

1 - Description du problème

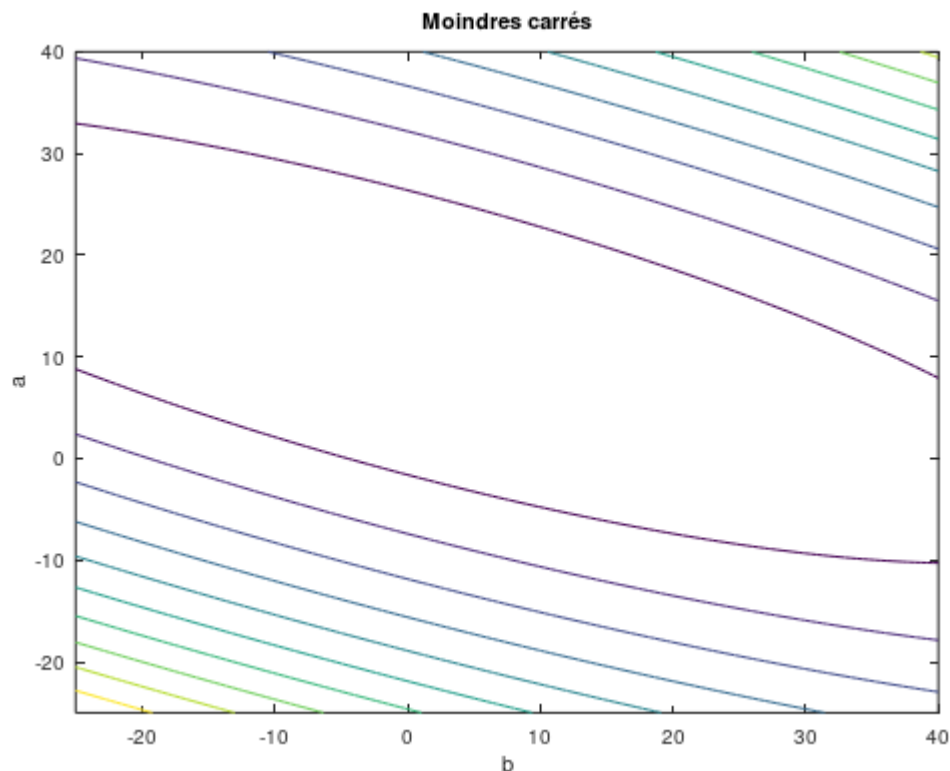
2 – Estimation au sens des moindres carrés

$$\hat{\theta} = \begin{pmatrix} \hat{a} \\ \hat{b} \end{pmatrix} = \min_{a,b} \arg \sum_{i=1}^n (ax_i + b - y_i)^2$$

$$C = \sum_{i=1}^n (ax_i + b - y_i)^2$$

Le but de cette partie est de déterminer le meilleur estimateur possible des moindres carrés.

On commence par représenter la fonction des moindres carrés dans un espace à deux dimensions pour pouvoir dégrossir l'espace de a et b . En effet, on cherche le minimum de C , ainsi à l'œil nue on peut trouver un intervalle de a et b encadrant ce minimum.

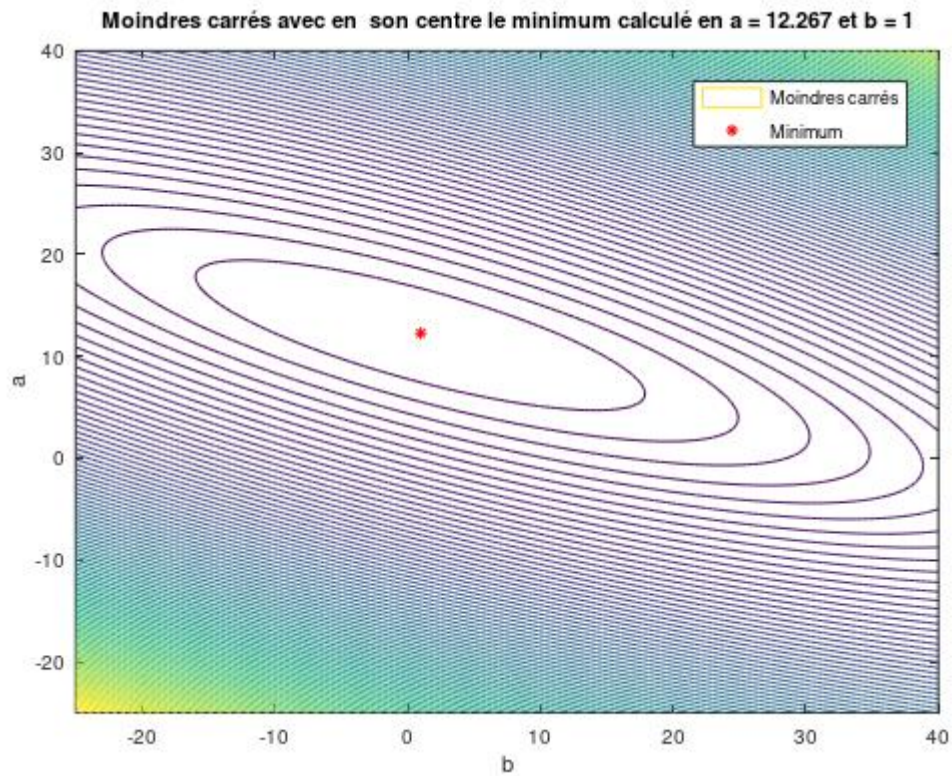


Représentation de la fonction des moindres carrés pour $a = b = [-25; 40]$

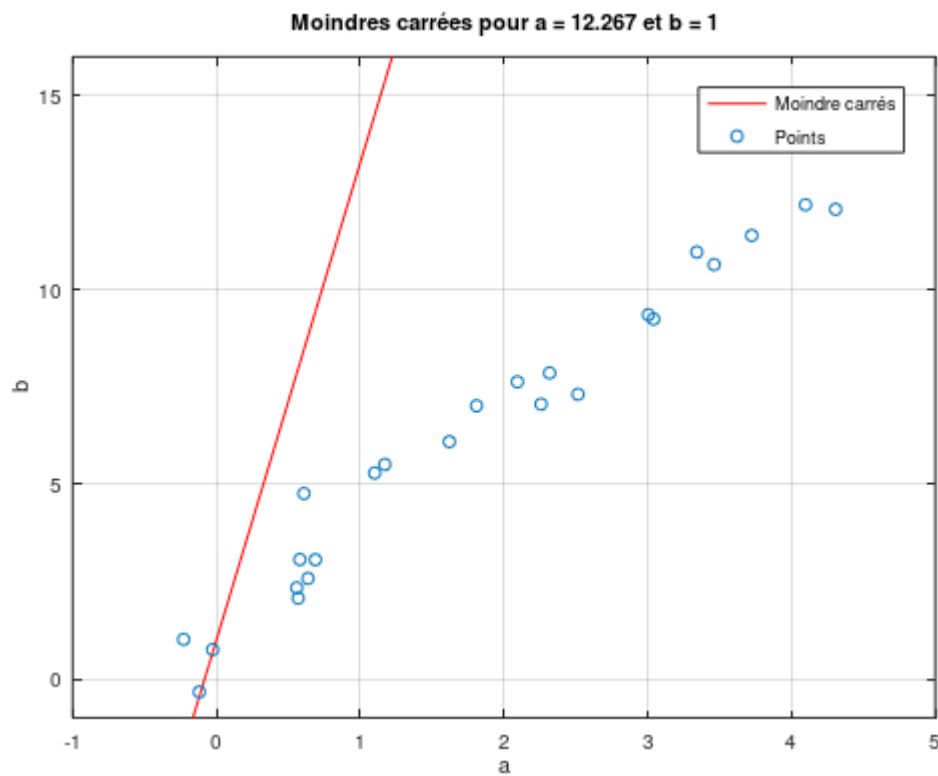
On remarque que le minimum se situe sur l'image ci-dessus pour a compris entre 10 et 20 et b compris entre 0 et 10.

Grâce à la matrice des moindres carrés construite précédemment, on va déterminer le minimum par le calcul pour trouver $\hat{\theta} = \begin{pmatrix} \hat{a} \\ \hat{b} \end{pmatrix}$, point minimum compris dans la matrice des moindres carrés.

Par le calcul, on trouve un point minimum en $\hat{\theta} = \begin{pmatrix} 12.267 \\ 1 \end{pmatrix}$.



Fonction des moindres carrés avec le minimum calculé (12.267 1) symbolisé par une étoile rouge



Représentation de la fonction des moindres carrés pour la solution précédente, $y = 12.267x + 1$

On remarque grâce à cette courbe que l'approximation n'est pas très proche de ce que l'on souhaite obtenir. En effet, cette approximation est très influencée par la présence du bruit, et l'approximation dévie beaucoup. Cette approximation n'est pas très approximative.

La détermination de $\hat{\theta}$ n'est pas très précise, ainsi donc, on passe par l'expression analytique de ce problème quadratique.

Expression analytique :

$$f\left(\begin{pmatrix} a \\ b \end{pmatrix}\right) = \sum_{i=1}^n (ax_i + b - y_i)^2 = \left(\begin{pmatrix} x_1 & 1 \\ \dots & \dots \\ x_n & 1 \end{pmatrix} * \begin{pmatrix} a \\ b \end{pmatrix} - \begin{pmatrix} y_1 \\ \dots \\ y_n \end{pmatrix} \right)^2 = (X * AB - Y)^2$$

$$\text{avec } AB = \begin{pmatrix} a \\ b \end{pmatrix}, X = \begin{pmatrix} x_1 & 1 \\ \dots & \dots \\ x_n & 1 \end{pmatrix} \text{ et } Y = \begin{pmatrix} y_1 \\ \dots \\ y_n \end{pmatrix}$$

On cherche le minimum de cette fonction. Par définition, le minimum d'une fonction est atteint lorsque son gradient est nul. On dérive donc la fonction f par AB .

$$\nabla f(AB) = 2X^T(X * AB - Y)$$

On multiplie par la transposée de X pour l'homogénéité de la fonction.

