{[1,q]})$ is modified by the loop **do**

2: SVP-reduce $B_{[q+1,n]}$

3: **if** $B_{[1,q+1]}$ is not $\sqrt{(1+\varepsilon)\gamma_{q+1}}$-DHSVP-reduced **then** $\sqrt{\gamma_{q+1}}$-DHSVP-reduce $B_{[1,q+1]}$

4: **end while**

5: **for** $i = q + 2$ to k **do** SVP-reduce $B_{[i,n]}$

6: SVP-reduce $B_{[1,k]}$

7: **return** The first basis vector.

★ Th: This algorithm terminates within $\mathrm{poly}(B_{\text{input}}, 1/\varepsilon)$ calls to SVP-oracle in rank k, and outputs a nonzero vector $\mathbf{b}$ of L s.t.

$$\|\mathbf{b}\| \leq \sqrt{\gamma_k}\left((1+\varepsilon)\gamma_{q+1}\right)^{\frac{q+1}{2k}} \lambda_1(L).$$

Algorithm 5 Approximating SVP with sublinear factor

Input: Blocksize $k \geq 2$, termination factor $\varepsilon > 0$, a basis B of an integer lattice L of rank $n = k + q$ where $1 \leq q < k$, and an SVP-oracle in rank k.

Output: A nonzero vector of L.

1: **while** $\mathrm{vol}(\mathbf{B}_{[1,q]})$ is modified by the loop **do**

2: SVP-reduce $B_{[q+1,n]}$

3: **if** $B_{[1,q+1]}$ is not $\sqrt{(1+\varepsilon)\gamma_{q+1}}$-DHSVP-reduced **then** $\sqrt{\gamma_{q+1}}$-DHSVP-reduce $B_{[1,q+1]}$

4: **end while**

5: **for** $i = q + 2$ to k **do** SVP-reduce $B_{[i,n]}$

6: SVP-reduce $B_{[1,k]}$

7: **return** The first basis vector.

★ Th: This algorithm terminates within $\mathrm{poly}(B_{\mathrm{input}}, 1/\varepsilon)$ calls to SVP-oracle in rank k, and outputs a nonzero vector $\mathbf{b}$ of L s.t.

$$\|\mathbf{b}\| \leq \sqrt{\gamma_k} \left((1+\varepsilon)\gamma_{q+1}\right)^{\frac{q+1}{2k}} \lambda_1(L).$$

Algorithm 6 Approximating SVP with sublinear factor

Input: Blocksize $k \geq 2$, termination factor $\varepsilon > 0$, a basis B of an integer lattice L of rank $n = k + q$ where $1 \leq q < k$, and an SVP-oracle in rank k.

Output: A nonzero vector of L.

1: **while** $\mathrm{vol}(\mathbf{B}_{[1,q]})$ is modified by the loop **do**

2: SVP-reduce $B_{[q+1,n]}$

3: **if** $B_{[1,q+1]}$ is not $\sqrt{(1+\varepsilon)\gamma_{q+1}}$-DHSVP-reduced **then** $\sqrt{\gamma_{q+1}}$-DHSVP-reduce $B_{[1,q+1]}$

4: **end while**

5: **for** $i = q+2$ to k **do** SVP-reduce $B_{[i,n]}$

6: SVP-reduce $B_{[1,k]}$

7: **return** The first basis vector.

★ Th: This algorithm terminates within $\mathrm{poly}(B_{\mathrm{input}}, 1/\varepsilon)$ calls to SVP-oracle in rank k, and outputs a nonzero vector **b** of L s.t.

$$\|\mathbf{b}\| \leq \sqrt{\gamma_k}\left((1+\varepsilon)\gamma_{q+1}\right)^{\frac{q+1}{2k}} \lambda_1(L).$$

Algorithm 7 Approximating SVP with sublinear factor

Input: Blocksize $k \geq 2$, termination factor $\varepsilon > 0$, a basis B of an integer lattice L of rank $n = k + q$ where $1 \leq q < k$, and an SVP-oracle in rank k.

Output: A nonzero vector of L.

1: **while** $\mathrm{vol}(\mathbf{B}_{[1,q]})$ is modified by the loop **do**

2: SVP-reduce $B_{[q+1,n]}$

3: **if** $B_{[1,q+1]}$ is not $\sqrt{(1+\varepsilon)\gamma_{q+1}}$-DHSVP-reduced **then** $\sqrt{\gamma_{q+1}}$-DHSVP-reduce $B_{[1,q+1]}$

4: **end while**

5: **for** $i = q+2$ to k **do** SVP-reduce $B_{[i,n]}$

6: SVP-reduce $B_{[1,k]}$

7: **return** The first basis vector.

★ Th: This algorithm terminates within $\mathrm{poly}(B_{\mathrm{input}}, 1/\varepsilon)$ calls to SVP-oracle in rank k, and outputs a nonzero vector **b** of L s.t.

$$\|\mathbf{b}\| \leq \sqrt{\gamma_k}\left((1+\varepsilon)\gamma_{q+1}\right)^{\frac{q+1}{2k}} \lambda_1(L).$$

Algorithm 8 Approximating SVP with sublinear factor

Input: Blocksize $k \geq 2$, termination factor $\varepsilon > 0$, a basis B of an integer lattice L of rank $n = k + q$ where $1 \leq q < k$, and an SVP-oracle in rank k.

Output: A nonzero vector of L.

1: **while** $\mathrm{vol}(\mathbf{B}_{[1,q]})$ is modified by the loop **do**

2: SVP-reduce $B_{[q+1,n]}$

3: **if** $B_{[1,q+1]}$ is not $\sqrt{(1+\varepsilon)\gamma_{q+1}}$-DHSVP-reduced **then** $\sqrt{\gamma_{q+1}}$-DHSVP-reduce $B_{[1,q+1]}$

4: **end while**

5: **for** $i = q + 2$ to k **do** SVP-reduce $B_{[i,n]}$

6: SVP-reduce $B_{[1,k]}$

7: **return** The first basis vector.

★ Th: This algorithm terminates within $\mathrm{poly}(B_{\text{input}}, 1/\varepsilon)$ calls to SVP-oracle in rank k, and outputs a nonzero vector $\mathbf{b}$ of L s.t.

$$\|\mathbf{b}\| \leq \sqrt{\gamma_k}\left((1+\varepsilon)\gamma_{q+1}\right)^{\frac{q+1}{2k}} \lambda_1(L).$$

Algorithm 9 Approximating SVP with sublinear factor

Input: Blocksize $k \geq 2$, termination factor $\varepsilon > 0$, a basis B of an integer lattice L of rank $n = k + q$ where $1 \leq q < k$, and an SVP-oracle in rank k.

Output: A nonzero vector of L.

1: **while** $\mathrm{vol}(\mathbf{B}_{[1,q]})$ is modified by the loop **do**

2: SVP-reduce $B_{[q+1,n]}$

3: **if** $B_{[1,q+1]}$ is not $\sqrt{(1+\varepsilon)\gamma_{q+1}}$-DHSVP-reduced **then** $\sqrt{\gamma_{q+1}}$-DHSVP-reduce $B_{[1,q+1]}$

4: **end while**

5: **for** $i = q + 2$ to k **do** SVP-reduce $B_{[i,n]}$

6: SVP-reduce $B_{[1,k]}$

7: **return** The first basis vector.

★ Th: This algorithm terminates within $\mathrm{poly}(B_{\mathrm{input}}, 1/\varepsilon)$ calls to SVP-oracle in rank k, and outputs a nonzero vector $\mathbf{b}$ of L s.t.

$$\|\mathbf{b}\| \leq \sqrt{\gamma_k}\left((1+\varepsilon)\gamma_{q+1}\right)^{\frac{q+1}{2k}} \lambda_1(L).$$

Approximating SVP with (at least) polynomial factor

GN-slide-reduction: A basis B of rank $n = pk \geq 2k$ is (ε, k)-slide-reduced if

1. Primal conditions: for all $i \in [0; p-1]$, $B_{[ik+1, ik+k]}$ is HKZ-reduced.
2. Dual condition: for all $i \in [0; p-2]$, $B_{[ik+2, ik+k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.

