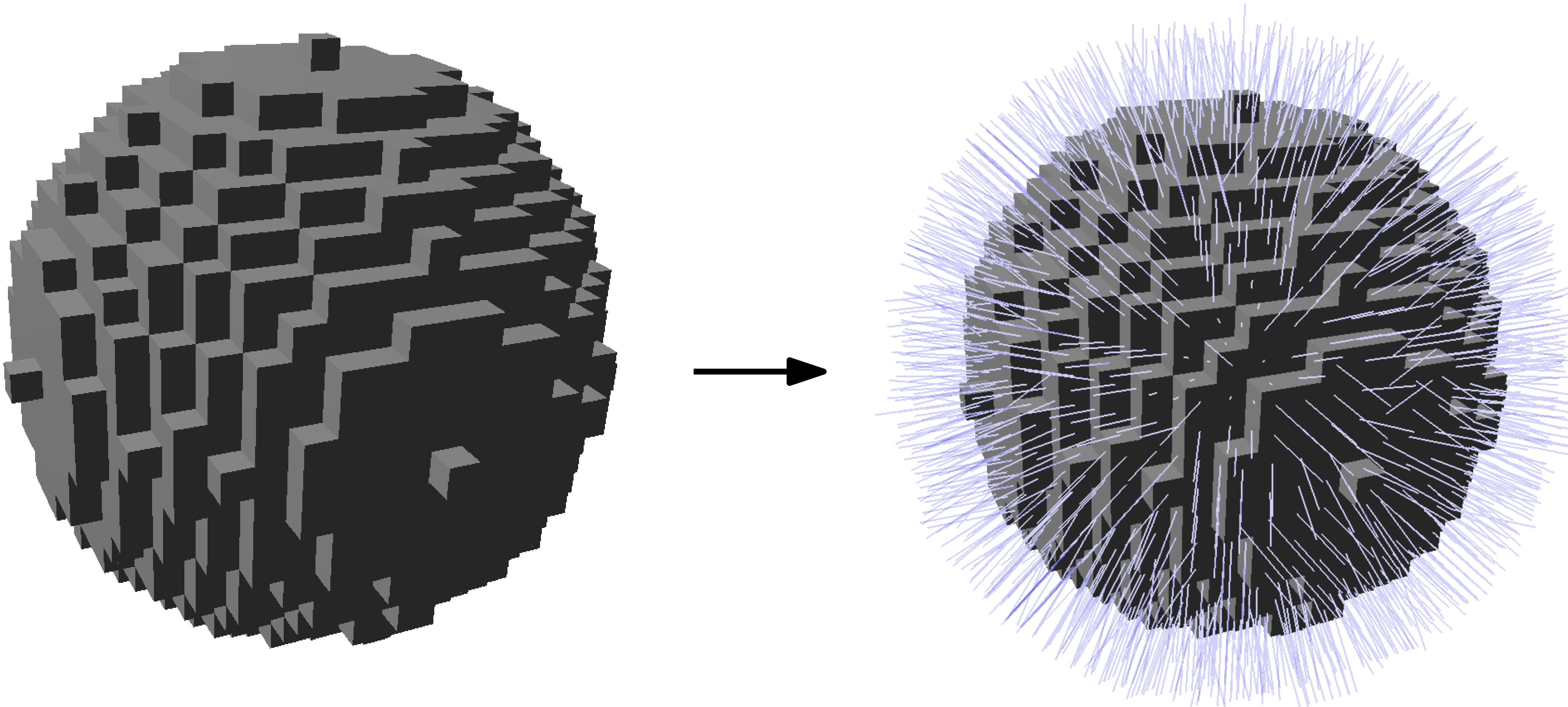
Une approche générale pour les algorithmes de plane-probing

Jacques-Olivier Lachaud, Univ. Savoie Mont Blanc, LAMA
Jocelyn Meyron, Univ. Lyon, INSA Lyon, LIRIS
Tristan Roussillon, Univ. Lyon, INSA Lyon, LIRIS

Journées de Géométrie Discrète et Morphologie Mathématique
Marseille, 12 novembre 2019

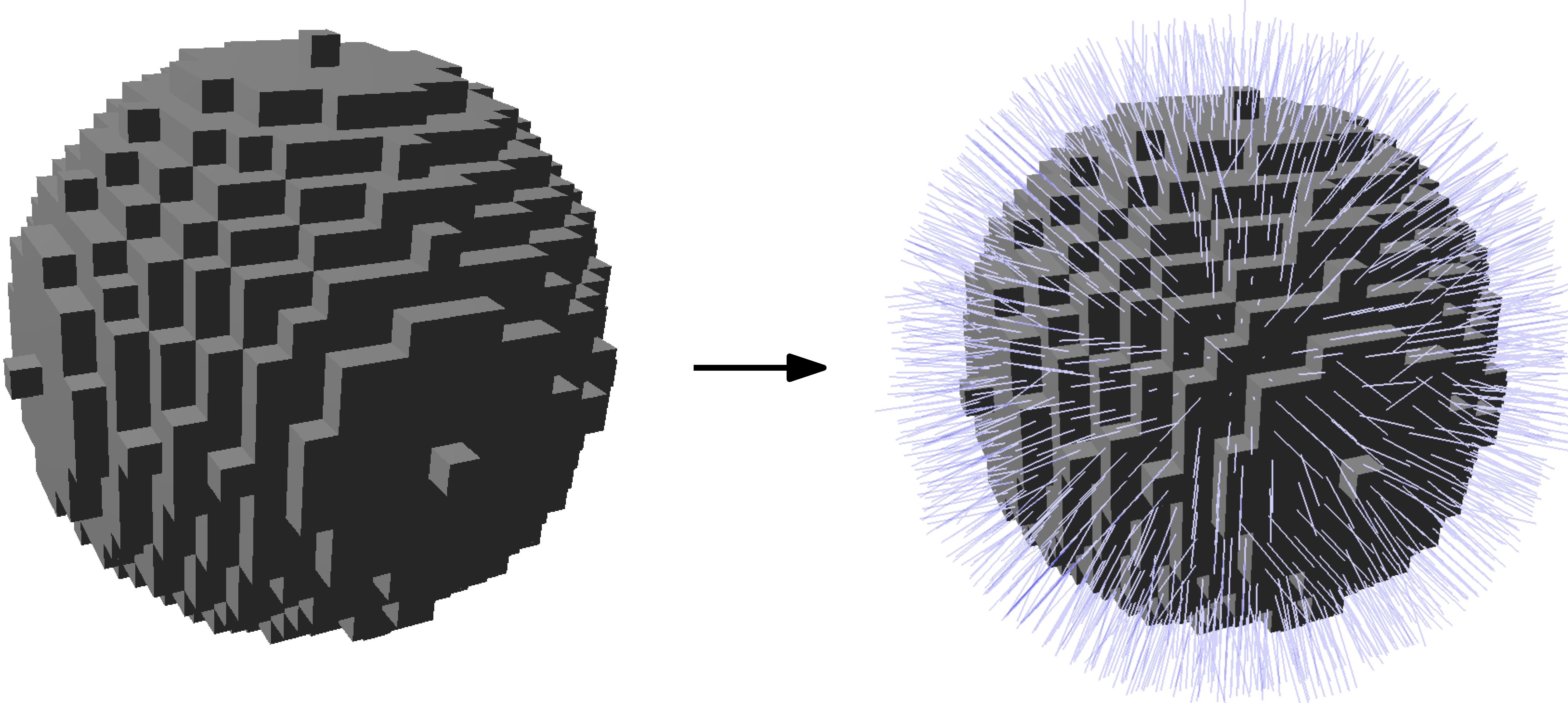
Contexte

Objectif : estimation sans paramètre des vecteurs normaux sur des surfaces digitales



Contexte

Objectif : estimation sans paramètre des vecteurs normaux sur des surfaces digitales



Approches possibles :

- Pour chaque surfel, regarder un voisinage et moyenner $\Longrightarrow$ paramètre de taille ?
- Utiliser des segments de plans digitaux $\Longrightarrow$ adaptation à la géométrie locale

Plans digitaux

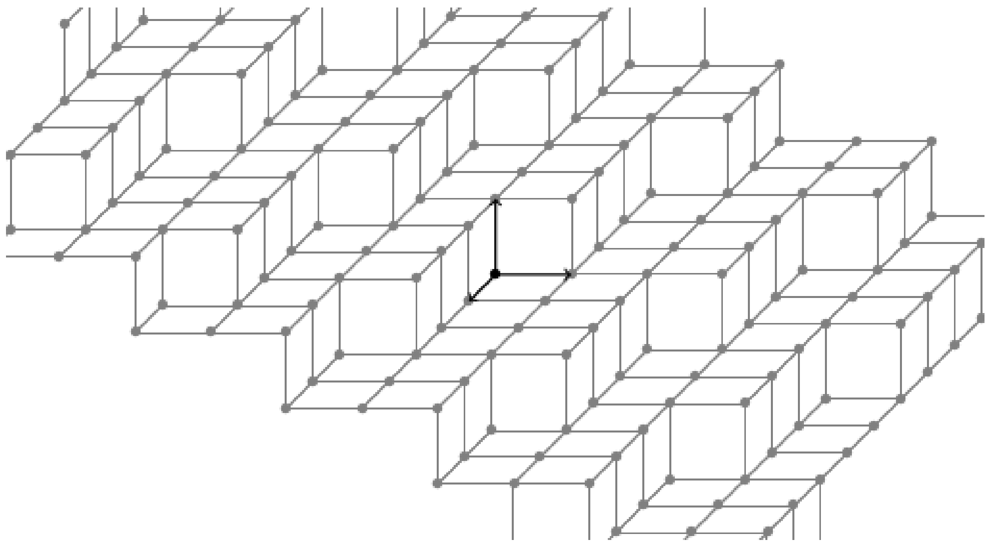
Plan digital standard et hauteur

$\mathbf{N}(a, b, c)$ un vecteur normal (avec $gcd(a, b, c) = 1$) et $\mu \in \mathbb{Z}$

$$\mathbf{P} = \{x \in \mathbb{Z}^3 \mid \mu \leq x \cdot \mathbf{N} < \mu + \|\mathbf{N}\|_1\}.$$

On note $\bar{x} = x \cdot \mathbf{N}$ (hauteur de x).

Segment de plan digital : sous ensemble 6-connexe d'un plan digital $\mathbf{P}$



Plans digitaux

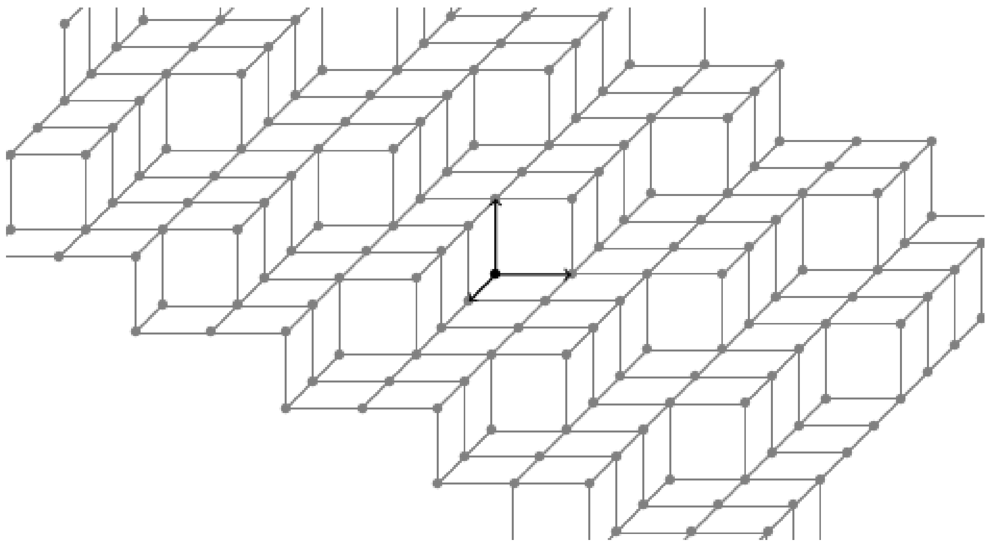
Plan digital standard et hauteur

$\mathbf{N}(a, b, c)$ un vecteur normal (avec $gcd(a, b, c) = 1$) et $\mu \in \mathbb{Z}$

$$\mathbf{P} = \{x \in \mathbb{Z}^3 \mid \mu \leq x \cdot \mathbf{N} < \mu + \|\mathbf{N}\|_1\}.$$

On note $\bar{x} = x \cdot \mathbf{N}$ (hauteur de x).

Segment de plan digital : sous ensemble 6-connexe d'un plan digital $\mathbf{P}$



Approche classique :

- Partir d'un ensemble de points S formant un segment de plan digital
- Choisir des points au fur et à mesure à ajouter tel que S reste un segment de plan digital $\implies$ heuristiques et paramètre de taille

Plans digitaux

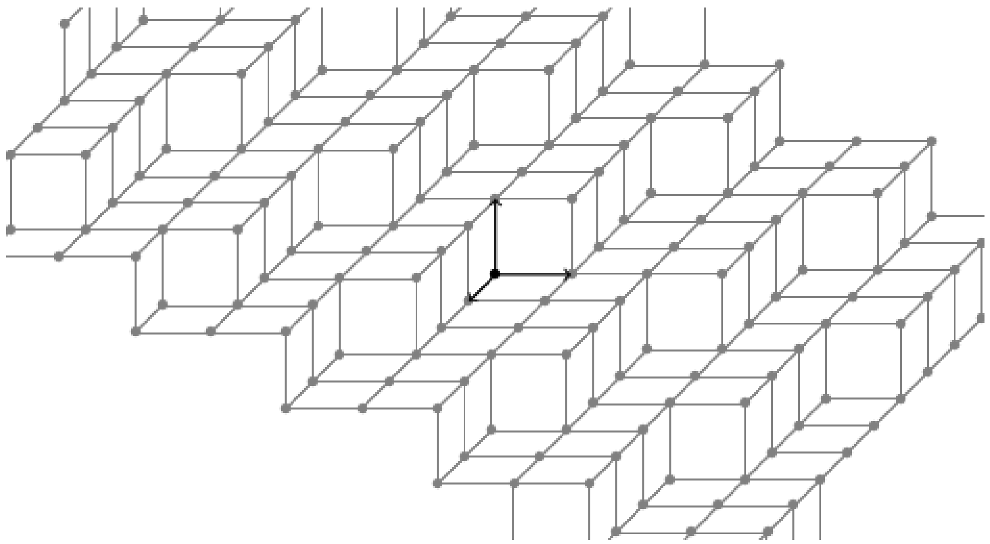
Plan digital standard et hauteur

$\mathbf{N}(a, b, c)$ un vecteur normal (avec $gcd(a, b, c) = 1$) et $\mu \in \mathbb{Z}$

$$\mathbf{P} = \{x \in \mathbb{Z}^3 \mid \mu \leq x \cdot \mathbf{N} < \mu + \|\mathbf{N}\|_1\}.$$

On note $\bar{x} = x \cdot \mathbf{N}$ (hauteur de x).

Segment de plan digital : sous ensemble 6-connexe d'un plan digital $\mathbf{P}$



Approche classique :

- Partir d'un ensemble de points S formant un segment de plan digital
- Choisir des points au fur et à mesure à ajouter tel que S reste un segment de plan digital $\Longrightarrow$ heuristiques et paramètre de taille

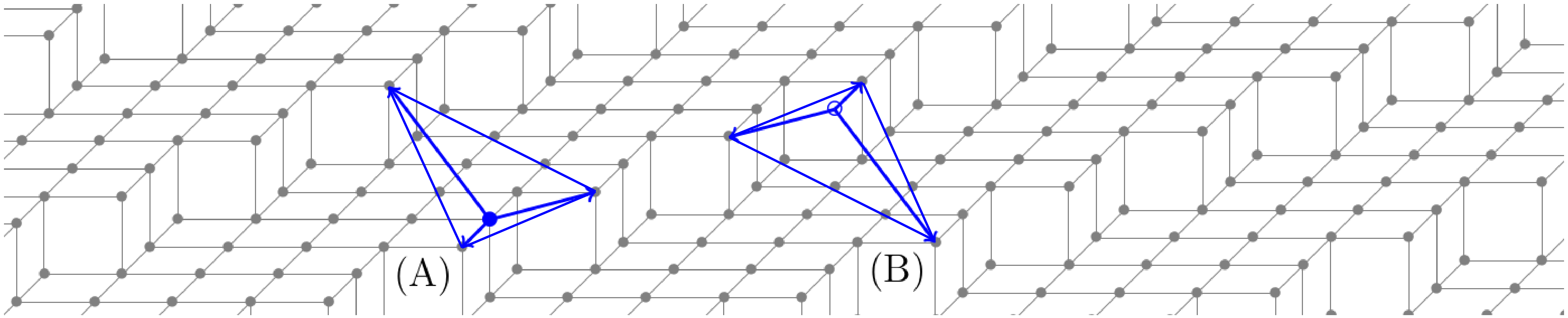
$\Longrightarrow$ Algorithmes de plane-probing : choix des points sans paramètre

Algorithmes de plane-probing

Prédicat $\mathrm{InPlane}$: $\mathrm{InPlane}(p)$ vrai $\iff p \in \mathbf{P}$

Initialisation sur un surfel : tétraèdre défini par p et (e_0, e_1, e_2)

Algorithme : déformer le tétraèdre jusqu'à ce que $\sum_k (e_k^{(i)} \times e_{k+1}^{(i)}) = \mathbf{N}$

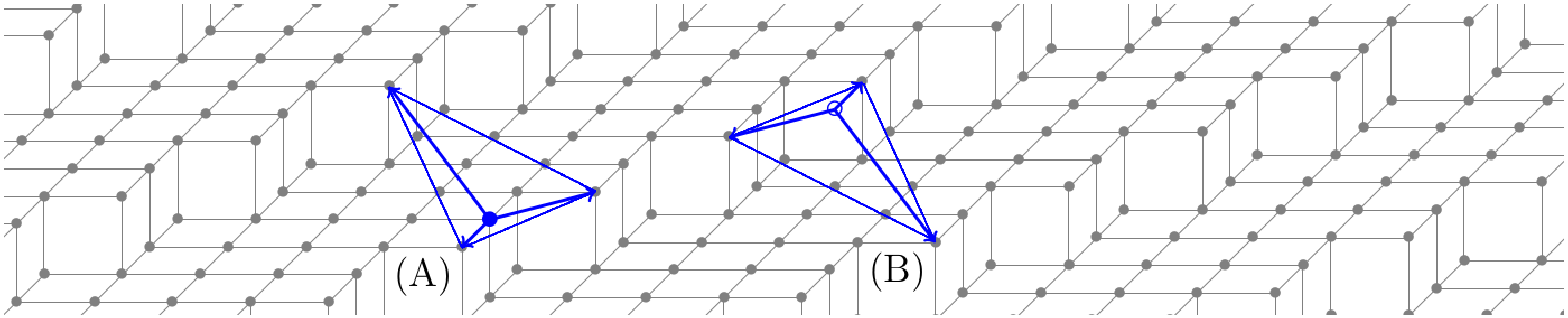


Algorithmes de plane-probing

Prédicat $\mathrm{InPlane}$: $\mathrm{InPlane}(p)$ vrai $\iff p \in \mathbf{P}$

Initialisation sur un surfel : tétraèdre défini par p et (e_0, e_1, e_2)

Algorithme : déformer le tétraèdre jusqu'à ce que $\sum_k (e_k^{(i)} \times e_{k+1}^{(i)}) = \mathbf{N}$

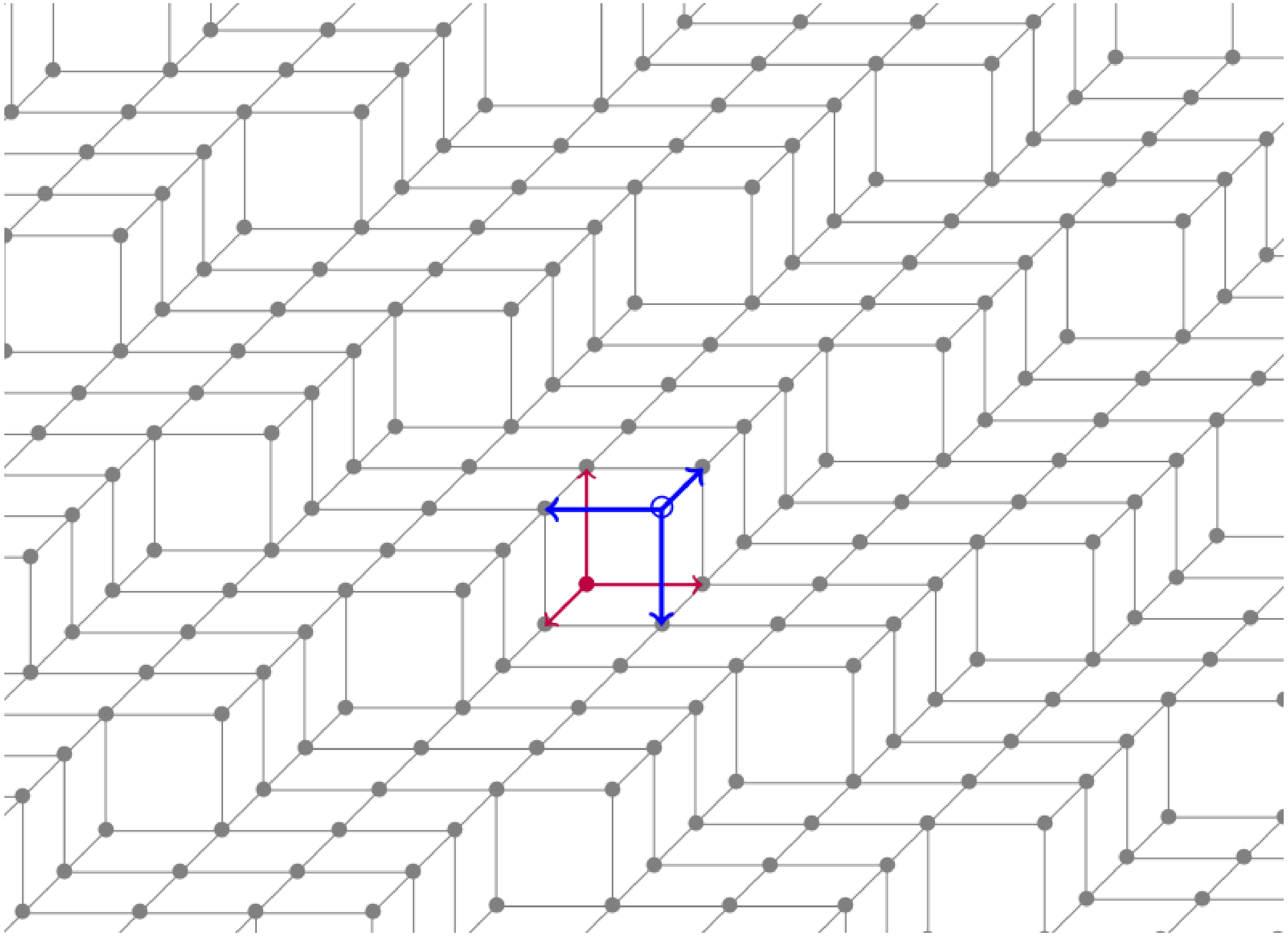


- Plusieurs types :
  - (A) Tétraèdre inférieur : pas de garantie de localité [Lachaud et al. '16]
  - (B) Tétraèdre supérieur : p ne bouge pas $\implies$ **local** [Lachaud et al. '17]
- Décider quels points ajouter à S grâce au prédicat $\mathrm{InPlane}$

