# TP4 – Estimation des paramètres d'une ellipse

À partir d'un ensemble de points dont la forme rappelle celle d'une ellipse, est-il possible de trouver l'ellipse qui passe « au plus près » de ces points ? Ce genre de question se pose très souvent en analyse d'images. Il s'agit d'un problème d'*estimation*. Dans ce TP, vous allez tester plusieurs méthodes d'estimation des paramètres d'une ellipse, puis vous utiliserez celle que vous jugerez « la meilleure » dans une application de réalité augmentée.

Commencez par lancer le script `donnees.m`, qui affiche une ellipse tirée aléatoirement, ainsi que $n$ points placés aléatoirement sur cette ellipse, dont chaque coordonnée est ensuite bruitée par un bruit additif gaussien.

## Exercice 1 : estimation par le maximum de vraisemblance

Le maximum de vraisemblance est une technique très générale d'estimation de paramètres, qui a déjà été illustrée à plusieurs reprises dans les TP de probabilités/statistiques en 1A.

Si $(X_1, \ldots, X_n)$ est un $n$-uplet de variables aléatoires continues « iid » (indépendantes et identiquement distribuées) de même densité de probabilité $f_p(x)$ dépendant d'une liste de paramètres que l'on désigne par $p$, alors la probabilité pour que $n$ observations $(x_1, \ldots, x_n)$ constituent une réalisation de ce $n$-uplet est appelée *vraisemblance* (souvent notée $L$, pour *Likelihood*). Grâce au caractère iid des variables aléatoires, elle s'écrit :

$$L_p(x_1, \ldots, x_n) = \prod_{i=1}^{n} f_p(x_i) \tag{1}$$

Une façon de caractériser une ellipse consiste à choisir deux points du plan $F_1$ et $F_2$, appelés *foyers*, et un réel $a$ tel que $2a > F_1F_2 = 2c$. L'ensemble des points $P$ tels que $PF_1 + PF_2 = 2a$ constitue une ellipse (cette méthode est connue sous le nom de « méthode du jardinier »). On dispose donc d'un critère permettant d'évaluer si un point $P$ du plan se trouve plus ou moins près de l'ellipse. En notant $E(P) = PF_1 + PF_2 - 2a$ l'écart entre les deux membres de l'équation précédente, il est légitime de modéliser ces écarts par une **loi normale tronquée** :

$$f_p(P) = \begin{cases} K \, \exp\left\{-\frac{E(P)^2}{2\,\sigma^2}\right\} & \text{si } E(P) \geq \min\limits_{P \in \mathbb{R}^2}\{E(P)\} = F_1F_2 - 2a = 2(c-a) < 0 \\ 0 \quad \text{sinon} \end{cases} \tag{2}$$

où $p = (F_1, F_2, a, \sigma)$. L'écart $E(P)$ prenant ses valeurs dans $[2(c-a), +\infty[$ et non dans $\mathbb{R}$, le facteur de normalisation $K$ n'est pas égal à $1/(\sigma\sqrt{2\pi})$. Néanmoins, si $2|c-a| \gg \sigma$, on montre que $K \approx 1/(\sigma\sqrt{2\pi})$.

L'*estimation par maximum de vraisemblance* consiste à chercher la valeur $\hat{p}$ de $p$ qui maximise la vraisemblance des données, données qui sont ici $n$ points du plan $(P_1, \ldots, P_n)$. Comme un produit est plus difficile à maximiser qu'une somme, et que la fonction logarithme est strictement croissante, il est préférable de maximiser le logarithme de la vraisemblance (ou *log-vraisemblance*) :

$$\hat{p} = \underset{p \in (\mathbb{R}^2)^2 \times (\mathbb{R}^+)^2}{\arg\max} \left\{ \ln \left[ \prod_{i=1}^{n} f_p(P_i) \right] \right\} = \underset{p \in (\mathbb{R}^2)^2 \times (\mathbb{R}^+)^2}{\arg\max} \left\{ n \ln K - \frac{1}{2\,\sigma^2} \sum_{i=1}^{n} E(P_i)^2 \right\} \tag{3}$$

Parmi les paramètres de la loi, l'écart-type $\sigma$ n'est pas caractéristique de l'ellipse. Si nous supposons que ce paramètre a une valeur suffisamment faible pour que la condition $2|c-a| \gg \sigma$ soit vérifiée, alors $K$ ne dépend que de $\sigma$. Cela permet de simplifier le problème (3) :

$$(\hat{F_1}, \hat{F_2}, \hat{a}) = \underset{(F_1, F_2, a) \in (\mathbb{R}^2)^2 \times \mathbb{R}^+}{\arg\max} \left\{ -\sum_{i=1}^{n} E(P_i)^2 \right\} = \underset{(F_1, F_2, a) \in (\mathbb{R}^2)^2 \times \mathbb{R}^+}{\arg\min} \left\{ \sum_{i=1}^{n} E(P_i)^2 \right\} \tag{4}$$

Complétez le script `exercice_1.m`, censé résoudre le problème (4) par tirages aléatoires selon des lois uniformes (fonction `rand` de Matlab). Chaque résultat est évalué par son score $(A_0 \cap \hat{A})/(A_0 \cup \hat{A}) \in [0, 1]$, où $A_0$ désigne l'aire de l'ellipse originale et $\hat{A}$ celle de l'ellipse estimée. Testez ce script en jouant sur ses paramètres.