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

- Goal: Extend GN-slide-reduction into the case $n = pk + q \geq 2k$ with $0 \leq q < k$ s.t. Eq. (1) still holds.
- Idea: Wrap "the extra q vectors" and "its nearby k vectors" into a bigger block of size $k + q$.
- Issue: With SVP-oracle in rank k, how to efficiently find a basis $(\mathbf{c}_1, \ldots, \mathbf{c}_m)$ for any lattice Λ of rank $m \in [k, 2k]$ s.t.

$$\|\mathbf{c}_1\| \lesssim \gamma_k^{\frac{m-1}{2(k-1)}} \cdot \mathrm{vol}(\Lambda)^{1/m}? \quad \Leftarrow \quad \text{DBKZ}$$

Approximating SVP with (at least) polynomial factor

GN-slide-reduction: A basis B of rank $n = pk \geq 2k$ is (ε, k)-slide-reduced if

1. Primal conditions: for all $i \in [0; p-1]$, $B_{[ik+1, ik+k]}$ is HKZ-reduced.
2. Dual condition: for all $i \in [0; p-2]$, $B_{[ik+2, ik+k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.

$$\Rightarrow \|\mathbf{b}_1\| \leq ((1+\varepsilon)\gamma_k)^{\frac{n-k}{k-1}} \cdot \lambda_1(L). \quad (1)$$

- Goal: Extend GN-slide-reduction into the case $n = pk + q \geq 2k$ with $0 \leq q < k$ s.t. Eq. (1) still holds.
- Idea: Wrap "the extra q vectors" and "its nearby k vectors" into a bigger block of size $k + q$.
- Issue: With SVP-oracle in rank k, how to efficiently find a basis $(\mathbf{c}_1, \ldots, \mathbf{c}_m)$ for any lattice Λ of rank $m \in [k, 2k]$ s.t.

$$\|\mathbf{c}_1\| \lesssim \gamma_k^{\frac{m-1}{2(k-1)}} \cdot \mathrm{vol}(\Lambda)^{1/m}? \quad \Leftarrow \quad \text{DBKZ}$$

Approximating SVP with (at least) polynomial factor

GN-slide-reduction: A basis B of rank $n = pk \geq 2k$ is (ε, k)-slide-reduced if

1. Primal conditions: for all $i \in [0; p-1]$, $B_{[ik+1, ik+k]}$ is HKZ-reduced.
2. Dual condition: for all $i \in [0; p-2]$, $B_{[ik+2, ik+k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.

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

- Goal: Extend GN-slide-reduction into the case $n = pk + q \geq 2k$ with $0 \leq q < k$ s.t. Eq. (1) still holds.
- Idea: Wrap "the extra q vectors" and "its nearby k vectors" into a bigger block of size $k+q$.
- Issue: With SVP-oracle in rank k, how to efficiently find a basis $(\mathbf{c}_1, \ldots, \mathbf{c}_m)$ for any lattice Λ of rank $m \in [k, 2k]$ s.t.

$$\|\mathbf{c}_1\| \lesssim \gamma_k^{\frac{m-1}{2(k-1)}} \cdot \mathrm{vol}(\Lambda)^{1/m}? \quad \Leftarrow \quad \text{DBKZ}$$

Approximating SVP with (at least) polynomial factor

GN-slide-reduction: A basis B of rank $n = pk \geq 2k$ is (ε, k)-slide-reduced if

1. Primal conditions: for all $i \in [0; p-1]$, $B_{[ik+1, ik+k]}$ is HKZ-reduced.
2. Dual condition: for all $i \in [0; p-2]$, $B_{[ik+2, ik+k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.

$$\Rightarrow \quad \|\mathbf{b}_1\| \leq ((1+\varepsilon)\gamma_k)^{\frac{n-k}{k-1}} \cdot \lambda_1(L). \tag{1}$$

- Goal: Extend GN-slide-reduction into the case $n = pk + q \geq 2k$ with $0 \leq q < k$ s.t. Eq. (1) still holds.
- Idea: Wrap “the extra q vectors” and “its nearby k vectors” into a bigger block of size $k + q$.
- Issue: With SVP-oracle in rank k, how to efficiently find a basis $(\mathbf{c}_1, \ldots, \mathbf{c}_m)$ for any lattice Λ of rank $m \in [k, 2k]$ s.t.

$$\|\mathbf{c}_1\| \lesssim \gamma_k^{\frac{m-1}{2(k-1)}} \cdot \mathrm{vol}(\Lambda)^{1/m}? \quad \Leftarrow \quad \text{DBKZ}$$

Approximating SVP with (at least) polynomial factor

GN-slide-reduction: A basis B of rank $n = pk \geq 2k$ is (ε, k)-slide-reduced if

1. Primal conditions: for all $i \in [0; p-1]$, $B_{[ik+1, ik+k]}$ is HKZ-reduced.
2. Dual condition: for all $i \in [0; p-2]$, $B_{[ik+2, ik+k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.

$$\Rightarrow \quad \|\mathbf{b}_1\| \leq ((1+\varepsilon)\gamma_k)^{\frac{n-k}{k-1}} \cdot \lambda_1(L). \tag{1}$$

- Goal: Extend GN-slide-reduction into the case $n = pk + q \geq 2k$ with $0 \leq q < k$ s.t. Eq. (1) still holds.
- Idea: Wrap "the extra q vectors" and "its nearby k vectors" into a bigger block of size $k + q$.
- Issue: With SVP-oracle in rank k, how to efficiently find a basis $(\mathbf{c}_1, \ldots, \mathbf{c}_m)$ for any lattice Λ of rank $m \in [k, 2k]$ s.t. $\|\mathbf{c}_1\| \lesssim \gamma_k^{\frac{m-1}{2(k-1)}} \cdot \mathrm{vol}(\Lambda)^{1/m}$? $\Leftarrow$ DBKZ

Approximating SVP with (at least) polynomial factor

GN-slide-reduction: A basis B of rank $n = pk \geq 2k$ is (ε, k)-slide-reduced if

1. Primal conditions: for all $i \in [0; p-1]$, $B_{[ik+1, ik+k]}$ is HKZ-reduced.
2. Dual condition: for all $i \in [0; p-2]$, $B_{[ik+2, ik+k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.

$$\Rightarrow \quad \|\mathbf{b}_1\| \leq ((1+\varepsilon)\gamma_k)^{\frac{n-k}{k-1}} \cdot \lambda_1(L). \tag{1}$$

- Goal: Extend GN-slide-reduction into the case $n = pk + q \geq 2k$ with $0 \leq q < k$ s.t. Eq. (1) still holds.
- Idea: Wrap "the extra q vectors" and "its nearby k vectors" into a bigger block of size $k + q$.
- Issue: With SVP-oracle in rank k, how to efficiently find a basis $(\mathbf{c}_1, \ldots, \mathbf{c}_m)$ for any lattice Λ of rank $m \in [k, 2k]$ s.t. $\|\mathbf{c}_1\| \lesssim \gamma_k^{\frac{m-1}{2(k-1)}} \cdot \mathrm{vol}(\Lambda)^{1/m}$? $\Leftarrow$ DBKZ

Approximating SVP with (at least) polynomial factor

GN-slide-reduction: A basis B of rank $n = pk \geq 2k$ is (ε, k)-slide-reduced if

1. Primal conditions: for all $i \in [0; p-1]$, $B_{[ik+1,ik+k]}$ is HKZ-reduced.
2. Dual condition: for all $i \in [0; p-2]$, $B_{[ik+2,ik+k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.

$$\Rightarrow \quad \|\mathbf{b}_1\| \leq ((1+\varepsilon)\gamma_k)^{\frac{n-k}{k-1}} \cdot \lambda_1(L). \tag{1}$$

- Goal: Extend GN-slide-reduction into the case $n = pk + q \geq 2k$ with $0 \leq q < k$ s.t. Eq. (1) still holds.
- Idea: Wrap "the extra q vectors" and "its nearby k vectors" into a bigger block of size $k + q$.
- Issue: With SVP-oracle in rank k, how to efficiently find a basis $(\mathbf{c}_1, \ldots, \mathbf{c}_m)$ for any lattice Λ of rank $m \in [k, 2k]$ s.t. $\|\mathbf{c}_1\| \lesssim \gamma_k^{\frac{m-1}{2(k-1)}} \cdot \mathrm{vol}(\Lambda)^{1/m}$? $\Leftarrow$ DBKZ

Approximating SVP with (at least) polynomial factor

GN-slide-reduction: A basis B of rank $n = pk \geq 2k$ is (ε, k)-slide-reduced if

1. Primal conditions: for all $i \in [0; p-1]$, $B_{[ik+1, ik+k]}$ is HKZ-reduced.
2. Dual condition: for all $i \in [0; p-2]$, $B_{[ik+2, ik+k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.

$$\Rightarrow \|\mathbf{b}_1\| \leq ((1+\varepsilon)\gamma_k)^{\frac{n-k}{k-1}} \cdot \lambda_1(L). \quad (1)$$

- Goal: Extend GN-slide-reduction into the case $n = pk + q \geq 2k$ with $0 \leq q < k$ s.t. Eq. (1) still holds.
- Idea: Wrap "the extra q vectors" and "its nearby k vectors" into a bigger block of size $k+q$.
- Issue: With SVP-oracle in rank k, how to efficiently find a basis $(\mathbf{c}_1, \ldots, \mathbf{c}_m)$ for any lattice Λ of rank $m \in [k, 2k]$ s.t. $\|\mathbf{c}_1\| \lesssim \gamma_k^{\frac{m-1}{2(k-1)}} \cdot \mathrm{vol}(\Lambda)^{1/m}$? $\Leftarrow$ DBKZ

Approximating SVP with (at least) polynomial factor

GN-slide-reduction: A basis B of rank $n = pk \geq 2k$ is (ε, k)-slide-reduced if

1. Primal conditions: for all $i \in [0; p-1]$, $B_{[ik+1, ik+k]}$ is HKZ-reduced.
2. Dual condition: for all $i \in [0; p-2]$, $B_{[ik+2, ik+k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.

