Méthodes numériques
approximation
et optimisation

INSTITUT
d'OPTIQUE
GRADUATE SCHOOL
ParisTech

Optimisation – Approximation
Approches non-linéaires

Méthodes non-linéaires

- Impossible de réduire le problème à un ensemble de contraintes linéaires

- Méthodes itératives
  - Aller pas à pas vers une solution
  - Convergence vers un minimum local
    - Lieu où les dérivées partielles sont nulles
  - Si solution unique, on la trouve

Rappel : Méthode de Newton 1D - e(x) = 0

- Approximation localement affine (développement de Taylor)

$$\mathrm{e}\left(x^{(k)}+h\right) \simeq \mathrm{e}\left(x^{(k)}\right) + \partial_x \mathrm{e}\left(x^{(k)}\right) h$$

- On cherche le zéro

$$h = -\frac{\mathrm{e}\left(x^{(k)}\right)}{\mathrm{d}_x \mathrm{e}\left(x^{(k)}\right)}$$

- On itère

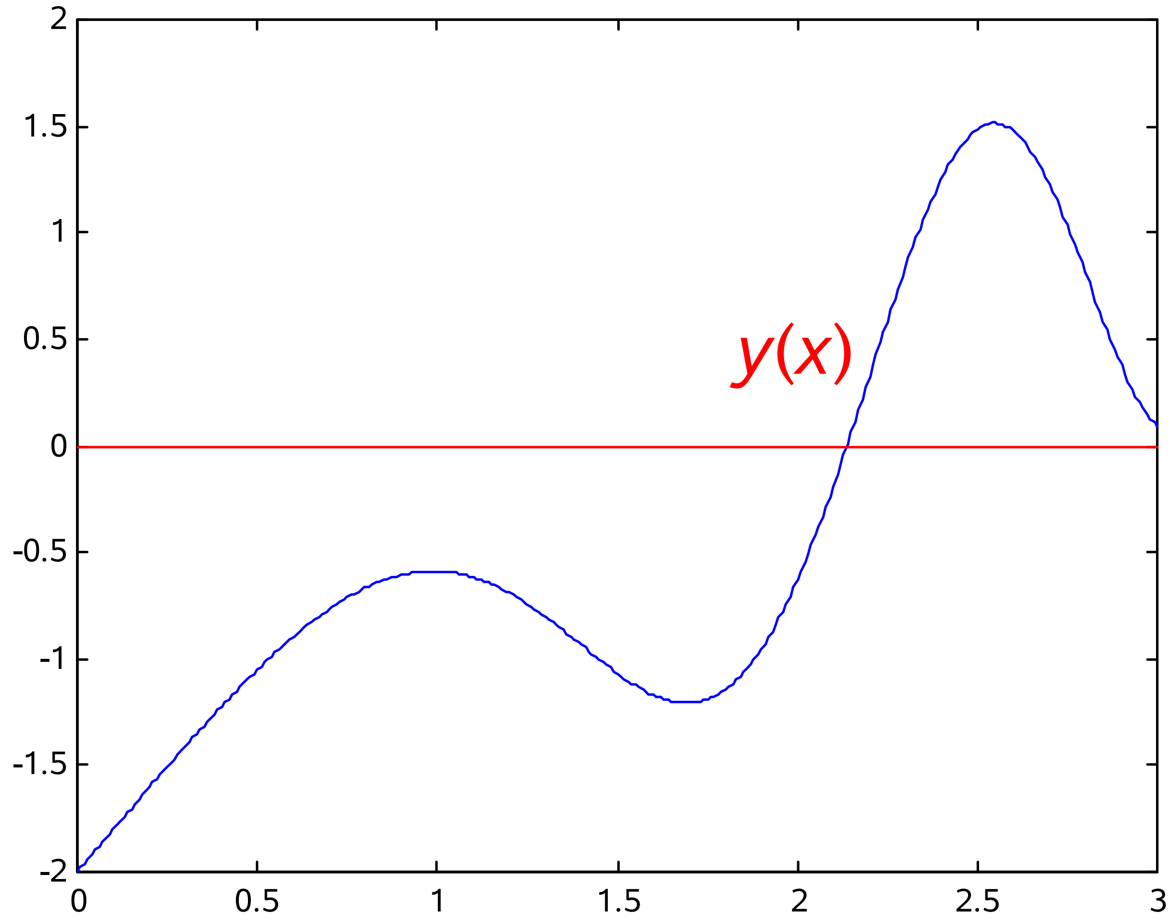
$$x^{(k+1)} = x^{(k)} - \frac{\mathrm{e}\left(x^{(k)}\right)}{\mathrm{d}_x \mathrm{e}\left(x^{(k)}\right)}$$

