Rappels et précisions, point du 21 décembre

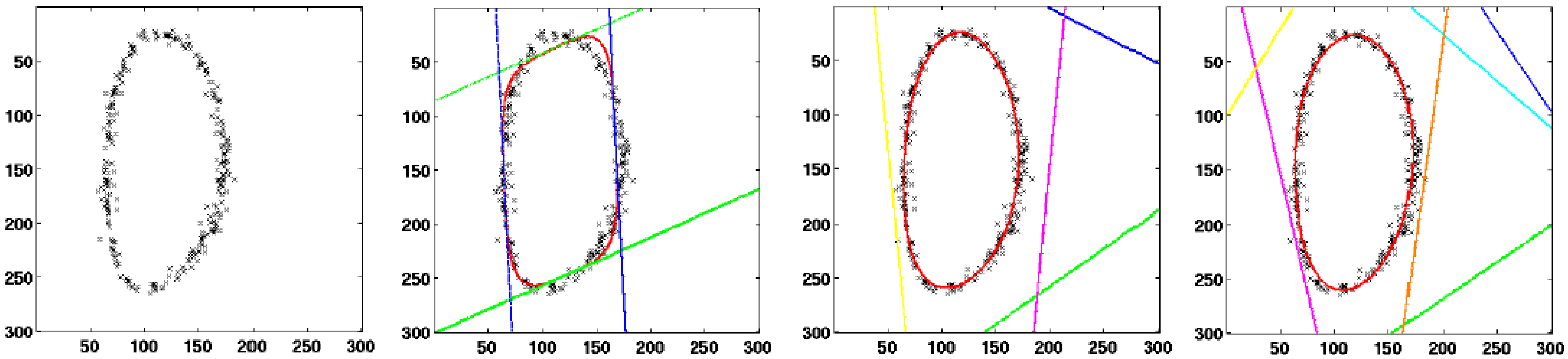
Présentation du projet : Reconnaissance d'hyperquadriques

MO4MEM08 : CALCUL SCIENTIFIQUE – 3 ECTS

RESPONSABLE D'UE ET COURS : FLORENCE OSSART

But du projet :

→ Déterminer la meilleure HQ (contour rouge ci-dessous) passant par un nuage de points (le nuage de points représente la frontière d'une cavité)



(a) Scattered data (b) Fit result with $N_h = 2$ (c) Fit result with $N_h = 4$ (d) Fit result with $N_h = 6$

Figure extraite de « Segmentation of 2D-echocardiographic sequences using level-set constrained with shape and motion priors », thèse de doctorat de T. Dietenbeck, Lyon 2012

Contour défini par une hyperquadrique

→ Une hyperquadrique (HQ) est une fonction définie par :

$$\varphi(x,y,\lambda) = \sum_{k=1}^{Nh} |A_k.x + B_k.y + C_k|^{\gamma_k}$$

Où Nh est le nombre de termes de la l'hyperquadrique HQ,

et $\lambda = \{A_k, B_k, C_k, \gamma_k, \quad \forall k = 1, Nh\}$ est le vecteur des paramètres de HQ, avec $\gamma_k > 0$

L'équation implicite $\psi(x,y,\lambda) = \varphi(x,y,\lambda) - 1 = 0$ définit le contour fermé qui nous intéresse.

Etapes du projet

1. Comprendre ce qu'est une hyperquadrique : visualiser ces fonctions et jouer avec

2. Fitter un nuage de points par une HQ dont une partie des paramètres est donnée.
   - Variables de décision : a et b, 2 des 8 paramètres de l'HQ à fitter sur le nuage de points
   - Fonction-objectif à minimiser : distance entre les points du nuage et l'HQ à paramètres a et b donnés
   - Méthodes mises en œuvre : descente de gradient et Newton

3. Fitter un nuage de points par une HQ dont seuls les exposants sont connus.
   - Contraintes : intervalles d'appartenance des paramètres de l'HQ
   - Contraintes prises en compte par un terme de pénalité rajouté à la fonction-objectif => on construit un problème de minimisation sans contraintes
   - Méthode de résolution : algorithme de Levenberq-Marquardt, basé sur une approximation