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

- Goal: Extend GN-slide-reduction into the case $n = pk + q \geq 2k$ with $0 \leq q < k$ s.t. Eq. (1) still holds.
- Idea: Wrap "the extra q vectors" and "its nearby k vectors" into a bigger block of size $k + q$.
- Issue: With SVP-oracle in rank k, how to efficiently find a basis $(\mathbf{c}_1, \ldots, \mathbf{c}_m)$ for any lattice Λ of rank $m \in [k, 2k]$ s.t. $\|\mathbf{c}_1\| \lesssim \gamma_k^{\frac{m-1}{2(k-1)}} \cdot \mathrm{vol}(\Lambda)^{1/m}$? $\Leftarrow$ DBKZ

Approximating SVP with (at least) polynomial factor

Algorithm 10 The Micciancio-Walter DBKZ algorithm

Input: Block size $k \geq 2$, Integer N, a basis $B = (\mathbf{b}_1, \cdots, \mathbf{b}_n)$, and an SVP oracle in rank k.

Output: A new basis of $L(B)$.

1: **for** $\ell = 1$ **to** N **do**

2: **for** $i = 1$ to $n - k$ **do** SVP-reduce $B_{[i,i+k-1]}$

3: **for** $j = n - k + 1$ to 1 **do** DSVP-reduce $B_{[j,j+k-1]}$

4: **end for**

5: SVP-reduce $B_{[1,k]}$.

6: **return** B.

★ Th: With $N = \mathrm{poly}(B_{\mathrm{input}}, 1/\varepsilon)$, the DBKZ algorithm outputs a basis $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ s.t.

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

Approximating SVP with (at least) polynomial factor

Algorithm 11 The Micciancio-Walter DBKZ algorithm

Input: Block size $k \geq 2$, Integer N, a basis $B = (\mathbf{b}_1, \cdots, \mathbf{b}_n)$, and an SVP oracle in rank k.

Output: A new basis of $L(B)$.

1: **for** $\ell = 1$ **to** N **do**
2: **for** $i = 1$ to $n - k$ **do** SVP-reduce $B_{[i,i+k-1]}$
3: **for** $j = n - k + 1$ to 1 **do** DSVP-reduce $B_{[j,j+k-1]}$
4: **end for**
5: SVP-reduce $B_{[1,k]}$.
6: **return** B.

★ Th: With $N = \mathrm{poly}(B_{\text{input}}, 1/\varepsilon)$, the DBKZ algorithm outputs a basis $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ s.t.

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

Approximating SVP with (at least) polynomial factor

Algorithm 12 The Micciancio-Walter DBKZ algorithm

Input: Block size $k \geq 2$, Integer N, a basis $B = (\mathbf{b}_1, \cdots, \mathbf{b}_n)$, and an SVP oracle in rank k.

Output: A new basis of $L(B)$.

1: **for** $\ell = 1$ **to** N **do**

2: **for** $i = 1$ to $n - k$ **do** SVP-reduce $B_{[i,i+k-1]}$

3: **for** $j = n - k + 1$ to 1 **do** DSVP-reduce $B_{[j,j+k-1]}$

4: **end for**

5: SVP-reduce $B_{[1,k]}$.

6: **return** B.

★ Th: With $N = \mathrm{poly}(B_{\text{input}}, 1/\varepsilon)$, the DBKZ algorithm outputs a basis $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ s.t.

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

Approximating SVP with (at least) polynomial factor

Algorithm 13 The Micciancio-Walter DBKZ algorithm

Input: Block size $k \geq 2$, Integer N, a basis $B = (\mathbf{b}_1, \cdots, \mathbf{b}_n)$, and an SVP oracle in rank k.

Output: A new basis of $L(B)$.

1: **for** $\ell = 1$ **to** N **do**

2: **for** $i = 1$ to $n - k$ **do** SVP-reduce $B_{[i,i+k-1]}$

3: **for** $j = n - k + 1$ to 1 **do** DSVP-reduce $B_{[j,j+k-1]}$

4: **end for**

5: SVP-reduce $B_{[1,k]}$.

6: **return** B.

★ Th: With $N = \mathrm{poly}(B_{\mathrm{input}}, 1/\varepsilon)$, the DBKZ algorithm outputs a basis $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ s.t.

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

Approximating SVP with (at least) polynomial factor

Algorithm 14 The Micciancio-Walter DBKZ algorithm

Input: Block size $k \geq 2$, Integer N, a basis $B = (\mathbf{b}_1, \cdots, \mathbf{b}_n)$, and an SVP oracle in rank k.

Output: A new basis of $L(B)$.

1: **for** $\ell = 1$ **to** N **do**

2: **for** $i = 1$ to $n - k$ **do** SVP-reduce $B_{[i,i+k-1]}$

3: **for** $j = n - k + 1$ to 1 **do** DSVP-reduce $B_{[j,j+k-1]}$

4: **end for**

5: SVP-reduce $B_{[1,k]}$.

6: **return** B.

★ Th: With $N = \mathrm{poly}(B_{\mathrm{input}}, 1/\varepsilon)$, the DBKZ algorithm outputs a basis $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ s.t.

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

Approximating SVP with (at least) polynomial factor

Algorithm 15 The Micciancio-Walter DBKZ algorithm

Input: Block size $k \geq 2$, Integer N, a basis $B = (\mathbf{b}_1, \cdots, \mathbf{b}_n)$, and an SVP oracle in rank k.

Output: A new basis of $L(B)$.

1: **for** $\ell = 1$ **to** N **do**

2: **for** $i = 1$ to $n - k$ **do** SVP-reduce $B_{[i,i+k-1]}$

3: **for** $j = n - k + 1$ to 1 **do** DSVP-reduce $B_{[j,j+k-1]}$

4: **end for**

5: SVP-reduce $B_{[1,k]}$.

6: **return** B.

★ Th: With $N = \mathrm{poly}(B_{\mathrm{input}}, 1/\varepsilon)$, the DBKZ algorithm outputs a basis $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ s.t.

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

Approximating SVP with (at least) polynomial factor

Algorithm 16 The Micciancio-Walter DBKZ algorithm

Input: Block size $k \geq 2$, Integer N, a basis $B = (\mathbf{b}_1, \cdots, \mathbf{b}_n)$, and an SVP oracle in rank k.

Output: A new basis of $L(B)$.

1: **for** $\ell = 1$ **to** N **do**
2:     **for** $i = 1$ to $n - k$ **do** SVP-reduce $B_{[i,i+k-1]}$
3:     **for** $j = n - k + 1$ to 1 **do** DSVP-reduce $B_{[j,j+k-1]}$
4: **end for**
5: SVP-reduce $B_{[1,k]}$.
6: **return** B.

★ Th: With $N = \mathrm{poly}(B_{\mathrm{input}}, 1/\varepsilon)$, the DBKZ algorithm outputs a basis $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ s.t.

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

Approximating SVP with (at least) polynomial factor

Algorithm 17 The Micciancio-Walter DBKZ algorithm

Input: Block size $k \geq 2$, Integer N, a basis $B = (\mathbf{b}_1, \cdots, \mathbf{b}_n)$, and an SVP oracle in rank k.

Output: A new basis of $L(B)$.

1: **for** $\ell = 1$ **to** N **do**
2: **for** $i = 1$ to $n - k$ **do** SVP-reduce $B_{[i,i+k-1]}$
3: **for** $j = n - k + 1$ to 1 **do** DSVP-reduce $B_{[j,j+k-1]}$
4: **end for**
5: SVP-reduce $B_{[1,k]}$.
6: **return** B.

★ Th: With $N = \mathrm{poly}(B_{\mathrm{input}}, 1/\varepsilon)$, the DBKZ algorithm outputs a basis $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ s.t.

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

Approximating SVP with (at least) polynomial factor

Observation: Twin in both DBKZ and GN-slide-reduction.

Algorithm 18 DBKZ with $n = k + 1$

1: **for** $\ell = 1$ **to** N **do**
2: SVP-reduce $B_{[1,k]}$
3: DSVP-reduce $B_{[2,k+1]}$
4: **end for**
5: δ-SVP-reduce $B_{[1,k]}$.
6: **return** B.

VS

$B_{[ik,ik+k+1]}$ in GN-slide-reduction: A basis B of rank $n = pk \geq 2k$ is (ε, k)-slide-reduced if

1. Primal conditions: for all $i \in [0; p-1]$, $B_{[ik+1,ik+k]}$ is HKZ-reduced.
2. Dual condition: for all $i \in [0; p-2]$, $B_{[ik+2,ik+k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.