Méthode de Newton

- Illustration

$y = \tanh(x)\cos(x^2) + x - 2$

$y' = (1 - \tanh^2(x))\cos(x^2)$
$\quad - 2\tanh(x)\sin(x^2)x + 1$



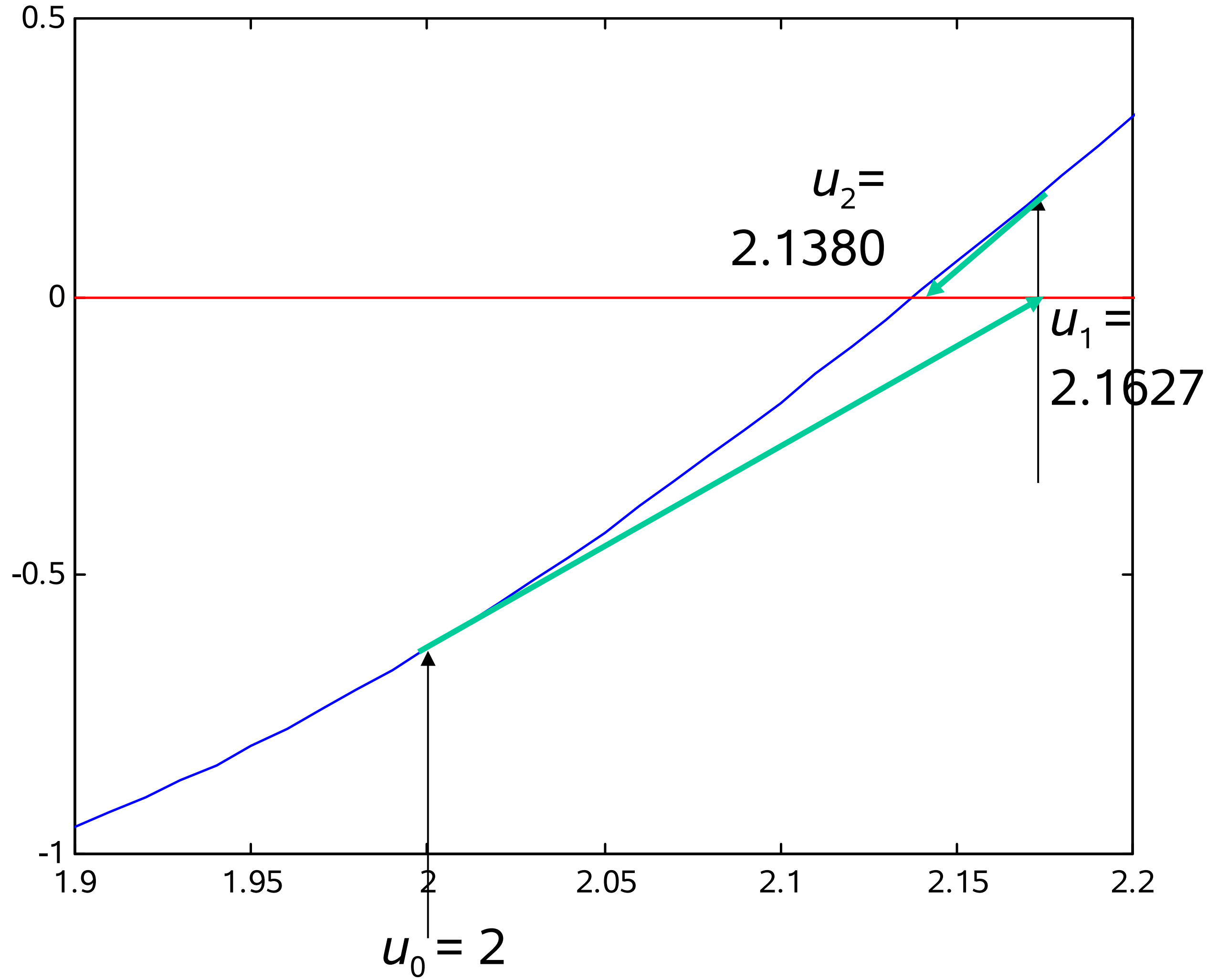
Méthode de Newton

- Illustration

$y = \tanh(x)\cos(x^2) + x - 2$

$y' = (1 - \tanh^2(x))\cos(x^2)$
$\quad - 2\tanh(x)\sin(x^2)x + 1$

$u_0 = 2$
$u_1 = 2.1627$
$u_2 = 2.1380$
$u_3 = 2.1378$
$u_4 = 2.1378$



Méthode de Newton : convergence

- Convergence quadratique

$$\left|x^{(k+1)}-x\right| \simeq \left|x^{(k)}-x\right|^2 \left|\frac{\mathrm{d}_{xx}^2 \mathrm{e}(x)}{2\,\mathrm{d}_x \mathrm{e}(x)}\right|$$

- Conditions
  - Dérivée analytique connue
  - Intersection de la tangente dans le domaine de définition

Optimisation 1D – e'(x) = 0

- Approximation localement affine

$$\mathrm{e}\left(x^{(k)}+h\right)\simeq\mathrm{e}\left(x^{(k)}\right)+\mathrm{d}_x\mathrm{e}\left(x^{(k)}\right)h+\frac{1}{2}\mathrm{d}_{xx}^2\mathrm{e}\left(x^{(k)}\right)h^2$$

- On cherche le zéro de la dérivée

- On itère

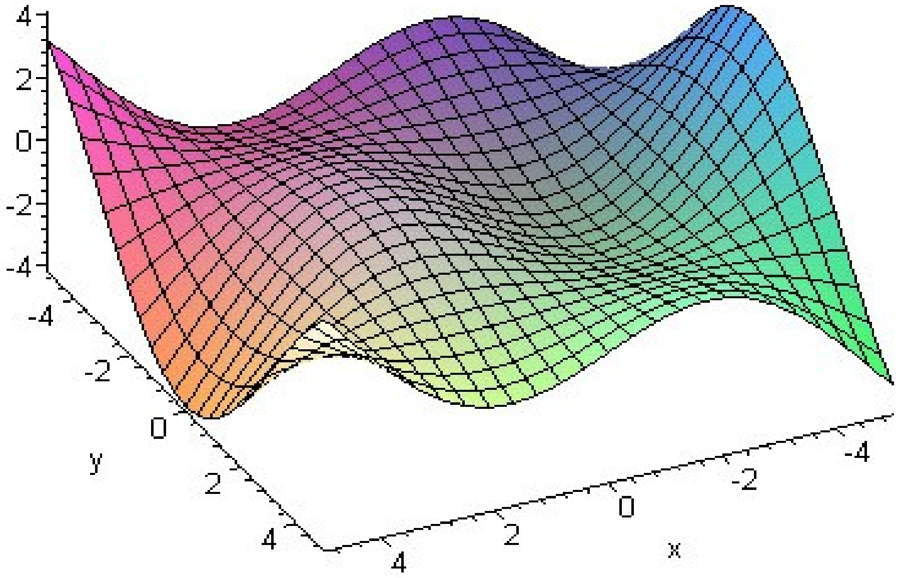
$$h=-\frac{\mathrm{d}_x\mathrm{e}\left(x^{(k)}\right)}{\mathrm{d}_{xx}^2\mathrm{e}\left(x^{(k)}\right)}$$

$$x^{(k+1)}=x^{(k)}-\frac{\mathrm{d}_x\mathrm{e}\left(x^{(k)}\right)}{\mathrm{d}_{xx}^2\mathrm{e}\left(x^{(k)}\right)}$$

Extension du développement : Fonction 2D

- Comment étendre le développement de Taylor ?

