Rapport MAP 568 : Gestion des incertitudes et analyse de risque

Plassier Vincent

1 Minimisation par processus gaussien

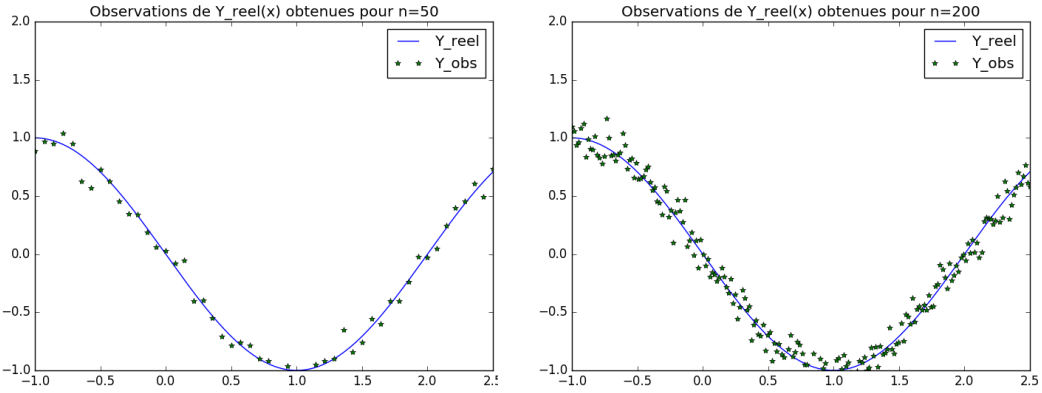
Dans cette partie nous considérons la fonction Y_{reel} donnée par :

$$Y_{reel}(x) = -\sin(\frac{\pi x}{2})$$

L'objectif sera d'estimer le minimum et la position du minimum de la fonction dérivée $Y_{reel}^{'}$.
Dans un premier temps nous considérons le métamodèle à 4 paramètres suivant :

$$Y_{meta}(x) = \beta_1 + \beta_2 x + \beta_3 x^2 + \beta_4 x^3$$

Nous commençons par générer un jeu de données, voici les observations Y_{obs} obtenues pour différentes valeurs de n :



(a) n=50 données (b) n=200 données

FIGURE 1: Observations de Y_{reel} pour différentes valeurs de n

Dans un second temps, nous déterminons σ_{mod} par la méthode de maximum de vraisemblance qui consiste à minimiser en σ_{mod} la fonction :

$$\log det(R) + (y_{obs} - H\beta_{post}))^T R^{-1}(y_{obs} - H\beta_{post})$$

Pour une valeur de n observations données, nous trouvons le σ_{mod} souhaité grâce à la fonction **fmin** présente dans le module **scipy.optimize** de python. Cependant il existe de nombreuses façons de déterminer le minimum d'une fonction. Les plus célèbres étant celles de descente de gradient comme celles d'Armijo ou de Wolfe, ainsi que celles de quasi Newton.