Phase 1 : Visualiser une HQ de paramètres donnés

1. 1 action (visualiser) => 1 fonction python

2. Données de travail => paramètres d'entrée de la fonction

3. HQ de paramètres λ donnés => λ est un paramètre d'entrée

   - Il faut donc écrire une fonction qui ressemble à quelque chose comme:

```
def HQ_trace(lambdas, …) :
     ……
```

   - λ : paramètres, peuvent être organisés sous la forme d'un tableau 2d

```
lambdas = [[a1, b1, c1, g1], [[a2, b2, c2, g2], … ]
```

Phase 1 : Visualiser une HQ de paramètres donnés

4. Comment faire le tracé ?
   - Le plus simple est de tracer l'isovaleur de niveau 0 de la fonction $\psi(x, y, \lambda)$
   - Donc, on va avoir besoin d'écrire une fonction qui calcule $\psi(x, y, \lambda)$
   - Il faut aussi déterminer les droites englobantes.
   - NB : 1 droite est définie par 2 points.
5. Sur quel domaine faire le tracé ?
   - Peut être choisi par l'utilisateur (c'est alors un paramètre d'entrée de la fonction **`HQ_trace`**)
   - Peut être calculé automatiquement à partir des droites englobantes
   - En général, besoin de quelques essais-erreurs pour adapter le domaine de tracé de l'HQ

Phase 1 : Visualiser une HQ de paramètres donnés

→ Discussion

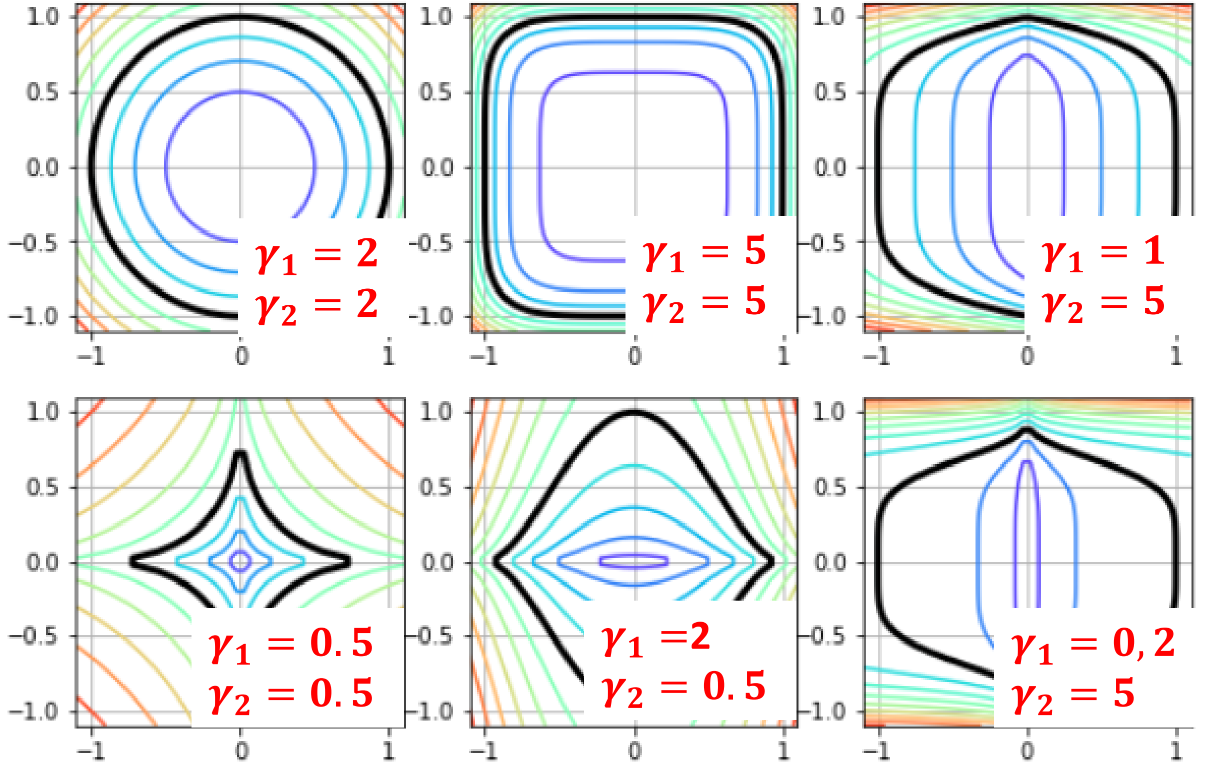
Exemples de HQ avec Nh=2

→ $\varphi(x,y,\lambda) = |A_1.x + B_1.y + C_1|^{\gamma_1} + |A_2.x + B_2.y + C_2|^{\gamma_2}$

