PMNS - Généralisation via l'algorithme de Babai

Nicolas Méloni

8 décembre 2022

Réduction modulo p

- $a, b \in \mathbb{F}_p, c = a \times b$
- cherche s réduit tel que $c \equiv s \mod p$

Réduction modulo p

- $a, b \in \mathbb{F}_p, c = a \times b$
- cherche s réduit tel que $c \equiv s \mod p$
- *Réduit: plein de 0.*

Réduction modulo p

$$c - c \times \frac{1}{p} \times p = 0$$

Réduction modulo p

$$c - c \times \frac{1}{p} \times p = 0$$

Barret

$$\underbrace{c - \left\lfloor c \times \frac{1}{p} \right\rfloor \times p}_{\leq p}$$

Réduction modulo p

$$c - c \times \frac{1}{p} \times p = 0$$

Barret

$$\underbrace{c - \left\lfloor c \times \frac{1}{p} \right\rfloor \times p}_{\leq p}$$

$$\begin{aligned} & 323434406780 - 308449 \times 1048583 \\ = \; & 323434406780 - 323434377767 \\ = \; & 000000029013 \end{aligned}$$

Réduction modulo p

$$c - c \times \frac{1}{p} \times p = 0$$

Barret

$$\underbrace{c - \left\lfloor c \times \frac{1}{p} \right\rfloor \times p}_{\leq p}$$

Montgomery

$$\underbrace{c - \left(c \times \frac{1}{p} \mod \phi \right) \times p}_{\equiv 0 \mod \phi}$$

$$\begin{aligned} & 323434406780 - 308449 \times 1048583 \\ = \; & 323434406780 - 323434377767 \\ = \; & 000000029013 \end{aligned}$$

Réduction modulo p

$$c - c \times \frac{1}{p} \times p = 0$$

Barret

$$\underbrace{c - \left\lfloor c \times \frac{1}{p} \right\rfloor \times p}_{\leq p}$$

$$\begin{aligned} & 323434406780 - 308449 \times 1048583 \\ = \; & 323434406780 - 323434377767 \\ = \; & 000000029013 \end{aligned}$$

Montgomery

$$\underbrace{c - \left(c \times \frac{1}{p} \mod \phi \right) \times p}_{\equiv 0 \mod \phi}$$

$$\begin{aligned} & 323434406780 - 8474660 \times 1048583 \\ = \; & 323434406780 - 8886384406780 \\ = \; & -8562950000000 \end{aligned}$$

PMNS

Base 2^w

$$a = \sum_{i=0}^{m-1} a_i 2^{wi}$$

$$\forall i \in \{0, \ldots, m-1\}, |a_i| < 2^w$$

$w = 5$

$$825663 = 25(2^5)^3 + 6(2^5)^2 + 9(2^5) + 31$$

PMNS

Base 2^w

$$a = \sum_{i=0}^{m-1} a_i 2^{wi}$$

$$\forall i \in \{0, \ldots, m-1\}, |a_i| < 2^w$$

$w = 5$

$$825663 = 25(2^5)^3 + 6(2^5)^2 + 9(2^5) + 31$$

Base γ

$$a = \sum_{i=0}^{n-1} a_i \gamma^i \mod p$$

$$\forall i \in \{0, \ldots, n-1\}, |a_i| < \rho$$

PMNS

Base 2^w

$$a = \sum_{i=0}^{m-1} a_i 2^{wi}$$

$$\forall i \in \{0, \ldots, m-1\}, |a_i| < 2^w$$

$w = 5$

$$825663 = 25(2^5)^3 + 6(2^5)^2 + 9(2^5) + 31$$

Base γ

$$a = \sum_{i=0}^{n-1} a_i \gamma^i \mod p$$

$$\forall i \in \{0, \ldots, n-1\}, |a_i| < \rho$$

$\gamma = 238019, \rho = 36$

$$\begin{aligned} 825663 &= -9(238019)^4 + 13(238019)^3 \\ &\quad -35(238019)^2 - 25 \mod 1048573 \\ &= -9(1028405) + 13(999523) \\ &\quad -35(742317) - 25 \mod 1048573 \end{aligned}$$

PMNS

Définition

Un tuple $(p, n, \gamma, \rho, E, I) \in \mathbb{Z}^4 \times \mathbb{Z}[X] \times \mathbb{Z} \cup \{\infty\}$, tel que $E(\gamma) \equiv 0 \mod p$ et pour tout $0 \leq v < p$, il existe un polynome $V(X)$ dans $\mathbb{Z}[X]/(E)$ (ou un vecteur $\vec{V} = (v_0, ..., v_{n-1}) \in \mathbb{Z}^n$) tel que:

$$V(\gamma) \equiv \sum_{i=0}^{n-1} v_i \gamma^i \equiv v \mod p \text{ et } \|\vec{V}\|_I < \rho.$$

Exemple

- $p = 1048573, n = 5, \gamma = 238019, \rho = 36, E = X^5 - 2$ alors $\mathcal{S} = (p, n, \gamma, \rho, E, \infty)$ est un PMNS.
- $A = -9X^4 + 13X^3 - 35X^2 - 25$ et $B = 24X^4 - 32X^3 + 21X^2 - 2X - 16$

Exemple

- $p = 1048573, n = 5, \gamma = 238019, \rho = 36, E = X^5 - 2$ alors $\mathcal{S} = (p, n, \gamma, \rho, E, \infty)$ est un PMNS.
- $A = -9X^4 + 13X^3 - 35X^2 - 25$ et $B = 24X^4 - 32X^3 + 21X^2 - 2X - 16$
- (multiplication)

$$\begin{aligned} AB &= -216X^8 + 600X^7 - 1445X^6 + 1411X^5 \\ &\quad -1217X^4 + 662X^3 + 35X^2 + 50X + 400 \end{aligned}$$

Exemple

- $p = 1048573, n = 5, \gamma = 238019, \rho = 36, E = X^5 - 2$ alors $\mathcal{S} = (p, n, \gamma, \rho, E, \infty)$ est un PMNS.
- $A = -9X^4 + 13X^3 - 35X^2 - 25$ et $B = 24X^4 - 32X^3 + 21X^2 - 2X - 16$
- (multiplication)

$$\begin{aligned} AB &= -216X^8 + 600X^7 - 1445X^6 + 1411X^5 \\ &\quad -1217X^4 + 662X^3 + 35X^2 + 50X + 400 \end{aligned}$$

- (réduction externe)

$$\begin{aligned} C &= AB \mod E \\ &= -1217X^4 + 230X^3 + 1235X^2 - 2840X + 3222 \end{aligned}$$

Exemple

- $p = 1048573, n = 5, \gamma = 238019, \rho = 36, E = X^5 - 2$ alors $\mathcal{S} = (p, n, \gamma, \rho, E, \infty)$ est un PMNS.
- $A = -9X^4 + 13X^3 - 35X^2 - 25$ et $B = 24X^4 - 32X^3 + 21X^2 - 2X - 16$
- (multiplication)

$$\begin{aligned} AB &= -216X^8 + 600X^7 - 1445X^6 + 1411X^5 \\ &\quad -1217X^4 + 662X^3 + 35X^2 + 50X + 400 \end{aligned}$$

- (réduction externe)

$$\begin{aligned} C &= AB \mod E \\ &= -1217X^4 + 230X^3 + 1235X^2 - 2840X + 3222 \end{aligned}$$

- (réduction interne)

$$\begin{aligned} S &= C - (-1211X^4 + 228X^3 + 1225X^2 - 2832X + 3236) \\ &= -6X^4 + 2X^3 + 10X^2 - 8X - 14 \\ &\quad (avec -1211\gamma^4 + 228\gamma^3 + 1225\gamma^2 - 2832\gamma + 3236 \equiv 0 \mod p) \end{aligned}$$

Réduction par l'algorithme Montgomery-like

Data: $\mathcal{B} = (p, n, \gamma, \rho, E, \infty)$, $V \in \mathbb{Z}_n[X]$, $M \in \mathcal{B}$ such that, $M(\gamma) \equiv 0 \mod p$, $\phi \in \mathbb{N} - \{0\}$ and $M' = -M^{-1} \mod (E, \phi)$.
Result: $S(\gamma) = V(\gamma)\phi^{-1} \mod p$
$Q \leftarrow V \times M' \mod (E, \phi)$;
$T \leftarrow Q \times M \mod E$;
$S \leftarrow (V + T)/\phi$;
return S;

Remarque

La justesse de l'algorithme repose sur le calcul préalable d'un polynome M inversible modulo E et modulo ϕ. Dans certain cas, la recherche d'un tel M nécessite d'explorer un espace de taille exponentielle en n.