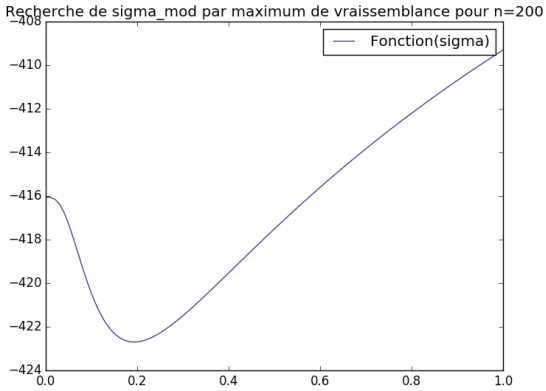


FIGURE 2: Fonction en fonction de σ_{mod} pour $n = 200$

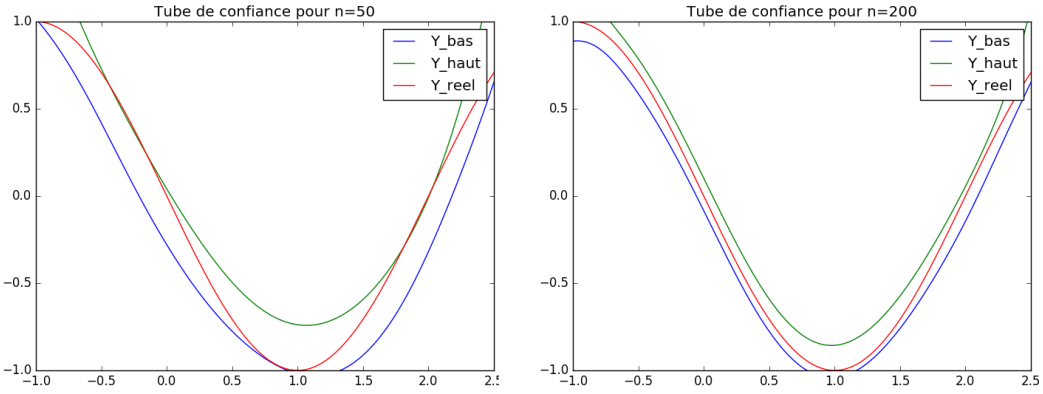
Par exemple, en générant $n = 50$ données, python nous renvoie σ_{mod} de l'ordre de 0.17 alors que pour $n = 200$ données nous trouvons $\sigma_{mod} \simeq 0.21$. La courbe 2 illustre les variations de la $Fonction$ en fonction de σ_{mod}.

Dès lors, il est possible de déterminer un tube de confiance pour Y_{reel}. Afin de calculer le tube de confiance nous utilisons le résultat :

$$Var_{post}(Y(x)) = \sigma^2_{mod} - \begin{pmatrix} h(x) \\ r(x) \end{pmatrix}^T \begin{pmatrix} 0 & H^T \\ H & R \end{pmatrix}^{-1} \begin{pmatrix} h(x) \\ r(x) \end{pmatrix}$$

Nous traçons dès lors les courbes $Y_{post} - 2\sqrt{Var_{post}(Y(x))}$ et $Y_{post} + 2\sqrt{Var_{post}(Y(x))}$.

Nous remarquons l'absence d'erreur de mesure dans notre formule, en effet nous modélisons Y_{reel} c'est pourquoi σ^2_{mes} n'intervient pas. Notre script python nous donne les résultats suivants, dans lesquels Y_{reel} est tracée en rouge :

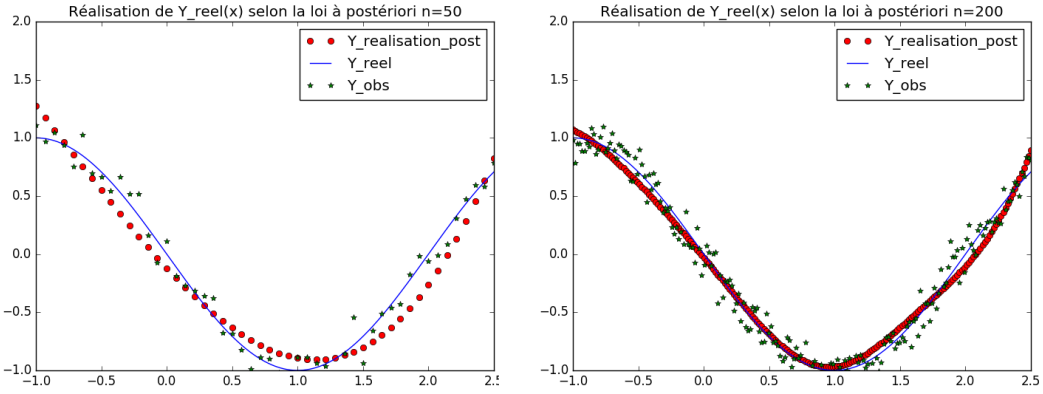


(a) n=50 données (b) n=200 données

FIGURE 3: Tube de confiance sur $Y_{reel}(x)$

Comme attendu, le tube de confiance se ressert autour de la fonction Y_{reel} lorsque nous augmentons le nombre de donnéees : n. Nous pouvons remarquer que la fonction Y_{reel} dépasse légèrement du tube de confiance pour $n = 50$, en effet, même en choisissant les meilleurs paramètres pour notre metamodèle cela ne signifie pas que celui-ci est parfait.

En s'inspirant de la remarque 1 présente à la page 100 de notre cours, en faisant l'hypothèse que nos observations sont quasi-parfaites nous sommes en mesure de tracer des réalisations de Y_{reel} selon la loi à *posteriori* (cf 4). Dans la prochaine figure nous traçons en rouge les réalisations et en vert les observations initiales.



