Une introduction à la réduction de réseaux (2ème partie)

Damien Stehlé

ÉNS de Lyon

JNCF, Mai 2013

Résumé de la 1ère partie

- Un réseau L de dimension ≥ 2 admet une infinité de bases.
- SVP : Pour de nombreuses applications, il est utile de savoir trouver des vecteurs de petite norme dans un réseau.
- La réduction de réseaux vise à tranformer une base en une autre t.q. les coefficients diagonaux du facteur R ne décroissent pas trop vite.
- La réduction HKZ remplit ces conditions mais coûte trop cher.
- La réduction LLL impose que deux coefficients diagonaux successifs ne décroissent jamais plus d'un facteur constant.

$\Rightarrow$ LLL résout SVP_γ pour $\gamma = 2^{O(n)}$ en temps polynomial.

Résumé de la 1ère partie

- Un réseau L de dimension ≥ 2 admet une infinité de bases.
- SVP : Pour de nombreuses applications, il est utile de savoir trouver des vecteurs de petite norme dans un réseau.
- La réduction de réseaux vise à tranformer une base en une autre t.q. les coefficients diagonaux du facteur R ne décroissent pas trop vite.
- La réduction HKZ remplit ces conditions mais coûte trop cher.
- La réduction LLL impose que deux coefficients diagonaux successifs ne décroissent jamais plus d'un facteur constant.

$\Rightarrow$ LLL résout SVP_γ pour $\gamma = 2^{O(n)}$ en temps polynomial.

Résumé de la 1ère partie

- Un réseau L de dimension ≥ 2 admet une infinité de bases.
- SVP : Pour de nombreuses applications, il est utile de savoir trouver des vecteurs de petite norme dans un réseau.
- La réduction de réseaux vise à tranformer une base en une autre t.q. les coefficients diagonaux du facteur R ne décroissent pas trop vite.
- La réduction HKZ remplit ces conditions mais coûte trop cher.
- La réduction LLL impose que deux coefficients diagonaux successifs ne décroissent jamais plus d'un facteur constant.

$\Rightarrow$ LLL résout SVP_γ pour $\gamma = 2^{O(n)}$ en temps polynomial.

Résumé de la 1ère partie

- Un réseau L de dimension ≥ 2 admet une infinité de bases.
- SVP : Pour de nombreuses applications, il est utile de savoir trouver des vecteurs de petite norme dans un réseau.
- La réduction de réseaux vise à tranformer une base en une autre t.q. les coefficients diagonaux du facteur R ne décroissent pas trop vite.
- La réduction HKZ remplit ces conditions mais coûte trop cher.
- La réduction LLL impose que deux coefficients diagonaux successifs ne décroissent jamais plus d'un facteur constant.

$\Rightarrow$ LLL résout SVP_γ pour $\gamma = 2^{O(n)}$ en temps polynomial.

Résumé de la 1ère partie

- Un réseau L de dimension ≥ 2 admet une infinité de bases.
- SVP : Pour de nombreuses applications, il est utile de savoir trouver des vecteurs de petite norme dans un réseau.
- La réduction de réseaux vise à tranformer une base en une autre t.q. les coefficients diagonaux du facteur R ne décroissent pas trop vite.
- La réduction HKZ remplit ces conditions mais coûte trop cher.
- La réduction LLL impose que deux coefficients diagonaux successifs ne décroissent jamais plus d'un facteur constant.

$\Rightarrow$ LLL résout SVP_γ pour $\gamma = 2^{O(n)}$ en temps polynomial.

Plan du cours

1. L'algorithme BKZ
2. Une application cryptographique
3. Un LLL hybride numérique-algébrique
4. Problèmes ouverts

Plan du cours

Hier, on avait un choix à faire

Objectif de la réduction de réseaux

Étant donnée $R \in \mathbb{R}^{n \times n}$ triangulaire supérieure, trouver $U \in \mathrm{GL}_n(\mathbb{Z})$ t.q. le facteur R de $R \cdot U$ ait de petits coefficients diagonaux

HKZ est trop cher... Que peut-on faire ?

1. Échanger deux vecteurs consécutifs t.q. $r_{i+1,i+1} \ll r_{i,i}$ [LLL82]
2. Équilibrer localement les coefficients diagonaux en appliquant HKZ à une sous-matrice sur la diagonale de R [Sch87]

$$\begin{bmatrix} \ddots & & & \vdots \\ & \begin{bmatrix} r_{ii} & \cdots & \cdots \\ & \ddots & \vdots \\ & & r_{jj} \end{bmatrix} & & \vdots \\ & & \ddots & \\ \cdots & \cdots & & \end{bmatrix} \cdot \begin{bmatrix} Id & \cdots & \vdots \\ & \begin{bmatrix} & \\ & U_{HKZ} \\ & \end{bmatrix} & \vdots \\ & & Id \end{bmatrix}$$

Multiples stratégies : alternative, entrelacement, ...

Hier, on avait un choix à faire

Objectif de la réduction de réseaux

Étant donnée $R \in \mathbb{R}^{n \times n}$ triangulaire supérieure, trouver $U \in \mathrm{GL}_n(\mathbb{Z})$ t.q. le facteur R de $R \cdot U$ ait de petits coefficients diagonaux

HKZ est trop cher... Que peut-on faire ?

1. Échanger deux vecteurs consécutifs t.q. $r_{i+1,i+1} \ll r_{i,i}$ [LLL82]
2. Équilibrer localement les coefficients diagonaux en appliquant HKZ à une sous-matrice sur la diagonale de R [Sch87]

$$\begin{bmatrix} \ddots & & & \vdots \\ & \begin{bmatrix} r_{ii} & \cdots & \cdots \\ & \ddots & \vdots \\ & & r_{jj} \end{bmatrix} & & \vdots \\ & & \ddots & \\ \cdots & \cdots & & \cdots \end{bmatrix} \cdot \begin{bmatrix} Id & \cdots & \vdots \\ & \begin{bmatrix} & U_{HKZ} & \end{bmatrix} & \vdots \\ & & Id \end{bmatrix}$$

Multiples stratégies : adaptative, non-adaptative, gloutonne, etc.

Hier, on avait un choix à faire

Objectif de la réduction de réseaux

Étant donnée $R \in \mathbb{R}^{n \times n}$ triangulaire supérieure, trouver $U \in \mathrm{GL}_n(\mathbb{Z})$ t.q. le facteur R de $R \cdot U$ ait de petits coefficients diagonaux

HKZ est trop cher... Que peut-on faire ?

1. Échanger deux vecteurs consécutifs t.q. $r_{i+1,i+1} \ll r_{i,i}$ [LLL82]
2. Équilibrer localement les coefficients diagonaux en appliquant HKZ à une sous-matrice sur la diagonale de R [Sch87]

$$\begin{bmatrix} \ddots & & \vdots & \\ & \begin{bmatrix} r_{ii} & \cdots & \cdots \\ & \ddots & \vdots \\ & & r_{jj} \end{bmatrix} & \vdots & \\ & & \ddots & \end{bmatrix} \cdot \begin{bmatrix} Id & \cdots & \vdots & \\ & \begin{bmatrix} & U_{HKZ} & \end{bmatrix} & \vdots & \\ & & & Id \end{bmatrix}$$

Multiples stratégies : adaptative, non-adaptative, gloutonne, etc

Hier, on avait un choix à faire

Objectif de la réduction de réseaux

Étant donnée $R \in \mathbb{R}^{n \times n}$ triangulaire supérieure, trouver $U \in \mathrm{GL}_n(\mathbb{Z})$ t.q. le facteur R de $R \cdot U$ ait de petits coefficients diagonaux

HKZ est trop cher... Que peut-on faire ?

1. Échanger deux vecteurs consécutifs t.q. $r_{i+1,i+1} \ll r_{i,i}$ [LLL82]
2. Équilibrer localement les coefficients diagonaux en appliquant HKZ à une sous-matrice sur la diagonale de R [Sch87]

$$\begin{bmatrix} \ddots & & \vdots & \\ & \begin{bmatrix} r_{ii} & \cdots & \cdots \\ & \ddots & \vdots \\ & & r_{jj} \end{bmatrix} & \vdots & \\ & & \ddots & \end{bmatrix} \cdot \begin{bmatrix} Id & \cdots & \vdots & \\ & \begin{bmatrix} & U_{HKZ} & \end{bmatrix} & \vdots & \\ & & & Id \end{bmatrix}$$

Multiples stratégies : adaptative, non-adaptative, gloutonne, etc

L'algorithme BKZ [Sch87,SchEuc91]

- Du local au global :
  Appliquer HKZ à des sous-blocs diagonaux de R
- La dimension k des sous-blocs détermine un compromis temps/qualité.

BKZ_k (version simplifiée)

Entrée : $R \in \mathbb{R}^{n \times n}$ triangulaire supérieure
Répéter
  Pour $i = 1..n-k+1$, faire
    HKZ-réduire le sous-bloc de R de dim k commençant à la position i
    Mettre à jour le facteur R et le proprifier

- Quand s'arrête-t-on ? Combien d'itérations ?
- Quelle est la qualité de la sortie ?

L'algorithme BKZ [Sch87,SchEuc91]

- Du local au global :
  Appliquer HKZ à des sous-blocs diagonaux de R
- La dimension k des sous-blocs détermine un compromis temps/qualité.

BKZ_k (version simplifiée)

Entrée : $R \in \mathbb{R}^{n \times n}$ triangulaire supérieure
Répéter
  Pour $i = 1..n-k+1$, faire
    HKZ-réduire le sous-bloc de R de dim k commençant à la position i
    Mettre à jour le facteur R et le proprifier

- Quand s'arrête-t-on ? Combien d'itérations ?
- Quelle est la qualité de la sortie ?

L'algorithme BKZ [Sch87,SchEuc91]

- Du local au global :
  Appliquer HKZ à des sous-blocs diagonaux de R
- La dimension k des sous-blocs détermine un compromis temps/qualité.

BKZ_k (version simplifiée)

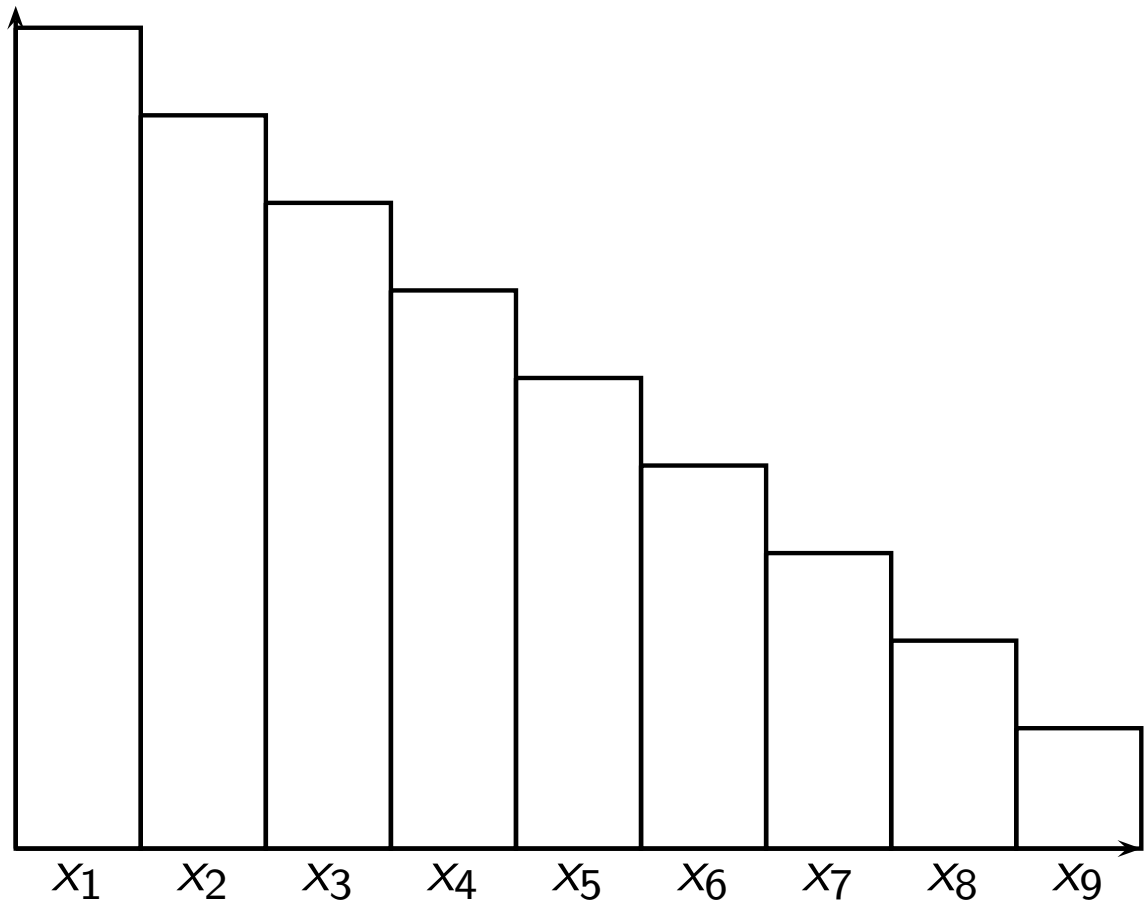
Entrée : $R \in \mathbb{R}^{n \times n}$ triangulaire supérieure
Répéter
  Pour $i = 1..n-k+1$, faire
    HKZ-réduire le sous-bloc de R de dim k commençant à la position i
    Mettre à jour le facteur R et le proprifier

- Quand s'arrête-t-on ? Combien d'itérations ?
- Quelle est la qualité de la sortie ?