Cas particulier :

$\varphi(x,y,\lambda) = |x|^{\gamma_1} + |y|^{\gamma_2}$

Contour noir : $\varphi(x,y,\lambda) - 1 = 0$



Droites enveloppantes

→ Contour définie par $\varphi(x, y, \lambda) = \sum_{k=1}^{Nh} |A_k.x + B_k.y + C_k|^{\gamma_k} = 1$ et $\gamma_k > 0$

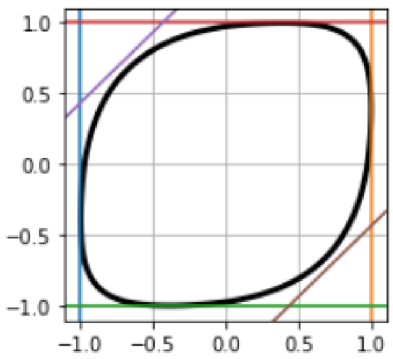
→ donc $\forall k, |A_k.x + B_k.y + C_k| \leq 1$, d'où $-1 \leq A_k.x + B_k.y + C_k \leq 1$

→ si : $B_k \neq 0$:

$$-\frac{A_k}{B_k}.x - \frac{C_k}{B_k} - \frac{1}{B_k} \leq y \leq -\frac{A_k}{B_k}.x - \frac{C_k}{B_k} + \frac{1}{B_k}$$

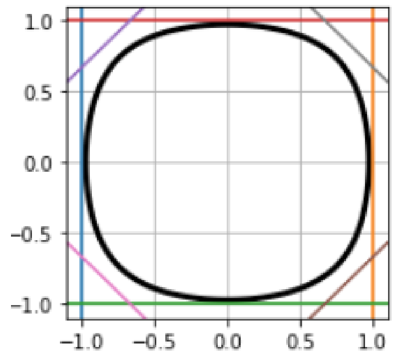
→ si : $B_k = 0$:

$$-\frac{C_k}{A_k} - \frac{1}{A_k} \leq x \leq -\frac{C_k}{A_k} + \frac{1}{A_k}$$




```
Hyper-quadrique avec 3 termes
1. a, b, c = 1 / 0 / 0 - gamma = 5
2. a, b, c = 0 / 1 / 0 - gamma = 5
3. a, b, c = 0.7 / -0.7 / 0 - gamma = 5
```

```
Hyper-quadrique avec 4 termes
1. a, b, c = 1 / 0 / 0 - gamma = 5
2. a, b, c = 0 / 1 / 0 - gamma = 5
3. a, b, c = 0.6 / -0.6 / 0 - gamma = 5
4. a, b, c = 0.6 / 0.6 / 0 - gamma = 5
```



Phase 2 : Fitter un nuage de points par une HQ dont une partie des paramètres est donnée

→ Données à fitter :

