EJCIM 2018

Géométrie Numérique (partie 2)

Gradient Conjugué

Plan

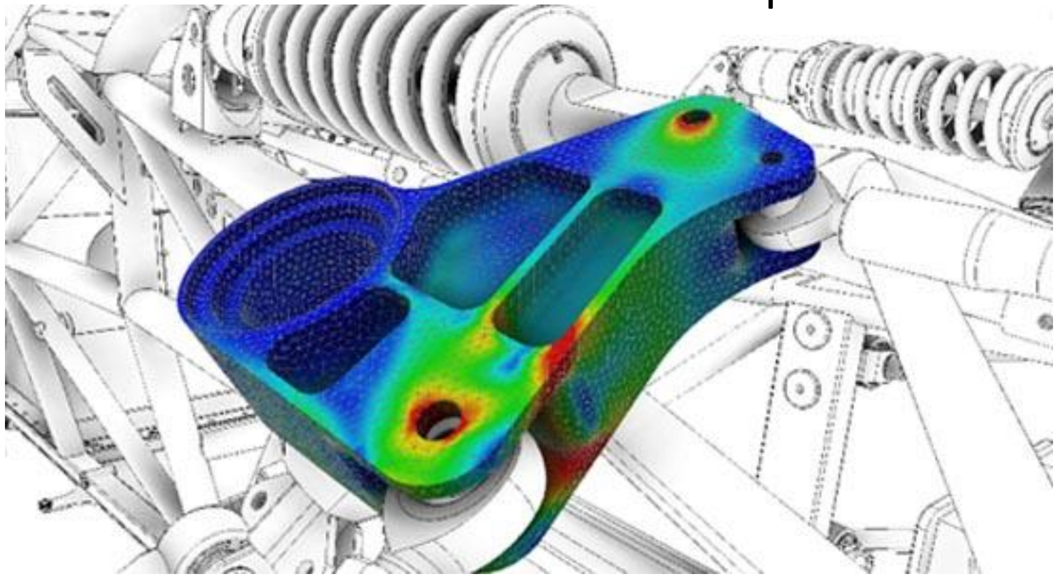
- **Introduction**
- Le problème
- Résolution par gradient
- Résolution par conjugaison
- Gradient conjugué
- Contemplation de performance pour des problèmes discrétisés

Introduction : problème

- Résoudre des systèmes linéaires
  - Soit une matrice A et un vecteur b,
  - on veut trouver x tel que Ax=b,
  - sans calculer A^{-1}
- Limitation: A est symétrique définie positive
  - A et b sont composés de réels
  - A est carrée
  - $A^T=A$
  - $x^TAx\geq 0$ pour tout x

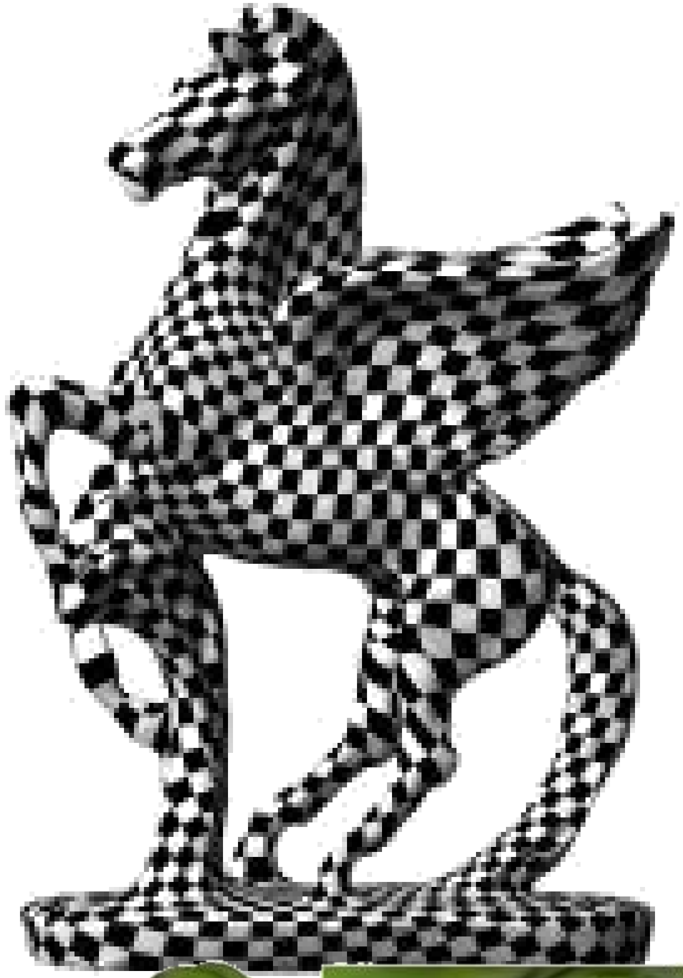
Introduction : applications

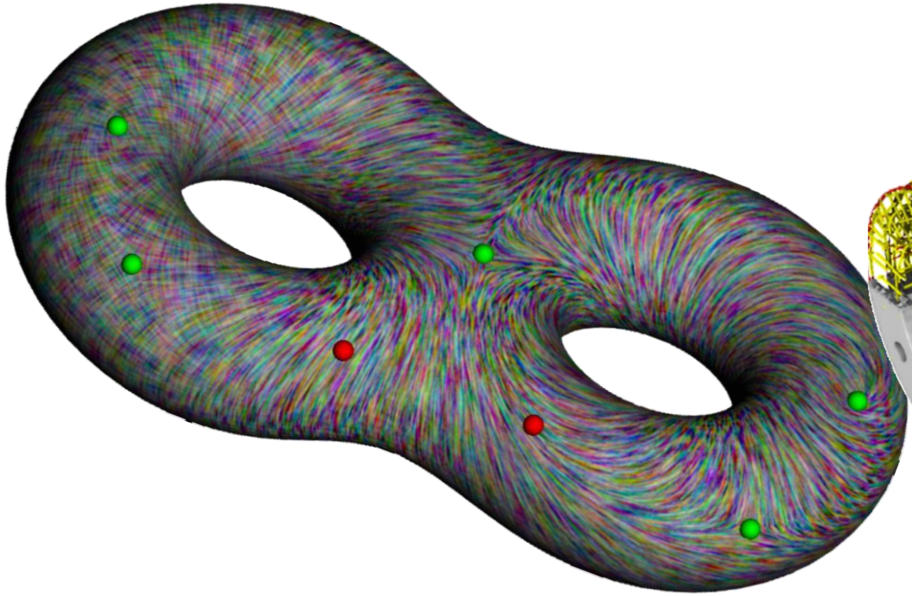
- Simulation numérique:
  - Définition d’un problème aux dérivées partielles
  - Discrétisation dans une base de fonction
  - Résolution des coefficients
    - Cas linéaire => juste résoudre Ax=b
    - Cas non linéaire => plusieurs itérations de Ax=b

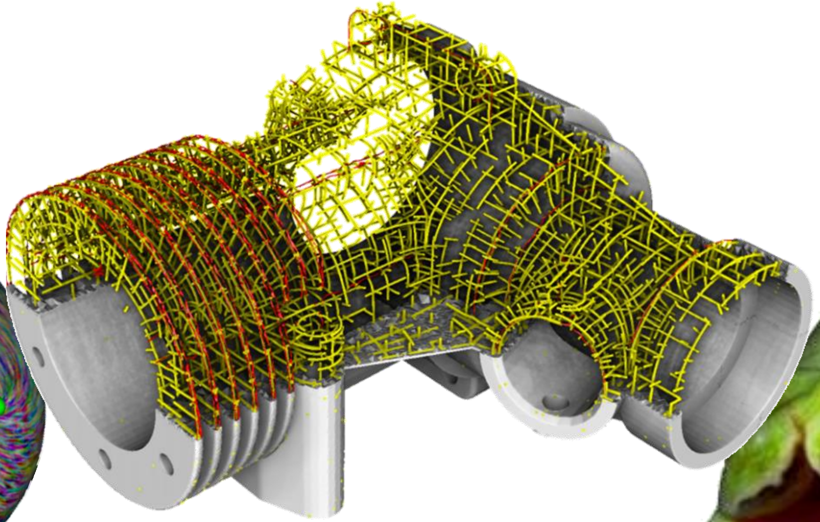


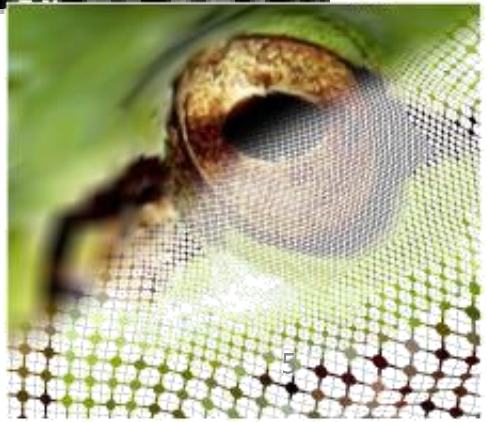
Introduction: applications

- Moindres carrés
  - On aimerait avoir Ax-b=0
  - Mais on a plus d'équations que d'inconnues
  - Alors, on minimize $|Ax-b|^2$
  - En résolvant $A^TAx=A^Tb$









Plan

- *Introduction*
- **Le problème**
  - **Problème équivalent**
  - **Intérêt des petites lignes du contrat… (Sym. Def. +)**
- Résolution par gradient
- Résolution par conjugaison
- Gradient conjugué
- Contemplation de performance pour des problèmes discrétisés

Problème équivalent

- Résoudre $Ax-b=0$ équivaut à minimiser $f(x) = x^T Ax-2b^T x$
- Pourquoi ?
  - Le vecteur x^* qui annule les dérivées df/dx_i est soit un extrema, soit un point selle de $f(x)$.

Problème équivalent

- Résoudre $Ax-b=0$ équivaut à minimiser $f(x) = x^TAx-2b^Tx$

(A symmétrique)

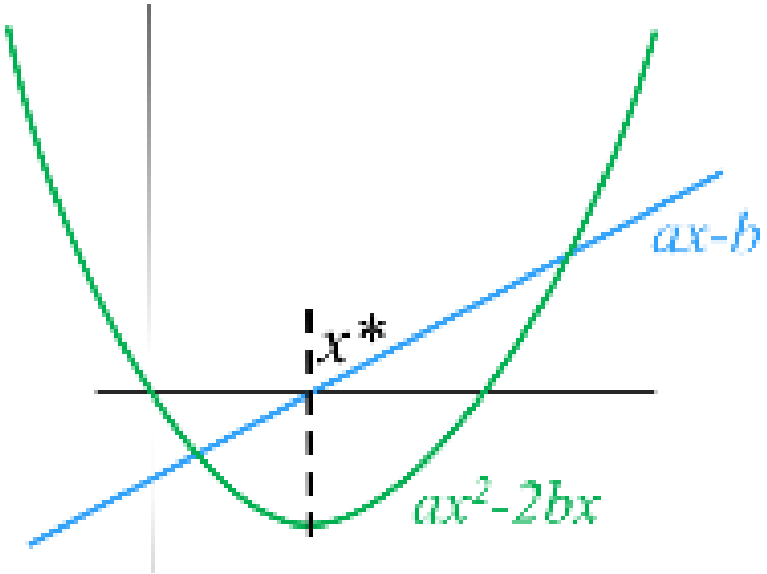
- Pourquoi ?
  - Le vecteur x^* qui annule les dérivées df/dx_i est soit un extrema, soit un point selle de $f(x)$.

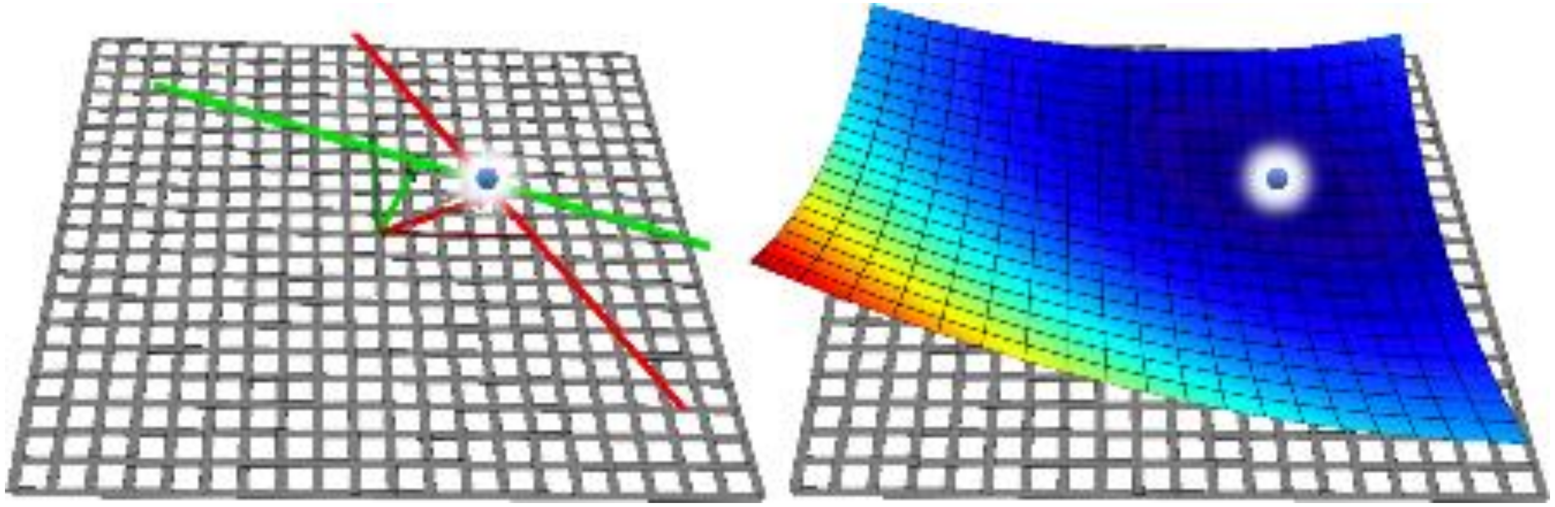
Problème équivalent

- Résoudre $Ax-b=0$ équivaut à minimiser $f(x) = x^TAx-2b^Tx$
- Pourquoi ?
  - Le vecteur x^* qui annule les dérivées df/dx_i est soit un extrema, soit un point selle de $f(x)$.
  - Avec A définie positive, x^* est forcément un minimum.

Problème équivalent

- Résoudre $Ax-b=0$ équivaut à minimiser $f(x) = x^TAx-2b^Tx$
- Pourquoi ?
  - Le vecteur x^* qui annule les dérivées df/dx_i est soit un extrema, soit un point selle de $f(x)$.
  - Avec A définie positive, x^* est forcément un minimum.





A est définie positive… pourquoi ?

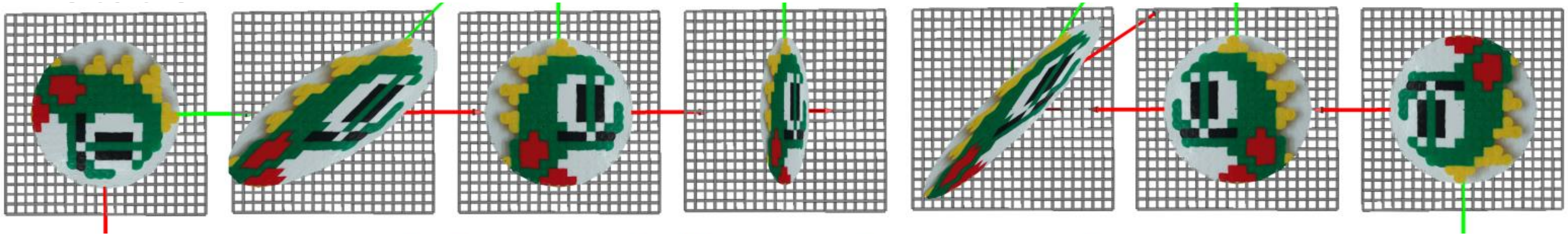
A est définie positive… pourquoi ?

$$\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \quad \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 1/2 & 0 \\ 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 1/2 \\ 1/2 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$$

définie positive

symétrique

A est définie positive… pourquoi ?

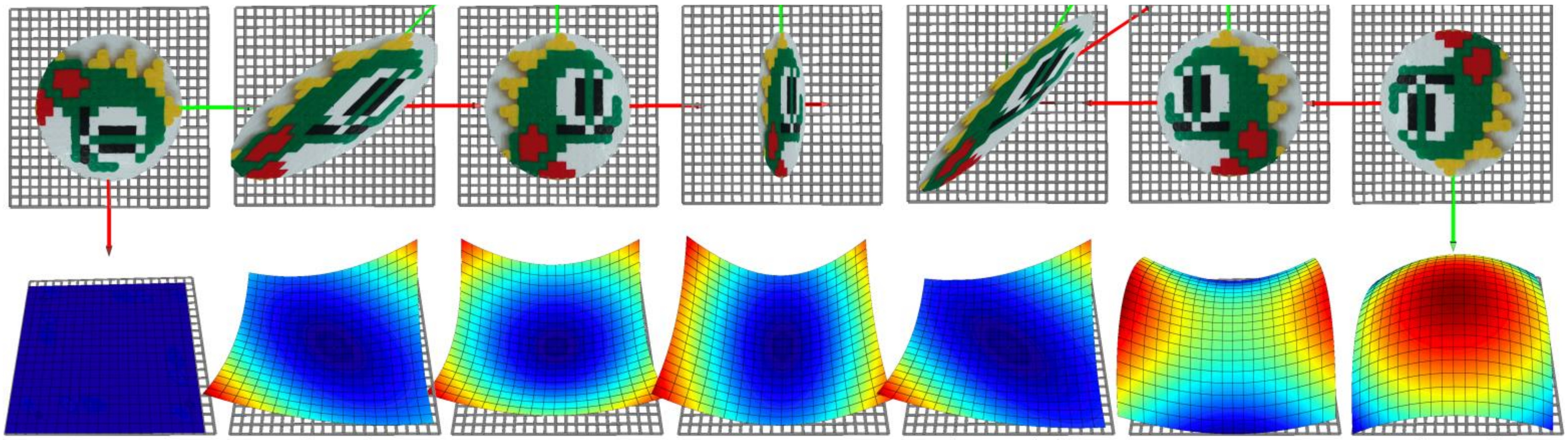


$$\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \quad \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 1/2 & 0 \\ 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 1/2 \\ 1/2 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$$

définie positive

symétrique

A est définie positive… pourquoi ?

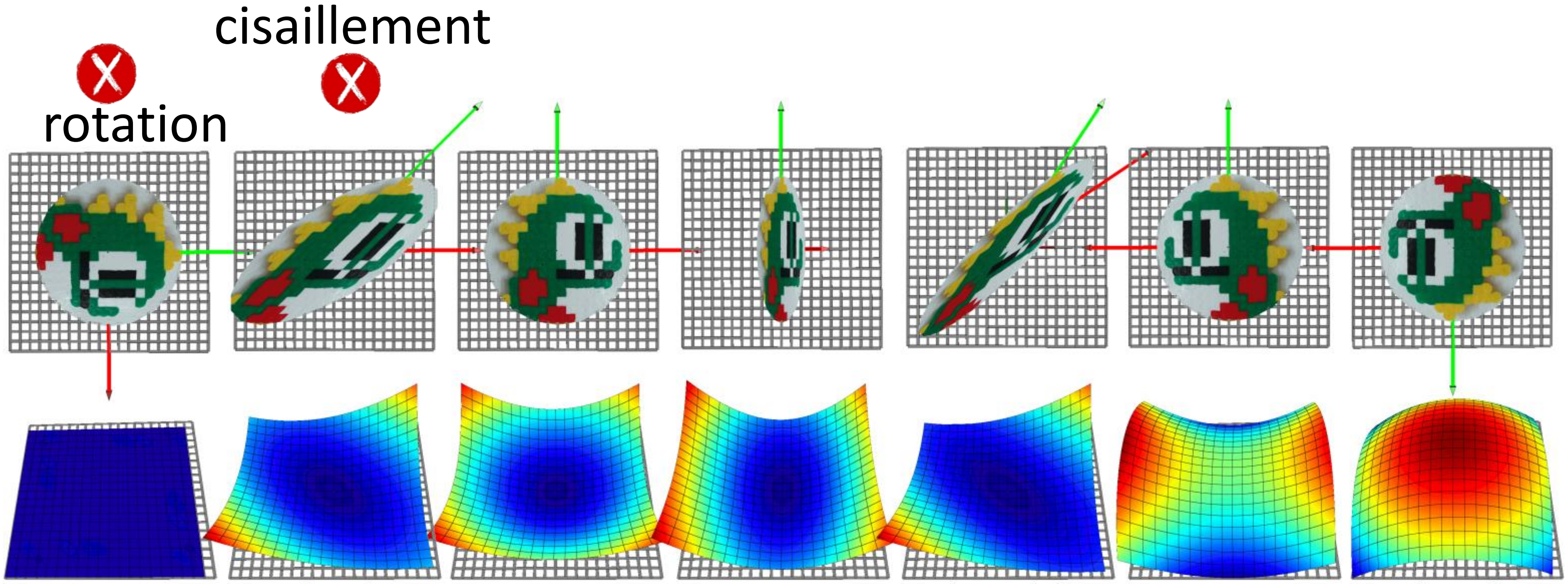


$$\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \quad \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 1/2 & 0 \\ 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 1/2 \\ 1/2 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$$

définie positive

symétrique

A est définie positive… pourquoi ?