Réseau associé à un PMNS

Soit $\mathcal{B} = (p, n, \gamma, \rho, E, l)$ un PMNS. Le réseau associé à $\mathcal{B}$ est l'ensemble

$$\mathcal{L}(\mathcal{B}) = \left\{ (v_0, \ldots, v_{n-1}) \in \mathbb{Z}^n : \sum_{i=0}^{n-1} v_i \gamma^i \equiv 0 \mod p \right\}.$$

Il est engendré par les lignes de la matrice

$$G = \begin{pmatrix} p & 0 & 0 & \ldots & 0 \\ -\gamma & 1 & 0 & \ldots & 0 \\ -\gamma^2 & 0 & 1 & \ldots & 0 \\ \vdots & & \ddots & & \vdots \\ -\gamma^{n-2} & 0 & \ldots & 1 & 0 \\ -\gamma^{n-1} & 0 & \ldots & 0 & 1 \end{pmatrix}.$$

On peut utiliser LLL pour obtenir une base réduite.

Réseau associé à un PMNS

Réduction interne

- $C = -1217X^4 + 230X^3 + 1235X^2 - 2840X + 3222$
- $C' = -1211X^4 + 228X^3 + 1225X^2 - 2832X + 3236$
  $C'(\gamma) \equiv 0 \mod p$
- $S = C - C' = -6X^4 + 2X^3 + 10X^2 - 8X - 14$

Réseau associé à un PMNS

Réduction interne

- $C = (3222, -2840, 1235, 230, -1217)$
- $C' = -1211X^4 + 228X^3 + 1225X^2 - 2832X + 3236$
  $C'(\gamma) \equiv 0 \mod p$
- $S = C - C' = -6X^4 + 2X^3 + 10X^2 - 8X - 14$

Réseau associé à un PMNS

Réduction interne

- $C = (3222, -2840, 1235, 230, -1217)$
- $C' = (3236, -2832, 1225, 228, -1211)$
  $C' \in \mathcal{L}(\mathcal{B})$
- $S = C - C' = -6X^4 + 2X^3 + 10X^2 - 8X - 14$

Réseau associé à un PMNS

Réduction interne

- $C = (3222, -2840, 1235, 230, -1217)$
- $C' = (3236, -2832, 1225, 228, -1211)$
  $C' \in \mathcal{L}(\mathcal{B})$
- $S = C - C' = (-14, -8, 10, 2, -6)$

Réseau associé à un PMNS

Réduction interne

- $C = (3222, -2840, 1235, 230, -1217)$
- $C' = (3236, -2832, 1225, 228, -1211)$
  $C' \in \mathcal{L}(\mathcal{B})$
- $S = C - C' = (-14, -8, 10, 2, -6)$

- Recherche de vecteur proche dans un réseau

Problème de réduction

PMNS reduction problem

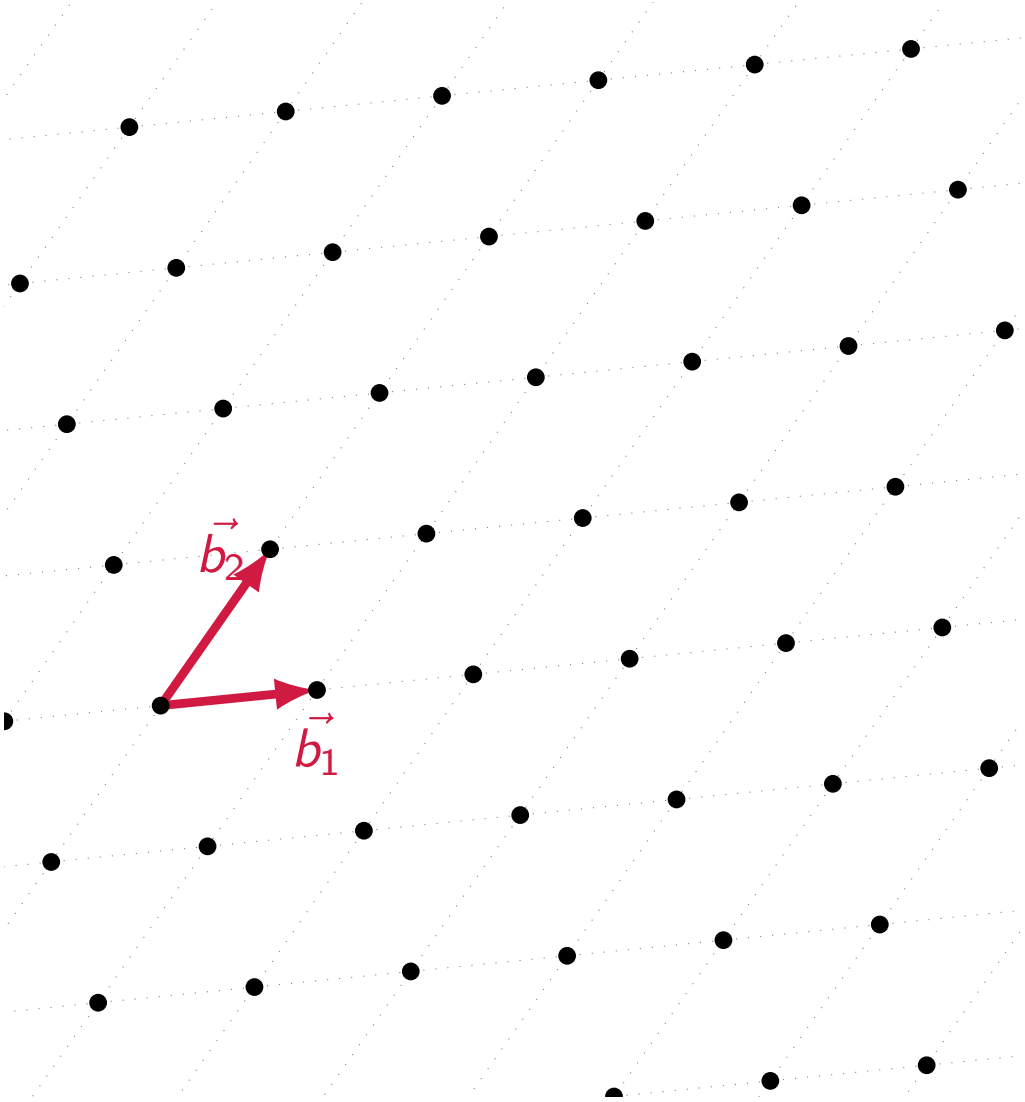
Soit $\mathcal{B} = (p, n, \gamma, \rho, E, I)$ un PMNS et un vecteur $v \in \mathbb{Z}^n$, trouver v' dans $\mathcal{L}(\mathcal{B})$ tel que