$$e(x, y) = x\cos\left(\frac{x}{2}\right)\sin\left(\frac{y}{2}\right)$$

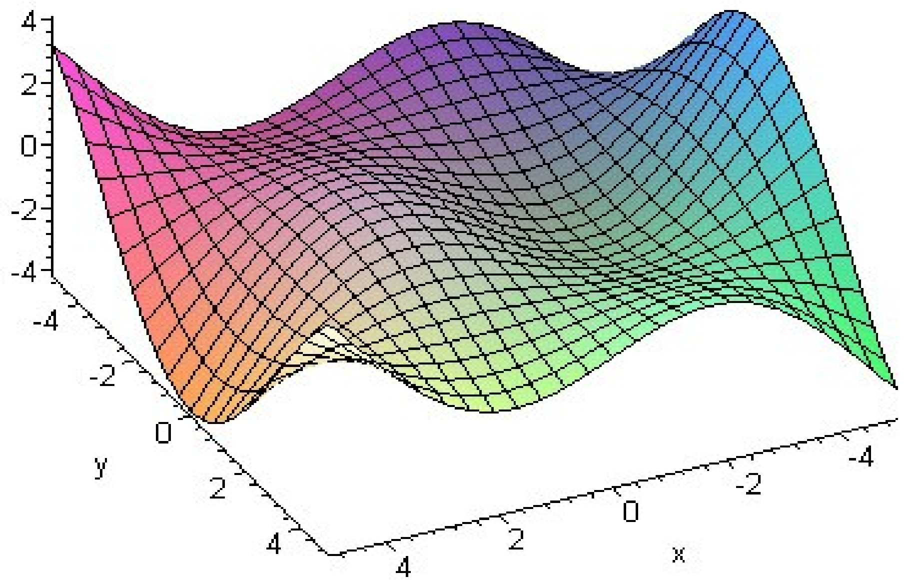


Extension du développement : Fonction 2D

- Comment étendre le développement de Taylor ?

- Gradient $\nabla e(x) = \begin{bmatrix} \partial_x e \\ \partial_y e \end{bmatrix}$

$$e(x, y) = x \cos\left(\frac{x}{2}\right) \sin\left(\frac{y}{2}\right)$$



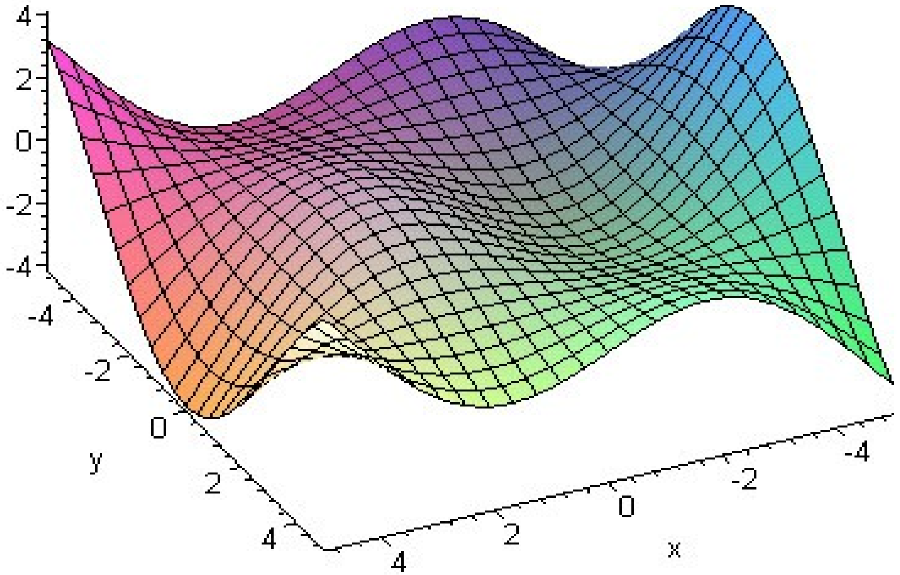
$$\nabla e(x) = \begin{bmatrix} \sin\left(\frac{y}{2}\right)\left(\cos\left(\frac{x}{2}\right) - \frac{x}{2}\sin\left(\frac{x}{2}\right)\right) \\ \frac{x}{2}\cos\left(\frac{x}{2}\right)\cos\left(\frac{y}{2}\right) \end{bmatrix}$$

Extension du développement : Fonction 2D

- Comment étendre le développement de Taylor ?

$$e(x, y) = x\cos\left(\frac{x}{2}\right)\sin\left(\frac{y}{2}\right)$$

- Gradient $\nabla e(x) = \begin{bmatrix} \partial_x e \\ \partial_y e \end{bmatrix}$
-
- Dérivée 1ere ordre
  - Produit scalaire direction/gradient



$$\boldsymbol{d} = (d_x, d_y) \in \Re^2$$

$$\partial_{\boldsymbol{d}} e = \langle \boldsymbol{d}, \nabla e \rangle$$

$$\partial_{\boldsymbol{d}} e = d_x \partial_x e + d_y \partial_y e$$

$$\nabla e(x) = \begin{bmatrix} \sin\left(\frac{y}{2}\right)\left(\cos\left(\frac{x}{2}\right) - \frac{x}{2}\sin\left(\frac{x}{2}\right)\right) \\ \frac{x}{2}\cos\left(\frac{x}{2}\right)\cos\left(\frac{y}{2}\right) \end{bmatrix}$$

Dérivation en dimension N

- Une équation, N inconnues :

$$\mathrm{e}(\boldsymbol{x}+\boldsymbol{h})=\mathrm{e}(\boldsymbol{x})+\nabla^{\mathrm{T}}\mathrm{e}(\boldsymbol{x})\boldsymbol{h}+\frac{1}{2}\boldsymbol{h}^{\mathrm{T}}\mathrm{H}_{\mathrm{e}}(\boldsymbol{x})\boldsymbol{h}+\mathrm{o}\left(\|\boldsymbol{h}\|_2^2\right)$$

gradient

matrice Hessienne

$$\nabla \mathrm{e}=\begin{bmatrix}\partial_{x_1}\mathrm{e}\\ \vdots\\ \partial_{x_i}\mathrm{e}\\ \vdots\\ \partial_{x_N}\mathrm{e}\end{bmatrix}$$

$$\mathrm{H}_{\mathrm{e}}(\boldsymbol{x})=\begin{bmatrix}\partial^2_{x_1x_1}\mathrm{e} & \cdots & \partial^2_{x_1x_i}\mathrm{e} & \cdots & \partial^2_{x_1x_N}\mathrm{e}\\ \vdots & \ddots & \vdots & ⋰ & \vdots\\ \partial^2_{x_ix_1}\mathrm{e} & \cdots & \partial^2_{x_ix_i}\mathrm{e} & \cdots & \partial^2_{x_ix_N}\mathrm{e}\\ \vdots & ⋰ & \vdots & \ddots & \vdots\\ \partial^2_{x_Nx_1}\mathrm{e} & \cdots & \partial^2_{x_Nx_i}\mathrm{e} & \cdots & \partial^2_{x_Nx_N}\mathrm{e}\end{bmatrix}$$