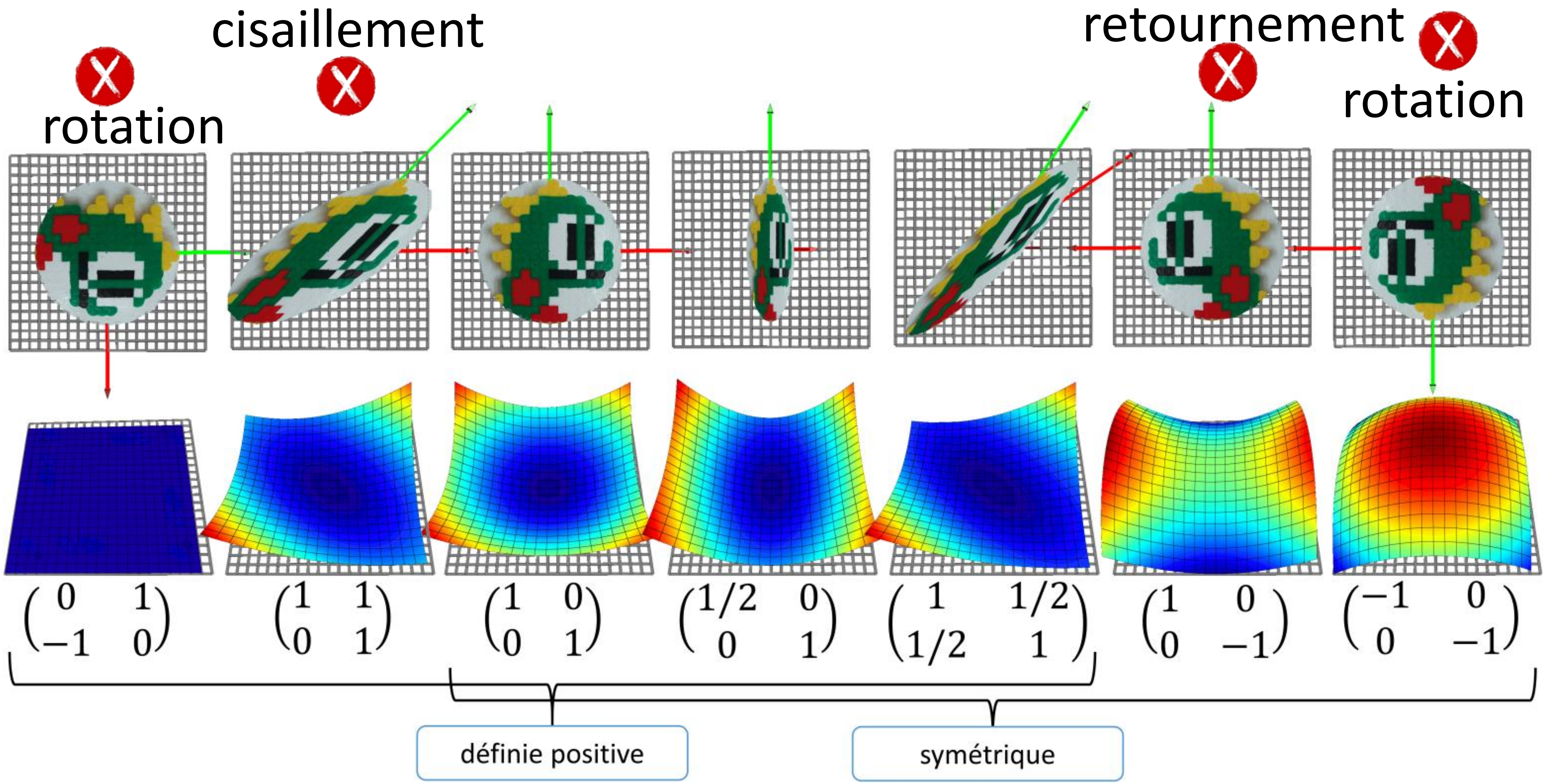


$$\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \quad \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 1/2 & 0 \\ 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 1/2 \\ 1/2 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$$

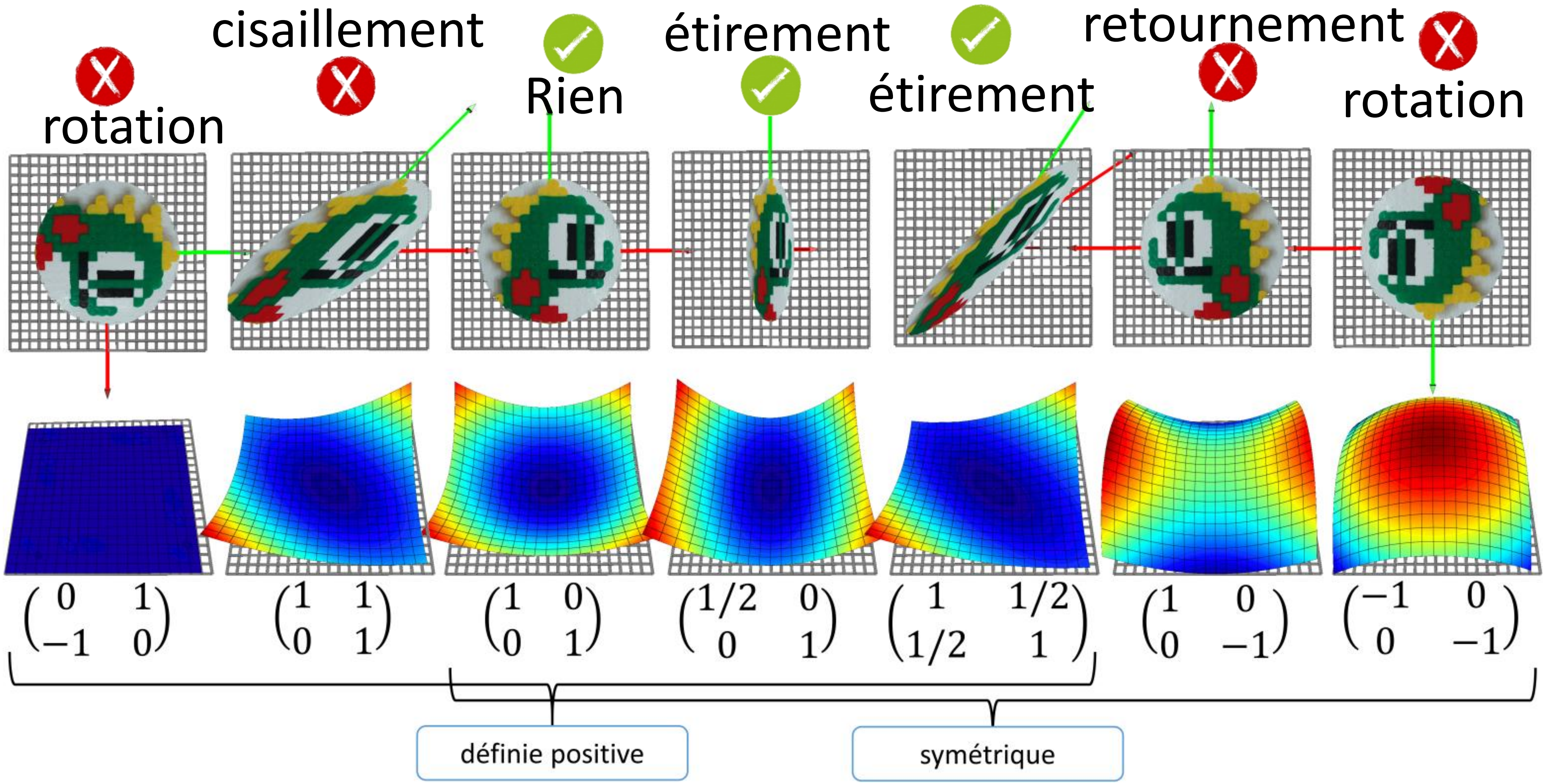
définie positive

symétrique

A est définie positive… pourquoi ?

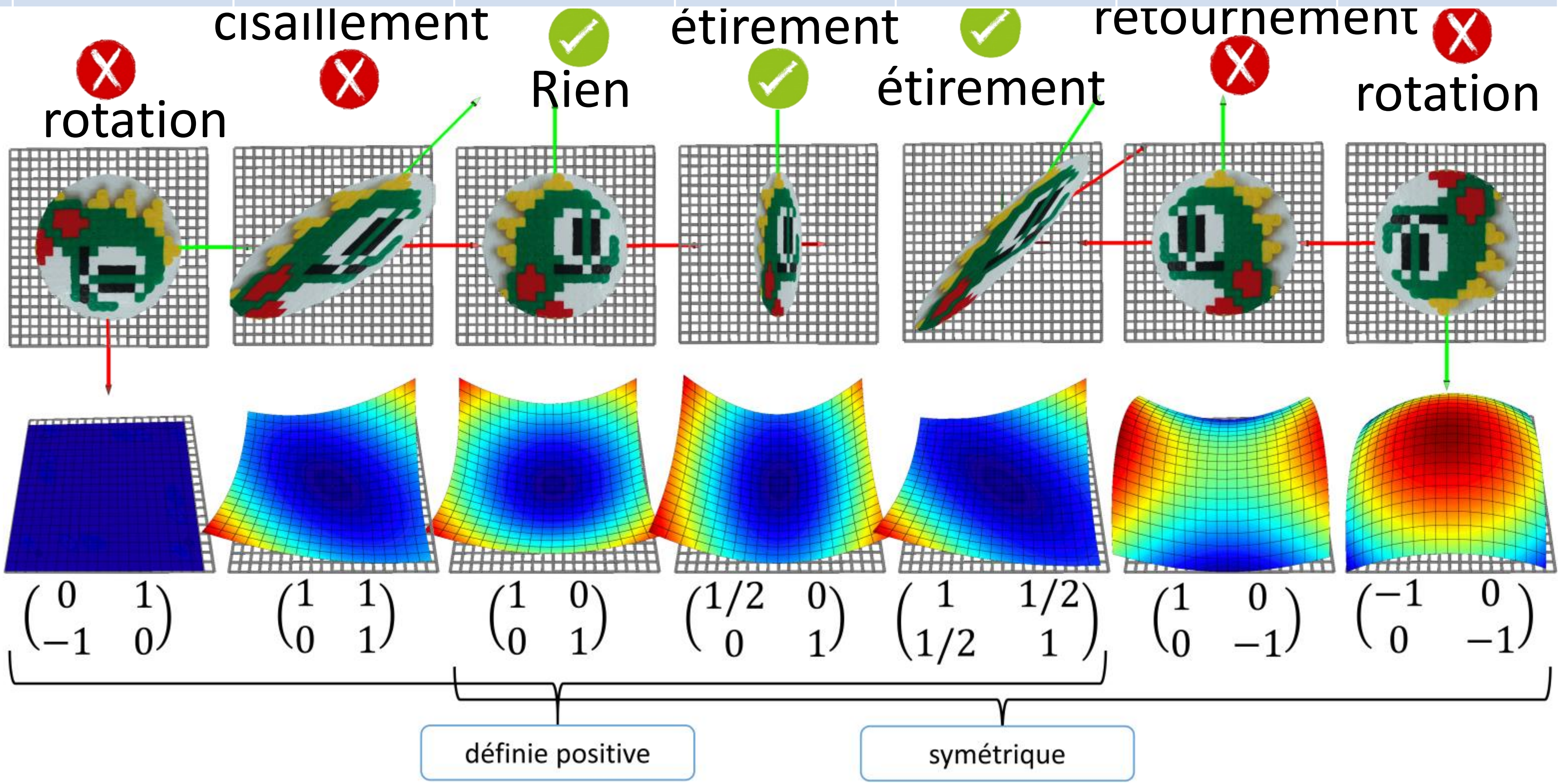


A est définie positive… pourquoi ?



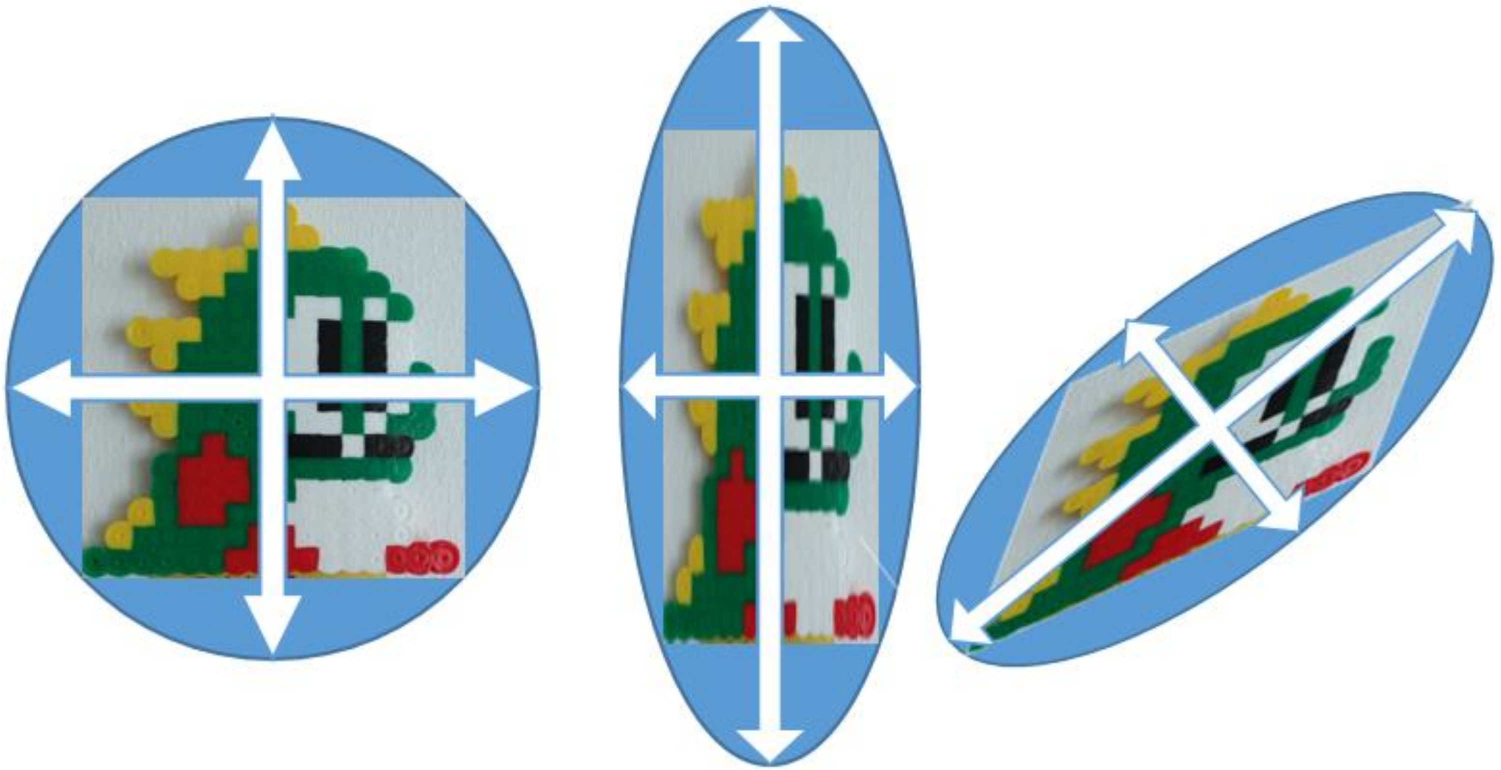
A est définie positive… pourquoi ?

Val. Propre	complexe	+	+	+	+	-+	--
Vect. propre	complexe	Non ortho	ortho	ortho	ortho	ortho	ortho



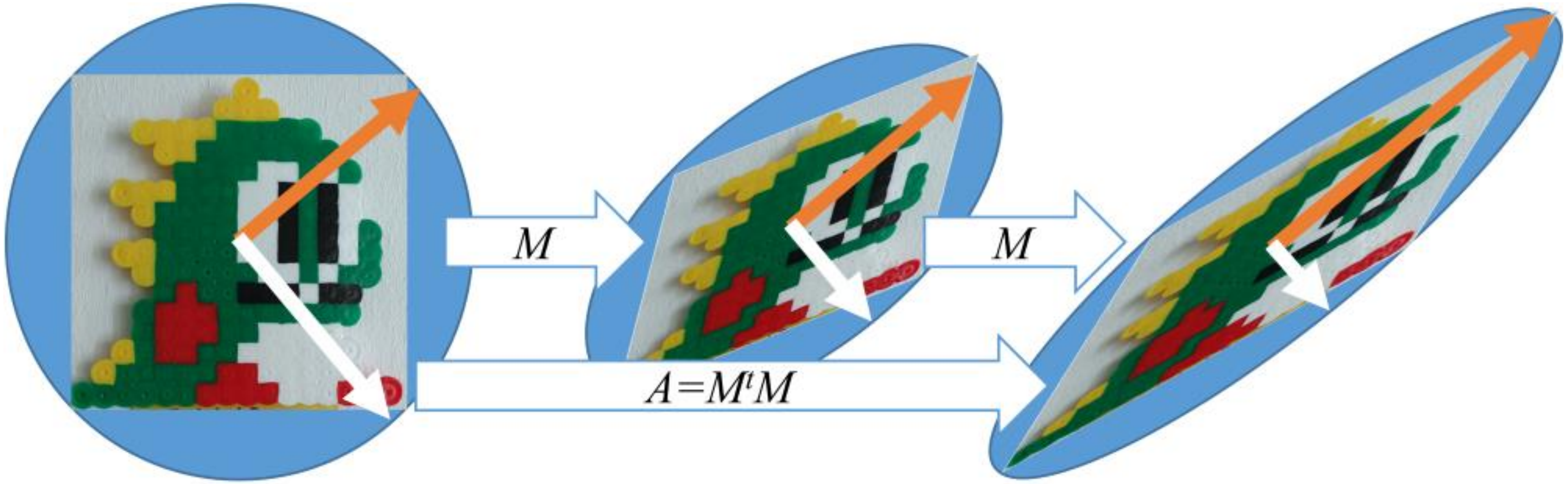
A est définie positive… pourquoi ?

- Cette application linéaire est simplement un étirement de l'espace dans des directions orthogonales



Soit M tel que A =$M^T M$: existence

- On peut même décomposer cette application linéaire en étirant en deux coups avec l'application associée à la matrice M définie comme suit
- M symétrique définie positive et étire dans les mêmes directions que A, mais avec des coefficients étant racines carrées de ceux de A.



Soit M tel que A =$M^T M$: intérêt

- On peut facilement calculer des produits scalaires entre des images de vecteurs par l'application M… et ce, sans expliciter M !
- Pour ce faire, on utilise
  - la définition de M i.e. $M^T M = A$, ce qui donne

$$\boxed{(Mu)^T.(Mv) = u^T M^T M v = u^T A v}$$

Soit M tel que A =$M^T M$: intérêt

- On peut facilement calculer des produits scalaires entre des images de vecteurs par l'application M… et ce, sans expliciter M !
- Pour ce faire, on utilise
  - la définition de M i.e. $M^T M = A$, ce qui donne

$$\boxed{(Mu)^T.(Mv) = u^T M^T M v = u^T A v}$$

  - et … $Ax^*=b$, qui permet de calculer

$$\boxed{(Mu)^T.(Mx^*) = u^T A x^* = u^T b}$$

Soit M tel que $A = M^T M$: intérêt

- On peut facilement calculer des produits scalaires entre des images de vecteurs par l'application M… et ce, sans expliciter M !
- Pour ce faire, on utilise
  - la définition de M i.e. $M^T M = A$, ce qui donne

$$\boxed{(Mu)^T.(Mv) = u^T M^T M v = u^T A v}$$

  - et … $Ax^* = b$, qui permet de calculer

$$\boxed{(Mu)^T.(Mx^*) = u^T A x^* = u^T b}$$

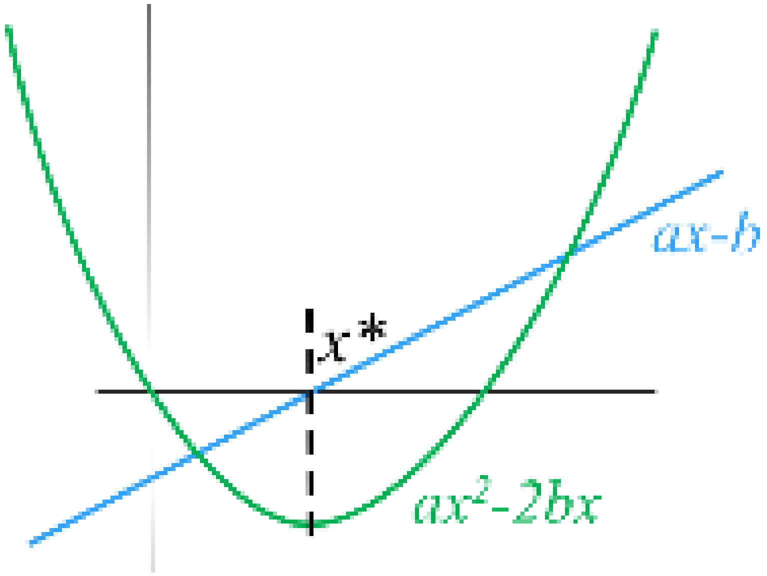
Ces produits scalaires donnent les **longueurs**, l'**orthogonalité**, et la **projection** des images de vecteurs par M.

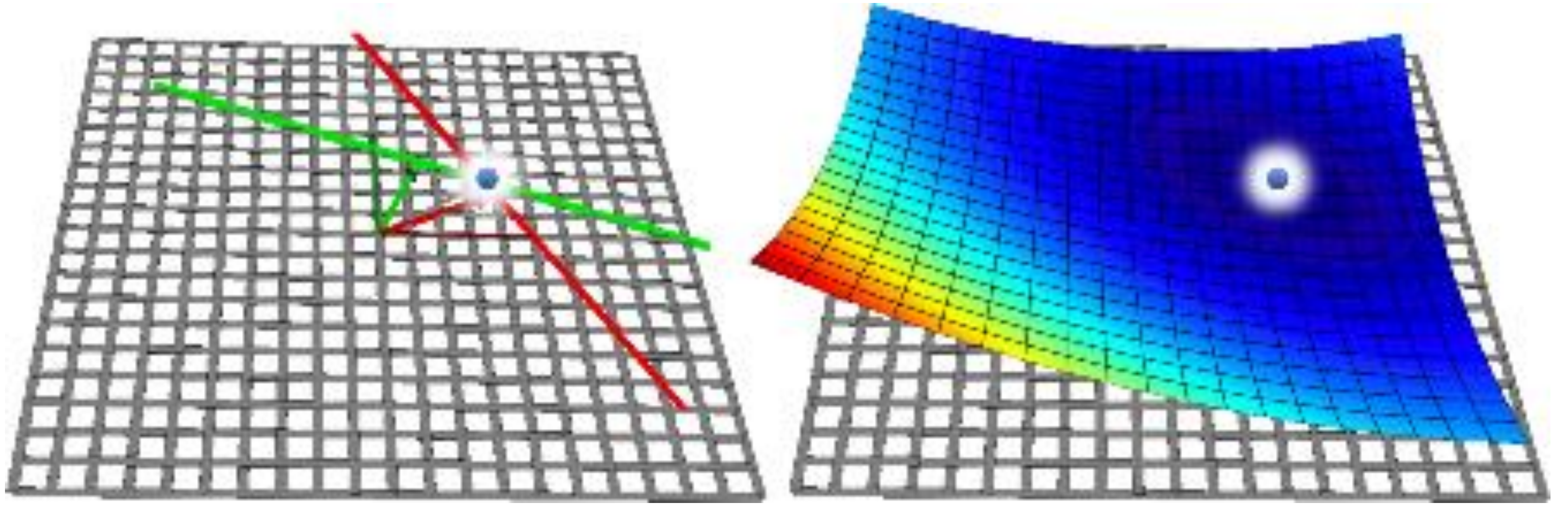
Plan

- *Introduction*
- *Le problème*
- **Résolution par gradient**
- Résolution par conjugaison
- Gradient conjugué
- Contemplation de performance pour des problèmes discrétisés

RAPPEL: Problèmes équivalents

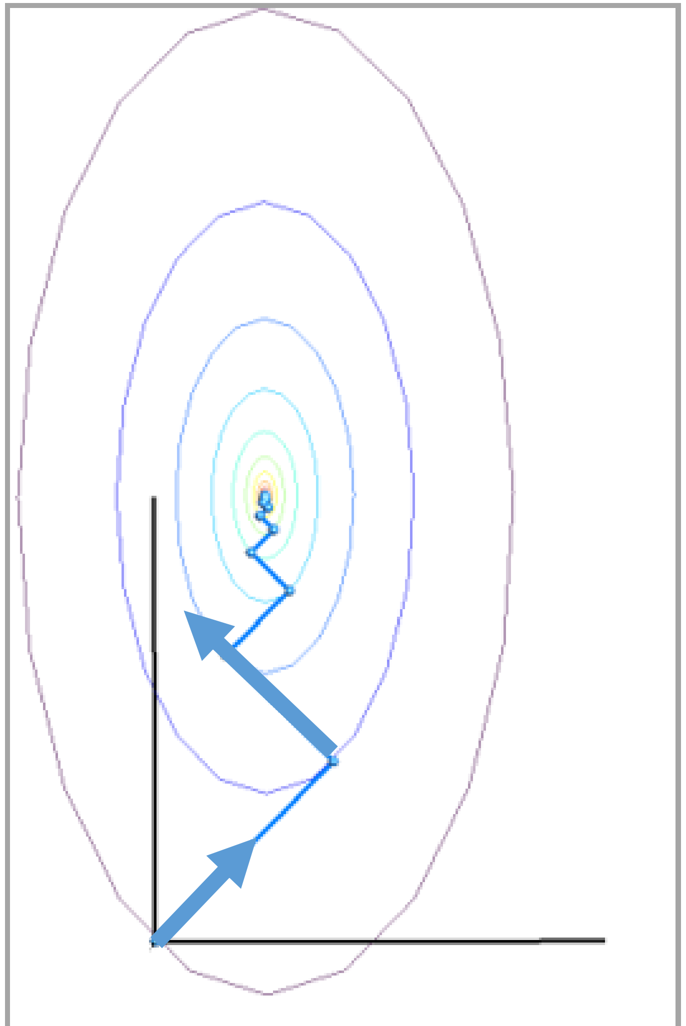
- Résoudre $Ax-b=0$ équivaut à minimiser $f(x) = x^TAx-2b^Tx$
- Pourquoi ?
  - Le vecteur x^* qui annule les dérivées df/dx_i est soit un extrema, soit un point selle de $f(x)$.
  - Avec A définie positive, x^* est forcément un minimum.