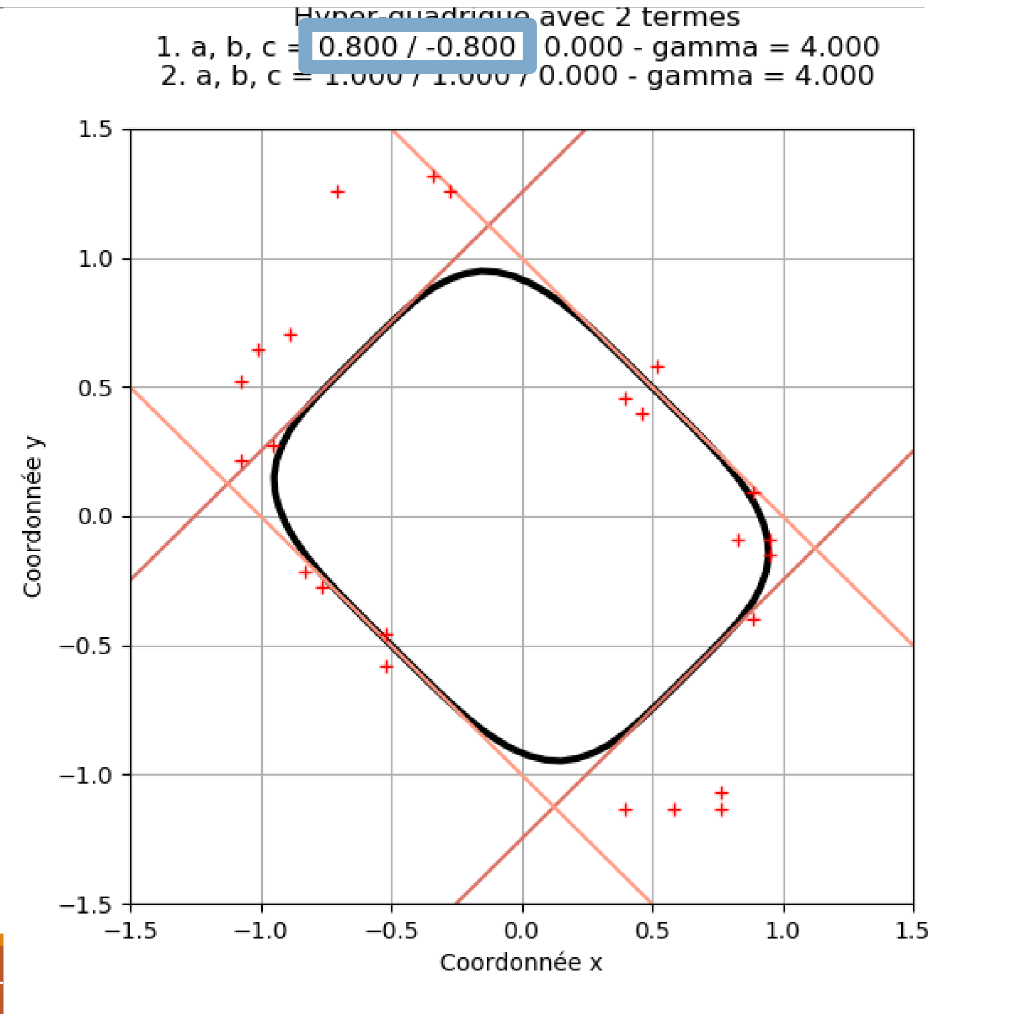
- Nuage de points : $\mathbf{data} = \{(\boldsymbol{x_i}, \boldsymbol{y_i})\}_{\boldsymbol{i=1,N}}$.

→ Fonction utilisée pour le fit :

- $\boldsymbol{\psi}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{a}, \boldsymbol{b}) = (\boldsymbol{a}\ \boldsymbol{x} + \boldsymbol{b}\ \boldsymbol{y})^4 + (\boldsymbol{x} + \boldsymbol{y})^4 - \mathbf{1}$

→ Problème à résoudre :

- $(\boldsymbol{a}, \boldsymbol{b})^* = \underset{\boldsymbol{a,b}}{\boldsymbol{argmin}}\, \boldsymbol{J_{data}}(\boldsymbol{a}, \boldsymbol{b})$,
- où $\boldsymbol{J_{data}}(\boldsymbol{a}, \boldsymbol{b}) = \sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} [\boldsymbol{\psi_i}(\boldsymbol{a}, \boldsymbol{b})]^2$
- et $\boldsymbol{\psi_i}(\boldsymbol{a}, \boldsymbol{b}) = \boldsymbol{\psi}(\boldsymbol{x_i}, \boldsymbol{y_i}, \boldsymbol{a}, \boldsymbol{b})$



Phase 2 : Fitter un nuage de points par une HQ dont une partie des paramètres est donnée

→ Données à fitter :