(a) n=50 données (b) n=200 données

FIGURE 4: Réalisations de $Y_{reel}(x)$ selon la loi à postériori

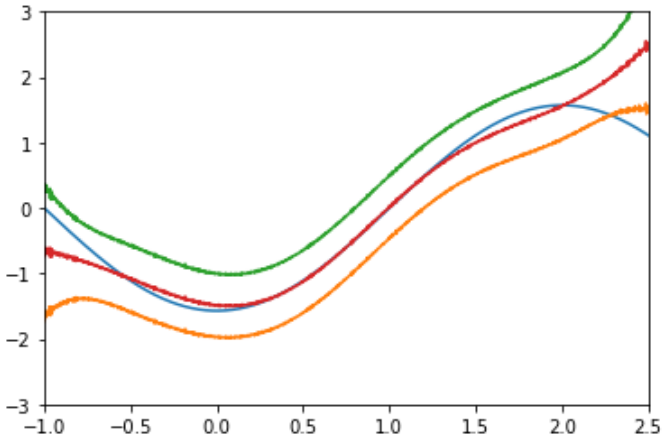
Attaquons-nous désormais au coeur de notre problème. L'objectif du projet étant d'obtenir de l'information sur Y'_{reel} nous commençons d'abord par déterminer un tube de confiance sur Y'_{reel}.
Pour cela, nous effectuons une méthode par différences finies sur des réalisations de Y_{post} que nous couplons avec une méthode de Monte-Carlo en chaque point $(x^{(j)})_{1\leq j\leq n}$.
Dans notre modèle, nous avons trois paramètres :

1. n : nombre d'observations,
2. M : nombre de répétitions de l'expérience,
3. P : nombre de points tirés selon la distribution a *postériori*.

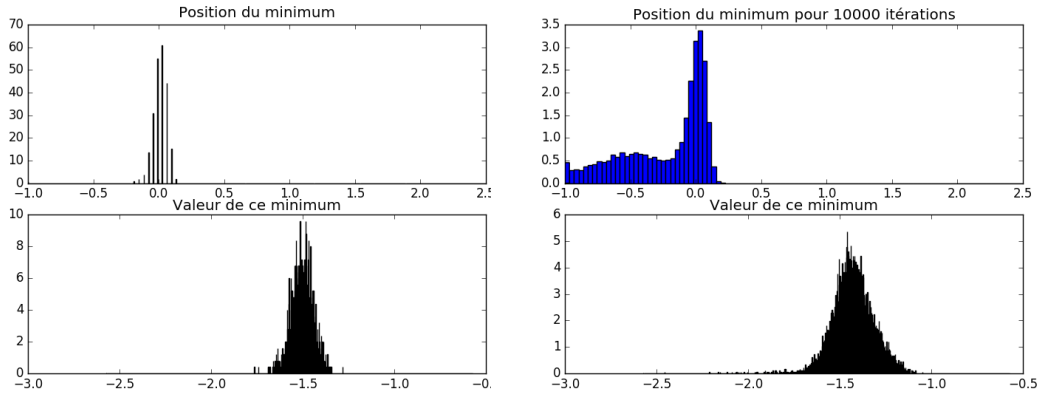
Remarque : Nous aurions pu directement prendre des réalisations de Y_{obs}, mais cela est en pratique coûteux.
Par cette méthode nous obtenons le tube de confiance à 95% en calculant expérimentalement la variance. En effet, nous utilisons que :

$$P(Y' \in [Y'_{mc} - 1,96\frac{\widehat{\sigma_{mc}}}{\sqrt{n}}, Y'_{mc} + 1,96\frac{\widehat{\sigma_{mc}}}{\sqrt{n}}]) \simeq 0.95$$

avec dans l'expression Y'_{mc} valant en x_j : $Y'_{mc,j} = \frac{1}{M}\sum_{i=1}^{M}\frac{Y_i[x_{j+1}]-Y_i[x_j]}{h}$.
Voici le tube de confiance obtenu sur Y', pour $n = P = 50$ et $M = 1000$:



Il est également possible de tracer des histogrammes sur la position et la valeur du minimum de Y'_{reel}. Pour cela, nous nous fixons $n = 100$ le nombre d'observations. Nous effectuons ensuite une méthode de différence finie sur des réalisations de $Y_{reel,post}$ obtenues via notre métamodèle. Grâce à la méthode de Monte Carlo effectuée $M = 500$ fois pour 5b nous obtenons la courbe de gauche. En réitérant le procédé sur 10000 jeux de données nous obtenons l'histogramme de droite 5b.