$\implies$ plusieurs versions : H, R (base réduite à la fin et plus local)

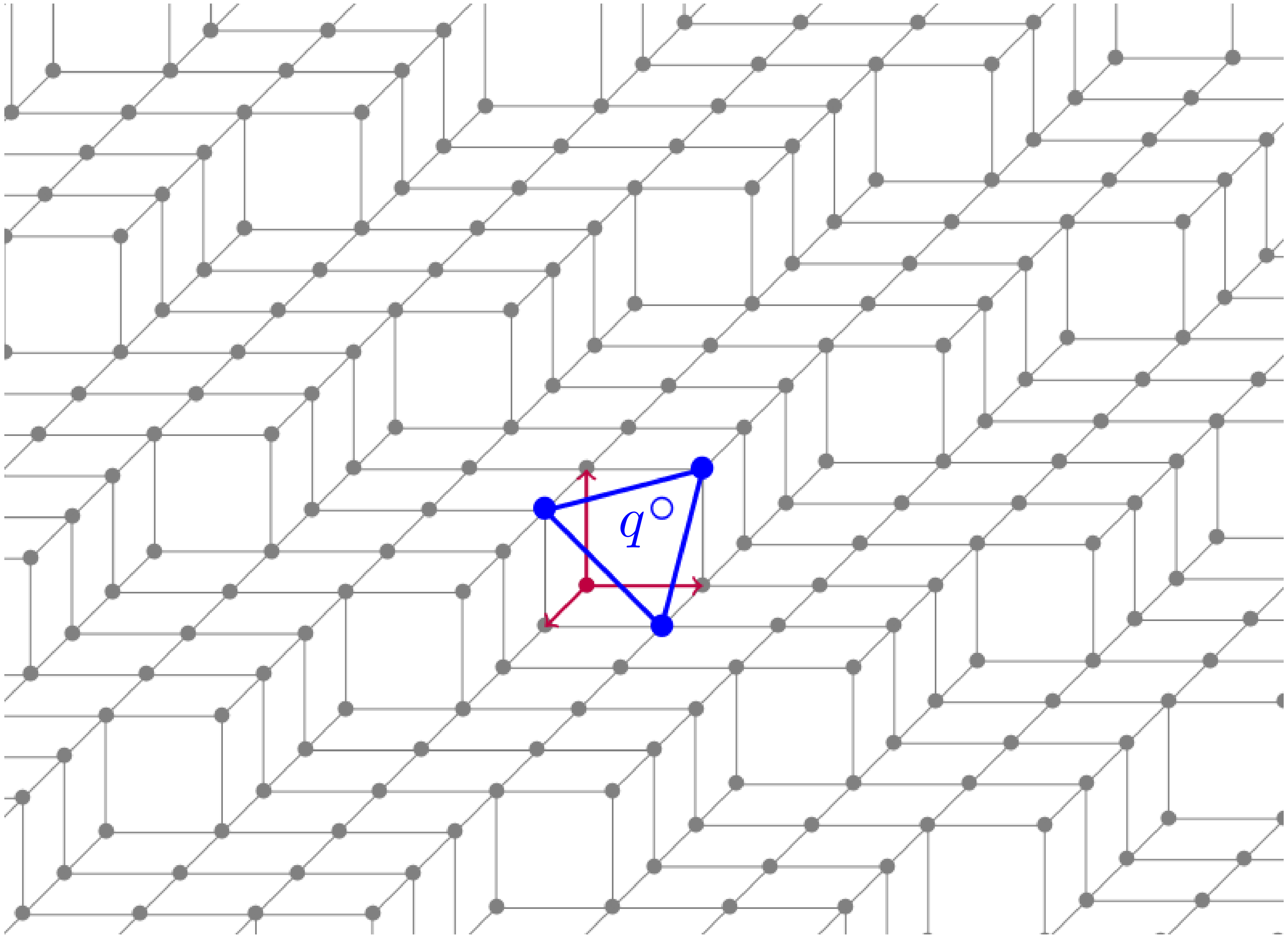
Exemple de plane-probing

Tétraèdre initial / Origine, $\mathbf{N} = (2, 6, 15)$



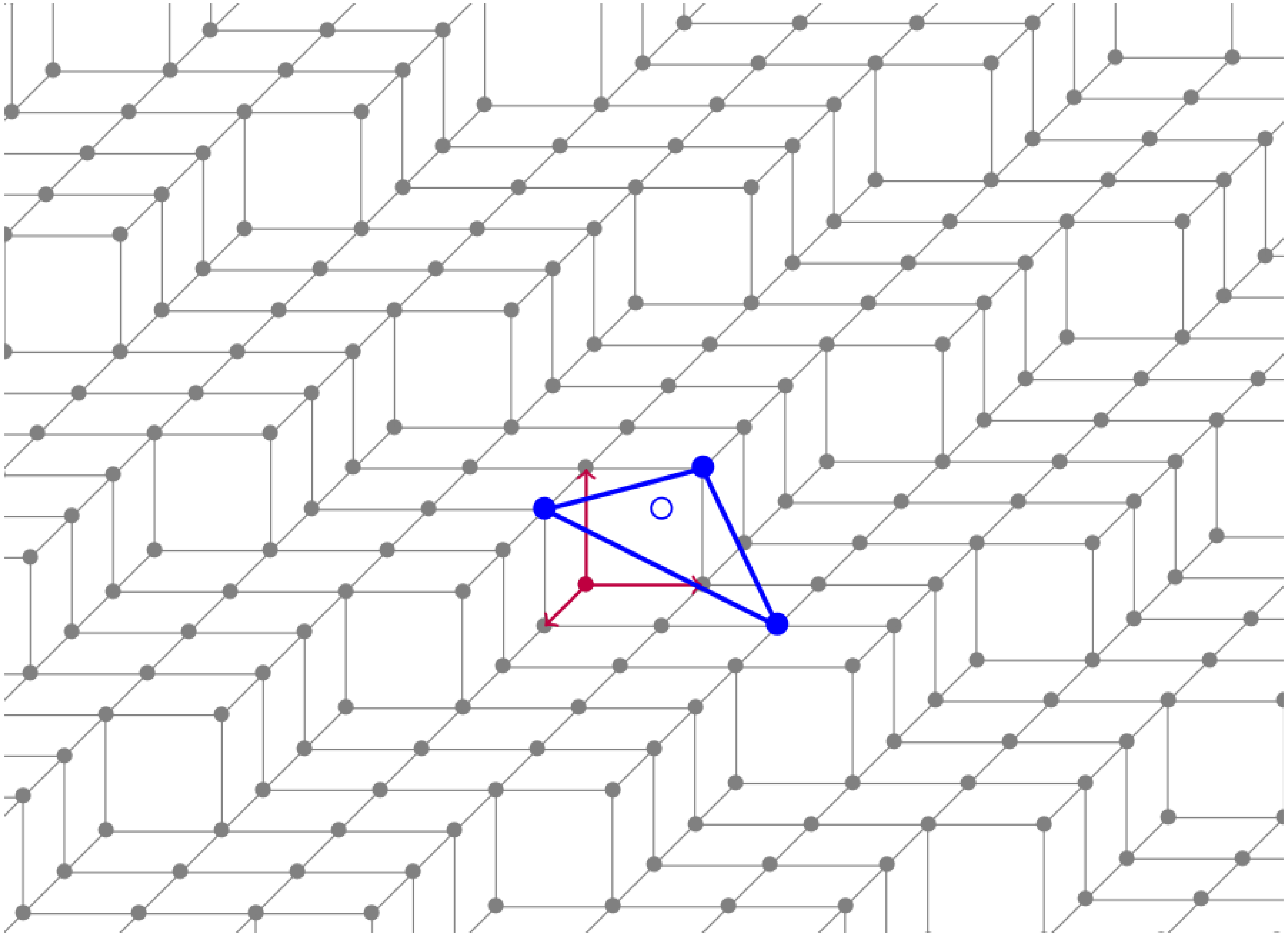
Exemple de plane-probing

Tétraèdre initial, $\mathbf{N}(T) = (1, 1, 1)$



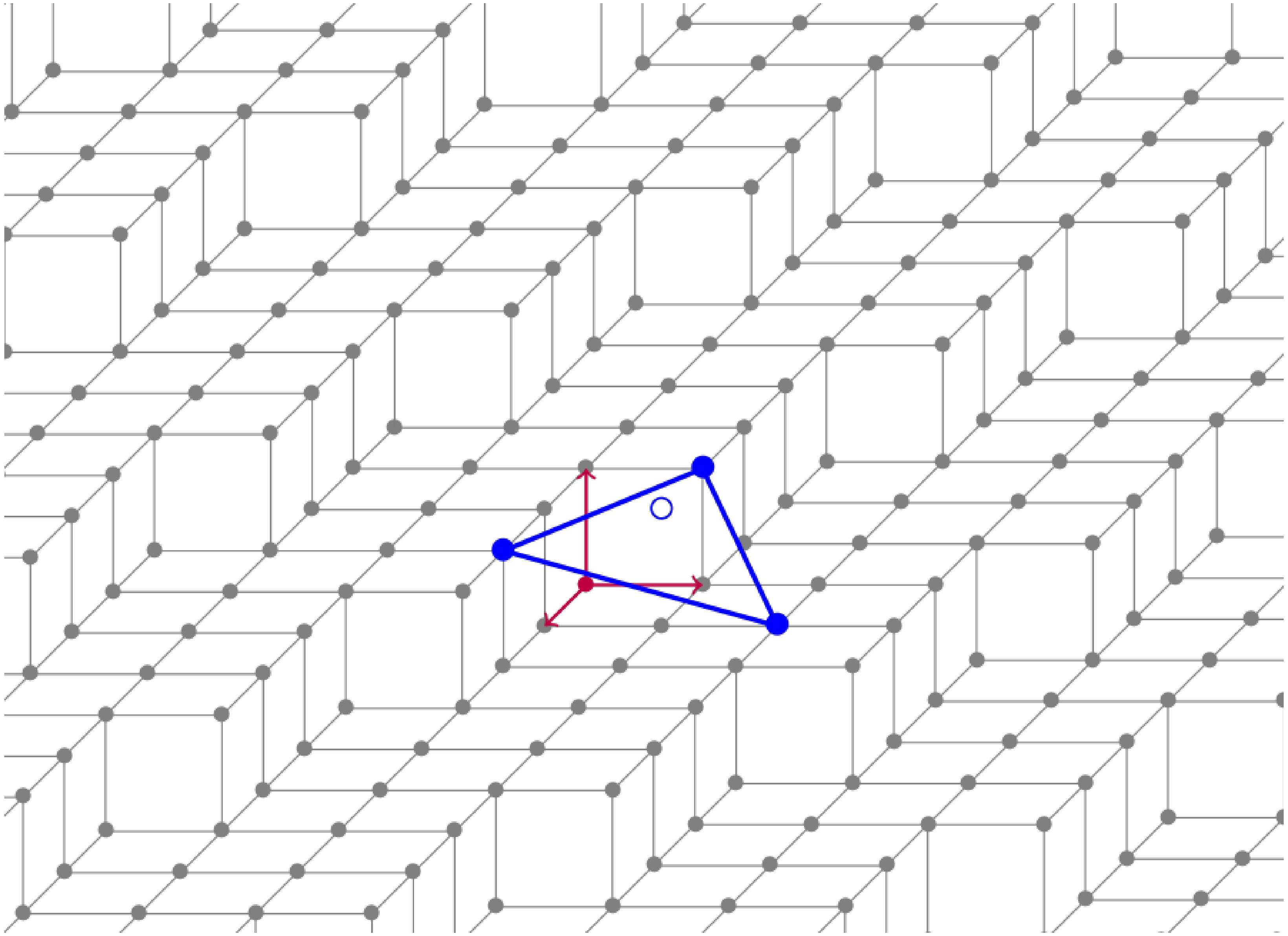
Exemple de plane-probing

$i = 1$, Tétraèdre



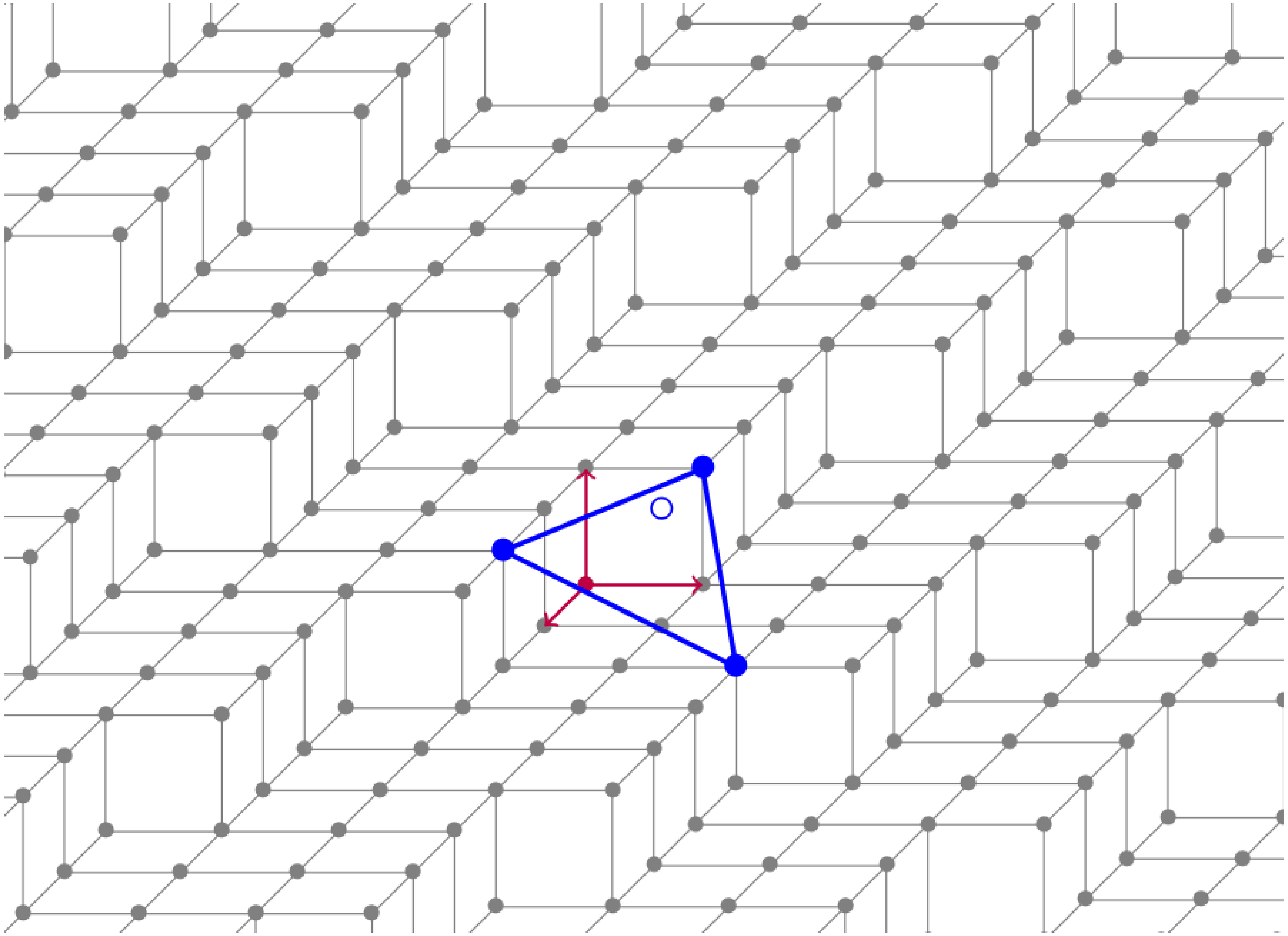
Exemple de plane-probing

$i = 2$, Tétraèdre



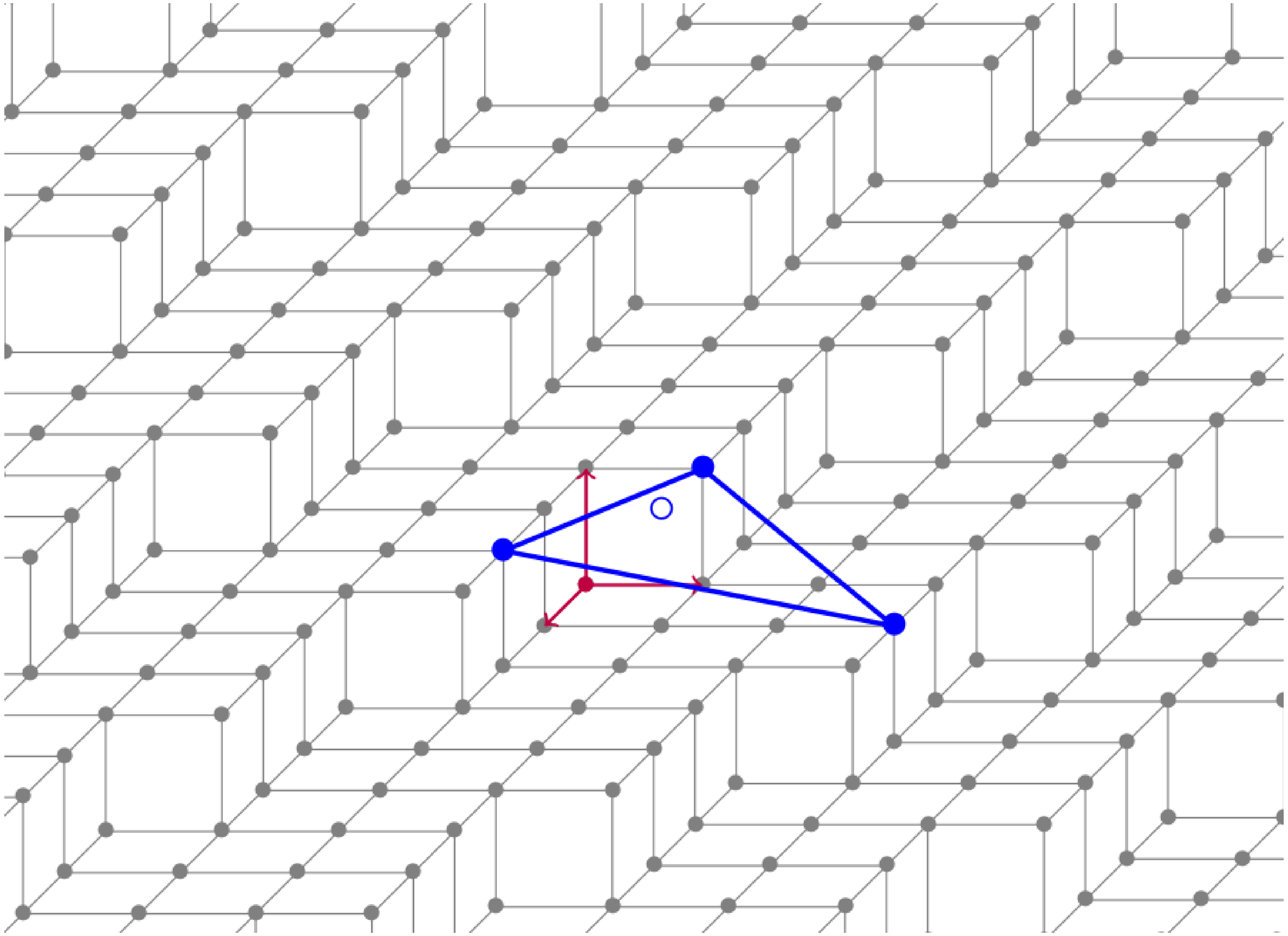
Exemple de plane-probing

$i = 3$, Tétraèdre



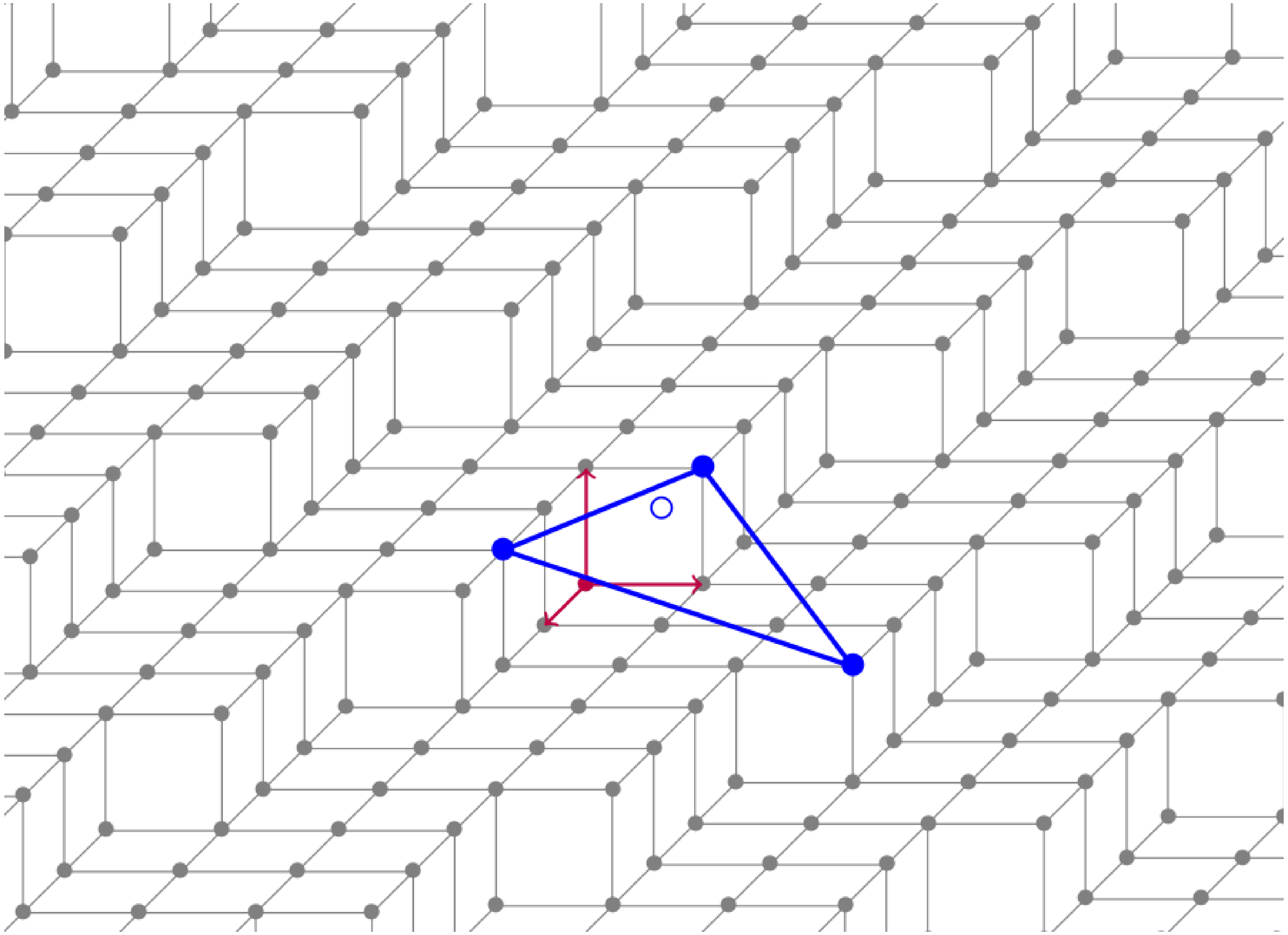
