Approximants polynomiaux efficaces en machine

Nicolas Brisebarre – CNRS / LIP

Guillaume Hanrot – ENS Lyon / LIP

Séminaire Apics, 29/01/2010

Plan

- Contexte ;

Plan

- Contexte ;
- Approximation L^2 ;

Plan

- Contexte ;
- Approximation L^2 ;
- Interlude : réseaux euclidiens ;

Plan

- Contexte ;
- Approximation L^2 ;
- Interlude : réseaux euclidiens ;
- Retour au cas L^2.

Contexte

Contexte

- But : évaluer une fonction f sur machine ;

Contexte

- But : évaluer une fonction f sur machine ;
- Principe :

Contexte

- But : évaluer une fonction f sur machine ;
- Principe :
  - Réduction d'argument : se ramener à $g : I \to \mathbb{R}$, I petit ;

Contexte

- But : évaluer une fonction f sur machine ;
- Principe :
  - Réduction d'argument : se ramener à $g : I \to \mathbb{R}$, I petit ;
  - Approcher g sur I par une fonction h facile à évaluer ;

Contexte

- But : évaluer une fonction f sur machine ;
- Principe :
  - Réduction d'argument : se ramener à $g : I \to \mathbb{R}$, I petit ;
  - Approcher g sur I par une fonction h facile à évaluer ;
  - (typiquement polynôme) ;

Contexte

- But : évaluer une fonction f sur machine ;
- Principe :
  - Réduction d'argument : se ramener à $g : I \to \mathbb{R}$, I petit ;
  - Approcher g sur I par une fonction h facile à évaluer ;
  - (typiquement polynôme) ;
  - Évaluer h.

Fonctions faciles à évaluer

- Polynôme = facile à évaluer ?

Fonctions faciles à évaluer

- Polynôme = facile à évaluer ?
- $P(x) = \sum_{i=0}^{n} p_k x^k$ s'évalue par le schéma de Horner ;

Fonctions faciles à évaluer

- Polynôme = facile à évaluer ?
- $P(x) = \sum_{i=0}^{n} p_k x^k$ s'évalue par le schéma de Horner ;
-

$$P(x) = (\ldots(p_n \cdot x + p_{n-1}) \cdot x + \ldots) + p_0.$$

Fonctions faciles à évaluer

- Polynôme = facile à évaluer ?
- $P(x) = \sum_{i=0}^{n} p_k x^k$ s'évalue par le schéma de Horner ;
-

$$P(x) = (\ldots(p_n \cdot x + p_{n-1}) \cdot x + \ldots) + p_0.$$

- Coeffs de P adaptés à une arithmétique efficace ?

Virgule fixe et virgule flottante

- Principe : nombre fini de “nombres réels” possibles

Virgule fixe et virgule flottante

- Principe : nombre fini de “nombres réels” possibles
- Choisir un ensemble fini de nombres représentables/tatifs M_p ;

Virgule fixe et virgule flottante

- Principe : nombre fini de “nombres réels” possibles
- Choisir un ensemble fini de nombres représentables/tatifs M_p ;
- $x \in M_p$ représente $I(x) \subset \mathbb{R}$ partition de $\mathbb{R}$;

Virgule fixe et virgule flottante (II)

- Virgule fixe :

Virgule fixe et virgule flottante (II)

- Virgule fixe :
  - $I(x)$ de taille constante ;

Virgule fixe et virgule flottante (II)

- Virgule fixe :
  - $I(x)$ de taille constante ;
  - $M_p = \{x/2^k, x \in [-2^{p-1}, 2^{p-1} - 1[\}$;

Virgule fixe et virgule flottante (II)

- Virgule fixe :
  - $I(x)$ de taille constante ;
  - $M_p = \{x/2^k, x \in [-2^{p-1}, 2^{p-1} - 1[\}$;
  - Erreur absolue ;

Virgule fixe et virgule flottante (II)

- Virgule fixe :
  - $I(x)$ de taille constante ;
  - $M_p = \{x/2^k, x \in [-2^{p-1}, 2^{p-1} - 1[\}$;
  - Erreur absolue ;
- Virgule flottante :

Virgule fixe et virgule flottante (II)

- Virgule fixe :
  - $I(x)$ de taille constante ;
  - $M_p = \{x/2^k, x \in [-2^{p-1}, 2^{p-1} - 1[\}$;
  - Erreur absolue ;
- Virgule flottante :
  - $I(x)$ de taille correspondant à taille de x ;

Virgule fixe et virgule flottante (II)

- Virgule fixe :
  - $I(x)$ de taille constante ;
  - $M_p = \{x/2^k, x \in [-2^{p-1}, 2^{p-1} - 1[\}$;
  - Erreur absolue ;
- Virgule flottante :
  - $I(x)$ de taille correspondant à taille de x ;
  - $M_p = \{x/2^p \cdot 2^e, x \in \pm[2^{p-1}, 2^p[\}$;

Virgule fixe et virgule flottante (II)

- Virgule fixe :
  - $I(x)$ de taille constante ;
  - $M_p = \{x/2^k, x \in [-2^{p-1}, 2^{p-1} - 1[\}$;
  - Erreur absolue ;
- Virgule flottante :
  - $I(x)$ de taille correspondant à taille de x ;
  - $M_p = \{x/2^p \cdot 2^e, x \in \pm[2^{p-1}, 2^p[\}$;
  - Erreur relative.

Virgule fixe et virgule flottante (II)

- Virgule fixe :
  - $I(x)$ de taille constante ;
  - $M_p = \{x/2^k, x \in [-2^{p-1}, 2^{p-1} - 1[\}$;
  - Erreur absolue ;
- Virgule flottante :
  - $I(x)$ de taille correspondant à taille de x ;
  - $M_p = \{x/2^p \cdot 2^e, x \in \pm[2^{p-1}, 2^p[\}$;
  - Erreur relative.
- x flottant + exposant fixé $\Leftrightarrow$ x virgule fixe dans $[\pm 1/2, 1[$

Le problème à résoudre

- Entrée : P à coefficients réels, précisions p_i, $i = 0..\deg P$;

Le problème à résoudre

- Entrée : P à coefficients réels, précisions p_i, $i = 0..\deg P$;
- Sortie : $\tilde{P} = \sum_{i=0}^{\deg P} \tilde{a}_i X^i$, a_i flottant en précision p_i ;

Le problème à résoudre

- Entrée : P à coefficients réels, précisions p_i, $i = 0..\deg P$;
- Sortie : $\tilde{P} = \sum_{i=0}^{\deg P} \tilde{a}_i X^i$, a_i flottant en précision p_i ;
- tel que $\tilde{P}$ minimise $\|P - \tilde{P}\|$.

Le problème à résoudre

- Entrée : P à coefficients réels, précisions p_i, $i = 0..\deg P$;
- Sortie : $\tilde{P} = \sum_{i=0}^{\deg P} \tilde{a}_i X^i$, a_i flottant en précision p_i ;
- tel que $\tilde{P}$ minimise $\|P - \tilde{P}\|$.
- Idée naturelle : arrondir a_i en précision p_i ;

Le problème à résoudre

- Entrée : P à coefficients réels, précisions p_i, $i = 0..\deg P$;
- Sortie : $\tilde{P} = \sum_{i=0}^{\deg P} \tilde{a}_i X^i$, a_i flottant en précision p_i ;
- tel que $\tilde{P}$ minimise $\|P - \tilde{P}\|$.
- Idée naturelle : arrondir a_i en précision p_i ;
- Loin d'être optimal ;

Le problème à résoudre

- Entrée : P à coefficients réels, précisions p_i, $i = 0..\deg P$;
- Sortie : $\tilde{P} = \sum_{i=0}^{\deg P} \tilde{a}_i X^i$, a_i flottant en précision p_i ;
- tel que $\tilde{P}$ minimise $\|P - \tilde{P}\|$.
- Idée naturelle : arrondir a_i en précision p_i ;
- Loin d'être optimal ;
- Empire quand le degré croît.

Le problème à résoudre

- Entrée : P à coefficients réels, précisions p_i, $i = 0..\deg P$;
- Sortie : $\tilde{P} = \sum_{i=0}^{\deg P} \tilde{a}_i X^i$, a_i flottant en précision p_i ;
- tel que $\tilde{P}$ minimise $\|P - \tilde{P}\|$.
- Idée naturelle : arrondir a_i en précision p_i ;
- Loin d'être optimal ;
- Empire quand le degré croît.
- Pratique : borne pour $\|P - \tilde{P}\|$ fixée, minimiser $\deg P$ et les p_i.

Réduction du cas flottant au cas fixe

- Choisir l'exposant de $\tilde{a}_i$ comme l'exposant a_i ;

Réduction du cas flottant au cas fixe

- Choisir l'exposant de $\tilde{a}_i$ comme l'exposant a_i ;
- $\Rightarrow$ virgule fixe ;

Réduction du cas flottant au cas fixe

- Choisir l'exposant de $\tilde{a}_i$ comme l'exposant a_i ;
- $\Rightarrow$ virgule fixe ;
- Vérifier que les résultats sont dans $\pm[1/2, 1]$;

Réduction du cas flottant au cas fixe

- Choisir l'exposant de $\tilde{a}_i$ comme l'exposant a_i ;
- $\Rightarrow$ virgule fixe ;
- Vérifier que les résultats sont dans $\pm[1/2, 1]$;
- Sinon modifier les exposants + itérer.

Réduction du cas flottant au cas fixe

- Choisir l'exposant de $\tilde{a}_i$ comme l'exposant a_i ;
- $\Rightarrow$ virgule fixe ;
- Vérifier que les résultats sont dans $\pm[1/2, 1]$;
- Sinon modifier les exposants + itérer.
- Pratique : marche en 2 itérations au plus.

Réduction du cas flottant au cas fixe

- Choisir l'exposant de $\tilde{a}_i$ comme l'exposant a_i ;
- $\Rightarrow$ virgule fixe ;
- Vérifier que les résultats sont dans $\pm[1/2, 1]$;
- Sinon modifier les exposants + itérer.
- Pratique : marche en 2 itérations au plus.
- Effet petit degré ?