Dérivation en dimension N

- Une équation, N inconnues :

$$\mathrm{e}(\boldsymbol{x}+\boldsymbol{h})=\mathrm{e}(\boldsymbol{x})+\nabla^{\mathrm{T}}\mathrm{e}(\boldsymbol{x})\boldsymbol{h}+\frac{1}{2}\boldsymbol{h}^{\mathrm{T}}\mathrm{H}_{\mathrm{e}}(\boldsymbol{x})\boldsymbol{h}+\mathrm{o}\left(\|\boldsymbol{h}\|_2^2\right)$$

gradient = tenseur 1D

matrice Hessienne = tenseur 2D

$$\nabla\mathrm{e}=\begin{bmatrix}\partial_{x_1}\mathrm{e}\\ \vdots\\ \partial_{x_i}\mathrm{e}\\ \vdots\\ \partial_{x_N}\mathrm{e}\end{bmatrix}$$

$$\mathrm{H}_{\mathrm{e}}(\boldsymbol{x})=\begin{bmatrix}\partial^2_{x_1x_1}\mathrm{e} & \cdots & \partial^2_{x_1x_i}\mathrm{e} & \cdots & \partial^2_{x_1x_N}\mathrm{e}\\ \vdots & \ddots & \vdots & ⋰ & \vdots\\ \partial^2_{x_ix_1}\mathrm{e} & \cdots & \partial^2_{x_ix_i}\mathrm{e} & \cdots & \partial^2_{x_ix_N}\mathrm{e}\\ \vdots & ⋰ & \vdots & \ddots & \vdots\\ \partial^2_{x_Nx_1}\mathrm{e} & \cdots & \partial^2_{x_Nx_i}\mathrm{e} & \cdots & \partial^2_{x_Nx_N}\mathrm{e}\end{bmatrix}$$

Matrice Hessienne

- Tenseur d'ordre 2
  - Associée à une forme quadratique

$$\frac{1}{2}\boldsymbol{h}^{\mathrm{T}}\mathrm{H}_{\mathrm{e}}(\boldsymbol{x})\boldsymbol{h}$$

- Symétrique
  - Théorème de Schawrz

    Pour une fonction n-dérivable et dont les dérivées sont continues, l'ordre de dérivation n'a pas d'influence sur le résultat.

  - Mise en application

$$\mathrm{e}(x,y)=x\cos\left(\frac{x}{2}\right)\sin\left(\frac{y}{2}\right)$$

$$\mathrm{H}_{\mathrm{e}}(\boldsymbol{x})=\begin{bmatrix} -\sin\left(\frac{y}{2}\right)\left(\sin\left(\frac{x}{2}\right)+\frac{x}{4}\cos\left(\frac{y}{2}\right)\right) & \frac{1}{2}\cos\left(\frac{y}{2}\right)\left(\cos\left(\frac{x}{2}\right)-\frac{x}{2}\sin\left(\frac{x}{2}\right)\right) \\ \frac{1}{2}\cos\left(\frac{y}{2}\right)\left(\cos\left(\frac{x}{2}\right)-\frac{x}{2}\sin\left(\frac{x}{2}\right)\right) & -\frac{x}{4}\cos\left(\frac{x}{2}\right)\sin\left(\frac{y}{2}\right) \end{bmatrix}$$

Rappel : dérivation en dimension N x M

- M équations, N inconnues :

$$\mathbf{e}(\boldsymbol{x}+\boldsymbol{h})=\mathbf{e}(\boldsymbol{x})+\mathrm{J}_{\mathbf{e}}(\boldsymbol{x})\boldsymbol{h}+\mathrm{o}\left(\|\boldsymbol{h}\|_2\right)$$

La matrice jacobienne

$$\mathrm{J}_{\mathbf{e}}(\boldsymbol{x})=\begin{bmatrix} \partial_{x_1}\mathrm{e}_1(\boldsymbol{x}) & \partial_{x_2}\mathrm{e}_1(\boldsymbol{x}) & \cdots & \partial_{x_N}\mathrm{e}_1(\boldsymbol{x}) \\ \partial_{x_1}\mathrm{e}_2(\boldsymbol{x}) & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \partial_{x_1}\mathrm{e}_N(\boldsymbol{x}) & \cdots & \cdots & \partial_{x_N}\mathrm{e}_N(\boldsymbol{x}) \end{bmatrix}$$

Matrice Jacobienne

- Attention : dérivée à l'ordre 1
  - Équivalent du gradient pour des fonctions multivaluées
  - Différente de la matrice hessienne

- Utilisé pour le changement de variable pour l'intégration
  - **e** est une fonction bijective
  - N = M (matrice carrée)

$$\iiint_D \mathrm{f}(\boldsymbol{x})\,\mathrm{d}x_{1\ldots}\,\mathrm{d}x_N = \iiint_{\mathbf{e}^{-1}(D)} \mathrm{f}(\mathbf{e}(\boldsymbol{x}))\left|\det \mathrm{J}_{\boldsymbol{e}}(\boldsymbol{x})\right|\mathrm{d}e_{1\ldots}\,\mathrm{d}e_N$$