(a) Obtenue pour un seul jeu de 100 données (b) Obtenue pour 10000 jeux de 100 données

FIGURE 5: Réalisations de $Y'_{reel}(x)$ par Monte Carlo

L'histogramme de gauche 5b représente la position et la valeur du minimum de Y'_{reel} obtenu à partir d'un jeu de 100 observations, alors que l'histogramme de droite 5b représente les résultats obtenus sur 10000 jeux de 100 observations. Nous constatons la présence du minimum aux alentours de 0, pour une valeur correspondante d'environ $-\frac{\pi}{2} \simeq -1.5$. Cela concorde avec nos résultats sur :

$$Y'_{reel}(x) = -\frac{\pi}{2}\cos(\frac{\pi}{2}x)$$

Des histogrammes similaires étaient également réalisables pour Y_{reel}. A l'aide de la fonction f_{min} du module $scipy.optimize$ nous pouvons en déduire l'abscisse du minimum x_{min} ainsi que la valeur du minimum de cette fonction selon la loi à posteriori $ymin$. Fixons-nous $n_{total} = 5000$ points équiréparties dans l'intervalle $[-1, 2.5]$, nous sommes en mesure de tirer n_{total} réalisations de Y_{reel}. Pour construire les histogrammes 6, nous créons d'abord une liste dans laquelle nous conservons toutes les abscisses et les réalisations de Y_{reel} dont la valeur est inférieure à $ymin$. Nous obtenons le résultat suivant :

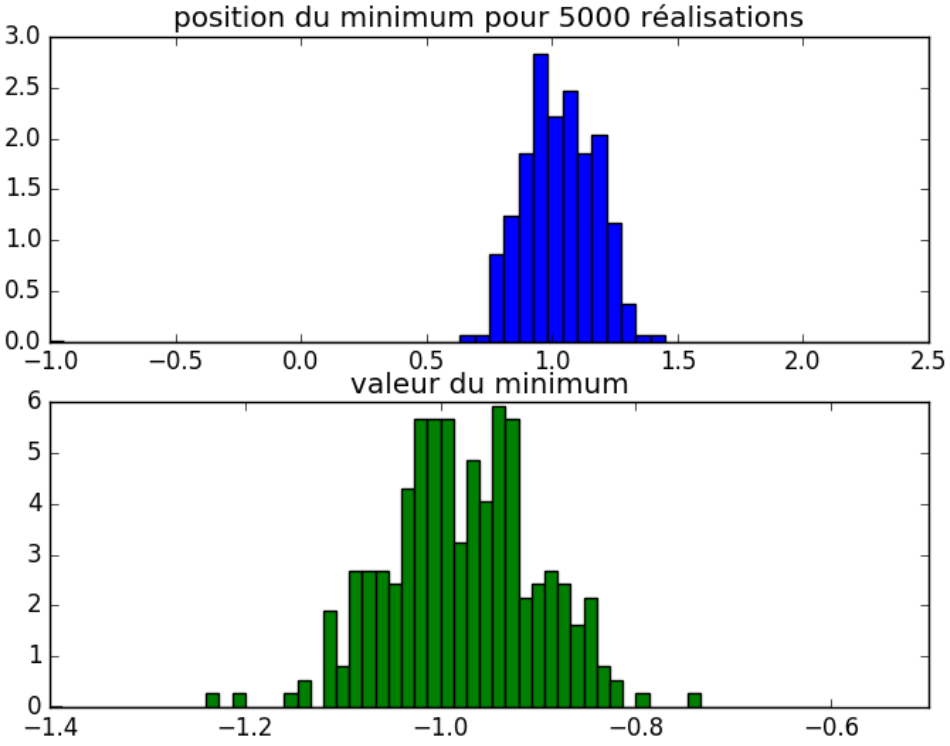


FIGURE 6: Histogrammes sur la position et la valeur du minimum de Y_{post}

Sur les histogrammes, nous constatons la présence du minimum de Y_{reel} aux alentours de $[0.8, 1.3]$ pour une valeur de ce minimum environ comprise dans l'intervalle $[-1.1, -0.9]$. Cela concorde avec la réalité puisque le minimum de Y_{reel} est atteint en 1 et celui-ci vaut -1.
Nous allons maintenant nous intéresser à l'étude du problème, mais cette fois-ci en considérant un métamodèle simplifié à p=2 paramètres, ce qui correpond à :