On cherche $\nabla f(AB) = 0$:

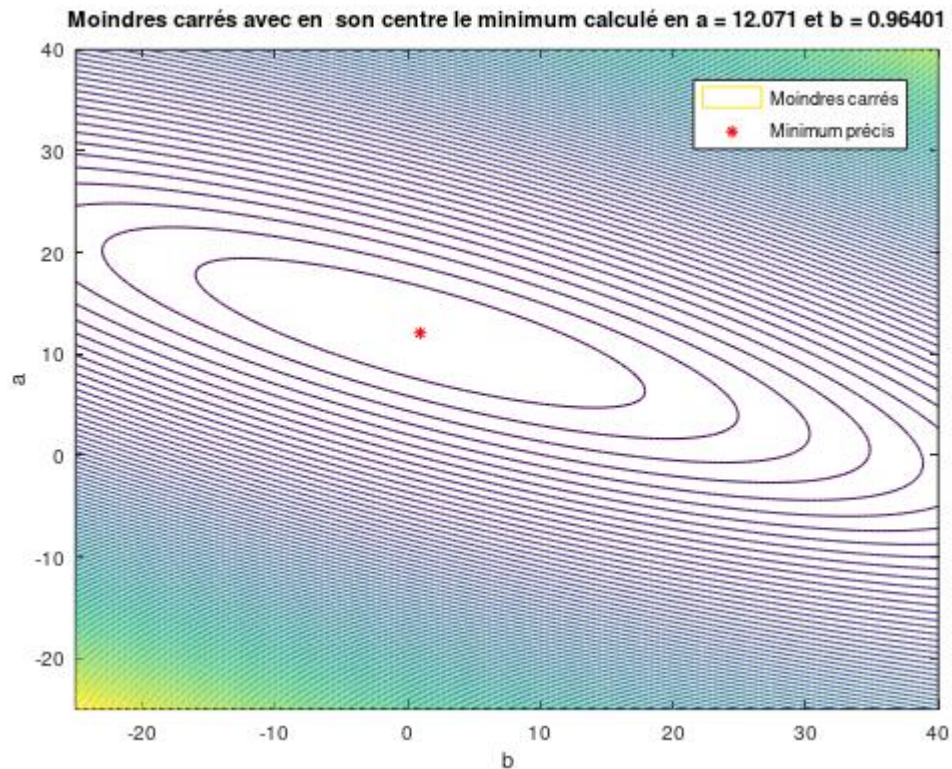
$$2X^T(X * AB - Y) = 0$$

$$2X^T X * AB = 2X^T Y$$

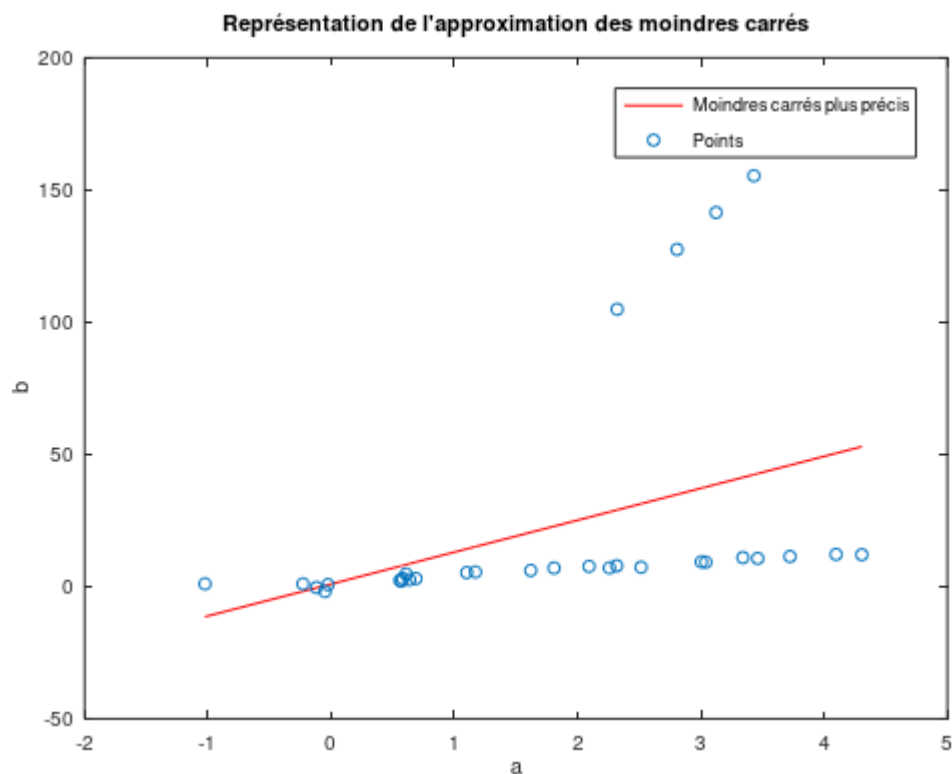
$$AB = (X^T X)^{-1} X^T Y$$

$$\begin{pmatrix} \hat{a} \\ \hat{b} \end{pmatrix} = \left(\begin{pmatrix} x_1 & \dots & x_n \\ 1 & \dots & 1 \end{pmatrix} * \begin{pmatrix} x_1 & 1 \\ \dots & \dots \\ x_n & 1 \end{pmatrix} \right)^{-1} * \begin{pmatrix} x_1 & 1 \\ \dots & \dots \\ x_n & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ \dots \\ y_n \end{pmatrix}$$

On trouve par la suite $\hat{\theta} = \begin{pmatrix} 12.071 \\ 0.96401 \end{pmatrix}$, qui est assez proche du résultat trouvé par le problème initial.



Fonction des moindres carrés avec le minimum précis symbolisé par une étoile rouge



Représentation de l'approximation des moindres carrés plus précise $y = 12.071x + 0.96401$

On remarque que passer par l'analyse quadratique du problème rend une meilleure approximation. Ce constat est fait notamment puisque sur le schéma ci-dessus, la courbe des moindres carrés est un compromis entre les points recherchés et le bruit, sans que cette courbe ne soit très fortement

influencée par le bruit. En passant par la méthode quadratique, nous avons trouvé un résultat en meilleur accord avec les valeurs recherchées et en atténuant l'influence du bruit.

Globalement, l'estimation des moindres carrés cherche à minimiser un résidu quadratique trop fort, ainsi l'approximation ne peut être que mauvaise dans notre cas.

On va donc essayer de résoudre le problème en passant par une estimation robuste, qui elle va se caler sur les points les plus groupés entre eux, en minimisant le résidu.

3 – Estimation robuste

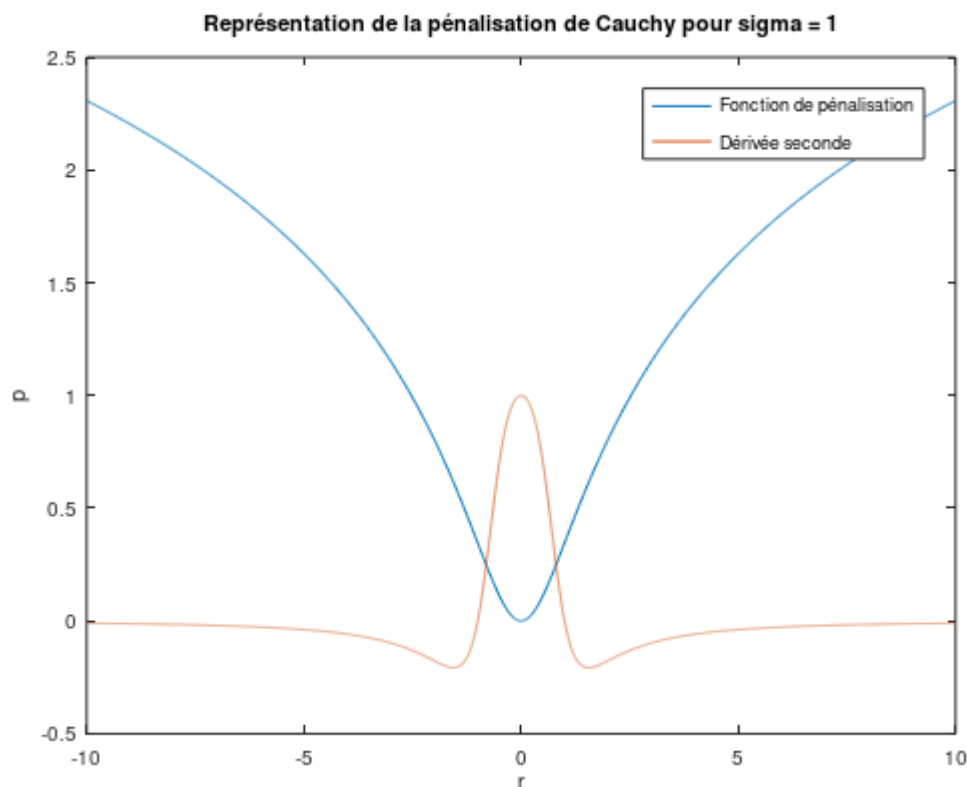
Pour pouvoir améliorer l'estimation de la droite, on passe par l'estimation robuste.

$$\hat{\theta} = (\hat{a}, \hat{b}) = \arg \min_{a,b} \sum_{i=1}^n \rho(r_i), \text{ avec } r_i = ax_i + b - y_i$$

Pénalisation de Cauchy :

$$\forall r \in \mathbb{R}, \quad \rho_{\sigma}(r) = \frac{1}{2} \log \left(1 + \left(\frac{r}{\sigma} \right)^2 \right)$$

On cherche maintenant à représenter la pénalisation de Cauchy avec $\sigma = 1$, pour avoir une idée de la forme de cette fonction.



Représentation de la pénalisation de Cauchy avec $\sigma = 1$

On remarque que la pénalisation de Cauchy est décroissante jusqu'à 0 pour $r \in \mathbb{R}^-$. De même, cette fonction est croissante pour $r \in \mathbb{R}^+$.

On a cherché à savoir si cette fonction de pénalisation est convexe ou non. On est donc par la dérivée seconde de la pénalisation de Cauchy.

Par définition, la pénalisation de Cauchy est convexe sur U , si et seulement si $\frac{\partial^2 \rho_\sigma(r)}{\partial r^2} \geq 0$ sur U .

Par définition, la pénalisation de Cauchy est concave sur U , si et seulement si $\frac{\partial^2 \rho_\sigma(r)}{\partial r^2} \leq 0$ sur U .

