

---

# Moindres carrés

Michel Bierlaire

`michel.bierlaire@epfl.ch`

Laboratoire Transport et Mobilité

EPFL - ENAC - TRANSP-OR

# Problème des moindres carrés

---

Problème d'optimisation de la forme

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} \|g(x)\|^2 = \frac{1}{2} \sum_{i=1}^m g_i(x)^2$$

avec  $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$  différentiable

Contexte : calibration de paramètres d'un modèle mathématique

# Résistivité du cuivre

---

Quelle est la résistivité du cuivre ?

- Barre de cuivre
  - 1 m de long,
  - 1 cm<sup>2</sup> de section
- Expérience
  - envoyer du courant
  - mesurer la différence de potentiel
- Modèle mathématique : loi d'Ohm  $v = ri$

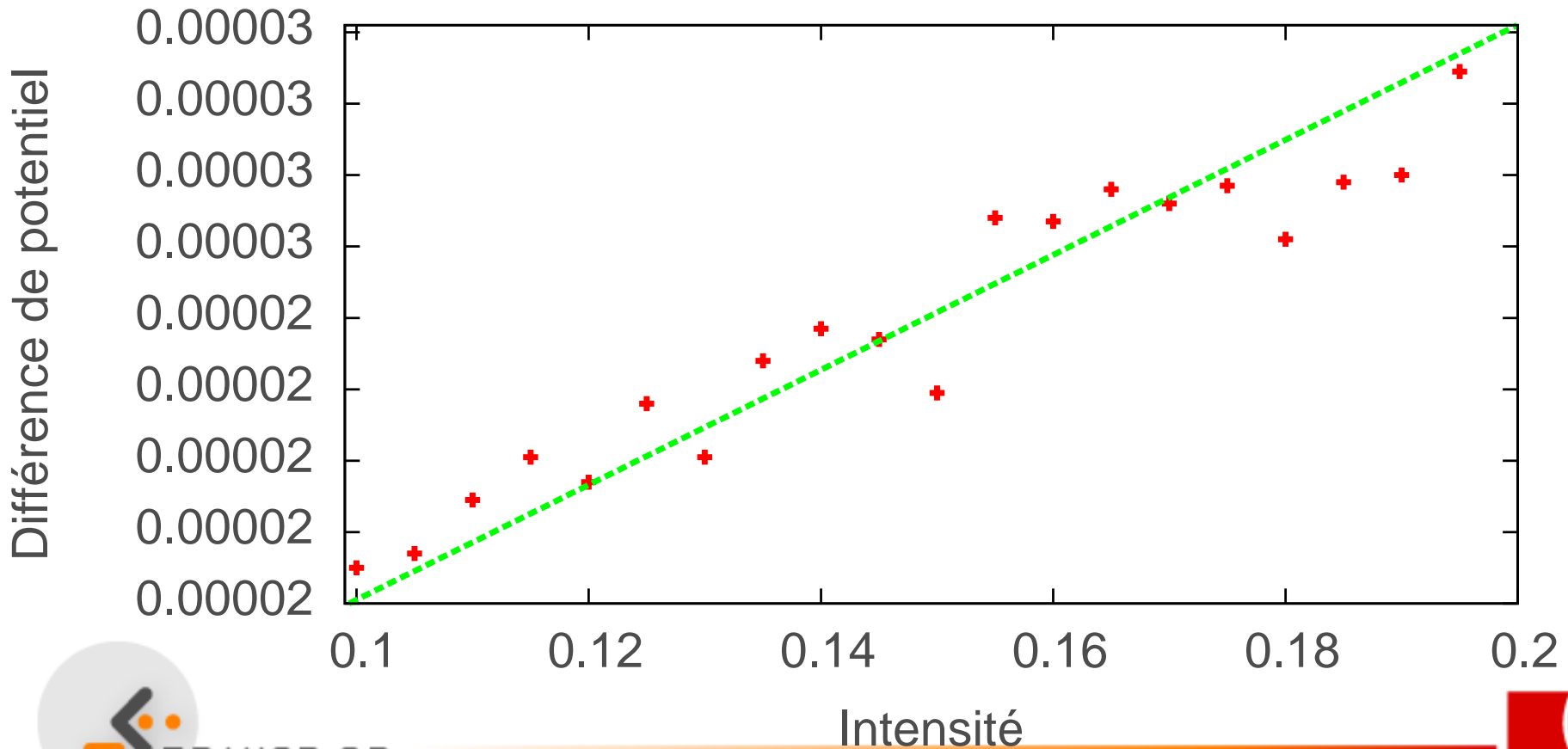
# Résistivité du cuivre

## Expériences

Intensité	Voltage	Intensité	Voltage
0.100	1.70E-05	0.150	2.19E-05
0.105	1.74E-05	0.155	2.68E-05
0.110	1.89E-05	0.160	2.67E-05
0.115	2.01E-05	0.165	2.76E-05
0.120	1.94E-05	0.170	2.72E-05