Exemple de plane-probing

$i = 4$, Tétraèdre



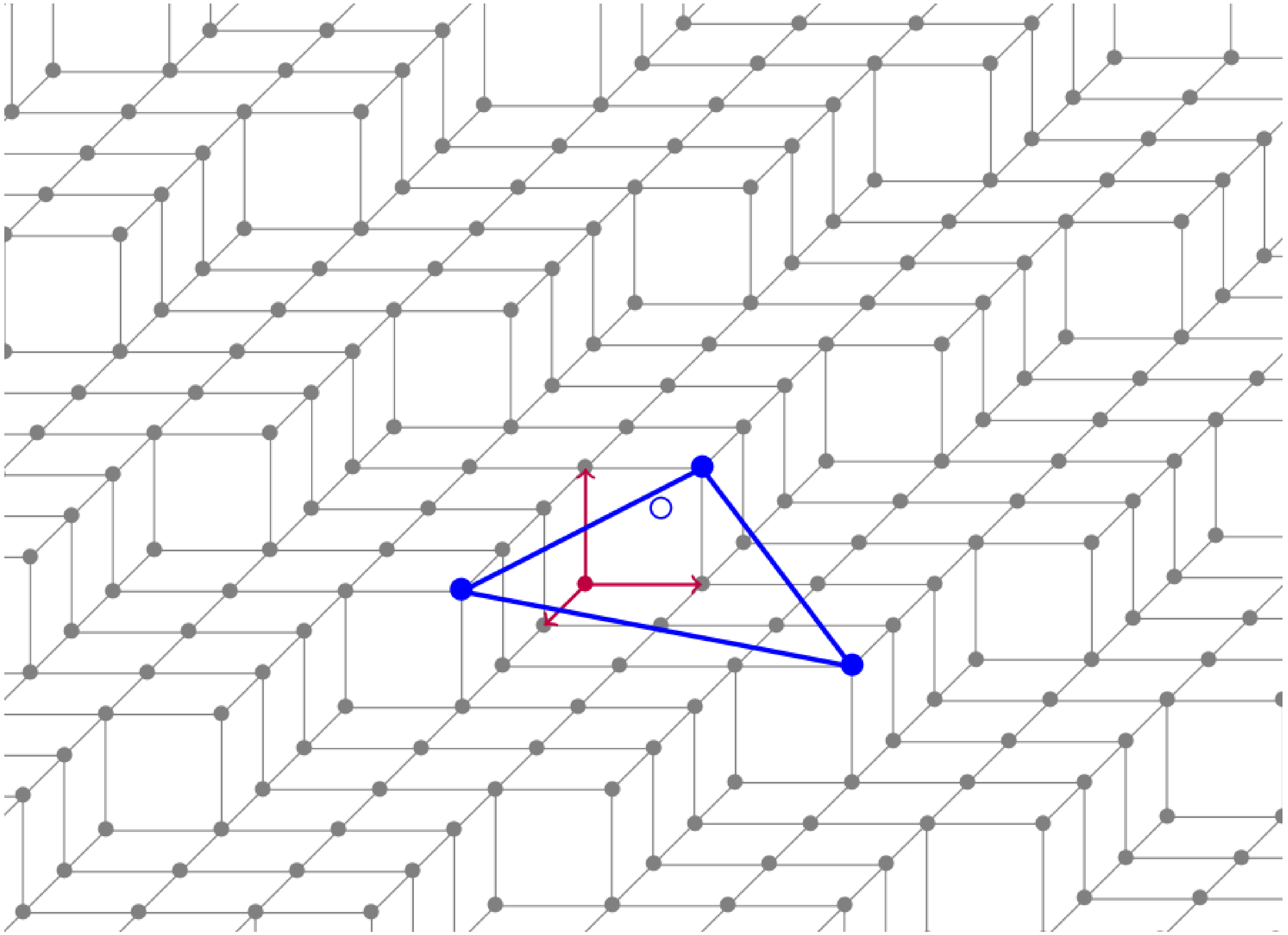
Exemple de plane-probing

$i = 5$, Tétraèdre



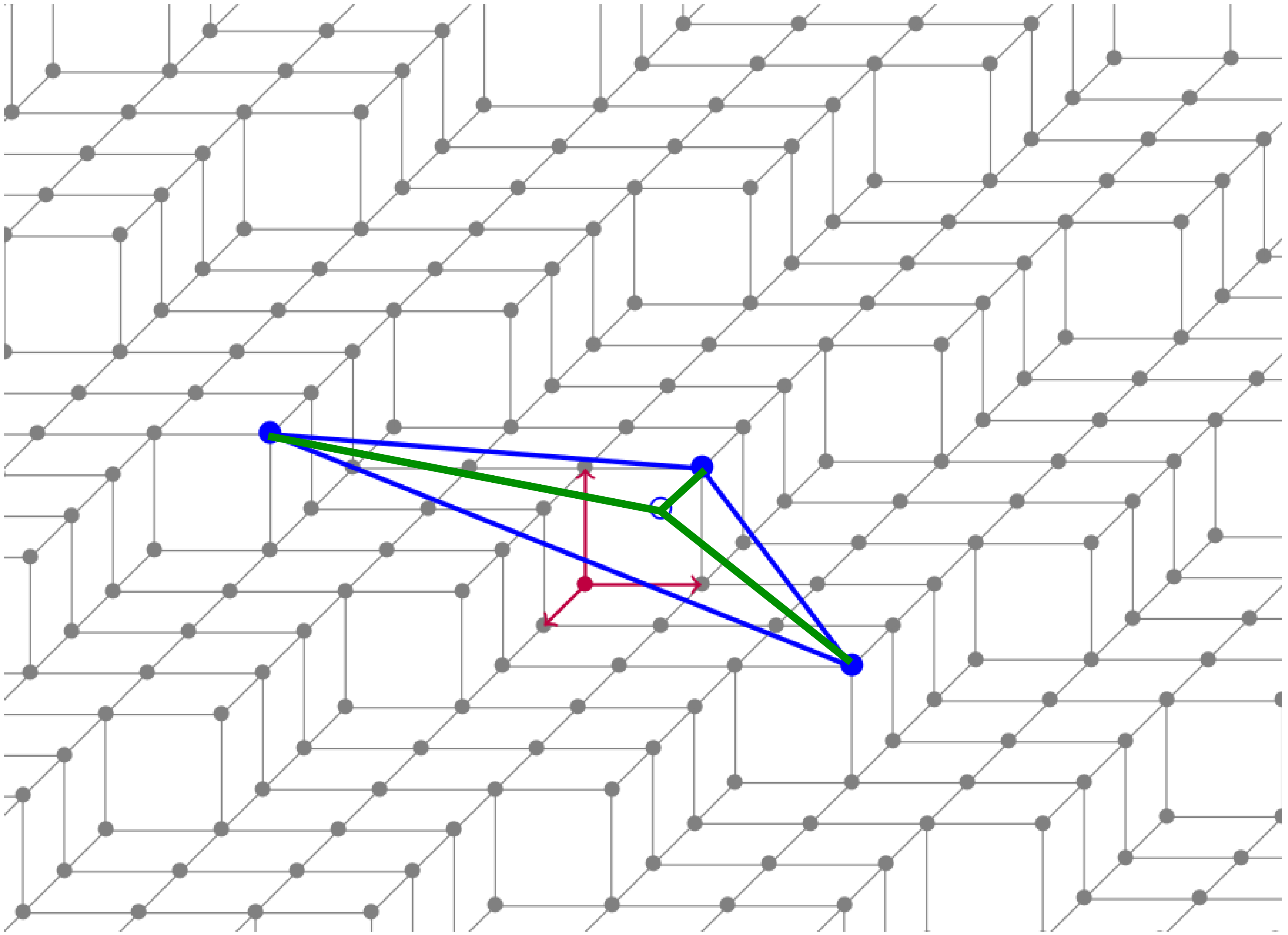
Exemple de plane-probing

$i = 6$, Tétraèdre



Exemple de plane-probing

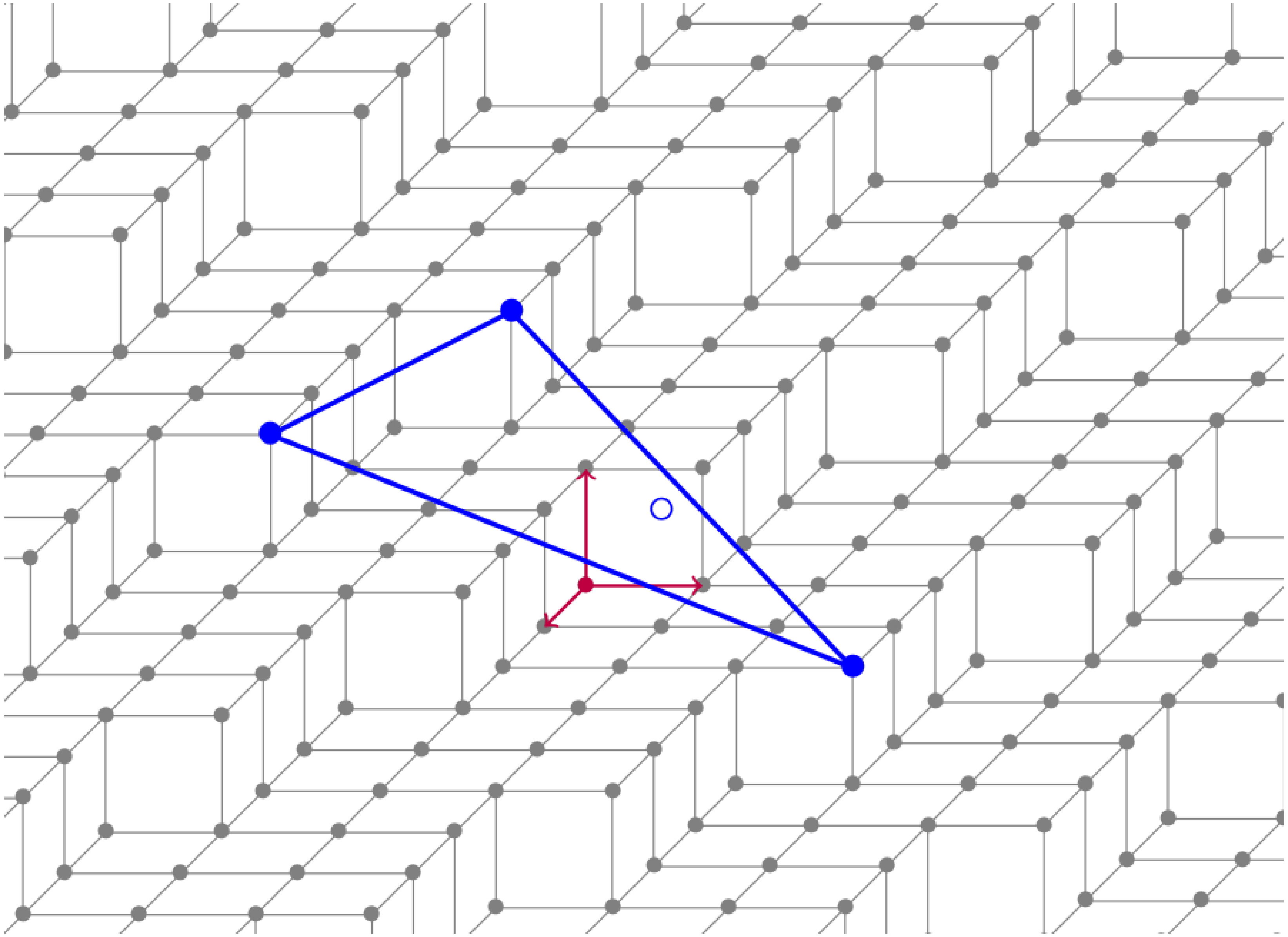
$i = 7$, Tétraèdre



$\overline{\vec{m_k}} > 0$

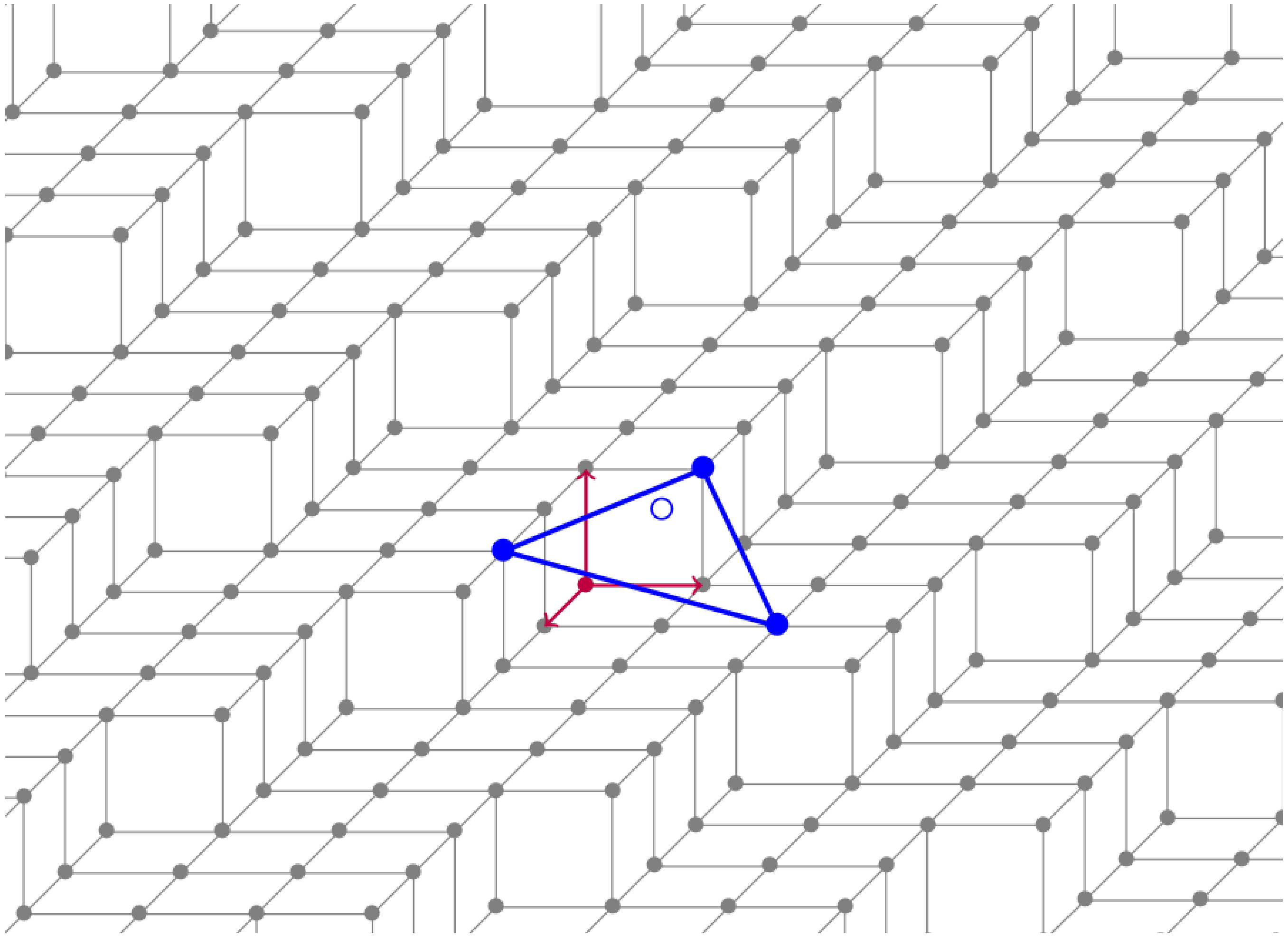
Exemple de plane-probing

$i = 8$, Tétraèdre, $\mathbf{N}(T) = (2, 6, 15)$



Procédure de mise à jour

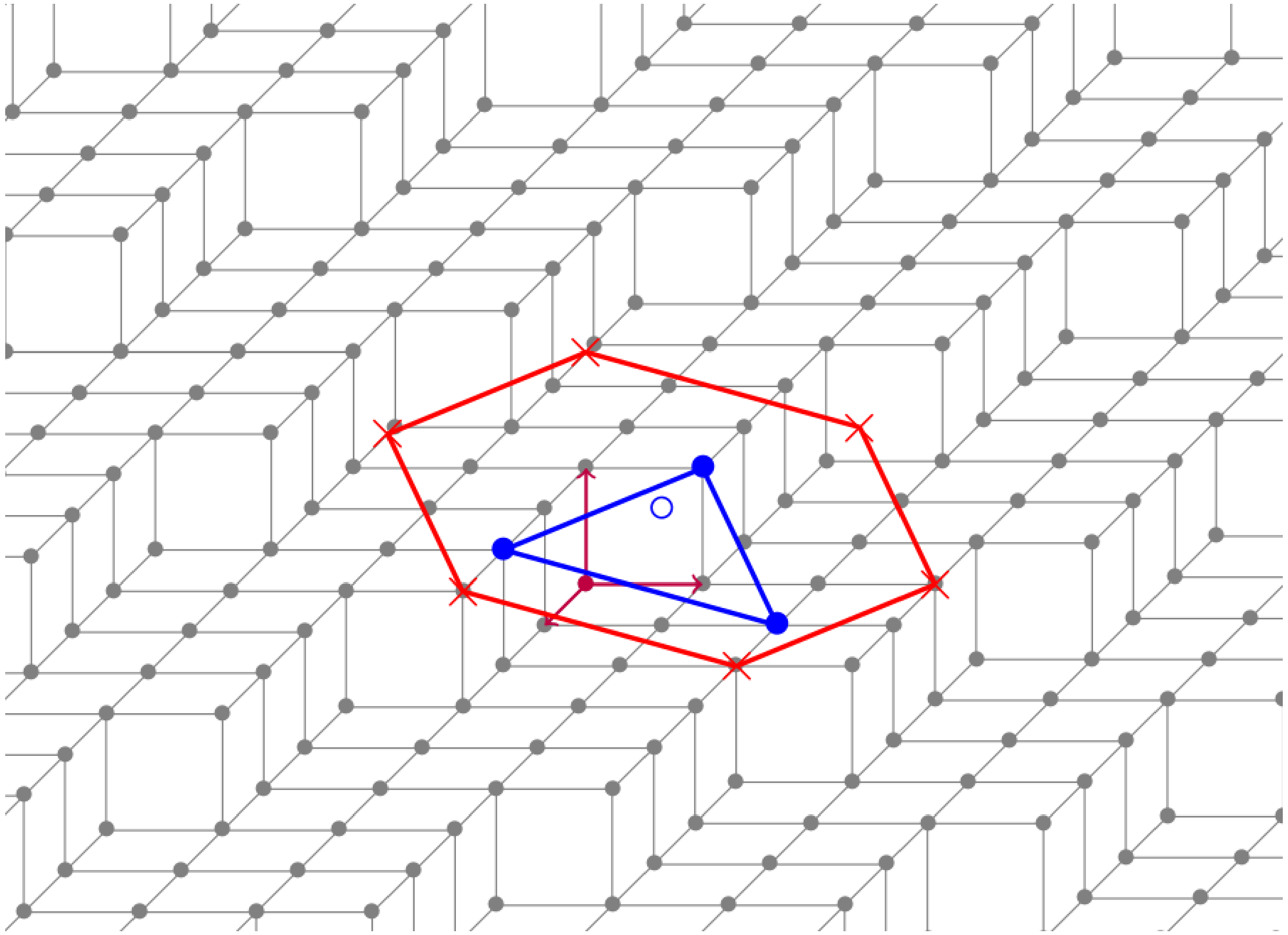
À chaque étape :



Procédure de mise à jour

À chaque étape :