$$\|v - v'\|_I < \rho.$$

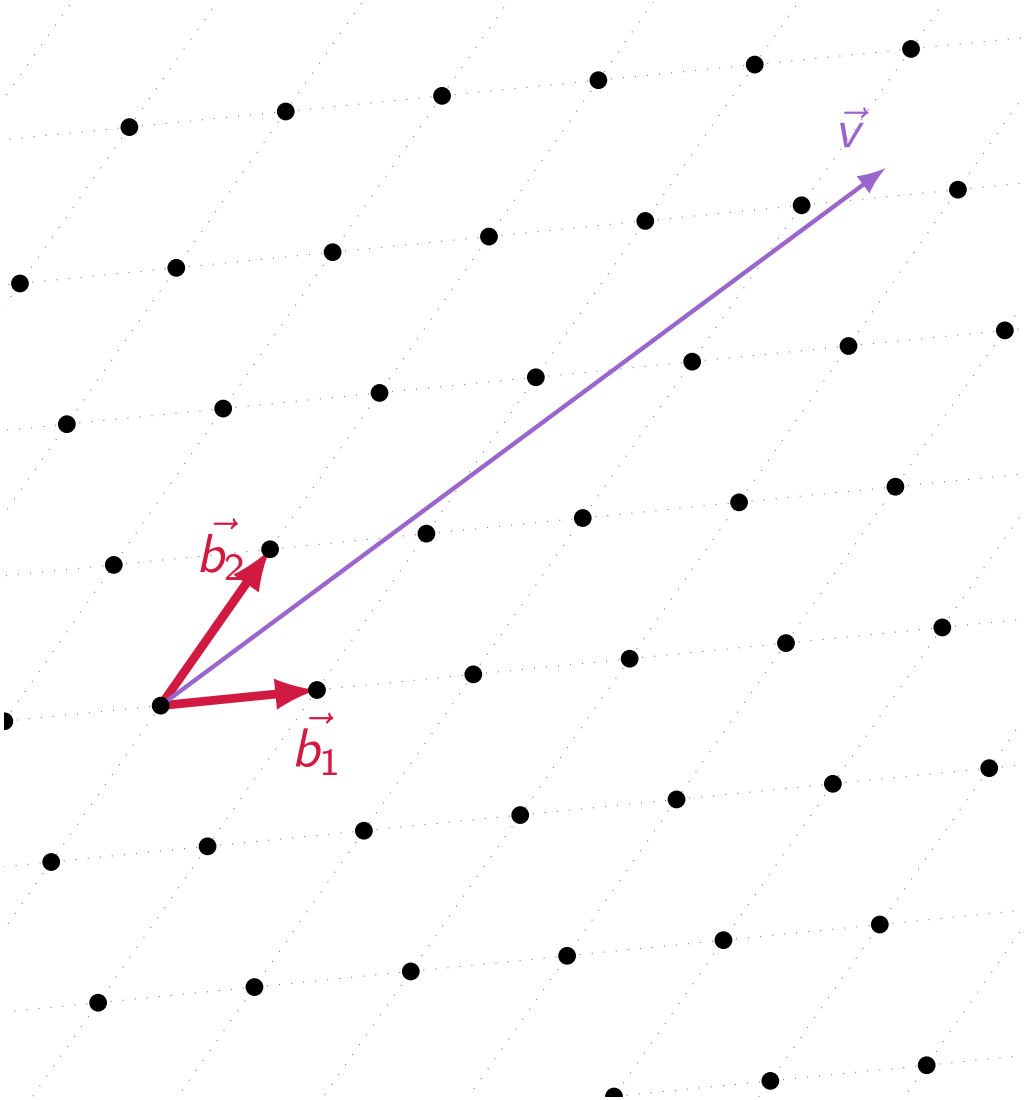
Remarque

Il s'agit d'un version faible du problème de recherche de plus proche vecteur dans un réseau (Closest Vector Problem) qui lui est NP-dur pour la norme infinie.

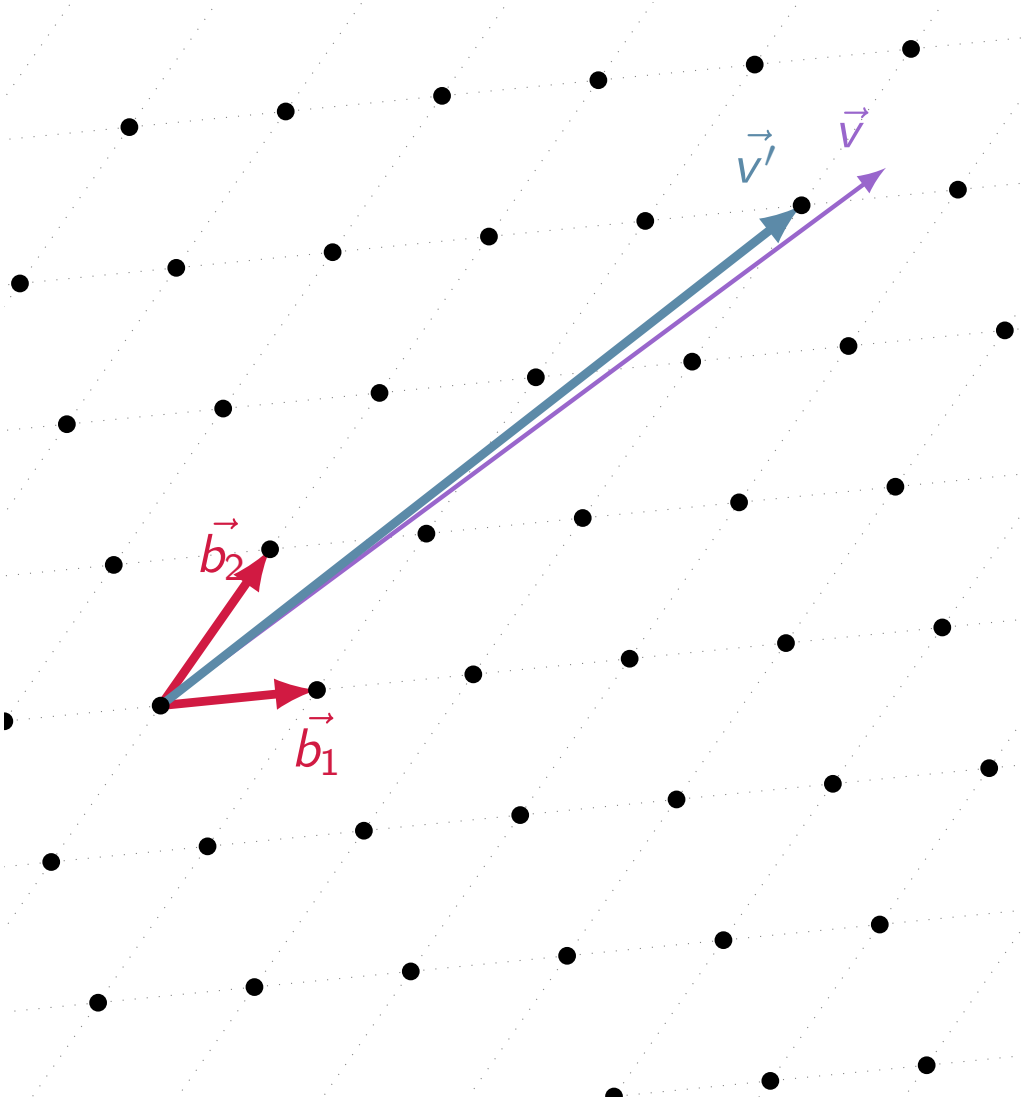
Résolution de CVP avec l'algorithme de Babai