0.125	2.16E-05	0.175	2.77E-05
0.130	2.01E-05	0.180	2.62E-05
0.135	2.28E-05	0.185	2.78E-05
0.140	2.37E-05	0.190	2.80E-05
0.145	2.34E-05	0.195	3.09E-05
		0.200	3.45E-05

# Résistivité du cuivre

$$r^* = \operatorname{argmin}_r \sum_{k=1}^{21} (ri_k - v_k)^2$$



# Modélisation

---

- Système comportant plusieurs configurations
- Chaque configuration  $i$  est définie par
  - des valeurs d'entrée  $\alpha_i$
  - des valeurs de sortie  $\beta_i$
- Modèle mathématique

$$\beta_i + \varepsilon_i = \Lambda(\alpha_i; x)$$

- $x$  : paramètres du modèle
- $\varepsilon_i$  : variable aléatoire représentant les erreurs de mesure et de modélisation

# Modélisation

- Minimiser l'erreur sous contrainte de reproduire les observations

$$\min_{x, \varepsilon} \sum_i \varepsilon_i^2$$

sous contrainte

$$\beta_i + \varepsilon_i = \Lambda(\alpha_i; x), \quad i = 1, \dots$$

- En éliminant  $\varepsilon$

$$\min_x \sum_i (\Lambda(\alpha_i; x) - \beta_i)^2, \quad g_i(x) = \Lambda(\alpha_i; x) - \beta_i$$