Partie 2 : Le cas L^2

Formalisation

Appr-L^2 :

- Données :
  - $\varphi_0, \ldots, \varphi_n$ des fonctions, typiquement $2^{-e_i} X^i$; μ une mesure ;
  - f une fonction à approcher ;
- Sortie :
  - $x \in \mathbb{Z}^n$ tq. $\int_I (f - \sum_{i=1}^d x_i \varphi_i)^2 d\mu$ petit (minimal ?).
  - ... ou encore $\int_I (\pi(f) - \sum_{i=1}^d x_i \varphi_i)^2 d\mu$;
  - ie. minimiser $q(x - t)$, q quadratique, $t \in \mathbb{R}^n$, $x \in \mathbb{Z}^n$

Interlude : Réseaux euclidiens

Réseaux euclidiens

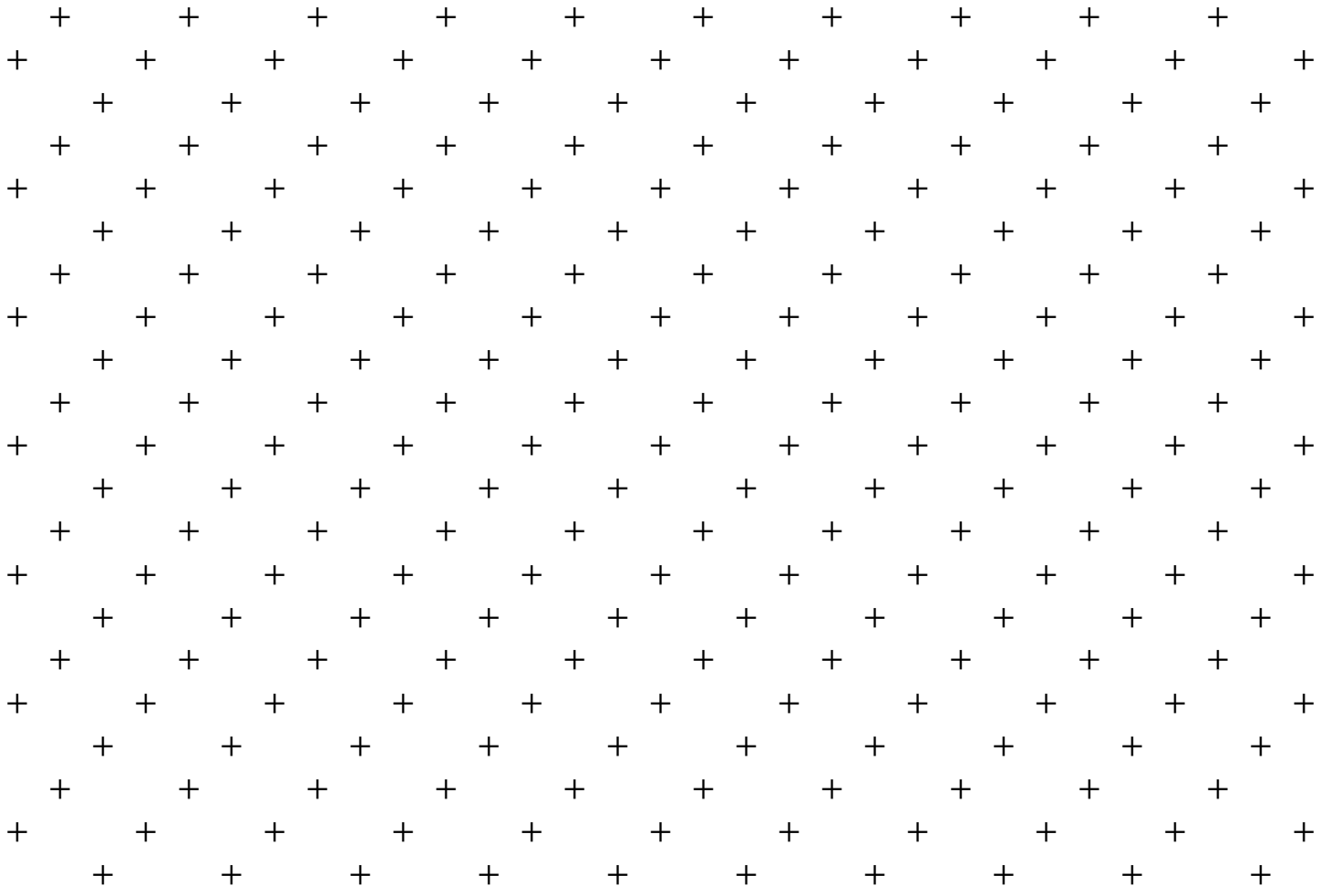
Réseau de $\mathbb{R}^d$ = grille régulière de $\mathbb{R}^d$

Réseaux euclidiens

Réseau de $\mathbb{R}^d$ = grille régulière de $\mathbb{R}^d$ (= sous-groupe discret de $\mathbb{R}^d$) ;

Réseaux euclidiens

Réseau de $\mathbb{R}^d$ = grille régulière de $\mathbb{R}^d$ (= sous-groupe discret de $\mathbb{R}^d$) ;

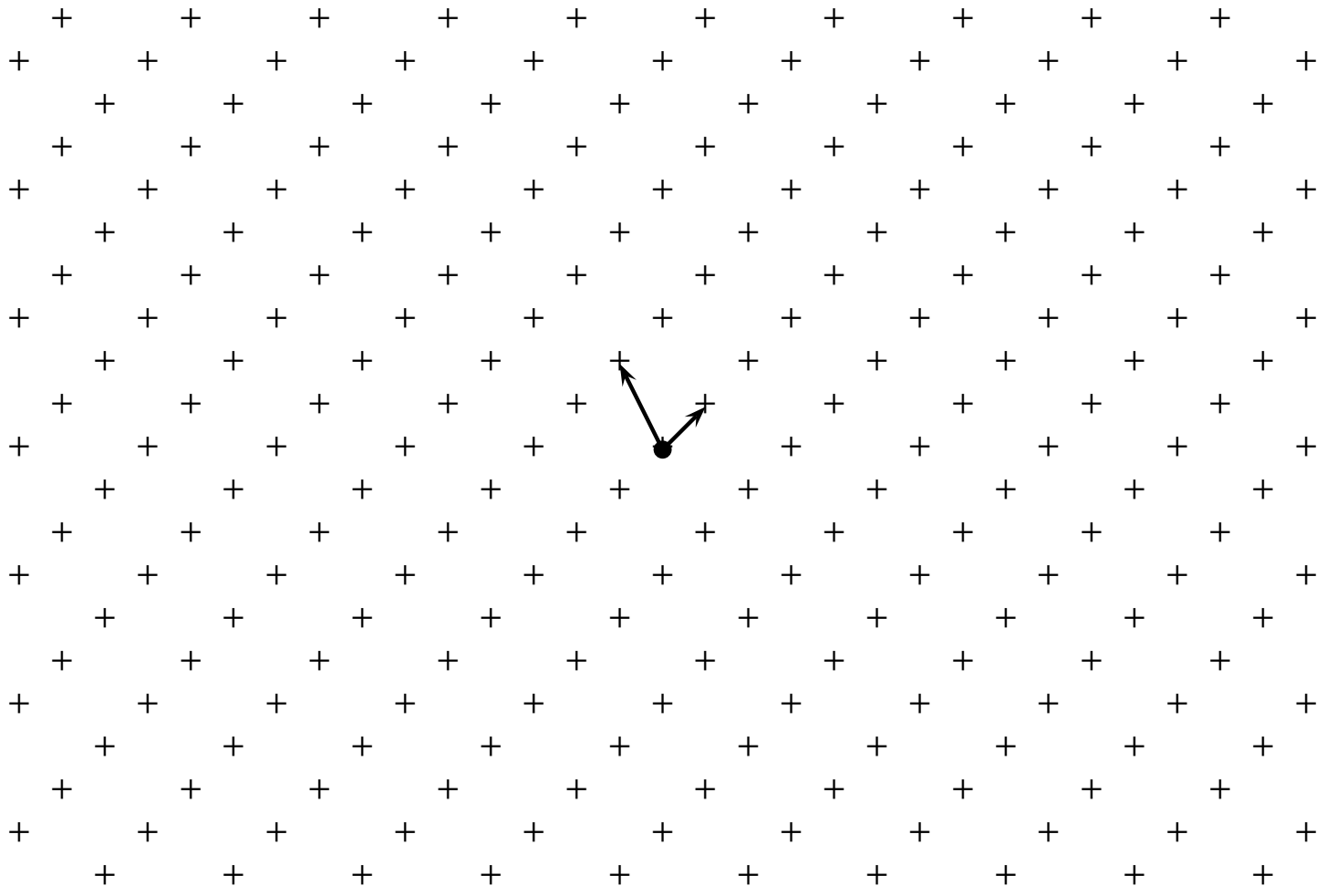


Réseaux euclidiens – bases

Réseau de $\mathbb{R}^d$ = combinaisons linéaires *entières* de vecteurs de base $b_1, \ldots, b_d$.

Réseaux euclidiens – bases

Réseau de $\mathbb{R}^d$ = combinaisons linéaires *entières* de vecteurs de base $b_1, \ldots, b_d$.



Réseaux euclidiens – point de vue équivalent

- (b_i) base d'un réseau $\Rightarrow (x_i) \mapsto \| \sum_{i=1}^{d} x_i b_i \|^2$ forme quadratique > 0 ;

Réseaux euclidiens – point de vue équivalent

- (b_i) base d'un réseau $\Rightarrow (x_i) \mapsto \| \sum_{i=1}^{d} x_i b_i \|^2$ forme quadratique > 0 ;
- Réciproquement q définie positive $\Rightarrow$ il existe une base (b_i) correspondante ;

Réseaux euclidiens – point de vue équivalent

- (b_i) base d'un réseau $\Rightarrow (x_i) \mapsto \| \sum_{i=1}^{d} x_i b_i \|^2$ forme quadratique > 0 ;
- Réciproquement q définie positive $\Rightarrow$ il existe une base (b_i) correspondante ;
- Matriciellement $(b_i) \hookrightarrow M \mapsto G := M^t M \hookrightarrow q$;

Réseaux euclidiens – point de vue équivalent

- (b_i) base d'un réseau $\Rightarrow (x_i) \mapsto \| \sum_{i=1}^{d} x_i b_i \|^2$ forme quadratique > 0 ;
- Réciproquement q définie positive $\Rightarrow$ il existe une base (b_i) correspondante ;
- Matriciellement $(b_i) \hookrightarrow M \mapsto G := M^t M \hookrightarrow q$;
- Réciproquement algorithme de Cholesky ;

Réseaux euclidiens – point de vue équivalent

- (b_i) base d'un réseau $\Rightarrow (x_i) \mapsto \| \sum_{i=1}^{d} x_i b_i \|^2$ forme quadratique > 0 ;
- Réciproquement q définie positive $\Rightarrow$ il existe une base (b_i) correspondante ;
- Matriciellement $(b_i) \hookrightarrow M \mapsto G := M^t M \hookrightarrow q$;
- Réciproquement algorithme de Cholesky ;
- Deux versions de l'algorithmique des réseaux : bases / formes quadratiques.
- Plus généralement, réseau = sous-groupe discret de $\mathbb{R}^d$ + produit scalaire.

Réseaux euclidiens – bonnes et mauvaises bases

- Qu’est-ce qu’une bonne base ?