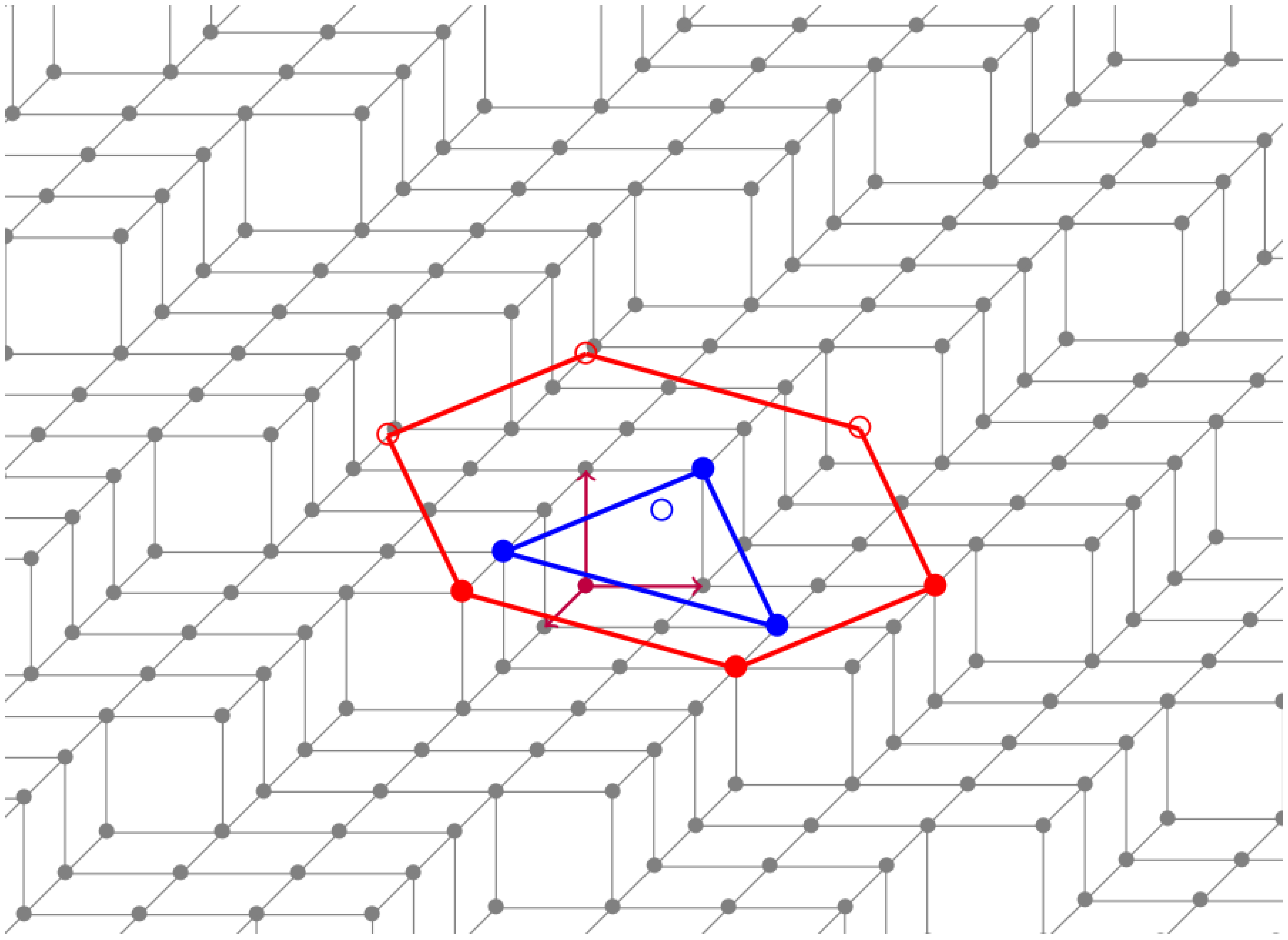
- considérer un ensemble candidat,



Procédure de mise à jour

À chaque étape :

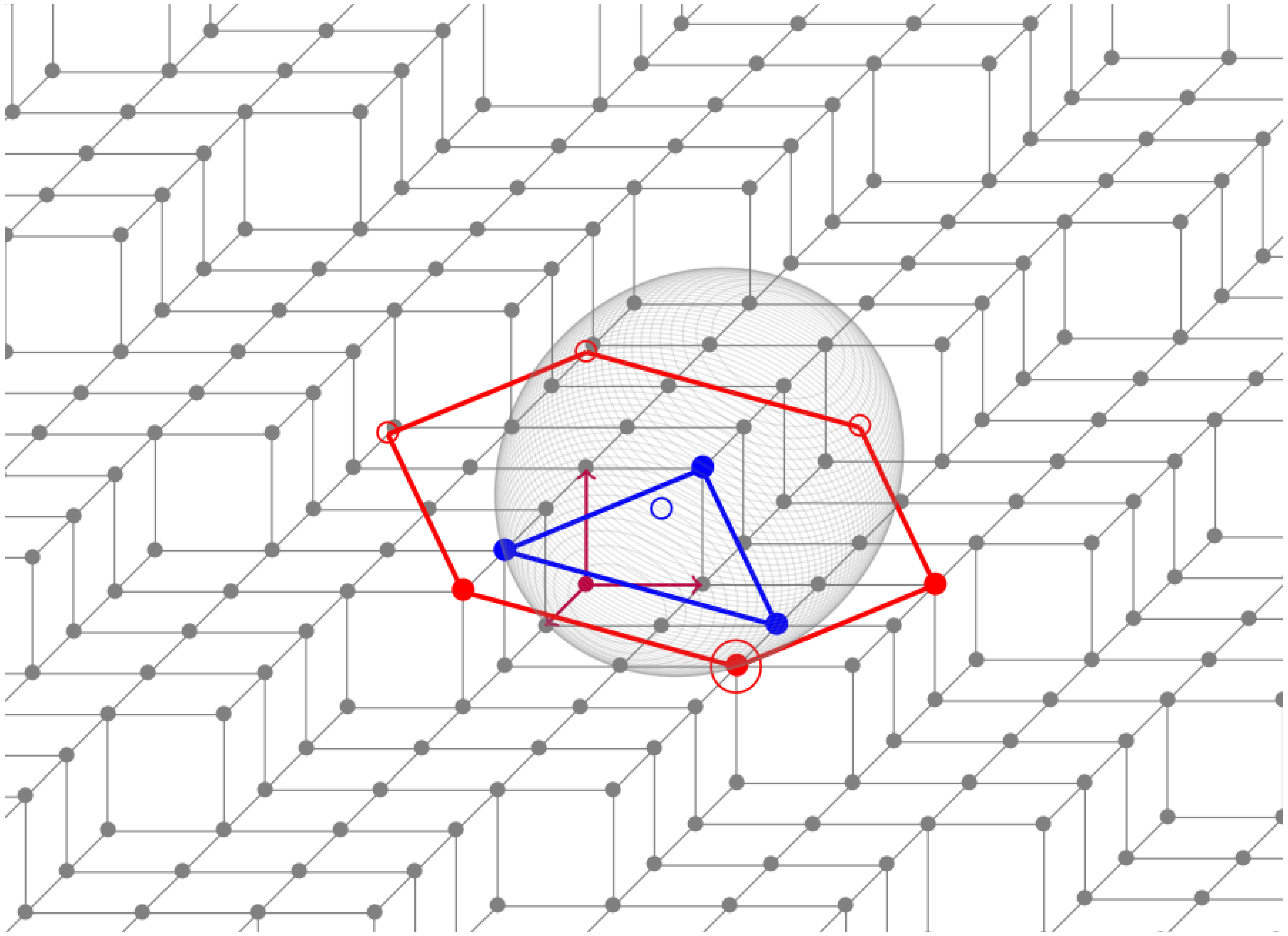
- considérer un ensemble candidat,
- filtrer les candidats par $\mathrm{InPlane}$,



Procédure de mise à jour

À chaque étape :

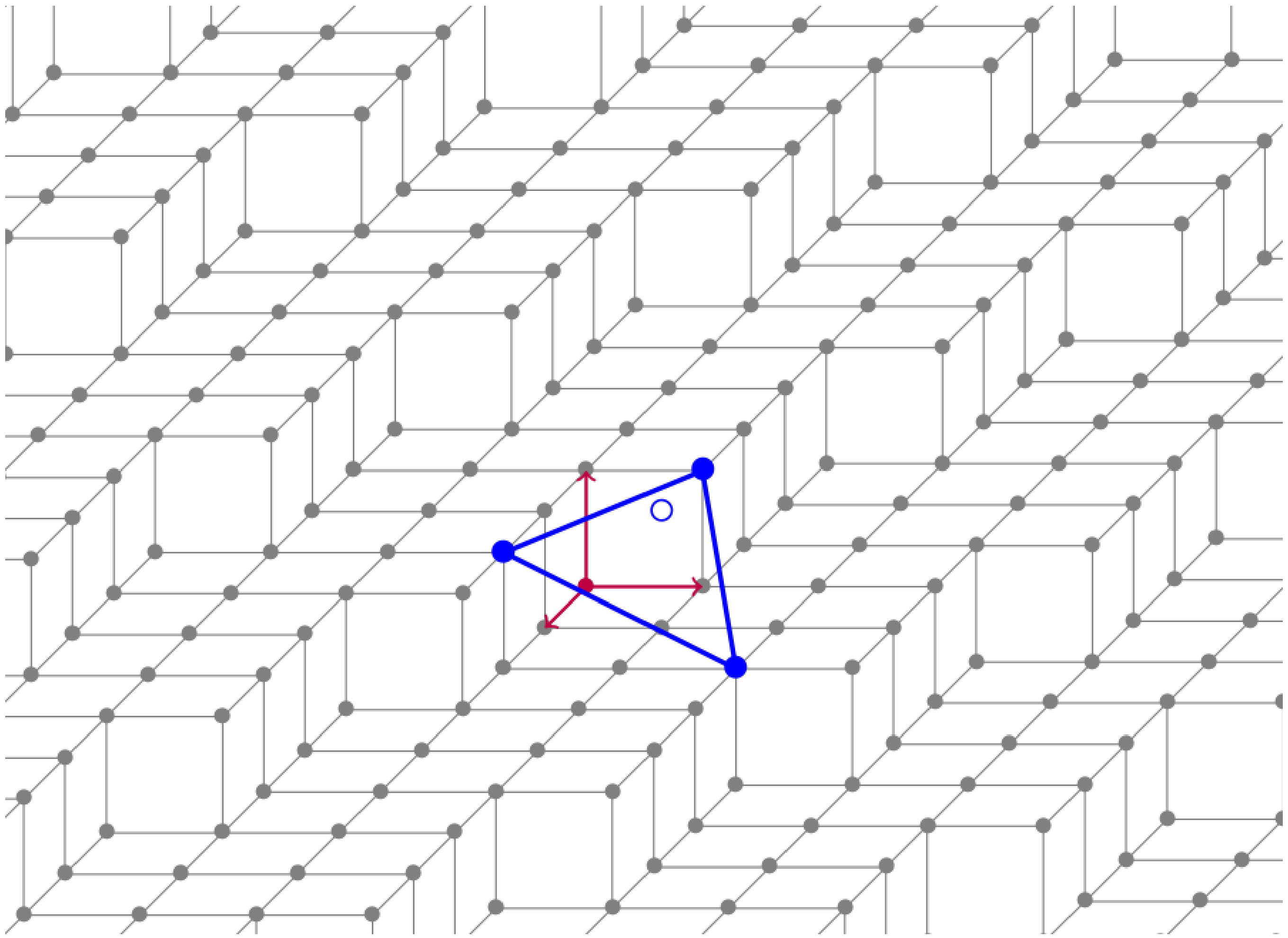
- considérer un ensemble candidat,
- filtrer les candidats par $\mathrm{InPlane}$,
- choisir le candidat le plus proche,



Procédure de mise à jour

À chaque étape :

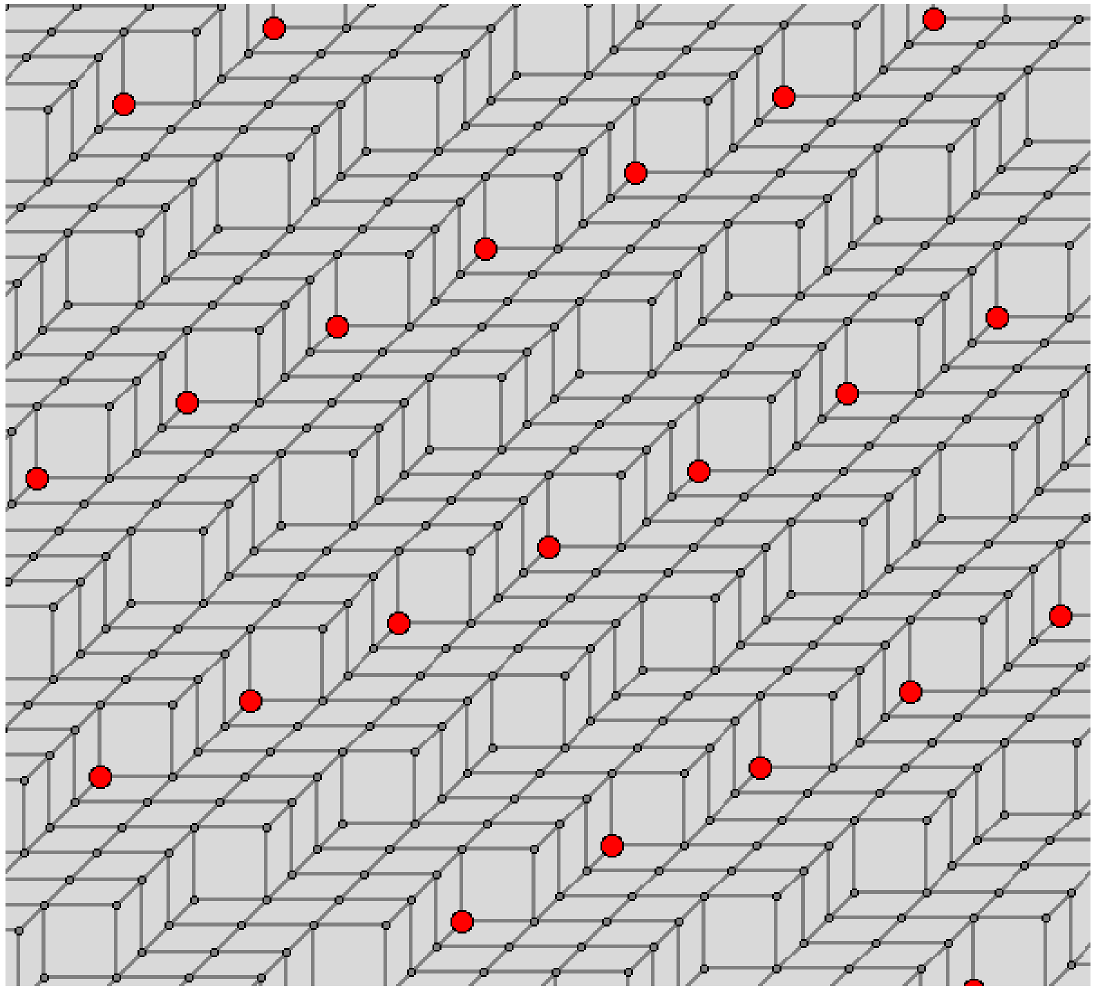
- considérer un ensemble candidat,
- filtrer les candidats par $\mathrm{InPlane}$,
- choisir le candidat le plus proche,
- mettre à jour le triangle.



Bilan des approches existantes

Limitations :

1. on ne retrouve la bonne normale que si l'on part d'un point d'appui inférieur

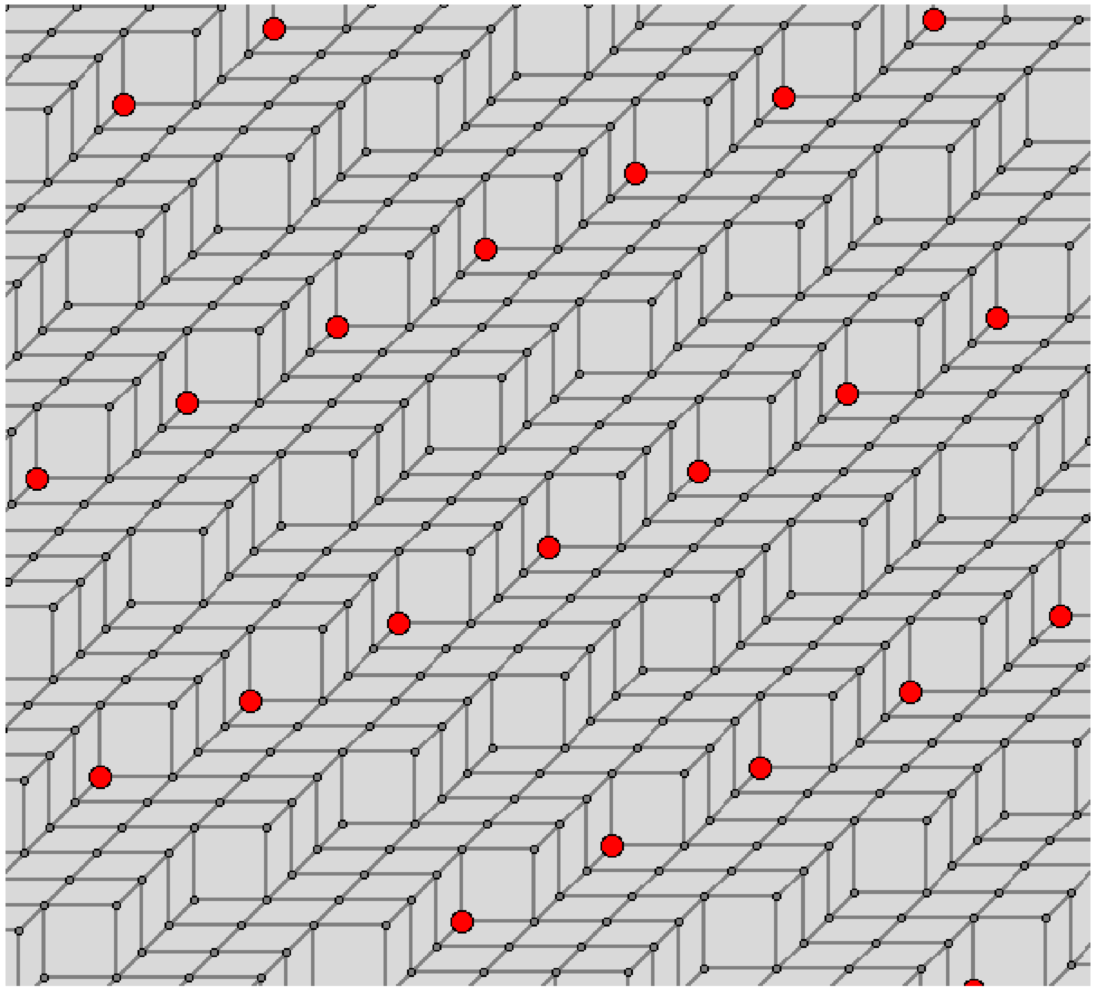


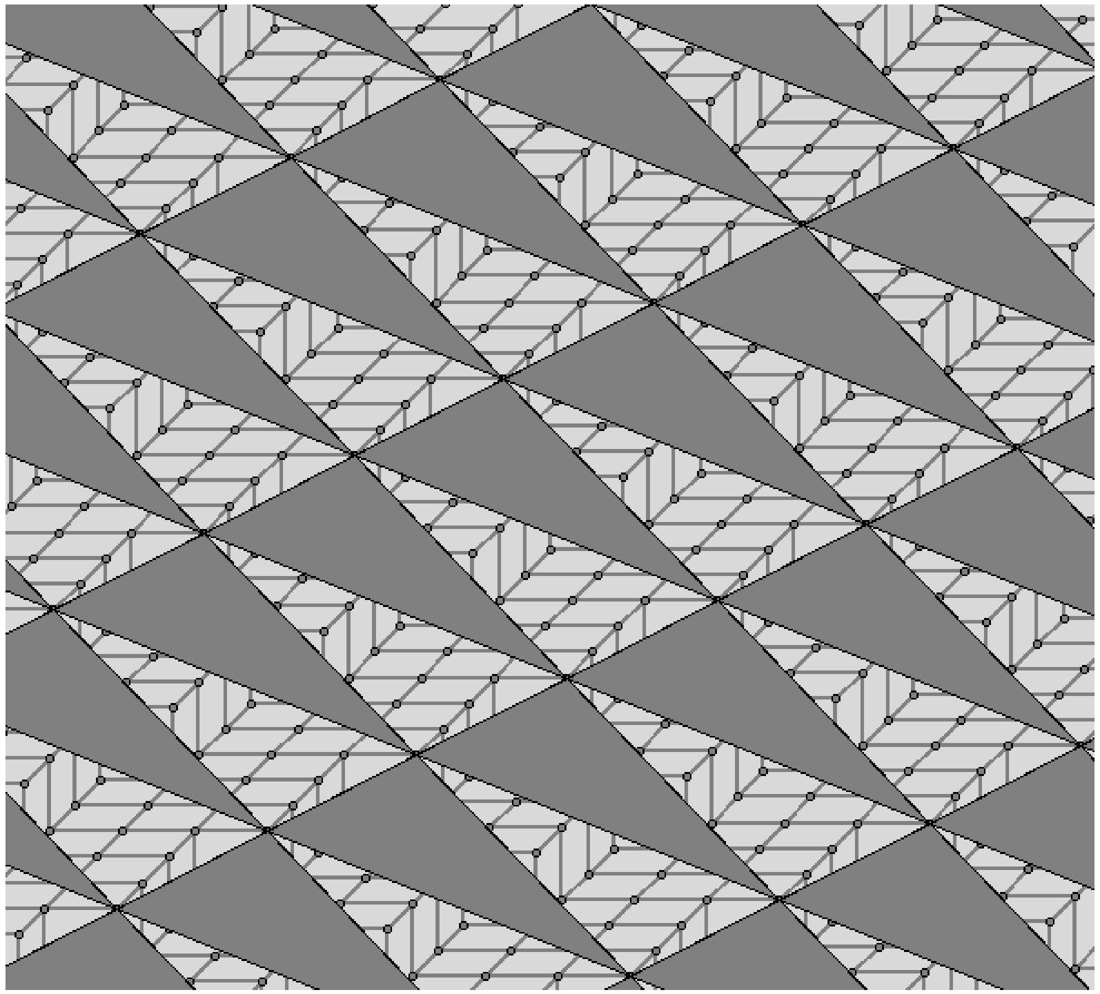
Bilan des approches existantes

Limitations :

1. on ne retrouve la bonne normale que si l'on part d'un point d'appui inférieur

2. on ne retrouve pas tous les triangles dans le réseau des points d'appui supérieur