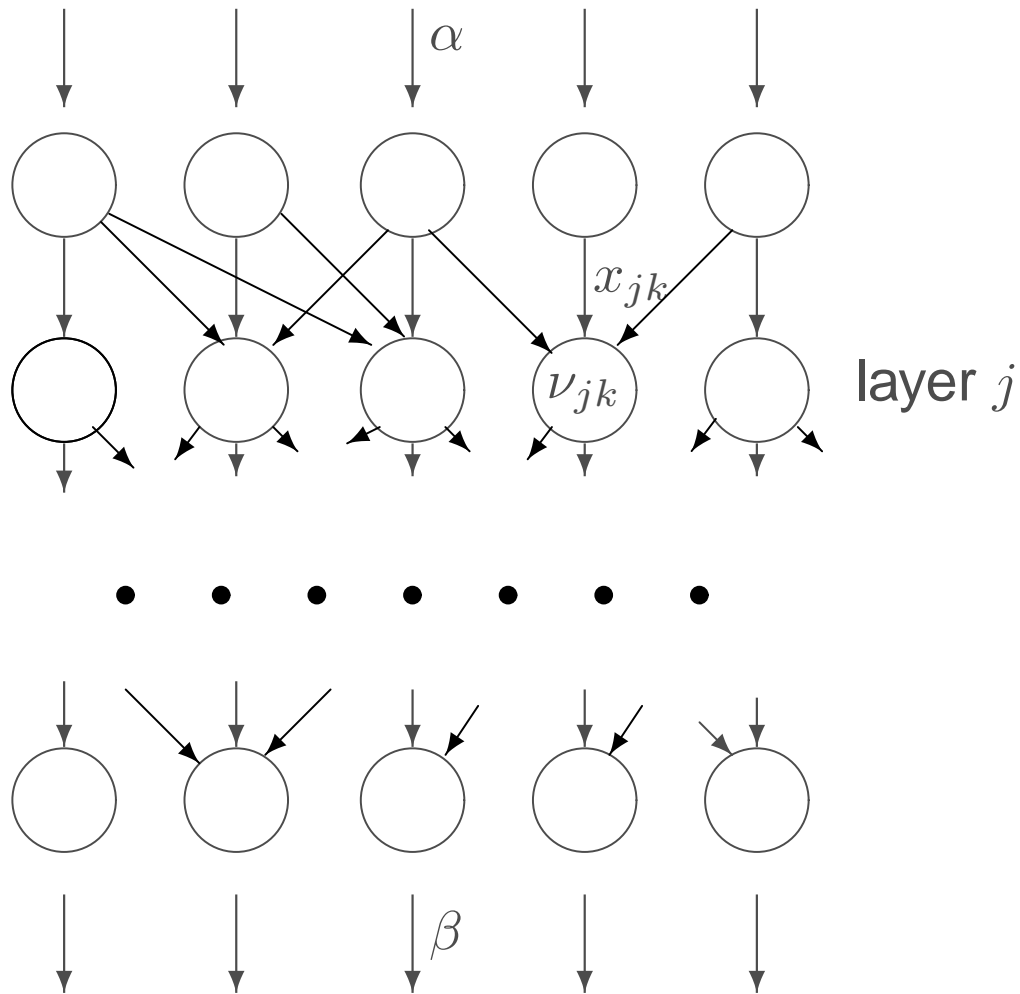
# Réseaux de neurones

---

- Modèle mathématique basé sur l'analogie biologique
- But : appréhender la complexité d'un système en utilisant un réseau d'unités simples
- Chaque unité (neurone) effectue une tâche simple, en utilisant l'information fournie par d'autres unités.
- Organisation en  $N$  couches



# Réseaux de neurones



# Réseaux de neurones

---

- Un neurone  $j$  de la couche  $k$  utilise des infos de la couche  $k - 1$

$$\nu_{j,k} = \phi \left( (x_{jk})_0 + \sum_{i=1}^{n_k} (x_{jk})_i \nu_{i,k-1} \right).$$

- Fonction sigmoïdale:  $\phi(\alpha) = \frac{1}{1+e^{-\alpha}}$
- Fonction hyperbolique tangente:  $\phi(\alpha) = \frac{e^{\alpha} - e^{-\alpha}}{e^{\alpha} + e^{-\alpha}}$ .