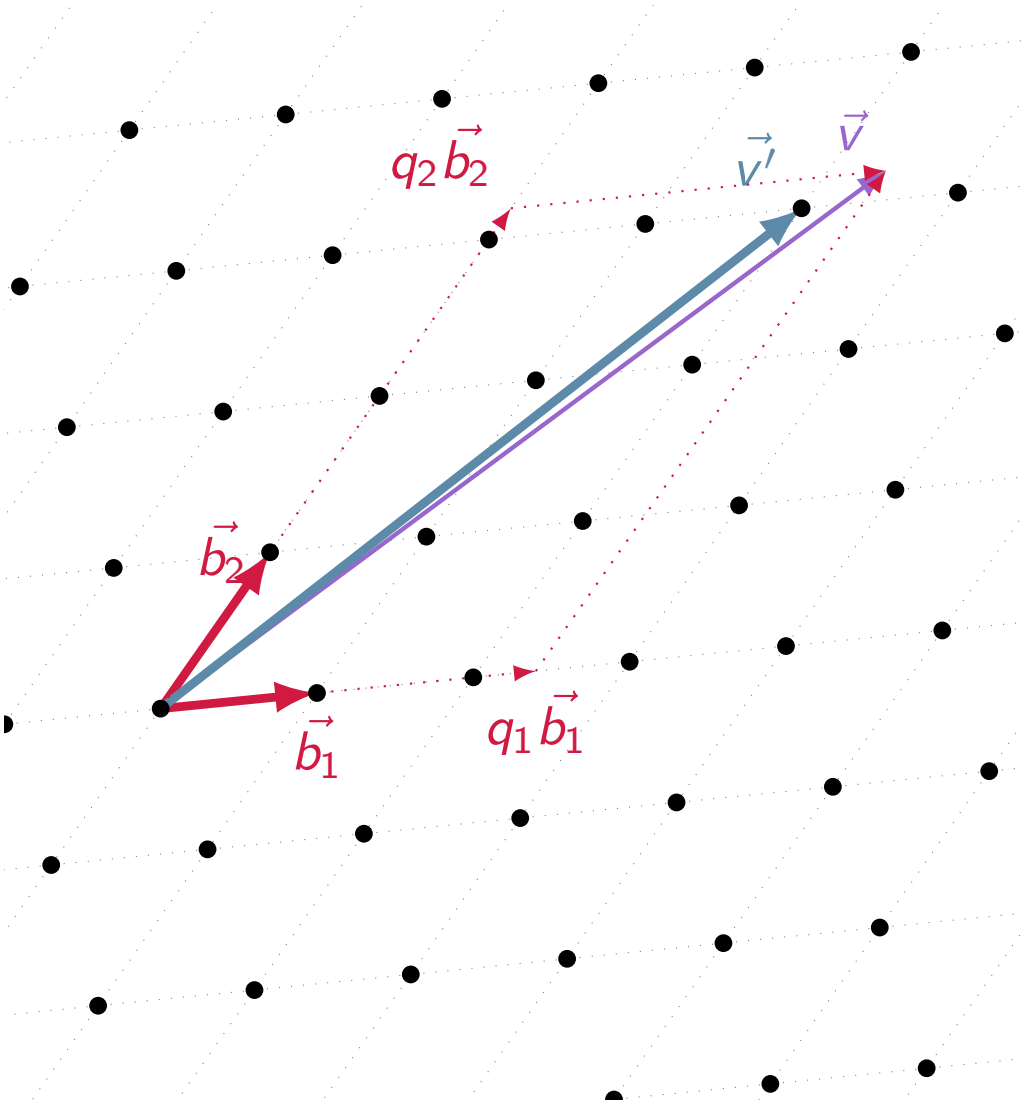
Résolution de CVP avec l'algorithme de Babai



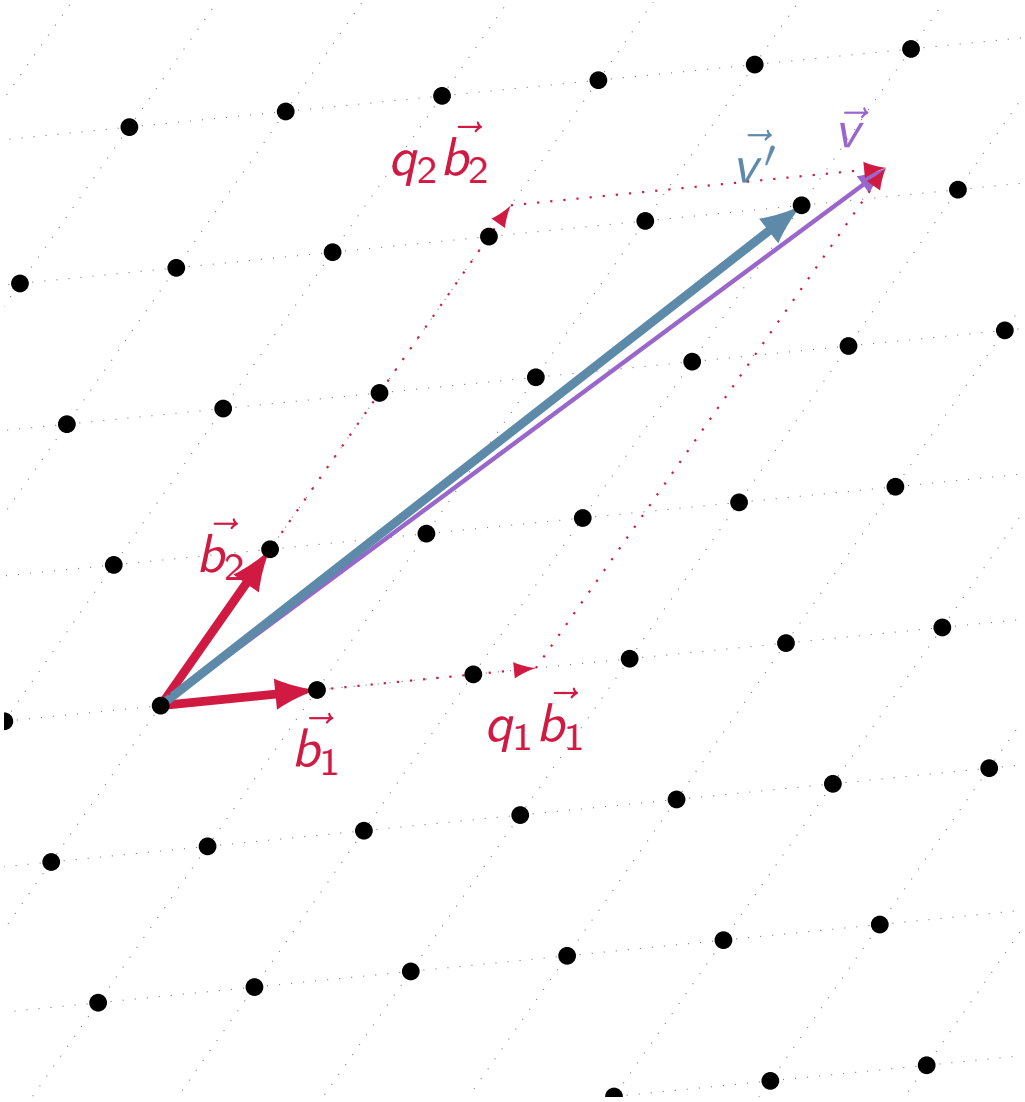
Résolution de CVP avec l'algorithme de Babai