Un modèle "tas-de-sable" pour BKZ_k [HaPuSt11]

Hypothèse de régularité :

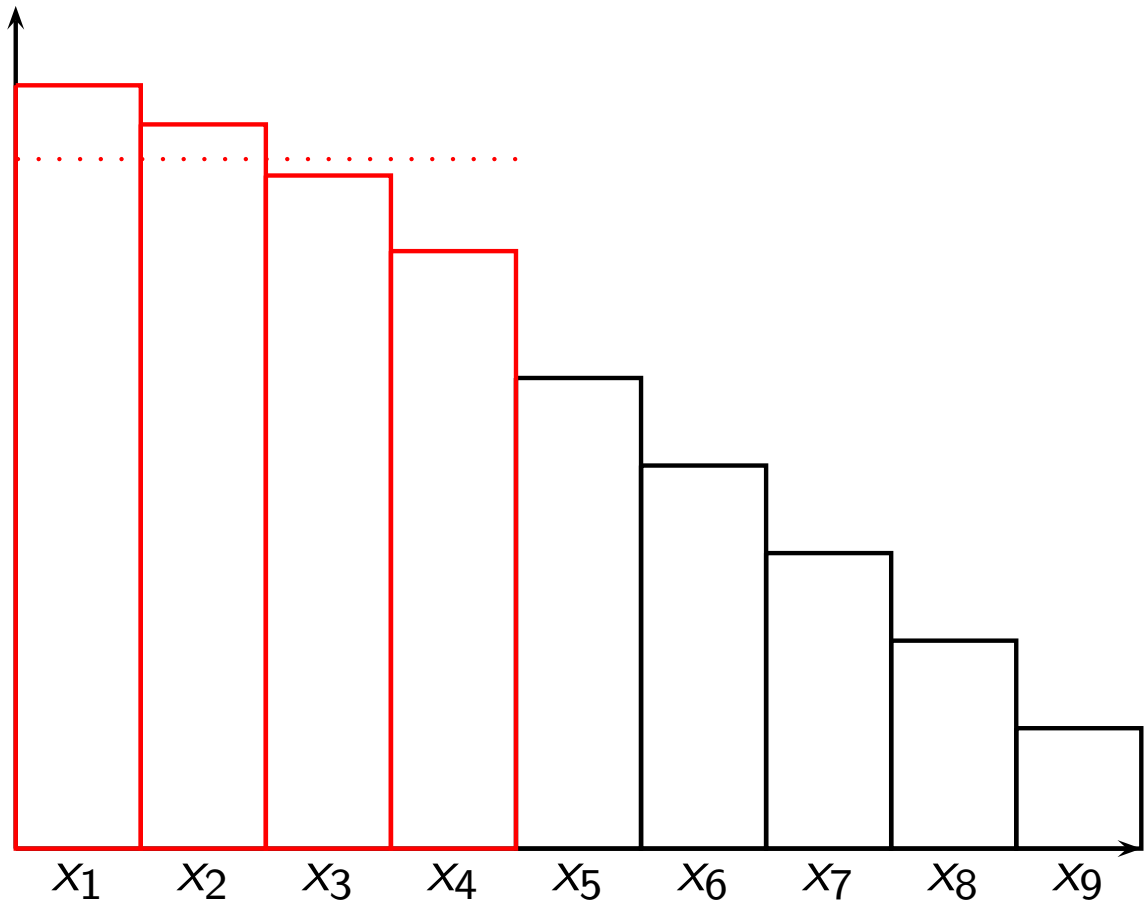
Chaque réduction HKZ aboutit à un profil pire-cas $(x_i = \log r_{ii})$



Un modèle "tas-de-sable" pour BKZ_k [HaPuSt11]

Hypothèse de régularité :

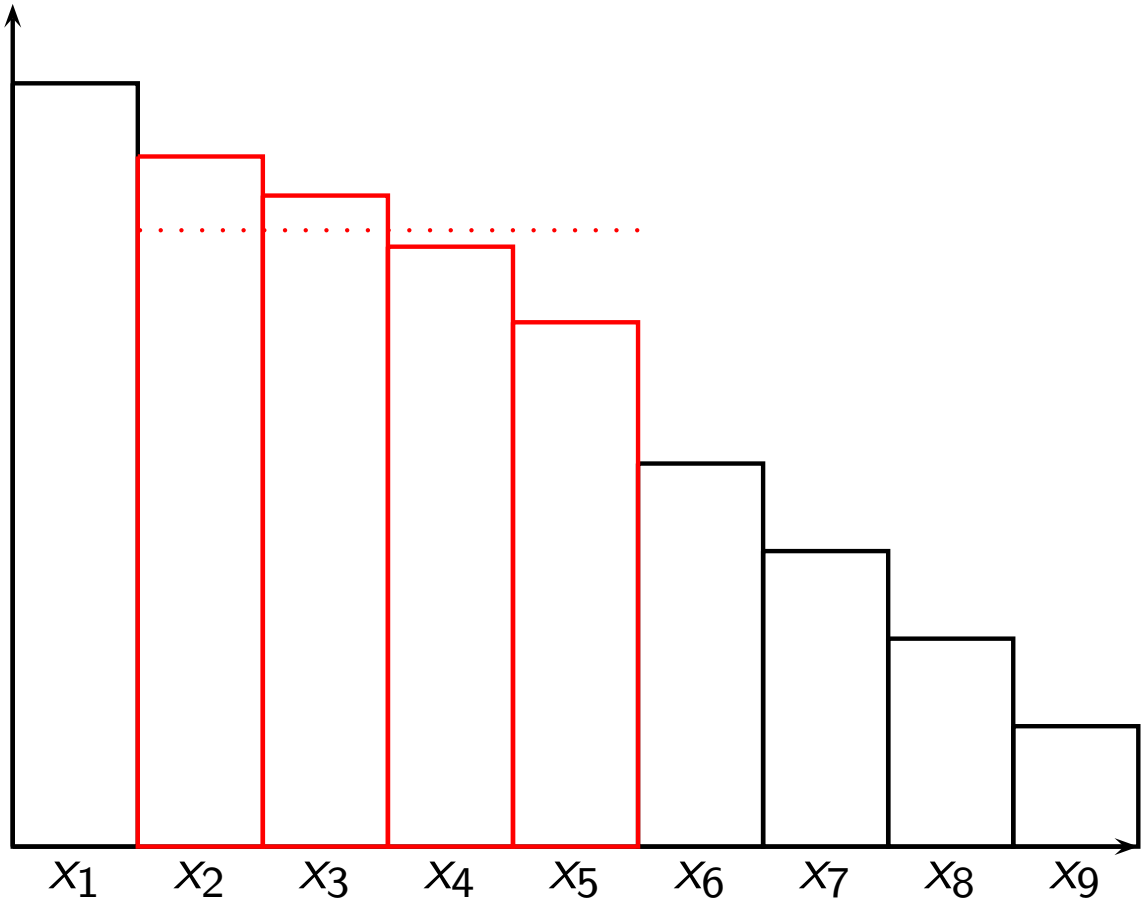
Chaque réduction HKZ aboutit à un profil pire-cas $(x_i = \log r_{ii})$



Un modèle "tas-de-sable" pour BKZ_k [HaPuSt11]

Hypothèse de régularité :

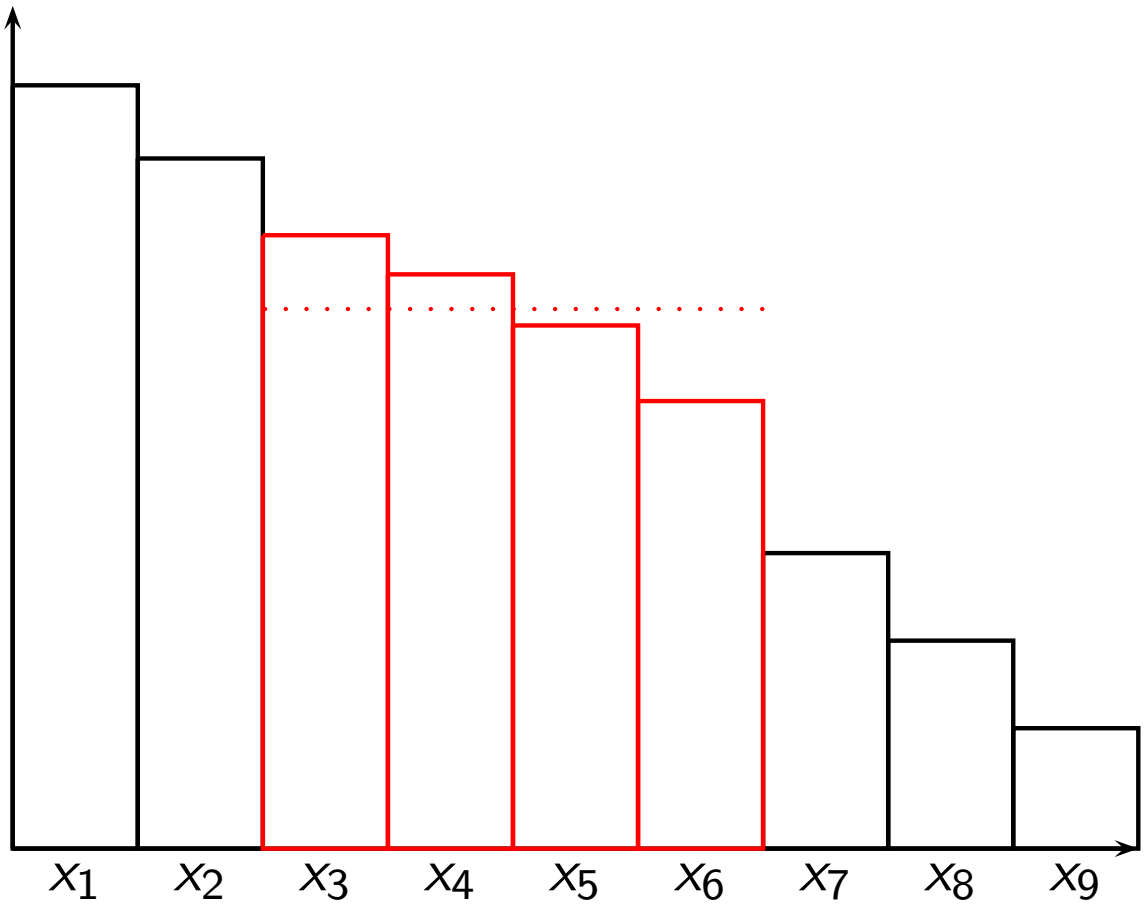
Chaque réduction HKZ aboutit à un profil pire-cas $(x_i = \log r_{ii})$



Un modèle "tas-de-sable" pour BKZ_k [HaPuSt11]

Hypothèse de régularité :

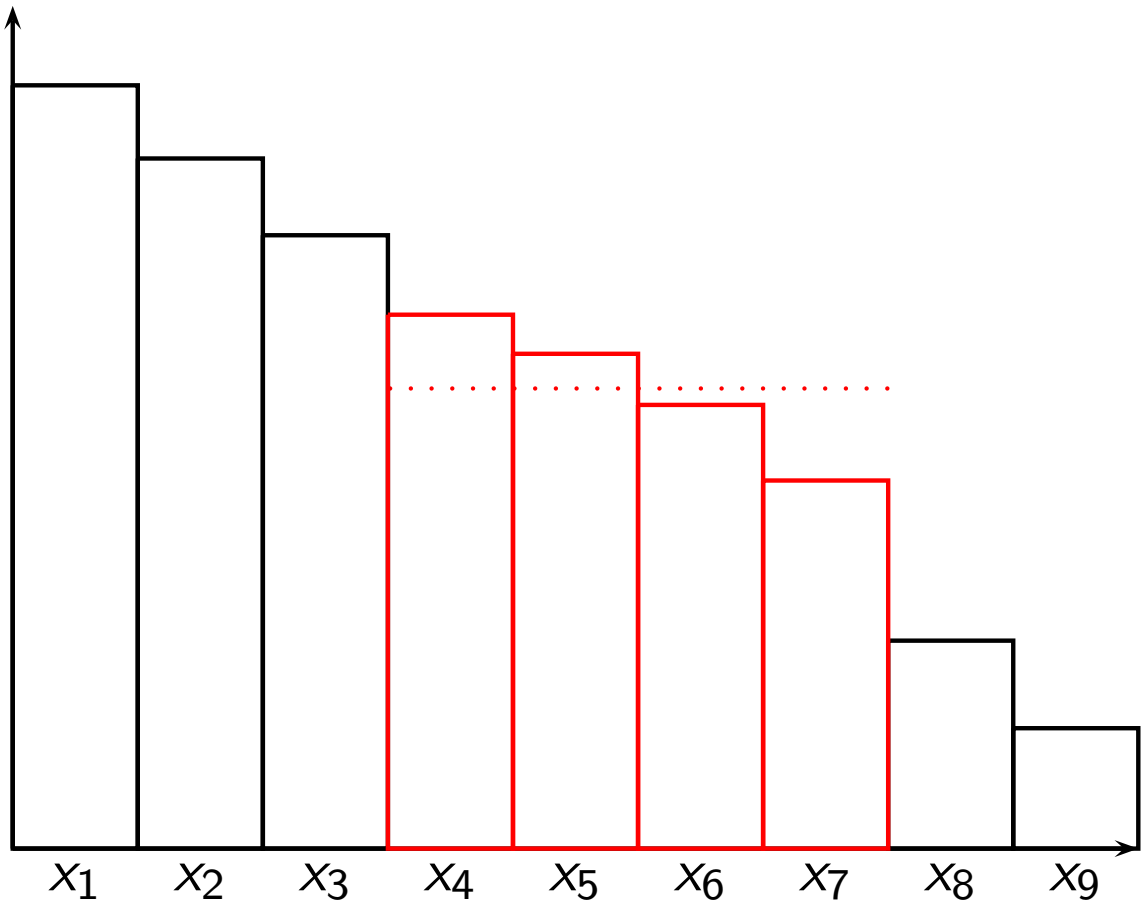
Chaque réduction HKZ aboutit à un profil pire-cas $(x_i = \log r_{ii})$



Un modèle "tas-de-sable" pour BKZ_k [HaPuSt11]

Hypothèse de régularité :

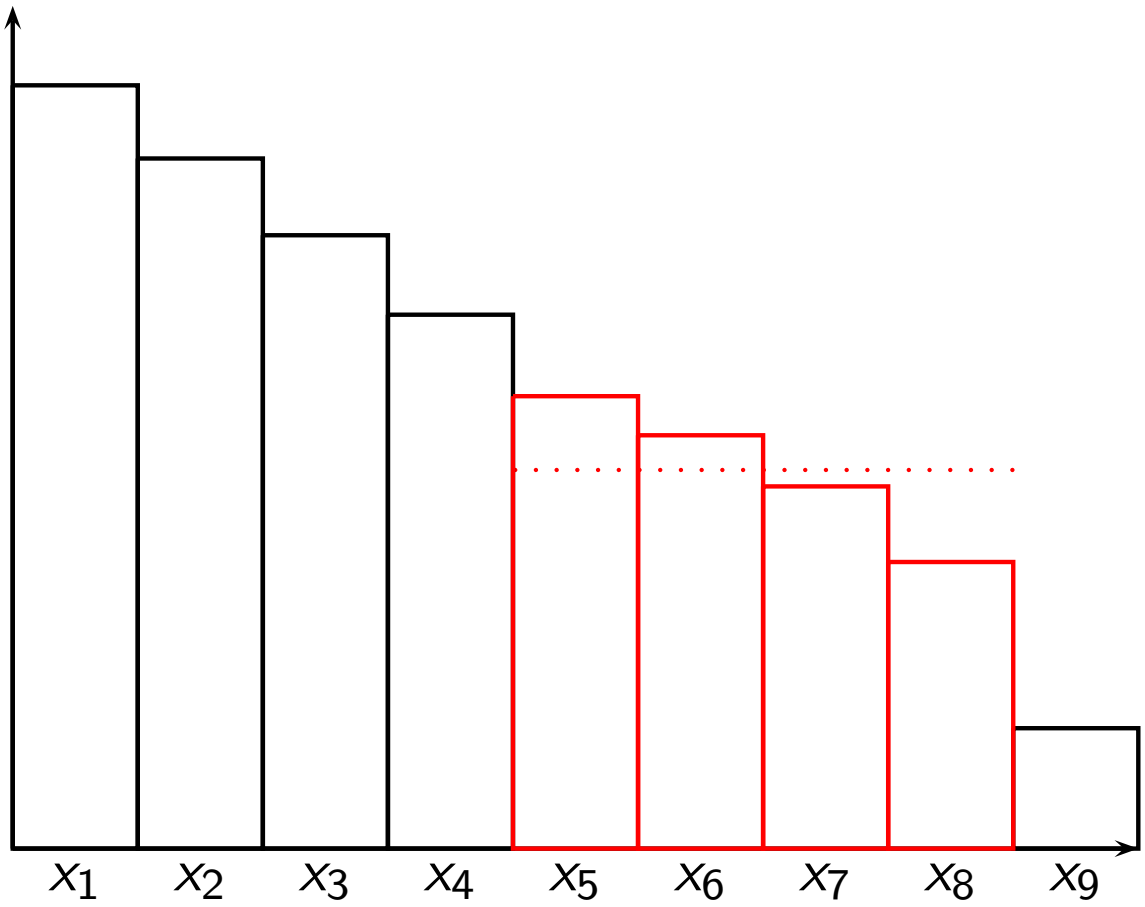
Chaque réduction HKZ aboutit à un profil pire-cas $(x_i = \log r_{ii})$