Calcul de $\frac{\partial \rho_\sigma(r)}{\partial r}$:

$$\frac{\partial \rho_\sigma(r)}{\partial r} = \frac{1}{2} \left(\frac{\frac{2r}{\sigma^2}}{1 + \left(\frac{r}{\sigma}\right)^2} \right)$$

$$\frac{\partial \rho_\sigma(r)}{\partial r} = \frac{r}{\sigma^2 + r^2}$$

Calcul de $\frac{\partial^2 \rho_\sigma(r)}{\partial r^2}$:

$$\frac{\partial^2 \rho_\sigma(r)}{\partial r^2} = \frac{\sigma^2 + r^2 - 2r^2}{(\sigma^2 + r^2)^2}$$

$$\frac{\partial^2 \rho_\sigma(r)}{\partial r^2} = \frac{(\sigma^2 - r^2)}{(\sigma^2 + r^2)^2}$$

Pour $\sigma = 1$:

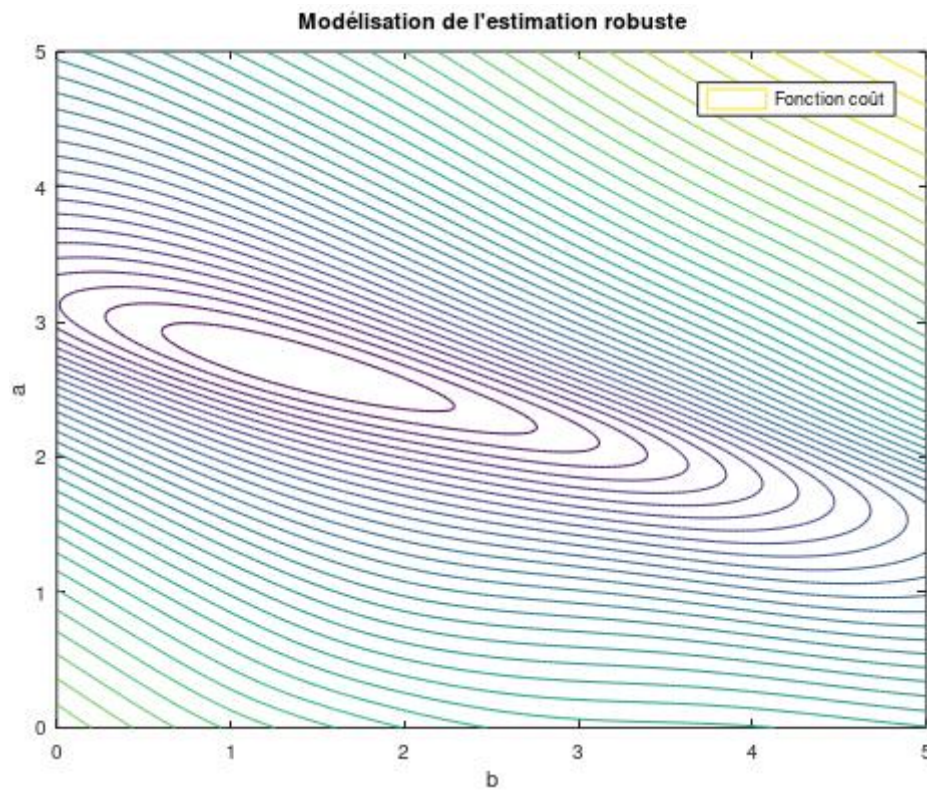
$$\nabla^2 \rho_1(r) = \frac{\partial^2 \rho_1(r)}{\partial r^2} = \frac{1 - r^2}{(1 + r^2)^2}$$

Par analyse numérique, on remarque $\nabla^2 \rho_1(r) \geq 0, \forall r \in [-1; 1]$.

Ainsi $\forall r \in [-1; 1]$, la pénalisation de Cauchy est convexe et concave en dehors de cet intervalle pour $\sigma = 1$.

On en déduit donc que cette fonction possède un minimum global. On va donc maintenant s'intéresser à ce minimum global. Pour obtenir la meilleure estimation possible, il faut que les résidus soient le plus proches de 0. Ainsi, on cherche $\begin{pmatrix} a \\ b \end{pmatrix}$, tel que $r = \begin{pmatrix} x_1 & 1 \\ \dots & \dots \\ x_n & 1 \end{pmatrix} * \begin{pmatrix} a \\ b \end{pmatrix} - \begin{pmatrix} y_1 \\ \dots \\ y_n \end{pmatrix}$ soit le minimum possible.

On passe maintenant à l'analyse de la fonction coût :



Fonction coût de l'estimation robuste

On remarque grâce à cette représentation que le minimum se situe en $\begin{pmatrix} 2.5 \\ 1.5 \end{pmatrix}$.

On calcul à présent le gradient de la fonction de coût, pour trouver les zones dans lesquelles le minimum est atteint. Celui-ci est atteint lorsque le gradient est nul.

Calcul du gradient de la fonction coût :

$$C_{\sigma} \left(\begin{pmatrix} a \\ b \end{pmatrix} \right) = \sum_{i=1}^n \rho_{\sigma}(r_i), \quad AB = \begin{pmatrix} a \\ b \end{pmatrix}$$

$$\nabla C_{\sigma} \left(\begin{pmatrix} a \\ b \end{pmatrix} \right) = \frac{\partial C_{\sigma} \left(\begin{pmatrix} a \\ b \end{pmatrix} \right)}{\partial AB} = \frac{\partial}{\partial AB} \sum_{i=1}^n \rho_{\sigma}(r_i) = \sum_{i=1}^n \frac{\partial \rho_{\sigma}(r_i)}{\partial AB}$$

$$\nabla C_{\sigma} \left(\begin{pmatrix} a \\ b \end{pmatrix} \right) = \begin{pmatrix} \sum_{i=1}^n \frac{\partial \rho_{\sigma}(r_i)}{\partial a} \\ \sum_{i=1}^n \frac{\partial \rho_{\sigma}(r_i)}{\partial b} \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^n \frac{\partial \rho_{\sigma}(r_i)}{\partial r_i} \times \frac{\partial r_i(a; b)}{\partial a} \\ \sum_{i=1}^n \frac{\partial \rho_{\sigma}(r_i)}{\partial r_i} \times \frac{\partial r_i(a; b)}{\partial b} \end{pmatrix}$$

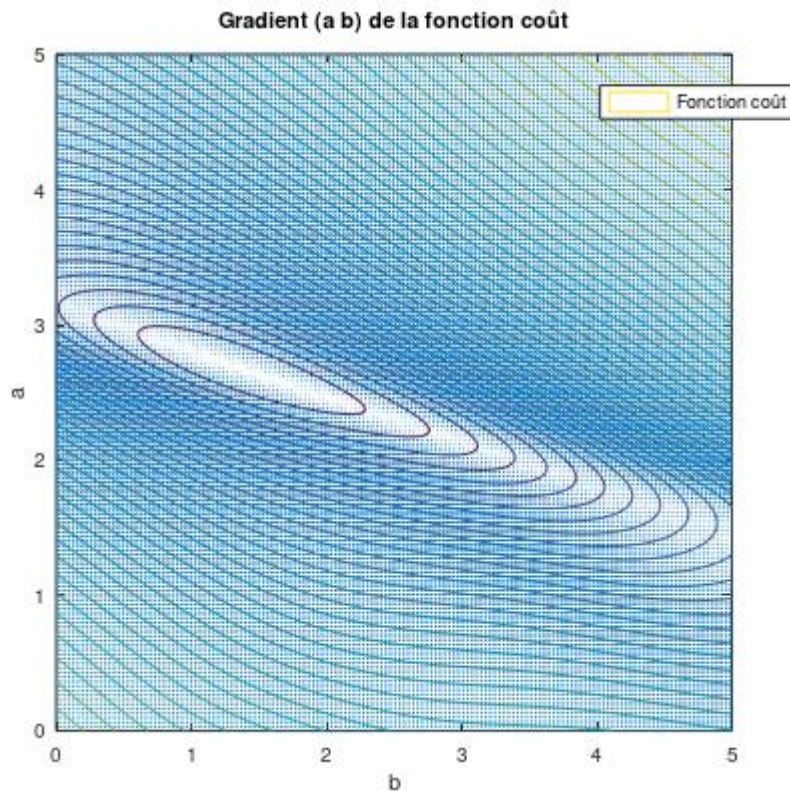
$$\frac{\partial \rho_{\sigma}(r_i)}{\partial r_i} = \frac{r_i}{\sigma^2 + r_i^2}, \quad \frac{\partial r_i(a; b)}{\partial a} = x_i, \quad \frac{\partial r_i(a; b)}{\partial b} = 1$$

$$\nabla C_{\sigma} \left(\begin{pmatrix} a \\ b \end{pmatrix} \right) = \begin{pmatrix} \sum_{i=1}^n \frac{x_i r_i}{\sigma^2 + r_i^2} \\ \sum_{i=1}^n \frac{r_i}{\sigma^2 + r_i^2} \end{pmatrix}$$

Pour $\sigma = 1$,

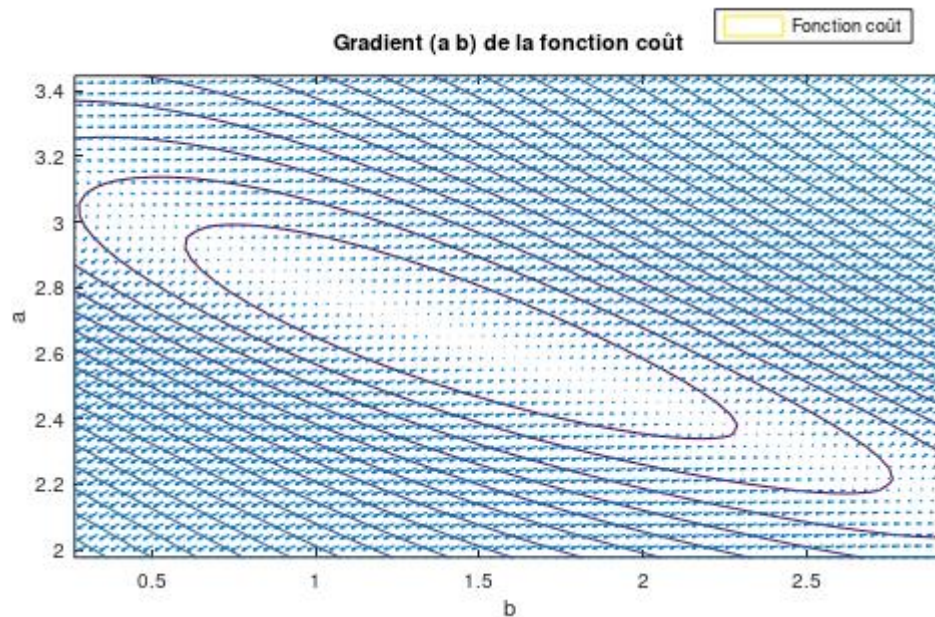
$$\nabla C_{\sigma=1} \left(\begin{pmatrix} a \\ b \end{pmatrix} \right) = \begin{pmatrix} \sum_{i=1}^n \frac{x_i r_i}{1 + r_i^2} \\ \sum_{i=1}^n \frac{r_i}{1 + r_i^2} \end{pmatrix}$$

Maintenant que le gradient de la fonction coût est établie, il est nécessaire de représenter ce gradient à deux dimensions dans l'espace a b.