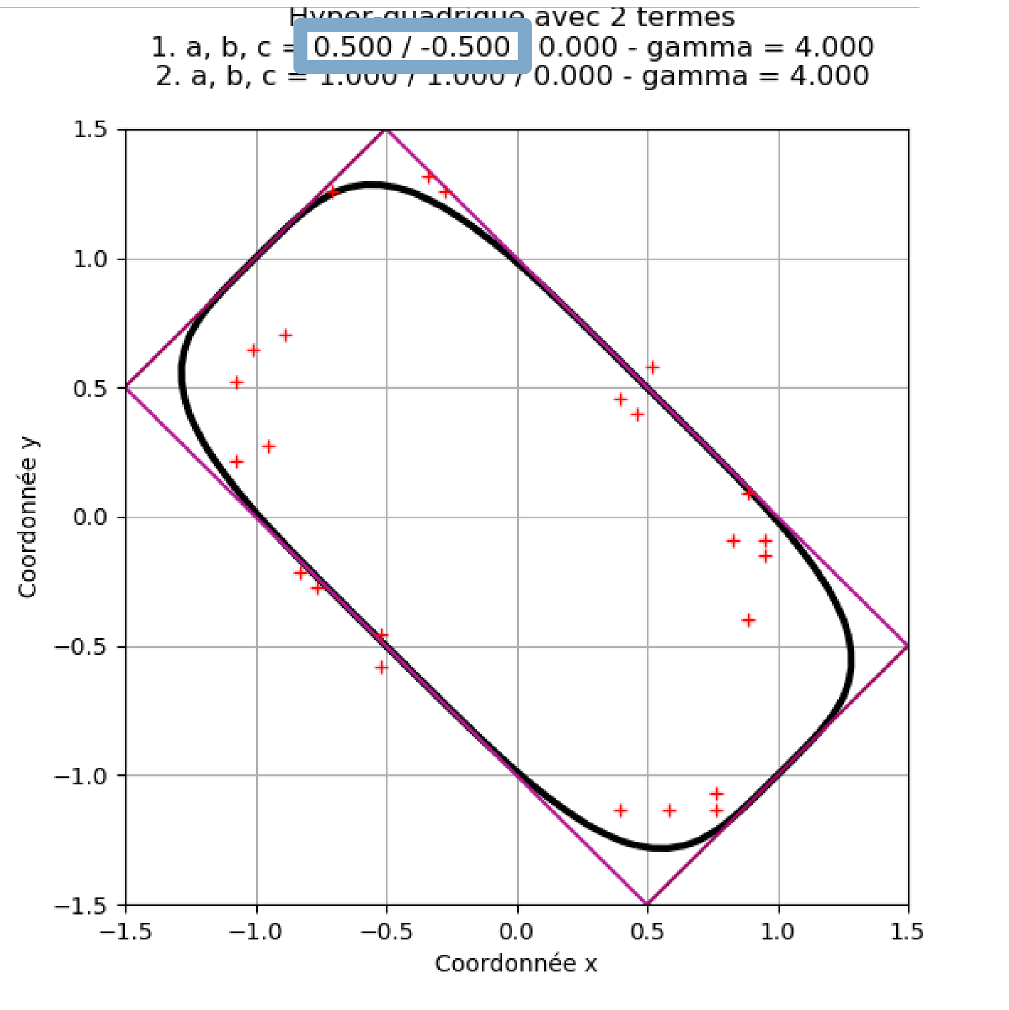
- Nuage de points : $\mathbf{data} = \{(x_i, y_i)\}_{i=1,N}$.

→ Fonction utilisée pour le fit :

- $\psi(x, y, a, b) = (a\,x + b\,y)^4 + (x + y)^4 - 1$

→ Problème à résoudre :

- $(a, b)^* = \underset{a,b}{\mathbf{argmin}}\, J_{data}(a, b),$
- où $J_{data}(a, b) = \sum_{i=1}^{N} [\psi_i(a, b)]^2$
- et $\psi_i(a, b) = \psi(x_i, y_i, a, b)$



Phase 2 : Fitter un nuage de points par une HQ dont une partie des paramètres est donnée

→ Données à fitter :