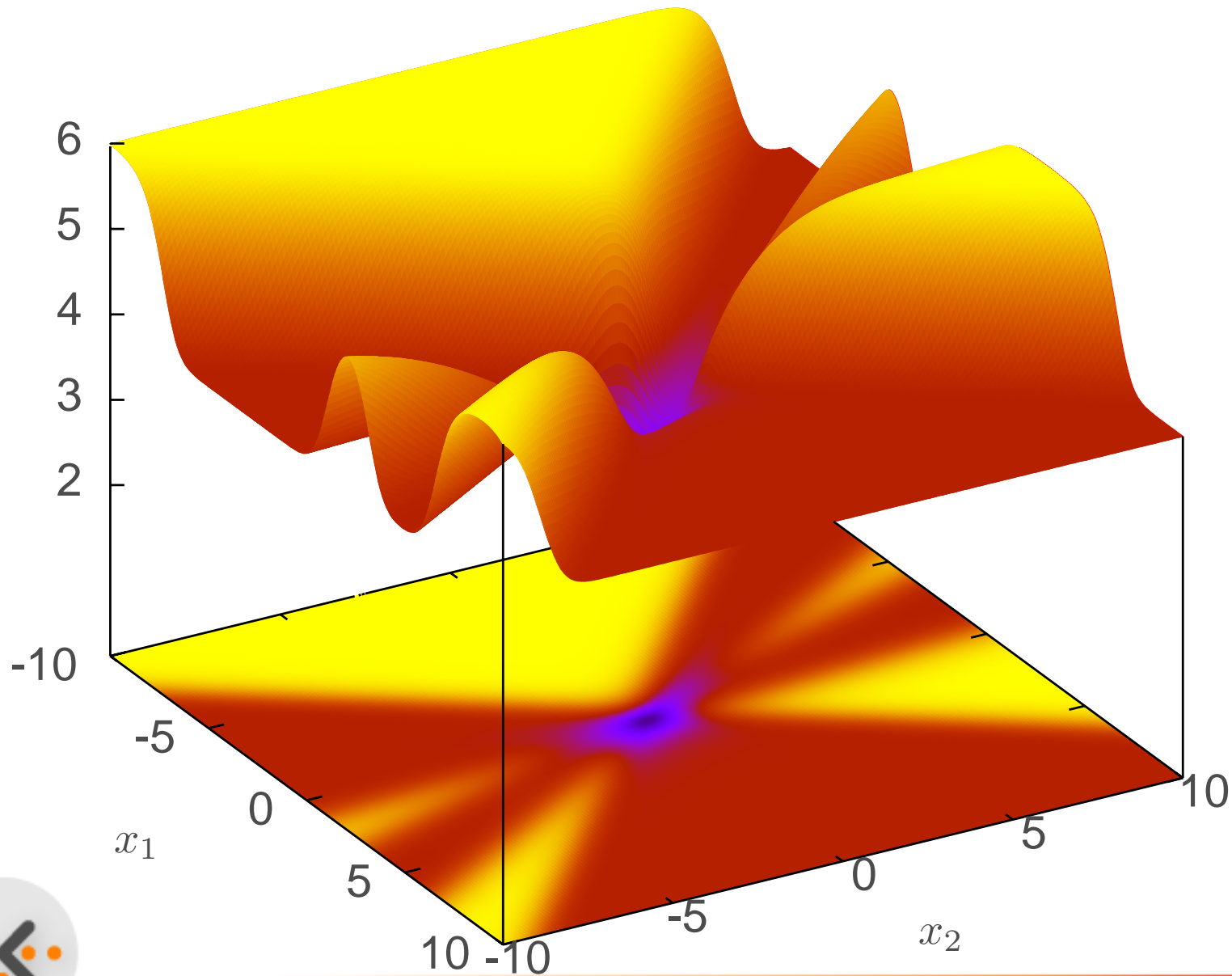
# Réseaux de neurones

- Apprentissage de réseau de neurones = moindres carrés
- Exemple avec la fonction hyperbolique tangente

$$\min_{x_0, x_1} \frac{1}{2} \sum_{i=1}^5 (\beta_i - \phi(x_1 \alpha_i + x_2))^2,$$

$\alpha_i$	$\beta_i$
1.165	1
0.626	-1
0.075	-1
0.351	1
-0.696	1

# Réseaux de neurones



# Gauss-Newton

---

- Algorithme pour résoudre

$$\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2} g(x)^T g(x)$$

- Gradient

$$\nabla f(x) = \nabla g(x) g(x) = \sum_{i=1}^m \nabla g_i(x) g_i(x),$$

avec  $\nabla g(x) \in \mathbb{R}^{n \times m}$  est la matrice gradient de  $g$

# Gauss-Newton

- Hessien

$$\begin{aligned}\nabla^2 f(x) &= \sum_{i=1}^m (\nabla g_i(x) \nabla g_i(x)^T + \nabla^2 g_i(x) g_i(x)) \\ &= \nabla g(x) \nabla g(x)^T + \sum_{i=1}^m \nabla^2 g_i(x) g_i(x).\end{aligned}$$

- Le second terme est très coûteux. Utilisons l'algorithme de plus forte pente préconditionnée avec

$$H_k = \nabla g(x) \nabla g(x)^T$$

# Algorithme : Gauss-Newton

---

## Objectif

Trouver une approximation de la solution du problème aux moindres carrés

$$\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2} g(x)^T g(x). \quad (1)$$

## Input

- La fonction  $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ ,
- La matrice gradient de  $g$ :  $\nabla g : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$ ;
- Une première approximation de la solution  $x_0 \in \mathbb{R}^n$ ;
- La précision demandée  $\varepsilon \in \mathbb{R}$ ,  $\varepsilon > 0$ .