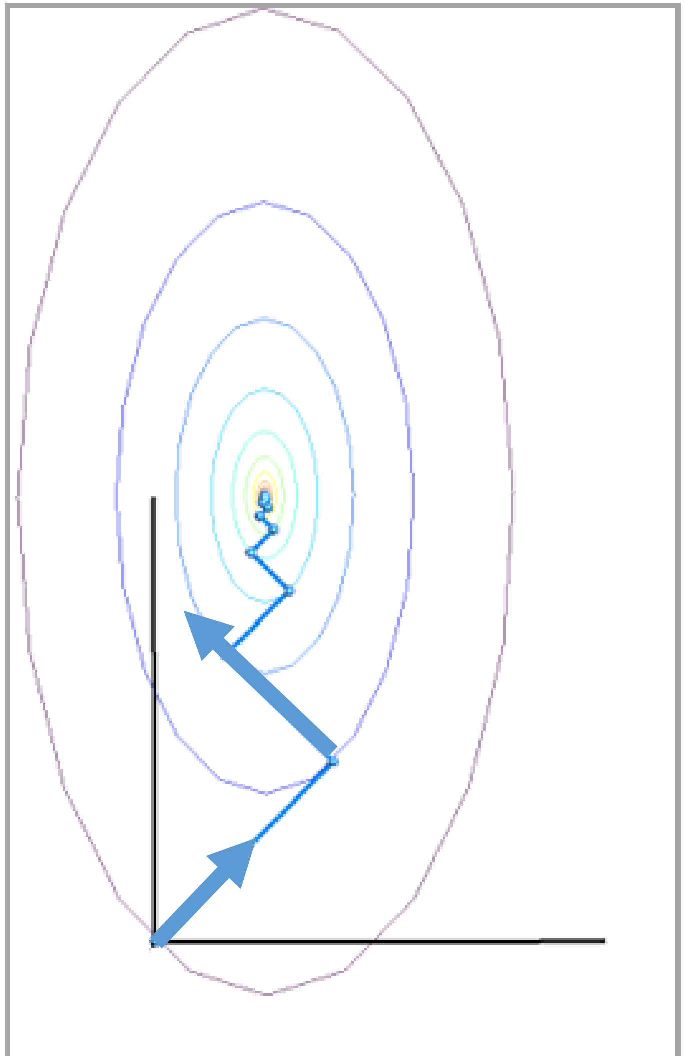
Descente de gradient: minimiser x^TAx-2b^Tx

- Initialiser $x^{(0)} = 0$
- On itère sur k:
  - On calcule la direction $d^{(k)} = r^{(k)} = -\nabla f(x^{(k)})/2 = b - Ax^{(k)}$
  - On calcule le pas de descente $\alpha^{(k)}$
  - On met à jour: $x^{(k+1)} = x^{(k)} + \alpha^{(k)} r^{(k)}$



Descente de gradient: minimiser x^TAx-2b^Tx

- Initialiser $x^{(0)} = 0$
- On itère sur k:
  - On calcule la direction $d^{(k)} = r^{(k)} = -\nabla f(x^{(k)})/2 = b-Ax^{(k)}$
  - On calcule le pas de descente $\alpha^{(k)}$
  - On met à jour: $x^{(k+1)} = x^{(k)} + \alpha^{(k)} r^{(k)}$

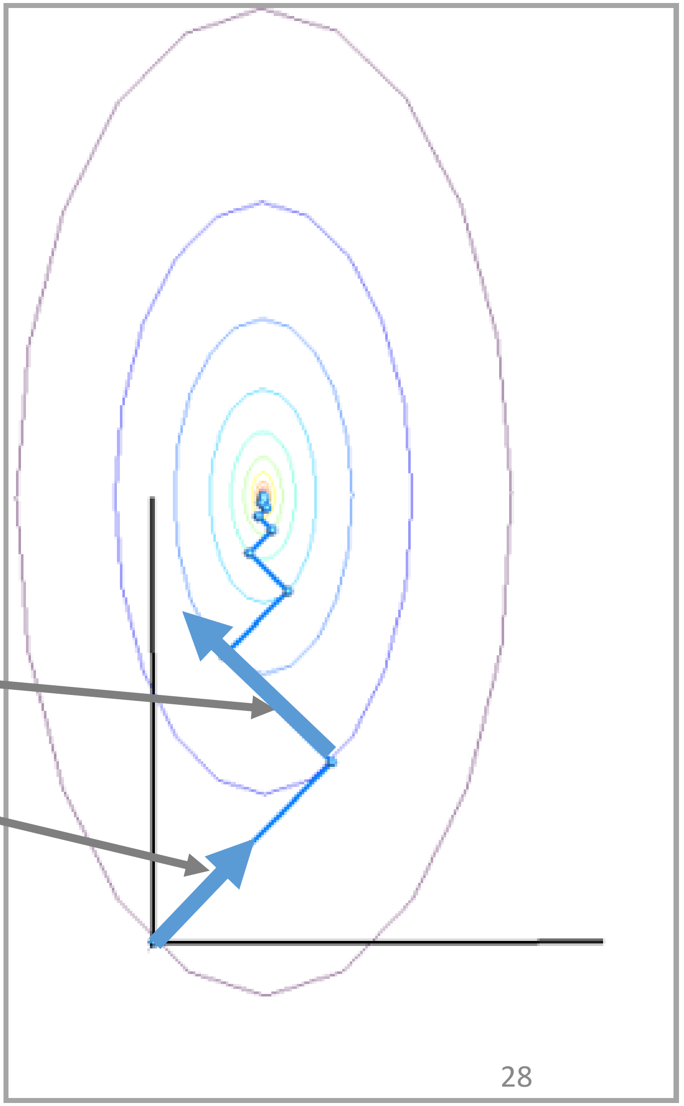


$\alpha^{(k)}$ est choisit pour que $r^{(k)}$ soit orthogonal à $\nabla f(x^{(k+1)})$

Descente de gradient: minimiser x^TAx-2b^Tx

- Initialiser $x^{(0)} = 0$
- On itère sur k:
  - On calcule la direction $d^{(k)} = r^{(k)} = -\nabla f(x^{(k)})/2 = b-Ax^{(k)}$
  - On calcule le pas de descente $\alpha^{(k)}$
  - On met à jour: $x^{(k+1)} = x^{(k)} + \alpha^{(k)}r^{(k)}$

$$\begin{aligned}
d^{(k)\top}r^{(k+1)} &= 0 \\
d^{(k)\top}(b - Ax^{(k+1)}) &= 0 \\
d^{(k)\top}(b - A(x^{(k)} + \alpha^{(k)}r^{(k)})) &= 0 \\
\alpha^{(k)} &= \frac{d^{(k)\top}(Ax^{(k)} - b)}{d^{(k)\top}Ar^{(k)}} \\
\alpha^{(k)} &= \frac{d^{(k)\top}r^{(k)}}{d^{(k)\top}Ar^{(k)}}
\end{aligned}$$

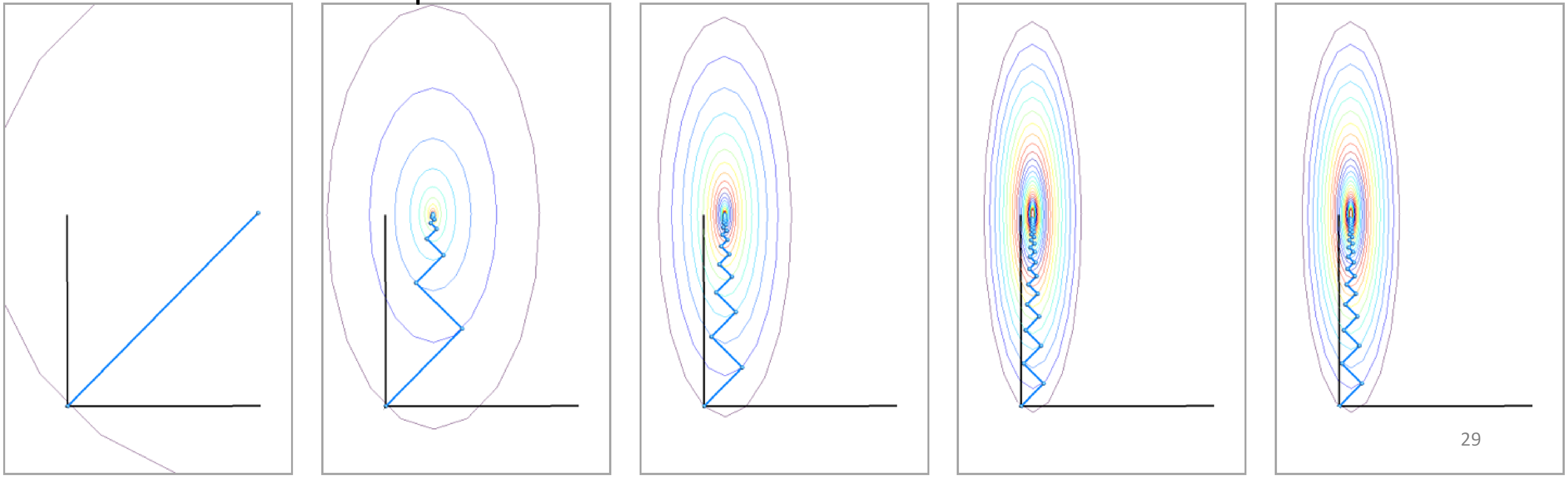


$\alpha^{(k)}$ est choisit pour que $r^{(k)}$ soit orthogonal à $\nabla f(x^{(k+1)})$

Descente de gradient: efficacité

La méthode est sensible au conditionnement de A (ratio entre valeurs propres extrêmes):

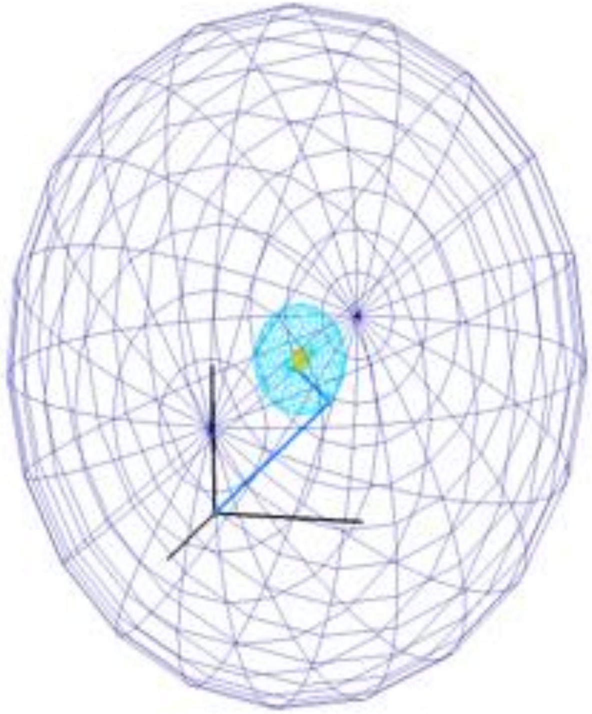
- Si f(x) est isotrope, la convergence est immédiate
- Sinon elle peut être très lente…

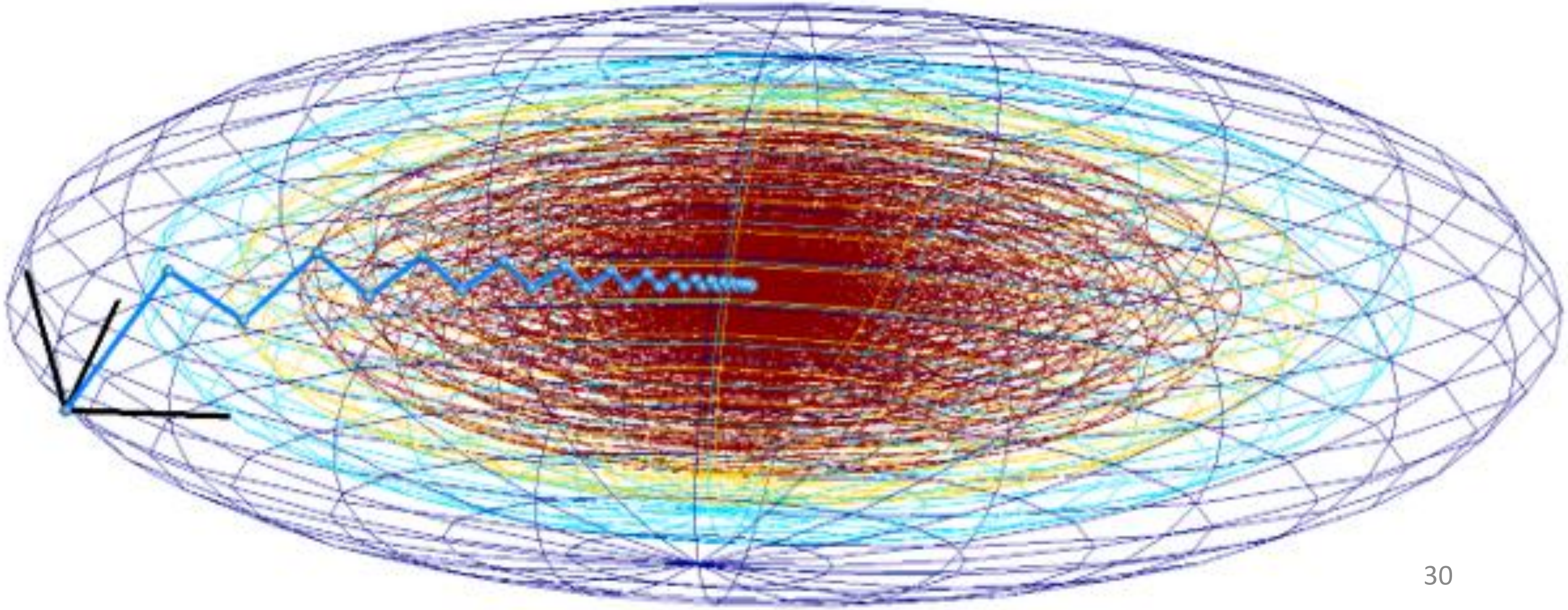


Descente de gradient: efficacité

La méthode est sensible au conditionnement de A (ratio entre valeurs propres extrêmes):

- Si f(x) est isotrope, la convergence est immédiate
- Sinon elle peut être très lente…



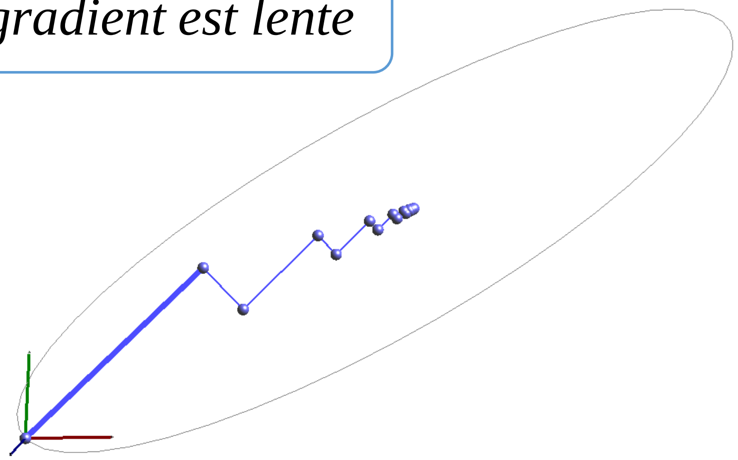


Plan

- *Introduction*
- *Le problème*
- *Résolution par gradient*
- **Résolution par conjugaison**
  - **Vue d’ensemble**
  - **Construction d’une base A-orthogonale**
- Gradient conjugué
- Contemplation de performance pour des problèmes discrétisés

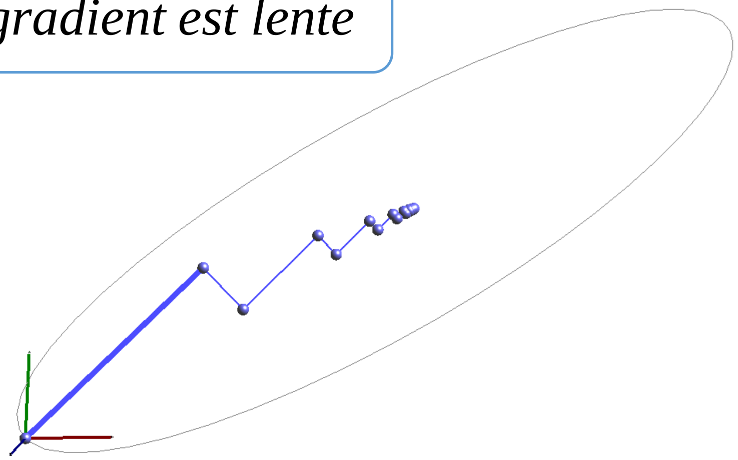
Conjugaison: idée de base

La descente de gradient est lente



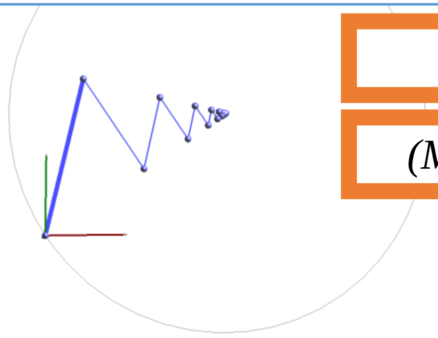
Conjugaison: idée de base

La descente de gradient est lente



M

La notion d'orthogonalité dans M est accessible

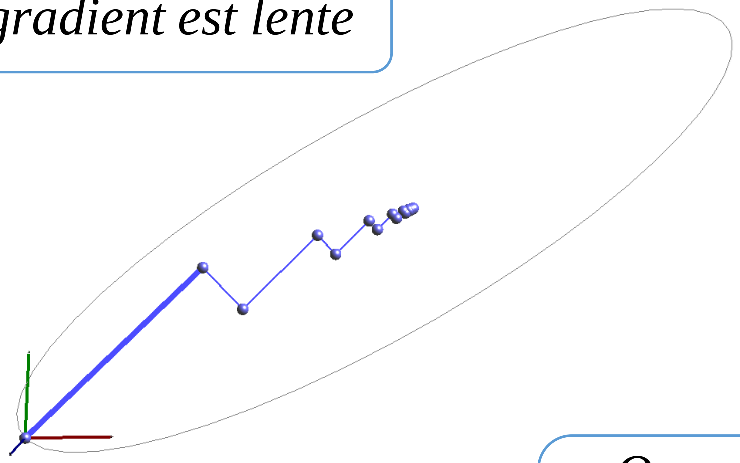


$$(Mu)^T(Mx^*)=u^TAx^*=u^Tb$$

$$(Mu)^T(Mv)=u^TM^TMv=u^TAv$$

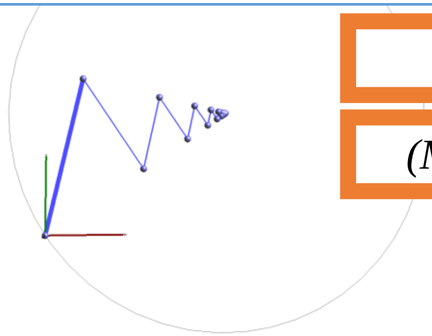
Conjugaison: idée de base

La descente de gradient est lente



M

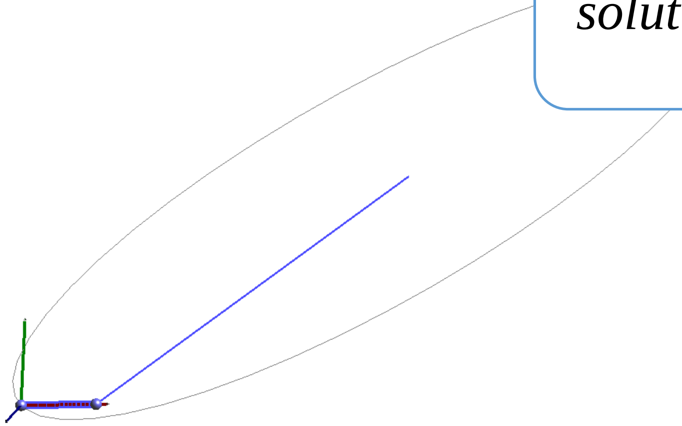
La notion d’orthogonalité dans M est accessible



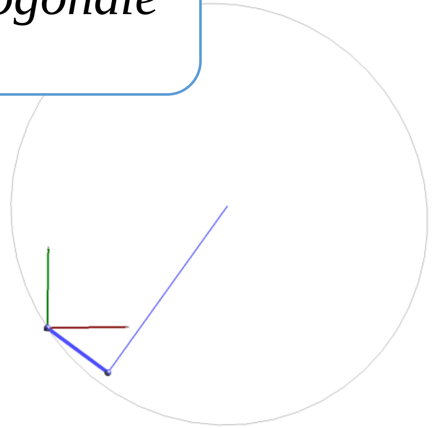
$$(Mu)^T(Mx^*)=u^TAx^*=u^Tb$$

$$(Mu)^T(Mv)=u^TM^TMv=u^TAv$$

On va essayer de décomposer la solution dans une base orthogonale dans M

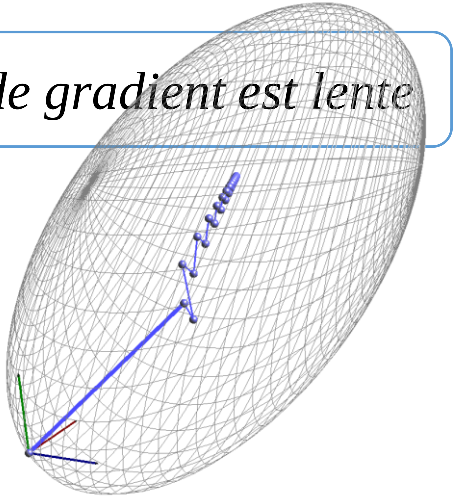


M



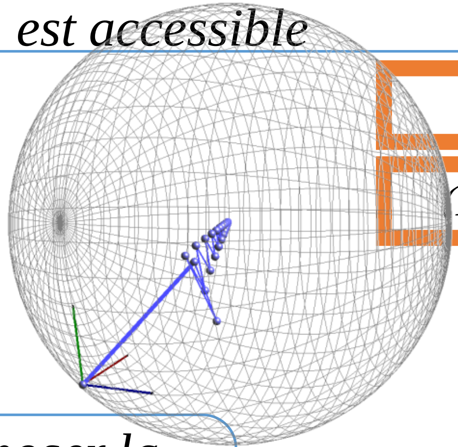
Conjugaison: idée de base

La descente de gradient est lente



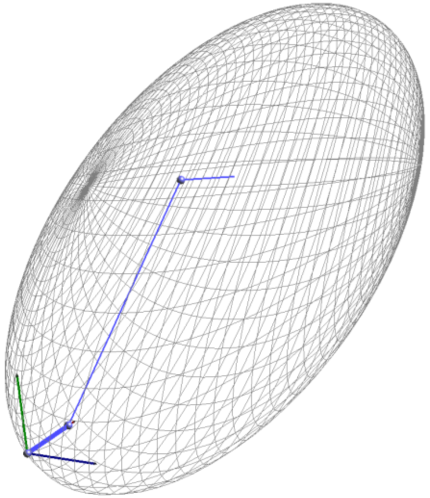
M

La notion d'orthogonalité dans M est accessible



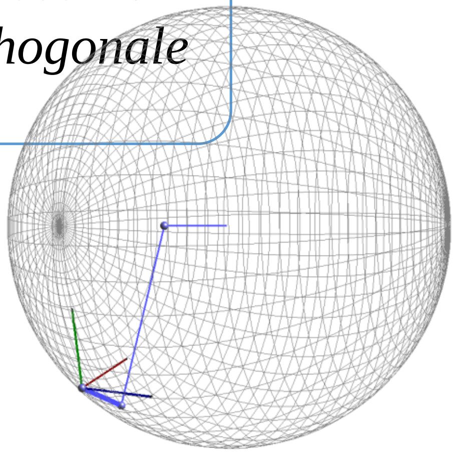
$(Mu)^T(Mx^*)=u^TAx^*=u^Tb$

$(Mu)^T(Mv)=u^TM^TMv=u^TAv$



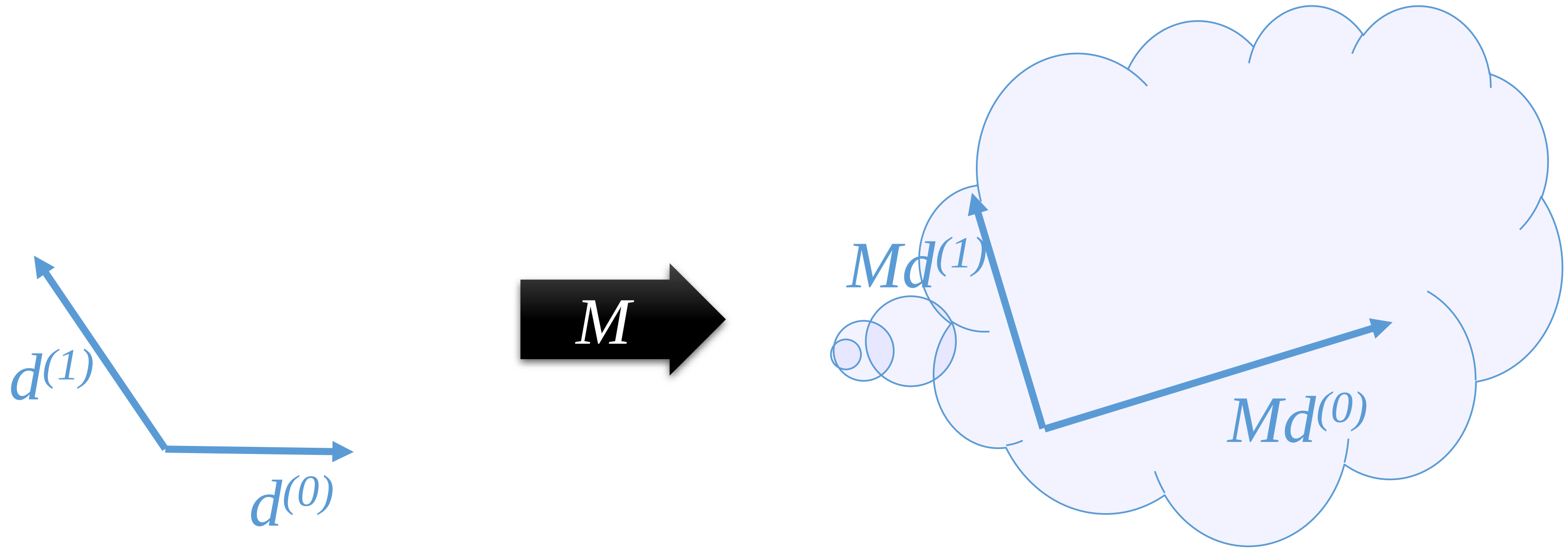
On va essayer de décomposer la solution dans une base orthogonale dans M