Résolution de CVP avec l'algorithme de Babai

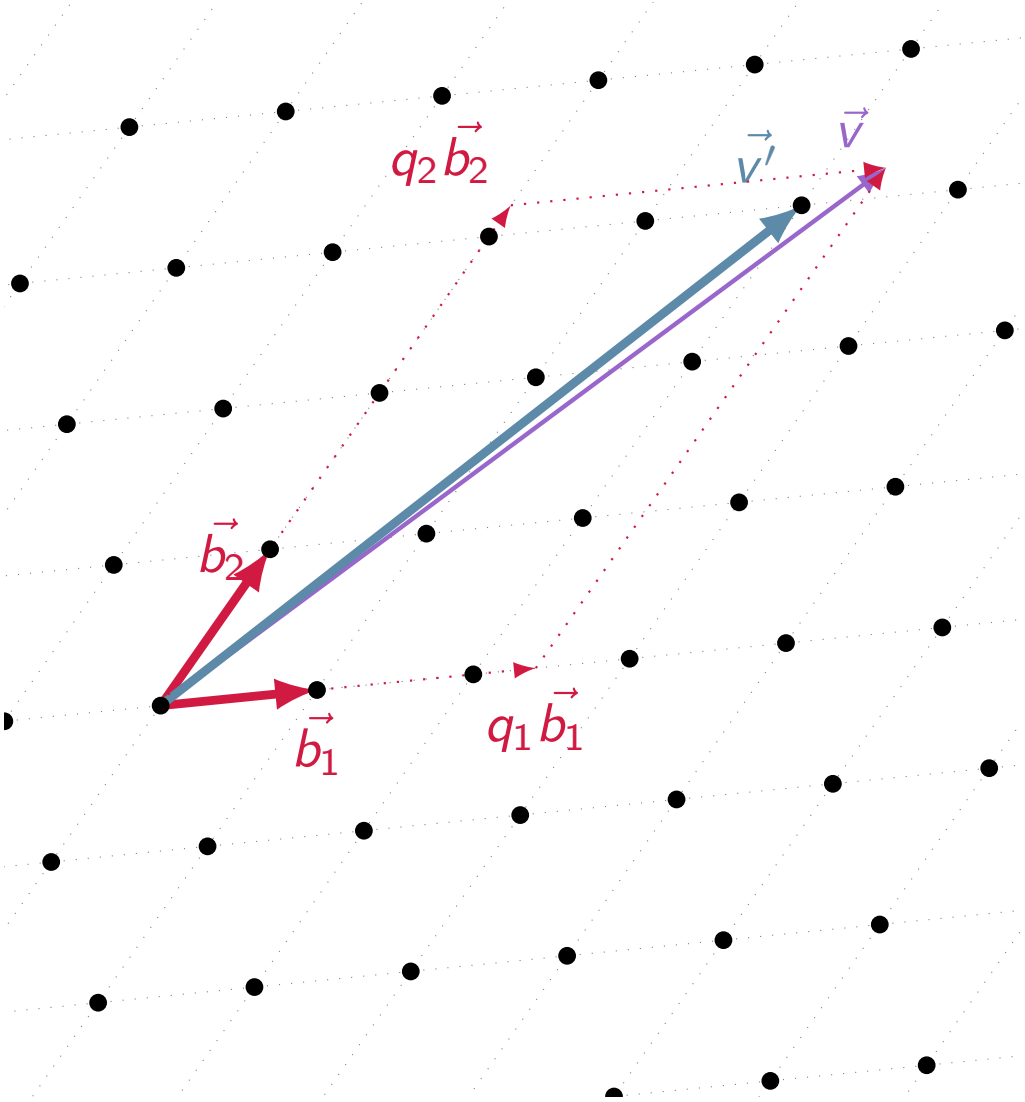


Résolution de CVP avec l'algorithme de Babai



$$\begin{aligned}\vec{v} &= (v_1, v_2) \\ &= q_1\vec{b_1} + q_2\vec{b_2}\end{aligned}$$

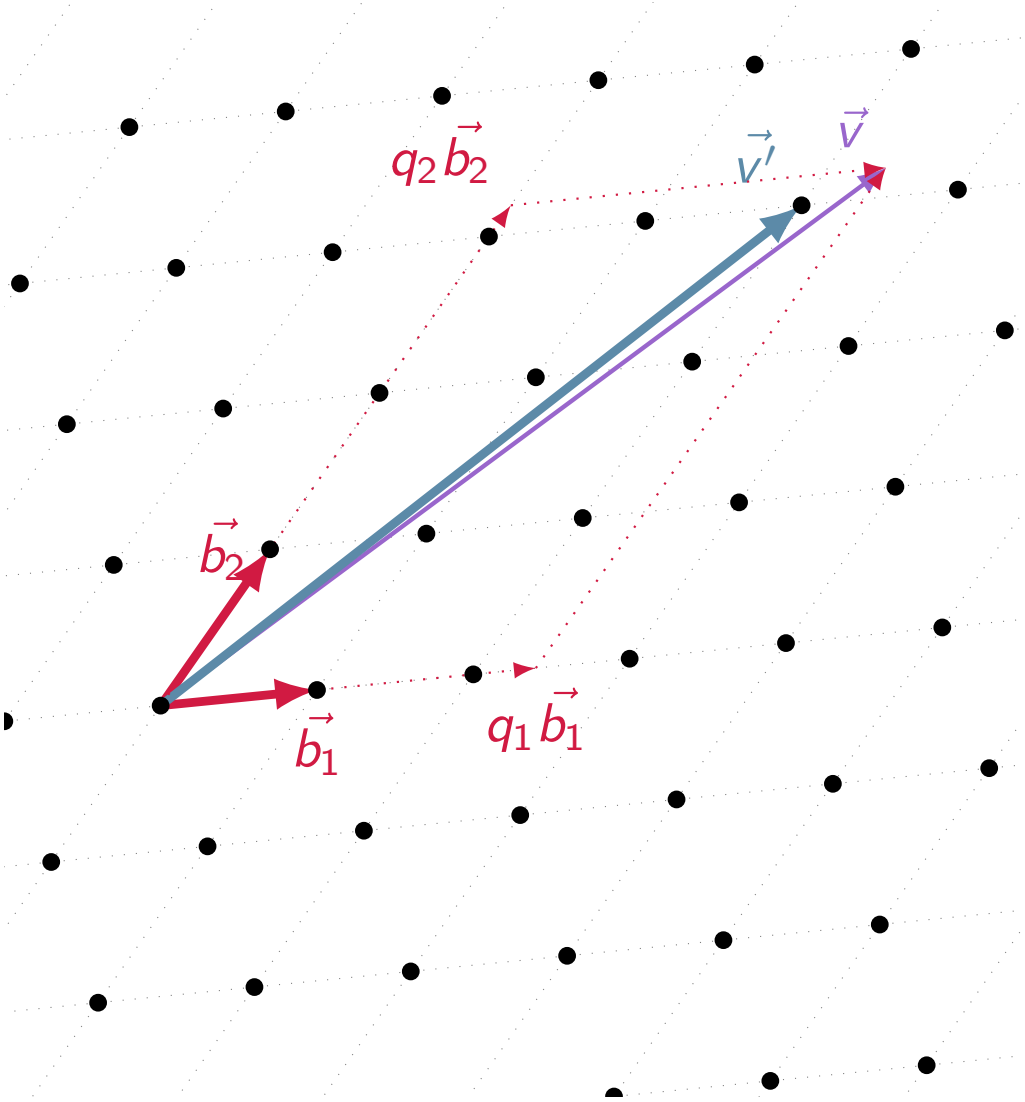
Résolution de CVP avec l'algorithme de Babai