Approximating SVP with (at least) polynomial factor

Observation: Twin in both DBKZ and GN-slide-reduction.

Algorithm 19 DBKZ with $n = k + 1$

for $\ell = 1$ **to** N **do**
    SVP-reduce $B_{[1,k]}$
    DSVP-reduce $B_{[2,k+1]}$
end for
δ-SVP-reduce $B_{[1,k]}$.
return B.

VS

$B_{[ik,ik+k+1]}$ in GN-slide-reduction: A basis B of rank $n = pk \geq 2k$ is (ε, k)-slide-reduced if

1. Primal conditions: for all $i \in [0; p-1]$, $B_{[ik+1,ik+k]}$ is HKZ-reduced.
2. Dual condition: for all $i \in [0; p-2]$, $B_{[ik+2,ik+k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.

Approximating SVP with (at least) polynomial factor

Observation: Twin in both DBKZ and GN-slide-reduction.

Algorithm 20 DBKZ with $n = k + 1$

1: **for** $\ell = 1$ **to** N **do**
2: SVP-reduce $B_{[1,k]}$
3: DSVP-reduce $B_{[2,k+1]}$
4: **end for**
5: δ-SVP-reduce $B_{[1,k]}$.
6: **return** B.

VS

$B_{[ik,ik+k+1]}$ in GN-slide-reduction: A basis B of rank $n = pk \geq 2k$ is (ε, k)-slide-reduced if

1. Primal conditions: for all $i \in [0; p-1]$, $B_{[ik+1,ik+k]}$ is HKZ-reduced.
2. Dual condition: for all $i \in [0; p-2]$, $B_{[ik+2,ik+k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.

Approximating SVP with (at least) polynomial factor

Observation: Twin in both DBKZ and GN-slide-reduction.

Algorithm 21 DBKZ with $n = k + 1$

1: **for** $\ell = 1$ **to** N **do**
2: SVP-reduce $B_{[1,k]}$
3: DSVP-reduce $B_{[2,k+1]}$
4: **end for**
5: δ-SVP-reduce $B_{[1,k]}$.
6: **return** B.

VS

$B_{[ik,ik+k+1]}$ in GN-slide-reduction: A basis B of rank $n = pk \geq 2k$ is (ε, k)-slide-reduced if

1. Primal conditions: for all $i \in [0; p-1]$, $B_{[ik+1,ik+k]}$ is HKZ-reduced.
2. Dual condition: for all $i \in [0; p-2]$, $B_{[ik+2,ik+k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.

Approximating SVP with (at least) polynomial factor

Observation: Twin in both DBKZ and GN-slide-reduction.

Algorithm 22 DBKZ with $n = k + 1$

1: **for** $\ell = 1$ **to** N **do**
2: SVP-reduce $B_{[1,k]}$
3: DSVP-reduce $B_{[2,k+1]}$
4: **end for**
5: δ-SVP-reduce $B_{[1,k]}$.
6: **return** B.

VS

$B_{[ik,ik+k+1]}$ in GN-slide-reduction: A basis B of rank $n = pk \geq 2k$ is (ε, k)-slide-reduced if

1. Primal conditions: for all $i \in [0; p-1]$, $B_{[ik+1,ik+k]}$ is HKZ-reduced.
2. Dual condition: for all $i \in [0; p-2]$, $B_{[ik+2,ik+k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.

Approximating SVP with (at least) polynomial factor

Observation: Twin in both DBKZ and GN-slide-reduction.

Algorithm 23 DBKZ with $n = k + 1$

1: **for** $\ell = 1$ **to** N **do**
2: SVP-reduce $B_{[1,k]}$
3: DSVP-reduce $B_{[2,k+1]}$
4: **end for**
5: δ-SVP-reduce $B_{[1,k]}$.
6: **return** B.

VS

$B_{[ik,ik+k+1]}$ in GN-slide-reduction: A basis B of rank $n = pk \geq 2k$ is (ε, k)-slide-reduced if

1. Primal conditions: for all $i \in [0; p-1]$, $B_{[ik+1,ik+k]}$ is HKZ-reduced.
2. Dual condition: for all $i \in [0; p-2]$, $B_{[ik+2,ik+k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.

Approximating SVP with (at least) polynomial factor

Observation: Twin in both DBKZ and GN-slide-reduction.

Algorithm 24 DBKZ with $n = k + 1$

1: **for** $\ell = 1$ **to** N **do**
2: SVP-reduce $B_{[1,k]}$
3: DSVP-reduce $B_{[2,k+1]}$
4: **end for**
5: δ-SVP-reduce $B_{[1,k]}$.
6: **return** B.

VS

$B_{[ik,ik+k+1]}$ in GN-slide-reduction: A basis B of rank $n = pk \geq 2k$ is (ε, k)-slide-reduced if

1. Primal conditions: for all $i \in [0; p-1]$, $B_{[ik+1,ik+k]}$ is HKZ-reduced.
2. Dual condition: for all $i \in [0; p-2]$, $B_{[ik+2,ik+k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.

Approximating SVP with (at least) polynomial factor

Observation: Twin in both DBKZ and GN-slide-reduction.

Algorithm 25 DBKZ with $n = k + 1$

1: **for** $\ell = 1$ **to** N **do**
2: SVP-reduce $B_{[1,k]}$
3: DSVP-reduce $B_{[2,k+1]}$
4: **end for**
5: δ-SVP-reduce $B_{[1,k]}$.
6: **return** B.

VS

$B_{[ik,ik+k+1]}$ in GN-slide-reduction: A basis B of rank $n = pk \geq 2k$ is (ε, k)-slide-reduced if

1. Primal conditions: for all $i \in [0; p-1]$, $B_{[ik+1,ik+k]}$ is HKZ-reduced.
2. Dual condition: for all $i \in [0; p-2]$, $B_{[ik+2,ik+k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.

Approximating SVP with (at least) polynomial factor

- Formalization: A basis B of rank $d+1$ is f-twin-reduced if $B_{[1,d]}$ is f-HSVP-reduced and $B_{[2,d+1]}$ is f-DHSVP-reduced.
- Fact: If $B = (\mathbf{b}_1, \ldots, \mathbf{b}_{d+1})$ is f-twin-reduced, then
$$\|\mathbf{b}_1\| \leq f^{2d/(d-1)} \|\mathbf{b}^*_{d+1}\|.$$
Further, $f^{-d/(d-1)}\|\mathbf{b}_1\| \leq \mathrm{vol}(B)^{1/(d+1)} \leq f^{d/(d-1)}\|\mathbf{b}^*_{d+1}\|$.
- Instantiation: Every block $B_{[ik+1,jk+1]}$ for any $i < j$ of a GN-slide-reduced basis is $\gamma_k^{\frac{(j-i)k-1}{2(k-1)}}$-twin-reduced.

Approximating SVP with (at least) polynomial factor

- Formalization: A basis B of rank $d+1$ is f-twin-reduced if $B_{[1,d]}$ is f-HSVP-reduced and $B_{[2,d+1]}$ is f-DHSVP-reduced.
- Fact: If $B = (\mathbf{b}_1, \ldots, \mathbf{b}_{d+1})$ is f-twin-reduced, then

$$\|\mathbf{b}_1\| \leq f^{2d/(d-1)} \|\mathbf{b}^*_{d+1}\|.$$

  Further, $f^{-d/(d-1)}\|\mathbf{b}_1\| \leq \mathrm{vol}(B)^{1/(d+1)} \leq f^{d/(d-1)}\|\mathbf{b}^*_{d+1}\|$.
- Instantiation: Every block $B_{[ik+1,jk+1]}$ for any $i < j$ of a GN-slide-reduced basis is $\gamma_k^{\frac{(j-i)k-1}{2(k-1)}}$-twin-reduced.

Approximating SVP with (at least) polynomial factor

- Formalization: A basis B of rank $d+1$ is f-twin-reduced if $B_{[1,d]}$ is f-HSVP-reduced and $B_{[2,d+1]}$ is f-DHSVP-reduced.
- Fact: If $B = (\mathbf{b}_1, \dots, \mathbf{b}_{d+1})$ is f-twin-reduced, then

$$\|\mathbf{b}_1\| \leq f^{2d/(d-1)} \|\mathbf{b}^*_{d+1}\|.$$

  Further, $f^{-d/(d-1)} \|\mathbf{b}_1\| \leq \mathrm{vol}(B)^{1/(d+1)} \leq f^{d/(d-1)} \|\mathbf{b}^*_{d+1}\|$.
- Instantiation: Every block $B_{[ik+1,jk+1]}$ for any $i < j$ of a GN-slide-reduced basis is $\gamma_k^{\frac{(j-i)k-1}{2(k-1)}}$-twin-reduced.