Un modèle "tas-de-sable" pour BKZ_k [HaPuSt11]

Hypothèse de régularité :

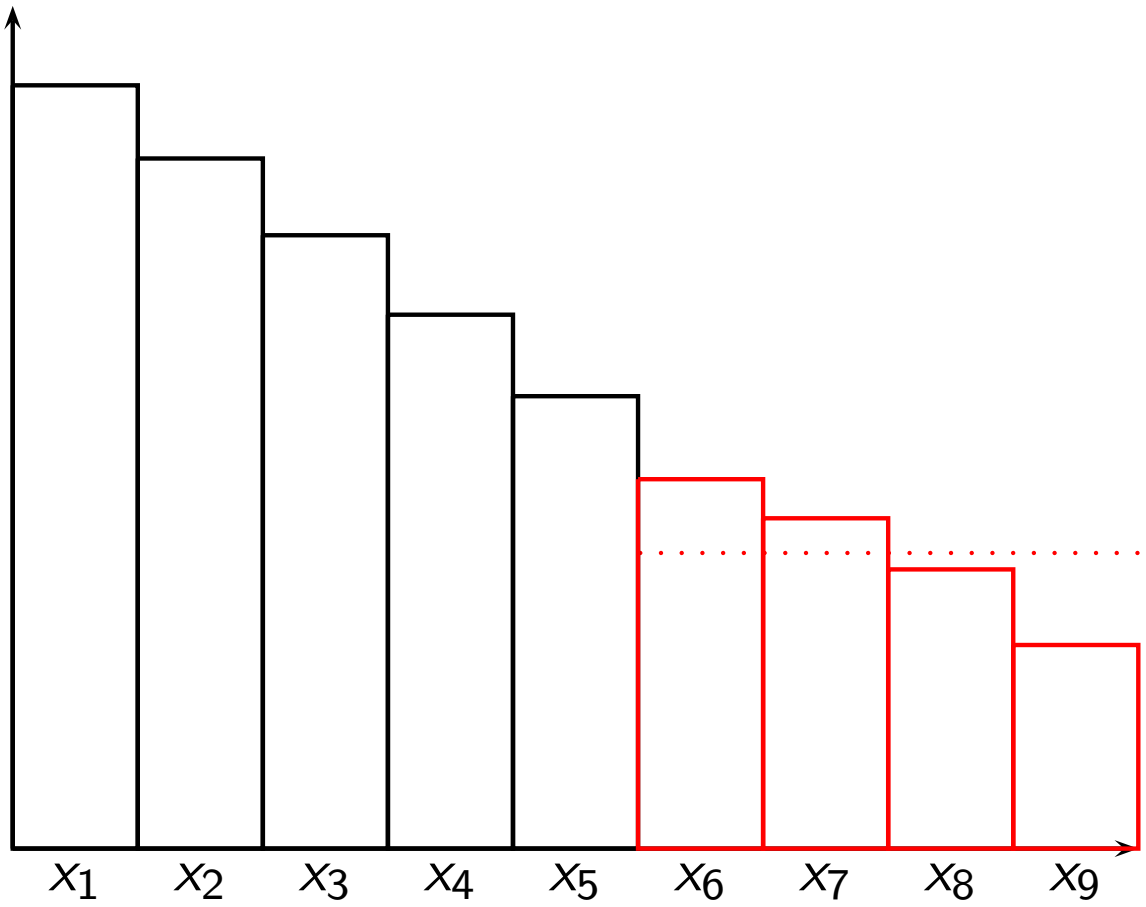
Chaque réduction HKZ aboutit à un profil pire-cas $(x_i = \log r_{ii})$



Un modèle "tas-de-sable" pour BKZ_k [HaPuSt11]

Hypothèse de régularité :

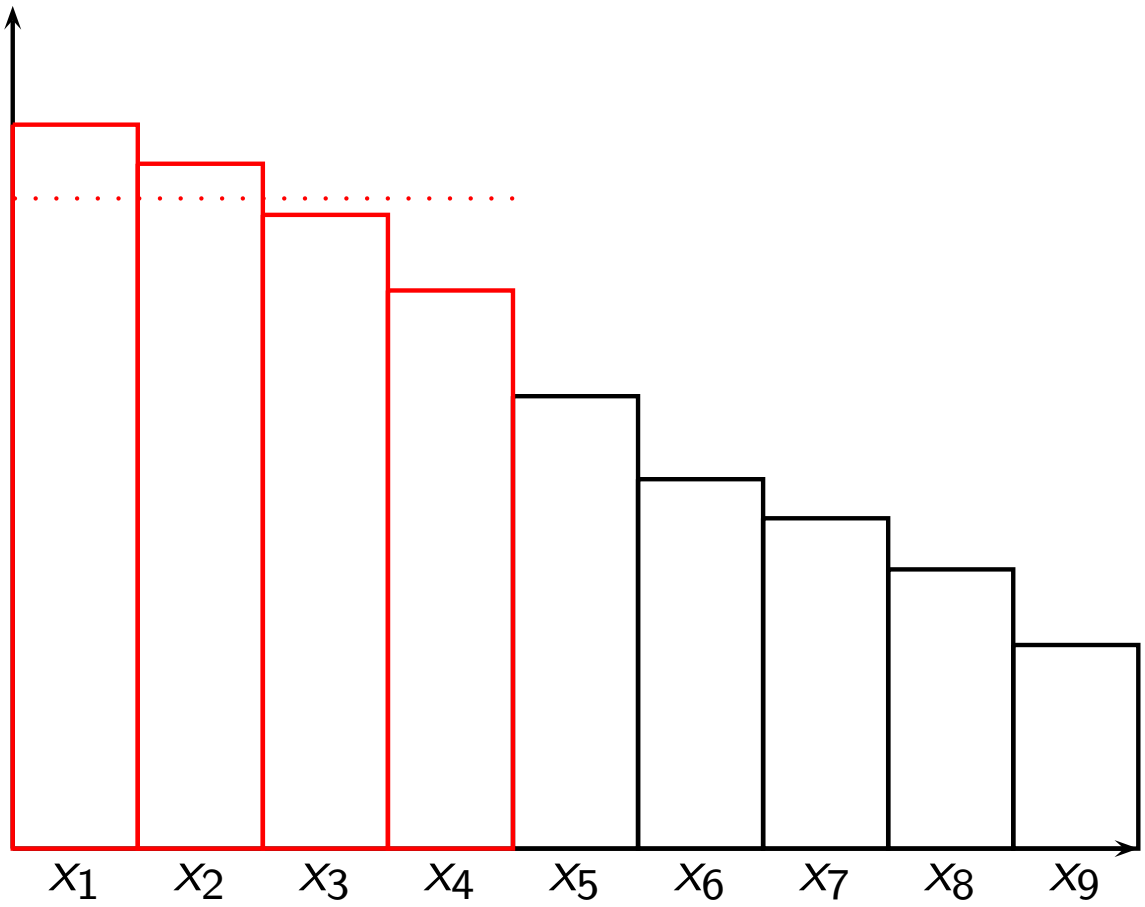
Chaque réduction HKZ aboutit à un profil pire-cas $(x_i = \log r_{ii})$



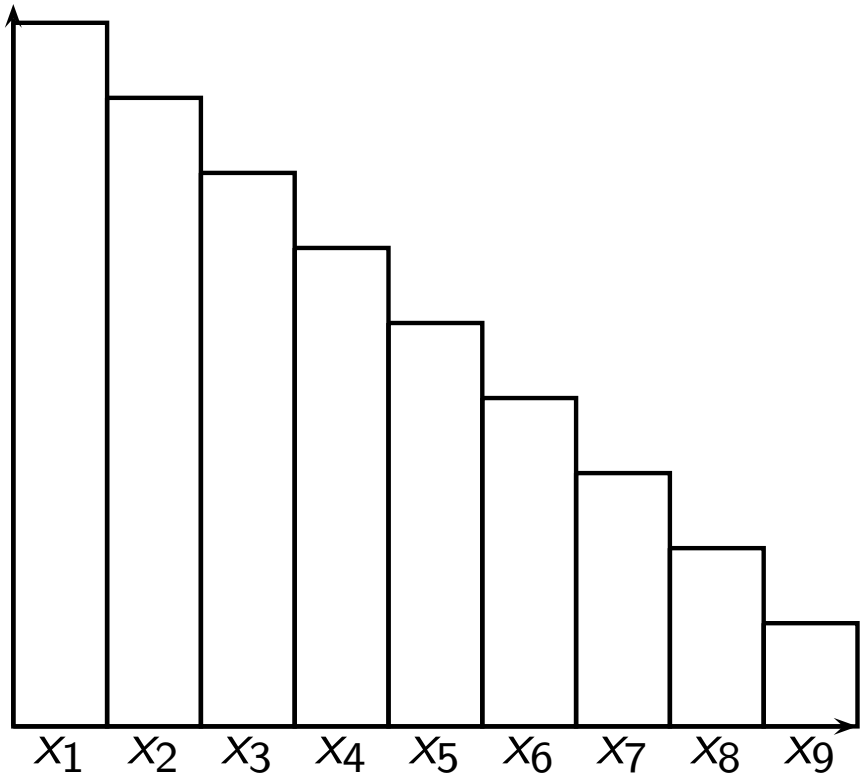
Un modèle "tas-de-sable" pour BKZ_k [HaPuSt11]

Hypothèse de régularité :

Chaque réduction HKZ aboutit à un profil pire-cas $(x_i = \log r_{ii})$



Dynamique du "tas-de-sable" de BKZ $(x_i = \log r_{ii})$



$$X = (x_1, \ldots, x_n)^T$$

$$X_{0.5} \leftarrow A_1 X$$

$$X_1 \leftarrow A_1 X + \Gamma_1$$

$$X_2 \leftarrow A_2 X_1 + \Gamma_2$$

$$\ldots$$

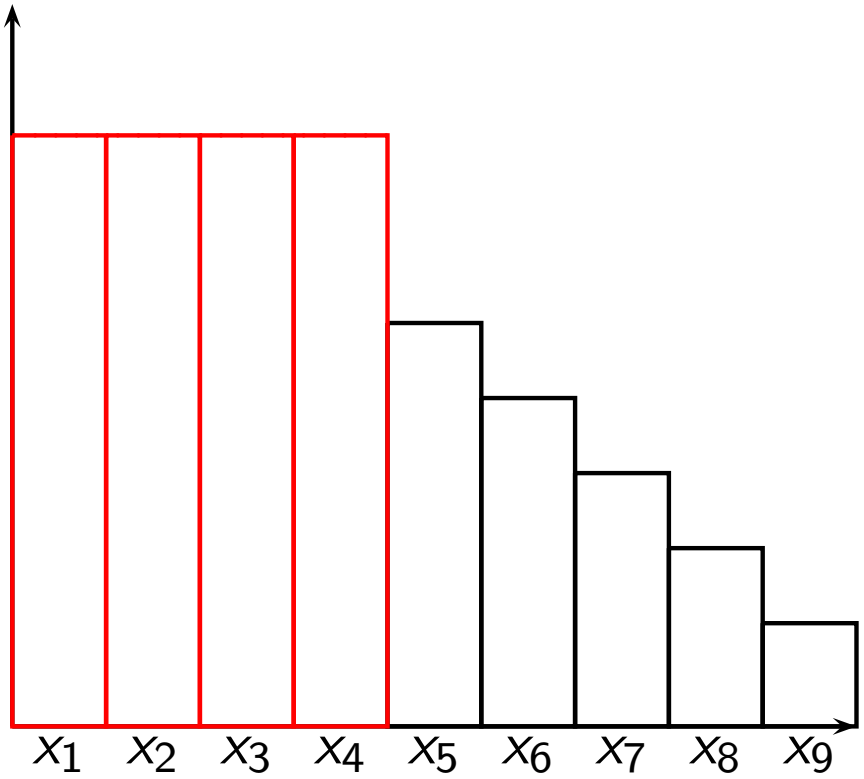
$$X_j = A_j X_j + \Gamma_j$$

avec $j = n - k + 1$

Un tour complet :

$$X' \leftarrow AX + \Gamma$$

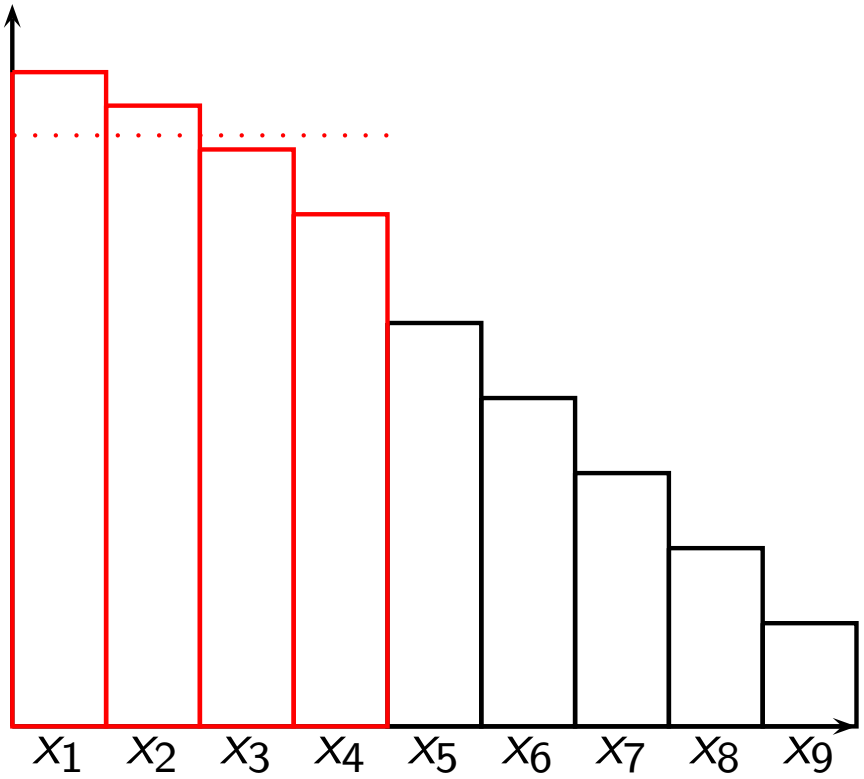
Dynamique du "tas-de-sable" de BKZ $(x_i = \log r_{ii})$



$$X = (x_1, \ldots, x_n)^T$$
$$X_{0.5} \leftarrow A_1 X$$
$$X_1 \leftarrow A_1 X + \Gamma_1$$
$$X_2 \leftarrow A_2 X_1 + \Gamma_2$$
$$\ldots$$
$$X_j = A_j X_j + \Gamma_j$$
avec $j = n - k + 1$