M



Conjugaison: vue d'ensemble

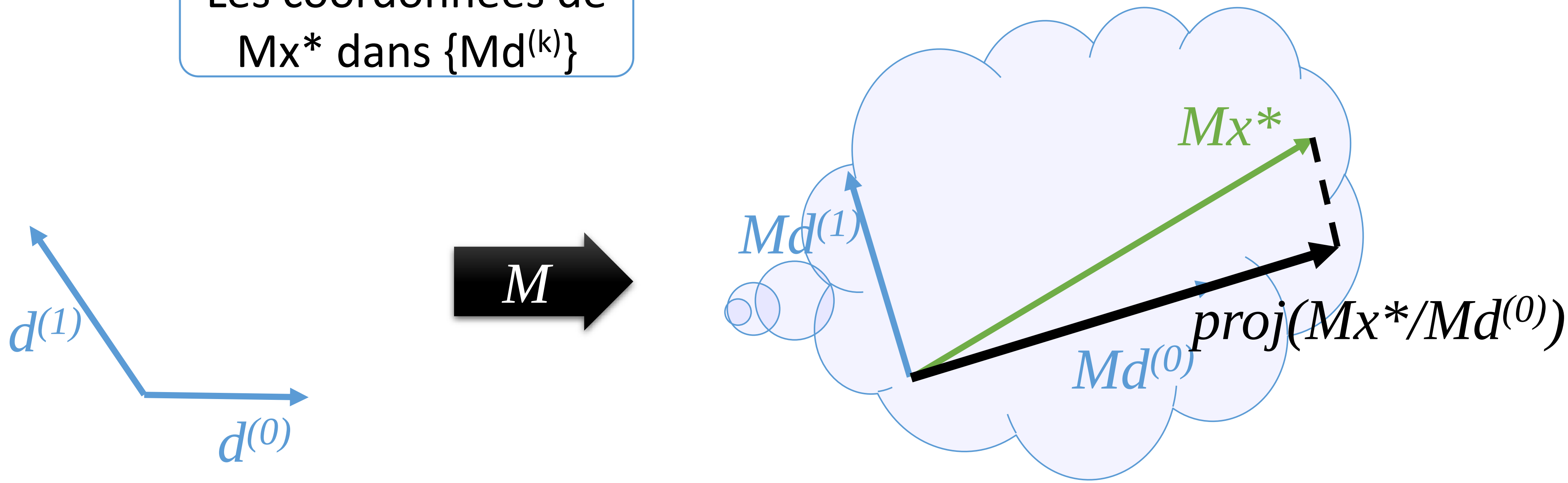
- Supposons que:
  - on connait $\{d^{(k)}\}$ une base telle que les $Md^{(k)}$ soient orthogonaux deux à deux



Conjugaison: vue d'ensemble

- Supposons que:
  - on connait $\{d^{(k)}\}$ une base telle que les $Md^{(k)}$ soient orthogonaux deux à deux
  - on sait calculer $\alpha^{(k)}$ tel que $proj(Mx^*/Md^{(k)})=\alpha^{(k)}Md^{(k)}$

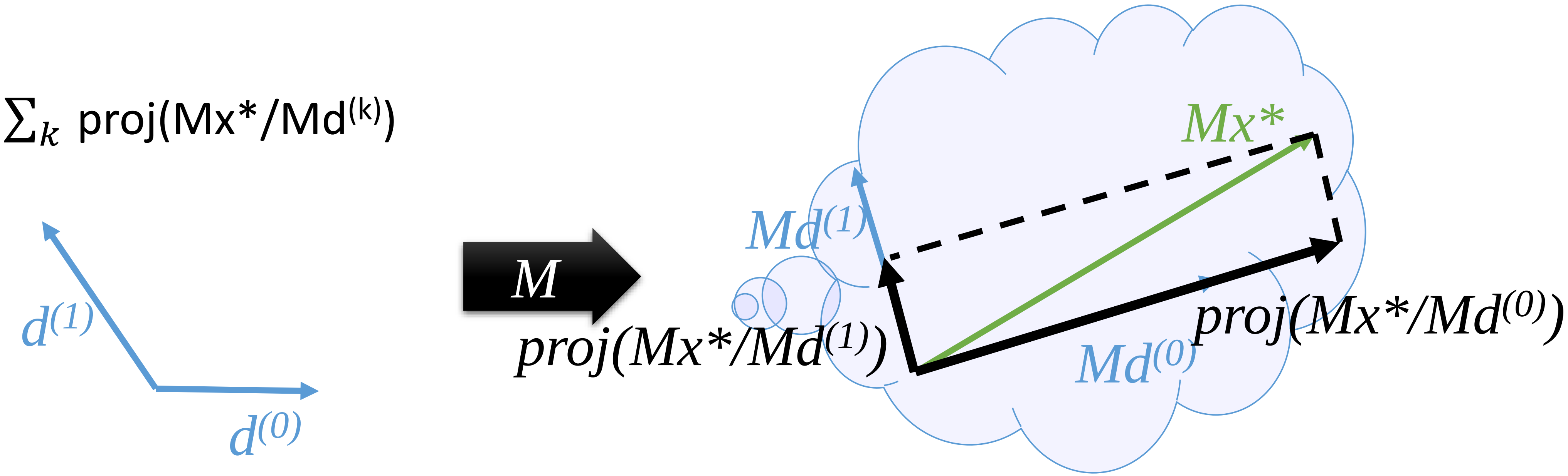
Les coordonnées de Mx^* dans $\{Md^{(k)}\}$



Conjugaison: vue d'ensemble

- Supposons que:
  - on connait $\{d^{(k)}\}$ une base telle que les $Md^{(k)}$ soient orthogonaux deux à deux
  - on sait calculer $\alpha^{(k)}$ tel que $proj(Mx^*/Md^{(k)})=\alpha^{(k)}Md^{(k)}$

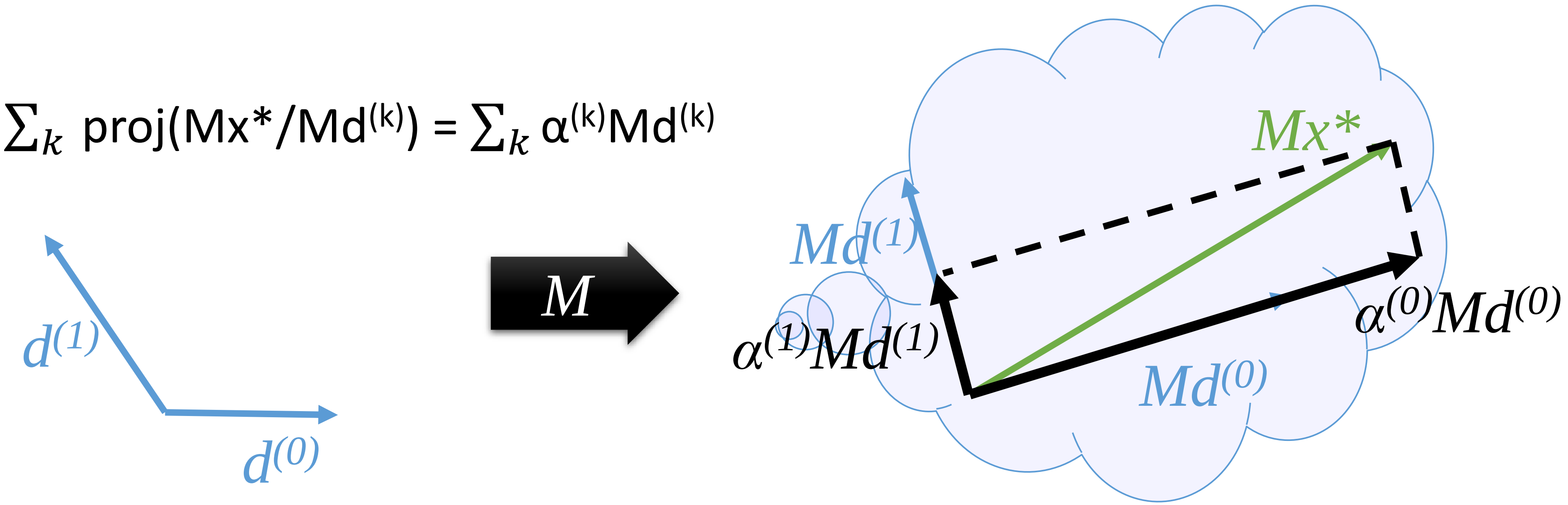
- Alors:
  - $Mx^* = \sum_k proj(Mx^*/Md^{(k)})$



Conjugaison: vue d'ensemble

- Supposons que:
  - on connait $\{d^{(k)}\}$ une base telle que les $Md^{(k)}$ soient orthogonaux deux à deux
  - on sait calculer $\alpha^{(k)}$ tel que $proj(Mx^*/Md^{(k)})=\alpha^{(k)}Md^{(k)}$

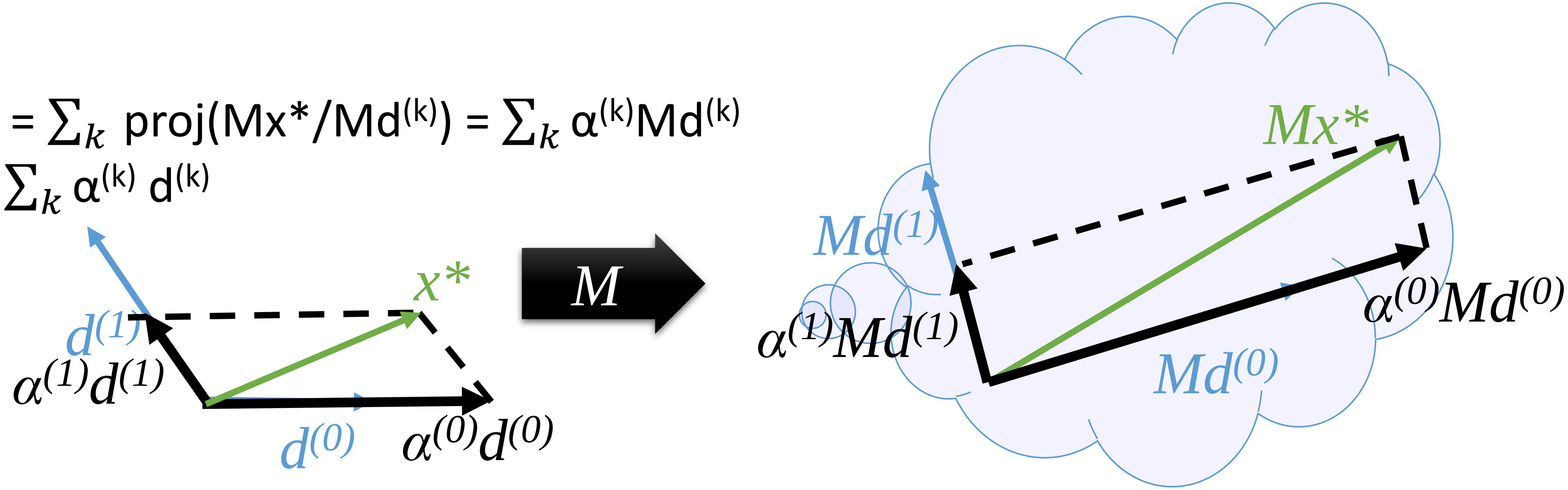
- Alors:
  - $Mx^* = \sum_k proj(Mx^*/Md^{(k)}) = \sum_k \alpha^{(k)}Md^{(k)}$



Conjugaison: vue d'ensemble

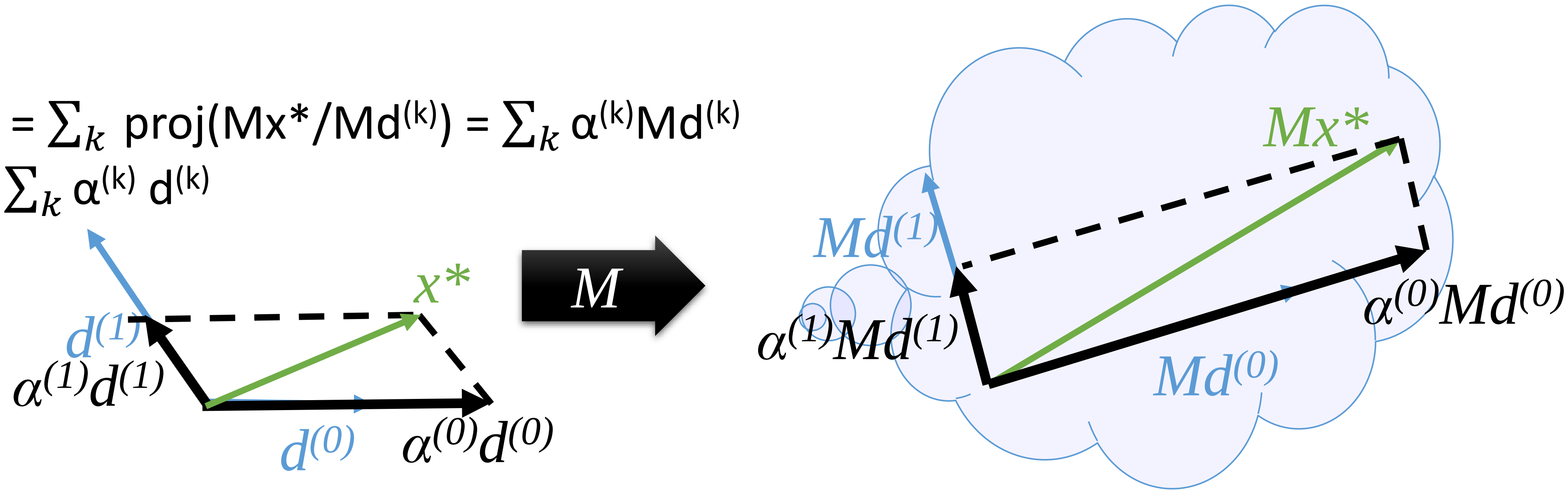
- Supposons que:
  - on connait $\{d^{(k)}\}$ une base telle que les $Md^{(k)}$ soient orthogonaux deux à deux
  - on sait calculer $\alpha^{(k)}$ tel que $\text{proj}(Mx^*/Md^{(k)})=\alpha^{(k)}Md^{(k)}$

- Alors:
  - $Mx^* = \sum_k \text{proj}(Mx^*/Md^{(k)}) = \sum_k \alpha^{(k)}Md^{(k)}$
  - $x^* = \sum_k \alpha^{(k)} d^{(k)}$



Conjugaison: vue d'ensemble

- Supposons que:
  - on connait $\{d^{(k)}\}$ une base telle que les $Md^{(k)}$ soient orthogonaux deux à deux
  - on sait calculer $\alpha^{(k)}$ tel que $proj(Mx^*/Md^{(k)})=\alpha^{(k)}Md^{(k)}$
- Alors:
  - $Mx^* = \sum_k proj(Mx^*/Md^{(k)}) = \sum_k \alpha^{(k)}Md^{(k)}$
  - $x^* = \sum_k \alpha^{(k)} d^{(k)}$

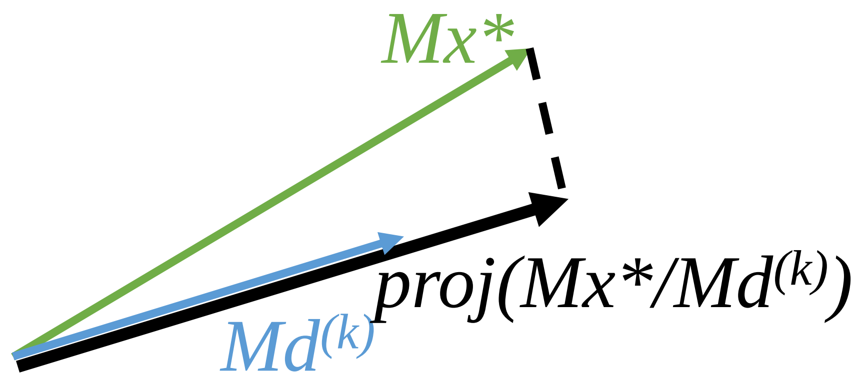


Conjugaison: vue d’ensemble

- Supposons que:
  - on connait $\{d^{(k)}\}$ une base telle que les $Md^{(k)}$ soient orthogonaux deux à deux
  - on sait calculer $\alpha^{(k)}$ tel que $proj(Mx^*/Md^{(k)})=\alpha^{(k)}Md^{(k)}$

Conjugaison: vue d'ensemble

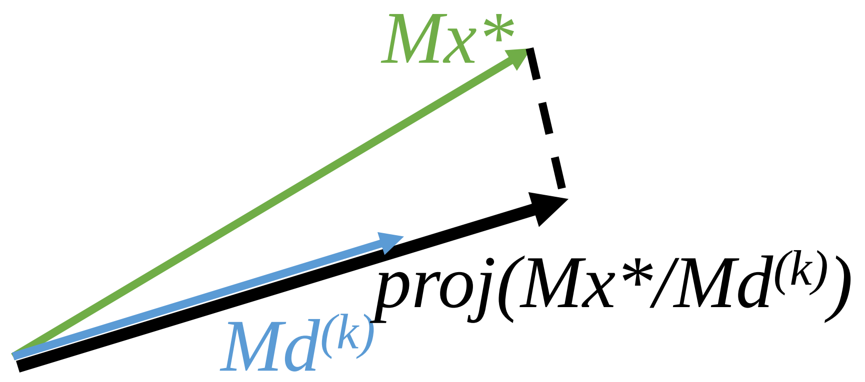
- Supposons que:
  - on connait $\{d^{(k)}\}$ une base telle que les $Md^{(k)}$ soient orthogonaux deux à deux
  - on sait calculer $\alpha^{(k)}$ tel que $proj(Mx^*/Md^{(k)})=\alpha^{(k)}Md^{(k)}$



Conjugaison: vue d'ensemble

- Supposons que:
  - on connait $\{d^{(k)}\}$ une base telle que les $Md^{(k)}$ soient orthogonaux deux à deux
  - on sait calculer $\alpha^{(k)}$ tel que $proj(Mx^*/Md^{(k)})=\alpha^{(k)}Md^{(k)}$

$$\begin{cases} proj(Mx^*/Md^{(k)})^T Md^{(k)} & = (Mx^*)^T Md^{(k)} \\ proj(Mx^*/Md^{(k)}) & = \alpha^{(k)} Md^{(k)} \end{cases}$$