Approximating SVP with (at least) polynomial factor

GN-slide-reduction in case $n = pk$: A basis B of rank $n = pk \geq 2k$ is (ε, k)-slide-reduced if

1. Primal conditions: for all $i \in [0; p-1]$, $B_{[ik+1, ik+k]}$ is HKZ-reduced.
2. Dual condition: for all $i \in [0; p-2]$, $B_{[ik+2, ik+k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.

Our variant of slide-reduction for $n \geq 2k$

Definition: Let $n = pk + q$ with $0 \leq q \leq k-1$ and $p, k \geq 2$. A basis B of rank n is *k-slide-reduced* if

- Twin condition: $B_{[1,k+q+1]}$ is $\gamma_k^{\frac{k+q-1}{2(k-1)}}$-twin-reduced;
- Primal conditions: for all $i = 1, \ldots, k$, $B_{[ik+q+1,(i+1)k+q]}$ is SVP-reduced;
- Dual condition: for all $i \in [1, p-2]$, $B_{[ik+q+2,(i+1)k+q+1]}$ is $\sqrt{\gamma_k}$-DHSVP-reduced.

Approximating SVP with (at least) polynomial factor

GN-slide-reduction in case $n = pk$: A basis B of rank $n = pk \geq 2k$ is (ε, k)-slide-reduced if

1. Primal conditions: for all $i \in [0; p-1]$, $B_{[ik+1, ik+k]}$ is HKZ-reduced.
2. Dual condition: for all $i \in [0; p-2]$, $B_{[ik+2, ik+k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.

Our variant of slide-reduction for $n \geq 2k$

Definition: Let $n = pk + q$ with $0 \leq q \leq k-1$ and $p, k \geq 2$. A basis B of rank n is *k-slide-reduced* if

- Twin condition: $B_{[1,k+q+1]}$ is $\gamma_k^{\frac{k+q-1}{2(k-1)}}$-twin-reduced;
- Primal conditions: for all $i = 1, \ldots, k$, $B_{[ik+q+1,(i+1)k+q]}$ is SVP-reduced;
- Dual condition: for all $i \in [1, p-2]$, $B_{[ik+q+2,(i+1)k+q+1]}$ is $\sqrt{\gamma_k}$-DHSVP-reduced.

Approximating SVP with (at least) polynomial factor

GN-slide-reduction in case $n = pk$: A basis B of rank $n = pk \geq 2k$ is (ε, k)-slide-reduced if

1. Primal conditions: for all $i \in [0; p-1]$, $B_{[ik+1, ik+k]}$ is HKZ-reduced.
2. Dual condition: for all $i \in [0; p-2]$, $B_{[ik+2, ik+k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.

Our variant of slide-reduction for $n \geq 2k$

Definition: Let $n = pk + q$ with $0 \leq q \leq k-1$ and $p, k \geq 2$. A basis B of rank n is *k-slide-reduced* if

- Twin condition: $B_{[1, k+q+1]}$ is $\gamma_k^{\frac{k+q-1}{2(k-1)}}$-twin-reduced;
- Primal conditions: for all $i = 1, \ldots, k$, $B_{[ik+q+1, (i+1)k+q]}$ is SVP-reduced;
- Dual condition: for all $i \in [1, p-2]$, $B_{[ik+q+2, (i+1)k+q+1]}$ is $\sqrt{\gamma_k}$-DHSVP-reduced.

Approximating SVP with (at least) polynomial factor

GN-slide-reduction in case $n = pk$: A basis B of rank $n = pk \geq 2k$ is (ε, k)-slide-reduced if

1. Primal conditions: for all $i \in [0; p-1]$, $B_{[ik+1,ik+k]}$ is HKZ-reduced.
2. Dual condition: for all $i \in [0; p-2]$, $B_{[ik+2,ik+k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.

Our variant of slide-reduction for $n \geq 2k$

Definition: Let $n = pk + q$ with $0 \leq q \leq k-1$ and $p, k \geq 2$. A basis B of rank n is *k-slide-reduced* if

- Twin condition: $B_{[1,k+q+1]}$ is $\gamma_k^{\frac{k+q-1}{2(k-1)}}$-twin-reduced;
- Primal conditions: for all $i = 1, \ldots, k$, $B_{[ik+q+1,(i+1)k+q]}$ is SVP-reduced;
- Dual condition: for all $i \in [1, p-2]$, $B_{[ik+q+2,(i+1)k+q+1]}$ is $\sqrt{\gamma_k}$-DHSVP-reduced.

Approximating SVP with (at least) polynomial factor

GN-slide-reduction in case $n = pk$: A basis B of rank $n = pk \geq 2k$ is (ε, k)-slide-reduced if

1. Primal conditions: for all $i \in [0; p-1]$, $B_{[ik+1, ik+k]}$ is HKZ-reduced.
2. Dual condition: for all $i \in [0; p-2]$, $B_{[ik+2, ik+k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.

Our variant of slide-reduction for $n \geq 2k$

Definition: Let $n = pk + q$ with $0 \leq q \leq k-1$ and $p, k \geq 2$. A basis B of rank n is *k-slide-reduced* if

- Twin condition: $B_{[1, k+q+1]}$ is $\gamma_k^{\frac{k+q-1}{2(k-1)}}$-twin-reduced;
- Primal conditions: for all $i = 1, \ldots, k$, $B_{[ik+q+1, (i+1)k+q]}$ is SVP-reduced;
- Dual condition: for all $i \in [1, p-2]$, $B_{[ik+q+2, (i+1)k+q+1]}$ is $\sqrt{\gamma_k}$-DHSVP-reduced.

Approximating SVP with (at least) polynomial factor

GN-slide-reduction in case $n = pk$: A basis B of rank $n = pk \geq 2k$ is (ε, k)-slide-reduced if

1. Primal conditions: for all $i \in [0; p-1]$, $B_{[ik+1, ik+k]}$ is HKZ-reduced.
2. Dual condition: for all $i \in [0; p-2]$, $B_{[ik+2, ik+k+1]}$ is $(1+\varepsilon)$-DSVP-reduced.

Our variant of slide-reduction for $n \geq 2k$

Definition: Let $n = pk + q$ with $0 \leq q \leq k-1$ and $p, k \geq 2$. A basis B of rank n is *k-slide-reduced* if

- Twin condition: $B_{[1, k+q+1]}$ is $\gamma_k^{\frac{k+q-1}{2(k-1)}}$-twin-reduced;
- Primal conditions: for all $i = 1, \ldots, k$, $B_{[ik+q+1, (i+1)k+q]}$ is SVP-reduced;
- Dual condition: for all $i \in [1, p-2]$, $B_{[ik+q+2, (i+1)k+q+1]}$ is $\sqrt{\gamma_k}$-DHSVP-reduced.

Approximating SVP with (at least) polynomial factor

Our variant of slide-reduction for $n \geq 2k$

Let $n = pk + q$ with $0 \leq q \leq k - 1$ and $p, k \geq 2$.

- Intuition: A basis B of rank n is *k-slide-reduced* if $B_{[1,k+q+1]}$ is $\gamma_k^{\frac{k+q-1}{2(k-1)}}$-twin-reduced and $B_{[k+q+1,n]}$ is k-GN-slide-reduced;
- Property: Let $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ be a k-slide-reduced basis of a lattice L. Then

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

Further, if either $\lambda_1(L(B_{[1,k+q]})) > \lambda_1(L)$ or $B_{[1,k+q]}$ is $\gamma_k^{\frac{n-k}{k-1}}$-SVP-reduced, then

$$\|\mathbf{b}_1\| \leq \gamma_k^{\frac{n-k}{k-1}} \lambda_1(L).$$

Approximating SVP with (at least) polynomial factor

Our variant of slide-reduction for $n \geq 2k$

Let $n = pk + q$ with $0 \leq q \leq k-1$ and $p, k \geq 2$.

- Intuition: A basis B of rank n is *k-slide-reduced* if $B_{[1,k+q+1]}$ is $\gamma_k^{\frac{k+q-1}{2(k-1)}}$-twin-reduced and $B_{[k+q+1,n]}$ is k-GN-slide-reduced;
- Property: Let $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ be a k-slide-reduced basis of a lattice L. Then

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

Further, if either $\lambda_1(L(B_{[1,k+q]})) > \lambda_1(L)$ or $B_{[1,k+q]}$ is $\gamma_k^{\frac{n-k}{k-1}}$-SVP-reduced, then

$$\|\mathbf{b}_1\| \leq \gamma_k^{\frac{n-k}{k-1}} \lambda_1(L).$$

Approximating SVP with (at least) polynomial factor

Our variant of slide-reduction for $n \geq 2k$

Let $n = pk + q$ with $0 \leq q \leq k - 1$ and $p, k \geq 2$.