Représentation du gradient de la fonction coût dans l'espace a, b

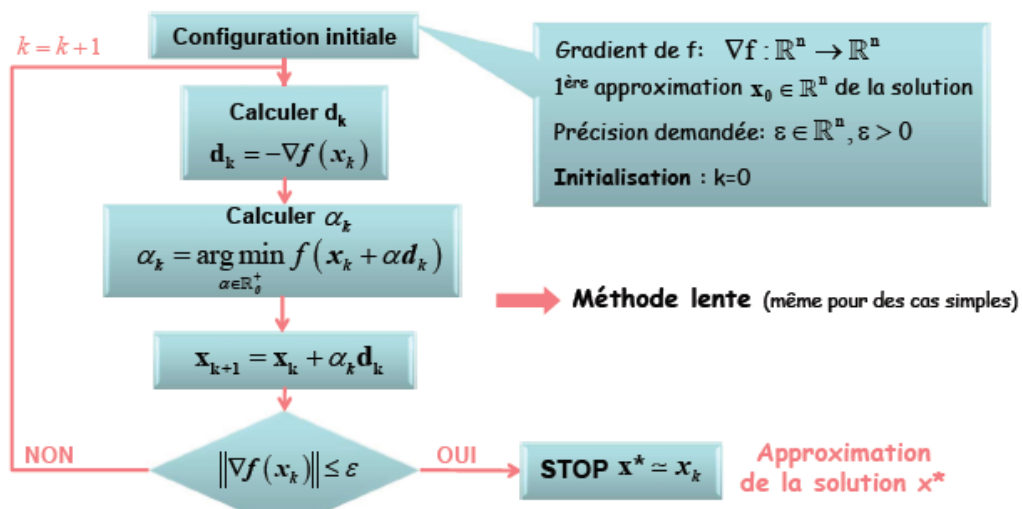
Grâce à cette représentation, on remarque que cette fonction admet plusieurs minima locaux. Sur le graphique l'ensemble des minima locaux est représenté par la zone blanche. On va donc par la suite essayer de déterminer le plus petits des minima locaux qui est le minimum global de la fonction coût.



Zoom du gradient sur le minimum global

Suite au zoom, sur l'extremum, on voit que le gradient autour de l'extremum est représenté par une zone centrifuge. La zone est donc un minimum global.

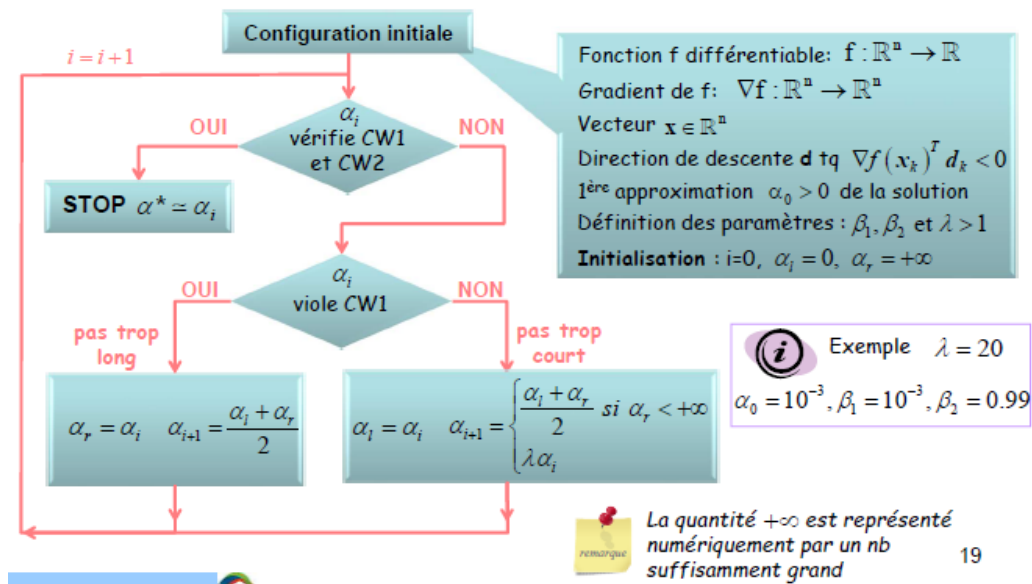
On passe maintenant à trouver la plus forte pente pour arriver jusqu'au minimum global de la fonction.



Algorithme de la plus forte pente

Dans cet algorithme, la fonction f représente la pénalisation de Cauchy.

Dans cet algorithme, on choisit la direction où la fonction décroît le plus fortement $d_k = -\nabla f\left(\begin{pmatrix} a \\ b \end{pmatrix}\right)$.



Algorithme de Fletcher-Lemarechal qui permet de déterminer le pas

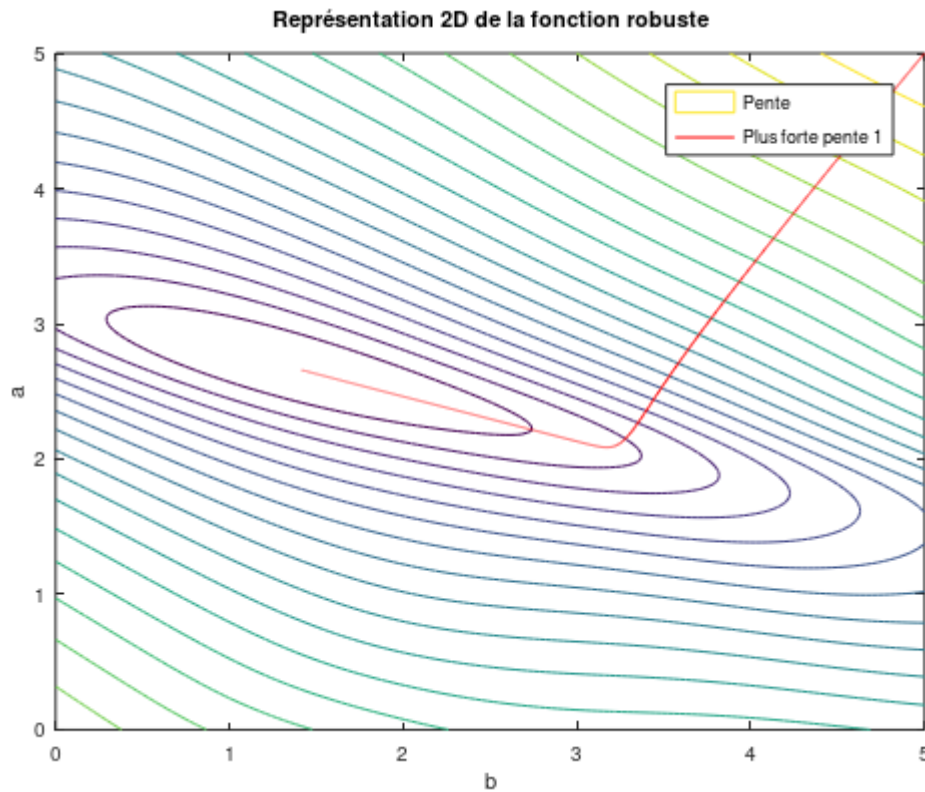
On doit maintenant déterminer le pas α qui permet d'incrémenter chaque point. Il nous faut une longueur de pas optimale, c'est-à-dire, ni trop longue, ni trop courte. L'utilisation du plus petit argument est un algorithme trop long pour notre utilisation. Pour ce faire, nous utilisons le minimum de l'algorithme Fletcher-Lemarechal, ici $\alpha_k = \min_{a,b} \text{Fletcher-Lemarechal} \left(\begin{pmatrix} a \\ b \end{pmatrix}_k + \alpha_k d_k \right)$ qui fait du tâtonnement pour trouver le pas optimal. Cette recherche est effectuée via les deux conditions de Wolfe :

- $CW1: f \left(\begin{pmatrix} a \\ b \end{pmatrix}_k + \alpha_k d_k \right) \leq f \left(\begin{pmatrix} a \\ b \end{pmatrix}_k \right) - \alpha_k \gamma$ avec $\gamma = -\beta_1 \nabla f \left(\begin{pmatrix} a \\ b \end{pmatrix}_k \right)^T d_k$ et $\beta_1 = 10^{-3}$
- $CW2: \frac{\nabla f \left(\begin{pmatrix} a \\ b \end{pmatrix}_k + \alpha_k d_k \right)^T d_k}{\nabla f \left(\begin{pmatrix} a \\ b \end{pmatrix}_k \right)^T d_k} \leq \beta_2$ avec $\beta_2 = 0.99$

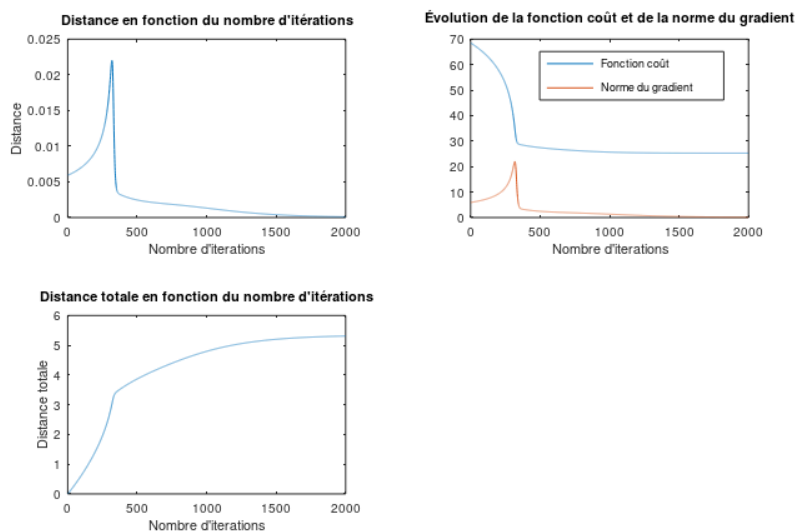
On incrémente maintenant pour obtenir le point suivant $\begin{pmatrix} a \\ b \end{pmatrix}_{k+1} = \begin{pmatrix} a \\ b \end{pmatrix}_k + \alpha_k d_k$.

Chaque nouveau point incrémenté est stocké dans un tableau qui contient l'ensemble des points de la plus forte pente.