# Algorithme : Gauss-Newton

---

## Output

Une approximation de la solution  $x^* \in \mathbb{R}^n$

## Initialisation

$$k = 0$$



# Algorithme : Gauss-Newton

---

## Itérations

1. Calculer  $d_{k+1}$  solution de

$$\nabla g(x_k) \nabla g(x_k)^T d_{k+1} = -\nabla g(x_k) g(x_k),$$

2.  $x_{k+1} = x_k + d_{k+1}$ ,

3.  $k = k + 1$ .

## Critère d'arrêt

Si  $\|\nabla g(x_k) g(x_k)\| \leq \varepsilon$ , alors  $x^* = x_k$ .

# Gauss-Newton

---

Mêmes problèmes que pour Newton :

- bien définie que si  $\nabla g(x_k)\nabla g(x_k)^T$  est inversible
- méthode de descente uniquement si  $\nabla g(x_k)\nabla g(x_k)^T$  est définie positive
- convergente si  $x_0$  n'est pas trop éloigné de  $x^*$ .

Solutions identiques

- Recherche linéaire
- Modification de la matrice pour la rendre définie positive

Levenberg-Marquardt

# Cas linéaire

---

$$g(x) = Ax - b$$

avec  $A \in \mathbb{R}^{m \times n}$

$$\nabla g(x) = A^T$$

Gauss-Newton:

$$\nabla g(x_k) \nabla g(x_k)^T d_{k+1} = -\nabla g(x_k) g(x_k)$$

$$A^T A d_{k+1} = -A^T (Ax_k - b)$$

Comme  $d_{k+1} = x_{k+1} - x_k$ , on obtient

$$A^T A x_{k+1} = A^T b, \quad \forall x_k$$

# Cas linéaire

## Equations normales

Soit le problème de moindres carrés linéaire

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} \|Ax - b\|_2^2,$$

avec  $A \in \mathbb{R}^{m \times n}$  et  $b \in \mathbb{R}^m$ . Le système d'équations

$$A^T Ax = A^T b$$

est appelé système d'équations normales du problème de moindres carrés.

# Cas linéaire

---

**Equations normales** Soient  $A \in \mathbb{R}^{m \times n}$  et  $b \in \mathbb{R}^m$ . Alors,  $x^*$  est solution des équations normales

$$A^T A x = A^T b$$

si et seulement si  $x^*$  est solution optimale de

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} \|Ax - b\|_2^2,$$

(p. 339)

# Intuitions pour la preuve

---

- Fonction objectif :

$$f(x) = \frac{1}{2}(Ax - b)^T(Ax - b)$$

- Gradient :

$$\nabla f(x) = A^T Ax - A^T b,$$

- Equations normales : gradient = 0.
- Hessien : semi défini positif

$$\nabla^2 f(x) = A^T A.$$

- $f$  est convexe.
- Théorème des conditions d'optimalité globale s'applique.

# Interprétation géométrique

- Analogie avec Newton
- Newton : modéliser  $f$  par une quadratique
- Gauss-Newton : modéliser  $g$  par un modèle linéaire

$$m_{x_k}(x) = g(x_k) + \nabla g(x_k)^T (x - x_k).$$

- Problème simplifié :

$$\min_x M(x) = \min_x \frac{1}{2} \|m_{x_k}(x)\|_2^2 = \min_x \frac{1}{2} m_{x_k}(x)^T m_{x_k}(x).$$

$$\begin{aligned} M(x) = \frac{1}{2} m_{x_k}(x)^T m_{x_k}(x) &= \frac{1}{2} g(x_k)^T g(x_k) \\ &+ (x - x_k)^T \nabla g(x_k) g(x_k) \\ &+ \frac{1}{2} (x - x_k)^T \nabla g(x_k) \nabla g(x_k)^T (x - x_k). \end{aligned}$$

# Interprétation géométrique

---

$$\begin{aligned}M(x) = \frac{1}{2}m_{x_k}(x)^T m_{x_k}(x) &= \frac{1}{2}g(x_k)^T g(x_k) \\ &+ (x - x_k)^T \nabla g(x_k) g(x_k) \\ &+ \frac{1}{2}(x - x_k)^T \nabla g(x_k) \nabla g(x_k)^T (x - x_k).\end{aligned}$$

Le gradient de cette expression est

$$\nabla M(x) = \nabla g(x_k) \nabla g(x_k)^T (x - x_k) + \nabla g(x_k) g(x_k).$$

Il s'annule en

$$x_{k+1} = x_k - (\nabla g(x_k) \nabla g(x_k)^T)^{-1} \nabla g(x_k) g(x_k),$$

si  $\nabla g(x_k) \nabla g(x_k)^T$  est inversible.