- Intuition: A basis B of rank n is *k-slide-reduced* if $B_{[1,k+q+1]}$ is $\gamma_k^{\frac{k+q-1}{2(k-1)}}$-twin-reduced and $B_{[k+q+1,n]}$ is k-GN-slide-reduced;
- Property: Let $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ be a k-slide-reduced basis of a lattice L. Then

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

  Further, if either $\lambda_1(L(B_{[1,k+q]})) > \lambda_1(L)$ or $B_{[1,k+q]}$ is $\gamma_k^{\frac{n-k}{k-1}}$-SVP-reduced, then

$$\|\mathbf{b}_1\| \leq \gamma_k^{\frac{n-k}{k-1}} \lambda_1(L).$$

Algorithm 26 The slide reduction algorithm for $n \geq 2k$

Input: Blocksize k, termination factor ε, a basis B of rank $n = pk + q$ where $0 \leq q < k$, and an SVP-oracle in rank k.

Output: An (almost) k-slide-reduced basis of $L(B)$.

while $\mathrm{vol}(B_{[1,ik+q]})$ is modified for some $i \in [1, p-1]$ **do**

  $(1+\varepsilon)\eta$-HSVP-reduce $B_{[1,k+q]}$ using DBKZ for $\eta := \gamma_k^{\frac{k+q-1}{2(k-1)}}$

  for $i = 1$ to $p-1$ **do** SVP-reduce $B_{[ik+q+1,(i+1)k+q]}$

  if $B_{[2,k+q+1]}$ is not $(1+\varepsilon)\eta$-DHSVP-reduced **then** $\sqrt{1+\varepsilon}\eta$-DHSVP-reduce $B_{[2,k+q+1]}$ using DBKZ

  if $B_{[ik+q+2,(i+1)k+q+1]}$ is not $\sqrt{(1+\varepsilon)\gamma_k}$-DHSVP-reduced for some $i \in [1, p-2]$ **then** $\sqrt{\gamma_k}$-DHSVP-reduce $B_{[ik+q+2,(i+1)k+q+1]}$

end while

Find a $\gamma_k^{\frac{n-k}{k-1}}$-SVP-reduced basis $C = (\mathbf{c}_1, \ldots, \mathbf{c}_{k+q})$ for the sublattice $B_{[1,k+q]}$ using our first algorithm

if $\|\mathbf{c}_1\| < \|\mathbf{b}_1\|$ **then** $B_{[1,k+q]} \leftarrow C$

return B.

Algorithm 27 The slide reduction algorithm for $n \geq 2k$

Input: Blocksize k, termination factor ε, a basis B of rank $n = pk + q$ where $0 \leq q < k$, and an SVP-oracle in rank k.

Output: An (almost) k-slide-reduced basis of $L(B)$.

while $\mathrm{vol}(B_{[1,ik+q]})$ is modified for some $i \in [1, p-1]$ **do**
  $(1+\varepsilon)\eta$-HSVP-reduce $B_{[1,k+q]}$ using DBKZ for $\eta := \gamma_k^{\frac{k+q-1}{2(k-1)}}$
  for $i = 1$ to $p-1$ **do** SVP-reduce $B_{[ik+q+1,(i+1)k+q]}$
  if $B_{[2,k+q+1]}$ is not $(1+\varepsilon)\eta$-DHSVP-reduced **then** $\sqrt{1+\varepsilon}\eta$-DHSVP-reduce $B_{[2,k+q+1]}$ using DBKZ
  if $B_{[ik+q+2,(i+1)k+q+1]}$ is not $\sqrt{(1+\varepsilon)\gamma_k}$-DHSVP-reduced for some $i \in [1, p-2]$ **then** $\sqrt{\gamma_k}$-DHSVP-reduce $B_{[ik+q+2,(i+1)k+q+1]}$
end while
Find a $\gamma_k^{\frac{n-k}{k-1}}$-SVP-reduced basis $C = (\mathbf{c}_1, \ldots, \mathbf{c}_{k+q})$ for the sublattice $B_{[1,k+q]}$ using our first algorithm
if $\|\mathbf{c}_1\| < \|\mathbf{b}_1\|$ **then** $B_{[1,k+q]} \leftarrow C$
return B.

Algorithm 28 The slide reduction algorithm for $n \geq 2k$

Input: Blocksize k, termination factor ε, a basis B of rank $n = pk + q$ where $0 \leq q < k$, and an SVP-oracle in rank k.

Output: An (almost) k-slide-reduced basis of $L(B)$.

while $\mathrm{vol}(B_{[1,ik+q]})$ is modified for some $i \in [1, p-1]$ **do**

  $(1+\varepsilon)\eta$-HSVP-reduce $B_{[1,k+q]}$ using DBKZ for $\eta := \gamma_k^{\frac{k+q-1}{2(k-1)}}$

  for $i = 1$ to $p-1$ **do** SVP-reduce $B_{[ik+q+1,(i+1)k+q]}$

  if $B_{[2,k+q+1]}$ is not $(1+\varepsilon)\eta$-DHSVP-reduced **then** $\sqrt{1+\varepsilon}\eta$-DHSVP-reduce $B_{[2,k+q+1]}$ using DBKZ

  if $B_{[ik+q+2,(i+1)k+q+1]}$ is not $\sqrt{(1+\varepsilon)\gamma_k}$-DHSVP-reduced for some $i \in [1, p-2]$ **then** $\sqrt{\gamma_k}$-DHSVP-reduce $B_{[ik+q+2,(i+1)k+q+1]}$

end while

Find a $\gamma_k^{\frac{n-k}{k-1}}$-SVP-reduced basis $C = (\mathbf{c}_1, \ldots, \mathbf{c}_{k+q})$ for the sublattice $B_{[1,k+q]}$ using our first algorithm

if $\|\mathbf{c}_1\| < \|\mathbf{b}_1\|$ **then** $B_{[1,k+q]} \leftarrow C$

return B.

Algorithm 29 The slide reduction algorithm for $n \geq 2k$

Input: Blocksize k, termination factor ε, a basis B of rank $n = pk + q$ where $0 \leq q < k$, and an SVP-oracle in rank k.

Output: An (almost) k-slide-reduced basis of $L(B)$.

while $\mathrm{vol}(B_{[1,ik+q]})$ is modified for some $i \in [1, p-1]$ **do**
  $(1+\varepsilon)\eta$-HSVP-reduce $B_{[1,k+q]}$ using DBKZ for $\eta := \gamma_k^{\frac{k+q-1}{2(k-1)}}$
  for $i = 1$ to $p-1$ **do** SVP-reduce $B_{[ik+q+1,(i+1)k+q]}$
  if $B_{[2,k+q+1]}$ is not $(1+\varepsilon)\eta$-DHSVP-reduced **then** $\sqrt{1+\varepsilon}\eta$-DHSVP-reduce $B_{[2,k+q+1]}$ using DBKZ
  if $B_{[ik+q+2,(i+1)k+q+1]}$ is not $\sqrt{(1+\varepsilon)\gamma_k}$-DHSVP-reduced for some $i \in [1, p-2]$ **then** $\sqrt{\gamma_k}$-DHSVP-reduce $B_{[ik+q+2,(i+1)k+q+1]}$
end while
Find a $\gamma_k^{\frac{n-k}{k-1}}$-SVP-reduced basis $C = (\mathbf{c}_1, \ldots, \mathbf{c}_{k+q})$ for the sublattice $B_{[1,k+q]}$ using our first algorithm
if $\|\mathbf{c}_1\| < \|\mathbf{b}_1\|$ **then** $B_{[1,k+q]} \leftarrow C$
return B.

Algorithm 30 The slide reduction algorithm for $n \geq 2k$

Input: Blocksize k, termination factor ε, a basis B of rank $n = pk + q$ where $0 \leq q < k$, and an SVP-oracle in rank k.

Output: An (almost) k-slide-reduced basis of $L(B)$.

1: **while** $\mathrm{vol}(B_{[1,ik+q]})$ is modified for some $i \in [1, p-1]$ **do**

2: $(1+\varepsilon)\eta$-HSVP-reduce $B_{[1,k+q]}$ using DBKZ for $\eta := \gamma_k^{\frac{k+q-1}{2(k-1)}}$

3: **for** $i = 1$ to $p-1$ **do** SVP-reduce $B_{[ik+q+1,(i+1)k+q]}$

4: **if** $B_{[2,k+q+1]}$ is not $(1+\varepsilon)\eta$-DHSVP-reduced **then** $\sqrt{1+\varepsilon}\eta$-DHSVP-reduce $B_{[2,k+q+1]}$ using DBKZ

5: **if** $B_{[ik+q+2,(i+1)k+q+1]}$ is not $\sqrt{(1+\varepsilon)\gamma_k}$-DHSVP-reduced for some $i \in [1, p-2]$ **then** $\sqrt{\gamma_k}$-DHSVP-reduce $B_{[ik+q+2,(i+1)k+q+1]}$

6: **end while**

7: Find a $\gamma_k^{\frac{n-k}{k-1}}$-SVP-reduced basis $C = (\mathbf{c}_1, \ldots, \mathbf{c}_{k+q})$ for the sublattice $B_{[1,k+q]}$ using our first algorithm

8: **if** $\|\mathbf{c}_1\| < \|\mathbf{b}_1\|$ **then** $B_{[1,k+q]} \leftarrow C$

9: **return** B.