Le dernier point nous indique le point central avec la précision demandée, ici $\epsilon = 10^{-4}$.



Plus forte pente en partant de $a = 5$ et $b = 5$ pour un centre $\begin{pmatrix} 2.6593 \\ 1.4139 \end{pmatrix}$



Avec une précision de 10^{-4} , on obtient les valeurs du centre $\begin{pmatrix} \hat{a} \\ \hat{b} \end{pmatrix} = \begin{pmatrix} 2.6593 \\ 1.4139 \end{pmatrix}$.

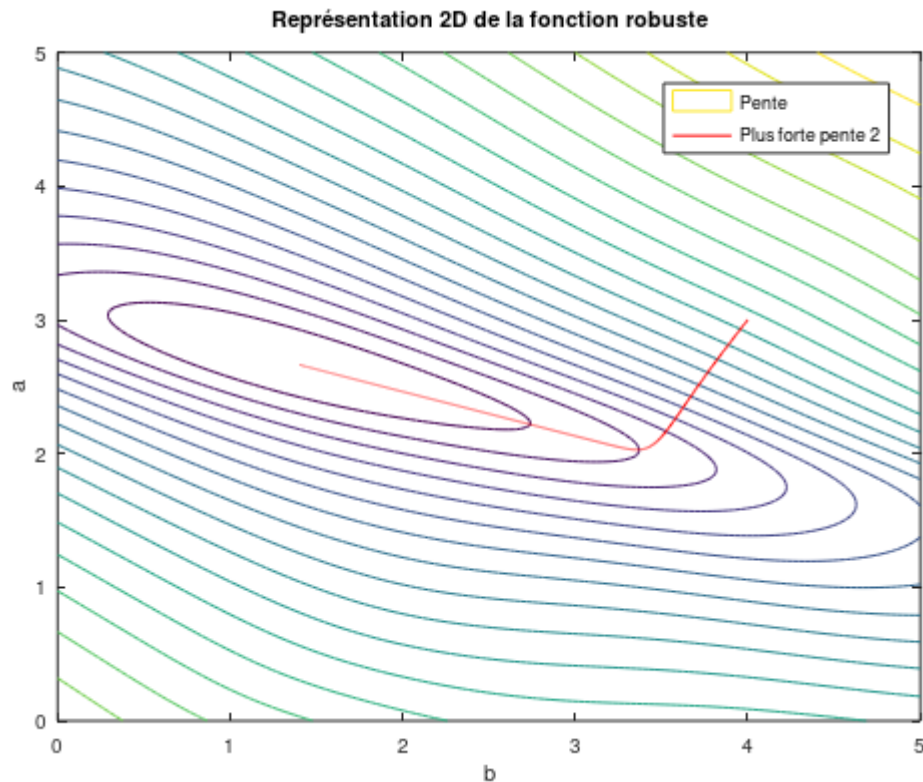
On remarque qu'à partir de 400 itérations environ, la fonction coût entre dans la zone stabilisée où elle varie assez peu. La norme du gradient, la distance et la fonction coût sont dans la partie stable à partir de 400 itérations car la pente n'évolue plus qu'à une seule dimension. En effet, celle-ci a atteint la ligne centrale et recherche simplement le centre de cette ligne.

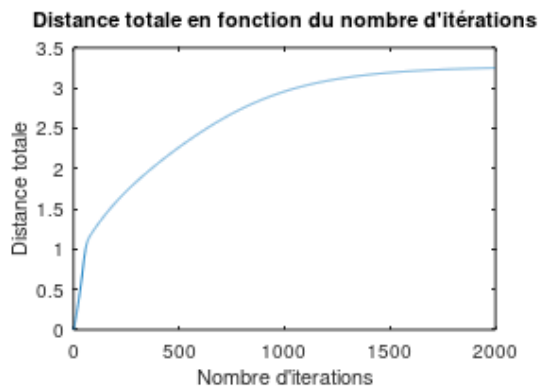
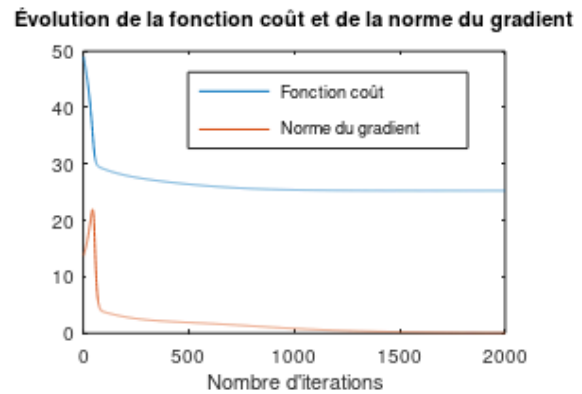
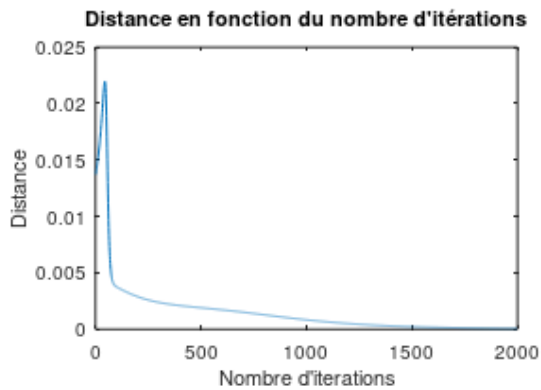
Les forts changements observables vers 400 itérations traduisent le changement de direction de la courbe.

À partir de 400 itérations environ, on observe une distance parcourue relative de plus en plus faible, une distance totale qui croît plus lentement et une norme du gradient proche de 0. Ce constat démontre que la plus forte pente se rapproche de la solution, du centre de la fonction robuste.

On remarque aussi qu'à partir de 1500 itérations, on peut affirmer avoir une solution assez précise.

Plus, le nombre d'itérations est important, plus la pente converge vers un centre précis, puisque tous les paramètres convergent.

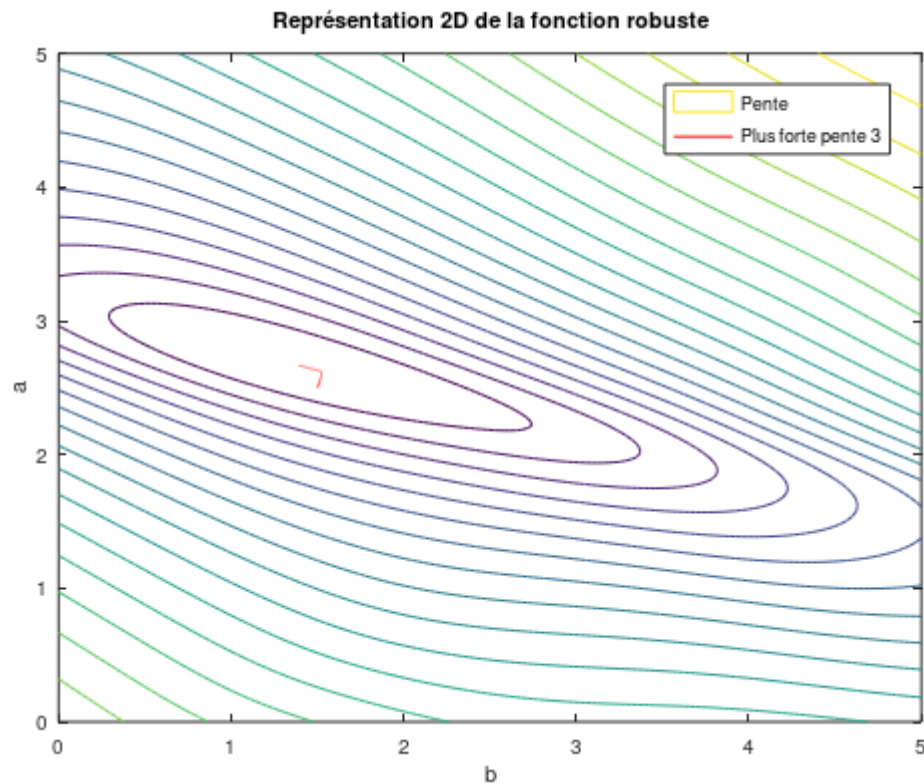




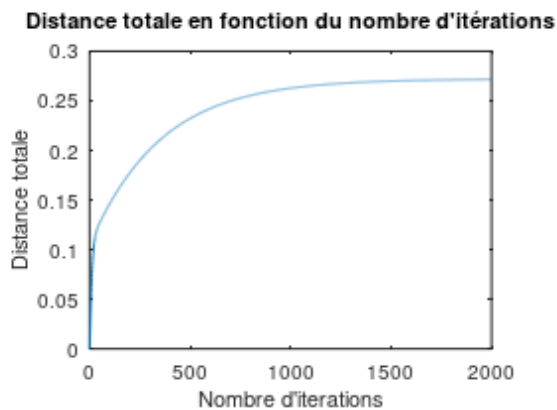
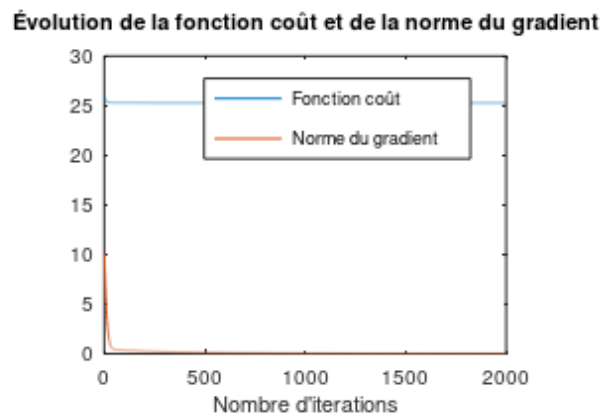
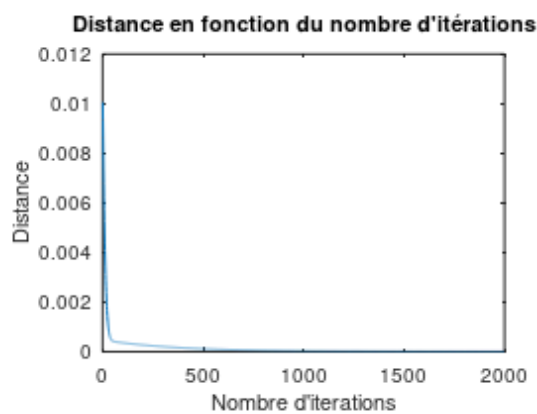
Dans cette configuration ($x_{initial} = \begin{pmatrix} 3 \\ 4 \end{pmatrix}$), on trouve un point central à $x_{central} = \begin{pmatrix} 2.6645 \\ 1.3986 \end{pmatrix}$. Plus le nombre d'itérations est élevé, plus la précision du point central l'est aussi. La précision du point central va être influencée par l'emplacement du point initial et le nombre d'itérations. Ainsi, pour avoir une forte précision, il est préférable de prendre un point initial assez proche du centre avec un nombre d'itérations important.