Réseaux euclidiens – bonnes et mauvaises bases

- Qu’est-ce qu’une bonne base ?
- Cas d’un espace vectoriel (euclidien) : base orthonormée ;

Réseaux euclidiens – bonnes et mauvaises bases

- Qu’est-ce qu’une bonne base ?
- Cas d’un espace vectoriel (euclidien) : base orthonormée ;
- Pas possible dans le cas des réseaux ;

Réseaux euclidiens – bonnes et mauvaises bases

- Qu’est-ce qu’une bonne base ?
- Cas d’un espace vectoriel (euclidien) : base orthonormée ;
- Pas possible dans le cas des réseaux ;
- Bases “presques orthonormées” :

Réseaux euclidiens – bonnes et mauvaises bases

- Qu’est-ce qu’une bonne base ?
- Cas d’un espace vectoriel (euclidien) : base orthonormée ;
- Pas possible dans le cas des réseaux ;
- Bases “presques orthonormées” :
  - Formées de vecteurs courts...

Réseaux euclidiens – bonnes et mauvaises bases

- Qu’est-ce qu’une bonne base ?
- Cas d’un espace vectoriel (euclidien) : base orthonormée ;
- Pas possible dans le cas des réseaux ;
- Bases “presques orthonormées” :
  - Formées de vecteurs courts...
  - et presque orthogonaux...

Réseaux euclidiens – bonnes et mauvaises bases

- Qu’est-ce qu’une bonne base ?
- Cas d’un espace vectoriel (euclidien) : base orthonormée ;
- Pas possible dans le cas des réseaux ;
- Bases “presques orthonormées” :
  - Formées de vecteurs courts...
  - et presque orthogonaux...

Point de vue quantitatif :

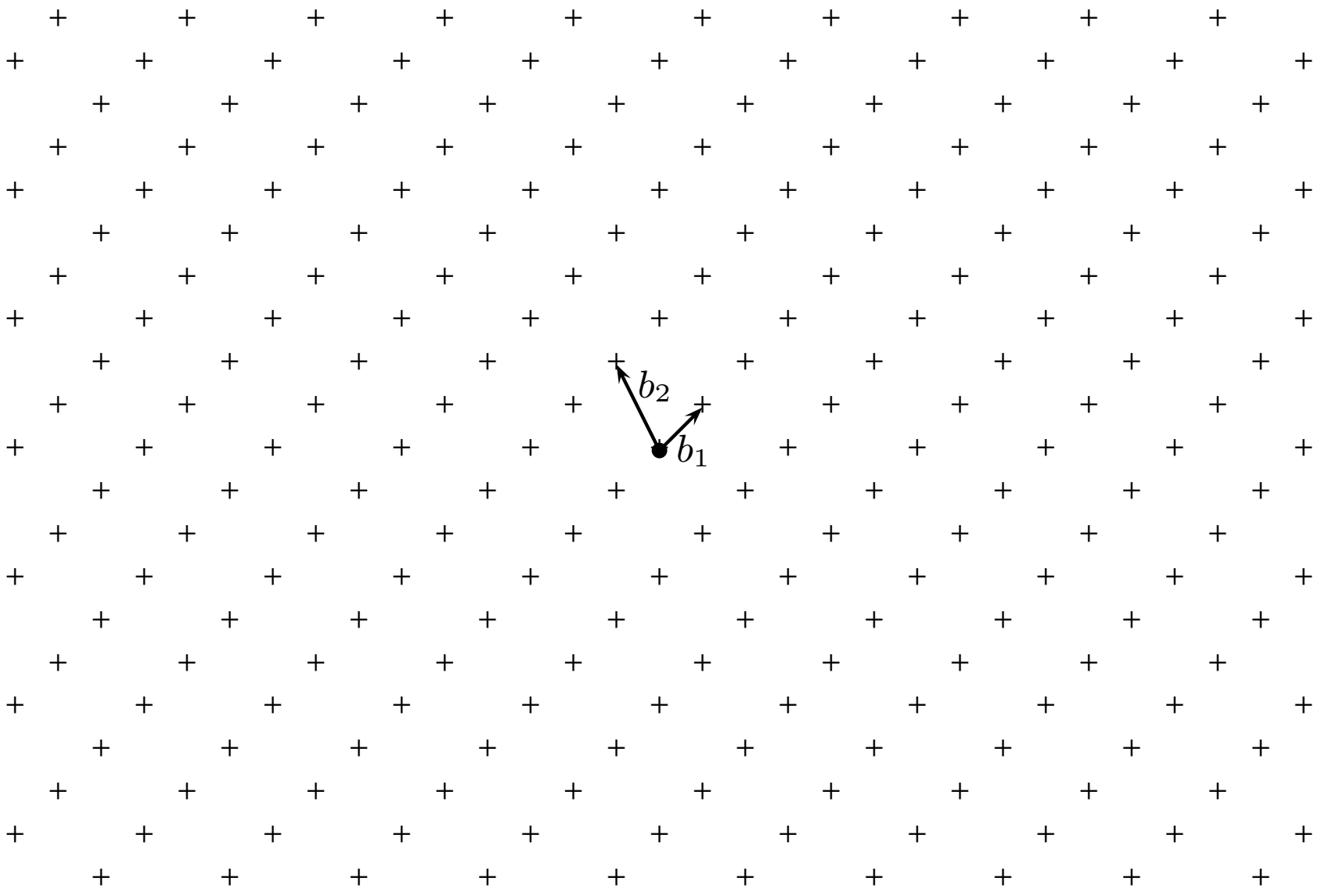
Réseaux euclidiens – bonnes et mauvaises bases

- Qu'est-ce qu'une bonne base ?
- Cas d'un espace vectoriel (euclidien) : base orthonormée ;
- Pas possible dans le cas des réseaux ;
- Bases "presques orthonormées" :
  - Formées de vecteurs courts...
  - et presque orthogonaux...

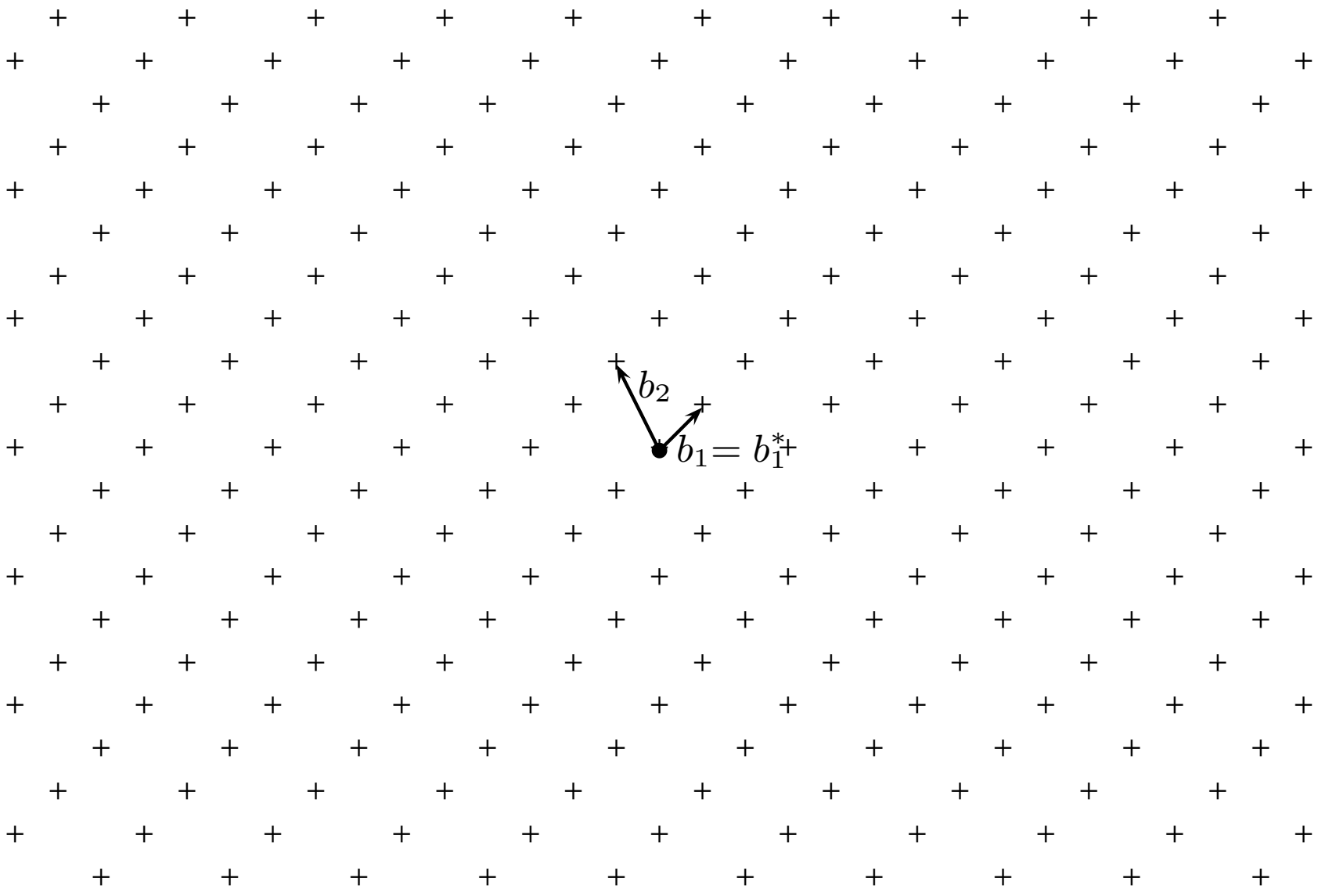
Point de vue quantitatif :

- Bonne base : base orthogonalisée de Gram-Schmidt $\left(b_i^*\right)$ est telle que $\left\|b_i^*\right\|$ décroît lentement.