$$\vec{v} = (v_1, v_2) = q_1\vec{b_1} + q_2\vec{b_2}$$

$$\vec{v'} = \lfloor q_1 \rfloor \vec{b_1} + \lfloor q_2 \rfloor \vec{b_2}$$

Résolution de CVP avec l'algorithme de Babai



$$\begin{aligned} \vec{v} &= (v_1, v_2) \\ &= q_1\vec{b_1} + q_2\vec{b_2} \end{aligned}$$

$$\vec{v'} = \lfloor q_1 \rfloor \vec{b_1} + \lfloor q_2 \rfloor \vec{b_2}$$

$$\begin{aligned} \|v - v'\|_1 \leq & \quad (q_1 - \lfloor q_1 \rfloor)\vec{b_1} \\ & + (q_2 - \lfloor q_2 \rfloor)\vec{b_2} \\ \leq & \quad \underbrace{\|\vec{b_1}\|_1 + \|\vec{b_2}\|_1}_{\rho} \end{aligned}$$

Algorithme de Babai "*Rounding*" (RND)

- Calculer les coordonnées (rationnelles) de v dans la base $\mathrm{B} = (\vec{b_1}, \ldots, \vec{b_n})$ du réseau

$$v = q_1 \vec{b_1} + \cdots + q_n \vec{b_n}$$

- Choisir un vecteur proche par arrondi des coordonnées

$$v' = \lfloor q_1 \rfloor \vec{b_1} + \cdots + \lfloor q_n \rfloor \vec{b_n}$$

- Le vecteur réduit $s = v - v'$ vérifie

$$\|s\|_1 \leq \underbrace{\sum_{i=1}^{n} \|\vec{b_i}\|_1}_{\rho}$$

Algorithme de Babai "*Rounding*" (RND)

- Calculer les coordonnées (rationnelles) de v dans la base $\mathrm{B} = (\vec{b_1}, \ldots, \vec{b_n})$ du réseau

$$(q_1, \ldots, q_n) = vB^{-1}$$

- Choisir un vecteur proche par arrondi des coordonnées

$$v' = (\lfloor q_1 \rfloor, \ldots, \lfloor q_n \rfloor)\mathrm{B}$$

- Le vecteur réduit $s = v - v'$ vérifie

$$\|s\|_1 \leq \underbrace{\sum_{i=1}^{n} \|\vec{b_i}\|_1}_{\rho}$$

Généralisation des algorithmes standards

$$c - c \times \frac{1}{p} \times p = 0$$

Barret

$$\underbrace{c - \left\lfloor c \times \frac{1}{p} \right\rfloor \times p}_{\leq p}$$

Généralisation des algorithmes standards

$$v - v \times \mathrm{B}^{-1} \times \mathrm{B} = (0, \ldots, 0)$$

Barret

$$\underbrace{c - \left\lfloor c \times \frac{1}{p} \right\rfloor \times p}_{\leq p}$$

Généralisation des algorithmes standards

$$v - v \times \mathrm{B}^{-1} \times \mathrm{B} = (0, \ldots, 0)$$

Barret

$$\underbrace{v - \left\lfloor v\mathrm{B}^{-1} \right\rfloor \times \mathrm{B}}_{\|.\|_1 \leq \rho}$$

Généralisation des algorithmes standards

$$v - v \times \mathrm{B}^{-1} \times \mathrm{B} = (0, \ldots, 0)$$

Barret

$$\underbrace{v - \left\lfloor v\mathrm{B}^{-1} \right\rfloor \times \mathrm{B}}_{\|.\|_1 \leq \rho}$$

Montgomery

$$\underbrace{c - \left(c \times \frac{1}{p} \mod \phi \right) \times p}_{\equiv 0 \mod \phi}$$

Généralisation des algorithmes standards

$$v - v \times \mathrm{B}^{-1} \times \mathrm{B} = (0, \ldots, 0)$$

Barret

$$\underbrace{v - \left\lfloor v\mathrm{B}^{-1} \right\rfloor \times \mathrm{B}}_{\|.\|_1 \leq \rho}$$

Montgomery

$$\underbrace{v - \left(v\mathrm{B}^{-1} \mod \phi\right) \times \mathrm{B}}_{\equiv (0,\ldots,0) \mod \phi}$$

Généralisation des algorithmes standards

$$v - v \times \mathrm{B}^{-1} \times \mathrm{B} = (0, \ldots, 0)$$

Barret **Montgomery**

$$\underbrace{v - \left\lfloor v\mathrm{B}^{-1} \right\rfloor \times \mathrm{B}}_{\|.\|_1 \leq \rho} \qquad \underbrace{v - \left(v\mathrm{B}^{-1} \mod \phi\right) \times \mathrm{B}}_{\equiv (0,\ldots,0) \mod \phi}$$

- L'inverse $\mathrm{B}^{-1} \mod \phi$ existe toujours. Plus besoin de rechercher le polynome M.

Généralisation des algorithmes standards

- $M(x) = m_0 + m_1 X + \ldots m_{n-1} X^{n-1}$
- $M'(X)M(X) \equiv 1 \mod E(X)$
- $m'_0 M(X) + m'_1 XM(X) + \ldots m'_{n-1} X^{n-1} M(X) \equiv 1 \mod E(X)$
-

$$(m'_0, \ldots, m'_{n-1}) = (1, 0, \ldots, 0) \begin{pmatrix} M(X) \\ XM(X) \mod E(X) \\ X^2 M(X) \mod E(X) \\ \vdots \\ X^{n-1} M(X) \mod E(X) \end{pmatrix}^{-1}.$$

Généralisation des algorithmes standards

- $M(x) = m_0 + m_1 X + \ldots m_{n-1} X^{n-1}$
- $M'(X)M(X) \equiv 1 \mod E(X)$
- $m'_0 M(X) + m'_1 XM(X) + \ldots m'_{n-1} X^{n-1} M(X) \equiv 1 \mod E(X)$
-

$$(m'_0, \ldots, m'_{n-1}) = (1, 0, \ldots, 0) \begin{pmatrix} M(X) \\ XM(X) \mod E(X) \\ X^2 M(X) \mod E(X) \\ \vdots \\ X^{n-1} M(X) \mod E(X) \end{pmatrix}^{-1}.$$

- Il s'agit de la matrice d'un sous réseau de $\mathcal{L}$, son determinant est un multiple de p.

Algorithme de Babai: *Rounding*

Data:

- Un réseau $\mathcal{L}$ défini par une base $B = (\vec{b_1}, \ldots, \vec{b_n})$,
- $\mathrm{B}^{-1} = (b'_{ij})$ la matrice inverse de B
- $v \in \mathbb{Z}^n$

Result: un vecteur s équivalent à v avec de "*petits*" coefficients

$s \leftarrow v$;
for $i = 1 \ldots n$ **do**
  $r \leftarrow \lfloor \sum_{j=1}^{n} v_j b'_{ij} \rceil$;
  $s \leftarrow s - r\vec{b_i}$;
end
return s;

Problème

Les coefficients sont "gros".