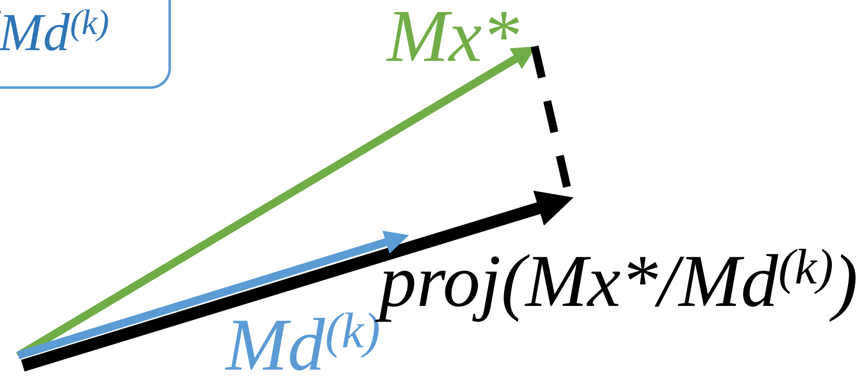


Conjugaison: vue d'ensemble

- Supposons que:
  - on connait $\{d^{(k)}\}$ une base telle que les $Md^{(k)}$ soient orthogonaux deux à deux
  - on sait calculer $\alpha^{(k)}$ tel que $proj(Mx^*/Md^{(k)})=\alpha^{(k)}Md^{(k)}$

$$\begin{cases} proj(Mx^*/Md^{(k)})^T Md^{(k)} & = (Mx^*)^T Md^{(k)} \\ proj(Mx^*/Md^{(k)}) & = \alpha^{(k)} Md^{(k)} \end{cases}$$

$$(\alpha^{(k)} Md^{(k)})^T Md^{(k)} = (Mx^*)^T Md^{(k)}$$



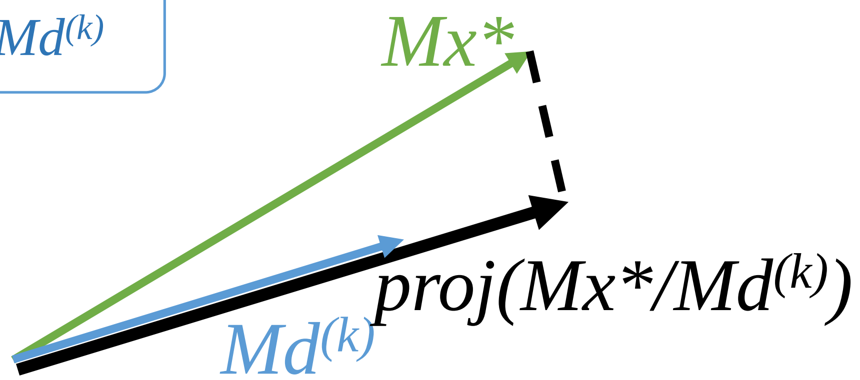
Conjugaison: vue d'ensemble

- Supposons que:
  - on connait $\{d^{(k)}\}$ une base telle que les $Md^{(k)}$ soient orthogonaux deux à deux
  - on sait calculer $\alpha^{(k)}$ tel que $proj(Mx^*/Md^{(k)})=\alpha^{(k)}Md^{(k)}$

$$\begin{cases} proj(Mx^*/Md^{(k)})^T Md^{(k)} & = (Mx^*)^T Md^{(k)} \\ proj(Mx^*/Md^{(k)}) & = \alpha^{(k)} Md^{(k)} \end{cases}$$

$$(\alpha^{(k)} Md^{(k)})^T Md^{(k)} = (Mx^*)^T Md^{(k)}$$

$$\alpha^{(k)} = \frac{(Mx^*)^T Md^{(k)}}{(Md^{(k)})^T Md^{(k)}}$$



Conjugaison: vue d'ensemble

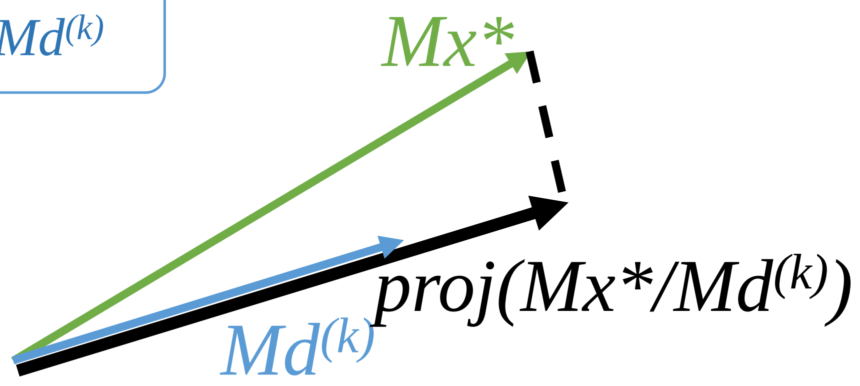
- Supposons que:
  - on connait $\{d^{(k)}\}$ une base telle que les $Md^{(k)}$ soient orthogonaux deux à deux
  - on sait calculer $\alpha^{(k)}$ tel que $proj(Mx^*/Md^{(k)})=\alpha^{(k)}Md^{(k)}$

$$\alpha^{(k)} = \frac{(Mx*)^T Md^{(k)}}{(Md^{(k)})^T Md^{(k)}}$$

$$\begin{cases} proj(Mx^*/Md^{(k)})^T Md^{(k)} & = (Mx^*)^T Md^{(k)} \\ proj(Mx^*/Md^{(k)}) & = \alpha^{(k)} Md^{(k)} \end{cases}$$

$$(\alpha^{(k)} Md^{(k)})^T Md^{(k)} = (Mx^*)^T Md^{(k)}$$

$$\alpha^{(k)} = \frac{(Mx*)^T Md^{(k)}}{(Md^{(k)})^T Md^{(k)}}$$



Conjugaison: vue d'ensemble

- Supposons que:
  - on connait $\{d^{(k)}\}$ une base telle que les $Md^{(k)}$ soient orthogonaux deux à deux
  - on sait calculer $\alpha^{(k)}$ tel que $proj(Mx^*/Md^{(k)})=\alpha^{(k)}Md^{(k)}$

$$\alpha^{(k)} = \frac{(Mx*)^T Md^{(k)}}{(Md^{(k)})^T Md^{(k)}}$$

$$\alpha^{(k)} = \frac{d^{(k)}b}{d^{(k)}Ad^{(k)}}$$

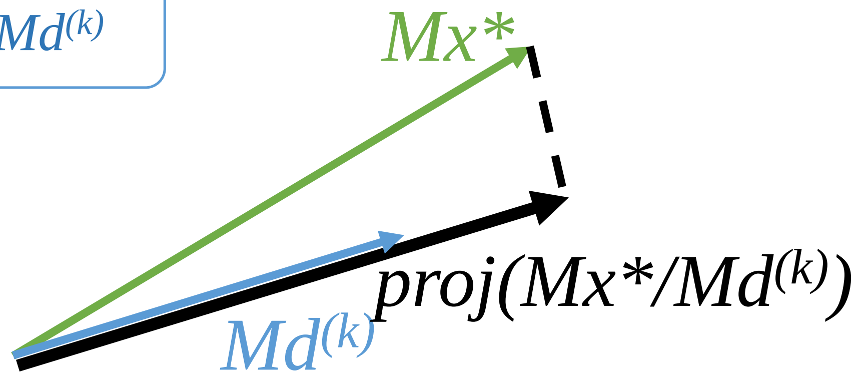
$$\begin{cases} proj(Mx^*/Md^{(k)})^T Md^{(k)} & = (Mx^*)^T Md^{(k)} \\ proj(Mx^*/Md^{(k)}) & = \alpha^{(k)} Md^{(k)} \end{cases}$$

$$(Mu)^T(Mx^*)=u^TAx^*=u^Tb$$

$$(Mu)^T(Mv)=u^TM^TMv=u^TAv$$

$$(\alpha^{(k)}Md^{(k)})^T Md^{(k)} = (Mx^*)^T Md^{(k)}$$

$$\alpha^{(k)} = \frac{(Mx*)^T Md^{(k)}}{(Md^{(k)})^T Md^{(k)}}$$



Conjugaison: vue d'ensemble

- Supposons que:
  - on connait $\{d^{(k)}\}$ une base telle que les $Md^{(k)}$ soient orthogonaux deux à deux
  - on sait calculer $\alpha^{(k)}$ tel que $proj(Mx^*/Md^{(k)})=\alpha^{(k)}Md^{(k)}$

Conjugaison: définir une base A-orthogonale

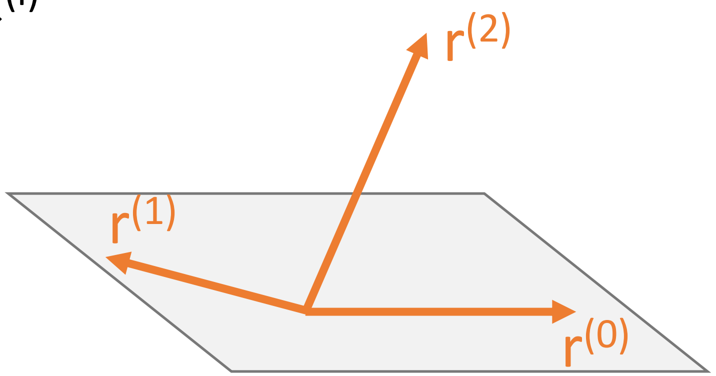
- Pour construite une base orthogonale, on peut utiliser l'algorithme de Gram Schmidt:
  - On choisit n vecteurs indépendants $\{r^{(k)}\}$
  - Pour chaque vecteur $r^{(k)}$
    - On lui retire sa projection sur les vecteurs précédents

$$\mathrm{d}^{(k)} = r^{(k)} - \sum_{i=0}^{k-1} \frac{r^{(k)}.\, d^{(i)}}{d^{(i)}.\, d^{(i)}} d^{(i)}$$

Conjugaison: définir une base A-orthogonale

- Pour construite une base orthogonale, on peut utiliser l'algorithme de Gram Schmidt:
  - On choisit n vecteurs indépendants $\{r^{(k)}\}$
  - Pour chaque vecteur $r^{(k)}$
    - On lui retire sa projection sur les vecteurs précédents

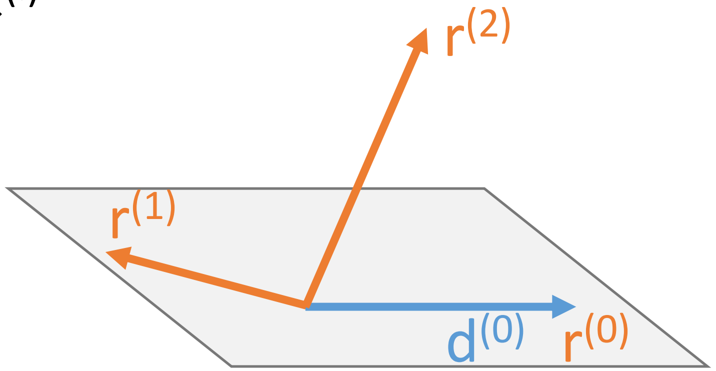
$$\mathrm{d}^{(k)} = r^{(k)} - \sum_{i=0}^{k-1} \frac{r^{(k)}.\,d^{(i)}}{d^{(i)}.\,d^{(i)}} d^{(i)}$$



Conjugaison: définir une base A-orthogonale

- Pour construite une base orthogonale, on peut utiliser l'algorithme de Gram Schmidt:
  - On choisit n vecteurs indépendants $\{r^{(k)}\}$
  - Pour chaque vecteur $r^{(k)}$
    - On lui retire sa projection sur les vecteurs précédents

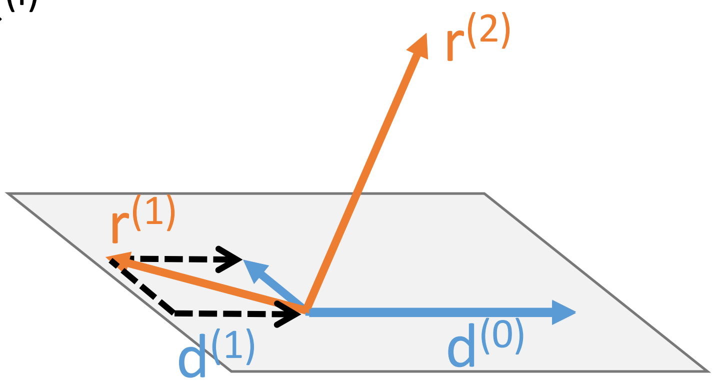
$$\mathrm{d}^{(k)} = r^{(k)} - \sum_{i=0}^{k-1} \frac{r^{(k)}.d^{(i)}}{d^{(i)}.d^{(i)}} d^{(i)}$$



Conjugaison: définir une base A-orthogonale

- Pour construite une base orthogonale, on peut utiliser l'algorithme de Gram Schmidt:
  - On choisit n vecteurs indépendants $\{r^{(k)}\}$
  - Pour chaque vecteur $r^{(k)}$
    - On lui retire sa projection sur les vecteurs précédents

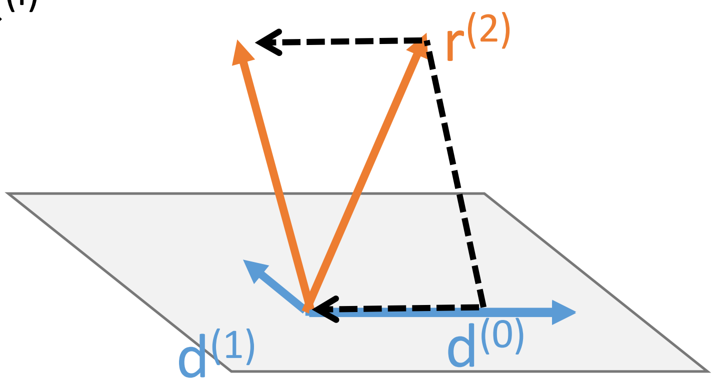
$$d^{(k)} = r^{(k)} - \sum_{i=0}^{k-1} \frac{r^{(k)} . d^{(i)}}{d^{(i)} . d^{(i)}} d^{(i)}$$



Conjugaison: définir une base A-orthogonale

- Pour construite une base orthogonale, on peut utiliser l'algorithme de Gram Schmidt:
  - On choisit n vecteurs indépendants $\{r^{(k)}\}$
  - Pour chaque vecteur $r^{(k)}$
    - On lui retire sa projection sur les vecteurs précédents

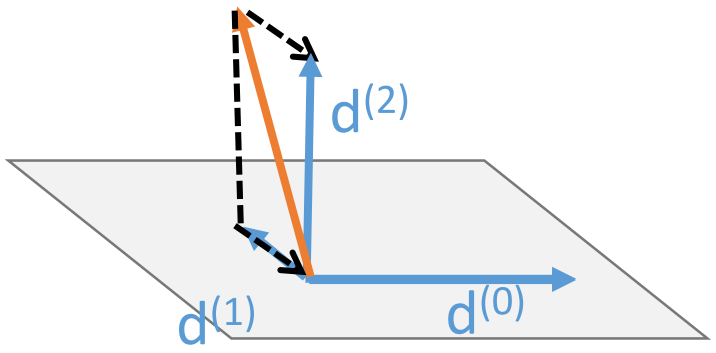
$$\mathrm{d}^{(k)} = r^{(k)} - \sum_{i=0}^{k-1} \frac{r^{(k)} . d^{(i)}}{d^{(i)} . d^{(i)}} d^{(i)}$$



Conjugaison: définir une base A-orthogonale

- Pour construite une base orthogonale, on peut utiliser l'algorithme de Gram Schmidt:
  - On choisit n vecteurs indépendants $\{r^{(k)}\}$
  - Pour chaque vecteur $r^{(k)}$
    - On lui retire sa projection sur les vecteurs précédents

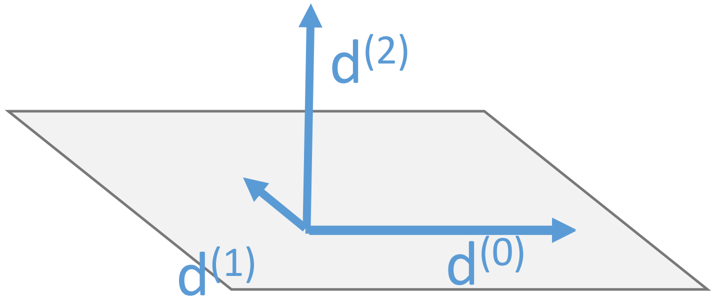
$$d^{(k)} = r^{(k)} - \sum_{i=0}^{k-1} \frac{r^{(k)}.\, d^{(i)}}{d^{(i)}.\, d^{(i)}} d^{(i)}$$



Conjugaison: définir une base A-orthogonale

- Pour construite une base orthogonale, on peut utiliser l'algorithme de Gram Schmidt:
  - On choisit n vecteurs indépendants $\{r^{(k)}\}$
  - Pour chaque vecteur $r^{(k)}$
    - On lui retire sa projection sur les vecteurs précédents

$$\mathrm{d}^{(k)} = r^{(k)} - \sum_{i=0}^{k-1} \frac{r^{(k)}.\, d^{(i)}}{d^{(i)}.\, d^{(i)}} d^{(i)}$$



Conjugaison: définir une base A-orthogonale

- Pour construite une base orthogonale, on peut utiliser l'algorithme de Gram Schmidt:
  - On choisit n vecteurs indépendants $\{r^{(k)}\}$
  - Pour chaque vecteur $r^{(k)}$
    - On lui retire sa projection sur les vecteurs précédents

$$\mathrm{d}^{(k)} = r^{(k)} - \sum_{i=0}^{k-1} \frac{r^{(k)} . d^{(i)}}{d^{(i)} . d^{(i)}} d^{(i)}$$

- Cet algorithme n'utilise que le produit scalaire…
- si on utilisait le produit scalaire des images par M…
- on aurait une base dont l'image par M est orthogonale !

Conjugaison: définir une base A-orthogonale

- Pour construite une base orthogonale, on peut utiliser l'algorithme de Gram Schmidt:
  - On choisit n vecteurs indépendants $\{r^{(k)}\}$
  - Pour chaque vecteur $r^{(k)}$
    - On lui retire sa projection sur les vecteurs précédents

$$\mathrm{d}^{(k)} = r^{(k)} - \sum_{i=0}^{k-1} \frac{Mr^{(k)}. Md^{(i)}}{Md^{(i)}. Md^{(i)}} d^{(i)}$$

- Cet algorithme n'utilise que le produit scalaire…
- si on utilisait le produit scalaire des images par M…
- on aurait une base dont l'image par M est orthogonale !

Conjugaison: définir une base A-orthogonale

- Pour construite une base orthogonale, on peut utiliser l'algorithme de Gram Schmidt:
  - On choisit n vecteurs indépendants $\{r^{(k)}\}$
  - Pour chaque vecteur $r^{(k)}$
    - On lui retire sa projection sur les vecteurs précédents

$$\mathrm{d}^{(k)} = r^{(k)} - \sum_{i=0}^{k-1} \frac{r^{(k)} A d^{(i)}}{d^{(i)} A d^{(i)}} d^{(i)}$$

- Cet algorithme n'utilise que le produit scalaire…
- si on utilisait le produit scalaire des images par M…
- on aurait une base dont l'image par M est orthogonale !

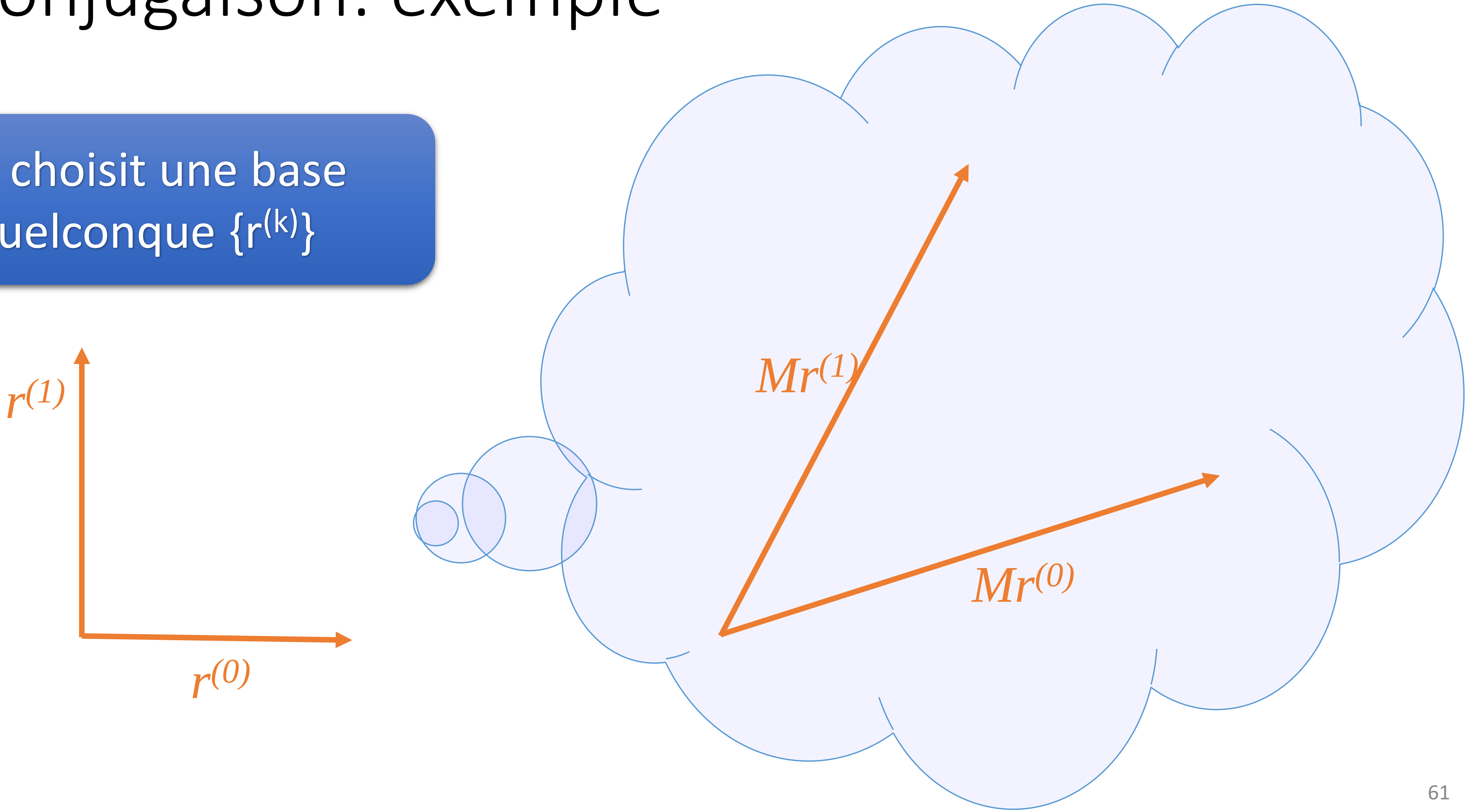
Conjugaison: vue d’ensemble

- On sait:
  - Calculer $\{d^{(k)}\}$ une base telle que les $Md^{(k)}$ soient orthogonaux deux à deux
  - Calculer $\alpha^{(k)}$ tel que $proj(Mx^*/Md^{(k)})=\alpha^{(k)}Md^{(k)}$

- Donc la solution est:
  - $x^* = \sum_k \alpha^{(k)} d^{(k)}$

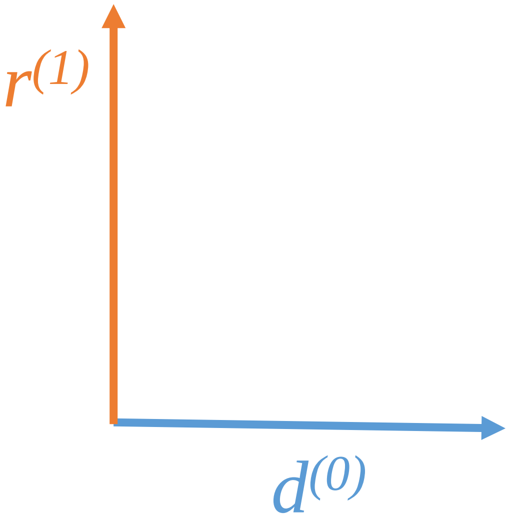
Conjugaison: exemple

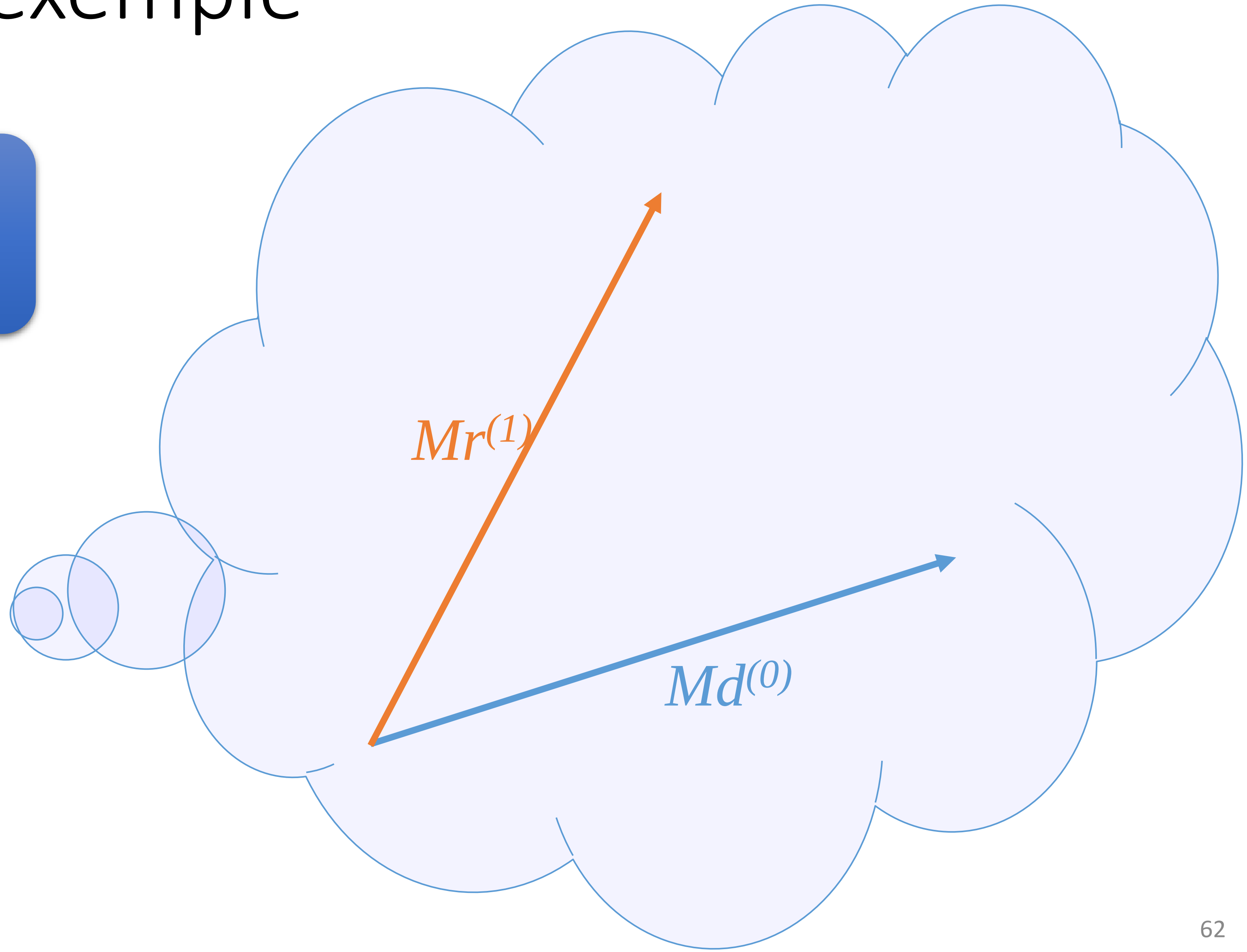
On choisit une base quelconque $\{r^{(k)}\}$