Un tour complet :
$$X' \leftarrow AX + \Gamma$$

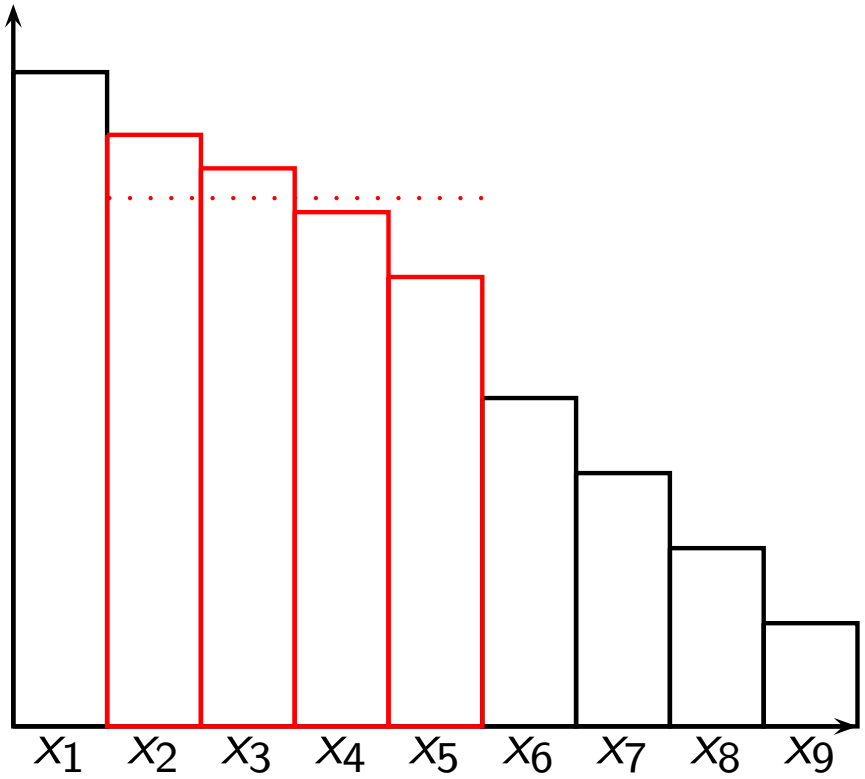
Dynamique du "tas-de-sable" de BKZ $(x_i = \log r_{ii})$



$$X = (x_1, \ldots, x_n)^T$$
$$X_{0.5} \leftarrow A_1 X$$
$$X_1 \leftarrow A_1 X + \Gamma_1$$
$$X_2 \leftarrow A_2 X_1 + \Gamma_2$$
$$\ldots$$
$$X_j = A_j X_j + \Gamma_j$$
avec $j = n - k + 1$

Un tour complet :
$$X' \leftarrow AX + \Gamma$$

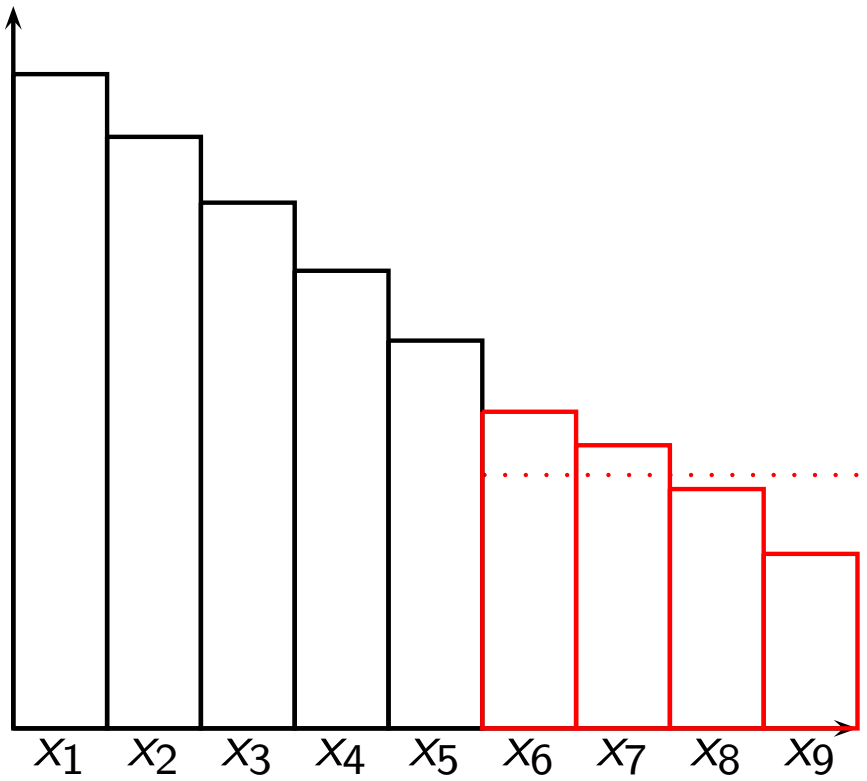
Dynamique du "tas-de-sable" de BKZ $(x_i = \log r_{ii})$



$$X = (x_1, \ldots, x_n)^T$$
$$X_{0.5} \leftarrow A_1 X$$
$$X_1 \leftarrow A_1 X + \Gamma_1$$
$$X_2 \leftarrow A_2 X_1 + \Gamma_2$$
$$\ldots$$
$$X_j = A_j X_j + \Gamma_j$$
avec $j = n - k + 1$

Un tour complet :
$$X' \leftarrow AX + \Gamma$$

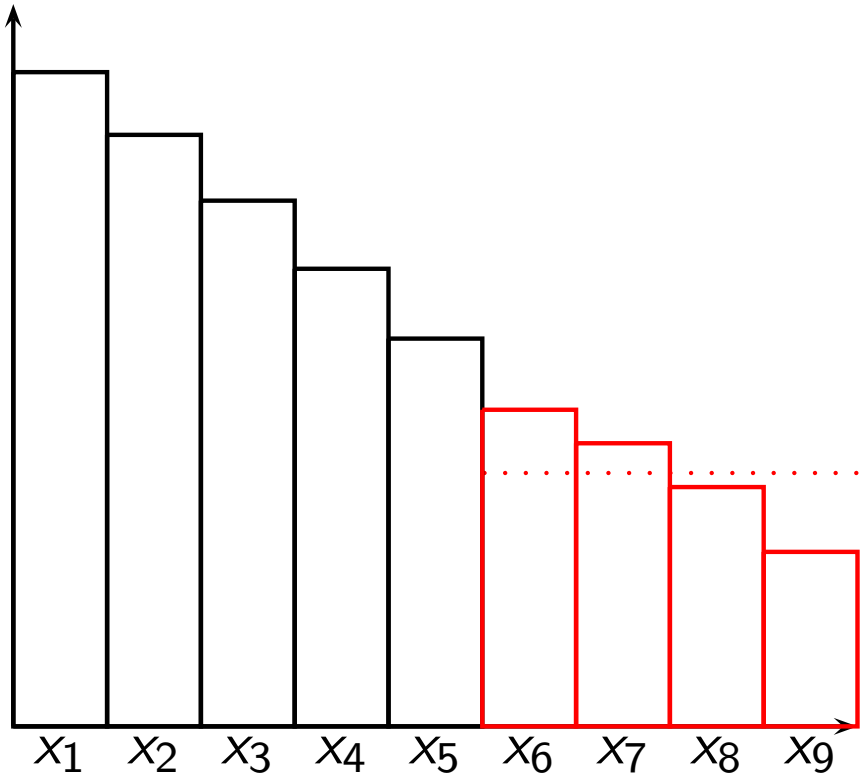
Dynamique du "tas-de-sable" de BKZ $(x_i = \log r_{ii})$



$$X = (x_1, \ldots, x_n)^T$$
$$X_{0.5} \leftarrow A_1 X$$
$$X_1 \leftarrow A_1 X + \Gamma_1$$
$$X_2 \leftarrow A_2 X_1 + \Gamma_2$$
$$\ldots$$
$$X_j = A_j X_j + \Gamma_j$$
avec $j = n - k + 1$

Un tour complet :
$X' \leftarrow AX + \Gamma$

Dynamique du "tas-de-sable" de BKZ $(x_i = \log r_{ii})$



$$X = (x_1, \ldots, x_n)^T$$
$$X_{0.5} \leftarrow A_1 X$$
$$X_1 \leftarrow A_1 X + \Gamma_1$$
$$X_2 \leftarrow A_2 X_1 + \Gamma_2$$
$$\ldots$$
$$X_j = A_j X_j + \Gamma_j$$
avec $j = n - k + 1$

Un tour complet :
$$X' \leftarrow AX + \Gamma$$

Analyse dans le modèle "tas-de-sable"

Système dynamique affine à temps discret

$$X \leftarrow AX + \Gamma$$

- A correspond aux moyennes locales successives
- Γ provient du facteur "$\sqrt{n}$" du théorème de Minkowski
- Qualité de la sortie $\Leftarrow$ points fixes
- Vitesse de convergence $\Leftarrow$ valeurs propres de $A^T A$

Analyse dans le modèle "tas-de-sable"

Système dynamique affine à temps discret

$$X \leftarrow AX + \Gamma$$

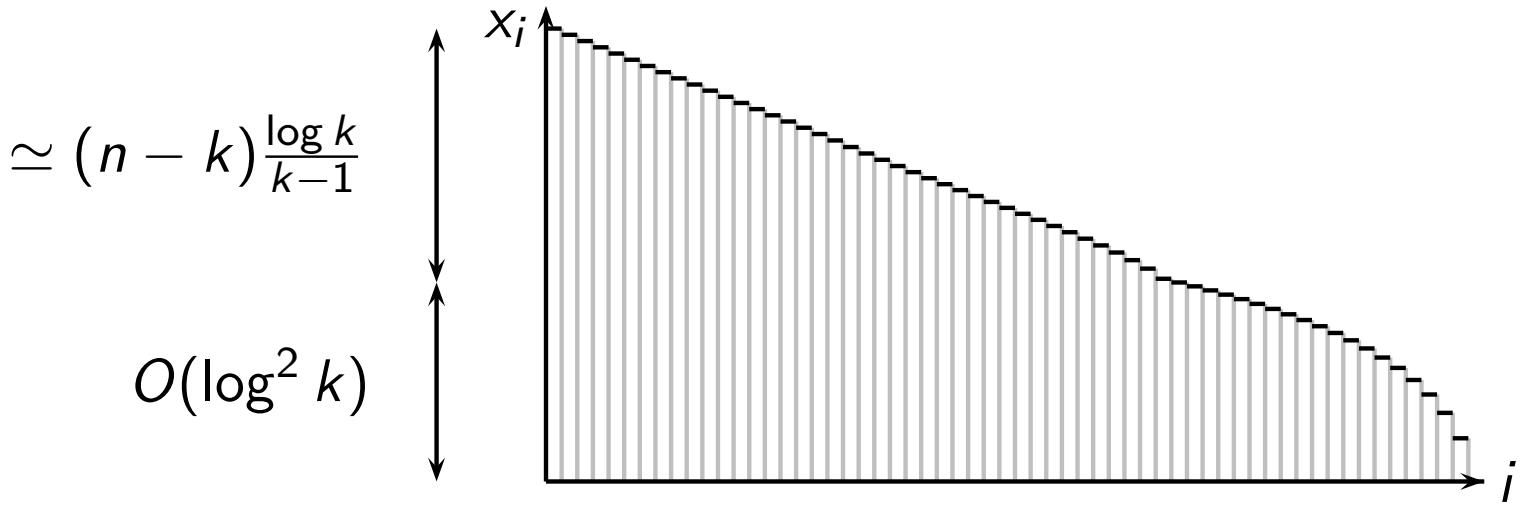
- A correspond aux moyennes locales successives
- Γ provient du facteur "$\sqrt{n}$" du théorème de Minkowski
- Qualité de la sortie $\Leftarrow$ points fixes
- Vitesse de convergence $\Leftarrow$ valeurs propres de $A^T A$

Analyse dans le modèle "tas-de-sable"

Système dynamique affine à temps discret

$$X \leftarrow AX + \Gamma$$

- A correspond aux moyennes locales successives
- Γ provient du facteur "$\sqrt{n}$" du théorème de Minkowski
- Qualité de la sortie $\Leftarrow$ points fixes



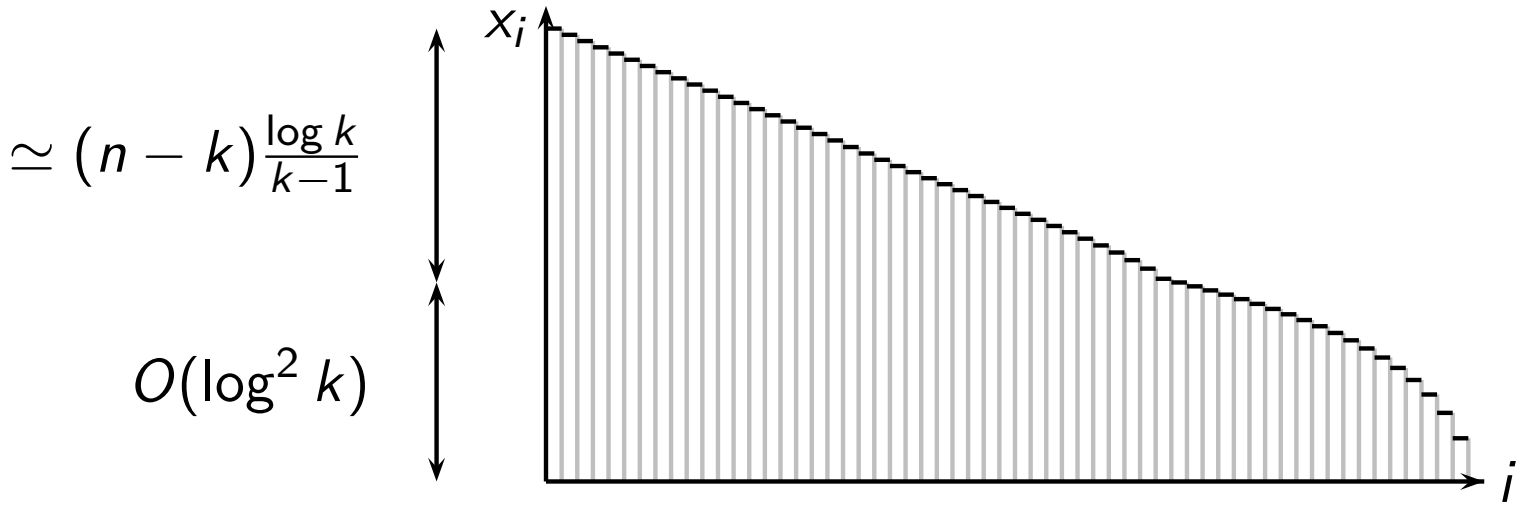
- Vitesse de convergence $\Leftarrow$ valeurs propres de $A^T A$

Analyse dans le modèle "tas-de-sable"

Système dynamique affine à temps discret

$$X \leftarrow AX + \Gamma$$

- A correspond aux moyennes locales successives
- Γ provient du facteur "$\sqrt{n}$" du théorème de Minkowski
- Qualité de la sortie $\Leftarrow$ points fixes



- Vitesse de convergence $\Leftarrow$ valeurs propres de $A^T A$

Analyse of BKZ

La plus grande valeur propre est 1... La seconde est $1 - \Omega(k^2/n^2)$.

$\Rightarrow$ Convergence quadratique : La distance au point fixe décroît d'un facteur 2 toutes les $O(n^2/k^2)$ itérations.