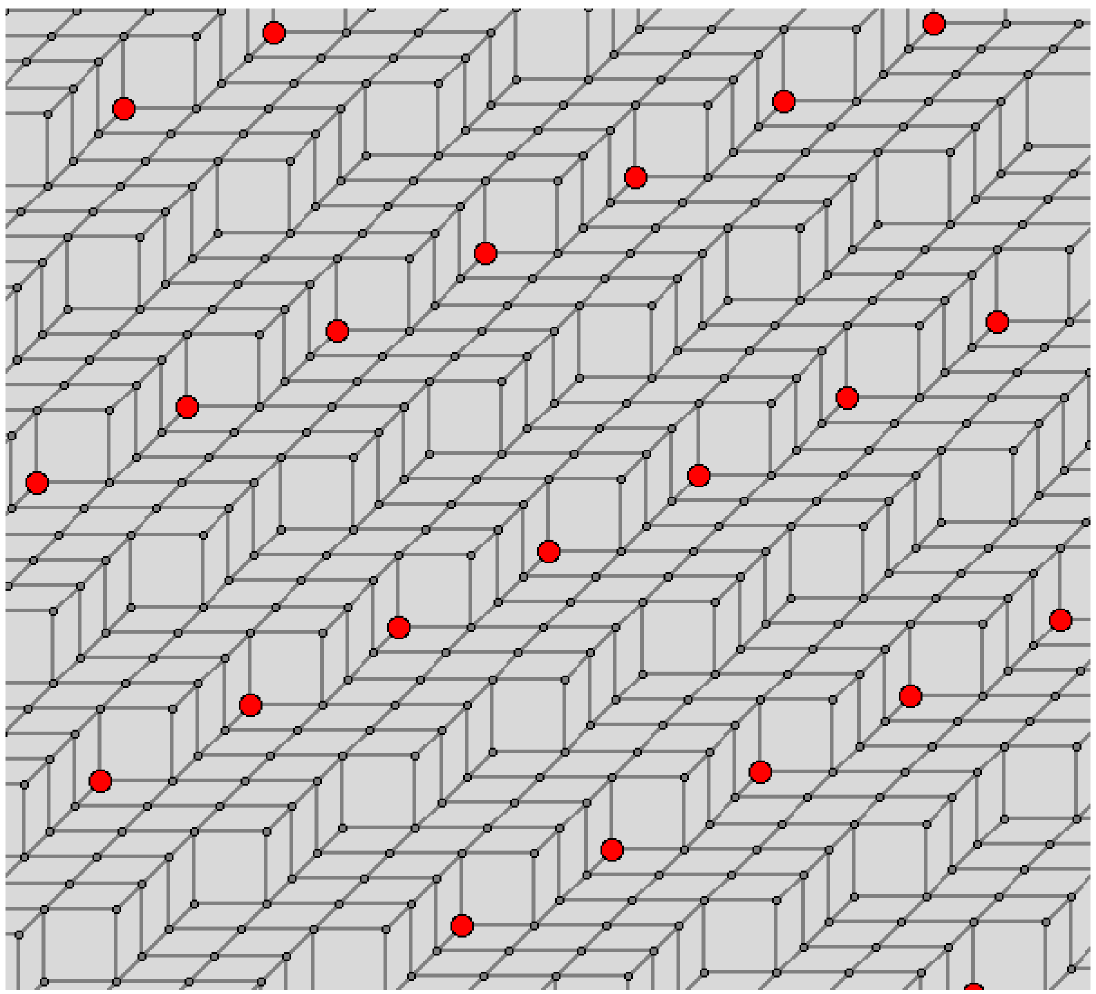


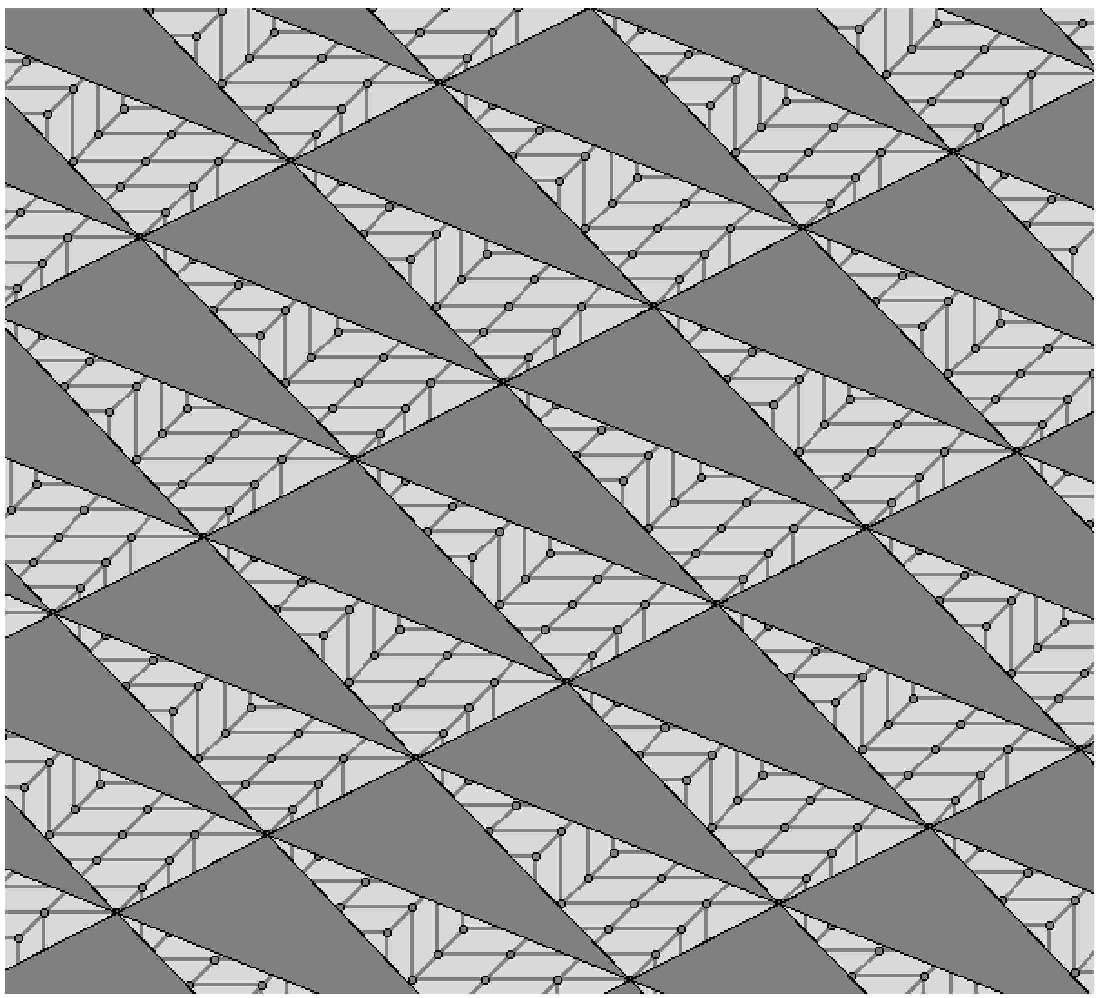
Bilan des approches existantes

Limitations :

1. on ne retrouve la bonne normale que si l'on part d'un point d'appui inférieur

2. on ne retrouve pas tous les triangles dans le réseau des points d'appui supérieur





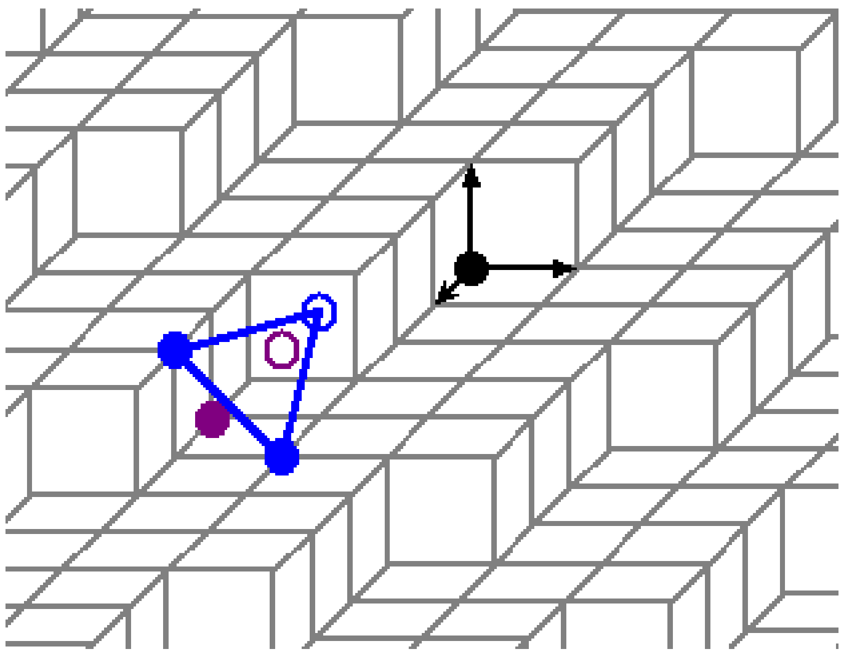
$\Longrightarrow$ Comment développer une approche générale ?

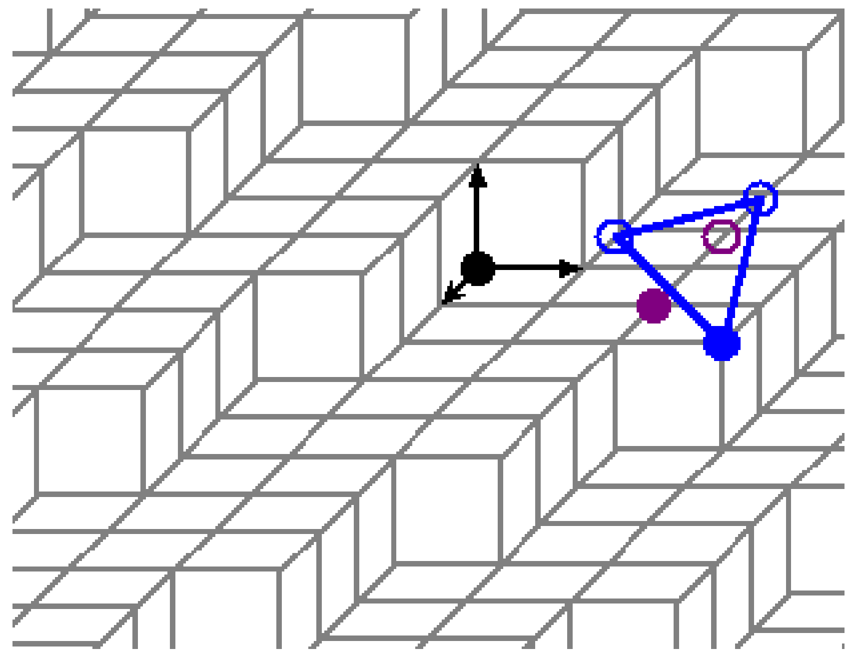
Généralisation des approches

1. Points du triangle pas forcément dans $\mathbf{P}$:

$\Longrightarrow$ généralisation du prédicat $\mathrm{InPlane} \rightarrow \mathrm{NotAbove}_q$

$\Longrightarrow$ trouver des points x pas plus hauts que q (i.e. $\bar{x} < \bar{q}$) pour la mise à jour



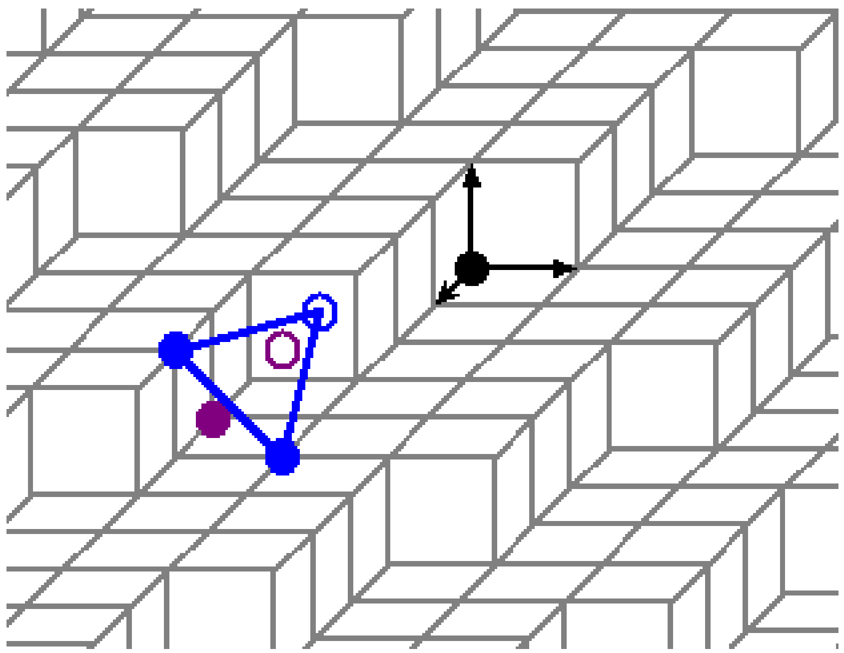


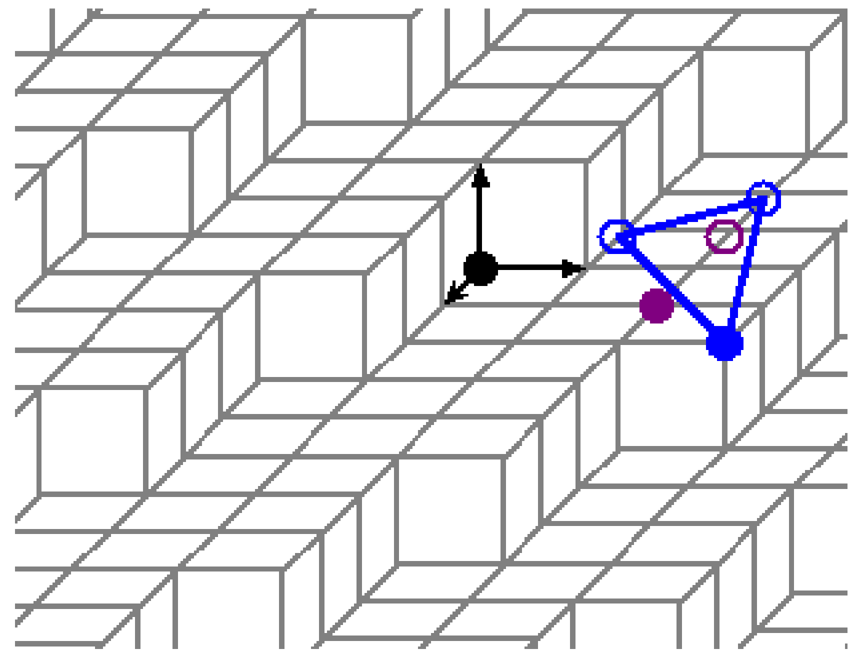
Généralisation des approches

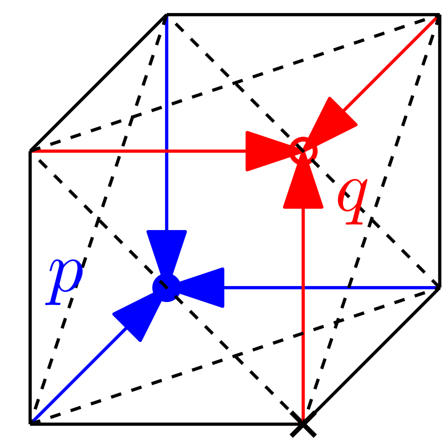
1. Points du triangle pas forcément dans $\mathbf{P}$:

$\Longrightarrow$ généralisation du prédicat $\mathrm{InPlane} \rightarrow \mathrm{NotAbove}_q$

$\Longrightarrow$ trouver des points x pas plus hauts que q (i.e. $\bar{x} < \bar{q}$) pour la mise à jour







2. Pouvoir retrouver tous les triangles :

$\Longrightarrow$ considérer les deux tétraèdres (inférieur et supérieur) $\rightarrow$ cube

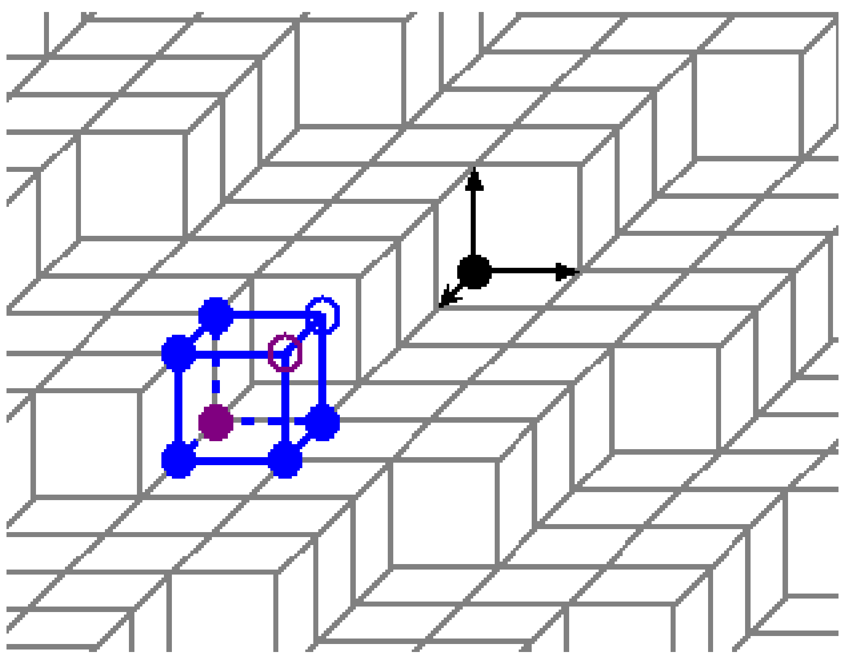
$\Longrightarrow$ passage d'un tétraèdre à l'autre si besoin

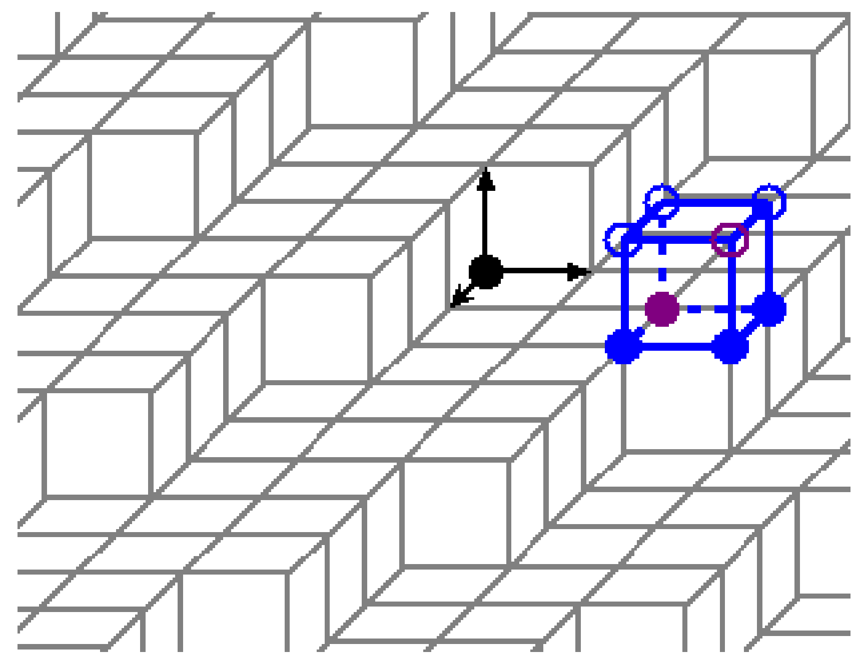
Généralisation des approches

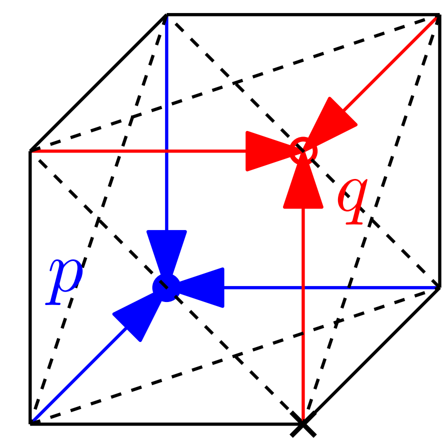
1. Points du triangle pas forcément dans $\mathbf{P}$:

$\Longrightarrow$ généralisation du prédicat $\mathrm{InPlane} \to \mathrm{NotAbove}_q$

$\Longrightarrow$ trouver des points x pas plus hauts que q (i.e. $\bar{x} < \bar{q}$) pour la mise à jour







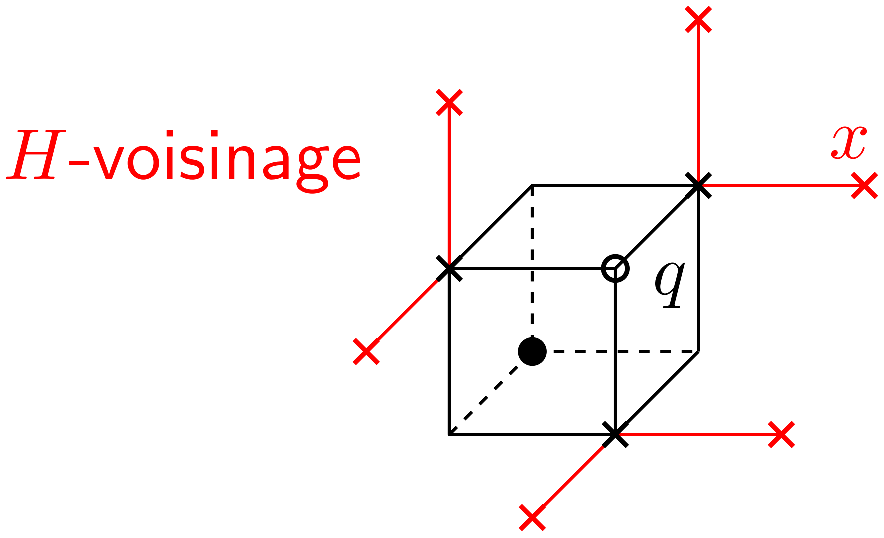
2. Pouvoir retrouver tous les triangles :

$\Longrightarrow$ considérer les deux tétraèdres (inférieur et supérieur) $\to$ cube