Optimisation trouver $\nabla \mathrm{e}(\boldsymbol{x}) = 0$

- Approximation localement polynomiale

$$\mathrm{e}\left(\boldsymbol{x}^{(k)}+\boldsymbol{h}\right)=\mathrm{e}\left(\boldsymbol{x}^{(k)}\right)+\boldsymbol{h}^{\mathrm{T}}\nabla\mathrm{e}\left(\boldsymbol{x}^{(k)}\right)+\frac{1}{2}\boldsymbol{h}^{\mathrm{T}}\mathrm{H}_{\mathrm{e}}\left(\boldsymbol{x}^{(k)}\right)\boldsymbol{h}$$

- On cherche le zéro du gradient

$$\boldsymbol{h}=-\left(\mathrm{H}_{\mathrm{e}}\left(\boldsymbol{x}^{(k)}\right)\right)^{-1}\nabla\mathrm{e}\left(\boldsymbol{x}^{(k)}\right)$$

- On itère

$$\boldsymbol{x}^{(k+1)}=\boldsymbol{x}^{(k)}-\left(\mathrm{H}_{\mathrm{e}}\left(\boldsymbol{x}^{(k)}\right)\right)^{-1}\nabla\mathrm{e}\left(\boldsymbol{x}^{(k)}\right)$$

Problème de la méthode de Newton

- Si la Hessienne n'est pas semi-définie positive
  - Le pas accroît l'erreur

Descente de gradient

- Suivre la pente de la fonction
  - Pente = gradient
- Choisir un pas pour avancer dans cette direction

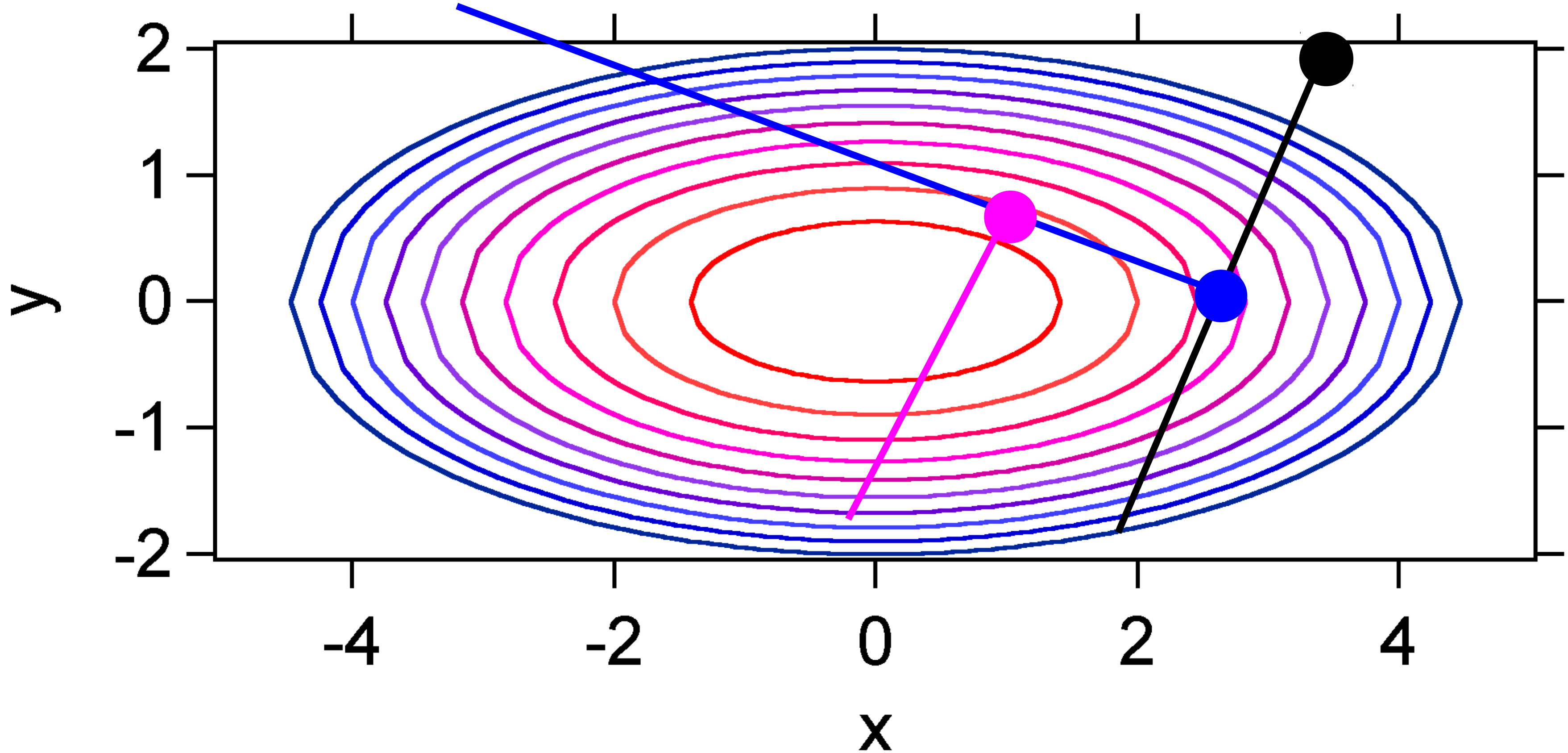
$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \rho \nabla \mathrm{e}\left(\boldsymbol{x}^{(k)}\right) \text{ avec } \rho \text{ tel que } \mathrm{e}\left(\boldsymbol{x}^{(k+1)}\right) < \mathrm{e}\left(\boldsymbol{x}^{(k)}\right)$$

$$\min_{\rho} \mathrm{e}\left(\boldsymbol{x}^{(k)} + \rho \nabla \mathrm{e}\left(\boldsymbol{x}^{(k)}\right)\right)$$

$$\partial_{\rho} \mathrm{e}\left(\boldsymbol{x}^{(k)} + \rho \nabla \mathrm{e}\left(\boldsymbol{x}^{(k)}\right)\right) = 0 = \left(\nabla \mathrm{e}\left(\boldsymbol{x}^{(k)}\right)\right)^{\mathrm{T}} \nabla \mathrm{e}\left(\boldsymbol{x}^{(k)} + \rho \nabla \mathrm{e}\left(\boldsymbol{x}^{(k)}\right)\right)$$