$$Y_{meta}(x) = \beta_1 + \beta_2 x$$

En reprenant l'étude précédente, nous trouvons désormais $\sigma_{mod} \simeq 0,66$. A parir de cette valeur nous pouvons de nouveau tracer des réalisations de Y_{reel} ainsi qu'un tube de confiance pour $Y_{reel}^{'}$:

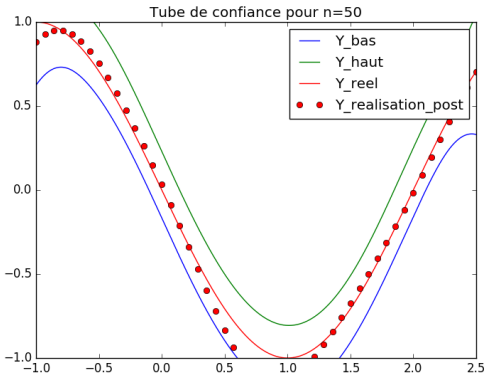


FIGURE 7: Tube de confiance et réalisations pour p=2

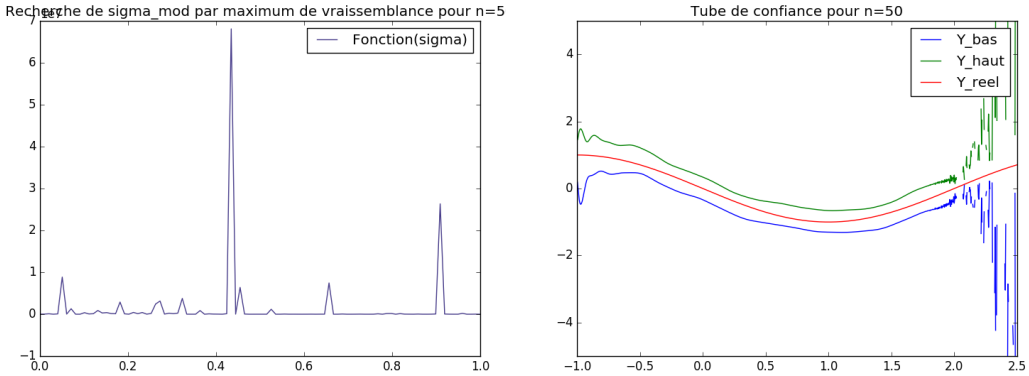
Cette fois-ci, le tube de confiance encadre fidèlement la fonction Y_{reel}, en effet, pour $n = 50$ observations, la fonction $\widehat{Y}_{post}$ fournit une bonne approximation de Y_{reel}.
A l'inverse, considérons un métamodèle plus sophistiqué en prenant $p = 20$:

$$Y_{meta}(x) = \beta_1 + \beta_2 x + \beta_3 x^2 + \ldots + \beta_{19} x^{18} + \beta_{20} x^{19}$$

Maintenant intéressons-nous à la recherche de σ_{mod}. En traçant la courbe 8a :

$\sigma_{mod} \mapsto \log det(R) + (y_{obs} - H\beta_{post}))^T R^{-1}(y_{obs} - H\beta_{post})$

Nous obtenons désormais $\sigma_{mod} \simeq 1,1$ mais la courbe tracée (cf 8a) à une allure complètement différente de celle obtenue pour $p = 4$ (cf 2). De plus, en traçant le tube de confiance pour Y_{reel} nous constatons de très fortes oscillations sur les bords de l'intervalle 8b.



(a) n=50 données (b) n=200 données

FIGURE 8: Réalisations de $Y_{reel}(x)$ selon la loi à postériori

Dans les cas de $n = 50$ observations, nous remarquons que le métamodèle pour $p = 2$ correspond davantage à la réalité que le métamodèle pour $p = 20$.
Ce phénomène illustre le *surapprentissage* (également appelé *overfitting*) qui apparaît lorsqu'on considère un modèle trop complexe compte-tenu du nombre d'informations disponibles.