Pour un point initial distant du centre plus petit, on aura le même nombre d'itérations que précédemment, ainsi les points sont moins espacés.

Essayons avec un point initial $x_{initial} = \begin{pmatrix} 2.5 \\ 1.5 \end{pmatrix}$, et un nombre d'itérations de 2000.



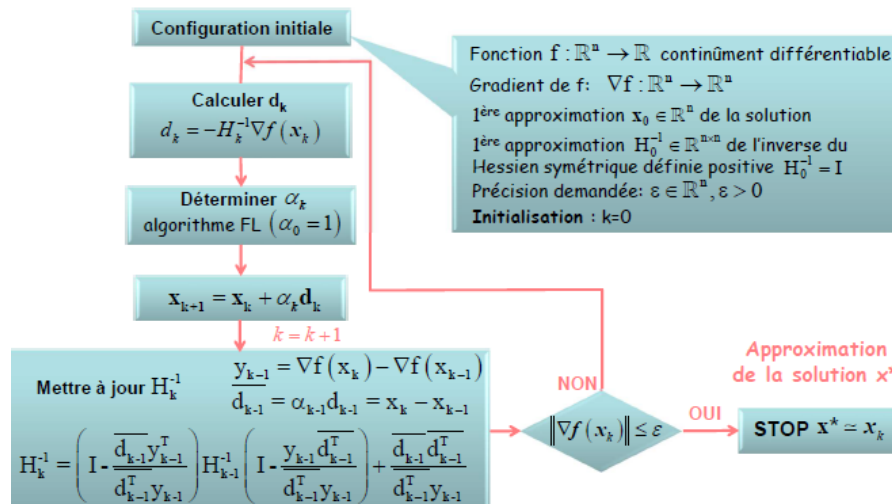
Plus forte pente en partant de $a = 2.5$ et $b = 1.5$ pour un centre $\begin{pmatrix} 2.6701 \\ 1.3821 \end{pmatrix}$



Ici, on observe un changement de direction pour des itérations proche du début, on observe donc uniquement les parties stables pour les distances et la norme.

On choisit donc un point central $x_k = \begin{pmatrix} 2.6701 \\ 1.3821 \end{pmatrix}$, car c'est le point le plus précis des 3 points précédemment déterminés.

Nous allons maintenant utiliser la méthode de Quasi-Newton. Cette méthode combine les méthodes de recherche de Newton et de la plus forte pente préconditionnée. On évalue la matrice Hessienne de la fonction robuste. On a donc un algorithme de recherche fonctionnel à convergence rapide.



Algorithme de Quasi-Newton

Dans cet algorithme, la fonction f représente la pénalisation de Cauchy.

Cet algorithme passe par un calcul avec la matrice hessienne pour trouver la direction de descente la plus forte. La direction est donc $d_k = -H_k^{-1} \nabla f \left(\begin{pmatrix} a \\ b \end{pmatrix}_k \right)$.

De même que dans l'algorithme de la plus forte pente, le pas α_k est déterminé via l'algorithme de Fletcher-Lemarechal.

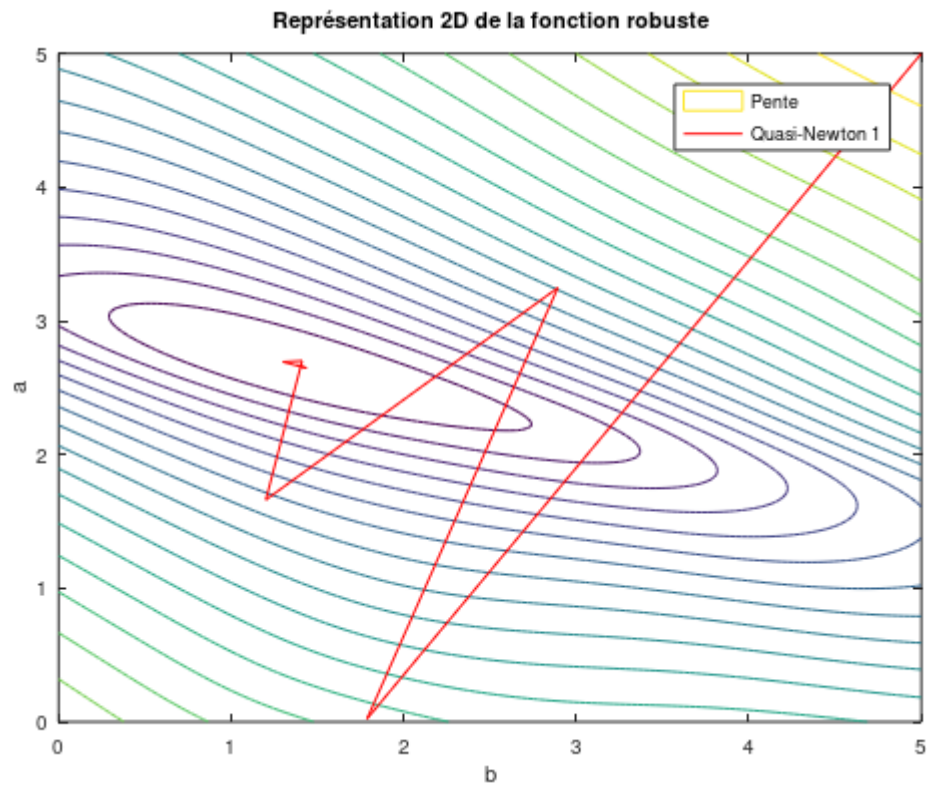
On incrémente le point suivant $\begin{pmatrix} a \\ b \end{pmatrix}_{k+1} = \begin{pmatrix} a \\ b \end{pmatrix}_k + \alpha_k d_k$.

Il faut maintenant mettre à jour la matrice hessienne par rapport au nouveau point.

$$y_{k-1} = \nabla f \left(\begin{pmatrix} a \\ b \end{pmatrix}_k \right) - \nabla f \left(\begin{pmatrix} a \\ b \end{pmatrix}_{k-1} \right)$$

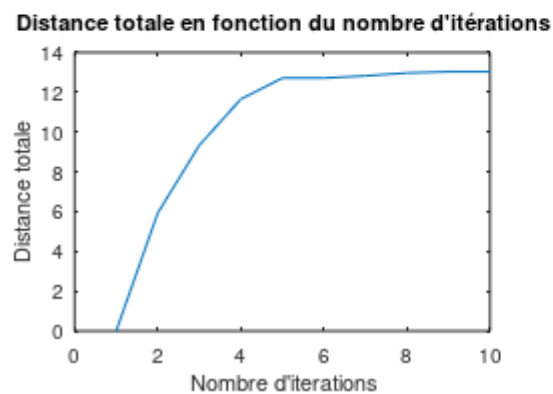
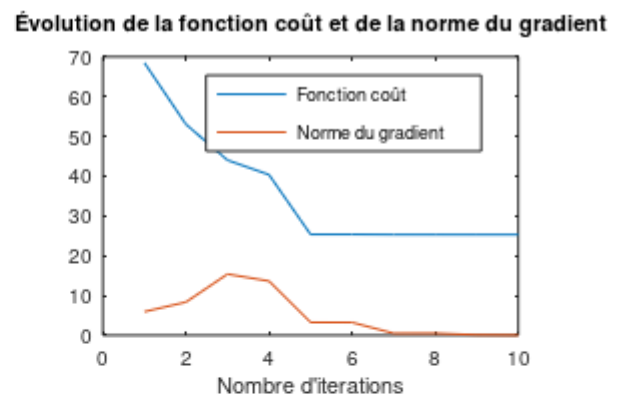
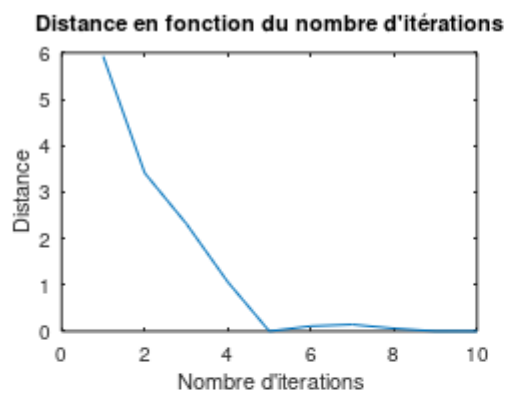
$$\overline{d_{k-1}} = \alpha_{k-1} d_{k-1}$$

$$H_k^{-1} = \left(I - \left(\frac{\overline{d_{k-1}} y_{k-1}^T}{\overline{d_{k-1}}^T y_{k-1}} \right) \right) H_{k-1}^{-1} \left(I - \left(\frac{y_{k-1} \overline{d_{k-1}}^T}{\overline{d_{k-1}}^T y_{k-1}} \right) \right) + \left(\frac{\overline{d_{k-1}} \overline{d_{k-1}}^T}{\overline{d_{k-1}}^T y_{k-1}} \right)$$

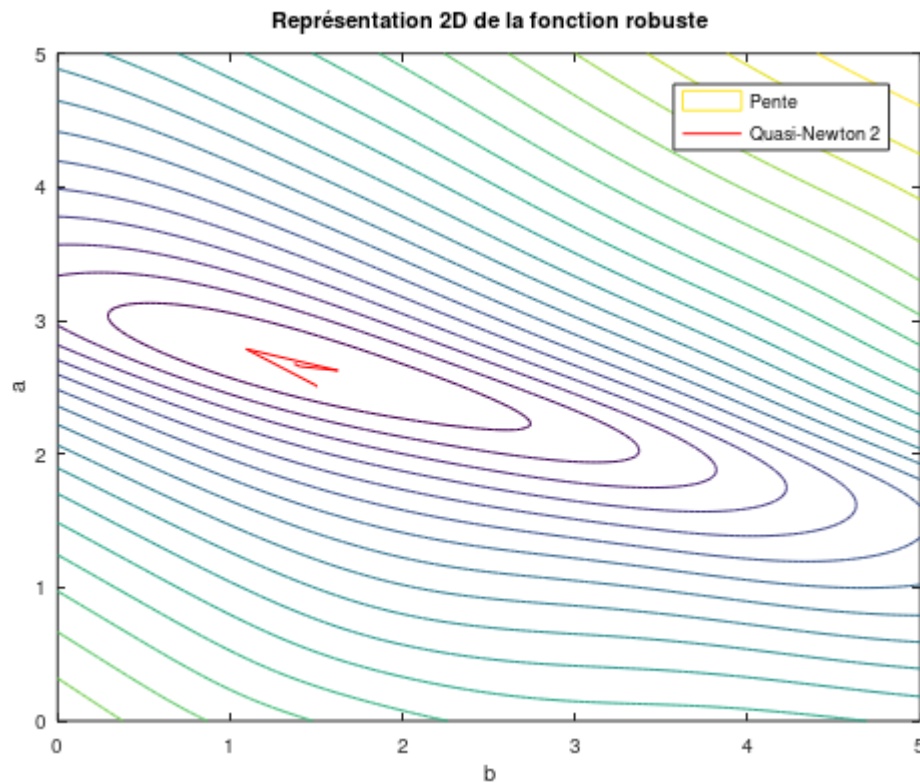


Quasi-Newton avec $x_{initial} = \begin{pmatrix} 5 \\ 5 \end{pmatrix}$ pour un centre estimé à $\begin{pmatrix} 2.6703 \\ 1.3816 \end{pmatrix}$

Grâce à cette figure, on remarque que la méthode de Quasi-Newton n'a besoin que de très peu d'itération, ce qui confirme la caractéristique de la convergence rapide de la méthode.