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \frac{\nabla^{\mathrm{T}} \mathrm{e}\left(\boldsymbol{x}^{(k)}\right) \nabla \mathrm{e}\left(\boldsymbol{x}^{(k)}\right)}{\nabla^{\mathrm{T}} \mathrm{e}\left(\boldsymbol{x}^{(k)}\right) \mathrm{H}_{\mathrm{e}}\left(\boldsymbol{x}^{(k)}\right) \nabla \mathrm{e}\left(\boldsymbol{x}^{(k)}\right)} \nabla \mathrm{e}\left(\boldsymbol{x}^{(k)}\right)$$

Descente de gradient



Descente de gradient : méthode itérative

- Soit K le conditionnement de M
  - Convergence

$$\| \boldsymbol{e} \|_{\mathrm{Q}}^{2} = \boldsymbol{e}^{\mathrm{T}} \mathrm{Q}\, \boldsymbol{e} \qquad \| \boldsymbol{e}^{(i)} \|_{\mathrm{Q}} \leq \left(\frac{K-1}{K+1} \right)^{(i)} \| \boldsymbol{e}^{(0)} \|_{\mathrm{Q}}$$

- Le point de départ est très important
  - Ex : Partir du résultat d'une méthode directe
- Autre question
  - Une meilleure direction ?

Gradient conjugué

- Principe
  - M est symétrique définie positive
  - Directions de descente orthogonales au sens M

$$\boldsymbol{x} \perp \boldsymbol{y} \Leftrightarrow \boldsymbol{x}^{\mathrm{T}} \mathrm{Q}\, \boldsymbol{x} = 0$$

- Pour chaque itération
  - Direction orthogonale (Gram-Schmidt)

$$\boldsymbol{d}_i = \boldsymbol{r}_i - \sum_{k<i} \frac{\boldsymbol{d}_k^{\mathrm{T}} \mathrm{Q}\, \boldsymbol{r}_k}{\boldsymbol{d}_k^{\mathrm{T}} \mathrm{Q}\, \boldsymbol{d}_k} \boldsymbol{d}_k$$

  - Nouveau pas

$$a = \frac{\boldsymbol{d}_i^{\mathrm{T}} \boldsymbol{r}_i}{\boldsymbol{d}_i^{\mathrm{T}} \mathrm{Q}\, \boldsymbol{d}_i}$$

Comparaison des convergences

- Soit K le conditionnement de la matrice M
- Descente de gradient

$$\| \boldsymbol{e}^{(i)} \|_{Q} \leq \left(\frac{K-1}{K+1} \right)^{(i)} \| \boldsymbol{e}^{(0)} \|_{Q}$$

- Gradient conjuguée

$$\| \boldsymbol{e}^{(i)} \|_{Q} \leq 2 \left(\frac{\sqrt{K}-1}{\sqrt{K}+1} \right)^{(i)} \| \boldsymbol{e}^{(0)} \|_{Q}$$

Préconditionnement

- Principe
  - Remplacer le système à résoudre par un système équivalent mieux conditionné

$$\mathrm{A}\,\boldsymbol{x}=\boldsymbol{b}\Leftrightarrow \mathrm{C}^{-1}\mathrm{A}\,\boldsymbol{x}=\mathrm{C}^{-1}\boldsymbol{b}$$

  - Idée : uniformiser les valeurs propres

- Ex : Préconditionnement de Jacobi
  - Prendre les éléments diagonaux de A

$$c_{ij}=\begin{cases} A_{ii} & \text{si } i=j \\ 0 & \text{sinon} \end{cases}$$

Formalisation en fonction du résidu r

- Cas scalaire
- Cas vectoriel
- Autre formalisation

Formalisation en fonction du résidu r

$$\mathrm{r}_n(\boldsymbol{v}) = y_n - \mathrm{f}_{\boldsymbol{v}}(\boldsymbol{x}_n)$$

$$\mathrm{e}(\boldsymbol{v}) = \sum_{n=1}^{N} \mathrm{r}_n^2(\boldsymbol{v})$$

$$\mathrm{J}_{\mathbf{r}} = \begin{bmatrix} \partial_{v_1}\mathrm{r}_1 & \partial_{v_2}\mathrm{r}_1 & \cdots & \partial_{v_M}\mathrm{r}_1 \\ \partial_{v_1}\mathrm{r}_2 & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \partial_{v_1}\mathrm{r}_N & \cdots & \cdots & \partial_{v_M}\mathrm{r}_N \end{bmatrix}$$

$$\nabla\mathrm{e} = 2\left(\mathrm{J}_{\mathbf{r}}^{\mathrm{T}}\mathbf{r}\right)$$

Matrice Hessienne

$$\mathrm{r}_n(\boldsymbol{v}) = \mathrm{y}_n - \mathrm{f}_{\boldsymbol{v}}(\boldsymbol{x}_n)$$

$$\mathrm{J}_{\mathbf{r}} = \begin{bmatrix} \partial_{v_1}\mathrm{r}_1 & \partial_{v_2}\mathrm{r}_1 & \cdots & \partial_{v_M}\mathrm{r}_1 \\ \partial_{v_1}\mathrm{r}_2 & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \partial_{v_1}\mathrm{r}_N & \cdots & \cdots & \partial_{v_M}\mathrm{r}_N \end{bmatrix}$$

$$\mathrm{e}(\boldsymbol{v}) = \sum_{n=1}^{N} \mathrm{r}_n^2(\boldsymbol{v})$$

$$\mathrm{H}_{\mathrm{e}} = 2\left(\mathrm{J}_{\mathbf{r}}^{\mathrm{T}}\mathrm{J}_{\mathbf{r}} + \sum_{n=1}^{N} \mathrm{r}_n \mathrm{H}_{\mathrm{r}_n}\right)$$

Méthode de Gauss-Newton

- Remplacer la Hessienne

$$\mathrm{H_e} \simeq \mathbf{J_r}^{\mathrm{T}} \mathbf{J_r} \qquad \boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \left(\mathrm{H_e}\left(\boldsymbol{x}^{(k)}\right)\right)^{-1} \nabla \mathrm{e}\left(\boldsymbol{x}^{(k)}\right)$$