Coût de BKZ dans le modèle "tas-de-sable"

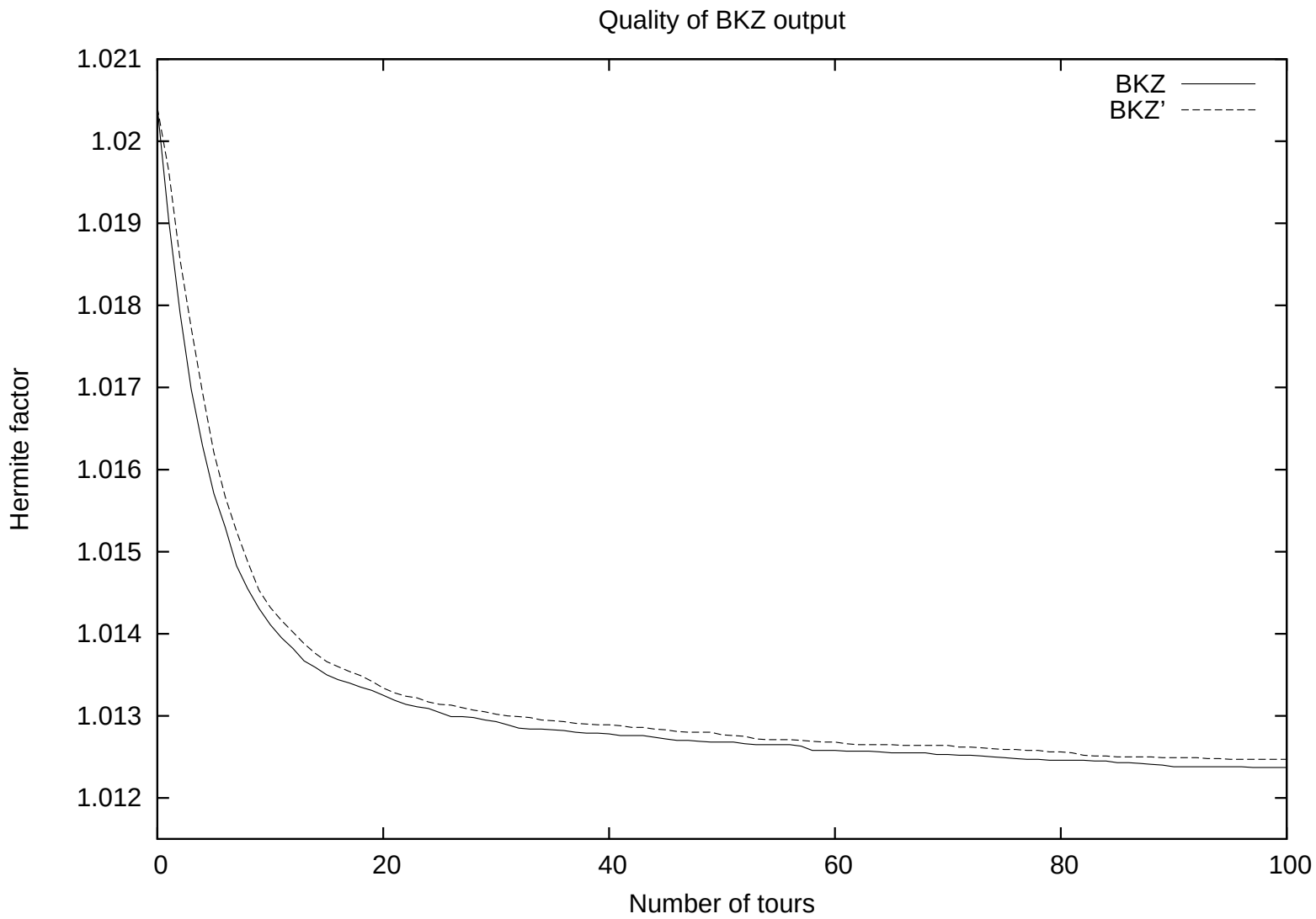
BKZ renvoie X à distance $\leq \varepsilon$ du point fixe en

$$O\left(\frac{n^3}{k^2}\left(-\log\varepsilon + \log\|X\|\right)\right) \text{ appels à } \mathrm{HKZ}_k.$$

Convergence très rapide : $\log\|X\| \leq \log\log\|B\|$ ($\|B\| = \max\|\mathbf{b}_i\|$).

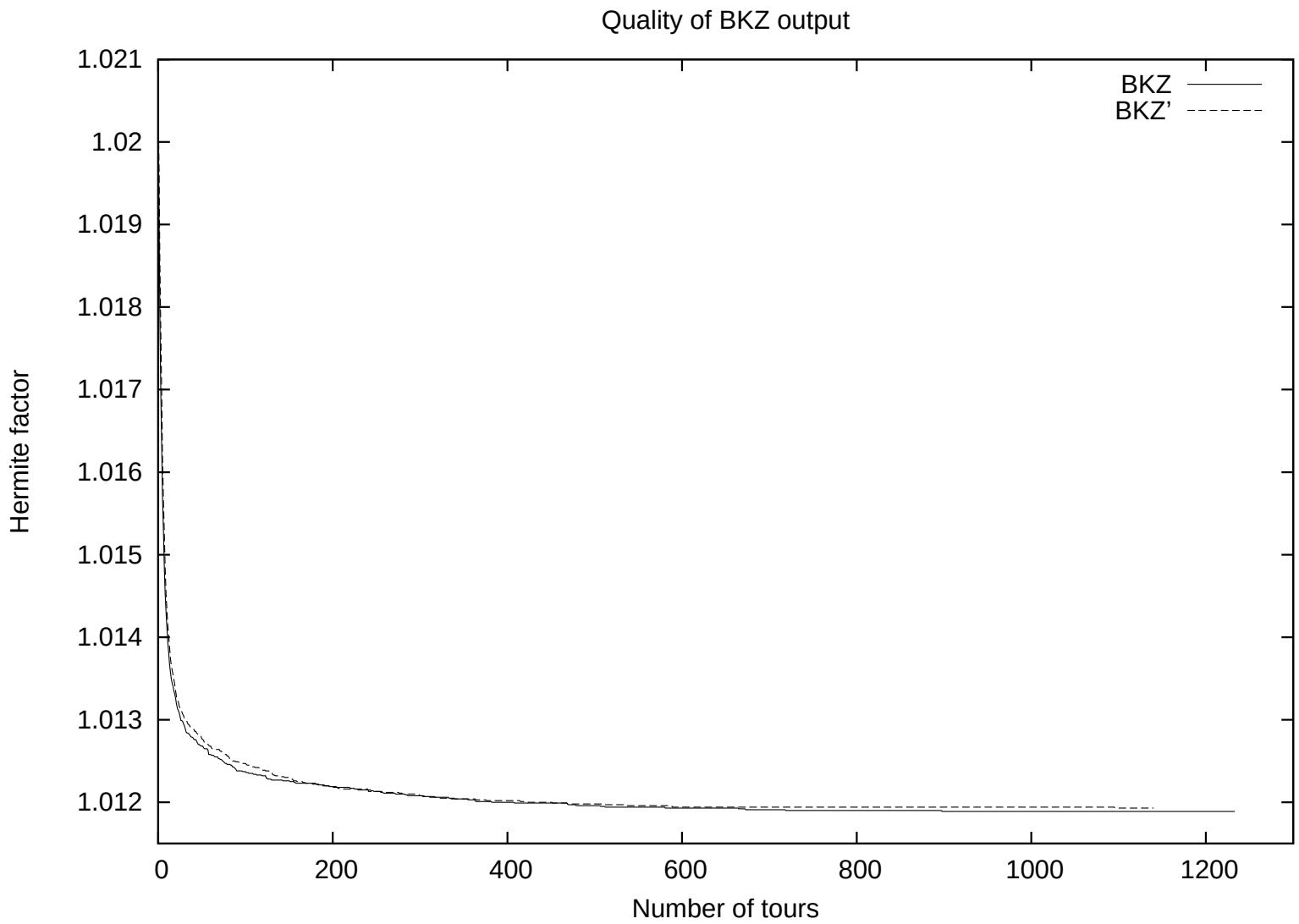
Les expériences tendent à confirmer le modèle

Norme de $\|\mathbf{b}_1\|$ au cours de l'éxécution :



Les expériences tendent à confirmer le modèle

Norme de $\|\mathbf{b}_1\|$ au cours de l'éxécution :



Analyse de BKZ

Peut-on se passer de l'hypothèse de régularité ?

- Pour $k = 2$, $X' = AX + \Gamma$ devient $X' \leq AX + \Gamma$.
- Pour $k > 2$ plus grand, il faut ruser...
  Pour lisser les irrégularités des x_i : $x_i \mapsto y_i = \frac{1}{i} \sum_{j \leq i} x_j$.

Qualité et coût de BKZ

Étant donnée une base $(\mathbf{b}_i)_i$ d'un réseau L en entrée,
BKZ renvoie une base $(\mathbf{c}_i)$ telle que

$$\|\mathbf{b}_1\| \leq 2 \cdot k^{\frac{n-k}{k-1}+\frac{3}{2}} \cdot (\det L)^{1/n}$$

en temps $O(\frac{n^3}{k^2} \log\log \|B\|)$ appels à HKZ_k.

$\Rightarrow$ Le coût est quasi-linéaire en $\log \|B\|$.

Analyse de BKZ

Peut-on se passer de l'hypothèse de régularité ?

- Pour $k = 2$, $X' = AX + \Gamma$ devient $X' \leq AX + \Gamma$.
- Pour $k > 2$ plus grand, il faut ruser...
  Pour lisser les irrégularités des x_i : $x_i \mapsto y_i = \frac{1}{i} \sum_{j \leq i} x_j$.

Qualité et coût de BKZ

Étant donnée une base $(\mathbf{b}_i)_i$ d'un réseau L en entrée,
BKZ renvoie une base $(\mathbf{c}_i)$ telle que

$$\|\mathbf{b}_1\| \leq 2 \cdot k^{\frac{n-k}{k-1}+\frac{3}{2}} \cdot (\det L)^{1/n}$$

en temps $O(\frac{n^3}{k^2} \log\log \|B\|)$ appels à HKZ_k.

$\Rightarrow$ Le coût est quasi-linéaire en $\log \|B\|$.

Analyse de BKZ

Peut-on se passer de l'hypothèse de régularité ?

- Pour $k = 2$, $X' = AX + \Gamma$ devient $X' \leq AX + \Gamma$.
- Pour $k > 2$ plus grand, il faut ruser...
  Pour lisser les irrégularités des x_i : $x_i \mapsto y_i = \frac{1}{i} \sum_{j \leq i} x_j$.

Qualité et coût de BKZ

Étant donnée une base $(\mathbf{b}_i)_i$ d'un réseau L en entrée,
BKZ renvoie une base $(\mathbf{c}_i)$ telle que

$$\|\mathbf{b}_1\| \leq 2 \cdot k^{\frac{n-k}{k-1}+\frac{3}{2}} \cdot (\det L)^{1/n}$$

en temps $O(\frac{n^3}{k^2} \log\log \|B\|)$ appels à HKZ_k.

$\Rightarrow$ Le coût est quasi-linéaire en $\log \|B\|$.

Réduction de réseaux, l'état de l'art

	HKZ	BKZ_k	$\mathrm{LLL} \approx \mathrm{BKZ}_2$
$\|\mathbf{b}_1\|/(\det L)^{\frac{1}{n}}$	$\sqrt{n}$	$\simeq k^{\frac{n}{2k}}$	$2^{O(n)}$
Time*	$2^{O(n)}$	$2^{O(k)} \times \mathrm{Poly}(n)$	$\mathrm{Poly}(n)$

*En négligeant les coûts arithmétiques.

Compromis qualité-temps (à facteurs polynomiaux près)

Temps $2^{O(k)} \Longrightarrow$ facteur d'approx. $\gamma = k^{O(n/k)}$

ou, de façon équivalente

Obtenir un facteur d'approx. γ coûte $\left(1 + \frac{n}{\log \gamma}\right)^{O\left(1+\frac{n}{\log \gamma}\right)}$.

Conséquence :

On peut résoudre SVP_γ en temps poly, pour $\gamma = 2^{O(\frac{n \log\log n}{\log n})}$.

Réduction de réseaux, l'état de l'art

	HKZ	BKZ_k	$\mathrm{LLL} \approx \mathrm{BKZ}_2$
$\|\mathbf{b}_1\|/(\det L)^{\frac{1}{n}}$	$\sqrt{n}$	$\simeq k^{\frac{n}{2k}}$	$2^{O(n)}$
Time*	$2^{O(n)}$	$2^{O(k)} \times \mathrm{Poly}(n)$	$\mathrm{Poly}(n)$

*En négligeant les coûts arithmétiques.

Compromis qualité-temps (à facteurs polynomiaux près)

Temps $2^{O(k)} \implies$ facteur d'approx. $\gamma = k^{O(n/k)}$

ou, de façon équivalente

Obtenir un facteur d'approx. γ coûte $\left(1 + \frac{n}{\log \gamma}\right)^{O\left(1+\frac{n}{\log \gamma}\right)}$.

Conséquence :

On peut résoudre SVP_γ en temps poly, pour $\gamma = 2^{O(\frac{n \log\log n}{\log n})}$.

Réduction de réseaux, l'état de l'art

	HKZ	BKZ_k	$\mathrm{LLL} \approx \mathrm{BKZ}_2$
$\|\mathbf{b}_1\|/(\det L)^{\frac{1}{n}}$	$\sqrt{n}$	$\simeq k^{\frac{n}{2k}}$	$2^{O(n)}$
Time*	$2^{O(n)}$	$2^{O(k)} \times \mathrm{Poly}(n)$	$\mathrm{Poly}(n)$

*En négligeant les coûts arithmétiques.

Compromis qualité-temps (à facteurs polynomiaux près)

Temps $2^{O(k)} \Longrightarrow$ facteur d'approx. $\gamma = k^{O(n/k)}$

ou, de façon équivalente

Obtenir un facteur d'approx. γ coûte $\left(1 + \frac{n}{\log \gamma}\right)^{O\left(1+\frac{n}{\log \gamma}\right)}$.

Conséquence :

On peut résoudre SVP_γ en temps poly, pour $\gamma = 2^{O(\frac{n \log\log n}{\log n})}$.

Réduction de réseaux, l'état de l'art

	HKZ	BKZ_k	$\mathrm{LLL} \approx \mathrm{BKZ}_2$
$\|\mathbf{b}_1\|/(\det L)^{\frac{1}{n}}$	$\sqrt{n}$	$\simeq k^{\frac{n}{2k}}$	$2^{O(n)}$
Time*	$2^{O(n)}$	$2^{O(k)} \times \mathrm{Poly}(n)$	$\mathrm{Poly}(n)$

*En négligeant les coûts arithmétiques.

Compromis qualité-temps (à facteurs polynomiaux près)

Temps $2^{O(k)} \implies$ facteur d'approx. $\gamma = k^{O(n/k)}$

ou, de façon équivalente

Obtenir un facteur d'approx. γ coûte $\left(1 + \frac{n}{\log \gamma}\right)^{O\left(1+\frac{n}{\log \gamma}\right)}$.

Conséquence :

On peut résoudre SVP_γ en temps poly, pour $\gamma = 2^{O(\frac{n \log\log n}{\log n})}$.

Réduction de réseaux, l'état de l'art

	HKZ	BKZ_k	$\mathrm{LLL} \approx \mathrm{BKZ}_2$
$\|\mathbf{b}_1\|/(\det L)^{\frac{1}{n}}$	$\sqrt{n}$	$\simeq k^{\frac{n}{2k}}$	$2^{O(n)}$
Time*	$2^{O(n)}$	$2^{O(k)} \times \mathrm{Poly}(n)$	$\mathrm{Poly}(n)$

*En négligeant les coûts arithmétiques.

Compromis qualité-temps (à facteurs polynomiaux près)

Temps $2^{O(k)} \implies$ facteur d'approx. $\gamma = k^{O(n/k)}$

ou, de façon équivalente

Obtenir un facteur d'approx. γ coûte $\left(1 + \frac{n}{\log \gamma}\right)^{O\left(1+\frac{n}{\log \gamma}\right)}$.