Algorithme de Babai: *Rounding*

$$B = \begin{pmatrix} -1581 & -2509 & -366 & -1233 & 1588 \\ 3176 & -1581 & -2509 & -366 & -1233 \\ 2466 & -3176 & 1581 & 2509 & 366 \\ 732 & 2466 & -3176 & 1581 & 2509 \\ -5018 & -732 & -2466 & 3176 & -1581 \end{pmatrix}$$

$$B^{-1} = \begin{pmatrix} \frac{-172526576471785}{2228155915641550789} & \frac{201547545314005}{2228155915641550789} & \frac{134398522401593}{2228155915641550789} & \frac{71679752211328}{2228155915641550789} & \frac{-185607823726855}{2228155915641550789} \\ \frac{-371215647453710}{2228155915641550789} & \frac{-172526576471785}{2228155915641550789} & \frac{-201547545314005}{2228155915641550789} & \frac{134398522401593}{2228155915641550789} & \frac{-71679752211328}{2228155915641550789} \\ \frac{-143359504422656}{2228155915641550789} & \frac{-371215647453710}{2228155915641550789} & \frac{172526576471785}{2228155915641550789} & \frac{-201547545314005}{2228155915641550789} & \frac{-134398522401593}{2228155915641550789} \\ \frac{-268797044803186}{2228155915641550789} & \frac{-143359504422656}{2228155915641550789} & \frac{371215647453710}{2228155915641550789} & \frac{172526576471785}{2228155915641550789} & \frac{201547545314005}{2228155915641550789} \\ \frac{403095090628010}{2228155915641550789} & \frac{-268797044803186}{2228155915641550789} & \frac{143359504422656}{2228155915641550789} & \frac{371215647453710}{2228155915641550789} & \frac{-172526576471785}{2228155915641550789} \end{pmatrix}$$

Algorithme de réduction interne de Babai RND

$\mathrm{B}^{-1} = (b'_{ij})$. On introduit h_1 and h_2 tels que $\left\lfloor 2^{h_1} \mathrm{b'}_{ij} \right\rceil$ et $\left\lfloor \frac{v_j}{2^{h_2}} \right\rfloor$ tiennent dans un mot machine.

$$
\begin{aligned}
s &= v - \left\lfloor v\mathrm{B}^{-1} \right\rceil \mathrm{B} && (1)\\
&= v - \left\lfloor \frac{\left\lfloor \frac{v}{2^{h_2}} \right\rfloor \left\lfloor 2^{h_1} \mathrm{B}^{-1} \right\rceil}{2^{h_1 - h_2}} \right\rfloor \mathrm{B} && (2)
\end{aligned}
$$

.

Algorithme de réduction interne de Babai RND

Data:

- Un réseau $\mathcal{L}$ défini par une base $(\vec{b_1}, \ldots, \vec{b_n})$,
- (H_{ij}) les coeffients précalculés de l'inverse $2^{h_1} \mathrm{B}^{-1}$
- $v \in \mathbb{Z}^n$

Result: un vecteur s équivalent à v avec de "*petits*" coefficients

$s \leftarrow v$;
$v' \leftarrow (v_1 \gg h_2, \ldots, v_n \gg h_2)$;
for $i = 1 \ldots n$ **do**
  $r \leftarrow 0$;
  for $j = 1 \ldots n$ **do**
    $r \leftarrow r + v'_j \times H_{ij}$
  end
  $r \leftarrow r \gg (h_1 - h_2)$;
  $s \leftarrow s - r \times \vec{b_i}$;
end
return s;

- Les premières additions sont des affectations
- Dans les additions/soustractions on peut ignorer les parties hautes:
- Compléxité:
  - $2n^2$ `mult_w`
  - $3n^2 - 2n$ `add_w`
  - $2n$ `rshift`
- En dimension 1 on retrouve exactement l'algorihtme de Barrett

Conditions de justesse

Lemma

Soit $s = \sum_{i=1}^{n} s_i \mathrm{b}_i$ *la sortie de l'algorihtme Barrett/Babai alors il* $1 \leq i \leq n$ *il existe une constante* K_i *ne dépendant pas de* v *telle que*

$$|s_i| \leq \frac{\|v\|_1}{2^{h_1+1}} + K_i.$$

Idée de preuve

$$\begin{aligned} s &= v - \left\lfloor \frac{\left\lfloor \frac{v}{2^{h_2}} \right\rfloor \left\lfloor 2^{h_1} \mathrm{B}^{-1} \right\rceil}{2^{h_1-h_2}} \right\rceil \mathrm{B} \\ s\mathrm{B}^{-1} &= v\mathrm{B}^{-1} - \left\lfloor \frac{\left\lfloor \frac{v}{2^{h_2}} \right\rfloor \left\lfloor 2^{h_1} \mathrm{B}^{-1} \right\rceil}{2^{h_1-h_2}} \right\rceil \end{aligned}$$

Conditions de justesse

Idée de preuve

$$
\begin{aligned}
|s_i| &= \left| \sum_j^{n-1} v_j {b'}_{ij} - \left\lfloor \frac{< \left\lfloor \frac{v}{2^{h_2}} \right\rfloor , \left\lfloor 2^{h_1} b'_i \right\rceil >}{2^{h_1-h_2}} \right\rfloor \right| \\
&\leq \sum_j^{n-1} \frac{|v_j|}{2^{h_1+1}} + 2^{h_2} \left| \sum_j^{n-1} {b'}_{ij} \right| + \frac{n}{2^{h_1-h_2+1}} + 1 \\
&\leq \frac{\|v\|_1}{2^{h_1+1}} + K_i
\end{aligned}
$$

Conditions de justesse

Proposition

Posons $\mu_1 = \sum_i \|\vec{b}_i\|_1$. *Il existe* K *tel que*

$$\|s\|_1 \leq (\frac{\|v\|_1}{2^{h_1+1}} + K)\mu_1.$$

Corollary

Soit $(p, n, \gamma, \rho, E, 1)$ *et* K *donné par la proposition précédente. Si* v *satisfait*

$$\|v\|_1 \leq 2^{h_1+1}\left(\frac{\rho}{\mu_1} - K\right)$$

l'algorithme résout le "PMNS reduction problem".

Majoration de $\|V\|_1$

- Soient $A = a_0 + a_1X + a_2X^2$, $B = b_0 + b_1X + b_2X^2$ et $E = X^3 - \lambda$. On veut traduire les opérations polynomiales en opérations sur les vecteurs.

$$\begin{pmatrix} v_0 \\ v_1 \\ v_2 \end{pmatrix} \quad = \quad \begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix}$$

Majoration de $\|V\|_1$

- Soient $A = a_0 + a_1 X + a_2 X^2$, $B = b_0 + b_1 X + b_2 X^2$ et $E = X^3 - \lambda$. On veut traduire les opérations polynomiales en opérations sur les vecteurs.

1. $AB = (a_0 b_0, a_1 b_0 + a_0 b_1, a_2 b_0 + a_1 b_1 + a_0 b_2, a_2 b_1 + a_1 b_2, a_2 b_2)$

$$\begin{pmatrix} v_0 \\ v_1 \\ v_2 \end{pmatrix} \quad = \quad \begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix}$$

Majoration de $\|V\|_1$

- Soient $A = a_0 + a_1X + a_2X^2$, $B = b_0 + b_1X + b_2X^2$ et $E = X^3 - \lambda$. On veut traduire les opérations polynomiales en opérations sur les vecteurs.