- Avantages
  - Pas de dérivée seconde
  - La matrice est semi-définie positive
- Limitations
  - Approximations valides si petits résidus
  - Le point initial doit être près de la solution finale

Levenberg-Marquardt

- Méthode très couramment utilisée
- A chaque étape, la direction de descente h est

$$\left(\mathrm{J}_{\boldsymbol{r}}^{\mathrm{T}}\mathrm{J}_{\boldsymbol{r}}+\lambda\,\mathrm{diag}\left(\mathrm{J}_{\boldsymbol{r}}^{\mathrm{T}}\mathrm{J}_{\boldsymbol{r}}\right)\right)\boldsymbol{h}=\mathrm{J}_{\boldsymbol{r}}^{\mathrm{T}}\boldsymbol{r}$$

- Le choix de λ conditionne la méthode
  - Petit, on a Newton
  - Grand, on a une méthode de descente rapide
- Meilleur conditionnement au bruit

Résolution itérative de système linéaire

- Rappel: résoudre

$$\mathrm{A}\,\boldsymbol{x}=\boldsymbol{b}$$

- Problèmes
  - Grande dimension : méthodes directes coûteuses
  - Problème de précision
- Principe
  - Reformuler le problème au moindres carrés

$$\min_{\boldsymbol{x}} \|\mathrm{A}\,\boldsymbol{x}-\boldsymbol{b}\|^2=\min_{\boldsymbol{x}} \frac{1}{2}\boldsymbol{x}^{\mathrm{T}} M\,\boldsymbol{x}-\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$$

  - Progressivement améliorer le résultat

Descente de gradient : méthode itérative

- M est symétrique définie positive
- Le gradient en x_i est le résidu
- Pas de descente a tel que

$$\min_{\boldsymbol{x}} \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\mathrm{M}\,\boldsymbol{x}-\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$$

$$\nabla_i=\mathrm{M}\,\boldsymbol{x}_i-\boldsymbol{c}=\boldsymbol{r}_i$$

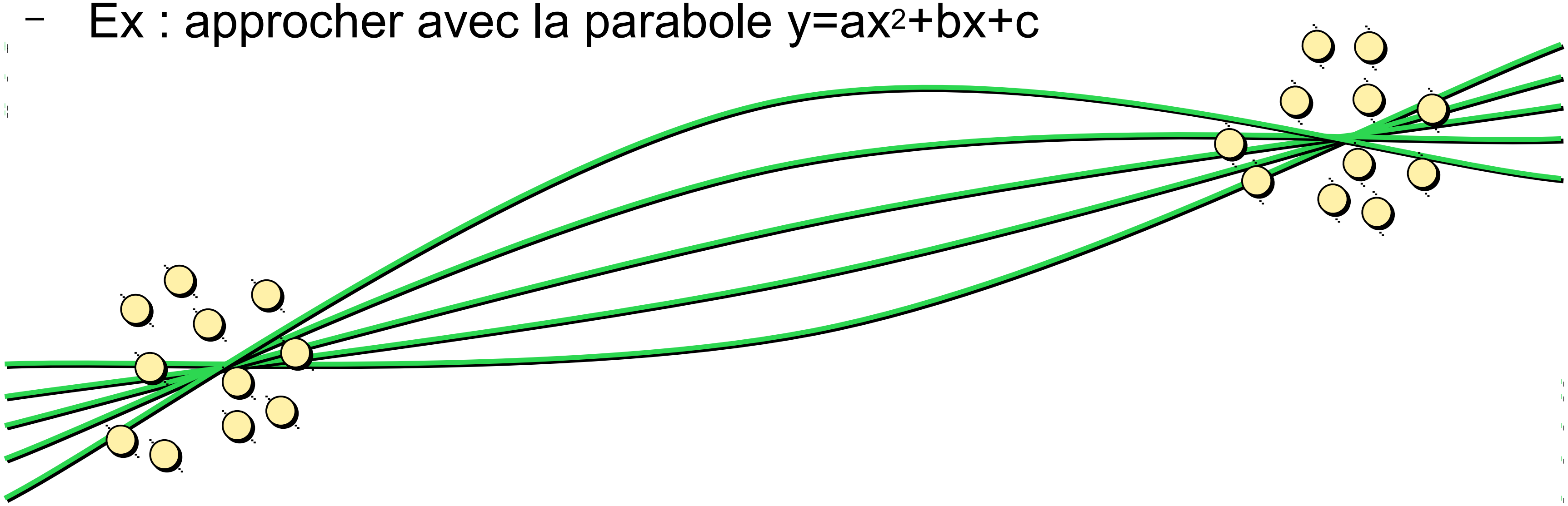
$$\min_{\boldsymbol{x}}\left\|\frac{1}{2}(\boldsymbol{x}_i+a\,\boldsymbol{r}_i)^{\mathrm{T}}\mathrm{M}(\boldsymbol{x}_i+a\,\boldsymbol{r}_i)-\boldsymbol{c}^{\mathrm{T}}(\boldsymbol{x}_i+a\,\boldsymbol{r}_i)\right\|^2$$

$$a=\frac{\boldsymbol{r}_i^{\mathrm{T}}\boldsymbol{r}_i}{\boldsymbol{r}_i^{\mathrm{T}}\mathrm{M}\,\boldsymbol{r}_i}$$

Autres systèmes :
Valeurs propres - SVD

Moindres carrés sous-constraint

- Moins de mesures que de variables
  - Impossible théoriquement d'utiliser les moindres carrés
    - Le système à résoudre a la forme $A^T A x = A^T b$
    - A a plus de colonnes que de lignes
    - $A^T A$ est donc singulière
- Les mesures sont de mauvaises contraintes
  - $A^T A$ est presque singulière : mauvais conditionnement
  - Ex : approcher avec la parabole $y=ax^2+bx+c$



Décomposition singulière SVD

- Soit une matrice A de dimension MxN
  - Il existe 2 matrices orthogonales U (*M*x*M*) et V (*N*x*N*) et une matrice diagonale Σ de dimension *P*= min(*M*,*N*) telle que

$$A = U \Sigma V^T$$

  - Avec

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_P \end{bmatrix} \qquad \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_P \geq 0$$

- Les σ_i sont les valeurs singulières
- Les colonnes de U et V sont les vecteurs singuliers de gauche et de droite