$\Longrightarrow$ passage d'un tétraèdre à l'autre si besoin

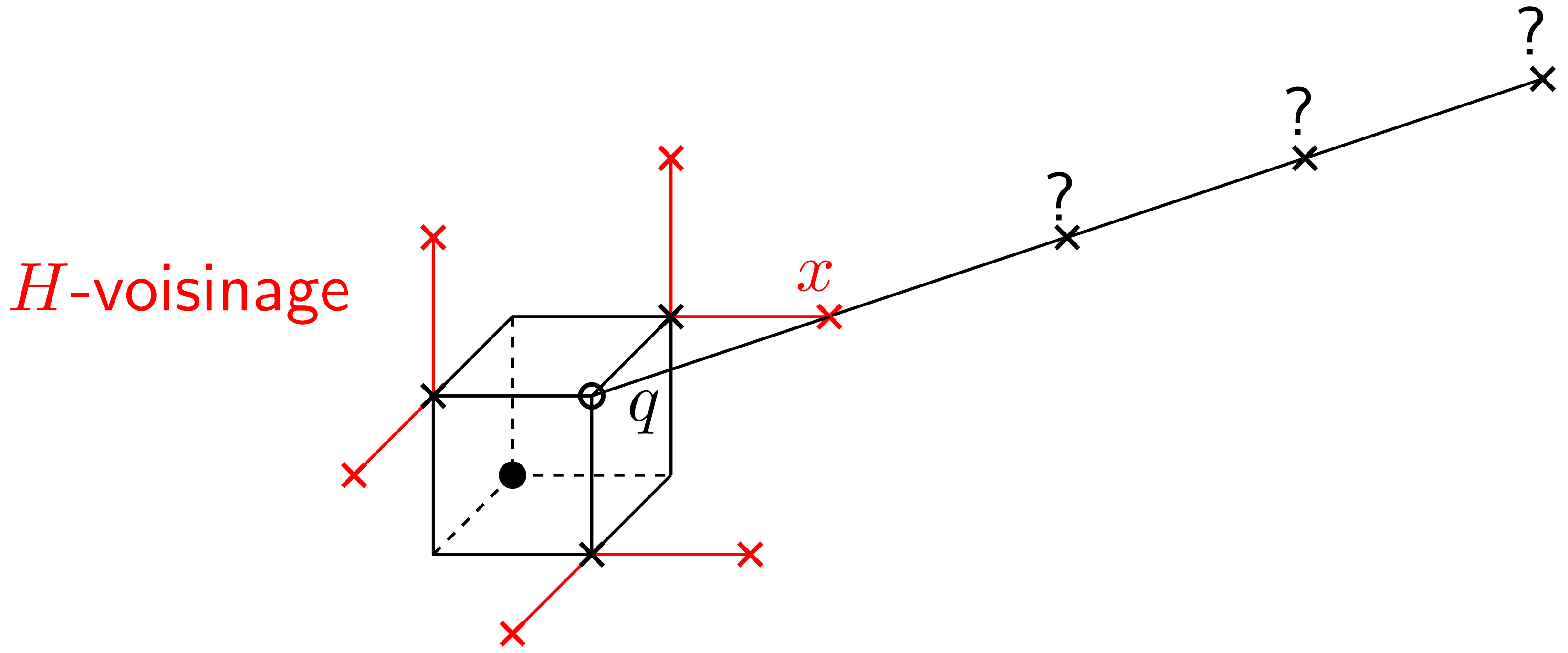
Prédicat $\mathrm{NotAbove}_q$

- Nouveau prédicat : $\mathrm{NotAbove}_q(x)$ vrai $\iff \bar{x} < \bar{q}$



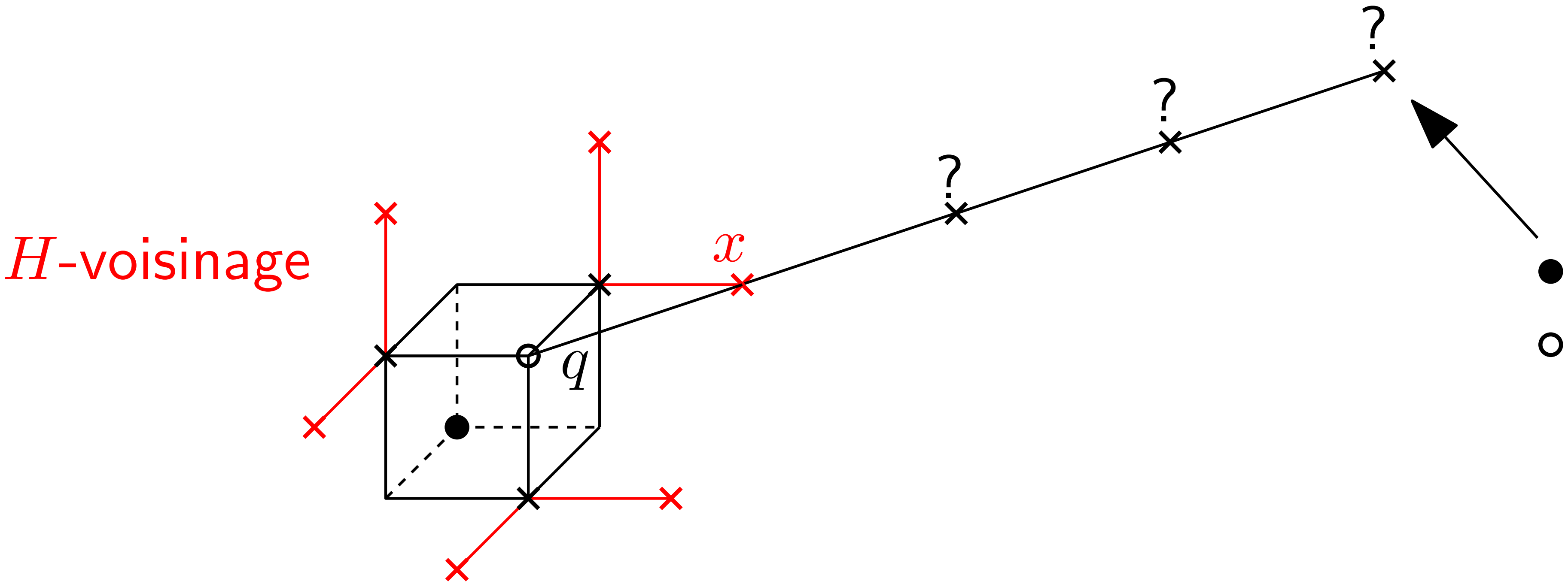
Prédicat $\mathrm{NotAbove}_q$

- Nouveau prédicat : $\mathrm{NotAbove}_q(x)$ vrai $\iff \bar{x} < \bar{q}$



Prédicat $\mathrm{NotAbove}_q$

- Nouveau prédicat : $\mathrm{NotAbove}_q(x)$ vrai $\iff \bar{x} < \bar{q}$



● $\implies \bar{x} < \bar{q}$
○ $\implies \bar{x} \geq \bar{q}$

Prédicat $\mathrm{NotAbove}_q$

- Nouveau prédicat : $\mathrm{NotAbove}_q(x)$ vrai $\iff \bar{x} < \bar{q}$
- Algorithme :

Entrée : point $x \in \mathbb{Z}^3$, prédicat $\mathrm{InPlane}$, cube courant, borne K

$(\vec{u}, s) \leftarrow (x - q, q)$ `// rayon partant de` s `de direction` $\vec{u}$

Si $\mathrm{InPlane}(q)$ **alors** $s \leftarrow s - \sum_{k=0}^{2} \vec{m_k}$ `//` s `en dehors de` $\mathbf{P}$

$l \leftarrow 1$

Tant que $l < K$ **faire**

  Si $\mathrm{InPlane}(s + l\vec{u})$ **alors retourner** $\neg\,\mathrm{InPlane}(q)$

  Si $\mathrm{InPlane}(s - l\vec{u})$ **alors retourner** $\mathrm{InPlane}(q)$

  $l \leftarrow 2l$

retourner `false`

$\Longrightarrow$ $O(\log \|\mathbf{N}\|_1)$ appels à $\mathrm{InPlane}$.

Changement de tétraèdre

- Rester proche de la surface $\Longrightarrow$ cube **séparant** (≥ 1 dedans, ≥ 1 dehors)

  $\Longrightarrow$ règle de majorité pour choisir le tétraèdre,

  $\Longrightarrow$ dès qu'il y a strictement moins de 4 points dans $\mathbf{P}$, on inverse.

Inversion : passage d'un tétraèdre à l'autre

Changement de tétraèdre

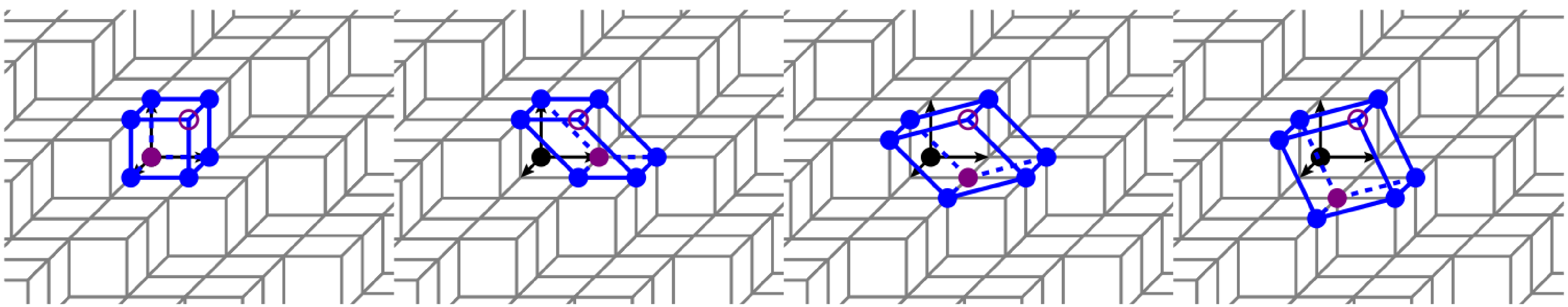
- Rester proche de la surface $\Longrightarrow$ cube **séparant** (≥ 1 dedans, ≥ 1 dehors)

  $\Longrightarrow$ règle de majorité pour choisir le tétraèdre,

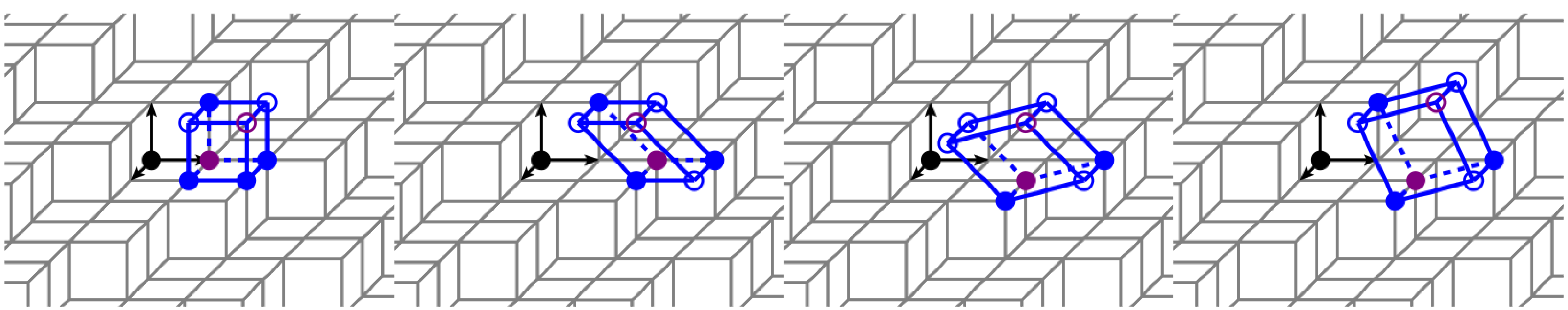
  $\Longrightarrow$ dès qu'il y a strictement moins de 4 points dans $\mathbf{P}$, on inverse.