1. $AB = (a_0b_0, a_1b_0 + a_0b_1, a_2b_0 + a_1b_1 + a_0b_2, a_2b_1 + a_1b_2, a_2b_2)$

$$\begin{pmatrix} v_0 \\ v_1 \\ v_2 \end{pmatrix} = \underbrace{\begin{pmatrix} a_0 & 0 & 0 \\ a_1 & a_0 & 0 \\ a_2 & a_1 & a_0 \\ 0 & a_2 & a_1 \\ 0 & 0 & a_2 \end{pmatrix}}_{\textit{multiplication}} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix}$$

Majoration de $\|V\|_1$

- Soient $A = a_0 + a_1X + a_2X^2$, $B = b_0 + b_1X + b_2X^2$ et $E = X^3 - \lambda$. On veut traduire les opérations polynomiales en opérations sur les vecteurs.

1. $AB = (a_0b_0, a_1b_0 + a_0b_1, a_2b_0 + a_1b_1 + a_0b_2, a_2b_1 + a_1b_2, a_2b_2)$
2. $AB \mod E =$
   $(a_0b_0 + \lambda(a_2b_1 + a_1b_2), a_1b_0 + a_0b_1 + \lambda a_2b_2, a_2b_0 + a_1b_1 + a_0b_2)$

$$\begin{pmatrix} v_0 \\ v_1 \\ v_2 \end{pmatrix} = \underbrace{\begin{pmatrix} a_0 & 0 & 0 \\ a_1 & a_0 & 0 \\ a_2 & a_1 & a_0 \\ 0 & a_2 & a_1 \\ 0 & 0 & a_2 \end{pmatrix}}_{\textit{multiplication}} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix}$$

Majoration de $\|V\|_1$

- Soient $A = a_0 + a_1X + a_2X^2$, $B = b_0 + b_1X + b_2X^2$ et $E = X^3 - \lambda$. On veut traduire les opérations polynomiales en opérations sur les vecteurs.

1. $AB = (a_0b_0, a_1b_0 + a_0b_1, a_2b_0 + a_1b_1 + a_0b_2, a_2b_1 + a_1b_2, a_2b_2)$
2. $AB \mod E =$
   $(a_0b_0 + \lambda(a_2b_1 + a_1b_2), a_1b_0 + a_0b_1 + \lambda a_2b_2, a_2b_0 + a_1b_1 + a_0b_2)$

$$\begin{pmatrix} v_0 \\ v_1 \\ v_2 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & 0 & \lambda & 0 \\ 0 & 1 & 0 & 0 & \lambda \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}}_{\textit{reduction externe}} \underbrace{\begin{pmatrix} a_0 & 0 & 0 \\ a_1 & a_0 & 0 \\ a_2 & a_1 & a_0 \\ 0 & a_2 & a_1 \\ 0 & 0 & a_2 \end{pmatrix}}_{\textit{multiplication}} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix}$$

$$=$$

Majoration de $\|V\|_1$

- Soient $A = a_0 + a_1X + a_2X^2$, $B = b_0 + b_1X + b_2X^2$ et $E = X^3 - \lambda$. On veut traduire les opérations polynomiales en opérations sur les vecteurs.

1. $AB = (a_0b_0, a_1b_0 + a_0b_1, a_2b_0 + a_1b_1 + a_0b_2, a_2b_1 + a_1b_2, a_2b_2)$
2. $AB \mod E =$
   $(a_0b_0 + \lambda(a_2b_1 + a_1b_2), a_1b_0 + a_0b_1 + \lambda a_2b_2, a_2b_0 + a_1b_1 + a_0b_2)$

$$\begin{pmatrix} v_0 \\ v_1 \\ v_2 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & 0 & \lambda & 0 \\ 0 & 1 & 0 & 0 & \lambda \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}}_{\textit{reduction externe}} \underbrace{\begin{pmatrix} a_0 & 0 & 0 \\ a_1 & a_0 & 0 \\ a_2 & a_1 & a_0 \\ 0 & a_2 & a_1 \\ 0 & 0 & a_2 \end{pmatrix}}_{\textit{multiplication}} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix}$$

$$= \underbrace{\begin{pmatrix} a_0 & \lambda a_2 & \lambda a_1 \\ a_1 & a_0 & \lambda a_2 \\ a_2 & a_1 & a_0 \end{pmatrix}}_{\textit{mult+red externe}} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix}$$

Majoration de $\|V\|_1$

$$\begin{pmatrix} v_0 \\ \vdots \\ v_{n-1} \end{pmatrix} = \begin{pmatrix} I_n & Red(E) \end{pmatrix} Conv(A) \begin{pmatrix} b_0 \\ \vdots \\ b_{n-1} \end{pmatrix} = RedMul(A,E) \begin{pmatrix} b_0 \\ \vdots \\ b_{n-1} \end{pmatrix}$$

$$\begin{aligned} \|V\|_1 &\leq \|RedMul(A,E)\|_1 \|B\|_1 && (3) \\ &\leq \|(I_n Red(E))\|_1 \|Conv(A)\|_1 \|B\|_1 && (4) \\ &\leq \|Red(E)\|_1 \|A\|_1 \|B\|_1 && (5) \end{aligned}$$

Majoration de $\|V\|_1$

Dans le cas $E = X^n - \lambda$, on a donc

$$Red(E) = \lambda \times \begin{pmatrix} & I_{n-1} & \\ 0 & \dots & 0 \end{pmatrix} = \begin{pmatrix} \lambda & 0 & \dots & 0 \\ 0 & \lambda & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \lambda \\ 0 & \dots & \dots & 0 \end{pmatrix}$$

et donc $\|Red(E)\|_1 = |\lambda|$. D'où

$$\|V\|_1 \leq |\lambda| \|A\|_1 \|B\|_1.$$

Majoration de $\|V\|_1$

$$RedMul(A,E) = \begin{pmatrix} a_0 & \lambda a_{n-1} & & \dots & \lambda a_1 \\ a_1 & a_0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & & \vdots \\ a_{n-2} & \dots & & a_0 & \lambda a_{n-1} \\ a_{n-1} & \dots & \dots & a_1 & a_0 \end{pmatrix}$$

et donc $\|RedMul(A,E)\|_\infty = (1+(n-1)|\lambda|)\|A\|_\infty$. D'où

$$\|V\|_\infty \leq (1+(n-1)|\lambda|)\|A\|_\infty \|B\|_\infty.$$

Majoration de $\|V\|_1$

Theorem

Soit $(p, n, \gamma, \rho, X^n - \lambda, 1)$ *un PMNS. Soient A, B et V trois polynomes de degré* $n-1$ *tels que* $V = AB \mod E$ *avec* $\|A\|_1 \leq \rho$ *et* $\|B\|_1 \leq \rho$. *Si*

$$|\lambda| K \mu_1^2 \leq 2^{h_1 - 1}$$

avec K donnée par la proposition précédente alors l'algorithme Barrett/Babai résoud le "PMNS reduction problem".

Résultats d'implantation

p size	192	224		256	384		512	
degree n	4	4	5	5	7	8	10	11
Montgomery	60	63	86	83	155	206	300	357
Nearest P.	83	83	143	133	265	331	494	606
Nearest P. (opt)	70	70	108	117	196	248	405	494
Rounding	50	52	76	74	155	205	321	362
Rounding (opt)	45	45	64	70	148	199	291	344

Table: Number of CPU clock cycles to perform an internal reduction using a polynomial of the form $X^n - \lambda$

- Générateur de code disponible sur gitlab
  `https://plmlab.math.cnrs.fr/melon359/pmns.git`

Conclusion

- Babai permet de généraliser les algos de réduction modulaires
- On a résolu le problème de la recherche du polynome M pour l'algo de Montgomery
- L'algo Babai / Barrett est tout aussi efficace que celui de Montgomery

Perspectives

- Arithmetique flottante
- Matrice de Toeplizt