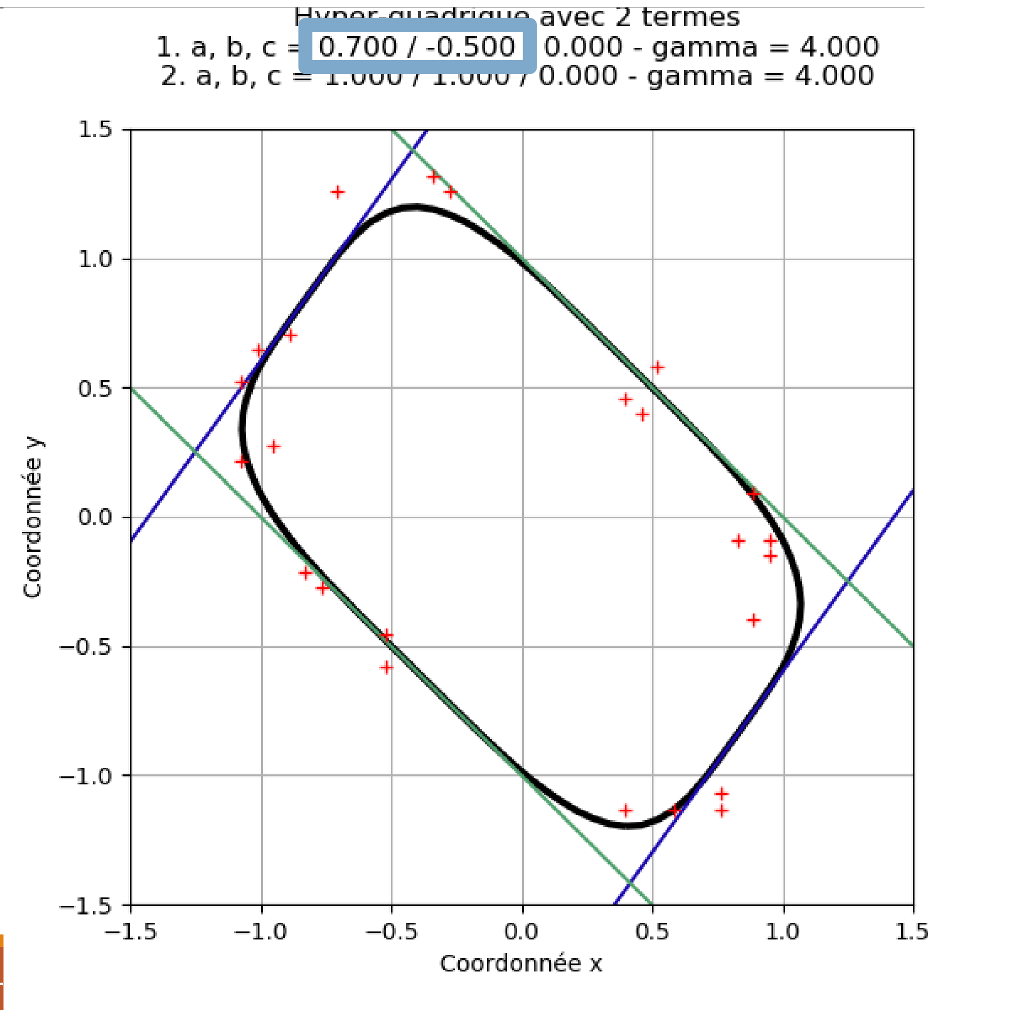
- Nuage de points : $\mathbf{data} = \{(x_i, y_i)\}_{i=1,N}$.

→ Fonction utilisée pour le fit :

- $\psi(x, y, a, b) = (a\,x + b\,y)^4 + (x + y)^4 - 1$

→ Problème à résoudre :

- $(a, b)^* = \underset{a,b}{\mathbf{argmin}}\, J_{data}(a, b)$,
- où $J_{data}(a, b) = \sum_{i=1}^{N} [\psi_i(a, b)]^2$
- et $\psi_i(a, b) = \psi(x_i, y_i, a, b)$



Phase 2 : Fitter un nuage de points par une HQ dont une partie des paramètres est donnée

→ Problème à résoudre : minimiser $\boldsymbol{J_{data}(a,b)} = \sum_{i=1}^{N}[\boldsymbol{\psi_i(a,b)}]^2$

→ Méthode 1 : descente de gradient

- S'appuie sur $\nabla \mathbf{J}(a,b) = \begin{bmatrix} \frac{\partial J}{\partial a}(a,b) \\ \frac{\partial J}{\partial b}(a,b) \end{bmatrix}$ <u>à calculer à la main</u> !!! (pas de dérivation automatique)
- À chaque itération : $(a_n, b_n) \leftarrow (a_{n-1}, b_{n-1}) - \alpha . \nabla J(a_{n-1}, b_{n-1})$
- Point délicat : choix du coefficient α
  - ajustement par essais-erreurs
  - il existe des algorithmes pour ajuster la valeur de α au cours des itérations