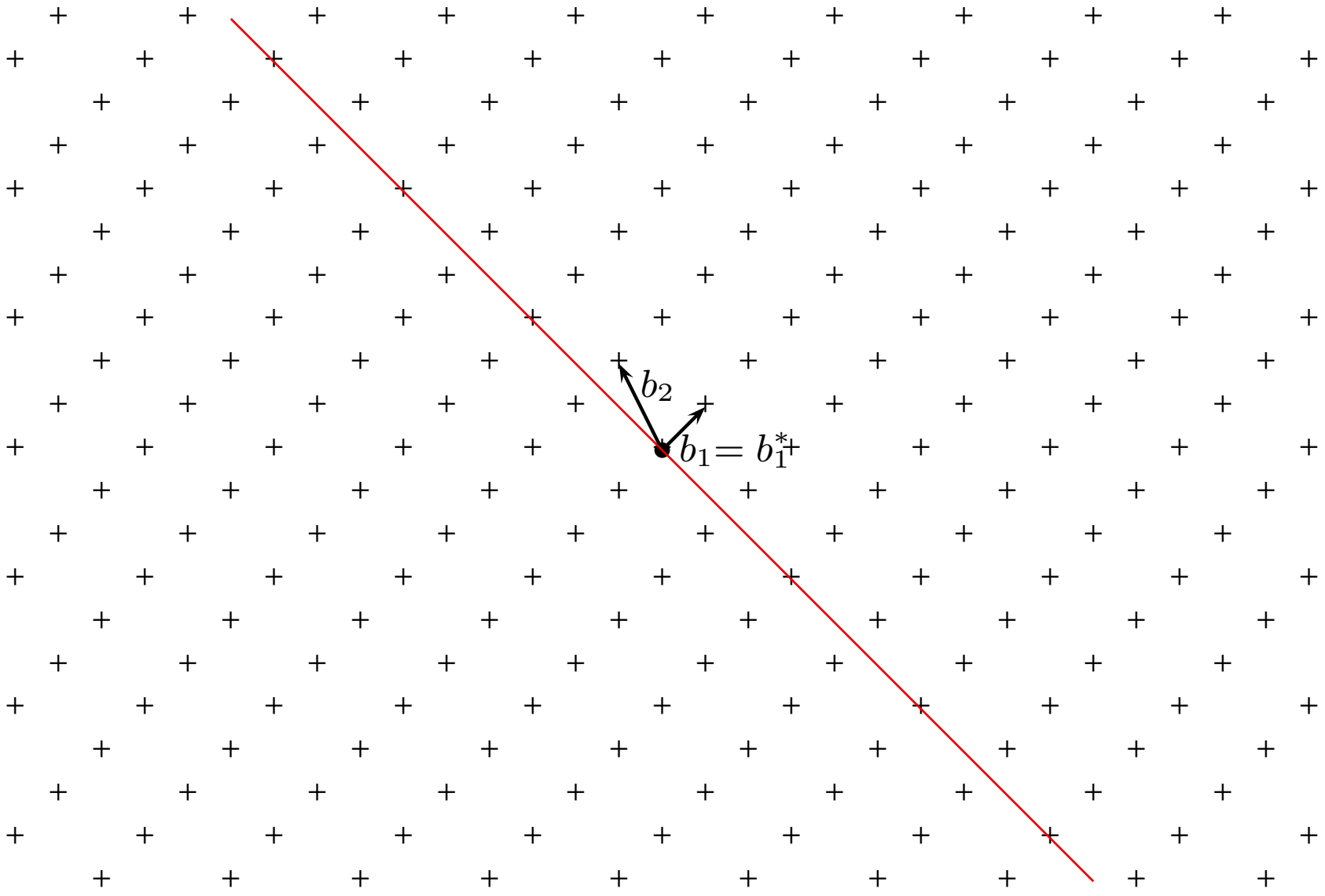
Réseaux euclidiens – bonnes et mauvaises bases



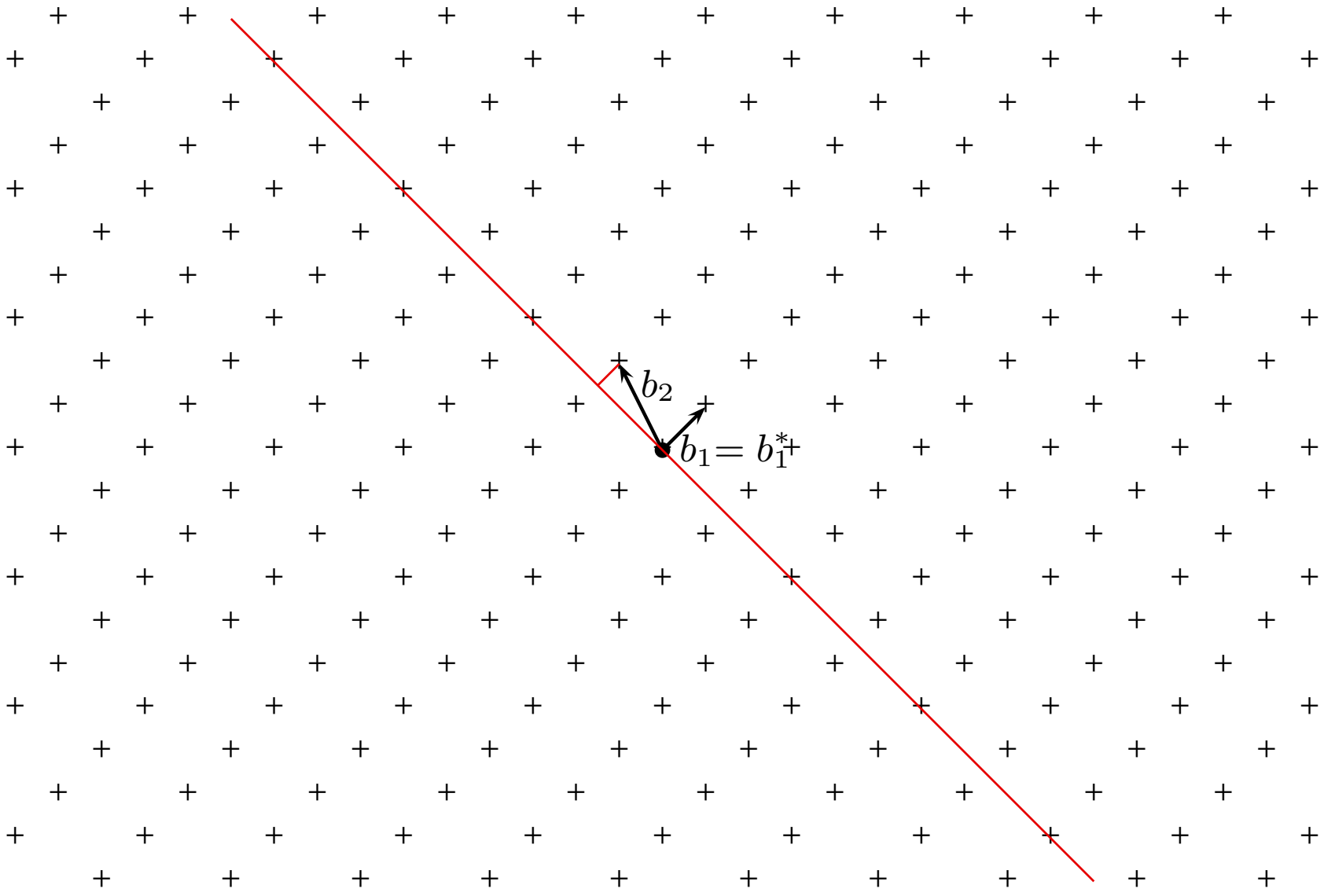
Réseaux euclidiens – bonnes et mauvaises bases



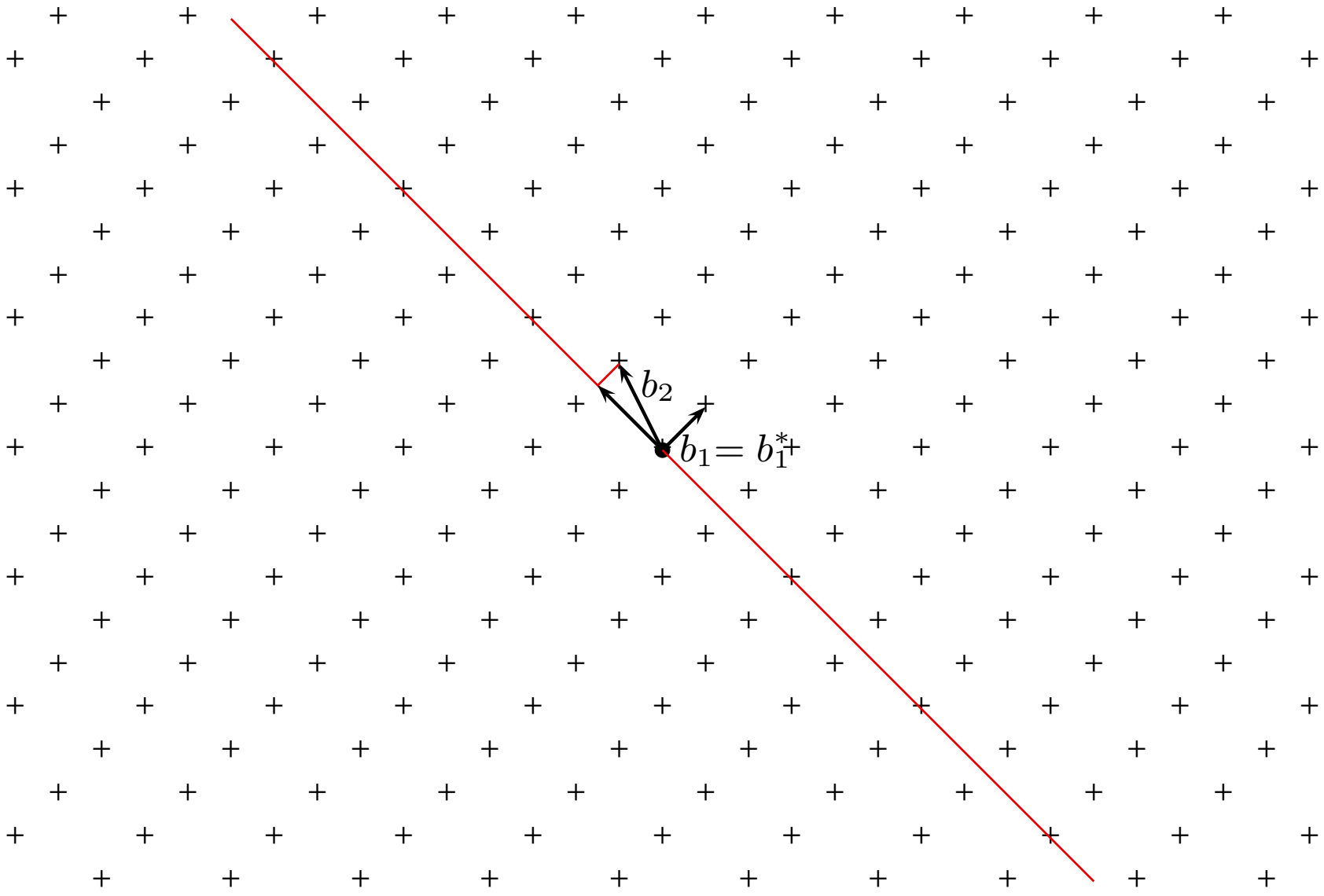
Réseaux euclidiens – bonnes et mauvaises bases