Conséquence :

On peut résoudre SVP_γ en temps poly, pour $\gamma = 2^{O(\frac{n \log\log n}{\log n})}$.

Plan du cours

La cryptographie reposant sur les réseaux

Principe général

- Clé secrète : une base très courte d'un réseau
- Clé publique : Une base non-réduite du même réseau
- Sécurité : héritée de la difficulté de SVP_γ et ses variantes.

Pourquoi s'y intéresse-t-on ?

- Plus sûr : sécurité reposant sur des hypothèses de difficulté pire-cas
- Plus sûre : semble résister aux calculs quantiques
- Plus efficace : pas d'exponentiations modulaires
- Plus efficace : opérations parallélisables
- Plus puissante : chiffrement totalement homomorphe

BKZ fournit la meilleure attaque générique connue

La cryptographie reposant sur les réseaux

Principe général

- Clé secrète : une base très courte d'un réseau
- Clé publique : Une base non-réduite du même réseau
- Sécurité : héritée de la difficulté de SVP_γ et ses variantes.

Pourquoi s'y intéresse-t-on ?

- Plus sûr : sécurité reposant sur des hypothèses de difficulté pire-cas
- Plus sûre : semble résister aux calculs quantiques
- Plus efficace : pas d'exponentiations modulaires
- Plus efficace : opérations parallélisables
- Plus puissante : chiffrement totalement homomorphe

BKZ fournit la meilleure attaque générique connue

La cryptographie reposant sur les réseaux

Principe général

- Clé secrète : une base très courte d'un réseau
- Clé publique : Une base non-réduite du même réseau
- Sécurité : héritée de la difficulté de SVP_γ et ses variantes.

Pourquoi s'y intéresse-t-on ?

- Plus sûr : sécurité reposant sur des hypothèses de difficulté pire-cas
- Plus sûre : semble résister aux calculs quantiques
- Plus efficace : pas d'exponentiations modulaires
- Plus efficace : opérations parallélisables
- Plus puissante : chiffrement totalement homomorphe

BKZ fournit la meilleure attaque générique connue

La cryptographie reposant sur les réseaux

Principe général

- Clé secrète : une base très courte d'un réseau
- Clé publique : Une base non-réduite du même réseau
- Sécurité : héritée de la difficulté de SVP_γ et ses variantes.

Pourquoi s'y intéresse-t-on ?

- Plus sûr : sécurité reposant sur des hypothèses de difficulté pire-cas
- Plus sûre : semble résister aux calculs quantiques
- Plus efficace : pas d'exponentiations modulaires
- Plus efficace : opérations parallélisables
- Plus puissante : chiffrement totalement homomorphe

BKZ fournit la meilleure attaque générique connue

La cryptographie reposant sur les réseaux

Principe général

- Clé secrète : une base très courte d'un réseau
- Clé publique : Une base non-réduite du même réseau
- Sécurité : héritée de la difficulté de SVP_γ et ses variantes.

Pourquoi s'y intéresse-t-on ?

- Plus sûr : sécurité reposant sur des hypothèses de difficulté pire-cas
- Plus sûre : semble résister aux calculs quantiques
- Plus efficace : pas d'exponentiations modulaires
- Plus efficace : opérations parallélisables
- Plus puissante : chiffrement totalement homomorphe

BKZ fournit la meilleure attaque générique connue

Des codes linéaires aux réseaux

- Un code linéaire C sur $\mathbb{Z}_p := \mathbb{Z}/p\mathbb{Z}$ pour p premier est un sous-espace vectoriel d'un $\mathbb{Z}_p^n$.
- Il existe une matrice génératrice $G \in \mathbb{Z}_p^{n \times k}$, avec $k = \dim C$, t.q. :

$$C = G \cdot \mathbb{Z}_p^k = \{G\mathbf{s} : \mathbf{s} \in \mathbb{Z}_p^k\}.$$

Construction A

Soit $C \subseteq \mathbb{Z}_p^n$ un code linéaire de dimension k.
La construction A associe à C le réseau :

$$L(C) = C + p\mathbb{Z}^n = \left\{\mathbf{x} \in \mathbb{Z}^n : \exists \mathbf{s} \in \mathbb{Z}_p^k,\ \mathbf{x} = G \cdot \mathbf{s} \bmod p\right\}.$$

Des codes linéaires aux réseaux

- Un code linéaire C sur $\mathbb{Z}_p := \mathbb{Z}/p\mathbb{Z}$ pour p premier est un sous-espace vectoriel d'un $\mathbb{Z}_p^n$.
- Il existe une matrice génératrice $G \in \mathbb{Z}_p^{n \times k}$, avec $k = \dim C$, t.q. :

$$C = G \cdot \mathbb{Z}_p^k = \{G\mathbf{s} : \mathbf{s} \in \mathbb{Z}_p^k\}.$$

Construction A

Soit $C \subseteq \mathbb{Z}_p^n$ un code linéaire de dimension k.
La construction A associe à C le réseau :

$$L(C) = C + p\mathbb{Z}^n = \left\{\mathbf{x} \in \mathbb{Z}^n : \exists \mathbf{s} \in \mathbb{Z}_p^k,\ \mathbf{x} = G \cdot \mathbf{s} \bmod p\right\}.$$

Propriétés de la construction A

$$L(C) = C + p\mathbb{Z}^n = \{\mathbf{x} \in \mathbb{Z}^n : \exists \mathbf{s} \in \mathbb{Z}_p^k,\ \mathbf{x} = G \cdot \mathbf{s} \bmod p\}.$$

- $p\mathbb{Z}^n \subseteq L(C) \subseteq \mathbb{Z}^n$. En particulier, $\dim(L(C)) = n$.
- Une base de $L(C)$ est obtenue en calculant la forme normale de Hermite de la matrice $[G|p \cdot Id_n]$.

Déterminant :

- HNF $\Rightarrow$ $\mathbb{Z}^n/L(C)$ est fini et $|\mathbb{Z}^n/L(C)| = \det(L(C))$.
- $\mathbb{Z}^n/L(C) \cong \mathbb{Z}_p^n/C$ implique : $\det(L(C)) = p^{n-k}$.

Minimum : Minkowski $\Rightarrow$ $\lambda_1(L(C)) \leq \sqrt{n} \cdot p^{1-k/n}$.

Propriétés de la construction A

$$L(C) = C + p\mathbb{Z}^n = \{\mathbf{x} \in \mathbb{Z}^n : \exists \mathbf{s} \in \mathbb{Z}_p^k,\ \mathbf{x} = G \cdot \mathbf{s} \bmod p\}.$$

- $p\mathbb{Z}^n \subseteq L(C) \subseteq \mathbb{Z}^n$. En particulier, $\dim(L(C)) = n$.
- Une base de $L(C)$ est obtenue en calculant la forme normale de Hermite de la matrice $[G|p \cdot Id_n]$.

Déterminant :

- HNF $\Rightarrow$ $\mathbb{Z}^n/L(C)$ est fini et $|\mathbb{Z}^n/L(C)| = \det(L(C))$.
- $\mathbb{Z}^n/L(C) \cong \mathbb{Z}_p^n/C$ implique : $\det(L(C)) = p^{n-k}$.

Minimum : Minkowski $\Rightarrow$ $\lambda_1(L(C)) \leq \sqrt{n} \cdot p^{1-k/n}$.

Propriétés de la construction A

$$L(C) = C + p\mathbb{Z}^n = \{\mathbf{x} \in \mathbb{Z}^n : \exists \mathbf{s} \in \mathbb{Z}_p^k,\ \mathbf{x} = G \cdot \mathbf{s} \bmod p\}.$$

- $p\mathbb{Z}^n \subseteq L(C) \subseteq \mathbb{Z}^n$. En particulier, $\dim(L(C)) = n$.
- Une base de $L(C)$ est obtenue en calculant la forme normale de Hermite de la matrice $[G|p \cdot Id_n]$.

Déterminant :

- HNF $\Rightarrow$ $\mathbb{Z}^n/L(C)$ est fini et $|\mathbb{Z}^n/L(C)| = \det(L(C))$.
- $\mathbb{Z}^n/L(C) \cong \mathbb{Z}_p^n/C$ implique : $\det(L(C)) = p^{n-k}$.

Minimum : Minkowski $\Rightarrow$ $\lambda_1(L(C)) \leq \sqrt{n} \cdot p^{1-k/n}$.

Le problème SIS [Ajt96]

Le problème *Small Integer Solution*

Étant donnée A choisie uniformément dans $\mathbb{Z}_p^{m \times n}$, trouver $\mathbf{x} \in \mathbb{Z}^m$ t.q.

$$0 < \|\mathbf{x}\| \leq \beta \quad \text{et} \quad \mathbf{x}^t \cdot A = \mathbf{0} \mod p.$$

SIS : trouver des vecteurs courts dans des réseaux issus de la construction A (appliquée au dual/orthogonal du code défini par A)

Difficulté de SIS $\gamma \approx \sqrt{n} \cdot \beta$

Tout algorithme efficace résolvant SIS_β avec probabilité non-negligeable permet de construire un algorithme efficace résolvant GapSVP_γ

- Fonctions de hachage [Ajt96,LyMi08,PeRo08]
- Signatures digitales [GePeVa08,Boy10,Lyu12]

Le problème SIS [Ajt96]

Le problème *Small Integer Solution*

Étant donnée A choisie uniformément dans $\mathbb{Z}_p^{m \times n}$, trouver $\mathbf{x} \in \mathbb{Z}^m$ t.q.

$$0 < \|\mathbf{x}\| \leq \beta \quad \text{et} \quad \mathbf{x}^t \cdot A = \mathbf{0} \mod p.$$

SIS : trouver des vecteurs courts dans des réseaux issus de la construction A (appliquée au dual/orthogonal du code défini par A)

Difficulté de SIS $\gamma \approx \sqrt{n} \cdot \beta$

Tout algorithme efficace résolvant SIS_β avec probabilité non-negligeable permet de construire un algorithme efficace résolvant GapSVP_γ

- Fonctions de hachage [Ajt96,LyMi08,PeRo08]
- Signatures digitales [GePeVa08,Boy10,Lyu12]

Le problème SIS [Ajt96]

Le problème *Small Integer Solution*

Étant donnée A choisie uniformément dans $\mathbb{Z}_p^{m\times n}$, trouver $\mathbf{x} \in \mathbb{Z}^m$ t.q.

$$0 < \|\mathbf{x}\| \leq \beta \quad \text{et} \quad \mathbf{x}^t \cdot A = \mathbf{0} \mod p.$$

SIS : trouver des vecteurs courts dans des réseaux issus de la construction A (appliquée au dual/orthogonal du code défini par A)

Difficulté de SIS $\gamma \approx \sqrt{n} \cdot \beta$

Tout algorithme efficace résolvant SIS_β avec probabilité non-negligeable permet de construire un algorithme efficace résolvant GapSVP_γ

- Fonctions de hachage [Ajt96,LyMi08,PeRo08]
- Signatures digitales [GePeVa08,Boy10,Lyu12]

Attaquer SIS avec la réduction de réseaux

Étant donnée A choisie uniformément dans $\mathbb{Z}_p^{m\times n}$, trouver $\mathbf{x}\in\mathbb{Z}^m$ t.q. :
$$0 < \|\mathbf{x}\| \leq \beta \quad \text{et} \quad \mathbf{x}^t \cdot A = \mathbf{0} \mod p.$$

Vecteur non-nul et court dans $\Lambda^\perp(A) = \{\mathbf{x}\in\mathbb{Z}^m : \mathbf{x}^t\cdot A = \mathbf{0}\ [p]\}$:

$$\det(\Lambda^\perp(A)) \approx p^{\frac{n}{m}} \quad \text{(Le code a dim } n \text{ avec proba. proche de 1)}$$

Réduction de réseaux dans L, avec coût $T = 2^{O(t)}$

$$t^{O(\frac{m}{t})}\cdot p^{\frac{n}{m}} \quad \text{doit être} \quad \leq \beta$$

On peut optimiser sur le choix de $m' \leq m$...

$\Rightarrow$ Prendre le plus petit t t.q. : $\min_{m'\leq m}\left(t^{O(\frac{m'}{t})}\cdot p^{\frac{n}{m'}}\right) \leq \beta$

$$m' \approx \min\left(\sqrt{tn\log p}\ ,\ m\right) \Rightarrow t \approx \frac{n\log p}{\log^2\beta} \quad \text{si } m \text{ est grand}$$

Attaquer SIS avec la réduction de réseaux

Étant donnée A choisie uniformément dans $\mathbb{Z}_p^{m\times n}$, trouver $\mathbf{x}\in\mathbb{Z}^m$ t.q. :
$$0 < \|\mathbf{x}\| \leq \beta \quad \text{et} \quad \mathbf{x}^t \cdot A = \mathbf{0} \mod p.$$

Vecteur non-nul et court dans $\Lambda^\perp(A) = \{\mathbf{x}\in\mathbb{Z}^m : \mathbf{x}^t\cdot A = \mathbf{0}\ [p]\}$:

$$\det(\Lambda^\perp(A)) \approx p^{\frac{n}{m}} \quad \text{(Le code a dim } n \text{ avec proba. proche de 1)}$$

Réduction de réseaux dans L, avec coût $T = 2^{O(t)}$

$$t^{O(\frac{m}{t})}\cdot p^{\frac{n}{m}} \quad \text{doit être} \quad \leq \beta$$

On peut optimiser sur le choix de $m' \leq m$...

$\Rightarrow$ Prendre le plus petit t t.q. : $\min_{m'\leq m}\left(t^{O(\frac{m'}{t})}\cdot p^{\frac{n}{m'}}\right) \leq \beta$

$$m' \approx \min\left(\sqrt{tn\log p}\ ,\ m\right) \Rightarrow t \approx \frac{n\log p}{\log^2\beta} \quad \text{si } m \text{ est grand}$$

Attaquer SIS avec la réduction de réseaux

Étant donnée A choisie uniformément dans $\mathbb{Z}_p^{m\times n}$, trouver $\mathbf{x}\in\mathbb{Z}^m$ t.q. :
$$0 < \|\mathbf{x}\| \leq \beta \quad \text{et} \quad \mathbf{x}^t \cdot A = \mathbf{0} \mod p.$$

Vecteur non-nul et court dans $\Lambda^\perp(A) = \{\mathbf{x}\in\mathbb{Z}^m : \mathbf{x}^t\cdot A = \mathbf{0}\ [p]\}$:

$$\det(\Lambda^\perp(A)) \approx p^{\frac{n}{m}} \quad \text{(Le code a dim } n \text{ avec proba. proche de 1)}$$

Réduction de réseaux dans L, avec coût $T = 2^{O(t)}$

$$t^{O(\frac{m}{t})}\cdot p^{\frac{n}{m}} \quad \text{doit être} \quad \leq \beta$$

On peut optimiser sur le choix de $m' \leq m$...