Conjugaison: exemple

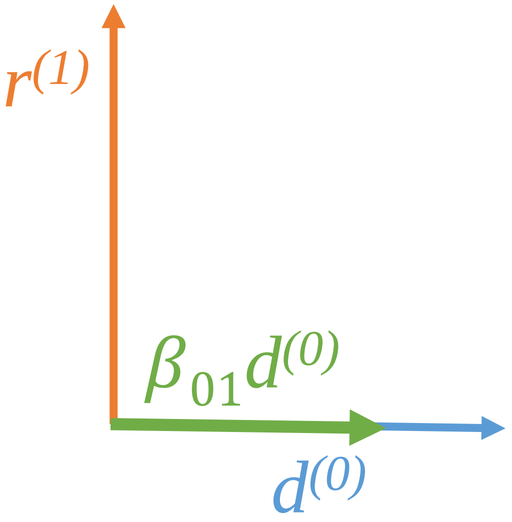
On fixe $d^{(0)} = r^{(0)}$

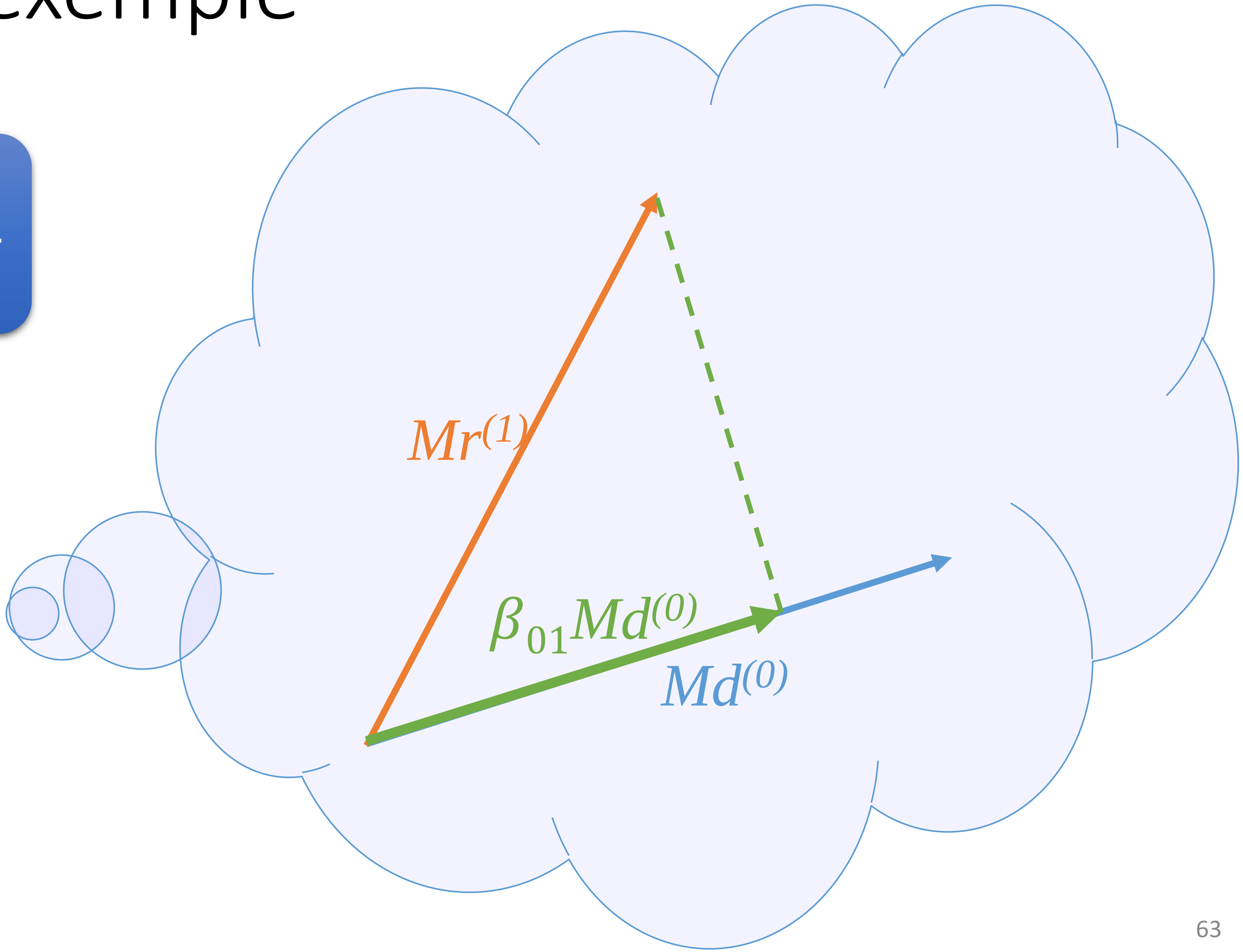




Conjugaison: exemple

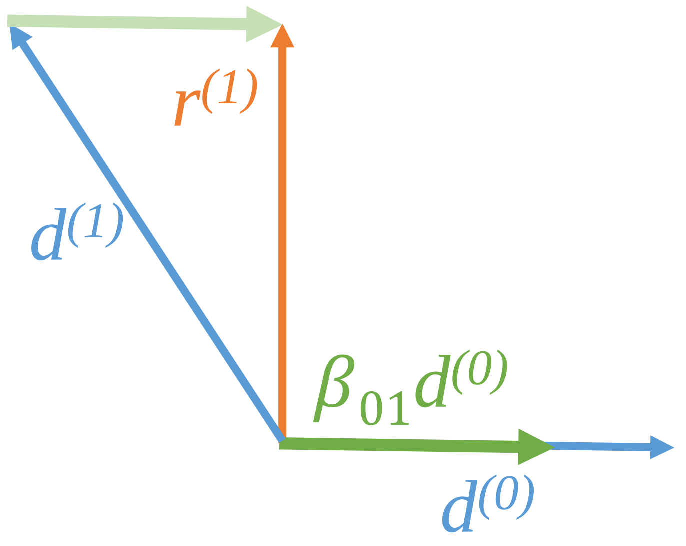
On calcule $\beta_{01} = \dfrac{r^{(0)}Ad^{(1)}}{d^{(0)}Ad^{(0)}}$

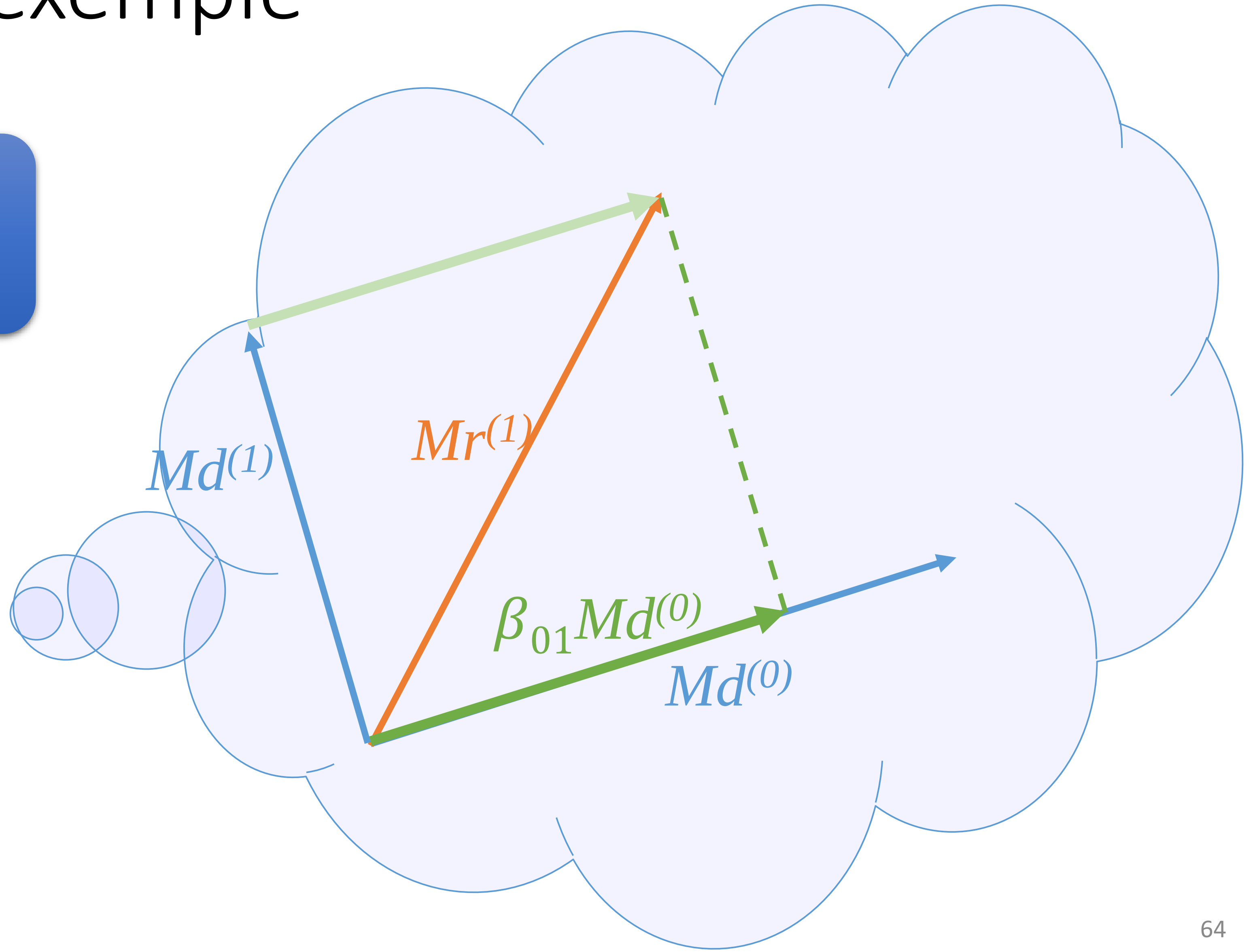




Conjugaison: exemple

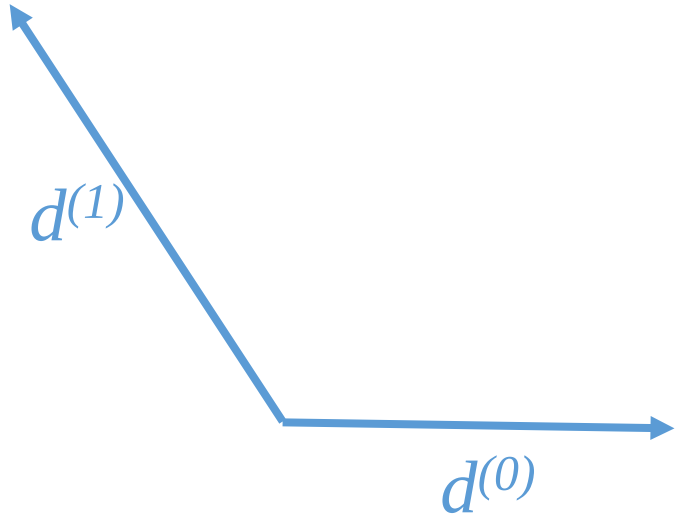
On choisit
$d^{(1)} = r^{(1)} - \beta_{01} d^{(0)}$

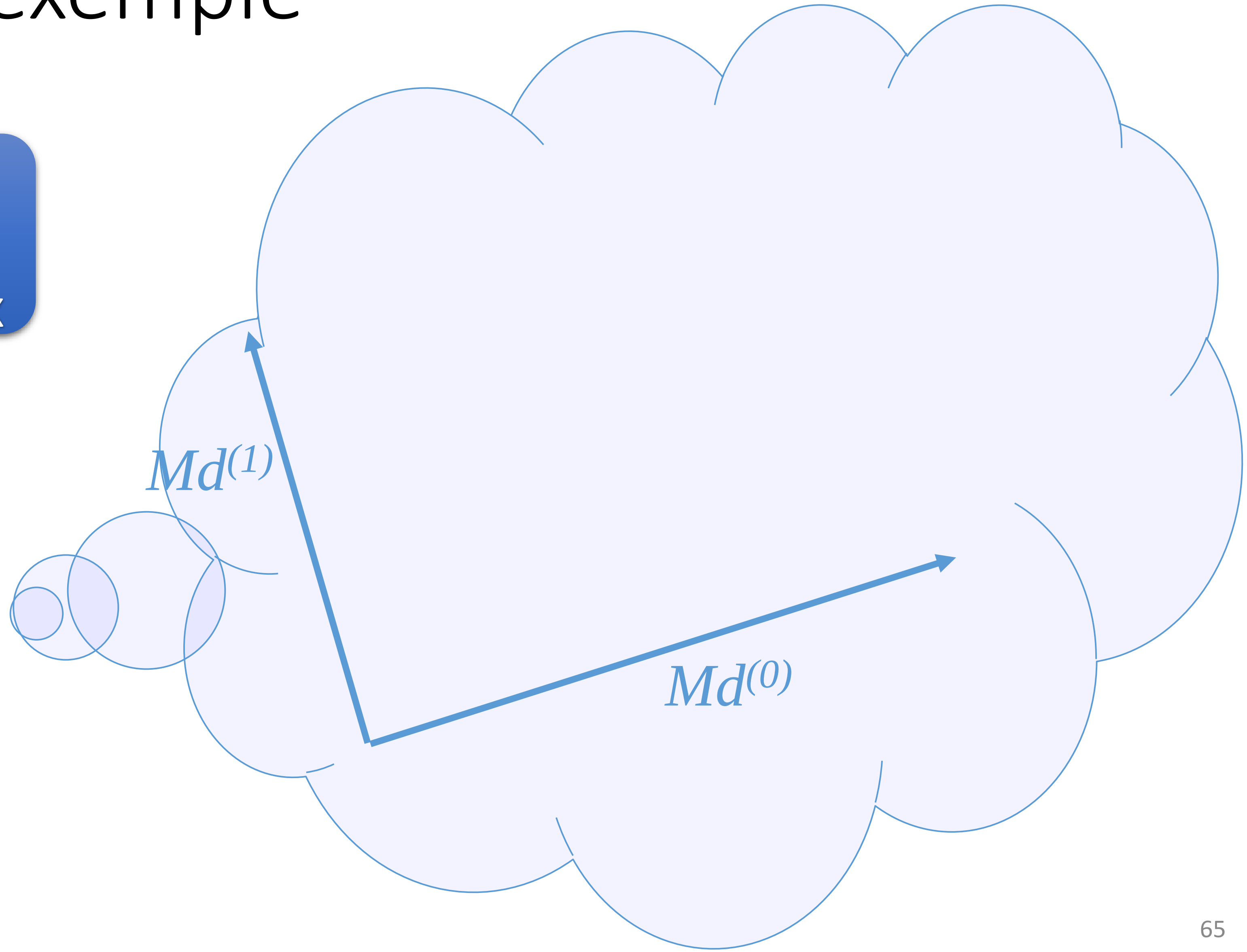




Conjugaison: exemple

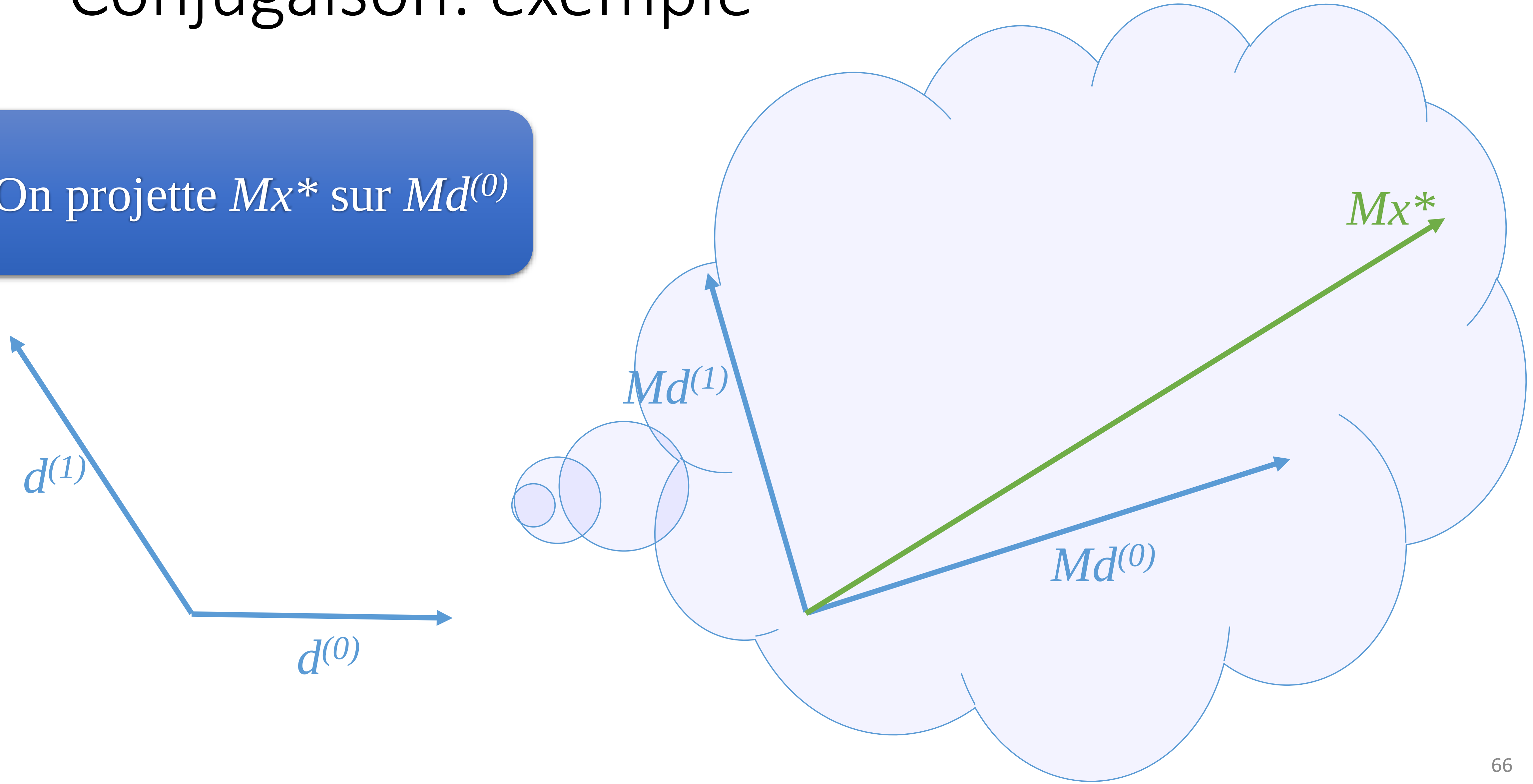
On a une base $\{d^{(k)}\}$ telle que les $Md^{(k)}$ soient orthogonaux deux à deux