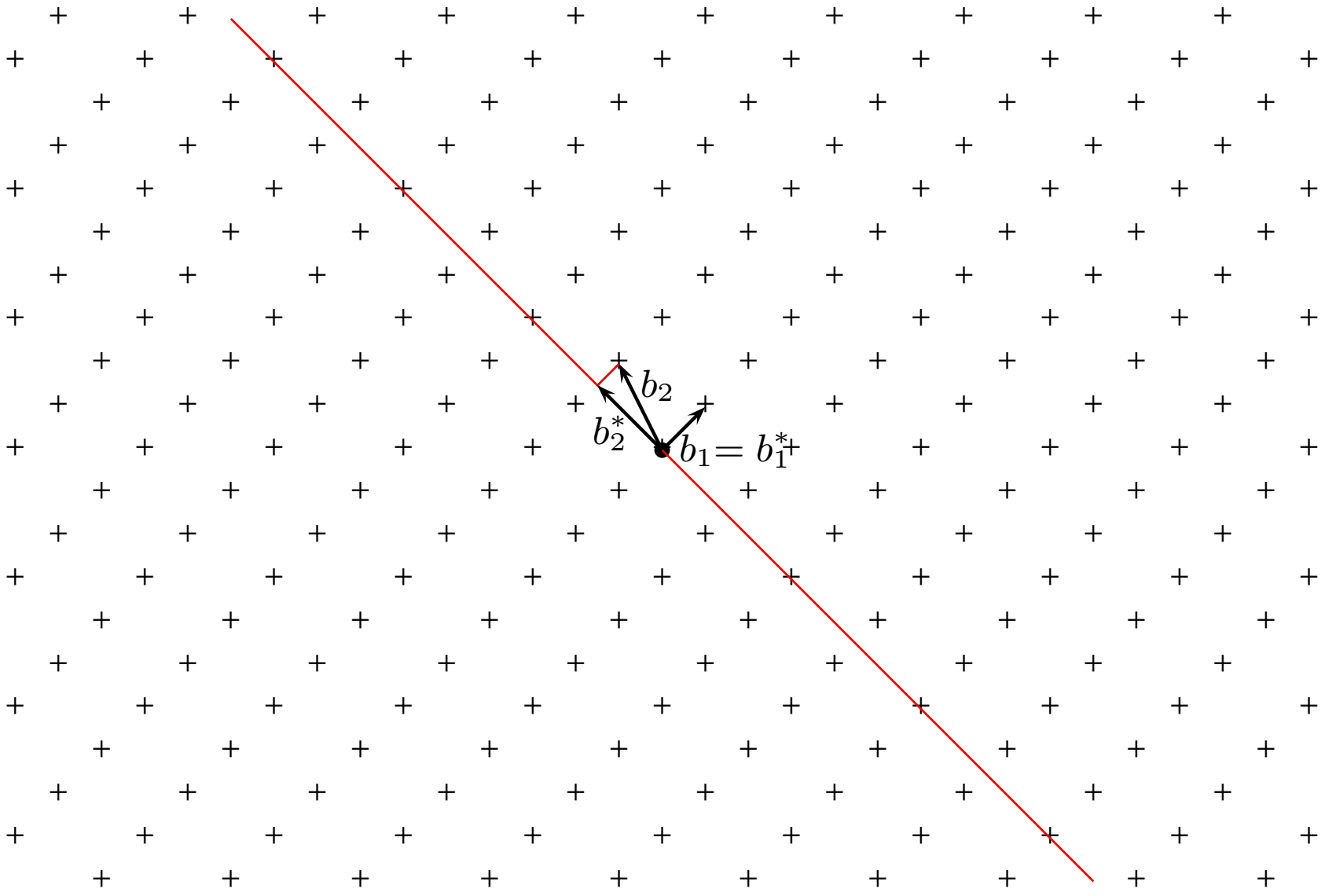
Réseaux euclidiens – bonnes et mauvaises bases



Réseaux euclidiens – bonnes et mauvaises bases

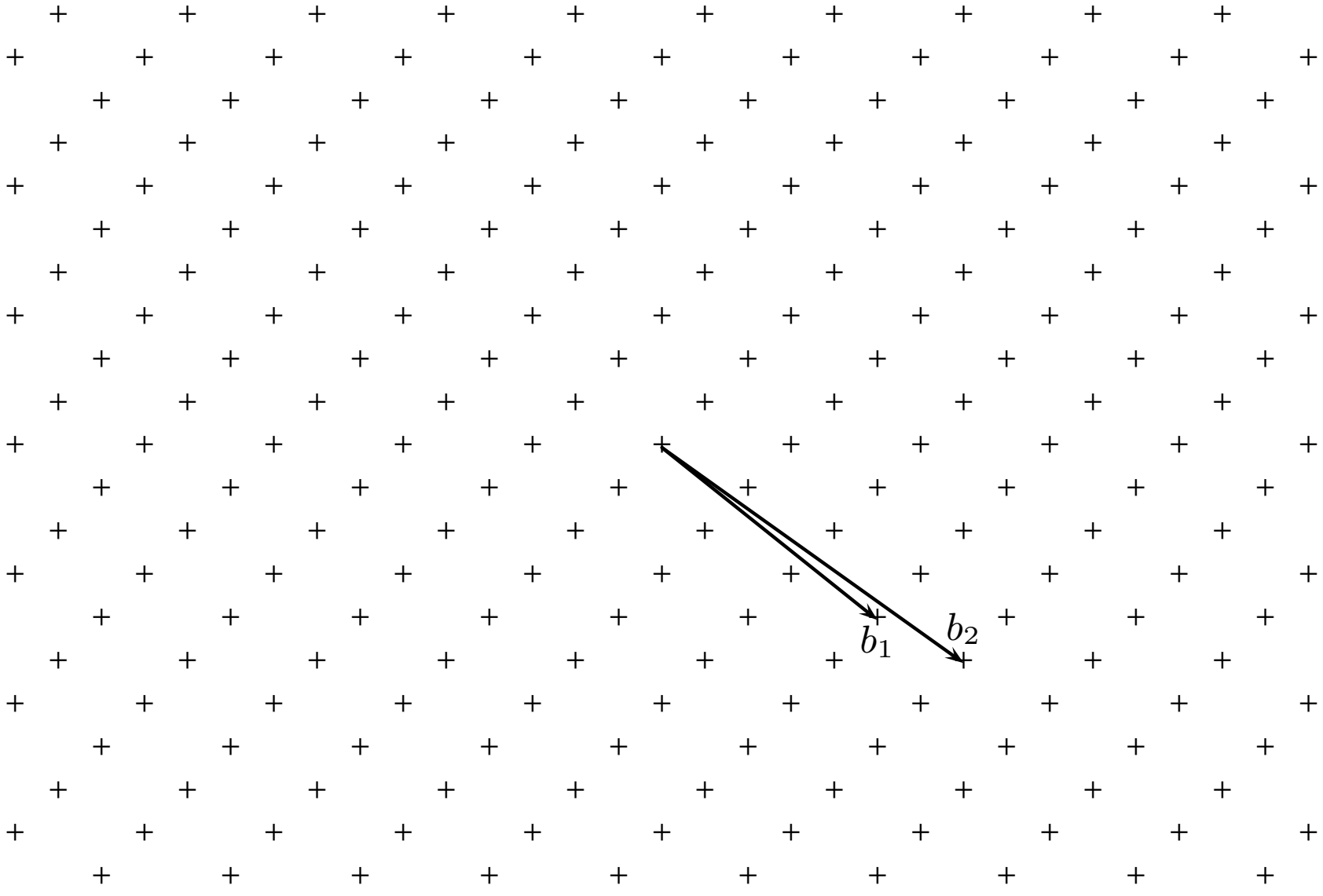


Réseaux euclidiens – bonnes et mauvaises bases



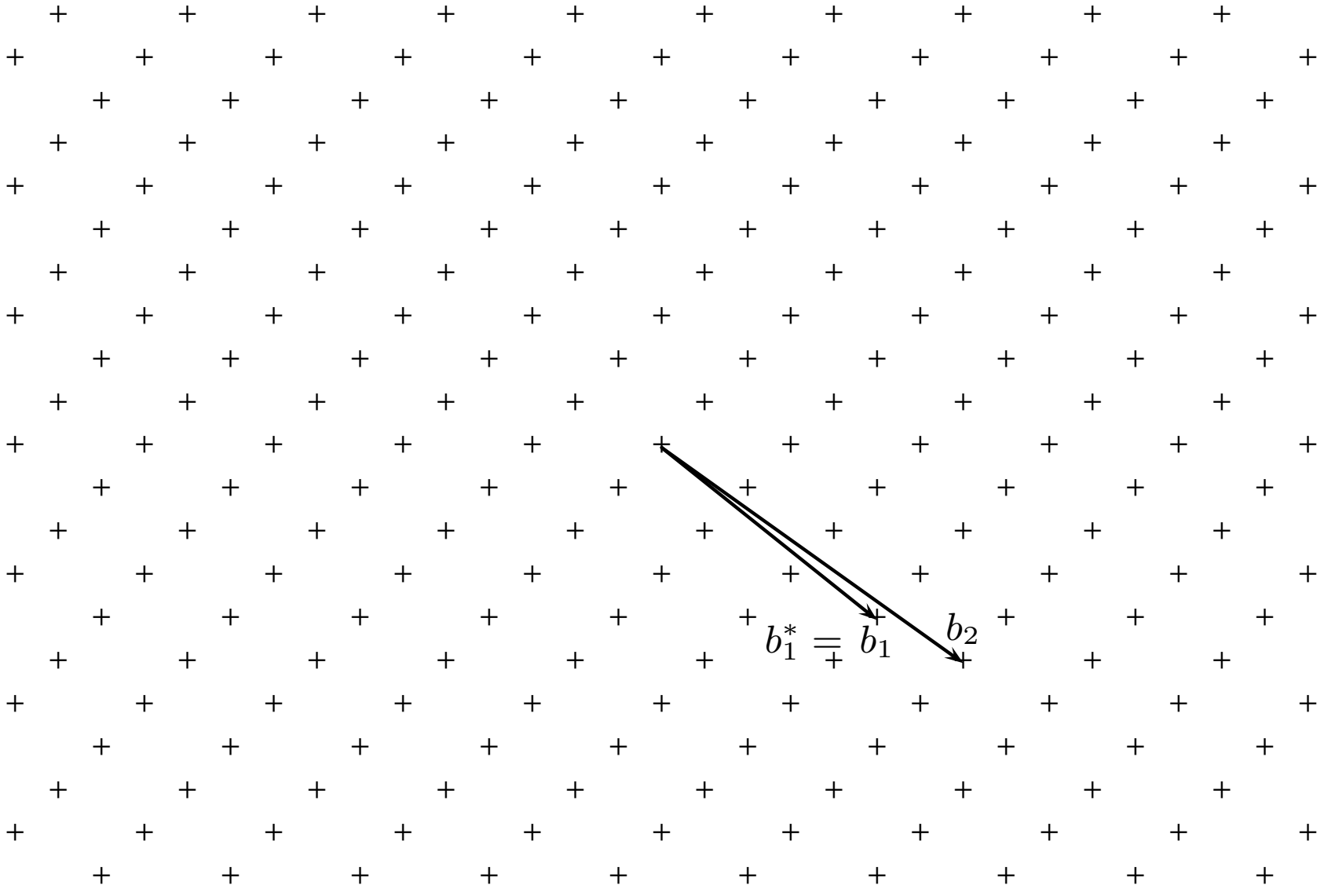
Réseaux euclidiens – bonnes et mauvaises bases