Algorithm 31 The slide reduction algorithm for $n \geq 2k$

Input: Blocksize k, termination factor ε, a basis B of rank $n = pk + q$ where $0 \leq q < k$, and an SVP-oracle in rank k.

Output: An (almost) k-slide-reduced basis of $L(B)$.

1: **while** $\mathrm{vol}(B_{[1,ik+q]})$ is modified for some $i \in [1, p-1]$ **do**

2: $(1+\varepsilon)\eta$-HSVP-reduce $B_{[1,k+q]}$ using DBKZ for $\eta := \gamma_k^{\frac{k+q-1}{2(k-1)}}$

3: **for** $i = 1$ to $p-1$ **do** SVP-reduce $B_{[ik+q+1,(i+1)k+q]}$

4: **if** $B_{[2,k+q+1]}$ is not $(1+\varepsilon)\eta$-DHSVP-reduced **then** $\sqrt{1+\varepsilon}\eta$-DHSVP-reduce $B_{[2,k+q+1]}$ using DBKZ

5: **if** $B_{[ik+q+2,(i+1)k+q+1]}$ is not $\sqrt{(1+\varepsilon)\gamma_k}$-DHSVP-reduced for some $i \in [1, p-2]$ **then** $\sqrt{\gamma_k}$-DHSVP-reduce $B_{[ik+q+2,(i+1)k+q+1]}$

6: **end while**

7: Find a $\gamma_k^{\frac{n-k}{k-1}}$-SVP-reduced basis $C = (\mathbf{c}_1, \ldots, \mathbf{c}_{k+q})$ for the sublattice $B_{[1,k+q]}$ using our first algorithm

8: **if** $\|\mathbf{c}_1\| < \|\mathbf{b}_1\|$ **then** $B_{[1,k+q]} \leftarrow C$

9: **return** B.

Algorithm 32 The slide reduction algorithm for $n \geq 2k$

Input: Blocksize k, termination factor ε, a basis B of rank $n = pk + q$ where $0 \leq q < k$, and an SVP-oracle in rank k.

Output: An (almost) k-slide-reduced basis of $L(B)$.

1: **while** $\mathrm{vol}(B_{[1,ik+q]})$ is modified for some $i \in [1, p-1]$ **do**

2: $(1+\varepsilon)\eta$-HSVP-reduce $B_{[1,k+q]}$ using DBKZ for $\eta := \gamma_k^{\frac{k+q-1}{2(k-1)}}$

3: **for** $i = 1$ to $p-1$ **do** SVP-reduce $B_{[ik+q+1,(i+1)k+q]}$

4: **if** $B_{[2,k+q+1]}$ is not $(1+\varepsilon)\eta$-DHSVP-reduced **then** $\sqrt{1+\varepsilon}\eta$-DHSVP-reduce $B_{[2,k+q+1]}$ using DBKZ

5: **if** $B_{[ik+q+2,(i+1)k+q+1]}$ is not $\sqrt{(1+\varepsilon)\gamma_k}$-DHSVP-reduced for some $i \in [1, p-2]$ **then** $\sqrt{\gamma_k}$-DHSVP-reduce $B_{[ik+q+2,(i+1)k+q+1]}$

6: **end while**

7: Find a $\gamma_k^{\frac{n-k}{k-1}}$-SVP-reduced basis $C = (\mathbf{c}_1, \ldots, \mathbf{c}_{k+q})$ for the sublattice $B_{[1,k+q]}$ using our first algorithm

8: **if** $\|\mathbf{c}_1\| < \|\mathbf{b}_1\|$ **then** $B_{[1,k+q]} \leftarrow C$

9: **return** B.

Algorithm 33 The slide reduction algorithm for $n \geq 2k$

Input: Blocksize k, termination factor ε, a basis B of rank $n = pk + q$ where $0 \leq q < k$, and an SVP-oracle in rank k.

Output: An (almost) k-slide-reduced basis of $L(B)$.

while $\mathrm{vol}(B_{[1,ik+q]})$ is modified for some $i \in [1, p-1]$ **do**
    $(1+\varepsilon)\eta$-HSVP-reduce $B_{[1,k+q]}$ using DBKZ for $\eta := \gamma_k^{\frac{k+q-1}{2(k-1)}}$
    for $i = 1$ to $p-1$ **do** SVP-reduce $B_{[ik+q+1,(i+1)k+q]}$
    if $B_{[2,k+q+1]}$ is not $(1+\varepsilon)\eta$-DHSVP-reduced **then** $\sqrt{1+\varepsilon}\eta$-DHSVP-reduce $B_{[2,k+q+1]}$ using DBKZ
    if $B_{[ik+q+2,(i+1)k+q+1]}$ is not $\sqrt{(1+\varepsilon)\gamma_k}$-DHSVP-reduced for some $i \in [1, p-2]$ **then** $\sqrt{\gamma_k}$-DHSVP-reduce $B_{[ik+q+2,(i+1)k+q+1]}$
end while
Find a $\gamma_k^{\frac{n-k}{k-1}}$-SVP-reduced basis $C = (\mathbf{c}_1, \ldots, \mathbf{c}_{k+q})$ for the sublattice $B_{[1,k+q]}$ using our first algorithm
if $\|\mathbf{c}_1\| < \|\mathbf{b}_1\|$ **then** $B_{[1,k+q]} \leftarrow C$
return B.

Algorithm 34 The slide reduction algorithm for $n \geq 2k$

Input: Blocksize k, termination factor ε, a basis B of rank $n = pk + q$ where $0 \leq q < k$, and an SVP-oracle in rank k.

Output: An (almost) k-slide-reduced basis of $L(B)$.

while $\mathrm{vol}(B_{[1,ik+q]})$ is modified for some $i \in [1, p-1]$ **do**
  $(1+\varepsilon)\eta$-HSVP-reduce $B_{[1,k+q]}$ using DBKZ for $\eta := \gamma_k^{\frac{k+q-1}{2(k-1)}}$
  for $i = 1$ to $p-1$ **do** SVP-reduce $B_{[ik+q+1,(i+1)k+q]}$
  if $B_{[2,k+q+1]}$ is not $(1+\varepsilon)\eta$-DHSVP-reduced **then** $\sqrt{1+\varepsilon}\eta$-DHSVP-reduce $B_{[2,k+q+1]}$ using DBKZ
  if $B_{[ik+q+2,(i+1)k+q+1]}$ is not $\sqrt{(1+\varepsilon)\gamma_k}$-DHSVP-reduced for some $i \in [1, p-2]$ **then** $\sqrt{\gamma_k}$-DHSVP-reduce $B_{[ik+q+2,(i+1)k+q+1]}$
end while
Find a $\gamma_k^{\frac{n-k}{k-1}}$-SVP-reduced basis $C = (\mathbf{c}_1, \ldots, \mathbf{c}_{k+q})$ for the sublattice $B_{[1,k+q]}$ using our first algorithm
if $\|\mathbf{c}_1\| < \|\mathbf{b}_1\|$ **then** $B_{[1,k+q]} \leftarrow C$
return B.

Algorithm 35 The slide reduction algorithm for $n \geq 2k$

Input: Blocksize k, termination factor ε, a basis B of rank $n = pk + q$ where $0 \leq q < k$, and an SVP-oracle in rank k.

Output: An (almost) k-slide-reduced basis of $L(B)$.

1: **while** $\mathrm{vol}(B_{[1,ik+q]})$ is modified for some $i \in [1, p-1]$ **do**

2: $(1+\varepsilon)\eta$-HSVP-reduce $B_{[1,k+q]}$ using DBKZ for $\eta := \gamma_k^{\frac{k+q-1}{2(k-1)}}$

3: **for** $i = 1$ to $p-1$ **do** SVP-reduce $B_{[ik+q+1,(i+1)k+q]}$

4: **if** $B_{[2,k+q+1]}$ is not $(1+\varepsilon)\eta$-DHSVP-reduced **then** $\sqrt{1+\varepsilon}\eta$-DHSVP-reduce $B_{[2,k+q+1]}$ using DBKZ

5: **if** $B_{[ik+q+2,(i+1)k+q+1]}$ is not $\sqrt{(1+\varepsilon)\gamma_k}$-DHSVP-reduced for some $i \in [1, p-2]$ **then** $\sqrt{\gamma_k}$-DHSVP-reduce $B_{[ik+q+2,(i+1)k+q+1]}$

6: **end while**

7: Find a $\gamma_k^{\frac{n-k}{k-1}}$-SVP-reduced basis $C = (\mathbf{c}_1, \ldots, \mathbf{c}_{k+q})$ for the sublattice $B_{[1,k+q]}$ using our first algorithm

8: **if** $\|\mathbf{c}_1\| < \|\mathbf{b}_1\|$ **then** $B_{[1,k+q]} \leftarrow C$

9: **return** B.