**C'est Gauss-Newton.**



# Filtre de Kalman

---

- Contexte : données organisées par blocs
- Motivation :
  - sources dispersées dans l'espace
  - sources dispersées dans le temps
  - applications en temps réel
  - très large base de données

# Filtre de Kalman

---

Le problème

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2^2$$

se décompose en  $J$  blocs, chacun contenant  $m_j$  données.

$$\min_{x \in \mathbb{R}^n} \sum_{j=1}^J \|A_j x - b_j\|_2^2,$$

où  $A_j \in \mathbb{R}^{m_j \times n}$  et  $b_j \in \mathbb{R}^{m_j}$ .

Calcul incrémental de la solution

(p. 341)

# Preuve

---

- Premier bloc :

$$\min_{x \in \mathbb{R}^n} \|A_1 x - b_1\|_2^2.$$

- Equations normales :

$$A_1^T A_1 x_1 = A_1^T b_1.$$

- Supposons  $A_1$  de rang plein, et donc  $A_1^T A_1$  inversible :

$$x_1 = (A_1^T A_1)^{-1} A_1^T b_1.$$

# Preuve (suite)

---

- Deux premiers blocs:

$$\min_{x \in \mathbb{R}^n} \|A_1 x - b_1\|_2^2 + \|A_2 x - b_2\|_2^2,$$

ou encore

$$\min_{x \in \mathbb{R}^n} \left\| \begin{pmatrix} A_1 \\ A_2 \end{pmatrix} x - \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \right\|_2^2,$$

- Equations normales :

$$(A_1^T A_1 + A_2^T A_2)x_2 = A_1^T b_1 + A_2^T b_2.$$

# Preuve (suite)

---

$$A_1^T A_1 x_1 = A_1^T b_1 \text{ et } (A_1^T A_1 + A_2^T A_2)x_2 = A_1^T b_1 + A_2^T b_2.$$

Donc

$$\begin{aligned}(A_1^T A_1 + A_2^T A_2)x_2 &= A_1^T b_1 + A_2^T b_2 \\ &= A_1^T A_1 x_1 + A_2^T b_2 \\ &= A_1^T A_1 x_1 + A_2^T A_2 x_1 - A_2^T A_2 x_1 + A_2^T b_2 \\ &= (A_1^T A_1 + A_2^T A_2)x_1 + A_2^T (b_2 - A_2 x_1).\end{aligned}$$

Comme  $A_1^T A_1$  est inversible,  $A_1^T A_1 + A_2^T A_2$  aussi, et

$$x_2 = x_1 + (A_1^T A_1 + A_2^T A_2)^{-1} A_2^T (b_2 - A_2 x_1).$$

# Filtre de Kalman

---

On peut montrer de manière similaire

$$x_3 = x_2 + (A_1^T A_1 + A_2^T A_2 + A_3^T A_3)^{-1} A_3^T (b_3 - A_3 x_2).$$

$$x_4 = x_3 + (A_1^T A_1 + A_2^T A_2 + A_3^T A_3 + A_4^T A_4)^{-1} A_4^T (b_4 - A_4 x_3).$$

# Algorithme : Filtre de Kalman

---

## Objectif

Trouver la solution  $x^*$  d'un problème aux moindres carrés linéaire

$$\min_{x \in \mathbb{R}^n} \sum_{j=1}^J \|A_j x - b_j\|_2^2,$$

de manière incrémentale.

## Input

- Les matrices  $A_j \in \mathbb{R}^{m_j \times n}$ ,  $j = 1, \dots, J$ .
- Les vecteur  $b_j \in \mathbb{R}^{m_j}$ ,  $j = 1, \dots, J$ .