Point de vue quantitatif :



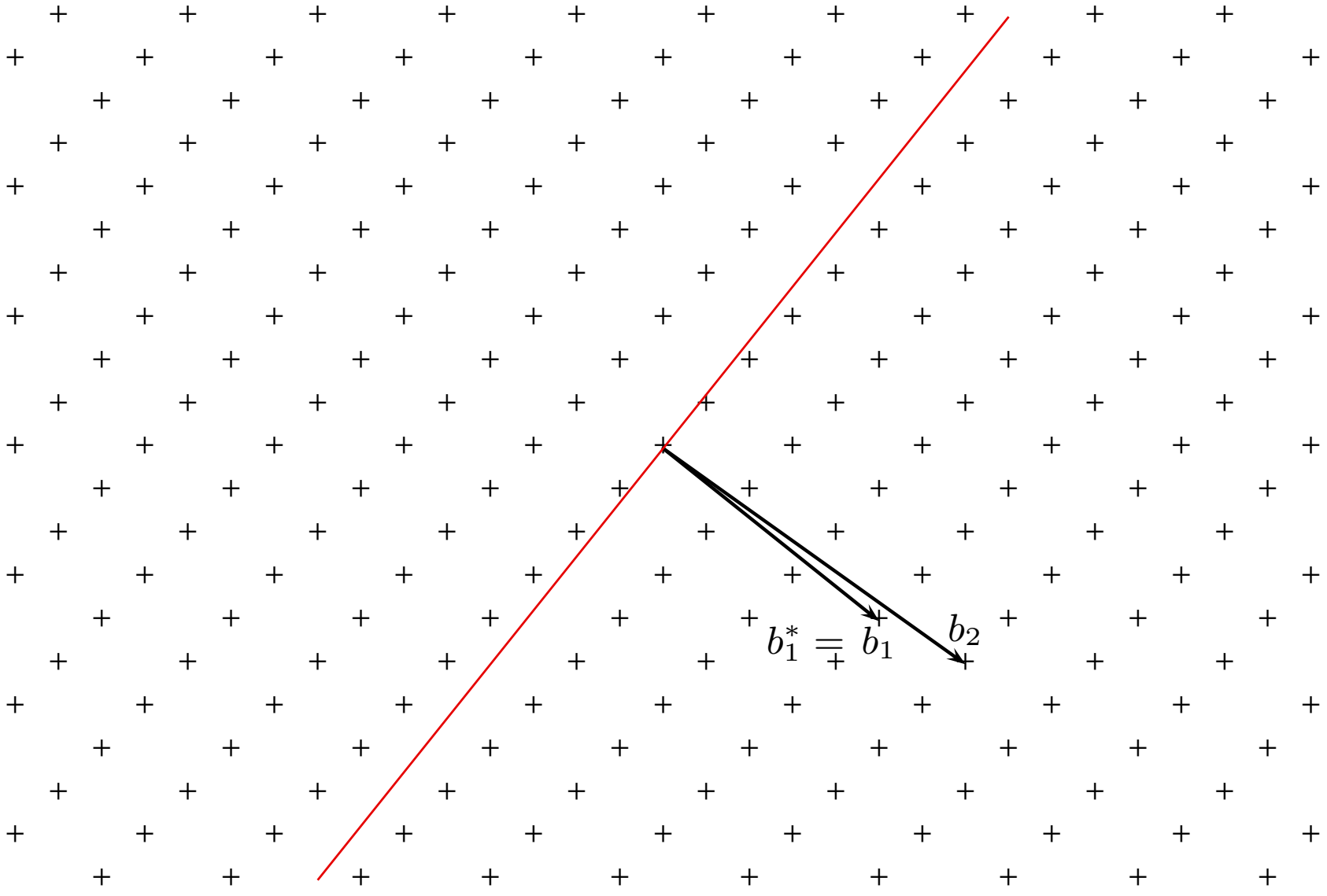
Réseaux euclidiens – bonnes et mauvaises bases

Point de vue quantitatif :



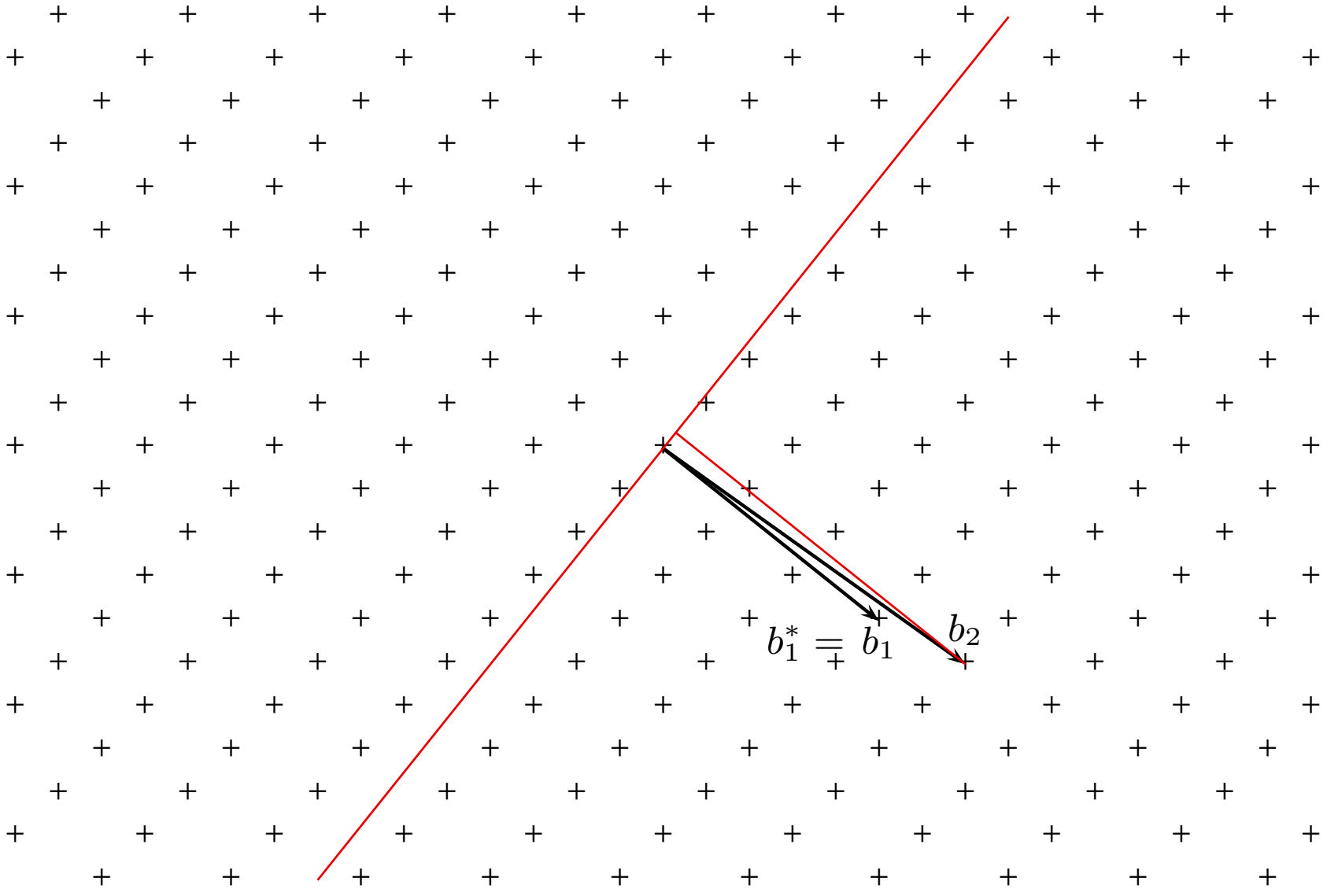
Réseaux euclidiens – bonnes et mauvaises bases

Point de vue quantitatif :



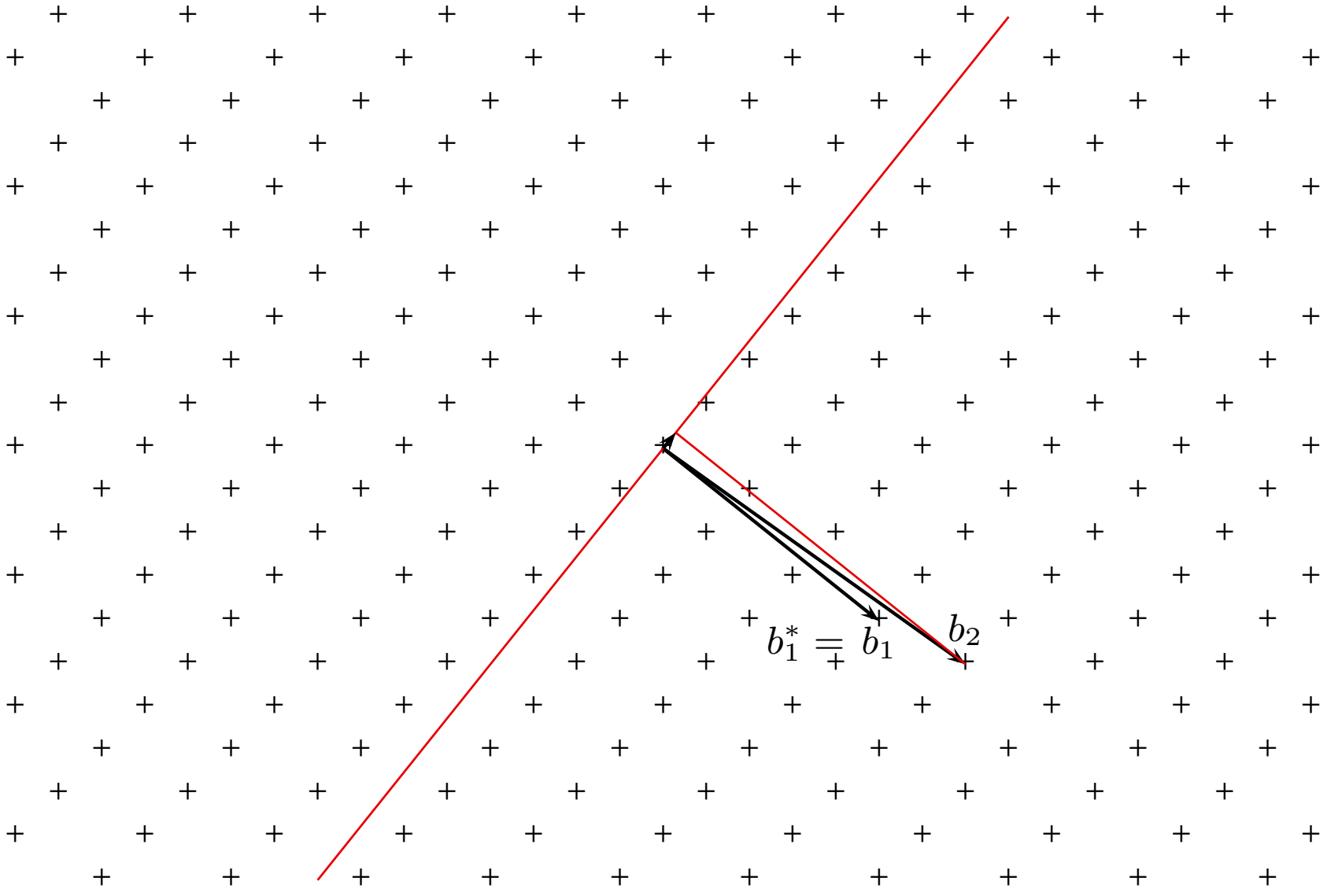
Réseaux euclidiens – bonnes et mauvaises bases