$\Rightarrow$ Prendre le plus petit t t.q. : $\min_{m'\leq m}\left(t^{O(\frac{m'}{t})}\cdot p^{\frac{n}{m'}}\right) \leq \beta$

$$m' \approx \min\left(\sqrt{tn\log p}\ ,\ m\right) \Rightarrow t \approx \frac{n\log p}{\log^2\beta} \quad \text{si } m \text{ est grand}$$

Attaquer SIS avec la réduction de réseaux

Étant donnée A choisie uniformément dans $\mathbb{Z}_p^{m\times n}$, trouver $\mathbf{x}\in\mathbb{Z}^m$ t.q. :
$$0 < \|\mathbf{x}\| \leq \beta \quad \text{et} \quad \mathbf{x}^t \cdot A = \mathbf{0} \mod p.$$

Vecteur non-nul et court dans $\Lambda^\perp(A) = \{\mathbf{x}\in\mathbb{Z}^m : \mathbf{x}^t\cdot A = \mathbf{0}\ [p]\}$:

$$\det(\Lambda^\perp(A)) \approx p^{\frac{n}{m}} \quad \text{(Le code a dim } n \text{ avec proba. proche de 1)}$$

Réduction de réseaux dans L, avec coût $T = 2^{O(t)}$

$$t^{O(\frac{m}{t})}\cdot p^{\frac{n}{m}} \quad \text{doit être} \quad \leq \beta$$

On peut optimiser sur le choix de $m' \leq m$...

$\Rightarrow$ Prendre le plus petit t t.q. : $\min_{m'\leq m}\left(t^{O(\frac{m'}{t})}\cdot p^{\frac{n}{m'}}\right) \leq \beta$

$$m' \approx \min\left(\sqrt{tn\log p}\ ,\ m\right) \Rightarrow t \approx \frac{n\log p}{\log^2\beta} \quad \text{si } m \text{ est grand}$$

Le problème LWE [Reg05]

Soit p premier, $n \geq 1$, $\alpha < 1$. Soit $\mathbf{s} \in \mathbb{Z}_p^n$.
Soit $D_{\mathbf{s},\alpha}$ la distribution :

$$(\mathbf{a}; \langle \mathbf{a}, \mathbf{s} \rangle + e \ [p]), \quad \text{avec} \quad \mathbf{a} \hookleftarrow U(\mathbb{Z}_p^n), \ e \hookleftarrow \nu_{\alpha p}$$

LWE_α

Soit $\mathbf{s} \hookleftarrow U(\mathbb{Z}_p^n)$. Distinguer $D_{\mathbf{s},\alpha}$ et $U(\mathbb{Z}_p^n \times [0, p[)$

LWE : résoudre BDD pour des réseaux issus de la construction A

Difficulté de LWE $\gamma \approx \sqrt{n}/\alpha$

Tout algorithme efficace résolvant LWE_α avec avantage non-négligeable fournit un algorithme efficace (quantique) pour GapSVP_γ

- Chiffrement [Reg05]
- Chiffrement totalement homomorphe [BraVai11]

Le problème LWE [Reg05]

Soit p premier, $n \geq 1$, $\alpha < 1$. Soit $\mathbf{s} \in \mathbb{Z}_p^n$.
Soit $D_{\mathbf{s},\alpha}$ la distribution :

$$(\mathbf{a}; \langle \mathbf{a}, \mathbf{s} \rangle + e \ [p]), \quad \text{avec} \quad \mathbf{a} \hookleftarrow U(\mathbb{Z}_p^n), \ e \hookleftarrow \nu_{\alpha p}$$

LWE_α

Soit $\mathbf{s} \hookleftarrow U(\mathbb{Z}_p^n)$. Distinguer $D_{\mathbf{s},\alpha}$ et $U(\mathbb{Z}_p^n \times [0, p[)$

LWE : résoudre BDD pour des réseaux issus de la construction A

Difficulté de LWE $\gamma \approx \sqrt{n}/\alpha$

Tout algorithme efficace résolvant LWE_α avec avantage non-négligeable fournit un algorithme efficace (quantique) pour GapSVP_γ

- Chiffrement [Reg05]
- Chiffrement totalement homomorphe [BraVai11]

Le problème LWE [Reg05]

Soit p premier, $n \geq 1$, $\alpha < 1$. Soit $\mathbf{s} \in \mathbb{Z}_p^n$.
Soit $D_{\mathbf{s},\alpha}$ la distribution :

$$(\mathbf{a}; \langle \mathbf{a}, \mathbf{s} \rangle + e \ [p]), \quad \text{avec} \quad \mathbf{a} \hookleftarrow U(\mathbb{Z}_p^n), \ e \hookleftarrow \nu_{\alpha p}$$

LWE_α

Soit $\mathbf{s} \hookleftarrow U(\mathbb{Z}_p^n)$. Distinguer $D_{\mathbf{s},\alpha}$ et $U(\mathbb{Z}_p^n \times [0, p[)$

LWE : résoudre BDD pour des réseaux issus de la construction A

Difficulté de LWE $\gamma \approx \sqrt{n}/\alpha$

Tout algorithme efficace résolvant LWE_α avec avantage non-négligeable fournit un algorithme efficace (quantique) pour GapSVP_γ

- Chiffrement [Reg05]
- Chiffrement totalement homomorphe [BraVai11]

Le problème LWE [Reg05]

Soit p premier, $n \geq 1$, $\alpha < 1$. Soit $\mathbf{s} \in \mathbb{Z}_p^n$.
Soit $D_{\mathbf{s},\alpha}$ la distribution :

$$(\mathbf{a}; \langle \mathbf{a}, \mathbf{s} \rangle + e \ [p]), \quad \text{avec} \quad \mathbf{a} \hookleftarrow U(\mathbb{Z}_p^n), \ e \hookleftarrow \nu_{\alpha p}$$

LWE_α

Soit $\mathbf{s} \hookleftarrow U(\mathbb{Z}_p^n)$. Distinguer $D_{\mathbf{s},\alpha}$ et $U(\mathbb{Z}_p^n \times [0, p[)$

LWE : résoudre BDD pour des réseaux issus de la construction A

Difficulté de LWE $\gamma \approx \sqrt{n}/\alpha$

Tout algorithme efficace résolvant LWE_α avec avantage non-négligeable fournit un algorithme efficace (quantique) pour GapSVP_γ

- Chiffrement [Reg05]
- Chiffrement totalement homomorphe [BraVai11]

Attaquer LWE avec la réduction de réseaux

$D_{\mathbf{s},\alpha}$: $(\mathbf{a}; \langle \mathbf{a}, \mathbf{s} \rangle + e \ [p])$ avec $\mathbf{a} \hookleftarrow U(\mathbb{Z}_p^n)$, $e \hookleftarrow \nu_{\alpha p}$

Soit $\mathbf{s} \hookleftarrow U(\mathbb{Z}_p^n)$. Distinguer entre $D_{\mathbf{s},\alpha}$ et $U(\mathbb{Z}_p^n \times [0, p[)$

Soit $(A, \mathbf{b}) \in \mathbb{Z}_p^{m \times n} \times [0, p[^{m \times 1}$ obtenue avec m échantillons.

Pour $\mathbf{x} \in \Lambda^\perp(A)$:

$$\left| \begin{array}{lcll} D_{\mathbf{s},\alpha} & \Rightarrow & \langle \mathbf{x}, \mathbf{b} \rangle & \sim \quad \nu_{\alpha p \|\mathbf{x}\|} \bmod p \\ \text{Unif} & \Rightarrow & \langle \mathbf{x}, \mathbf{b} \rangle & \sim \quad U([0, p[) \end{array} \right.$$

Il suffit de résoudre SIS_β avec $\beta \approx 1/\alpha$

Avec un coût $T = 2^{O(t)}$:

Prendre $m \approx \sqrt{tn \log p}$ permet d'aboutir à $t \approx \frac{n \log p}{\log^2 \alpha}$

Attaquer LWE avec la réduction de réseaux

$D_{\mathbf{s},\alpha}$: $(\mathbf{a}; \langle \mathbf{a}, \mathbf{s} \rangle + e \ [p])$ avec $\mathbf{a} \hookleftarrow U(\mathbb{Z}_p^n)$, $e \hookleftarrow \nu_{\alpha p}$

Soit $\mathbf{s} \hookleftarrow U(\mathbb{Z}_p^n)$. Distinguer entre $D_{\mathbf{s},\alpha}$ et $U(\mathbb{Z}_p^n \times [0, p[)$

Soit $(A, \mathbf{b}) \in \mathbb{Z}_p^{m \times n} \times [0, p[^{m \times 1}$ obtenue avec m échantillons.

$$\text{Pour } \mathbf{x} \in \Lambda^\perp(A) : \left| \begin{array}{lcll} D_{\mathbf{s},\alpha} & \Rightarrow & \langle \mathbf{x}, \mathbf{b} \rangle & \sim \quad \nu_{\alpha p \|\mathbf{x}\|} \bmod p \\ \text{Unif} & \Rightarrow & \langle \mathbf{x}, \mathbf{b} \rangle & \sim \quad U([0, p[) \end{array} \right.$$

Il suffit de résoudre SIS_β avec $\beta \approx 1/\alpha$

Avec un coût $T = 2^{O(t)}$:

Prendre $m \approx \sqrt{tn \log p}$ permet d'aboutir à $t \approx \frac{n \log p}{\log^2 \alpha}$

Attaquer LWE avec la réduction de réseaux

$D_{\mathbf{s},\alpha}$: $(\mathbf{a}; \langle \mathbf{a}, \mathbf{s} \rangle + e \ [p])$ avec $\mathbf{a} \hookleftarrow U(\mathbb{Z}_p^n)$, $e \hookleftarrow \nu_{\alpha p}$

Soit $\mathbf{s} \hookleftarrow U(\mathbb{Z}_p^n)$. Distinguer entre $D_{\mathbf{s},\alpha}$ et $U(\mathbb{Z}_p^n \times [0, p[)$

Soit $(A, \mathbf{b}) \in \mathbb{Z}_p^{m \times n} \times [0, p[^{m \times 1}$ obtenue avec m échantillons.

$$\text{Pour } \mathbf{x} \in \Lambda^\perp(A) : \left| \begin{array}{lcll} D_{\mathbf{s},\alpha} & \Rightarrow & \langle \mathbf{x}, \mathbf{b} \rangle & \sim \quad \nu_{\alpha p \|\mathbf{x}\|} \bmod p \\ \text{Unif} & \Rightarrow & \langle \mathbf{x}, \mathbf{b} \rangle & \sim \quad U([0, p[) \end{array} \right.$$

Il suffit de résoudre SIS_β avec $\beta \approx 1/\alpha$

Avec un coût $T = 2^{O(t)}$:

Prendre $m \approx \sqrt{tn \log p}$ permet d'aboutir à $t \approx \frac{n \log p}{\log^2 \alpha}$

Attaquer LWE avec la réduction de réseaux

$D_{\mathbf{s},\alpha}$: $(\mathbf{a}; \langle \mathbf{a}, \mathbf{s} \rangle + e \ [p])$ avec $\mathbf{a} \hookleftarrow U(\mathbb{Z}_p^n)$, $e \hookleftarrow \nu_{\alpha p}$

Soit $\mathbf{s} \hookleftarrow U(\mathbb{Z}_p^n)$. Distinguer entre $D_{\mathbf{s},\alpha}$ et $U(\mathbb{Z}_p^n \times [0, p[)$

Soit $(A, \mathbf{b}) \in \mathbb{Z}_p^{m \times n} \times [0, p[^{m \times 1}$ obtenue avec m échantillons.

$$\text{Pour } \mathbf{x} \in \Lambda^\perp(A) : \left| \begin{array}{lcll} D_{\mathbf{s},\alpha} & \Rightarrow & \langle \mathbf{x}, \mathbf{b} \rangle & \sim \quad \nu_{\alpha p \|\mathbf{x}\|} \bmod p \\ \text{Unif} & \Rightarrow & \langle \mathbf{x}, \mathbf{b} \rangle & \sim \quad U([0, p[) \end{array} \right.$$

Il suffit de résoudre SIS_β avec $\beta \approx 1/\alpha$

Avec un coût $T = 2^{O(t)}$:

Prendre $m \approx \sqrt{tn \log p}$ permet d'aboutir à $t \approx \frac{n \log p}{\log^2 \alpha}$

Plan du cours

1. L'algorithme BKZ
2. Une application cryptographique
3. **Un LLL hybride numérique-algébrique**
4. Problèmes ouverts

Complexité de LLL et coût en pratique

Borne de complexité de [LLL82]

LLL termine en $O(n^4 \log \|B\| \cdot \mathcal{M}(n \log \|B\|)$ opérations binaires

Avec MAGMA V2.16:

```
> n := 25; B := RMatrixSpace(Integers(),n,n)!0;
> for i:=1 to 25 do
>    B[i][i]:=1; B[i][1]:=RandomBits(2000);
> end for;
> time C := LLL(B:Method:=``Integral'');
```

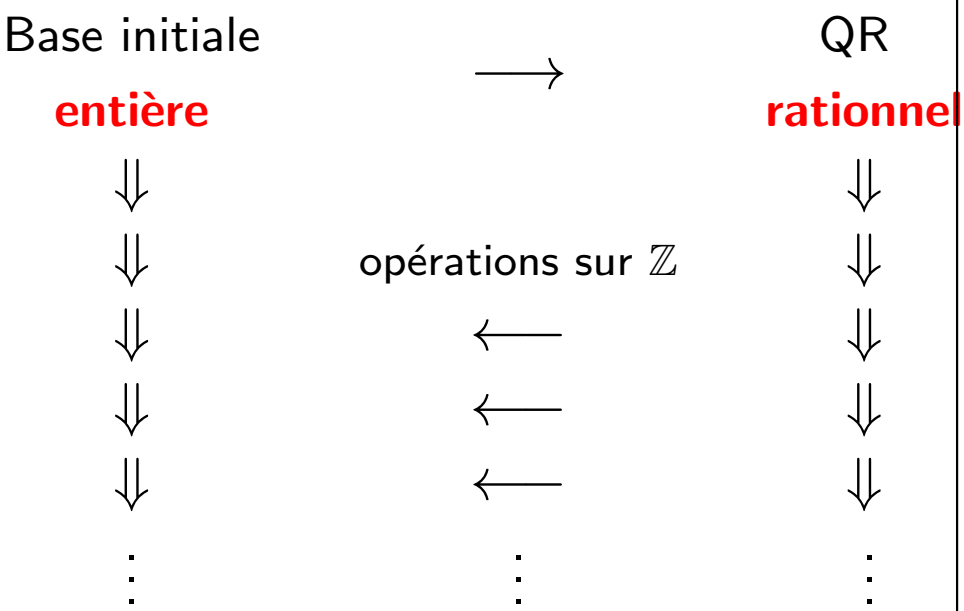
Time: 11.700

```
> time C := LLL(B);
```

Time: 0.240

Les approches exactes et numériques

L'approche exacte



On obtient une base réduite, mais...

QR domine le coût

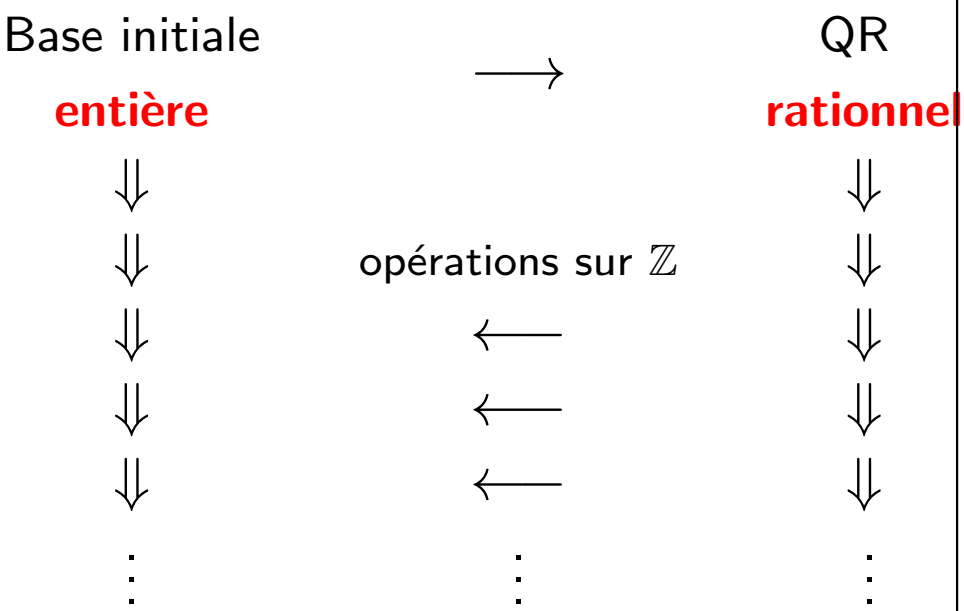
L'approche numérique

Plus rapide... mais

LLL exhibe des annulations : c'est très instable

Les approches exactes et numériques

L'approche exacte



On obtient une base réduite, mais...