# Algorithme : Filtre de Kalman

---

## Input (suite)

- Une solution initiale  $x_0 \in \mathbb{R}^n$  (défaut :  $x_0 = 0$ ).
- Un filtre initial  $H_0 \in \mathbb{R}^{n \times n}$  (défaut :  $H_0 = 0$ ).

## Output

La solution  $x^* \in \mathbb{R}^n$



# Algorithme : Filtre de Kalman

---

## Initialisation

$$j = 1$$

## Itérations

1.  $H_j = H_{j-1} + A_j^T A_j.$

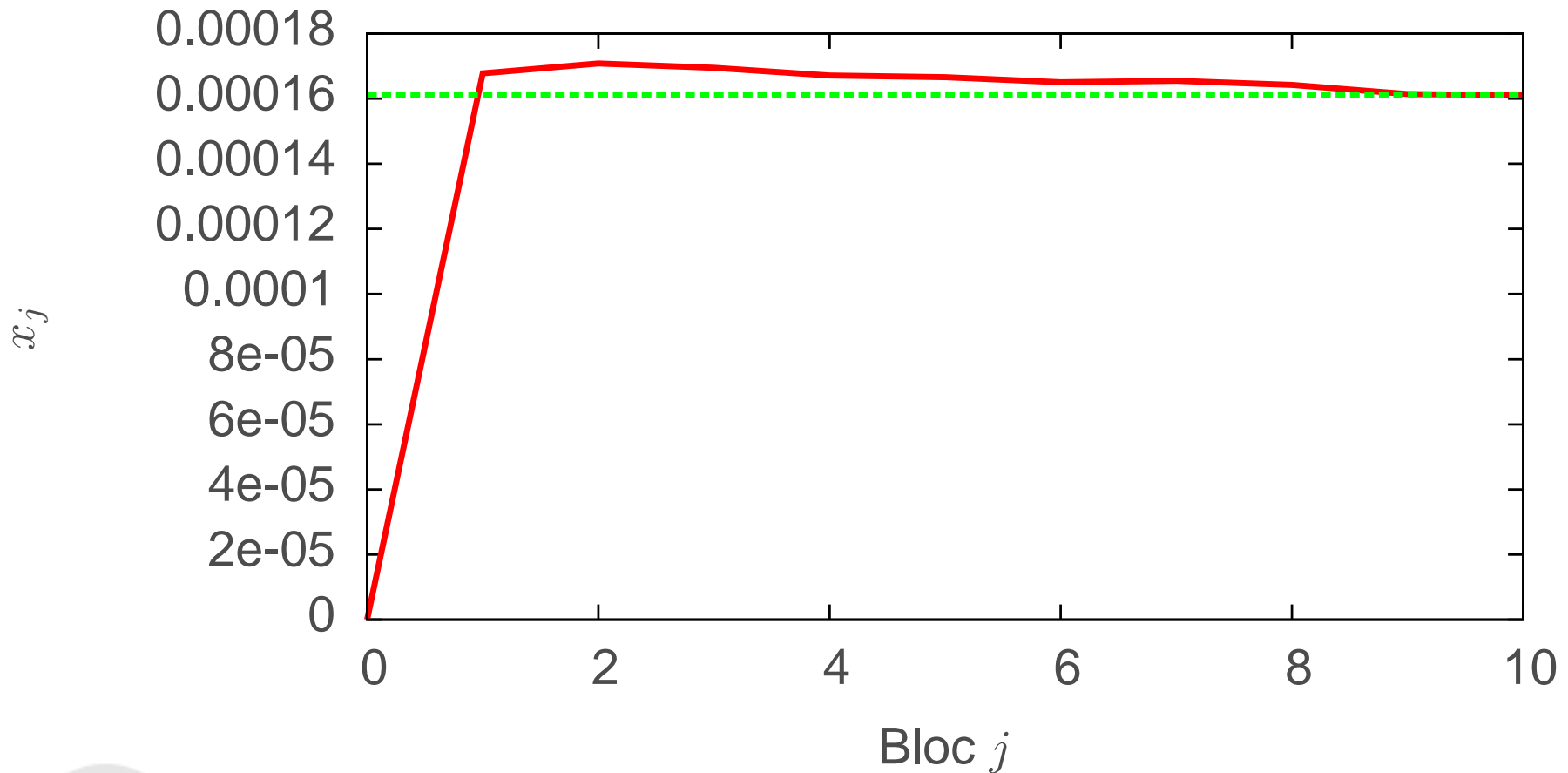
2.  $x_j = x_{j-1} + H_j^{-1} A_j^T (b_j - A_j x_{j-1})$