Point de vue quantitatif :



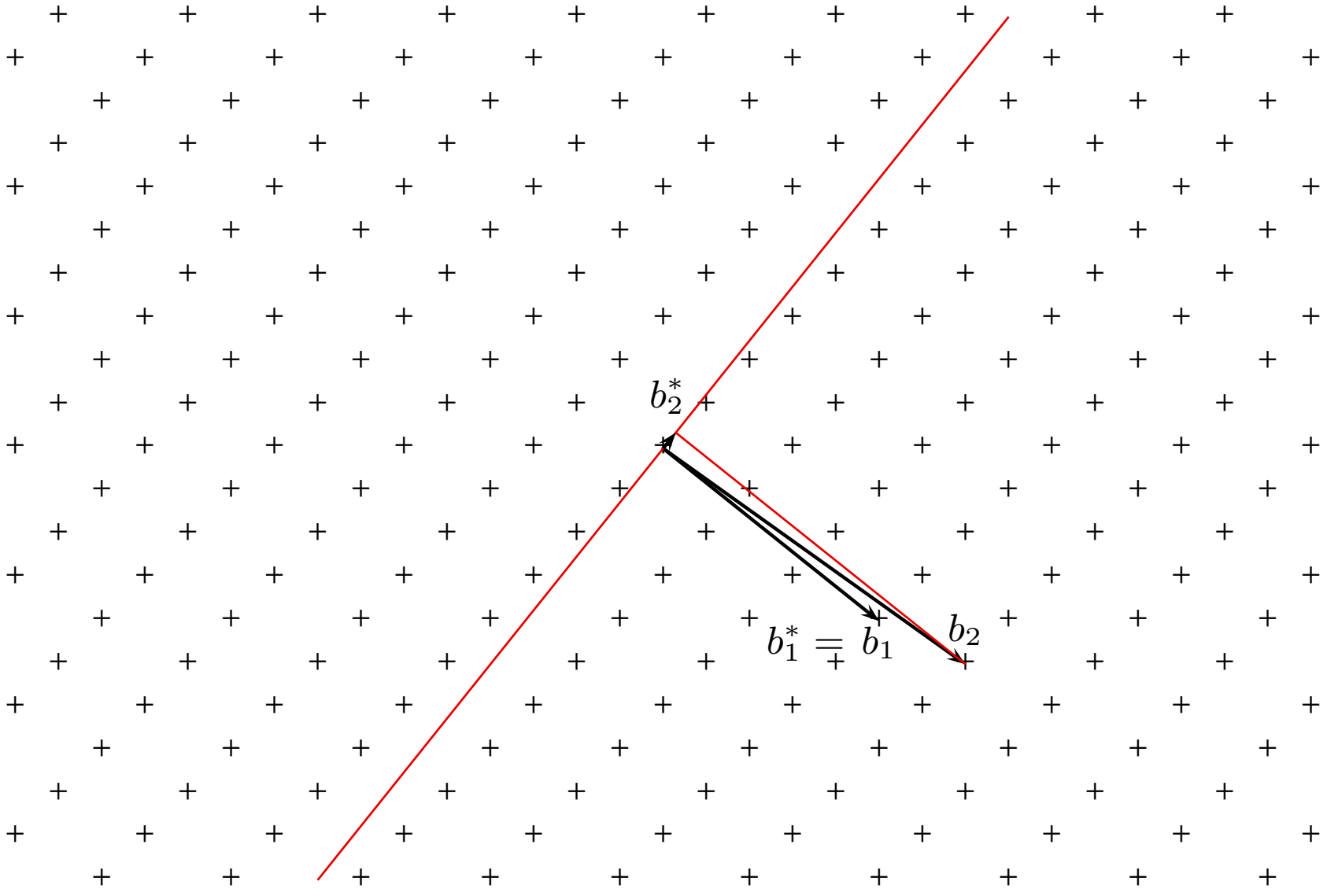
Réseaux euclidiens – bonnes et mauvaises bases

Point de vue quantitatif :



Réseaux euclidiens – bonnes et mauvaises bases

Point de vue quantitatif :



Réseaux euclidiens – problèmes fondamentaux

Donnée : base d'un réseau L de $\mathbb{R}^d$.

Réseaux euclidiens – problèmes fondamentaux

Donnée : base d'un réseau L de $\mathbb{R}^d$.

- Un vecteur le plus court non nul de L ; **SVP**

Réseaux euclidiens – problèmes fondamentaux

Donnée : base d'un réseau L de $\mathbb{R}^d$.

- Un vecteur le plus court non nul de L ; **SVP**
- Étant donné $x \in \mathbb{R}^d$, un vecteur $v \in L$ qui minimise $\|v - x\|$; **CVP**

Réseaux euclidiens – problèmes fondamentaux

Donnée : base d'un réseau L de $\mathbb{R}^d$.

- Un vecteur le plus court non nul de L ; **SVP**
- Étant donné $x \in \mathbb{R}^d$, un vecteur $v \in L$ qui minimise $\|v - x\|$; **CVP**
- Étant donné une mauvaise base $(b_1, \ldots, b_d)$, construire une bonne base $(b'_1, \ldots, b'_d)$; **Réduction**

Réseaux euclidiens – problèmes fondamentaux

Donnée : base d'un réseau L de $\mathbb{R}^d$.

- Un vecteur le plus court non nul de L ; **SVP**
- Étant donné $x \in \mathbb{R}^d$, un vecteur $v \in L$ qui minimise $\|v - x\|$; **CVP**
- Étant donné une mauvaise base $(b_1, \ldots, b_d)$, construire une bonne base $(b'_1, \ldots, b'_d)$; **Réduction**

Difficulté :

Réseaux euclidiens – problèmes fondamentaux

Donnée : base d'un réseau L de $\mathbb{R}^d$.

- Un vecteur le plus court non nul de L ; **SVP**
- Étant donné $x \in \mathbb{R}^d$, un vecteur $v \in L$ qui minimise $\|v - x\|$; **CVP**
- Étant donné une mauvaise base $(b_1, \ldots, b_d)$, construire une bonne base $(b'_1, \ldots, b'_d)$; **Réduction**

Difficulté :

- SVP, CVP = (presque) NP-dur ;

Réseaux euclidiens – problèmes fondamentaux

Donnée : base d'un réseau L de $\mathbb{R}^d$.

- Un vecteur le plus court non nul de L ; **SVP**
- Étant donné $x \in \mathbb{R}^d$, un vecteur $v \in L$ qui minimise $\|v - x\|$; **CVP**
- Étant donné une mauvaise base $(b_1, \ldots, b_d)$, construire une bonne base $(b'_1, \ldots, b'_d)$; **Réduction**

Difficulté :

- SVP, CVP = (presque) NP-dur ;
- Réduction : de polynômial à NP-dur selon les exigences.

CVP – algorithmes

- Méthodes exactes, de complexité exponentielle voire super-exponentielle ;

CVP – algorithmes

- Méthodes exactes, de complexité exponentielle voire super-exponentielle ;
- Méthodes approchées :

CVP – algorithmes

- Méthodes exactes, de complexité exponentielle voire super-exponentielle ;
- Méthodes approchées :
  - Deux algorithmes dus à Babai ;

CVP – algorithmes

- Méthodes exactes, de complexité exponentielle voire super-exponentielle ;
- Méthodes approchées :
  - Deux algorithmes dus à Babai ;
  - de complexité essentiellement cubique ;

CVP – algorithmes

- Méthodes exactes, de complexité exponentielle voire super-exponentielle ;
- Méthodes approchées :
  - Deux algorithmes dus à Babai ;
  - de complexité essentiellement cubique ;
  - donnant des approximations à un facteur $2^{O(d)}$ près.

CVP – application à des relations linéaires approchées

- Données : $x_{i,j} \in \mathbb{R}^{d \times k}$ une famille de vecteurs, $k \leq d$, et $y_i \in \mathbb{R}^k$;

CVP – application à des relations linéaires approchées

- Données : $x_{i,j} \in \mathbb{R}^{d \times k}$ une famille de vecteurs, $k \leq d$, et $y_i \in \mathbb{R}^k$;
- Objectif : $\lambda_1, \ldots, \lambda_d \in \mathbb{Z}$ tels que $\left| \sum_{j=1}^{d} \lambda_j x_{i,j} - y_i \right|$ petit pour tout i.

CVP – application à des relations linéaires approchées

- Données : $x_{i,j} \in \mathbb{R}^{d \times k}$ une famille de vecteurs, $k \leq d$, et $y_i \in \mathbb{R}^k$;
- Objectif : $\lambda_1, \ldots, \lambda_d \in \mathbb{Z}$ tels que $\left| \sum_{j=1}^{d} \lambda_j x_{i,j} - y_i \right|$ petit pour tout i.
- L réseau engendré par les colonnes de

$$\begin{pmatrix} x_{1,1} & \cdots & x_{1,k+1} & \cdots & x_{1,d} \\ x_{2,1} & \cdots & x_{2,k+1} & \cdots & x_{2,d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{k,1} & \cdots & x_{k,k+1} & \cdots & x_{k,d} \end{pmatrix}$$

CVP, SVP – application à des relations linéaires approc

- $v \in L$ est de la forme $M \cdot (\lambda_i)_{1 \leq i \leq d}$;

CVP, SVP – application à des relations linéaires approc

- $v \in L$ est de la forme $M \cdot (\lambda_i)_{1 \leq i \leq d}$;
- $v = \textbf{CVP}\big(L, (y_i)_{1 \leq i \leq d}\big)$ veut dire

$$\left\| \sum_{j=1}^{d} \lambda_j x_{ij} - y_i \right\|_2^2$$

minimal ;

CVP, SVP – application à des relations linéaires approc

- $v \in L$ est de la forme $M \cdot (\lambda_i)_{1 \leq i \leq d}$;
- $v = \textbf{CVP}\big(L, (y_i)_{1 \leq i \leq d}\big)$ veut dire

$$\left\| \sum_{j=1}^{d} \lambda_j x_{ij} - y_i \right\|_2^2$$

  minimal ;
- Cauchy-Schwarz (ou équivalence des normes) $\Rightarrow \max_i \left| \sum_{j=1}^{d} \lambda_j x_{ij} - y_i \right|$ presque minimal (on perd $\sqrt{d}$) ;