<u>Inversion</u> : passage d'un tétraèdre à l'autre

<u>Exemple</u> : pas d'inversion v.s. 1 inversion, $\mathbf{N} = (1, 2, 4)$

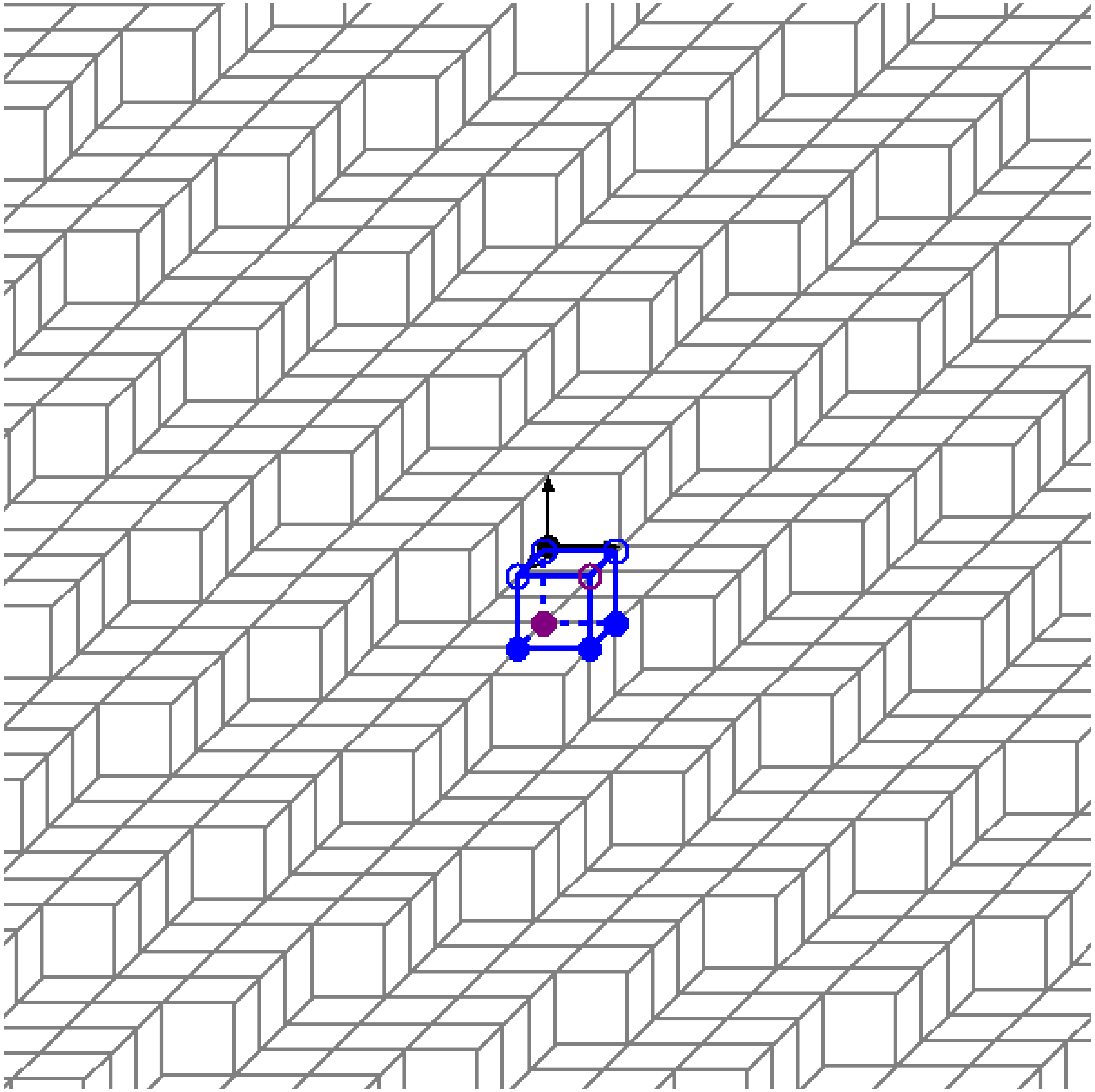


(a) $\sharp 1$ (b) $\sharp 2$ (c) $\sharp 3$ (d) $\sharp 4$

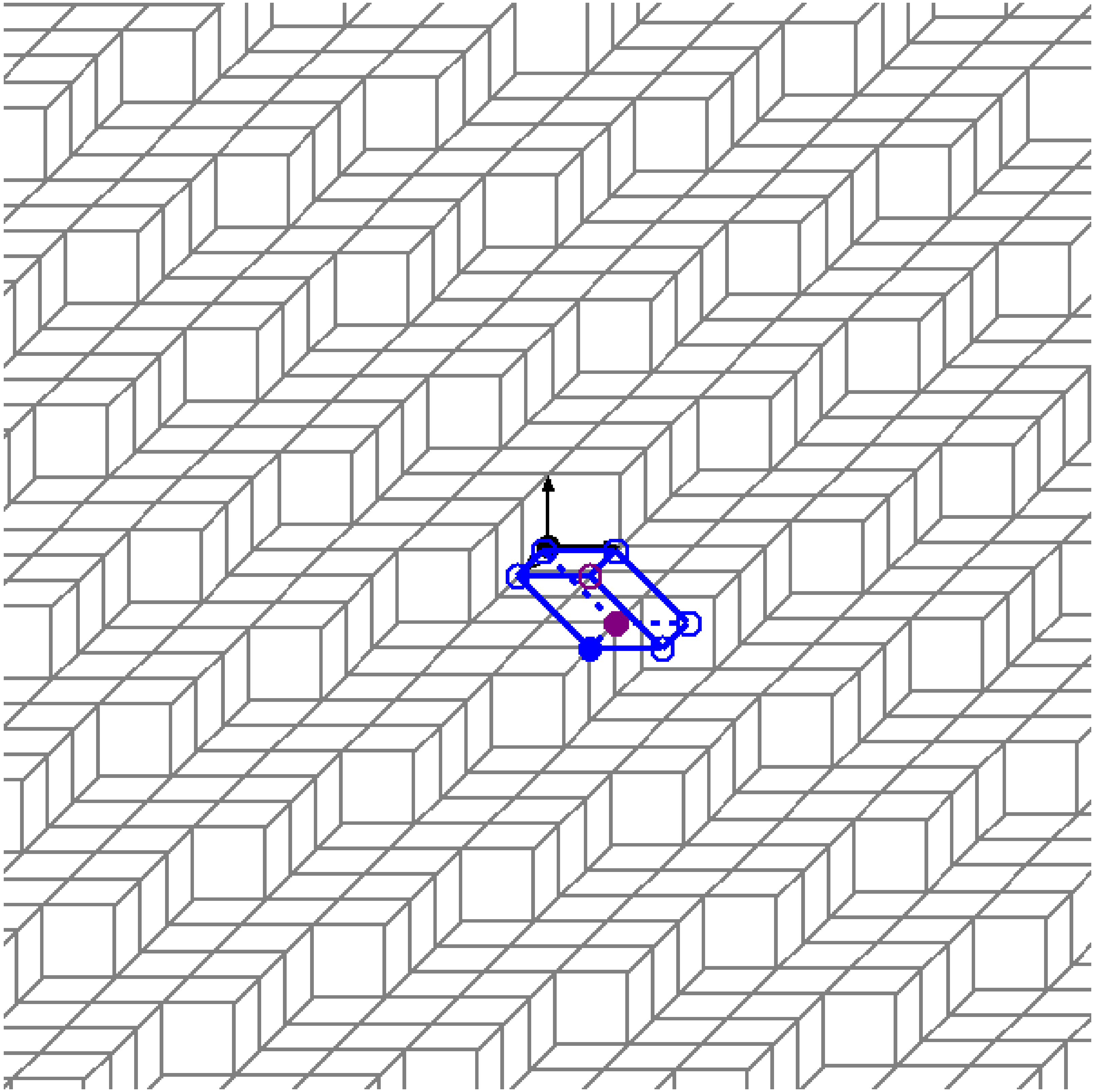


(e) $\sharp 1$ (f) $\sharp 2$ (g) $\sharp 3$ (h) $\sharp 4$

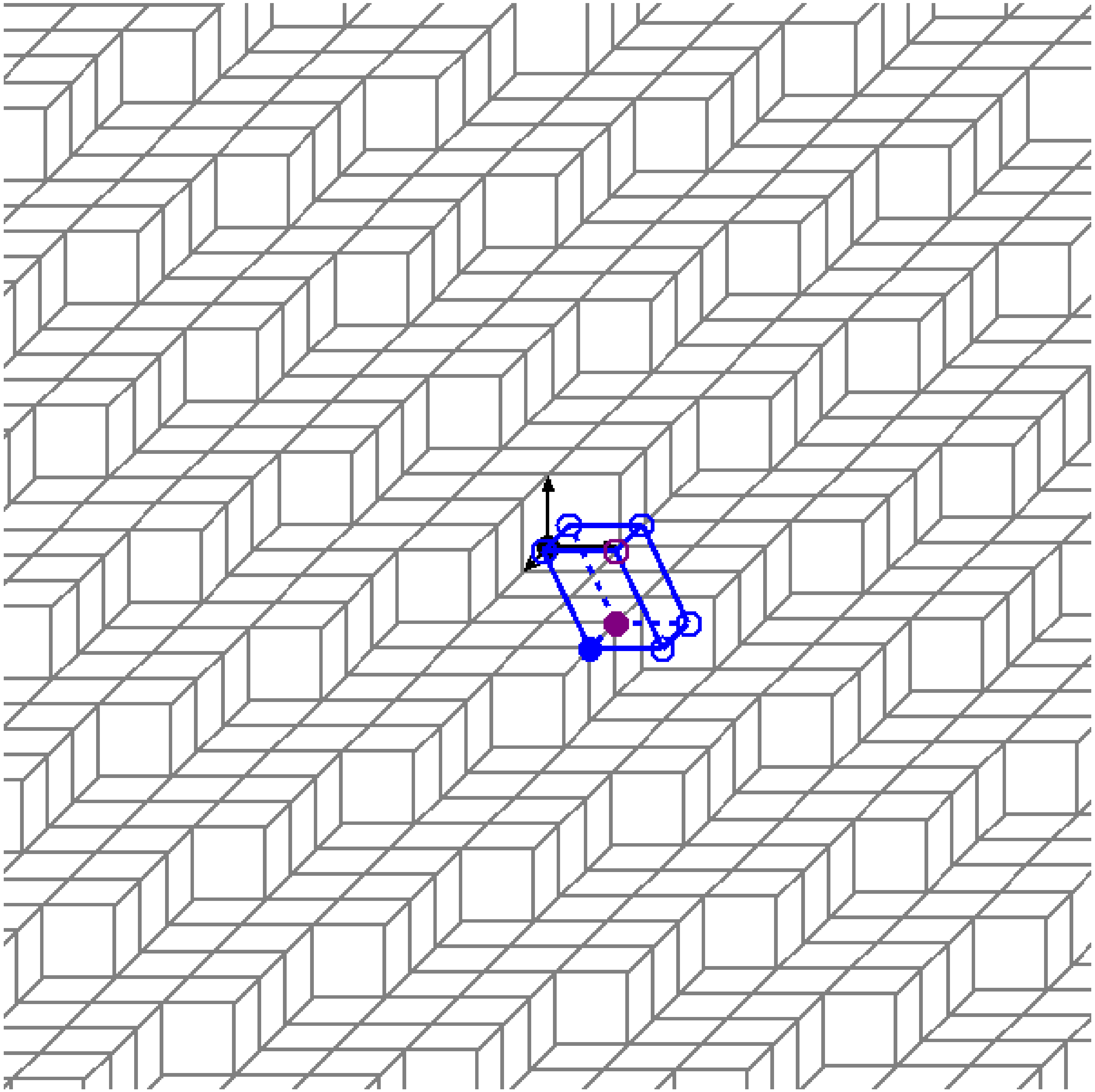
Illustration



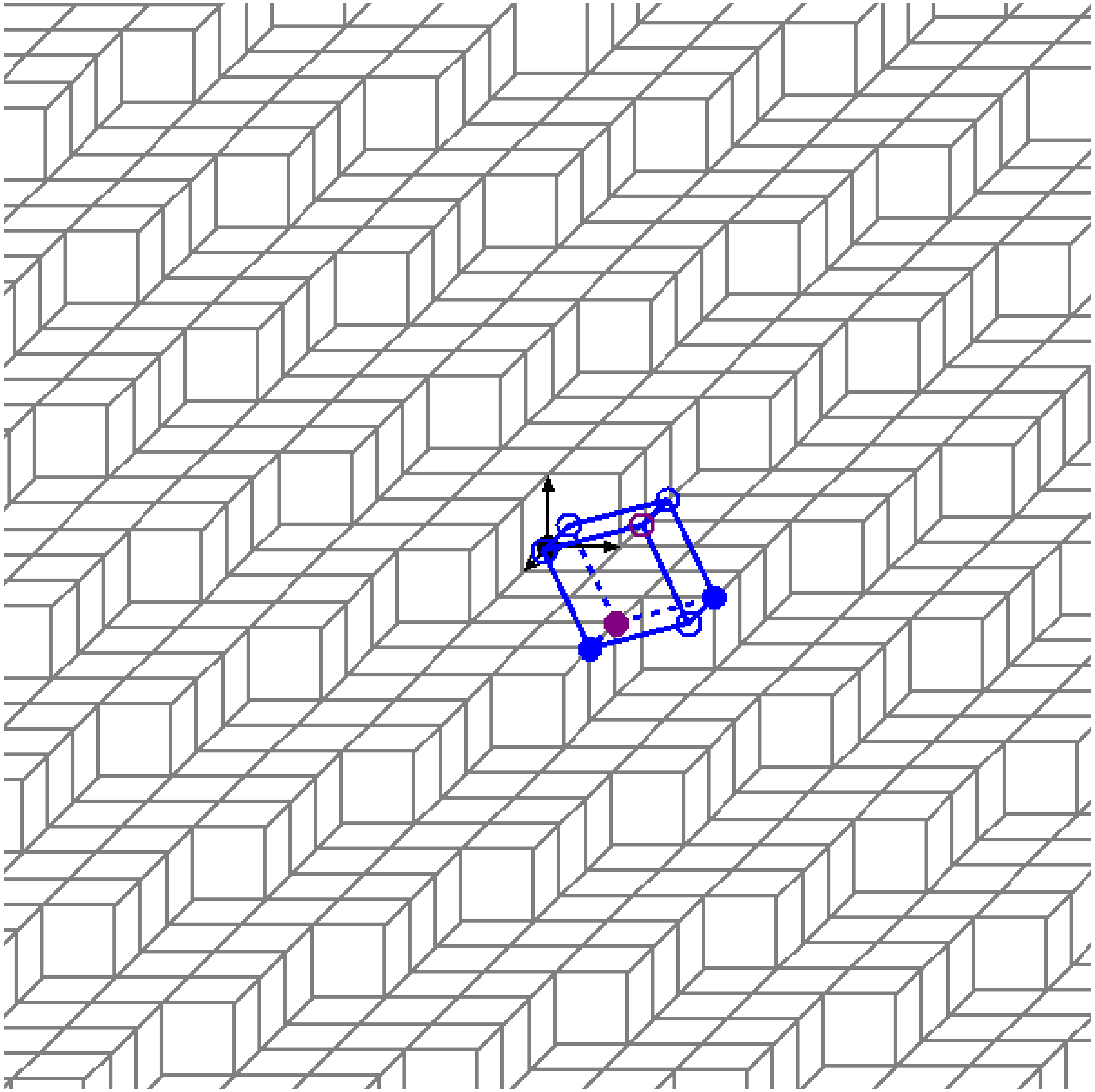
Illustration



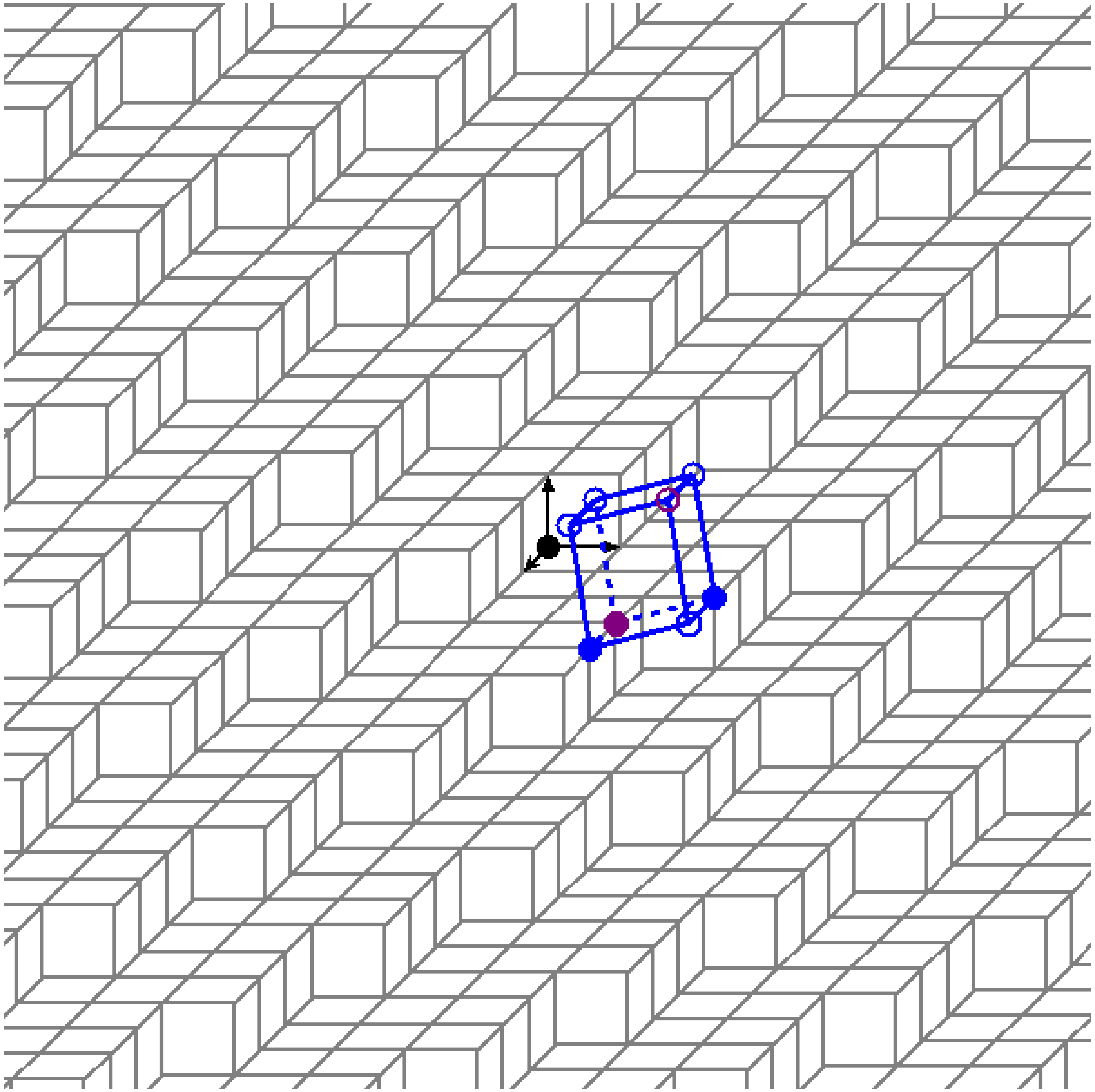
Illustration



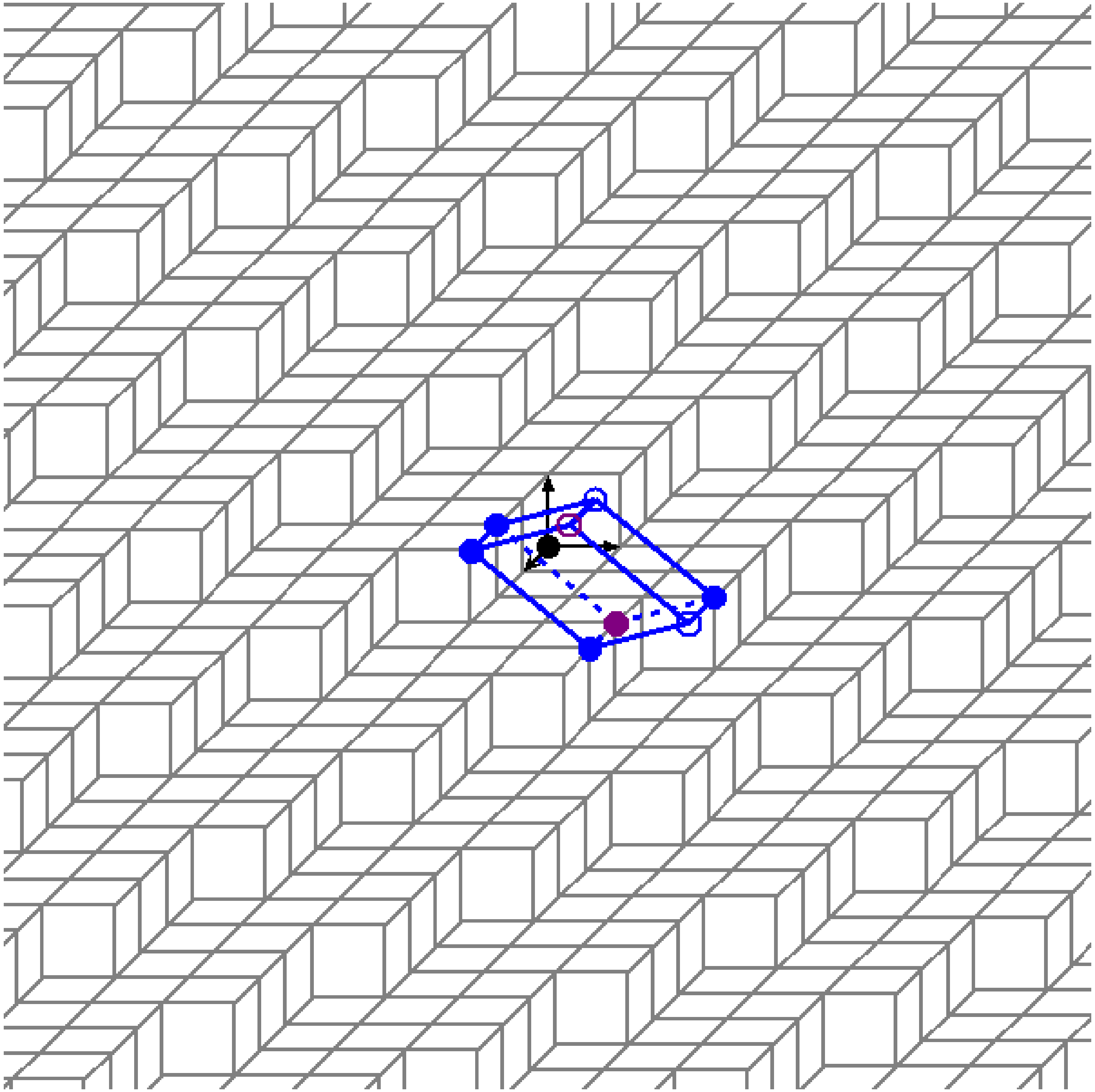
Illustration



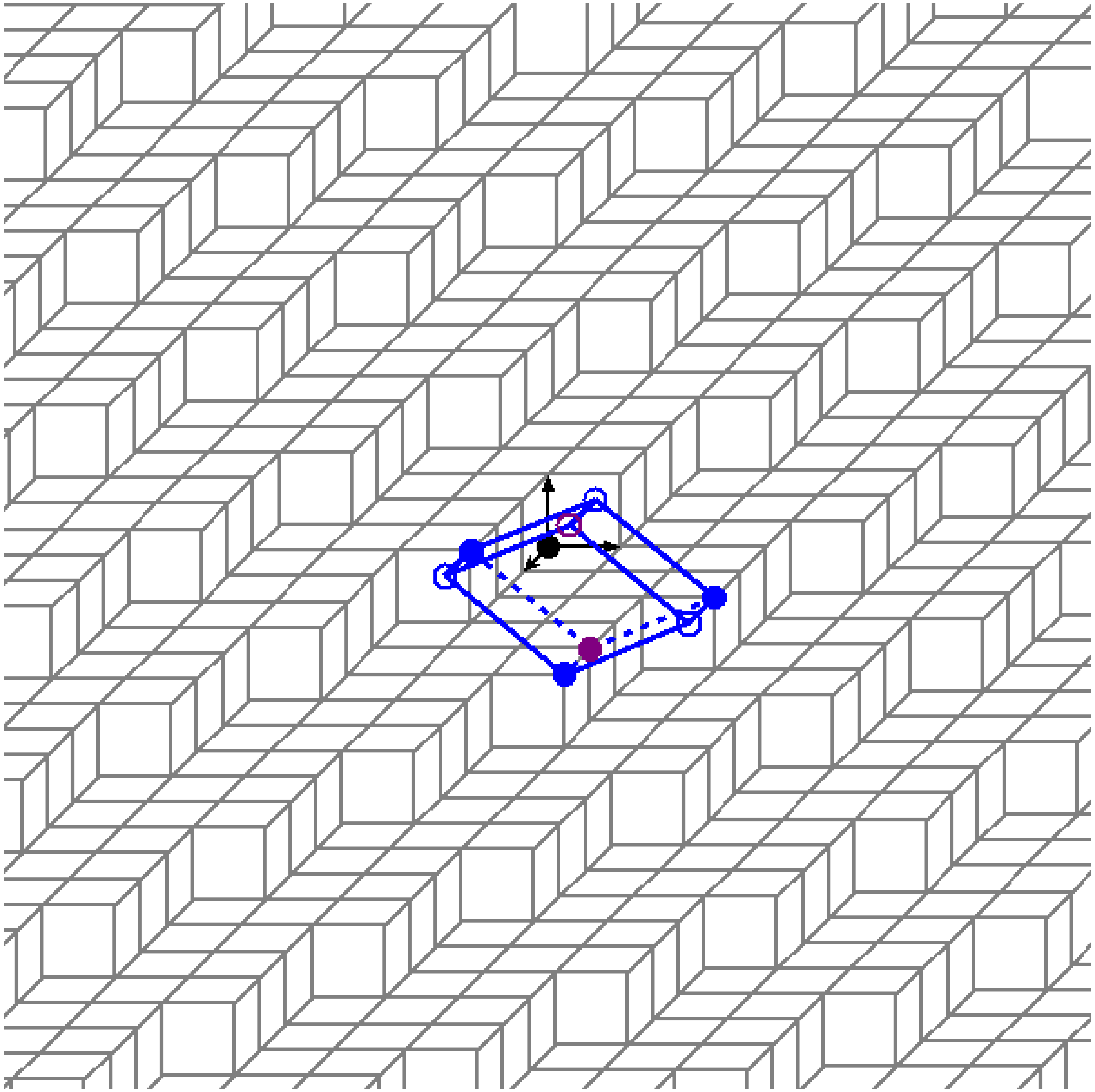
Illustration



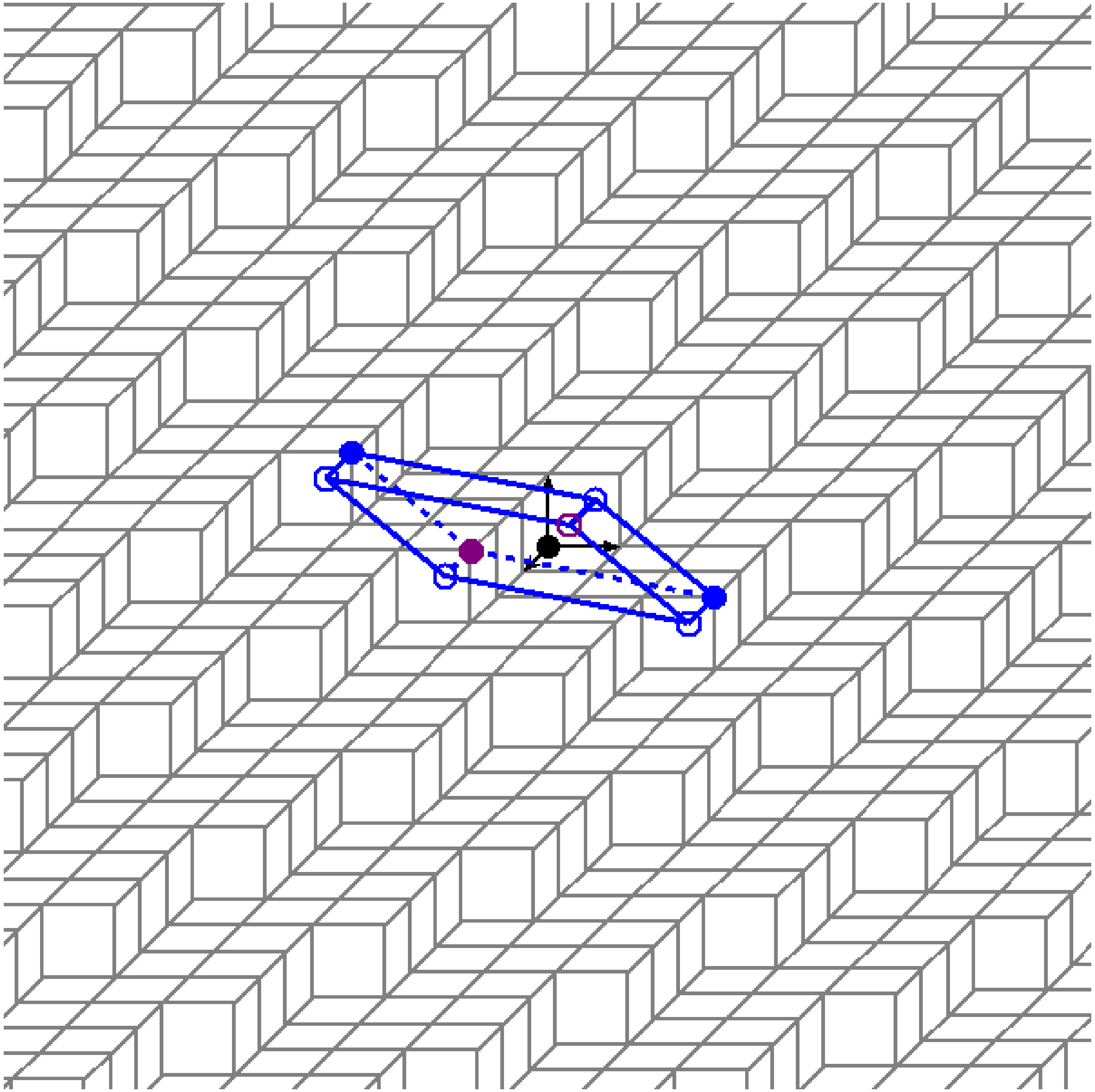
Illustration



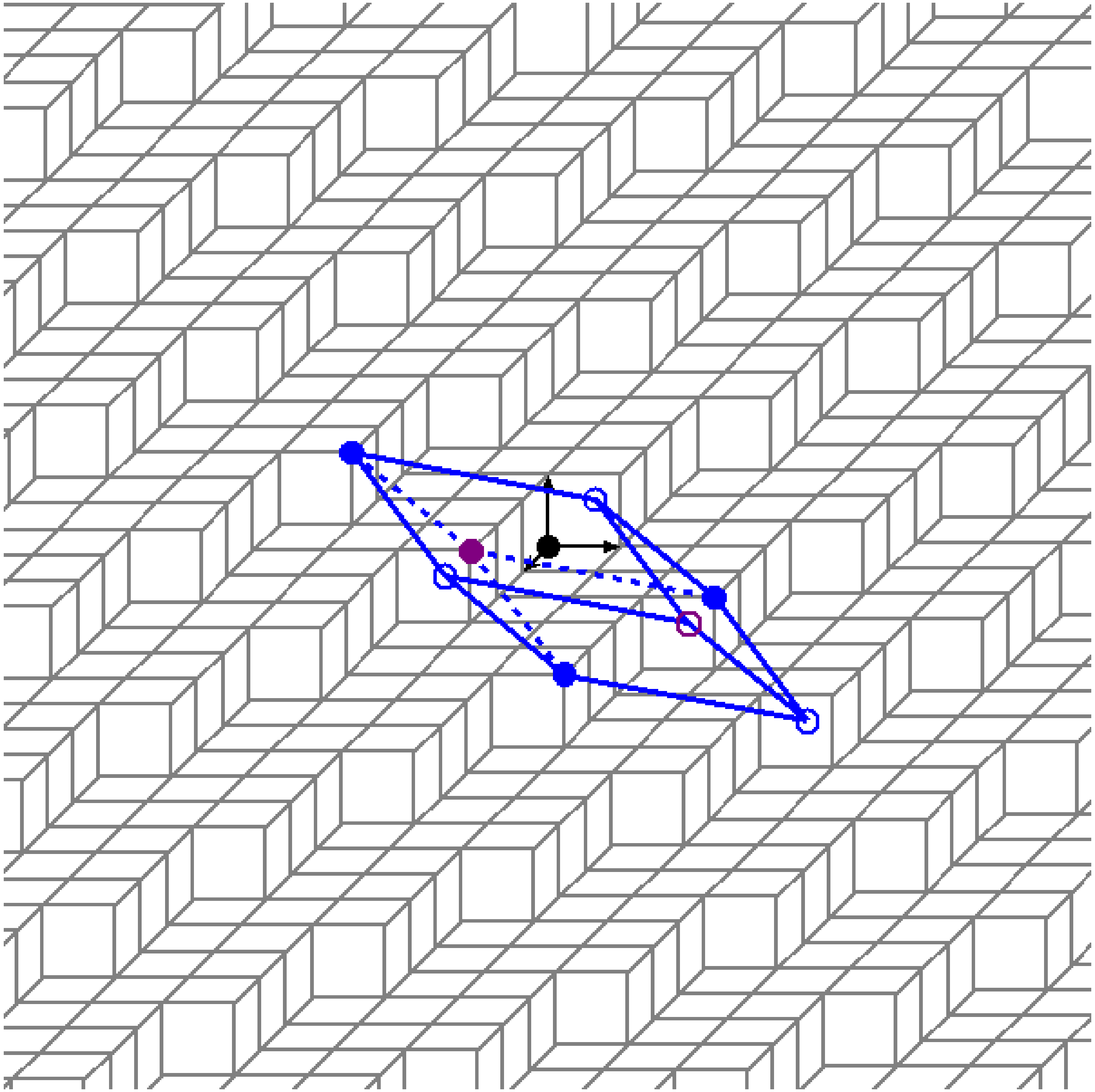
Illustration



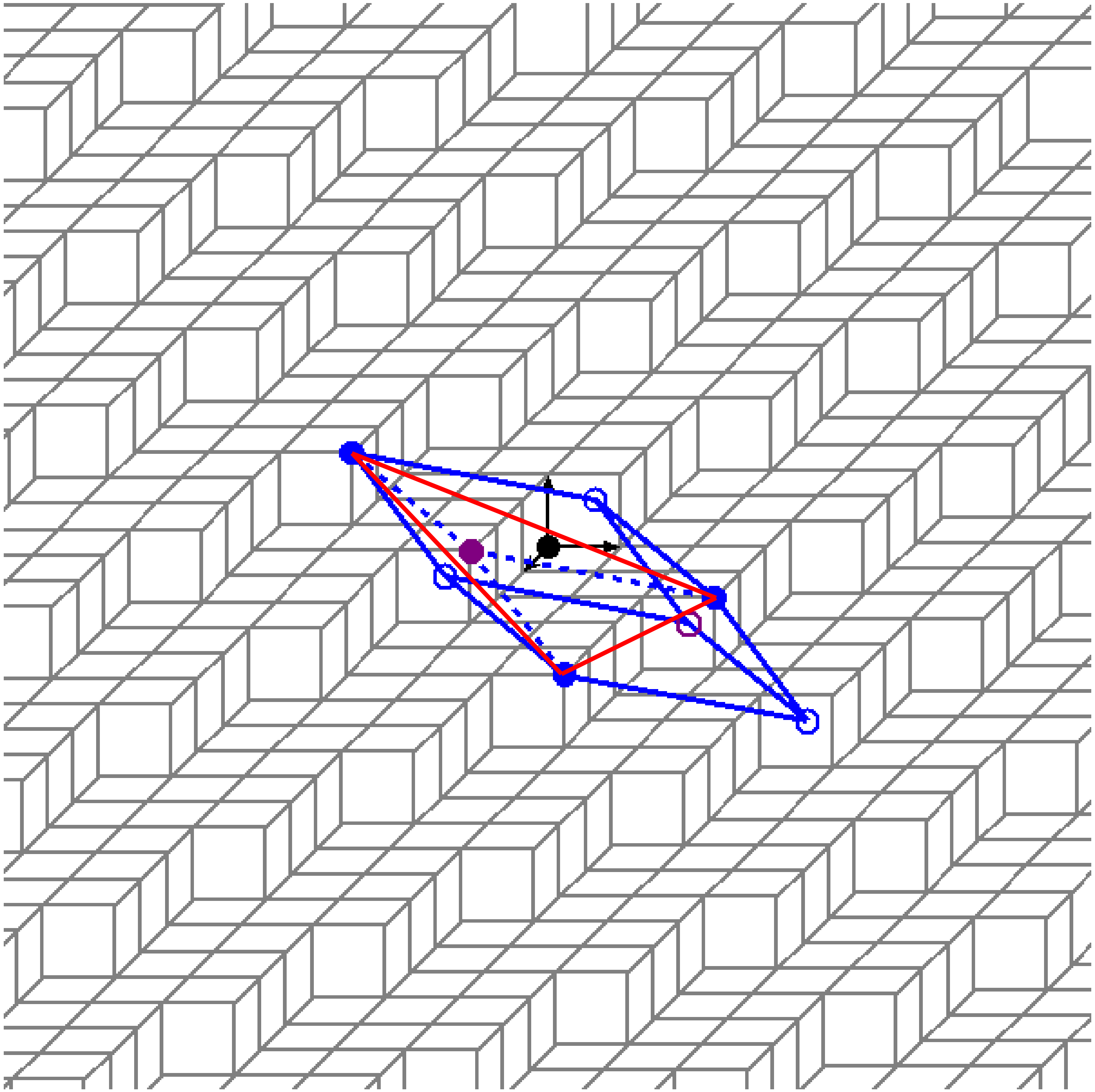
Illustration



Illustration



Illustration



Garanties

Convergence et correction

- le nombre d'étapes est majoré par $\|\mathbf{N}\|_1 - 3$,
- la normale à la dernière étape est $\mathbf{N}$ ou $-\mathbf{N}$ (suivant le tétraèdre),
- si $\bar{q} \in \{\|\mathbf{N}\|_1, \|\mathbf{N}\|_1 + 1\}$, alors pas d'inversion,
- si $\bar{q} \geq \|\mathbf{N}\|_1 + 2$, alors au moins une inversion.

Garanties

Convergence et correction

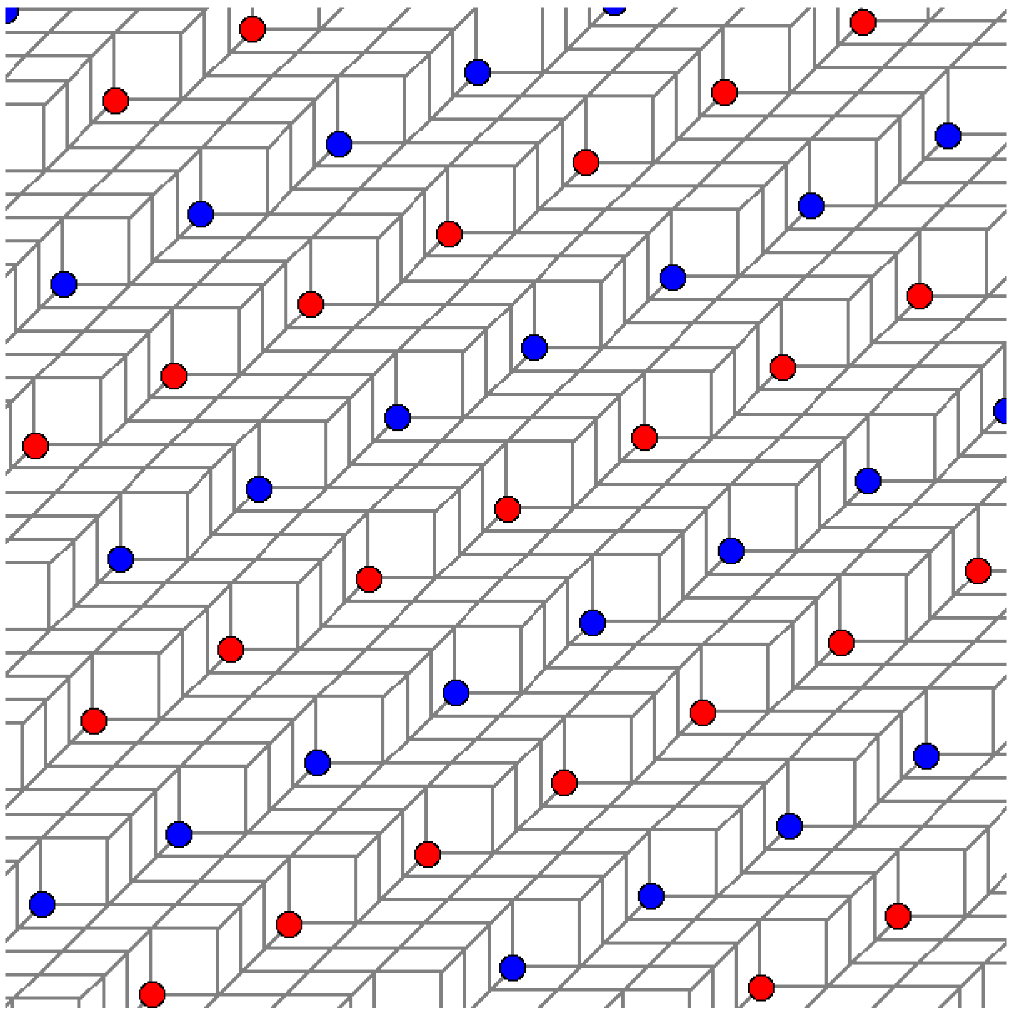
- le nombre d'étapes est majoré par $\|\mathbf{N}\|_1 - 3$,
- la normale à la dernière étape est $\mathbf{N}$ ou $-\mathbf{N}$ (suivant le tétraèdre),
- si $\bar{q} \in \{\|\mathbf{N}\|_1, \|\mathbf{N}\|_1 + 1\}$, alors pas d'inversion,
- si $\bar{q} \geq \|\mathbf{N}\|_1 + 2$, alors au moins une inversion.

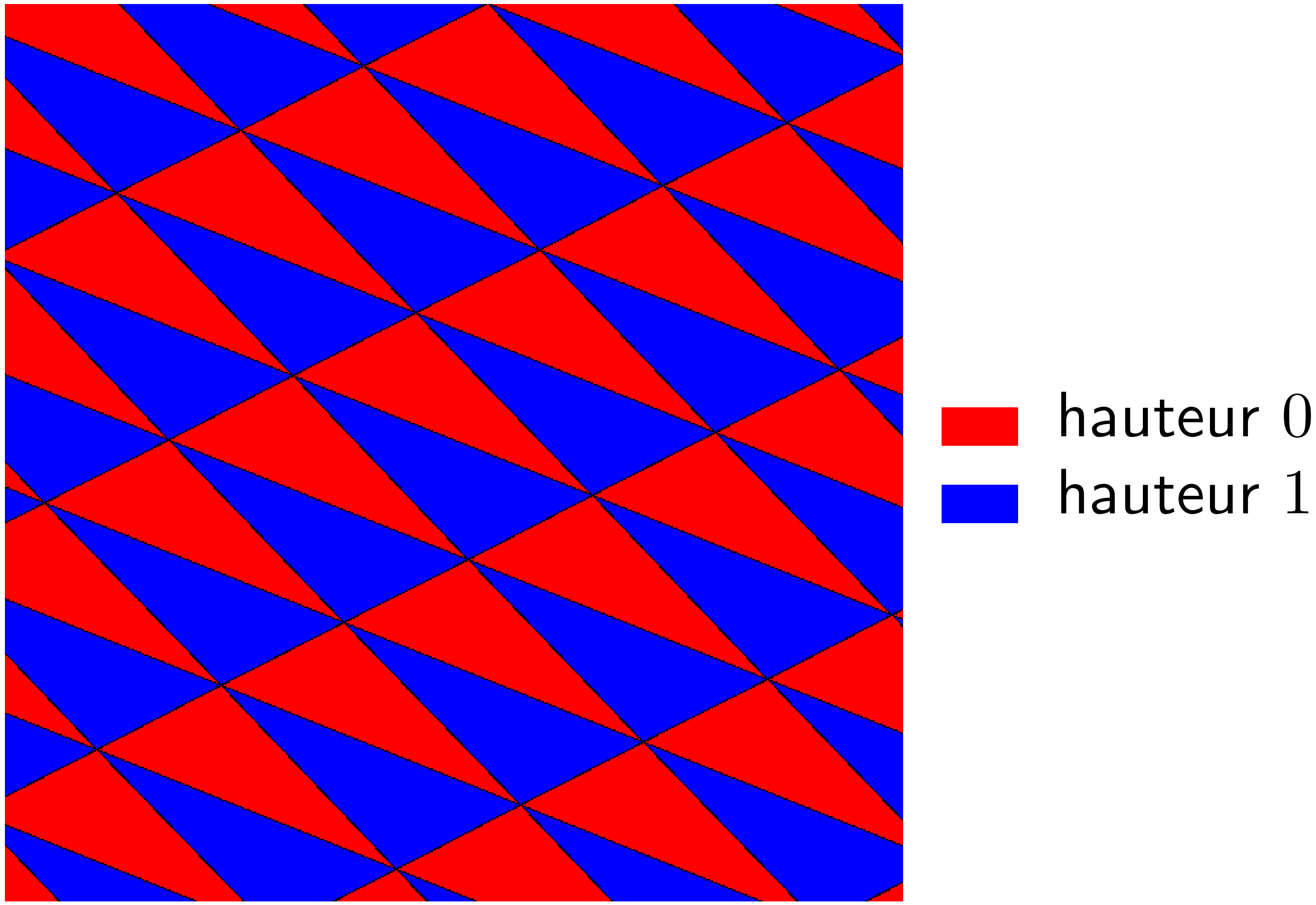
Complexité

- le nombre d'appels à $\mathrm{NotAbove}_q$ est majoré par $O(\|\mathbf{N}\|_1)$,
- le nombre d'appels à $\mathrm{InPlane}$ est majoré par $O(\|\mathbf{N}\|_1 \log(\|\mathbf{N}\|_1))$,
- en pratique, plus proche de $O(\log \|\mathbf{N}\|_1)$.

Réponses aux limitations I

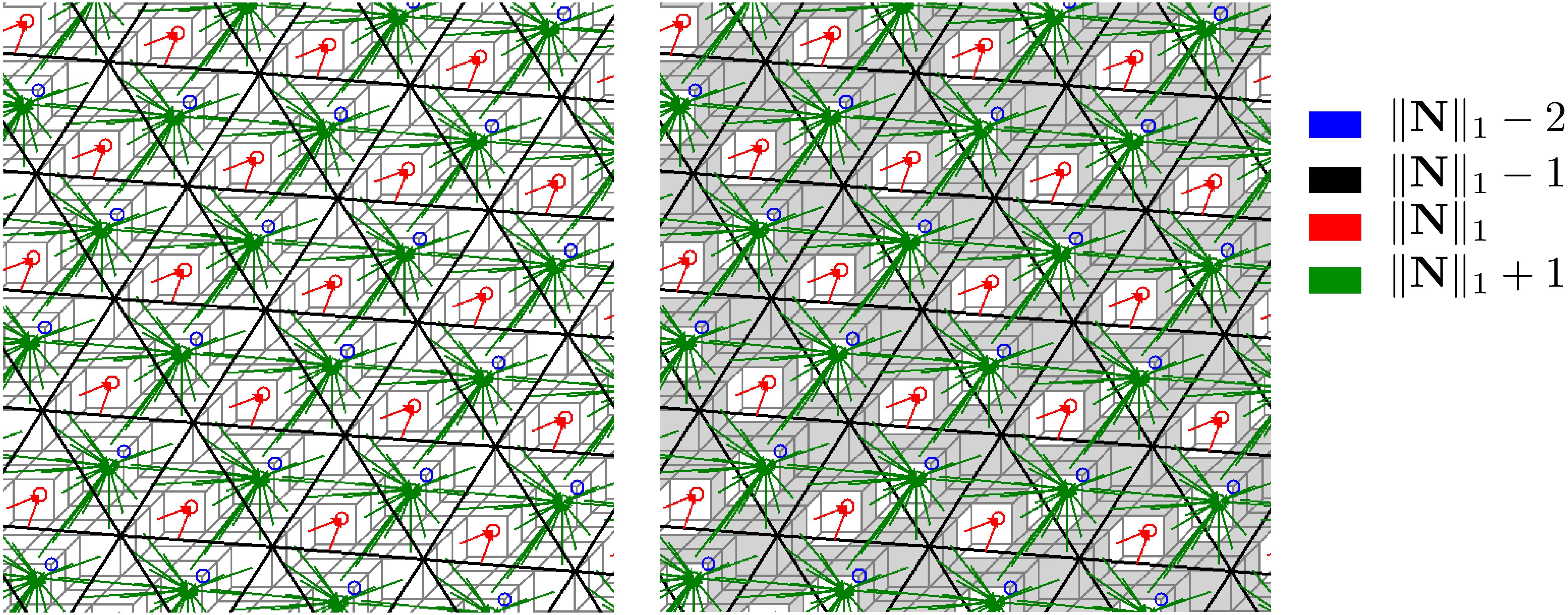
$\Longrightarrow$ on retrouve tous les triangles dans le réseau des points d'appui supérieur





Réponses aux limitations II

$\Longrightarrow$ on retrouve la bonne normale en partant de n'importe quel point de départ



Algorithme lancé sur 7618 surfels du plan de normale $\mathbf{N} = (3, 7, 15)$