## Exercice 2 : estimation par les moindres carrés ordinaires

L'équation cartésienne d'une conique (ellipse, parabole, hyperbole) est une équation polynomiale de degré 2 :

$$\alpha\, x^2 + \beta\, x\, y + \gamma\, y^2 + \delta\, x + \epsilon\, y + \phi = 0 \tag{5}$$

Bien sûr, les six paramètres $(\alpha, \beta, \gamma, \delta, \epsilon, \phi)$ ne sont pas indépendants : la conique ne change pas si tous ces paramètres sont multipliés par un même facteur. Cela signifie qu'une conique possède cinq « degrés de liberté », ce qui est cohérent avec le modèle d'ellipse de l'exercice 1 utilisant comme paramètres $(F_1, F_2, a)$. Il semble donc possible de fixer a priori la valeur d'un des six paramètres de l'équation (5), par exemple $\phi = 1$, ou d'imposer toute autre contrainte linéaire sur les paramètres, par exemple $\alpha + \gamma = 1$. Remarquez quand même que cela n'est pas toujours possible : par exemple, la contrainte $\phi = 1$ ne peut pas être imposée à une conique passant par l'origine. Si l'on ne se soucie pas de ce risque, la contrainte $\phi = 1$ permet d'écrire, pour chaque point $P_i = (x_i, y_i)$ censé se trouver sur la conique recherchée, une équation linéaire du type suivant :

$$x_i^2\, \alpha + x_i\, y_i\, \beta + y_i^2\, \gamma + x_i\, \delta + y_i\, \epsilon = -1 \tag{6}$$

Si l'on dispose de $n > 5$ points $P_i$, les équations (6) forment un système linéaire $A\,X = B$ *sur-contraint* qui n'admet généralement pas de solution exacte. Il convient de le résoudre de manière approchée, au sens des *moindres carrés ordinaires*, c'est-à-dire en utilisant soit la pseudo-inverse `pinv(A)`, soit l'expression `A\B`.

Complétez le script `exercice_2.m` pour effectuer cette estimation de $p = (\alpha, \beta, \gamma, \delta, \epsilon, \phi)$ sous chacune des deux contraintes suivantes : $\alpha + \gamma = 1$, puis $\phi = 1$. Dans chaque cas, affichez la solution et calculez son score.

## Exercice 3 : estimation par les moindres carrés totaux

Comme cela a déjà été dit, il est facile d'imposer une contrainte linéaire sur un ou plusieurs des paramètres à estimer, mais cela est risqué. Sans contrainte, la résolution des équations (5) pose problème car ces équations forment un système linéaire *homogène* $A\,X = 0$, où $X = [\alpha, \beta, \gamma, \delta, \epsilon, \phi]^\top$ et $A$ est une matrice de taille $n \times 6$. Afin d'éviter la solution évidente $\hat{X} = 0$, la parade usuelle consiste à imposer la contrainte non linéaire $\|X\| = 1$. Ce problème de *moindres carrés totaux* s'écrit donc :

$$\hat{X} = \underset{\|X\|=1}{\arg\min} \left\{ \|A\,X\|^2 \right\} \tag{7}$$

La résolution d'un problème d'optimisation différentiable sous contrainte se fait en introduisant un *lagrangien* :

$$\mathcal{L}(X, \lambda) = \|A\,X\|^2 + \lambda\,(1 - \|X\|^2) \tag{8}$$

où $\lambda$ est un *multiplicateur de Lagrange*. La condition d'optimalité $\nabla\mathcal{L} = 0$ donne les deux équations suivantes :

$$\left\{ \begin{array}{l} \nabla_X \mathcal{L}(X, \lambda) = 0 \\ \nabla_\lambda \mathcal{L}(X, \lambda) = 0 \end{array} \right. \iff \left\{ \begin{array}{l} A^\top A\, X = \lambda X \\ \|X\| = 1 \end{array} \right. \tag{9}$$

La solution du problème (7) est donc un vecteur propre de norme 1 de la matrice $A^\top A$. Sachant que cette matrice $6 \times 6$ est symétrique réelle, elle admet une base orthonormée de vecteurs propres. On en conclut donc que la solution $\hat{X}$ de (7) est le vecteur propre de $A^\top A$ associé à sa plus petite valeur propre (les valeurs propres de $A^\top A$ sont toutes positives ou nulles, car cette matrice est semi-définie positive).

Cette solution du problème (7) pourrait également s'obtenir par *décomposition en valeurs singulières* (SVD), car chaque valeur singulière non nulle de $A$ est égale à la racine carrée d'une valeur propre non nulle de $A^\top A$.

Faites une copie du script `exercice_2.m`, de nom `exercice_3.m`, que vous compléterez de manière à comparer, pour chaque tirage, les scores obtenus par ces différentes méthodes d'estimation aux moindres carrés.

## Application à la réalité augmentée

Parmi les méthodes d'estimation testées dans ce TP, sélectionnez celle qui vous semble « la meilleure ». Complétez la fonction `estimation_ellipse` inspirée de cette méthode, dans le fichier de nom `estimation_ellipse.m`. Pendant la mise au point, n'hésitez pas à modifier les valeurs de `pas_frame` et `taille_sequence` du script `augmentation.m`. Après avoir lu les directives indiquées au début de ce script, lancez-le. Vous êtes en train d'effectuer l'*augmentation* d'une vidéo... mais il faut vous armer de patience. Enfin, lancez le script `cinema.m` !