CVP, SVP – application à des relations linéaires approc

- $v \in L$ est de la forme $M \cdot (\lambda_i)_{1 \leq i \leq d}$;
- $v = \textbf{CVP}\big(L, (y_i)_{1 \leq i \leq d}\big)$ veut dire

$$\left\| \sum_{j=1}^{d} \lambda_j x_{ij} - y_i \right\|_2^2$$

  minimal ;
- Cauchy-Schwarz (ou équivalence des normes) $\Rightarrow \max_i \left| \sum_{j=1}^{d} \lambda_j x_{ij} - y_i \right|$ presque minimal (on perd $\sqrt{d}$) ;
- Deux difficultés techniques :

CVP, SVP – application à des relations linéaires approc

- $v \in L$ est de la forme $M \cdot (\lambda_i)_{1 \leq i \leq d}$;
- $v =$ **CVP**$\left(L, (y_i)_{1 \leq i \leq d}\right)$ veut dire

$$\left\| \sum_{j=1}^{d} \lambda_j x_{ij} - y_i \right\|_2^2$$

  minimal ;
- Cauchy-Schwarz (ou équivalence des normes) $\Rightarrow \max_i \left| \sum_{j=1}^{d} \lambda_j x_{ij} - y_i \right|$ presque minimal (on perd $\sqrt{d}$) ;
- Deux difficultés techniques :
  - L ainsi défini n'est pas un réseau

CVP, SVP – application à des relations linéaires approc

- $v \in L$ est de la forme $M \cdot (\lambda_i)_{1 \leq i \leq d}$;
- $v = \textbf{CVP}\big(L, (y_i)_{1 \leq i \leq d}\big)$ veut dire

$$\left\| \sum_{j=1}^{d} \lambda_j x_{ij} - y_i \right\|_2^2$$

  minimal ;
- Cauchy-Schwarz (ou équivalence des normes) $\Rightarrow \max_i \left| \sum_{j=1}^{d} \lambda_j x_{ij} - y_i \right|$ presque minimal (on perd $\sqrt{d}$) ;
- Deux difficultés techniques :
  - L ainsi défini n'est pas un réseau [*ajouter des lignes "parasites"*] ;

CVP, SVP – application à des relations linéaires approc

- $v \in L$ est de la forme $M \cdot (\lambda_i)_{1 \leq i \leq d}$;
- $v = \textbf{CVP}\big(L, (y_i)_{1 \leq i \leq d}\big)$ veut dire

$$\left\| \sum_{j=1}^{d} \lambda_j x_{ij} - y_i \right\|_2^2$$

  minimal ;
- Cauchy-Schwarz (ou équivalence des normes) $\Rightarrow \max_i \left| \sum_{j=1}^{d} \lambda_j x_{ij} - y_i \right|$ presque minimal (on perd $\sqrt{d}$) ;
- Deux difficultés techniques :
  - L ainsi défini n'est pas un réseau [*ajouter des lignes "parasites"*] ;
  - les lignes parasites ajoutées modifient le problème

CVP, SVP – application à des relations linéaires approc

- $v \in L$ est de la forme $M \cdot (\lambda_i)_{1 \leq i \leq d}$;
- $v = \textbf{CVP}\big(L, (y_i)_{1 \leq i \leq d}\big)$ veut dire

$$\left\| \sum_{j=1}^{d} \lambda_j x_{ij} - y_i \right\|_2^2$$

  minimal ;
- Cauchy-Schwarz (ou équivalence des normes) $\Rightarrow \max_i \left| \sum_{j=1}^{d} \lambda_j x_{ij} - y_i \right|$ presque minimal (on perd $\sqrt{d}$) ;
- Deux difficultés techniques :
  - L ainsi défini n'est pas un réseau [*ajouter des lignes "parasites"*] ;
  - les lignes parasites ajoutées modifient le problème [*ajouter des poids*]

CVP, SVP – application à des relations linéaires approc

L engendré par les colonnes de

$$\begin{pmatrix} x_{1,1} & \cdots & x_{1,k+1} & \cdots & x_{1,d} \\ x_{2,1} & \cdots & x_{2,k+1} & \cdots & x_{2,d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{k,1} & \cdots & x_{k,k+1} & \cdots & x_{k,d} \\ & & & & \\ & & & & \\ & & & & \end{pmatrix}$$

CVP, SVP – application à des relations linéaires approc

L engendré par les colonnes de

$$\begin{pmatrix} x_{1,1} & \dots & x_{1,k+1} & \dots & x_{1,d} \\ x_{2,1} & \dots & x_{2,k+1} & \dots & x_{2,d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{k,1} & \dots & x_{k,k+1} & \dots & x_{k,d} \\ 0 & \dots & 1 & \dots & 0 \\ 0 & \vdots & 0 & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 1 \end{pmatrix}$$

CVP, SVP – application à des relations linéaires approc

L engendré par les colonnes de

$$\begin{pmatrix} Cx_{1,1} & \dots & Cx_{1,k+1} & \dots & Cx_{1,d} \\ Cx_{2,1} & \dots & Cx_{2,k+1} & \dots & Cx_{2,d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ Cx_{k,1} & \dots & Cx_{k,k+1} & \dots & Cx_{k,d} \\ 0 & \dots & 1 & \dots & 0 \\ 0 & \vdots & 0 & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 1 \end{pmatrix}$$

Difficulté

- **Thm.** (B.-H.) L réseau, v vecteur, il existe

Difficulté

- **Thm.** (B.-H.) L réseau, v vecteur, il existe
  - $\varphi_0, \ldots, \varphi_n$ fonctions

Difficulté

- **Thm.** (B.-H.) L réseau, v vecteur, il existe
  - $\varphi_0, \ldots, \varphi_n$ fonctions
  - μ une mesure positive

  tq.

Difficulté

- **Thm.** (B.-H.) L réseau, v vecteur, il existe
  - $\varphi_0, \ldots, \varphi_n$ fonctions
  - μ une mesure positive

  tq. CVP(L, v) équivaut à **Appr-**$L^2(\varphi_0, \ldots, \varphi_n, \mu)$.

Difficulté

- **Thm.** (B.-H.) L réseau, v vecteur, il existe
  - $\varphi_0, \ldots, \varphi_n$ fonctions
  - μ une mesure positive

  tq. CVP(L, v) équivaut à **Appr-**$L^2(\varphi_0, \ldots, \varphi_n, \mu)$.
- **Corollaire.**

Difficulté

- **Thm.** (B.-H.) L réseau, v vecteur, il existe
  - $\varphi_0, \ldots, \varphi_n$ fonctions
  - μ une mesure positive

  tq. CVP(L, v) équivaut à **Appr-**$L^2(\varphi_0, \ldots, \varphi_n, \mu)$.
- **Corollaire.**
  - **Appr-**L^2 NP-difficile,

Difficulté

- **Thm.** (B.-H.) L réseau, v vecteur, il existe
  - $\varphi_0, \ldots, \varphi_n$ fonctions
  - μ une mesure positive

  tq. CVP(L, v) équivaut à **Appr-**$L^2(\varphi_0, \ldots, \varphi_n, \mu)$.
- **Corollaire.**
  - **Appr-**L^2 NP-difficile,

Difficulté

- **Thm.** (B.-H.) L réseau, v vecteur, il existe
  - $\varphi_0, \ldots, \varphi_n$ fonctions
  - μ une mesure positive

  tq. CVP(L, v) équivaut à **Appr-**$L^2(\varphi_0, \ldots, \varphi_n, \mu)$.
- **Corollaire.**
  - **Appr-**L^2 NP-difficile,
  - même avec un facteur d'approximation $n^{1/\log\log n}$.

Difficulté

- **Thm.** (B.-H.) L réseau, v vecteur, il existe
  - $\varphi_0, \ldots, \varphi_n$ fonctions
  - μ une mesure positive

  tq. CVP(L, v) équivaut à **Appr-**$L^2(\varphi_0, \ldots, \varphi_n, \mu)$.
- **Corollaire.**
  - **Appr-**L^2 NP-difficile,
  - même avec un facteur d'approximation $n^{1/\log\log n}$.
- Donne un algorithme de résolution (exponentiel).

Pratique

Pratique

- Efficace en petite-moyenne dimension (jusqu’à 20-30) ;

Pratique

- Efficace en petite-moyenne dimension (jusqu'à 20-30) ;
- Largement suffisant pour les problèmes pertinents ;

Pratique

- Efficace en petite-moyenne dimension (jusqu'à 20-30) ;
- Largement suffisant pour les problèmes pertinents ;
- Très adaptable : polynômes de format imposé (pairs, impairs), cas bivarié, etc.

Exemple

- $\varphi_i(x) = x^i$, $i = 0..8$, $f(x) = \sin(\pi\sqrt{x})/\pi\sqrt{x}$ sur $I = [0, 1]$.

Exemple

- $\varphi_i(x) = x^i$, $i = 0..8$, $f(x) = \sin(\pi\sqrt{x})/\pi\sqrt{x}$ sur $I = [0, 1]$.
- Coefficients : flottants simple précision ; ($x = a2^{-24-e}, a \in [2^{23}, 2^{24} - 1]$.

Exemple

- $\varphi_i(x) = x^i$, $i = 0..8$, $f(x) = \sin(\pi\sqrt{x})/\pi\sqrt{x}$ sur $I = [0, 1]$.
- Coefficients : flottants simple précision ; ($x = a2^{-24-e}, a \in [2^{23}, 2^{24} - 1]$.
- Optimum réel : distance L^2 $3.06 \cdot 10^{-14}$;

Exemple

- $\varphi_i(x) = x^i$, $i = 0..8$, $f(x) = \sin(\pi\sqrt{x})/\pi\sqrt{x}$ sur $I = [0, 1]$.
- Coefficients : flottants simple précision ; ($x = a2^{-24-e}, a \in [2^{23}, 2^{24} - 1]$.
- Optimum réel : distance L^2 $3.06 \cdot 10^{-14}$;
- Arrondi de l'optimum réel : distance L^2 $5.06 \cdot 10^{-9}$;

Exemple

- $\varphi_i(x) = x^i$, $i = 0..8$, $f(x) = \sin(\pi\sqrt{x})/\pi\sqrt{x}$ sur $I = [0, 1]$.
- Coefficients : flottants simple précision ; ($x = a2^{-24-e}, a \in [2^{23}, 2^{24} - 1]$.
- Optimum réel : distance L^2 $3.06 \cdot 10^{-14}$;
- Arrondi de l'optimum réel : distance L^2 $5.06 \cdot 10^{-9}$;
- Produit par notre méthode : distance L^2 $4.34 \cdot 10^{-11}$;
- Facteur 100...