Algorithm 36 The slide reduction algorithm for $n \geq 2k$

Input: Blocksize k, termination factor ε, a basis B of rank $n = pk + q$ where $0 \leq q < k$, and an SVP-oracle in rank k.

Output: An (almost) k-slide-reduced basis of $L(B)$.

while $\mathrm{vol}(B_{[1,ik+q]})$ is modified for some $i \in [1, p-1]$ **do**

  $(1+\varepsilon)\eta$-HSVP-reduce $B_{[1,k+q]}$ using DBKZ for $\eta := \gamma_k^{\frac{k+q-1}{2(k-1)}}$

  for $i = 1$ to $p-1$ **do** SVP-reduce $B_{[ik+q+1,(i+1)k+q]}$

  if $B_{[2,k+q+1]}$ is not $(1+\varepsilon)\eta$-DHSVP-reduced **then** $\sqrt{1+\varepsilon}\eta$-DHSVP-reduce $B_{[2,k+q+1]}$ using DBKZ

  if $B_{[ik+q+2,(i+1)k+q+1]}$ is not $\sqrt{(1+\varepsilon)\gamma_k}$-DHSVP-reduced for some $i \in [1, p-2]$ **then** $\sqrt{\gamma_k}$-DHSVP-reduce $B_{[ik+q+2,(i+1)k+q+1]}$

end while

Find a $\gamma_k^{\frac{n-k}{k-1}}$-SVP-reduced basis $C = (\mathbf{c}_1, \ldots, \mathbf{c}_{k+q})$ for the sublattice $B_{[1,k+q]}$ using our first algorithm

if $\|\mathbf{c}_1\| < \|\mathbf{b}_1\|$ **then** $B_{[1,k+q]} \leftarrow C$

return B.

Approximating SVP with (at least) polynomial factor

★ Th: Our slide reduction algorithm for $n \geq 2k$ terminates within $\mathrm{poly}(B_{\mathrm{input}}, 1/\varepsilon)$ calls to SVP-oracle in rank k, and outputs a basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of the input lattice L s.t.

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

- It includes both GN-slide-reduction and DBKZ as special cases.

Approximating SVP with (at least) polynomial factor

★ Th: Our slide reduction algorithm for $n \geq 2k$ terminates within $\mathrm{poly}(B_{\mathrm{input}}, 1/\varepsilon)$ calls to SVP-oracle in rank k, and outputs a basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of the input lattice L s.t.

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

$$\|\mathbf{b}_1\| \leq (1+\varepsilon)^{O(1)}((1+\varepsilon)\gamma_k)^{\frac{n-k}{k-1}} \lambda_1(L).$$

- It includes both GN-slide-reduction and DBKZ as special cases.

Conclusion

The best polynomial-time lattice reduction in theory, including the first non-trivial algorithm for approximating SVP with sublinear factors $n^{\frac{1}{2}} \leq f \leq n^{1-\varepsilon}$:

- The significantly exponentially faster provable/heuristic algorithm for approximating SVP with factor $n^{1/2} \leq f \leq n^{O(1)}$;

  $\Rightarrow$ The regime most relevant for cryptography.

  $\Rightarrow$ Security estimates of lattice-based cryptosystems.

$n^{0.99}$-SVP	Provable: $2^{0.802n} \rightarrow 2^{0.405n}$
	Heuristic: $2^{0.292n} \rightarrow 2^{0.148n}$
$n^{1.99}$-SVP	Provable: $2^{0.401n} \rightarrow 2^{0.269n}$
	Heuristic: $2^{0.146n} \rightarrow 2^{0.098n}$

- For solving $n^{c \in [\frac{1}{2}, O(1)]}$-SVP, it is more efficient to run blockwise lattice reduction with an approximate rather than exact SVP-oracle in low ranks.

Conclusion

The best polynomial-time lattice reduction in theory, including the first non-trivial algorithm for approximating SVP with sublinear factors $n^{\frac{1}{2}} \leq f \leq n^{1-\varepsilon}$:

- The significantly exponentially faster provable/heuristic algorithm for approximating SVP with factor $n^{1/2} \leq f \leq n^{O(1)}$;
  $\Rightarrow$ The regime most relevant for cryptography.
  $\Rightarrow$ Security estimates of lattice-based cryptosystems.

$n^{0.99}$-SVP	Provable: $2^{0.802n} \rightarrow 2^{0.405n}$
	Heuristic: $2^{0.292n} \rightarrow 2^{0.148n}$
$n^{1.99}$-SVP	Provable: $2^{0.401n} \rightarrow 2^{0.269n}$
	Heuristic: $2^{0.146n} \rightarrow 2^{0.098n}$

- For solving $n^{c \in [\frac{1}{2}, O(1)]}$-SVP, it is more efficient to run blockwise lattice reduction with an approximate rather than exact SVP-oracle in low ranks.

Conclusion

The best polynomial-time lattice reduction in theory, including the first non-trivial algorithm for approximating SVP with sublinear factors $n^{\frac{1}{2}} \leq f \leq n^{1-\varepsilon}$:

- The significantly exponentially faster provable/heuristic algorithm for approximating SVP with factor $n^{1/2} \leq f \leq n^{O(1)}$;
  $\Rightarrow$ The regime most relevant for cryptography.
  $\Rightarrow$ Security estimates of lattice-based cryptosystems.

$n^{0.99}$-SVP	Provable: $2^{0.802n} \rightarrow 2^{0.405n}$
	Heuristic: $2^{0.292n} \rightarrow 2^{0.148n}$
$n^{1.99}$-SVP	Provable: $2^{0.401n} \rightarrow 2^{0.269n}$
	Heuristic: $2^{0.146n} \rightarrow 2^{0.098n}$

- For solving $n^{c \in [\frac{1}{2}, O(1)]}$-SVP, it is more efficient to run blockwise lattice reduction with an approximate rather than exact SVP-oracle in low ranks.

Open problems

Q1 Can we rigorously prove (without any heuristic assumption) that within polynomial calls to SVP-oracle, the (original) BKZ algorithm outputs an almost BKZ-reduced basis?
$\Rightarrow$ Can we rigorously prove that within polynomial calls to SVP-oracle, the BKZ algorithm achieves almost the same quality guarantees as that of our slide-reduction algorithm?

Q2 Is there an non-trivial (lattice reduction) algorithm for approximating SVP with sublinear factors $n^{\varepsilon} \leq f \leq n^{\frac{1}{2}}$?

Q3 . . . .

Open problems

Q1 Can we rigorously prove (without any heuristic assumption) that within polynomial calls to SVP-oracle, the (original) BKZ algorithm outputs an almost BKZ-reduced basis?
$\Rightarrow$ Can we rigorously prove that within polynomial calls to SVP-oracle, the BKZ algorithm achieves almost the same quality guarantees as that of our slide-reduction algorithm?

Q2 Is there an non-trivial (lattice reduction) algorithm for approximating SVP with sublinear factors $n^{\varepsilon} \leq f \leq n^{\frac{1}{2}}$?

Q3 $\cdots\cdots\cdots$

Open problems

Q1 Can we rigorously prove (without any heuristic assumption) that within polynomial calls to SVP-oracle, the (original) BKZ algorithm outputs an almost BKZ-reduced basis?
$\Rightarrow$ Can we rigorously prove that within polynomial calls to SVP-oracle, the BKZ algorithm achieves almost the same quality guarantees as that of our slide-reduction algorithm?

Q2 Is there an non-trivial (lattice reduction) algorithm for approximating SVP with sublinear factors $n^{\varepsilon} \leq f \leq n^{\frac{1}{2}}$?

Q3 . . . .

Open problems

Q1 Can we rigorously prove (without any heuristic assumption) that within polynomial calls to SVP-oracle, the (original) BKZ algorithm outputs an almost BKZ-reduced basis?
$\Rightarrow$ Can we rigorously prove that within polynomial calls to SVP-oracle, the BKZ algorithm achieves almost the same quality guarantees as that of our slide-reduction algorithm?

Q2 Is there an non-trivial (lattice reduction) algorithm for approximating SVP with sublinear factors $n^{\varepsilon} \leq f \leq n^{\frac{1}{2}}$?

Q3 . . . .

$n^{0.99}$-SVP	Provable: $2^{0.802n} \to 2^{0.405n}$
	Heuristic: $2^{0.292n} \to 2^{0.148n}$
$n^{1.99}$-SVP	Provable: $2^{0.401n} \to 2^{0.269n}$
	Heuristic: $2^{0.146n} \to 2^{0.098n}$

Thank you!