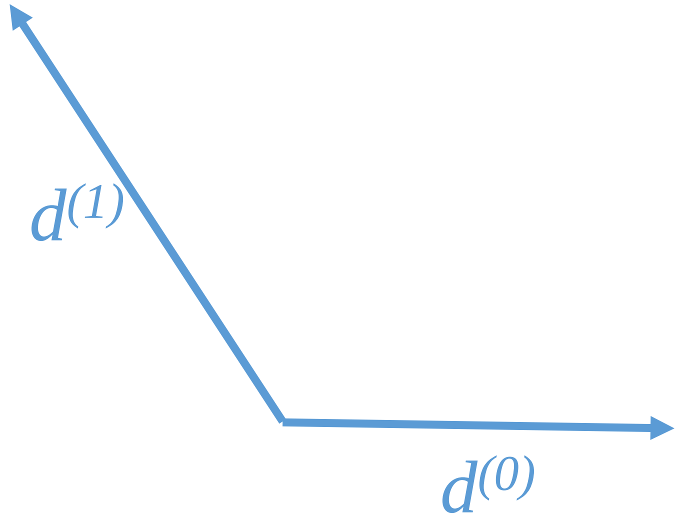
Conjugaison: exemple

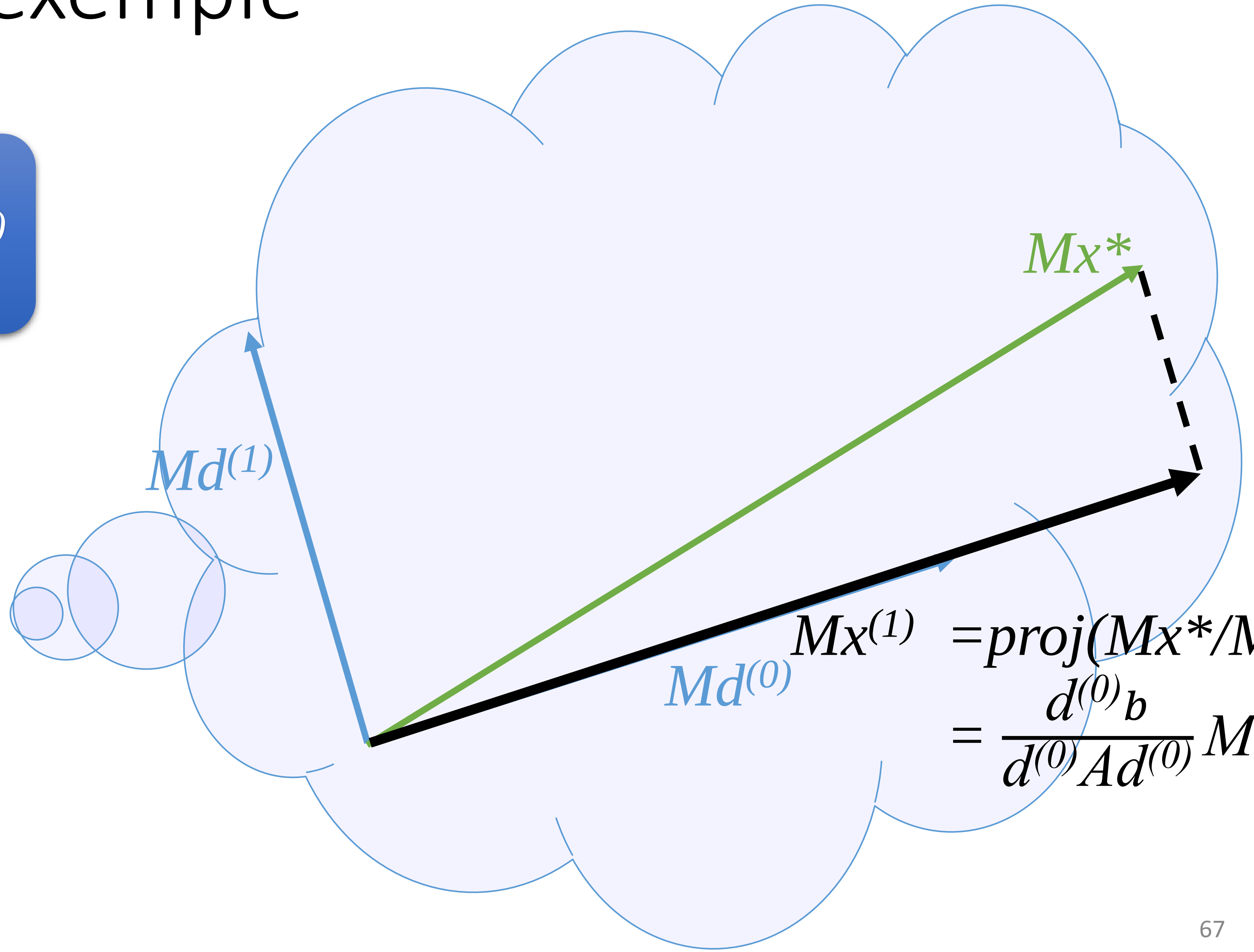
On projette Mx^* sur $Md^{(0)}$



Conjugaison: exemple

On projette Mx^* sur $Md^{(0)}$

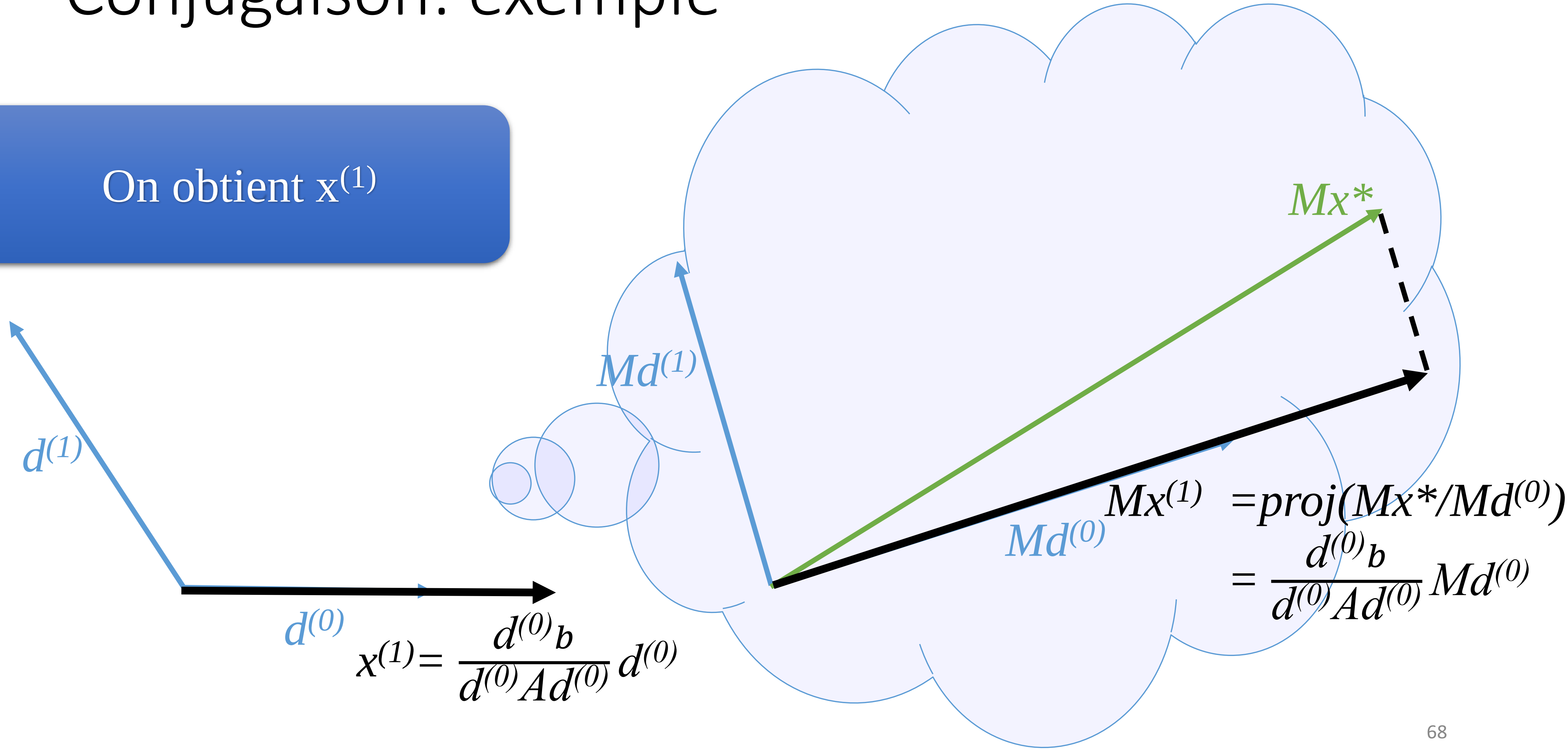




$$Mx^{(1)} = proj(Mx^*/Md^{(0)}) = \frac{d^{(0)}b}{d^{(0)}Ad^{(0)}} Md^{(0)}$$

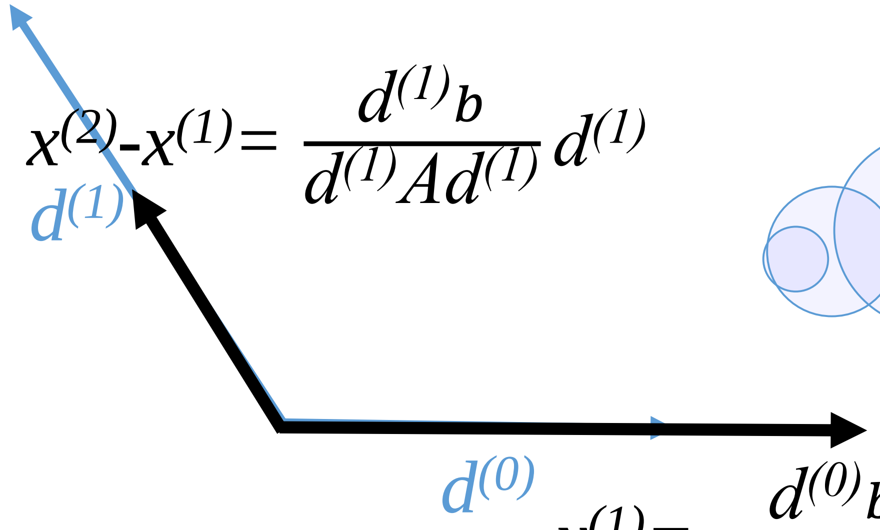
Conjugaison: exemple

On obtient $x^{(1)}$



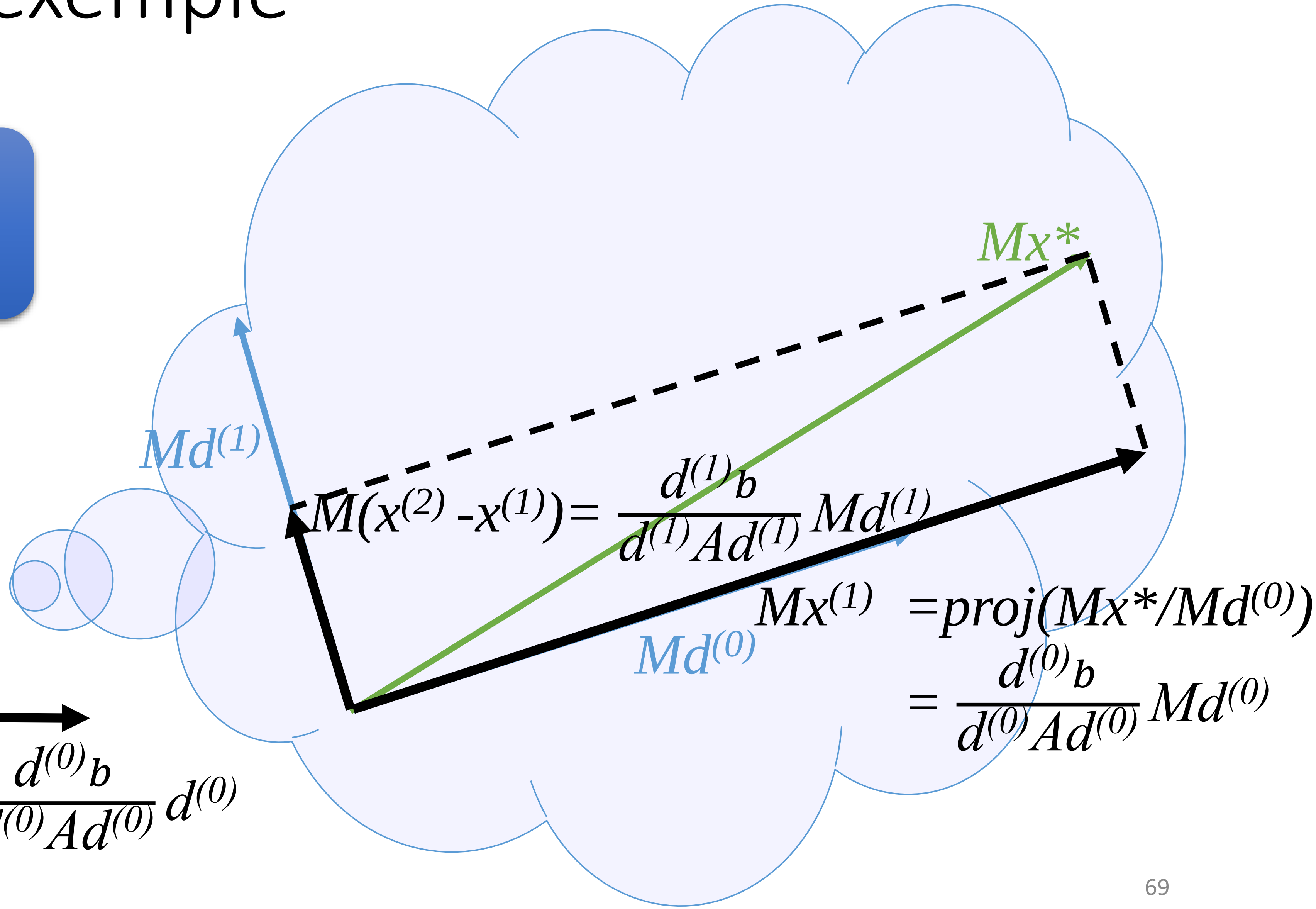
Conjugaison: exemple

On obtient $x^{(2)}-x^{(1)}$



$$x^{(2)}-x^{(1)} = \frac{d^{(1)}b}{d^{(1)}Ad^{(1)}}\, d^{(1)}$$

$$x^{(1)} = \frac{d^{(0)}b}{d^{(0)}Ad^{(0)}}\, d^{(0)}$$

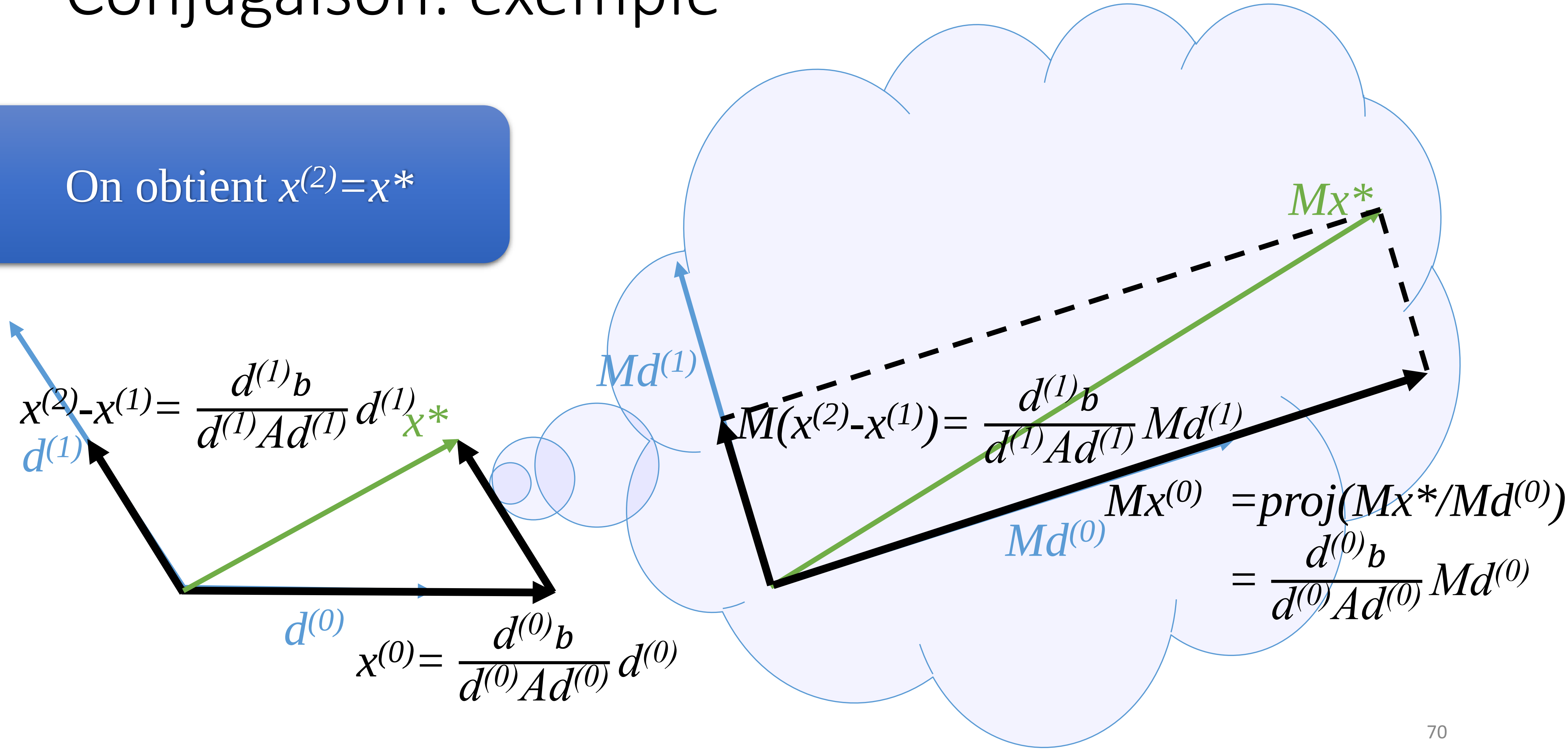


$$M(x^{(2)}-x^{(1)}) = \frac{d^{(1)}b}{d^{(1)}Ad^{(1)}}\, Md^{(1)}$$

$$Mx^{(1)} = proj(Mx^*/Md^{(0)}) = \frac{d^{(0)}b}{d^{(0)}Ad^{(0)}}\, Md^{(0)}$$

Conjugaison: exemple

On obtient $x^{(2)}=x^*$



Conjugaison: algorithme

// construction de la base

- On choisit n vecteurs indépendants $\{r^{(k)}\}$
- Pour chaque vecteur $r^{(k)}$
  - $d^{(k)}$ reçoit $r^{(k)}$ moins sa M-projection sur les $d^{(i)}$ précédents

// projection de x* sur la base

- On calcule les coefficients $\alpha^{(k)}$ tels que $Mx^* = \sum \alpha^{(k)} Md^{(k)}$, en projetant Mx^* sur les $Md^{(k)}$
- On définit $x^* = \sum \alpha^{(k)} d^{(k)}$

Conjugaison: algorithme

// construction de la base

- On choisit n vecteurs indépendants $\{r^{(k)}\}$
- Pour chaque vecteur $r^{(k)}$
  - $d^{(k)}$ reçoit $r^{(k)}$ moins sa M-projection sur les $d^{(i)}$ précédents

// projection de x* sur la base

- On calcule les coefficients $\alpha^{(k)}$ tels que $Mx^* = \sum \alpha^{(k)} Md^{(k)}$, en projetant Mx^* sur les $Md^{(k)}$
- On définit $x^* = \sum \alpha^{(k)} d^{(k)}$

On va factoriser les boucles différemment

Conjugaison: algorithme

// construction de la base

- On choisit n vecteurs indépendants $\{r^{(k)}\}$
- Pour chaque vecteur $r^{(k)}$
  - $d^{(k)}$ reçoit $r^{(k)}$ moins sa M-projection sur les $d^{(i)}$ précédents

// projection de x* sur la base

- Initialiser $x^{(0)} = 0$
- On itère sur k:
  - On calcule le coefficient $\alpha^{(k)}$, en projetant Mx* sur $Md^{(k)}$
  - On définit $x^{(k)} = x^{(k-1)} + \alpha^{(k)} d^{(k)}$

Conjugaison: algorithme

// construction de la base

- On choisit n vecteurs indépendants $\{r^{(k)}\}$
- Pour chaque vecteur $r^{(k)}$
  - $d^{(k)}$ reçoit $r^{(k)}$ moins sa M-projection sur les $d^{(i)}$ précédents

// projection de x* sur la base

- Initialiser $x^{(0)} = 0$
- On itère sur k:
  - On calcule le coefficient $\alpha^{(k)}$, en projetant Mx* sur $Md^{(k)}$
  - On définit $x^{(k)} = x^{(k-1)} + \alpha^{(k)} d^{(k)}$

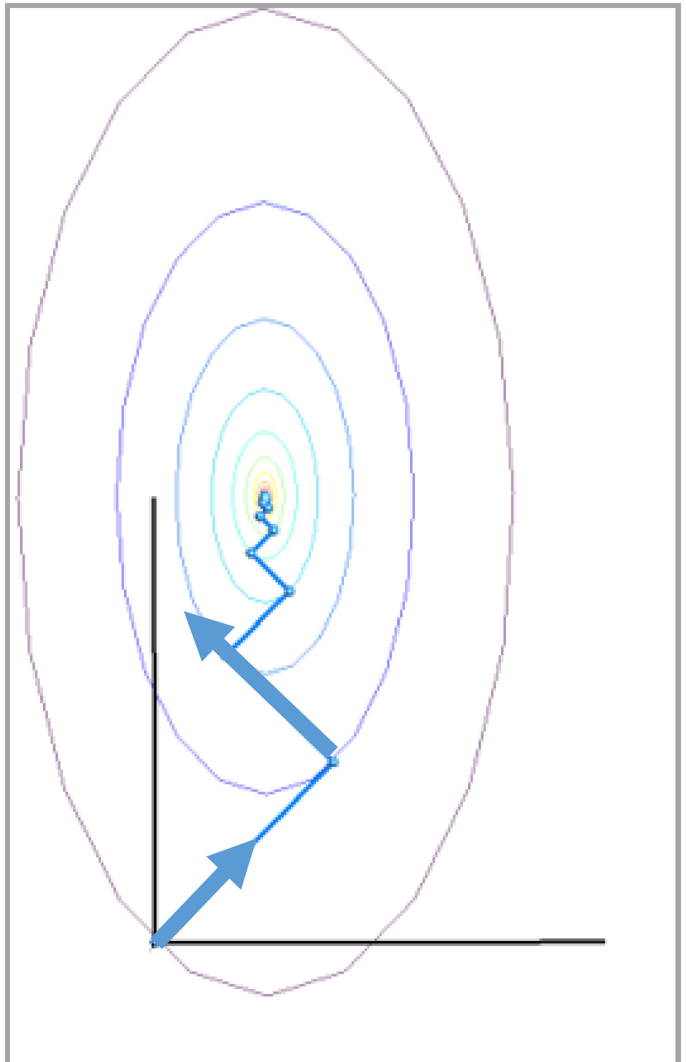
On va factoriser les boucles différemment

Conjugaison: algorithme

- On choisit n vecteurs indépendants $\{r^{(k)}\}$
- Initialiser $x^{(0)} = 0$
- Pour chaque dimension k
  - // construction de la base
  - $d^{(k)}$ reçoit $r^{(k)}$ moins sa M-projection sur les $d^{(i)}$ précédents
  - // projection de x* sur la base
  - On calcule le coefficient $\alpha^{(k)}$, en projetant Mx^* sur $Mr^{(k)}$
  - On définit $x^{(k+1)} = x^{(k)} + \alpha^{(k)} r^{(k)}$

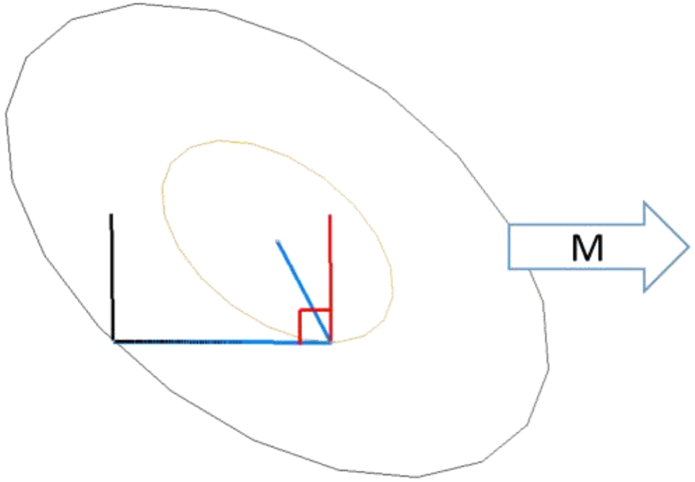
Descente de gradient: minimiser $x^T A x - 2b^T x$

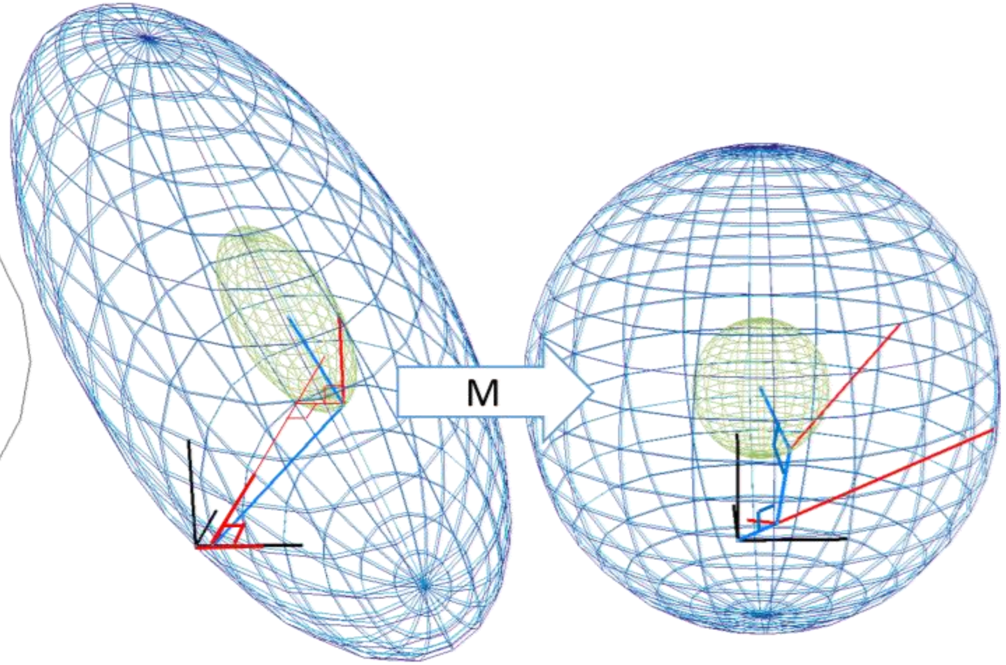
- Initialiser $x^{(0)} = 0$
- On itère sur k:
  - // calcul de la direction de descente
  - On calcule la direction $r^{(k)} = d^{(k)} = b - Ax^{(k)} = -\nabla f(x^{(i)})/2$
  - // déplacement de la solution
  - On calcule le pas de descente $\alpha^{(k)}$
  - On définit : $x^{(k+1)} = x^{(k)} + \alpha^{(k)} r^{(k)}$



Conjugaison: récapitulatif

- Algorithme:
  - On construit une base orthogonale dans M
  - On projette la solution sur cette base
- Performances:
  - Méthode directe en n itérations
  - Mais Gram-Schmidt est en n^2





Plan

- *Introduction*
- *Le problème*
- *Résolution par gradient*
- *Résolution par conjugaison*
- **Gradient conjugué**
  - **Utiliser le gradient pour construire la base A-orthogonale**
  - **Une simplification bienvenue**
- Contemplation de performance pour des problèmes discrétisés
  - Laplacien 1D
  - Laplacien 2D

Gradient conjugué: idée de base

- Remarque:
  - Choisir les vecteurs r_i au hasard est sous-optimal
- Idée… et si on prenait $r^{(k)} = -\nabla f(x^{(k)})$?
- Conséquence:
  - On se rapproche plus vite de la solution au début (comme le gradient)
  - Mais on reste assuré de finir en n itérations (par la conjugaison)
  - En pratique, ca converge bien plus vite !