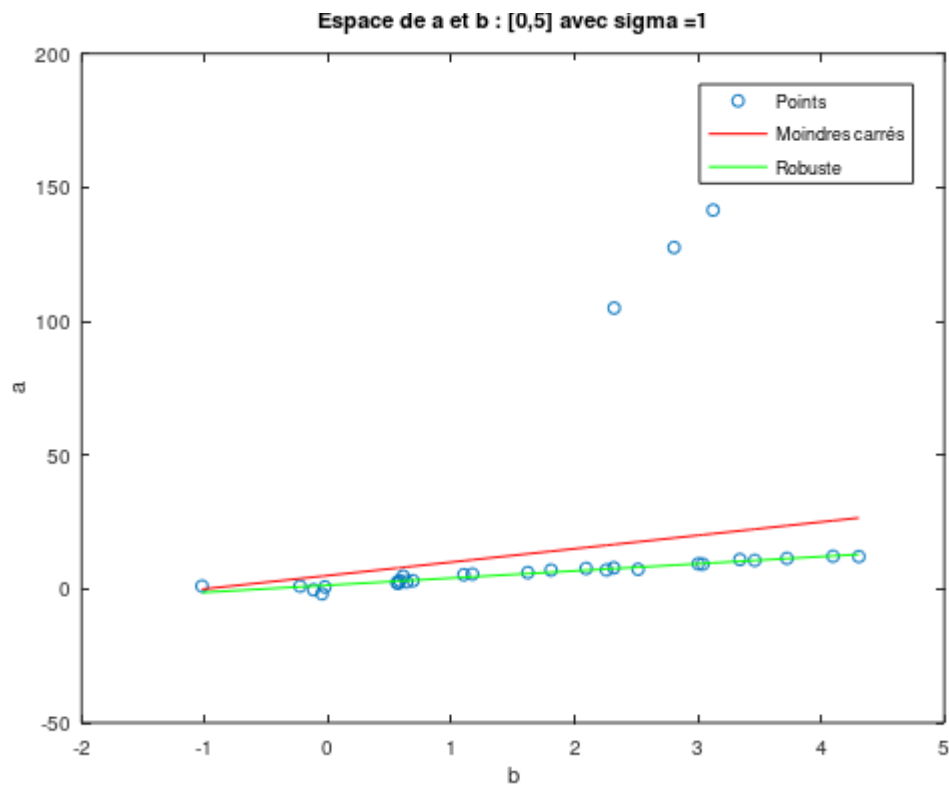
On remarque que cette méthode n'a besoin que d'une dizaine d'itérations pour un point initial $x_{initial} = \begin{pmatrix} 5 \\ 5 \end{pmatrix}$ pour pouvoir estimer précisément le centre de la fonction robuste. En effet, la distance relative est proche de 0 dès la cinquième itération. Ainsi, le rendement efficacité par itération de cette méthode est grandement supérieur à la méthode de la plus forte pente, car notamment la distance parcourue entre 2 itérations est beaucoup plus grande que la distance entre 2 itérations pour la méthode de la plus forte descente.



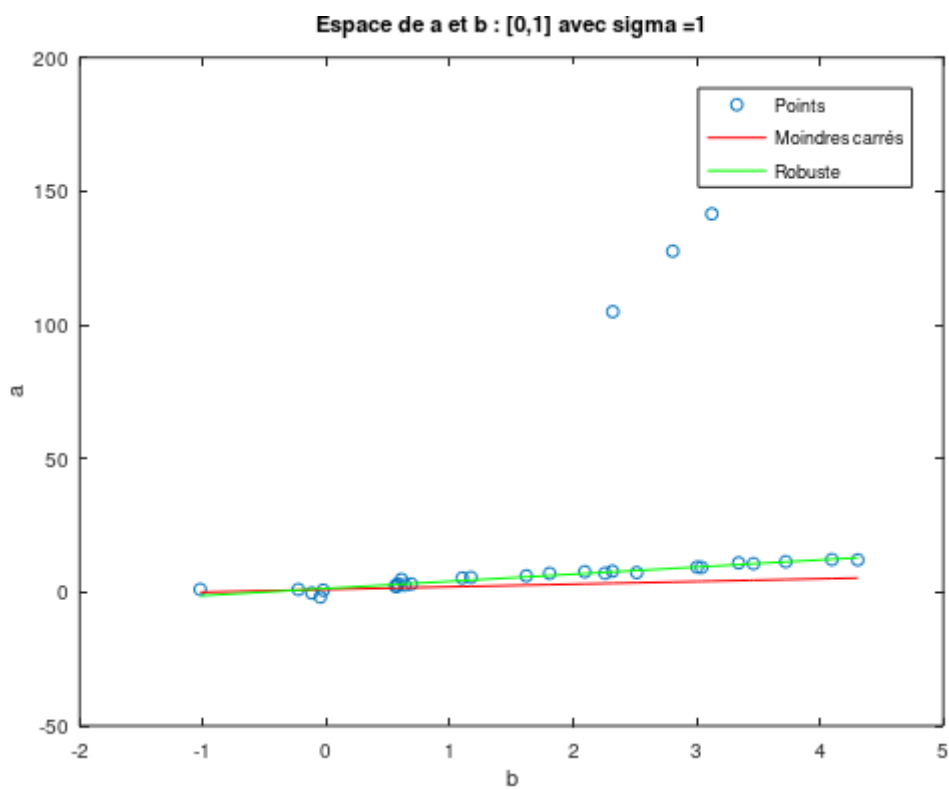
Quasi-Newton avec $x_{initial} = \begin{pmatrix} 2.5 \\ 1.5 \end{pmatrix}$ pour un centre estimé à $\begin{pmatrix} 2.6703 \\ 1.3816 \end{pmatrix}$

On remarque que la précision n'a pas été améliorée contrairement alors qu'on est parti d'un point plus proche du centre contrairement à la méthode de la plus forte descente. Cela est dû au fait qu'on a atteint l'itération maximale pour la précision demandée de 10^{-4} . Lorsqu'on a utilisé la méthode de la plus forte descente, on n'avait pas atteint l'itération maximale qui est supérieure à 2000.

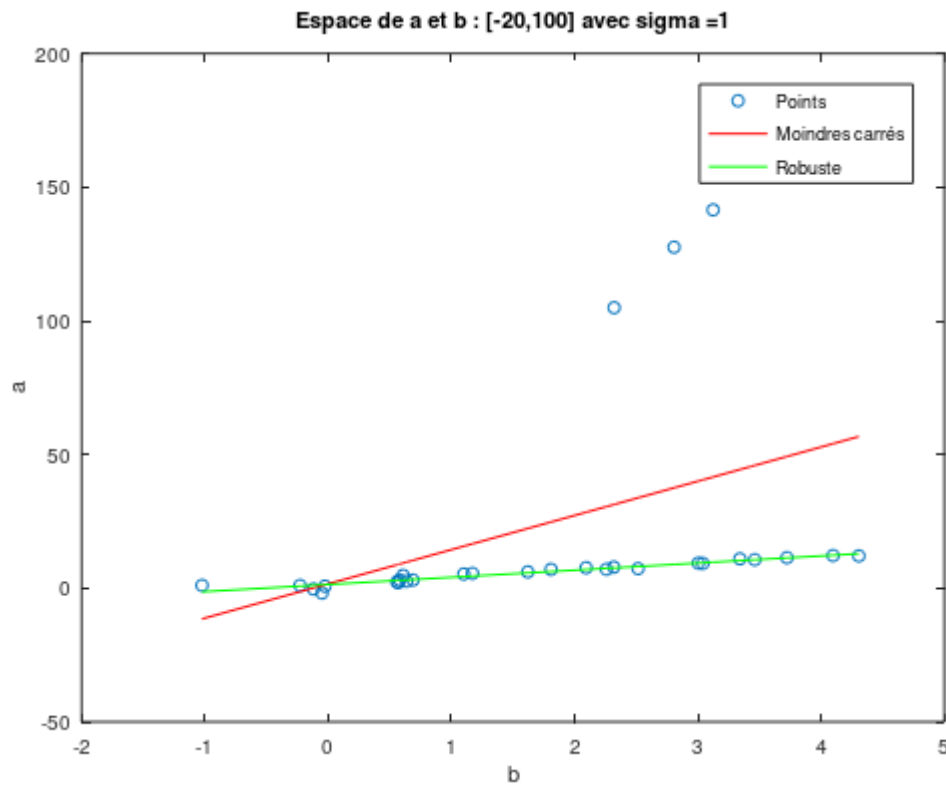
Ainsi, on a un rapport de précision/itérations supérieur à $\frac{2000}{5} = 400$ entre Quasi-Newton et la méthode de la plus forte descente. Avec un rapport de norme de $\frac{|\sqrt{2.6701^2 + 1.3821^2} - \sqrt{2.6703^2 + 1.3816^2}|}{\sqrt{2.6703^2 + 1.3816^2}} = 1.73 \cdot 10^{-5}$ entre Quasi-Newton et la plus forte pente.