## Critère d'arrêt

Lorsque  $j = J$ ,  $x^* = x_J.$

# Résistivité du cuivre

10 blocs de données



# Validité de l'algorithme

---

- Il faut que  $H_1 = A_1^T A_1$  soit inversible
- S'il y a suffisamment de données dans le bloc 1,  $A_1$  est de rang plein, et  $H_1$  est inversible.
- Sinon, poser  $H_0 = \tau I$ , avec  $\tau$  petit, et  $H_1 = H_0 + A_1^T A_1$ .

Attention: cela modifie la solution

# Temps réel

---

- Paramètres doivent être mis à jour continuellement
- Concept d'“âge” des données
- Vieilles données moins représentatives que les récentes
- Paramètre de dépréciation :  $0 < \lambda \leq 1$
- Pendant la période  $j$ , on reçoit les données  $(A_j, b_j)$ .

# Algorithme : Filtre de Kalman en temps réel

---

## Objectif

Mettre à jour les paramètres d'un modèle linéaire au fur et à mesure que de nouvelles données sont disponibles. A chaque intervalle de temps  $J$ , on résoud le problème

$$\min_{x \in \mathbb{R}^n} \sum_{j=1}^J \lambda^{J-j} \|A_j x - b_j\|_2^2,$$

en mettant à jour la solution de l'intervalle de temps  $J - 1$ .

# Algorithme : Filtre de Kalman en temps réel

---

## Input

- La matrice  $A_J \in \mathbb{R}^{m_J \times n}$ .
- Le vecteur  $b_J \in \mathbb{R}^{m_J}$ .
- La solution précédente  $x_{J-1} \in \mathbb{R}^n$ .
- Le filtre précédent  $H_{J-1} \in \mathbb{R}^{n \times n}$ .
- Un facteur de dépréciation  $\lambda$  tel que  $0 < \lambda \leq 1$

## Output

$x_J$  et  $H_J$ .

# Algorithme : Filtre de Kalman en temps réel

---

## Mise à jour

1.  $H_J = \lambda H_{J-1} + A_J^T A_J.$
2.  $x_J = x_{J-1} + H_J^{-1} A_J^T (b_J - A_J x_{J-1})$

# Régression orthogonale

---

Hypothèses des moindres carrés :

- variables dépendantes  $\beta_i$  sujettes à des erreurs aléatoires (distribution normale)
- variables indépendantes  $\alpha_i$  connues avec exactitude

Pour le problème du cuivre

- les différences de potentiel sont entachées d'erreur
- les intensités sont exactes

Hypothèse souvent trop forte en pratique



# Régression orthogonale

---

Supposons que les deux soient entachées d'erreur

$$\beta_i + \varepsilon_i = \Lambda(\alpha_i + \xi_i; x),$$

avec  $\varepsilon_i$  et  $\xi_i$  variables aléatoires

- indépendantes
- de moyenne nulle
- de variances identiques

# Régression orthogonale

---

Problème d'optimisation :

$$\min_{x, \varepsilon, \xi} \sum_i (\varepsilon_i^2 + \xi_i^2)$$

sous contrainte

$$\beta_i + \varepsilon_i = \Lambda(\alpha_i + \xi_i; x), \quad i = 1, \dots$$

ou encore

$$\min_{x, \xi} \sum_i ((\Lambda(\alpha_i + \xi_i; x) - \beta_i)^2 + \xi_i^2).$$

# Régression orthogonale

---

## Problème plus compliqué

- Si  $\Lambda$  est linéaire, ce n'est plus un problème de moindres carrés linéaire standard
- Nombre d'inconnues =  $n + m$
- S'il y a beaucoup de données, problème de grande taille

# Régression orthogonale

## Interprétation géométrique