Conjugaison: algorithme

- On choisit n vecteurs indépendants $\{r^{(k)}\}$
- Initialiser $x^{(0)} = 0$
- Pour tout k:
  - // construction de la base
  - On retire à $r^{(k)}$ sa projection sur les vecteurs précédents
  - // projection de x* sur la base
  - On calcule le coefficient $\alpha^{(k)}$, en projetant Mx^* sur $Mr^{(k)}$
  - On définit $x^{(k+1)} = x^{(k)} + \alpha^{(k)} r^{(k)}$

Gradient conjugué : algorithme

- ~~On choisit n vecteurs indépendants $\{r^{(k)}\}$~~
- Initialiser $x^{(0)}, r^{(0)} = 0$
- Pour tout k:
  - // construction de la base
  - On retire à $r^{(k)}$ sa projection sur les vecteurs précédents
  - // projection de x* sur la base
  - On calcule le coefficient $\alpha^{(k)}$, en projetant Mx^* sur $Mr^{(k)}$
  - On définit $x^{(k+1)} = x^{(k)} + \alpha^{(k)} r^{(k)}$

On calcule la direction $r^{(k)} = d^{(k)} = b - Ax^{(k)} = -\nabla f(x^{(k)})/2$

Gradient conjugué

Adaptation directe	Adaptation optimisée
$x^{(0)} \leftarrow 0$	$x^{(0)} \leftarrow 0$
$r^{(0)} \leftarrow b - Ax^{(0)}$	$r^{(0)} \leftarrow b - Ax^{(0)}$
$d^{(0)} \leftarrow r^{(0)}$	$d^{(0)} \leftarrow r^{(0)}$
Pour k de 0 à $n-1$:	Pour k de 0 à $n-1$:
$\alpha^{(k)} \leftarrow \frac{b^\top d^{(k)}}{d^{(k)\top} A d^{(k)}}$	$\alpha^{(k)} \leftarrow \frac{r^{(k)\top} r^{(k)}}{d^{(k)\top} A d^{(k)}}$
$x^{(k+1)} \leftarrow x^{(k)} + \alpha^{(k)} d^{(k)}$	$x^{(k+1)} \leftarrow x^{(k)} + \alpha^{(k)} d^{(k)}$
$r^{(k+1)} \leftarrow b - Ax^{(k+1)}$	$r^{(k+1)} \leftarrow r^{(k)} - \alpha^{(k)} A d^{(k)}$
$\beta_i^{k+1} \leftarrow \frac{r^{(k+1)\top} A d^{(i)}}{d^{(i)\top} A d^{(i)}}, \forall i$	$\beta^{(k+1)} \leftarrow -\frac{r^{(k+1)\top} r^{(k+1)}}{r^{(k)\top} r^{(k)}}$
$d^{(k+1)} \leftarrow r^{(k+1)} - \sum_{i=0}^{k} \beta_i^{(k+1)} d^{(k)}$	$d^{(k+1)} \leftarrow r^{(k+1)} - \beta^{(k+1)} d^{(k)}$

Gradient conjugué: final trick

Adaptation directe	Adaptation optimisée
$x^{(0)} \leftarrow 0$	$x^{(0)} \leftarrow 0$
$r^{(0)} \leftarrow b - Ax^{(0)}$	$r^{(0)} \leftarrow b - Ax^{(0)}$
$d^{(0)} \leftarrow r^{(0)}$	$d^{(0)} \leftarrow r^{(0)}$
Pour k de 0 à $n-1$:	Pour k de 0 à $n-1$:
$\alpha^{(k)} \leftarrow \frac{b^\top d^{(k)}}{d^{(k)\top} A d^{(k)}}$	$\alpha^{(k)} \leftarrow \frac{r^{(k)\top} r^{(k)}}{d^{(k)\top} A d^{(k)}}$
$x^{(k+1)} \leftarrow x^{(k)} + \alpha^{(k)} d^{(k)}$	$x^{(k+1)} \leftarrow x^{(k)} + \alpha^{(k)} d^{(k)}$
$r^{(k+1)} \leftarrow b - Ax^{(k+1)}$	$r^{(k+1)} \leftarrow r^{(k)} - \alpha^{(k)} A d^{(k)}$
$\beta_i^{k+1} \leftarrow \frac{r^{(k+1)\top} A d^{(i)}}{d^{(i)\top} A d^{(i)}}, \forall i$	$\beta^{(k+1)} \leftarrow -\frac{r^{(k+1)\top} r^{(k+1)}}{r^{(k)\top} r^{(k)}}$
$d^{(k+1)} \leftarrow r^{(k+1)} \boxed{- \sum_{i=0}^{k} \beta_i^{(k+1)} d^{(k)}}$	$d^{(k+1)} \leftarrow r^{(k+1)} \boxed{- \beta^{(k+1)} d^{(k)}}$

Le gradient est naturellement orthogonal avec toutes les directions précédentes sauf la dernière

Gradient conjugué: final trick

Adaptation directe	Adaptation optimisée
$x^{(0)} \leftarrow 0$	$x^{(0)} \leftarrow 0$
$r^{(0)} \leftarrow b - Ax^{(0)}$	$r^{(0)} \leftarrow b - Ax^{(0)}$
$d^{(0)} \leftarrow r^{(0)}$	$d^{(0)} \leftarrow r^{(0)}$
Pour k de 0 à $n-1$:	~~Pour k de 0 à $n-1$:~~ Tant que $r^{(k)} > \varepsilon$
$\alpha^{(k)} \leftarrow \frac{b^\top d^{(k)}}{d^{(k)\top} A d^{(k)}}$	$\alpha^{(k)} \leftarrow \frac{r^{(k)\top} r^{(k)}}{d^{(k)\top} A d^{(k)}}$
$x^{(k+1)} \leftarrow x^{(k)} + \alpha^{(k)} d^{(k)}$	$x^{(k+1)} \leftarrow x^{(k)} + \alpha^{(k)} d^{(k)}$
$r^{(k+1)} \leftarrow b - Ax^{(k+1)}$	$r^{(k+1)} \leftarrow r^{(k)} - \alpha^{(k)} A d^{(k)}$
$\beta_i^{k+1} \leftarrow \frac{r^{(k+1)\top} A d^{(i)}}{d^{(i)\top} A d^{(i)}}, \forall i$	$\beta^{(k+1)} \leftarrow -\frac{r^{(k+1)\top} r^{(k+1)}}{r^{(k)\top} r^{(k)}}$
$d^{(k+1)} \leftarrow r^{(k+1)} \boxed{- \sum_{i=0}^{k} \beta_i^{(k+1)} d^{(k)}}$	$d^{(k+1)} \leftarrow r^{(k+1)} \boxed{- \beta^{(k+1)} d^{(k)}}$

Le gradient est naturellement orthogonal avec toutes les directions précédentes sauf la dernière

Gradient conjugué: final trick

- Grosso modo, la dérivée (Ax-b) est linéaire donc quand une direction est annulée, elle le reste…
- La preuve par calcul est plus laborieuse qu’élégante
- Ca ne veut rien dire en 2D, et ce n’est pas simple en 3D
- On va essayer de s’en convaincre sur le cas du Laplacien 1D…

Plan

- *Introduction*
- *Le problème*
- *Résolution par gradient*
- *Résolution par conjugaison*
- *Gradient conjugué*
- **Contemplation de performance pour des problèmes discrétisés**
  - **Laplacien 1D**
  - **Laplacien 2D**

Gradient conjugué: Laplacien discretisé

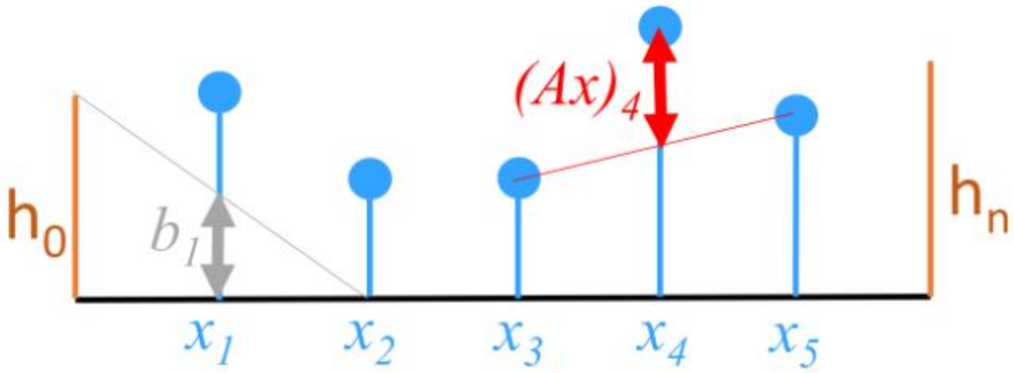
- Objectif:
  - Observer CG sur des problèmes plus intéressants
- Problème de Dirichlet: trouver une fonction f sur Ω
  - Dont le Laplacien est nul
  - Qui soit connue sur le bord de Ω
- Intérêt:
  - On obtient le même système pour différences finies, FEM linéaire, et minimisation L^2 d'un gradient
  - Méthode de lissage / terme de régulation extrêmement utilisés
  - Types de matrices creuses extrêmement commun

Gradient conjugué: Laplacien 1D/2D

- Discrétisation du Laplacien 1D/2D
  - On place une grille sur le domaine Ω
  - f est représentée par sa valeur aux nœuds de la grille (le vecteur x)
  - Le Laplacien est approximé par différences finie (la matrice A)
  - La valeur du Laplacien (le vecteur b) assure les contraintes du bord

Gradient conjugué: Laplacien 1D

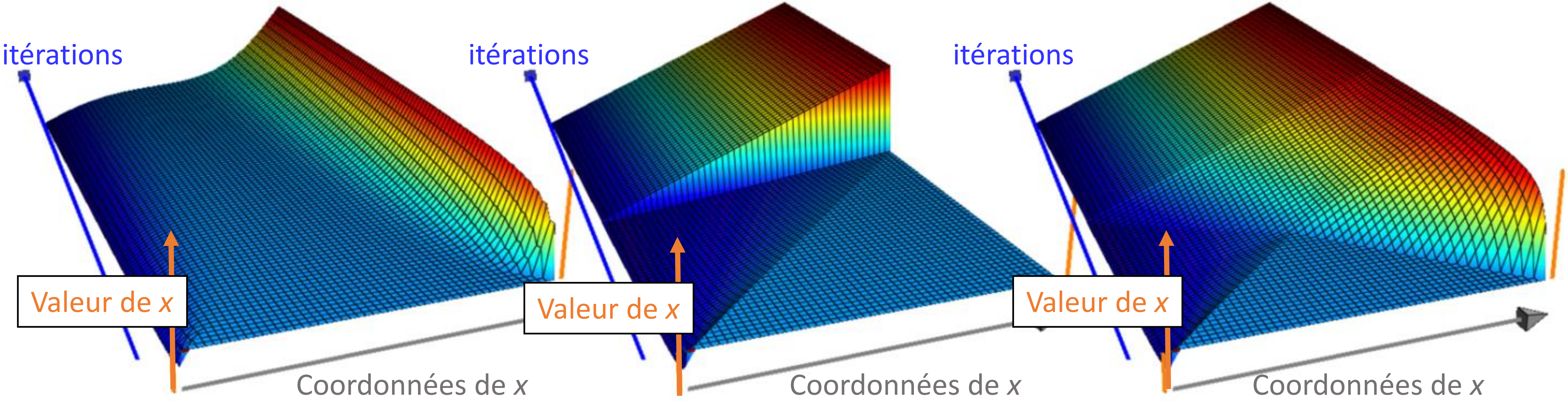
- Problème du Laplacien 1D:
  - $x_i = f(i)$
  - La $i^{\text{ème}}$ ligne de A est $-1/2\, x_{i-1} + x_i - 1/2\, x_{i+1}$
  - b représente des contraintes (au bord)



$$A = \begin{pmatrix} 1 & -1/2 & 0 & 0 & 0 \\ -1/2 & 1 & -1/2 & 0 & 0 \\ 0 & -1/2 & 1 & -1/2 & 0 \\ 0 & 0 & -1/2 & 1 & -1/2 \\ 0 & 0 & 0 & -1/2 & 1 \end{pmatrix}, b = \begin{pmatrix} h_0/2 \\ 0 \\ 0 \\ 0 \\ h_n/2 \end{pmatrix}$$

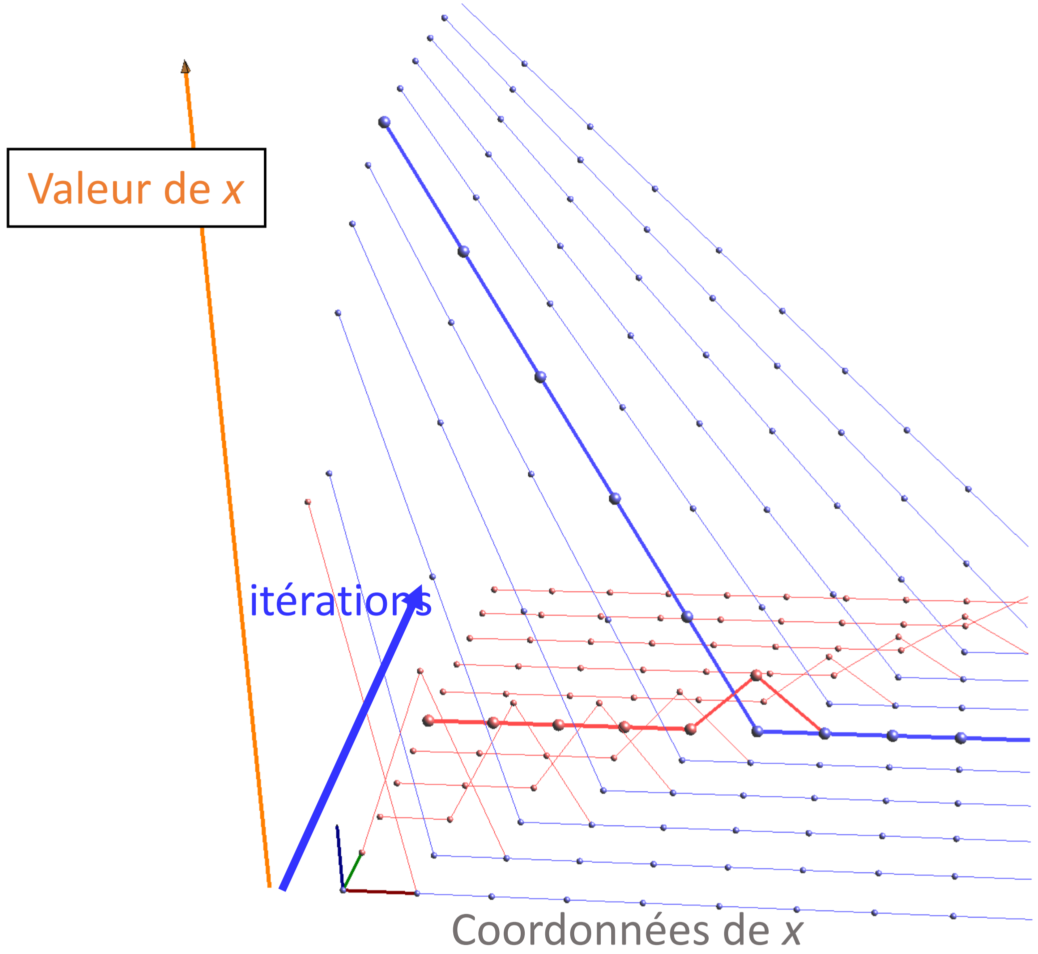
Laplacien 1D

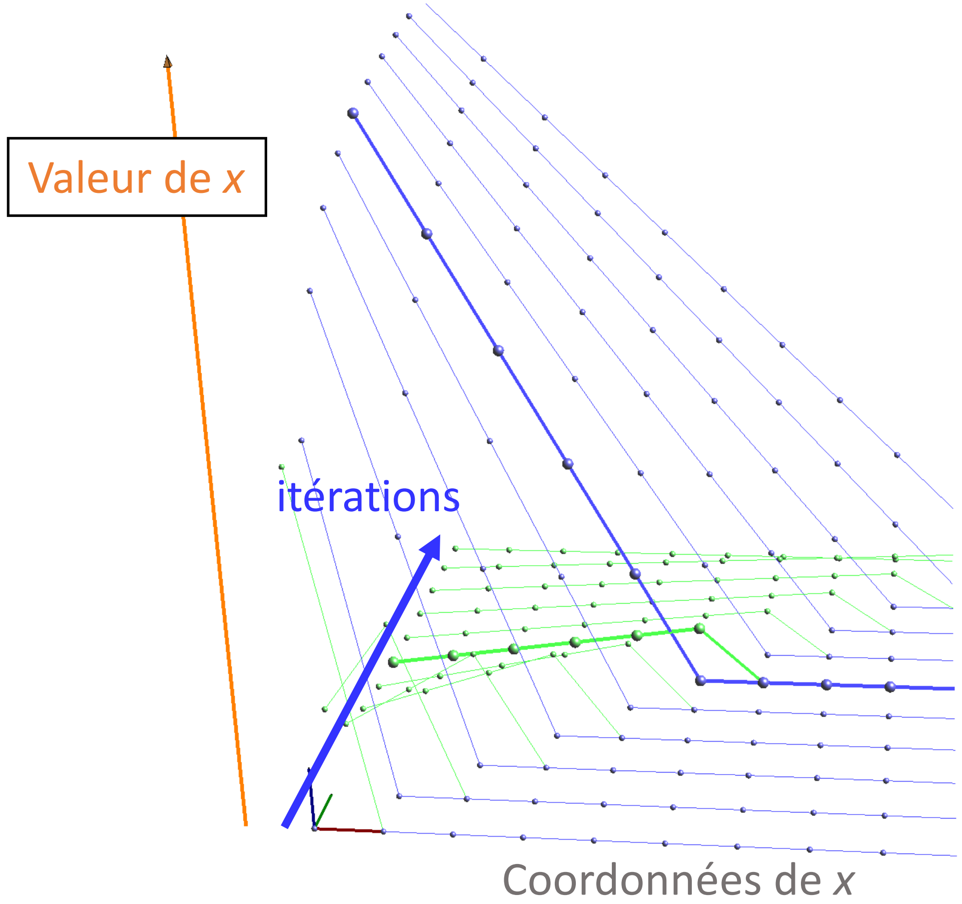
Gradient · Conjugaison · CG



Gradient conjugué: Laplacien 1D

- Final trick ?





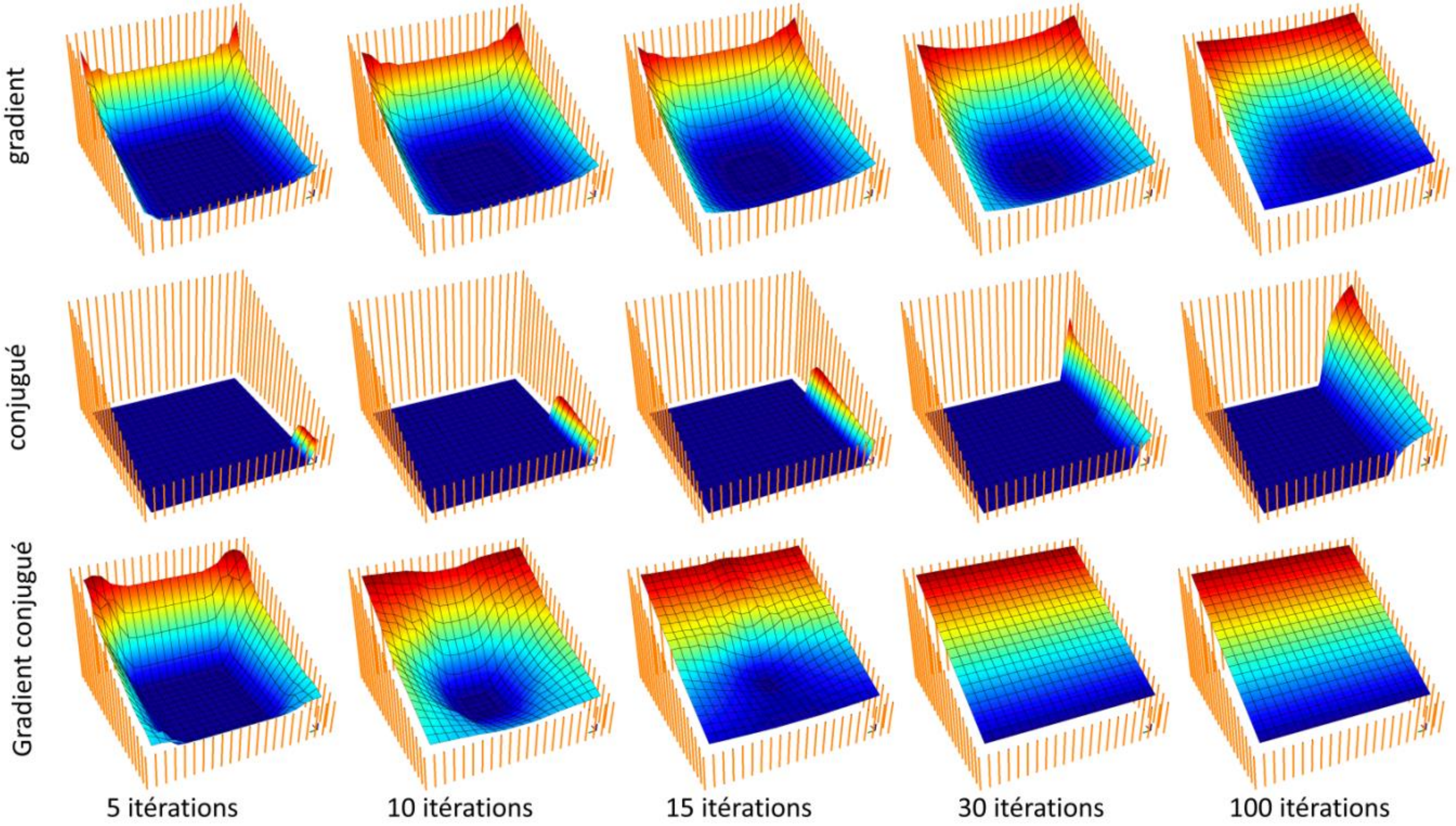
Gradient conjugué: Laplacien 2D

- Problème du Laplacien 2D:
  - x_k = f(i,j) avec i,j =k mod n, k / n
  - La $k^{ème}$ ligne de A est f(i,j) $- 1/4$ (f(i-1,j)+ f(i+1,j)+ f(i,j-1)+ f(i,j+1))
  - b représente des contraintes (au bord)

$$\begin{bmatrix}
4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 4 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 4 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4
\end{bmatrix}$$

[Introduction pratique aux EDP, JP Chehab]

Laplacien 2D



Conclusion

- A retenir --- culture générale:
  - CG est extrêmement simple à implémenter
  - Utilisé comme brique de base de l'optimisation numérique (FEM, LS, pbs non linéaires, A non sym. def. +, CG pré-conditionné, etc.)
- A retenir --- techniquement:
  - On peut séparer le « gradient » du « conjugué »
  - On sait calculer des produits scalaires dans $\mathrm{M} = \sqrt{A}$
  - Le gradient est déjà orthogonal avec les précédents
- En pratique:
  - Pour des problèmes discrétisés, chaque itération découvre un peu plus de domaine.
    - le nombre d'itérations doit permettre aux contraintes de « se voir ».