QR domine le coût

L'approche numérique

Plus rapide... mais

LLL exhibe des annulations : c'est très instable

Les approches exactes et numériques

L'approche exacte

Base initiale **entière** $\longrightarrow$ QR **rationnel**

$\Downarrow$ $\Downarrow$ $\Downarrow$ $\Downarrow$ $\Downarrow$ $\vdots$

opérations sur $\mathbb{Z}$: $\longleftarrow$ $\longleftarrow$ $\longleftarrow$ $\vdots$

On obtient une base réduite, mais...

QR domine le coût

L'approche numérique

Base initiale **entière** $\longrightarrow$ QR **numérique**

$\Downarrow$ $\Downarrow$ $\Downarrow$ $\Downarrow$ $\Downarrow$ $\vdots$

opérations sur $\mathbb{Z}$: $\longleftarrow$ $\longleftarrow$ $\longleftarrow$ $\vdots$

Plus rapide... mais

LLL exhibe des annulations : c'est très instable

Les approches exactes et numériques

L'approche exacte

Base initiale **entière** $\longrightarrow$ QR **rationnel**

$\Downarrow$ $\Downarrow$ $\Downarrow$ $\Downarrow$ $\Downarrow$ $\vdots$

opérations sur $\mathbb{Z}$ $\longleftarrow$ $\longleftarrow$ $\longleftarrow$ $\vdots$

On obtient une base réduite, mais...

QR domine le coût

L'approche numérique

Base initiale **entière** $\longrightarrow$ QR **numérique**

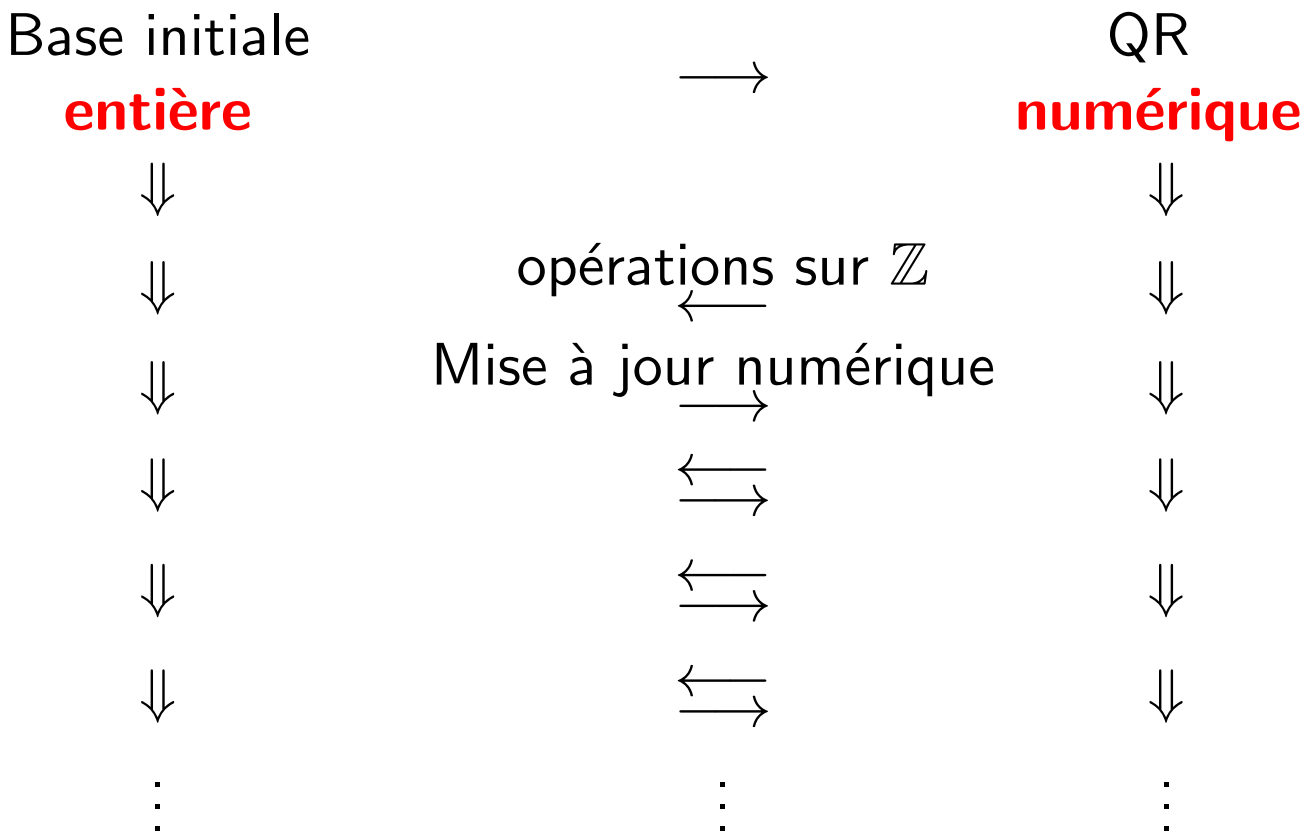
$\Downarrow$ $\Downarrow$ $\Downarrow$ $\Downarrow$ $\Downarrow$ $\vdots$

opérations sur $\mathbb{Z}$ $\longleftarrow$ $\longleftarrow$ $\longleftarrow$ $\vdots$

Plus rapide... mais

LLL exhibe des annulations : c'est très instable

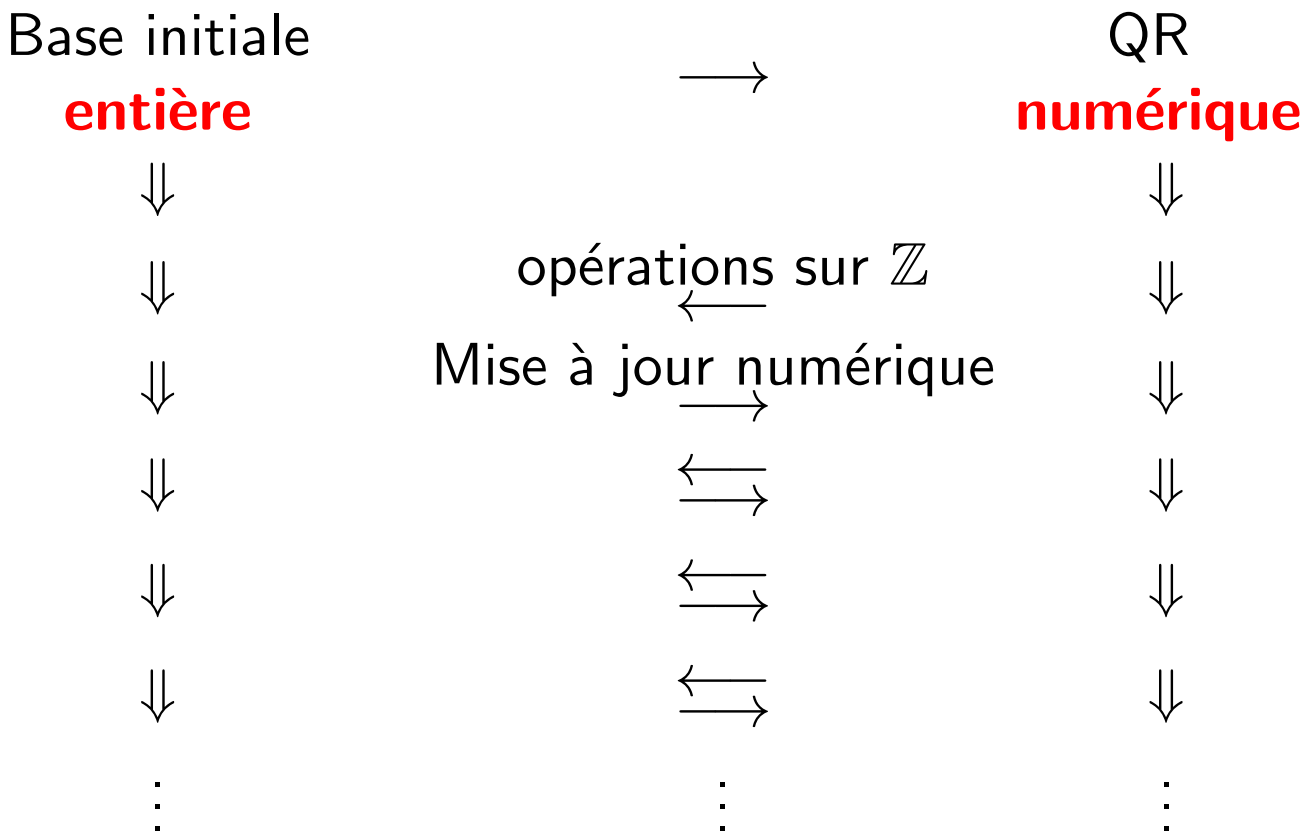
L'approche hybride d'Odlyzko (1982)



Idéalement...

On aurait le meilleur des deux mondes :
Correction de l'approche exacte, efficacité de l'approche numérique

L'approche hybride d'Odlyzko (1982)



Idéalement...

On aurait le meilleur des deux mondes :
Correction de l'approche exacte, efficacité de l'approche numérique

Arithmétique sous-jacente

Nombres flottants: $x_1.x_2x_3\ldots x_p \cdot B^e$, avec :

- p la précision
- B la base, et $x_i \in \{0,\ldots,B-1\}$
- $e \in \mathbb{Z}$ l'exposant

Arithmétique flottante :
$fl(a\ op\ b)$ est le nombre flottant le plus proche de $(a\ op\ b)$,
pour tout $op \in \{+,-,/,\times,\sqrt{\cdot}\}$

Pour tous ces op, on a : $|fl(a\ op\ b) - (a\ op\ b)| \leq 2^{-p} \cdot |a\ op\ b|$.

- L'arithmétique flottante simule les calculs exacts
- Plus p est petit, plus c'est efficace
- En pratique : on aime prendre $p = 53$ (précision machine)

Arithmétique sous-jacente

Nombres flottants: $x_1.x_2x_3\ldots x_p \cdot B^e$, avec :

- p la précision
- B la base, et $x_i \in \{0, \ldots, B-1\}$
- $e \in \mathbb{Z}$ l'exposant

Arithmétique flottante :
$fl(a\ op\ b)$ est le nombre flottant le plus proche de $(a\ op\ b)$,
pour tout $op \in \{+, -, /, \times, \sqrt{\cdot}\}$

Pour tous ces op, on a : $|fl(a\ op\ b) - (a\ op\ b)| \leq 2^{-p} \cdot |a\ op\ b|$.

- L'arithmétique flottante simule les calculs exacts
- Plus p est petit, plus c'est efficace
- En pratique : on aime prendre $p = 53$ (précision machine)

Arithmétique sous-jacente

Nombres flottants: $x_1.x_2x_3\ldots x_p \cdot B^e$, avec :

- p la précision
- B la base, et $x_i \in \{0, \ldots, B-1\}$
- $e \in \mathbb{Z}$ l'exposant

Arithmétique flottante :
$fl(a\ op\ b)$ est le nombre flottant le plus proche de $(a\ op\ b)$,
pour tout $op \in \{+, -, /, \times, \sqrt{\cdot}\cdot\}$

Pour tous ces op, on a : $|fl(a\ op\ b) - (a\ op\ b)| \leq 2^{-p} \cdot |a\ op\ b|$.

- L'arithmétique flottante simule les calculs exacts
- Plus p est petit, plus c'est efficace
- En pratique : on aime prendre $p = 53$ (précision machine)

Arithmétique sous-jacente

Nombres flottants: $x_1.x_2x_3\ldots x_p \cdot B^e$, avec :

- p la précision
- B la base, et $x_i \in \{0, \ldots, B-1\}$
- $e \in \mathbb{Z}$ l'exposant

Arithmétique flottante :
$fl(a\ op\ b)$ est le nombre flottant le plus proche de $(a\ op\ b)$,
pour tout $op \in \{+, -, /, \times, \sqrt{\cdot}\cdot\}$

Pour tous ces op, on a : $|fl(a\ op\ b) - (a\ op\ b)| \leq 2^{-p} \cdot |a\ op\ b|$.

- L'arithmétique flottante simule les calculs exacts
- Plus p est petit, plus c'est efficace
- En pratique : on aime prendre $p = 53$ (précision machine)

Calcul numérique du facteur R

Un sujet étudié depuis longtemps :

- Plusieurs algorithmes sont stables dans le sens inverse : MGS, Givens, Householder, ...

$\Rightarrow$ Le facteur R calculé est le vrai facteur R d'une matrice $B + \Delta B$ avec $\|\Delta B\|$ petit.

- La stabilité inverse peut être combinée avec une analyse de perturbation, pour décrire la stabilité dans le sens direct
- ... à condition que l'entrée soit bien conditionnée

Mais des contraintes inhabituelles :

- Bornes **exactes** requises pour **garantir** correction et terminaison
- On peut se permettre de modifier le format flottant habituel
  - $\rightsquigarrow$ exposants non bornés
  - $\rightsquigarrow$ précision ajustable

Calcul numérique du facteur R

Un sujet étudié depuis longtemps :

- Plusieurs algorithmes sont stables dans le sens inverse : MGS, Givens, Householder, ...

$\Rightarrow$ Le facteur R calculé est le vrai facteur R d'une matrice $B + \Delta B$ avec $\|\Delta B\|$ petit.

- La stabilité inverse peut être combinée avec une analyse de perturbation, pour décrire la stabilité dans le sens direct
- ... à condition que l'entrée soit bien conditionnée

Mais des contraintes inhabituelles :

- Bornes **exactes** requises pour **garantir** correction et terminaison
- On peut se permettre de modifier le format flottant habituel
  - $\rightsquigarrow$ exposants non bornés
  - $\rightsquigarrow$ précision ajustable

Calcul numérique du facteur R

Un sujet étudié depuis longtemps :

- Plusieurs algorithmes sont stables dans le sens inverse : MGS, Givens, Householder, ...

$\Rightarrow$ Le facteur R calculé est le vrai facteur R d'une matrice $B + \Delta B$ avec $\|\Delta B\|$ petit.

- La stabilité inverse peut être combinée avec une analyse de perturbation, pour décrire la stabilité dans le sens direct
- ... à condition que l'entrée soit bien conditionnée

Mais des contraintes inhabituelles :

- Bornes **exactes** requises pour **garantir** correction et terminaison
- On peut se permettre de modifier le format flottant habituel
  - $\rightsquigarrow$ exposants non bornés
  - $\rightsquigarrow$ précision ajustable

A COMPLETER

Plan du cours

1. L'algorithme BKZ
2. Une application cryptographique
3. Un LLL hybride numérique-algébrique
4. **Problèmes ouverts**

A COMPLETER