Représentation des deux estimations sur un intervalle $[0, 5]$ pour $\sigma = 1$

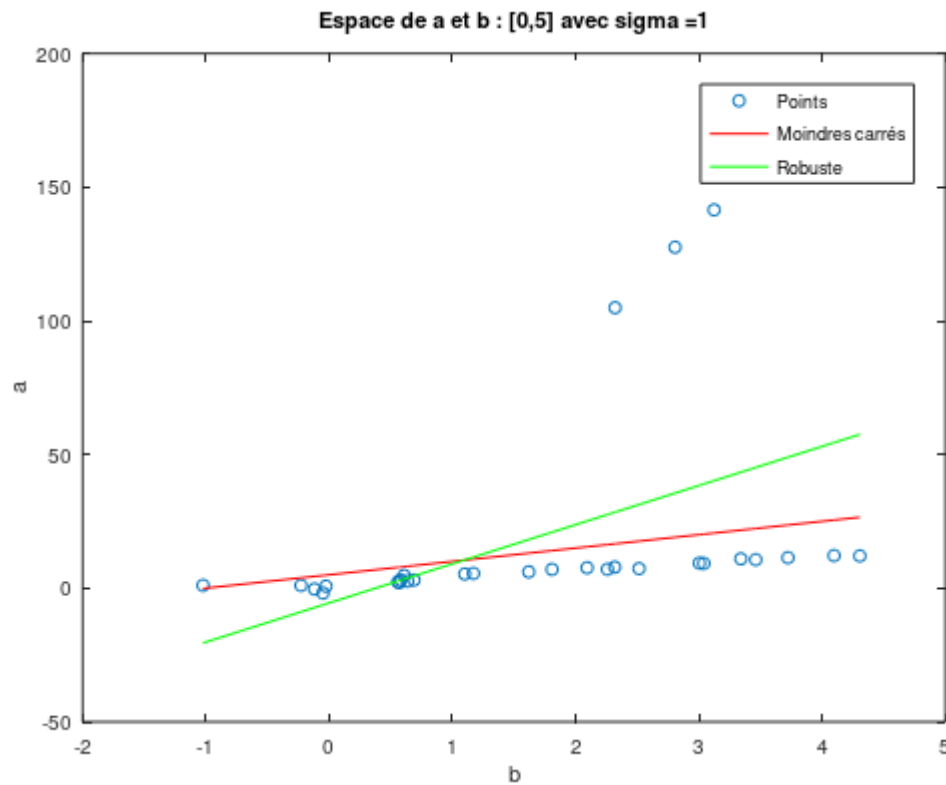


Représentation des deux estimations sur un intervalle $[0, 1]$ pour $\sigma = 1$



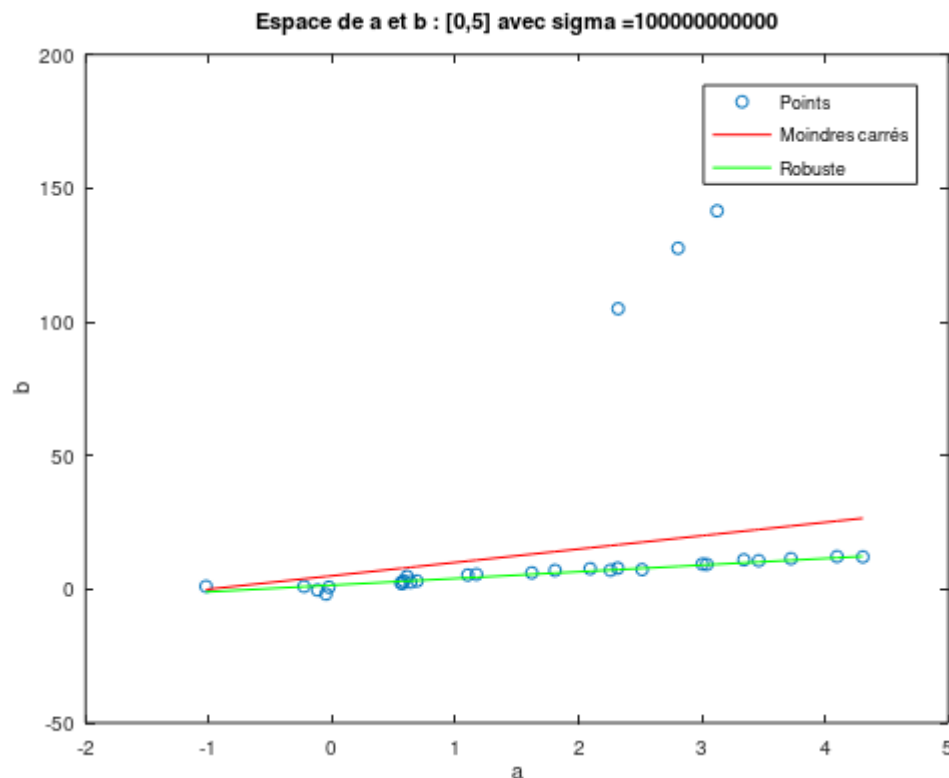
Représentation des deux estimations sur un intervalle [-20, 100] pour $\sigma = 1$

Suite à ces différents tests, on remarque que le choix de l'intervalle $[a, b]$ est primordial pour avoir la meilleure estimation possible. En effet, si l'intervalle est trop court, alors les a et b seront les valeurs extrêmes de cet intervalle (voir $[0, 1]$), mais ne seront pas les valeurs les plus optimales possibles. En effet, les valeurs seront les meilleurs possibles de l'intervalle. Il est donc préférable de choisir un intervalle plus grand, ce qui permet de prendre les valeurs de a et b les plus précises possibles. Mais choisir un intervalle large entraîne beaucoup plus de calculs, il faut donc choisir un compromis.



Représentation de la courbe robuste pour un point initial $x_{initial} = \begin{pmatrix} 10 \\ 10 \end{pmatrix}$ avec Quasi-Newton

On remarque que la courbe robuste n'est pas une bonne estimation, car elle passe sur très peu de points. Cela est dû au fait que le point initial est très éloigné du centre et le nombre d'itérations est faible (10). Ainsi, l'algorithme a été stoppé avant d'avoir pu trouver les coordonnées optimales. Ainsi, le choix du point initial est primordial pour obtenir une bonne estimation avec la méthode de Quasi-Newton.



Représentation des approximations avec $\sigma = 100000000000$

On remarque que sigma n'a quasiment aucune influence sur la courbe robuste.

En effet, on passe par le gradient,

$$\nabla f(b) = \sum_{i=1}^n \frac{r_i}{\sigma^2 + r_i^2}, \text{ et } \nabla f(a) = \sum_{i=1}^n \frac{x_i r_i}{\sigma^2 + r_i^2}$$

$$\text{et } \lim_{r_i \rightarrow \infty} \nabla f(a) = \lim_{r_i \rightarrow +\infty} \nabla f(b) = 0, \forall \sigma \in \mathbb{R}$$

Ainsi, σ n'a que très peu d'influence sur la courbe robuste.

Enfin, la fonction robuste est une meilleure approximation que la fonction des moindres carrés car celle-ci ne prend pas en compte les valeurs aberrantes. Le modèle des moindres carrés est plus adapté à un système linéaire.

Les méthodes que Quasi-Newton et de la plus forte pente sont qu'en a elles plus adaptées aux modèles non-linéaires.

On remarque que l'estimateur robuste correspond en fait au maximum de vraisemblance. En effet, celui-ci s'exprime sous la forme $L_Y(\theta) = \sum_{n=0}^{N-1} \ln(f_{Y|\theta}(y_n|\theta))$.

$$Y \sim \text{Cauchy}(ax + b, \sigma)$$

soit $f_{Y|\theta}(y|\theta)$ la densité de probabilité de Y avec $\theta = \{ax + b, \sigma\}$

$$f_{Y|\theta}(y_n|\theta) = \frac{\sigma}{\pi[\sigma^2 + (y_n - ax - b)^2]}$$

$$f_{Y|\theta}(y|\theta) = \frac{\sigma}{\Pi[\sigma^2 + r_n^2]}, \text{ avec } r = ax + b - y_n$$

On recherche la vraisemblance maximale :

$$\hat{\theta} = \arg \max_{\theta} L_Y(\theta) = \arg \max_{\theta} \sum_{n=0}^{N-1} \ln \left(\frac{\sigma}{\Pi[\sigma^2 + r_n^2]} \right)$$

Dans notre problème nous cherchons le maximum de l'opposé de notre fonction pour ainsi trouver le minimum.

$$\hat{\theta} = \arg \min_{\theta} -L_Y(\theta)$$

$$\hat{\theta} = \arg \min_{\theta} \sum_{n=0}^{N-1} \ln \left(\frac{\Pi[\sigma^2 + r_n^2]}{\sigma} \right)$$

$$\hat{\theta} = \arg \min_{\theta} N \ln(\Pi\sigma) + 2 \sum_{n=0}^{N-1} \frac{1}{2} \ln \left(\frac{\sigma^2 + r_n^2}{\sigma^2} \right)$$

Notre fonction robuste s'exprime $C(r) = \sum_{n=0}^N \frac{1}{2} \ln \left(1 + \left(\frac{r_n}{\sigma} \right)^2 \right)$.

$$\hat{\theta} = \arg \min_{\theta} N \ln(\Pi\sigma) + 2C(r)$$

Cet estimateur est un estimateur du maximum de vraisemblance sur des variables indépendantes et identiquement distribuées $Y \sim \text{Cauchy}(ax + b, \sigma)$. Ainsi, l'estimateur robuste revient à faire un maximum de vraisemblance sur l'estimateur ci-dessus auquel un offset a été ajouté $N \ln(\Pi\sigma)$.

Conclusion

Dans cette activité, nous avons pu voir les différentes estimations possibles pour différents points exploitables et bruités. Nous avons principalement vu deux méthodes, la méthode des moindres carrés et la méthode de la robustesse. La méthode des moindres carrés est fortement influencée par la présence du bruit, ce qui n'est pas le cas de la méthode robuste. L'estimation des moindres carrés est donc moins efficace que la méthode de la robustesse si l'on souhaite extraire l'information. De plus, nous avons pu remarquer que l'on obtient une meilleure précision de l'estimation en passant par le calcul des dérivées premières et secondes via les méthodes de la plus forte pente et de quasi-newton. Enfin, l'algorithme le plus efficace est celui de quasi-newton car il converge très rapidement avec la précision demandée si les paramètres de position initiale, d'itération et de précision sont bien paramétrés.

Enfin, cette estimation robuste peut être calculée via le maximum de vraisemblance d'une loi de Cauchy de paramètres $ax + b$ et σ .