Interpretations

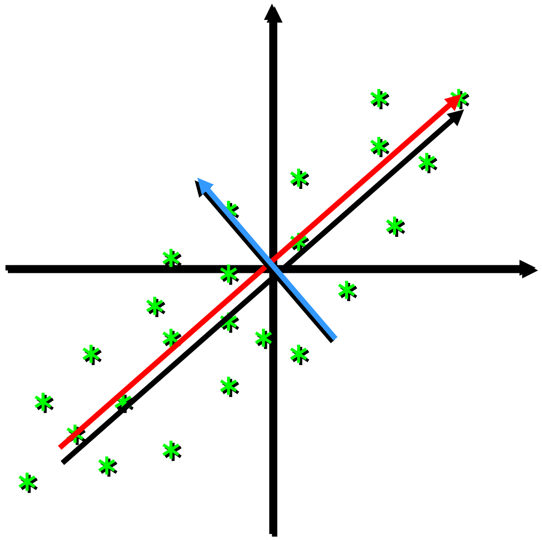
- Lien avec les valeurs propres

$$\Sigma^T \Sigma = (U^T A V)^T U^T A V = V^T A^T U U^T A V = V^T A^T A V$$

  - Donc : les σ_i^2 sont les valeurs propres de A^TA et AA^T

- Demi-longueurs des axes de l'hyperellipse

$$\boldsymbol{x}^T A^T A \boldsymbol{x} - 1 = 0$$



- Conditionnement

$$K(A) = \frac{\sigma_{max}}{\sigma_{min}} \qquad K(A) = \frac{\sigma_1}{\sigma_p}$$

Calcul de pseudo inverse

- On définie la matrice

$$A^{+} = V\,\Sigma^{+}\,U^{T}$$

$$\Sigma^{+} = \begin{bmatrix} \frac{1}{\sigma_1} & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & & & & \vdots \\ \vdots & \ddots & \frac{1}{\sigma_R} & \ddots & & & \vdots \\ \vdots & & \ddots & 0 & \ddots & & \vdots \\ \vdots & & & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & 0 & 0 & 0 \end{bmatrix}$$

- Principes
  - On annule les valeurs inverses trop grandes
  - On laisse libre un certain nombre de paramètres

Applications

- Avantage : méthode numérique stable
  - Coût : $0(n^3)$
  - Plus rapide si l'on ne cherque Σ, U et Σ, ou Σ et V
- Calcul du rang R

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_R > \sigma_{R+1} = \cdots = \sigma_P = 0$$

  - En matlab

```
s=svd(X);
tol=max(size(X))*s(1)*eps;
rang=sum(s > tol);
```

- Approximation par une somme de matrice de rang 1 $A = \sum_{i=1}^{P} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^{\mathrm{T}}$

$$A_K = \sum_{i=1}^{K} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^{\mathrm{T}} \qquad \left\| A - A_K \right\|_2 = \sigma_{K+1}$$

Analyse en composantes principales (PCA)

- Une méthode de réduction de dimension des données
  - Projection au sens des moindres carrés
  - Capturer les principales variations
- Une première approximation
  - Meilleur représentant au sens des moindres carrés $\min_{\boldsymbol{a}} \sum_{m=1}^{M} \|\boldsymbol{y}_m - \boldsymbol{a}\|_2^2$

$$\boldsymbol{a} = \frac{1}{M} \sum_{m=1} \boldsymbol{y}_m$$

  - Quelles sont le ou les directions principales ?
- Représenter les données par une ligne passant par la moyenne

$$\boldsymbol{y}_m = \boldsymbol{a} + \alpha_m \boldsymbol{e}$$

$$\min_{\alpha_1, \cdots, \alpha_M, \boldsymbol{e}} \sum_{m=1}^{M} \|\boldsymbol{y}_m - \boldsymbol{a} - \alpha_m \boldsymbol{e}\|_2^2$$

PCA – Intuitions (suite)

- Calculer les α_m par différentiation partielle

$$\alpha_m = \boldsymbol{e}^{\mathrm{T}}(\boldsymbol{y}_m - \boldsymbol{a})$$

- Nouvelle erreur pour calculer la direction principale

$$\sum_{m=1}^{M} \|\boldsymbol{y}_m - \boldsymbol{a} - \alpha_m \boldsymbol{e}\|_2^2 = \sum_{m=1}^{M} \alpha_m^2 - 2\sum_{m=1}^{M} \alpha_m \boldsymbol{e}^{\mathrm{T}}(\boldsymbol{y}_m - \boldsymbol{a}) + \sum_{m=1}^{M} \|\boldsymbol{y}_m - \boldsymbol{a}\|^2$$

$$= -\sum_{m=1}^{M} \boldsymbol{e}^{\mathrm{T}}(\boldsymbol{y}_m - \boldsymbol{a})(\boldsymbol{y}_m - \boldsymbol{a})^{\mathrm{T}} \boldsymbol{e} + \sum_{m=1}^{M} \|\boldsymbol{y}_m - \boldsymbol{a}\|^2$$

$$= -\boldsymbol{e}^{\mathrm{T}} \mathrm{S}\, \boldsymbol{e} + \sum_{m=1}^{M} \|\boldsymbol{y}_m - \boldsymbol{a}\|^2$$

- Matrice de covariance (scatter matrix)

$$\mathrm{S} = \sum_{m=1}^{M} (\boldsymbol{y}_m - \boldsymbol{a})(\boldsymbol{y}_m - \boldsymbol{a})^{\mathrm{T}}$$

- Minimisation sous contrainte

$$\max_{\boldsymbol{e}} \boldsymbol{e}^{\mathrm{T}} \mathrm{S}\, \boldsymbol{e}$$

$$\|\boldsymbol{e}\|^2 = 1$$

PCA – Intuitions (fin)

- Utilisation des multiplicateurs de Lagrange

$$e^T S e - \lambda \left(e^T e - 1 \right)$$

- Par dérivation

$$S e = 2 \lambda e$$

  - **e** est un vecteur propre de la matrice S
- Peut-être étendue à *M* directions principales
  - Calcul d'une SVD
  - Amplitude de variation= valeur propre
  - Direction de variation= vecteur propre