Phase 2 : Fitter un nuage de points par une HQ dont une partie des paramètres est donnée

→ Problème à résoudre : minimiser $\boldsymbol{J_{data}}(\boldsymbol{a}, \boldsymbol{b}) = \sum_{i=1}^{N}[\boldsymbol{\psi_i}(\boldsymbol{a}, \boldsymbol{b})]^2$

→ Méthode 2 : méthode de Newton

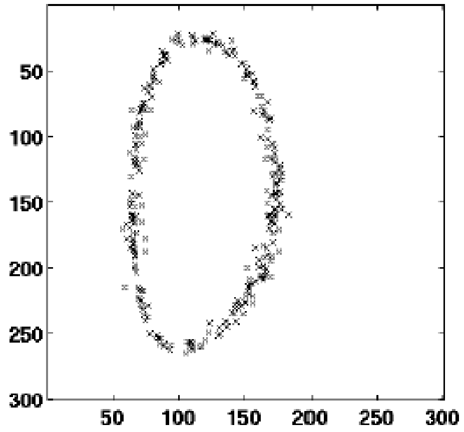
- S'appuie sur $\nabla \mathbf{J}(a,b) = \begin{bmatrix} \frac{\partial J}{\partial a}(a,b) \\ \frac{\partial J}{\partial b}(a,b) \end{bmatrix}$ et $H_J(a,b) = \begin{bmatrix} \frac{\partial^2 J}{\partial a^2}(a,b) & \frac{\partial^2 J}{\partial a \partial b}(a,b) \\ \frac{\partial^2 J}{\partial b \partial a}(a,b) & \frac{\partial^2 J}{\partial b^2}(a,b) \end{bmatrix}$ <u>à calculer à la main</u> !!!
- À chaque itération : $(a_n, b_n) \leftarrow (a_{n-1}, b_{n-1}) + (\Delta a_{n-1}, \Delta b_{n-1})$

où $(\Delta a_{n-1}, \Delta b_{n-1})$ est solution du système d'équations $H_J(a_{n-1}, b_{n-1}).\begin{pmatrix} \Delta a_{n-1} \\ \Delta b_{n-1} \end{pmatrix} = -\nabla J(a_{n-1}, b_{n-1})$

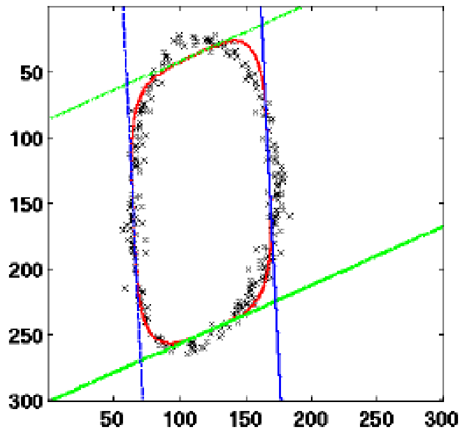
- Point délicat : l'algorithme peut diverger si le point initial (a_0, b_0) est mal choisi
  - ➢ On peut limiter $\begin{pmatrix} \Delta a_{n-1} \\ \Delta b_{n-1} \end{pmatrix}$: $(a_n, b_n) \leftarrow (a_{n-1}, b_{n-1}) + \alpha\ (\Delta a_{n-1}, \Delta b_{n-1})$
  - ➢ Il existe des algorithmes pour ajuster la valeur de α au cours des itérations

Phase 3 : Fitter un nuage de points par une HQ dont seuls les exposants sont connus

- Fitter un nuage de points par une HQ dont seuls les exposants sont connus.
  - Contraintes : intervalles d'appartenance des paramètres de l'HQ
  - Contraintes prises en compte par un terme de pénalité rajouté à la fonction-objectif => on construit un problème de minimisation sans contraintes
  - Méthode de résolution : algorithme de Levenberq-Marquardt, basé sur une approximation de $H_J(a,b)$ et un ajustement du pas de déplacement au cours des itérations.



(a) Scattered data



(b) Fit result with $N_h = 2$