Observations :

- l'algorithme semble local pour tous les surfels (pas plus de 2 triangles),
- la base finale est toujours réduite. (R voisinage)

Conclusion et perspectives

Nous avons présenté :

- Un algorithme générique pour l'estimation de vecteurs normaux,
- Convergence et correction en partant de n'importe quel point de départ,
- Complexité comparable aux algorithmes existants.

Conclusion et perspectives

Nous avons présenté :

- Un algorithme générique pour l'estimation de vecteurs normaux,
- Convergence et correction en partant de n'importe quel point de départ,
- Complexité comparable aux algorithmes existants.

Perspectives :

Conclusion et perspectives

Nous avons présenté :

- Un algorithme générique pour l'estimation de vecteurs normaux,
- Convergence et correction en partant de n'importe quel point de départ,
- Complexité comparable aux algorithmes existants.

Perspectives :

- Amélioration de la complexité,

Conclusion et perspectives

Nous avons présenté :

- Un algorithme générique pour l'estimation de vecteurs normaux,
- Convergence et correction en partant de n'importe quel point de départ,
- Complexité comparable aux algorithmes existants.

Perspectives :

- Amélioration de la complexité,
- Convergence et stabilité de l'estimateur,

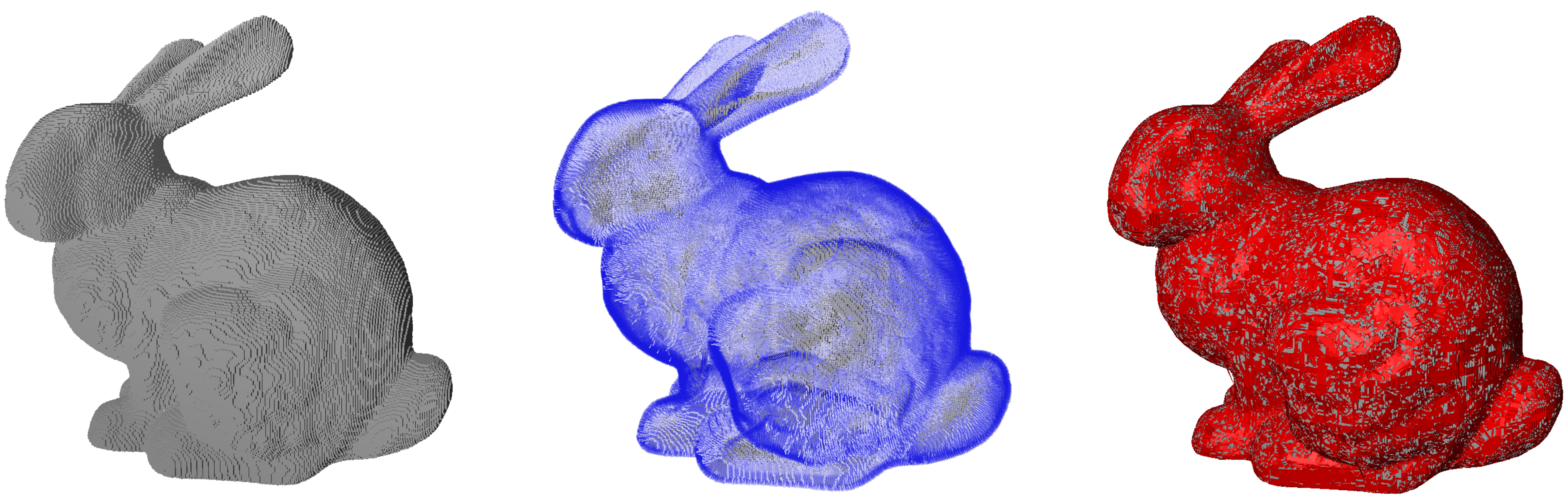
Conclusion et perspectives

Nous avons présenté :

- Un algorithme générique pour l'estimation de vecteurs normaux,
- Convergence et correction en partant de n'importe quel point de départ,
- Complexité comparable aux algorithmes existants.

Perspectives :

- Amélioration de la complexité,
- Convergence et stabilité de l'estimateur,
- Surfaces digitales : estimation, reconstruction, plans maximaux...



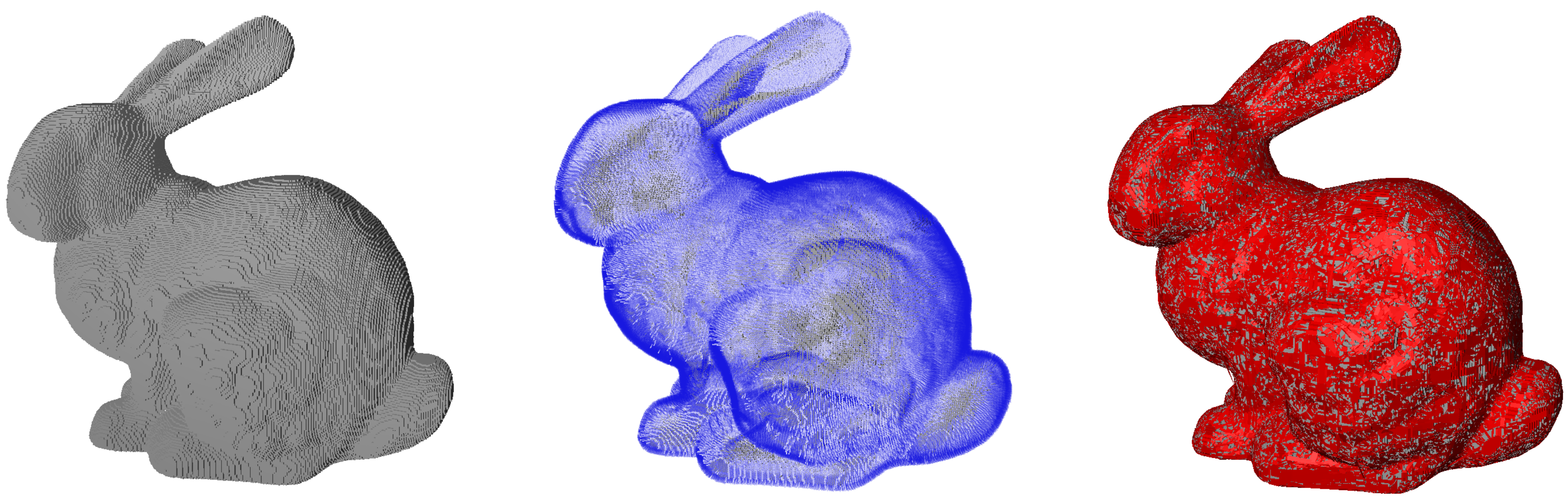
Conclusion et perspectives

Nous avons présenté :

- Un algorithme générique pour l'estimation de vecteurs normaux,
- Convergence et correction en partant de n'importe quel point de départ,
- Complexité comparable aux algorithmes existants.

Perspectives :

- Amélioration de la complexité,
- Convergence et stabilité de l'estimateur,
- Surfaces digitales : estimation, reconstruction, plans maximaux...



Merci pour votre attention